AMATH 482/582: HOME WORK 1

REBECCA WANG

Amath Department, University of Washington, Seattle, WA
lufanw@uw.edu

Abstract. This report presents an analysis of acoustic data to determine the trajectory of a submarine moving through Puget Sound. Utilizing Fourier Analysis, the dominant frequency signature of the submarine was identified. A Gaussian filter was designed and applied to denoise the data, revealing the submarine's 2D and 3D movement paths. By comparing noisy and denoised data, we demonstrate the effectiveness of the filtering process in improving accuracy. This work highlights the value of signal processing in noisy environments, providing insights for future tracking systems.

1. Introduction and Overview

A submarine using a new, unknown technology is moving through Puget Sound, emitting acoustic frequencies that must be analyzed to determine its location. 3D acoustic pressure measurements were taken at 30-minute intervals over a 24-hour period in the area where the submarine is known to be. However, these measurements are noisy and require processing to extract meaningful information. By extracting the x and y coordinates of the submarine's path, this analysis can aid future tracking efforts.

This report focuses on analyzing the data to (1) determine the submarine's frequency signature using Fourier Analysis, (2) design a filter to denoise the data, and (3) reconstruct the 2D and 3D paths of the submarine.

2. Theoretical Background

2.1. Fourier Transform and Inverse Fourier Transform. The Fourier transform decomposes a signal into its constituent frequency components, offering insights into its periodic structure. The continuous 3D Fourier transform is defined as:

$$\hat{f}(k_x, k_y, k_z) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x, y, z)e^{-i(k_x x + k_y y + k_z z)}dx\,dy\,dz.$$

For computational purposes, the discrete Fourier transform (DFT) is used, which is implemented efficiently using the Fast Fourier Transform (FFT). For a one-dimensional discrete signal of length N, the DFT is mathematically represented as:

$$F(k) = \sum_{n=0}^{N-1} f(n)e^{-\frac{2\pi i}{N}kn},$$

where k refers to discrete frequency values. Extending this to three dimensions, for a grid of size $N \times N \times N$, the 3D DFT is given by:

$$F(k_x, k_y, k_z) = \sum_{x=0}^{N-1}\sum_{y=0}^{N-1}\sum_{z=0}^{N-1} f(x, y, z)e^{-\frac{2\pi i}{N}(k_x x + k_y y + k_z z)}.$$

Date: January 27, 2025.

The inverse Fourier transform reconstructs the original signal in the spatial domain from its frequency representation. The 3D discrete inverse Fourier transform (IDFT) is expressed as:

$$f(x,y,z) = \frac{1}{N^3} \sum_{k_x=0}^{N-1} \sum_{k_y=0}^{N-1} \sum_{k_z=0}^{N-1} F(k_x,k_y,k_z) e^{\frac{2\pi i}{N}(k_x x + k_y y + k_z z)}.$$

The inverse transform is efficiently executed using the inverse Fast Fourier Transform (IFFT). After applying a filter in the frequency domain, the IFFT is utilized to reconstruct a clean 3D spatial signal. This allows for the extraction of the submarine's position while effectively suppressing noise.

The Fast Fourier Transform (FFT) is an efficient algorithm for computing the discrete Fourier transform (DFT) and its inverse. Unlike the direct Fourier Transform, which has a computational complexity of $O(N^2)$, the FFT reduces this to $O(N \log N)$, making it well-suited for large datasets.

2.2. Denoise Data. The data contains noise, which must be filtered to extract the signal of interest. Assuming the noise is white Gaussian noise with a mean of zero, its effect on the Fourier series coefficients is uniform, as noted in the assignment: *"It is known that adding mean zero white noise to a signal (Gaussian noise) is equivalent to adding mean zero white noise (Gaussian noise) to its Fourier series coefficients."* This implies that the Fourier coefficients of the signal will have significantly larger amplitudes than those of the noise.

To denoise and extract signal features, the following methods are applied:

- **Filter application:** Remove noise by filtering out irrelevant Fourier coefficients.
- **Fourier coefficient averaging:** Average coefficients over all time samples to leverage the random, zero-mean nature of noise and further suppress its impact.

The averaging of the Fourier spectra for all time samples is mathematically expressed as:

$$\text{Average Spectrum: } \operatorname{avg}(F(k)) = \frac{1}{T} \sum_{t=1}^{T} F_t(k),$$

where $T = 49$ represents the number of time samples and k as wavenumbers

$$k = \frac{2\pi}{2L} \text{ for } k \in \{-\frac{n}{2}, \dots, \frac{n}{2} - 1\}$$

This approach effectively isolates the signal of interest by averaging out the noise contributions.

To identify the dominant frequency of the submarine's acoustic signal, the maximum coefficient in the averaged Fourier spectrum is located:

$$k_{\max} = \underset{k}{\operatorname{argmax}} \, |F(k)|.$$

This dominant frequency corresponds to the wavenumber $k_{\max}$ with the highest amplitude, representing the primary signal component amidst the noise.

2.3. Gaussian Filter. The Gaussian filter is a key tool in signal processing for isolating desired frequencies while suppressing noise. Its 3D mathematical formulation in the frequency domain is given by:

$$\mathcal{F}(k_x,k_y,k_z) = e^{-\tau\left((k_x-k_{x_0})^2+(k_y-k_{y_0})^2+(k_z-k_{z_0})^2\right)},$$

where:

- $\tau = \frac{1}{2\sigma^2}$, controlling the filter bandwidth,
- σ^2 denotes the variance,
- $(k_{x_0}, k_{y_0}, k_{z_0})$ is the center frequency of interest.

In this 3D implementation, terms inside the exponent can be adjusted to match the data's dimensionality. The filter's effect is visualized in 2D and 3D plots, demonstrating its ability to suppress noise while preserving key signal features.

3. Algorithm Implementation and Development

The implementation of this project leverages the following Python packages for efficient computation and visualization:

- **NumPy:** Used for numerical computations, including multidimensional array manipulations, FFT operations, and mathematical functions such as exponentials for constructing filters.
- **Matplotlib:** Facilitates 2D and 3D visualizations to analyze data slices, frequency spectra, and the submarine's reconstructed path.
- **MPL Toolkits:** The module is used to create interactive 3D plots for the frequency spectrum and spatial domain reconstructions.

3.1. Define Spatial and Frequency Domains.

3.1.1. Grid Initialization. Define a 3D spatial domain $[-10, 10]$ with 64 intervals using `np.linspace()` and generate mesh grids (X, Y, Z) with `np.meshgrid()`.

3.1.2. Frequency Modes. Compute wavenumbers and center them around zero using `np.fft.fftshift()` to form frequency grids (K_x, K_y, K_z).

3.2. Fourier Transform and Averaging.

3.2.1. Transform Data. Apply the 3D FFT (`np.fft.fftn()`) to convert spatial data into the frequency domain and `np.fft.fftshift()` to center the zero frequency. The inverse FFT (`np.fft.ifftn()`) reconstructs the denoised signal in the spatial domain.

3.2.2. Average Fourier Coefficients. To reduce noise, compute the average of Fourier coefficients over all time samples.

3.3. Dominant Frequency. The dominant frequency of the submarine's acoustic signal is identified by finding the maximum coefficient in the averaged Fourier spectrum. The corresponding wavenumbers $(k_{x_0}, k_{y_0}, k_{z_0})$ are extracted from the frequency grids (K_x, K_y, K_z) and represent the dominant frequency component in the 3D frequency domain, then further analyzed by visualizing:

- **2D Frequency Slice:** A slice of the frequency spectrum at $K_z = 10$ to highlight the intensity distribution in the K_x-K_y plane.
- **3D Visualization:** A 3D representation of the frequency spectrum, with the dominant frequency marked explicitly.

3.4. Apply Gaussian Filter.

3.4.1. Filter Construction. Construct a 3D Gaussian filter centered at $(k_{x_0}, k_{y_0}, k_{z_0})$ use $\tau = 0.2$ determines the bandwidth of the filter. Larger τ values introduced noise, so $\tau = 0.2$ was chosen for optimal noise suppression and signal preservation.

3.4.2. Filter Application. Multiply the Fourier-transformed signal by the Gaussian filter to reduce noise and isolate the frequencies of interest.

3.5. Inverse Fourier Transform for Denoising. Reconstruct the denoised signal by applying the inverse FFT as following and extract and visualize the middle slice ($z = 0$).

$$f(x, y, z) = |\mathcal{F}^{-1}(F(k))|.$$

3.6. Submarine Path Detection.

3.6.1. Path Using Denoised Data. Identify the maximum intensity in the denoised signal for each time step using `np.argmax()` and `np.unravel_index()` to construct the submarine's path.

3.6.2. Path Using Noisy Data. Repeat the procedure for the noisy data without applying the Gaussian filter to compare the effects of denoising, see the comparison.

3.7. Visualization.

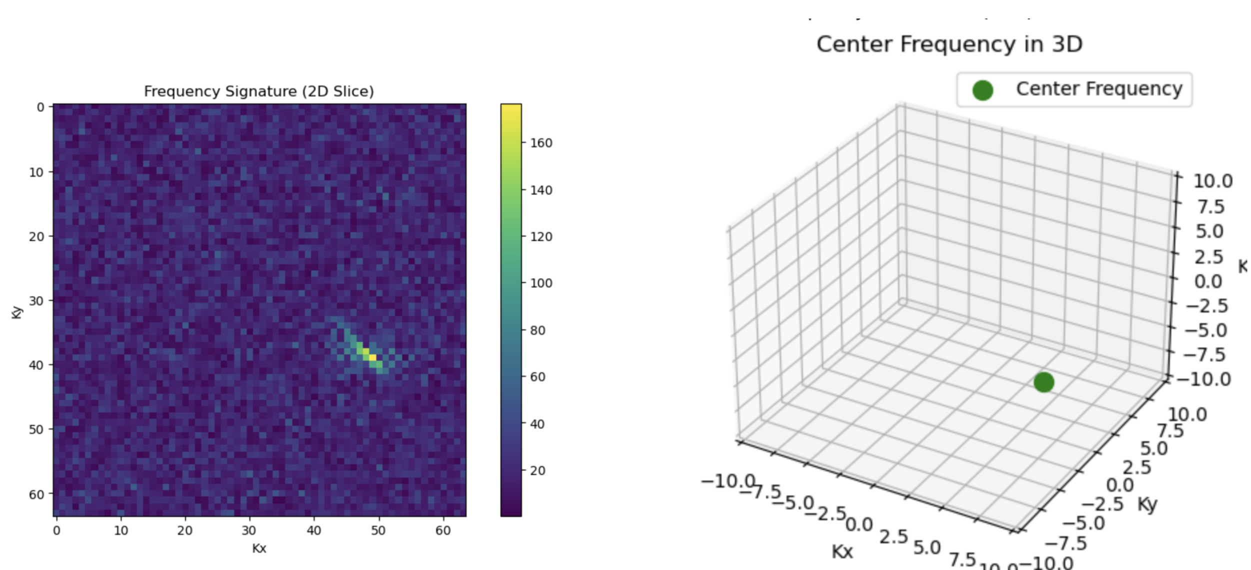
3.7.1. Noisy and Denoised Data Visualization. Plot the 2D and 3D visualizations of the noisy and denoised data to demonstrate the filter's effectiveness using `matplotlib.pyplot` and `mpl_toolkits.mplot3d.Axes` to demonstrate the filter's effectiveness.

3.7.2. Submarine Path Visualization. Plot the submarine's path in both noisy and denoised contexts in 3D, along with their 2D projections, using `matplotlib.pyplot` to compare accuracy and robustness.

4. Computational Results

4.1. Dominant Frequency Analysis. The averaged frequency spectrum over all time steps reveals the dominant frequency component corresponding to the submarine's acoustic signal. The center frequency identified in the spectrum is $(5.34070751, 2.19911486, -6.91150384)$, which indicates the spatial periodicity of the signal. This is accompanied by its corresponding grid location, referred to as the center frequency index, $(39, 49, 10)$.

Figure 1a illustrates a 2D slice of the frequency spectrum at $K_z = 10$, highlighting the frequency intensity distribution in the $K_x - K_y$ plane. Figure 1b shows the 3D representation of the entire frequency spectrum, with the center frequency explicitly marked.



(A) 2D Slice of the Frequency Spectrum at $K_z = 10$.

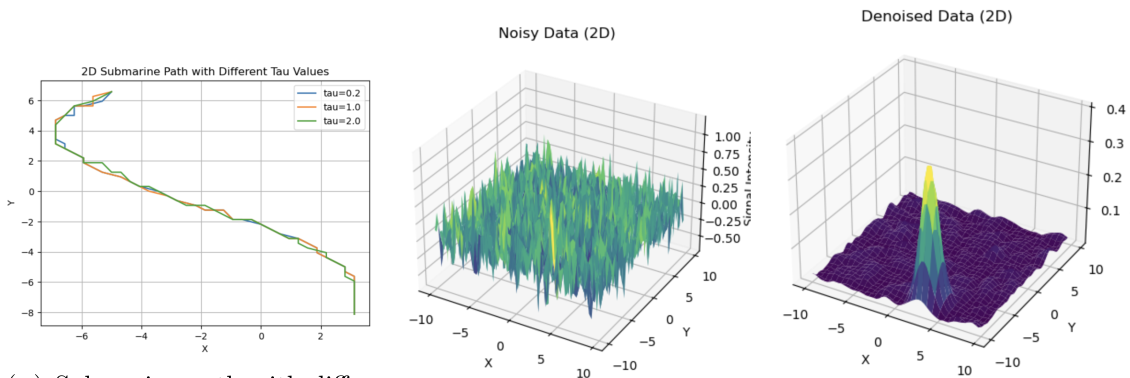
(B) 3D Visualization of the Frequency Spectrum

FIGURE 1. Dominant Frequency Component Analysis

4.2. Gaussian Filter Effectiveness. To determine the best diameter for the Gaussian function, the path of the submarine was analyzed using several different values of τ, as shown in Figure 2a. $\tau = 0.2$ was chosen because it results in the smoothest path of the submarine.

The impact of the Gaussian filter on the data is evident when comparing the noisy and denoised signal plots. The denoised measurement of the 3D path of the submarine in Section 4.3 also demonstrates the filter's denoising validity.

Figure 2b illustrates the noisy 2D signal slice extracted from the original data. Noise dominates the spatial domain, masking the underlying signal of interest. The intensity values are distributed randomly, with no visible structure of the submarine's position. In contrast, Figure 2c shows the same 2D slice after applying the Gaussian filter and reconstructing the signal. The denoised signal demonstrates a clear peak, corresponding to the submarine's acoustic signal. Noise has been effectively reduced, allowing the spatial coordinates of the submarine to be identified with high accuracy.



(A) Submarine path with different τ values

(B) Noisy data

(C) Denoised data

FIGURE 2. Effectiveness of the Gaussian Filter

4.3. Submarine Path Detection. The submarine's path was successfully determined from both the noisy and denoised data. Figures 3 and 4 present the comparisons between noisy and denoised data in 3D and 2D projections, respectively. These visualizations illustrate the effectiveness of the Gaussian filter in improving the accuracy of the submarine's trajectory detection. We find the final position of the submarine determined from the denoised data is $(x, y, z) = (-5.0, 6.5625, 0.9375)$.

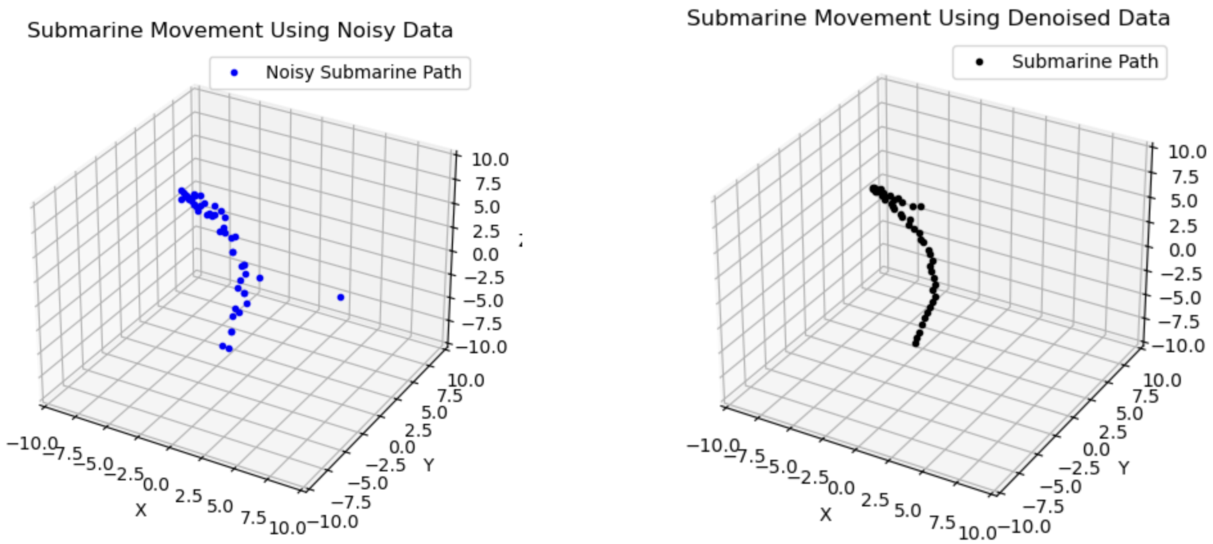
TABLE 1. Submarine path coordinates (denoised data).

Time Step	x	y	z
1	3.125	-8.125	0.0000
2	3.125	-7.8125	0.3125
⋮	⋮	⋮	⋮
49	-5.0000	6.5625	0.9375

5. Summary and Conclusions

In this study, we processed 3D noisy acoustic data collected over 24 hours to identify the path of a submarine. The dominant frequency was identified as $(5.341, 2.199, -6.912)$ by Fourier analysis. The denoised data after the Gaussian filter revealed a clear trajectory, culminating in a final submarine position of $(-5.0, 6.5625, 0.9375)$. Our findings highlight the critical role of signal processing techniques in extracting meaningful patterns from noisy datasets.

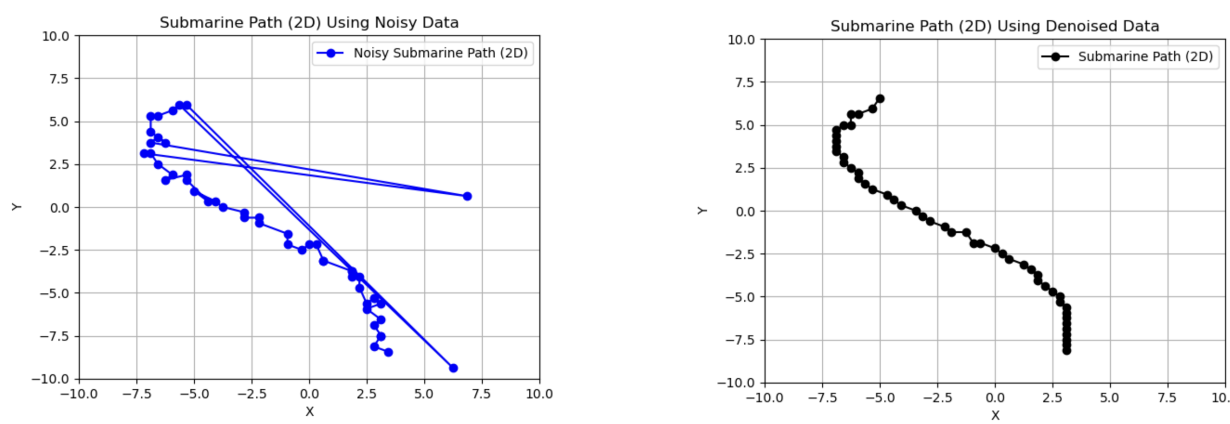
Future work could involve real-time implementation of this method for continuous tracking, as well as exploring advanced machine learning techniques to improve noise filtering and detection efficiency.



(a) Submarine Movement Using Noisy Data (3D)

(b) Submarine Movement Using Denoised Data (3D)

FIGURE 3. Comparison of Submarine Movement in 3D using Noisy and Denoised Data



(a) Submarine Path (2D) Using Noisy Data

(b) Submarine Path (2D) Using Denoised Data

FIGURE 4. Comparison of Submarine Movement in 2D using Noisy and Denoised Data

Acknowledgements

This author thanks Professor Frank for providing useful information to tackle the problem through the hints in the assignment, as well as the HW1-Helper.ipynb code. The author is also grateful to TA Rohin Gilman for providing theoretical background to assist in constructing my code. Finally, my peers Zhaohan Yin were also helpful when I ran into issues related to coding.

References

[1] J.N. Kutz, *Methods for Integrating Dynamics of Complex Systems and Big Data*, Oxford, 2013.
[2] N. Frank, *Lecture Notes*, January 2025.
[3] N. Frank, *Homework 1-TA Slides*, January 2025.