



Adaptive Signal Processing and Machine Intelligence

Coursework

Angelos Filos

March 23, 2018

Supervisor: Prof. Danilo Mandic

Department of Electrical and Electronic Engineering, Imperial College London

Contents

1	Spectrum Estimation	1
1.1	Discrete Fourier Transform Basics	1
1.2	Properties of Power Spectral Density (PSD)	2
1.3	Resolution and Leakage of Periodogram-based Methods	4
1.4	Periodogram-based Methods Applied to Real-World Data	8
2	Parametric and Line Spectra	10
2.1	Correlation Estimation	10
2.2	Spectrum of Autoregressive Processes	13
2.3	Principal Component Analysis	14
2.4	Real World Signals: Respiratory Sinus Arrhythmia from RR-Intervals	16
3	Adaptive Signal Processing	19
3.1	The Least Mean Square (LMS) Algorithm	19
3.2	Adaptive Step Sizes	22
3.3	Adaptive Noise Cancellation	24
4	Widely Linear Filtering and Adaptive Spectrum Estimation	29
4.1	Complex LMS and Widely Linear Modelling	29
4.2	Adaptive AR Model Based Time-Frequency Estimation	34
4.3	A Real Time Spectrum Analyser Using Least Mean Square	35

Chapter 1

Spectrum Estimation

1.1 Discrete Fourier Transform Basics

- a) The Fourier Transform (FT) of a sine wave at $f_0 = 20\text{Hz}$ is discrete, since the time series is periodic, comprising of Dirac δ functions at $\pm f_0$, provided at figure 1.1. Since the windowing (multiplication) operation in time domain is equivalent to convolution in the frequency domain, the Discrete Time Fourier Transform (DTFT) of a windowed sine wave at 20Hz is derived by convolution of the ideal FT of the sine wave, Dirac δ functions, with the FT of the rectangular window, *sinc* function, illustrated at figure 1.1 for different window lengths τ . The spectrum leakage is demonstrated, as well as the trade-off between side-lobes peak and window length τ (inverse relationship).

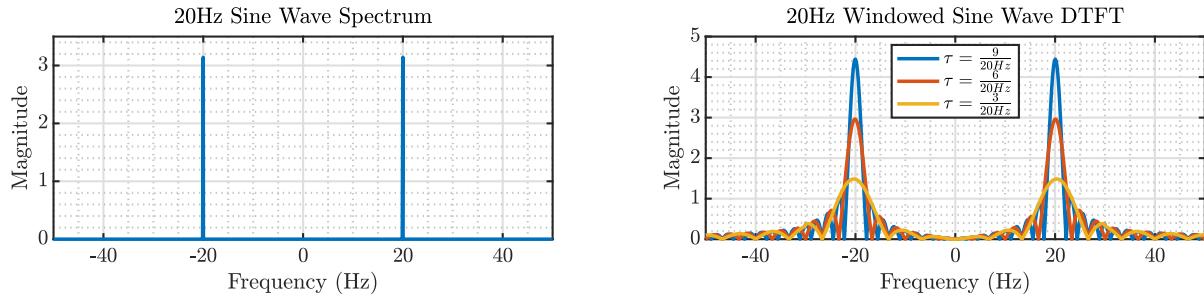


Figure 1.1: Ideal magnitude spectrum (left) and theoretical DTFT (right) of 20Hz sine wave.

- b) The Discrete Fourier Transform (DFT) spectrum of a finite length sine wave at $f_0 = 20\text{Hz}$ is considered. The signal can be treated as a windowed and sampled version of the continuous, infinite length sine wave. Consequently, its DFT is expected to be a sampled version of the DTFT shown in figure 1.1, at frequency intervals defined by the frequency resolution $\Delta f = \frac{F_s}{K}$, where F_s and K the sampling frequency and the sequence length, respectively. In case of $K = 100$, we obtain $F_s = 1000\text{Hz}$ and $\Delta f = 10\text{Hz}$, leading to elimination of spectrum leakage or **coherent sampling**, since at the integer multiples of the frequency resolution 10Hz , the *sinc* function is zero, excluding the peaks at $\pm f_0$. We highlight that this is not the same with the ideal FT spectrum, but just the result of frequency quantisation. If f_0 or F_s are slightly perturbed then the leakage will be inevitable. When zero-padding is performed and $K = 1000$, then $\Delta f = 1\text{Hz}$ and the impact of the spectrum leakage is evident. Figure 1.2 illustrated the DFT spectra of those experiments.

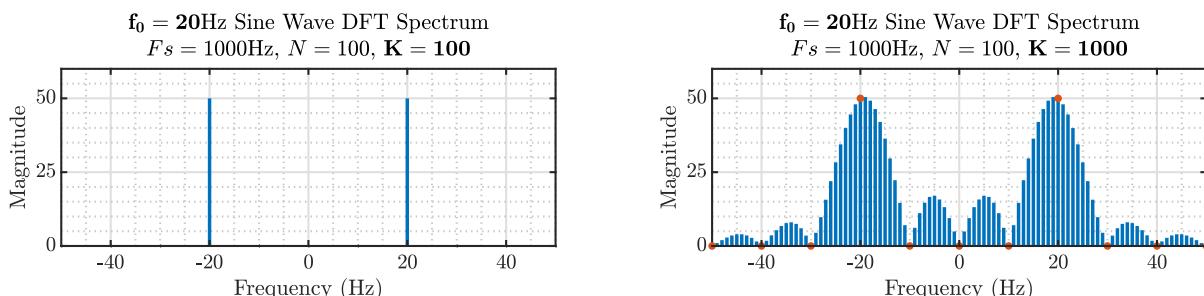


Figure 1.2: 20Hz sampled sine wave DFT spectra and coherent sampling.

- c) The experiment is repeated for a finite length sine wave at $f_0 = 24\text{Hz}$ and its spectra are provided at 1.3. As expected this time for both $K = 100$ and $K = 1000$ spectral leakage is unavoidable. However, especially in case of $K = 100$ when $\Delta f = 10\text{Hz}$, we note that the peaks at $\pm f_0 = \pm 24\text{Hz}$ are undetected and therefore the sequence is incoherently sampled. Increasing the zero-padding samples, such that the frequency resolution Δf is a factor of f_0 resolves this issue, therefore nothing can be done for the spectral leakage effect. Candidate values for K then are: $125, \frac{1000}{6}, 250, 500, 1000, \dots$.

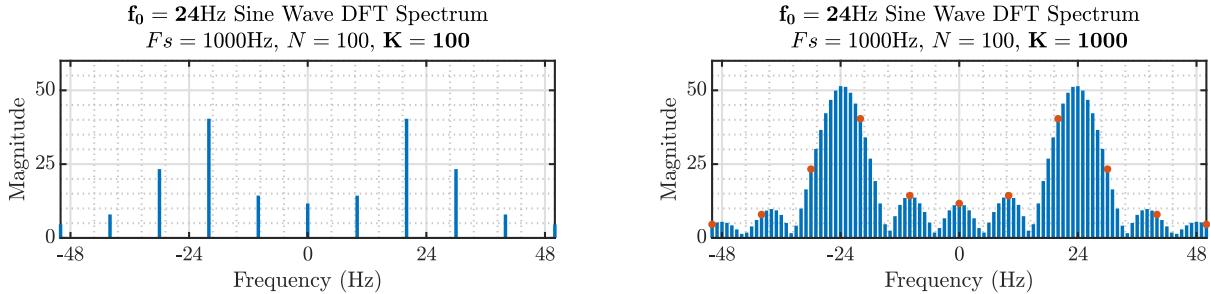


Figure 1.3: 24Hz sampled sine wave DFT spectra and incoherent sampling.

1.2 Properties of Power Spectral Density (PSD)

Approximation in the definition of PSD

Equivalence is shown by starting from equation (9), as indicated in the coursework instructions.

$$P(w) = \lim_{N \rightarrow \infty} \mathbb{E} \left\{ \frac{1}{N} \left| \sum_{n=0}^{N-1} x(n) e^{-jwn} \right|^2 \right\} \quad (1.1)$$

$$\begin{aligned} &= \lim_{N \rightarrow \infty} \mathbb{E} \left\{ \frac{1}{N} \sum_{m=0}^{N-1} x(m) e^{-jwm} \sum_{k=0}^{N-1} x^*(k) e^{jwk} \right\} \\ &= \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{m=0}^{N-1} \sum_{k=0}^{N-1} \mathbb{E} \left\{ x(m) e^{-jwm} x^*(k) e^{jwk} \right\} \\ &= \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{m=0}^{N-1} \sum_{k=0}^{N-1} \mathbb{E} \left\{ x(m) x^*(k) \right\} e^{-jw(m-k)} \\ &= \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{m=0}^{N-1} \sum_{k=0}^{N-1} r_{xx}(m-k) e^{-jw(m-k)} = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{m=0}^{N-1} \sum_{k=0}^{N-1} g(m-k) \end{aligned} \quad (1.2)$$

where $g(\tau) = r_{xx}(\tau) e^{-jw\tau}$. Converting the double into a single summation according to:

$$\sum_{m=-N}^N \sum_{k=-N}^N g(m-k) = \sum_{\tau=-2N}^{2N} (2N+1-|\tau|)g(\tau) \quad (1.3)$$

(1.2) can be written as:

$$\begin{aligned} P(w) &= \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{\tau=-(N-1)}^{N-1} (N-|\tau|)r_{xx}(\tau) e^{-jw\tau} \\ &= \lim_{N \rightarrow \infty} \sum_{\tau=-(N-1)}^{N-1} r_{xx}(\tau) e^{-jw\tau} - \lim_{N \rightarrow \infty} \frac{1}{N} |\tau| \sum_{\tau=-(N-1)}^{N-1} r_{xx}(\tau) e^{-jw\tau} \\ &\approx \sum_{\tau=-\infty}^{\infty} r_{xx}(\tau) e^{-jw\tau} \end{aligned} \quad (1.4)$$

where the last equality in (1.4) is obtained under the mild assumption that the covariance sequence $r_{xx}(k)$ decays rapidly, or equivalently:

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{\tau=-(N-1)}^{N-1} |\tau| |r_{xx}(\tau)| = 0 \quad (1.5)$$

Thus under the mild assumption that the covariance sequence decays rapidly we have shown that (1.1) is equal to (1.4), which is the DTFT of the autocovariance function (ACF).

$$P(w) = \sum_{t=-\infty}^{\infty} r_{xx}(t) e^{-jw t} \quad (1.6)$$

- a) Power Spectrum Estimation is performed by taking the Fourier Transform of the ACF, $r_{xx}(k)$. Due to the even symmetry of ACF, its Fourier Transform, $P(w)$, is expected to be purely real, as well as non-negative since the positive semi-definiteness of the covariance matrix, xx^T . These properties are preserved after zero-padding in the middle of the ACF, around zero, while at the same time, frequency resolution is increased. Figure 1.4 verifies our claims in the case of the rectangular window signal, whose width is controlled by varying M . Unsurprisingly, the increase of M from 10 to 128 results in an increase of the rectangular width and and inversely proportional decrease in the width of the mainlobe (frequency domain), while the power concentration is increased around 0.

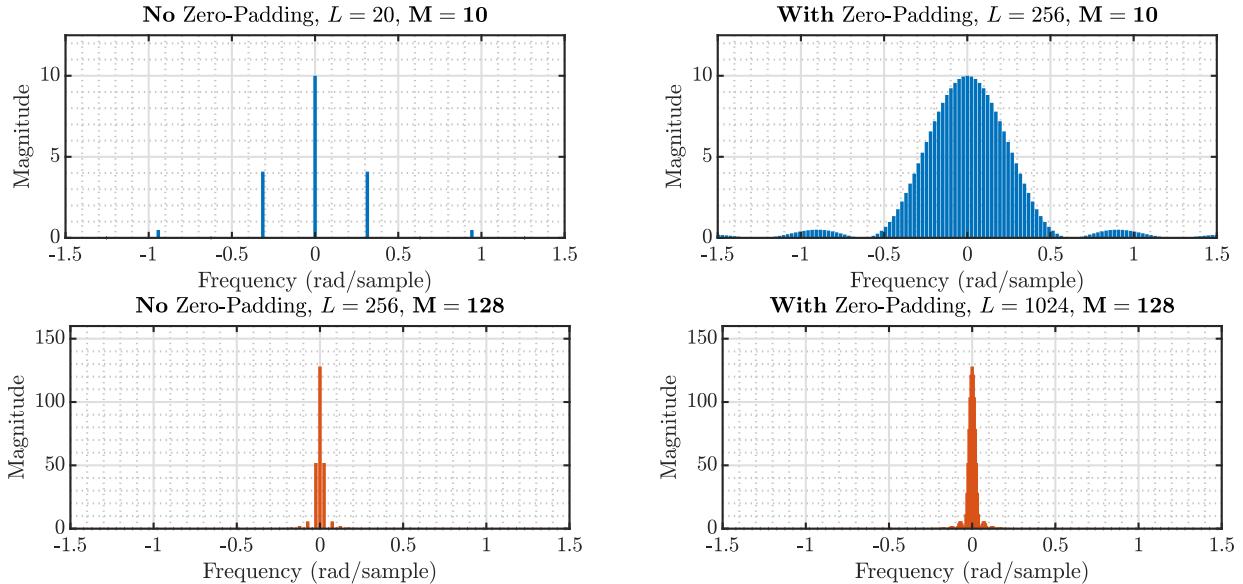


Figure 1.4: Spectrum estimation using symmetric ACF and zero-padding effect.

- b) Because of the symmetry of ACF, $P(w)$ is expected to be purely real. Nonetheless, the numerical methods used for the implementation of `fft` and the finite precision using in computers (64-bits in this case) lead to round-off errors, introducing a negligible imaginary part `imag(xf)`. Figure 1.5 demonstrates this point, verifying how trivial (16 orders of magnitude smaller) the imaginary part is, compared to the real part, `real(xf)`, while figure 1.6 depicts the PSD estimates with and without the imaginary part.

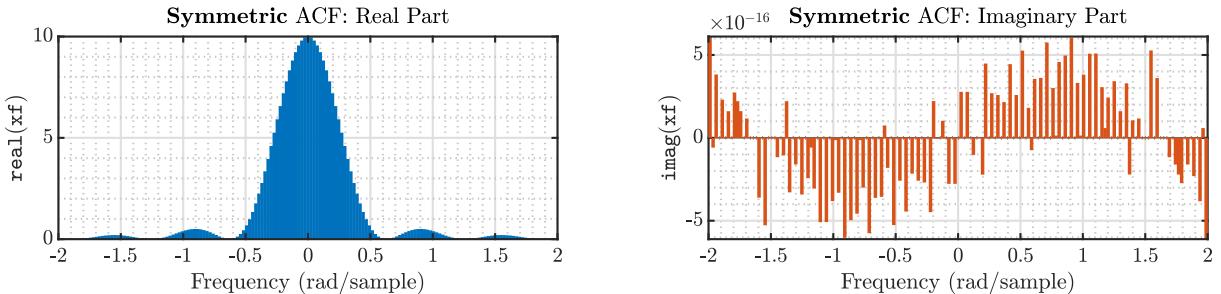
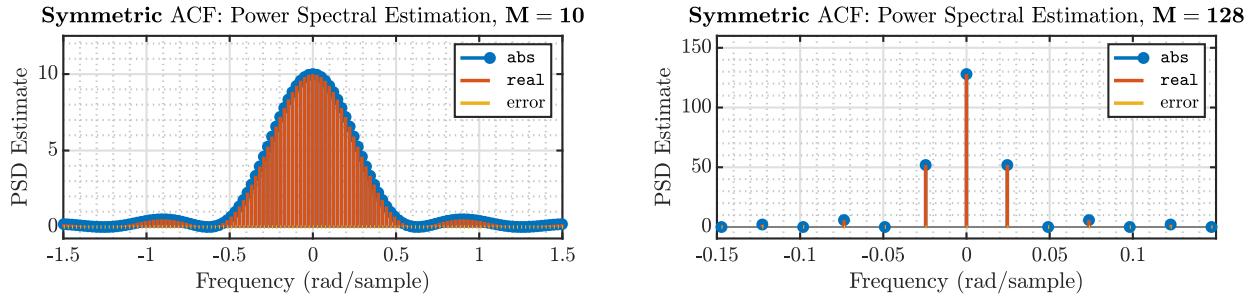
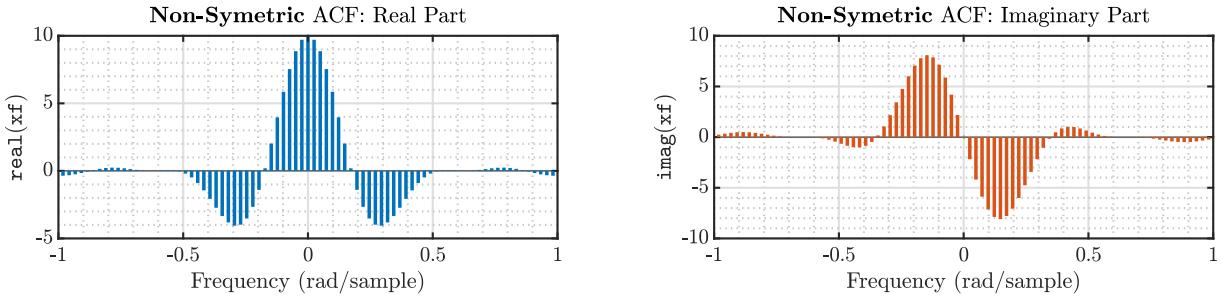
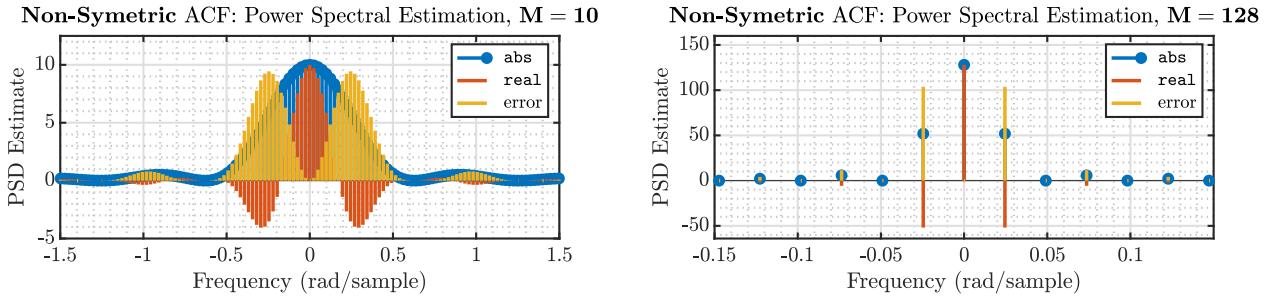


Figure 1.5: Symmetric $r_{xx}(k)$: real and imaginary parts of $P(w)$ and round-off errors.


 Figure 1.6: Symmetric $r_{xx}(k)$: $P(w)$ estimate using abs and real.

- c) On the other hand, in case of non-symmetric ACF, $r_{zz}(k)$, the imaginary part is of the same order with the real part, as illustrated in figure 1.7 and the PSD estimates at figure 1.8 have significant deviation. Moreover, it is also important to highlight that when **real** is used instead of the **abs**, the Fourier Transform, $P_z(w)$, is not guaranteed to be positive, resulting in a meaningless PSD estimate. Consequently, the magnitude (**abs**) should be used in case of non-symmetric ACF.


 Figure 1.7: Non-Symmetric $r_{xx}(k)$: real and imaginary parts of $P(w)$.

 Figure 1.8: Non-Symmetric $r_{xx}(k)$: $P(w)$ estimate using abs and real.

- d) The figures in all the parts are generated, having previously centered the frequency axis via `fftshift`. Moreover, the signal length is important when the frequency index, w , or the time index, n , are generated. Table 1.1 summarized the MATLAB commands used for any case.

Vector Length, L	Frequency Index, w	Time Index, n
Even	$-\pi : \frac{2\pi}{L} : \pi - \frac{2\pi}{L}$	$-\frac{L}{2} : 1 : \frac{L}{2} - 1$
Odd	$-\pi + \frac{\pi}{L} : \frac{2\pi}{L-1} : \pi - \frac{\pi}{L}$	$-\frac{L-1}{2} : 1 : \frac{L-1}{2} - 1$

 Table 1.1: MATLAB commands for frequency w and time n indexes.

1.3 Resolution and Leakage of Periodogram-based Methods

- a) We investigate the magnitude spectrum of the N -points Bartlett window, $W_B(w)$, for several values of N , illustrated at figure 1.9. We note that its $3dB$ width of the main lobe varies as a function of N . Their inverse relationship is empirically shown in figure 1.10. Lastly, the peaks of the side lobes as a function of N are depicted in figure 1.11. Interestingly, the side lobes peaks are almost unaffected by N ($1.5dB$

change from $N = 8$ to $N = 1024$), unlike the $3dB$ width of the main lobe, which shrinks considerably for increasing values of N .

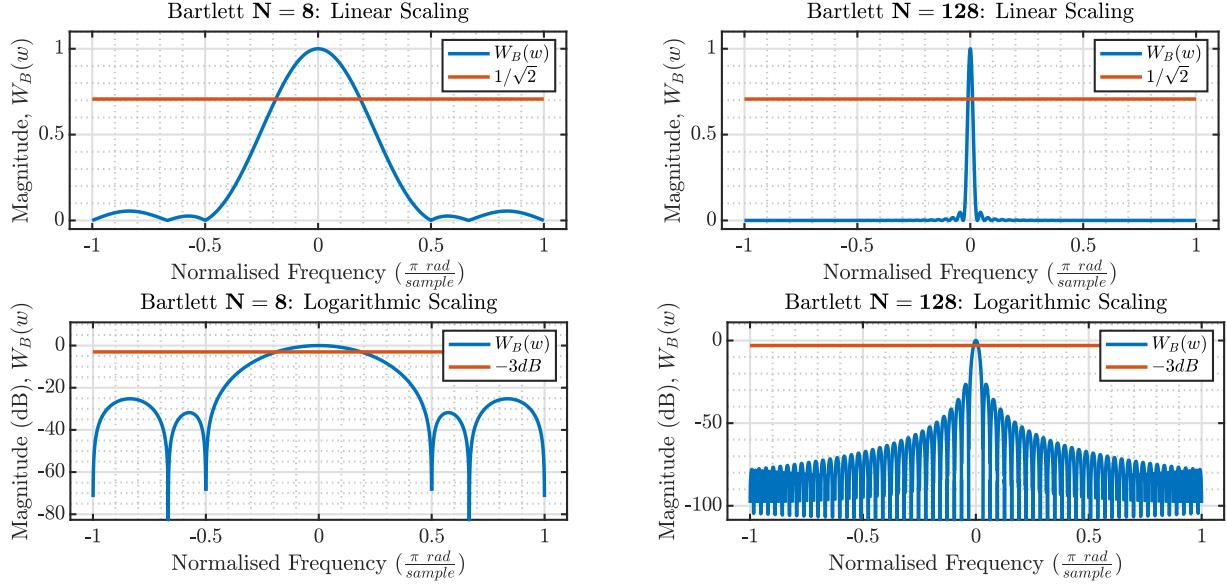


Figure 1.9: N -points magnitude spectrum of Bartlett window, $W_B(w)$.

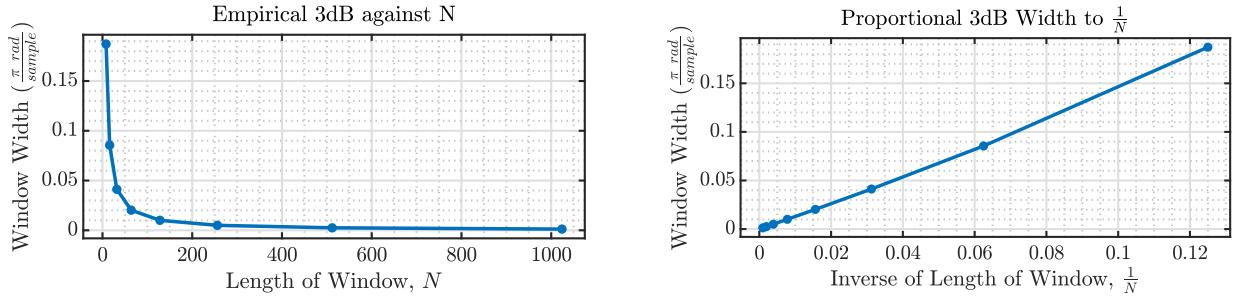


Figure 1.10: Bartlett window $3dB$ width of the main lobe as a function of N and $1/N$.

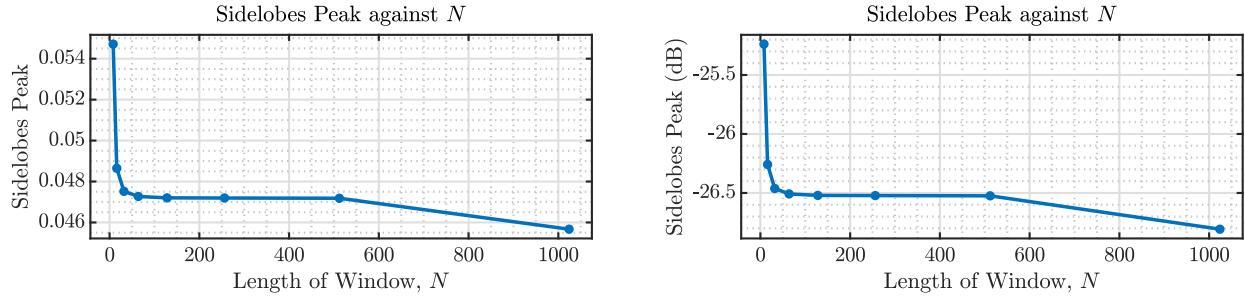


Figure 1.11: Bartlett window peaks of the side lobes as a function of N .

b) Let the signal

$$x(n) = \sin(2\pi f_0 n) + \sin(2\pi(f_0 + \frac{\alpha}{N})n) \quad (1.7)$$

where α is a varying parameter and $N = 256$ the fixed signal length. The PSD of $x(n)$ is ideally expected to have two peaks at frequencies f_0 and $f_0 + \frac{\alpha}{N}$ (in Hz), therefore if the frequency resolution Δf is not sufficiently small, the two peaks are not distinguishable, thus $\Delta f \leq \frac{\alpha}{N}$ is required. Moreover, due to the spectral leakage, the side lobes height make differentiation even harder, thus we will find the minimum α value by experiments. Figure 1.12 shows the empirical determination of α using a **rectangular window periodogram**. We note that for $\alpha \lesssim 0.62$ the two peaks are indistinguishable, while for greater α values the peaks can be discriminated. Lastly, figure 1.13 illustrates some examples for different α values.

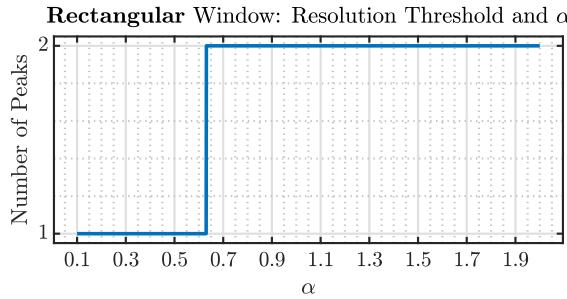


Figure 1.12: Rectangular Window: Number of Peaks in Periodogram for varying α .

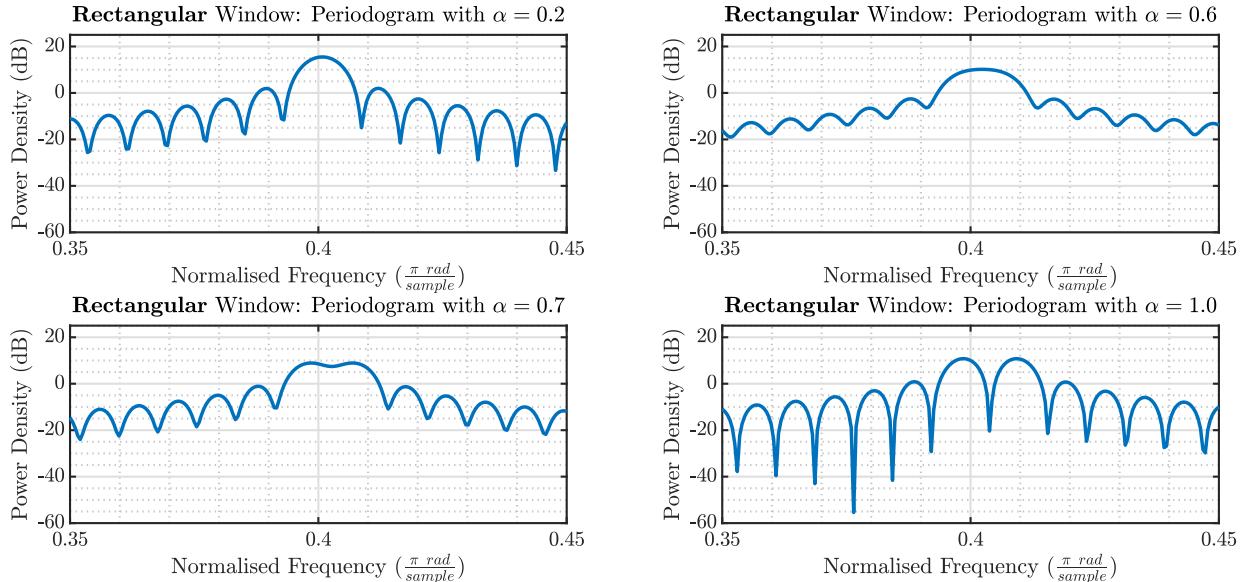


Figure 1.13: Rectangular window: Periodograms of $x(n)$ for varying α .

- c) Repeating the experiment using the **Hamming-windowed periodogram method**, we obtain figures 1.14 and 1.15. Note that since the Hamming window has a wider main lobe than the rectangular window, a larger value of α is required, $\alpha \gtrsim 0.71$, in order to distinguish the two peaks. Nonetheless, the Hamming window has better attenuation of the side lobes, depicted in figure 1.15.

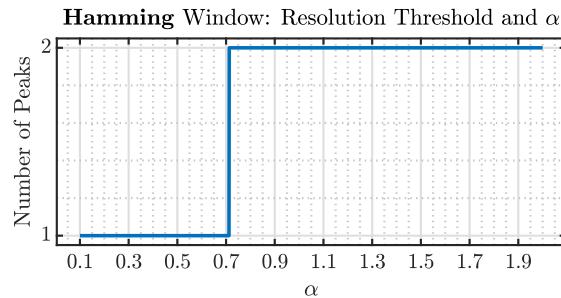
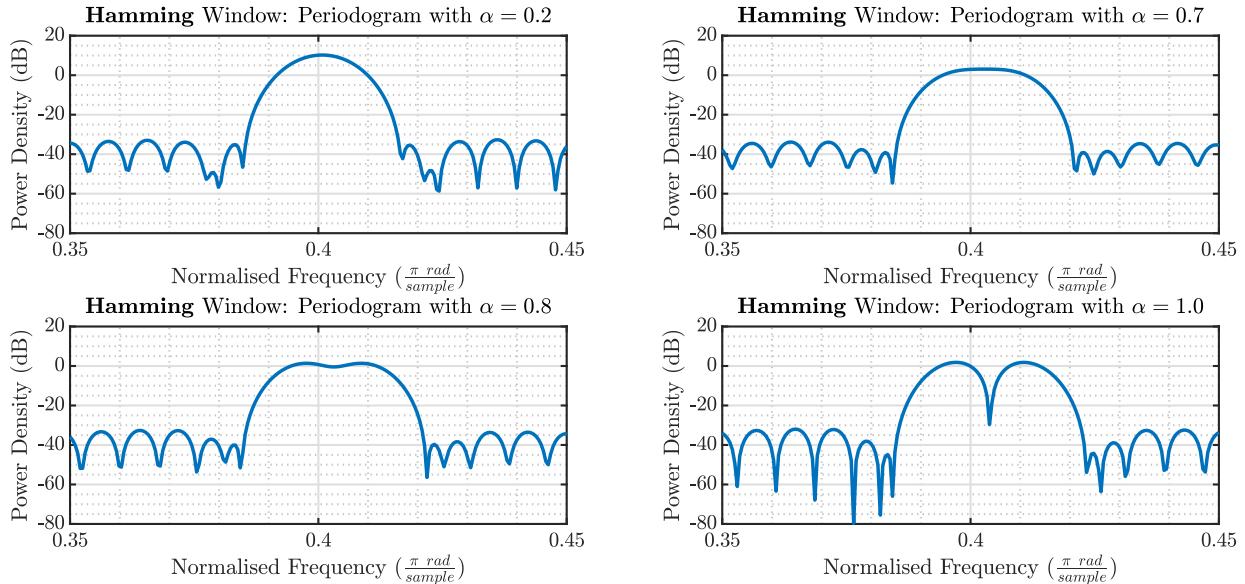


Figure 1.14: Hamming Window: number of peaks in periodogram for varying α .

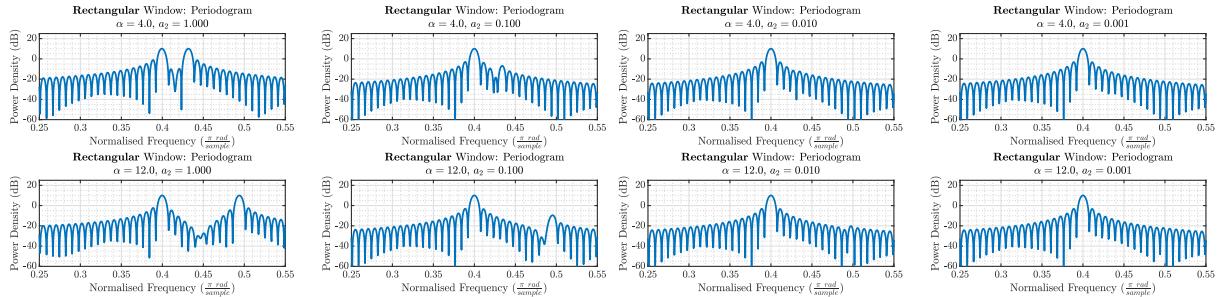
- d) Let the signal

$$x(n) = \sin(2\pi f_0 n) + a_2 \sin\left(2\pi\left(f_0 + \frac{\alpha}{N}\right)n\right) \quad (1.8)$$

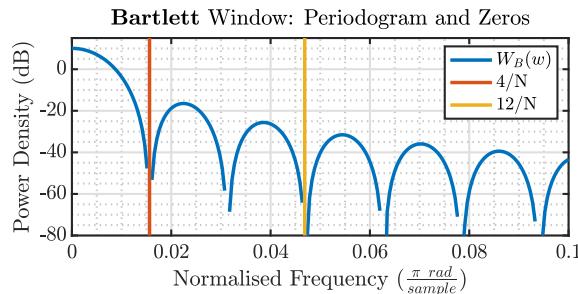
where $a_2 \in \{1, 0.1, 0.01, 0.001\}$ and $\alpha \in \{4, 12\}$ are varying parameters and $N = 256$ the fixed signal length. Ideally, for an infinitely long $x(n)$, the PSD is expected to comprise of two dirac deltas, however, since a finitely long sequence is used, spectral leakage is expected to degrade the quality of the periodogram. Different window functions trade $3dB$ width of the main lobe and the relative heights of the side lobes. The rectangular window used, has the smallest main lobe, however it also has the highest side


 Figure 1.15: Hamming Window: periodograms of $x(n)$ for varying α .

lobes, resulting in the greatest spectral leakage. Consequently, the amplitude of the second sinusoid, a_2 , affects significantly our ability to identify the second sinusoidal term in the spectral estimate. Figure 1.16 illustrates the periodograms for the different values of a_2 and α . We observe that for higher a_2 values, 1.0 and 0.1, the second peak can be identified for both α values. When $\alpha = 12$, the peak can also be observed for $a_2 = 0.01$, however it is rather unclear. Hence, even when the peaks are moved further apart (increased α) the spectral leakage deteriorates peak identification considerably.


 Figure 1.16: Rectangular window spectral leakage: periodograms of $x(n)$ for varying a_2 and α .

- e) In figure 1.17 the amplitude of the Fourier Transform of the Bartlett window is provided. We note that at frequencies $k\frac{2}{N}, k \in \mathbb{Z}$ there are zeros and thus at frequencies $\frac{4}{N}$ and $\frac{12}{N}$, too. Additionally, we observe that the side lobes of the window are not constant and they decrease as distance from the main lobe increases (increasing α). As a result, the amplitude threshold identification is slightly easier at $\alpha = 12$ because the peak of side lobes around $f = \frac{12}{N}$ is slightly lower than the peak of side lobes around $f = \frac{4}{N}$.


 Figure 1.17: Bartlett window: periodogram and zeros at $f = \frac{12}{N}$ and $f = \frac{4}{N}$.

- f) Repeating the experiment using the **Chebyshev-windowed periodogram method**, we obtain figure 1.18. Thanks to the Chebyshev window side lobes high attenuation, we notice that the two frequency com-

ponents are distinguishable even in the case of the smallest $a_2 = 0.001$. However, given the trade-off between side lobes attenuation and main lobe bandwidth, for $\alpha = 4$, the two peaks overlap, making discrimination more difficult. Overall, this represents the tradeoff that windows have to make between the width of the mainlobe as the height of the sidelobes. The rectangular window has a small mainlobe and thus the trouble in identification comes about because of the leakage effects whereas the Chebyshev window cause more smearing and less leakage.

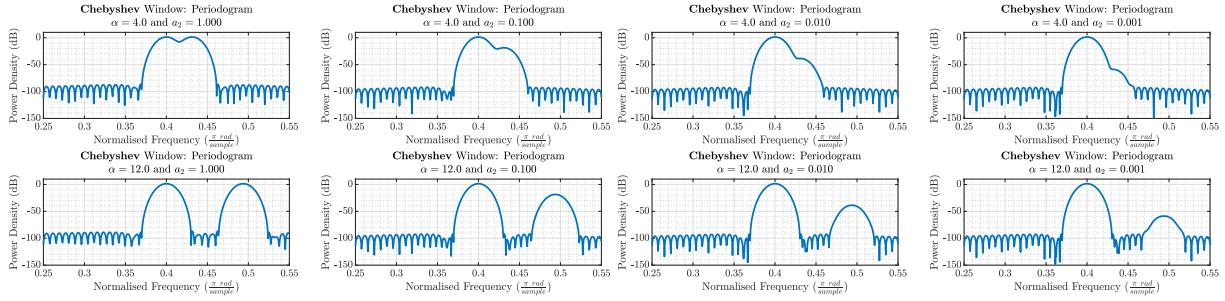


Figure 1.18: Chebyshev window: periodograms of $x(n)$ for varying a_2 and α .

Using the **Blackman-Tukey periodogram method** with $M = \frac{N}{4} = 64$ lags for spectral estimation of $x(n)$, figure 1.19 is obtained. We notice that the two frequency peaks are obtained only in the case of $a_2 = 1.0$, failing in all other cases, regardless α . This method trades resolution, $\Delta f_{BT} \sim \frac{1}{M}$ compared to $\Delta f_{Per} \sim \frac{1}{N}$ for a rectangular window periodogram, for variance. However, the signal under investigation, $x(n)$, is purely deterministic (no stochastic term, since $\sigma^2 = 0$) and thus the reduction in variance does not add any value to our estimate, leading solely to reduced resolution and hence unsuccessful identification of the two frequency components.

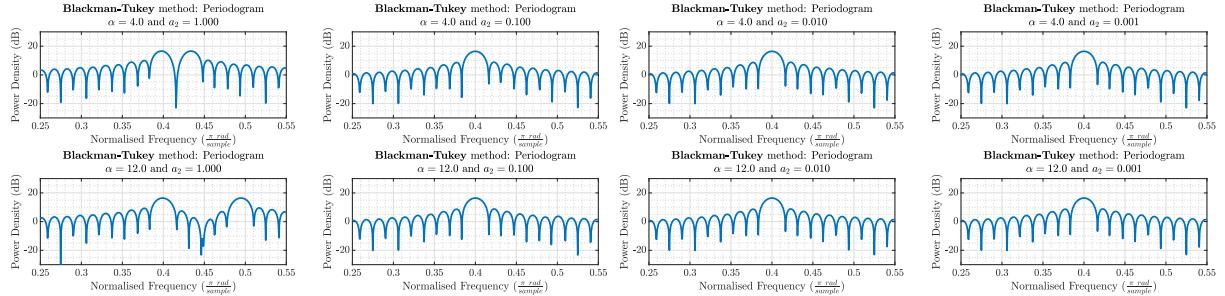


Figure 1.19: Blackman-Tukey method: periodograms of $x(n)$ for varying a_2 and α .

1.4 Periodogram-based Methods Applied to Real-World Data

- The sunspot time series and its Chebyshev-windowed periodogram method are provided at figure 1.20. The raw time series (blue) spectral estimates are compared to two preprocessed series:
 - a centered and detrended series (red, mean & detrend)
 - a logarithmic centered series (yellow, log & mean)

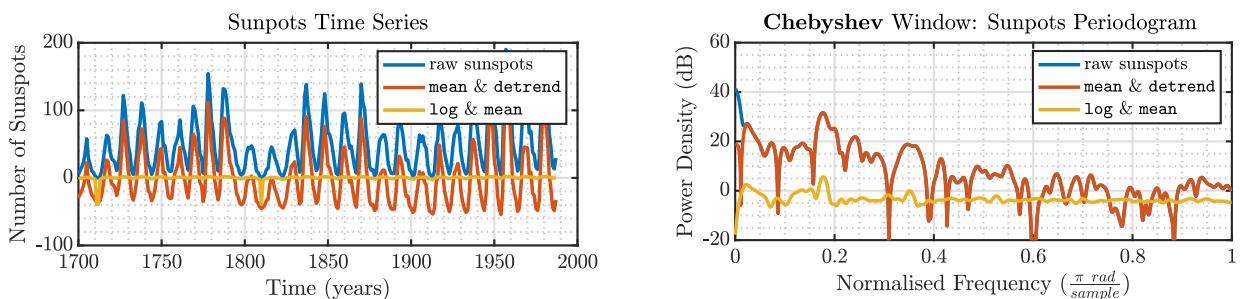


Figure 1.20: Sunspots: Chebyshev-windowed periodogram method, to raw and preprocessed series.

Subtraction of the mean, results in attenuation of the DC component ($f = 0$), while detrend removed low-frequency components. Consequently, the spectral estimate of the centered and detrended series is almost identical to the raw sunspots time series for frequencies $f \gtrsim 0.02$ (in rad/sample), while the lower frequency components are eliminated.

To avoid logarithms of zero, the logarithmic series is obtained by first adding a small constant to the raw sunspots time series (MATLAB $\text{eps} = 2.2204e - 16$) and then taking natural \log and removing the mean. Similarly, the DC component is also removed and the peaks at the same frequencies are observed, but now they are more noticeable (only peaks of interest are above 0dB).

- b) The Standard and Bartlett method periodograms of the EEG signal are provided at figure 1.21. The latter is easier to interpret since its peaks are accentuated. The peaks distinguished from the periodograms are at frequencies $f = 13, 26, 39, 50$ Hz as well as a wider peak at 8 – 10 Hz.

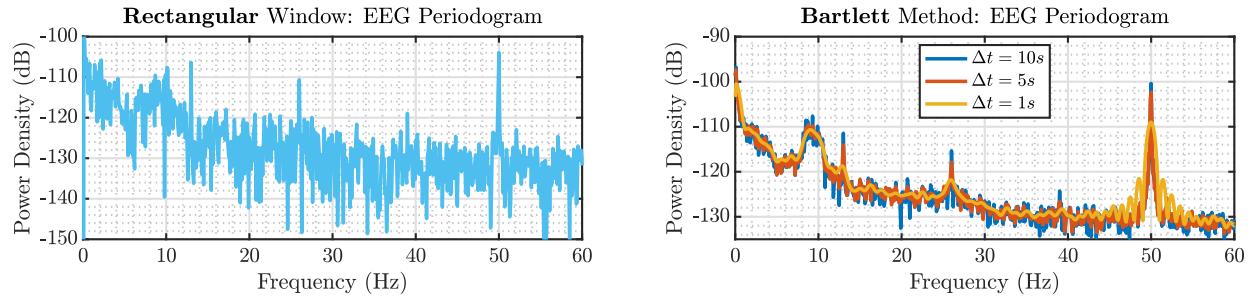


Figure 1.21: EEG: standard and Bartlett method periodogram.

As suggested by the instructions, the components in 8 – 10 Hz are due to the tiredness of the subject during the recording, adequately captured by the Bartlett method periodograms, regardless the window length Δt .

The peak at $f_0^{\text{SSEVP}} = 13$ Hz corresponds to the fundamental frequency of the SSEVP, whose harmonics can be noticed at frequencies $f_1^{\text{SSEVP}} = 26$ Hz and $f_2^{\text{SSEVP}} = 39$ Hz. Its third harmonic $f_3^{\text{SSEVP}} = 52$ Hz is hardly visible at the standard periodogram and the Bartlett method periodogram with $\Delta t = 10$ s, highly affected by the strong $f^{\text{PLI}} = 50$ Hz component due to the power-line interference.

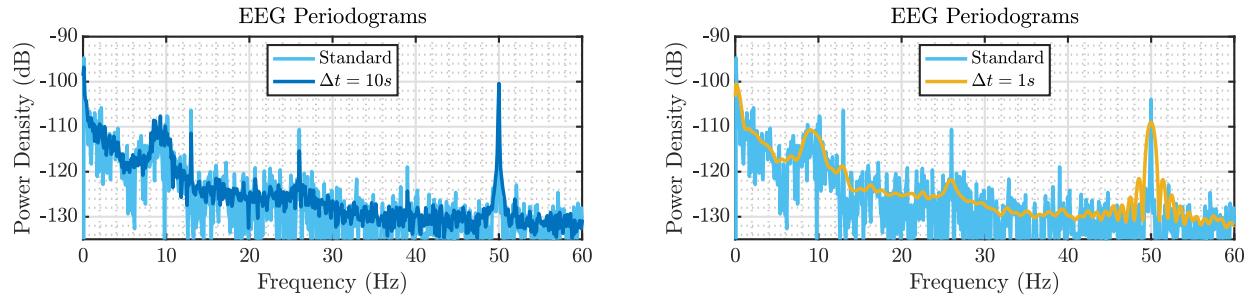


Figure 1.22: EEG: Bartlett method periodogram averaging window Δt .

Figure 1.22 depicts the comparison between the Standard and Bartlett method periodograms for averaging window lengths $\Delta t = 10$ s and $\Delta t = 1$ s. For $\Delta t = 10$ s, the periodogram has reduced variance compared to the standard periodogram, but inevitably reduced resolution. Nonetheless, the peaks of interest (harmonics of SSEVP, power-line interference frequency and 8 – 10 Hz band) are observable. On the other hand, for $\Delta t = 1$ s despite the even more reduced variance (by a factor of 10) the resolution is insufficient to capture the 3rd harmonic of SSEVP, but the rest of the peaks are visible.

Overall, the trade-off between variance and precision is illustrated, from the standard periodogram with best precision and maximum variance to the Bartlett method periodogram with $\Delta t = 1$ s with worst precision and least variance.

Chapter 2

Parametric and Line Spectra

2.1 Correlation Estimation

- a) Figure 2.1 shows the biased (red) and unbiased (blue) estimates of the autocorrelation function (ACF) as well as the correlogram spectral estimates obtained for: white gaussian noise (WGN), a noisy sinusoidal signal ($\sigma^2 = 1$) and filtered white gaussian noise.

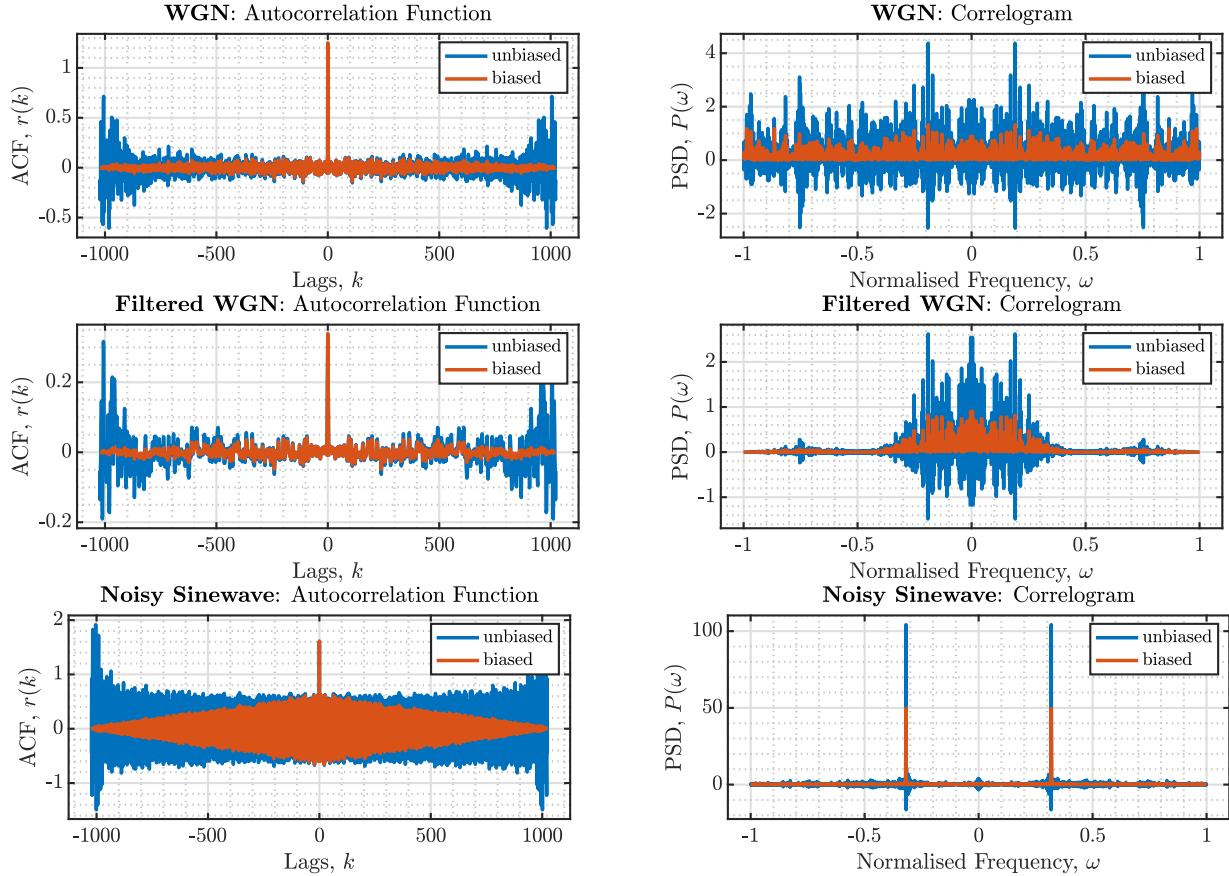


Figure 2.1: ACF and Correlogram: biased and unbiased estimates of various signals.

Regarding the ACF estimates, we verify that for small lags, $k \lesssim 150$, the two estimates are very similar, but for larger lags the unbiased estimates increase in value while the biased estimates fade out to zero. We now show that the biased and unbiased ACF estimates are obtained by windowing the ideal ACF $r_{xx}(k)$ with the Bartlett window, $w_B(k)$, and the rectangular window, $w_R(k)$, respectively.

$$\mathbb{E}[\hat{r}_{biased}(k)] = \sum_{n=k+1}^N \frac{N-k}{N} r_{xx}(k) = w_B(k) r_{xx}(k) \quad (2.1)$$

$$\mathbb{E}[\hat{r}_{unbiased}(k)] = \sum_{n=k+1}^N \frac{N-k}{N-K} r_{xx}(k) = w_R(k) r_{xx}(k) \quad (2.2)$$

Thus, using the Fourier transform pair ACF-PSD, we obtain the expected values of the correlograms $\mathbb{E}[\hat{P}_{biased}]$ and $\mathbb{E}[\hat{P}_{unbiased}]$ as the convolution of the true power spectrum with the Fourier transform of the Bartlett and rectangular window, respectively. The Fourier transform of the rectangular window is the *sinc* function, which introduces **negative** values, and does not preserve the positive semi-definiteness of the PSD. On the other hand, the Fourier transform of the Bartlett window is strictly non-negative¹, guaranteeing positive semi-definiteness. The correlograms in figure 2.1 verify our theoretical argument, where the unbiased estimates lead to negative PSD values while this is not the case for the biased estimates.

- b) Figure 2.2 illustrates the periodogram of 100 realisations of the random process $x(n)$, as well as the ensemble mean and standard deviation, where:

$$x(n) = 1.5\sin(2\pi 0.3n) + \sin(2\pi 0.6n) + 2\sin(2\pi 0.9n) + w(n) \quad w \sim \mathcal{N}(0, 1) \quad (2.3)$$

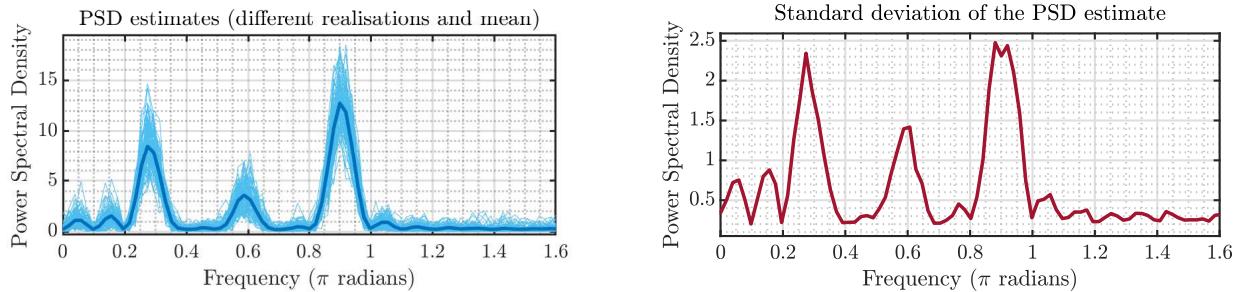


Figure 2.2: Correlogram: mean and standard deviation of $x(n)$.

When the biased ACF estimate is used, the obtained correlogram is an **inconsistent estimator** since:

$$\text{Var}[\hat{P}_{biased}(f)] = P_{xx}^2(f) \left[1 + \left(\frac{\sin(2\pi Nf)}{N \sin(2\pi f)} \right)^2 \right] \quad (2.4)$$

and as a result when $N \rightarrow \infty$, $\text{Var}[\hat{P}_{biased}(f)] \rightarrow P_{xx}^2(f) \gg 0$. This also explains the increased standard deviation of the periodogram at the peak frequencies of the ideal PSD, $P_{xx}(f)$, of the process.

- c) The experiments are repeated and the periodograms are expressed in *dB* this time, as illustrated in figure 2.3. Interestingly, the variance close to the peak frequencies, $f = 0.3, 0.6, 0.9$ Hz, is decreased. This is a counter-intuitive result, but it can be explained by the logarithmic function's gradient ($\frac{1}{x}$). Fluctuations around zero are significantly amplified, while values greater than 1 are attenuated (sub-linear trend). Consequently, since the power at peak frequencies is much greater than 1, its variance is squeezed, while at all other frequencies power is concentrated around zero, leading to amplified variance.

Clearly, this is an advantageous representation, since at frequencies of interest the spread-out is reduced and the interpretation of the periodograms is easier.

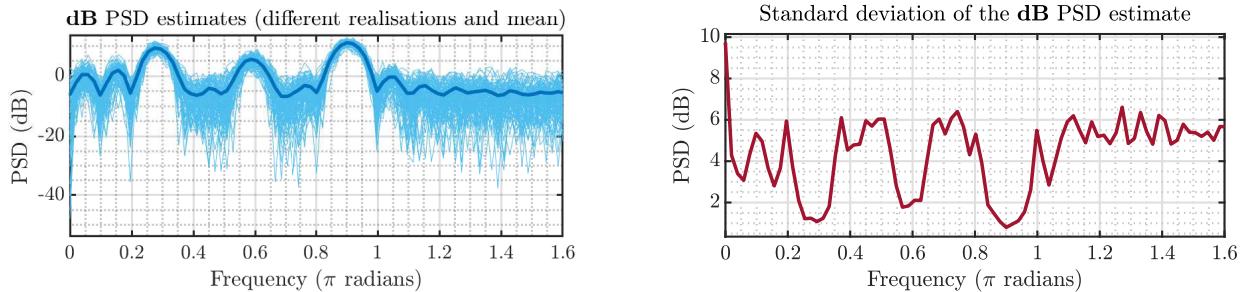


Figure 2.3: Correlogram: mean and standard deviation of $x(n)$ in *dB*.

¹see figure 1.10 from Assignment 1

- d) Figure 2.4 shows the periodograms of complex exponential signals (two sine waves) in noise for different number of samples and therefore frequency resolutions. As expected, for small number of samples ($n \leq 40$) the two peaks cannot be distinguished but for larger values ($n \geq 45$) the two peaks are visible. This observation agrees with theory, since the resolution of the (standard) periodogram is given by $\Delta f = \frac{0.89}{N}$ and the two complex signals have frequencies $f_1 = 0.3 \text{ Hz}$ and $f_2 = 0.32 \text{ Hz}$, hence discrimination is possible when:

$$\Delta f \leq f_2 - f_1 \implies N \geq \frac{0.89}{f_2 - f_1} = \frac{0.89}{0.32 - 0.3} = 44.5 \quad (2.5)$$

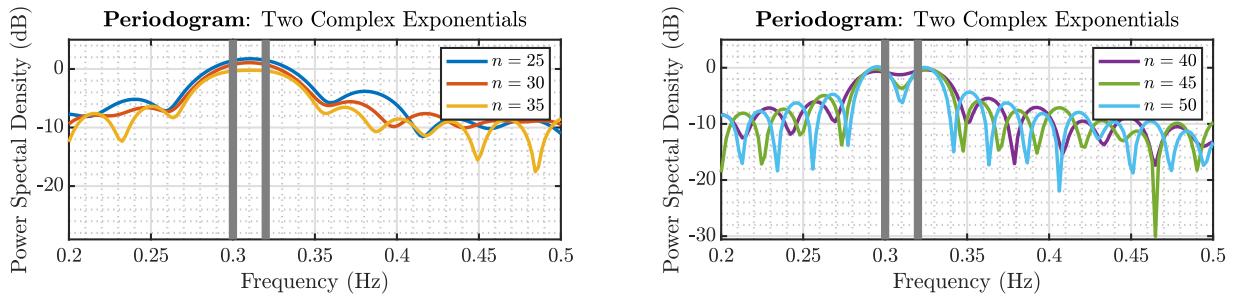


Figure 2.4: Periodogram: frequency resolution and peak identification.

Hence, despite the simplicity of the signals under investigation (two sine waves), the periodogram fails to adequately estimate their power spectral density when a small number of samples is available. Therefore, alternative methods (i.e subspace methods) must be used in these cases.

- e) The MULTiple SIgnal Classification algorithm (MUSIC) estimates the spectral density of a signal, using a **subspace** method. It assumes that the signal, $x(n)$, consists of p complex exponentials in the presence of Gaussian white noise. Given an autocorrelation matrix, $\mathbf{R}_{xx} \in \mathbb{R}^{M \times M}$, if its eigenvalues are sorted in decreasing order, the eigenvectors corresponding to the p largest eigenvalues (i.e. directions of largest variability) span the signal subspace, \mathbf{R}_s . The remaining $M - p$ eigenvectors span the orthogonal space, \mathbf{R}_n , where there is only noise. Let the noise eigenvectors \mathbf{v}_i and the helper vector \mathbf{e} such that:

$$\mathbf{v}_i, \quad i = p + 1, \dots, M \quad \text{and} \quad \mathbf{e} = [1 \quad e^{jw} \quad e^{j2w} \quad \dots \quad e^{j(M-1)w}]^T \quad (2.6)$$

then the MUSIC spectral estimate \hat{P}_{MU} is given by:

$$\hat{P}_{MU}(e^{jw}) = \frac{1}{\sum_{i=p+1}^M |e^H \mathbf{v}_i|^2} \quad (2.7)$$

where at signal frequencies $w_1, \dots, w_k, \dots, w_p$ the noise eigenvectors \mathbf{v}_i and the signal eigenvectors \mathbf{e}_k will be orthogonal (since \mathbf{R}_{xx} is Hermitian) and therefore \hat{P}_{MU} will have p peaks, as expected for a spectrum estimator of p complex exponentials.

The modified autocorrelation matrix \mathbf{R}_{xx} with $M = 14$ is obtained using MATLAB command:

```
[X, Rxx] = corrmtx(x, 14, "mod")
```

which is used by the MUSIC algorithm for spectral estimation

```
[S, F] = pmusic(Rxx, p, [], 1, "corr")
```

where p is the signal subspace dimensionality, a hyperparameter that must be tuned/selected.

In figure 2.5, the 100 realisations overlay of a two complex exponentials signal ($f_1 = 0.3 \text{ Hz}$ and $f_2 = 0.32 \text{ Hz}$) MUSIC spectrum estimate is provided, along with their standard deviation, for different values of hyperparameter p .

Despite the small number of samples available ($n = 30$) and thus the poor frequency resolution, the algorithm successfully identifies the two frequency components at f_1 and f_2 , when $p = 2$ is selected, unlike

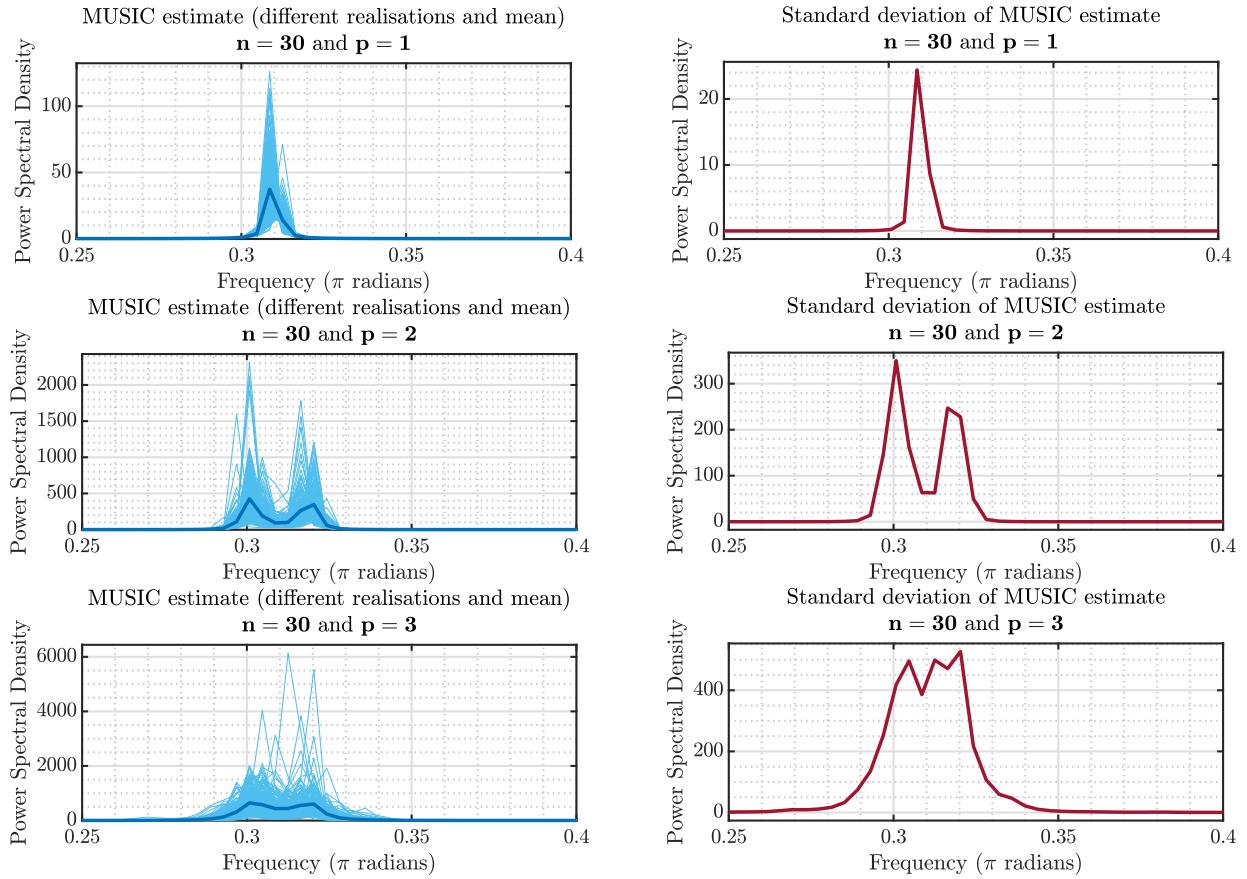


Figure 2.5: MUSIC: spectral estimation of two complex exponentials and signal space size p .

the periodogram that requires a larger amount of samples. Similar to the periodogram, the standard deviation of the estimator increases closer to the peak values, but most importantly, the reliability of the estimator deteriorates significantly when p is not matched with the nature of signal $x(n)$.

To sum up, MUSIC method is very powerful when a small number of samples is available but its chief disadvantage is that it requires the number of components p to be known in advance, so **it cannot be used in more general cases**, when prior knowledge of the signal is not provided. On the other hand, the periodogram needs a finer frequency resolution to detect closely-spaced sine waves (larger n), but it does not require any knowledge about the signal $x(n)$.

2.2 Spectrum of Autoregressive Processes

a) A p order Autoregressive process with parameters $\mathbf{a} \in \mathbb{R}^p$ satisfies the Yule-Walker (or normal) equation:

$$\mathbf{r}_{xx} = \mathbf{R}_{xx}\mathbf{a} \Rightarrow \mathbf{a} = \mathbf{R}_{xx}^{-1}\mathbf{r}_{xx} \quad (2.8)$$

where \mathbf{R}_{xx} the autocorrelation matrix (ACF) of signal $x(n)$. Equation (2.8) is meaningful for non-singular and thus invertible \mathbf{R}_{xx} . The biased estimator of ACF guarantees positive semi-definiteness and as a result the \mathbf{R}_{xx} can be inverted, providing solutions for the autoregressive parameters \mathbf{a} . On the other hand, as depicted in figure 2.1, the unbiased ACF estimator leads to indefiniteness and thus \mathbf{R}_{xx} may be singular.

b) The power spectral density of an AR process with parameters $\mathbf{a} = [2.76, -3.81, 2.65, -0.92]$ in Gaussian white noise ($\sigma^2 = 1$) is estimated using different AR model order $p = 2, 3, \dots, 14$. As illustrated in figure 2.6, low order models (i.e $p = 2$) fail to capture the behaviour of the process, identifying a single peak in the spectrum, while two are expected. Higher order models (i.e $p = 9$) provide better estimates, able to find the two peaks.

Intuitively, the larger the model order p , the more degrees of freedom available to capture the nature of the process. In figure 2.7 the noise power (mean squared prediction error) is illustrated as a function

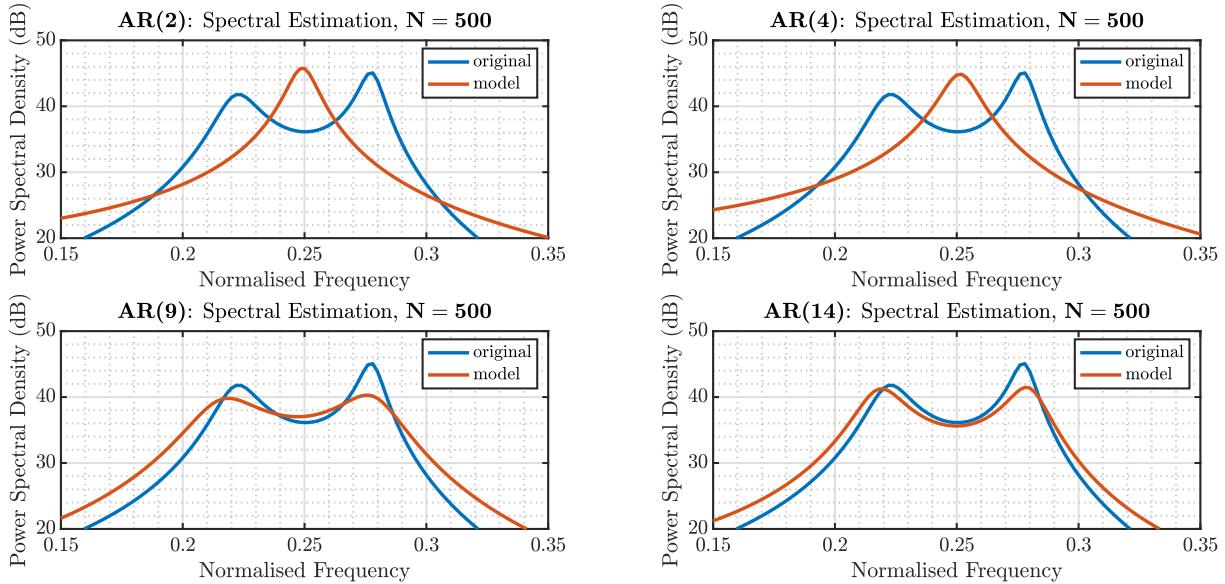


Figure 2.6: AR: spectrum estimates and model order p with $N = 500$ samples.

of model order p . Unsurprisingly, the error decreases for increasing model order though it plateaus for $p \leq 9$. As a result, to minimize model complexity and avoid fitting error (overfitting) model order $p = 9$ is selected.

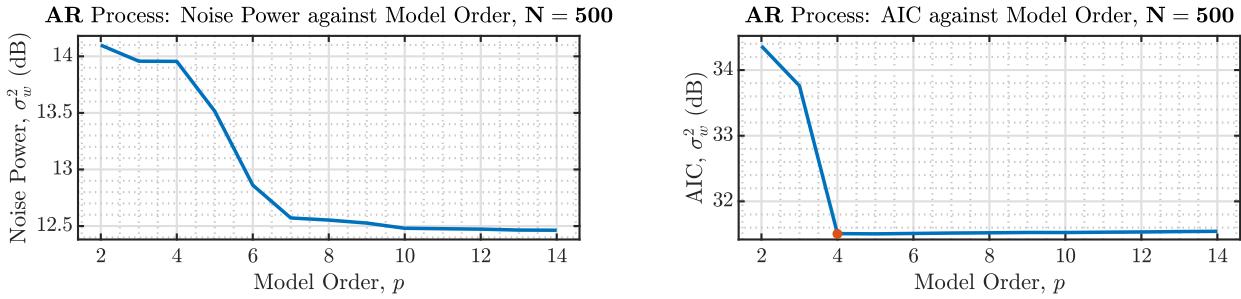


Figure 2.7: AR: noise power and model order p with $N = 500$ samples.

We also highlight that despite the fact that $p_{\text{original}} = 4$, due to the small number of available sample ($N = 500$), when $p = 4$ is selected the fitted model performs very poorly, thus either the number of samples should be increased or a higher order model should be used instead.

- c) The experiment is repeated for the same AR process, but using $N = 10000$ samples and figures 2.8 and 2.9 are obtained. When $p < p_{\text{original}} = 4$, the model still does not have the capacity to model the process (under-modelling) while for $p > p_{\text{original}} = 4$ the error plateaus. Despite the fact that AR(4) model identifies the two peaks, we note that AR(5) and all higher order models track spectrum much better. To avoid overfitting the noise and preserve generalisation of the model, criteria such as the Akaike Information Criterion (AIC) or the Bayesian Information Criterion can be used to penalise higher order models. According to the AIC, AR(4) model is selected for this experiment, agreeing with the true model order.

2.3 Principal Component Analysis

- a) The singular values of \mathbf{X} and $\mathbf{X}_{\text{noise}}$ as well as their squared error between each singular value are depicted in figure 2.10.

The noiseless input matrix, \mathbf{X} , has 3 non-zero singular values, which reflect its rank. The noise corrupted matrix, $\mathbf{X}_{\text{noise}}$, has 3 dominant singular values, corresponding to the eigenvectors spanning the signal subspace, while the rest non-zero singular values correspond to the noise subspace. Moreover, the signal subspace singular values are offset from the true singular values of \mathbf{X} , due to the noise corruption.

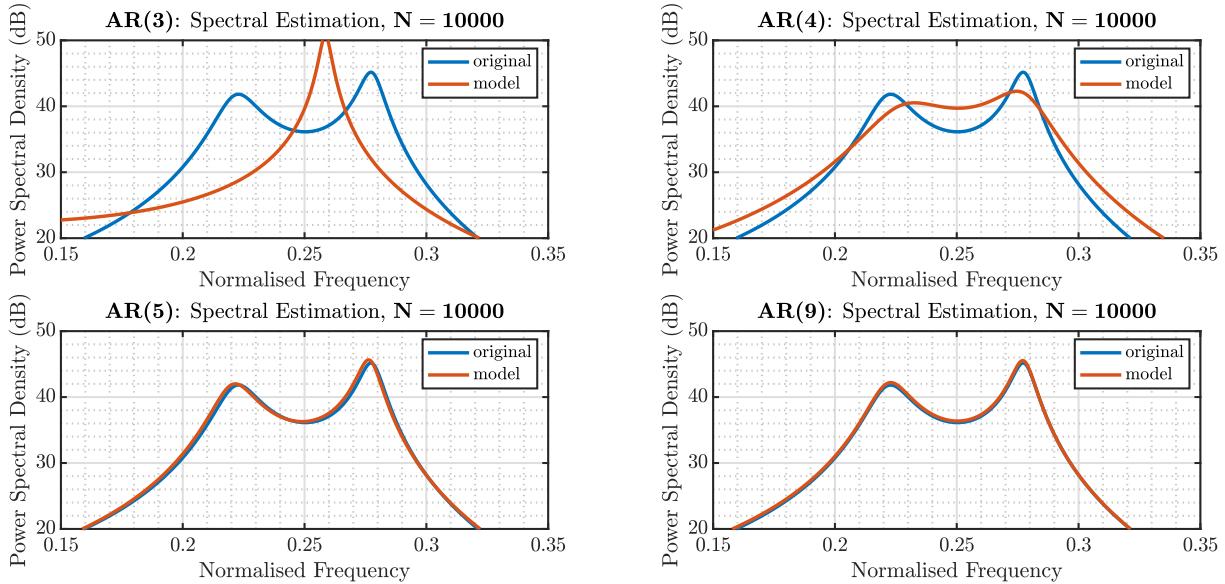


Figure 2.8: AR: spectrum estimates and model order p with $N = 10000$ samples.

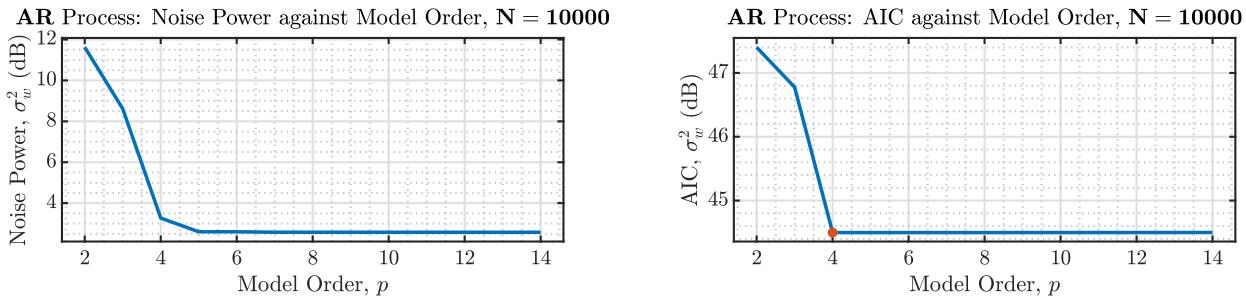


Figure 2.9: AR: soise power and model order p with $N = 10000$ samples.

The noise subspace eigenvalues are half the magnitude compared to the signal subspace eigenvalues and as a result they can be distinguished by thresholding. Nonetheless, if the noise power is increased and its singular values are of comparable magnitude to the signal subspace eigenvalues, then the rank of \mathbf{X}_{noise} becomes hard to identify.

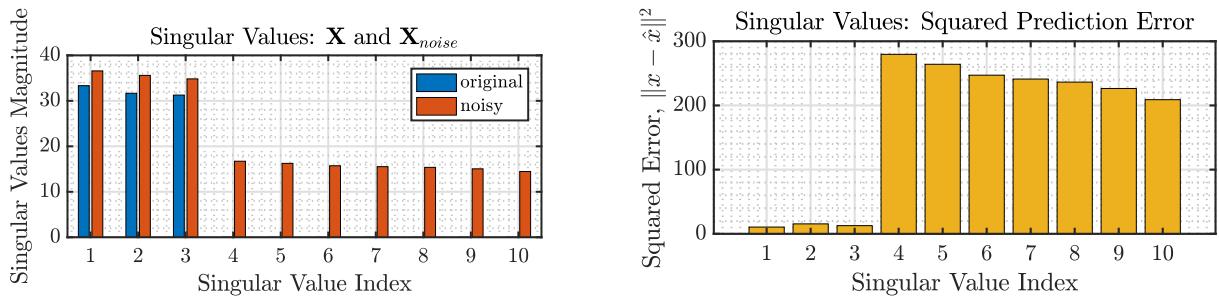


Figure 2.10: Singular Value Decomposition: squared prediction error for corrupted signal.

- b) Low-rank approximation of the noisy data \mathbf{X}_{noise} are obtained by keeping only its r most principle components. This denoising operation relies on the assumption that the r most significant components can adequately explain the data, while the rest are pure noise. Figure 2.11 shows the approximation error ($\|\mathbf{X} - \tilde{\mathbf{X}}_{noise}\|_F$) for different values of r . Interestingly, the error is minimised for $r = r_{true} = 3$, demonstrating the power of this method, Principle Component Analysis (PCA).
- c) The parameter matrix \mathbf{B} estimation is performed using OLS and PCR methods. The estimation errors between $\mathbf{Y}-\mathbf{Y}_{OLS}$ and $\mathbf{Y}-\mathbf{Y}_{PCR}$ for the train and test datasets are illustrated for different values of r in figure 2.12. We notice that for $r \geq 3$ the OLS and PCR methods are score equally well at both the train and the test datasets. In more detail, while OLS performs better in training by 0.4% PCR outperforms by

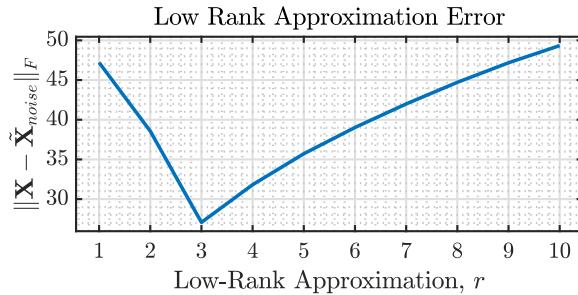


Figure 2.11: Singular Value Decomposition: squared prediction error for corrupted signal.

0.7% in test set.

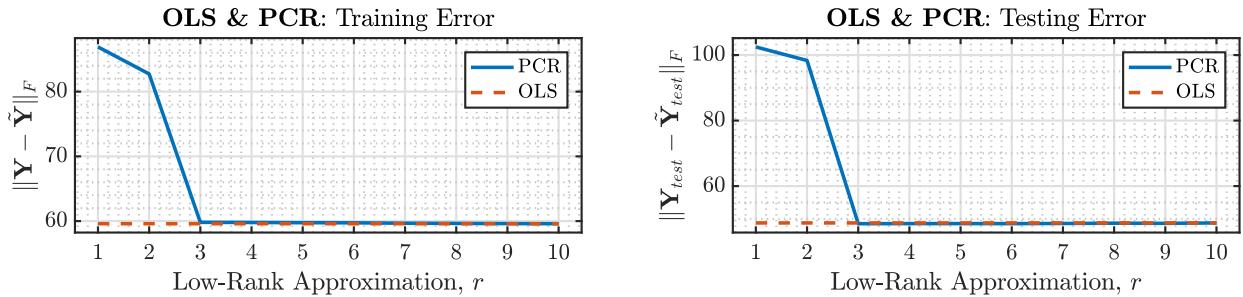


Figure 2.12: Singular Value Decomposition: train & test error.

- d) Model evaluation of OLS and PCR models is performed over an ensemble of 100 test realisations of the stochastic process generating \mathbf{X}_{noise} . Figure 2.13 summarises the prediction errors of the two models, verifying the results of the previous part, where PCR outperforms OLS by 1.2%.

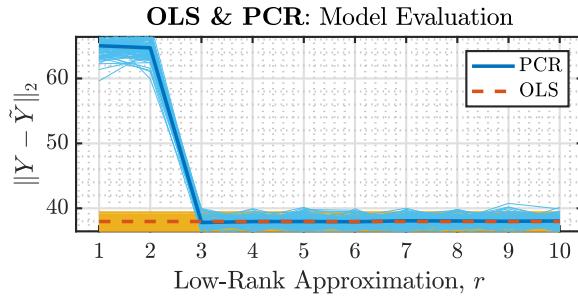


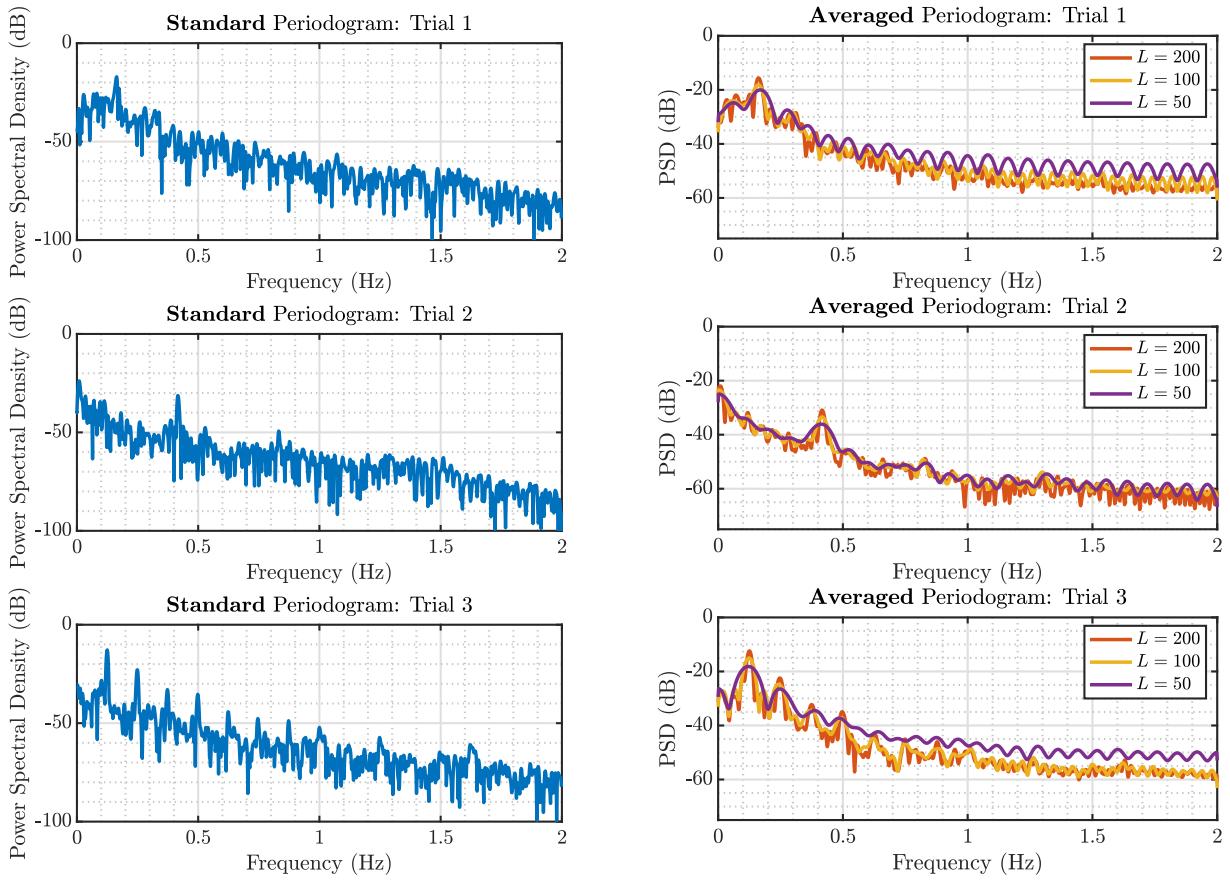
Figure 2.13: Singular Value Decomposition: model evaluation.

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

- a) The standard and averaged periodograms with different window lengths $L \in \{50, 100, 200\}$ are obtained and illustrated in figure 2.14. In detail, Bartlett's method periodogram is used, where:
- 1) the signal is split to M non-overlapping segments of length L
 - 2) the standard periodogram (rectangular window) of each segment is calculated
 - 3) the M periodograms are averaged

Increasing M (or decreasing L) trades frequency resolution for variance, which is reflected in 2.14, where the standard periodogram, which can be treated as a special case of the Bartlett's method with $M = 1$, has finest frequency resolution but largest variance, while for $L = 50$ and $M \rightarrow \max$, variance is minimised in price of larger frequency resolution.

- b) The three trials were carried out under different breathing conditions, which is reflected by the Breaths per Minute (BPM) rate. Table 2.1 summarises the conditions and the corresponding expected and observed BPM measurements for each trial.


 Figure 2.14: RRI: standard and averaged periodograms with different window lengths L .

	Conditions	Expected Range (BPM)	Observation (BPM)
Trial 1	normal & unconstraint breathing	12 – 35	19.69
Trial 2	fast breathing	35 – 50	49.92
Trial 3	slow breathing	8 – 15	15

Table 2.1: RRI: conditions and observed BPM.

The observed peaks for the three trials are at frequencies $f_1 = 0.1641 \text{ Hz}$, $f_2 = 0.416 \text{ Hz}$ and $f_3 = 0.125 \text{ Hz}$. For trial 1, both the standard and the averaged periodograms have a peak at $f_1 = 0.1641 \text{ Hz}$, however the averaged periodograms due to their inadequate frequency resolution fail to capture its harmonics, which are hardly visible also at the standard periodogram. Trial 2 peak at $f_2 = 0.416 \text{ Hz}$ and its second and third harmonics are visible using any of the provided methods. Lastly, trial 3 peak at $f_3 = 0.125 \text{ Hz}$ and its harmonics are clearly captured by both the standard and the averaged periodograms.

- c) The power spectral density of the three trials is estimated by fitting p order AR processes and figures 2.15, 2.16 and 2.17 are obtained.

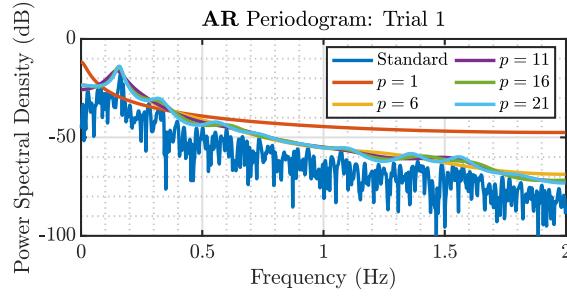


Figure 2.15: RRI Trial 1: AR process spectral estimation.

For trial 1, for $p \geq 11$, the fundamental frequency $f_1 = 0.1641$ Hz and its 3 first harmonics are identified, corroborating with the results of the previous part.

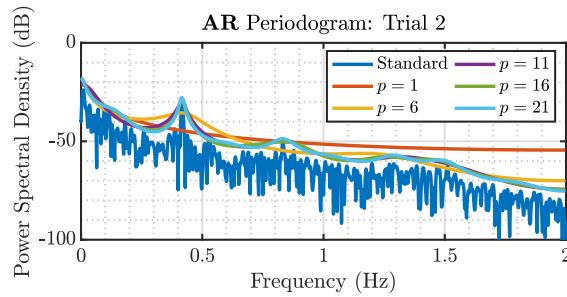


Figure 2.16: RRI Trial 2: AR process spectral estimation.

Figure 2.16 depicts the AR spectral estimate of trial 2, which again for $p \geq 11$ adequately captures the fundamental frequency and its first three harmonics.

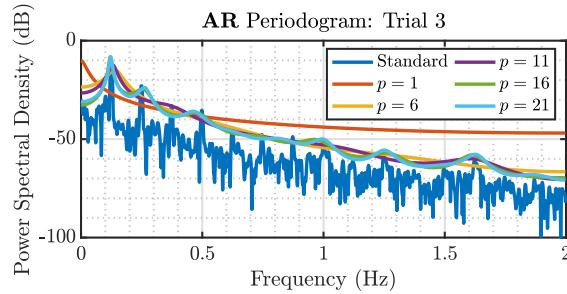


Figure 2.17: RRI Trial 3: AR process spectral estimation.

Last but not least, for $p \geq 16$ trial 3 AR(p) process model discriminated all six first harmonics, verifying the respiration rate reported in the previous part.

Overall, the AR process models can provide reliable estimates when the generative process order p is known, or noise power is negligible compared to the signal power. When under-modelling is performed ($p < p_{true}$) the estimates cannot identify the frequency peaks, while in presence of noise, over-modelling ($p > p_{true}$) can lead to overfitting noise and thus providing faulty peaks. Nonetheless, compared to the standard and averaged periodogram methods, the variance, frequency resolution trade-off is not present in case of AR models.

Chapter 3

Adaptive Signal Processing

3.1 The Least Mean Square (LMS) Algorithm

a) A general AR(2) process $x(n)$ with parameters a_1, a_2 satisfies the difference equation:

$$x(n) = a_1 x(n-1) + a_2 x(n-2) + \eta(n) \quad (3.1)$$

where $\eta(n) \sim \mathcal{N}(0, \sigma_\eta^2)$. The Least Mean Square (LMS) algorithm is used to approximate the autoregressive parameters a_1, a_2 from data, treating $\mathbf{x}(n) = [x(n-1), x(n-2)]^T$ and $y(n) = x(n)$ as the input (features) vector and the output (target), respectively.

The correlation matrix \mathbf{R}_{xx} of the input vector $\mathbf{x}(n)$ is given by:

$$\mathbf{R}_{xx} = \mathbb{E}[\mathbf{x}(n)\mathbf{x}^T(n)] = \mathbb{E} \begin{bmatrix} x(n-1)x(n-1) & x(n-1)x(n-2) \\ x(n-1)x(n-2) & x(n-2)x(n-2) \end{bmatrix} = \begin{bmatrix} r_{xx}(0) & r_{xx}(1) \\ r_{xx}(1) & r_{xx}(0) \end{bmatrix} \quad (3.2)$$

where $r_{xx}(k)$ the autocorrelation function (ACF) of $x(n)$. To obtain the ACF of the AR(2) process, multiply equation (3.1) by $x(n-k)$ and take expectations:

$$\begin{aligned} r_{xx}(k) &= \mathbb{E}[x(n)x(n-k)] = \mathbb{E}\left[a_1 x(n-1)x(n-k) + a_2 x(n-2)x(n-k) + \eta(n)x(n-k)\right] \\ r_{xx}(k) &= \mathbb{E}[x(n)x(n-k)] = a_1 \mathbb{E}[x(n-1)x(n-k)] + a_2 \mathbb{E}[x(n-2)x(n-k)] + \mathbb{E}[\eta(n)x(n-k)] \end{aligned} \quad (3.3)$$

Notice that $\mathbb{E}[\eta(n)x(n-k)]$ vanishes when $k > 0$, then:

$$r_{xx}(0) = a_1 r_{xx}(1) + a_2 r_{xx}(2) + \sigma_\eta^2 \quad (3.4)$$

$$r_{xx}(k) = a_1 r_{xx}(k-1) + a_2 r_{xx}(k-2), k > 0 \quad (3.5)$$

Note that ACF is an even function or equivalently $r_{xx}(-k) = r_{xx}(k), \forall k \in \mathbb{Z}$, thus using the true process parameters $a_1 = 0.1$, $a_2 = 0.8$ and $\sigma_\eta^2 = 0.25$, we obtain the three simultaneous equations with three unknowns:

$$r_{xx}(0) = a_1 r_{xx}(1) + a_2 r_{xx}(2) + \sigma_\eta^2 \quad (3.6)$$

$$r_{xx}(1) = a_1 r_{xx}(0) + a_2 r_{xx}(-1) = a_1 r_{xx}(0) + a_2 r_{xx}(1) \quad (3.7)$$

$$r_{xx}(2) = a_1 r_{xx}(1) + a_2 r_{xx}(0) \quad (3.8)$$

with the unique solution $r_{xx}(k) = [\frac{25}{27}, \frac{25}{54}, \frac{85}{108}]$ for $k = 1, 2, 3$.

Hence by substitution in equation (3.2) we obtain the correlation matrix of the input vector $\mathbf{x}(n)$:

$$\mathbf{R}_{xx} = \frac{25}{54} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \quad (3.9)$$

Convergence of the LMS algorithm depends on the step size μ , which should satisfy:

$$0 < \mu < \frac{2}{\lambda_{max}} \quad (3.10)$$

where λ_{max} the largest eigenvalue of the correlation matrix \mathbf{R}_{xx} . Performing the eigendecomposition of \mathbf{R}_{xx} , we obtain $\lambda_1 = 0.4630$ and $\lambda_2 = \lambda_{max} = 1.3889$ and as a result, convergence to the Wiener optimal solution is guaranteed for:

$$0 < \mu < 1.44 \quad (3.11)$$

- b) The LMS algorithm is implemented and tested on $N = 1000$ samples of 100 different realisations of $x(n)$ process. The squared prediction error over time for one and the average of the 100 realisations is provided at figure 3.1 for two different step-sizes $\mu_1 = 0.01$ and $\mu_2 = 0.05$.

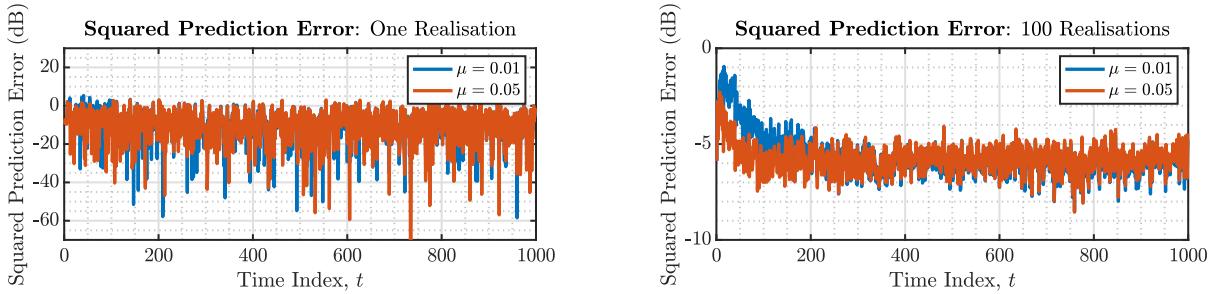


Figure 3.1: LMS: squared prediction error over time and μ convergence rate.

Concentrating on the averaged of 100 realisation prediction error (reduced variance of estimate), we notice that the larger step-size $\mu_2 = 0.05$ converges within 50 timesteps, while μ_1 converges in 200 timesteps, verifying the theoretical argument that larger μ values allow steeper decent of the error surface, as long as the condition (3.11) is satisfied. Nonetheless, fast convergence is traded with larger oscillations around the true parameter values, motivating the use of an adaptive step-size, which decays over time.

- c) The theoretical Misadjustment, \mathcal{M}_{LMS} , of the LMS algorithm is obtained by the approximation formula:

$$\mathcal{M}_{LMS} \approx \frac{\mu}{2} \text{Tr}\{\mathbf{R}_{xx}\} = \frac{\mu}{2} 1.8519 \quad (3.12)$$

where \mathbf{R}_{xx} from (3.9) is used.

According to figure 3.1 the squared prediction error plateaus for both μ values after $t > 200$. To guarantee that a steady-state has been reached, we time average the squared prediction error for $t_0 > 500$, in order to obtain a Mean Squared Error (MSE) estimate. The empirical misadjustment, \mathcal{M}_{emp} , is then obtained using the formula:

$$\mathcal{M}_{emp} = \frac{\text{MSE}}{\sigma_\eta^2} - 1 \quad (3.13)$$

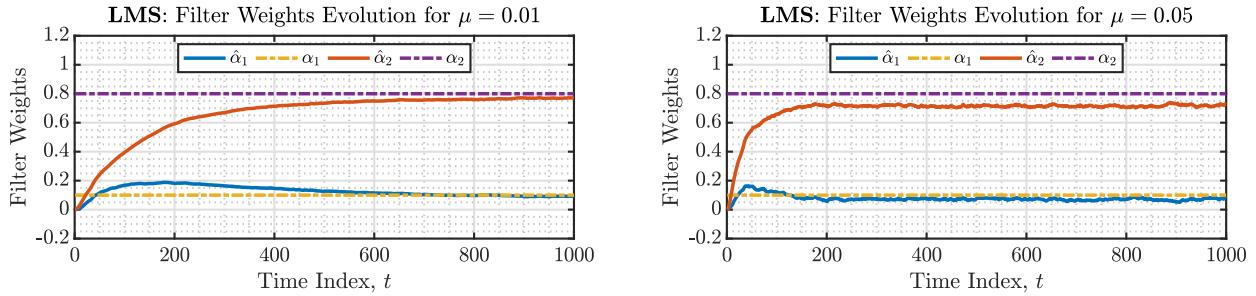
Table 3.1 summarises the empirical and theoretical misadjustments of the simple LMS algorithm for the $x(n)$ process.

- d) The evolution of the LMS filter coefficients over time is shown in figure 3.2, along with the true AR(2) process parameters, for step-sizes $\mu_1 = 0.01$ and $\mu_2 = 0.05$. The illustrated curves represent the average of the coefficients of 100 independently trained LMS filters on different realisations of the same $x(n)$ stochastic process.

μ	\mathcal{M}_{LMS}	\mathcal{M}_{emp}
0.01	0.0093	0.00766
0.05	0.0463	0.04912

Table 3.1: LMS: empirical and theoretical approximation of misadjustment comparison matrix.

Clearly, for both μ values, $\hat{\alpha}_1$ coefficient converges to the true α_1 parameter, while the second coefficient $\hat{\alpha}_2$ has a negative offset (3.75% for $\mu_1 = 0.01$ and 10% for $\mu_2 = 0.05$). Moreover, we note that a larger step-size (i.e μ_2) compromises oscillations around the true values and a greater offset for faster convergence. In more detail, within 200 timesteps μ_1 reaches the 10% error bounds, while μ_2 needs 350 timesteps.


 Figure 3.2: LMS: steady state values of the adaptive filter coefficients for different step-sizes μ .

e) Let the objective function \mathcal{J}_2 , such that:

$$\mathcal{J}_2(n) = \frac{1}{2} \left(e^2(n) + \gamma \| \mathbf{w}(n) \|_2^2 \right) \quad (3.14)$$

Expressing the error term, $e(n)$, in terms of the input vector \mathbf{x}_n and the target value y_n :

$$\mathcal{J}_2(n) = \frac{1}{2} \left(\| y(n) - \mathbf{w}(n)^T \mathbf{x}(n) \|_2^2 + \gamma \| \mathbf{w}(n) \|_2^2 \right) \quad (3.15)$$

$$\mathcal{J}_2(n) = \frac{1}{2} \left((y(n) - \mathbf{w}(n)^T \mathbf{x}(n))^T (y(n) - \mathbf{w}(n)^T \mathbf{x}(n)) + \gamma \mathbf{w}(n)^T \mathbf{w}(n) \right) \quad (3.16)$$

Calculating the gradient of the objective function, $\nabla_{\mathbf{w}} \mathcal{J}_2$, with respect to the weights \mathbf{w} :

$$\nabla_{\mathbf{w}} \mathcal{J}_2(n) = -(y(n) - \mathbf{w}(n)^T \mathbf{x}(n)) \mathbf{x}(n) + \gamma \mathbf{w}(n) \quad (3.17)$$

$$\nabla_{\mathbf{w}} \mathcal{J}_2(n) = -e(n) \mathbf{x}(n) + \gamma \mathbf{w}(n) \quad (3.18)$$

Applying the gradient descent update to weights \mathbf{w} to minimise the objective function \mathcal{J}_2 :

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu \nabla_{\mathbf{w}} \mathcal{J}_2(n) \quad (3.19)$$

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu (-e(n) \mathbf{x}(n) + \gamma \mathbf{w}(n)) \quad (3.20)$$

$$\mathbf{w}(n+1) = (1 - \mu \gamma) \mathbf{w}(n) + \mu e(n) \mathbf{x}(n) \quad (3.21)$$

Hence we proved that the Leaky LMS algorithm following the update rule in (3.21) is equivalent to the minimisation of the objective function \mathcal{J}_2 defined in (3.14).

f) In figure 3.3 the Leaky LMS filter coefficients for different leakage coefficient γ and step-sizes μ are provided, against the true AR(2) process parameters.

The predicted filter coefficients do not converge to the true parameters, and increasing γ values introduce a greater bias between the estimates and the process autoregressive parameters.

The offline, non-adaptive but optimal solution to this linear system is the Wiener filter, which has access to all the observed samples and relies on the construction of the autocorrelation matrix, $\mathbf{R} = \mathbb{E}[\mathbf{XX}^T]$, and the crosscorrelation vector (between desired output value and input vector), \mathbf{p} , then the optimal filter weights, \mathbf{w}_* , are given by:

$$\mathbf{w}_* = \mathbf{R}^{-1} \mathbf{p} \quad (3.22)$$

where \mathbf{R} needs to be invertible. The LMS algorithm can be shown to converge to this solution if (3.10) is true. Invertibility of the autocorrelation matrix is not guaranteed, since it is positive semi-definite and not strictly positive definite. The Leaky LMS algorithm can be shown to converge to the solution:

$$\mathbf{w}_{reg} = (\mathbf{R} + \gamma \mathbf{I})^{-1} \mathbf{p} \quad (3.23)$$

where the leakage coefficient $\gamma > 0$ acts as a regularization parameter ensuring invertibility of the "modified" autocorrelation matrix ($\mathbf{R} + \gamma \mathbf{I}$). Nonetheless, this is a common example of bias-variance trade-off since larger values of γ introduce bias but penalise variance of estimates (due to the L_2 norm term in \mathcal{J}_2).

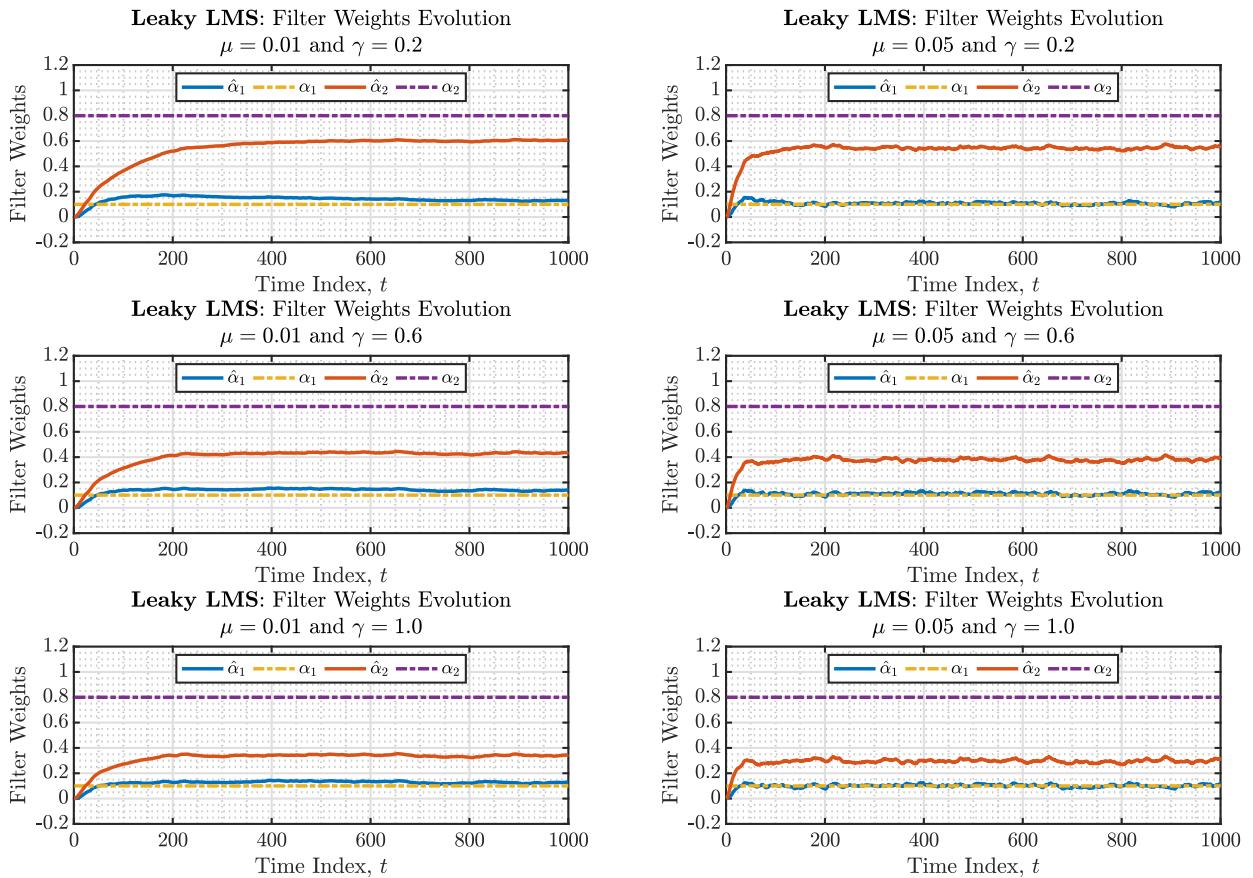


Figure 3.3: Leaky LMS: filter weights evolution for different leakage coefficient γ .

3.2 Adaptive Step Sizes

- a) Let a Moving Average process $x(n)$ of order $q = 1$, MA(1), such that:

$$x(n) = 0.9\eta(n-1) + \eta(n), \quad \eta \sim \mathcal{N}(0, 0.5) \quad (3.24)$$

When LMS is used with a constant step-size μ , a trade-off between convergence speed, smoothness and oscillations magnitude at steady-state is inevitable. Gradient Adaptive Step-Size (GASS) algorithms

attempt to combine the positive aspects of both large and small values of μ by adapting its magnitude according to the gradient $\nabla_\mu \mathcal{J}$, where \mathcal{J} the objective function minimised by the adaptive filter.

Different GASS algorithms, namely Benvenist, Ang & Farhang and Matthews & Xi, are implemented and their performance is compared to simple constant step-size LMS algorithm. Weight error and squared prediction error curves are provided at figure 3.4.

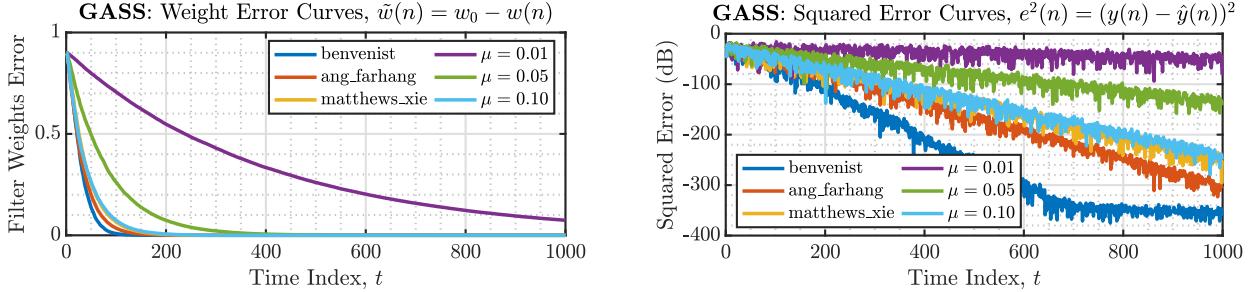


Figure 3.4: GASS LMS: weight error and squared prediction error curves.

All GASS algorithms converge to the true process parameters faster than simple LMS (within less than 50 steps), while unsurprisingly, the most computationally intensive algorithm, Benvenist ($\mathcal{O}(M^2)$), adapts first of all and scores the smallest squared prediction error. Moreover, their steady-state error is much smaller (below $-300dB$) than the fixed step-size LMS algorithm (at $-200dB$, $-120dB$, $-30dB$ for $\mu = 0.1, 0.05, 0.01$, respectively), improving the EMSE from the previous part.

- b) Starting from the update equation based on the *a posteriori* error $e_p(n) = d(n) - \mathbf{x}(n)^T \mathbf{w}(n+1)$:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e_p(n) \mathbf{x}(n) \quad (3.25)$$

Multiply both sides with $-\mathbf{x}(n)^T$ and add $d(n)$ to construct $e_p(n)$ on the LHS:

$$d(n) - \mathbf{x}(n)^T \mathbf{w}(n+1) = d(n) - \mathbf{x}(n)^T \mathbf{w}(n) - \mathbf{x}(n)^T \mu e_p(n) \mathbf{x}(n) \quad (3.26)$$

Note that the LHS term is the *a posteriori* error, $e_p(n)$, while the first RHS term the *a priori* error, $e(n)$:

$$e_p(n) = e(n) - \mu e_p(n) \|\mathbf{x}(n)\|^2 \quad (3.27)$$

$$e_p(n) = e(n) \frac{1}{1 + \mu \|\mathbf{x}(n)\|^2} \quad (3.28)$$

Substituting (3.28) in update equation (3.25):

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu \frac{1}{1 + \mu \|\mathbf{x}(n)\|^2} e(n) \mathbf{x}(n) \quad (3.29)$$

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{1}{\frac{1}{\mu} + \|\mathbf{x}(n)\|^2} e(n) \mathbf{x}(n) \quad (3.30)$$

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\beta}{\varepsilon + \|\mathbf{x}(n)\|^2} e(n) \mathbf{x}(n) \quad (3.31)$$

where for $\beta = 1$ and $\varepsilon = \frac{1}{\mu}$ we showed that the NLMS update (3.31) is equivalent to the update equation based on the *a posteriori* error, given by (3.25).

- c) The implementation of Generalized Nnornalized Gradient Descent (GNGD) algorithm is compared with the Benvenist GASS algorithm, and the weight error and squared prediction error curves are provided at figure 3.5. We notice that the GNGD algorithm converges to the true process parameters within 30

timesteps, faster than the Benvenist GASS algorithm which reaches a steady-state in 50 timesteps. This is also reflected in the squared prediction error curves, where the GNGD error is always smaller than the Benvenist GASS prediction error.

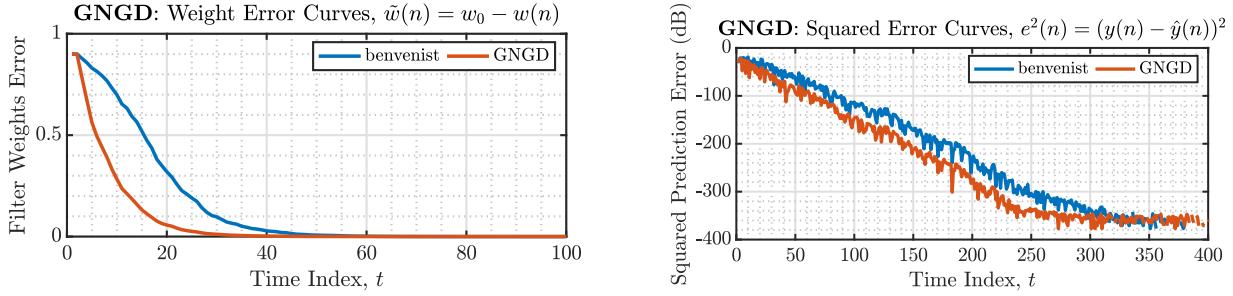


Figure 3.5: GNGD vs Benvenist GASS: weight error and squared prediction error curves.

In the previous part we empirically showed the superiority of the Benvenist GASS algorithm over the other GASS algorithms and the standard LMS, in terms of performance (speed & prediction error). Nonetheless, this improved performance comes with an increased computational complexity load.

Let M the model order (number of lags in MA or AR model process), then the input vector $\mathbf{x}(n) \in \mathbb{R}^M$.

Each update of the Benvenist GASS algorithm involves the calculation of

- the outer product $\mathbf{x}(n-1)\mathbf{x}(n-1)^T$
- the matrix product $[I - \mu(n-1)\mathbf{x}(n-1)\mathbf{x}(n-1)^T]\boldsymbol{\psi}(n-1)$

which both have quadratic complexity in M . Hence Benvenist GASS algorithm is $\mathcal{O}(M^2)$.

Each update of the GNGD algorithm involves only inner product calculations, additions of M -dimensional vectors and scalar operations, all bounded by linear complexity in M . Therefore, GNGD algorithm is $\mathcal{O}(M)$.

Surprisingly, the GNGD algorithm does not only perform better (convergence speed and prediction error) than the Benvenist GASS algorithm, but it is also computationally less expensive.

3.3 Adaptive Noise Cancellation

- a) Let the pure sine wave $x(n)$, the noisy signal $s(n)$, the noise term $\eta(n)$ and the ALE filter output $\hat{x}(n; \Delta)$, parametrised by the delay parameter Δ . Then the Mean Squared Error (MSE) is given by:

$$\text{MSE} = \mathbb{E} \left[(s(n) - \hat{x}(n; \Delta))^2 \right] = \mathbb{E} \left[(x(n) + \eta(n) - \hat{x}(n; \Delta))^2 \right] \quad (3.32)$$

$$= \mathbb{E} \left[(\eta(n) + (x(n) - \hat{x}(n; \Delta)))^2 \right] \quad (3.33)$$

$$= \mathbb{E} [\eta^2(n)] + \mathbb{E} [(x(n) - \hat{x}(n; \Delta))^2] + 2\mathbb{E} [\eta(n)(x(n) - \hat{x}(n; \Delta))] \quad (3.34)$$

The first term, noise power $\mathbb{E}[\eta^2(n)]$, is independent of Δ , while the second term, Mean Squared Prediction Error $\mathbb{E}[(x(n) - \hat{x}(n; \Delta))^2]$ is not a function of noise $\eta(n)$. Hence, the last term only involves both the delay Δ (through $\hat{x}(n)$) and the noise term, so we will minimise it:

$$\min_{\Delta \in \mathbb{N}} \mathbb{E} [\eta(n)(x(n) - \hat{x}(n; \Delta))] \quad (3.35)$$

Using the fact that $x(n)$ and $\eta(n)$ are uncorrelated the term $\mathbb{E}[\eta(n)x(n)]$ vanished:

$$\min_{\Delta \in \mathbb{N}} \mathbb{E} \left[\eta(n) \hat{x}(n; \Delta) \right] \rightarrow \min_{\Delta \in \mathbb{N}} \mathbb{E} \left[(u(n) + 0.5u(n-2)) \mathbf{w}^T \mathbf{u}(n; \Delta) \right] \quad (3.36)$$

$$\rightarrow \min_{\Delta \in \mathbb{N}} \mathbb{E} \left[(u(n) + 0.5u(n-2)) \sum_{i=0}^{M-1} w_i s(n - \Delta - i) \right] \quad (3.37)$$

$$\rightarrow \min_{\Delta \in \mathbb{N}} \mathbb{E} \left[(u(n) + 0.5u(n-2)) \sum_{i=0}^{M-1} w_i (x(n - \Delta - i) + \eta(n - \Delta - i)) \right] \quad (3.38)$$

$$\rightarrow \min_{\Delta \in \mathbb{N}} \mathbb{E} \left[(u(n) + 0.5u(n-2)) \sum_{i=0}^{M-1} w_i (\eta(n - \Delta - i)) \right] \quad (3.39)$$

$$\rightarrow \min_{\Delta \in \mathbb{N}} \mathbb{E} \left[(u(n) + 0.5u(n-2)) \sum_{i=0}^{M-1} w_i (u(n - \Delta - i) + 0.5u(n - 2 - \Delta - i)) \right] \quad (3.40)$$

$$\rightarrow 0, \quad \Delta > 2 \quad (3.41)$$

Since $u(n)$ is identically and **independently** distributed white noise:

$$\mathbb{E} \left[u(n) u(n-j) \right] = 0, \quad \forall j \neq 0 \quad (3.42)$$

therefore the expectation in (3.40) is zero and thus minimised for $\Delta > 2$, since the terms are non time-overlapping. This is an expected result, since the colored noise signal $\eta(n)$ is a second order MA process.

The theoretical optimal delay range, $\Delta > 2$ is also verified empirically. In figure 3.6 the clean signal $x(n)$ against, $s(n)$ and filter output $\hat{x}(n)$ are illustrated for different delay Δ values. Moreover, the MPSE as a function of Δ is also plotted in figure 3.7, verifying the improved performance $\Delta > 2$. All experiments are conducted using a fixed model order $M = 5$ LMS filter.

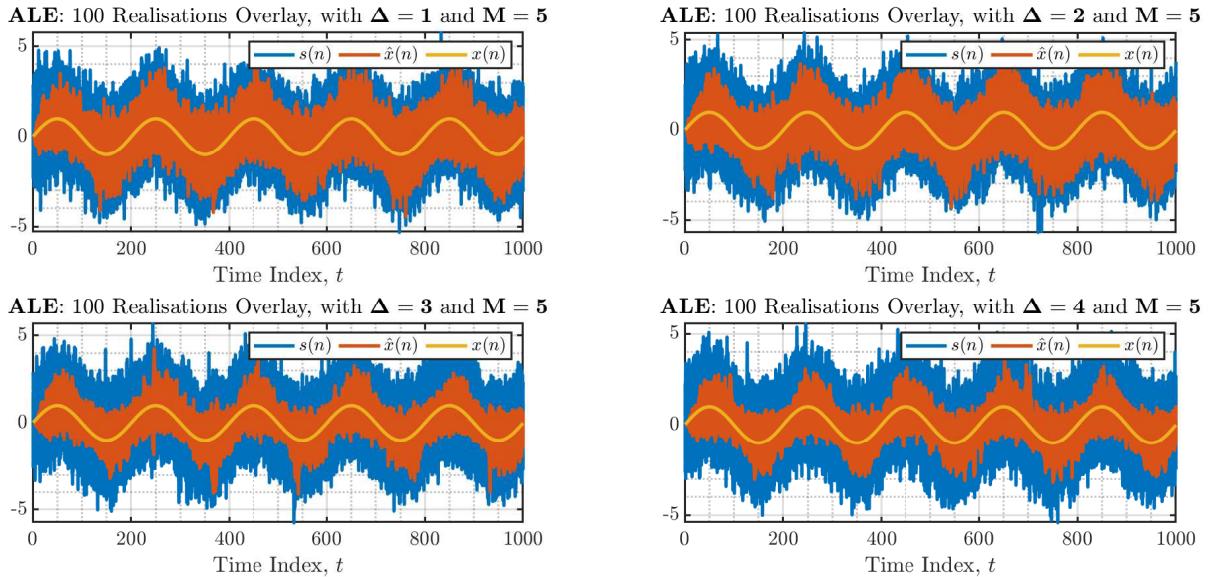
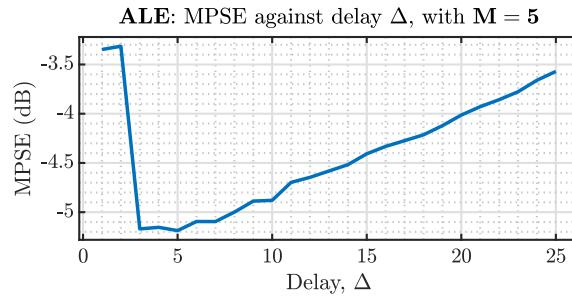


Figure 3.6: ALE: overlay plots for various Δ delays, for fixed $M = 5$.

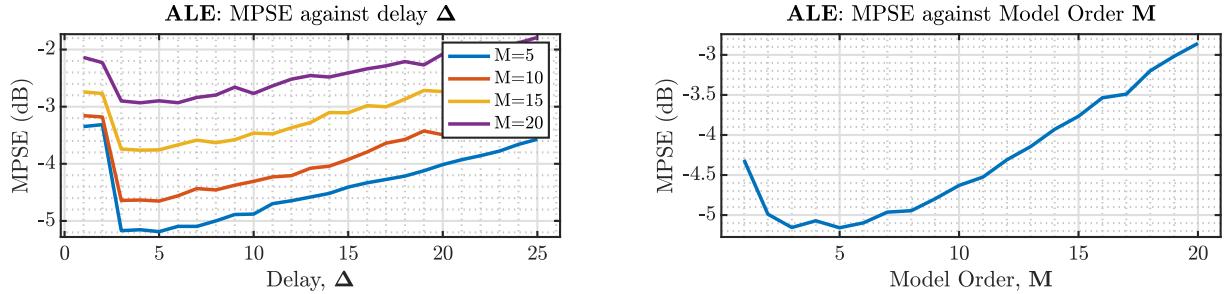
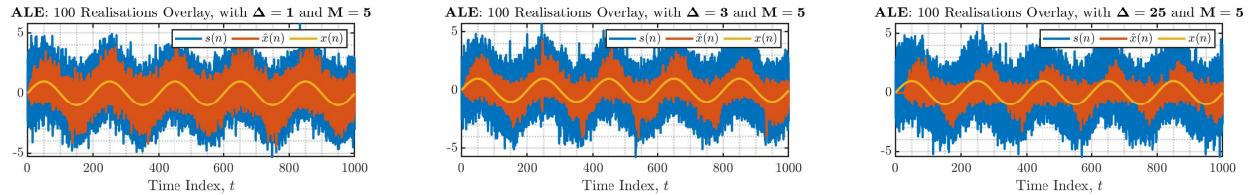
- b) The experiments are repeated varying now both the delay parameter, Δ , and the model order, M , obtaining figures 3.8, 3.9. We notice that the mean squared prediction error (MPSE) is minimised for the hyperparameters pair $(\Delta, M) = (3, 6)$.

Over-modelling (large M) results in excess degrees of freedom that increase computational complexity and over-fit noise, degrading model performance. For model order $M = 6$ MPSE is minimised, while the model computational load is still not prohibitive.

In the previous part we showed theoretically that for $\Delta > 2$ the noise and the filter output are uncorrelated thus MSE is minimised. Nonetheless, the second term in (3.34) was ignored. The impact of this term on


 Figure 3.7: ALE: MPSE against Δ , for fixed $M = 5$.

the MPSE is illustrated in figure 3.9, where very large Δ (i.e $\Delta = 25$) inevitably cause a time-shift between the filter output $\hat{x}(n)$ and the true sine wave $x(n)$. Hence, $\Delta = 3$ is the optimal parameter, minimising delay effects between $x(n)$ and $\hat{x}(n)$, as well as guaranteeing uncorrelation between $\hat{x}(n)$ and $\eta(n)$.


 Figure 3.8: ALE: MPSE against delay Δ and model order M .

 Figure 3.9: ALE: model deterioration for large delays Δ .

- c) We compare the performance of the Adaptive Noise Cancellation (ANC) configuration to the Adaptive Line Enhancer (ALE) configuration, used in previous parts. The colored noise $\varepsilon(n)$ is used as input to the LMS filter to perform ANC, such that:

$$\varepsilon(n) = 0.9\eta(n) + 0.05 \quad (3.43)$$

and as a result $\varepsilon(n)$ is the secondary noise signal, correlated to the $\eta(n)$, primary noise signal. Note that the relationship between the two signals is unknown to the ANC algorithm.

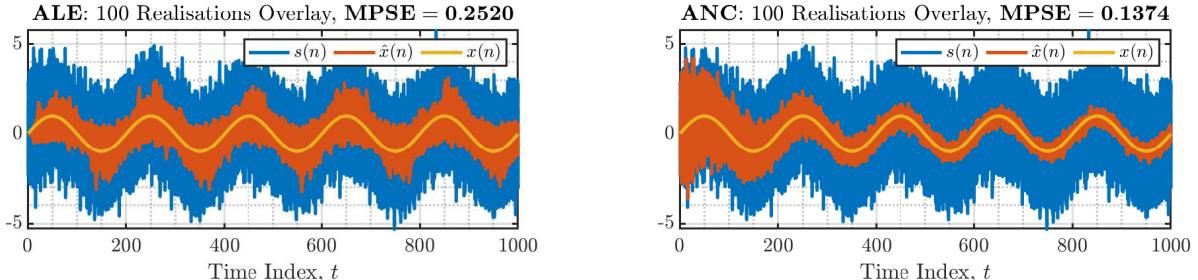


Figure 3.10: ALE vs ANC: overlay plots and mean prediction squared error.

In figure 3.10 the overlay plots for 100 realisations of the process $x(n)$ are provided, along with the

denoised versions of both the ALE and ANC configurations. ANC performs overall better, with MPSE = 0.1374, than the ALE configuration, which scores MPSE = 0.2520. Nonetheless, we highlight the fact that the ANC algorithm does poorly in the first timesteps, but for $t > 400$, it tracks the true signal $x(n)$ much better. An ensemble of realisations of the process is also simulated and its mean are illustrated in figure 3.11.

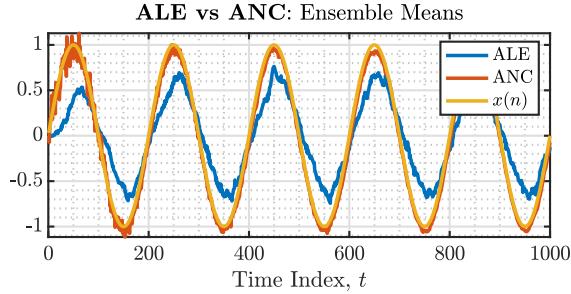


Figure 3.11: ALE vs ANC: ensemble mean comparison.

- d) Let a synthetic reference input, $\varepsilon(n)$, composed of a sinusoid of 50 Hz corrupted by white Gaussian noise. The ANC configuration is used with inputs $\varepsilon(n)$ and the P0z EEG time-series, in order to remove the strong 50Hz frequency component due to power-line interference (mains).

For illustration purposes spectrograms are plotted using a rectangular window of length $L = 4096$ and 80% overlap. The obtained spectrograms are provided in figures 3.12 and 3.13. As expected the original P0z signal has a strong 50 Hz frequency component.

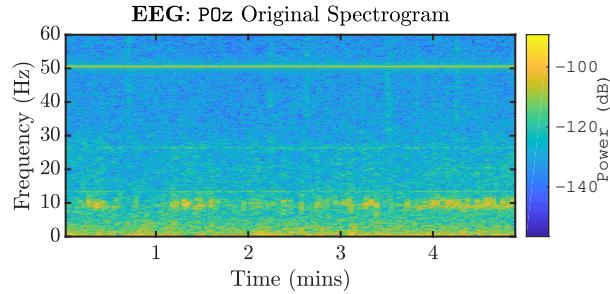


Figure 3.12: EEG: original, reference spectrogram.

The LMS filter order M and the step-size μ is varied, and the impact on the spectrogram is shown in figure 3.13.

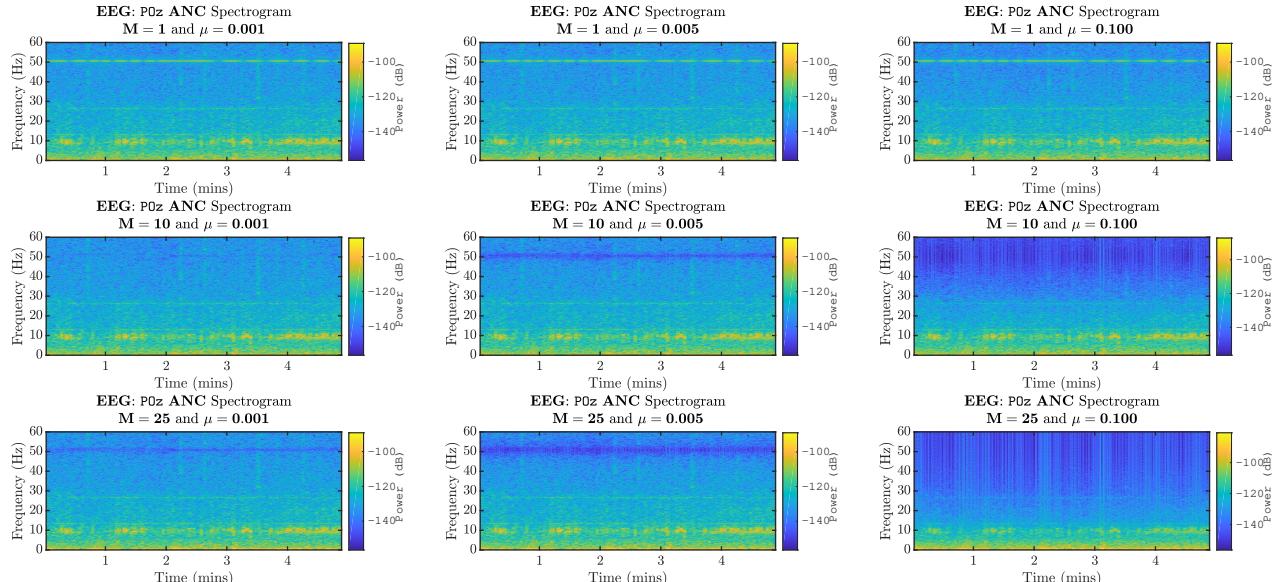


Figure 3.13: EEG: ANC denoised spectrogram for different model order, M , and step-sizes, μ .

Large step-sizes (i.e $\mu = 0.1$) affect significantly the spectral components around 50 Hz, degrading ANC performance. On the other hand, small step-sizes (i.e $\mu = 0.001$) take more time to reach steady-state, however provide successful denoising, without disrupting the frequencies close to 50 Hz.

On the other hand, under-modelling (i.e $M = 1$) leads to poor noise cancellation, since the 50 Hz power-line interface component has not been attenuated. However, over-modelling (i.e $M = 25$) degrades quality of neighbour frequencies, while a medium size model, such as $M = 10$, achieves satisfying performance, by eliminating the 50 Hz component of interest, without affecting any other components.

Overall, an ANC configuration with $(M, \mu) = (10, 0.001)$ is selected. The corresponding denoised periodogram is also provided at figure 3.14.

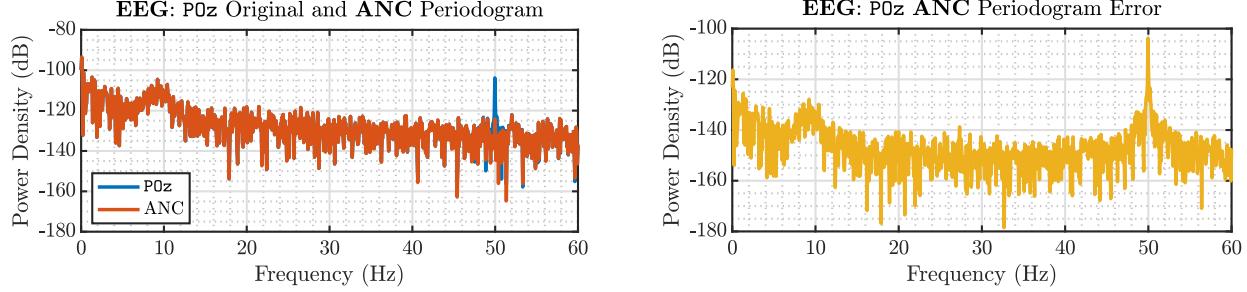


Figure 3.14: EEG: periodograms of original and denoised signals.

Widely Linear Filtering and Adaptive Spectrum Estimation

4.1 Complex LMS and Widely Linear Modelling

- a) A Wide Linear Moving Average order 1, WLMA(1), process $y(n)$ is generated, whose real and imag parts are provided in figure 4.1. As expected, the process is not circular, and we expect simple Complex Least Mean Squared (CLMS) algorithm to fail to model it.

The mean squared prediction error curve in figure 4.1 agrees with our assumption, though the Augmented Complex Least Mean Squared (ACLMS) algorithm successfully learns the process parameters. In more detail, the ACLMS achieves an $MPSE = -300dB$ in steady-state, while CLMS saturates at $MPSE = 7.6$. This is the consequence of the inability of the CLMS to adapt to non-circular signals, due to its limited capacity.

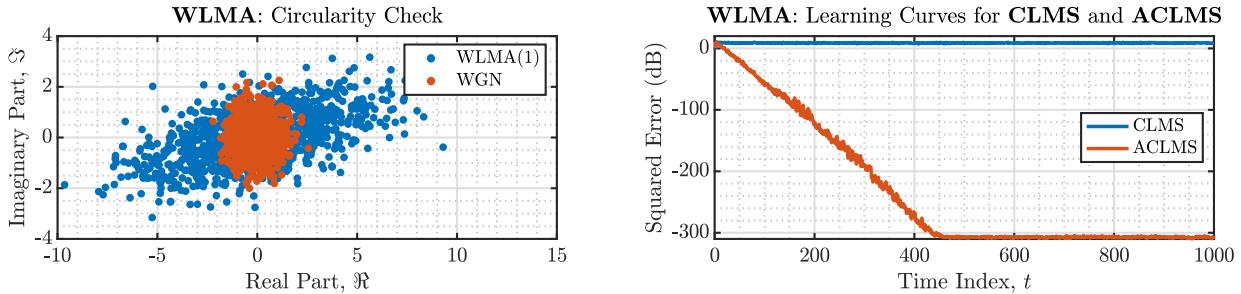


Figure 4.1: CLMS vs ACLMS: non-circularity and (A)CLMS learning curves.

- b) Figures 4.2, 4.3, 4.4 illustrate the circularity plots (real-img scatter plots) along with the learning curves of the ACLMS and CLMS algorithms for different model orders, on the wind-dataset.

Despite its non-obviously symmetric shape, the low regime wind data has lowest circularity coefficient, $\rho = 0.159$, while the medium and high regimes have $\rho = 0.454$ and $\rho = 0.624$, respectively. A complex-valued random variable is said to be circular if its probability distribution is not dependent on the angle, that is, the distribution is rotationally invariant. In other words the variable's probability distribution function should only depend on the Euclidean distance from the origin in the complex domain. By definition, the higher the circularity coefficient, ρ , the less circular the data is.

Hence, as shown in the previous part, ACLMS algorithm outperforms CLMS on non-circular data. This is verified again on the wind-dataset, where the ACLMS has lower mean squared prediction error (MPSE) for any wind regime. More interesting is the fact that at high regime data, the least circular, the ACLMS has a larger margin in performance ($0.1dB$). Lastly, we observe that the MPSE is minimised for small model orders, $M \in [3, 6]$, since over-modelling leads to over-fitting the noise and fail to generalise, despite the excess degrees of freedom. For larger model order, $M > 10$, we observe that the ACLMS performs worse than the CLMS, due to its extra capacity (more parameters), which lead to overfitting.

- c) Complex voltages are synthetically generated for various magnitude, angle combinations leading to balanced and unbalanced configurations, while their illustrations in the complex plane are provided in figures 4.5, 4.6. Overall, we notice that balanced systems have a circular shape, while unbalanced don't. This is also collaborated by the fact that the balanced system has circularity coefficient $\rho = 0$, while the unbalanced system has a high circularity coefficient, $\rho = 0.757$. Finally, the impact of the angle and magnitude distortions, $\Delta_b \neq \Delta_c \neq 0$ and $V_a \neq V_b \neq V_c$, respectively, is depicted in figure 4.6.

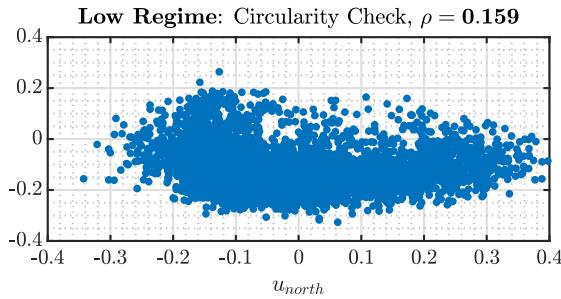


Figure 4.2: Low Regime:circularity plot and (A)CLMS learning curves.

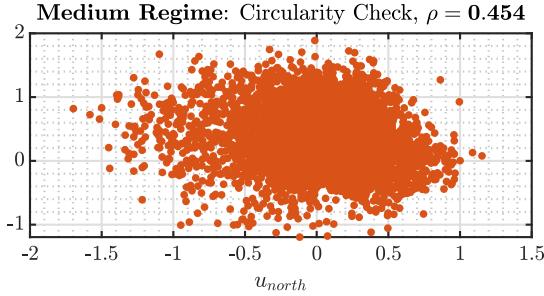


Figure 4.3: Medium Regime:circularity plot and (A)CLMS learning curves.

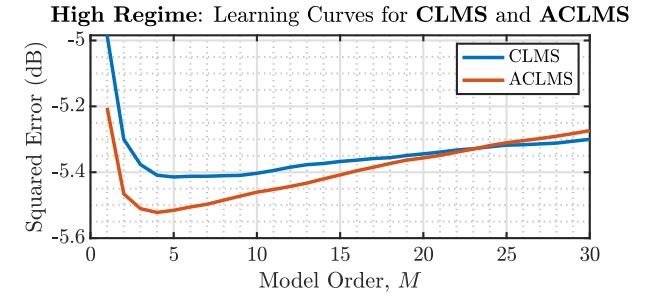
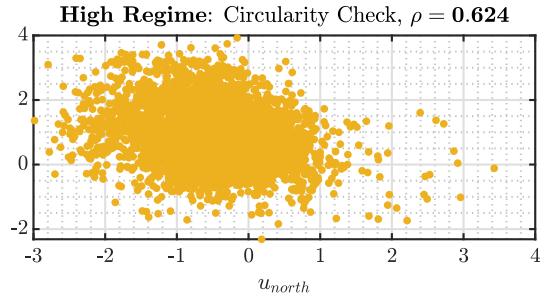


Figure 4.4: High Regime:circularity plot and (A)CLMS learning curves.

d) Balanced complex $\alpha - \beta$ voltages satisfy:

$$u(n) = \sqrt{\frac{3}{2}} V e^{j(2\pi \frac{f_o}{f_s} n + \varphi)} \quad (4.1)$$

For time index $n+1$:

$$u(n+1) = \sqrt{\frac{3}{2}} V e^{j(2\pi \frac{f_o}{f_s} (n+1) + \varphi)} \quad (4.2)$$

$$= \sqrt{\frac{3}{2}} V e^{j(2\pi \frac{f_o}{f_s} n + \varphi)} e^{j2\pi \frac{f_o}{f_s}} \quad (4.3)$$

$$= u(n) e^{j2\pi \frac{f_o}{f_s}} \quad (4.4)$$

Using the strictly linear autoregressive model of order 1, equation, satisfying:

$$u(n+1) = u(n) h^*(n) \quad (4.5)$$

we express the complex exponential in (4.4) as a function of the model parameter $h(n)$:

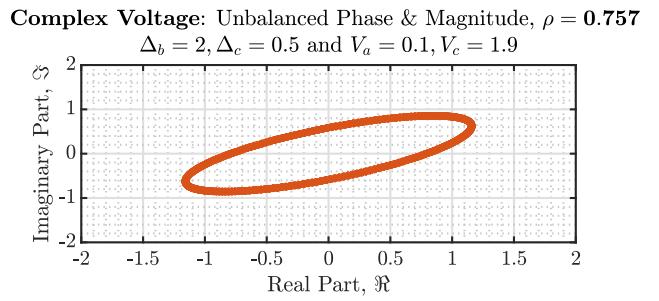
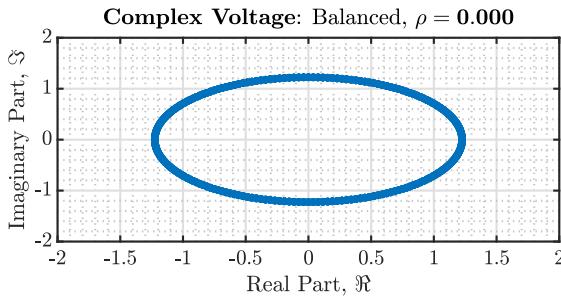


Figure 4.5: Complex Voltage: balanced and unbalanced voltages.

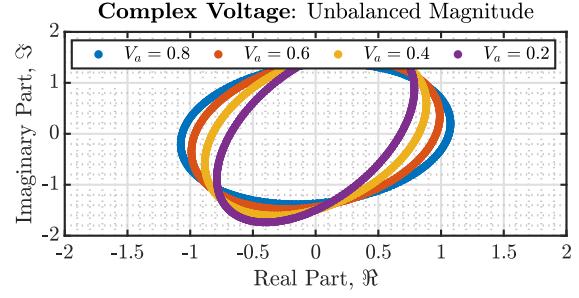
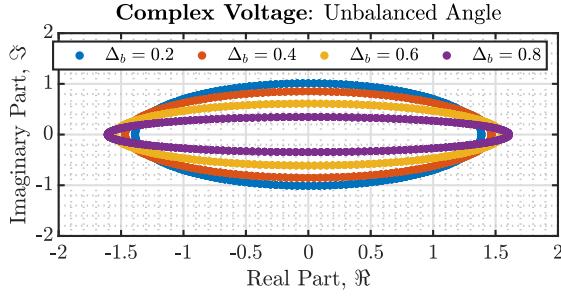


Figure 4.6: Complex Voltage: unbalanced voltages angle and magnitude distortions.

$$e^{j2\pi \frac{f_o}{f_s}} = h^*(n) \quad (4.6)$$

$$e^{-j2\pi \frac{f_o}{f_s}} = h(n) \quad (4.7)$$

$$= |h(n)| e^{j(\arctan(\frac{\Im\{h(n)\}}{\Re\{h(n)\}}))} \quad (4.8)$$

Two complex numbers are equal if and only of both their magnitudes and their angles are equal, therefore we set the angles to be equal, obtaining:

$$2\pi \frac{f_o}{f_s} = \arctan\left(\frac{\Im\{h(n)\}}{\Re\{h(n)\}}\right) \quad (4.9)$$

Solving for f_o concludes the proof:

$$f_o = \frac{f_s}{2\pi} \arctan\left(\frac{\Im\{h(n)\}}{\Re\{h(n)\}}\right) \quad (4.10)$$

The unbalanced system satisfies:

$$u(n) = A(n)e^{j(2\pi \frac{f_o}{f_s} n + \varphi)} + B(n)e^{-j(2\pi \frac{f_o}{f_s} n + \varphi)} \quad (4.11)$$

Using the widely linear autoregressive model of order 1, equation, satisfying:

$$u(n+1) = h^*(n)u(n) + g^*(n)u^*(n) \quad (4.12)$$

we substitute (4.11) terms:

$$\begin{aligned} u(n+1) &= h^*(n) \left[A(n)e^{j(2\pi \frac{f_o}{f_s} n + \varphi)} + B(n)e^{-j(2\pi \frac{f_o}{f_s} n + \varphi)} \right] + \\ &\quad g^*(n) \left[A^*(n)e^{-j(2\pi \frac{f_o}{f_s} n + \varphi)} + B^*(n)e^{j(2\pi \frac{f_o}{f_s} n + \varphi)} \right] \end{aligned} \quad (4.13)$$

For time index $n+1$, the complex voltage satisfies:

$$u(n+1) = A(n+1)e^{j(2\pi \frac{f_o}{f_s} (n+1) + \varphi)} + B(n+1)e^{-j(2\pi \frac{f_o}{f_s} (n+1) + \varphi)} \quad (4.14)$$

Equating (4.13) and (4.14), collecting common exponential terms:

$$A(n+1)e^{j(2\pi \frac{f_o}{f_s} (n+1) + \varphi)} = \left[h^*(n)A(n) + g^*(n)B^*(n) \right] e^{j(2\pi \frac{f_o}{f_s} n + \varphi)} \quad (4.15)$$

$$B(n+1)e^{-j(2\pi \frac{f_o}{f_s} (n+1) + \varphi)} = \left[h^*(n)B(n) + g^*(n)A^*(n) \right] e^{-j(2\pi \frac{f_o}{f_s} n + \varphi)} \quad (4.16)$$

Assuming that the amplitude change over time is negligible, or equivalently $A(n+1) \approx A(n)$ and $B(n+1) \approx B(n)$, the equations simplify as:

$$e^{j2\pi \frac{f_o}{f_s}} = \frac{h^*(n)A(n) + g^*(n)B^*(n)}{A(n+1)} \approx h^*(n) + g^*(n) \frac{B^*(n)}{A(n)} \quad (4.17)$$

$$e^{-j2\pi \frac{f_o}{f_s}} = \frac{h^*(n)B(n) + g^*(n)A^*(n)}{B(n+1)} \approx h^*(n) + g^*(n) \frac{A^*(n)}{B(n)} \quad (4.18)$$

Note that (4.17) is the complex conjugate of (4.18), we reach:

$$h^*(n) + g^*(n) \frac{B^*(n)}{A(n)} = h(n) + g(n) \frac{A(n)}{B^*(n)} \quad (4.19)$$

Multiplying with $\frac{B^*(n)}{A(n)}$ both sides:

$$\left(h^*(n) - h(n) \right) \frac{B^*(n)}{A(n)} + g^*(n) \left(\frac{B^*(n)}{A(n)} \right)^2 - g(n) = 0 \quad (4.20)$$

We note that (4.20) is quadratic in $\frac{B^*(n)}{A(n)}$, hence solving it we obtain:

$$\frac{B^*(n)}{A(n)} = \frac{-\left(h^*(n) - h(n) \right) \pm \sqrt{\left(h^*(n) - h(n) \right)^2 + 4g^*(n)g(n)}}{2g^*(n)} \quad (4.21)$$

$$= \frac{2\Im\{h(n)\}j \pm j\sqrt{-4\Im\{h(n)\}^2 + 4|g(n)|^2}}{2g^*(n)} \quad (4.22)$$

$$= \frac{\Im\{h(n)\}j \pm j\sqrt{\Im\{h(n)\}^2 - |g(n)|^2}}{g^*(n)} \quad (4.23)$$

Substitution in (4.17), yields:

$$e^{j2\pi\frac{f_0}{f_s}} = h^*(n) + \Im\{h(n)\}j \pm j\sqrt{\Im\{h(n)\}^2 - |g(n)|^2} \quad (4.24)$$

$$= \Re\{h(n)\} \pm j\sqrt{\Im\{h(n)\}^2 - |g(n)|^2} \quad (4.25)$$

$$(4.26)$$

Keeping one solution (+ sign), since $f_s \gg f_0 > 0$:

$$e^{j2\pi\frac{f_0}{f_s}} = \Re\{h(n)\} + j\sqrt{\Im\{h(n)\}^2 - |g(n)|^2} \quad (4.27)$$

$$= \rho e^{j\left(\frac{\sqrt{\Im\{h(n)\}^2 - |g(n)|^2}}{\Re\{h(n)\}}\right)} \quad (4.28)$$

where $\rho > 0$. Set the angles to be equal and solving for f_0 we complete the proof:

$$f_0 = \frac{f_s}{2\pi} \arctan\left(\frac{\sqrt{\Im\{h(n)\}^2 - |g(n)|^2}}{\Re\{h(n)\}}\right) \quad (4.29)$$

- e) First order CLMS and ACLMS filters are trained on both the balanced and unbalanced (magnitude and angle) complex voltage synthetic data. The derived formulae (4.10) and (4.29) are used along with the learnt filter parameters, h_{CLMS} , h_{ACLMS} & g_{ACLMS} , while the frequency estimates over time and the error curves are provided in figures 4.7, 4.8 for the balanced and unbalanced complex voltage, respectively.

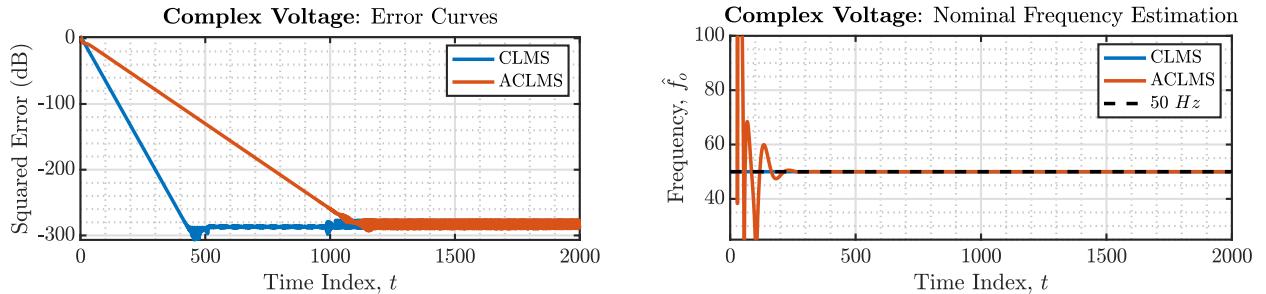


Figure 4.7: Balanced Complex Voltage: mean squared prediction error (MPSE) and frequency estimation.

In case of the balanced grid, the CLMS filter excels, since it both converges faster to the nominal frequency $f_0 = 50\text{Hz}$, without overshooting (transient behaviour). On the other hand, the ACLMS oscillates for the first 300 timesteps and finally converges to the true frequency after that. This is an expected result, since in figure 4.5 we showed that the balanced system has a circular complex voltage, which can be sufficiently modelled by a CLMS filter. The extra degrees of ACLMS freedom, put a burden on the model, which needs more time to converge.

Figure 4.8 depicts the mean squared prediction error and frequency estimates for the unbalanced system, with $(V_a, V_b, V_c) = (0.1, 1.0, 1.9)$ and $(\Delta_b, \Delta_c) = (2.0, 0.5)$. In this scenario, the ACLMS converges to the true nominal frequency value after 450 timesteps, though the CLMS filter oscillates around $\hat{f}_o = 37\text{Hz}$ and never adapts to the true f_0 . Unsurprisingly, CLMS does not have the capacity to model non-circular distributions and hence fails to model unbalanced configurations ($\rho = 0.757$).

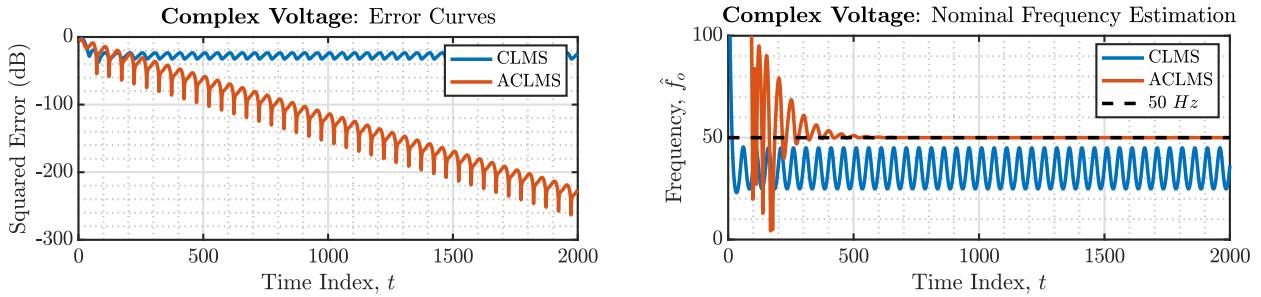


Figure 4.8: Unbalanced Complex Voltage: mean squared prediction error (MPSE) and frequency estimation.

4.2 Adaptive AR Model Based Time-Frequency Estimation

- a) In figure 4.9 the non-stationary frequency, $f(n)$, and the corresponding phase, $\phi(n)$, time-series are provided. Clearly, the frequency is time-variant, hence a model capable of capturing non-stationarity is required.

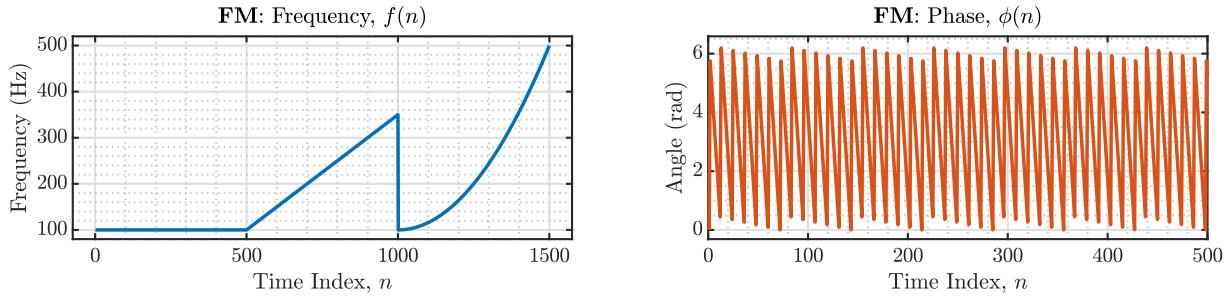


Figure 4.9: FM: non-stationary frequency and phase time-series.

As expected, the static autoregressive order 1 model, AR(1), in figure 4.10, fails to adapt to $f(n)$ over time, since a single peak is detected. Increasing model order does not improve performance, since the static approach is incapable of dealing with non-stationary series, regardless the capacity of the model.

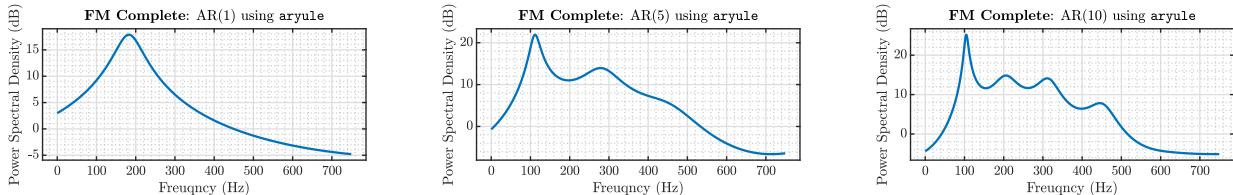


Figure 4.10: FM: power spectrum and AR(p) models.

The frequency $f(n)$ is a branch function, a constant, a linear and a quadratic function, for different values of n . Splitting the signal into three series of segment length $N_{segment} = 500$, we confirm that only the first segment (constant frequency $f = 100\text{Hz}$ over time) can be adequately modelled, while the rest are failed due to their non-stationary nature. Figure 4.10 illustrates the power spectral density estimates for the three segments.

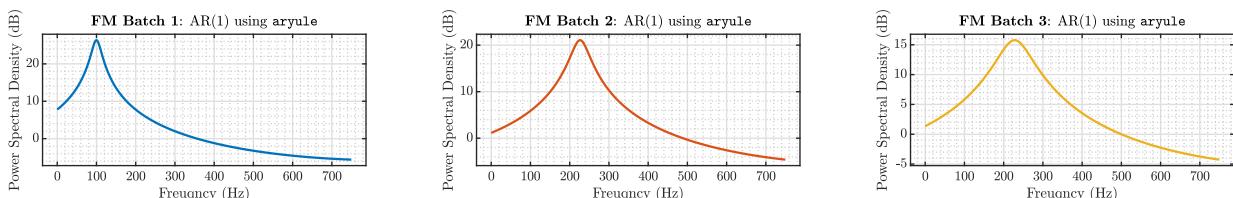


Figure 4.11: FM: power spectrum of segments.

- b) Having considered the deficiencies of the static AR models, a dynamic approach is now taken, using the CLMS algorithm, since the signal is complex. Comparing figure 4.12, the time-frequency spectrum

plots, with the time-series $f(n)$ in figure 4.9, we verify that the dynamic CLMS AR(1) model enables the modelling of non-stationary processes.

The step-size μ of the CLMS filter is also varied, where for small values (i.e $\mu = 0.001$) the filter does not converge in-time to the correct frequencies, while large μ values (i.e $\mu = 0.1$) lead to oscillations around the target value. This reflect once again the trade-off between convergence rate and steady-state error. The wider bounds in the time-frequency plots (i.e for $\mu = 0.1$) the less certain the estimate is.

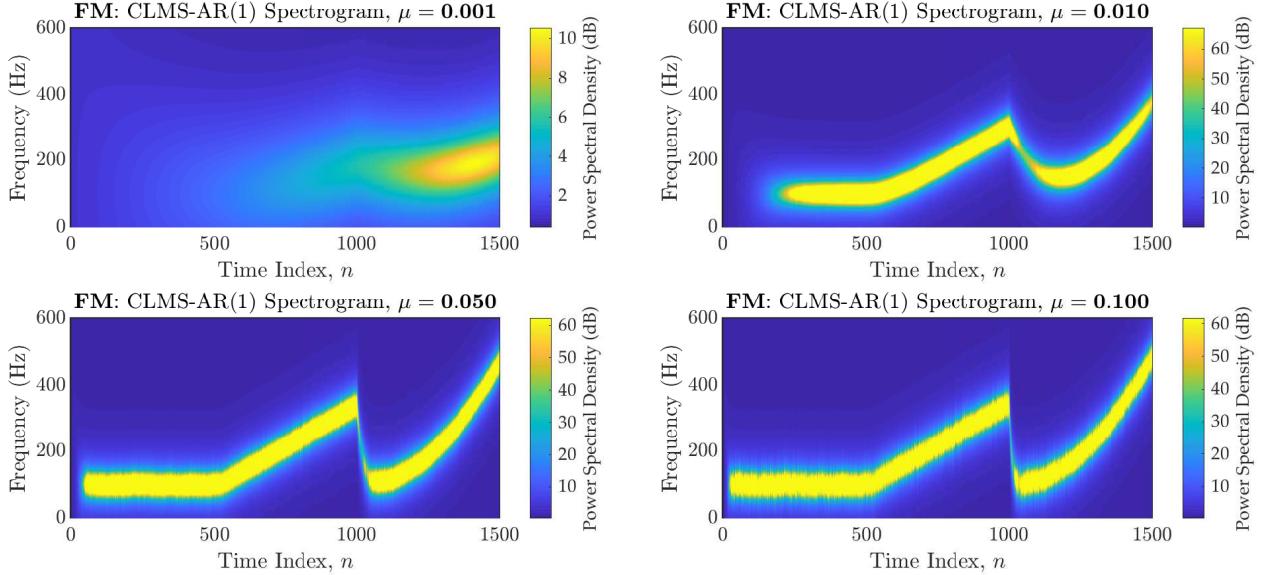


Figure 4.12: FM: CLMS time-frequency plots.

4.3 A Real Time Spectrum Analyser Using Least Mean Square

a) Let the L_2 -norm of the prediction error being the objective function \mathcal{J} :

$$\mathcal{J}(\mathbf{w}) = \|\mathbf{y} - \hat{\mathbf{y}}\|^2 = \|\mathbf{y} - \mathbf{F}\mathbf{w}\|^2 = (\mathbf{y} - \mathbf{F}\mathbf{w})^H(\mathbf{y} - \mathbf{F}\mathbf{w}) \quad (4.30)$$

In order to minimise the objective function \mathcal{J} with respect to the parameters (weights) \mathbf{w} , the first order condition is:

$$\frac{\partial \mathcal{J}(\mathbf{w})}{\partial \mathbf{w}} \Big|_{\mathbf{w}=\mathbf{w}_*} = 0 \quad (4.31)$$

Substituting from (4.30):

$$\frac{\partial}{\partial \mathbf{w}} \left(\mathbf{y}^H \mathbf{y} - \mathbf{y}^H \mathbf{F} \mathbf{w} - \mathbf{w}^H \mathbf{F}^H \mathbf{y} + \mathbf{w}^H \mathbf{F}^H \mathbf{F} \mathbf{w} \right) \Big|_{\mathbf{w}=\mathbf{w}_*} = 0 \quad (4.32)$$

$$0 - \mathbf{F}^H \mathbf{y} - \mathbf{F}^H \mathbf{y} + 2\mathbf{F}^H \mathbf{F} \mathbf{w}_* = 0 \quad (4.33)$$

Assuming that $\mathbf{F}^H \mathbf{F}$ is invertible (semi-positive as covariance matrix), we solve for \mathbf{w}_* , concluding the proof:

$$\mathbf{w}_* = \left(\mathbf{F}^H \mathbf{F} \right)^{-1} \mathbf{F}^H \mathbf{y} \quad (4.34)$$

The Inverse Discrete Fourier Transform (IDFT) of a signal $x(n)$ is given by:

$$\hat{x}(n) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} X(k) e^{j \frac{2\pi}{N} nk} = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} X(k) F_N^{nk} \quad (4.35)$$

where $F_N = e^{j \frac{2\pi}{N}}$. If we define:

$$f_n^H = \frac{1}{\sqrt{N}} [1, F_N^n, F_N^{2n}, \dots, F_N^{n(N-1)}] \quad (4.36)$$

Arranging the sample estimates in a vector, we obtain:

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{x}(0) \\ \hat{x}(1) \\ \vdots \\ \hat{x}(N-1) \end{bmatrix} = \begin{bmatrix} f_0^H \mathbf{X} \\ f_1^H \mathbf{X} \\ \vdots \\ f_{N-1}^H \mathbf{X} \end{bmatrix} = \mathbf{F} \mathbf{X} \quad (4.37)$$

where:

$$\mathbf{F} = \begin{bmatrix} f_0^H \\ f_1^H \\ \vdots \\ f_{N-1}^H \end{bmatrix} = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & F_N & F_N^2 & \cdots & F_N^{N-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & F_N^{N-1} & F_N^{2(N-1)} & \cdots & F_N^{(N-1)^2} \end{bmatrix} \quad (4.38)$$

Hence from (4.37), we conclude that IDFT is a linear transformation, where \mathbf{F} the transformation matrix comprises of N harmonically related sinusoids, with mutually orthonormal columns¹. Then the Fourier coefficients, \mathbf{X} , are given by (4.34):

$$\mathbf{X} = (\mathbf{F}^H \mathbf{F})^{-1} \mathbf{F}^H \hat{\mathbf{x}} = \mathbf{F}^H \hat{\mathbf{x}} \quad (4.39)$$

where the fact that \mathbf{F} is unitary is used². Therefore, since (4.34) is the optimal least mean squares solution, minimising the squared error and the Fourier coefficients, \mathbf{w} , follow (4.34), then Inverse Discrete Fourier Transform, $\hat{x}(n)$, is a linear approximation of the original signal $x(n)$, minimising the squared error:

$$\min_{\mathbf{X}} \| \mathbf{x} - \hat{\mathbf{x}} \|^2 \quad (4.40)$$

- b) According to equation (4.39), the Fourier coefficients, \mathbf{X} , are a linear combination of the columns of the transformation matrix \mathbf{F} , whose columns are orthonormal:

$$f_k^H f_l = \frac{1}{N} \sum_{k=0}^{N-1} e^{j \frac{2\pi}{N} (k-l)} = \begin{cases} 1 & \text{if } k = l \\ 0 & \text{otherwise} \end{cases} \quad (4.41)$$

Therefore DFT operation can be seen as a projection of a vector, \mathbf{x} , in time domain to the \mathbf{F} matrix subspace, spanned by their orthonormal columns, representing N harmonically related sinusoids, where each sinusoid is a multiple of the frequency $\frac{f_s}{N}$, with f_s the sampling frequency.

¹Note that the $\frac{1}{\sqrt{N}}$ term is introduced to deal with unit-length columns.

²See Problem & Answer Sets for the proof.

- c) The K -points DFT-CLMS method is applied to the frequency modulated (FM) non-stationary signal, $f(n)$. Figure 4.13 illustrates the time-frequency diagram obtained. The algorithm adapts perfectly to the behaviour of the frequency signal, $f(n)$, for $n < 500$, where the frequency is constant over time, while for larger values we notice that the trends (both linear and quadratic) are generally captured. Surprisingly though, once a strong frequency component is picked up by the DFT-CLMS filter, this component is not updated, hence the diagram has Fourier coefficients superimposed over time. CLMS is a gradient algorithm, which updates weights (DFT coefficients) towards the mean squared error descent direction. The coefficients are not block-based calculated, but in an adaptive manner, hence error back-propagation is slow, causing this lasting effect. Moreover, the K -points DFT-CLMS filter has K dimensional weights, where $K = 2048$, thus the curse of dimensionality prevents error gradients to propagate back and update the weights.

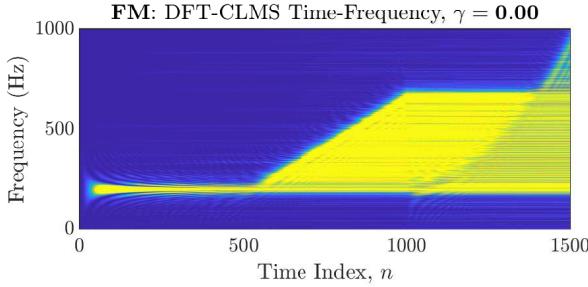


Figure 4.13: FM: DFT-CLMS time-frequency plot with $\gamma = 0$ (unbiased).

Regularisation, such as adding a leakage coefficient γ , similar to the Leaky LMS variant, enables accurate time-frequency modelling, as shown in figure 4.14. This is achieved thanks to the forget mechanism that the Leaky CLMS algorithm provides:

$$\mathbf{w}(n+1) = (1 - \gamma\mu)\mathbf{w}(n) + \mu e^*(n)\mathbf{x}(n) \quad (4.42)$$

where the greater γ values allow to ignore, forget previous timestep's weights $\mathbf{w}(n)$.

Note that for very small values of the leakage coefficient (i.e $\gamma = 0.01$) the Fourier coefficient superimposing effect is still visible. However, larger values of γ (i.e $\gamma = 0.05, 0.1$) introduce some bias, but obtain correct modelling. Finally, bias for larger values of γ (i.e $\gamma \geq 0.5$) lead to inaccurate results.

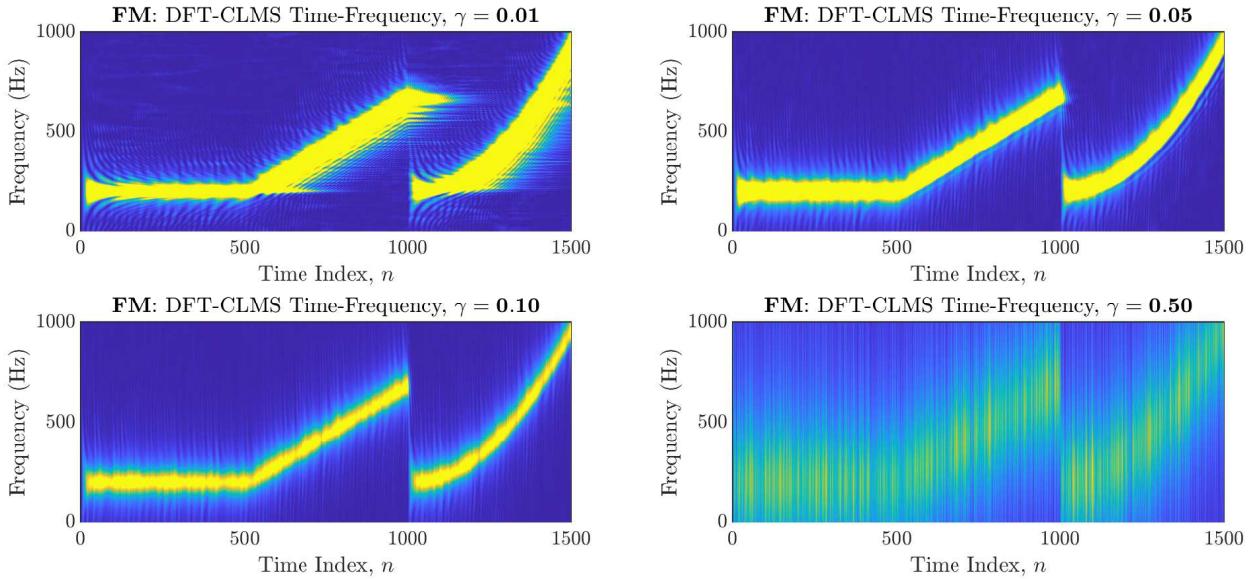


Figure 4.14: FM: DFT-CLMS time-frequency plot with $\gamma > 0$ (biased).

- d) The K -points DFT-CLMS algorithm is applied to the EEG P0z data, of length $N = 1200$. The time-frequency diagrams for different γ values are obtained and illustrated in figure 4.15. The Leaky CLMS does not perform any better than the standard CLMS algorithm, due to the stationary nature of the signal under study.

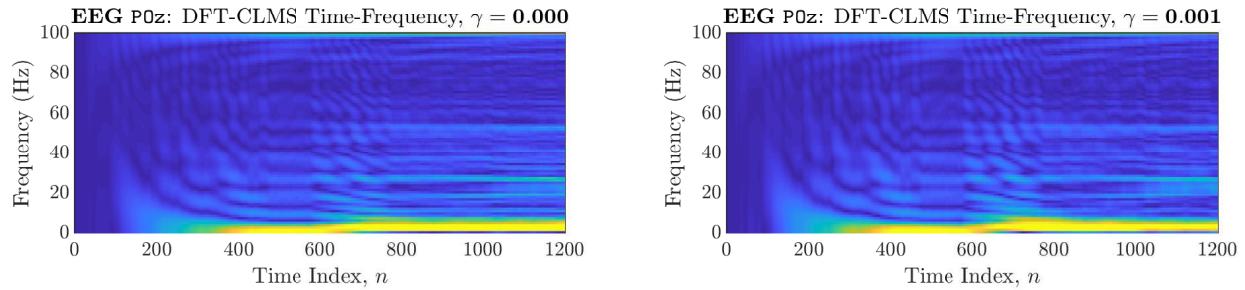


Figure 4.15: EEG P0z: DFT-CLMS time-frequency plots.

The strong 50Hz component is visible in both implementations and so are the first two harmonics of SSEVP, at frequencies $f_1 = 13\text{Hz}$ and $f_2 = 26\text{Hz}$, respectively. The third harmonic is not distinguishable though, with any of the two algorithms.