HW1: FINDING SUBMARINES

TIANBO ZHANG

University of Washington, Department of Applied Mathematics, Seattle

Abstract. Maritime acoustic monitoring is an extremely important topic in the military field. In particular, the development of modern science has made this task significantly more difficult. Hence, this report focuses on the detection and tracking of a submarine in the Puget Sound using noisy acoustic data. By introducing techniques such as taking the average of the signal's Fourier transform and using custom filter for data denoising, we will found a reliable solution for submarines monitoring in the Puget Sound.

1. Introduction and Overview

Introduction: Maritime surveillance is a critical aspect of national security and naval operations. The ability to detect and track underwater war machines such as submarines is matter of both strategic and tactical importance. This report aims on the application of signal processing on acoustic data for the purpose of locating a submarine that emits an unknown acoustic frequency.

Overview: The setting of the acoustic data take place in Puget Sound, and the data has been collected over a 24-hour period with half-hour increments. The data is in the form of a 49 columns matrix corresponding to the measurements of acoustic pressure taken over 24 hours. And note that the measurements are 3D and taken on a uniform grid of size $64 \times 64 \times 64$. We will implement the process of submarine locating with 4 tasks:

(1) Averaging the Fourier transform and visual inspection to determine the frequency signature.
(2) Implement a filter to extract the frequency signature in order to denoise the data and determine a more robust path.
(3) Determine and plot the x, y coordinates of the submarine during the 24 hour period.
(4) Implement an alternative technique for noise filtering.

2. Theoretical Background

2.1. Fourier Series and Fourier Transform.

Fourier Series and Fourier Transform. The first step in our task is to find a way to translate our data collected during the 24 hour period (time based data) into frequency based data. And the fundamental theory behind this concept is Fourier Series and Fourier Transforms.

(1) **Fourier Series:** The concept of Fourier Series is to represent a given function by a trigonometric series of sines and cosines:

$$f(x) = \frac{b_0}{2} + \sum_{k=1}^{\infty}(a_k \cos kx + b_k \sin kx), x \in [0, 2\pi]$$

Note that finding the Fourier Series of a function is equivalent to computing a_k, b_k for the given signal.

(2) **Fourier Transform:** Fourier Transform is an commonly used technique to transform a time-domain signal into a frequency-domain representation. Then if $f(x)$ is a time domain signal, its Fourier Transform $F(k)$ over the entire line $x \in [-\infty, \infty]$ is defined as:
$$F(k) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} f(x) e^{-ik} dx$$
And its inverse is defined as:
$$f(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} e^{ikx} F(k) dk$$

2.2. **Discrete Fourier Transform (DFT).** Note that since our data are a collection of samples, standard Fourier Transformation is not suitable for us to compute the signal. Hence we introduced the DFT, which is a way for us to approximate Fourier Series with finite number of samples.
Let there be N samples of $f(x_0), \ldots, f(x_{N-1})$, then we have:
$$f(\hat{k}_n) \approx \frac{1}{N} \sum_{n=0}^{N-1} f(x_n) e^{-i2\pi \frac{k_n n}{N}}$$
And its inverse is defined as:
$$f(x_n) \approx \sum_{n=0}^{N-1} f(\hat{k}_n) e^{i2\pi \frac{k_n n}{N}}$$

2.3. **Averaging Fourier Transform.** Note that a common fact about white noise data is that they have mean zero. Then also note that "*It is known that adding mean zero white noise to a signal (Gaussian noise) is equivalent to adding mean zero white noise (Gaussian noise) to its Fourier series coefficients.*" Hence this fact enable us to perform a preliminary noise filtering to the data by taking the average. Then after averaging we can find the central frequency to use in the Gaussian filter.

2.4. **Gaussian Filter.** Since we are dealing with noisy acoustic data, the implementation of noise filtering is essential. Then note a Gaussian filter will have the best combination of suppression of high frequencies while also minimizing spatial spread. This characteristic enables us to perform a descent noise filtering for our data. Gaussian Filter for 3 dimensional data can be define as:
$$g(x, y, z) = \frac{1}{(2\pi)^{3/2}\sigma^3} e^{-(x^2+y^2+z^2)/2\sigma^2}$$
Then note to denoise the data we have to filter out those that are not a around the center frequency. Then we will have the following formula:
$$g(x, y, z) = e^{-((x-x_k)^2+(y-y_k)^2+(z-z_k)^2)/2\sigma^2}$$
Note that we took out the constant terms since it won't effect the data themselves.

3. Algorithm Implementation and Development

Note that since we have multiple tasks, we will have specific implementations for each task.

3.1. **Condition Setup.** Note there are several base condition we have to define:
Number of grid points in each direction: N grid = 64
Length of spatial domain (cube of side): L = 10

3.2. **Data Preprocessing.** Since that data file is a (262144, 49) matrix. We have to divide the data into points along axes, then scale them (multiply them by $\frac{2\pi}{2L}$) to frequency domain.

3.3. **Averaging Fourier Transform.** Note that our data set has 49 columns corresponding to the measurements of acoustic pressure taken over 24 hours in half-hour increments. Hence we will perform the following algorithm:

Algorithm 1 Averaging Fourier Transform

```
for each time step from 1 to 49 do
   Reshape the data set to cubic representation:(64, 64, 64)
   Calculate the fft using fftn from numpy package
   Add the fft to the sum of previous ffts
end for
Shift the fft to correct order by fftshift from numpy package
Normalized the data by taking its absolute value and divide by its
maximum absolute value
```

3.4. **Find Center Frequency.** Center frequency has strongest connection with the maximum frequency. Note that after averaging the ffts from previous step, we just have to find the index of the maximum value. Then find its corresponding x, y, z values to get the center frequency.

3.5. **Gaussian Filter and Butter Bandpass Filter Application.** Take the center frequency we got from previous step and plug it in the Gaussian filter formula for 3 dimension, we will have our filter. Then we can perform the following algorithm:

Algorithm 2 Apply Gaussian Filter

```
Set up the Gaussian function as formula stated in 2.4
Plug in σ and the center frequency k_x0, k_y0, k_z0
for each time step from 1 to 49 do
   Reshape the data set to cubic representation:(64, 64, 64)
   Calculate the fft using fftn from numpy package
   Shift the fft to correct order by fftshift from numpy package
   Apply the filter to the data
   Normalized the data and find the maximum frequency
   Transform the data back into position form and add it to the
path list of x, y, z
end for
```

3.6. **Hyper-parameter Tuning.** After the application of two types of filters, we will notice that the value of σ and n has an noticeable impact to the filtered data. So we have to perform a hyper-parameter tuning for σ and n to find the proper value. Hence we will use $\{1, 2, 3, 4\}$ for σ and $\{2, 4, 6, 8\}$ for n.

4. **Computational Results**

4.1. **Original Data Visualization.** First note our acoustic data before denoising looks like this:

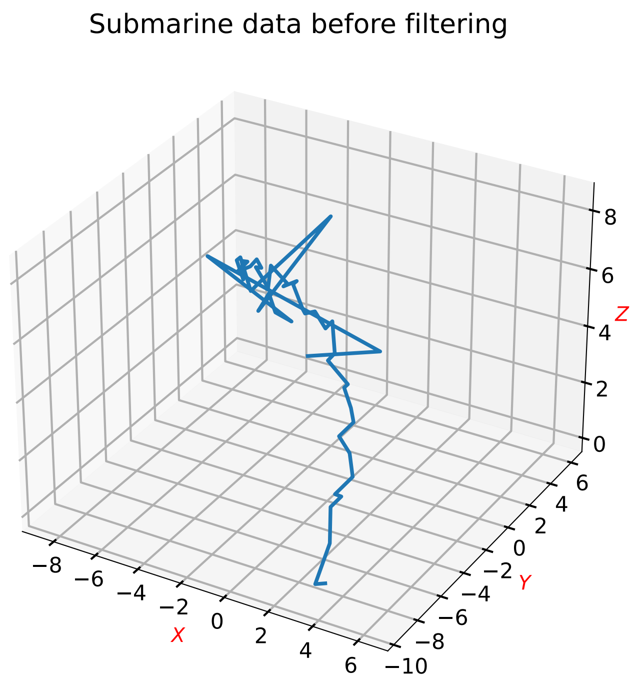

FIGURE 1. Original submarine path data without noise filtering

Note that from Figure 1 we can see that the location of the submarine is really spiky due to the un-filtered noisy data. Hence in order to approximate submarine's path we will perform the preliminary filtering.

4.2. **Noise Filtering.** After implementing the algorithm from 3.3 we found the central frequency to be:

$(k_{x0}, k_{y0}, k_{z0}) = [5.340707511102648, 2.199114857512855, -6.911503837897545]$

Then plug this into the Gaussian filter formula we will have the Gaussian filter to denoise the original data. Note that with different values of σ we will have different path. Hence we have to find the appropriate hyper-parameter σ. We can accomplish this step by visualize the effect with different σ value on the resultant submarine path.

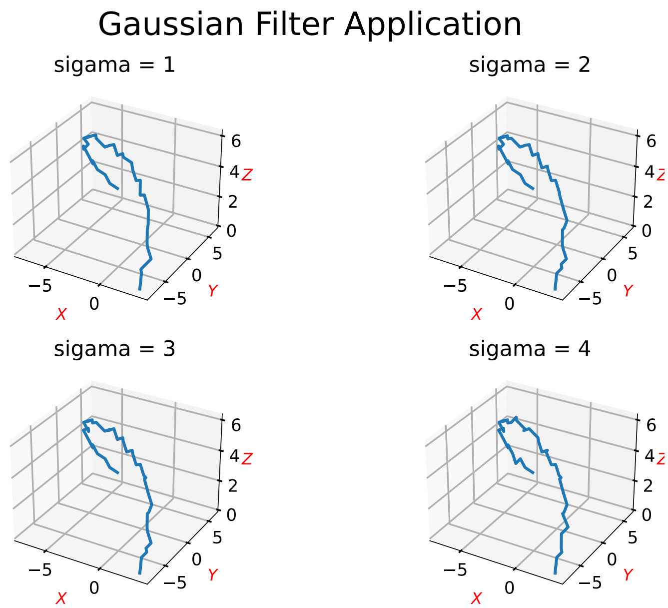

FIGURE 2. Submarine path with different σ

As we can see that as σ grows bigger in Figure 2, the path shows a more spiky pattern. Hence we choose $\sigma = 1$ for our Gaussian filter. We can also see the effect of Gaussian filter by look at the XY plane

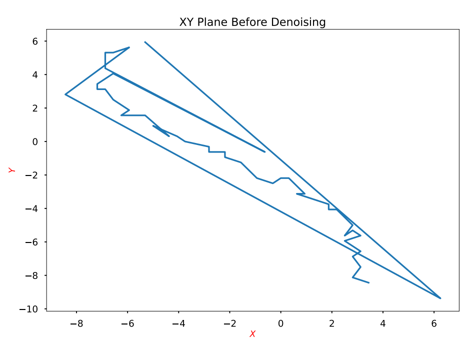

FIGURE 3. XY Original

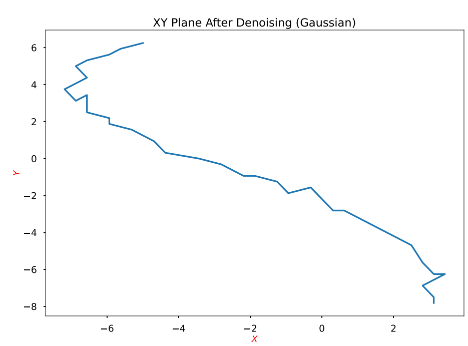

FIGURE 4. XY Gaussian

In general these two figures demonstrate the importance of noise filtering. We can see that the path becomes much more clear than before.

5. Summary and Conclusions

In conclusion, the process of locating moving submarine with noisy acoustic data is indeed challenging. But with the power of Mathematics, nothing is impossible. This report has demonstrated the utility of signal processing techniques for the challenges of detection and tracking of submarine. With a careful application of

Fourier Transform, we identified the distinct central frequency of the submarine. Then the implementation of Gaussian filter effectively isolated this signature and provided us a clear path of the submarine's movements over a 24-hour period.

The path determined from analysis not only give us the submarine's coordinates with high precision but also suggested a consistent operational pattern. These tracking data provided us a framework that could enhance the effectiveness of maritime monitoring in the future.

In addition we introduced an alternative noise filtering approach using Band-pass filter, which yielded a some what accurate result. These findings emphasizes the importance of adaptable and multi-faceted strategies in acoustic signal processing.

While the results are promising, the limited number of sample measurements and the complexity of higher dimensional data suggest that further research is needed. The continued studies in filtering techniques could very much improve the accuracy and reliability. This study investigation will pave the way for more sophisticated surveillance methods and contributes to the ongoing enhancement of maritime security measures.

6. **Acknowledgements**

The author is thankful to Pro.Zhang from Beijing University of Post and Telecommunications for useful discussions about potential three-dimensional filters. I am also thankful to Ms.Haravi for the suggestions in creating more informative plots for this report.

7. **References**

References

[1] J.N. Kutz (2013) *Methods for Integrating Dynamics of Complex Systems and Big Data*, Oxford.
[2] Alan V. Oppenheim, Alan S. Willsky (1997) *Signals and Systems*, 2nd Edition.
[3] Electrical4U (2021) *Butterworth Filter: What is it? (Design and Applications)*.
[4] J.L. Davy, I.P. Dunn (2003) *The statistical bandwidth of Butterworth filters*, Journal of Sound and Vibration.

8. **Appendix (Extra Credit)**

8.1. **Theoretical Background:** Another choice for denoising 3 dimensional acoustic data is Band-pass filter. It allows through components in a specified band of frequencies, but blocks components with frequencies above or below this band. We will simplify this filter into a more manageable formula:

$$h(x,y,z) = \frac{1}{\sqrt{1+\sqrt{(x-x_k)^2+(y-y_k)^2+(z-z_k)^2}^{2n}}}$$

8.2. **Algorithm:** Take the center frequency we got from previous step and plug it in the Bandpass filter formula for 3 dimension, we will have our filters. Then we can perform the following algorithm:

Algorithm 3 Apply Bandpass Filter

```
Set up the Bandpass function as formula
Plug in n and the center frequency kx0, ky0, kz0
The for loop is identical in Algorithm 2
```

8.3. **Computational Results:** We use $\{2, 4, 6, 8\}$ for n in the Band-pass filter.

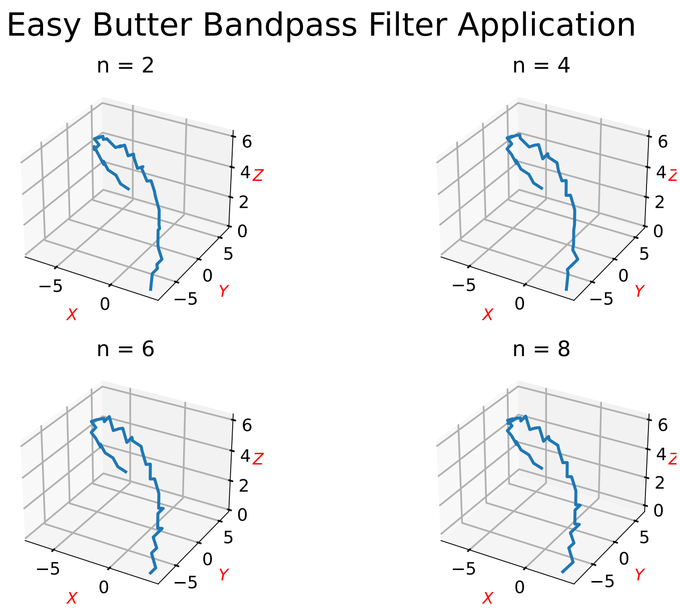

FIGURE 5. Submarine path with different n values

Note that from Figure 5 we can see that with $n = 4$, the graph shows the clearest path of the submarine. Then we will have the following graph for XY plane.

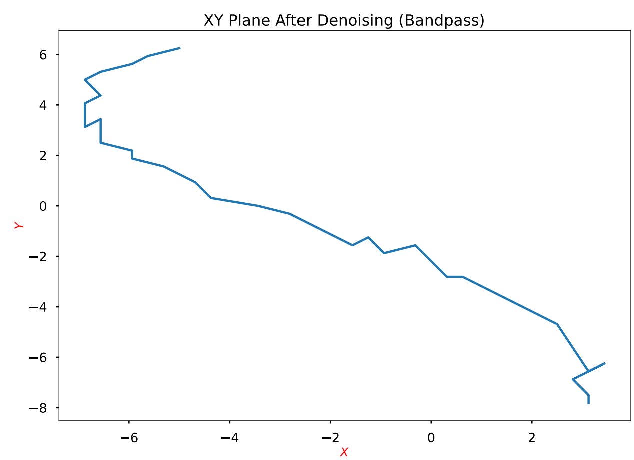

FIGURE 6. XY Band-pass

9. **Conclusion:**

Included in section 5.