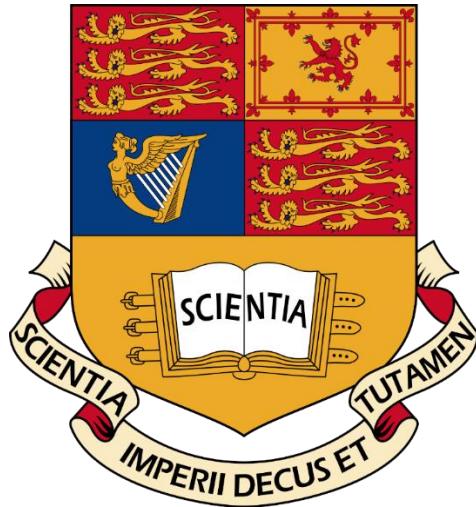


Imperial College London
EEE Department
MSc Communications & Signal Processing



Adaptive Signal Processing & Machine Intelligence

Professor: Danilo P. Mandic

Teaching Assistants: Giuseppe Calvi, Ilia Kasil, Ahmad Moniri, Takashi Nakamura

MSc Student: Ilias Chrysovergis, ic517@ic.ac.uk, 01449042

Monday 9th April 2018

Table of Contents

1.	Spectrum Estimation	3
1.1	Discrete Fourier Transform Basics	3
1.2	Properties of Power Spectral Density (PSD)	4
1.3	Resolution and Leakage of Periodogram-based Methods.....	7
1.4	Periodogram-based Methods Applied to Real-World Data	11
2.	Parametric line spectra.....	13
2.1.	Correlation Estimation	13
2.2.	Spectrum of Autoregressive Processes	17
2.3.	Principal Component Analysis.....	19
2.4.	Real World Signals: Respiratory Sinus Arrhythmia from RR-Intervals.....	21
3.	Adaptive Signal Processing	24
3.1.	The Least Mean Square (LMS) Algorithm	24
3.2.	Adaptive Step Sizes	27
3.3.	Adaptive Noise Cancellation	30
4.	Widely Linear Filtering & Adaptive Spectrum Estimation	34
4.1.	Complex LMS & Widely Linear Modelling	34
4.2.	Adaptive AR Model Based Time-Frequency Estimation	39
4.3.	A Real Time Spectrum Analyser Using Least Mean Square	40

1. SPECTRUM ESTIMATION

1.1 Discrete Fourier Transform Basics

- a) The Fourier spectrum of a 20 Hz sine wave (Fig.1 a) is a Dirac function at 20 Hz since the sine wave is a periodic function with only one frequency. The application of a rectangular window (multiplication) into the sinewave in the time domain is translated into convolution with the sinc function in the frequency domain. Since convolution with the Dirac function only shifts the convolved function according to the Dirac's frequency, the theoretical DTFT Spectrum of a rectangular-windowed sine wave is two sinc functions in -20 Hz and 20 Hz (Fig.1 b).

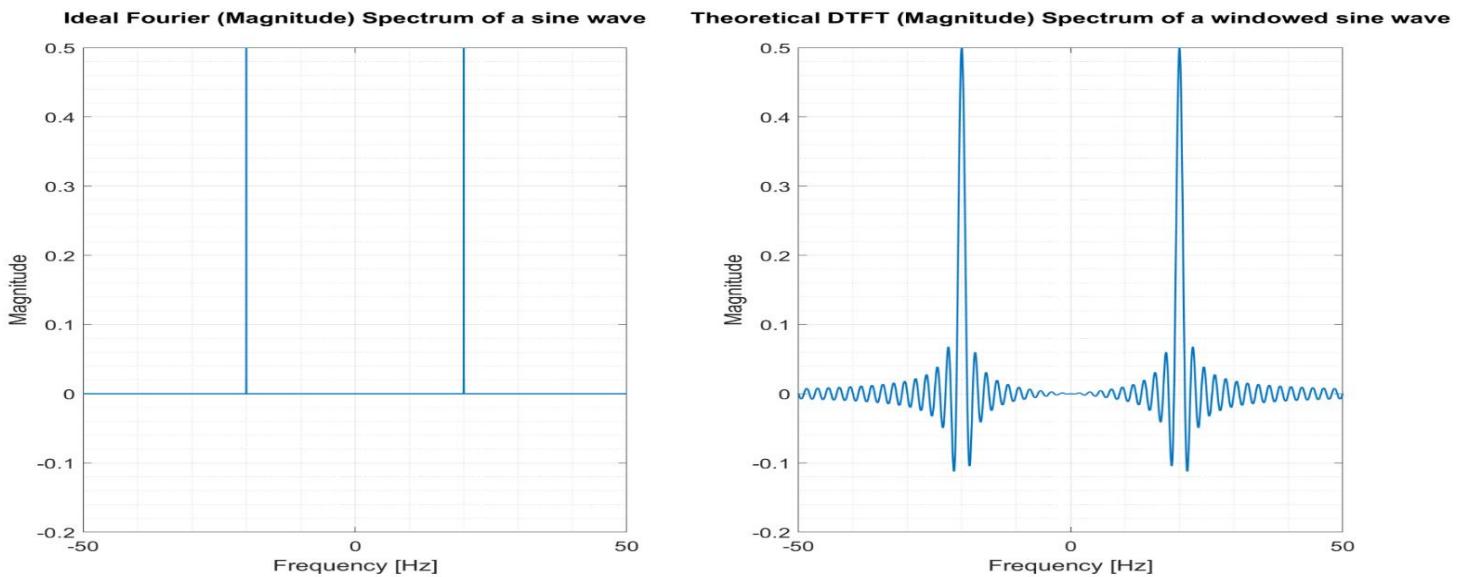


Figure 1: The Ideal Fourier Spectrum of a sine wave and the DTFT Spectrum of a rectangular windowed sine wave.

- b) The $K = 100$ -point DFT (Fig.2 a) of the $N = 100$ -point 20 Hz sine wave is equal to the ideal Fourier spectrum of a 20 Hz sine wave (Fig.1 a), because the sampling is coherent and therefore all the zeroes of the sinc functions lie at multiples of $1/N$, and hence the outputs of the DFT are all zero except at $f = \pm \frac{1}{N}$. In Fig.2 b zero-padding is used and therefore the signal length and the frequency resolution have increased. However, the use of zero-padding has introduced rectangular windowing in the time domain and therefore the $K = 1000$ -point DFT of a $N = 100$ -point 20 Hz sine wave looks like the theoretical DTFT Spectrum of a rectangular-windowed sine-wave. As $K \rightarrow \infty$ the K -point DFT will be equal with the theoretical DTFT spectrum of a windowed sine wave.

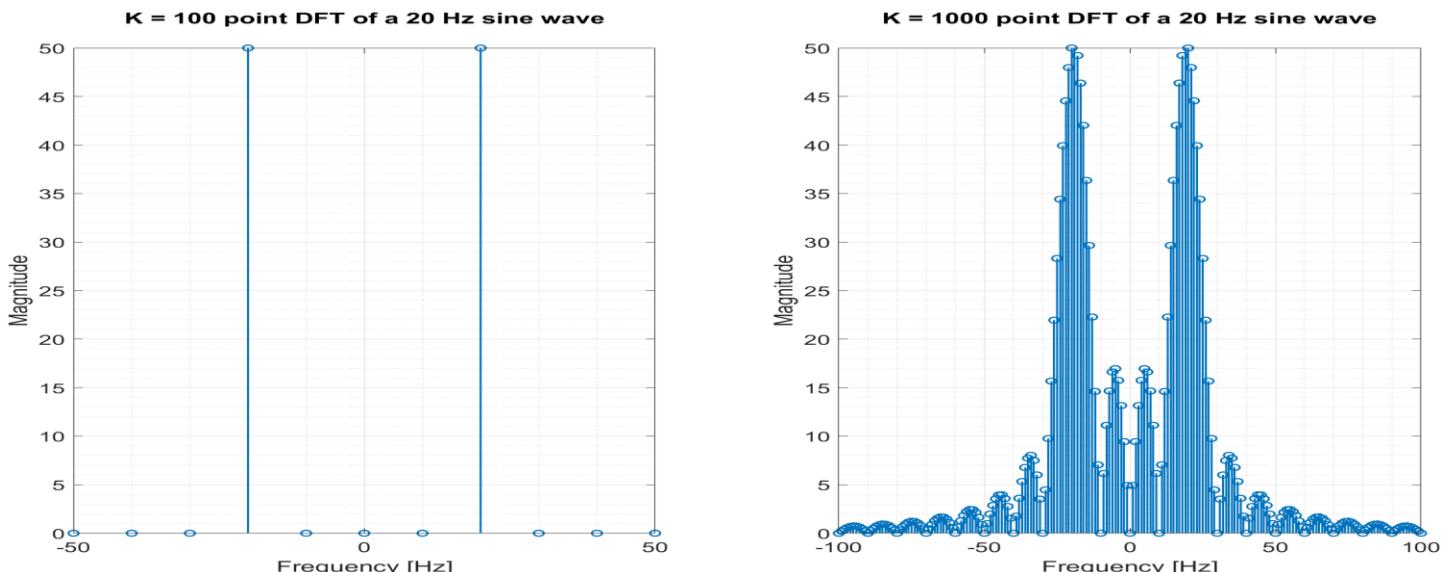


Figure 2: The $K = 100$ and $K = 1000$ DFT spectra of a 20 Hz sine wave

- c) When a 24 Hz sine wave is generated with $N = 100$, the phenomenon of spectral leakage can be observed. This happens because the length of the sine wave is not a multiple of its' period. Therefore, if the Fourier transform is applied without zero padding, a clear peak at 24 Hz will not be present in the spectrum (Fig.3 a). If zero padding is applied the peak is distinguishable, but this is not a solution to incoherent sampling and therefore to spectral leakage if we do not want to increase the length of the signal. In order to solve that problem, the length of the sine wave should be a multiple of its' period. This can be realized by applying a window into the sine wave of length which will be a multiple of its' period.

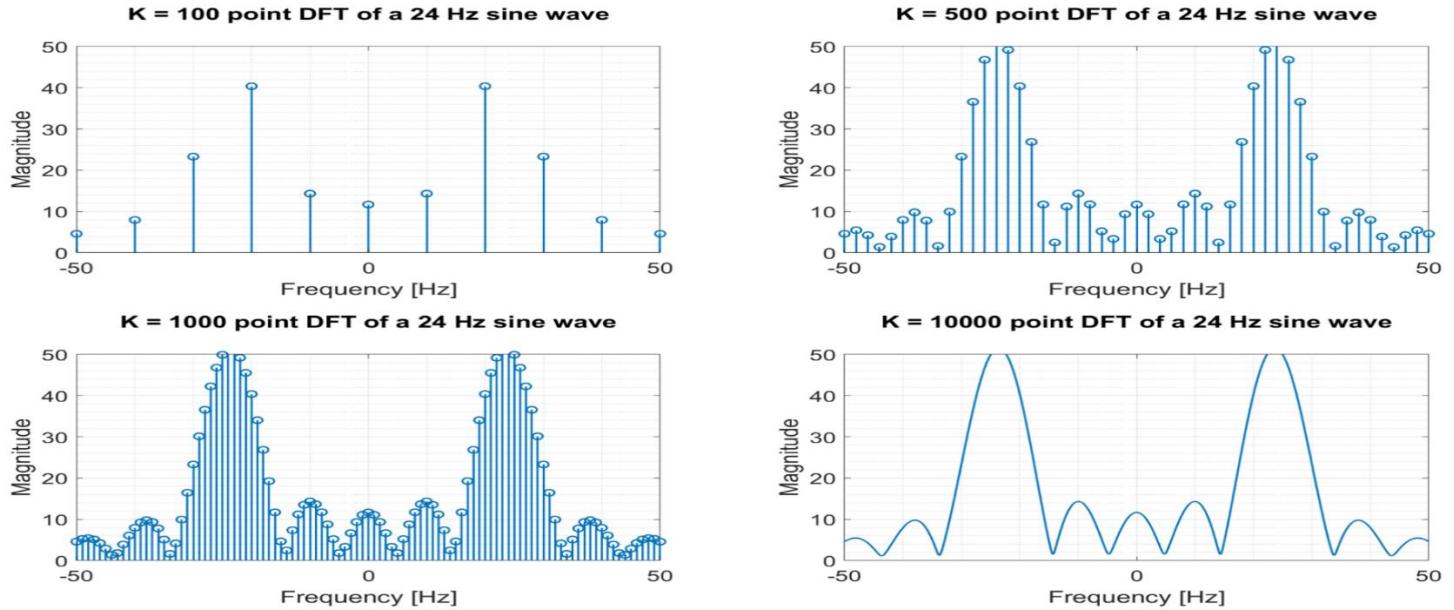


Figure 3: The $K = 100$, $K = 500$, $K = 1000$ and $K = 10000$ DFT spectra of a 24 Hz sine wave

1.2 Properties of Power Spectral Density (PSD)

- a) Zero-padding of a signal leads to increased frequency resolution. When $M = 10$ zero padding has positive effects since it leads to smoothing of the spectrum of the signal as seen in Fig. 4. The particular choice of x is very useful since its Fourier transform is equal to the zero-padded power spectral density (periodogram) of the autocovariance function (ACF) $r(k)$. Therefore, that choice of x gives us the ability to calculate the periodogram of the ACF without having to think how to introduce zero-padding to increase the frequency resolution. Also, this choice makes a symmetric signal, a very convenient property when dealing with Fourier transforms. In Fig.4 this claim is clear.

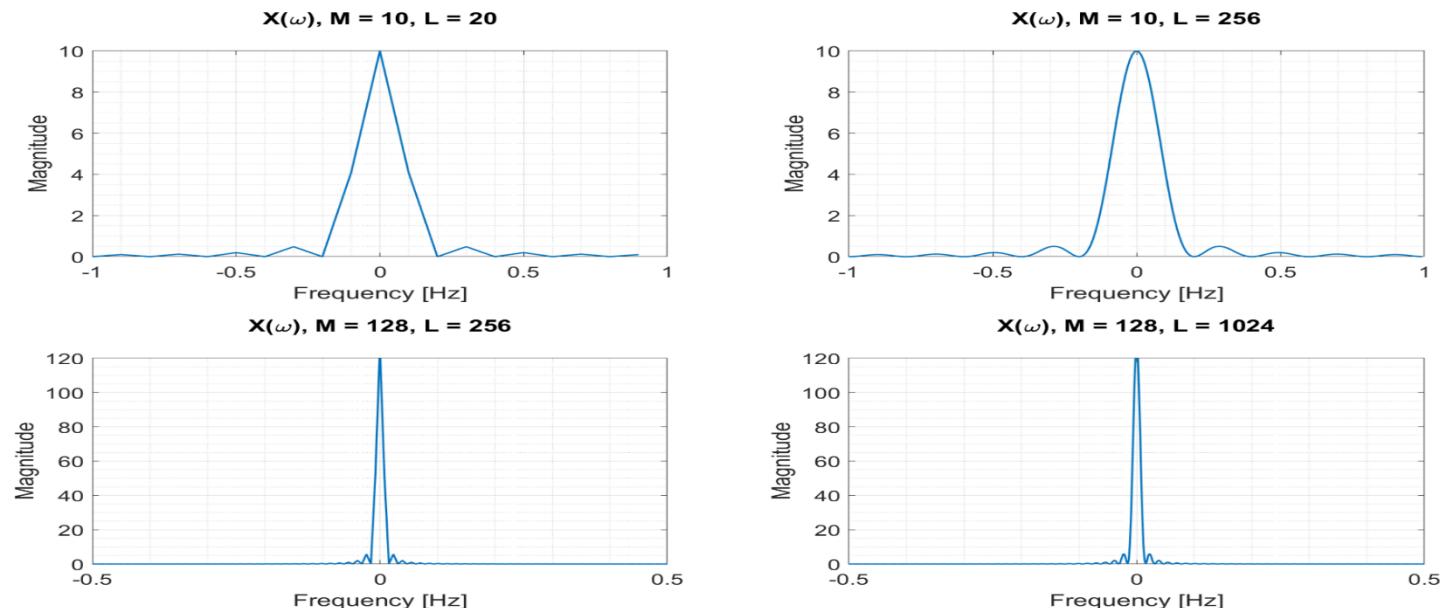


Figure 4: The zero padding properties of the DTFT of $x(k)$ generated by $r(k)$.

- b) Fig. 5 shows the negligible imaginary part of the spectrum of \mathbf{x} . Since the signal is real and symmetric also its' Fourier transform is real and symmetric. That small imaginary component has generated due to very small round-off errors. Therefore, we need to take the real value of the Fourier spectrum of \mathbf{x} because matlab will encounter problems when we want to plot it, since its' value will be complex otherwise.

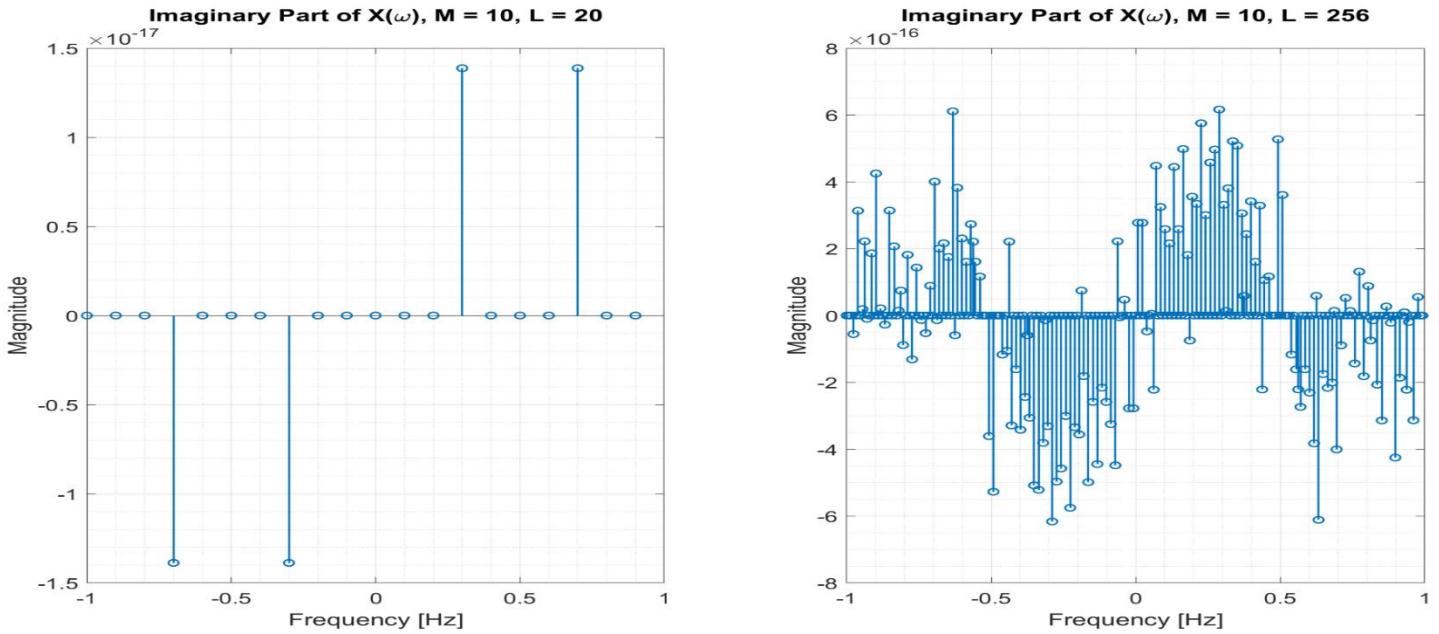


Figure 5: The negligible imaginary part of $X(\omega)$ with and without zero padding.

- c) Fig. 6 shows the real, imaginary and magnitude of $Z(\omega)$ (a mistake has been made while plotting the diagrams and $X(\omega)$ is used in the title falsely). That choice of \mathbf{z} leads to erroneous “spectral” values when $\text{real}(\text{fft}(\mathbf{z}))$ is used, since the imaginary part is not negligible now. This happens because that choice does not zero-pads correctly the autocovariance function and therefore the signal is not symmetric now. Thus, the imaginary part is not insignificant now and $\text{abs}(\text{fft}(\mathbf{z}))$ should be used.

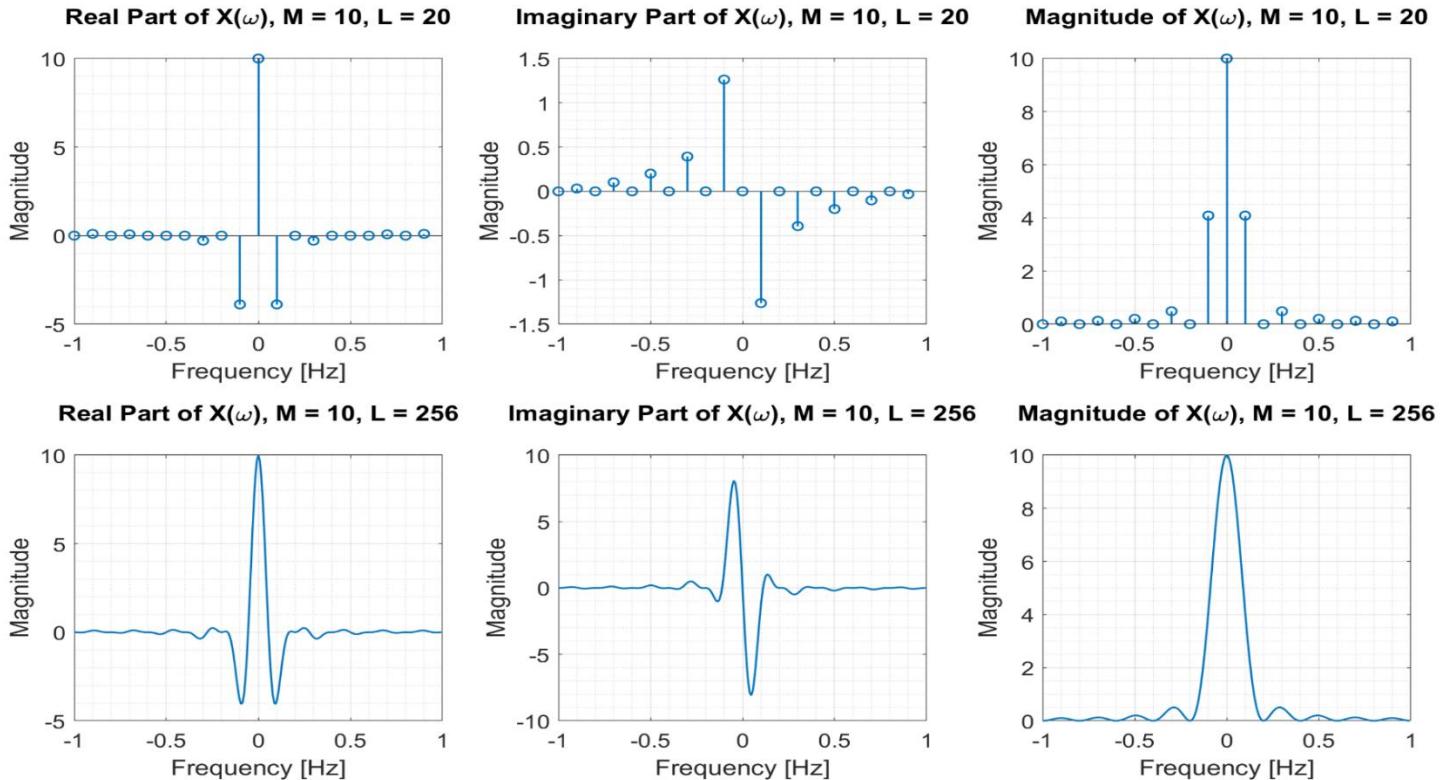


Figure 6: The Real and Imaginary as well as the Magnitude of $Z(\omega)$ with and without zero padding.

- d) The use of the `fftshift` command is very important for convenient plotting in matlab. In Fig.3 a and c the command has not been used. This leads to very inconvenient insights when plotting the Fourier transform or the autocovariance function. The use of the command (Fig.3 b and d) gives the plots that everyone is used to, where the zero frequency and time lag are centered, and the magnitudes are presented correctly.

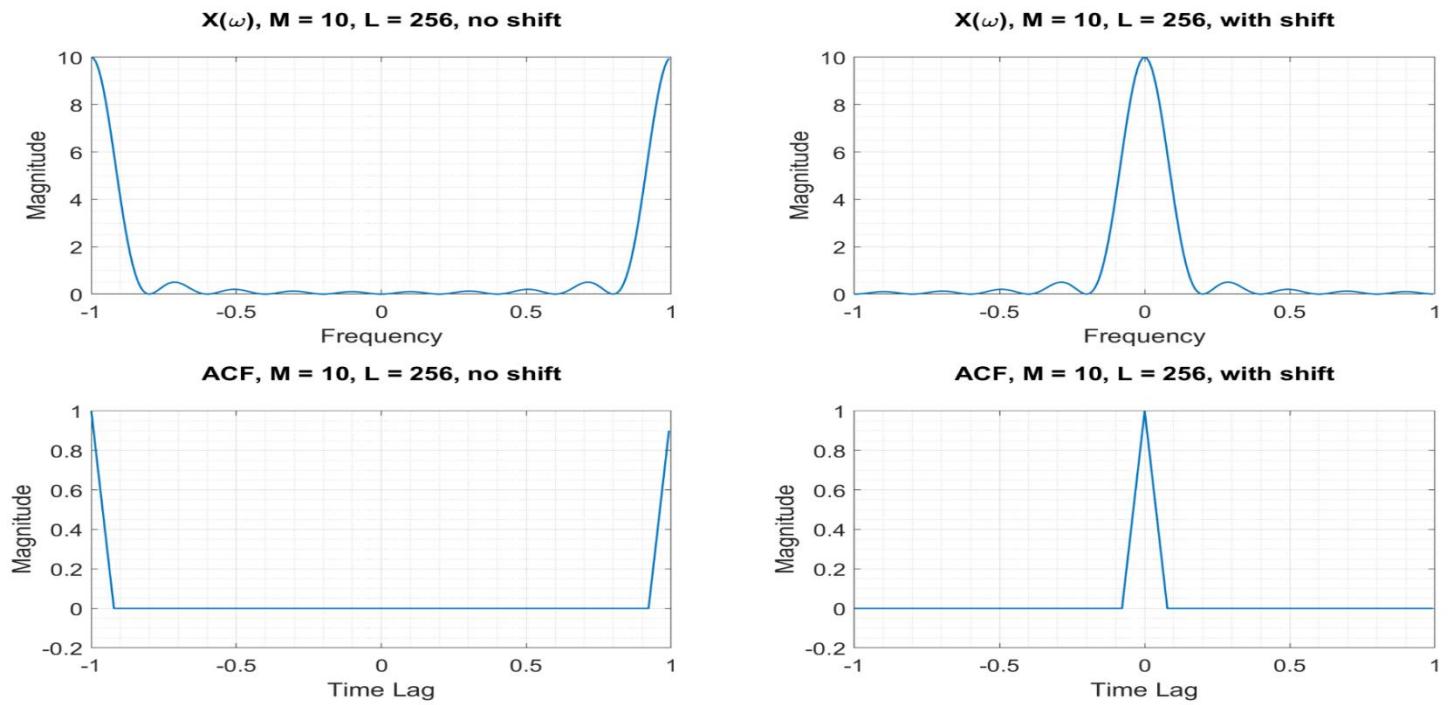


Figure 7: The effect of `fftshift` on the DFT spectrum and the ACF.

When L is odd the frequency and time lags vector must be different than the ones used when L is even. In the first row the false vectors have been used, while in the second row they have been changed with the correct ones. The false vectors cause a very small shift and the maximums are not found in the zero frequency and zero time lag for the Fourier transform and the autocovariance function respectively. The correct frequency and time lag vectors can be found on the appendix, where the code is provided.

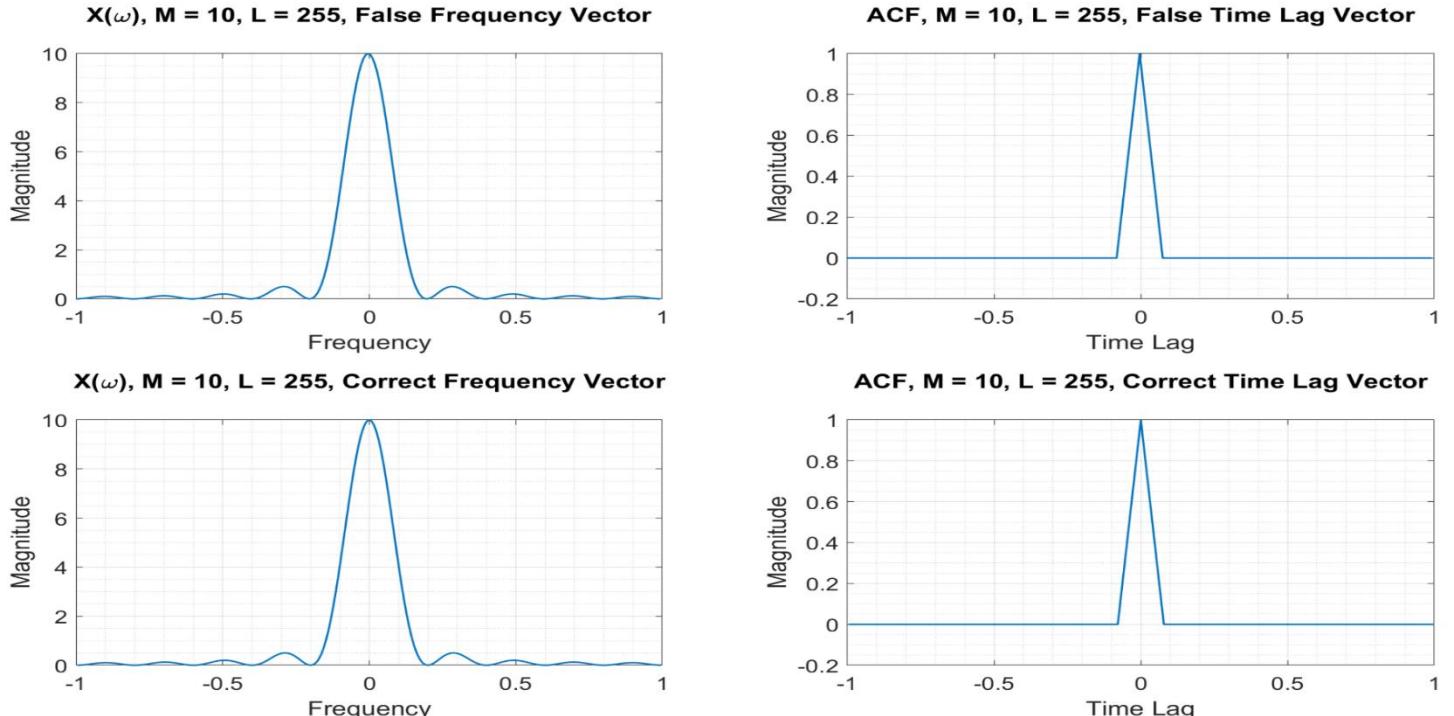


Figure 8: The dependence of the frequency and time lag vectors on their length.

1.3 Resolution and Leakage of Periodogram-based Methods

- a) The peak of the first three sidelobes and the 3dB bandwidth of the main lobe of the Bartlett window is presented on fig.9 as a function of N. The 3dB bandwidth is inversely proportional to N. That means that as N increases the bandwidth of the main lobe of the window decreases. This can be also seen in fig 10 where the magnitude of the Bartlett window is presented in linear and logarithmic scales for N = 8, N = 64 and N = 128. Furthermore, it is obvious that the peak of the sidelobes is constant for different values of N as seen in fig.9 a. The peak of the first and second sidelobes are on -27 dB, whereas the peak of the third sidelobe is on -36 dB almost.

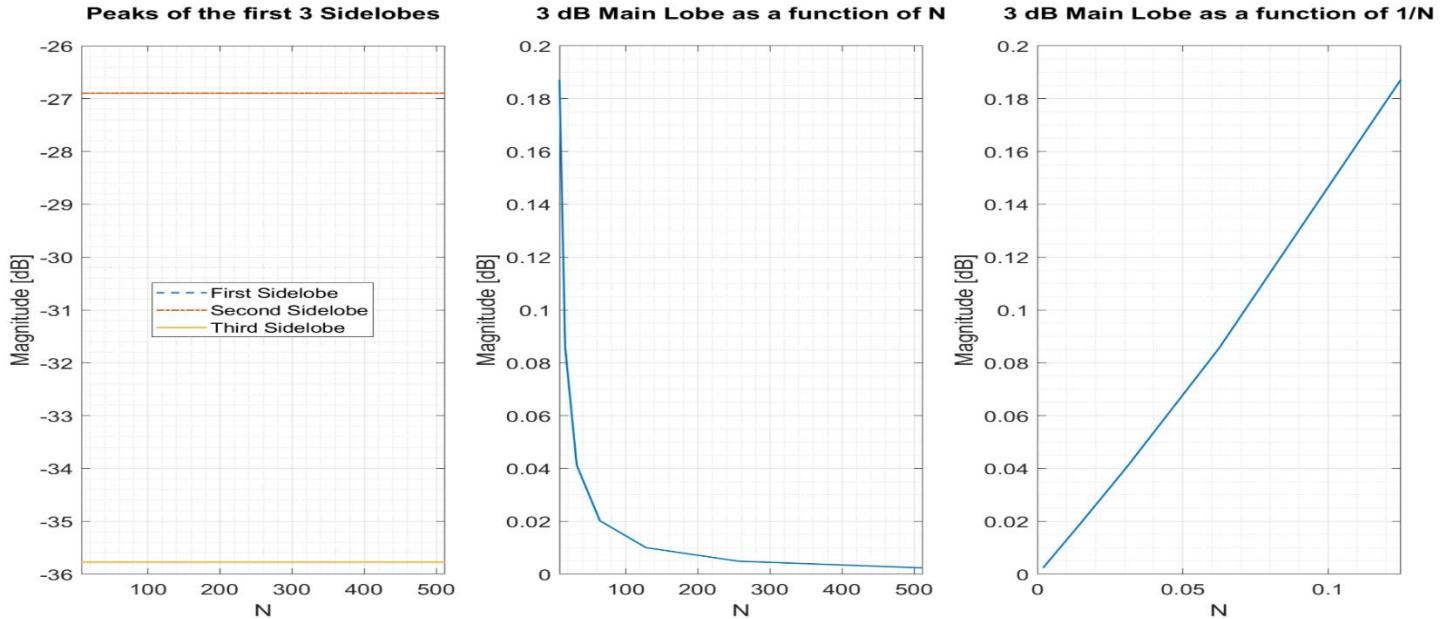


Figure 9: The peaks of the first 3 sidelobes of a Bartlett window as a function of N and the 3dB width of its main lobe as a function of N and $1/N$.

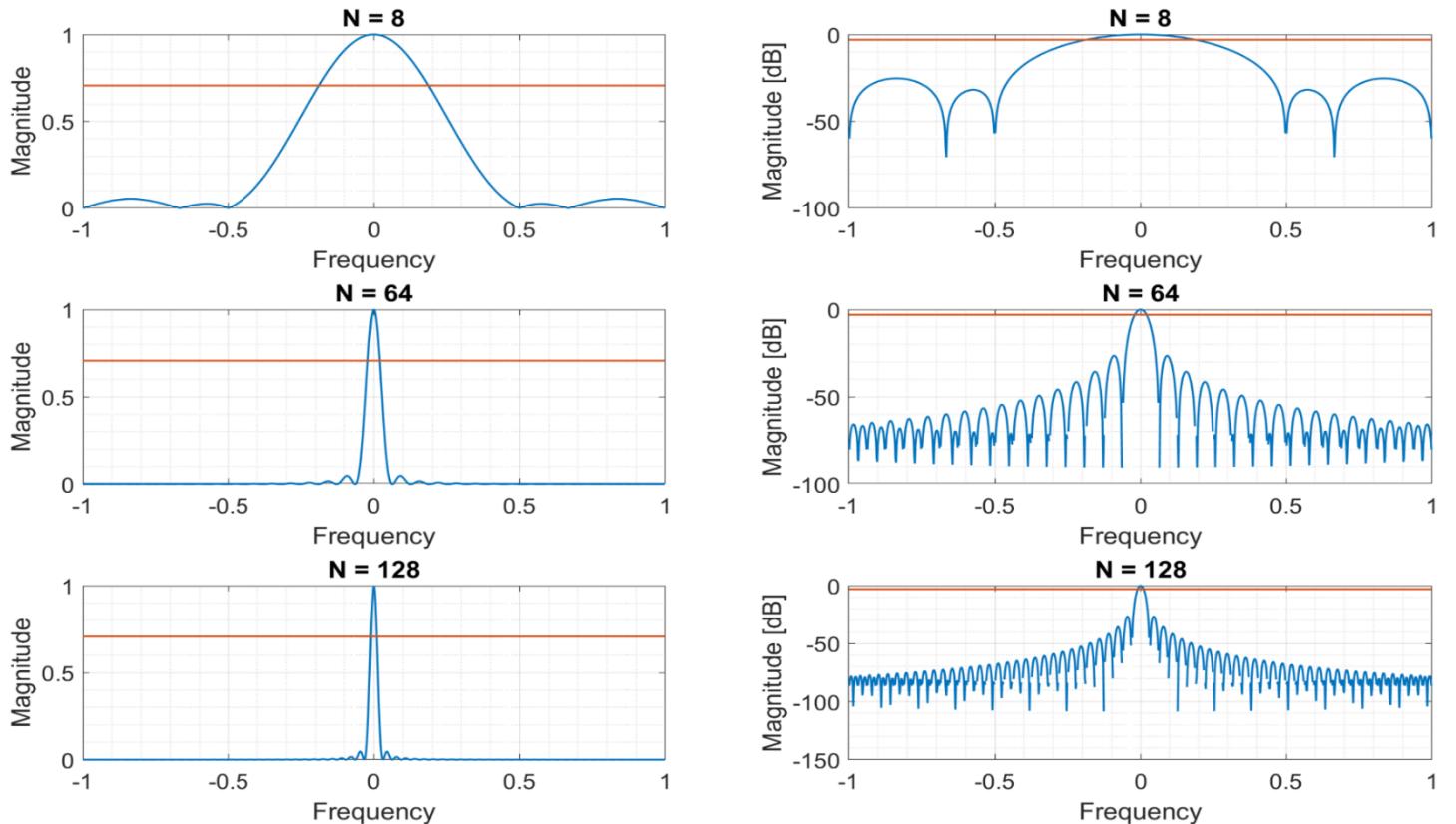


Figure 10: The magnitude of the Bartlett window and the 3dB line for $N = 8$, $N = 64$ and $N = 128$ in both linear and dB scales.

- b) The standard periodogram of two sine waves, with equal amplitudes and differing frequencies for a value of $\frac{\alpha}{N}$ for various α , is presented in figure 11. It is obvious that the minimum value of α for which the two frequency components can be resolved is equal to 0.6. When the value is equal to 1 the two frequencies are clearly resolved. Finally, for values greater than 1 the frequencies can be resolved too. It seems that in multiples of $\alpha = 1$ the two frequencies can be clearly resolved. This does not happen in the hamming-windowed periodogram of the next question.

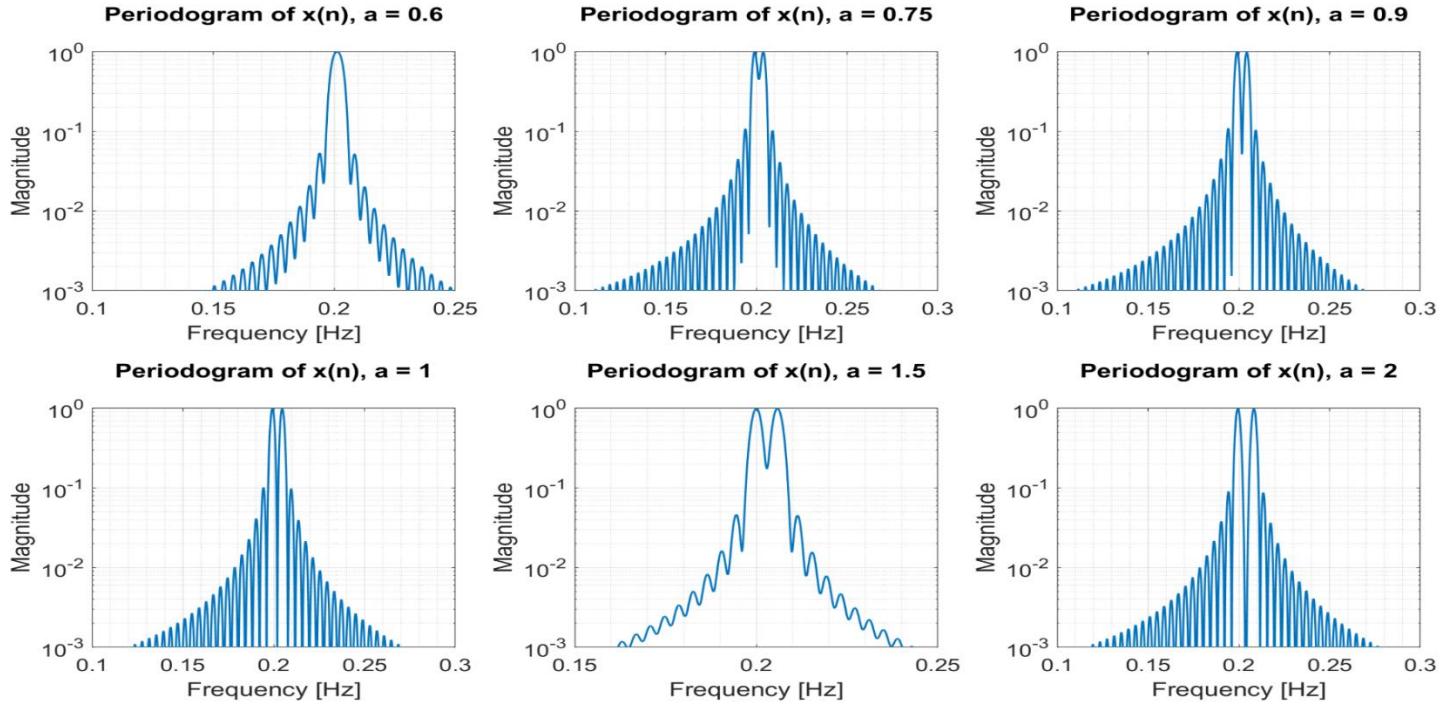


Figure 11: The periodogram of two sine waves for various α .

- c) The hamming-windowed periodogram is presented in figure 12. With that type of window, the two frequencies are resolved clearly only when $\alpha = 1$. When the value is equal to 0.9 for example it is much more difficult to resolve the two sine waves. Furthermore, when $\alpha = 2$ the frequencies cannot be resolved neither. Thus, if there was white gaussian noise in the signal the standard periodogram would perform better than the hamming windowed one for those values of α .

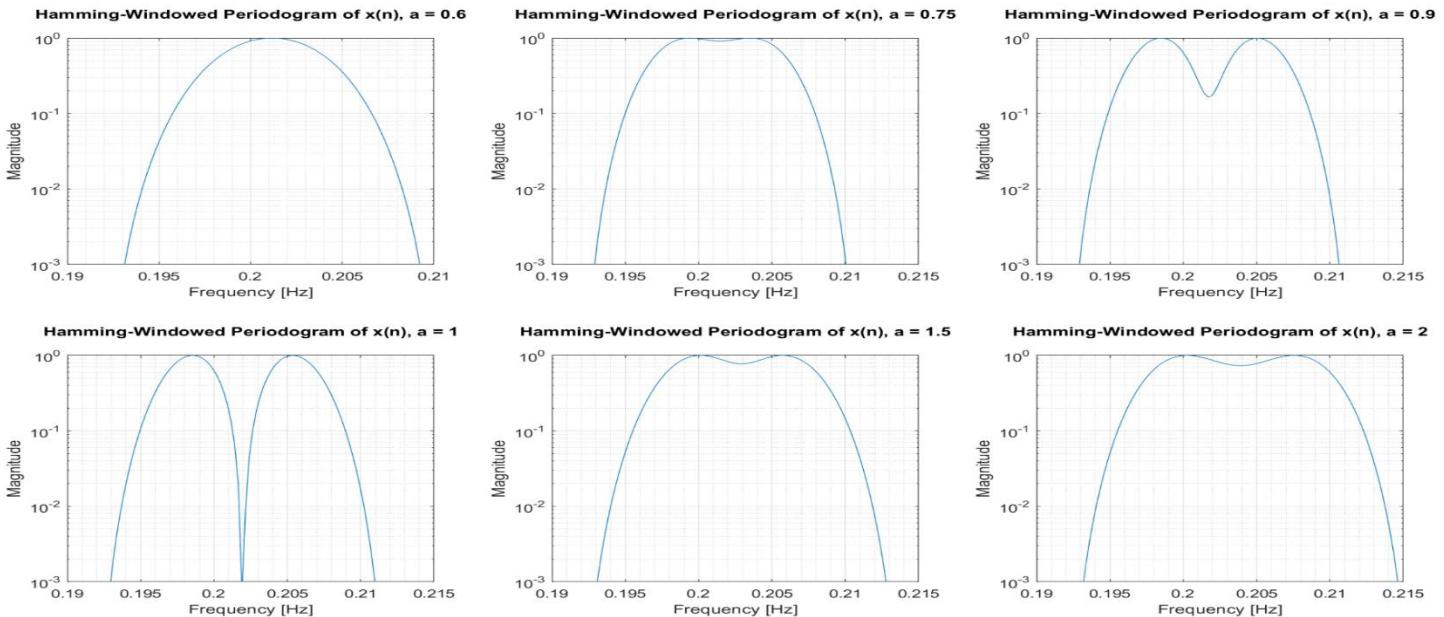


Figure 12: The Hamming-Windowed Periodogram of two sine waves for various α .

- d) When the amplitudes of the two sinusoidal sequences are different, it is difficult to resolve clearly the two sequences. The two frequencies must not be so close now, and therefore greater values of α should be used. For example, when $\alpha = 4$ and the amplitude of the second sine wave is ten times smaller than the first it is difficult to observe the second frequency on the periodogram, while when the amplitude is 100 times smaller it is impossible. However, if $\alpha = 12$ the two frequencies can be resolved more easily. Finally, if the hamming-windowed periodogram is used the results are even better.

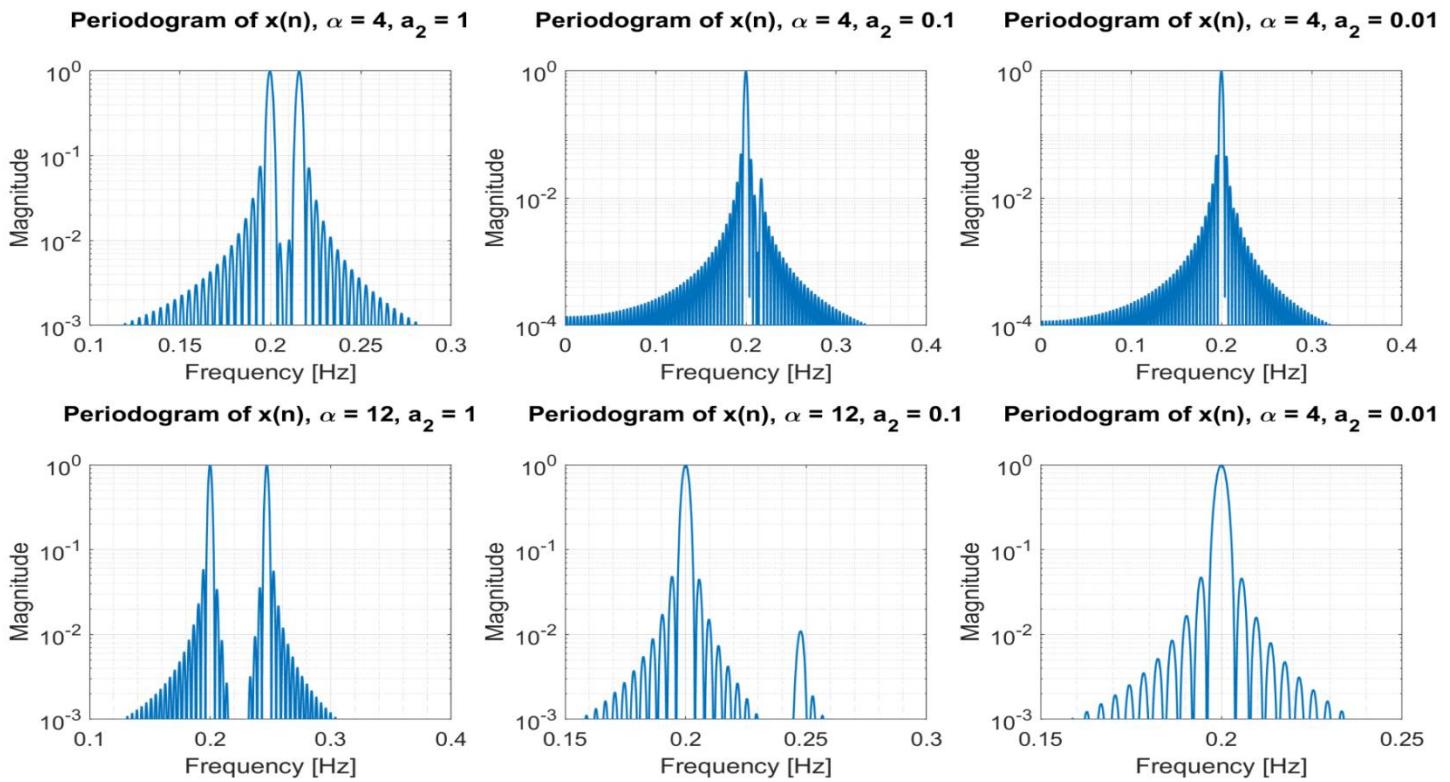


Figure 14: The periodogram of two sine waves with differing amplitudes for $\alpha = 4$ & $\alpha = 12$.

- e) Figure 15 shows that both $\alpha = 4$ & $\alpha = 12$ correspond to almost zero values of the amplitude of the Bartlett window. This is the reason why these two values were used in the previous question. Furthermore, the second value give better resolution, because the peaks of the neighboring sidelobes to that frequency ($12/N$) are smaller than the peaks of the sidelobes close to the first frequency ($4/N$).

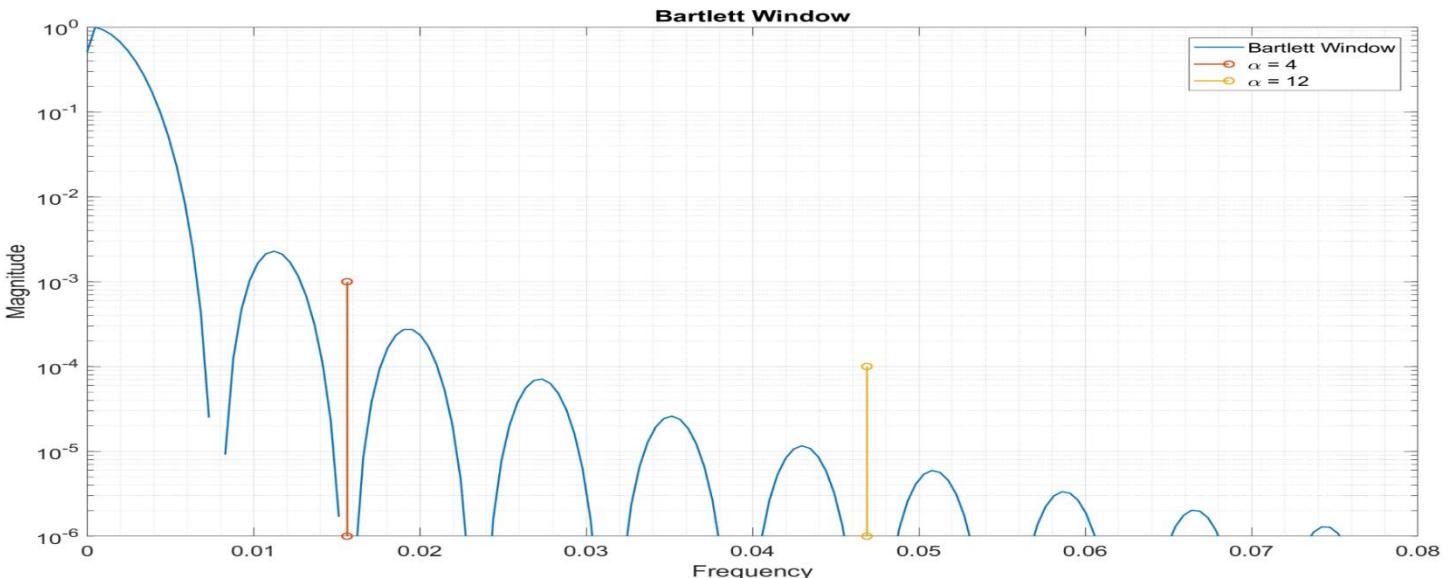


Figure 15: The Fourier Transform of the Bartlett Window and the frequencies corresponding to $\alpha = 4$ & $\alpha = 12$.

- f) With the Chebyshev-windowed periodogram (Fig. 15) the two frequencies can be very easily resolved when $\alpha = 12$. However, when $\alpha = 4$ it is very difficult to observe the two sine waves even when they have the same amplitude. If white gaussian noise is added to the signal, it will be impossible to resolve these two frequencies. The Blackman-Tukey method can resolve the two frequencies when $\alpha = 12$ very easily. Also, when larger points of the ACF function are not used (the large lags have extrapolated) the two frequencies can be resolved. Thus, these large lags do not provide any information and therefore they can be dismissed.

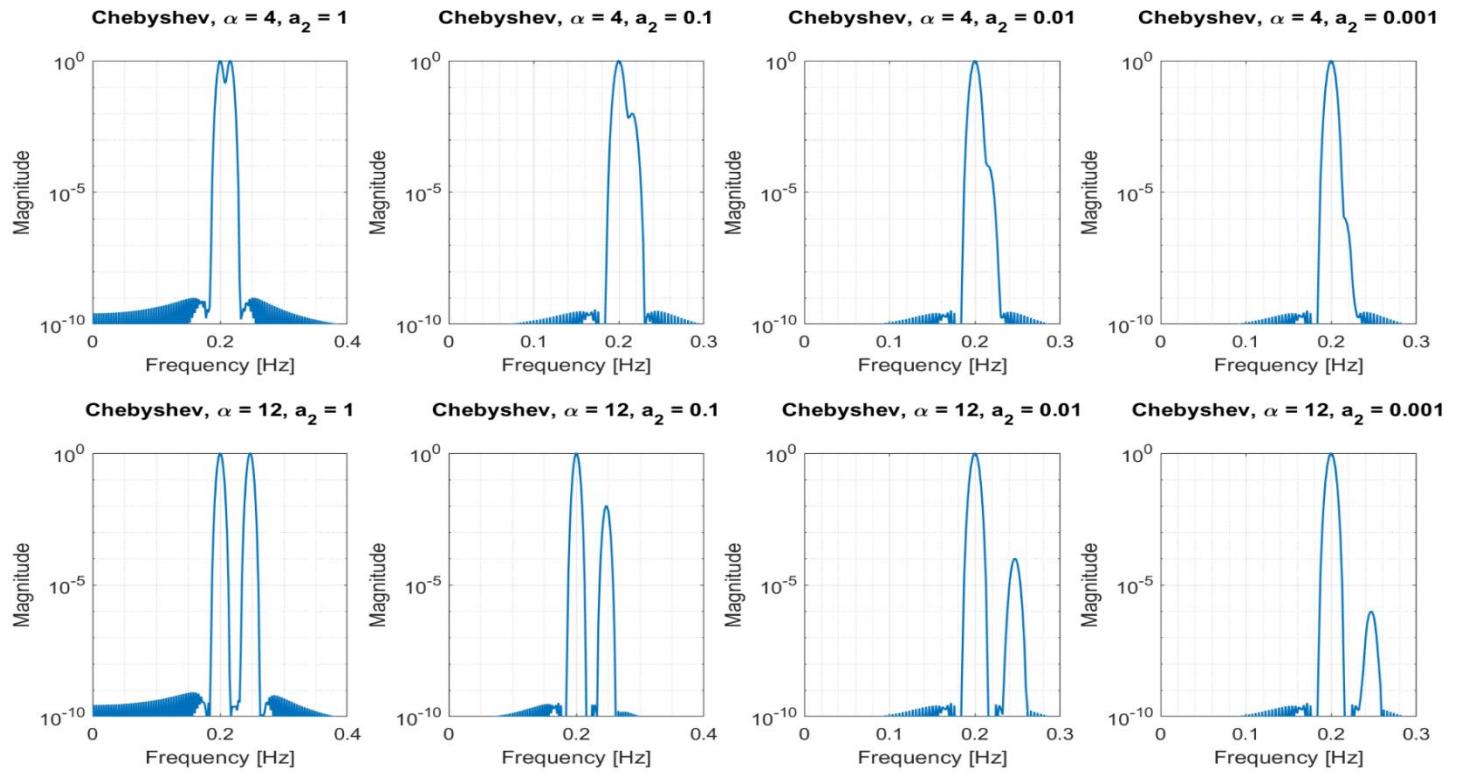


Figure 15: The Chebyshev Windowed periodogram of two sinewaves with differing amplitudes for $\alpha = 4$ & $\alpha = 12$.

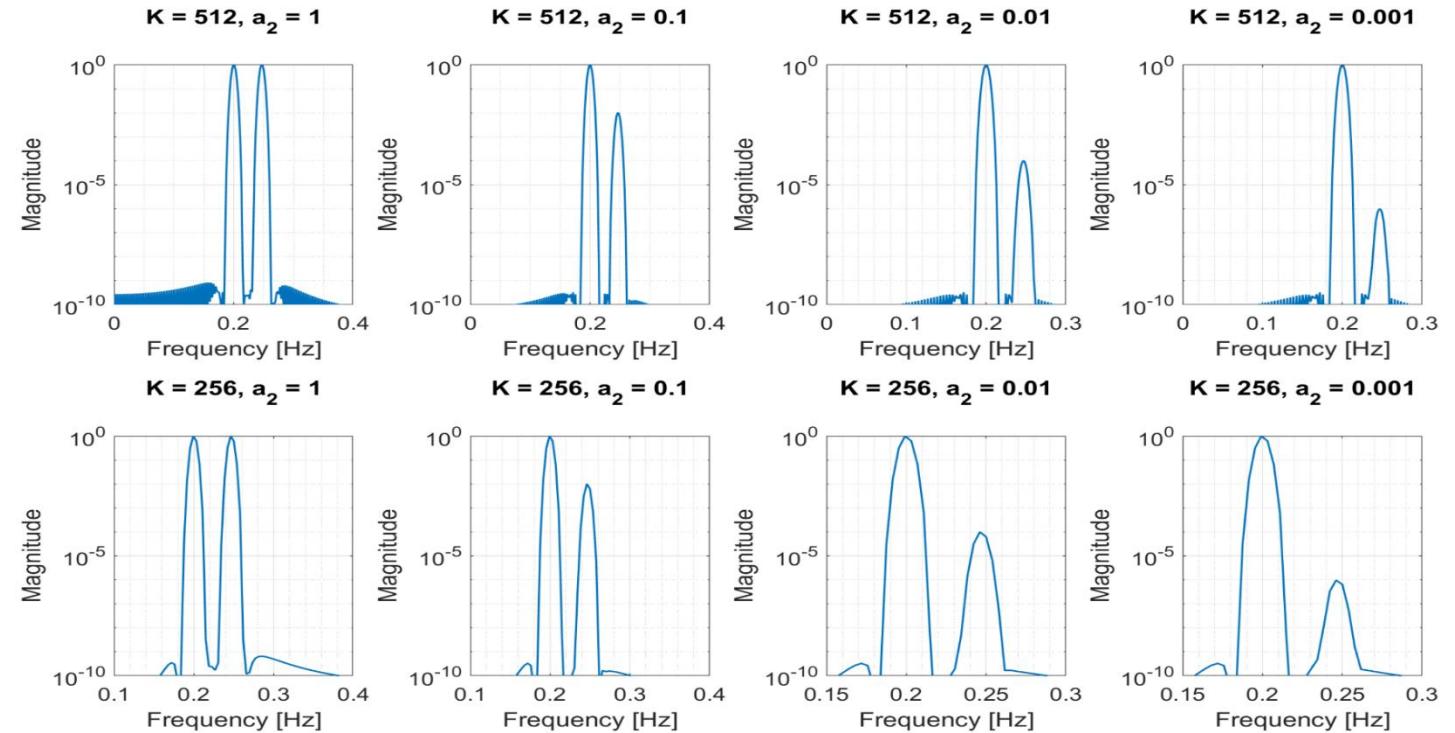


Figure 16: The Blackman-Tukey method using a Chebyshev Window for $K = 512, K = 256$ and differing sine wave amplitudes

1.4 Periodogram-based Methods Applied to Real-World Data

- a) The Chebyshev-windowed and hamming-windowed periodogram have been used to detect the periodicities of the sunspot time series. After removing the mean or the trend from the time series it is obvious that at $f = 0.09$ there is a peak. This means that there is a period of 11 year at the data since the sunspots are measured every year. Also, a peak at $f = 0.025$ can be observed. It is important to state that if the mean or the trend are not removed it is more difficult to find the periodicities of these data. Furthermore, in figure 18 the standard periodogram is used and the application of the logarithm to the data is examined. It is obvious that when the logarithm is applied the peak at $f = 0.09$ is much higher than all the other peaks, while when it is not applied there is a very high peak at $f = 0.1$ which may confuse the analyst of the data. Therefore, applying the logarithm to that data can lead to find their periodicities with more confidence.

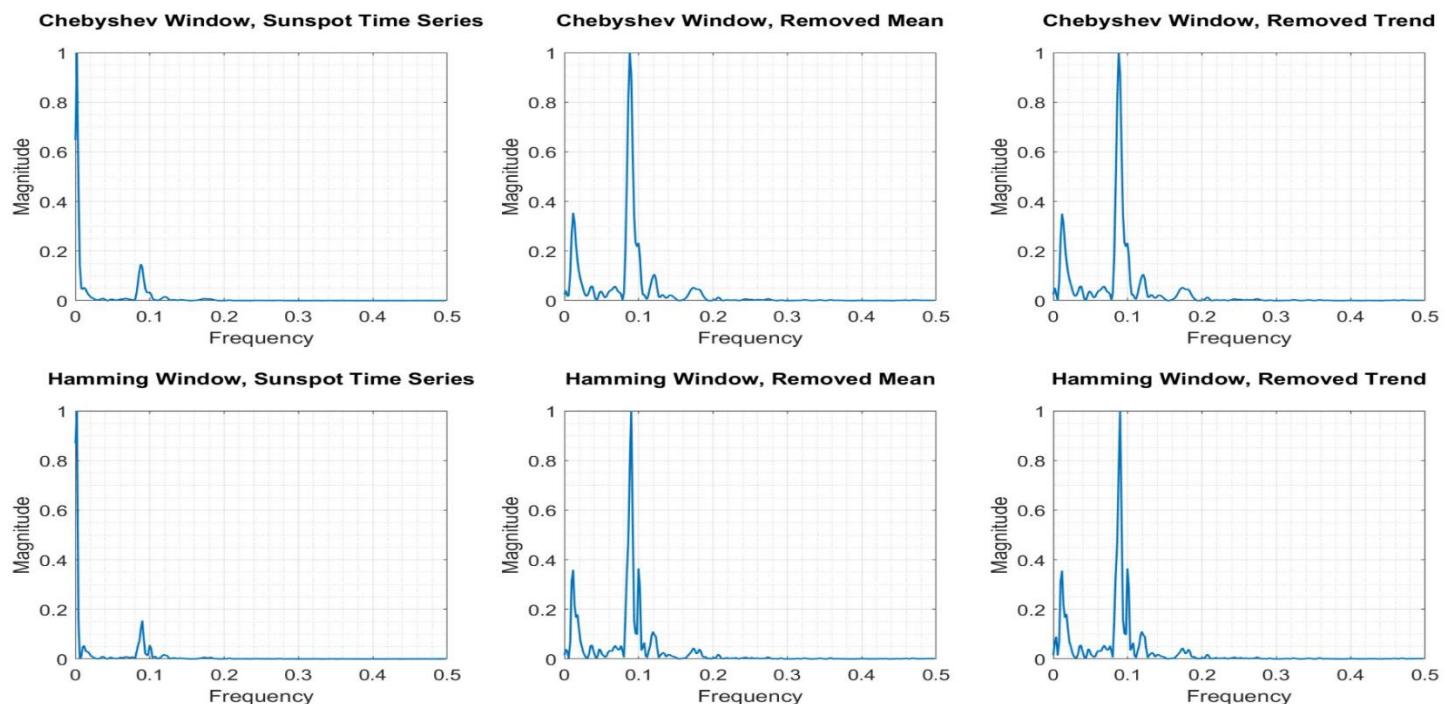


Figure 17: The Chebyshev-Windows and Hamming-Windows Periodogram of the original Sunspot Time series and after removing the mean and the trend.

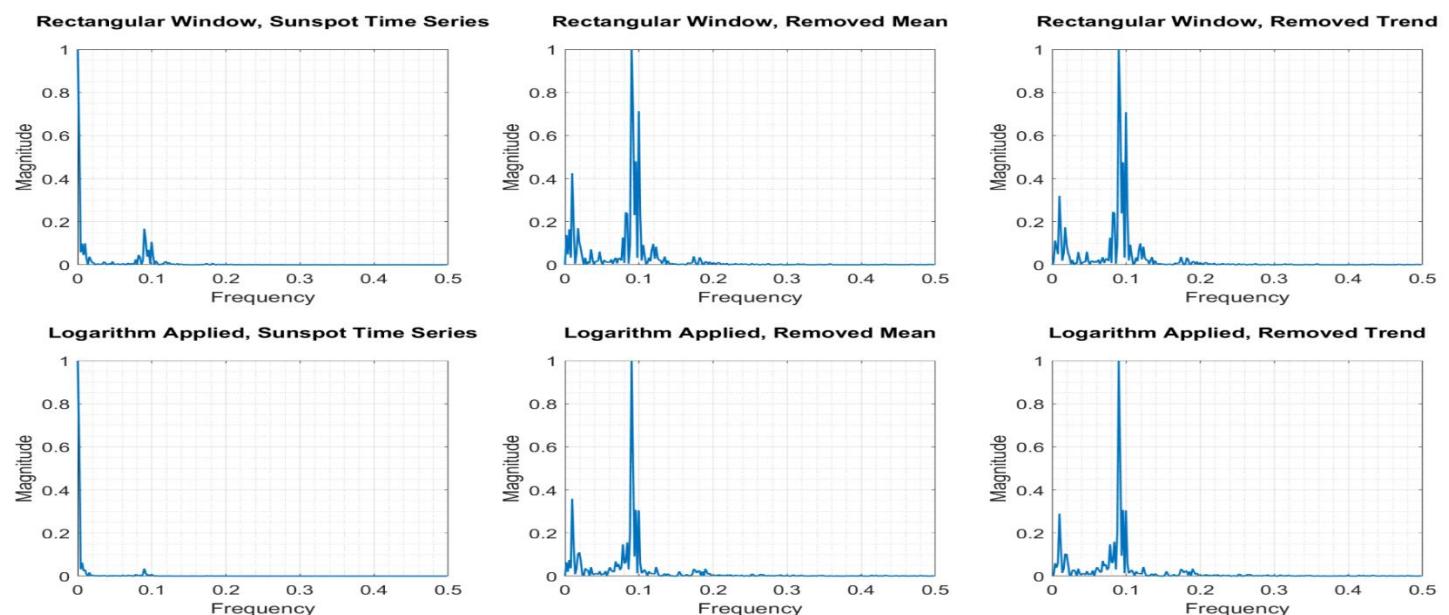


Figure 18: The periodogram of the Sunspot Time Series and their derivatives as well as the consequence of applying the logarithm.

- b) The standard and three averaged (using the Welch method) periodograms of the EEG signal are presented in the following figure. The standard periodogram provides us with some insights for the SSVEP and the power-line interference, but the spikes are not very clear since the noise is not reduced significantly. Furthermore, the alpha-rhythm is very difficult to be observed. For the 10s window the alpha-rhythm, the fundamental frequency and the first and second harmonics of the SSVEP as well as the power-line interference can be clearly observed, since the peaks in these frequencies are much higher than their neighboring ones. As the length of the time window decreases the spectral resolution decreases too and important information about the signal is lost. Therefore, for the 5s window the 39 Hz harmonic of the SSVEP can be hardly observed. Furthermore, when the 1s window is used, the 39 Hz harmonic is lost, due to the high value of the power-line interference, which is very close to it. Additionally, the fundamental frequency is now not so easily observed, because the alpha-rhythm is very close to it and it dominates the frequency area close to its' center. Finally, the first harmonic at 26 Hz is more easily observed because its neighboring frequencies do not have high magnitudes.

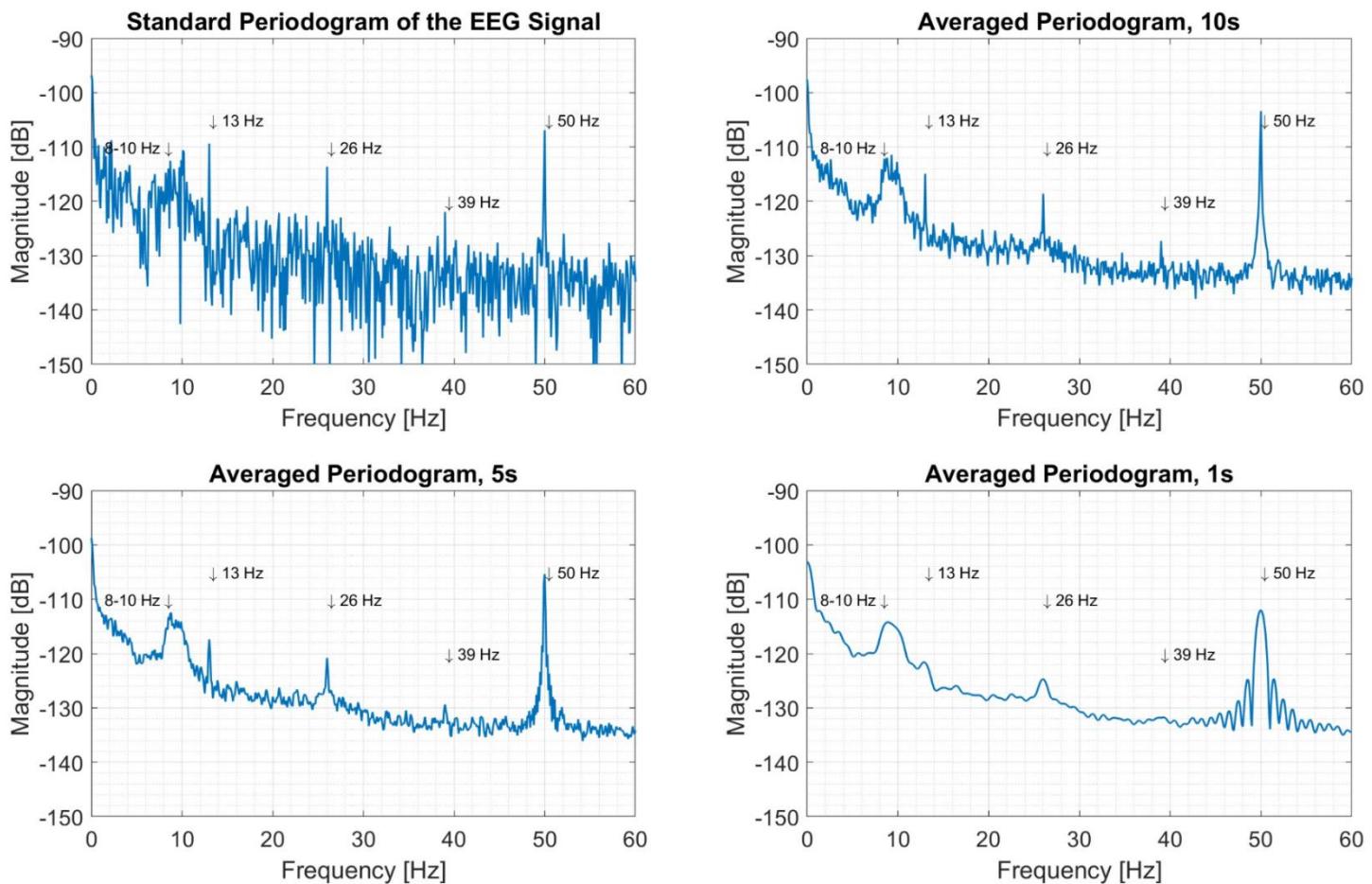


Figure 19: The Standard and the Averaged Periodogram (for window lengths 10s, 5s & 1s) of the EEG data.

2. PARAMETRIC LINE SPECTRA

2.1. Correlation Estimation

- a) In fig. 20 the ACF and PSD unbiased and biased estimates of WGN, filtered WGN and noisy sinusoidal signal are depicted. Unbiased ACF estimates have very high values for large time lags as bias performance is always traded with variance performance. Although the unbiased estimate seems more appropriate as its mean matches to the true mean of PSD, there is a big variance of the ACF estimate for large time lags since the assumption that $N \rightarrow \infty$ is not met. These highly erratic values of the ACF for large time lags leads to the fact that the ACF is not positive semidefinite and therefore result to negative PSD values.

It is obvious that the biased estimator gives more consistent results for the WGN, the filtered WGN and the noisy sinusoid. Firstly, the biased ACF estimate of the WGN is a Dirac's delta function, while its' PSD estimate is flatter than the unbiased case, corresponding to the Fourier Transform of a Dirac's delta function.

The biased ACF of the filtered WGN has some differences from the non-filtered since its' PSD tends to zero for large frequencies and an amplification has appeared. Finally, the biased PSD Estimate corresponds to what is expected, since the Fourier Transform of a sinusoid is a Dirac's delta function in the frequency domain.

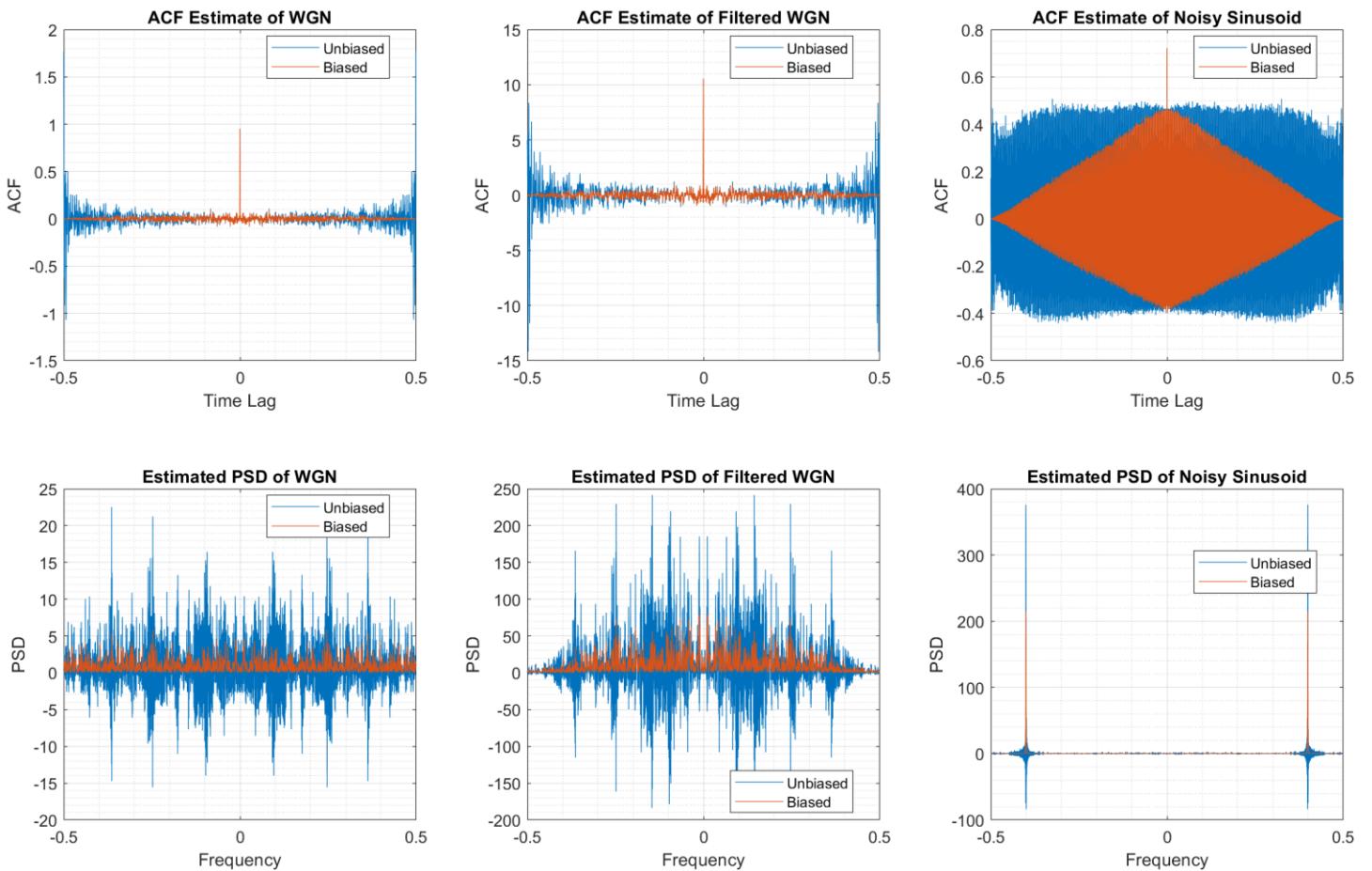


Figure 20: ACF & PSD unbiased and biased estimates of WGN, filtered WGN and noisy sinusoid.

- b) Since the biased estimator ensures that the ACF is positive definite and therefore the PSD estimate is non-negative, it is more appropriate than the unbiased, although its' mean does not match the true mean of PSD. In this question a signal composed of two sinusoids of normalized frequencies 0.1 & 0.2 Hz/sample corrupted by white gaussian noise have been generated and 100 different realizations have been implemented.

The PSD estimates of the different realizations and the ensemble average are presented in figure 21 (left). It is obvious that each realization's PSD estimate does not resemble the true power spectrum since the variance is significantly high. Although the power of the ensemble average in frequencies that do not correspond to the sinusoids is equal to -6 , a large deviation is observed in the power of the different realizations. Additionally, in the frequencies that the sinusoids are occurred, although there are realizations that have peaks, there are a lot of other that have power less than -25 dB.

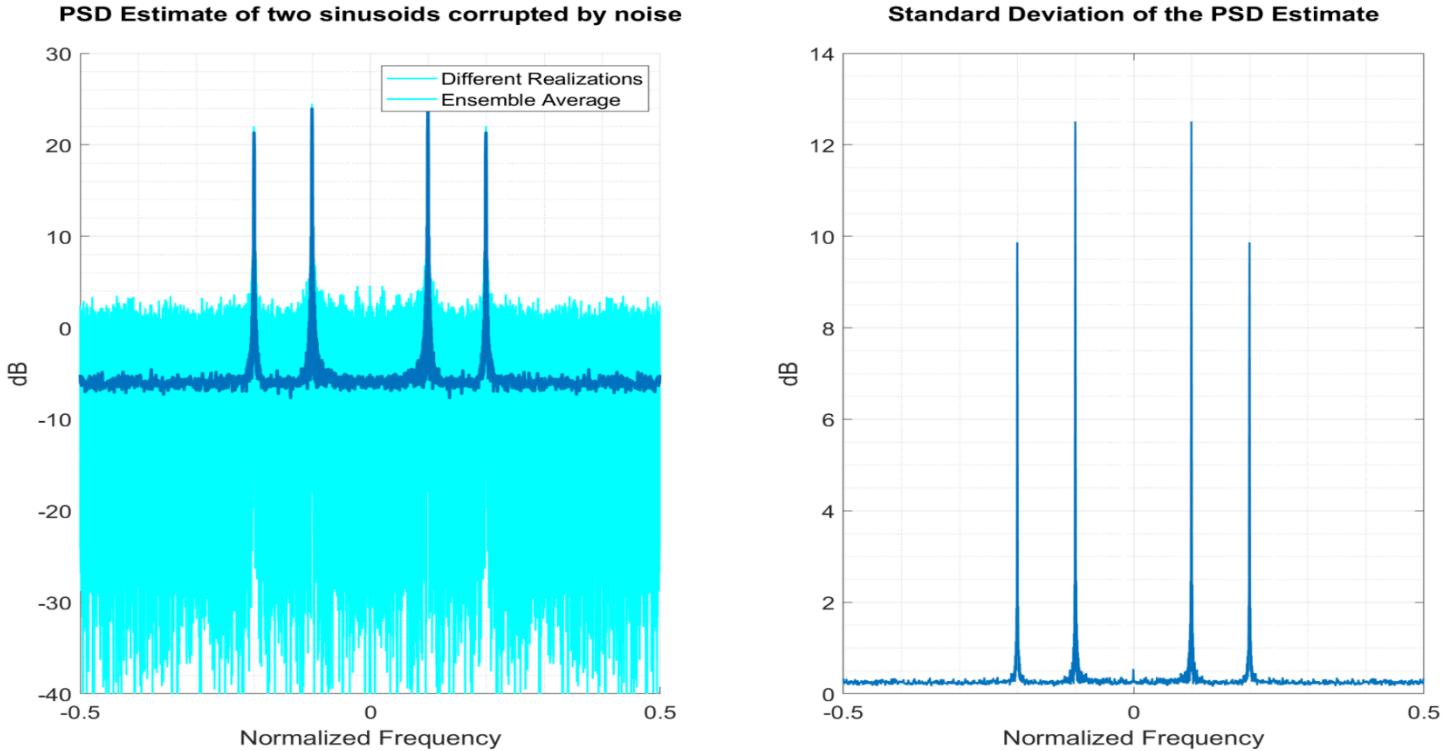


Figure 21: The PSD Estimate (left) of two sinusoids corrupted by noise and its' Standard deviation (right).

- c) In this question the confidence interval of the PSD estimate is examined, which is defined in the following equation:

$$CI = \hat{P}(\omega) \pm \mu \hat{\sigma}_{\hat{P}}(\omega) \quad (2.1.1)$$

meaning that for high PSD estimates values the confidence interval widens, since the standard deviation is proportional to the value of the PSD estimate.

Figure 21 (right) presents the standard deviation of the PSD estimate of the signal generated for 2.1.b. It is clear that the standard deviation is proportional to the PSD estimate and also that the constant μ is almost equal to 0.5.

Finally, comparing the dB plot of the PSD estimate to the linear plot of figure 1 of the coursework assignment, one can realize that the logarithm provides with more representation benefits. This occurs because the logarithm filters the PSD estimate and it can express the differences in more detail for lower power frequencies, while it attenuates higher powers. This property is very convenient since it gives a better representation of the whole frequency spectrum, providing with better insights for data and signal analysis.

- d) Figure 22 shows the PSD Estimate (in dB) of a noise-corrupted signal comprising of two sinusoids. Different signal lengths and frequencies have been implemented and the differences can be observed in the figure. It is clear that when $N = 30$ & $\Delta f = 0.02$ the periodogram cannot distinguish the two frequencies, since the periodogram is proportional to $1/N = 0.033$. Increasing the signal length to $N = 35$ & $N = 40$ the periodogram starts to identify the two frequencies, as one can observe in fig. 22 (top-middle and top-right).

When $\Delta f = 0.04$ the frequencies can be more easily identified, since resolution is smaller than the frequency difference. The second row of figure 22 presents this fact. It is obvious that for the same signal length as before, the frequencies can be more easily distinguished if their difference is larger.

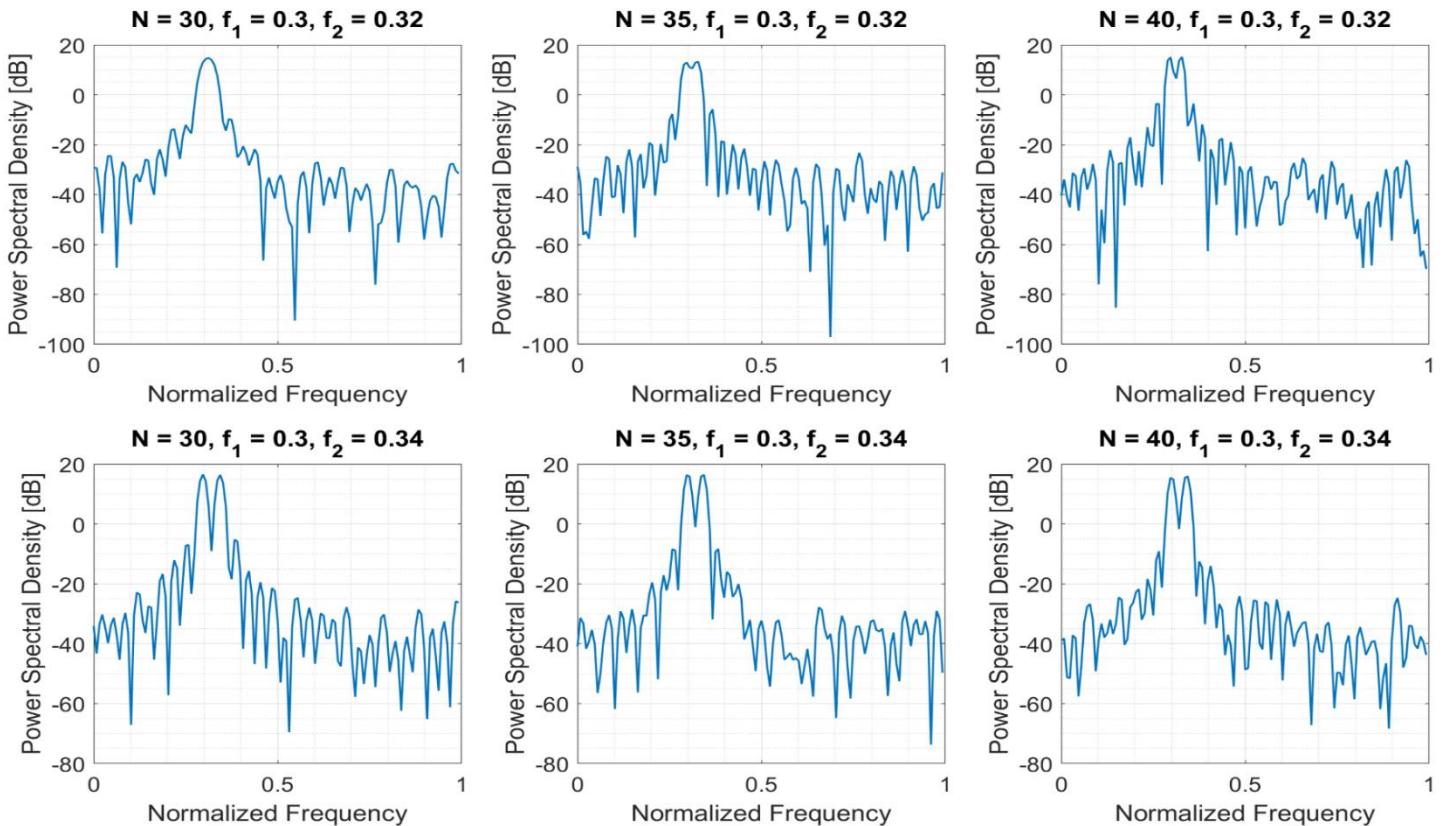


Figure 22: The PSD estimate of a noise-corrupted two-sinusoidal signal for different frequencies and lengths.

- e) Multiple Signal Classification (MUSIC) is a spectral estimation subspace method, which uses the eigenvalue decomposition. Firstly, the correlation matrix is estimated using the `corrmtx(x,m,'mod')` matlab function, where the first argument is the signal vector, the second is the dimensionality of the correlation matrix and the third states how the autocorrelation matrix will be estimated. The ‘modified’ method derives the autocorrelation estimate by using forward and backward prediction error estimates. The autocorrelation matrix has the following form after applying the eigenvalue decomposition:

$$\mathbf{R}_{xx} = \mathbf{E}\mathbf{P}\mathbf{E}^H + \sigma_w^2\mathbf{I}, \quad (2.1.2)$$

where \mathbf{E} is comprised of the m eigenvectors, \mathbf{P} is the eigenvalue diagonal matrix and σ_w^2 is the power of the noise. The first p (in magnitude) eigenvectors span the signal subspace, while the remaining $m - p$ span the noise subspace.

The pseudospectrum (PSD) estimate using the MUSIC algorithm makes use of the eigenvectors of the signal and the noise subspaces. The estimate is calculated using the following equation:

$$\hat{P}_{MU}(\omega) = \frac{1}{\sum_{i=p+1}^N |\mathbf{e}^H \mathbf{v}_i|^2}, \quad (2.1.3)$$

where ν_i 's are the eigenvectors corresponding to the noise subspace. The matlab implementation uses the pmusic(R,p,nfft,fs,'corr') function, where the first argument is the correlation matrix as stated by the fifth argument, the second is the dimensionality of the signal subspace, the third shows the integer length of the DFT and the fourth represents the sampling frequency.

In figure 23 the PSD estimate (or pseudospectrum) using the MUSIC algorithm is presented. It is obvious that this algorithm provides with more detailed information than the non-parametric periodogram-based method used in the previous question. Therefore, the superresolution property of the MUSIC method is observed. That property holds because this method can estimate frequencies with higher resolution, since the estimation function is not only evaluated for the DFT bins. Thus, the MUSIC method can identify the two different frequencies even when $N = 20$, while the periodogram-based method could not distinguish them even when $N = 30$.

However, one can observe that the estimated frequencies using that method have a small divergence by their true value. In fig. 23 (middle) the estimated frequencies are equal to 0.305 & 0.325, an error that is not observed in the periodogram based method. In addition to this the MUSIC method is more computational expensive than the periodogram-based. Furthermore, that method requires the knowledge of the dimensionality of the signal space and, also, assumes that these signals are uncorrelated. However, there are algorithms that overcome the assumption of the uncorrelated signals, such as the spatial smoothing algorithm for direction of arrival (DoA) applications.

Concludingly, the MUSIC method is more accurate than the periodogram-based method when there is knowledge of the system, although it is more computational expensive. However, a general spectrum estimate would not be so accurate when using that method and, therefore, it cannot give good results for every case.

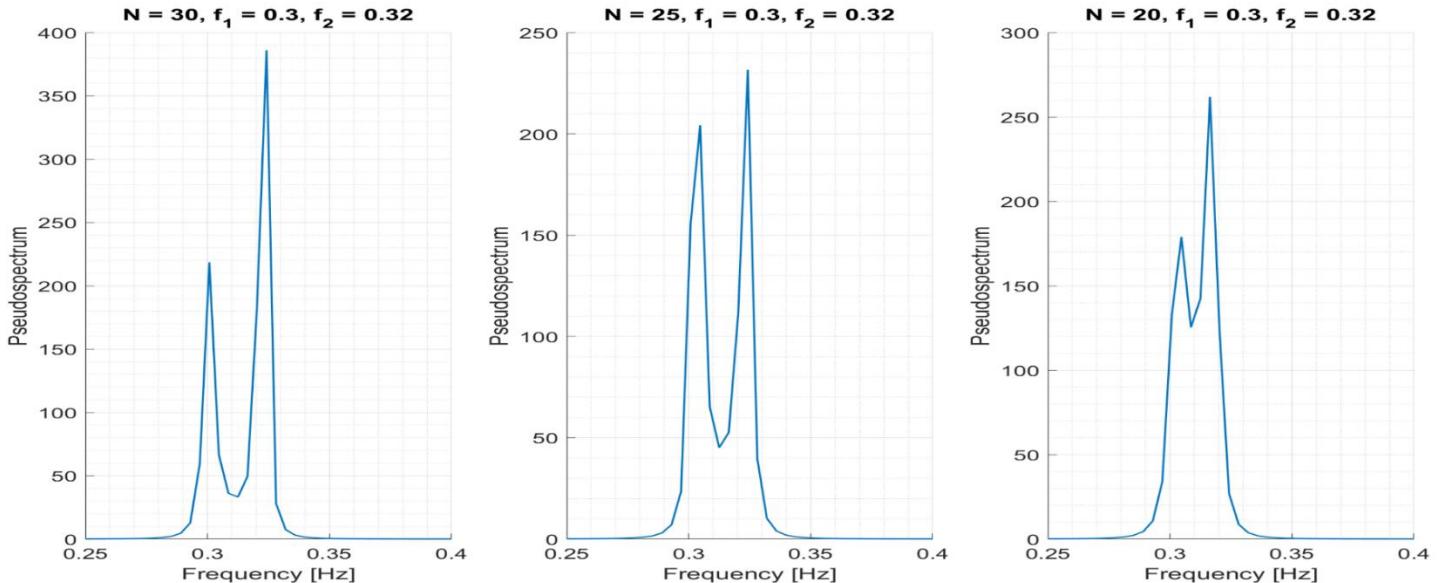


Figure 23: The pseudospectrum of the noise-corrupted two-sinusoidal signal, generated using the MUSIC method.

2.2. Spectrum of Autoregressive Processes

- a) An autoregressive (AR) process is defined in the following equation:

$$y(n) = \sum_{k=1}^p a_k y(n-k) + w(n) \quad (2.2.1),$$

where a_k are the AR parameters and $w \sim N(0, \sigma_w^2)$ is additive white Gaussian noise of zero mean and variance equal to σ_w^2 . The AR parameters can be found by using the Yule-Walker equations, which have the following form:

$$\mathbf{R}_{xx} * \mathbf{a}_p = \sigma_w^2 * \mathbf{b} \quad (2.2.2)$$

where $\mathbf{R}_{xx} \in \mathbb{R}^{p \times p}$ is the autocovariance matrix and $\mathbf{b} = [1, 0, \dots, 0]^T$.

Due to the fact that for large time lags the unbiased ACF estimate has highly erratic values, if the order of the AR process is close to the data length, then the estimated parameters will not be the correct ones. Therefore, the biased ACF estimate should be used. Furthermore, the unbiased ACF estimate may not be positive definite resulting in not being invertible. This means that if the matrix inversion property is not met, then the AR parameters cannot be estimated correctly.

- b) Modern spectrum estimation methods use parametric approaches in order to achieve high accuracy. Along with the MUSIC subspace method analyzed on 2.1.e the ARMA spectrum estimation method can achieve very low estimation errors. In this approach an assumption of an underlying model is made, and the parameters of that model need to be found. The Yule-Walker equations, discussed above, and the minimization of the MSE are used in order to derive the parameters.

Figure 24 shows the MSE of estimated models of different orders (left) and the true and the PSD estimates of the two lowest-error models. It is obvious by the left figure that increasing the order after the true value does not provide much better accuracy. Therefore, $p = 4$ provides with an accurate estimation and further increase of the model order is not suggested, since high model orders are more vulnerable to noise.

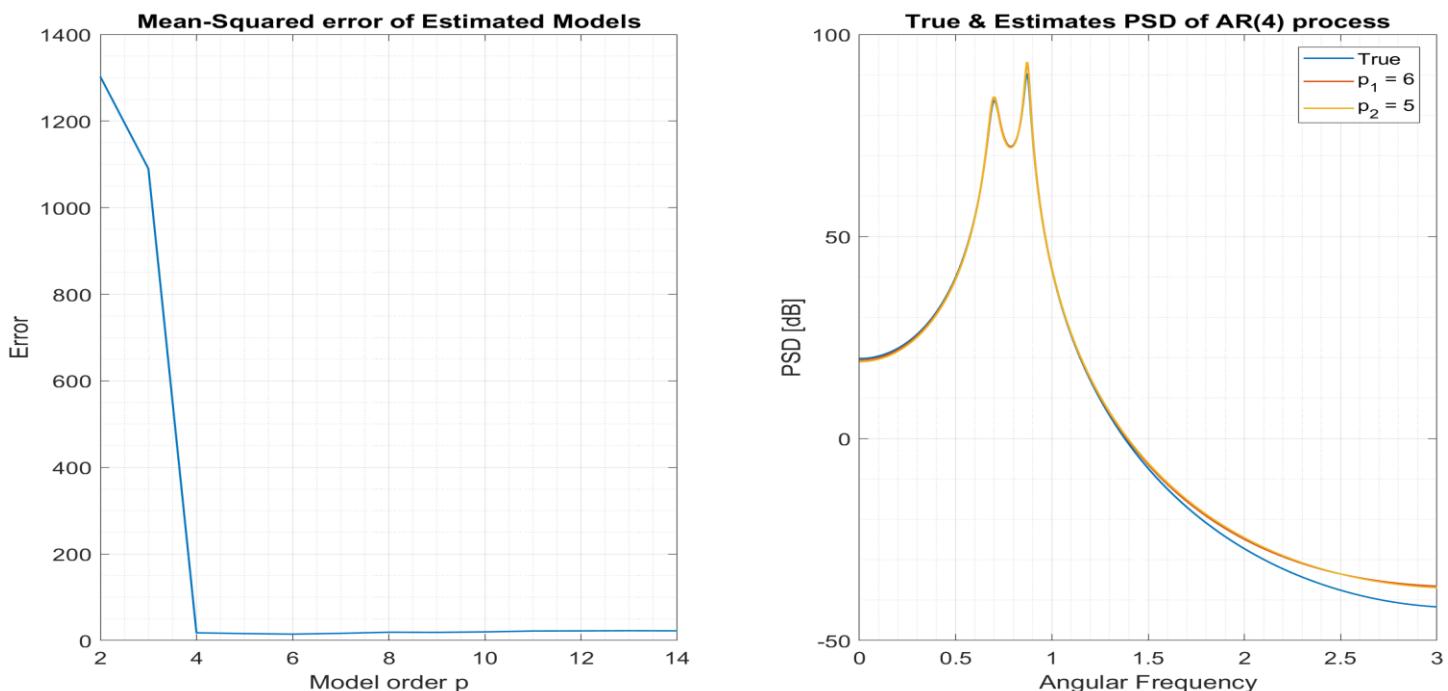


Figure 24: The Mean-Square Error of the AR estimated models (left). The true and the best two PSD estimates of an AR(4) process (right).

In addition to the above, for better model order estimates, there exist model order estimation algorithms which take into consideration the increase of the model order and penalize it, such as the AIC, BIC and MDL algorithms. The right figure shows the true and the PSD estimates for $p = 5$ & $p = 6$. It is obvious that the models are very accurate in estimating the peaks of the spectrum, while they do not provide with the optimal values when the angular frequency is high.

- c) Figure 23 shows the true and the estimated power spectral density (PSD) of an AR(4) signal of length $N = 10000$. When the model order is equal to the true order, the PSD estimate is very close to the true power spectrum when peaks occur and when the angular frequency is high. However, the local minimum in the middle of the two peaks has a relative high divergence from the true. Under-modelling, i.e. $p = 2 < 4$ results in false estimation, since the two frequencies cannot be identified, and the PSD estimate has only one peak, which occurs in the middle of the two real peaks. Finally, over-modelling results in accurate estimation of the PSD in low frequencies as well as in the local minimum between the two peaks. However, there is bigger deviation in the peaks than using the correct model order. Also, in high angular frequencies over-modelling is worse than taking $p = 4$.

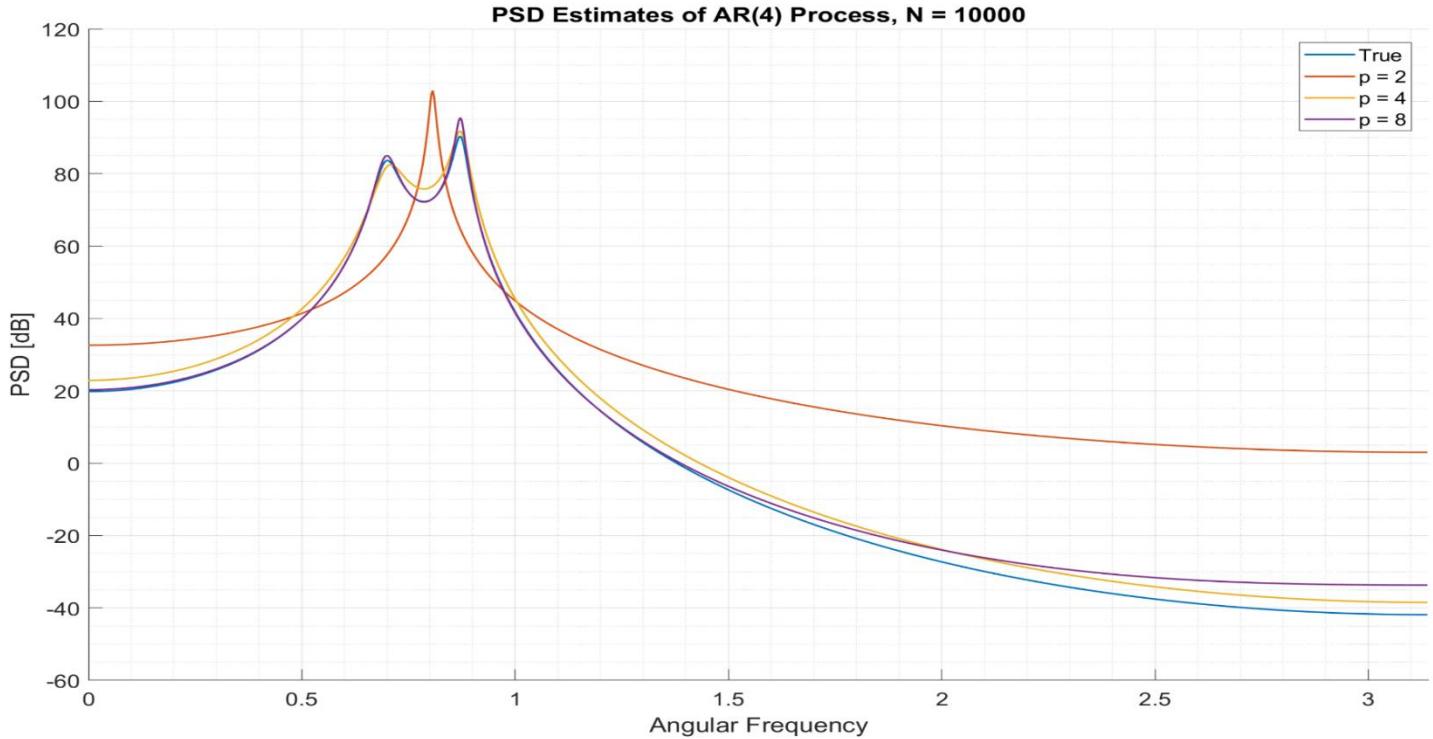


Figure 25: PSD Estimates of an AR(4) process and the effect of under and over-modelling.

2.3. Principal Component Analysis

- a) The singular values of X & X_{noise} are depicted in Fig.26 (left and middle). It is obvious that the rank of the input data X is equal to three, since only the first three singular values are non-zero. The most important effect of noise on the singular values of the noisy signal is that it increases the rank of the matrix, since there are not non-zero singular values. Another effect is that the noise is introduced to the first three singular values, changing their amplitude. However, the effect of the noise to these three singular values is not significant, because the difference of the amplitudes of the clean and the noisy signal is much smaller comparing to the other singular values, as one can notice in the Square Error diagram between the singular values (Fig. 26 right). Finally, if the effect of noise makes the amplitude of the singular values of the noisy subspace equal to the ones of the signal subspace, it would be difficult to identify the rank of the matrix X_{noise} .

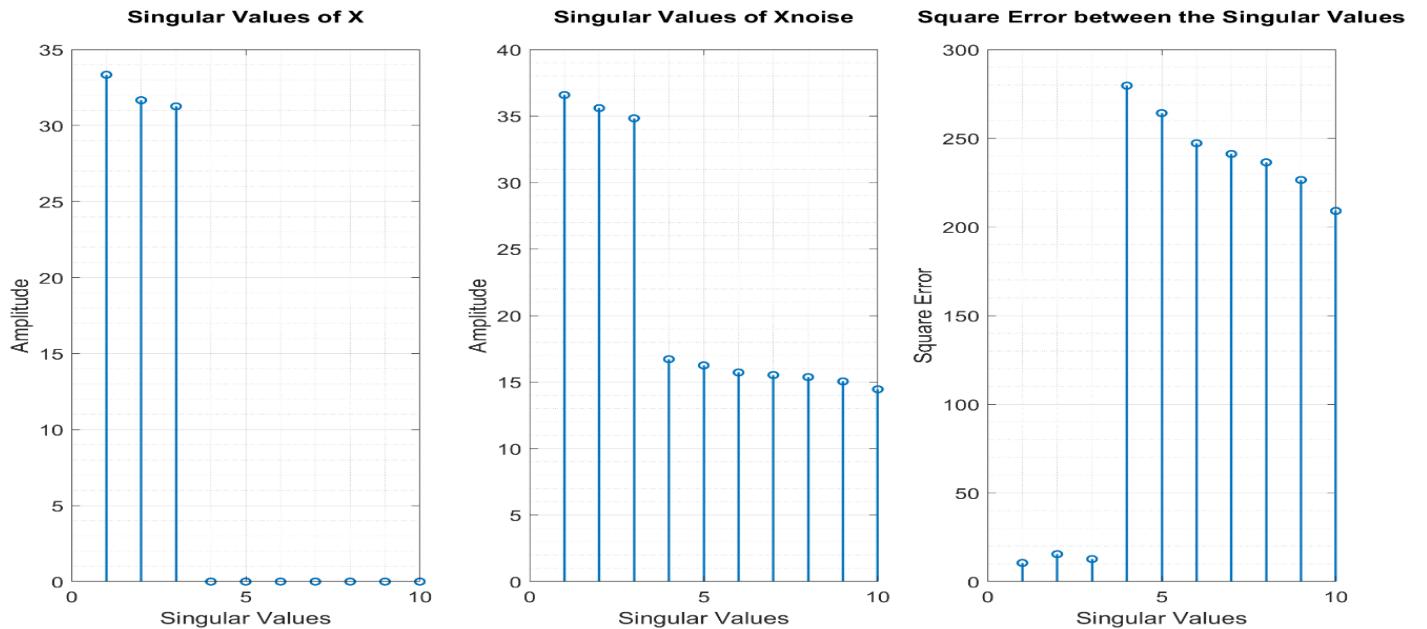


Figure 26: The singular values of X (left), X_{noise} (middle) and their square error.

- b) The mean squared error of the noiseless input matrix, the noisy input matrix and its' low-rank approximation can be found on Table 1. It is obvious that the error is smaller when the three most significant principal components of the noisy signal are used instead of the initial noisy signal. A useful insight is that the mean squared error of the noisy signal X_{noise} and its' low-rank approximation \tilde{X}_{noise} is equal to the difference of the mean squared errors of the noiseless signal and the two noisy signals.

Matrices	Mean Squared Error
$X - X_{noise}$	0.2435
$X - \tilde{X}_{noise}$	0.0733
$X_{noise} - \tilde{X}_{noise}$	0.1704

Table 1: The Mean Squared Error of the three different signals X, X_{noise} & \tilde{X}_{noise} .

- c) The estimation errors between Y & \hat{Y}_{OLS} & \hat{Y}_{PCR} are given on Table 2. Although one would believe that the Principal Component Regression (PCR) method would provide a better estimator than the Ordinary Least Squares (OLS) method, the results show exactly the opposite. When the test-set is used, while the regression coefficients $\hat{\mathbf{B}}_{OLS}$ & $\hat{\mathbf{B}}_{PCR}$ are computed using the training set, the PCR method performs better than the OLS one.

Matrices	Mean Squared Error
$\mathbf{Y} - \hat{\mathbf{Y}}_{OLS}$	0.7103
$\mathbf{Y} - \hat{\mathbf{Y}}_{PCR}$	0.7154
$\mathbf{Y}_{test} - \hat{\mathbf{Y}}_{test-OLS}$	0.4755
$\mathbf{Y}_{test} - \hat{\mathbf{Y}}_{test-PCR}$	0.4708

Table 2: The estimation error of the training and testing signal for the OLS and PCR methods.

- d) Due to the fact that in real world machine intelligence applications the models must be a good fit not only for the training data, but also for the test data, we make use of the regval script provided for the completion of the coursework. The following table shows that the PCR method performs better than the OLS, something that one could predict, since the OLS method introduces spurious correlations in the calculation of regression coefficients, as we know from theory.

Matrices	Mean Squared Error
$\mathbf{E}\{\mathbf{Y} - \hat{\mathbf{Y}}_{OLS}\}$	0.4715
$\mathbf{E}\{\mathbf{Y} - \hat{\mathbf{Y}}_{PCR}\}$	0.4666

Table 3: The estimation error of the new realization of the test data \mathbf{Y} and its' estimate $\hat{\mathbf{Y}}$ for the OLS and PCR methods.

2.4. Real World Signals: Respiratory Sinus Arrhythmia from RR-Intervals

- a) The standard periodogram of the three RRI trials are depicted in figure 27. It is obvious that the three trials provide different peaks in the frequency domain, since the breathing rate was different in each of them. Therefore, it is obvious that difference on the breathing rate can be reflected in the heart rate of an object.

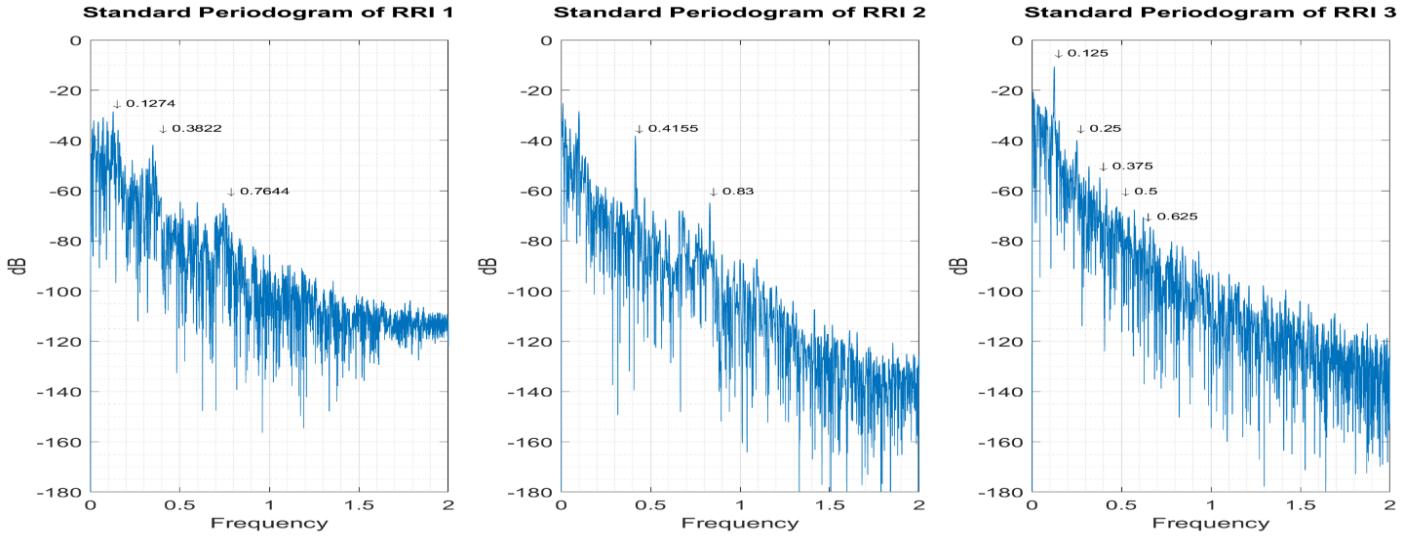


Figure 27: The Standard Periodogram of the three RRI signals.

The use of the average (Welch) periodogram can reduce the effect of noise which is very high in the standard periodogram, as one can observe in figure 27. In figure 28 the average periodogram of the three trials is depicted for window length of 50 seconds and 150 seconds. It is obvious that the window length of 50s provides with the neatest diagrams where the harmonics of the signal can still be observed.

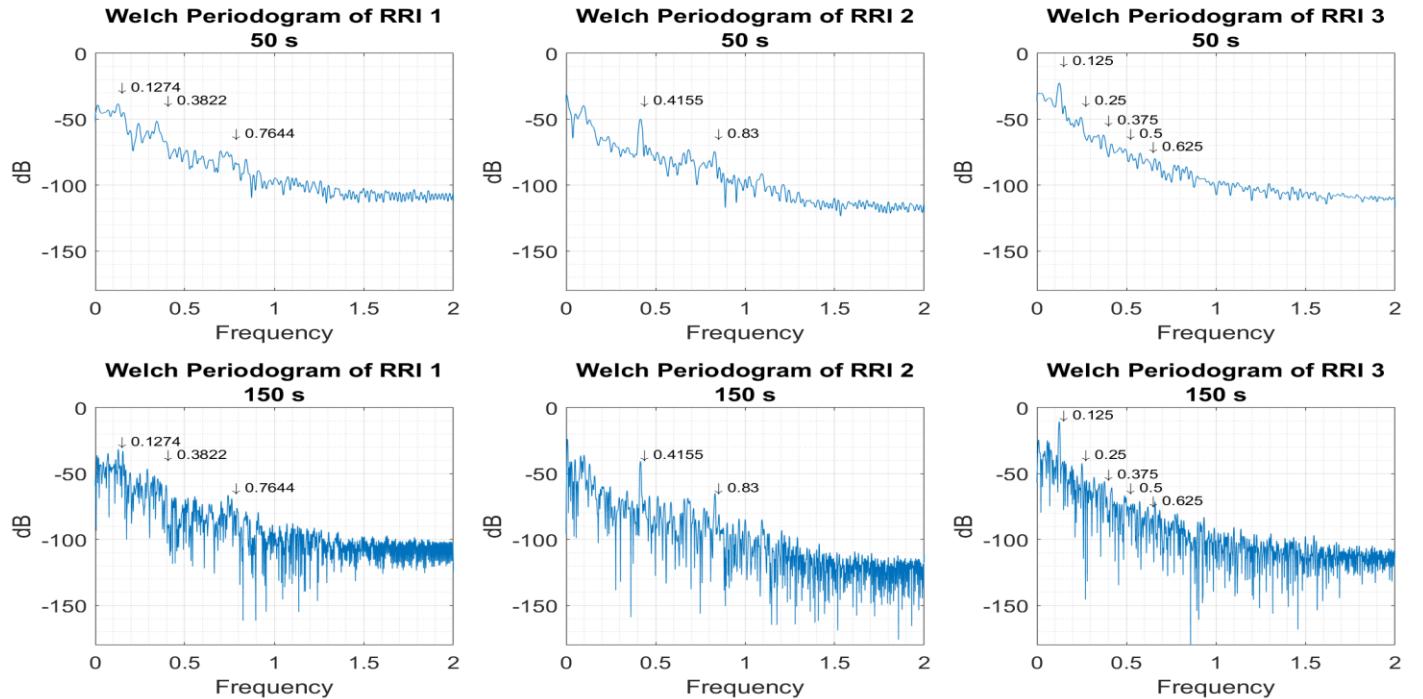


Figure 28: The average (Welch) periodogram of the three RRI signals for window lengths of 50s and 150s.

- b) Respiratory sinus arrhythmia (RSA) refers to the modulation of cardiac function by respiratory effort. During inspiration (“breathing in”) a speeding of the heart rate is observed, while during expiration (“breathing out”) a slowing down is detected.

In the first trial the breathing rate was not constrained and since the fundamental frequency can be found in 0.1274 Hz, the ventilation rate of the object is equal to $0.1274 \times 60 = 7.644$ Breaths Per Minute (BPM), which corresponds to a slow breathing rate. In the periodogram of the first RRI trial the third and the sixth harmonics of the fundamental frequency can be detected.

The second trial required the object to breath fast and this fact is reflected to the periodogram of the RR-Interval of the second trial. The fundamental frequency has a peak in 0.4155 Hz, meaning that the ventilation rate is equal to 24.93 BPM. That number is very close to the theoretical one of 25 BPM, in which the object was instructed to breathe in the experiment. Furthermore, a peak in the first harmonic of the fundamental frequency can be observed in 0.83 Hz.

Finally, the periodogram of the third trial has a sharp peak on 0.125 Hz, which is the fundamental frequency, while smaller peaks on the first, second, third and fourth harmonics can be observed. The fundamental frequency detected is equal to the value that one would predict to see, since the object was instructed to take 7.5 BPM, meaning that the frequency is equal to $7.5/60 = 0.125$ Hz.

- c) Figure 29 presents the Mean Squared Error of the AR estimate models of the three trials for model order $p \in [5,30]$. Although the MSE does not differ too much for these model orders, a minimum can be observed for $p_1 = 12, p_2 = 11$ & $p_3 = 9$ for the three trials respectively.

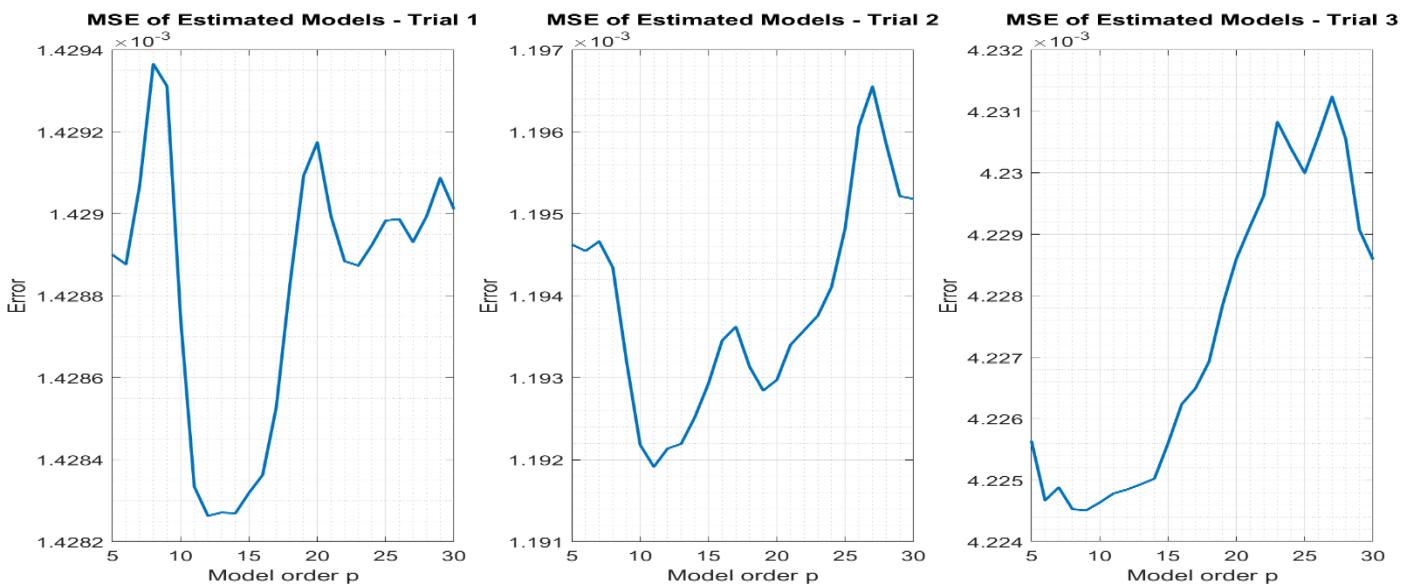


Figure 29: The Mean Squared Error of the AR Estimated Models of the three trials for different model orders.

Figure 30 presents the AR spectrum estimate of the RRI data from the three trials. For the first trial, model orders equal to 12, 16 & 22 have been used. Although it seems that all the three model orders fit well to the data, the AR(22) model provides the best spectrum estimate, since its complexity corresponds in the best way with the RRI signal.

For the second trial, the highest model order provides with the best approximation of the spectrum. Finally, for the third trial $p = 9$ & $p = 14$ agrees with the periodogram of the signal, while for high frequencies a significant error is observed.

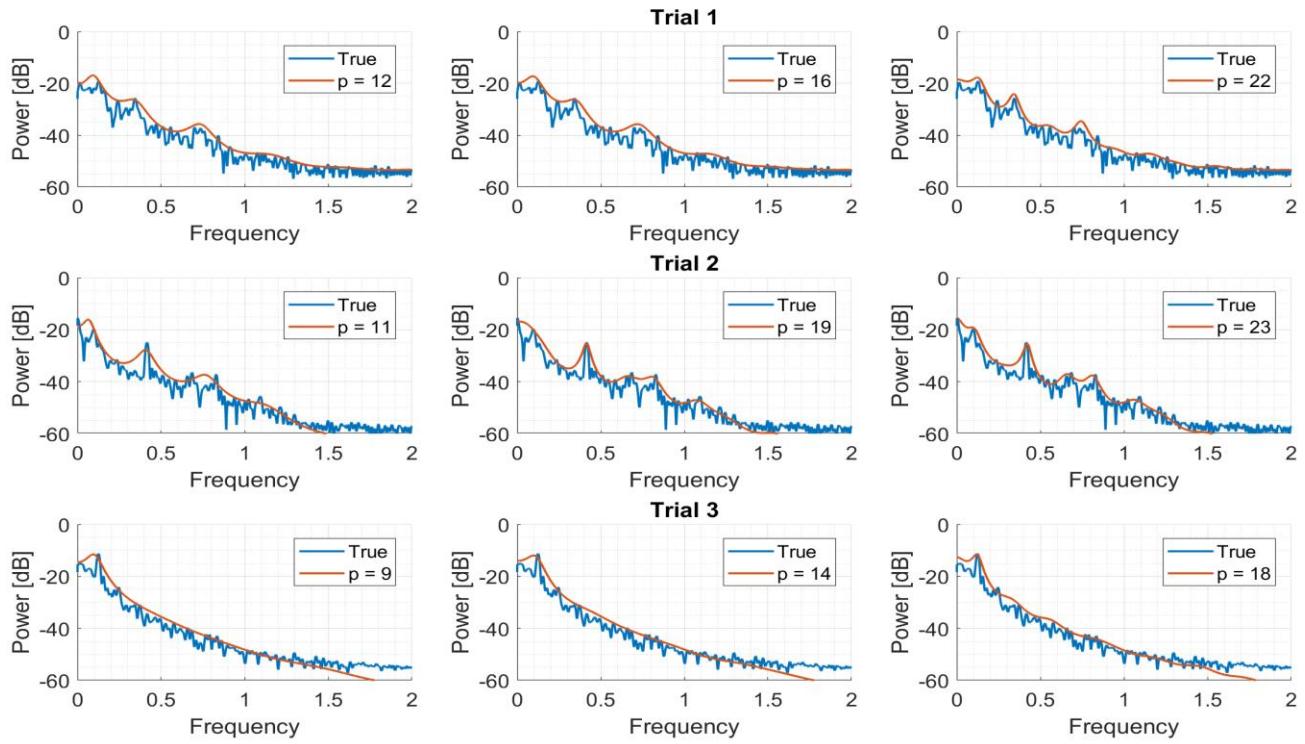


Figure 30: The AR Spectrum estimate for the three RRI signals for different model orders.

3. ADAPTIVE SIGNAL PROCESSING

3.1. The Least Mean Square (LMS) Algorithm

- a) The entries of the autocorrelation matrix of the input vector $\mathbf{x}(n) = [x(n-1), x(n-2)]^T$ are given below:

$$\mathbf{R}_{xx} = E\{\mathbf{x}\mathbf{x}^T\} = \begin{bmatrix} E\{x(n-1)x(n-1)\} & E\{x(n-1)x(n-2)\} \\ E\{x(n-2)x(n-1)\} & E\{x(n-2)x(n-2)\} \end{bmatrix} = \begin{bmatrix} r_{xx}(0) & r_{xx}(1) \\ r_{xx}(1) & r_{xx}(0) \end{bmatrix} \quad (3.1.1)$$

The $r_{xx}(0)$ & $r_{xx}(1)$ can be found by using the difference equation of the second-order auto-regressive process, where the expectation is taken after multiplying by $x(n-l)$ with $l = 0, 1, 2$ in order to form three equations. Thus:

$$\begin{aligned} r_{xx}(l) &= E\{x(n)x(n-l)\} = E\{a_1x(n-1)x(n-l) + a_2x(n-2)x(n-l) + \eta(n)x(n-l)\} \Rightarrow \\ r_{xx}(l) &= a_1r_{xx}(l-1) + a_2r_{xx}(l-2) + r_{\eta\eta}(l) \end{aligned} \quad (3.1.2)$$

Therefore, by solving the system of equations below, the entries of the autocorrelation matrix can be determined.

$$\begin{aligned} r_{xx}(0) &= a_1r_{xx}(1) + a_2r_{xx}(2) + \sigma_\eta^2 \\ r_{xx}(1) &= a_1r_{xx}(0) + a_2r_{xx}(1), \\ r_{xx}(2) &= a_1r_{xx}(1) + a_2r_{xx}(0). \end{aligned} \quad (3.1.3)$$

By solving the above system of equations, the result is that $r_{xx}(0) = \frac{25}{27}$ & $r_{xx}(1) = \frac{25}{54}$. Thus,

$$\mathbf{R}_{xx} = \begin{bmatrix} \frac{25}{27} & \frac{25}{54} \\ \frac{25}{54} & \frac{25}{27} \end{bmatrix}, \text{with } \lambda_1 = \frac{25}{18}, \lambda_2 = \frac{25}{54} \quad (3.1.4)$$

Finally, since the step size μ must be less than two over the maximum eigenvalue of the autocorrelation matrix, in order to achieve convergence in the mean with the LMS algorithm, the range of its' values are given below:

$$0 < \mu < 1.44 \quad (3.1.5)$$

- b) In figure 31 the learning curve and the ensemble average learning curve of the LMS adaptive predictor for an AR(2) process are presented.

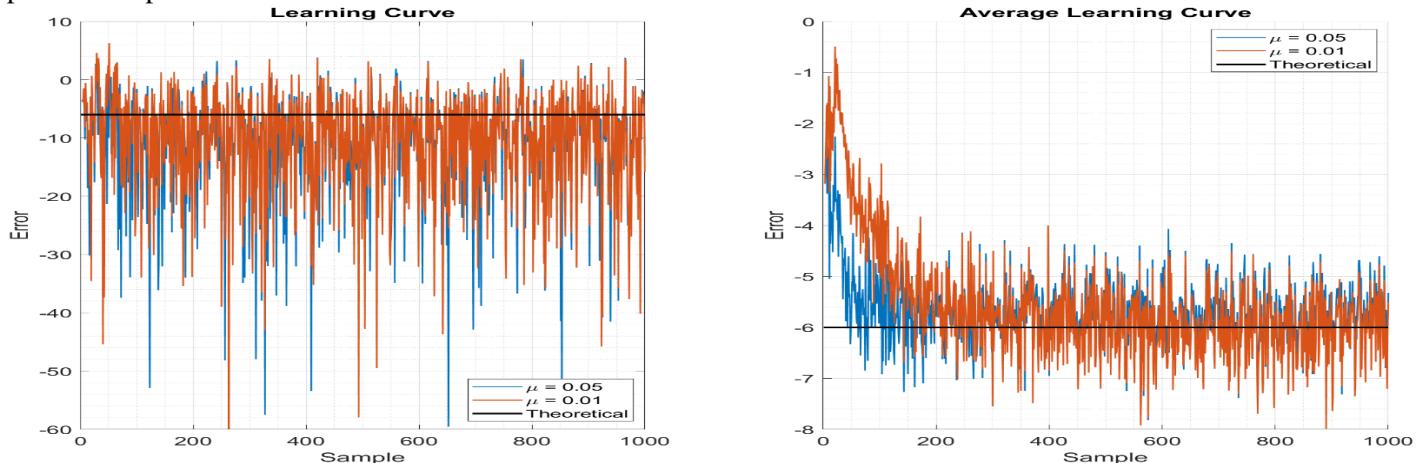


Figure 31: The learning curve and the average learning curve for the LMS adaptive predictor for $\mu = 0.05$ & $\mu = 0.01$.

The adaptation gain is equal to 0.05 and 0.01, values much smaller than the theoretical upper bound discussed in 3.1.a. The learning curve of one realization of the process (Fig.31 left) has very high variance and thus it is not so efficient to provide insights about the convergence and generally the performance of the predictor. However, the squared prediction error seems to fluctuate rapidly around the theoretical MSE, which is equal to the power of the noise that drives the AR process. The average learning curve (Fig.31 right) is much more capable of showing the differences between the two adaptation gains. Firstly, the variance is much slower in that case, since the squared error fluctuates much more slightly around the theoretical MSE which is equal to -6 dB since the power of the noise is equal to 0.25. Furthermore, it is obvious that the adaptation gain equal to 0.05 provides with faster convergence than the higher one. Additionally, the smaller adaptation gain seems to result in slightly higher errors, since the blue plot lies above the orange one on steady state. Finally, the insights would have been even more certain if the number of different realizations was even bigger, i.e. 1000 instead of 100.

- c) Table 4 shows the Excess Mean Square Error (EMSE), the corresponding misadjustment (\mathcal{M}) and the approximated misadjustment (\mathcal{M}_{LMS}). As discussed in 3.1.b the error is higher for bigger adaptation gains. This fact is reflected on the EMSE, which is the difference of the MSE and the power of the noise on steady state and agrees with the findings of figure 31. Therefore, the tradeoff of error and convergence is confirmed one more time, because fast convergence requires higher error and vice versa. The corresponding misadjustment is bigger for bigger adaptation gains since it is proportional to the EMSE. Furthermore, the approximation of the misadjustment agrees with the theoretical one with the insights that it provides, but it gives smaller values for both step sizes. Finally, as it is known from theory the approximation of the misadjustment is closer to the theoretical for smaller step sizes in terms of how close is its' value to the theoretical one.

μ	EMSE	\mathcal{M}	\mathcal{M}_{LMS}
0.05	0.0151	0.0604	0.0463
0.01	0.0034	0.0135	0.0093

Table 4: The EMSE, the corresponding Misadjustment and the approximated Misadjustment of the LMS algorithm.

- d) Figure 32 shows the evolution of the estimates of the AR coefficients for the two step sizes discussed above. The conclusions made in the previous questions can be verified here too. $\mu = 0.05$ provides with faster convergence but higher error, while $\mu = 0.01$ approximates better the coefficients although it is slower.

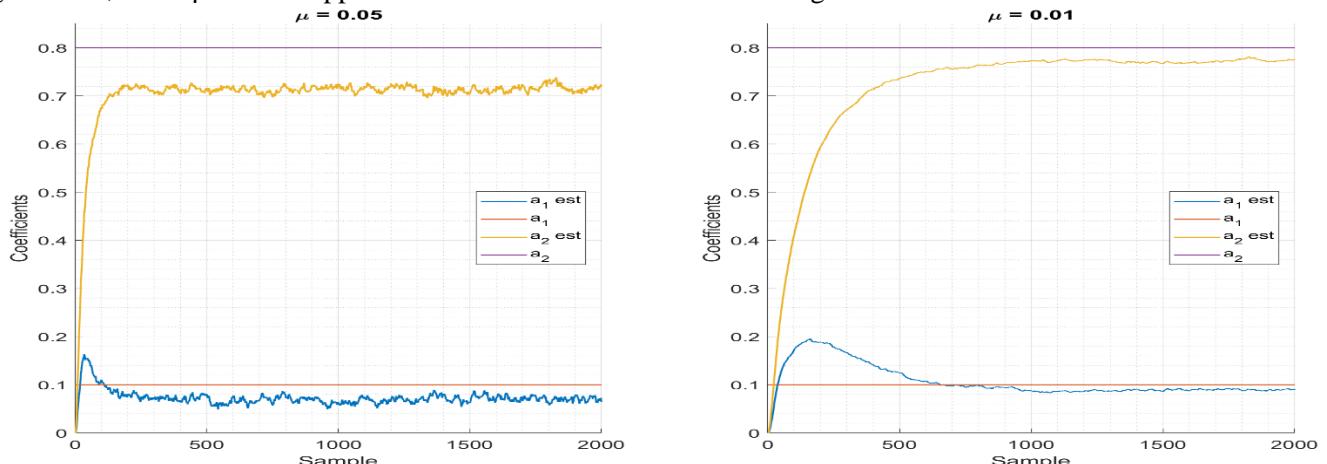


Figure 32: The evolution of the Weight Coefficients of the LMS predictor for $\mu = 0.05$ & $\mu = 0.01$.

Table 5 summarizes the steady state estimates of the adaptive filter coefficients for the two different step sizes. As discussed on figure 32 and in the previous questions, the smaller adaptation gain provides with better estimates of the AR coefficients, since it achieves smaller MSE and EMSE (3.1.c). This conclusion can be also derived by the last two columns of table 5, where the difference of the AR coefficients and the adaptive filter coefficients is presented.

μ	\hat{a}_1	\hat{a}_2	$a_1 - \hat{a}_1$	$a_2 - \hat{a}_2$
0.05	0.07	0.7143	0.03	0.0857
0.01	0.0885	0.7722	0.0115	0.0278

Table 5: The steady state values of the adaptive filter coefficients for the step sizes of $\mu = 0.05$ & $\mu = 0.01$.

- e) The leaky LMS algorithm minimizes the cost function $J_2(n) = \frac{1}{2}(e^2(n) + \gamma\|\mathbf{w}(n)\|_2^2)$. The update equation for the steepest descent method to find the minimum of J_2 is given by:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu(-\nabla J_2|_{\mathbf{w}(n)}) \quad (3.1.6)$$

Since $e^2(n) = (x(n) - \hat{x}(n))^2 = (x(n) - \mathbf{w}^T(n)x(n))^2$, the cost function becomes

$$J_2(n) = \frac{1}{2}\left(\left(x(n) - \mathbf{w}^T(n)x(n)\right)^2 + \gamma\|\mathbf{w}(n)\|_2^2\right) \quad (3.1.7)$$

and therefore

$$-\nabla J_2(n)|_{\mathbf{w}(n)} = -\frac{1}{2}\left(\nabla\left(x(n) - \mathbf{w}^T(n)x(n)\right)^2 + \gamma\nabla\|\mathbf{w}(n)\|_2^2\right) = -\frac{1}{2}\left(2(-x(n))(x(n) - \mathbf{w}^T(n)x(n)) + 2\gamma\mathbf{w}(n)\right).$$

Thus:

$$-\nabla J_2(n)|_{\mathbf{w}(n)} = \mathbf{x}(n)e(n) - \gamma\mathbf{w}(n) \quad (3.1.8)$$

and

$$\begin{aligned} \mathbf{w}(n+1) &= \mathbf{w}(n) + \mu(\mathbf{x}(n)e(n) - \gamma\mathbf{w}(n)) \Rightarrow \\ \mathbf{w}(n+1) &= (1 - \mu\gamma)\mathbf{w}(n) + \mu e(n)\mathbf{x}(n). \end{aligned} \quad (3.1.9)$$

- f) Figure 33 shows the results of the leaky LMS predictor for an AR(2) process. It is obvious that as the leakage coefficient increases, the error of the a_2 estimates increases for both step sizes. As far as the first order parameter of the process is concerned, for $\mu = 0.05$ its' estimate seems to have the smaller error for $\gamma = 0.1$, while for $\mu = 0.01$ the convergence is faster and the steady state error is bigger as the leakage parameter increases.

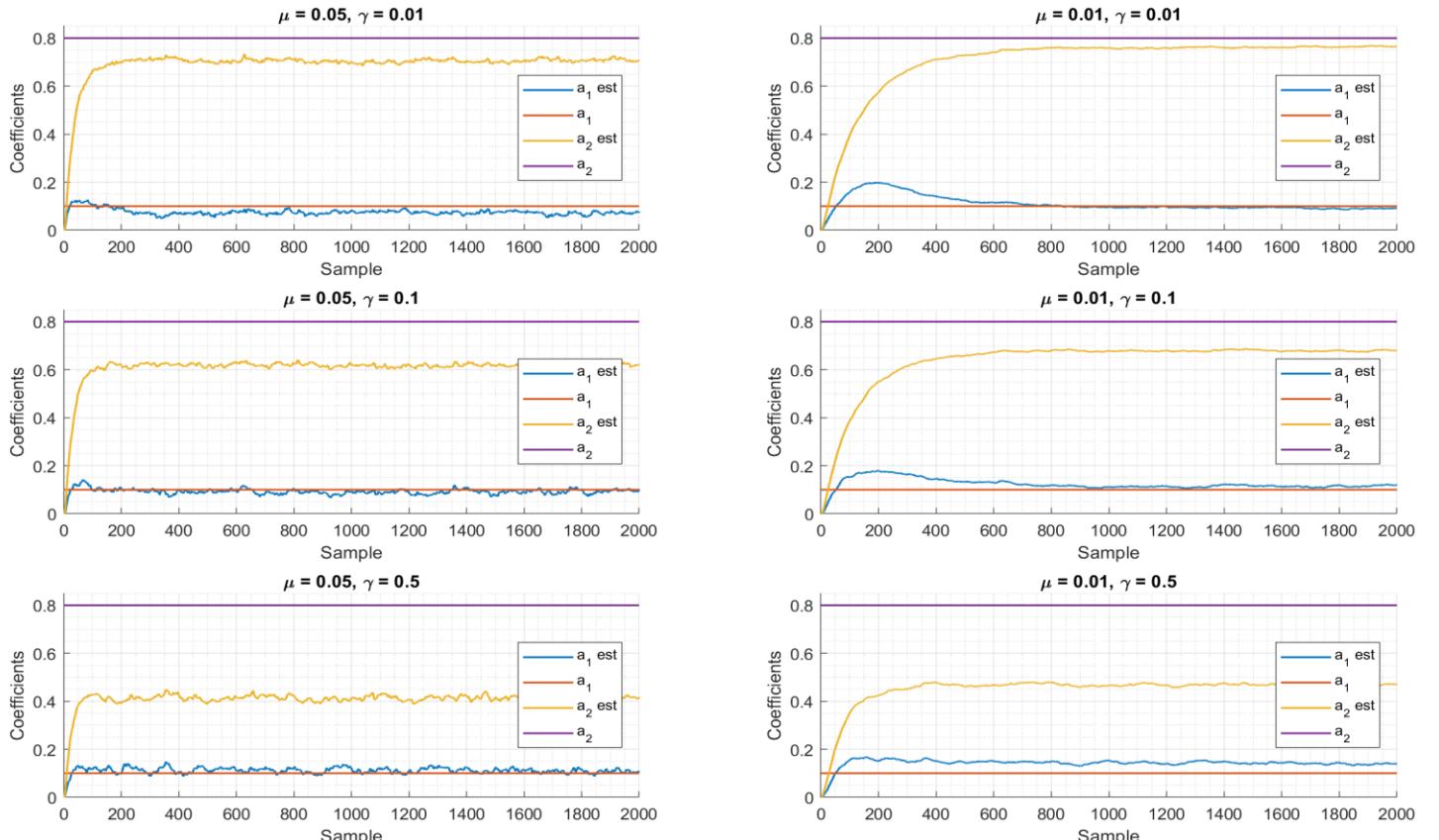


Figure 33: The AR coefficient estimates of the leaky LMS algorithm for different μ & γ values.

3.2. Adaptive Step Sizes

- a) The choice of the adaptation gain for the LMS algorithm is one of the most challenging and difficult tasks when designing an adaptive filter, because it depends on the signal or the process that it is going to adapt to. Additionally, the trade-off between convergence speed and steady state error variance is a major problem, as discussed and shown in 3.1. Therefore, the step size needs to be large in the beginning of the adaptation and it should become smaller as the filter estimates approach the true values. Thus, a time-varying step size is the best solution for achieving best performance for both convergence and steady state error.

As a result of the above concerns, variable step-size (VSS) algorithms have been implemented, which simultaneously solve two unconstrained optimization procedures. These algorithms minimize the cost function with respect to the weight vector as well as to the step size in the same time. Therefore, the LMS algorithm has the following form:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu(n)e(n)\mathbf{x}(n) \quad (3.2.1)$$

where the adaptation gain has become adaptive utilizing the gradient adaptive step-size (GASS) algorithms. The learning rate is updated according to:

$$\mu(n+1) = \mu(n) + \rho e(n)\mathbf{x}^T(n)\boldsymbol{\psi}(n). \quad (3.2.2)$$

For that question of the coursework, three different GASS algorithms have been implemented, which were introduced by Benveniste, Ang & Farhang and Matthews & Xie. The vector $\boldsymbol{\psi}$ takes different forms for the three different GASS algorithms, as presented in the coursework assignment and at the lectures. By analyzing the implementation of that vector, one can realize that the Matthews and Xie implementation is the least computational expensive, while the Benveniste implementation is the most computational expensive, since it requires two matrix/vector multiplications and two matrix/vector additions. Therefore, one can assume that Benveniste GASS algorithm has the best performance, Ang & Farhang provides with a trade-off between computational complexity and performance, while Matthews and Xie is the least time demanding algorithm with the worst performance among the three algorithms.

Figure 34 depicts the weight error of the three GASS LMS algorithms and of the LMS algorithm for an MA(1) process. Different ρ , α & μ parameters have been used for the different algorithms and a comparison of the best GASS algorithms with the LMS is presented in figure 34 bottom-right. To begin with, the GASS Benveniste algorithm has only one parameter, ρ , and as one can observe that parameter should be relatively small of order 10^{-3} in order to achieve convergence (Fig. 34 top-left). In addition to this, $\rho = 0.003$ provides with the best convergence speed and the least weight error and therefore is the best choice.

The Ang & Farhang algorithm has two different parameters that need to be decided. The weight error of that GASS algorithm is depicted in fig.34 top-right, where different values of the parameters ρ & α have been used. Best performance, i.e. fast convergence and low steady state error, is achieved for $\rho = 0.025$ & $\alpha = 0.5$.

The Matthews and Xie algorithm is examined in figure 34 bottom-left. That algorithm has only one degree of freedom, which is the ρ parameter and same values with the ones from the Benveniste algorithm have been used in order to achieve convergence. In that case, the bigger value of the parameter achieves the best performance. However, comparing with the two prior algorithms the performance is worse since convergence is not fast and the steady state error is bigger. Therefore, the assumption made above about the performance of the algorithms is verified with the results of the simulations shown in figure 34.

Concludingly, figure 34 bottom-right compares the three GASS algorithms -using their best parameters ρ & α - with the LMS algorithm for $\mu = 0.1$ & $\mu = 0.01$. Matthews & Xie and LMS(0.01) algorithms converge very slowly and therefore, they do not provide good estimates for MA(1) processes. LMS(0.1) provides with the fastest convergence among the algorithms, but as mentioned in previous questions the steady state error is significantly big. Benveniste and Ang & Farhang algorithms provide with the smallest steady state error, with Benveniste having the smallest divergence, conforming with the assumptions made above. Finally, Ang & Farhang algorithm converges faster than the Benveniste one, but its' steady state error is roughly bigger. This result can be derived more easily by using a dB scale for the weight error diagram.

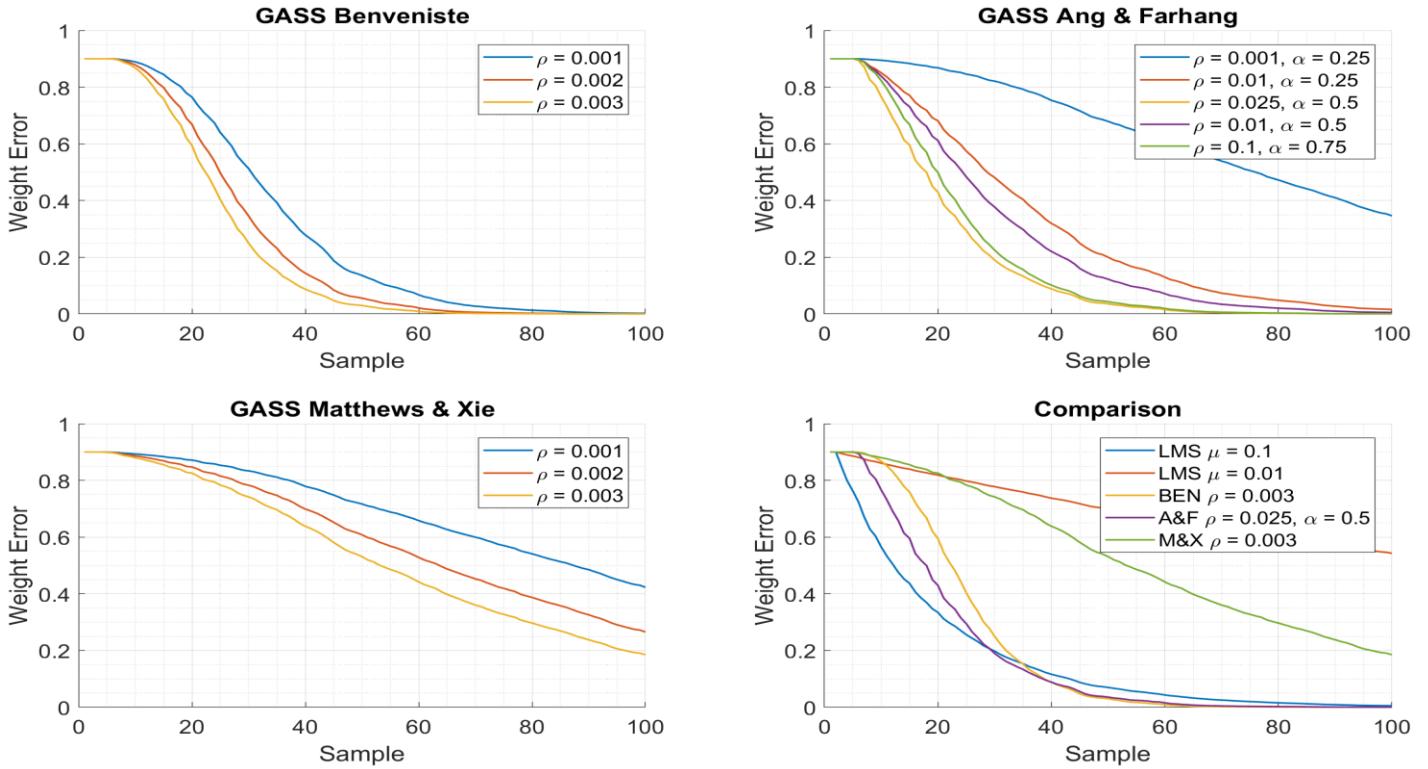


Figure 34: The Weight Error of the three GASS algorithms and the LMS algorithm for different parameters of an MA(1) process.

b) The update equation of the NLMS algorithm is the following one:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\beta}{\epsilon + \|\mathbf{x}\|^2} e(n) \mathbf{x}(n) \quad (3.2.1)$$

Using the a posteriori error the update equation is:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e_p(n) \mathbf{x}(n) \quad (3.2.2)$$

By multiplying both sides by $\mathbf{x}^T(n)$ and then adding the desired response $d(n)$ we have:

$$d(n) - \mathbf{x}^T(n) \mathbf{w}(n) = d(n) - \mathbf{x}^T(n) \mathbf{w}(n+1) + \mu e_p(n) \|\mathbf{x}\|^2 \quad (3.2.3)$$

Thus:

$$e(n) = e_p(n) + \mu e_p(n) \|\mathbf{x}\|^2 = e_p(n)(1 + \mu \|\mathbf{x}\|^2) \Rightarrow e_p(n) = \frac{e(n)}{1 + \mu \|\mathbf{x}\|^2} \quad (3.2.4)$$

By substituting (3.2.4) to (3.2.2) the a posteriori error update equation is:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\mu e(n)}{1 + \mu \|\mathbf{x}\|^2} \mathbf{x}(n) \Rightarrow \mathbf{w}(n+1) = \mathbf{w}(n) + \frac{e(n)}{\frac{1}{\mu} + \|\mathbf{x}\|^2} \mathbf{x}(n) \quad (3.2.5)$$

Therefore, the update equation based upon the a posteriori error is equivalent to the one used in the NLMS algorithm and the parameters have the following relationship:

$$\beta = 1, \epsilon = \frac{1}{\mu} \quad (3.2.6)$$

- c) The normalized LMS (NLMS) algorithm normalizes the adaptation gain with respect to the input vector $\mathbf{x}(n)$. The NLMS algorithm is useful for real-world applications, because it makes the adaptive filter independent of the power in the tap input. However, a regularization factor ϵ is vital for that algorithm, since the filter may become unstable if the input vector is zero.

That regularization factor is constant most of the times, and its choice is as challenging as having to decide for the value of the step size of the classic LMS algorithm. Thus, algorithms that make the gain of the NLMS algorithm adaptive have been introduced, making the regularization factor time-varying. One of the most popular algorithms is the Generalized Normalized Gradient Descent (GNGD) algorithm, which minimizes the cost function with respect to the regularization factor along to the weight vector.

Figure 35 shows the weight error curves and the weight estimates of the GNGD algorithm and the Benveniste's GASS algorithm. Examining the left figure, one can realize that the convergence of the GNGD algorithm is extremely faster than the Benveniste's one and that the steady state error variance is zero. The right figure provides with the same results where the weight estimates of the MA(1) process are presented. It is obvious that the GNGD algorithm has the best performance among the LMS, NLMS and GASS algorithms, considering the results of question 3.2.a.

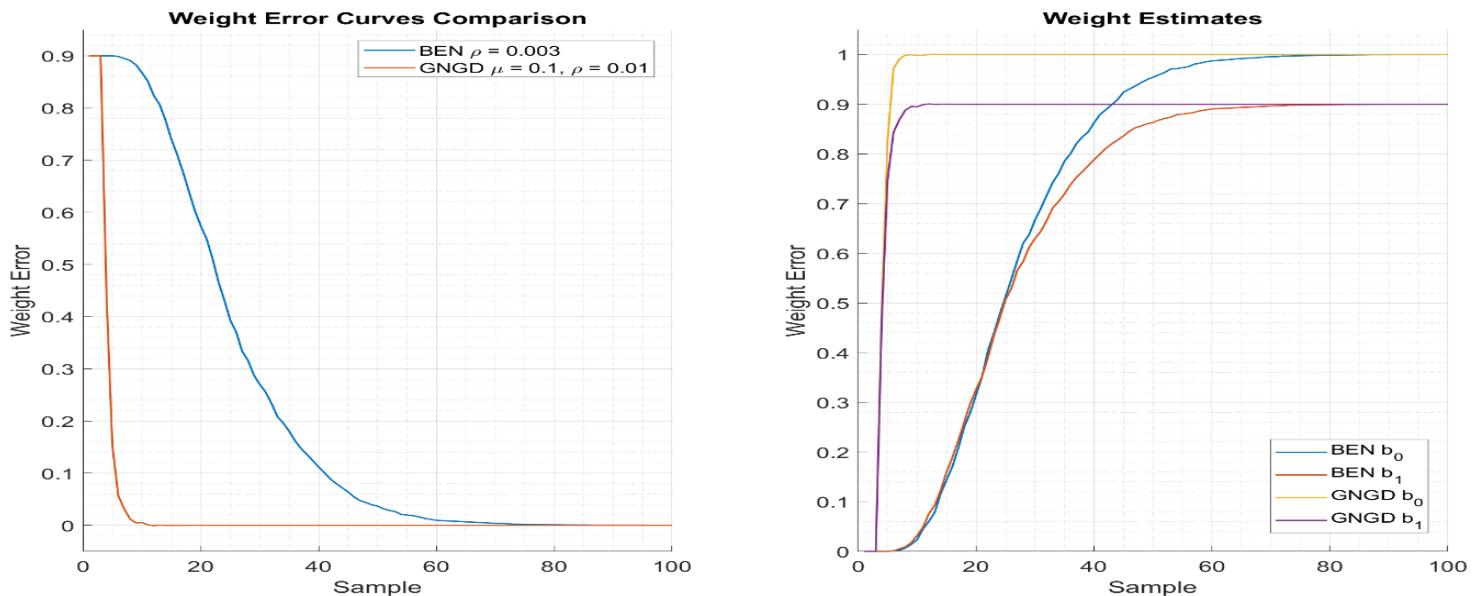


Figure 35: Weight error curves comparison and weight estimates of the GNGD algorithm and the Benveniste's GASS algorithm for an MA(1) process.

Having shown that the GNGD algorithm provides with the best performance for the system identification problem of an MA(1) process, the computational complexity among the two algorithms above is compared. Table 6 compares the computational complexity of the two algorithms in terms of number of multiplications and additions at every time instant, where L is the order of the adaptive filter. It is obvious that the GNGD has less additions and multiplications than Benveniste's algorithm and therefore it is the best choice for the system identification problem since it provides with fast convergence, negligible steady state error and significantly small computational complexity.

Algorithm	Additions	Multiplications
Benveniste's GASS	$5L$	$2L^2 + 5L + 2$
GNGD	$4L$	$4L + 8$

Table 6: Computational Complexity of the Benveniste's GASS algorithm and the GNGD algorithm. L is the order of the adaptive filter.

3.3. Adaptive Noise Cancellation

- a) The Mean Square Error of the adaptive line canceller is the following:

$$\begin{aligned} MSE_{ALE} &= E\{(s(n) - \hat{x}(n))^2\} = E\{(x(n) + \eta(n) - \hat{x}(n))^2\} \Rightarrow \\ MSE_{ALE} &= E\{\eta(n)^2\} + E\{(x(n) - \hat{x}(n))^2\} + 2E\{\eta(n)x(n) - \eta(n)\hat{x}(n)\} \end{aligned} \quad (3.3.1)$$

Since $x(n)$ and $\eta(n)$ are not correlated and $E\{\eta(n)^2\}$ cannot be controlled, the MSE is only affected by the noise in the following term:

$$E\{\eta(n)\hat{x}(n)\} = E\left\{\left(\eta(n)\left(\sum_{k=0}^M w_k(n)s(n-\Delta-k)\right)\right)\right\} \quad (3.3.2)$$

$$E\{\eta(n)\hat{x}(n)\} = E\left\{\left((v(n) + 0.5v(n-2))\left(\sum_{k=0}^M w_k(n)(x(n-\Delta-k) + \eta(n-\Delta-k))\right)\right)\right\} \quad (3.3.3)$$

Since $v(n)$ & $x(n)$ are uncorrelated we have:

$$E\{\eta(n)\hat{x}(n)\} = E\left\{\left((v(n) + 0.5v(n-2))\left(\sum_{k=0}^M w_k(n)(v(n-\Delta-k) + 0.5v(n-\Delta-k-2))\right)\right)\right\} \Rightarrow \quad (3.3.4)$$

$$E\{\eta(n)\hat{x}(n)\} = E\left\{\left(\left(\sum_{k=0}^M w_k(n)(v(n) + 0.5v(n-2))(v(n-\Delta-k) + 0.5v(n-\Delta-k-2))\right)\right)\right\} \quad (3.3.5)$$

Therefore, the MSE is minimized when the autocorrelation of $\eta(n)$ & $x(n)$ is equal to zero. This is true when $\Delta \geq 3$, because $v(n)$ is white noise, meaning that $E\{v(i)v(j)\} = 0 \forall i \neq j$.

The above choice of minimum Δ can be justified by simulating the adaptive line enhancement configuration using the LMS algorithm for the noise-corrupted signal $s(n) = x(n) + \eta(n)$. Figure 36 shows the estimated sinusoidal signal for different delays.

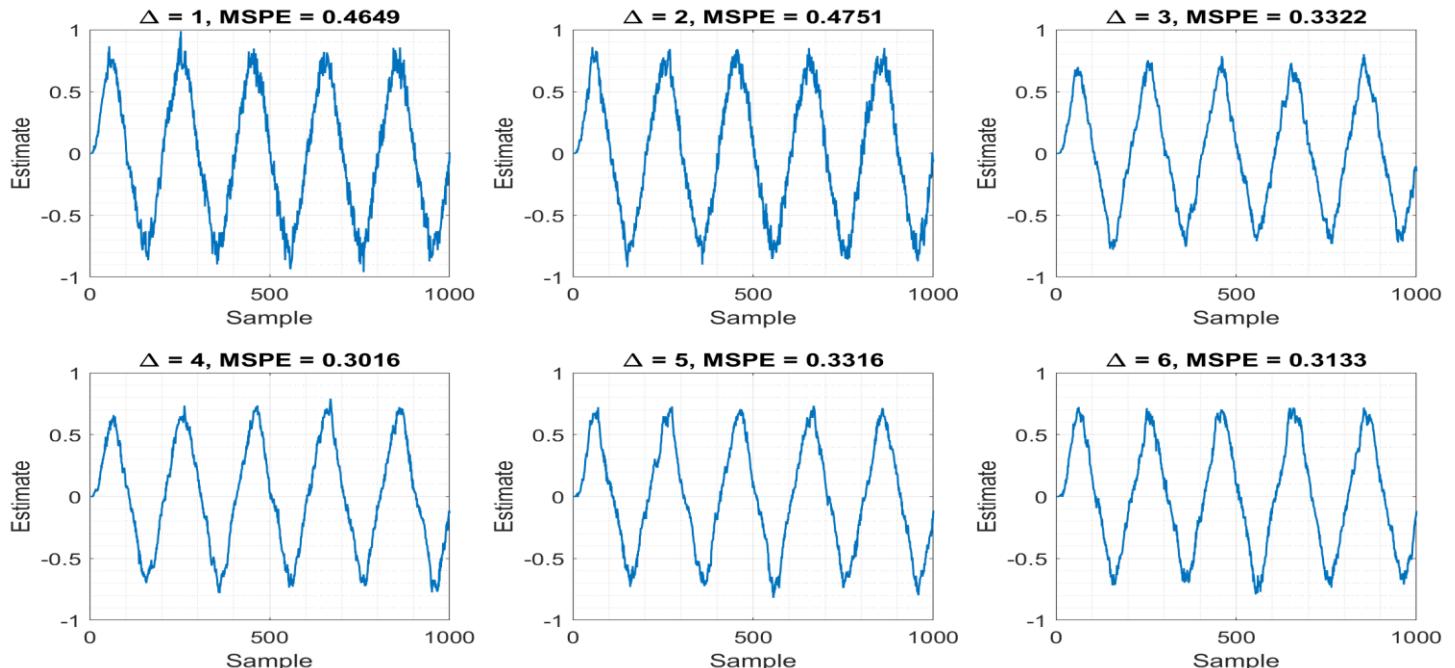


Figure 36: The estimated output $\hat{x}(n)$ of the ALE for $\Delta \in [1, 6]$ and the corresponding MSPEs.

It is obvious that when the delay is equal to three, the mean square prediction error (MSPE) has a rapid decrease, which reflects the minimization of the autocorrelation of $\eta(n)$ & $\hat{x}(n)$. As the delay Δ increases, the MSPE decreases most of the times, but a monotonically decrease is not observed. In the next question the relationship between the delay and the MSPE is discussed.

- b) In this question the effect of the delay Δ of the adaptive line enhancer configuration and the filter order M of the LMS FIR adaptive filter is examined. Generation of 1000 noisy-corrupted sinusoids has been implemented in order to reassure that the average MSPE has minimized the variance introduced by the finite implementation of the configuration.

Figure 37 shows the Mean Squared Prediction Error (MSPE) vs the delay $\Delta \in [3,25]$ and the filter order $M \in \{5,10,15,20\}$. By examining the diagram, one can realize that as the delay increases after the value of 3, because when it is equal to 1 & 2 the MSPE has higher values, the MSPE is a monotonically increasing function until the delay is equal to 25. Although one would believe that the MSPE would become smaller as the delay increases, the simulation tells us that this is not real. That fact is caused by the nature of the signal, since it is a sinusoid of fixed frequency. Therefore, by delaying the noisy-corrupted signal too much the difference between $x(n)$ & $\hat{x}(n)$ will become bigger and therefore the error will increase. However, since the signal is periodic one can assume that the MSPE will not increase forever. That happens because after a certain delay the real and the predicted signals will be in phase again and therefore the MSPE will become the minimum again. Thus, the MSPE is a periodic function of Δ , but the periodicity is higher than 25, since the length of the signal $x(n)$ is bigger, meaning that in the following diagram this assumption cannot be verified. Concludingly, $\Delta = 3$ is the best choice for the delay, since it provides with the smaller prediction error as well as the implementation of the filter in hardware is always easier, cheaper and more convenient for smaller delays.

The effect of the filter order M on the MSPE is presented in figure 37, too. It is obvious that as the filter order increases, the MSPE increases too. This fact is also counterintuitive as the previous one, but if someone tries to think a little bit more what causes that effect, they will understand that this fact should be anticipated. Although, in theory we know that higher filter orders provide with better accuracy and bigger noise suppression, in practice we see that this fact is not met. That happens because there exist coefficient and arithmetic precision errors, because the values of the coefficients have finite precision and arithmetic calculations introduce noise, meaning that they are not exact. Therefore, as verified by the following diagram, higher order filters should be avoided for a lot of reasons, except from the fact that they increase the prediction error. Except from the above, higher order filters increase the computational cost. Thus, $M = 5$ is the best choice for the filter order, since it provides with small prediction error, small computational complexity and, also, its' implementation is cheaper and easier in hardware.

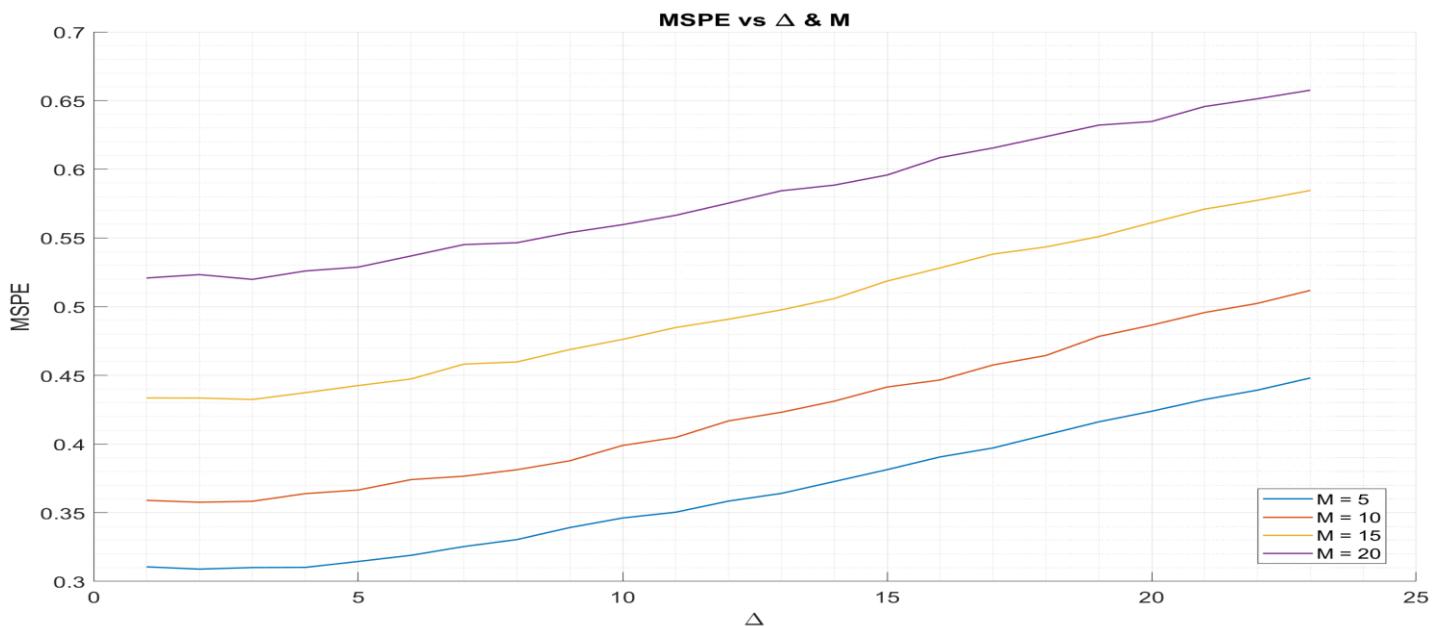


Figure 37: The effect of the delay Δ and the filter order M on the MSPE.

- c) The Adaptive Noise Cancellation (ANC) configuration differs from the Adaptive Line Enhancer (ALE) since it does not make use of a delay, but it needs to have a second sensor that senses noise correlated to the one that corrupts the primary input. That configuration is more difficult to implement in hardware but it can achieve better performance as it is clear in the following figures.

Figure 38 (left) shows the MSPE of the ANC and the ALE($\Delta = 3$) configurations for different filter orders. As discussed in the previous question, the increase in the order of the filter causes increase in the prediction error too. It is obvious that the ANC configuration achieves smaller errors and therefore is a better noise cancellation configuration in terms of accuracy. In fig.38 (right) the predicted output of the ANC configuration is depicted. Comparing with the one from fig.36 (3.3.a) the better accuracy of the ANC configuration is verified, since the noise is suppressed in a bigger extent than in the ALE configuration.

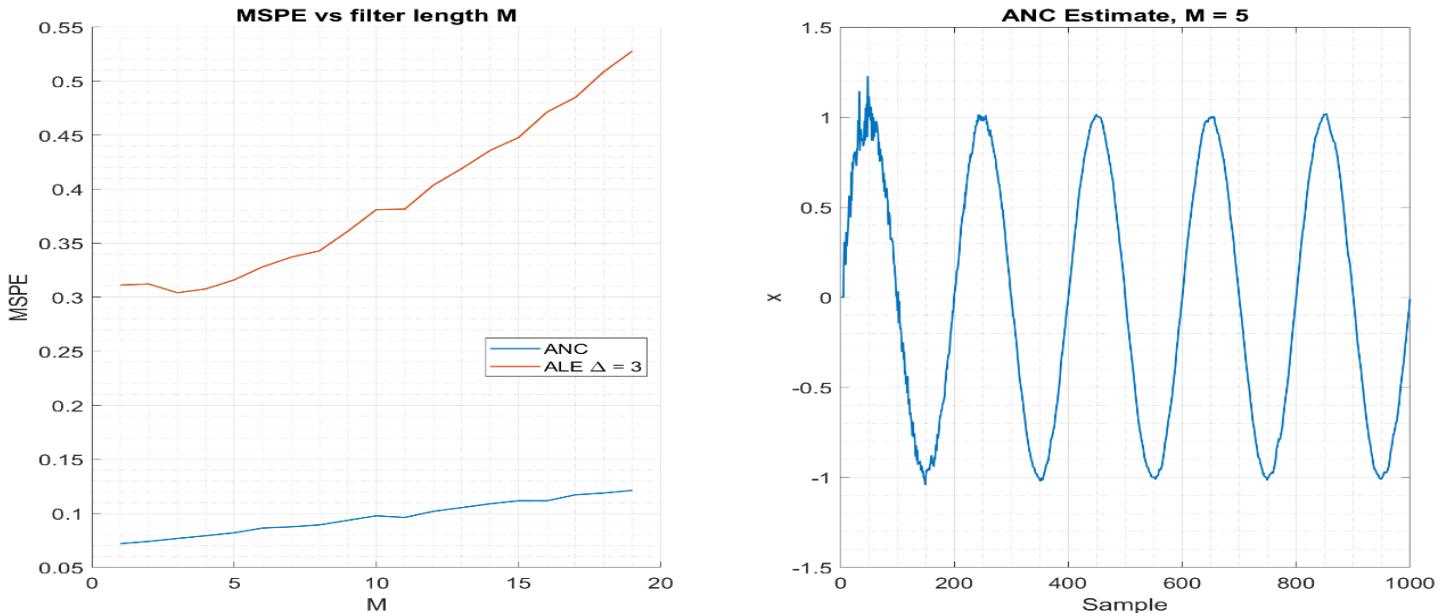


Figure 38: The MSPE for the two different de-noising adaptive filtering configurations (left). The ANC estimated signal for an adaptive filter of order $M = 5$ (right).

- d) In question 1.4.b the basis for brain computer interface (BCI) has been examined. An electroencephalogram (EEG) signal from the posterior/occipital (POz) region of the head was analyzed in the time/frequency domain. There, a power-line interference was found, which corrupted the signal. Having implemented adaptive noise cancellation configurations using the LMS algorithm we know that we can suppress the noise from noise-corrupted signals with high efficiency. In order to denoise the EEG data, the ANC configuration is utilized, since it can achieve smaller prediction error than the ALE as shown in the previous question.

In figure 39 the noise-corrupted POz signal and the POz denoised estimates are presented for different learning rates, while the filter order is equal to 10. A hanning filter of order $M = 4096$ has been used and the best overlap that was found is $overlap = 0.75M = 0.75 \times 4096 = 3072$ samples. It is obvious from the first two spectrograms that even for a very small step size the noise has been significantly suppressed.

Furthermore, as the step size increases the noise is increasingly cancelled and for $\mu = 0.1$ the noise is not noticeable in the spectrogram at all. Therefore, this adaptive signal processing configuration is capable to suppress the noise in a very small region of the frequency domain, without distorting other useful signals recorded. In the spectrograms of figure 39 it is clear that the frequencies 0 to 25 have remained unchanged and, thus, there is no loss of useful information for the steady state visual evoked potentials (SSVEP).

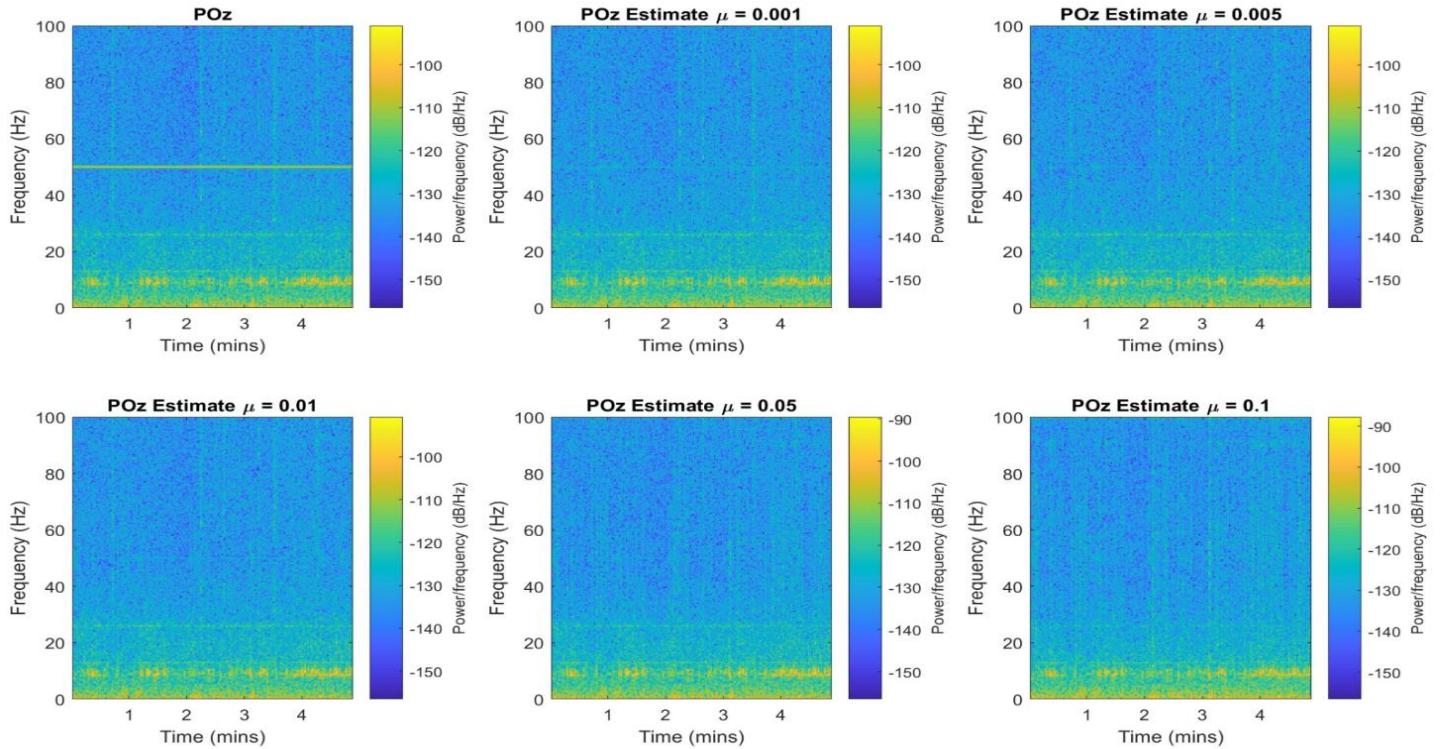


Figure 39: The POz Estimate after removing the 50 Hz component by using the ANC configuration. The LMS algorithm have been used for filter order $M = 10$. The effect of different adaptation gains is depicted.

In addition to the above, the filter order plays a significant role on the efficiency and correct noise suppression. Figure 40 shows that a small filter order ($M = 5$) does not manage to cancel the effect of the noise, while a big filter order ($M = 20$) is unstable and therefore all the information is lost. Additionally, $M = 15$ seems to have bad effects on the spectrogram, because one can notice that at 50 Hz the power has lower values than it should, while at 100 Hz harmonic components of the noise have been introduced. Therefore, $M = 10$ & $\mu = 0.1$ are the best parameters for the problem.

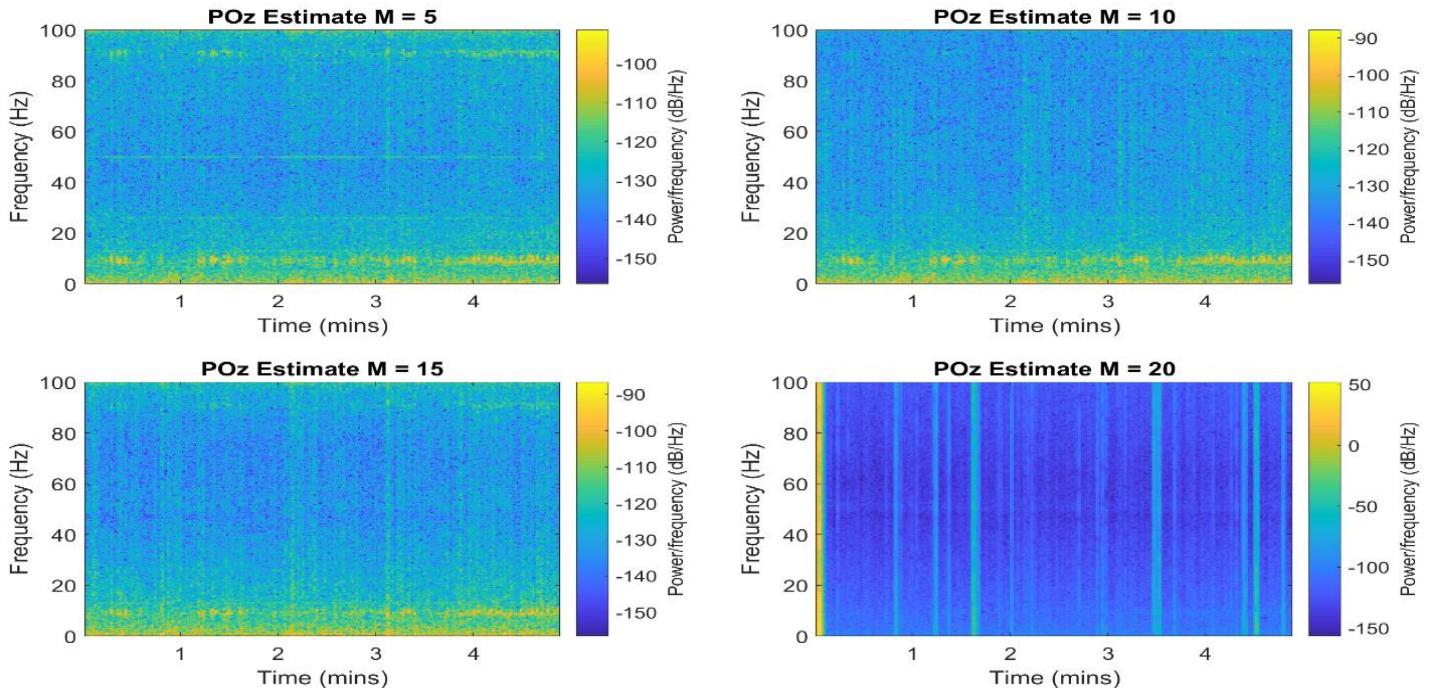


Figure 40: The POz Estimate after removing the 50 Hz component by using the ANC configuration. The LMS algorithm have been used for step size $\mu = 0.1$. The effect of different filter orders is depicted.

4. WIDELY LINEAR FILTERING & ADAPTIVE SPECTRUM ESTIMATION

4.1. Complex LMS & Widely Linear Modelling

- a) Complex-valued signals are found everywhere in communications, renewable energy, smart grid and array and biomedical signal processing since they are more convenient for many different reasons. In that question a first-order widely-linear-moving-average process WLMA(1), driven by circular white Gaussian noise is analyzed. The process is determined by the following equation:

$$y(n) = x(n) + b_1 x(n-1) + b_2 x^*(n-1),$$

where $b_1 = 1.5 + 1j$, $b_2 = 2.5 - 0.5j$ and $x \sim N(0,1)$. The Complex LMS (CLMS) and Augmented CLMS (ACLMS) are implemented, in order to identify the WLMA model, i.e. its' parameters. Since the circularity quotient is:

$$\rho = \frac{p}{c} = \frac{E\{|y|^2\}}{E\{y^2\}} = 0.8 + 0.3j$$

the circularity coefficient is non-zero and therefore the signal is noncircular. This means that the CLMS algorithm is not capable of identifying the system, and thus we are sure that the ACLMS will perform significantly better than the CLMS.

In figure 41 the ensemble average learning curves from 100 independent realizations (simulations) for the CLMS and ACLMS algorithms of a WLMA(1) process driven by circular white Gaussian noise is depicted. It is clear that the ACLMS algorithm performs significantly better than the CLMS, since the process is not circular. The learning rate used for the algorithm implementations is equal to 0.05. The steady state error (SSE) for the CLMS algorithm is equal to 6.59 dB, while for the ACLMS is -306 dB. Therefore, the importance of the widely linear modelling for noncircular complex data is clear.

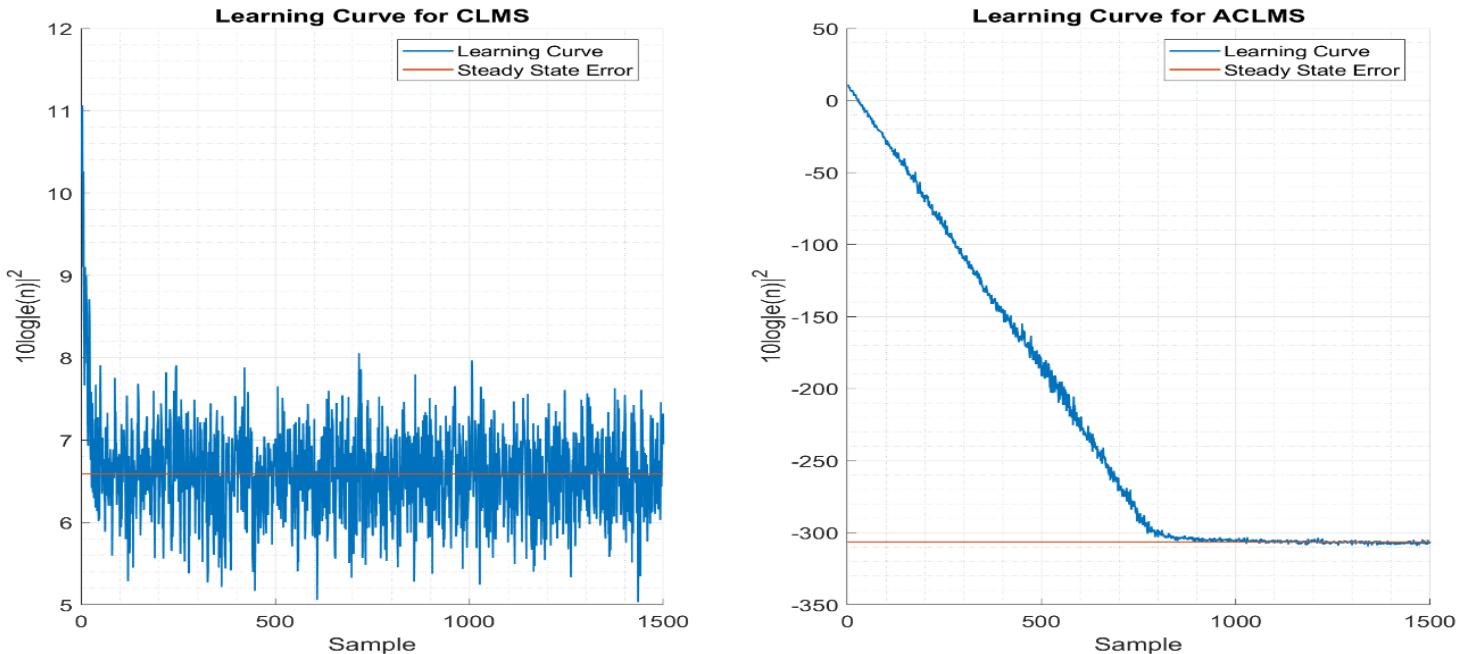


Figure 41: The learning curve for the CLMS and ACLMS algorithms for $\mu = 0.05$. The steady-state error for the CLMS algorithm is SSE1 = 6.59 dB, while for the ACLMS algorithm is SSE2 = -306 dB.

- b) As mentioned in the previous question, complex-valued signals are used because they are more convenient for a lot of reasons. As far as 2D wind speed is concerned, complex signals are used due to their convenience of representation. Anemometers work by measuring the wind speed in the East-West (EW) direction and the North-South (NS) separately and they represent them as one complex number, having the EW direction on the x-plane and the NS direction on the y-plane. The third dimension (height) can be introduced by using the z-plane. In figure 42 the complex valued wind speed for three wind regimes is presented. It is obvious that the high and low wind speeds are highly non-circular signals, while the medium wind speed is closer to be a circular signal.

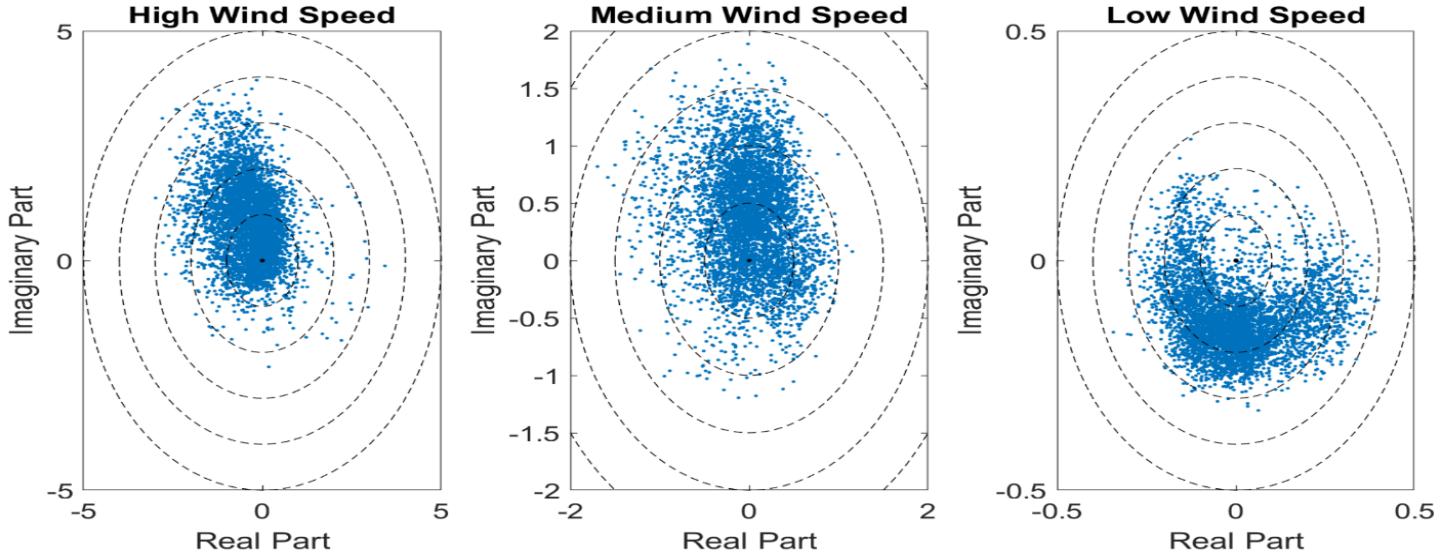


Figure 42: The complex-valued wind signal for the three wind regimes (high, medium, low) also known as circularity plot.

The one-step prediction error of the CLMS and ACLMS filter is shown in figure 43. Different adaptation gains have been used for the three wind regimes, since the range of values that the corresponding algorithms converge differs. This happens because the correlation matrices of the three regimes are different, because the wind signals are not the same, as presented in the figure above. For all three wind speeds the ACLMS algorithm performs better than the CLMS, managing to predict better the signals. The optimal filter length for the high wind speed is equal to 7, for the medium one 8 and for the low wind speed a filter length equal to 10 should be chosen, taking into consideration the prediction error and the complexity of the model.

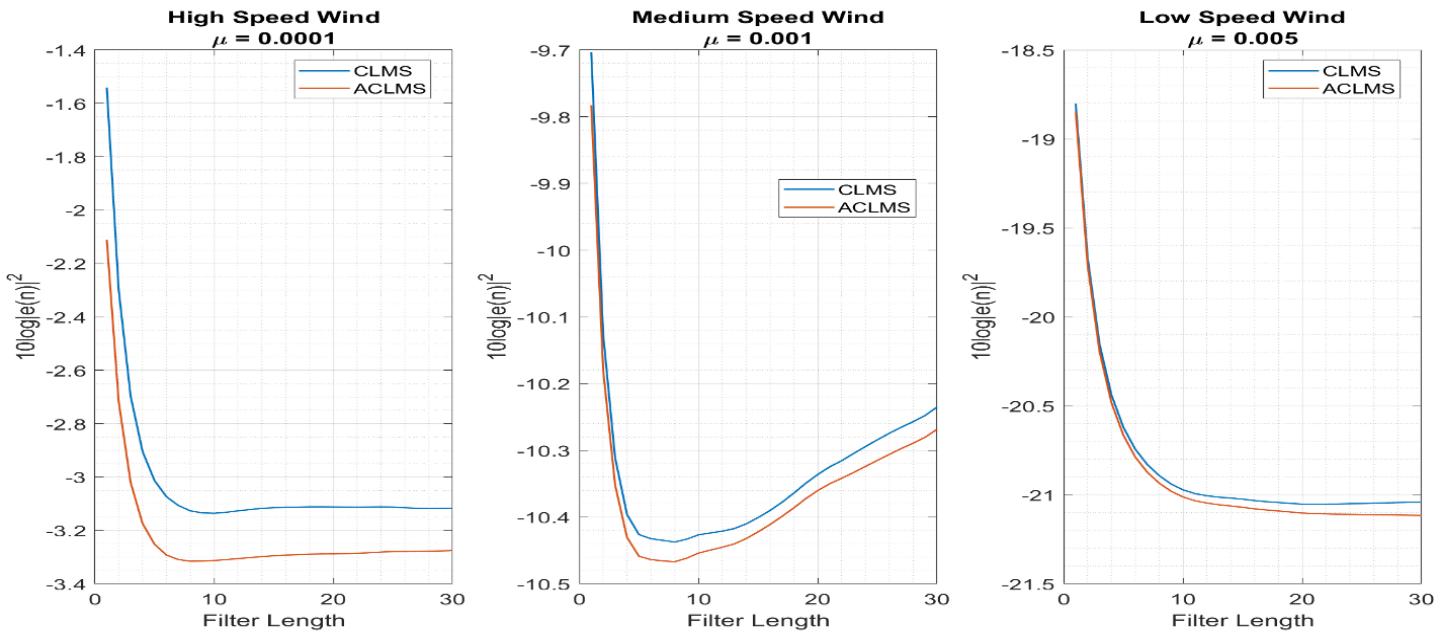


Figure 43: The one-step ahead prediction error of the complex wind data for different filter lengths.

- c) Power systems operate in three phases since they are more convenient, energy efficient and economically viable. However, these systems do not function perfectly sometimes, resulting in faults. By using the Clarke Transform (as introduced in the coursework assignment and the lectures) one can represent the whole system with one complex voltage, called the Clarke Voltage. This voltage can be represented in the complex plane by using its' real and imaginary part. In figure 44 the circularity diagrams of the Clarke voltages of three different three-phase systems are depicted. The left plot shows a balanced system, which we can easily realize that is circular. The other two plots present the C and D unbalanced systems (different phases and magnitudes), which show that the signals are not circular. Therefore, it is obvious that by using the circularity diagram and the circularity quotient and coefficient, faults in the system can be identified.

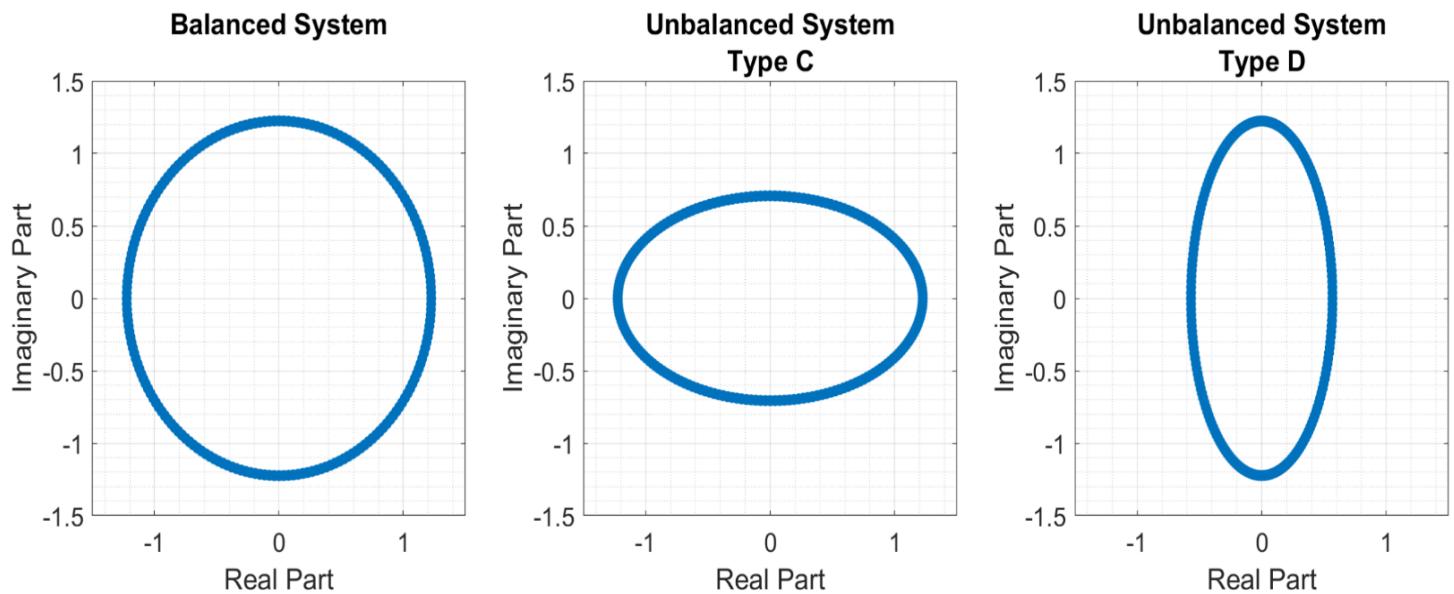


Figure 44: The circularity diagrams of three complex $\alpha - \beta$ voltages.

- d) The balanced complex $\alpha - \beta$ voltage is:

$$v(n) = \sqrt{\frac{3}{2}} V e^{j\left(\frac{2\pi f_0}{f_s} n + \phi\right)} \quad (4.1.1)$$

Using the strictly linear autoregressive model of order 1 we have:

$$v(n+1) = h^*(n)v(n) \Rightarrow \sqrt{\frac{3}{2}} V e^{j\left(\frac{2\pi f_0}{f_s}(n+1) + \phi\right)} = h^*(n) \sqrt{\frac{3}{2}} V e^{j\left(\frac{2\pi f_0}{f_s} n + \phi\right)} \Rightarrow e^{j2\pi\left(\frac{f_0}{f_s}\right)} = h^*(n) \quad (4.1.2)$$

Since the angle of complex number can be found by calculating the arc tangent of its' imaginary part over its' real part:

$$\begin{aligned} \frac{2\pi f_0(n)}{f_s} &= \arctan\left(\frac{\Im(h^*(n))}{\Re(h^*(n))}\right) \Rightarrow \\ f_0(n) &= \frac{f_s}{2\pi} \arctan\left(\frac{\Im(h^*(n))}{\Re(h^*(n))}\right) \end{aligned} \quad (4.1.3)$$

The complex-valued voltage that arises from the Clarke Transform for the unbalanced system is:

$$v(n) = A(n)e^{j\left(\frac{2\pi f_0}{f_s} n + \phi\right)} + B(n)e^{-j\left(\frac{2\pi f_0}{f_s} n + \phi\right)} \quad (4.1.4)$$

Using the widely linear autoregressive model of order 1 we have:

$$v(n+1) = h^*(n)v(n) + g^*(n)v^*(n) \quad (4.1.5)$$

Separating each part of the complex-valued voltage in (4.1.4), taking (4.1.5) and dividing with $e^{j(\frac{2\pi f_0 n}{f_s})}$ we have:

$$A(n+1)e^{j2\pi(\frac{f_0}{f_s})} = h^*(n)A(n) + g^*(n)B^*(n) \quad (4.1.6)$$

$$B(n+1)e^{-j2\pi(\frac{f_0}{f_s})} = h^*(n)B(n) + g^*(n)A^*(n) \quad (4.1.7)$$

Since $f_s \gg f_0$, we have that $A(n+1) \approx A(n)$ & $B(n+1) \approx B(n)$ and by subtracting the conjugate of (4.1.7) from (4.1.6):

$$h^*(n) - h(n) + g^*(n)\left(\frac{B^*(n)}{A(n)}\right) + g(n)\left(\frac{A(n)}{B^*(n)}\right) = 0 \Rightarrow g^*(n)x^2(n) + (h^*(n) - h(n))x(n) + g(n) = 0, \quad (4.1.8)$$

where $x(n) = \frac{B^*(n)}{A(n)}$. The solution of the quadratic equation in (4.1.8) is:

$$x_{1,2}(n) = \frac{\left(j\Im(h^*(n)) \pm j\sqrt{(\Im(h(n))^2 - |g(n)|^2)}\right)}{g^*(n)} \quad (4.1.9)$$

Substituting (4.1.9) to the modified (4.1.6) we have that:

$$e^{j2\pi(\frac{f_0}{f_s})} = h^*(n) + g^*(n)x_{1,2}(n) = h^*(n) + g^*(n) \frac{\left(j\Im(h^*(n)) \pm j\sqrt{(\Im(h(n))^2 - |g(n)|^2)}\right)}{g^*(n)} \Rightarrow$$

$$e^{j2\pi(\frac{f_0}{f_s})} = h^*(n) + j\Im(h^*(n)) \pm j\sqrt{(\Im(h(n))^2 - |g(n)|^2)} = \Re(h(n)) \pm j\sqrt{(\Im(h(n))^2 - |g(n)|^2)} \quad (4.1.10)$$

Since $f_s \gg f_0$ the imaginary part of the exponential in (4.1.10) is positive. Therefore, the second solution of (4.1.9) is not valid. By following the same procedure with (4.1.3) the frequency of the unbalanced voltage is:

$$f_0(n) = \frac{f_s}{2\pi} \arctan \left(\frac{\left(\sqrt{(\Im(h(n))^2 - |g(n)|^2)} \right)}{\Re(h(n))} \right) \quad (4.1.11)$$

- e) Frequency estimation in three phase power systems is crucial, since electrical energy plays a pivotal role in our everyday life. In question 4.1.c the representation of the Clarke complex voltage of a power system was discussed and the fact that faults in the system can be identified by the circularity diagram was shown. Also, figure 44 shows that a balanced system provides with a circular Clarke voltage, while unbalanced systems have non-circular voltages. As discussed in 4.1.a the CLMS algorithm is appropriate for the estimation of circular data, while it does not provide with high accuracy when the data are non-circular.

Therefore, in 4.1.d the frequency estimation of balanced and unbalanced systems was examined, and the results were that balanced systems can be estimated with the CLMS algorithm, while unbalanced systems should be estimated using the ACLMS algorithm, agreeing with what was previously stated.

Figure 45 presents the frequency estimation problem of a balanced and two unbalanced systems (Type C & Type D) using these two algorithms. In the left figure, a balanced system's frequency is estimated, and one can observe that the

CLMS algorithm converges instantaneously to the correct frequency of 50 Hz. Although, the ACLMS also provides with the correct estimation, it needs almost 40 samples to converge.

As far as the type C unbalanced system is concerned, it is obvious that the ACLMS algorithm gives the correct frequency estimate, while the CLMS algorithm oscillates about 45 Hz, verifying the ineffectiveness of strictly linear models in noncircular data.

Finally, for the type D unbalanced system, ACLMS provides with the best estimate, while the CLMS algorithm oscillates around an even lower frequency, causing a higher error in the frequency estimation. In addition to this, the high non-circularity attribute of the type D system leads to even slower convergence of the ACLMS algorithm.

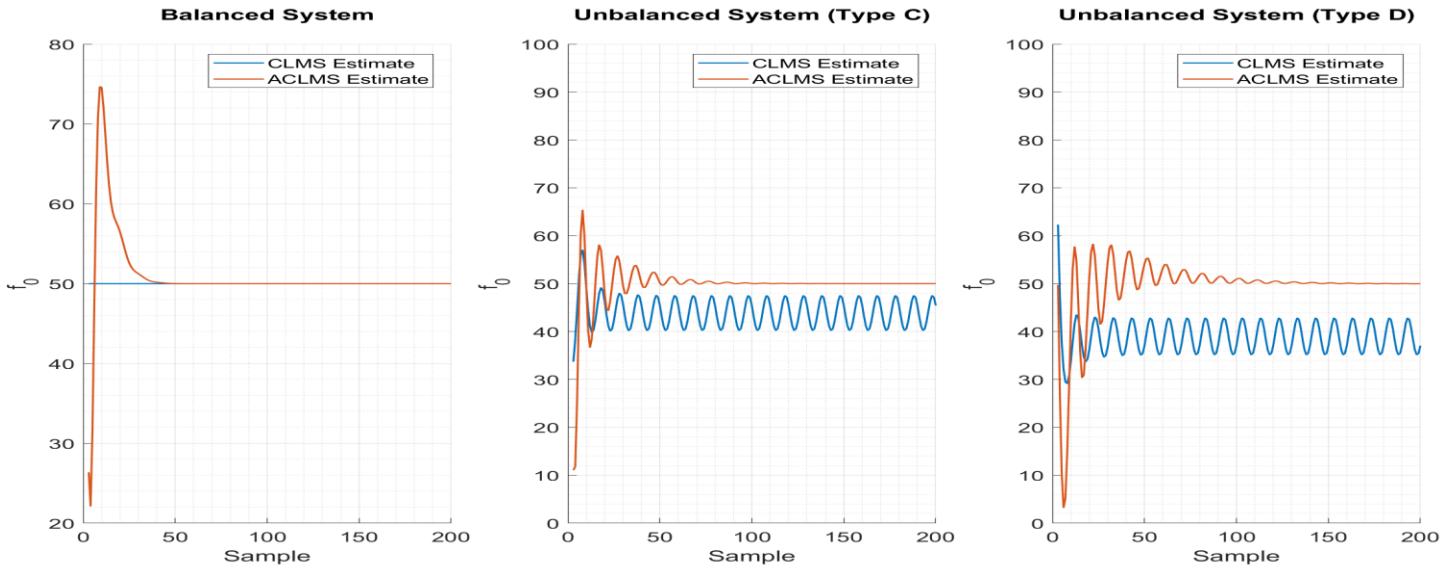


Figure 45: Frequency Estimation in Balanced & Unbalanced Three-Phase Power Systems using the CLMS and ACLMS algorithms.

4.2. Adaptive AR Model Based Time-Frequency Estimation

- a) For that question a frequency modulated (FM) signal with additive circular complex-valued white noise with zero mean and variance equal to 0.05 has been generated. That signal is non-stationary since it has a time-varying instantaneous frequency and therefore the block-based estimation of the AR coefficients is not an accurate solution. In figure 46 (left) the frequency of the signal is depicted, while in the right one the AR(1) PSD estimate is shown. It is obvious that this method cannot capture the changes in frequency. The peak of the PSD estimate is found at 184 Hz, which is the value of the mean of the instantaneous frequency of the FM signal. By using autoregressive processes of higher orders, a better estimator could be implemented, but this method should be avoided since many coefficients must be estimated to achieve high accuracy.

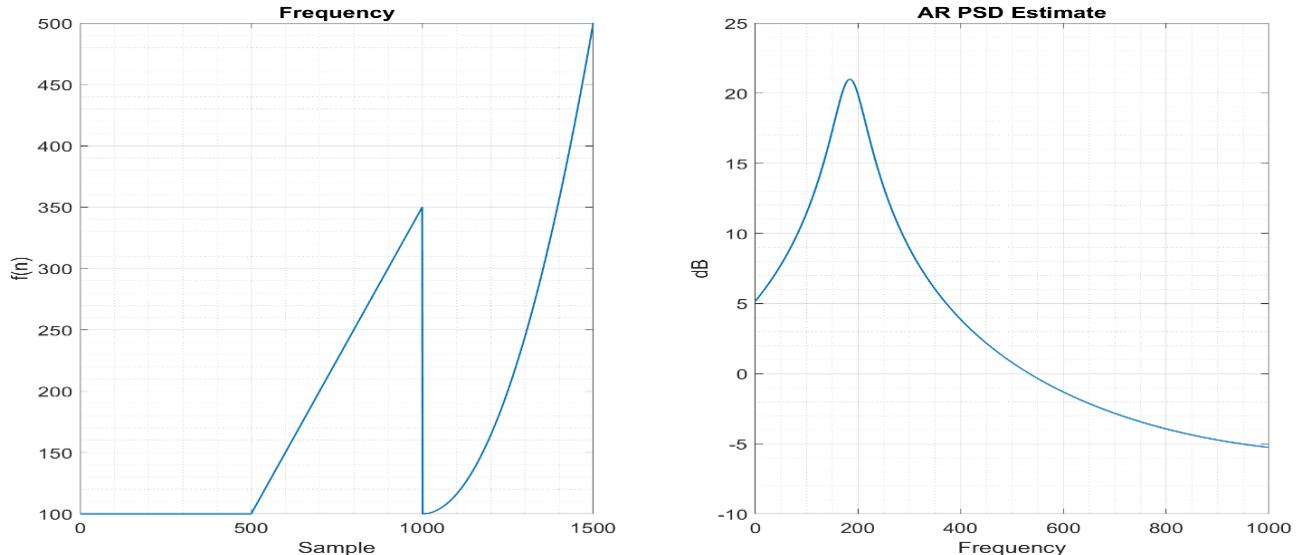


Figure 46: The frequency of the FM signal (left). The AR(1) Power Spectral Density estimate of the signal (right).

- b) In that question the CLMS algorithm is used in order to estimate the AR(1) coefficient at each time instant. Afterwards, the power spectrum of an AR process with that coefficient is estimated. When the iteration has finished the time-frequency spectrum is plotted, as depicted in figure 47. In that experiment three different learning rates have been used. When the learning rate is very small (fig. 47 left) the spectrum is falsely estimated, while when it is big (fig. 47 right) it seems to diverge. When $\mu = 0.05$ the CLMS algorithm manages to estimate correctly the spectrum, since it can capture the changes in frequency of the FM signal, shown in fig.46 (left).

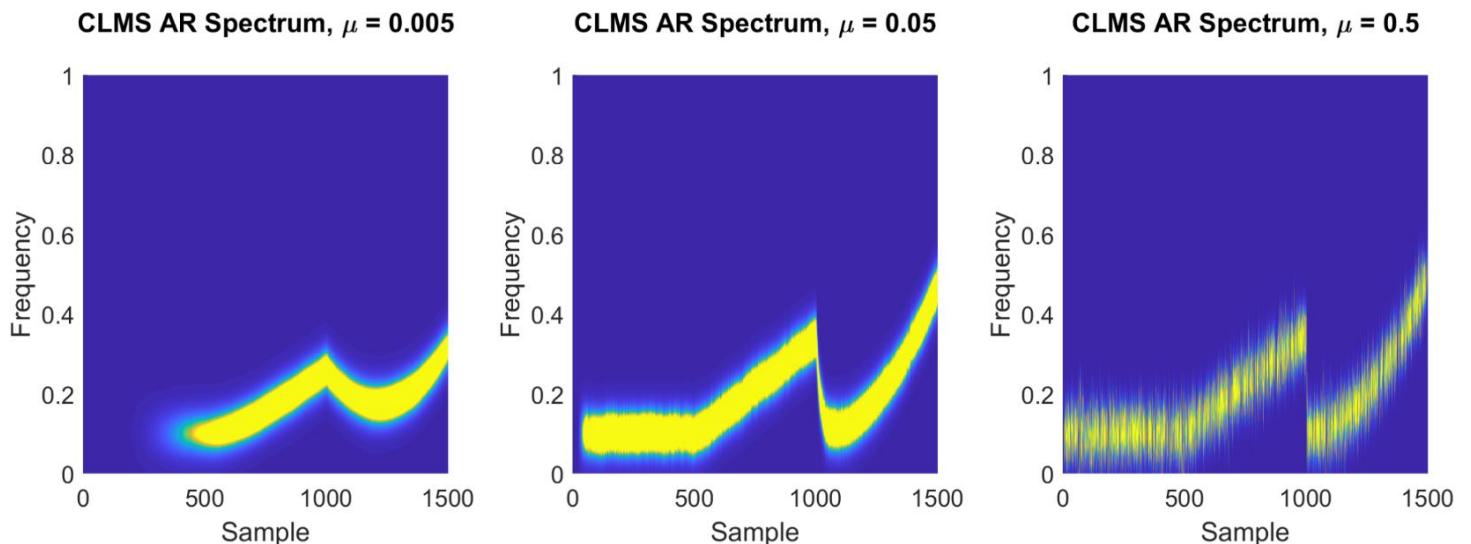


Figure 47: The CLMS based time-frequency spectrum estimate of the FM signal for different learning rates.

4.3. A Real Time Spectrum Analyser Using Least Mean Square

- a) The cost function of the least squares framework is the square of the norm of the error, i.e. $\|\mathbf{y} - \hat{\mathbf{y}}\|^2$. Expanding it we have:

$$J(\mathbf{w}) = \|\mathbf{y} - \hat{\mathbf{y}}\|^2 = \|\mathbf{y} - \mathbf{F}\mathbf{w}\|^2 = (\mathbf{y} - \mathbf{F}\mathbf{w})^H(\mathbf{y} - \mathbf{F}\mathbf{w}) = \mathbf{y}^H\mathbf{y} - \mathbf{y}^H\mathbf{F}\mathbf{w} - \mathbf{w}^H\mathbf{F}^H\mathbf{y} + \mathbf{w}^H\mathbf{F}^H\mathbf{F}\mathbf{w}. \quad (4.3.1)$$

In order to minimize with respect to \mathbf{w} , we differentiate (4.3.1). Thus:

$$\nabla_{\mathbf{w}} J(\mathbf{w}) = \mathbf{0} - \mathbf{F}^H\mathbf{y} - \mathbf{F}^H\mathbf{y} + 2\mathbf{F}^H\mathbf{F}\mathbf{w} = -2\mathbf{F}^H\mathbf{y} + 2\mathbf{F}^H\mathbf{F}\mathbf{w} \quad (4.3.2)$$

Taking the derivative of the cost function to be equal to zero, the least squares solution is:

$$\mathbf{w} = (\mathbf{F}^H\mathbf{F})^{-1}\mathbf{F}^H\mathbf{y} = \mathbf{F}^{-1}\mathbf{y}. \quad (4.3.3)$$

It is clear that the equation $\hat{\mathbf{y}} = \mathbf{F}\mathbf{w}$ is the IDFT of \mathbf{w} , where \mathbf{F} is the basis matrix and \mathbf{w} are the Fourier coefficients of \mathbf{y} , i.e. the DFT of \mathbf{y} . Therefore, the best linear estimator of the signal \mathbf{y} in the least squares sense is derived by using the DFT formula.

- b) As shown in the previous question, the least squares solution of estimating a signal $y(n)$ using a linear combination of N harmonically related sinusoids as

$$\hat{y}(n) = \sum_{k=0}^{N-1} w(k)e^{\frac{j2\pi kn}{N}}, \quad (4.3.4)$$

corresponds to the Inverse Discrete Fourier Transform (IDFT) of length N , i.e

$$y(n) = \sum_{k=0}^{N-1} Y(k)e^{\frac{j2\pi kn}{N}}. \quad (4.3.5)$$

Thus, DFT converts a finite signal in the time domain into another finite signal in the frequency domain of the same or different length (e.g. by using zero-padding) and then the signal can be reconstructed again by the coefficients of the frequency domain with or without loss of information.

The DFT formula makes use of the complex basis functions $e^{-\frac{j2\pi kn}{N}}$, where k corresponds to the frequency sample and n corresponds to the time sample. Each time sample of the sequence $y(n)$ is projected to each complex function in the frequency domain, constructing the Discrete Fourier Transform. Thus, the first two Fourier coefficients are:

$$Y(0) = y(0) + y(1) + y(2) + \cdots + y(N-1), \quad (4.3.6)$$

$$Y(1) = y(0) + y(1)e^{-\frac{j2\pi}{N}} + y(2)e^{-\frac{j4\pi}{N}} + \cdots + y(N-1)e^{-\frac{j2\pi(N-1)}{N}}. \quad (4.3.7)$$

Therefore, the signal transforms into the frequency domain by projecting each time sample to the orthonormal basis function of the frequency domain, and then it is able to return to the time domain by projecting the Fourier coefficients into the orthonormal basis functions of the time domain, i.e. $e^{\frac{j2\pi kn}{N}}$.

The DFT algorithm is extended to the DTFT (Discrete-Time Fourier Transform) when infinite samples are available, resulting in a continuous signal in the frequency domain, and to the CTFT (Continuous-Time Fourier Transform) when the signal is continuous in both time and frequency. The CTFT of a signal $y(t)$ is:

$$Y(j\Omega) = \int_{-\infty}^{\infty} y(t)e^{-j\Omega t}dt, \quad (4.3.8)$$

meaning that the time-continuous signal is projected into infinite continuous orthonormal basis functions in the frequency domain, in order to construct the Fourier coefficients.

- c) The magnitude of the weight vector of the DFT-CLMS algorithm for the FM signal from question 4.2.a. is given in the time-frequency diagram of figure 48 (left). The spectrum obtained does not resemble the true power spectrum found by using the adaptive AR-spectrum analyzer in question 4.2.b. This happens because in each iteration the estimate weight does not only adapt to the data, but also accumulates the value of the weight of the previous iteration. That fact can be realized by the update equation of the CLMS algorithm given below:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e^*(n) \mathbf{x}(n). \quad (4.3.9)$$

Therefore, a forgetting factor should be introduced in order to suppress that effect. The leaky LMS algorithm, introduced in 3.1.e can solve that issue. Thus, the leaky CLMS update equation is shown below:

$$\mathbf{w}(n+1) = (1 - \mu\gamma)\mathbf{w}(n) + \mu e^*(n) \mathbf{x}(n). \quad (4.3.10)$$

In figure 48 (middle and right) the leaky DFT-CLMS algorithm manages to resemble the true power spectrum of the signal. It is also important to state that the value of the leakage coefficient plays a significant role, because if it is very small it will not achieve to resemble the power spectrum correctly and the result will be almost the same with the one provided by the DFT-CLMS algorithm, since the forgetting factor will be rather insufficient.

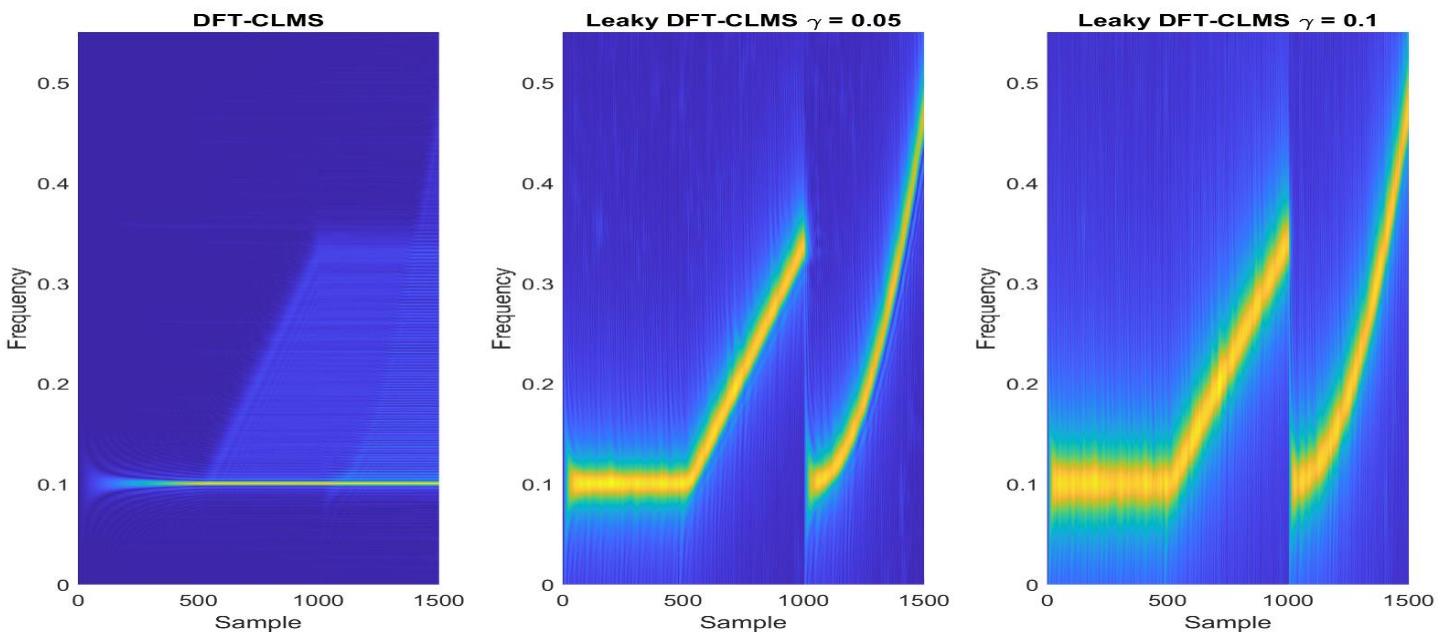


Figure 48: The DFT-CLMS and the Leaky DFT-CLMS algorithms for the estimation of the frequency of the FM signal from question 4.2.a.

- d) The EEG POz signal processed in 1.4.b and 3.3.d is also analyzed by using the DFT-CLMS algorithm. In figure 49 the PSD estimate of the signal is depicted, for different number of DFT points. It is obvious that bigger number of DFT points leads to better estimation of the time-frequency spectrum of the EEG signal. For $N = 2400$ a spectral leakage can be identified, since there are a lot of frequencies that are not observed in the other two spectrograms. Therefore, N needs to be big enough in order to distinguish the different frequencies of the EEG signal and realize the cause of each frequency peak. In figure 49 (right) the following signals can be observed:

- i. Alpha-rhythm: 8 – 10 Hz
- ii. Steady state visual evoked potential (SSVEP): 13 Hz (fundamental frequency)
- iii. Steady state visual evoked potential (SSVEP): 26 Hz (first harmonic)
- iv. Power-Line interference: 50 Hz

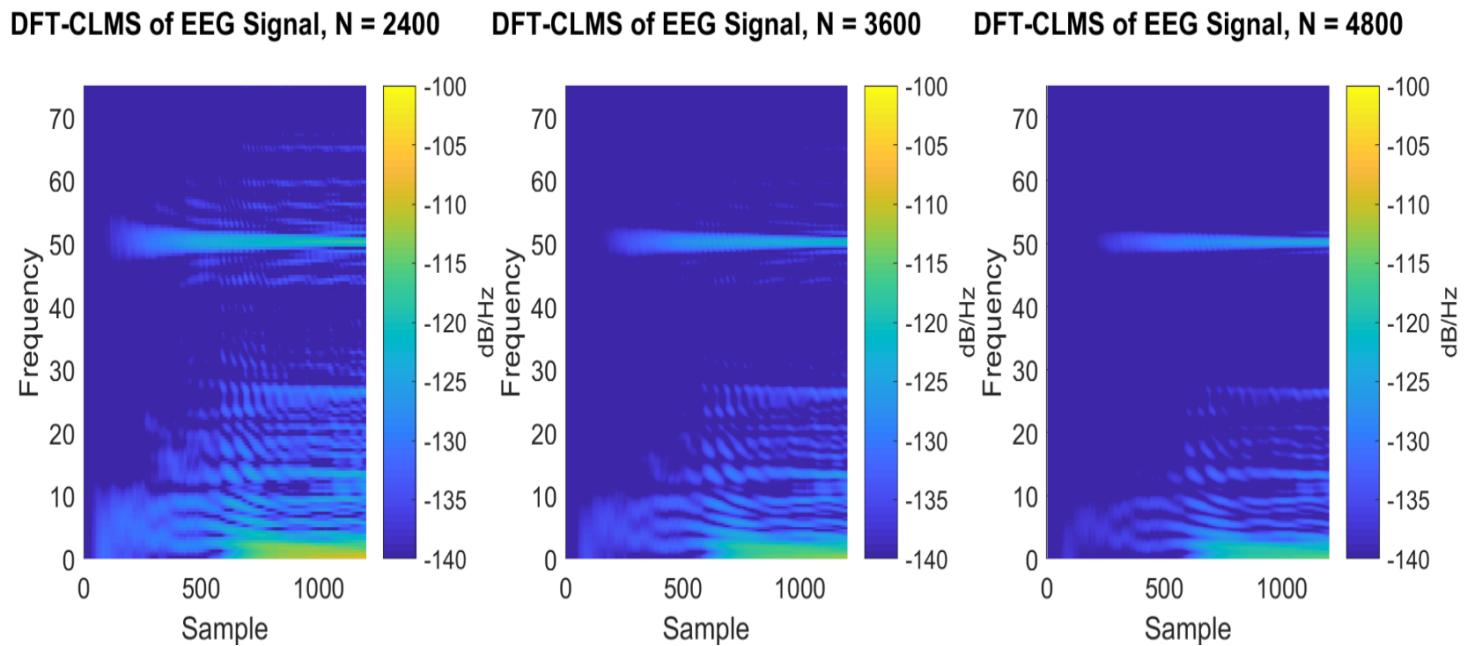


Figure 49: The PSD estimate of the EEG signal using the DFT-CLMS for different number of DFT points.