AMATH 482/582: HOME WORK 1

SHENGBO JIN

CFRM Program, University of Washington, Seattle, WA
shengboj@uw.edu

Abstract. This report aims at locating a submarine from noisy acoustic data. The main method is using frequency analysis to denoise 3D signals. With the property of Fourier Transform and white noise, pre-filter the signal to explore the center frequency. Then, use Gaussian filter to denoise the signals further, then use Inverse Fourier Transform to get clear data in real domain. At last, with denoised data, locate the submarine and determine its path.

1. Introduction and Overview

A submarine is moving underwater in Puget Sound. This submarine is a new technology that emits an unknown acoustic frequency that needs to be detected. The only available resource is a broad-spectrum recording of acoustics data obtained over 24 hours in half-hour increments with noise. This data contains a matrix with 49 columns of data corresponding to the measurements of acoustic pressure taken over 24 hours. The measurements themselves are 3D and taken on a uniform grid of size $64 \times 64 \times 64$. For locating the submarine and determining its path, denoised 3D signals need to be acquired through Fourier Transform and appropriate filter.

In this report, under Python environment, Fast Fourier Transform is realized by *NumPy* [1] package. *Matplotlib* [2] and *Plotly* [3] packages are used for visualization.

2. Theoretical Background

In the general signal processing field, we use Discrete Fourier Transform (*abbr.* DFT) which is truncated from Fourier Series (*abbr.* FS) to transform the signal from real domain into frequency domain. As usual, we use Fast Fourier Transform*abbr.* FFTto compute DFT. For this problem, we also need frequency analysis to denoise the signal, with extending the application of Fourier Transform into three dimensions.

$$f(x) \approx \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} c_k e^{-\frac{ik\pi x}{L}}, c \in \mathbb{C}$$

$$f(x_0), ..., f(x_{N-1}) \underset{IFFT}{\overset{FFT}{\rightleftarrows}} (c_{-\frac{N}{2}}, ..., c_{\frac{N}{2}-1})$$

Based on the hint, *It is known that adding mean zero white noise to a signal is equivalent to adding mean zero white noise to its Fourier series coefficients.* In the equation format:

$$\frac{1}{k}\sum_{k=0}^{n-1}\widehat{(f+\xi)}^{(k)} \sim \frac{1}{k}\sum_{k=0}^{n-1}\hat{f}^{(k)} + \frac{1}{k}\sum_{k=0}^{n-1}\xi^{(k)}$$

Date: January 29, 2022.

Thus, we can use this technique to effectively denoise signals in the situation where 49 measurements over the 24-hour period are available that are subject to the same noise. By averaging the measurements in the Fourier domain, the noise should reduce to zero, then the center frequency will stand out. However, we can find that submarine data is not sufficient, so we still need to do extra filtering.

We have known the Gaussian filter in 1D and 2D, which widely used in digital signal processing and image processing. The distribution function of them is just like:

$$G(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}}$$

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Now, extend the definition to the 3D domain, and build a 3D Gaussian filter shifted to match the center frequency. Apply this filter to remove noise in the signal of each timestamp, we can get clear data to locate the submarine.

3. Algorithm Implementation and Development

Step 1 Find the frequency signature of the submarine
  a. Load data with dimensions $64^3 * 49$
  b. Loop through the columns(i.e., 49 measurements over 24 hours):
    i. Reshape the data to be 3D cube
    ii. Implement 3D FFT in original signals using *fftn()*
    iii. Use *fftshift()* for sake of visualization
  c. Take the absolute value after averaging the frequency signals
  d. Find location of the peak in frequency space
    i. Find index using *max()* and *where()*
    ii. Visualize the peak by plotting 3D isosurface with Plotly
    iii. Take the 2D slices and create contour plots for two frequency signatures using Matplotlib

Step 2 Build Filter centered at the center frequency
  a. Create two Gaussian filters corresponding to pairs of center frequencies, and make a linear combination of them
  b. Visualize the designed filter by plotting 3D isosurface with Plotly
  c. Take the 2D slices and create contour plots for the designed filter using Matplotlib

Step 3 Apply the filter to each time step to denoise the signal and find submarine location from that denoised signal
  a. Apply the filter to the averaged frequency signal and visualize it to check the performance
  b. Loop through the columns:
    i. Reshape the data to be 3D cube
    ii. Take the 3D FFT and shift it using *fftn()* and *fftshift()*
    iii. Multiply the shifted frequency signal by the deigned filter
    iv. Shift and take the IFFT using *ifftshift()* and *ifftn()*
    v. Take the real part using *real()*
    vi. Find index of max entry using argmax and unravel_index
    vii. Locate x, y, z coordinates based on index

Step 4 Plot the 3D path of the submarine and plot the x, y coordinates of the submarine during the 24-hour period using Matplotlib

4. Computational Results

- Task 1

  Average the Fourier transform each time step, most of the noise should average to zero. Then, we assume the location in the frequency domain with maximum value as the frequency signature. Visualize the peak by plotting 3D isosurface:

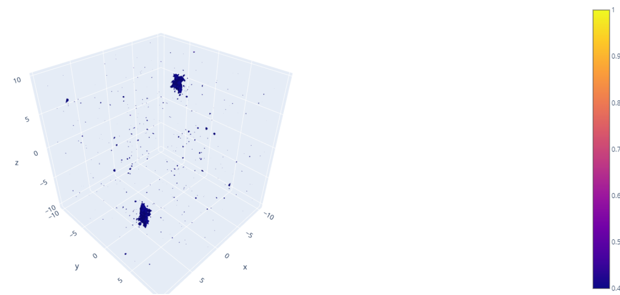

Figure 1. 3D isosurface of the average FFT

We can see pairs of center frequency in the domain. In addition, these two frequency signatures are symmetric in space. The specific frequency locations are around (5.34, 2.20, -6.91) and (-5.34, -2.20, 6.91) in Kx, Ky, Kz axis.

	Kx	Ky	Kz
1st frequency signature	-5.341	-2.199	6.912
2st frequency signature	5.341	2.199	-6.912

Table 1. Frequency signature pair

Visualize it by taking 2D slices of the frequency signals and making contour plots:

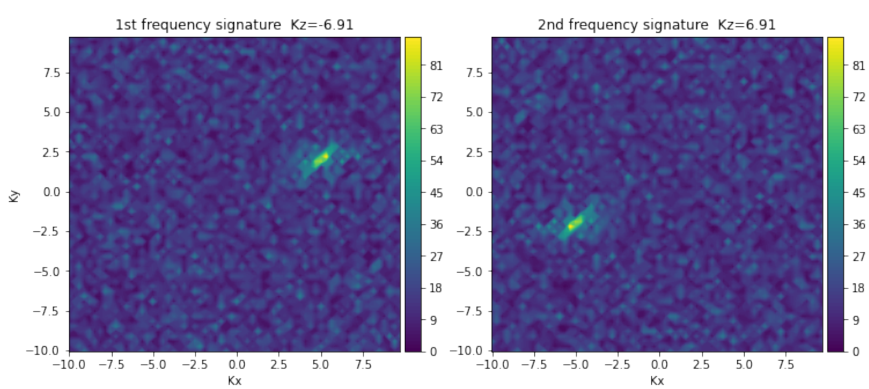

Figure 2. Frequency signature pair shown in 2D slices

- Task 2

  First, create two different Gaussian filters and shift them for matching two frequency signatures in the frequency domain. Then, design the final filter by making a linear combination of them. After many attempts and comparisons, set sigma of Gaussian filter equal to 2. Visualize the designed filter by plotting 3D isosurface:

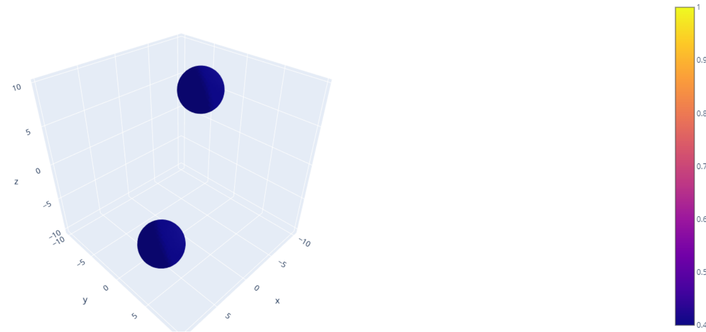

Figure 3. 3D isosurface of the designed filter

For showing the effect of the designed filter, create 2D slices of the domain and make contour plots. We can check the center of filters match the center frequency pair.

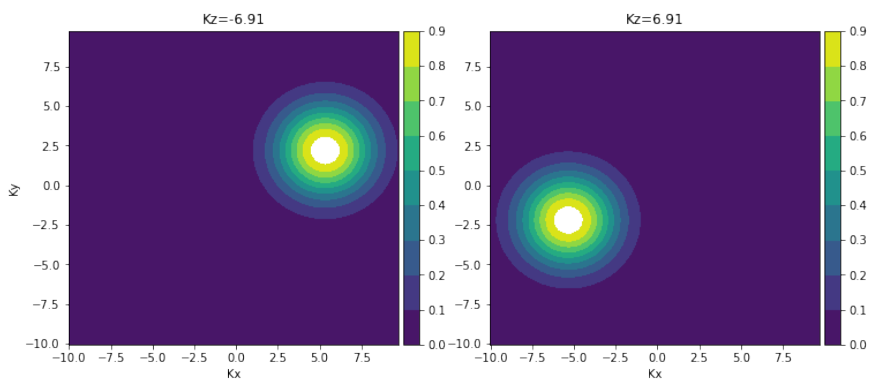

Figure 4. Filter effect shown in 2D slices

Apply the filter to the averaged frequency signals in the domain, we can see the signal has been denoised successfully. Visualize the filter performance against noise:

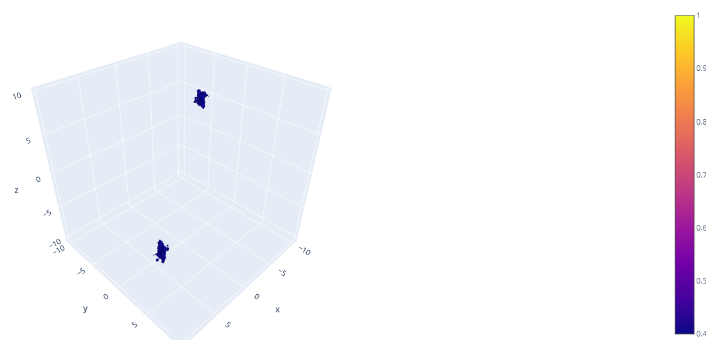

Figure 5. Filter performance shown by 3D isosurface

Next, apply the designed filter to the each FFT of the original signals over 49 measurements, and use IFFT to get denoised signals in the real domain. Then, for each timestamp, we assume the location with maximum value in the domain as the submarine location. Using these locations, we get the 3D path of the submarine over 24-hour period.

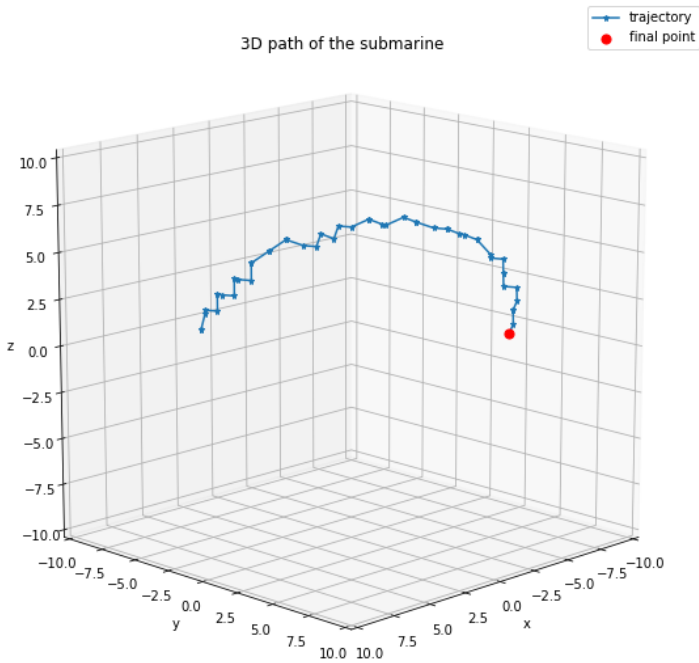

FIGURE 6. 3D path of the submarine

- Task 3

  At last, based on the coordinates data of the location, determine the x,y coordinates of the submarine during the 24-hour period, i.e., 2D path of the submarine in xy-plane.

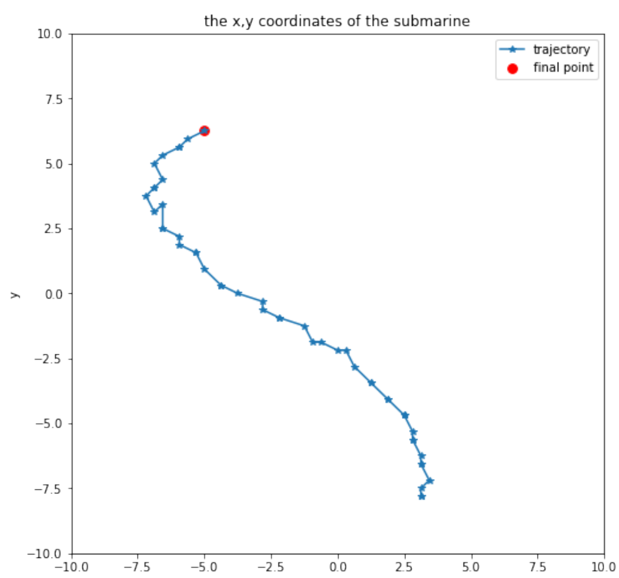

FIGURE 7. 2D path of the submarine

5. Summary and Conclusions

Firstly, I apply 3D FFT to transform acoustic signals into frequency domain for each measurement and average them to denoise some kind of white noise. With these pre-filter frequency signals, I locate center frequency pair. At the same time, I also find the frequency signature is symmetric in frequency space.

Secondly, according to this finding, I try to design two 3D Gaussian filters with different peaks which match with center frequency pair. And make a linear combination of them to design the final filter. After applying to the averaged frequency signal and visualization, I can make sure the performance of the filter is very ideal.

Finally, apply the designed filter to each signal over a 24-hour period, and inverse them to get clear data in the real domain. Reasonably, the location with maximum value for each timestamp is considered to be the submarine location. Then, use this location data to determine its path underwater.

In conclusion, I learned that Fourier Transform can be easily extended to higher dimensions for frequency analysis, like 3D in this report. For multiple center frequencies, we can combine Gaussian filters with different peaks together. In addition, from this simple report, I know about wide applications of Fourier Transform in the real world, even military use.

Acknowledgements

The author is thankful to Prof. Hosseini for lectures about signal processing. I am also thankful to Dr. Owens and Dr. Levin for careful guidance about the report. Furthermore, My peers Xuqing Hu and Xiangyu Gao were helpful in the use of Latex.

References

[1] 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.

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

[3] P. T. Inc. Collaborative data science, 2015.