SIGNAL PROCESSING AND FINDING SUBMARINE

SAMIP KARKI

Applied Mathematics, University of Washington, Seattle, WA
karki1@uw.edu

Abstract. Fourier transform is a powerful tool used in signal processing. In this work, I demonstrate how a noisy dataset can be cleaned up by accessing the k-space and filtering out unwanted frequencies. With these techniques, I am able to find the exact trajectory of a submarine hidden in the Puget Sound area as it moves in a 24 hour time interval. I compare two methods, one which uses Fourier transforms, and another without using Fourier transforms and see which produces a clearer trajectory of the submarine.

1. Introduction and Overview

In this report, I explored signal processing techniques involving fast Fourier transform to denoise 3D time series data. The problem is as follows. I have obtained a 262144 by 49 data array which corresponds to acoustical measurements of the Puget Sound area over a 24 hour interval. Measurements where made every 30 minutes, corresponding to the 49 columns of the array. Each column has 262144 entries which is the flattened version of a 64 by 64 by 64 array of acoustical amplitudes in a cube with side length 20. Hidden is in this noisy data is a moving submarine. A preliminary plotting of the unfiltered data can be seen in figure 1) and a projection of the entire dataset onto the $x-y$ plane can be seen on the left in figure 2). As it can be seen, noise in the data obstructs the exact location of the submarine. In this report, I use signal processing techniques to filter the original data and locate the submarine at every point in time.

Date: August 30, 2024.

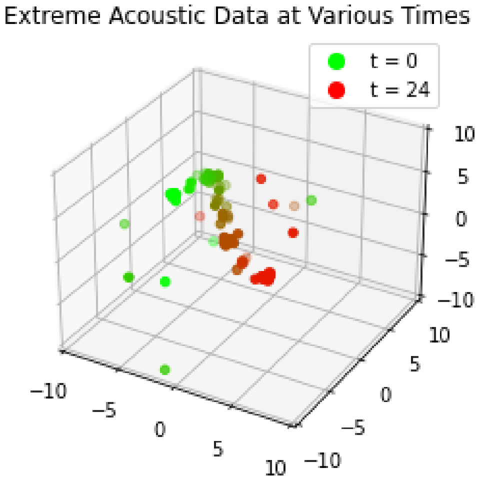

Figure 1. Plot of the largest absolute amplitudes of the original submarine data for all measurements.

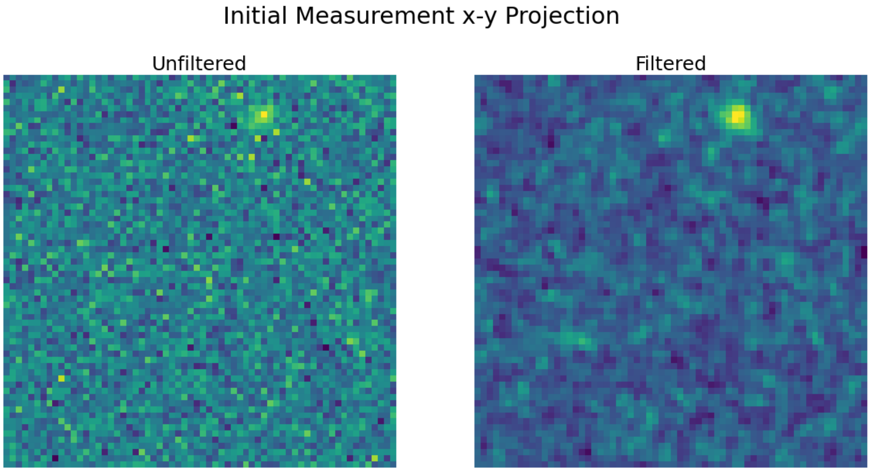

FIGURE 2. Projection onto x-y plane of unfiltered data (**left**) vs filtered data (**right**). These plots were created by taking the 64 × 64 × 64 array of spatial acoustical measurements at the first time-step and averaging across the z-coordinate.

2. Theoretical Background

This project uses Fourier transform and inverse Fourier transform extensively. Fourier transform and its inverse are given by the following:

$$F(k) = \int_{-\infty}^{\infty} f(x)\, e^{-ikt}\, dx$$

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(k)\, e^{ikt}\, dk$$

The Fourier transform of a some signal, $F(k)$ at a frequency k represents the amplitude of a sinusoid with the frequency k in the original signal. In this way, we can say the the Fourier transform takes us from position-space (x-space) to frequency-space (k-space). In many different applications, it can be insightful to analyze a signal in k-space rather than x-space. In our case, as will be discussed further in this report, by filtering out noisy frequencies in k-space, we denoise the original signal in x-space.

The Fourier transform can be generalized to three dimensions. In this project, I use the `numpy` implementation of the fast Fourier transform algorithm, `numpy.fft.fft()`, or in 3D, `numpy.fft.fftn()`

3. Algorithm Implementation and Development

I was able to find the submarine's trajectory in two ways, which I will call method-A and method-B, repectfully. Method-A used extensive use of `numpy.fft.fftn()`, while method-B did not.

Method-A

First, I transform the 49 temporal measurements into k-space. I average the transformed measurements, smoothing out much of the noise in the original data while keeping the strong amplitude associated with the submarine frequency common in all temporal measurements. Figure 3) compares a single measurement in k-space to the average of the k-space measurements. The exact frequency vector $\vec{k} = [k_x, k_y, k_z]$ of the submarine is clearly visible after the averaging. Then, each k-space measurement is multiplied by a gaussian centered at $\vec{k} = [k_x, k_y, k_z]$, the frequency of

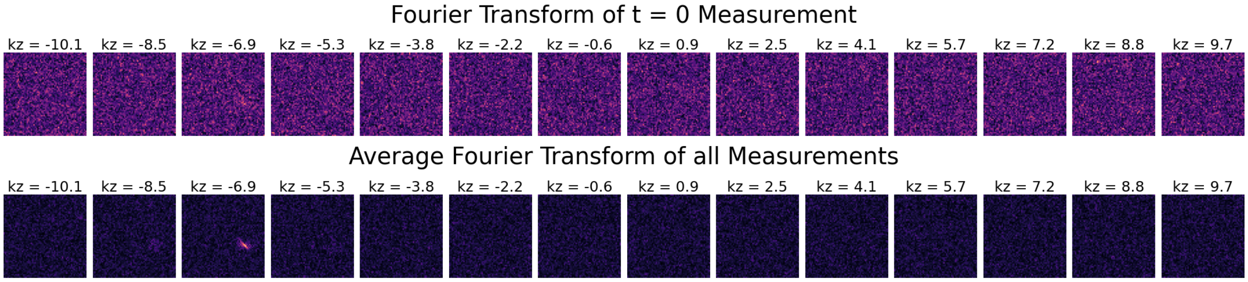

FIGURE 3. (**top**) Fourier transform of the first temporal measurement vs (**bottom**) Average of all the Fourier transformed measurements. k_x and k_y are the horizontal and vertical axis of all plots.

the submarine, before applying the inverse Fourier transform to return to x-space. The result is a filtered version of the x-space measurements where noisy signals other than the submarine are suppressed. Figure 2) compares the $x - y$ projection of unfiltered data against the filtered data. The maximum amplitude of the filtered x-spaced data at each time are plotted in the left of figure 4).

Method-B

This method consists of simply taking the absolute value of each unfiltered measurement and then finding $[x, y, z]$ coordinates which had the largest amplitude. Figure 1) was created using this method. This method used basic `numpy` commands like `abs`, and `argmax`.

4. COMPUTATIONAL RESULTS

The effect of averaging the Fourier transformed measurements can be seen clearly in figure 3). Much of the noise that was in the top plot is gone in the bottom plot after averaging, and there is a clear bright spot in the data located at in the $k_z = -6.9$ plane that was difficult to see in the top plot. The exact location of this bright spot was $\vec{k} = [2.2, 5.3, -6.9]$. Likewise, after filtering with a gaussian centered at this frequency in k-space, we can see in figure 2) that the location of submarine became clearly visible as the only bright spot in the filtered data.

Comparison between the two different methods can be seen in figure 4). Method-A, involving the filtering of the noisy data in k-space resulted in a much more consistent trajectory of the submarine. However, except for a few points, method-B of simply finding the maximum value in all the measurement still was able to trace the submarine's trajectory successfully.

5. SUMMARY AND CONCLUSIONS

This was a simple yet instructive project in demonstrating how Fourier transforms can be used in signal processing to filter noisy data. In addition, this project made extensive use of `numpy` and `matplotlib`, allowing me to practice fundamental data analysis skills like working with 3D data arrays and making informative plots using foundational python packages.

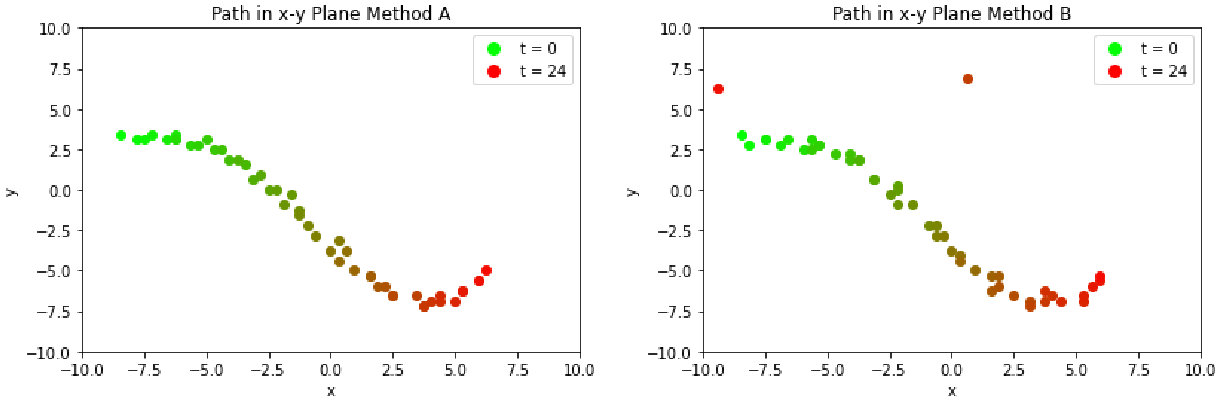

Figure 4. Comparison of the tracing of the submarine's trajectory by method-A (**left**) and method-B (**right**).