ADSP Homework#1 – Frequency Estimation and CRB
Author: Pouya Aminaie
A.Y. 2023-2024

Section 1: Theory

To generate the noise in the first part of the homework, we need to compute a sample covariance matrix that is called Ĉ. To do so, we use the statistical properties of the original sample C_{ww} to generate a new sample. Then, compute the new covariance matrix. Finally, it will be compared to the original sample by using mean squared error. Once we generate samples, we need to create a covariance matrix. To do so, based on the definition of covariance matrix as below, we can compute $\hat{C}$:

In a generic form, we can write the covariance matrix for N number of observations as below:

$$\hat{C} = \frac{1}{N-1} \cdot \sum_{i=1}^{N} (W_i - \bar{W}).(W_i - \bar{W})^T$$

Which:

- W_i is *i-th* observation vector
- $\bar{W}$ is the sample mean vector

However, we are in the case where we generate only one sample at a time in each Monte Carlo iteration. Therefore, for a single vector W, if we have only one observation, the sample mean $\bar{W}$ is simply W itself since there are no other observations to average. Therefore, the product of ww^T is used to estimate the covariance matrix in Monte Carlo iteration. Finally, for Monte Carlo iterations, we calculate the sample covariance matrix by averaging the normalized outer product over all iterations, and we simply can write as below:

$$\hat{C} = \frac{W\,W^T}{N-1} = \frac{1}{N-1} \times \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_{N-1} \\ w_N \end{bmatrix} \times [w_1 \quad w_2 \quad \cdots \quad w_{N-1} \quad w_N] = \begin{bmatrix} w_1^2 & \cdots & w_1 w_N \\ \vdots & \ddots & \vdots \\ w_N w_1 & \cdots & w_N^2 \end{bmatrix}$$

After the calculation of the covariance matrix, we need to use the mean squared error (MSE) function to see the difference between $\hat{C}$ and C_{ww} and it can be written as below based on the Frobenius norm squared:

$$MSE = \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} (\hat{C}_{ij} - C_{ij})^2$$

We use the PSD method to estimate the frequency. We can estimate the waveform's frequency by taking the maximum power spectral density. First, we calculate the FFT and plot power spectrum density. Then, we consider only the positive side of the spectrum. Finally, we find the frequency corresponding to the maximum point on the PSD plot.

Section 2: Solutions

This section explains the MATLAB kernels, solutions, and plots. For each plot, we considered the vertical axis in both CRB and MSE, which is the result of the estimation.

2.1. Noise Generation

As already mentioned, we generate new samples based on the statistical properties of the original sample vectors. In MATLAB, there is a function called *mvrnd(mu, sigma, M),* which takes the mean and variance of the samples and generates a set of random variables based on these statistical properties. However, in our problem, M equals 1, which means we generate only one vector at a time in each Monte Carlo iteration. Therefore, we are in a single sample observation.

The observation of waveform can be done by:

- ten values for N starting from 20 to 200
- five values for ρ starting from 0 to 0.99

Therefore, for each N corresponding to each ρ, we have one MSE value, and the total size of the MSE matrix is 10x5. It is necessary to mention that after iterations of Monte Carlo, we take the average of all and considered it as an estimated sample covariance matrix $\hat{C}$. In MATLAB, I used *norm* function

To apply Monte Carlo analysis, we used a loop of 500 iterations in MATLAB. Then, we take the average over all experiments and consider it as an estimated value. In summary, the Figure.1 in the Appendix summarizes the procedure that we need to perform with MATLAB:

1. Shaping covariance matrix C_{ww} as defined in the problem

2. Generate the random vector by the following steps:
   - Using mean=0 and C_{ww}
   - Put them inside *mvnrnd* function

3. estimated sample covariance matrix $\hat{C}$

4. Compute MSE

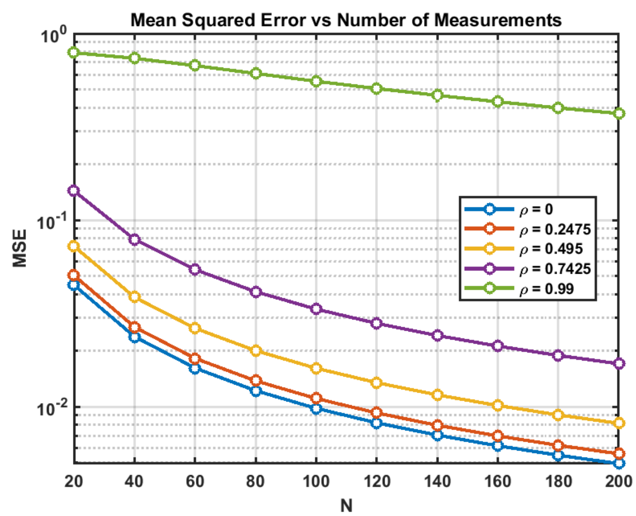
It is also necessary to mention that there could be two different approaches to illustrate the result of our experiment as below:

1. MSE vs. ρ → with different values of N
2. MSE vs. N → with different values of ρ

Result

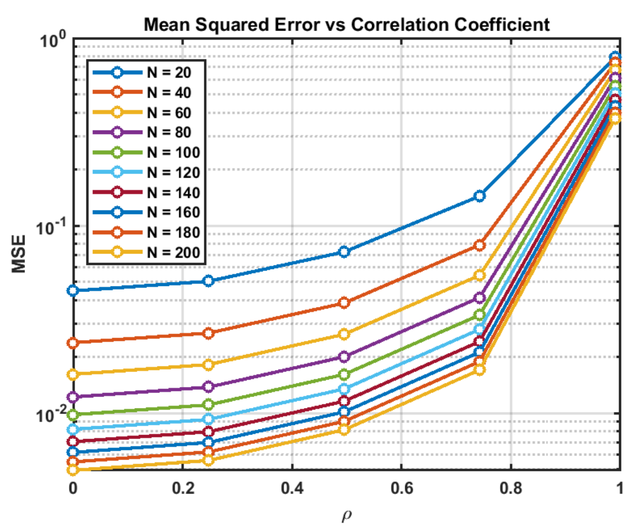
MSE vs. N:

The result of the experiment based on MSE with respect to different values of N is shown below. With a constant value of ρ, MSE decreases when the value of N increases.



MSE vs. ρ:

The result of our experiment based on MSE with respect to different values of ρ is shown below. With a constant value of N, MSE increases when the value of ρ increase. In other words, it is obvious that if we increase the power of noise, we increase the error of estimation.



2.2 Frequency Estimation

In this part, a single sinusoid (L=1) has been considered, and its frequency has been estimated. However, there are two approaches for calculating the covariance matrix.

- The first one is to use the covariance matrix $C_{ww,1} = \sigma_w^2 . I$
- The second one is to use $C_{ww,2} = \sigma_w^2 . \rho^{|i-j|}$

There are different approaches to attack the problem. One approach is to fix the of the amplitude of waveform equal to 1 (a=1) and then compute the power of the noise depending on the value of SNR that is considered. After generating the noisy waveform, we have to take the FFT and compute the power of the signal. Here is the explanation of the function used in MATLAB during implementation of PSD method.

Explain the functions of MATLAB:

1. **fftshift** ➔ why should we use fftshift:
   When we do fft on MATLAB, we take x multiplied by a matrix which is computed for indexes that goes from 0 to N. Therefore, the output without fftshift is signal which does not have the negative frequency. It has only the positive frequency. Therefore, what fftshift does is that it takes the upper frequencies and put them in negative one. So we can see odd fourier transform of the signal.
2. **waveform** → Generating the waveform and sum the noise:

$$\texttt{x=}\underbrace{\texttt{waveform(a,w\_try,N\_i,phi\_try)}}_{\textit{Generating Waveform}} \texttt{ + }\underbrace{\texttt{sqrt(sigma\_w).*randn(1,N\_i)}}_{\textit{Sum up the noise}}\texttt{;}$$

3. **rand** ➔ to generate random phase:

   - **Distribution**: Uniform distribution.
   - **Range**: Generates numbers in the range (0,1)
   - **Usage**: Useful when you need uniformly distributed random numbers.

It is also necessary to mention that we normalized the frequency. Therefore, it will range from -π to π. Therefore, we have a widow of N. We also included up-sampling. The axis of non-normalized frequency ranges from -N/2 to N/2-1. The space between the index of frequency depends on the value of the up-sampling factor, which is 8 in our case. We normalized by the factor of 2π /N_i to scale it on [-π π]

$$\texttt{f\_ax = }\underbrace{\texttt{(-N\_i/2:1/n\_us:N\_i/2-1/n\_us)}}_{\substack{\textit{Non-normalized index} \\ \textit{From -N/2 to N/2+1}}} \texttt{ * }\underbrace{\texttt{2*pi/N\_i;}}_{\substack{\textit{Normalization Factor} \\ \textit{From } -\pi \textit{ to } \pi}}$$

The flowchart of Figure.2 in the Appendix summarize all the steps that is needed to be done. Now, we discuss about the result that we get from. When we take the maximum of PSD, it gives us the maximum value and also the index of maximum value which corresponds to the argument that maximize the power spectral density.

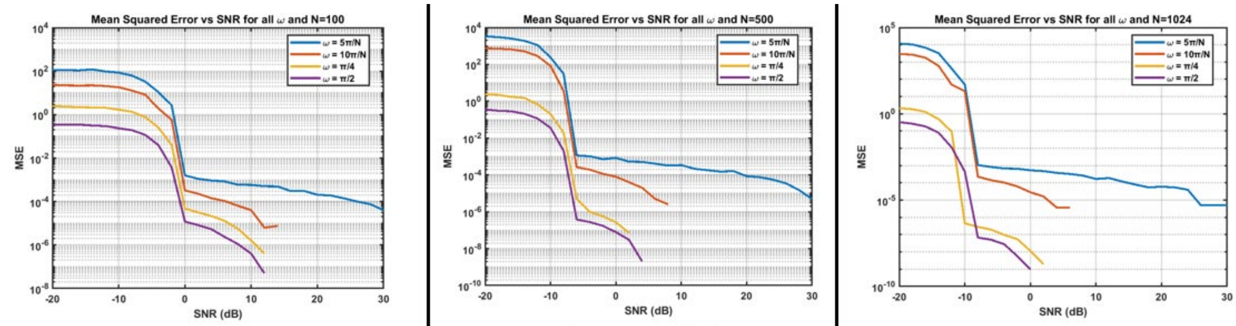
In the MATLAB code, as written below, we only take the positive side of PSD. The reason is that when we look at the maximum point, it can be in a positive index or in a negative index. So, there will be ambiguity. In order to remove the ambiguity, we can take only the positive portion of PSD and positive frequency.

```
posPSD = PSDx((N_i*n_us)/2 : end); % psd shift
 pos_f_ax = f_ax((N_i*n_us)/2 : end); % freq shift
```

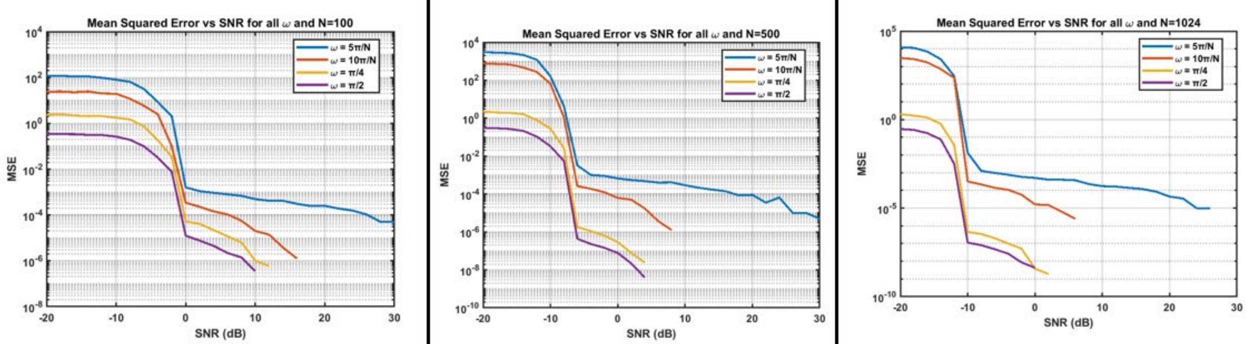
Result

MSE vs SNR

1. For all ω and N= {100, 500, 1024}
   By increasing the SNR, our estimation error decreased for all frequency values decreased. The plots are on a logarithmic scale.

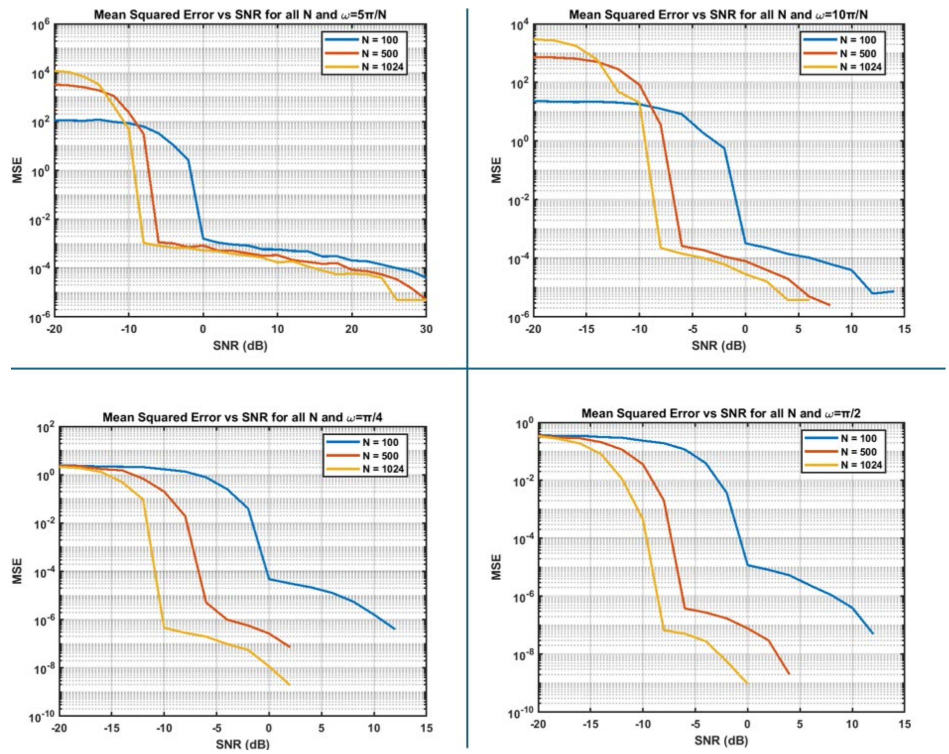


$\boldsymbol{C_{ww,1}} = \sigma_w^2 . I$



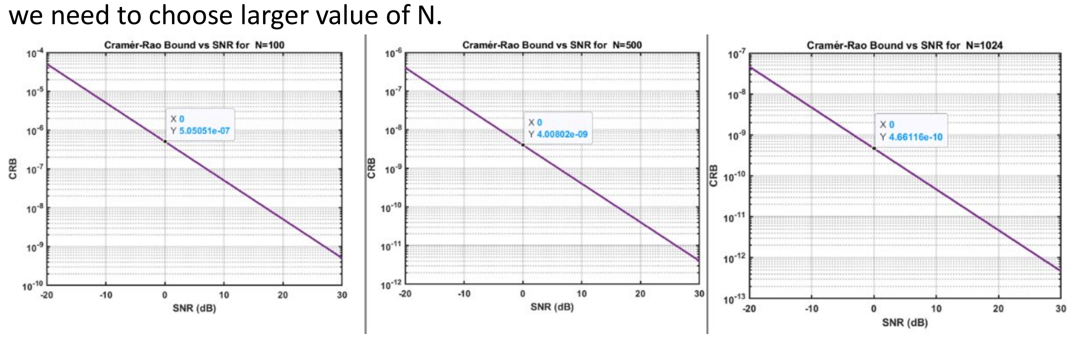
$\boldsymbol{C_{ww,2}} = \sigma_w^2 . \rho^{|i-j|}$

2. For all N and $\omega = \{\frac{5\pi}{N}, \frac{10\pi}{N}, \frac{\pi}{4}, \frac{\pi}{2}\}$ for Cww1
   The plots below clearly show the behaviour such that, with the same SNR, increasing the number of N decreases the estimation error. Obviously, we need to consider that MSE is always less than 1 for the positive SNR.



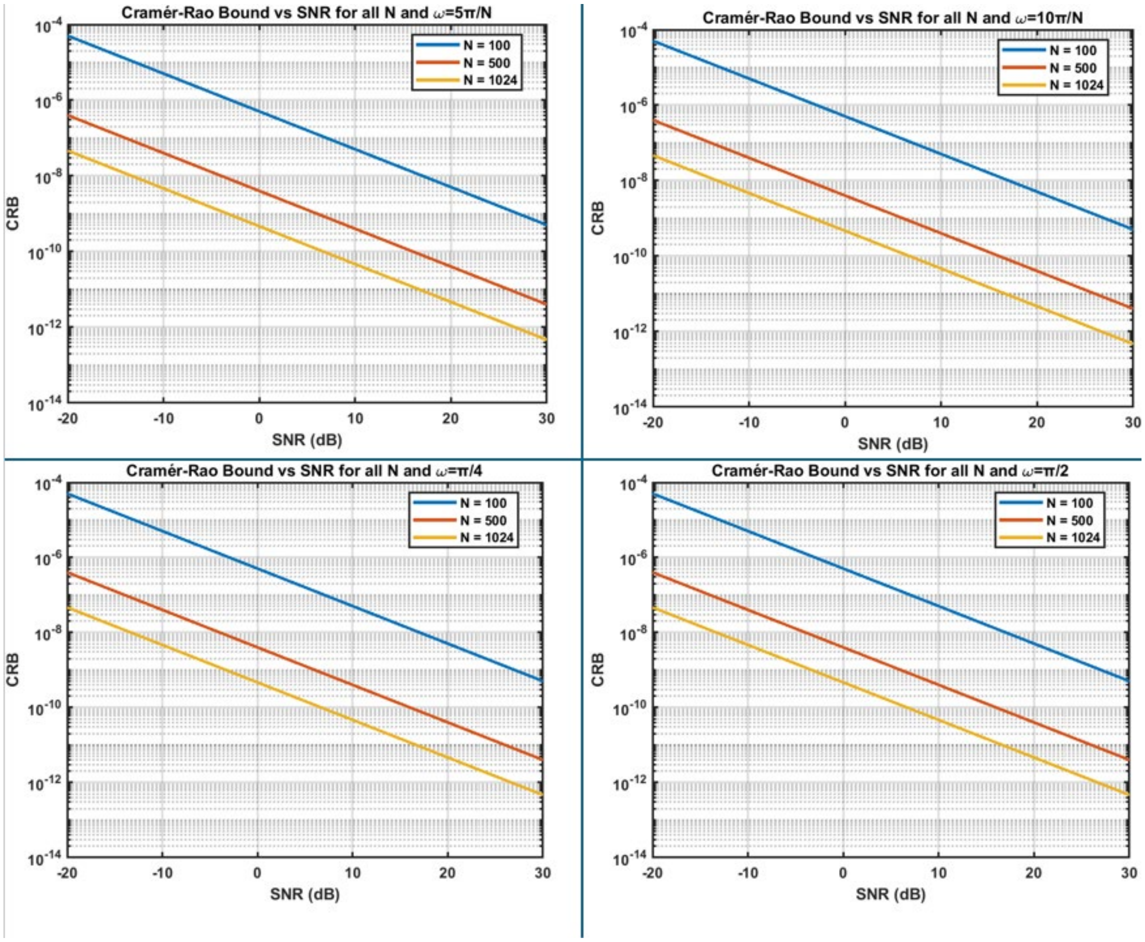
CRB vs SNR

1. For all N= {100, 500, 1024}
   The plot below shows CRB vs SNR for all values of N. The plots are logarithmic. It is clear that with the same SNR, the larger value of N have lower CRB. In other words, in order to minimize CRB, we need to choose larger value of N.



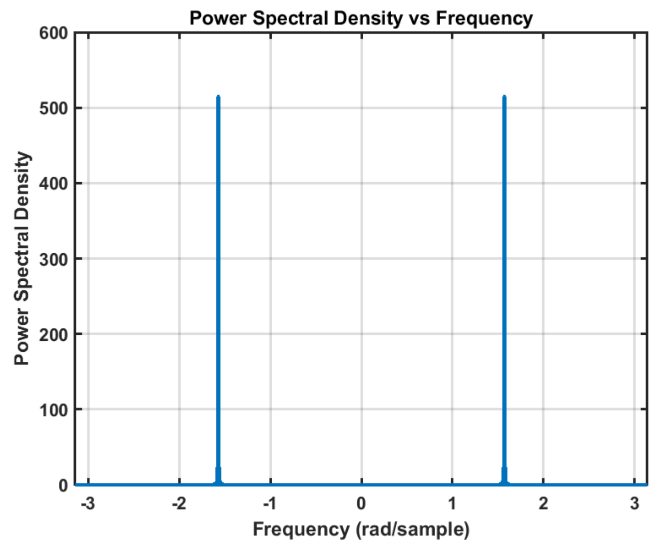
2. For all N and $\omega = \{\frac{5\pi}{N}, \frac{10\pi}{N}, \frac{\pi}{4}, \frac{\pi}{2}\}$

   As shown below, CRB vs. SNR has decreasing behaviors. The plots are on a logarithmic scale. With the same SNR, the larger N causes lower CRB.

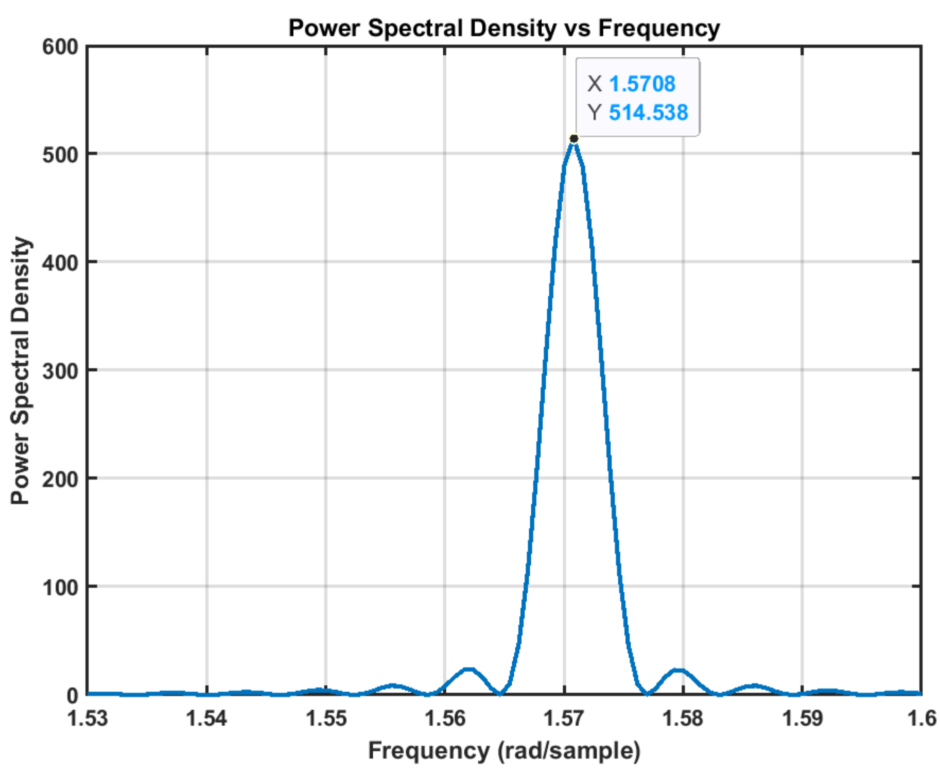


PSD

Two-sided spectrum between $-\pi$ and π is as below. The frequency we assumed to estimate by the PSD method is $\omega = \frac{\pi}{2}$ [rad/sample]

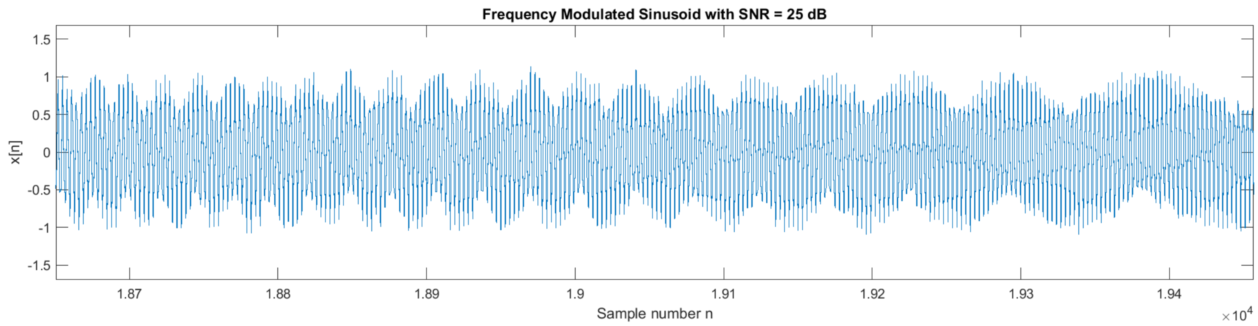


However, to find the maximum power, we need to consider only the positive part of the spectrum. The zoomed version of the one-sided spectrum is shown below. By taking a look at the PSD plot below, the maximum happened at the frequency of 1.5708 rad/sample, which can be approximated as $\frac{\pi}{2}$ and proves the method that we implemented.

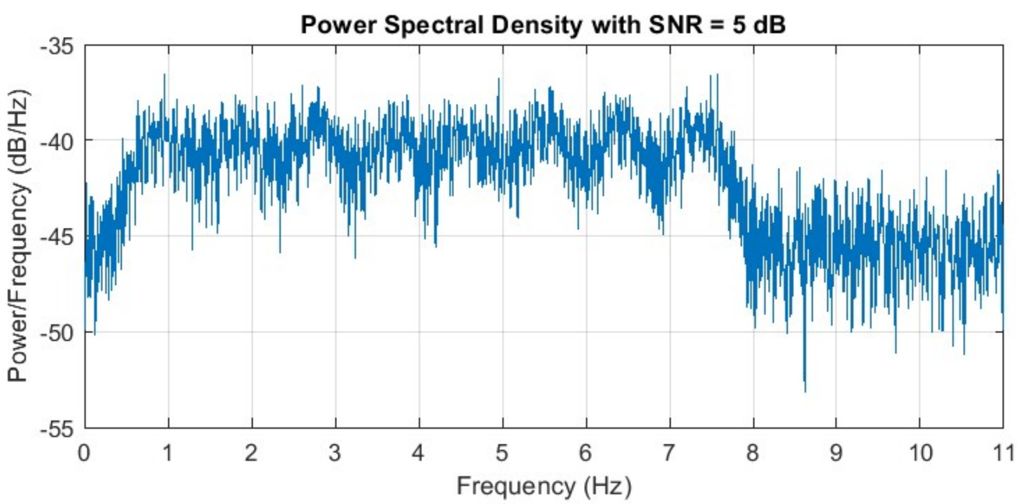


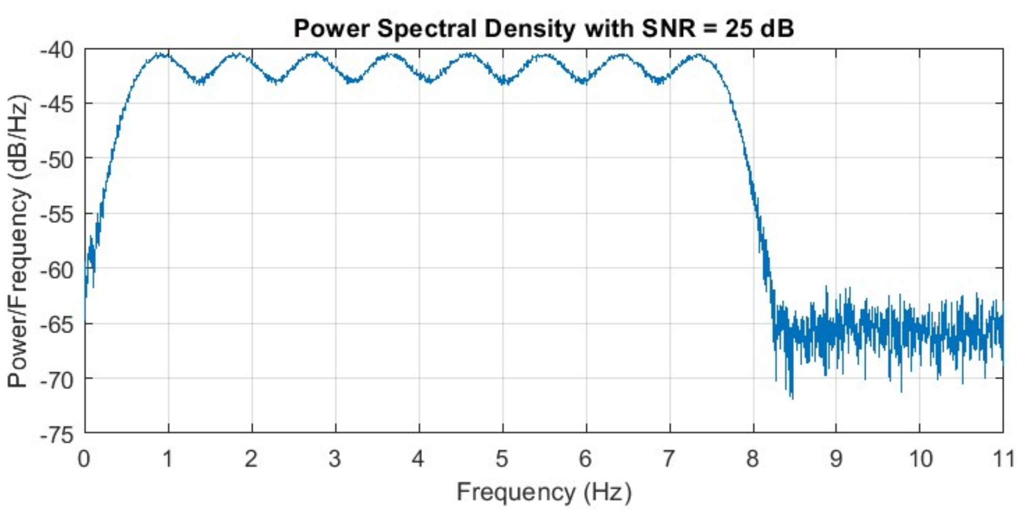
2.3 Frequency Modulation

In this part, the frequency is modulated. In the time domain, we get something like this:



If we take the PSD of the modulated signal. It is shown below:



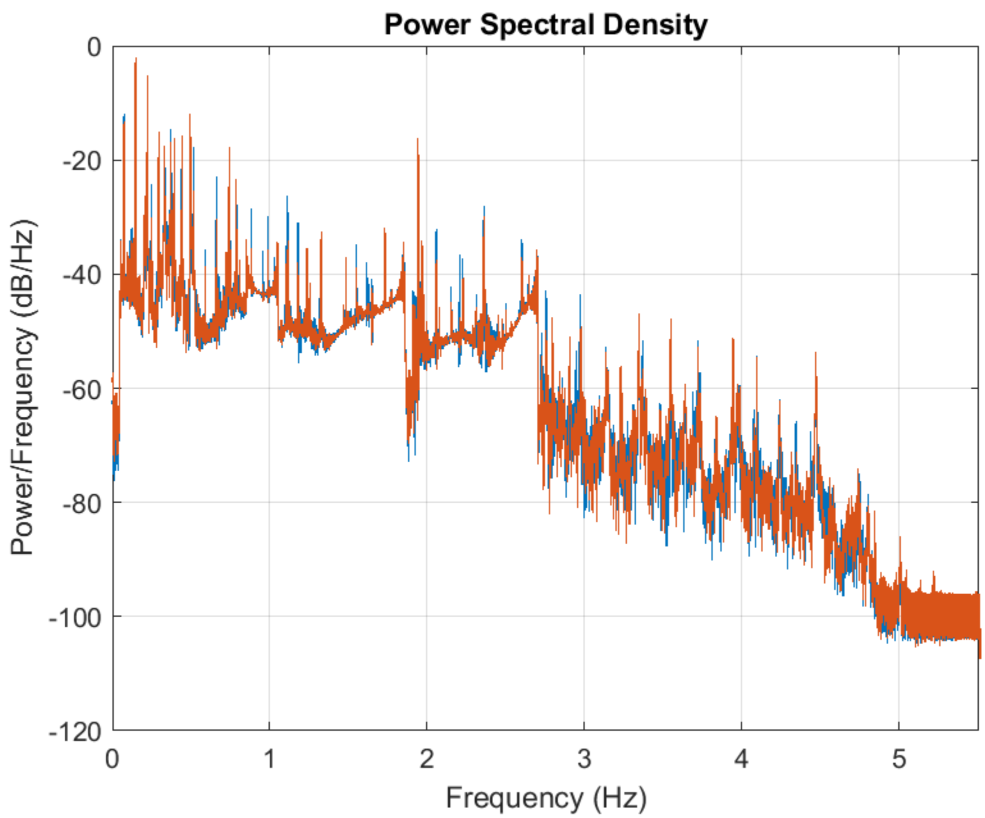


As it is shown above, PSD oscillates with frequency.

2.4 Arbitrary Modulation

In this section, the possible strategy can be the detection of harmonics corresponding to the maximum PSD points.

For example, for one of the stereo, we have:



One way to estimate the harmonics is to search for the maximum point on PSD. In the above plot, we need to find the peak point and read the frequency corresponding to that peak point.

Section 3: Conclusion

In this homework, we generated a random noise using a given statistical property. Then, we added this noise to our sinusoid waveform. Finally, we estimated the frequency of the generated waveform based on the PSD method. One possible future work can be estimating spatial frequency in radar systems, especially in array antennas. The PSD of the received signal at each element of the array should be taken when needed, and the spatial frequency corresponding to the peak point should be detected. Since the spatial frequency is proportional to the angular distance, the direction of arrival can be calculated.

Appendix

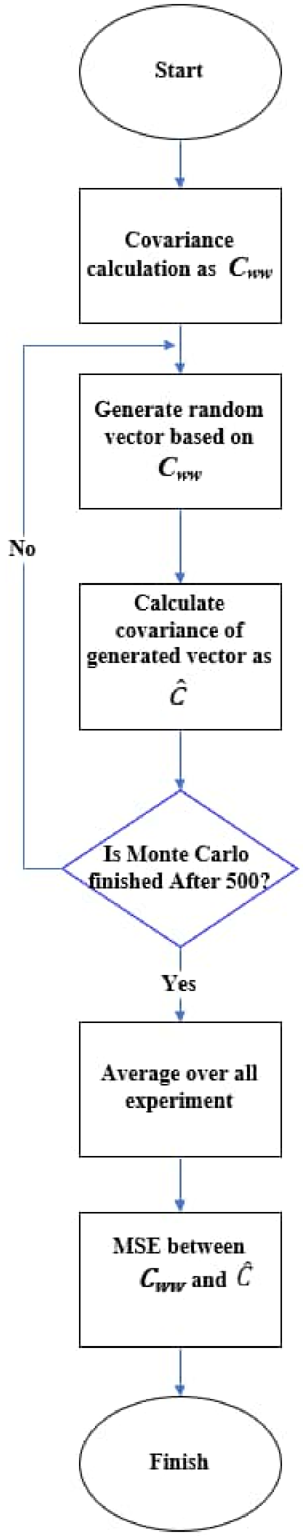


Figure 1: Noise Generation

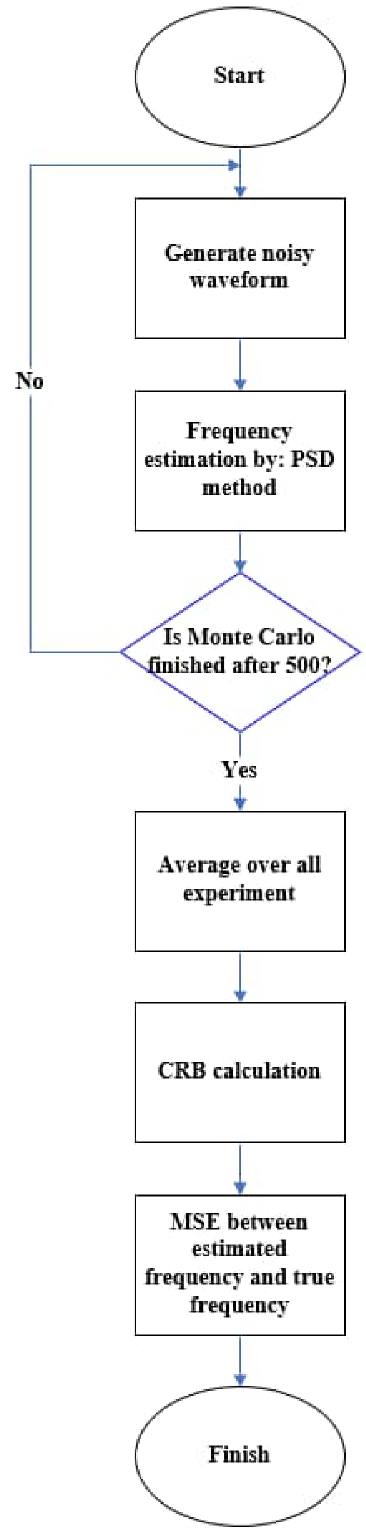


Figure 2: Frequency Estimation