AMATH 582: HOMEWORK 1

CASSIA CAI

School of Oceanography, University of Washington, Seattle, WA
fmc2855@uw.edu

Abstract. The three-dimensional underwater movements of an object that vibrates with a frequency signature can be tracked using acoustic pressure measurements. Vibrations result in mechanical disturbances that propagate through the fluid, increasing or decreasing pressure with time. Acoustic pressure data can be noisy. Using a noisy dummy three-dimensional acoustic pressure data set where a submarine is moving, we demonstrate that the submarine's trajectory can be tracked using Fourier transform to extract the submarine frequency signature, implementing a filter that extracts this frequency signature to denoise the data, and detecting the submarine over time. In this data set, central frequencies are (-4.712, -1.885, 7.226) and (4.712, 1.885, -7.226), and the filter used is Gaussian. This method has a wide range of applicability in biological oceanography and fisheries research to track marine animal movement.

1. Introduction

Given noisy three-dimensional (3D) dynamic acoustic pressure data of unknown orientation where a submarine is moving in time, we aim to detect its location and path. The data was obtained over 24 hours in half-hour increments, and the noise is random and mean zero. The vibrations of a submarine is generated by its propeller, engine, and pump among other parts, making up the submarine's acoustic signature. This submarine emits at an unknown frequency, which can be determined by transforming the data into the frequency domain. Using a filter to extract this signature frequency and thereby denoising the data, the submarine path can be detected. We demonstrate that we can indeed detect extract this signature frequency, apply a filter to denoise the data, and detect the submarine trajectory.

Object detection is an active field of research with applications in many other fields [1, 2]. Given the complexity of the marine environment, acoustic pressure data are often interrupted by noise [3]. The goal of many tracking algorithms is to obtain better tracking performance in a noisy environment. The method used here is a simple example of what Fourier transform allows us to do, which is to identify an object within a noisy data set. This method can be extended and combined with statistical methods to address more complex problems [4, 5].

2. Theoretical Background

The Fourier transformation (FT), which is the continuous version of the Fourier Series (FS), is used to represent a signal as a sum of sine and cosine functions [6]. The FT is defined as Given

$$F(f) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \exp(-ikx) f(x) dx$$

and its inverse:

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

Date: January 27 2022.

A Discrete Fourier Transform (DFT) is the FT applied to a discrete signal.

$$f(x) \approx \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} c_k \exp(-\frac{ikRx}{L}), c_k \in \mathbb{C} \tag{1}$$

We can write the DFT of f(x) as: $\hat{f} := c_{-\frac{N}{2}}, ..., c_{\frac{N}{2}-1}) \in \mathbb{C}^N$. The time complexity to compute DFT is O(N^2).

The Fast Fourier Transform (FFT) is an algorithm that computes the DFT quickly, converting a signal from a time or space domain into the frequency domain [7]. The FFT computes $\hat{f}$ for a function f(x) given a uniform mesh.

The FFT computes the DFT quickly because it costs O(NlogN) operations to compute if N is the number samples in signal. It does this by discretizing the range $x \in [L, L]$ into 2^n points.

For a function $f : [0, 2L] \to \mathbb{R}$ where $f_n := f(x_n)$ for n=0,...,N-1. FFT returns coefficients:

$$\tilde{c}_K = \sum_{n=0}^{N-1} f_n \exp(-2\pi i \frac{nk}{N})$$

FS returns:

$$\tilde{c}_k = \sum_{n=0}^{N-1} f_n \exp(-2\pi i \frac{kn}{N})$$

$$c_k = \frac{1}{2L} \int_0^{2L} f(x) \exp(-i\pi \frac{kn}{L}) dx$$

Approximating with a Riemann sum and breaking [0,2L] into N intervals of length $\delta_x = \frac{2L}{N}$:

$$c_k \approx \frac{1}{2L} \sum_{n=0}^{N-1} f(x_n) \exp(-i\pi \frac{k_n}{L}) \delta x = \frac{1}{N} \sum_{n=0}^{N-1} f(x_n) \exp(-i\pi \frac{k_n}{N}) = \frac{1}{N} \tilde{c}$$

To approximate FS, we scale the output of FFT by $\frac{1}{N}$. With discrete transforms, FFT does not need to be rescaled.

In summary, FS is used to represent a periodic function by a discrete sum of complex exponentials. FT is a generalized version of FS, where a non-periodic function can be represented by complex exponentials. DFT is the FT applied to a discrete signal. Finally, FFT is a fast algorithm to compute the DFT.

MEAN-ZERO NOISE: Adding random noise to a signal is is equivalent to adding mean zero noise to the signal's FS coefficients [8, 9]. To illustrate this, let ξ be mean-zero noise:

$$f_n + \xi_n$$

$$\widehat{f + \xi} \stackrel{d}{=} \widehat{f} + \xi$$

$$\widehat{(f + \xi)}^k \stackrel{d}{=} \widehat{f_n}^k + \xi^k$$

$$\frac{1}{k} \sum_{k=0}^{k-1} \widehat{(f + \xi)}^k \stackrel{d}{=} \frac{1}{k} \sum_{k=0}^{k-1} \widehat{f}^k + \frac{1}{k} \sum_{k=0}^{k-1} \xi^k$$

While unable to extract the signal in the time- or space-domain, we can do the extraction in the Fourier domain.

3D GAUSSIAN FILTER: The 3D Gaussian filter is defined as:

$$G(x, y, z) = e^{\frac{-((x-x_{shift})^2 + (y-y_{shift})^2 + (z-z_{shift})^2)}{2\sigma^2}}$$

Because there are two central frequencies, the final Gaussian filter is the two centered Gaussian filters added together. The shifted coordinates allows allows the Gaussian to be centered at the central frequency.

3. Algorithm Implementation and Development

The provided noisy acoustic pressure data is a flattened 3D matrix of size 64x64x64x49. To effectively denoise the data, we apply a filter centered at the center frequency. In the pseudo-code, a Gaussian filter is applied. However, other filters can be used such as the hard spherical filter and filters found in Scipy's multi-dimensional image processing package among others. The steps undertaken to denoise data and detect the submarine's location are outlined below in the pseudo-code. The pseudo-code is purposely minimal. I coded Python and used Numpy for mathematical calculations and Matplotlib for visualizations [10, 11].

Pseudo-code: Apply Gaussian filter to denoise data and detect object location

---

Step 1: Find the frequency signature of the submarine

- a. Reshape data
- b. Apply FFT and FFT shift
- c. Take the time-average of the frequency signals
- d. Find the indices of the peak in the frequency space
- e. Find the corresponding coordinates from the 3D frequency domain

Step 2: Build Gaussian filter centered at the center frequency

- a. Build two 3D Gaussian filters by centering each around the two peaks by shifting
- b. Add the 2 Gaussian filters to create the final Gaussian filter

Step 3: Apply filter to each time-step to denoise the data

- a. Multiply the filter by the data in the frequency space
- b. Apply inverse FFT and inverse FFT shift to generate the denoised data

Step 4: Find the submarine location

- a. Find the indices of location with the strongest signal at each time step

---

Signals in the raw data are evidently noisy (Figure 1). Applying FFT and FFT shift (for visualization purposes) transforms this data to the frequency domain, where the signals are significantly clearer. NumPy's fftn, fftshift, ifftn, and ifftshift function are used. Numpy's fftn function computes the N-dimensional discrete Fourier Transform. Similarly, ifftn computes the N-dimensional inverse discrete Fourier transform. fftshift and ifftshift both shift to the center of the spectrum. By time-averaging the Fourier transform and through visual inspection, we can detect the index coordinates of the two peaks (Figs. 1b and 1d), where one peak is the reflection of the other across the central line of our space. However, a more robust method to find the peak coordinates is to exploit NumPy.

Our definition of the 3D Gaussian filter allows the σ to be changeable. σ acts to control variation around the mean value. When σ is large (Figure 2), the kernel function is larger, effectively meaning a larger smoothing filter radius (Figure 3). Alternatively, if σ is small, the blurring occurs across a smaller radius. However, then σ is chosen to be too small, the filter does not sufficiently allow the the data to be denoised, leading to a submarine trajectory that is similar to the predicted trajectory from the raw signal data. In this study, σ is chosen to be 3, which allows a reasonably smooth submarine trajectory.

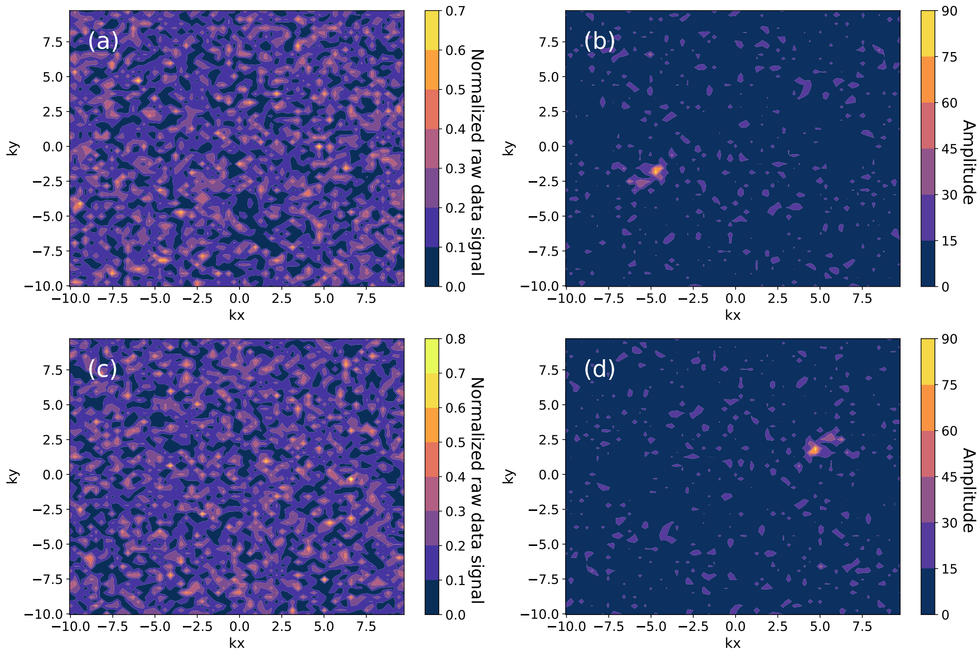

FIGURE 1. **(a)** Normalized data at z-level $= 55$ **(b)** Signal in frequency domain at z-level $= 55$ **(c)** Same as (a) but at z-level $= 9$ **(d)** Same as (b) but at z-level $= 9$

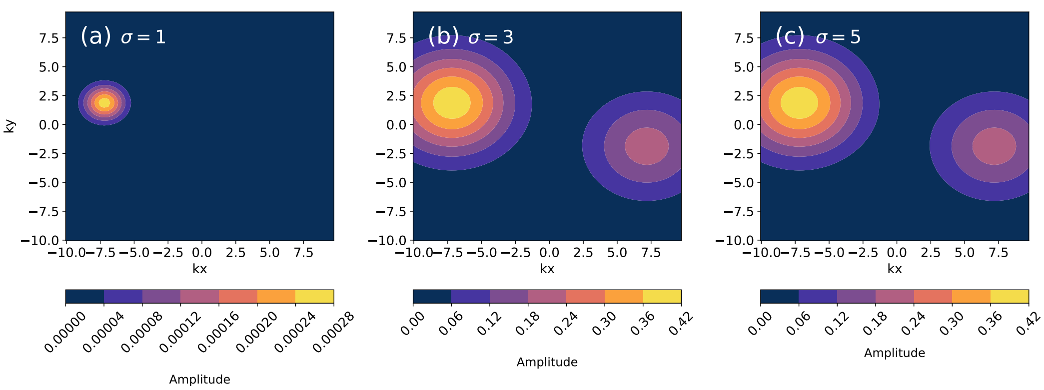

FIGURE 2. **(a)** 2D slice of time-averaged Gaussian filter at z-level$= 34$ with $\sigma = 1$ **(b)** Same as (a) but for $\sigma = 3$ **(c)** Same as (a) but for $\sigma = 5$

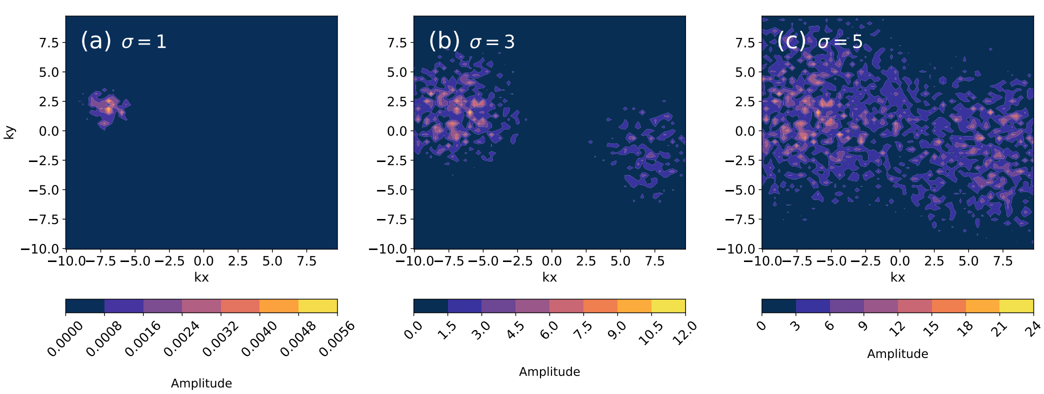

FIGURE 3. **(a)** 2D slice of time-averaged filtered fft at z-level $= 34$ with $\sigma = 1$ **(b)** Same as (a) but for $\sigma = 3$ **(c)** Same as (a) but for $\sigma = 5$

4. Computational Results

The coordinates of the peak frequency of the submarine were detected (Table 1). Visually, peak locations can also be approximated from Figure 1. Because of the nature of the Fourier transform, there are two peaks, one of which is the reflection of the other across the central line in our data space. It is important to consider both peaks when constructing the filter.

coordinate	center frequency 1	center frequency 2
kx	-4.712	4.712
ky	-1.885	1.885
kz	7.226	-7.226

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

The x,y, coordinates of the submarine during the 24-hour period and the 3D path of the submarine are shown in Figure 4. The submarine follows a curved path. It is difficult to make any conclusions the orientation of this path since no information was provided about the orientation of the data space. Two other filters were also implemented (the algorithms used to design and implement them are not discussed here) (Figure 5), one of which is a hard spherical filter which allows a radius to be defined and does not apply any blurring and the other is the Scipy percentile filter. The submarine trajectories are comparable, with the exception of the hard spherical filter.

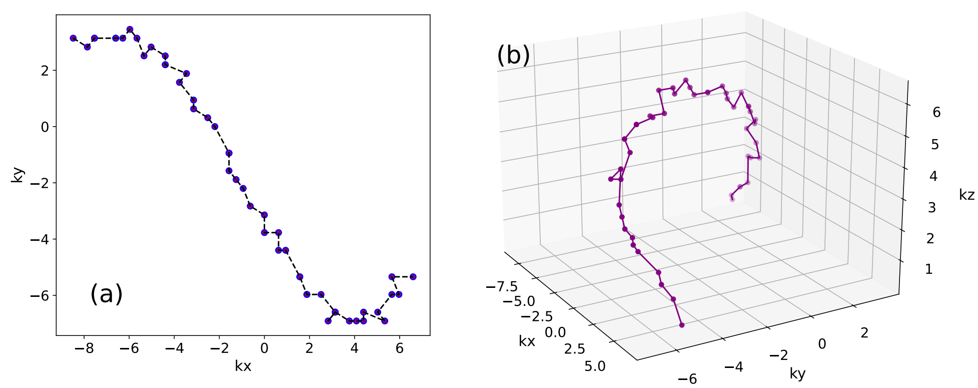

Figure 4. **(a)** 24-hour submarine trajectory in (x,y) **(b)** Same as (a) but showing the 3D submarine

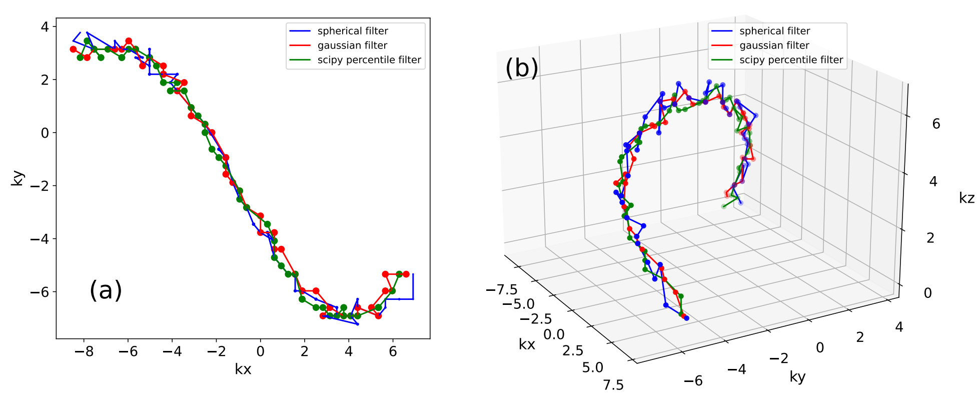

Figure 5. **(a)** Same as Fig. 4a **(b)** Same as Fig. 4b

5. Summary and Conclusions

Given noisy 3D dynamic acoustic pressure data,the trajectory of a submarine by implementing Fourier analysis. First, the signal data was converted to the fourier domain via FFT. Then, the central frequency was found by time-averaging the FFT-data. This central frequency was then used to localize a Gaussian filter. The data was converted back to the physical domain via inverse FFT, and the 3D submarine trajectory was found by extracting the coordinates of the maximum pressure at each time step. The submarine follows a curved path.

Future directions of this study could be to apply and compare the efficiency of different filtering methods. Another direction is to extend the Gaussian filter by developing a robust way to determine the optimal σ.

Acknowledgements

The author is thankful for the fruitful conversations with and algorithm-implementation suggestions from peers in AMATH 482-582. The author is thankful to Prof. Bamdad Hosseini for providing the problem set-up and data as well as AMATH 482-582 Teaching Assistant Katherine Owens for productive discussions about creating the filter and denoising algorithm.

References

[1] Douglas Gillespie, Laura Palmer, Jamie Macaulay, Carol Sparling, and Gordon Hastie. Passive acoustic methods for tracking the 3d movements of small cetaceans around marine structures. *PLOS ONE*, 15(5), 2020.

[2] K. Lopatka, J. Kotus, and A. Czyzewski. Detection, classification and localization of acoustic events in the presence of background noise for acoustic surveillance of hazardous situations. *Multimedia Tools and Applications*, 75(17):10407–10439, 2015.

[3] David L. Bradley and Richard Stern. *Underwater Sound and the Marine Mammal Acoustic Environment: A Guide to Fundamental Principles*. U.S. Marine Mammal Commission, 2008.

[4] Mausumi Pohit. Object recognition in presence of blur from fourier transform spectrum. *2009 Seventh International Conference on Advances in Pattern Recognition*, 2009.

[5] D. Toth and T. Aach. Detection and recognition of moving objects using statistical motion detection and fourier descriptors. *12th International Conference on Image Analysis and Processing, 2003.Proceedings.*

[6] Steve Brunton and Nathan Kutz. *Data Driven Science Engineering*. databook.uw.edu, 2017.

[7] Bamdad Hosseini. Signal processing with dft. University of Washington (LOW 216), Jan 2022. AMATH 482/582.

[8] E. Oran Brigham. *The Fast Fourier Transform and Its Applications*. Prentice Hall, Inc., 1988.

[9] Bamdad Hosseini. High-dimensional extensions amp; image processing. University of Washington-Seattle (LOW 216), Jan 2022. AMATH 482/582.

[10] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

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