# AMATH 482/582: HOME WORK 1

SARAYU GUNDLAPALLI

*Applied Mathematics, University of Washington, Seattle, WA*
*saru07g@uw.edu*

Abstract. The trajectory of a submarine in Puget Sound is located using noisy acoustic data. Through Fourier analysis and Filtering, it was possible to locate the spatial frequency and track the location of the submarine in 3D space. By comparing the results before and after filtering, an improvement in measurement precision is observed. This information can be used to deploy a sub-tracking aircraft to keep an eye on your submarine in the future.

## 1. Introduction and Overview

Given noisy acoustic pressure data of unknown orientation of a submarine moving in time, we aim to detect its location and path. The broad spectrum recording of acoustics data was obtained over 24 hours in half-hour increments. The goal is to de-noise this data and find the frequency signature of the set of signals, and subsequently the trajectory of the submarine. Several concepts are implemented to achieve this goal, including the FFT algorithm, averaging the frequency of the signals, and Gaussian filtering. These are described in detail in the next section, followed by a discussion of the algorithm used and a compilation of the computational results.

Object detection is an active field of research with applications in many fields [2]. Underwater, one of the signal propagation parameters to be observed is ambient noise. Ambient noise causes the channel condition to become non-ideal and thus requires further research [1].

The method used here is a simple example of what Fourier transform allows us to do.

## 2. Theoretical Background

### 2.1. The Fourier Transform.

The Fourier Transform can be used to get spatial frequency information about the acoustic signal. The Fourier transform decomposes a signal in time into its frequency components. It is defined over the entire line, i.e., $x \in [-1, 1]$, and is an integral transform. The FT is given as:

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

and its inverse is given as:

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

*Date*: January 25, 2024.

### 2.2. **The Fast Fourier Transform.**

In this problem, we apply the Fast Fourier transform (FFT) to perform Fourier transform and inverse Fourier transform. It is a widely used algorithm, developed by Cooley and Tukey, to compute the DFT (discrete Fourier transform) because of its low operation count O(NlogN) and accuracy [5]. In python, we use functions np.fft.fft(x), np.fft.ifftn(x) and np.fft.fftn(x) to perform forward, backward and multidimensional FFT.

### 2.3. **Time Averaging and Gaussian Filtering.**

Time averaging the frequency spectrum of a signal over several time steps is an effective way to white noise from the signal. The idea behind this technique is that noise over a signal should add up to zero on average. This can be used to filter signals where the frequency is unknown but constant, as we see in this report. Once the frequency signature is isolated by averaging, a Gaussian filter can be designed around this frequency and applied to all the signals used in time averaging.

Gaussian filtering is a commonly used technique used to filter data where the frequency is constant over time. It can filter out the required frequency by removing most of the white noise. As mentioned, a 3D Gaussian filter centered around the frequency signature is constructed. It is defined as:

$$G(\kappa_x, \kappa_y, \kappa_z) = \exp(-\tau((\kappa_x - \kappa_x^*)^2 + (\kappa_y - \kappa_y^*)^2 + (\kappa_z - \kappa_z^*)^2)) \tag{3}$$

where

$$\tau = \frac{1}{2\sigma^2} \tag{4}$$

The Gaussian function acts like a filter that passes frequencies near its center but suppresses frequencies away from the center. Multiplying it with the signal spectrum keeps the frequencies near the Gaussian peak intact while attenuating other frequencies. This effectively isolates just the frequencies of interest around the Gaussian center.

## 3. **Algorithm Implementation and Development**

This section lists the algorithms used to find the frequency signature, attenuate noise in the data and find the trajectory of the Submarine.

- Load **subdata.npy**
- The spatial domain and frequency domain are defined.
  - x, y and $z \in [-L, L]$ are partitioned into n discrete modes in the spatial domain. The frequency domain is multiplied by $\frac{2\pi}{L}$ to rescale as FFT requires $2\pi$ periodic domain.
- **meshgrid** is used to create a cartesian grid with the X, Y, Z coordinate system and Kx, Ky, Kz frequency scheme.
- For averaging, the data is reshaped into three dimensions before multi-dimensional FFT is performed using *np.fft.fftn()* to add up the frequency. The gathered frequencies are rescaled and the absolutes of the cumulative values are taken (as the data may be in the form of complex values). Next, we look at the max signal and use *np.unravel index*
- The Gaussian filter is designed according to the equation (3). The filter is applied by multiplying each signal in the Fourier space by the Gaussian filter. Inverse FFT is performed using *np.fft.ifftn()*, the data is shifted using *np.fft.fftshift()* and the absolute value is taken.

- The path of the submarine is determined by using the coordinates of the maximum signal for each realization.

Python packages NumPy[3], Scipy[7], Pandas[6], matplotlib[4] are used.

## 4. Computational Results

The coordinates of the frequency signature were detected 1. The Gaussian function was modeled around this frequency signature using equation (3) and (4).

| | Center Frequency | Index of Max: |
|---|---|---|
| 0 | 2.199115 | 39 |
| 1 | 5.340708 | 49 |
| 2 | -6.911504 | 10 |

Table 1. The frequency signature (center frequency) generated by the submarine.

The $\sigma$ in the Gaussian function acts to control variation around the mean value. Our definition of the Gaussian function allows the $\sigma$ to be changeable. When $\sigma$ is too small, the filter doesn't allow for the data to be de-noised sufficiently2a. If $\sigma$ is chosen to be large, the kernel size is larger, so there is more "blurring"2c. In this study, $\sigma$ is chosen to be 2, which allows for a sufficiently smooth submarine trajectory1.

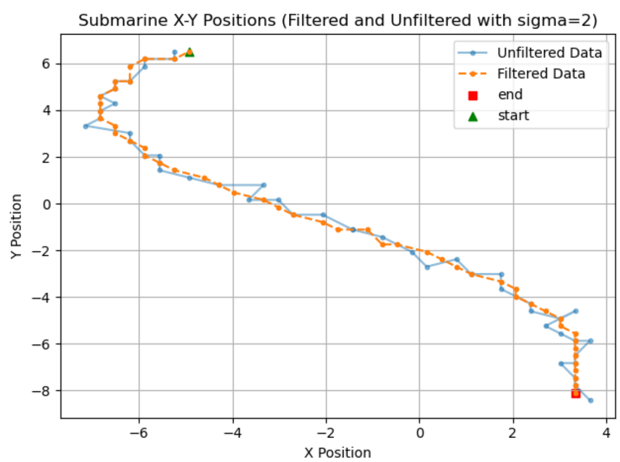


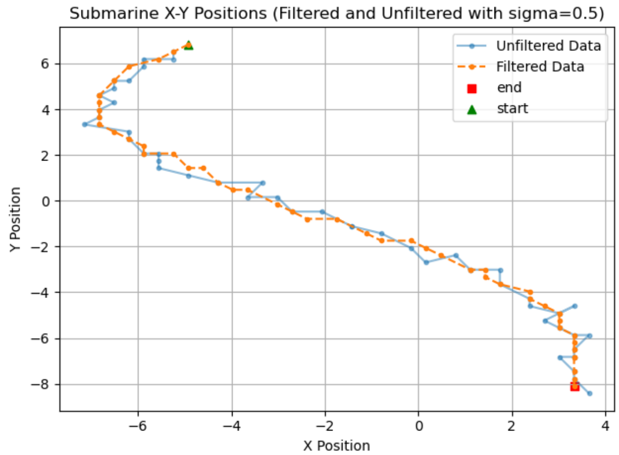

Figure 1. Submarine trajectory along x-y (unfiltered vs filtered at $\sigma = 2$)

The submarine's location was determined at each time point over the 24-hr period and the trajectory is mapped out in the figures. The figures compare the path the submarine takes over time mapped with the original (unfiltered data) vs filtered data1. By plotting the (x, y) points of the submarine's path, we get a clear picture of the trajectory w.r.t time and any trends/patterns. This information can be used to deploy a sub-tracking aircraft to keep an eye on the submarine in the future.

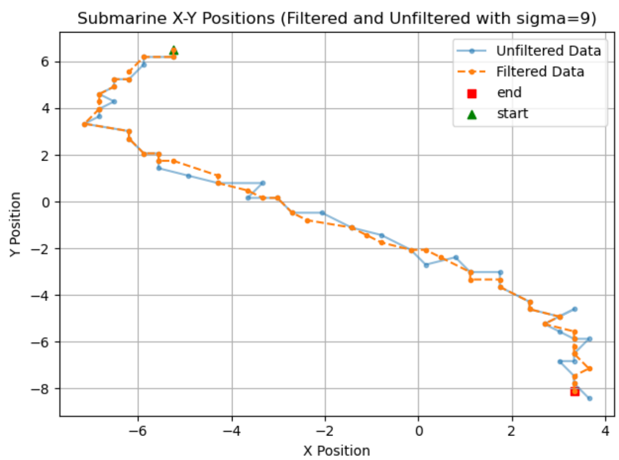
## 5. Summary and Conclusions

Gaussian filtering around the time-averaged frequency signature successfully de-noised the 3D data by isolating the key frequencies. When the choice of $\sigma$ was varied, the effects on the filtering were observed. The trajectory of the submarine was seen to be a curved path.

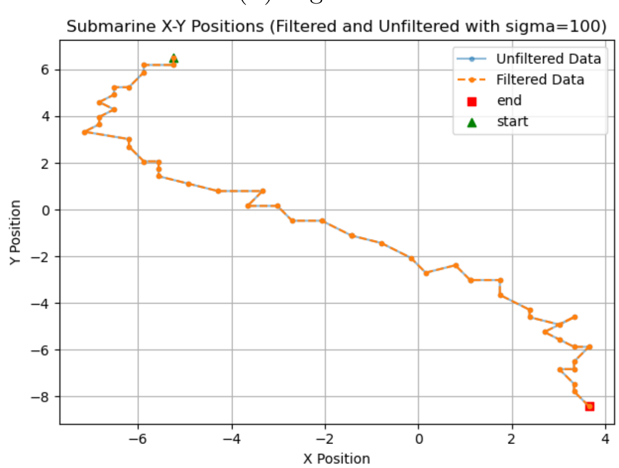
The future direction of this study could be to compare the efficiency of different filters applied to noisy acoustic data or extend the study to find an optimal $\sigma$ value.




(A) Figure A




(B) Figure B




(C) Figure C

FIGURE 2. Submarine trajectory along x-y (unfiltered vs filtered at $\sigma = 0.5$, 9 and 100 respectively)

## Acknowledgements

The author is thankful for the suggestions/advice about the algorithm implementation from peers in AMATH 482/582. The author is thankful to the Teaching Assistants, Saba Heravi and Juan Felipe Osorio Ramírez, for their invaluable time and fruitful discussions about the implementation of the Gaussian filter. Further, the author is thankful to Prof. Eli Shlizerman for useful discussions about the course content.

## References

[1] S. I. Adzhani, H. Mahmudah, and T. B. Santoso. Time-frequency analysis of underwater ambient noise of mangrove estuary. In *2016 International Electronics Symposium (IES)*, pages 223–227, 2016.

[2] D. Gillespie, L. Palmer, J. Macaulay, C. Sparling, and G. Hastie. Passive acoustic methods for tracking the 3d movements of small cetaceans around marine structures. *Plos One*, 15(5):e0229058, 2020.

[3] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, Sept. 2020.

[4] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[5] J. Kutz. *Methods for Integrating Dynamics of Complex Systems and Big Data*. Oxford, 2013.

[6] W. McKinney et al. Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, volume 445, pages 51–56. Austin, TX, 2010.

[7] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.