ARISTOTLE UNIVERSITY OF THESSALONIKI

2nd Assignment

«Analysis of Discrete Processes: Power Spectrum and Bispectrum Estimation»

Advanced Signal Processing Techniques

in MATLAB

May 2023

2022-2023 | AEM: 10265

STAVROS KIIGMAS

ECE AUTH Student

1. Statement

Consider the real discrete process given by: $X[k] = \sum_{i=1}^{6} \cos(\omega_i + \varphi_i)$, $k = 0,1,\dots,N-1$, Where $\omega_i = 2\pi\lambda_i$, $\lambda_3 = \lambda_1 + \lambda_2$ and $\lambda_6 = \lambda_4 + \lambda_5$, $\varphi_3 = \varphi_1 + \varphi_2$, $\varphi_6 = \varphi_4 + \varphi_5$ are independent and uniformly distributed random variables on $[0, 2\pi]$. Consider that $\lambda_1 = 0.12Hz$, $\lambda_2 = 0.30Hz$, $\lambda_4 = 0.19Hz$ and $\lambda_5 = 0.17Hz$ (hence, $\lambda_3 = 0.42Hz$ and $\lambda_6 = 0.36Hz$). Moreover, let $N = 8192$ as the data length.

1. Construct the X[k].
2. Estimate the power spectrum $C_2^x(f)$. Use $L_2 = 128$ max shiftings for autocorrelation.
3. Estimate the bispectrum (only in the primary area) $C_3^x(f_1, f_2)$ using:
    a. The indirect method with $K = 32$ and $M = 256$. Use $L_3 = 64$ max shiftings for the third-order cumulants. Use:
        i. Rectangular window
        ii. Parzen Window
4. Plot $\boldsymbol{X}[\boldsymbol{k}]$, $C_2^x(f)$, $C_3^x(f_1, f_2)$ (all estimations).
5. Compare the estimations of $C_3^x(f_1, f_2)$ amongst $\{a_i, a_{ii}, b\}$ settings. Comment on the comparisons.
6. What can you deduce regarding the frequency content from the comparison of $C_2^x(f)$ and $C_3^x(f_1, f_2)$ (all estimations)?
7. How the results will change if you repeat the process from 1 to 5 taking into account:
    a. Different segment length:
        i. $K = 16$ and $M = 512$
        ii. $K = 64$ and $M = 128$
    b. 50 realizations of the $\boldsymbol{X}[\boldsymbol{k}]$ and comparing the mean values of the estimated $C_2^x(f)$, $C_3^x(f_1, f_2)$?

2. Analysis

2.1 Construction of X[k]

In this task, we aim to construct the discrete process $X[k]$ based on the given formula. We implemented the code in MATLAB to generate the values of $X[k]$.

First, we define the values of λ (lambdas) as $[0.12, 0.3, 0.42, 0.19, 0.17, 0.36]$ to represent the frequencies of the process. These values are used to calculate the corresponding values of ω_i (omegas) by multiplying each λ_i with 2π.

Next, we generate random phase offsets (phis) for each component of $X[k]$. These phase offsets are uniformly distributed random variables between 0 and 2π. Additionally, we set $\boldsymbol{\varphi_3 = \varphi_1 + \varphi_2}$ and $\boldsymbol{\varphi_6 = \varphi_4 + \varphi_5}$, as specified in the assignment statement.

To generate the values of $X[k]$, we iterate over each component of the process (i = 1 to 6) and calculate the values of $\boldsymbol{\cos(\omega_i + \varphi_i)}$ for all k. These values are stored in the cos_values matrix. Finally, we sum the values of $\boldsymbol{\cos(\omega_i + \varphi_i)}$ for all components i to obtain $X[k]$. The constructed $X[k]$ is plotted using a line plot, where the x-axis represents the index k and the y-axis represents the values of $X[k]$ time series.

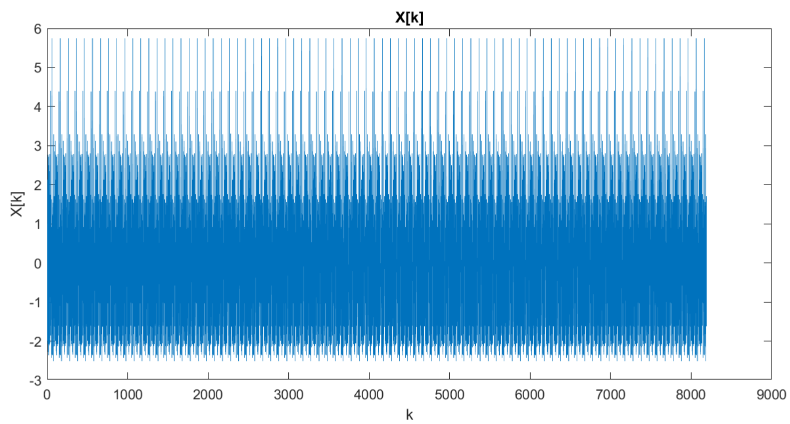


Figure 2: Visual representation of the constructed X[k] based on the provided parameters.

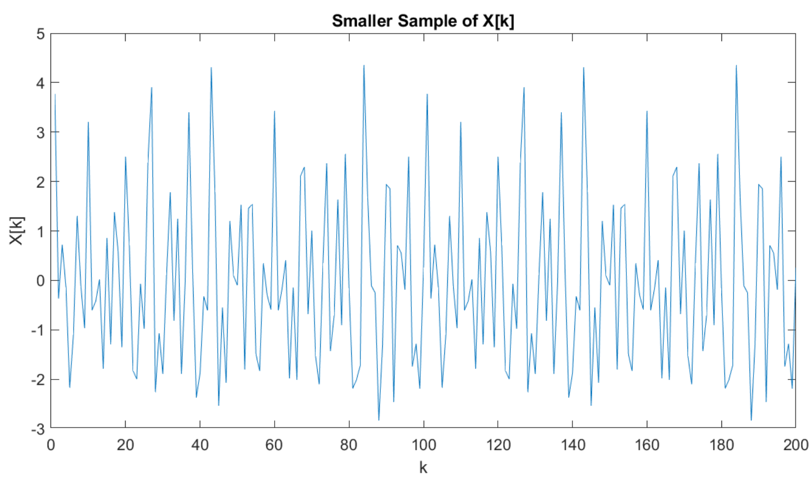


Figure 1: Smaller sample of the data.

2.2 Power Spectrum Estimation

This task aims to estimate the power spectrum of the signal X[k] using the autocorrelation method.

First, we define the maximum number of shiftings (L2) as 128 according to the assignment statement. This value determines the lags for the autocorrelation calculation.

We calculate the full cross-correlation of the signal X, including negative lags, using the "xcorr" function. Then, we normalize the cross-correlation values by dividing them by the maximum value to obtain normalized cross-correlation values (normCrossCorrFull).

Next, we calculate the autocorrelation of the signal X using the "autocorr" function. The autocorrelation values are stored in the "autoCorr" variable.

We plot the autocorrelation using the "autocorr" function, which generates an autocorrelation plot with the time axis ranging from -L2 to L2.

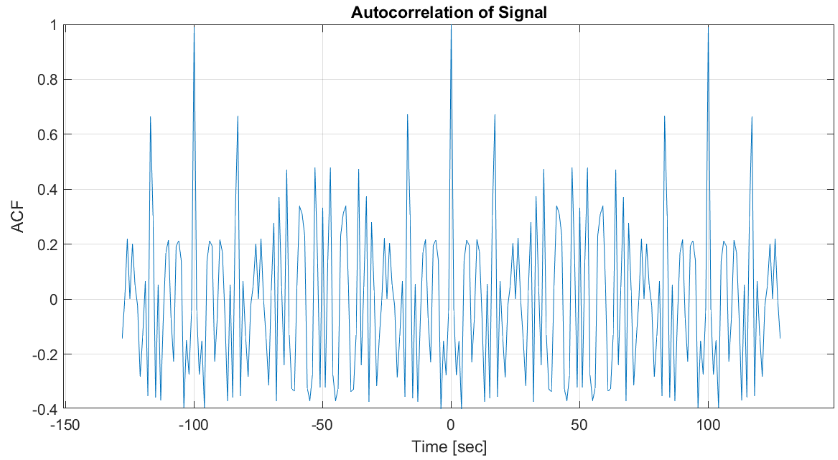


Figure 3: Autocorrelation of signal.

The normalized cross-correlation values (normCrossCorrFull) are also plotted against the time axis.

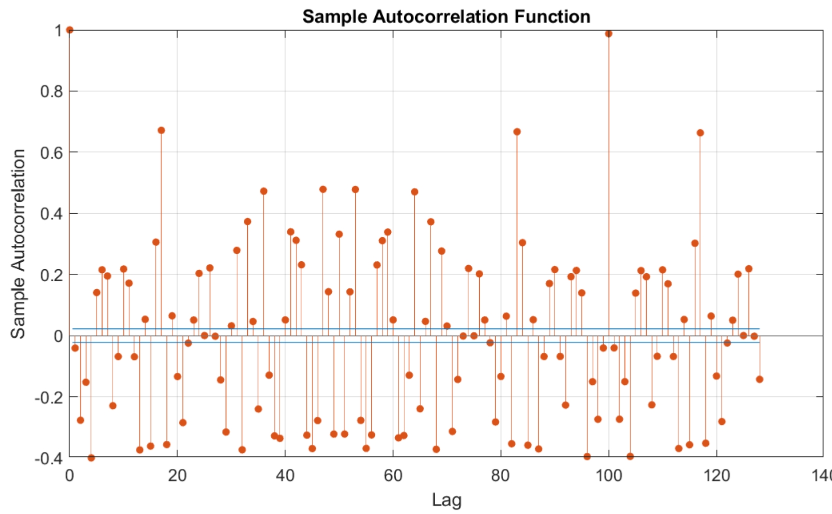
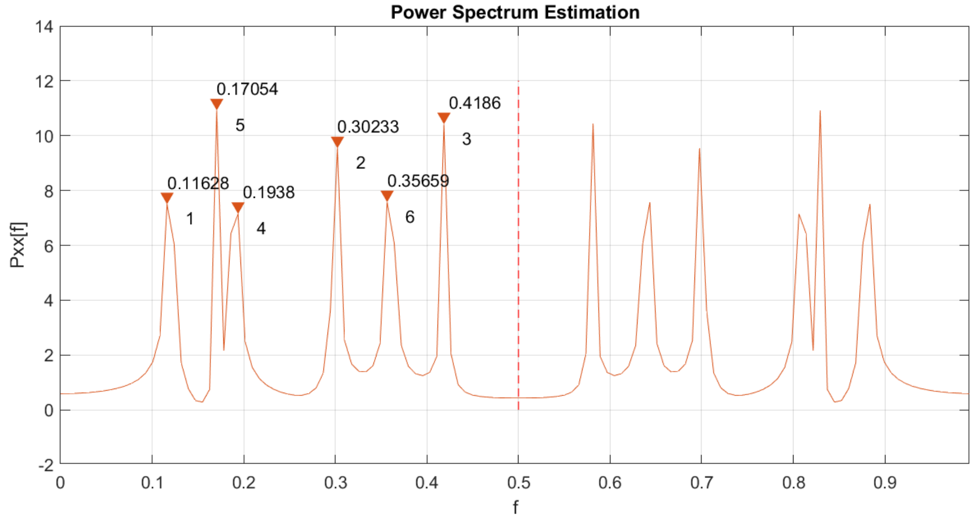


Figure 4: Normalized autocorrelation values.

To estimate the power spectrum, we take the absolute value of the Fourier transform of the autocorrelation values using the "fft" function. The resulting values are stored in the "powerSpec" variable. We create a frequency axis (freqAxis) corresponding to the power spectrum using the length of the powerSpec array.

The power spectrum is plotted using the "plot" function, with the frequency axis on the x-axis and the power spectrum values on the y-axis. Additionally, a red dashed line is added at frequency 0.5 along with the six highest peaks.



The comment we can make here is that the frequencies on the peaks are approximately the given lambdas. Theoretically, any frequencies other than these six should cause the power spectrum to zero out. The concept of spectral leakage [1]is evident here since we see an approximation of the power spectrum in a segment of finite length 8192 samples. In other words, the spectrum estimate has non-zero values in areas where there should not typically be any content. In addition, the power spectrum is symmetrically distributed around the zero, as expected.

2.3 Bispectrum Estimation

In Task 3, our goal is to estimate the bispectrum of the signal X[k]. We approach this task by using different methods and window types[2]. Let us go through each subtask:

a1) Indirect Method with Rectangular Window:

We begin by applying the indirect method with a rectangular window. The chosen parameters for this method are K=32, M=256, and L=64. We split the data into K subsets of size M using the "reshape" function. Then, we calculate the bispectrum using the "indirectBispectrum" function

[1] https://en.wikipedia.org/wiki/Spectral_leakage

[2] Harris. (1978). On then Use of Windows for Harmonic Analysis with the Discrete Fourier Transform. Proceedings of the IEEE, 66(1)

with the specified parameters. The resulting bispectrum is plotted, and the primary area, where $f_1 = f_2$, $f_1 + f_2 = 0.5$ and $f_2 = 0$, is also plotted for reference and definition of this area.

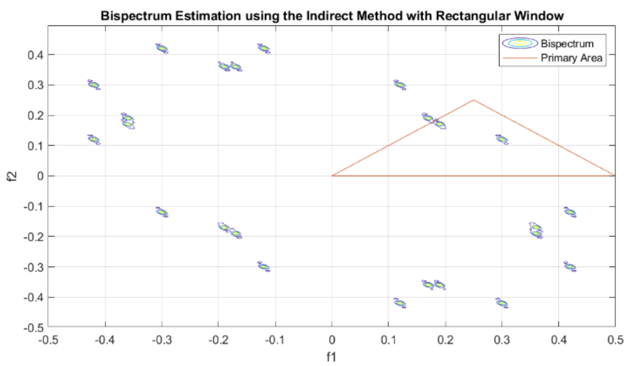
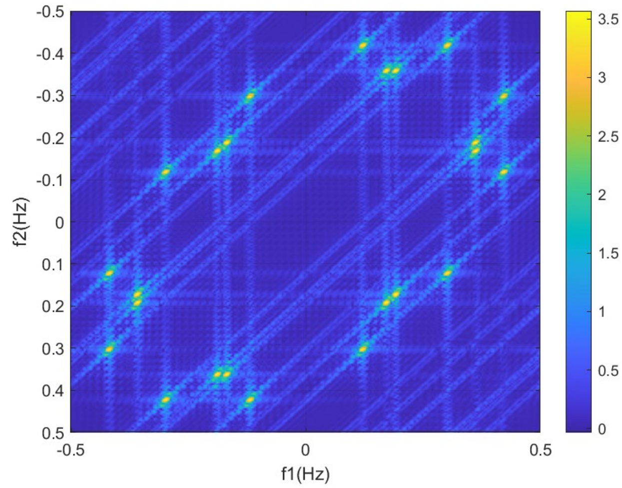


Figure 5: Indirect method with rectangular window.



a2) Indirect Method with Parzen Window:

Next, we repeat the same process as in a1, but this time we use the Parzen window instead of the rectangular window. The remaining steps and plots are identical to a1.

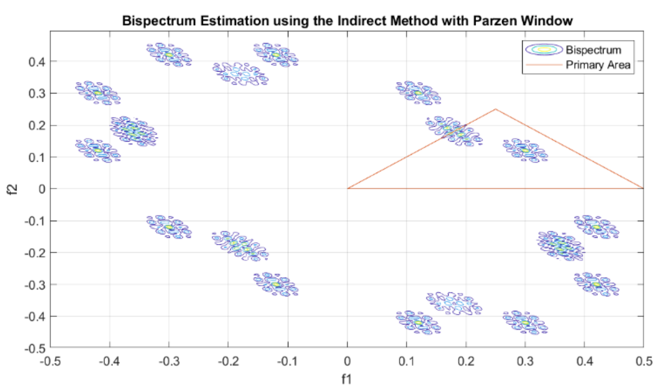
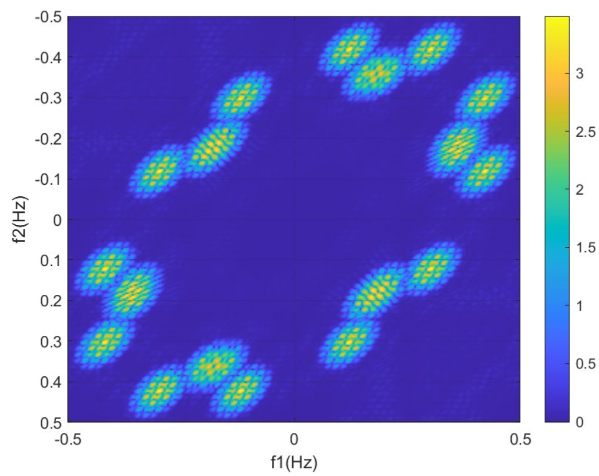


Figure 6: Indirect method with Parzen window.

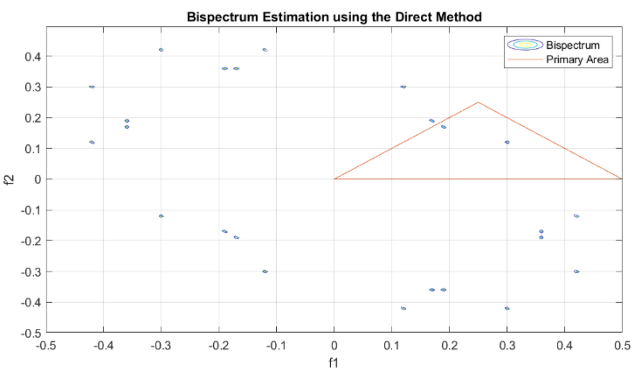


The bispectrum displays twelve areas of symmetry in both methods. When we concentrate on the primary area, we realize that we have large values at $(f_1, f_2) = (0.3, 0.12)$ and $(f_1, f_2) = (0.19, 0.17)$.

Taking a look at the diagrams we observe that the signal exhibits a large level of spectral leakage when the rectangular window is applied. The energy from each frequency component spreads into neighboring frequency bins due to the abrupt cutoff characteristic of the rectangular window. The energy of the 0.19 Hz component leaks into the 0.17 Hz bin as a result, giving the spectrum a smeared appearance. This leakage makes it difficult to distinguish between the two frequencies, which can make the predicted bispectrum inaccurate. In general, hen studying signals with tightly spaced frequency components or when the signal's spectral composition changes over time, the Parzen window is especially helpful. The spectral content is concentrated in particular regions, as can be seen in the diagram above. However, these regions are larger than the corresponding in the rectangle method, which is a negative characteristic. So, this is a kind o trade-off, accuracy over leakage.

b) Direct Method:

In this method, we employ the direct method for bispectrum estimation. The parameters chosen for this method are K=32, M=256, and J=0. The value of J determines the size of the secondary area in the bispectrum plot. We calculate the bispectrum using the "directBispectrum" function with the specified parameters. Similar to the previous methods, we plot the bispectrum and the primary area.



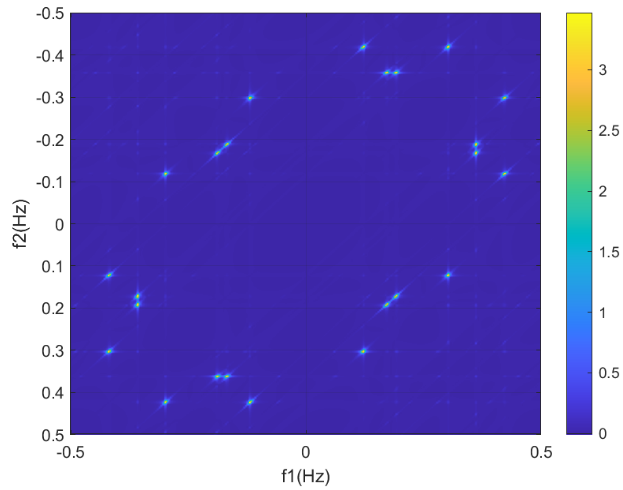


Figure 7: Direct method.

This method depicts the larger values the same frequencies as the indirect one. It manages the spectral leakage as good as the Parzen method before without the need of windows. At the same time, the areas of spectral content are sufficiently small.

2.4 Plots

The required plots are depicted in the above tasks.

2.5 – 2.6 Bispectrum Estimation Comparison

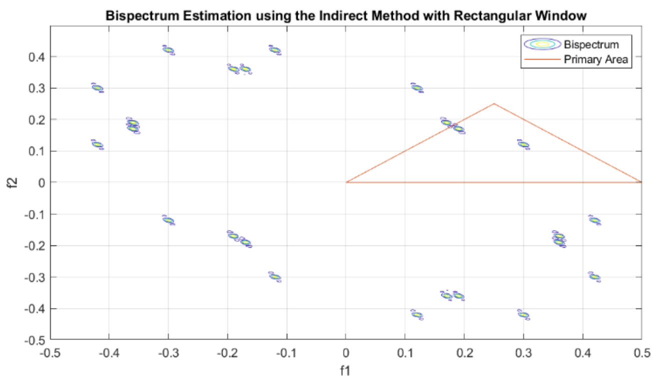
The abovementioned comments make direct method a more suitable option for our signal. We also explained the trade-off in the selection of window function in the indirect method.

2.7 Parameter Dependence

In this task, we will repeat the processes with different parameter values.

2.7.a Different Segment Length

i. $K = 16$ and $M = 512$



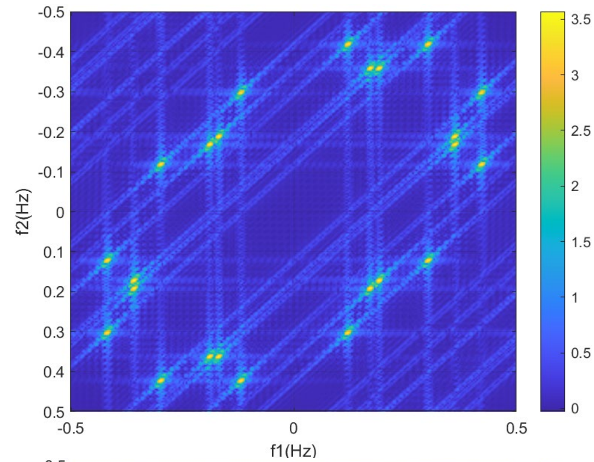
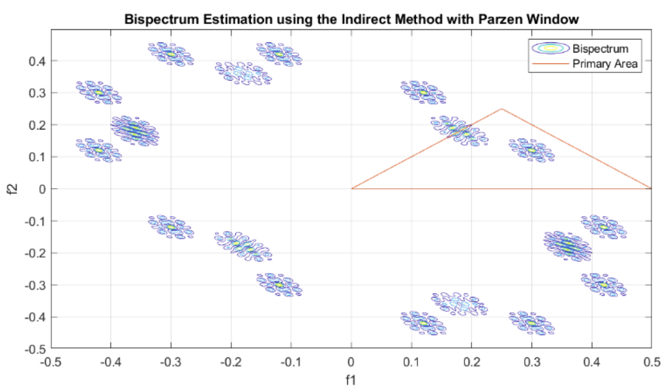


Figure 8: Indirect method with rectangular window.



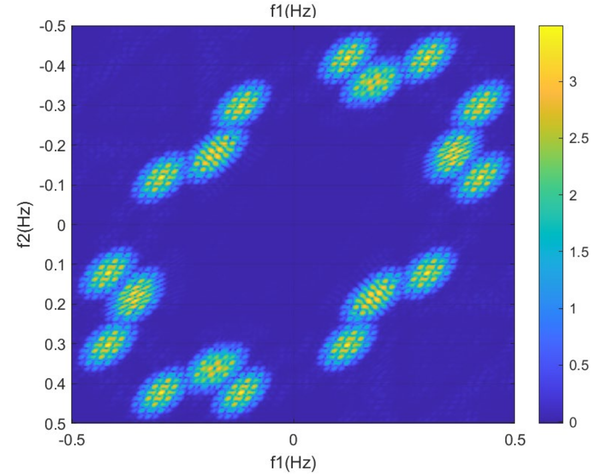
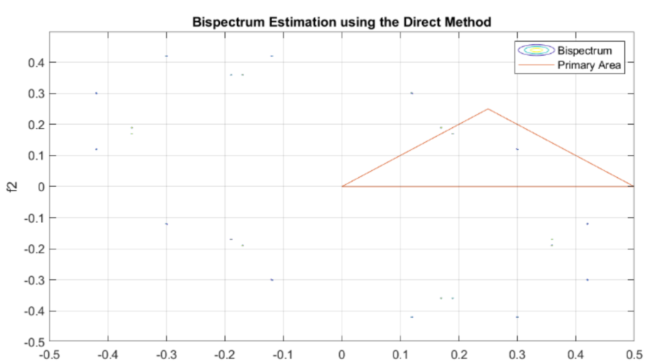


Figure 10: Indirect method with Parzen window.



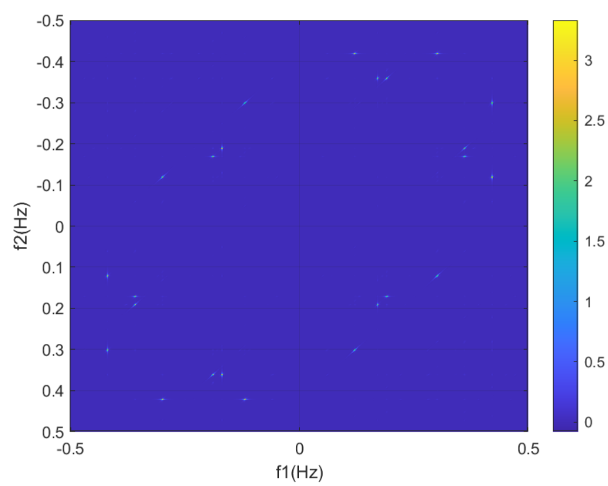
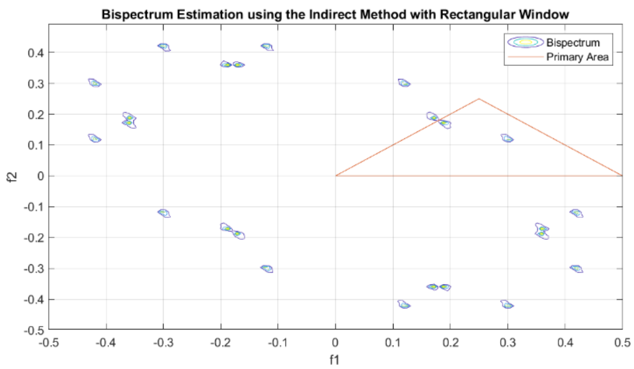


Figure 9: Direct method.

ii. $K = 64$ and $M = 128$



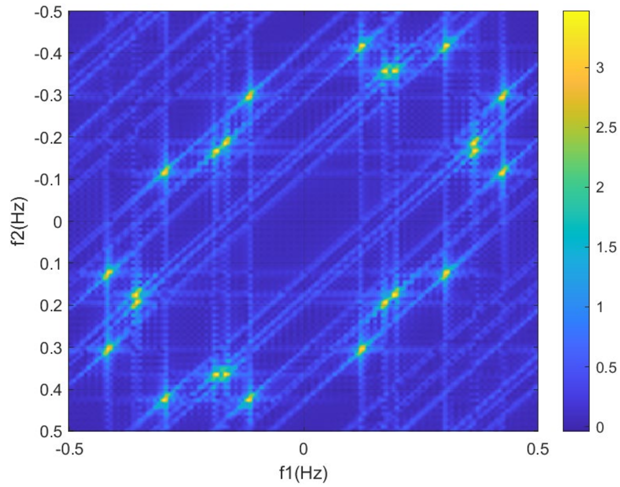
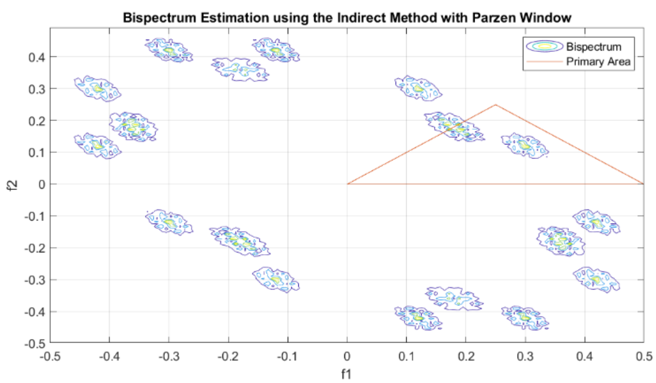


Figure 11: Indirect method with rectangular window.



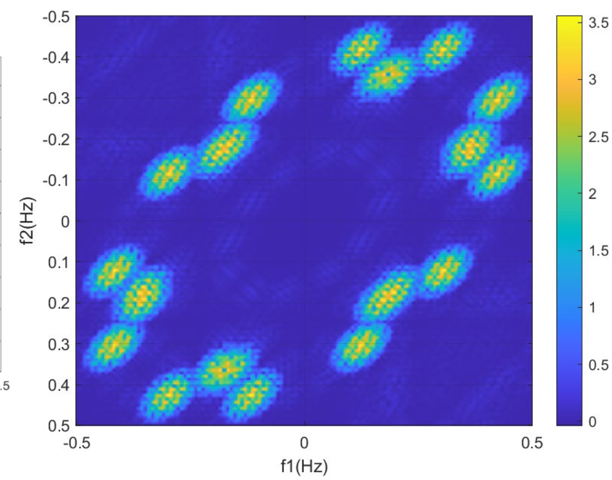
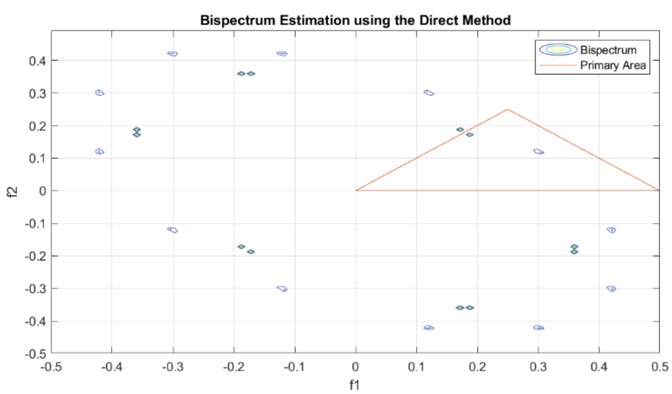


Figure 12: Indirect method with Parzen window.



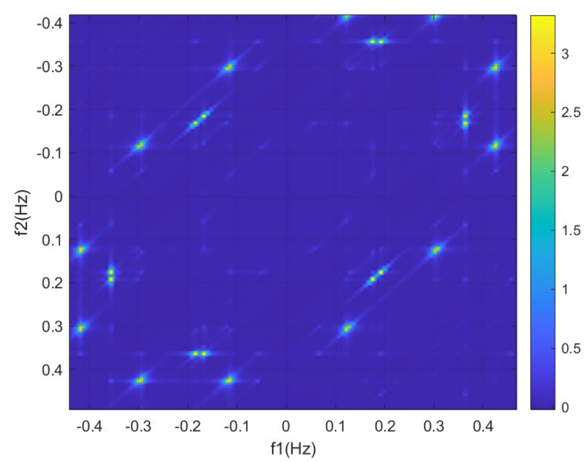


Figure 13: Direct method.

Comparing the results obtained for different segment lengths, we observed notable differences in the characteristics of the bispectrum. With shorter segments (K = 16 and M = 512), the spectral resolution improved, enabling the detection of finer frequency components. However, this improvement came at the expense of decreased spectral smoothing and increased spectral leakage, leading to higher variability in the estimated spectra. Conversely, longer segments (K = 64 and M = 128) provided smoother spectra with reduced spectral leakage, but sacrificed resolution.

2.7.a Multiple Realizations

Next, we examined the impact of considering multiple realizations of the signal X[k]. We aimed to assess the variability and robustness of the estimated spectra by generating and analyzing 50 results.

Comparing the averages of the power spectrum and bispectrum over the 50 realizations reveals a reduction in the variance of the estimates compared to a single realization. The average spectrum provided a more reliable representation of the signal's spectral properties and minimized the effects of random variations in individual realizations. The accuracy of the estimation improved as the mean approximated the true underlying spectrum.

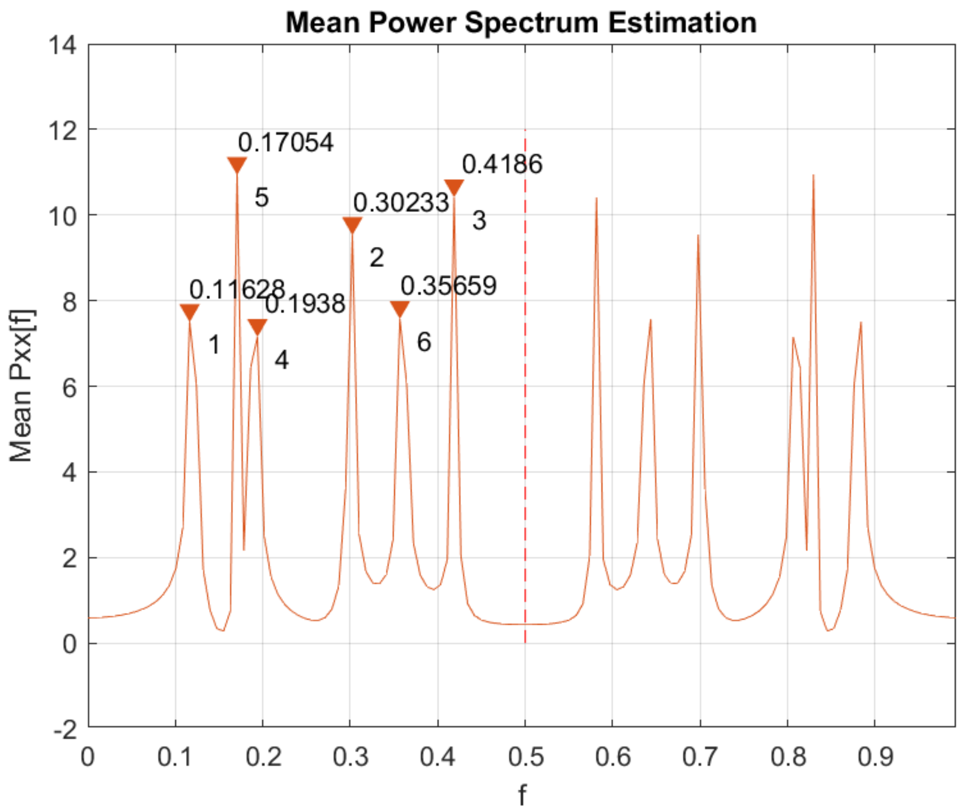


Figure 14: Mean power spectrum.

In fact, the only difference after this analysis is the values of the initial phases of the signal. The amplitude and frequency of the signal's cosines remain unchanged. However, the phase is not reflected in the power spectrum, so the average power spectrum is constant.

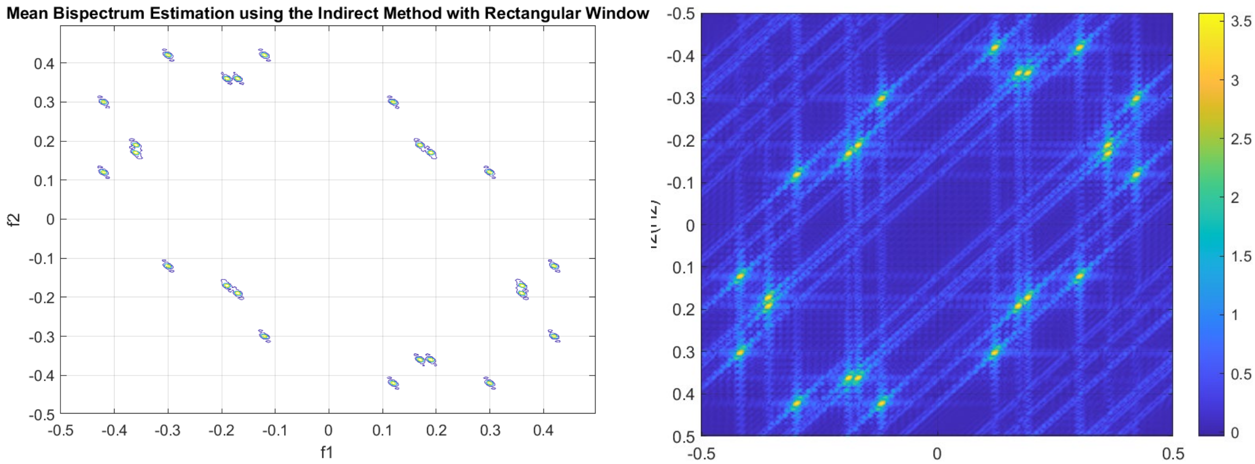


Figure 15: Indirect method with rectangle window.

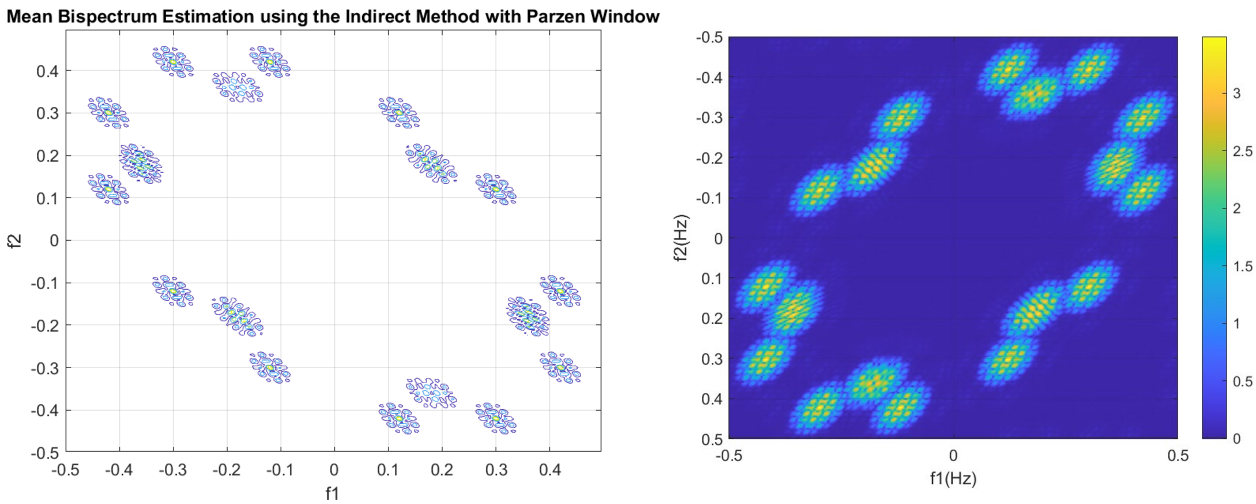


Figure 16: Indirect method with Parzen window.

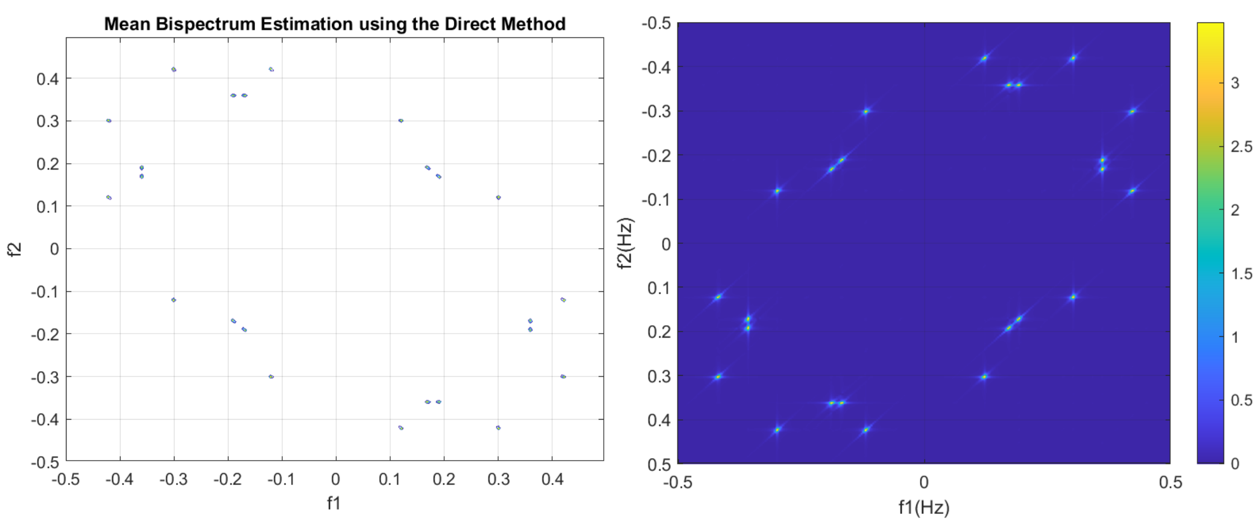


Figure 17: Direct method.