

AMATH 582 Homework 1: Denoising Stationary Signal Using FFT

Hongda Li

January 15, 2021

Abstract

The paper seeks to explain the process of using FFT as a way to denoise static signal. Specially using the assumption of the presence of whitenosie within the data, we can leverage the statistic feature of normal distribution to reduce the amount of noise in the frquency domain. To demonstrate it, we were given the hypothetical scenario of searching for a submarine and synthetic data to work with, and the goal is to recover the full path of the submarine.

1 Introduction and Overview

We are interested in locating a submaring in the Puget Sound area given some noisy acoustic data. The assignment is designed to test the skill of filtering out noise in stationary waves, with the reward of figuring out the full path that the submarine takes.

The data given is a cube of data, collection every 30 mins during a 24 hours period. The data is in the format of complex double and the magnitude of each of the data point represents the presence and absence of the submarine. And that is, if it's zoro then it marks its absence else it marks its presence.

2 Theoretical Background

2.1 Fourier Series/Transform, and Frequency Domain

As we learned from our textbook [1], Fourier introduced the concept of representing a given function $f(x)$ by a trigonometric series of sines and cosines:

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n \cos nx + b_n \sin nx) \quad x \in (-\pi, \pi]. \quad (1)$$

And the coefficients of the sin and cos is up to our interests because it transform the signal with additional useful mathematical properties. The continuation of the Fourier Series is Fourier Transform:

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

And the inverse Fourier Transform is:

$$f(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \exp(ikx) F(k) dk \quad (3)$$

The Fourier Transform extend the domain for the coefficients of sin and cos to the real domain.

For computational reasons, signal is discretized into 2^n floats and then a discrete Fourier Transform is performed on the signal. The algorithm is relatively efficient with $\mathcal{O}(N \log(N))$ complexity where N is the length of the signal.

In addition, the Fourier Transform above is appicable for signal in $[-\pi, \pi]$ range, but in actual application, the signal locates in $[-L, L]$ or $[L/2, L/2]$ (depends on convention) range. The the additional caveats that, the FFT algorithm shifts the domain in its implementation.

Therefore, for any given signal in the domain $[-L/2, L/2]$, the multiplier k equals to $\frac{2\pi}{L}$, and when the signal has been discretized: $k_i = \frac{2\pi i}{L}$ where $-\frac{N}{2} \leq i < \frac{N}{2}$. Therefore, our frequency domain goes from $-\frac{\pi N}{L}$ to $\frac{\pi N}{L}$.

2.2 Denoising in the Frequency Domain

The idea here is to observe the same signal over different time frame of times and realize that the expected value of white noise in the frequency domain is zero. This idea is Further Discussed in The data science textbook[2].

Let ϵ_i to be a white noise signal sampled with discretization of N , then:

$$\Re(\hat{\epsilon}_i) \sim \text{Normal}(0, \sigma) \wedge \Im(\hat{\epsilon}_i) \sim \text{Normal}(0, \sigma) \implies \mathbb{E}(\hat{\epsilon}_i) = 0 \quad (4)$$

And this means that, summing over the same stationary signal over different time frame will reduce the magnitude of white noise in the frequency domain because the real and imaginary part of the white noise has an expectation value of zero.

2.3 Filtering and Reconstruction of the Submarine Path

Because the signal is stationary, therefore the frequency component is unchanging, and in this particular case, the submarine emits a particular frequency that is unknown.

Therefore after the denoising process, we seek to find the unknown frequencies by simply looking for the frequency component that has maximum magnitude.

After the identification of the signal, we use a Gaussian filter to boost this particular frequencies in the original signal. The Gaussian filter is given as:

$$\exp(\mu_x(x - x_0)^2 + \mu_y(y - y_0)^2 + \mu_z(z - z_0)^2) \quad (5)$$

Where x_0, y_0, z_0 is the frequencies we try to boost (Should be the signal that the submarine is emitting) and the multiplier μ_x, μ_y, μ_z determines the window size of the Gaussian filter.

After the desired signal emitted by the submarine is identified, using the Gaussian filter and applied it to the signal in the Fourier domain will allow us to listen to this particular frequency. A similar idea but for 1D signal is discussed in the textbook[3], here we extend the same idea into 3D.

Because white noise is random, there will be significant amount of phase cancellation. Therefore, if we inverse Fourier Transform the filtered signal, looking for the signal that is peaking at each time frame will give us the approximate location of the submarine.

3 Algorithm Implementation and Development

The process of identifying the path requires the following procedures

1. Establish basic parameters such as the spatial and Frequency domain.
2. Taking the average on the frequencies domain for all realization and identify the frequency.
3. Use the frequency to construct a Gaussian Filter, apply the filter to the data and then reconstruct the signal.
4. Identify the path of the submarine from the reconstructed signal.

3.1 Signal Averaging and Frequencies Identification

For the first part, we need to establish the vectors that represents the spital and frequence domain. On line

Algorithm 1: Setup Spatial and Frequency Domain

- 1: **Input:** Subdata
 - 2: Reshape Subdat into $49 \times 64 \times 64 \times 64$ sequence of data cubes
 - 3: **Initialize:** $L = 10$
 - 4: **Initialize:** \mathbf{k}_s , to be the frequency domain vector shifted
 - 5: **Initlize:** X, Y, Z to be 3d meshgrid of equally spaced 64 points (excludes endpoints) on $[-L, L]$
 - 6: **Initliaze:** $\mathbf{k}_x, \mathbf{k}_y, \mathbf{k}_z$, to be 3d meshgrid using \mathbf{k}_s
-

4 of [Algorithm 1](#), the frequency domain vector is given by:

$$\text{fftshift} \left(\frac{2\pi}{2L} \left[\frac{-n}{2} : \left(\frac{n}{2} - 1 \right) \right] \right) \quad (6)$$

3.2 Averaging and Identify Unknown Frequencies

Algorithm 2: Identifying Unknown Frequencies

- 1: **Precondition:** Algorithm 1 has been run
 - 2: Average := zeros
 - 3: **for** TimeFrame T in Subdata **do**
 - 4: Perform FFT on T , with shifts
 - 5: Average += T
 - 6: **end for**
 - 7: Average /= Total Number of Realizations
 - 8: Let T be the the magnitude
 - 9: Let T be normalized
 - 10: Identify indices that has element in T which are larger than a given threshold
 - 11: Let the list of indices be: \mathbb{I}
-

Take note that, we have selected frequencies with a magnitude that is larger than a certain threshold. This choice is made for the purpose of relexing the constraint, giving us the possibility see just how much higher is the magnitude of the peaking frequencies compare to all the others.

3.3 Applying Filter and Reconstruction

Algorithm 3: Filtering, Reconstructing Path

- 1: **Precondition:** Algorithm 2 has been run
 - 2: Use \mathbb{I} and $\mathbf{k}_x, \mathbf{k}_y, \mathbf{k}_z$, to find a list of coordinates
 - 3: Take the average of the above list of coordinate, it's an estimation of the unknown frequencies, name it \mathbf{P}
 - 4: Construct a guassian filter using \mathbf{P} , name it \mathbf{GF}
 - 5: **Initlize** Filtered: to store the signal after frequencies has been filtered
 - 6: **for** each time frame: F in Signal **do**
 - 7: Apply shifted FFT on F
 - 8: Apply \mathbf{GF} on F
 - 9: Apply FFT shifts on F
 - 10: Apply Inverse FFT on F
 - 11: Store to: Filtered
 - 12: **end for**
-

3.4 Tracing the Path

Algorithm 4: Algorithm 4

- 1: **Input:** "Filtered" from Algorithm 3
 - 2: Initlized: Results, to store a list of 3d coordinates for all timeframes
 - 3: **for** Timeframe T in Filtered **do**
 - 4: Take the magnitude for T, normalized it, and figure out the index with the maximum magnitude
 - 5: Find the coordinates using the index and X, Y, Z meshgrid
 - 6: Append it to "Results"
 - 7: **end for**
 - 8: Visualize Results
-

4 Computational Results

The distribution of the magnitude of the averaged signal in the Fourier Domain is the most convincing evidence that, a stationary signal exists in the frequencies domain. As a common knowledge, the distribution of the modulus of a complex number is the Rayleigh Distribution, a Chi Square distribution with $DF = 2$. Therefore, we expect low level confidence of observing magnitude on the tail of the distribution. Anything contrary observation will yield existence of a stationary signal, intuitively. This is showed in [Figure 1](#), the first plot is fitting the Rayleigh Distribution, however there are a fair amount of frequencies that is higher than the threshold 0.7.

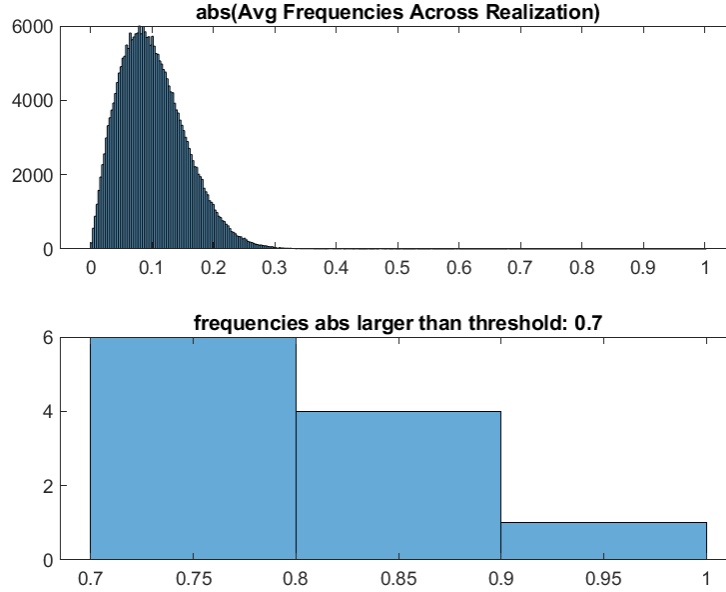


Figure 1: Distribution of the magnitude of averaged frequencies

If the frequencies that is peaking indeed is the frequency that the submarine is emitting, then it will be in a narrow range. Which is indeed the case and the frequencies is plotted in [Figure 2](#), and the red dot is the location of frequency with maximum magnitude, and all other frequencies that is above the threshold close to each other. Next, we use the approximate location of the frequency, which in the box $[4.6, 5.4] \times [-7.2, -6.6] \times [1.6, 2.2]$, and that is the approximately where the gaussian filter is covering. The filter is plotted in [Figure 3](#). Finally, the filter is applied as described in [Algorithm 4](#), and after selecting all the peaking signal in the Spatial domain, we were able to recover the full path of the submarine, as showed in [figure 4](#).

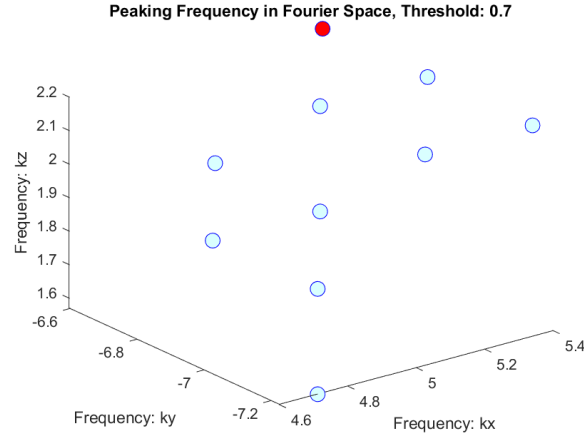


Figure 2: Distribution of the magnitude of averaged frequencies

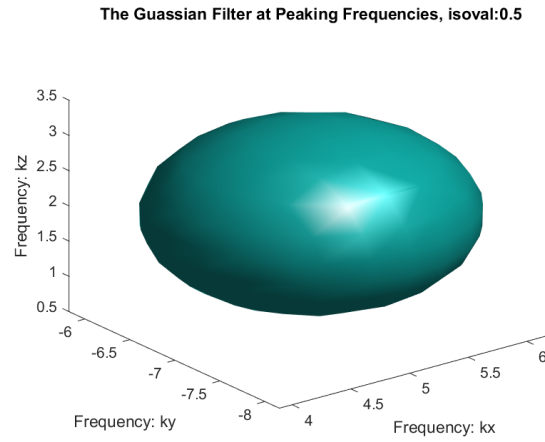


Figure 3: The Gaussian Filter

5 Summary and Conclusions

Add your summary and conclusions here.

References

- [1] Jose Nathan Kutz. *Data-driven modeling & scientific computation: methods for complex systems & big data*. Oxford University Press, 2013.
- [2] Jose Nathan Kutz. *Data-driven modeling & scientific computation: methods for complex systems & big data*. Oxford University Press, 2013, p. 316.
- [3] Jose Nathan Kutz. *Data-driven modeling & scientific computation: methods for complex systems & big data*. Oxford University Press, 2013, p. 312.

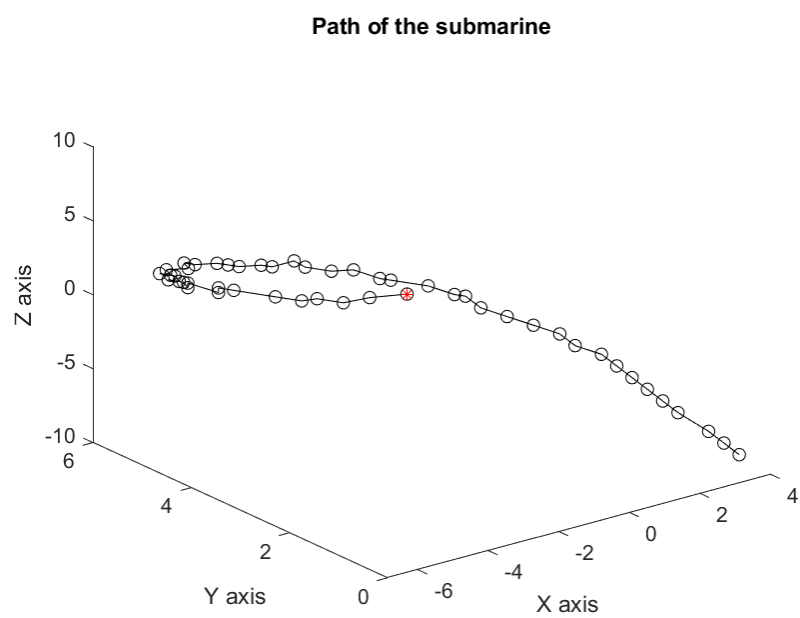


Figure 4: Path of the Submarine