UNIVERSITÀ DEGLI STUDI DI TRENTO

Dipartimento di Ingegneria Industriale

Master Degree in Mechatronics Engineering

Synchrophasor Estimation

Systems and Techniques for Digital Signal Processing

Academy year: 2018/2019

Students:
Donato d'Acunto ID: 198393
Riccardo Maistri ID: 198392

Professor: David Macii

Contents

Abstract

In this report there will be explained how a Synchrophasor is estimated. A synchrophasor is defined as the phasor of a voltage waveform at a given reference time. All the project is developed in Matlab by Donato d'Acunto and Riccardo Maistri.

1 Synchrophasor

Synchrophasors are numbers that are time-synchronized for accuracy and represent the magnitude and phase angle of a sine wave. They are measured by high-speed monitors called Phasor Measurement Units (PMUs). PMU measurements record grid conditions with great accuracy and offer insight into grid stability or stress. Synchrophasor technology is used for off-line engineering analysis and also for real-time operations to improve grid reliability and efficiency and lower operating costs.

2 Exercise Questions

As known, in Europe the electric voltage waveforms in low-voltage distribution grids have a nominal frequency of 50 Hz and a root mean square (RMS) amplitude of 220 V. A synchrophasor is defined as the phasor (namely the magnitude and the phase) of a voltage waveform at a given reference time.

1. Assume to collect windows of N samples (N being a generic odd number) with a random initial phase at the reference time tr chosen in the center of each observation interval (the sampling rate can be set equal to fs= 2.4 kHz). Using both the plain DFT (not the FFT) and the coherent sampling (with a sampling period of your choice) estimate the synchrophasor at tr. Evaluate (and try to minimize) the magnitude and phase error at 50 Hz.

2. Assume that the nominal waveform is corrupted by a second- and third-order harmonic equal to 10% of the nominal amplitude and by a random noise with a SNR= 50 dB. Estimate mean value and variance of the estimation errors in such conditions.

3. Check what happens in the case of noncoherent sampling and if you change the number of observed cycles. Compare and comment the results.

4. Repeat steps 1-3 using the Kaiser window with different values of parameter . Compare the results.

3 Data

In the following table are summarized all the data built for this computation.

Data	**Unit**	**Value**
f_0	[Hz]	50
f_s	[kHz]	2.45
RMS	[V]	220
N	-	98
M	-	113
L	-	161
ϕ	**[rad]**	5.2
SNR	dB	50

Table 1: Overview table of Data.

The waveform is designed using the following formula:

$$x(n) = A \cdot sin(\frac{2\pi f_0}{f_s} \cdot n + \phi) \tag{1}$$

Where A is the amplitude of the sinusoidal function and it is defined as:

$$A = \sqrt{2} \cdot RMS$$

n is defined in the interval $[0, L]$ where L is the window of records. The last 2 variables are N and M that are the number of samples respectively for the coherent sampling and for the non coherent sampling, they were calculated as follows:

$$N = \frac{2f_s}{f_0}$$

$$M = \frac{2.3f_s}{f_0}$$

With the explained data the waveform is created and the discrete sinusoidal wave is displayed in the following figure.

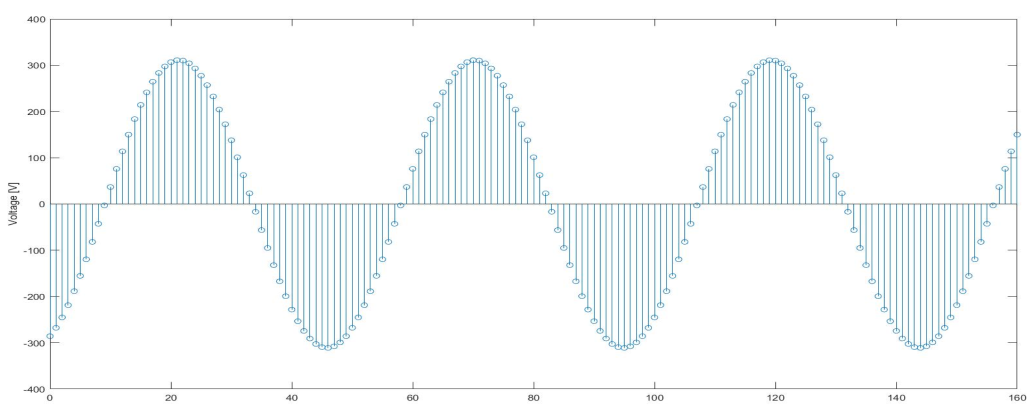


Figure 1: Waveform 220 RMS

4 DFT computation in with coherent and non-coherent sampling

To answer the first question at first the dft is computed both in the coherent sampling case and in the non coherent sampling one. Starting from the first case, the dft is computed on the signal x_1 over N samples and on the signal x_2 over M samples. The dft is defined as follows:

$$\sum_{k=0}^{N|M-1} x(k)e^{-\frac{j2\pi}{N}\cdot kn}$$

The results of the dft computations for the 2 cases are shown below.

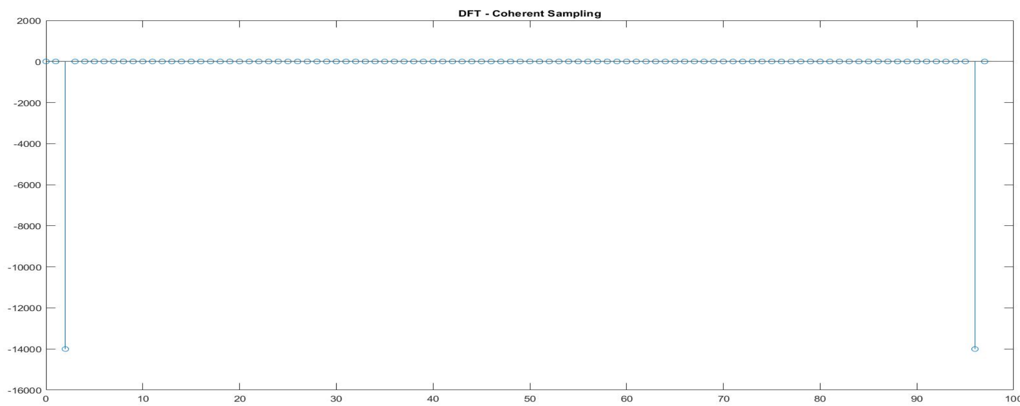


Figure 2: Dft computed on x(n) with Coherent Sampling over N

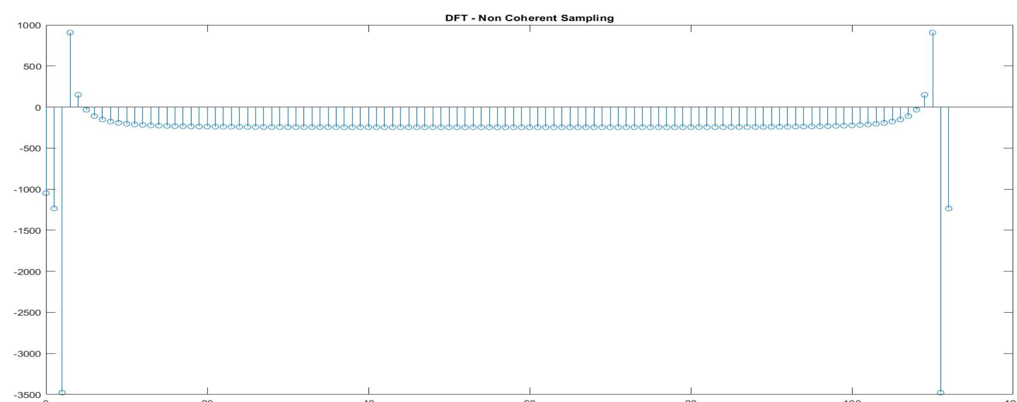
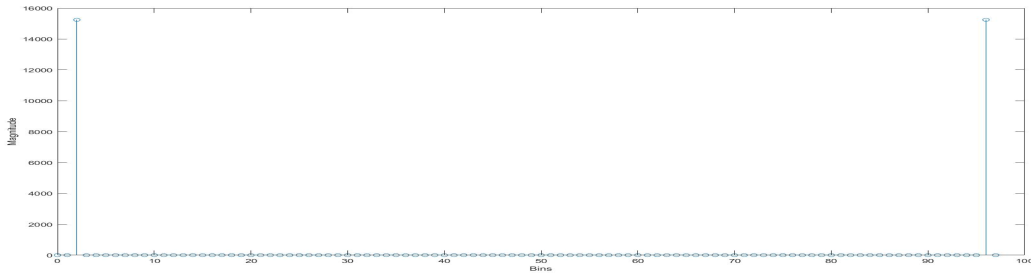
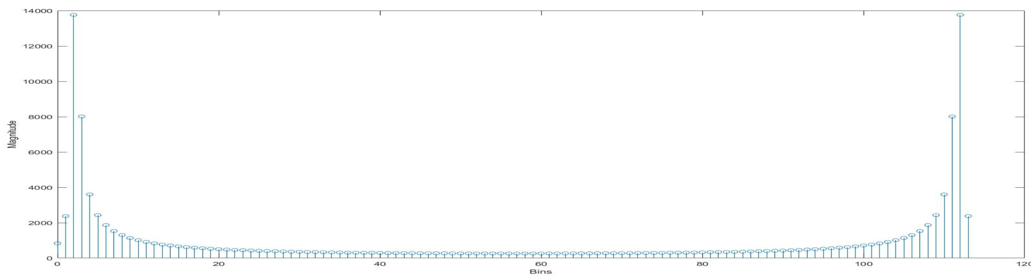


Figure 3: Dft computed on x(n) with Non Coherent Sampling over M

The difference between the 2 cases is clear, in the first the sampling is perfect on the zeros of the signal and on the 2 peaks of the sinc function while in the second case the signal is not sampled in the correct way, so would be losses of information.
After the computation of the dft the Magnitude and the Phase of the signal are calculated even this time for both cases. Magnitude and phase for both coherent and non coherent sampling are shown in the figures below:

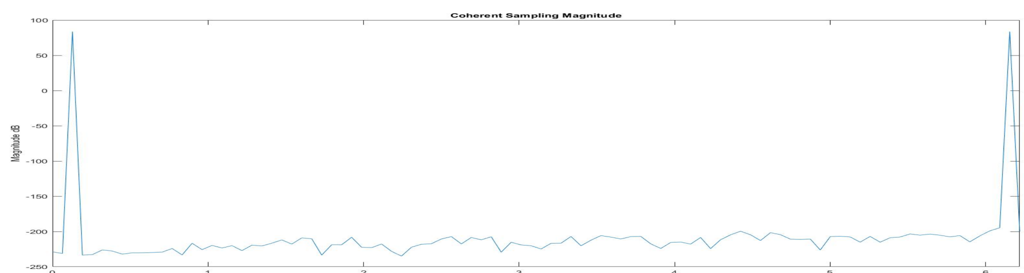


(a) Coherent Sampling

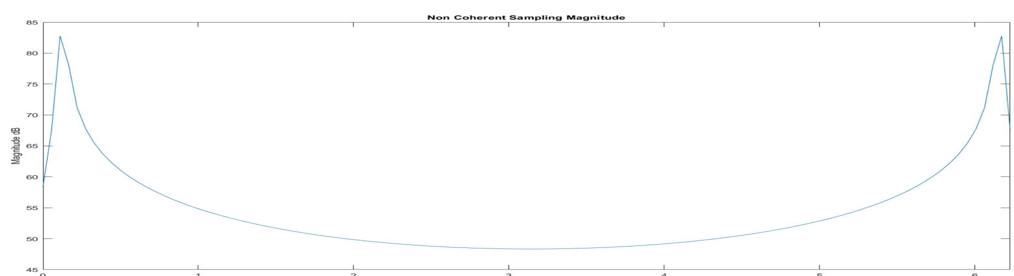


(b) Non Coherent Sampling

Figure 4: Magnitude linear in both cases



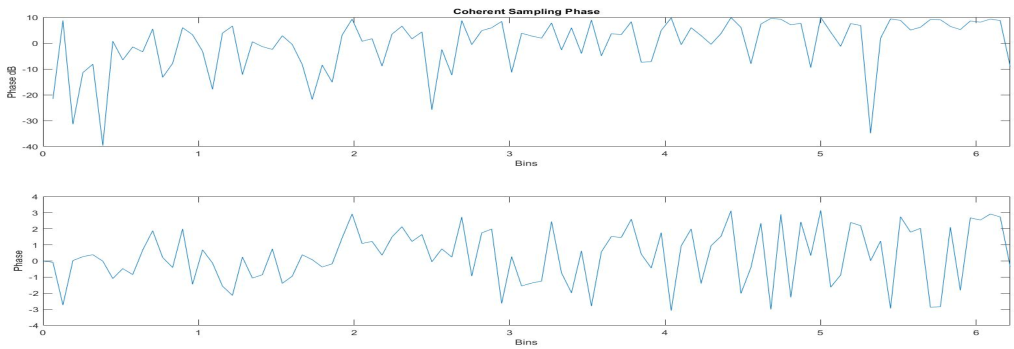
(a) Coherent Sampling



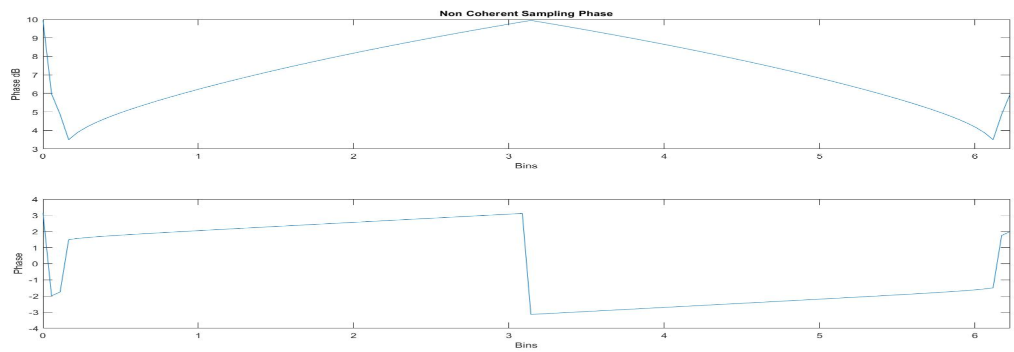
(b) Non Coherent Sampling

Figure 5: Magnitude logarithmic scale in both cases

The syncrhophasor at time t_r, which is the time taken in the middle of the observed window, is given from the calculations of the magnitude and phase in the two cases, and are collected in the following table.
As we can see from the plots and from the results the coherent sampling case is very close to the ideal sampling, but there is a small error, which is possible to minimize trying to improve the coherent sampling.



(a) Coherent Sampling



(b) Non Coherent Sampling

Figure 6: Phase in both cases

	x_1	x_2
Magnitude	$5.7 \cdot 10^{-12}$	252.2
Phase	-1.4	2.9
Magnitude dB	-224.9	48.03
Phase dB	$2.7 + j27$	9.3

Table 2: Syncrhophasor Results

Anyway the order of the error is about 10^{-12} so really low. The non coherent sampling case presents different results as shown in the figures and in the results, this problem can be partially solved by increasing the number of samples, but anyway the spectral leakage does not decrease, as we can see in the following figures where M is increased to 2M. It is evident that the problem isn't solved only increasing M. The spectral leakage is caused by the sidelobes of the replica that tend to infiltrate the main lobe, this is due to the side lobes of the non coherent sampling case that are larger than in the other case.

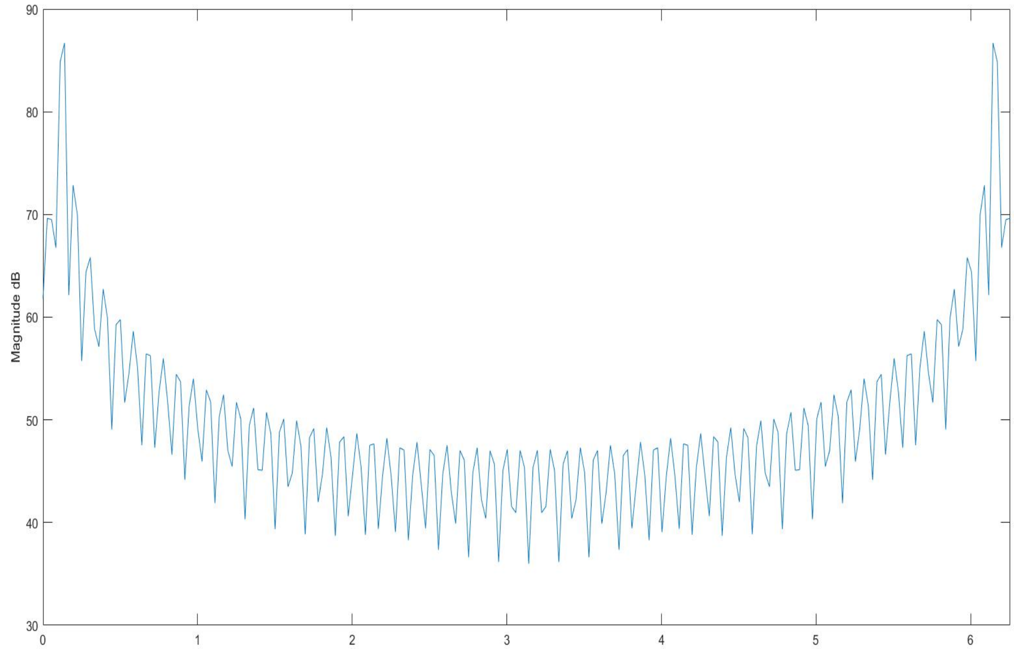


Figure 7: Magnitude in logarithmic scale Non Coherent Sampling with M = 2*M

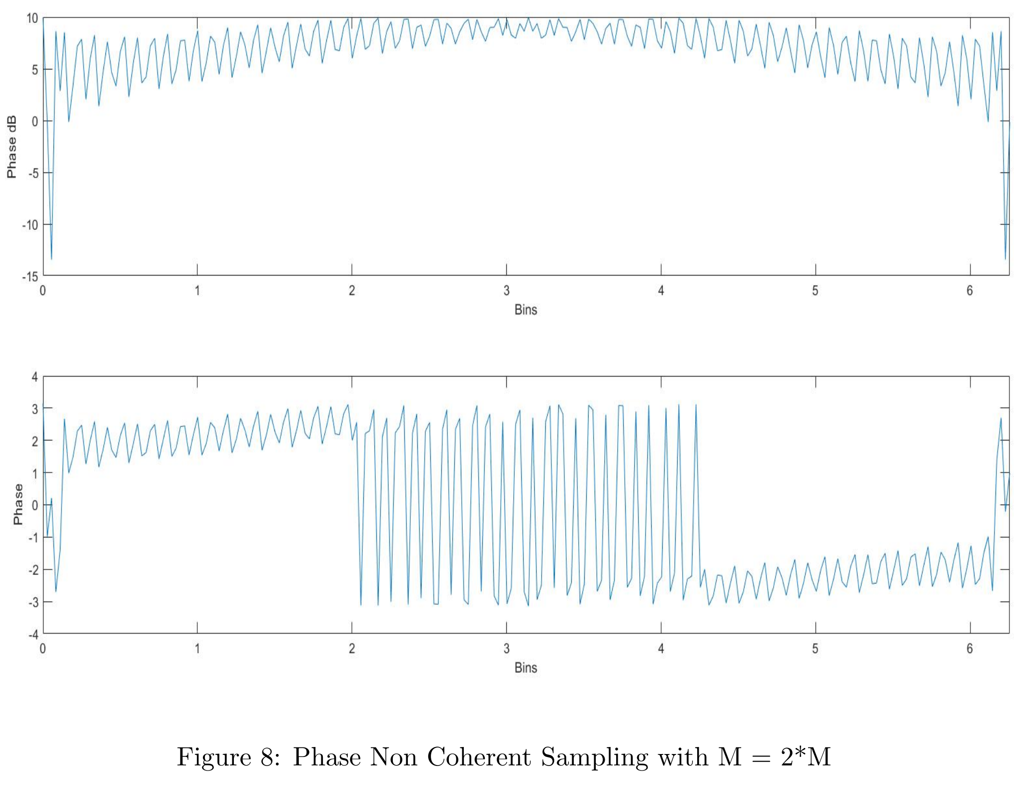


Figure 8: Phase Non Coherent Sampling with M = 2*M

5 Corrupting the signal

The signal is going to be corrupted by 3 sources of noise, let's define these noises as follows.

- Second-order harmonic:
$$A_1 \cdot sin(\frac{2\pi f_0}{f_s} \cdot 2n + \phi)$$
- Third-order harmonic:
$$A_1 \cdot sin(\frac{2\pi f_0}{f_s} \cdot 3n + \phi)$$
- Random noise: in matlab the function randn() is used, it returns a matrix of normally distributed random numbers, so this specific noise is defined as
$$\sqrt{10^{\frac{SNR}{20}}} \cdot randn(L)$$

A_1 is the amplitude of the harmonics defined as the 10% of the original amplitude of the sinusoidal signal and SNR is 50 dB as defined at the beginning. The total corrupted signal is given by the sum of these 3 noise sources and the original signal.

$$x_{corrupted} = x(n) + rand_{noise}(n) + x_{HarmII}(n) + x_{HarmIII}(n)$$

In the following figures the noises and the output corrupted signal are shown.

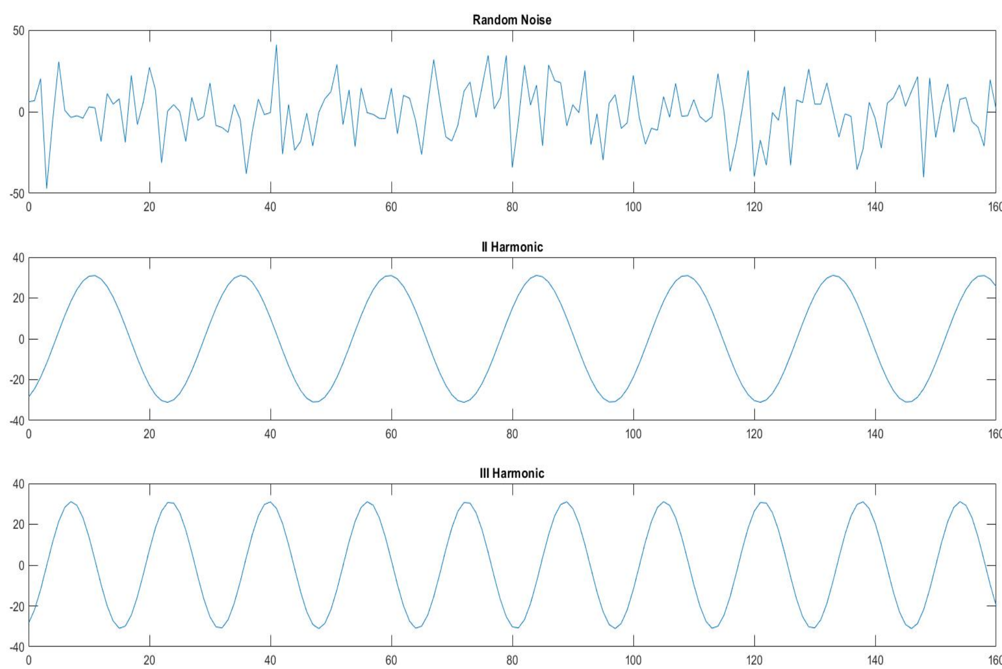


Figure 9: The noises

The mean and the variance are calculated on the total noise computed by the summation of the 3 noise sources which is shown in the figure below. The values are calculated through the functions *mean* and *var* in matlab, they gave the following results.

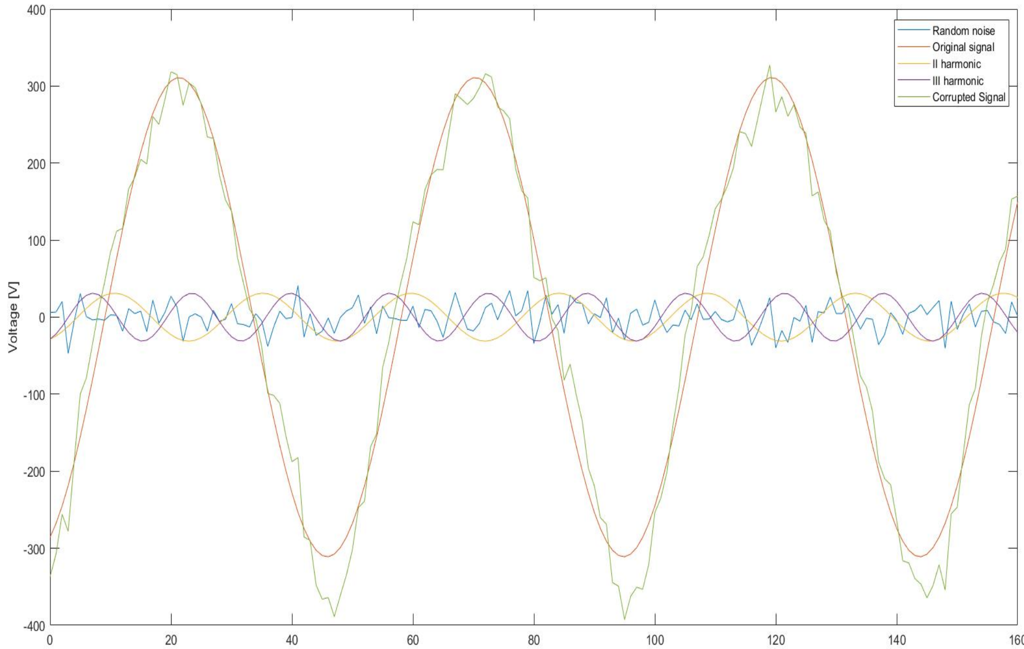


Figure 10: The signal input, the output and the noises

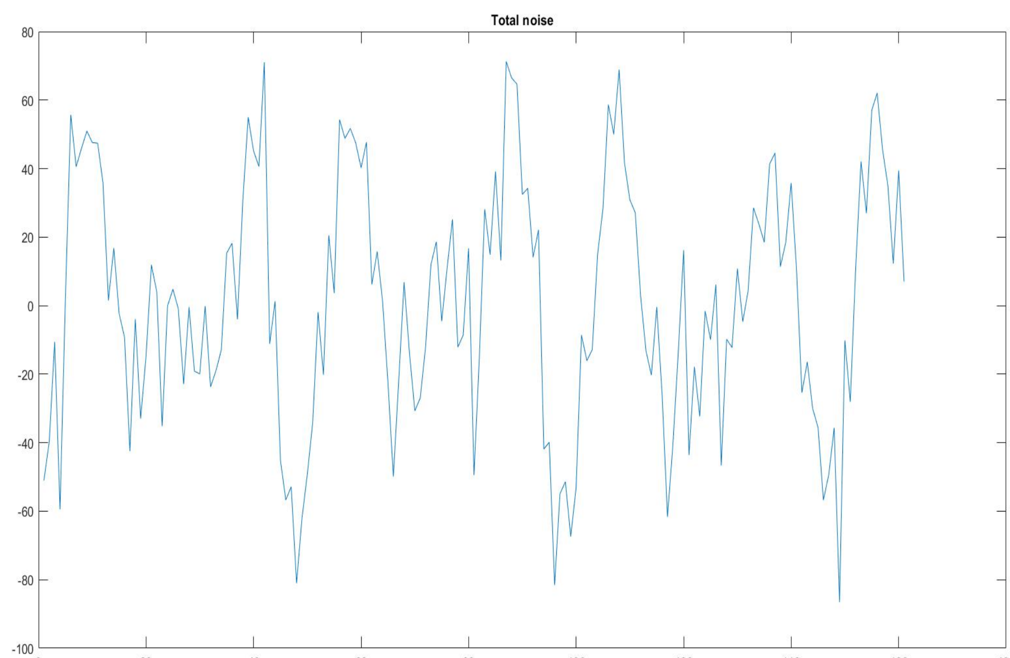


Figure 11: The signal input, the output and the noises

	x_1	Random Noise	Hamonic II	Harmonic III
Mean	0.12	-1.01	0.71	0.43
Variance	$1.2 \cdot 10^3$	258.2	489.3	481.2

Table 3: Syncrhophasor Results

6 Kaiser Windowing

The aim of the second part of the project is to repeat all the work done in the first steps but using kaiser window changing parameters and see the changes in behavior. Kaiser windowing method is based on a special type of window function which tries to improve the windowing method using a parametric window function. Kaiser parametric window function is defined as:

$$w_k(n) = \frac{I_0(\beta(1 - (\frac{2n-M}{M})^2)^{\frac{1}{2}})}{I_0(\beta)}; 0 \leq n \leq M$$

Where β, M and I_0 are the parameters defining the Bessel function of first order.

Implementing Kaiser window in matlab is very easy, it is computed with the function kaiser(L,beta) which returns an L-point Kaiser window with shape factor beta. An example of the input sinewave windowed with the kaiser window and a generic β is provided below.

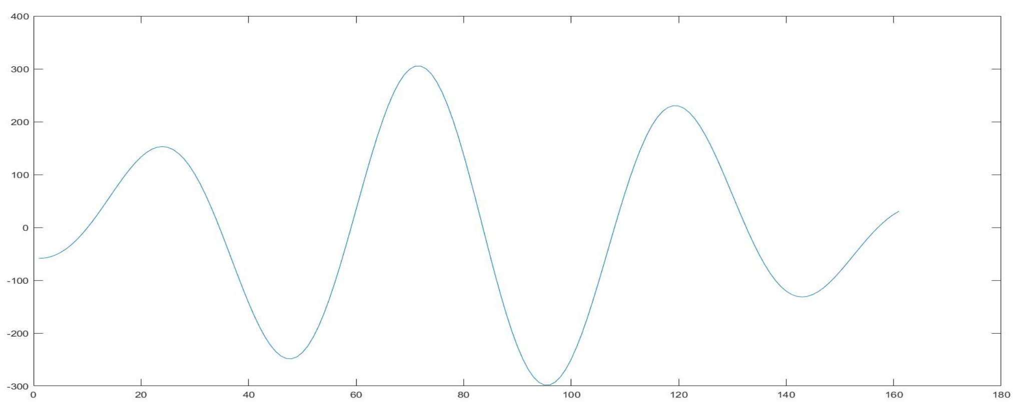


Figure 12: The original sinewave windowed with a generic kaiser window

As we can see the side lobes of the sinewave are attenuated, this is due to the windowing operation. Let's see how the signal changes tuning the β parameter. β is taken in the interval [1:10] each step to clearly see the differences in the output plots.
At first the Kaiser windows are shown in figure 13, is clearly visible how the width of the window becomes narrower and narrower increasing β.
In figure 14 is possible to see how the side lobes of the original sinewave become smaller and smaller increasing β.
Now the dft is computed, as at the beginning, for the coherent sampling case and the non coherent sampling one. Then the magnitude and the phase are computed in the same way. They are shown in figure 15-16.
As we can see the magnitude is strongly affected by the changing of the window while the phase responses isn't affected so much, the behavior is more or less the same both in the coherent sampling case and in the non coherent one. Focusing on the Magnitude the difference between the 2 cases is interesting, in fact while in the first case increasing β the error increases significantly in the non coherent sampling case increase β means an improvement, the attenuation of the error of the magnitude.

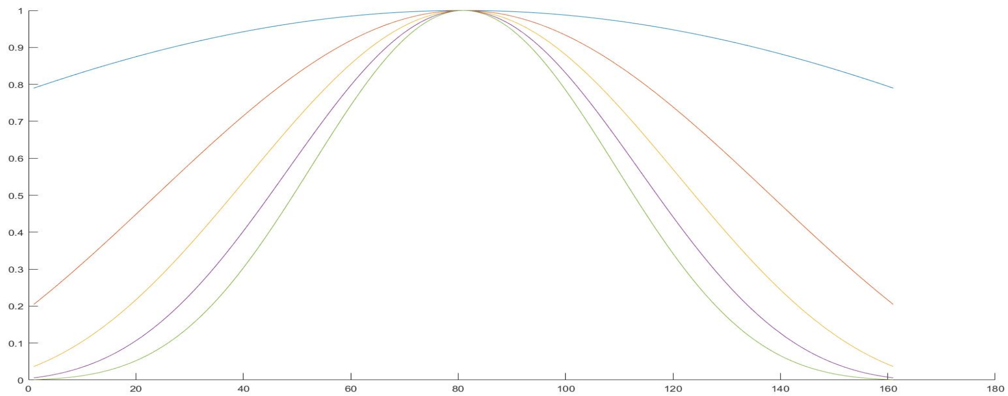


Figure 13: Kaiser windows changing β

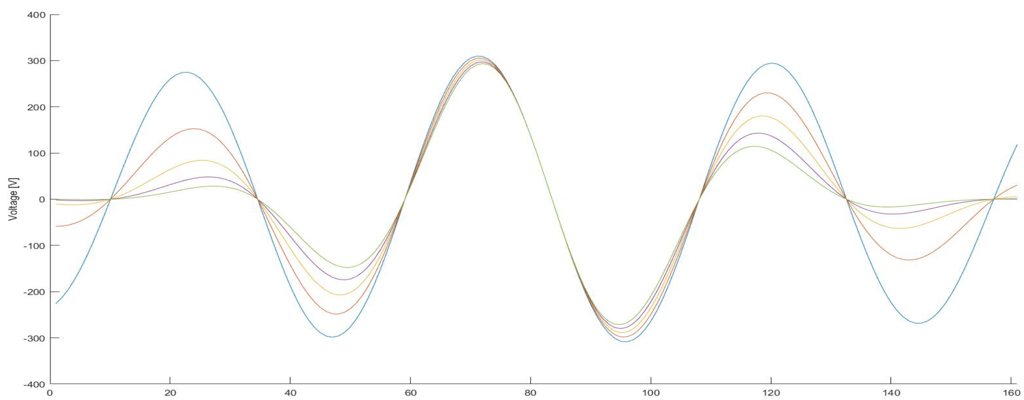


Figure 14: Original signal windowed changing β

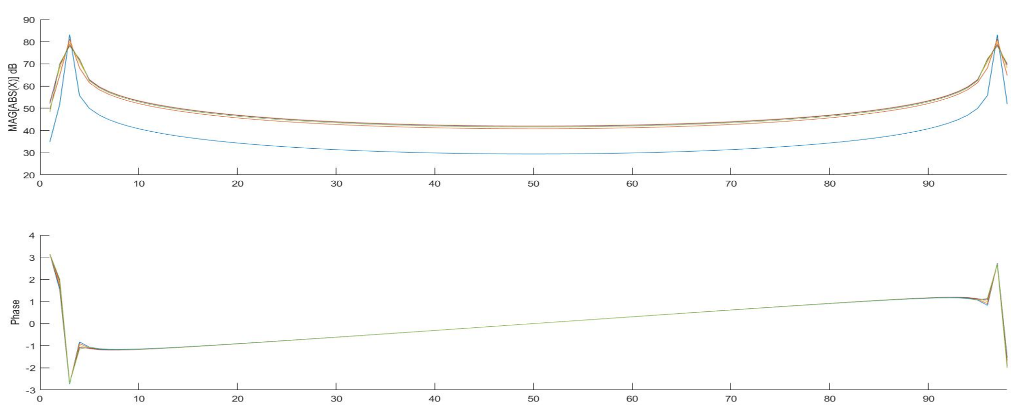


Figure 15: Magnitude and phase of the windowed signal changing β with coherent sampling

Also in this case we tried to increase the M number which became 2M and the results of the magnitude and phase are shown in figure 17. Here the behavior is the same described before for the non coherent sampling,

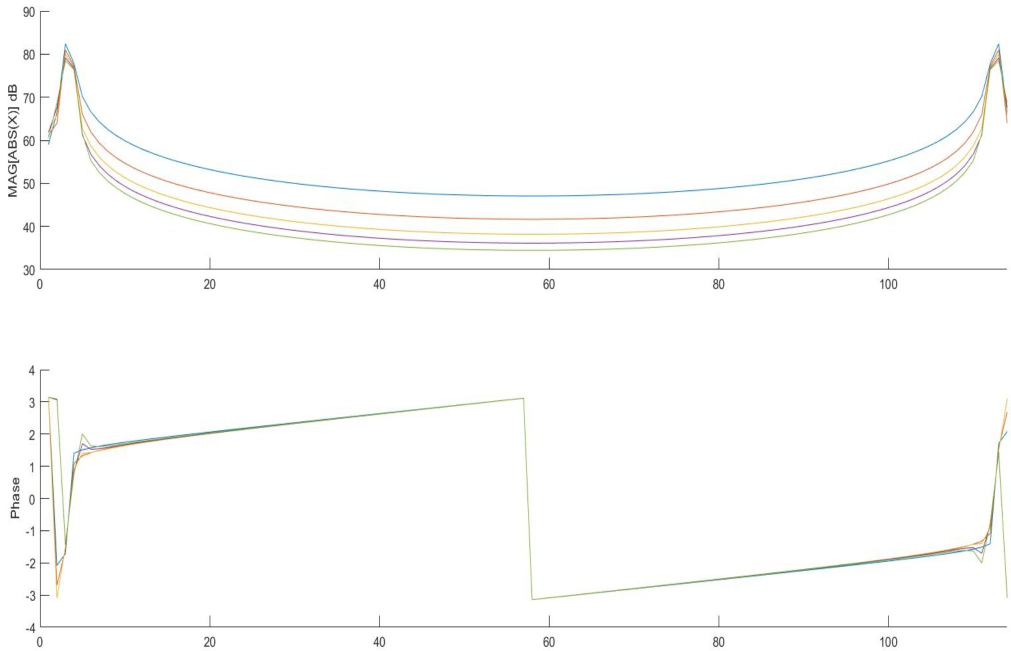


Figure 16: Magnitude and phase of the windowed signal changing β with non coherent sampling

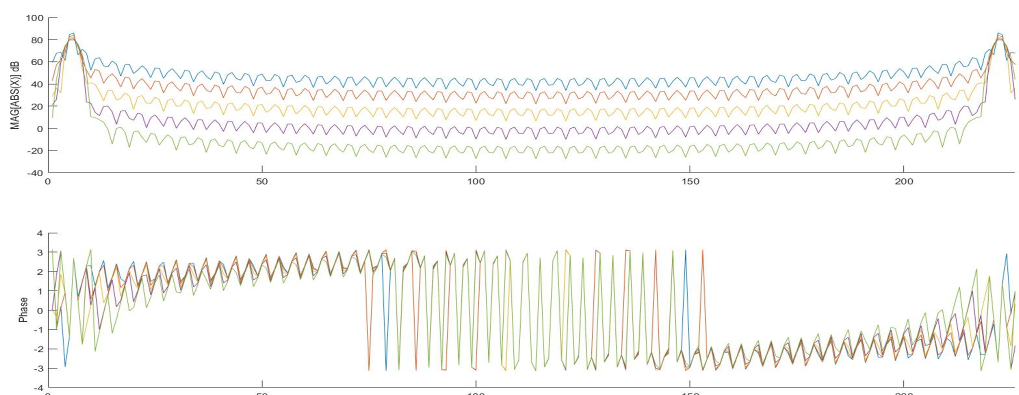


Figure 17: Magnitude and phase of the windowed signal changing β with non coherent sampling and M = 2M

with the only difference, as in the first part, that the spectral leakage is really high, figure 17.
A solution to this problem would be increase the β parameter. In fact is known that the choice of the type of window affects the spectral leakage amount, but at the same time also the spectral resolution. The spectral resolution is defined as the capability to distinguish 2 sequences, so the minimum frequency interval over which it's possible to distinguish 2 signals. Anyway increasing β we pass from windows similar to the rectangular window to types similar to the cosine class windows, with lower β is possible to obtain a really good spectral resolution, but an high spectral leakage, while with higher β the spectral leakage tends to decrease and so the effect of the ripples become less important, but at the same time the spectral resolution grows. So it is important to find a good trade off between these 2 phenomena.

7 Corrupting the windowed signal

As in the first part, in this section the windowed signal will be corrupted by 3 sources of noises. The 3 noise sources are the same as in the first part of the project but with the window applied to each of it minus the random noise:

- Second-order harmonic:
$$A_1 \cdot sin(\frac{2\pi f_0}{f_s} \cdot 2n + \phi) * w(n)$$
- Third-order harmonic:
$$A_1 \cdot sin(\frac{2\pi f_0}{f_s} \cdot 3n + \phi) * w(n)$$
- Random noise: in matlab the function randn() is used, it returns a matrix of normally distributed random numbers, so this specific noise is defined as
$$\sqrt{10^{\frac{SNR}{20}}} \cdot randn(L)$$

Also in this case the results are summarize in a single plot to better see the behavior changing the β parameter. As we can see here and in the following plots (single cases are shown to better understand the

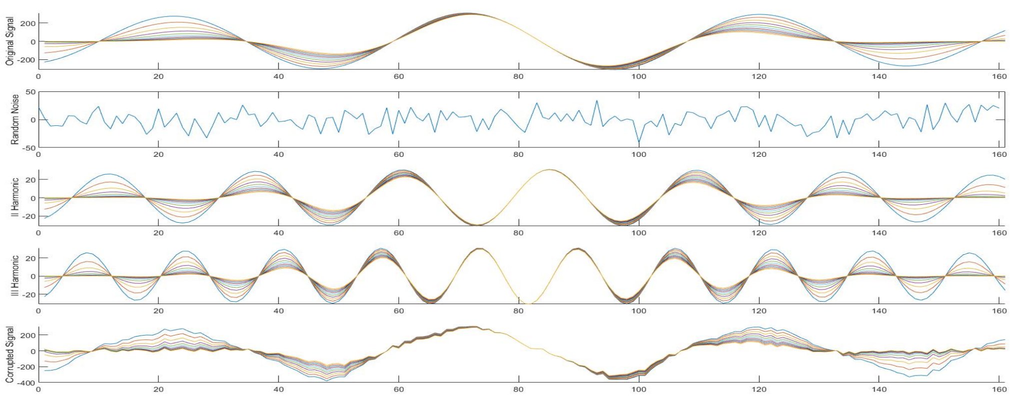


Figure 18: Noises and windowed signal changing β

behavior of the windowing effect on the noises) the windowing effect is more or less the same as in the original signal, the side lobes are smoothed increasing β.

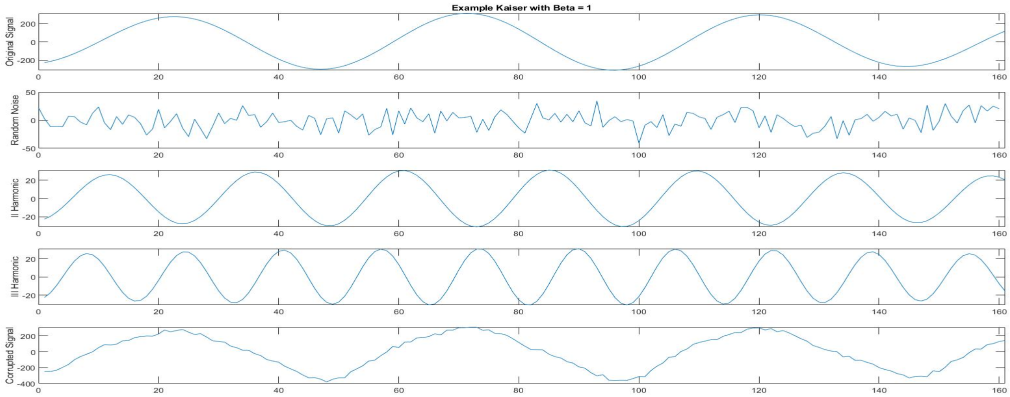


Figure 19: Magnitude and phase of the windowed signal changing β with non coherent sampling and M = 2M

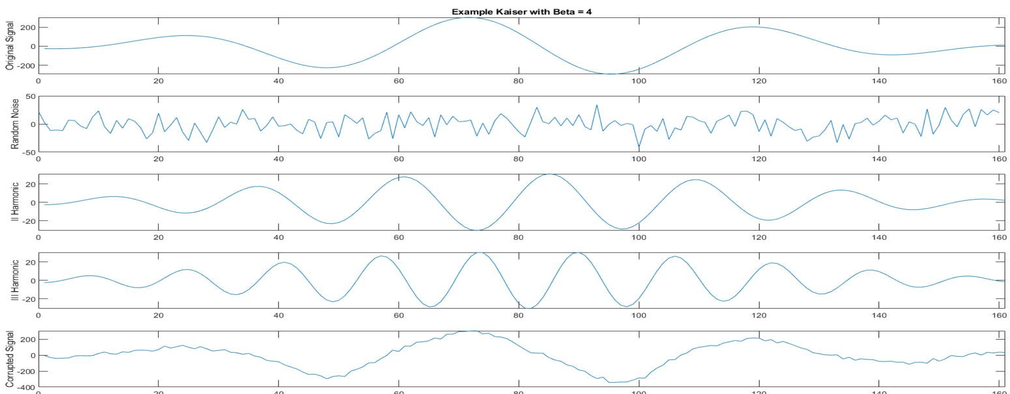


Figure 20: Magnitude and phase of the windowed signal changing β with non coherent sampling and M = 2M

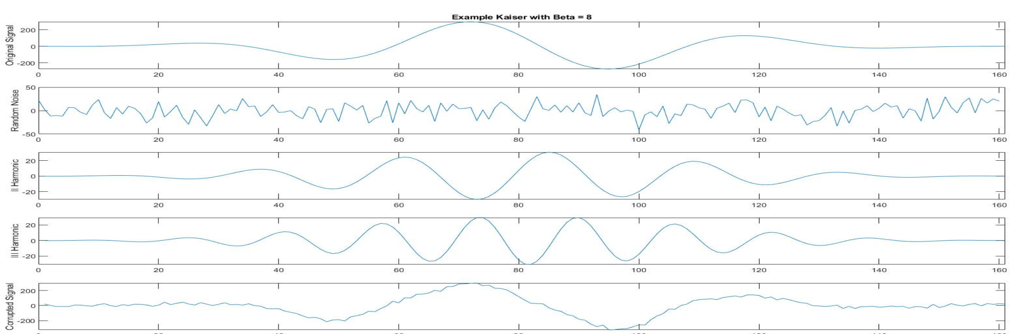


Figure 21: Magnitude and phase of the windowed signal changing β with non coherent sampling and M = 2M

8 Overview of the Matlab Code

The Code is developed entirely in Matlab in one file called DSP_Project.m and 3 functions: dft.m, printGraph.m and printGraph_2.m. The code is mainly divided into 4 parts, but before all there is a section where the variables are declared as explained in the first part of the report. Then in the first part of the code the dfts are computed both for the coherent sampling case and the non coherent sampling case. In the second part the noises are created and applied to the input signal and at the end the means and variances are computed for each type of noise. In part 3 and 4 is done the same with the only difference that the kaiser window is applied. Also in these 2 last part the procedure is iterated to see how the results change tuning the parameter of the kaiser window. The function dft.m is used to compute the dft as definition and the other 2 funtions are used to plot data and to obtain values from the magnitude and phase of the computed signals.

9 Conclusions

About the first part of the project the system's behavior is more or less as we expected, in fact the coherent sampling makes the analysis more accurate with an estimation error much lower than in the other case with respect to the ideal case. As we have seen changes in the observation interval make the analysis more accurate, more in the non coherent sampling case, but anyway does not solve the problem of the spectral leakage. In the second part the results are different because of the windowing operation which makes the input signal different and a generalized linear phase response is obtained being the filter a finite impulse response (FIR) filter.