Imperial College London

Spectrum Estimation & Adaptive Signal Processing

Author: Vijay S. Gami
CID: 00739377

March 2016

Contents

1 Non Parametric Spectrum Estimation

1.1 Discrete Fourier Transform Basics

1.1.a Fourier Transform (FT) and Discrete Time Fourier Transform (DTFT):

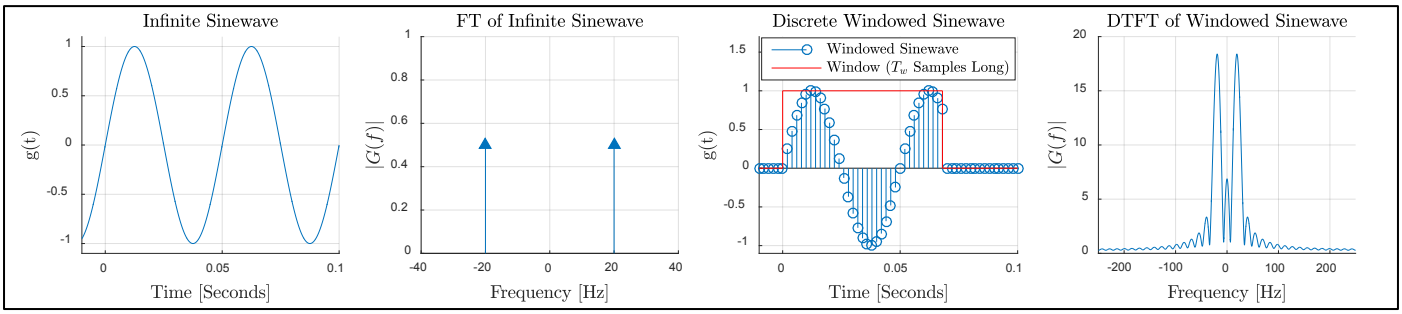

Figure 1: FT of an $20Hz$ infinite continuous sinusoid, and DTFT of windowed discrete sinusoid. In this example, $f_s = 500\ Hz, T_w = 35\ samples$

Figure 1 shows the FT of a $20\ Hz$ continuous infinite duration sinewave and the DTFT ($X(\omega) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n}$) of a discrete windowed sinewave. Applying a rectangular window in time corresponds to convolution with a Sinc function $\left(X_{sinc}(\omega) = sin(\omega T_w/2)/sin(\omega/2)\right)$ in the frequency domain. Convolution of any function with an impulse results in the same function just scaled and shifted to the location of the impulse. This is why we see two Sinc functions in the spectra of the discrete windowed sinewave. A Sinc has a peak height of T_w, and a lobe width of $f = 2f_s/T_w$ $(in\ Hz)$, where T_w is the window length in samples and f_s is the sampling frequency. In this particular example $T_w = 35$ and $f_s = 500\ Hz$, thus we expect a peak Sinc amplitude of $0.5 \times 35 = 17.5$. Here, the 0.5 factor comes from the amplitude of the impulse which is 0.5. The DTFT shown in Figure 1 differs slightly from this since there are two Sinc functions which overlap and interfere with each other, and as a result, the amplitude is higher than expected. Note that the DTFT is periodic with a period of $f = f_s$ (or normalised frequency, $\omega = 2\pi$), but only one period is shown in Figure 1 (right).

1.1.b Discrete Fourier Transform (DFT):

Figure 2 below shows the K point DFT of a $20\ Hz$ sinewave of length N with a sample frequency $f_s = 1000\ Hz$. For the plot on the right $K > N$ which means the signal is zero padded which $K - N$ zeros. This increases the signal length and hence frequency bin resolution, but essentially introduces rectangular windowing. This means the estimated spectrum equals the true spectrum convolved with a Sinc function which is why this plot resembles the rightmost plot in Figure 1. This similarity is expected since the DFT is just a sampled version of the DTFT. However since f_s and T_w are now changed, there are some differences; now the peak height is approximately $T_w/2 = 50$ and the mainlobe width is $2f_s/T_w = 20\ Hz$.

The DFT plot shown in the left Figure 2 has been scaled by $1/K$ since this makes the DFT more relatable to the FT and this is why it matches the FT plot in Figure 1. When $K = 100$, the DFT is an ideal Kronecker delta, and this can be explained by the fact that we have an essentially infinite sinusoid in the time domain. This is because the time series is actually periodic, (with a period of 100 samples), because the frequency spectrum is discrete. Another way to explain this is to consider the DTFT and recall the fact that the DFT is just a sampled version of the DTFT. In this case the sampling is coherent; the zeroes of the Sinc function all lie at multiples of $1/N$, and hence the outputs of DFT (which is a sampled version of the DTFT at sample locations $1/N$) are all zero except at $f = \pm 20\ Hz$.

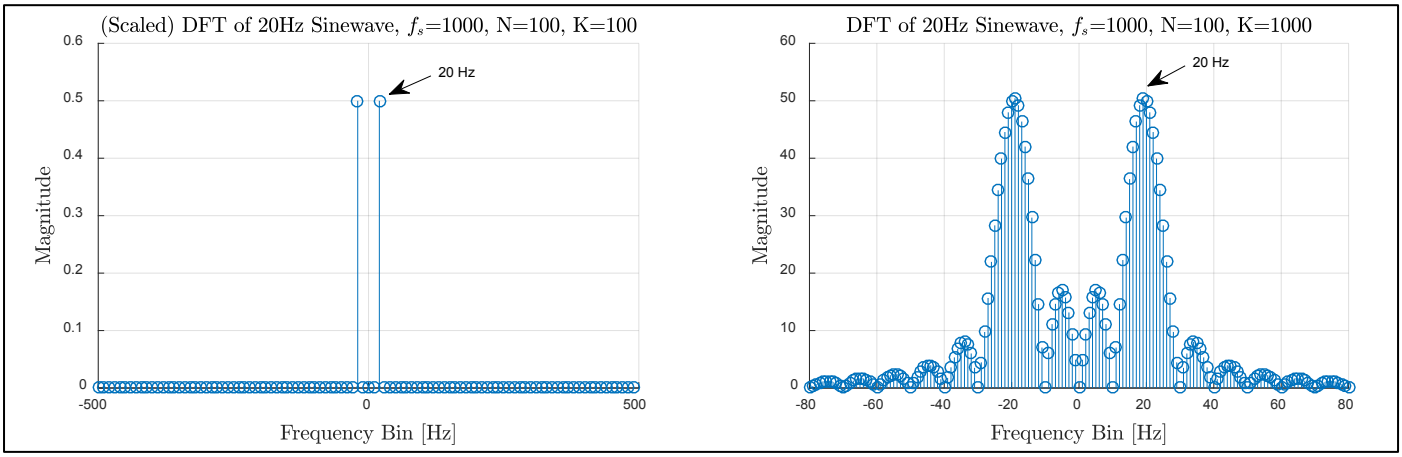

Figure 2: K point DFT of a $20\ Hz$ sinewave of length N. The sample frequency is $1000\ Hz$, and the DFT on the left is scaled by $1/K$ to make it more relatable to the Fourier transform.

1.1.c Incoherent Sampling:

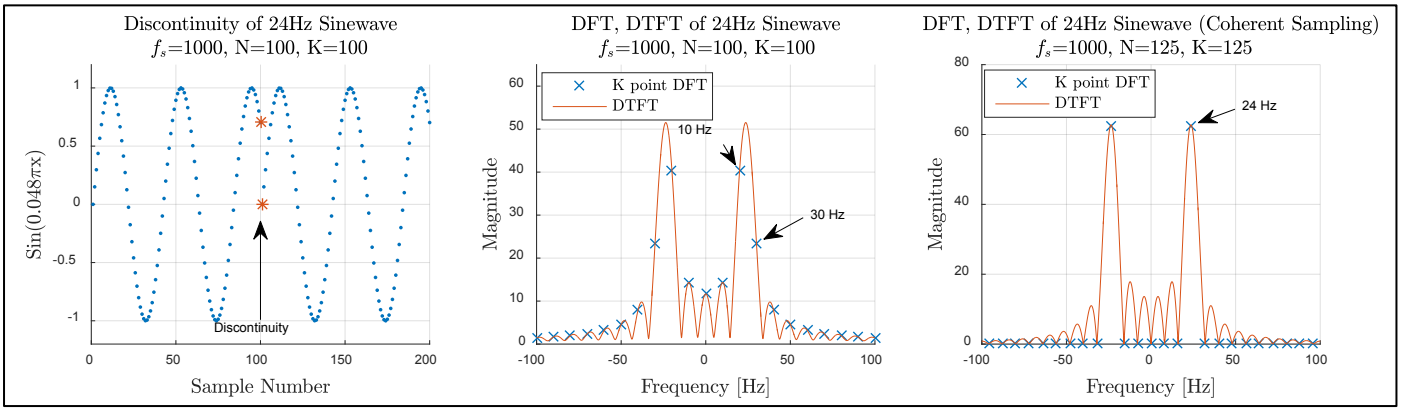

Figure 3: Showing the discontinuity in the 24 Hz sinewave and its DFT. The rightmost plot illustrates coherent sampling.

The middle plot of Figure 3 shows the DFT and DTFT of a 24 Hz sinewave sampled at 1000 Hz with a signal and DFT length of 100. From this we see that there is no single peak in the DFT spectrum. This is because there is a 'discontinuity' when the signal is repeated in the time domain (Figure 3, left plot) so this time we do not have an infinite sinewave unlike the previous case of the 20 Hz sinusoid. This 'discontinuity' essentially introduces additional spectral components in the DFT.

Another way to explain this DFT result is to consider the frequency resolution which is $fs/N = 10\ Hz$. Since 24 Hz is not an integer multiple of 10 Hz, the sinewave does not fall into exactly one frequency bin; so instead there is high power in the 20 Hz and 30 Hz bins. This is called incoherent sampling and it cannot be fixed by increasing the frequency resolution by zero padding since this corresponds to windowing and results in the same spectral shape but with more detail. It can be fixed by changing the sinewave length N until 24 is an integer multiple of the frequency resolution. When this is true, the sinewave has no discontinuities in the time domain (and the sampling is coherent), so we get the ideal Kronecker delta in the frequency domain. The right plot of Figure 3 achieves this with $N = 125$ which gives a frequency resolution of $8Hz$. If this is not possible, an alternative is to use a window function or tapering to remove the discontinuity and reduce spectral leakage (at the expense of more spectral smearing).

1.2 Properties of Power Spectral Density (PSD)

Approximation in the definition of power spectral density:

Equations 1.1 show two different definitions for the PSD. Definition 1 defines the PSD as the DTFT of the autocorrelation function (ACF). Definition 2 defines the PSD as the ensemble average of the PSD's of each of the sample functions. These two definitions can be shown to be equal under an assumption that the ACF decays sufficiently rapidly as shown by equation 1.2.

$$p(\omega) = \sum_{k=-\infty}^{\infty} r_{xx}(k)e^{-j\omega k} \quad (Def\ 1), \qquad p(\omega) = \lim_{N\to\infty} E\left\{\frac{1}{N}\left|\sum_{n=0}^{N-1} x(n)e^{-jn\omega}\right|^2\right\} \quad (Def\ 2) \tag{1.1}$$

$$\lim_{N\to\infty} \frac{1}{N} \sum_{k=-(N-1)}^{N-1} |k||r_{xx}(k)| = 0 \tag{1.2}$$

The proof is below. First we start with Definition 2 from equation 1.1:

$$p(\omega) = \lim_{N\to\infty} E\left\{\frac{1}{N}\left|\sum_{n=0}^{N-1} x(\mathrm{n})e^{-jn\omega}\right|^2\right\} = \lim_{N\to\infty} \frac{1}{N}\sum_{n=0}^{N-1}\sum_{m=0}^{N-1} E\{x(\mathrm{n})x^*(\mathrm{m})\}e^{-j\omega(n-m)} \tag{1.3}$$

Assuming $x(n)$ is zero mean and wide sense stationary (WSS) then $E\{x(n)x^*(m)\} = r_{xx}(n-m)$. WSS implies the ACF depends only on the time difference $n - m = k$. Therefore:

$$p(\omega) = \lim_{N\to\infty} \frac{1}{N}\sum_{n=0}^{N-1}\sum_{m=0}^{N-1} r_{xx}(n-m)e^{-j\omega(n-m)} = \lim_{N\to\infty} \frac{1}{N}\sum_{k=-(N-1)}^{N-1} (N-|k|)r_{xx}(k)e^{-j\omega k} \tag{1.4}$$

The step in 1.4 can be seen by expanding the double sum and considering symmetry and is clarified by 1.6 later on. The final step in 1.5 makes use of assumption 1.2 and completes the proof. Figure 4 shows an example when these two definitions give different results. This is because N is too small for the assumption 1.2 to be valid. Increasing N improves this case.

$$= \lim_{N\to\infty} \sum_{k=-(N-1)}^{N-1} r_{xx}(k)e^{-j\omega k} - \lim_{N\to\infty} \frac{1}{N}\sum_{k=-(N-1)}^{N-1} |k| r_{xx}(k)e^{-j\omega k} = \lim_{N\to\infty} \sum_{k=-(N-1)}^{N-1} r_{xx}(k)e^{-j\omega k} \tag{1.5}$$

$$\sum_{n=0}^{N-1}\sum_{m=0}^{N-1} f(n-m) = \sum_{n=0}^{N-1}\Big(f(n-0)+f(n-1)+\cdots+f\big(n-(N-1)\big)\Big)$$

$$\begin{aligned}
&= \quad f(0) \quad + \quad f(1) \quad + \ \ldots \ + \ f[-(N-2)] \ + \ f[-(N-1)] \\
&+ \quad f(1) \quad + \quad f(2) \quad + \ \ldots \ + \ f[-(N-3)] \ + \ f[-(N-2)] \\
&\quad\ \vdots \qquad\quad \vdots \qquad\qquad\quad \vdots \qquad\qquad \vdots \\
&+ \ f(N-2) \ + \ F(N-3) \ + \ \ldots \ + \quad f(-2) \quad + \quad f(-1) \\
&+ \ f(N-1) \ + \ f(N-2) \ + \ \ldots \ + \quad f(-1) \quad + \quad f(0)
\end{aligned} \qquad 1.6$$

$$= \sum_{k=-(N-1)}^{N-1} (N-|k|)f(k)$$

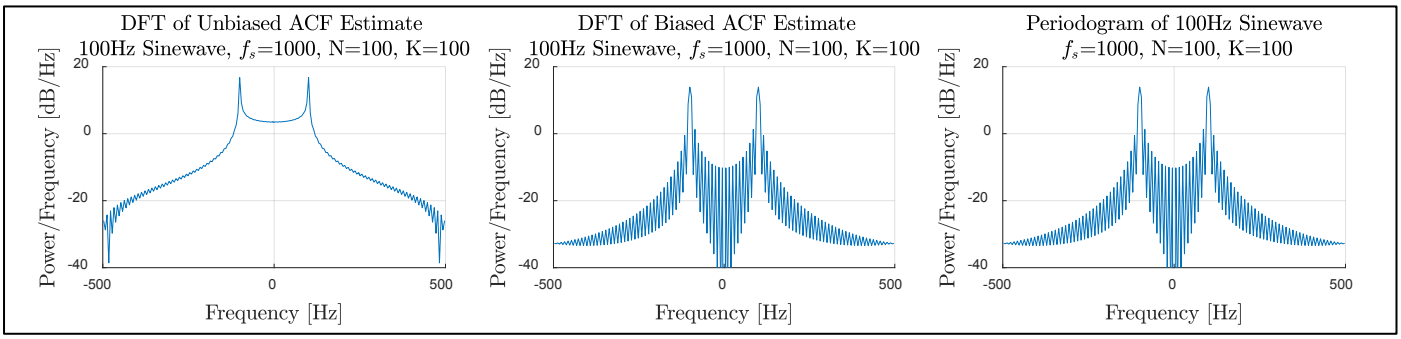

Figure 4: A case when the two definitions give different results. This is because the ACF of a sinewave does not die down to zero, therefore 1.2 is not valid. Note that if the biased ACF estimate is used, then two definitions are always exact regardless of the input signal and length.

1.2.a Zero Padding:

Zero padding is a computationally efficient method to increase the frequency bin resolution since it increases the number of frequency bins. However it does not necessarily increase spectral resolution which is the ability to distinguish the two spectral peaks of two equal amplitude sinusoids. This depends on the length of the data and the window function used. Furthermore, the FFT is most efficient when its length N has small prime factors and this can be achieved by zero padding the time series until N is close to an integer power of 2. One disadvantage is that variance increases by a factor N_{pad}/N_{orig}, i.e. the total signal length with zero padding divided by the original signal length.

$$\mathbf{x} = [r(0), r(1), \ldots, r(M-1), 0, \ldots, 0, r(-M+1), \ldots, r(-1)]^T \qquad 1.7$$

The particular choice of $\mathbf{x}$ given (eqn 1.7) is necessary due to the way FFT is computed in MATLAB. Since MATLAB's indexing starts from 1, there can be no negative indexes, and so the time series must be shifted in a circular fashion. If the DFT length is L, then the real time index $-k$ corresponds to a MATLAB index of $L+1-k$. Therefore the particular value of $\mathbf{x}$ is justified and necessary to maintain the conjugate symmetric property of the ACF. Figure 5 below justifies this form of $\mathbf{x}$. It shows that if no zero padding is used then the spectral resolution is low. It shows that if circular shifting is not used (i.e. circular symmetry is not maintained), then the PSD has complex and negative values (PSD should not be negative). Note that for the case of a complex FFT vector, only the real value is plotted since this is sufficient to demonstrate the need for a symmetric $\mathbf{x}$.

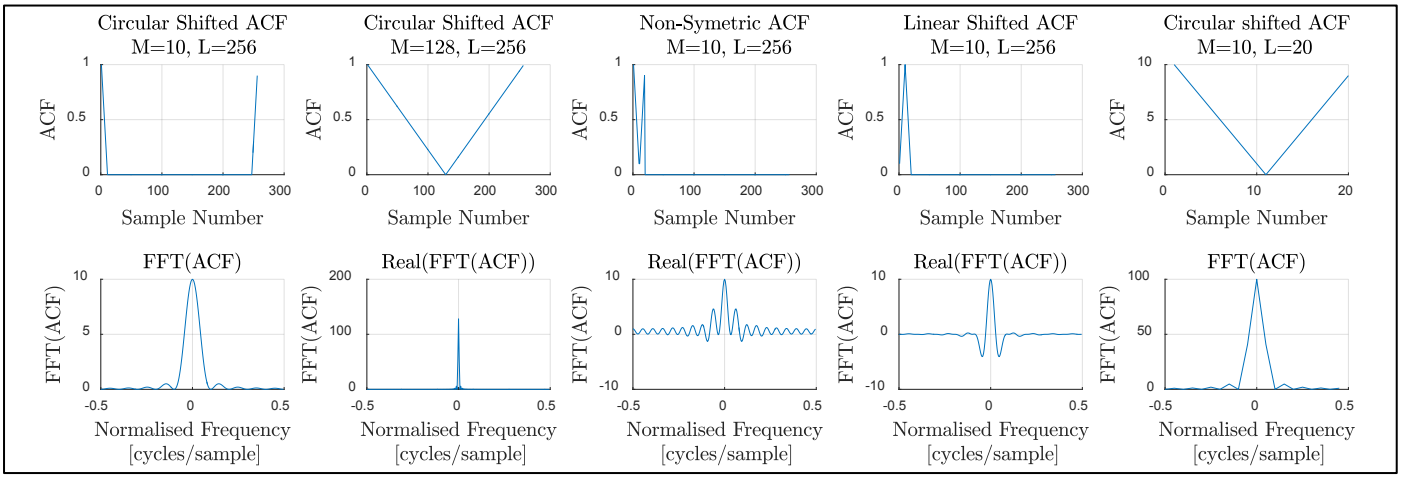

Figure 5: The proposed form of $\mathbf{x}$ for different M, and various unsuitable alternatives along with their corresponding $FFT(ACF) = PSD$

1.2.b Imaginary Components due to Round off Errors:

Since the ACF is symmetric we expect no imaginary components in the DFT of the ACF since they should ideally cancel out when the DFT sum is performed. However due to numerical errors, negligible but nonzero imaginary components are present in the DFT vector **xf**. Since we expect no imaginary part, is seems logical that the estimate can be improved removing the imaginary part. In reality this makes almost no difference since the imaginary part is so small (see Figure 6). In order to show how small the imaginary component is compared to the real part, some computations on the ratio of the imaginary part of **xf** to the real part of **xf** of have been performed and are summarised in Table 1 below. Note that Figure 6 is an odd function implying that the numerical errors are consistent and not random.

	$M = 10, L = 256$:	$M = 128, L = 256$:
Max:	$1.3148e-12$	$1.3907e-13$
$Mean$:	$-8.0973e-32$	$-6.8919e-31$
$Variance$:	$2.7331e-26$	$5.6807e-28$

Table 1 Max, mean and variance of $\Im(\mathbf{xf})/\Re(\mathbf{xf})$.

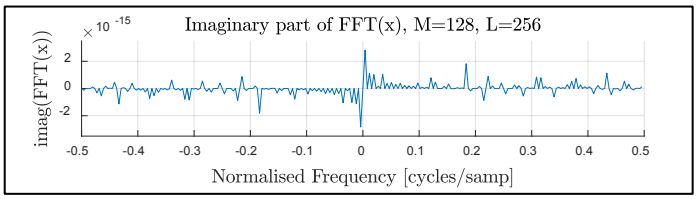

Figure 6: Imaginary part of the FFT vector. Note the odd symmetry.

1.2.c Invalid ACF, Negative and Complex PSD:

$$\mathbf{z} = [r(-M+1), \dots, r(-1), r(0), r(1), \dots, r(M-1), 0, \dots, 0]^T \quad 1.8$$

When the ACF is not symmetric the FFT is complex, and this is the case for the given vector **z** (eqn 1.8) when $M = 10$ and $L = 256$. When $M = 128$ and $L = 256$, the ACF is still symmetric so its FFT is real. However both these ACF functions still give erroneous spectral value as shown in Figure 7. This is because they have negative real parts and the PSD must be non-negative at all frequencies. This is because the area within a band of frequencies represents the power of those frequencies in the signal, and the area under the whole PSD represents the total signal power. We also know that this signal power cannot be negative. For a valid PSD this is not a problem because we know the ACF matrix for a WSS process is conjugate symmetric, toeplitz and positive semi-definite. Being positive semi-definite means the PSD can never be negative as explained in section 2.1.a.

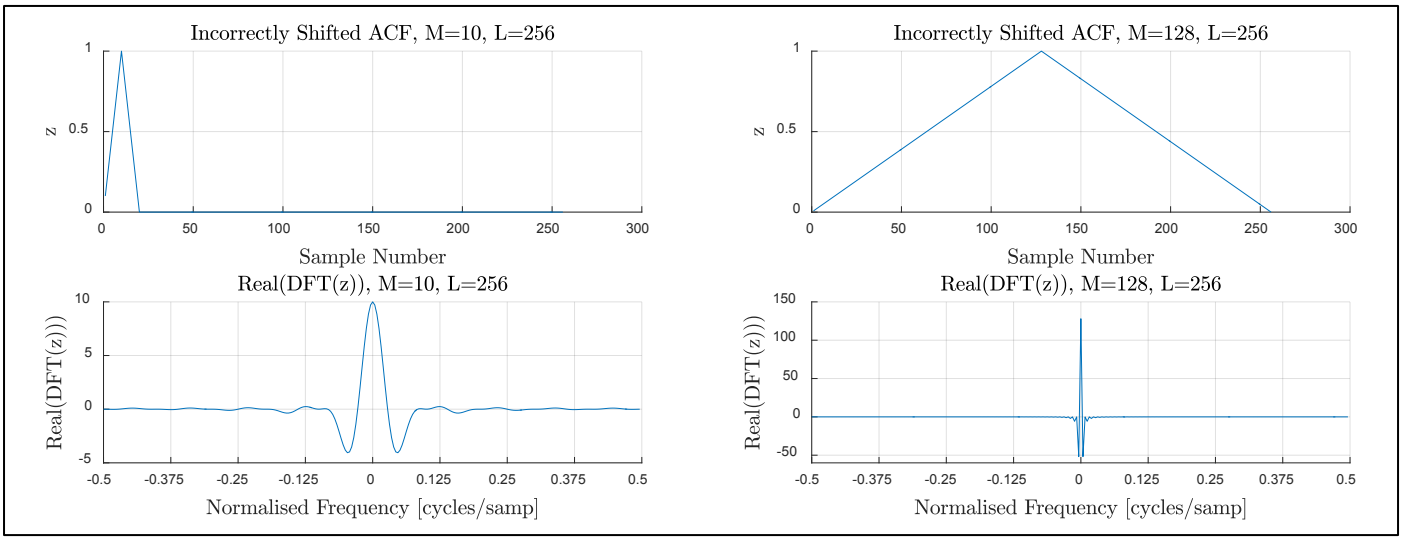

Figure 7: Incorrectly shifted ACF as given by **z** (eqn 1.8), and the corresponding real part of its DFT.

1.2.d FFT Shift Function:

The `fftshift` function swaps the first and second halves of the FFT output. This is useful since it allows the frequency axis to run from $[-\pi, \pi]$ radians per sample, and this is more convenient than $[0, 2\pi]$. This function has been used in the previous sections except the normalised frequency in cycles per sample is used as the x-axis. This ranges from $[-0.5, 0.5]$ as opposed to $[-\pi, \pi]$. The `fftshift` function can also be used to centre the ACF function on zero lag. The `ifftshift` function performs the opposite shifting, and this is useful if you have an ACF function centred on zero lag and you wish to correctly shift it in order to perform the FFT in order to find the PSD.

The required frequency vector is **w** is dependent on if N is even or odd. For convenience a MATLAB function called `x_axis` has been written (see below) which generates a correct frequency vector based on if N is even or odd.

```
function [ x ] = x_axis( N, F )
% Generates vector of x-axis values so they can be used when plotting FFT or ACF centred around 0.
% N = Number of samples, F = Max freq. For example, use pi when making a normalised frequency axis
    if mod(N,2) == 0    x=[-(N/2):1:(N/2)-1]*2*F/N;        % N is Even
    else                x=[-((N-1)/2):1:((N-1)/2)]*2*F/N;  % N is Odd
    end
end
```

Figure 8: MATLAB function to generate the correct frequency vector for centring plots around 0 .

1.3 Resolution and Leakage of Periodogram-Based Methods

1.3.a Bartlett Window 3dB Bandwidth and Sidelobe Peaks:

The definition of the Bartlett window and its DTFT is shown by equation 1.9. The 3dB points (in radians per sample) and the sidelobe gains have been empirically determined for a rage of N as shown by Figure 9 below.

$$w_B(k) = \begin{cases} 1 - \dfrac{|k|}{N}, & for\ |k| \le N \\ 0, & otherwise \end{cases} \quad \overset{DTFT}{\Longleftrightarrow} \quad W_B(\omega) = \frac{sin^2(N\omega/2)}{Nsin^2(\omega/2)} \tag{1.9}$$

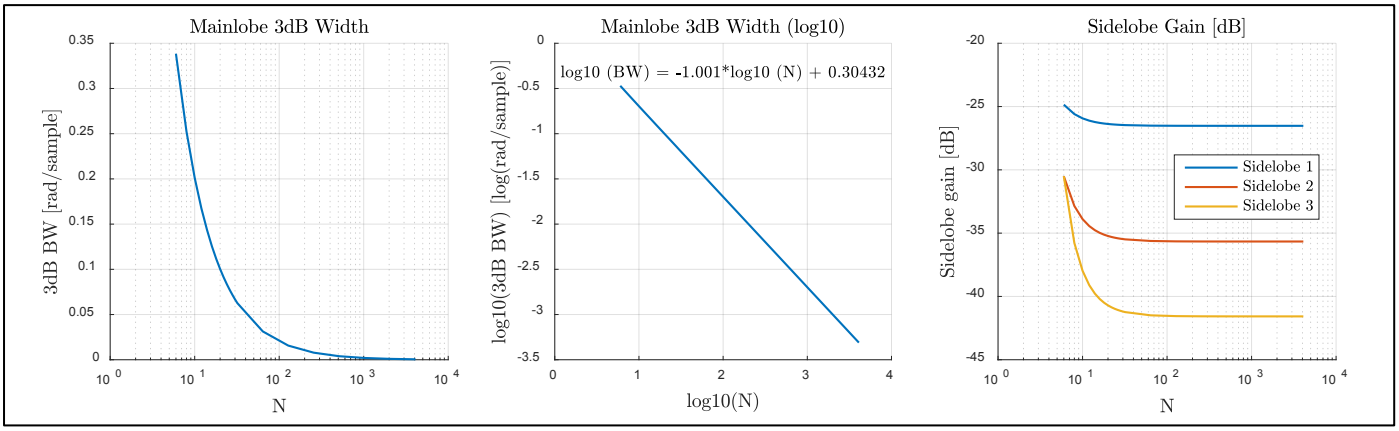

Figure 9: 3dB bandwidth and sidelobe gain for Bartlett window (eqn 1.9) of different lengths

The log-log plot in the centre of Figure 9 above shows the relationship between the 3dB points and N to be approximately: $\omega_{3dB} = 10^{0.30432}/N = 2.02/N$. Note that this result is a factor of 4 smaller than the theoretical relationship given in the lecture notes which states that the 3dB bandwidth is $1.28 \times 2\pi/N = 8.04/N$. This is because in the relation from the lecture notes N is defined as the total window length, whereas the analysis here is based on equation 1.9 which defines the total window length to be $2N + 1$. Furthermore, in the lecture notes, the 3dB bandwidth refers the total 3dB width of the mainlobe, whereas here only half the mainlobe is considered here as shown in Figure 10 below. The empirically derived relation can be shown to be roughly correct (for large N) by substituting ω_{3dB} into the DTFT equation given by 1.9:

$$\frac{W_B(\omega_{3dB})}{W_B(0)} = \frac{\dfrac{sin^2(N\omega_{3dB}/2)}{Nsin^2(\omega_{3dB}/2)}}{N} = \frac{sin^2(1.01)}{N^2 sin^2(1.01/N)} \approx \frac{sin^2(1.01)}{1} \approx \frac{1}{\sqrt{2}} \tag{1.10}$$

The sidelobe gain is roughly constant for $N > 50$ as shown by Figure 9, with the first sidelobe takes a minimum of -26.5dB, the second one -35.66dB, and the third sidelobe -41.58dB. The 3dB bandwidth and sidelobe attenuation are graphically verified by Figure 10 for a 3 different values of N. Note that the amplitude of the spectra have been normalized to 1 for convenience.

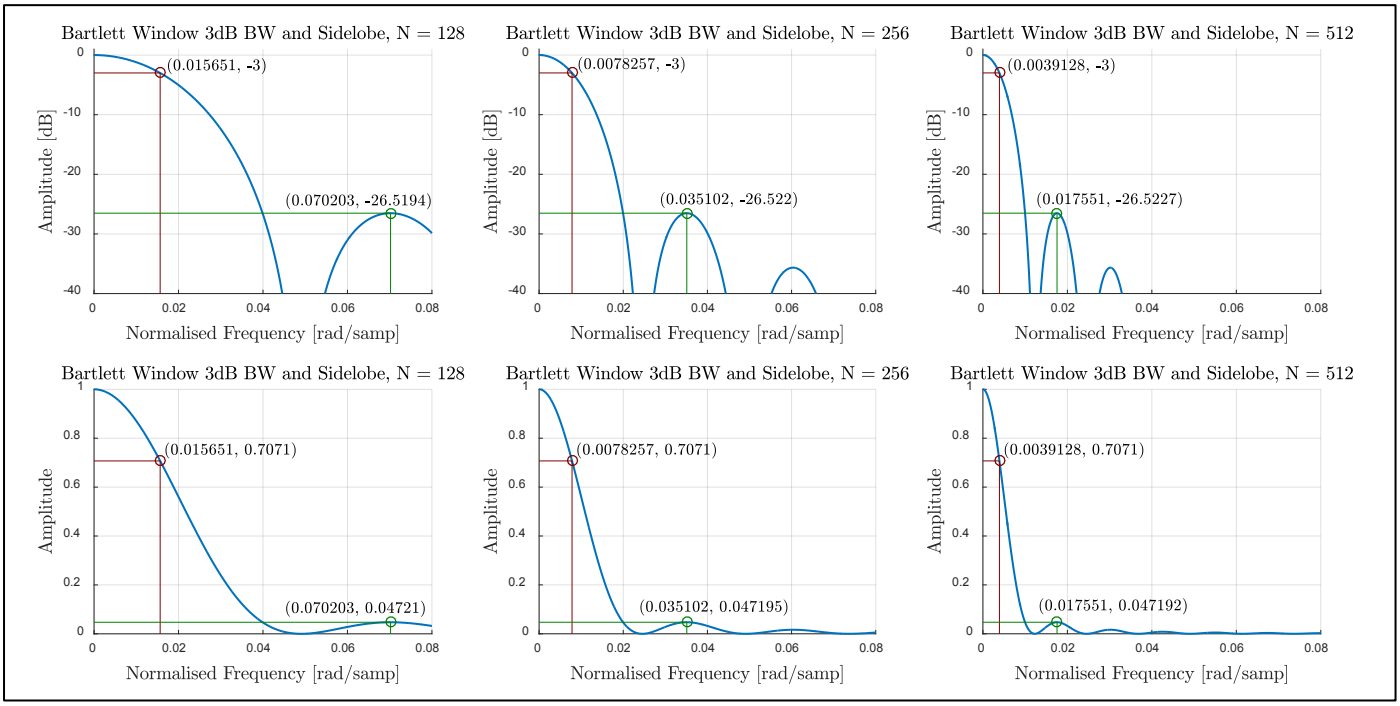

Figure 10: Amplitude normalised Bartlett window spectra in both dB and linear scales. The half power bandwidth point is shown by the red circle and the green circle shows the first sidelobe's amplitude. Note that the total window length is $2N + 1$ according to def. 1.9

1.3.b Periodogram Resolution:

$$\hat{P}_{per}(\omega) = \frac{1}{N}\left|\sum_{k=0}^{N-1} x(k)e^{-j\omega k}\right|^2 \qquad (The\ Periodogram\ Estimator\ of\ the\ PSD) \qquad 1.11$$

$$x(n) = a_1 sin(0.2 \times 2\pi n) + a_2 sin\big((0.2 + \alpha/N) \times 2\pi n\big) + w(n), \qquad where \quad w \sim \mathcal{N}(0, \sigma^2) \qquad 1.12$$

The periodogram (eqn 1.11) for the function $x(n)$ (eqn 1.12) is shown by Figure 11 below. Here $N = 256, \sigma = 0, a_1 = a_2 = 1$, and α is varying. The parameter α controls the frequency separation of the two sine waves. By finding the minimum α for which we can distinguish the sinusoidal peaks we can find the spectral resolution of the periodogram. Note that the variance of the PSD estimator is zero since $\sigma = 0$ and this allows us to purely focus on the bias of the periodogram. Evidence of two sinusoids first appears when $\alpha = 0.65$, and when $\alpha > 0.7$, the two sinusoids are clearly resolvable. This minimum value of α for which the sinusoids are resolvable does not change with N. This implies that the normalized frequency resolution is around $0.65 \times 2\pi/N$ radians per sample. Since the estimated PSD equals the true PSD convolved with the Bartlett window, we expect to be able to distinguish two sinusoids if they are separated by $2 \times 2.02/N$. This is the full 3dB mainlobe width of the Bartlett window as determined in section 1.3.a. This matches the observations here since $0.65 \times 2\pi = 4.08 \approx 2 \times 2.02$.

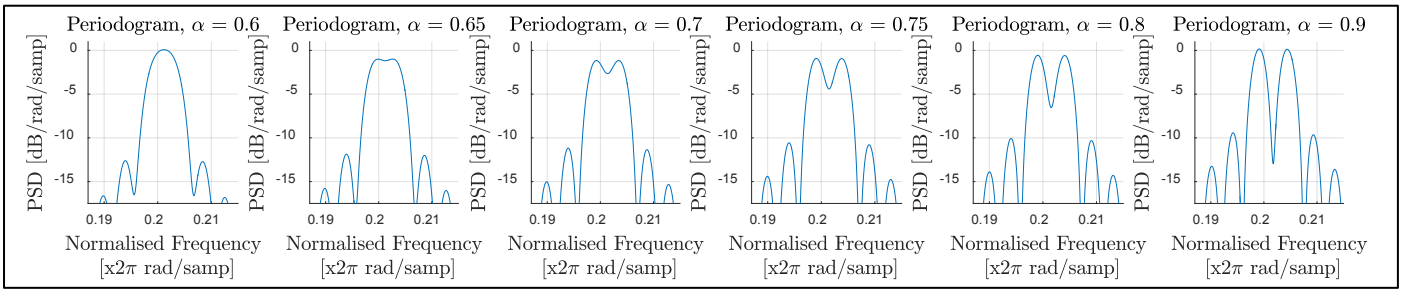

Figure 11: Zero padded Periodogram of $x(n)$ (eqn 1.12) with $N = 256,\ \sigma = 0,\ a_1 = a_2 = 1$ and various α.

1.3.c Periodogram Resolution with a Hamming Window:

The process in 1.3.b is repeated but this time with a Hamming window. Figure 12 shows that a minimum alpha of around 0.75 to 0.8 allows the two sinusoids to be resolved. This means using a Hanning window results in less spectral resolution compared to a rectangular window. This is expected since the rectangular window has the narrowest mainlobe of all the window functions, thus resulting in the least amount of spectral smearing and hence greatest resolution. The Hanning window is smoother in the time domain so its sidelobes in the frequency domain are smaller and decay faster. However, this smoothness reduces effective window width so its mainlobe is wider compared to an equivalent width rectangular window.

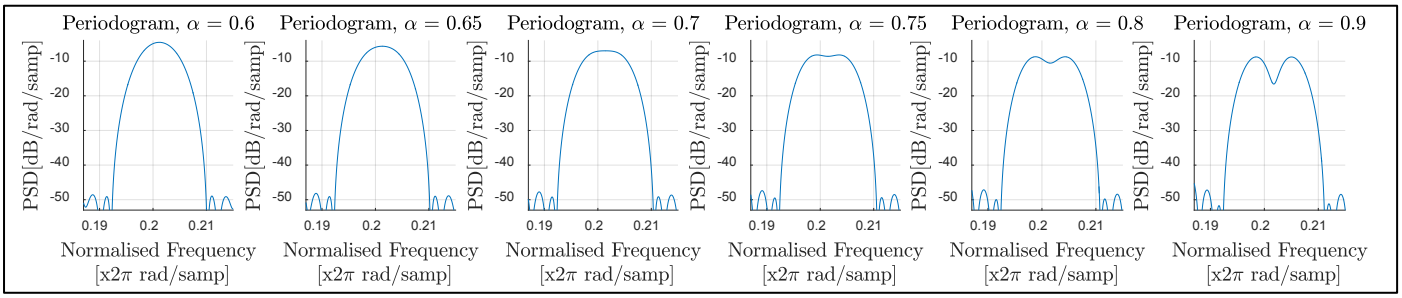

Figure 12: Zero padded Periodogram of $x(n)$ with $N = 256,\ \sigma = 0,\ a_1 = a_2 = 1$ and various α. A Hamming window function is used.

1.3.d Spectral Leakage of Rectangular Window:

The PSD of $x(n)$ is computed again with a rectangular window. This time the two sinusoids have a well separated frequency but greatly different amplitudes. Figure 13 shows the PSD for $\alpha = 4$, $a_1 = 1$ and various a_2. Figure 14 shows the PSD for $\alpha = 12$, $a_1 = 1$, and various a_2. For $\alpha = 4$, the second sinusoid is indistinguishable when $a_2 \leq 0.01$. With a greater frequency separation ($\alpha = 12$), the second sinusoid is still identifiable for $a_2 = 0.01$, but not for $a_2 \leq 0.001$. Figure 14 and Figure 13 essentially demonstrate how spectral leakage can obscure a smaller amplitude sinewave especially if the frequencies are close.

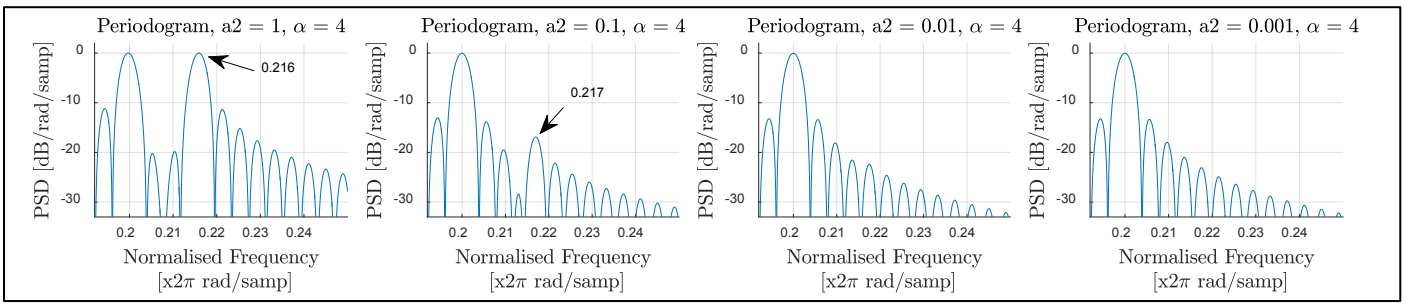

Figure 13: The PSD of $x(n)$ *for* $\alpha = 4, a_1 = 1$ and various a_2.

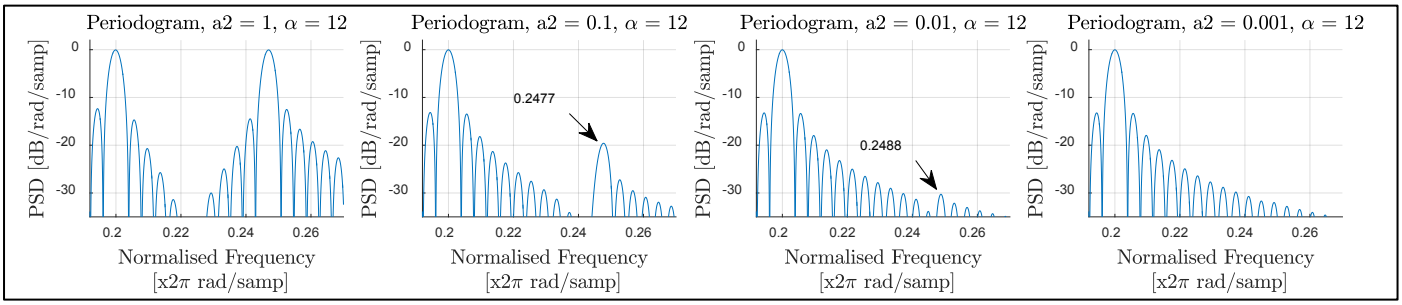

Figure 14: The PSD of $x(n)$ *for* $\alpha = 12, a_1 = 1$ and various a_2.

1.3.e Effects of Finite Data Lengths:

The data length is finite so the periodogram is actually a biased estimator of the true PSD. Since we have set the variance of the additive white Gaussian noise (AWGN) to 0, the variance of the PSD estimate is 0, and this allows us to purely focus on the bias. The expected value of the periodogram is the convolution between the DTFT of the Bartlett window, $W_B(\omega)$ (eqn 1.9), and the true PSD as shown by eqn 1.13. Note that the Periodogram estimator is asymptotically unbiased since $W_B(\omega) \to \delta(\omega)$ as $N \to \infty$. For the special case of white noise, the variance is given by equation 1.14. As $N \to \infty$, the variance tends to a non-zero value of $P_{xx}^2(\omega)$, hence the periodogram estimator is inconsistent.

$$E\{\hat{P}_{per}(\omega)\} = W_B(\omega) * P_{xx}(\omega) \quad 1.13$$

$$var\{\hat{P}_{per}(\omega)\} = P_{xx}^2(\omega)\left[1 + \left(\frac{\sin(\omega N)}{N\sin(\omega)}\right)^2\right] \quad 1.14$$

$$P_{xx}(\omega) = 0.5\pi {a_1}^2[\delta(\omega + 0.4\pi) + \delta(\omega - 0.4\pi)] + 0.5\pi {a_2}^2[\delta(\omega + 0.4\pi + \alpha\pi/128) + \delta(\omega - 0.4\pi - \alpha\pi/128)] \quad 1.15$$

$$P_{per}(\omega) = 0.5\pi {a_1}^2[W_b(\omega + 0.4\pi) + W_b(\omega - 0.4\pi)] + 0.5\pi {a_2}^2\left[W_b\left(\omega + 0.4\pi + \frac{\alpha\pi}{128}\right) + W_b(\omega - 0.4\pi - \frac{\alpha\pi}{128})\right] \quad 1.16$$

The true PSD of $x(\mathrm{n})$ is given by equation 1.15, and the periodogram estimate is given by equation 1.16. W_b is shown by Figure 15, along with the relative frequencies corresponding to the two values of α. From this, it is clear that the mainlobe of the smaller amplitude sinusoid is in-between two sidelobes of the larger sinusoid. Therefore it is essentially hidden if its amplitude is lower than the sidelobes amplitude. Since the sidelobes further from the mainlobe are smaller, it is easier to resolve the smaller amplitude sinusoid if it has a large frequency separation and this is confirmed by the observations in section 1.3.d. Figure 15 shows the amplitude of the sidelobes relative to the mainlobe is approximately -40dB, and -60dB for $\alpha = 4$ and $\alpha = 12$ respectively. Hence the minimum relative amplitude of the smaller sinusoid for it to be distinguishable is approximately -40dB, and -60dB. This corresponds to a minimum value of a_2 to be 0.01 and 0.001 for $\alpha = 4$ and $\alpha = 12$ respectively which agrees with the results in 1.3d.

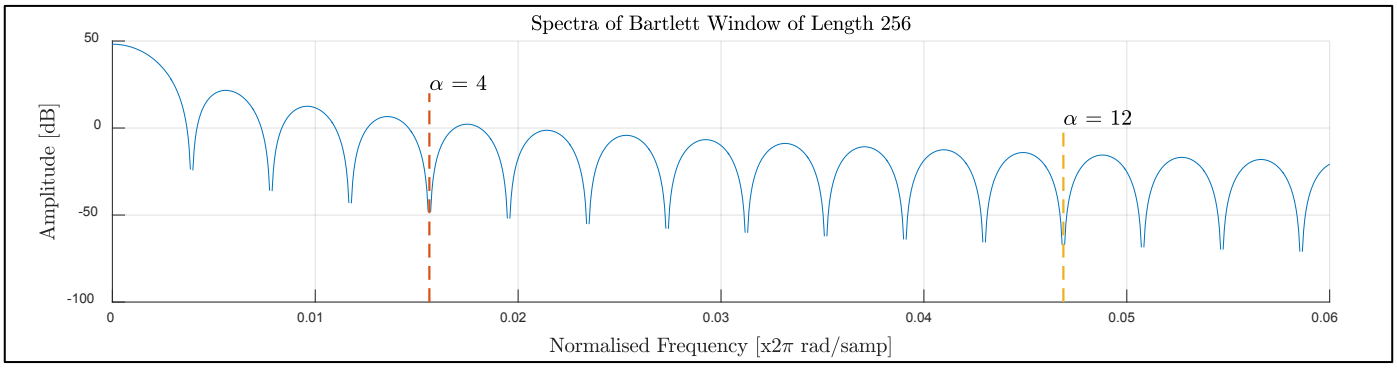

Figure 15: DTFT of the Bartlett window of length 256. The dotted lines show the relative positions of the two sinusoid peaks

1.3.f Chebyshev and Blackman-Tukey Windows:

$$\hat{P}_{per}(\omega) = \sum_{k=-N+1}^{N-1} \hat{r}_{xx}(k)e^{-j\omega k} \qquad (Periodogram\ alternate\ expression) \quad 1.17$$

$$\hat{P}_{BT}(\omega) = \sum_{k=-M}^{M} w(k)\hat{r}_{xx}(k)e^{-j\omega k} \qquad (Blackman - Tukey\ method) \quad 1.18$$

The Chebyshev window has a constant sidelobes whose amplitudes can be designed and traded off with the mainlobe width. A wider mainlobe means more spectral smearing, and larger sidelobes mean more spectral leakage. Therefore there is a trade-off between spectral resolution due to the mainlobe, and spectral masking due to the sidelobes. Therefore, a narrow mainlobe is needed if the sine waves to be resolved are close in frequency, and low sidelobes are needed if the sine waves to be resolved have greatly different amplitudes.

This is illustrated by Figure 16 which shows that a window with smaller sidelobes ($-120dB$) allows us to distinguish the two sine waves when $\alpha = 12$ and $a_2/a_1 = 0.0001 = -80dB$. This is expected since $-80dB > -120dB$ so the smaller sinewave is not masked by the sidelobes. However, we cannot really distinguish the two sine waves for the case when $\alpha = 4$ since they are too close in frequency and the smearing is too much. Figure 17 shows the case when the sidelobe attenuation is -90dB. In this case we are able to just about distinguish the two sine waves for when $\alpha = 4$ and $a_2/a_1 = 0.001$ as required.

Equation 1.17 shows an alternate representation of the periodogram and equation 1.18 shows a modification of this which is referred to as the Blackman-Tukey method. It involves selecting $M < N - 1$ and applying a window to the ACF estimate in order to reduce the contribution of unreliable estimates at high lags. ACF estimates at high lags (k) have high variance since only $N - |k|$ points are used to calculate them. Since the PSD estimate is the DFT of the ACF estimate, windowing the ACF estimate corresponds to convolution with spectra of window in the power spectral domain. This results in smoothing of the power spectra (and also introduces bias). With smaller M the bias is worse since the mainlobe is wider, but the variance of the periodogram reduces since the variance of the ACF estimate is less since only robust ACF values at low lags are used. The figures below also show the Blackman-Tukey estimate with $M = 0.5N$ and a Chebyshev window. The smoothing of the spectra is clear, and it is more evident when the Chebychev window has smaller sidelobes since its mainlobe is wider.

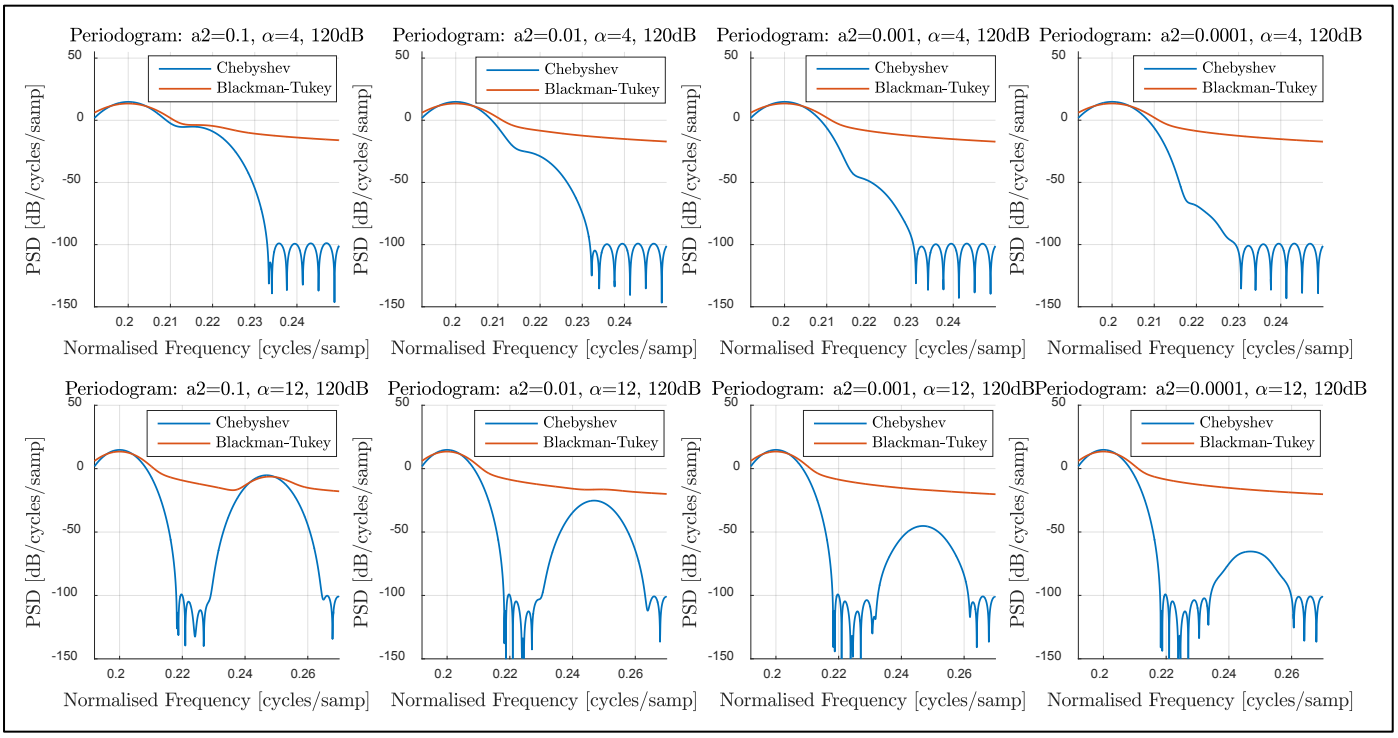

Figure 16: $120dB$ Chebyshev window and corresponding Blackman-Tukey periodograms for $a_2 = \{0.1, 0.01, 0.001, 0.0001\}$ and $\alpha = \{4, 12\}$

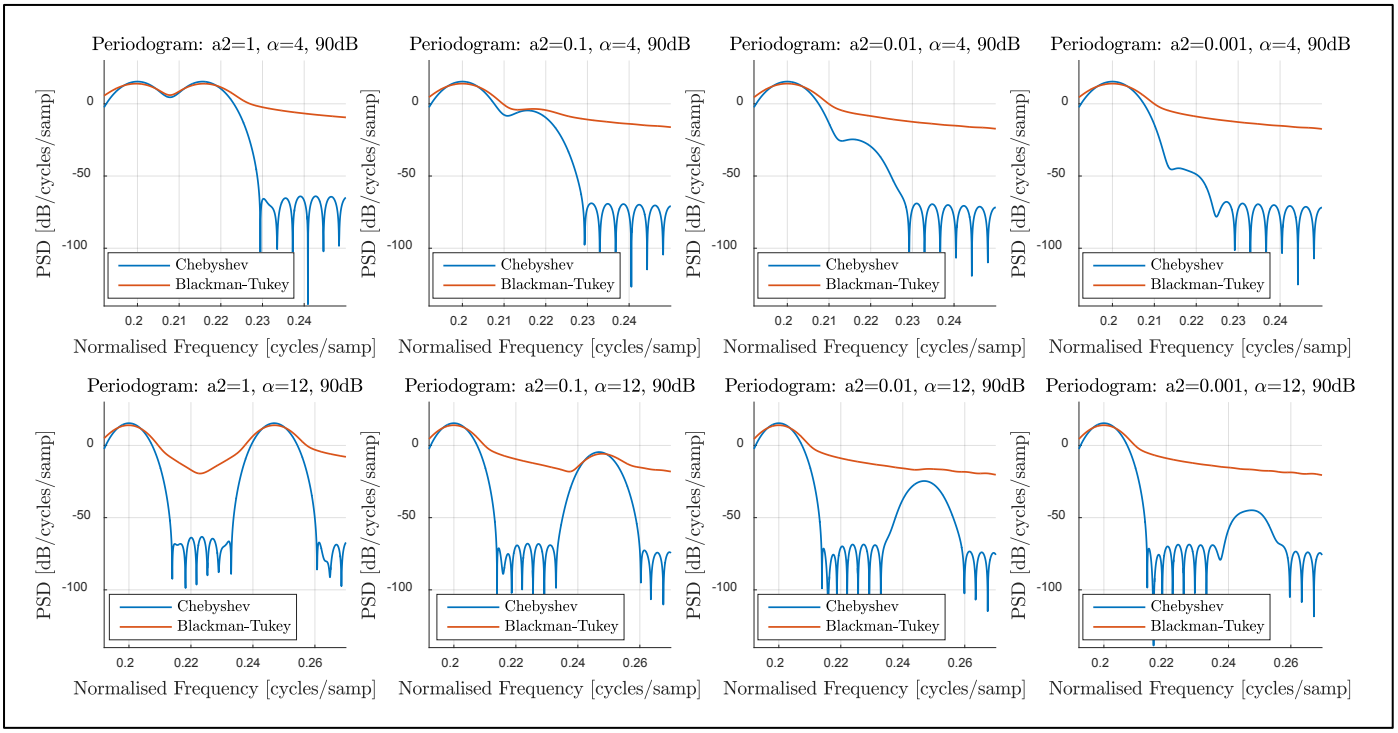

Figure 17: $90dB$ Chebyshev window and corresponding Blackman-Tukey periodograms for $a_2 = \{1, 0.1, 0.01, 0.001\}$ and $\alpha = \{4, 12\}$

1.4 Periodogram Based Methods Applied to Real-World Data

1.4.a Sunspot Time Series

The top right plot of Figure 18 shows the zero padded PSD estimate for the sunspot time series with various pre-processing methods. Removing the mean results in no DC component and hence no power in the $0\,Hz$ frequency bin. De-trending the series almost completely removes the mean while also attenuating some low frequency components, i.e. the range 0 to 0.003 cycles per year. Beyond around 0.025 cycles per year, the de-trended PSD estimate and zero mean PSD estimate are almost exactly the same.

The top left plot of Figure 18 shows the normalized time series and the normalized log of the time series. From these two it is clear that the sunspot data is cyclic with a period of 11 years, and so we expect a spectral peak in the PSD estimates at 0.909 cycles per year. Taking the log of the data has the effect of compressing values in the range $(1,\infty)$, and amplifying values in the range $(0,1)$. The values in the range $(0,1)$ are mapped to the range $(-\infty,0)$. This is why the time series is more rounded at the peaks but spikier at the troughs. The PSD estimate of this log data is shown in the bottom left plot, and from this we can see a spectral peak at 0.906 cycles per year. Taking the log has made this spectral peak clearer, and this is due to the compression properties of the log. Previously the time series was very spikey at the peaks but the log compresses this making it closer to a sinewave. Note that before taking the log of the data, a minimum floor is set on the sunspot count since $(0) \rightarrow -\infty$. Finally the averaged periodogram is shown for 4, 8 and 16 averages by the bottom right plot of Figure 18. This averaging results in less variance but also less resolution and from this plot it is easier to see the spectral peak at around 0.91 cycles per year.

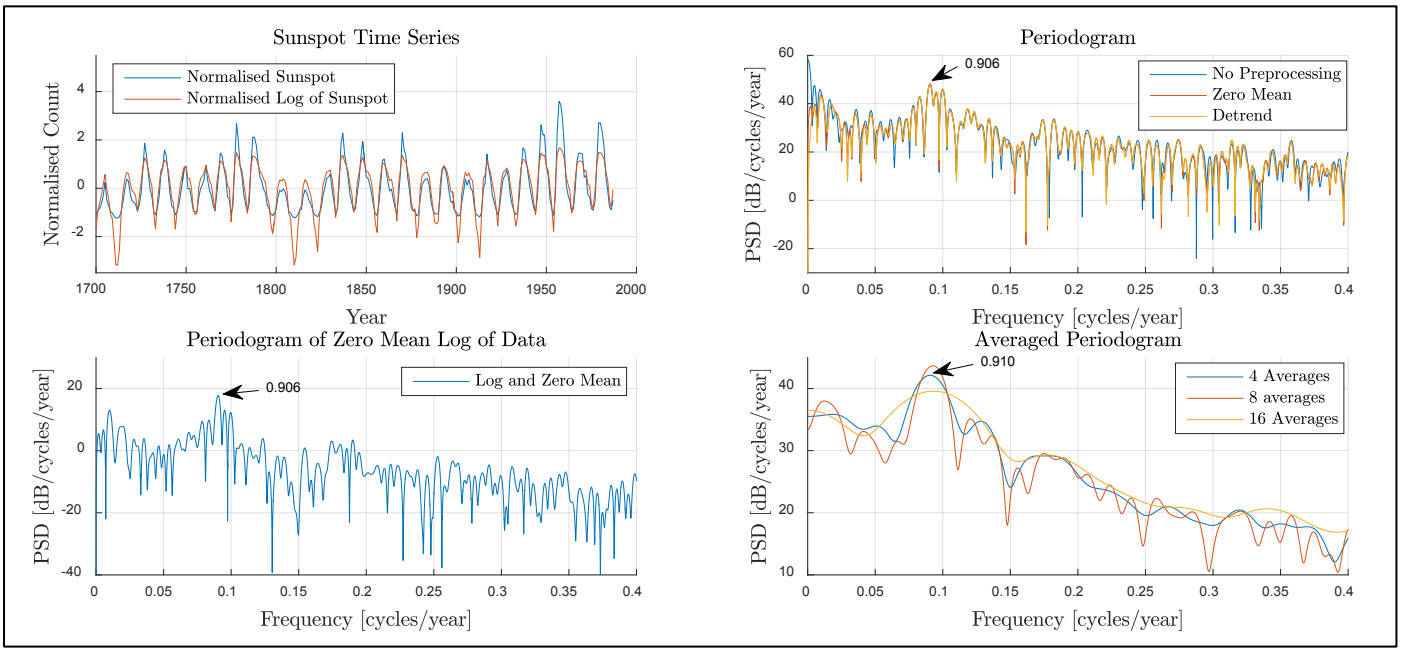

Figure 18: PSD estimates for the sunspot time series using various pre-processing methods and periodogram averages. The sample frequency is once per year.

1.4.b Brain Computer Interface (BCI)

Figure 19 shows various Periodogram based methods applied to the EEG data. The first plot shows the standard periodogram applied to the full time series. From this we can suggest the SSVEP to be at $13\,Hz$. We can see up to the third harmonic at $39\,Hz$, but confidence is low since the variance is high; the peaks we see could just be random variations. Averaging M periodograms (Bartlett's method) reduces the variance by a factor of M assuming statistical independence between segments (but this is not likely). For non-white data, the variance reduction is less than this. However the averaging means shorter segments of data have to be used resulting in less spectral resolution. Thus there is a trade-off between variance reduction and spectral resolution, and this is evident in the averaged periodogram in Figure 19. In this plot, the 10 second window shows a noticeable reduction in variance compared to the standard periodogram. From this we can confidently identify the SSVEP peak at $13\,Hz$, and the first and second harmonics at 26 and $39\,Hz$ respectively. The 1 second window has even less variance, but as a downside in that it does not have enough spectral resolution to precisely identify the SSVEP peaks and harmonics.

Windowing the time series segments reduces spectral leakage and this is clear by considering the 1 second window length around the $50\,Hz$ power line interference peak. However this is not really needed here since the SSVEP of interest is known to be far away from $50\,Hz$ and the leakage does not obscure the SSVEP peak. The Welch method (overlapping modified periodograms) with a Hanning window results in even less variance and hence a smoother PSD estimate. Overlapping by more than 70% doesn't result in a noticeable reduction in variance but increases computation time. If the data was white, we would expect variance to keep reducing as overlap increases. The Blackman-Tukey method and the welch method with a 10 second window result in the most useful PSD estimates since the SSVEP peaks and harmonics are most narrow, distinct and clear.

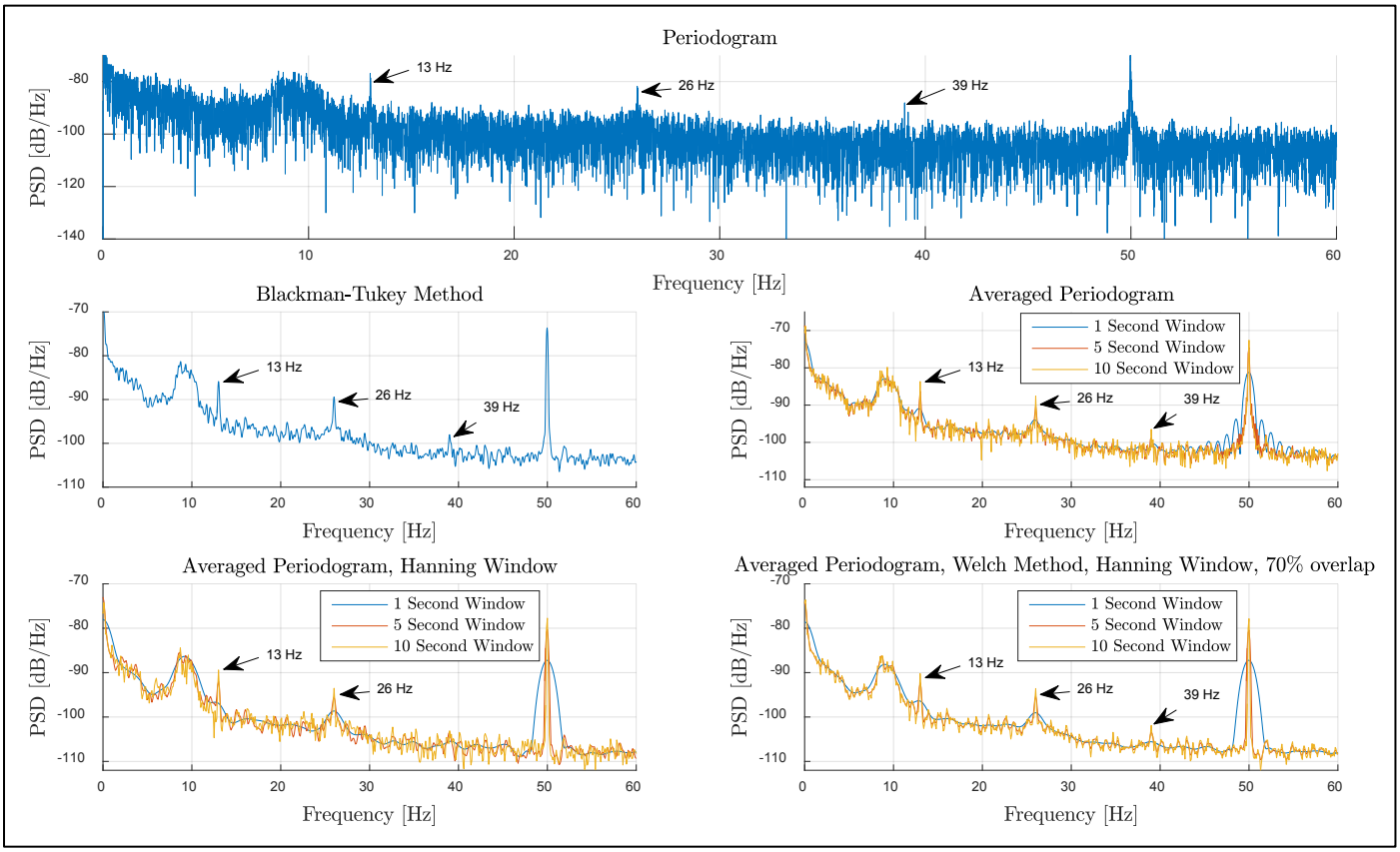

Figure 19: Various Periodogram methods applied to EEG data. 10 frequency bins per Hz are used in all the averaged periodograms for fairness.

2 Parametric and Line Spectra

2.1 Correlation Estimation

2.1.a Biased and Unbiased Correlogram:

Considering definition 1 of the PSD (eqn 1.1), it seems intuitive to use the unbiased estimate for the ACF since on average this will give the true ACF and hence on average the true PSD (since the FT is a linear operation). However, in reality the time series is finite meaning that for large autocorrelation lags, i.e. lag $|k|$ close to N, the ACF estimate has high variance and is highly erratic. This is because only $N - |k|$ samples are used to estimate the ACF at lag $|k|$. Additionally, the higher lag ACF values are given a higher weighting due to the $1/(N - |k|)$ factor. As a result of these inaccuracies the ACF may not be positive definite.

To guarantee a non-negative PSD, the ACF matrix ($\mathbf{R}_{xx}$) must be positive semi-definite. This can proved as follows. Consider any linear system given by the vector coefficients $\mathbf{a} = [a_0 \dots a_{N-1}]^T$ with an input sequence $\mathbf{x}$ and the output sequence $\mathbf{y}$ given by $y(n) = \sum_{k=0}^{N-1} a(k)x(n-k) = \mathbf{x}^{\mathrm{T}}\mathbf{a} = \mathbf{a}^{\mathrm{T}}\mathbf{x}$. Also note that $E\{y^2[n]\} = E\{y[n]y^T[n]\} = E\{\mathbf{a}^{\mathrm{T}}\mathbf{x}\mathbf{x}^{\mathrm{T}}\mathbf{a}\} = \mathbf{a}^{\mathrm{T}}E\{\mathbf{x}\mathbf{x}^{\mathrm{T}}\}\mathbf{a} = \mathbf{a}^{\mathrm{T}}\mathbf{R}_{xx}\mathbf{a}$, and that $E\{y^2[n]\}$ is the output power and is therefore non-negative. Thus to preserve non-negative power the ACF must be positive semi-definite, i.e. $\mathbf{a}^{\mathrm{T}}\mathbf{R}_{xx}\mathbf{a} \geq \mathbf{0} \ \forall \ \mathbf{a}$.

The biased estimate for the ACF has the same properties as the true ACF, i.e. $\mathbf{R}_{xx} \geq 0$ $r_{xx}(0) \geq |r_{xx}(k)|, r_{xx}(0) \geq 0$ and $r_{xx}(k) = r_{xx}(-k)$, (proof Priestly 1981). The biased estimator is essentially equal to the unbiased estimator multiplied by the triangular window given by 1.9. Multiplying in time is the same as convolution in the frequency domain, so we can use the spectra of the window (also given by equation 1.9) to understand the bias in the correlogram estimate. In fact we have the exact same spectral smearing and leakage effects as the periodogram estimate (eqn 1.11) as expected.

Figure 20 below shows the biased and unbiased ACF estimates along with their corresponding correlograms for various example signals. The sinusoid example illustrates the bias in the biased ACF estimate at high lags since the ACF dies down as the lag increases when ideally it should not. The second example of white Gaussian noise (WGN) shows the erratic nature of the unbiased ACF estimate at high correlation lags. The high variance at high lags makes the estimate worse, for example we know that the ACF of WGN should be a Kronecker delta. The biased estimate is closer to this compared to the unbiased estimate particularly at higher lags. Figure 20 also shows the correlograms corresponding to the unbiased ACF estimates result in negative values for the estimated PSD which is incorrect since signal power cannot be negative in any band as previously explained.

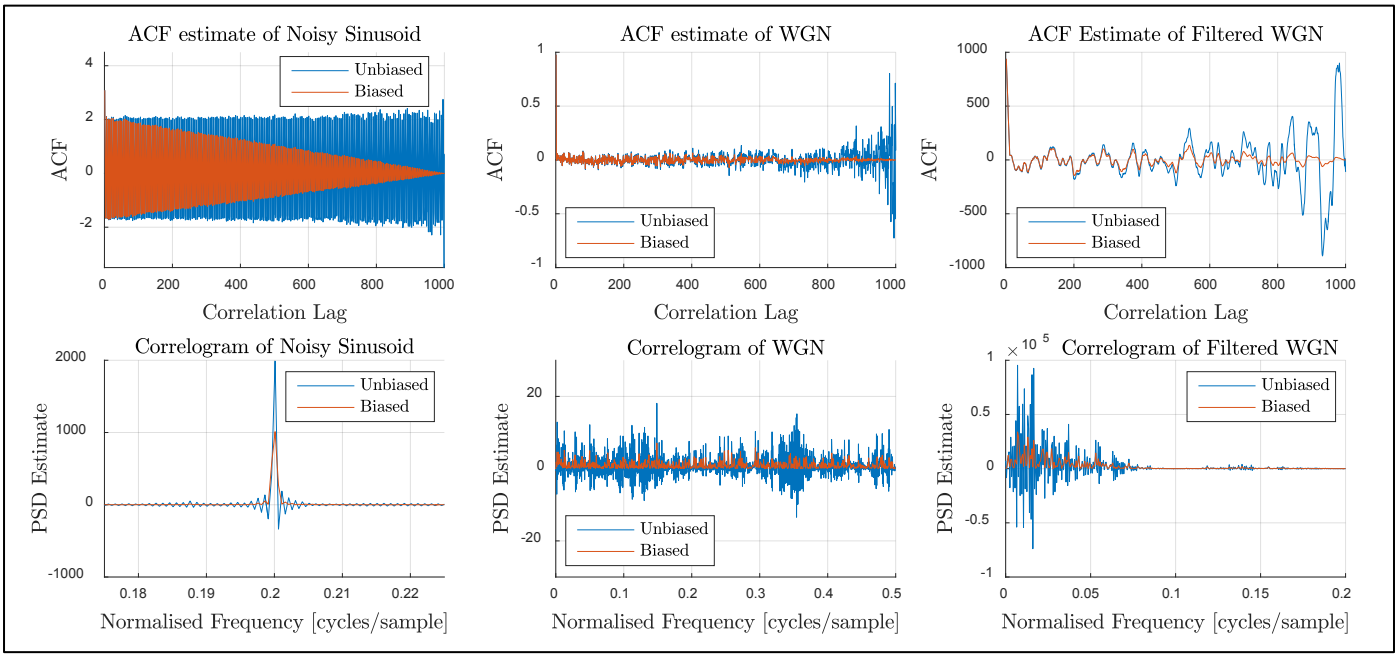

Figure 20: Biased and unbiased ACF estimate and their corresponding correlograms for various noisy signals. A log scale is not used in the PSD estimates since the unbiased ACF estimate results in negative spectra. The filter used in the rightmost plot is a low pass MA filter of order 9 which has a half power bandwidth of around 0.045 cycles per sample.

2.1.b Dispersion of PSD Estimates:

The top left plot of Figure 21 shows the biased PSD estimates for 500 independent realisations of a random process of length 128. The signal consists of two sinusoids with different frequencies $(0.15, 0.3$ cycles per sample) and amplitudes $(1.5, 0.75)$ in additive white Gaussian noise (AWGN) of variance 1. The top right plot shows the same processes but this time filtered with a MA low pass filter of order 9 which has a half power bandwidth of around 0.045 cycles per sample. From the plot of the filtered processes, it is easier to see that the variance of the estimate is proportional to power. In fact the asymptotic variance $(N \to \infty)$ is given by $\mu {P_{xx}}^2(\omega)$ where $\mu > 0$ is a constant. For the special case of zero mean WGN with variance σ^2, the asymptotic variance of the periodogram is given by ${P_{xx}}^2(f) = \sigma_w^4$, or by $\frac{1}{(2\pi)^2}\sigma_w^4$ if the PSD is radians per sample.

2.1.c Decibel Plots of PSD Estimates and their Standard Deviation:

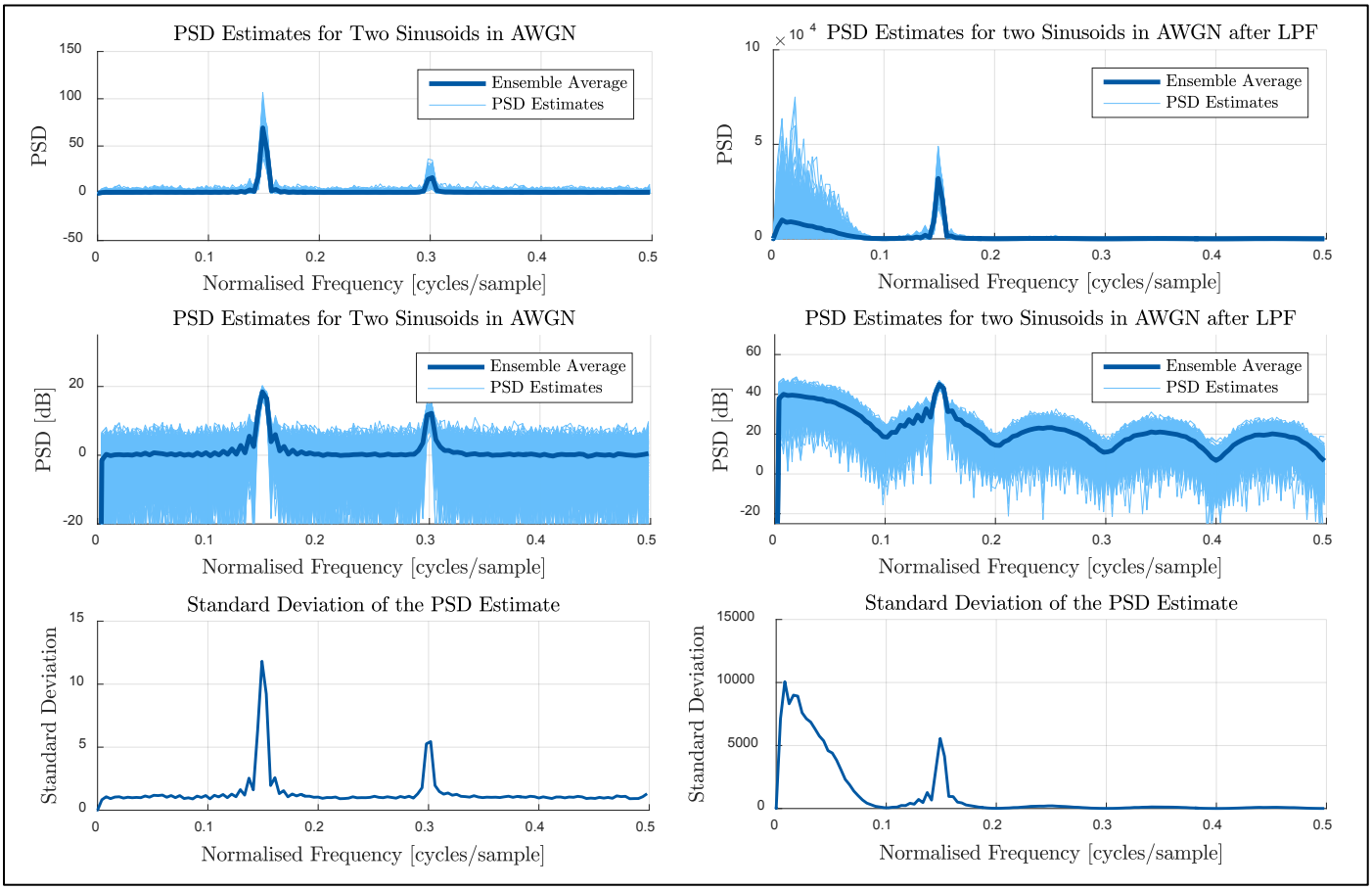

Figure 21: Periodograms of 500 independent realisations of two random processes. Left plots show two sinusoids in AWGN, and the right ones show the same process but filtered by a MA low pass filter of order 9 which has a half power bandwidth of around 0.045 cycles per sample.

The second row of plots in Figure 21 show the same plots in a decibel scale. The compression property of the logarithm allows us to compare relative differences more easily. For example, we know the standard deviation is proportional to PSD so on the log scale we would just see a shift rather than a multiplication, i.e. note that $\log(kx) = \log(k) + \log(x)$. This therefore allows us to see more detail in the dispersion of lower power spectral regions; regions which were otherwise overshadowed by high powers in the non-log plots. However, we are usually more interested in the high power regions so this may not be so relevant. The last row of plots in Figure 21 shows how the variance is proportional to the PSD. Note that the asymptotic relation described in section 2.1.b is not necessarily valid here since we only used sample functions of length 128 samples.

2.1.d Complex Exponential Generation:

The periodogram of two complex exponentials (sine waves) in complex AWGN is shown in Figure 22. A rectangular window is used and the FFT length is 128. The first row of plots show how two complex exponentials with frequencies 0.3 and 0.32 Hz ($f_s = 1$) cannot be distinguished if the signal length N is only 30 samples, but can be if N is increased to 35 samples. The second row of plots shows a constant signal length of 25 but with varying frequency separation. Only for a frequency separation $\geq 0.03\ Hz$ are we able to distinguish the two frequencies. This is expected since we know the spectral resolution is proportional to $1/N$. Since the 3dB bandwidth of the rectangular window is $0.89/N\ Hz$, we expect to be able to comfortably distinguish frequencies separated by $0.02\ Hz$ if $N \geq 45$, and this is confirmed by the first row of plots. Note that the PSD is no longer symmetric sine the time series is complex, therefore the full range of normalized frequencies from 0 to $f_s = 1$ have been plotted.

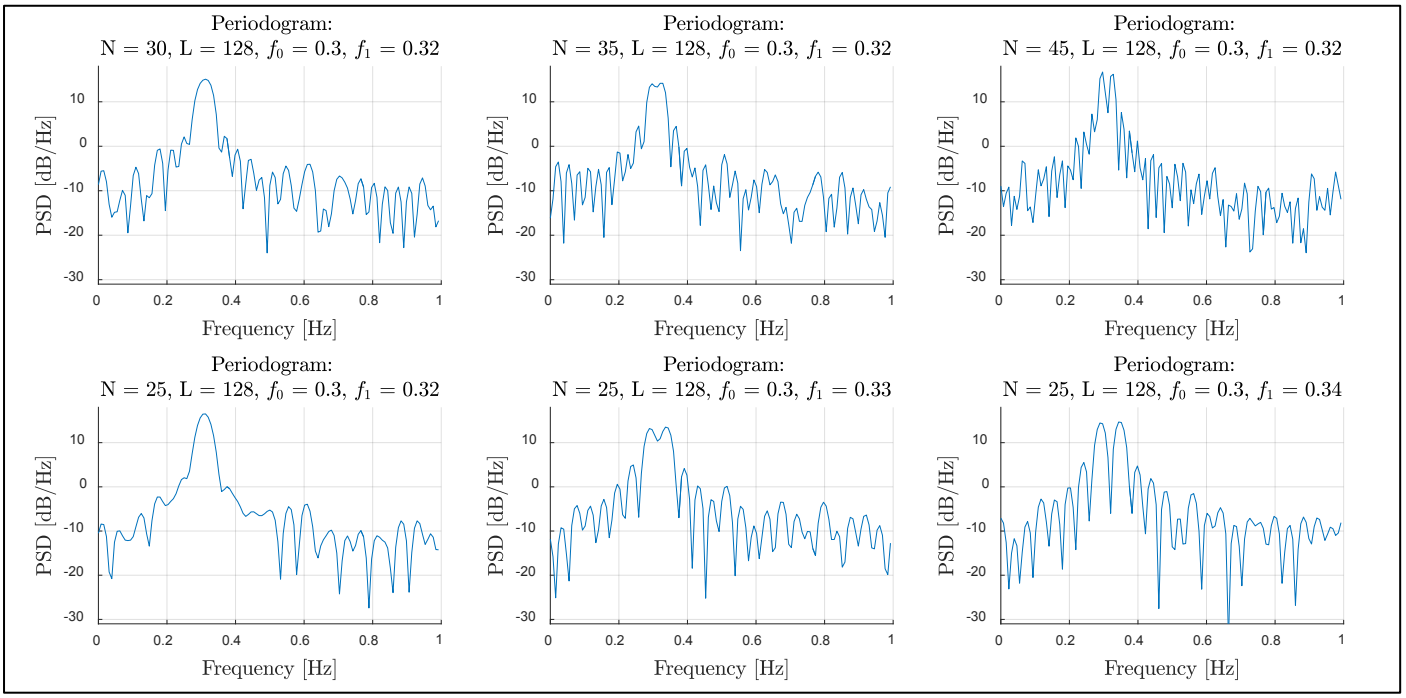

Figure 22: Periodogram of two complex exponentials with closely spaced frequencies in complex AWGN with variance of 0.2. The top row of plots shows two frequencies at 0.3 and 0.32 Hz ($f_s = 1$) with varying signal lengths (N). The second row shows a fixed signal length but with varying frequency separation.

2.1.e MUltiple SIgnal Classication Method (MUSIC):

$$V_i(z) = \sum_{k=0}^{M-1} v_i(k) z^{-k} \tag{2.1}$$

$$|V_i(\omega)|^2 = \frac{1}{|\sum_{k=0}^{M-1} v_i(k) e^{-jk\omega}|^2} \tag{2.2}$$

$$P_{MU}(\omega) = \frac{1}{\sum_{i=p+1}^{M} |\mathbf{e}^H \mathbf{v}_i|^2}, \quad where \quad \mathbf{e} = \left[1, e^{j\omega}, e^{j2\omega}, \ldots, e^{j(M-1)\omega}\right]^T, \quad \mathbf{v}_i \text{ are the ordered eigenvectors of } \mathbf{R}_{xx} \tag{2.3}$$

The MUSIC method is an eigen-based method for the estimation of line spectra and is an improvement to the Pisarenko Harmonic Decomposition method. It assumes a signal of the form $x(n) = \sum_{i=1}^{p} A_i e^{jn\omega_i} + w(n)$, where $w(n)$ is white noise of variance σ_w^2. Let $\mathbf{R}_{xx}$ denote the $M \times M$ Autocorrelation matrix where $M > p + 1$. It is worth noting that when $M = p + 1$, this method is equivalent to the Pisarenko decomposition. Also let λ_i denote the eigenvalues of $\mathbf{R}_{xx}$ arranged in decreasing order, and $\mathbf{v}_i$ denote the corresponding eigenvectors. The p largest eigenvectors corresponding to the p largest eigenvalues of $\mathbf{R}_{xx}$ span the p dimensional signal subspace. The remaining $M - p$ eigenvectors corresponding to the remaining $M - p$ eigenvalues span the orthogonal, (since $\mathbf{R}_{xx}$ is hermitian), $M - p$ dimension noise subspace. Ideally the noise eigenvalues are all equal to σ_w^2, but in reality $\mathbf{R}_{xx}$ is estimated so this is not true. Therefore a better way to obtain σ_w^2 is by averaging the smallest $M - p$ eigenvalues.

Estimating the frequencies of the complex exponentials is more involved. Each eigenvector $\mathbf{v}_i$ of $\mathbf{R}_{xx}$ is of length M, so each of the noise subspace eigenfilters (eqn 2.1) will have $M-1$ roots. In an ideal case, p of these roots will lie on the unit circle at locations corresponding to the frequencies of the complex exponentials in $x(n)$, thus the eigenspectrum (eqn 2.2) associated with each eigenvector $\mathbf{v}_i$ will have peaks at these frequencies. The problem is that the remaining $M-p-1$ zeros may also lie close to the unit circle, thus resulting in spurious peaks in the eigenspectrum. Additionally, since the $\mathbf{R}_{xx}$ is estimated, the zeros corresponding to the complex exponential frequencies may not lie exactly on the unit circle, thus making it harder to distinguish the actual frequencies from the spurious peaks. The MUSIC algorithm improves on this by making use of all the noise eigenvectors and averaging, i.e. using the frequency estimation function given by equation 2.3. The p largest peaks in the spectrum given by 2.3 correspond to the frequencies of the complex exponentials, and once these frequencies are determined we can find the power of each complex exponential by solving a set of p linear equations.

```
(1)  [X,R] = corrmtx(x,14,'mod');
(2)  [S,F] = pmusic(R,2,[],1,'corr');
(3)  plot(F,S,'linewidth',2); set(gca,'xlim',[0.25 0.40]);
(4)  grid on; xlabel('Frequency [Hz]'); ylabel('Pseudospectrum');
```

Figure 23: MATLAB script for the MUSIC algorithm as given in the CW hand-out.

The MATLAB code shown in Figure 23 above finds the line spectra of the inputted signal $\mathbf{x}$ which is composed of two complex exponentials with frequency 0.3 and $0.32\ Hz$ $(f_s = 1\ Hz)$ in complex AWGN which has a of variance 0.2. The first line returns $\mathbf{X} \in \mathbb{Z}^{(2N-2M)\times(M+1)}$ which is the modified (`'mod'`) rectangular Toeplitz matrix that is used to generate the autocorrelation estimate for $\mathbf{x}$. $M = 14$ as specified by the second argument and $N = 30$ (the length of the input). $\mathbf{X}$ is computed using forward and backward prediction error estimates based on an 14^{th} (M^{th}) order prediction error model. It also returns $\mathbf{R} \in \mathbb{Z}^{(M+1)\times(M+1)}$ which is the autocorrelation matrix estimate $\widehat{\mathbf{R}}_{xx}$ and is computed by $\widehat{\mathbf{R}}_{xx} = \mathbf{X}^T\mathbf{X}$.

The second line computes the pseudo-spectrum in the vector $\mathbf{S}$ at the frequencies given in the vector $\mathbf{F}$ which are in Hz. The first argument is to be interpreted as the autocorrelation matrix as instructed by the option `'corr'`. The sampling frequency is $1\ Hz$ and specified by the third argument, and the second argument is empty meaning the default number frequencies to evaluate at is 256 (or 128 is $\mathbf{R}$ is real). The second argument is the signal subspace dimension which is two since we know we have two complex exponentials. The third line simply plots the pseudospectrum and sets the x-axis range to the region of interest which is around the 0.3 and $0.32\ Hz$ sinusoids. Figure 24 below shows the spectra.

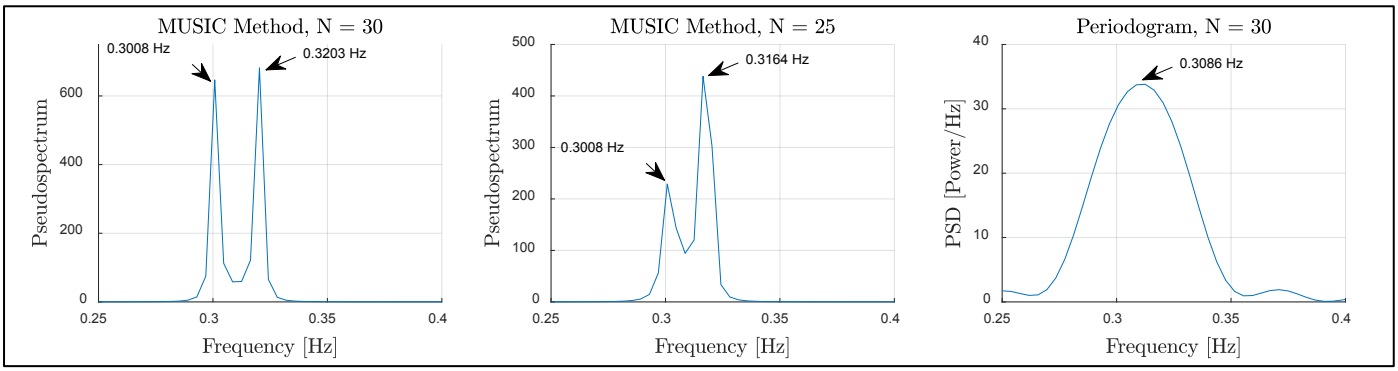

Figure 24: MUSIC and periodogram applied to a short signal consisting of two complex exponentials with frequencies at $0.3, 0.32\ Hz$, $(f_s = 1)$

Figure 24 shows that the MUSIC method is able to correctly identify the frequencies with high resolution even in cases when the periodogram cannot (i.e. for small $N = 30$). There is also less bias in the pseudospectrum compared to the PSD estimate obtained by the periodogram since there is no leakage and less smearing. There is also less variance.

Despite all the advantages mentioned, the MUSIC method is a limited because it is only applicable to signals which consist of complex exponentials in white noise whereas the periodogram is applicable for any signal provided it is sufficiently long. The music method is a 'denominator' method, which means it would find it hard to estimate a non-peaky spectra; something which the periodogram is particularly suited for. Furthermore, if the noise is not white, the frequencies estimated are biased and the algorithm is very sensitive to estimation error of the estimated ACF matrix. It is also computationally expensive for higher order problems compared to the periodogram (since the FFT is $O(Nlog(N))$) and further computations are required to estimate the powers once the frequencies are determined.

2.2 Spectrum of Autoregressive Processes

2.2.a Limits of Unbiased ACF Estimate:

$$y(n) = \sum_{k-1}^{p} a_k y(n-k) + w(n), \qquad where\ w(n) is\ white\ noise \tag{2.4}$$

$$P_{AR}(\omega) = \frac{\sigma_w^2}{\left|1 + \sum_{k=1}^{p} a_k e^{-jk\omega}\right|^2} \tag{2.5}$$

$$\mathbf{R}_{xx}\mathbf{a}_p = \sigma_w^2\mathbf{b}, \qquad where \qquad \mathbf{a}_p = \left[1, a_1, \dots, a_p\right]^T, \qquad and \qquad \mathbf{b} = [1, 0, \dots, 0]^T \tag{2.6}$$

An autoregressive (AR) processes has the form given by equation 2.4, and its power spectrum is given by equation 2.5. The AR parameters can be found using the Yule Walker method which makes use of the autocorrelation matrix up to lag p where p is the AR model order, so $\mathbf{R}_{xx} \in \mathbb{R}^{p\times p}$. The method involves solving p linear equations as given by equation 2.6. Therefore any inaccuracies in the estimated ACF matrix $\hat{\mathbf{R}}_{xx}$ will result in inaccuracies in the estimated model parameters $\hat{\mathbf{a}}_p$ and $\hat{\sigma}_w^2$. The unbiased ACF estimate has high variance at large lags (as discussed in 2.1.a) and so can cause significant inaccuracies in the estimate of $\mathbf{R}_{xx}$, especially if p is close to N which is the data length. Note that the unbiased and biased ACF matrices are similar for relatively small lags, so if $N \gg P$, then either the biased or the unbiased method can be used. However the biased one is generally preferred. Another shortcoming is that the unbiased ACF estimate is not guaranteed to be positive definite. In fact the unbiased $\mathbf{R}_{xx}$ estimate can actually be singular meaning there may be no solution, or an infinite amount of solutions to equation 2.6. In this case we can still find the least squares solution, or the minimum norm solution respectively. Note that if $\hat{\mathbf{R}}_{xx} > 0$, then the roots of the $\mathbf{A}_p(z)$ lie inside the unit circle and so an estimated AR process suing the biased ACF estimate is always stable. A final remark is that the variance of the spectrum tends to be large when $\hat{\mathbf{R}}_{xx}$ is ill conditioned.

2.2.b Modelling of an AR(4) Process:

Figure 25 investigates the effect of model order in autoregressive spectrum estimation. An AR(4) process of length 500 is generated using the `filter` function with the coefficients $\mathbf{a}_4 = [1, -2.76, 3.81, -2.65, 0.92]$, $\mathbf{b} = [1]$, and with white noise input. The first 500 output samples are discarded to remove the transient output of the filter. Then the model is estimated using `aryule` (which uses the biased ACF estimate and solves the Yule walker equations), and the power spectrum is plotted using the frequency response generated by `freqz`. The black plot in Figure 25 shows the true PSD which is determined based on the underlying model used to generate the process. There are two spectral peaks in the true spectrum and this is expected since the process order is 4. This can be conceptualised by considering the pole-zero locations of the process as shown by the leftmost pole zero diagram in Figure 26.

When a model order of 4 is chosen, there is only one peak in the spectrum and an order of 6 is required to start developing two peaks. When the order is 10, the estimated PSD resembles the true PSD. We expect the optimal model order to be 4 since this is the order of the generated AR process. However since there are only 500 samples, the ACF estimate used to compute the AR coefficients has too much variance and bias and as a result is not accurate enough and a higher order is necessary to see two peaks. Figure 26 shows the pole-zero locations of the various estimated models and explains why the AR(10) estimate resembles the true PSD while the lower order estimates do not. Figure 25 also includes the averaged periodogram for comparison. It shows that the periodogram struggles to model the peaky spectra and this is because the periodogram is of MA type (no poles) which means it is not ideal for peaky spectra.

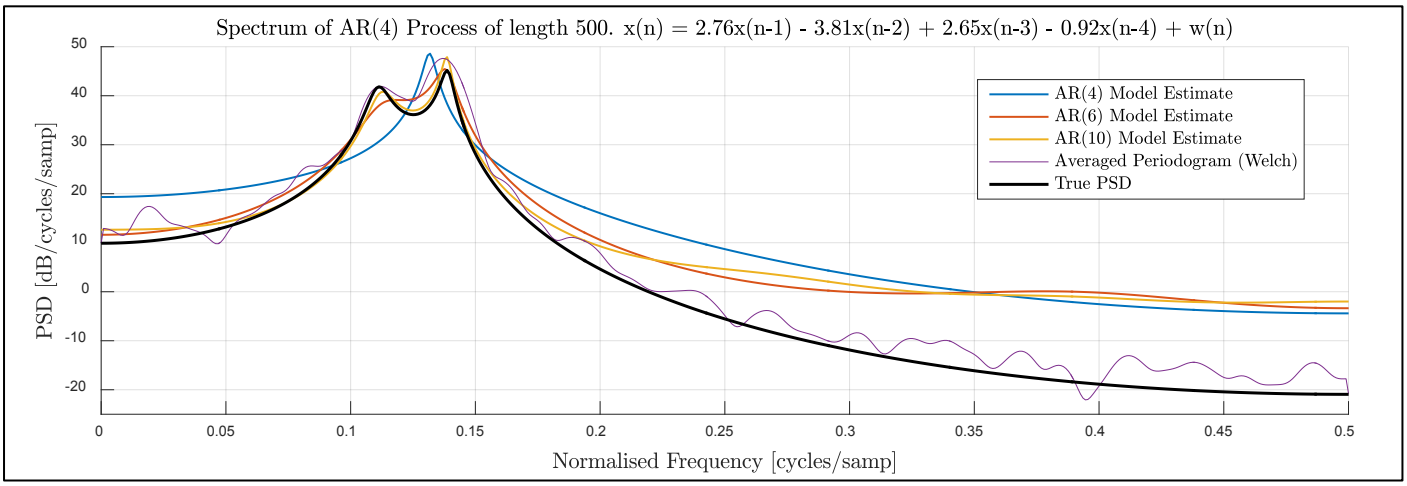

Figure 25: Parametric (AR) PSD estimates with different model orders of an AR(4) process. The Welch method periodogram is also shown with 90% overlap and a Hanning window of length 150. The true PSD is plotted in black using `freqz` and the known AR(4) process coefficients.

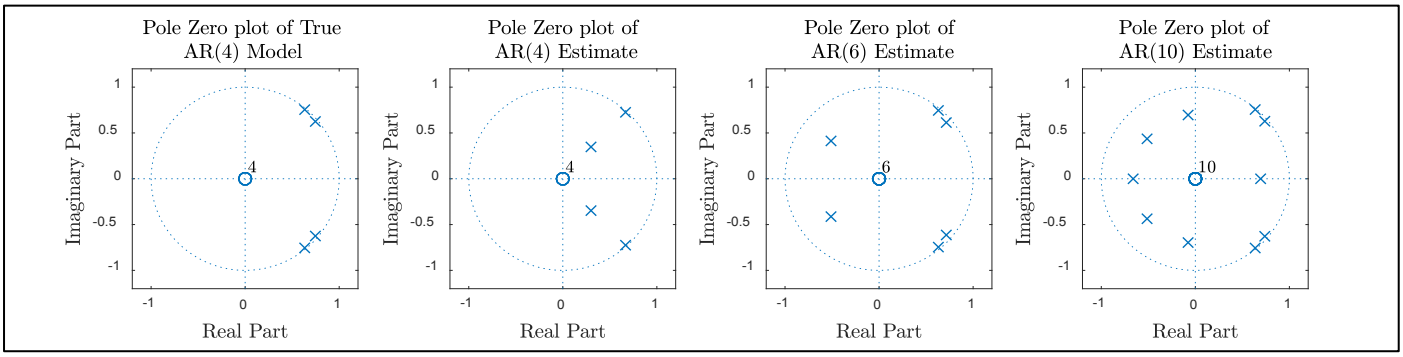

Figure 26: Pole zero plots of the true AR process and the various estimates of different orders. From this we can see that the AR(6) PSD estimate will have two clear peaks which match the true PSD since it has two complex conjugate pole pairs close to the unit circle.

2.2.c Over and Under Modelling:

Figure 27 shows the estimated spectra of the same process from section 2.2.b, but now its length N is increased to 1,000 and 10,000 samples. When $N = 10{,}000$, the AR(4) model PSD estimate closely resembles the true PSD. When the model order is too small, the corresponding spectrum is smoothed and has poor resolution. In this specific case, we know the optimal model order to be 4, and so for any $p < 4$, we expect an inaccurate spectrum. For example considering $p = 2$ in Figure 27, we see that there is only one peak regardless of how long the data is (even when $N = 10{,}000$). This is expected since a model with order less than 4 does not have sufficient orders of freedom to replicate the AR(4) process. With two coefficients, we can only ever have one pair of complex conjugate poles and hence only one spectral peak. With only one coefficient, we can only have one real pole so the spectral peak is restricted to the frequency 0 or π radians/sample.

When the model order is 200 (over-modelling), the spectrum contains spurious peaks as shown by the purple plot in Figure 27. These occur since the noise input is not perfectly white and because the ACF estimate is also not perfect. If everything were ideal, we would simply expect the extra coefficients to become 0. Another artefact of over-modelling is spectral line splitting, which is when a single peak is split into two or more distinct peaks. This has not occurred in Figure 27, and its occurrence depends on the specific white noise input. Furthermore, spectral line splitting is generally easier to observe when there is a single peak in the true spectrum, or when the peaks are more spread out.

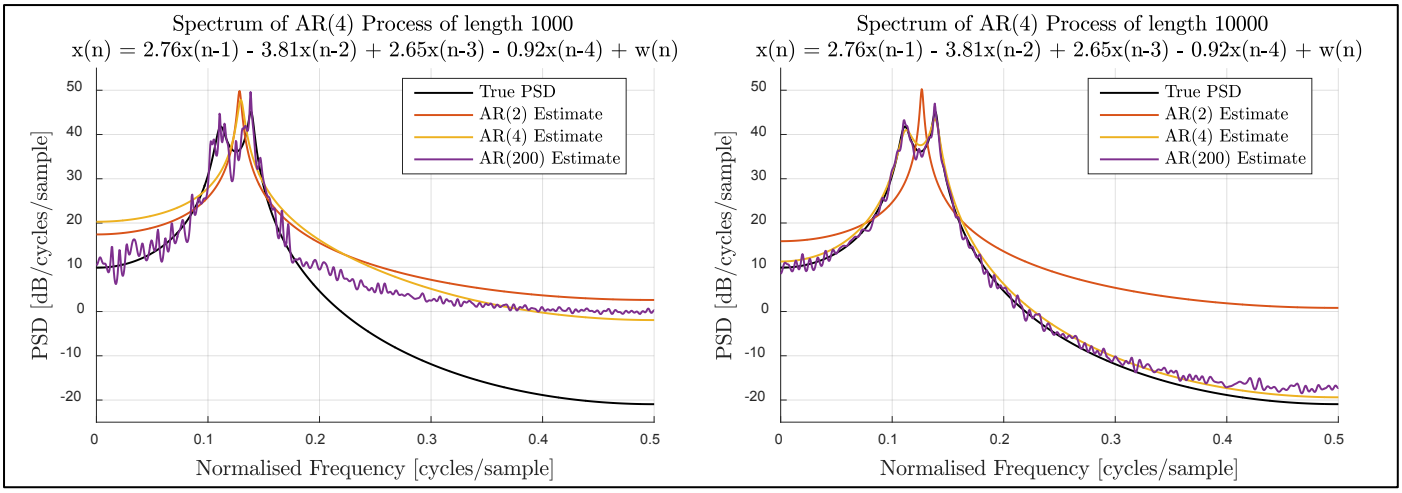

Figure 27: Parametric (AR) PSD estimate of an AR(4) process of length 1,000 and 10,000. Effects of over and under modelling are shown.

Table 2 below shows how the estimation error in the AR parameters varies with the length of the AR process. As expected a longer process results in less error since the biased ACF estimate has less variance and bias.

	$N = 1{,}000$	$N = 10{,}000$	$N = 100{,}000$	$N = 1{,}000{,}000$
Mean absolute percentage error in $\hat{\mathbf{a}}_4$:	3.29	1.11	0.0368	0.000147
Percentage Error in $\hat{\sigma}_w$:	195.8	63.2	3.33	0.512

Table 2: Shows how the estimation error in the AR parameters varies with the length (N) of the AR process.

2.3 Time-Frequency Estimation

2.3.a Time-Frequency Representation – Spectrogram:

The short time Fourier transform (STFT) allows us to see the frequency content of a signal as a function of time; something which is essential for analysing non-stationary processes. The MATLAB function `spectrogram` shows the PSD estimate for overlapping segments of data, and hence shows a time localized estimate for the PSD at various time instants. In this exercise, various spectrogram parameters (window length, window type, overlap factor and FFT length) are individually investigated in order to produce an optimal spectrogram of a quadratic frequency sweeping chirp signal sampled at 1 KHz.

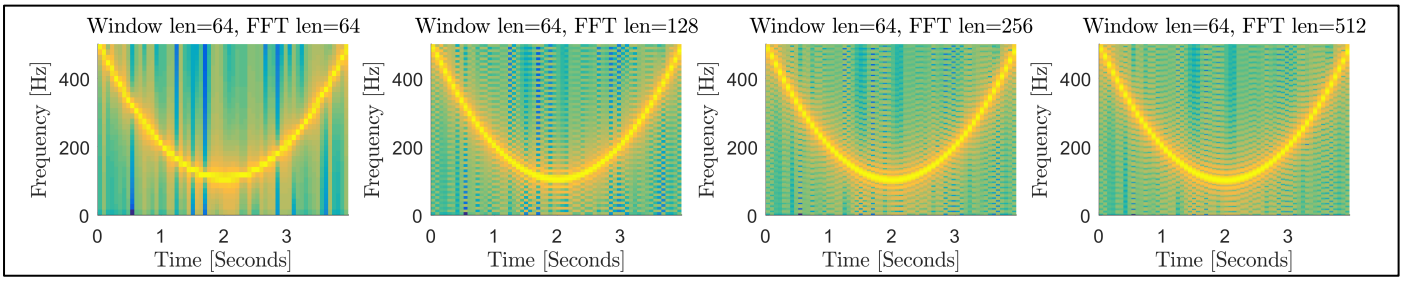

Figure 28: Effect of zero padding. A rectangular window of length **64** with no overlap is used for all spectrograms to allow a fair comparison.

Figure 28 above shows that increasing the FFT length (zero padding) increases the smoothness of the frequency spectrum for a given time instant. This is because it increases the number of frequency bins. Beyond a certain limit (256 in this particular case) the additional zero padding provides no extra benefit. Note that zero padding does not increase spectral resolution since this is dependent on the window type and length. Figure 29 below shows how the frequency resolution increases with window length, i.e. there is greater contrast in the spectrogram colours when the window length is longer meaning the spectral frequency peak is more distinct. However, increasing the window length reduces time resolution, and this can be seen by the larger and more distinct steps in the time axis for larger window lengths. This compromise is related to the Gabor uncertainty principle ($\sigma_t \sigma_f \geq 1/4\pi$) or the Heisenberg uncertainty principle in a more general context. The Gabor limit states that the product of the standard deviations in frequency and time are greater a constant ($1/4\pi$).

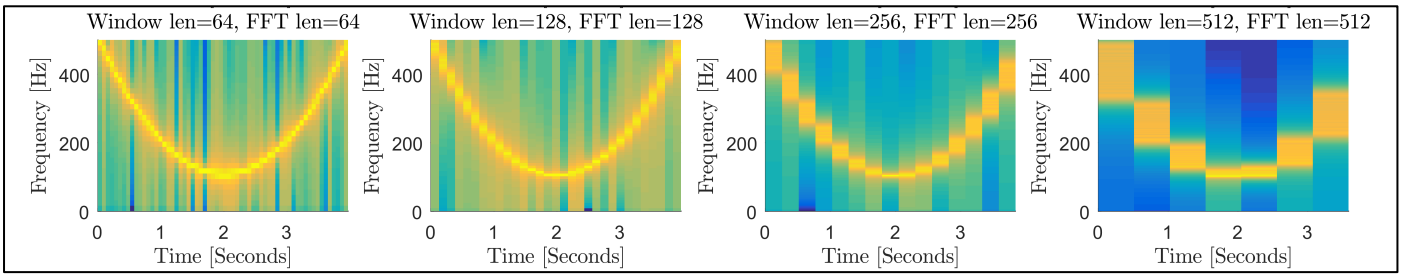

Figure 29: Effect of window length. A rectangular window with no overlap is used for all spectrograms to allow a fair comparison.

The large steps in the time axis due to large window lengths can be smoothed by increasing the overlap between the windows as shown by Figure 30 below. More overlap means a smoother periodogram and gives the appearance of more time resolution just like zero padding gives the appearance of more frequency resolution. However, it also means more computation, and actual time resolution is determined by window length. In the particular case of Figure 30, it was found that overlapping beyond 80% (205 samples) had negligible additional benefit.

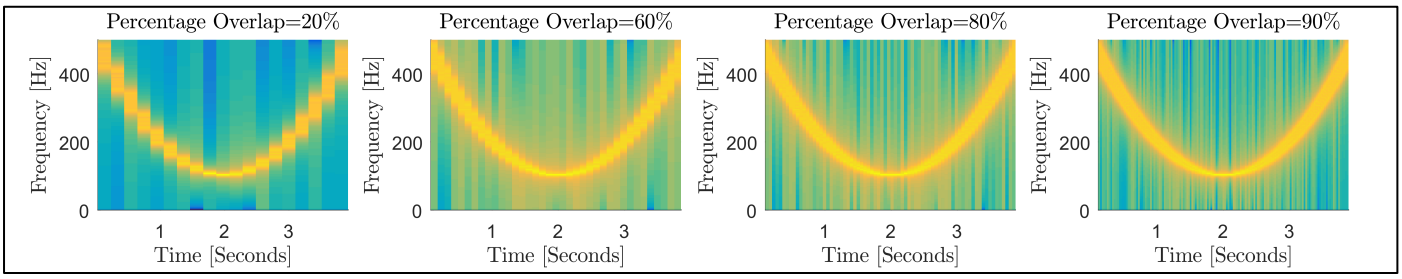

Figure 30: Effect of overlap. A rectangular window of length **256** with no zero padding is used for all spectrograms to allow a fair comparison

Figure 31 shows the effects of different window functions. It has been produced using the optimal combination of parameters based on experimentation and the discussion above: Window length = 256, FFT length = 2048, percentage overlap = 90%.

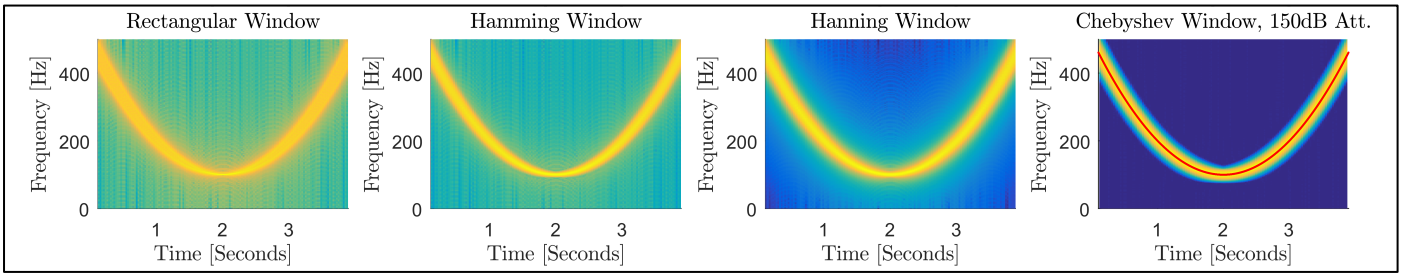

Figure 31: Effects of different window functions. Optimal parameters are used: Window length = 256, FFT length = 2048, percentage overlap = 90%. The rightmost figure has an overlay plot (in red) of the actual chirp frequency sweep.

Using a smoother window compared to the rectangular one reduces spectral leakage at the expense of spectral smearing. In Figure 31, the Chebychev window with −150dB sidelobes has the smallest leakage and his is shown by the greatest colour contrast. However, its mainlobe is wider than the Hanning window so the yellow line is slightly thicker (more smearing). Since

there is only one frequency at any given time instant, leakage and smearing are not too important; i.e. we can still determine the frequency easily regardless of the window. However, a window with smaller sidelobes produces a more visually appealing spectrogram, and for this reason the Hanning and Chebyshev windows are considered to be most suitable. Note that the red line overlaid on the rightmost plot shows the actual chirp frequency.

2.3.b Spectrogram of Real-World EEG Data:

From the experimentation in section 1.4.b we found that at least a 1 second long window is necessary to see the SSVEP spectral peak, and that a window of 2.5 or more seconds results in a well-defined SSVEP peak. Based on this result and a sample frequency of 1.2 KHz, the very minimum FFT length should be 1200 samples. A more theoretical method to find the minimum window length makes use of the 3dB bandwidth of the window function. E.g. a Hanning window has a 3dB bandwidth of $1.44 \times \pi/N$, which implies we need more than 864 samples to distinguish frequencies separated by 2 Hz, i.e. the alpha rhythm response at $8 - 11$ Hz and the SSVEP at 13 Hz. Through more experimentation, it is found that a length of 4096 provides a better balance between spectral and time resolution. A Hanning window is best, and an overlap factor greater than 75% provides almost no extra benefit while increasing computation time. The same goes for a FFT length more than 16384 samples. Therefore these values are chosen as the spectrogram parameters and the result is shown by Figure 32 below.

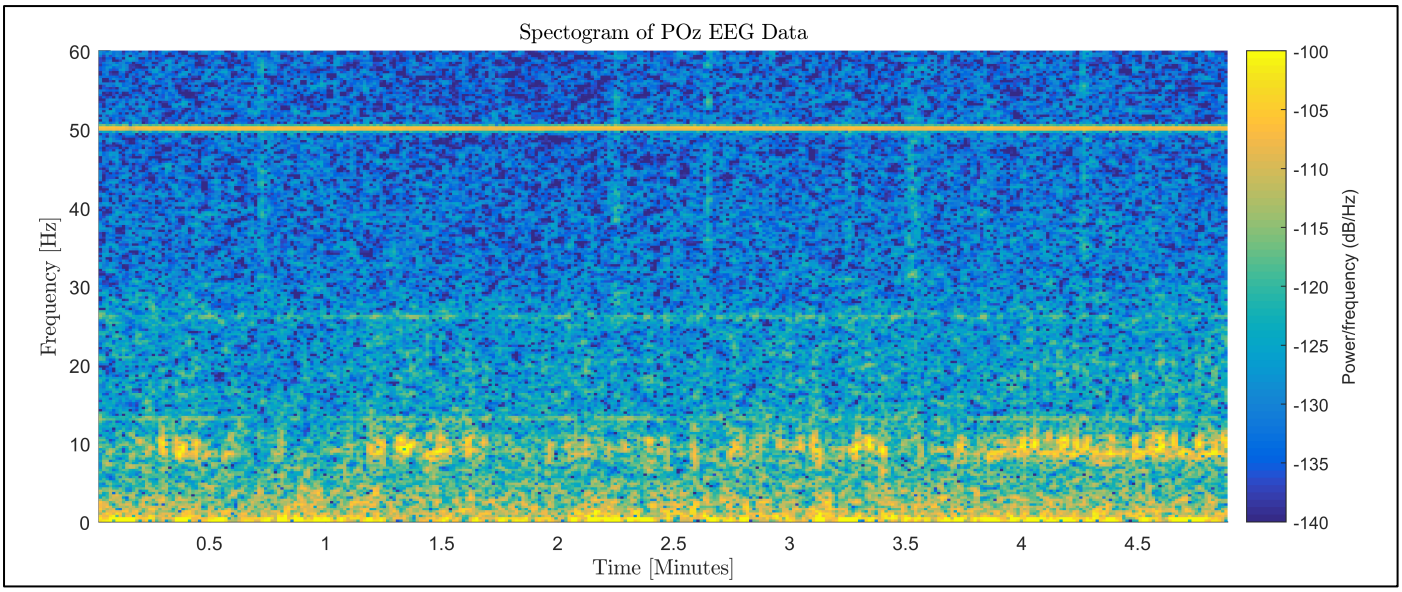

Figure 32: POz EEG spectrogram. Hanning window of length 4096 (3.41 seconds), overlap factor = 75% (3072 samples), FFT length = 16384.

From Figure 32 we can see the SSVEP at 13 Hz, and the second harmonic at 26 Hz. Unlike when we used the averaged periodogram method in section 1.4.b, it is not possible to see the third harmonic at 39 Hz. This could be due to less averaging, or the fact that it is harder to distinguish colours in a spectrogram compared to finding peaks in periodogram line plots.

The spectrogram shows that the SSVEP is not consistent with time, for example it is low amplitude (or non-existent) around a time of 1 minute. The strong and constant 50 Hz mains interference can also be seen as well as an intermittent alpha rhythm spectral activity at around $8 - 11$ Hz. There is high signal power at low frequencies, and this is a common artefact of biological signals since biological signal noise power is inversely proportional to frequency.

2.4 Real World Signals: Sinus Arrhythmia from RR-Intervals

2.4.a PSD Estimation of RRI Data:

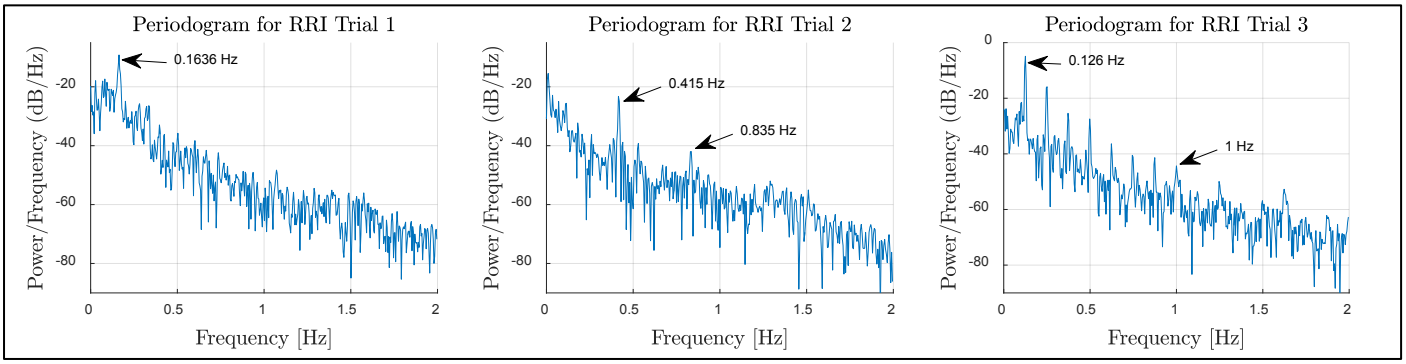

Figure 33: Periodogram of the three RRI trials. A Hanning window is used to reduce spectral leakage.

Figure 33 above shows the periodograms for the three RRI trials. Spectral peaks and harmonics are visible, but there is a lot of noise. The variance of the periodogram is reduced by averaging windowed segments as shown by Figure 34. From this it is easier to identify the spectral peaks, (for the cases when the window is 100 or 150 samples long), some of which have been pointed

out by the arrows. When the window is too small (50 samples), there is not enough spectral resolution to easily identify the fundamental peaks, especially for trial 1. Also with a 50 sample long window, the harmonics are not clearly visible in any of the three trials. The periodograms obtained using Welch's method with 80% overlap show even less noise in the PSD estimate as theoretically expected. Trial 3 appears to be the most successful since we can see clearly see the fundamental frequency and all the way up to the 8^{th} harmonic.

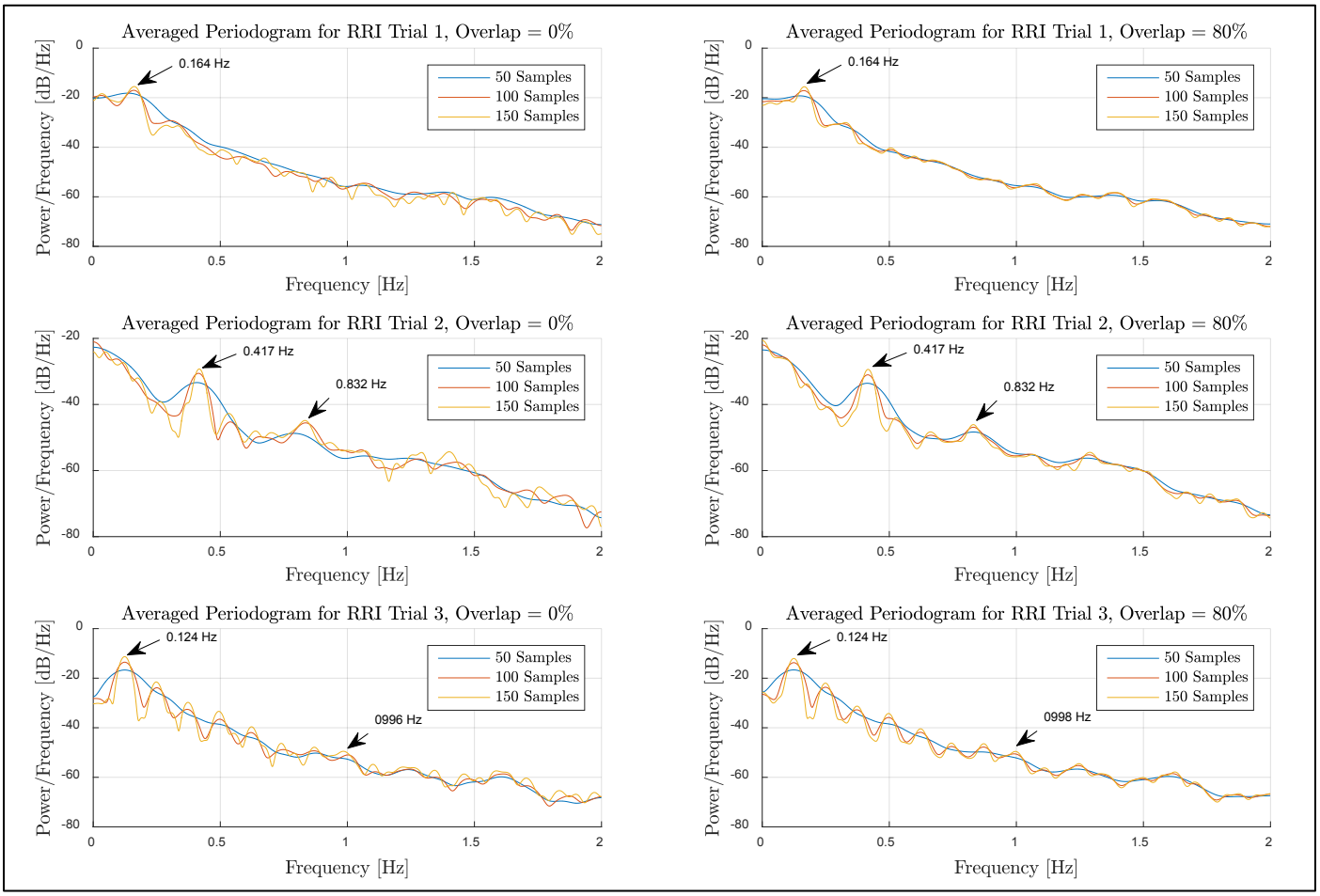

Figure 34: Averaged Periodogram with different Hanning window lengths; 50, 100, 150 samples, and 0% or 80% overlap.

2.4.b Explanation of the RRI Trials:

Respiratory sinus arrhythmia (RSA) is a naturally occurring modulation in heart rate due to breathing. The heart rate increases during inhalation and decreases during exhalation. For trial one, the breathing rate was unconstrained. According to the periodograms in section 2.4.a, this heart rate modulation occurred at a rate of $0.164\ Hz$ which means my natural breathing rate is $0.164 \times 2 \times 60 = 19.7$ breaths per minute (BPM). For trial 2, the breathing rate was controlled to $50\ BPM$ or $0.4167\ Hz$ by a metronome. There is evidence of a spectral peak at this frequency in both the averaged and non-averaged periodograms. We expect harmonics since this modulation is not a perfect sinusoid, (it is closer to a triangular wave), and the second harmonic can be seen in both figures. For trial 3, the rate was controlled to $15\ BPM$ or $0.125\ Hz$. Looking at Figure 34 (100 or 150 sample long window) we can see the spectral peak which corresponds to this. We can also clearly see up to the 8^{th} harmonic which we expect to occur at $1\ Hz$. Since the signal is biological, the noise is proportional to $1/f$. This is why the periodogram is generally downward sloping and has high power at low frequencies.

2.4.c AR Spectrum Estimate:

An AR model is best suited to model the RRI data compared to a moving average (MA) model. This is implied because the unbiased ACF estimate for the three trials does not decay to zero, and the periodogram of the RRI data is peaky. A MA model would struggle to model peaky spectra since it has an all zero transfer function, and the unbiased ACF of a MA process is zero for lag greater than model order. Initial investigations to determine a suitable model order for the three trials are first conducted using the partial autocorrelation function (PACF) (Figure 35), the minimum description length criterion (MDL) and Akaike information criterion (AIC) (Figure 36).

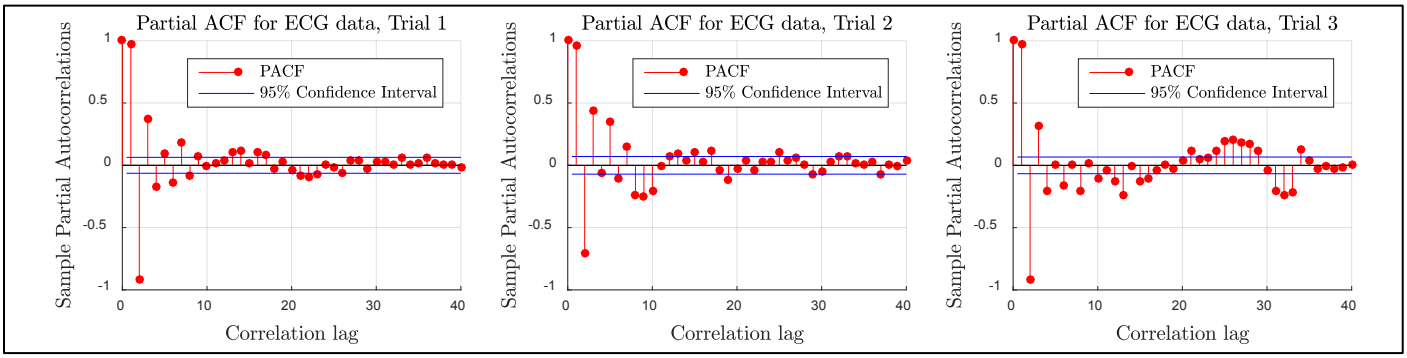

Figure 35: Sample PACF of the RRI data from the three trials and 95% confidence interval.

The PACF for correlation lags greater than p are normally distributed with $\mu = 0$ and $\sigma^2 = 1/N$. This can therefore be used to help determine a model order. For example the required order for trial 3 seems to be 34 since the PACF values for lags greater than this all lie within 95% confidence bounds. In general, the higher model orders result in more accuracy (the error is a monotonically non-increasing function of the model order), but there are some issues with over-modelling as discussed in section 2.2.c. Furthermore, in reality there is usually a trade-off between model accuracy and computational complexity. The Akaike information criterion (AIC) and the minimum description length criterion (MDL) consider modelling error and also introduce a penalty for high orders. According to these criterions, the optimal order for trial 1 is 2, and for trial 2 is 10, and for trial 3 is 34 since this is the order at which the MDL and AIC are minimized.

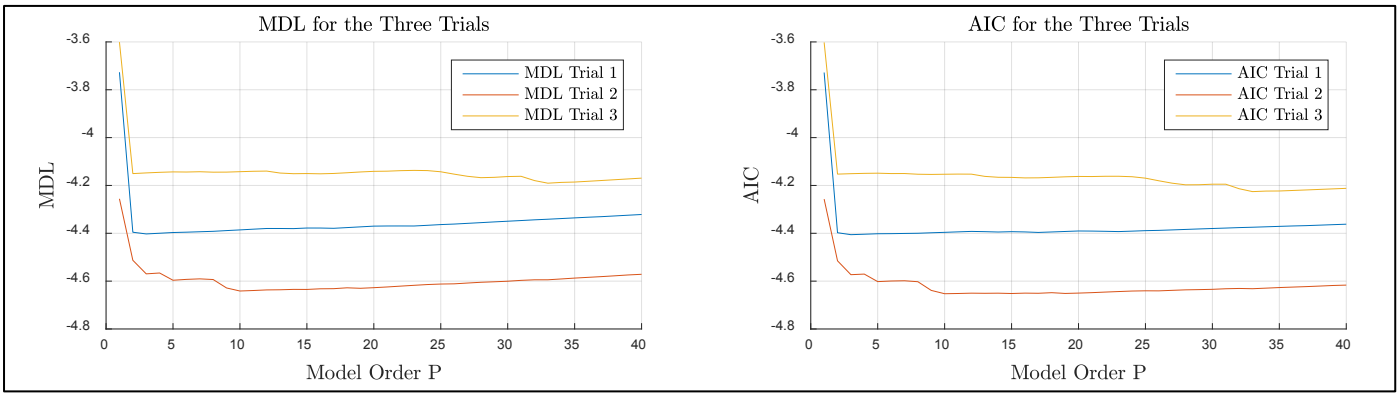

Figure 36: MDL and AIC criterion for the RRI data from the three trials.

Figure 37 below shows an AR model for each of the three trials. The model order chosen is the smallest such a peak in the spectrum which corresponds to the respiration rate first starts to develop. From this we can see that the AR based estimates are smoother than the welch periodograms. Another difference is that the peak frequency for the AR estimates is slightly different from the theoretical expected values. This simply due to the low resolution of the AR model based PSD estimate when the model order is low.

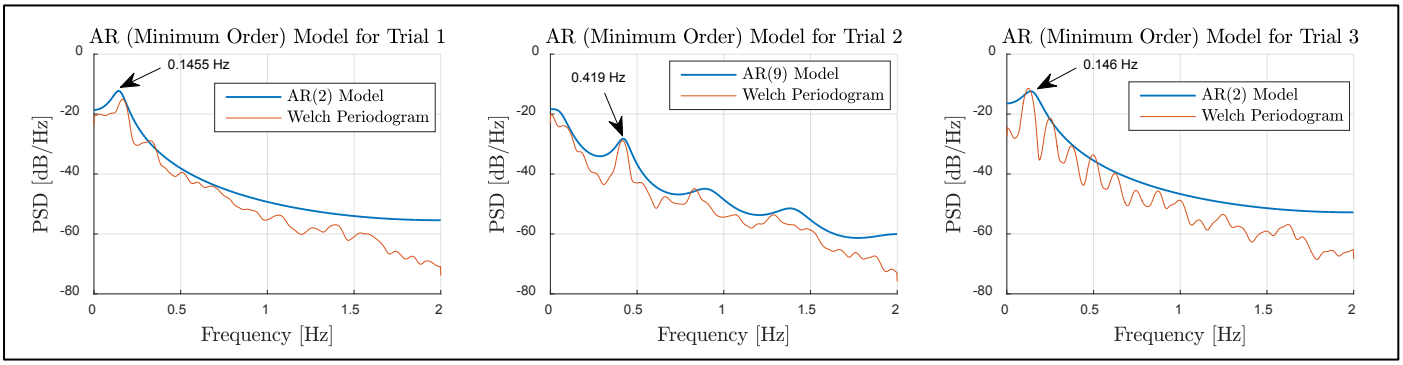

Figure 37: AR modelling of the RRI data for the three trials. These plots show the minimum order model required until a peak corresponding to the RSA first appears. The Welch method is used for the periodogram with a Hanning window of length 128, and an overlap of 80% and an FFT length of 1024.

Figure 38 below shows higher order AR models based PSD estimates whose orders have been determined by considering the PACF, AIC and MDL criteria. These higher order spectra are peakier and less noisy compared to the welch periodograms. Additionally, the peaks are narrower and more precise, for example the main fundamental peak for trial 3 occurs exactly at $0.125\ Hz$ as expected based on the known breathing rate. This is because the periodogram is of a MA type and so it struggles to model peaky spectra unless the data sequence is long. The PSD estimate based on the MA(34) model of trial 3 actually allows us to confidently determine up to the 12^{th} harmonic compared to only the 8^{th} when using the periodogram based method. The final main difference is that the model estimates are slightly higher up than the periodogram based method, but this is only because the Hanning window is used for the periodogram based method and this removes some of the signal power.

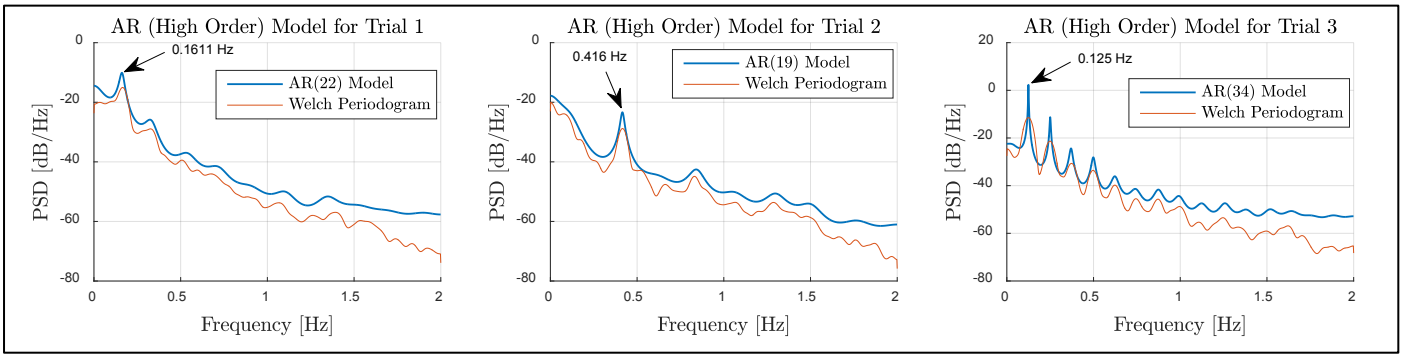

Figure 38: AR modelling of the RRI data. The model orders are determined from consideration of the PACF, AIC and MDL criteria. The Welch method is used for the periodogram with a Hanning window of length 128, and an overlap of 80% and an FFT length of 1024.

3 Adaptive Signal Processing

3.1 The Least Mean Square (LMS) Algorithm

3.1.a Correlation Matrix and LMS Convergence Criteria:

$$x(n) = a_1 x(n-1) + a_2 x(n-2) + \eta(n), \qquad \eta(n) \sim \mathcal{N}(0, \sigma_\eta^2), \qquad a_1 = 0.1,\ a_2 = 0.8,\ \sigma_\eta^2 = 0.25 \tag{3.1}$$

The autocorrelation matrix ($\mathbf{R}_{xx}$) for the AR(2) process as described by equation 3.1 above is derived as follows:

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

The last step of 3.2 follows from the fact that the process is wide sense stationary (WSS) so it's ACF depends only on the relative time difference. Now considering equation 3.1 and multiplying in turn by $x(n-k)$ for $k \in \{0,1,2\}$, and then taking the expectation results in the three following equations:

$$r_{xx}(0) = a_1 r_{xx}(1) + a_2 r_{xx}(2) + \sigma_\eta^2, \qquad r_{xx}(1) = a_1 r_{xx}(0) + a_2 r_{xx}(1), \qquad r_{xx}(2) = a_1 r_{xx}(1) + a_2 r_{xx}(0) \tag{3.3}$$

These three equations can be written in compact matrix form and solved as follows:

$$\begin{bmatrix} 1 & -a_1 & -a_2 \\ -a_1 & 1-a_2 & 0 \\ -a_2 & -a_1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} r_{xx}(0) \\ r_{xx}(1) \\ r_{xx}(2) \end{bmatrix} = \begin{bmatrix} \sigma_\eta^2 \\ 0 \\ 0 \end{bmatrix} \Rightarrow \begin{bmatrix} 1 & -0.1 & -0.8 \\ -0.1 & 0.2 & 0 \\ -0.8 & -0.1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} r_{xx}(0) \\ r_{xx}(1) \\ r_{xx}(2) \end{bmatrix} = \begin{bmatrix} 0.25 \\ 0 \\ 0 \end{bmatrix} \Rightarrow \begin{bmatrix} r_{xx}(0) \\ r_{xx}(1) \\ r_{xx}(2) \end{bmatrix} = \begin{bmatrix} 25/27 \\ 25/54 \\ 85/108 \end{bmatrix} \tag{3.4}$$

Therefore the autocorrelation matrix and its eigenvalues are given by equation 3.5. Table 3 summaries the convergence requirements for the LMS algorithm in an AR configuration:

$$\mathbf{R}_{xx} = \begin{bmatrix} 25/27 & 25/54 \\ 25/54 & 25/27 \end{bmatrix}, \qquad \lambda_1 = \frac{25}{18}, \quad \lambda_2 \frac{25}{54} \tag{3.5}$$

	Definition:	*Requirement:*	*Evaluated Requirement:*
Convergence in mean	$E\{\mathbf{w}(n) - \mathbf{w}_{opt}\} \xrightarrow{n\to\infty} 0$	$0 < \mu < (2/\lambda_{max})$	$0 < \mu < 1.44$
Convergence in mean square	$E\left\{(\mathbf{w}(n) - \mathbf{w}_{opt})^2\right\} \xrightarrow{n\to\infty} J_{min} + J_{ex}(\infty)$	$\mu \sum_{k=1}^{p} \frac{\lambda_k}{1-\mu\lambda_k} < 1$	$0 < \mu < 0.325$

Table 3: Summary of the convergence criteria for the adaptation gain, μ, for the LMS algorithm for AR identification. $\mathbf{w}$ denotes the AR weight vector. J_{min} denotes the minimum MSE, and $J_{ex}(\infty)$ denotes the excess steady state MSE which occurs due to gradient noise and a non-zero fixed step size which cause the weights to fluctuate around their steady state values.

3.1.b Learning Curves for LMS:

Figure 39 below shows the learning curve of the LMS filter for the AR(2) process given by equation 3.1 for different adaptation gains. For both values of μ we see convergence of the error power to be close to the theoretical minimum which is given by the variance of the noise input $\left(10 \times log10(\sigma_\eta^2) = -6.02dB\right)$. the bottom plot of Figure 39 shows the ensemble average of 100 independent trials. From this it is easier to see that the smaller adaptation gain takes around 200 iterations to converge, whereas the larger gain takes around 80 iterations. It is not clear in Figure 39, but the smaller the step, the smaller the excess MSE. This is the MSE in excess of the theoretical minimum, and this is investigated further in the section 3.1.c. In summary, there is a trade-off between speeds of convergence and steady state MSE. A final point to note is that a large gain can result in overshoot or even instability.

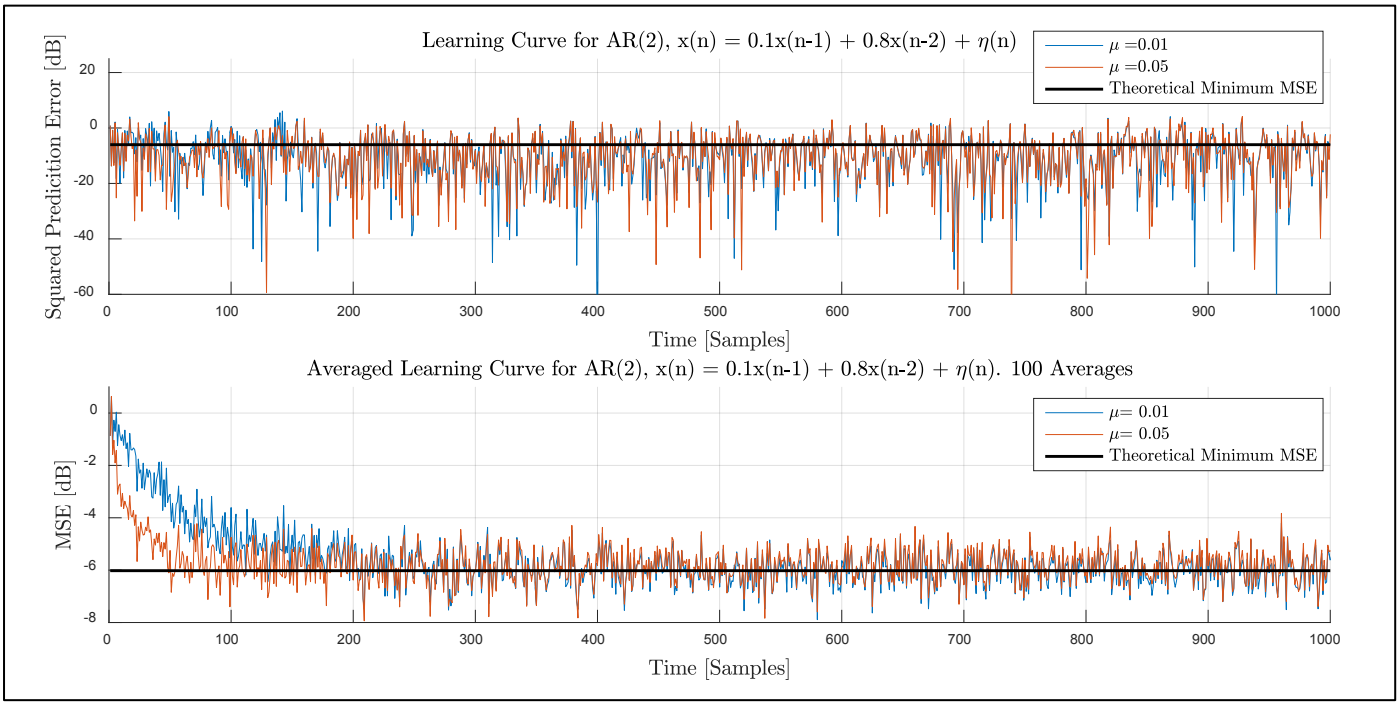

Figure 39: Learning curves of the AR(2) process given by equation 3.1. The lower plot is an average of 100 independent trials.

3.1.c Excess MSE and Missadjustment:

The experimental steady state value of the ensemble averaged squared error is shown in Table 4 below. These experimental averages have been computed by time averaging the last 1000 samples of a 2000 sample long ensemble averaged error squared vector. 100 Independent trials are used for the ensemble averaging. Table 4 shows that the experimental misadjustments agree with the theoretical missadjustment computations. It is clear that a larger gain results in a larger excess mean square error (EMSE). Intuitively this is because the prediction error, $x(n) - \hat{x}(n)$, can never be zero due to non-zero σ_η^2. As a consequence even when the mean of $\mathbf{w}(n)$ converges $\mathbf{w}_{opt}$, the time varying weights will continue to oscillate around the true values. When μ is larger, each step in the direction of (instantaneous) steepest decent is greater, and so the oscillation of the weights around the true values is more erratic. This results in a larger steady state error variance.

Adaptation Gain (μ)	*Experimental MSE*: $E\{(\hat{x}(n) - x(n)^2)\}$	*Experimental Excess MSE*: $\text{MSE} - \sigma_\eta^2$	*Experimental Missadjustment*: $\frac{\text{MSE} - \sigma_\eta^2}{\sigma_\eta^2}$	*Approximate Theoretical Missadjustment (for small μ)*: $\frac{\mu}{2} Tr(R_{xx}) = \frac{25\mu}{27}$	*Exact Theoretical Missadjustment*: $\sum_{k=1}^{p} \frac{\mu\lambda_k}{2 - \mu\lambda_k}$
0.01	0.2528	0.0028	0.0111	0.009259	0.009313
0.05	0.2639	0.0139	0.0557	0.046296	0.047681

Table 4: Experimental and theoretical missadjustment. The experimental averages have been computed by time averaging the last 1000 samples of a 2000 sample long ensemble averaged error squared vector. 100 Independent trials are used for the ensemble averaging.

3.1.d Steady State Coefficient Error:

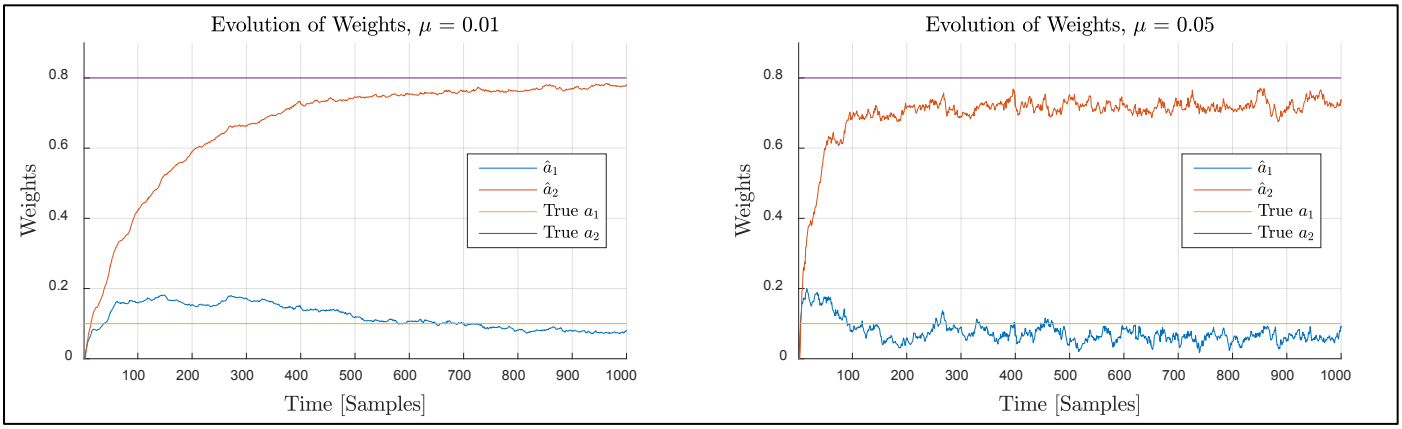

Figure 40: Evolution of filter weights (coefficients) for two different adaptation gains.

Figure 40 shows the time evolution of the coefficients for different adaptation gains. With a larger learning rate we expect the coefficient rise time to be smaller, and the coefficient steady state error variance to be higher. This is because there is a larger correction to the weights during each iteration of the algorithm. Once the weights reach the steady state values, there is still a source of error (since this is an AR configuration) and so the weights are still adapting. With a larger step this adaptation is more significant, and so the weights would oscillate more erratically around the true coefficients hence resulting in a larger error variance. Interestingly in this particular case we do not actually get convergence to the weights regardless of how small μ is. This is likely because $x(n)$ (eqn 3.1) is very close to instability; the Z-plane poles are given by $p_1 = (0, 0.948)$ and $p_2 = (0, -0.846)$, and these are close to the unit circle. It could be that with a large adaptation gain the adaptive filter has difficulty converging close to poles close to the unit circle due to stability issues. This explains why steady state error increases with μ. It was found that this is not the case when a more stable system is identified, e.g. $\mathbf{a} = [1, -0.1, -0.2]^T$. For this process the steady state mean weight error is zero for both values of μ as expected.

We experience a significant overshoot in coefficient a_1 even for small μ, and this is likely a consequence of the poles being close to the unit circle. The percentage overshoot for a_1 is the same for both adaptation gains and is around 95%. Table 5 shows the steady state coefficients and steady state value of the mean squared coefficient error (MSCE) in dB. The steady state coefficients are obtained by ensemble averaging 100 independent trials of length 1000 and then time averaging the last 200 (steady state) values of the coefficients. Figure 41 below shows the MSCE and MSE for a wider range of adaptation gains. It shows instability to occur when the step is too high.

Adaptation gain μ	*Steady state* a_1	*Steady state* a_2	*MSCE*: $10log_{10}(\|a - \hat{a}\|^2)$
0.01	0.0889	0.7692	$-25.25\ dB$
0.05	0.0673	0.7172	$-17.22\ dB$

Table 5: Steady state adaptive filter coefficients obtained by ensemble averaging 100 independent trials of length 1000 and then time averaging the last 200 (steady state) values.

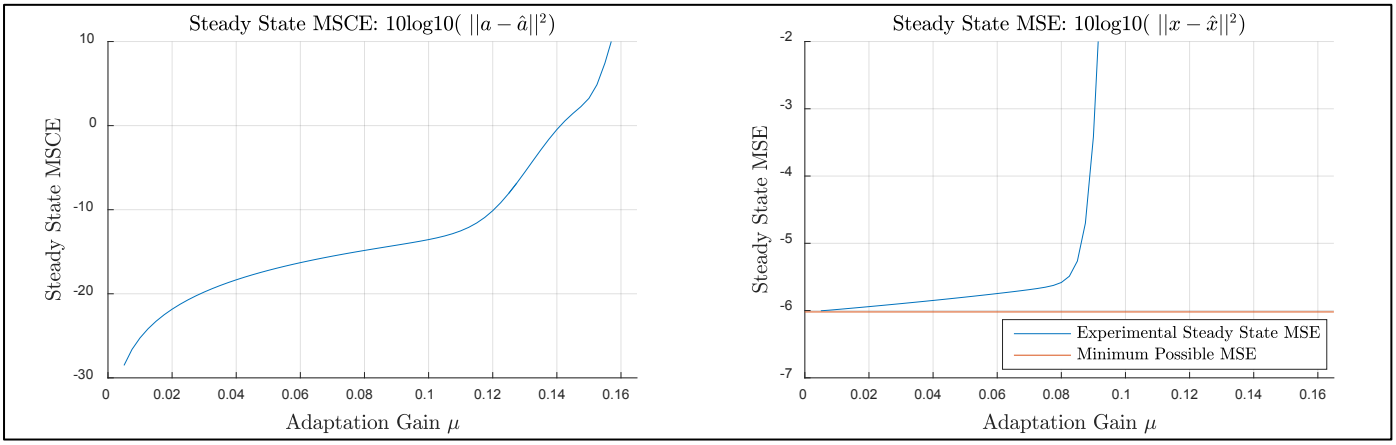

Figure 41: steady state MSCE and MSE for different adaptation gains. Obtained by ensemble averaging 100 independent trials of length 5000 and then time averaging the last 1000 (steady state) values

3.1.e Leaky LMS Derivation:

The leaky LMS algorithm minimises the leaky cost function $J_2(n)$ which is given by equation 3.7. To derive the coefficient update equation (3.10) for the leaky LMS we make use of the update equation for the method of steepest decent (3.6). This in turn requires the anti-gradient of $J_2(n)$ (3.9). The proof is as follows:

The update equation for the method of steepest descent:	$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu\left(-\nabla_w J_2(n)\|_{\mathbf{w}_n}\right)$	3.6
In which $J_2(n)$ *is the leaky cost function*:	$J_2(n) = 0.5(e^2(n) + \gamma\|\mathbf{w}(n)\|_2^2)$	3.7
In which the error square term is:	$e^2(n) = \left(x(n) - \hat{x}(n)\right)^2 = \left(x(n) - \sum_{m=1}^{p} w_m(n)x(n-m)\right)^2$	
Define $\mathbf{x}(n) = [x(n-1), \dots, x(n-p)]^T$:	$e^2(n) = \left(x(n) - \mathbf{w}^T(n)\mathbf{x}(n)\right)^2$	3.8
Substitute 3.8 *into* 3.7	$J_2(n) = 0.5\left(\left(x(n) - \mathbf{w}^T(n)\mathbf{x}(n)\right)^2 + \gamma\|\mathbf{w}(n)\|_2^2\right)$	
$Finding\ the\ the\ anti-gradient$:	$-\nabla_w J_2(n)\|_{\mathbf{w}_n} = -0.5\left(\nabla_w\left(x(n) - \mathbf{w}^T(n)\mathbf{x}(n)\right)^2 + \gamma\nabla_w\|\mathbf{w}(n)\|_2^2\right)$	
Chain rule to find $\nabla_{\mathbf{w}}\left(x(n) - \mathbf{w}^T(n)\mathbf{x}(n)\right)^2$:	$-\nabla_w J_2(n)\|_{\mathbf{w}_n} = -0.5\left(2\left(-\mathbf{x}(n)\right)\left(x(n) - \mathbf{w}^T(n)\mathbf{x}(n)\right) + \gamma\nabla_w\|\mathbf{w}(n)\|_2^2\right)$	
use $\nabla_{\mathbf{w}}\|\mathbf{w}(n)\|_2^2 = 2\mathbf{w}(n)$:	$-\nabla_w J_2(n)\|_{\mathbf{w}_n} = -0.5\left(2\left(-\mathbf{x}(n)\right)\left(x(n) - \mathbf{w}^T(n)\mathbf{x}(n)\right) + 2\gamma\mathbf{w}(n)\right)$	

Simplify expression:	$-\nabla_w J_2(n)\|_{\mathbf{w}_n} = \mathbf{x}(n)e(n) - \gamma\mathbf{w}(n)$	3.9
Substituting 3.9 *into* 3.6:	$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu\big(\mathbf{x}(n)e(n) - \gamma\mathbf{w}(n)\big)$	
Final result:	$\mathbf{w}(n+1) = (1-\mu\gamma)\mathbf{w}(n) + \mu e(n)\mathbf{x}(n)$	3.10

3.1.f Leaky LMS Implementation:

The leaky LMS algorithm minimises a cost function which includes an additional penalty term which is proportional to the norm of the weight vector. The name 'leaky' comes from the fact that once the input $\mathbf{x}$ is stopped, the weight estimates leak and exponentially decay to zero. When the input signal has an autocorrelation matrix with zero eigenvalues, one or more weights do not converge. The leaky LMS algorithm introduces a leakage coefficient and essentially forces convergence. It practice it has been shown to improve stability in a finite precision implementation and reduce undesirable effects like stalling and bursting. However the leaky LMS is biased. Comparing the leaky solution (eqn 3.11) to the Wiener solution, $\mathbf{w}_{opt} = \mathbf{R}^{-1}\mathbf{p}$, we see that the leaky LMS can be interpreted as adding zero-mean white noise with autocorrelation matrix $\gamma\mathbf{I}$ to the input $\mathbf{x}$.

$$\mathbf{w}_{leaky} = (\mathbf{R} + \gamma\mathbf{I})^{-1}\mathbf{p}\,, \quad where\ \mathbf{R} = E\{\mathbf{x}(n)\mathbf{x}^T(n)\}, \quad \boldsymbol{p} = E\{\mathbf{d}(n)\mathbf{x}(n)\}, \quad (d = desired\ output, x = input) \qquad 3.11$$

Figure 42 below shows how the steady state weights of the leaky LMS adaptive filter changes with the leakage coefficient, and it clearly illustrates the bias which increases as the leakage coefficient increases as predicted by equation 3.11. The steady state averages have been obtained by ensemble averaging 100 realisations each of length 2000, and then time averaging the last 1000 samples. Figure 43 shows the evolution of the weights and also illustrates the increased steady state error. The leaky algorithm could be improved if the leakage coefficient was time varying and allowed to reduce as the weights approach the steady state.

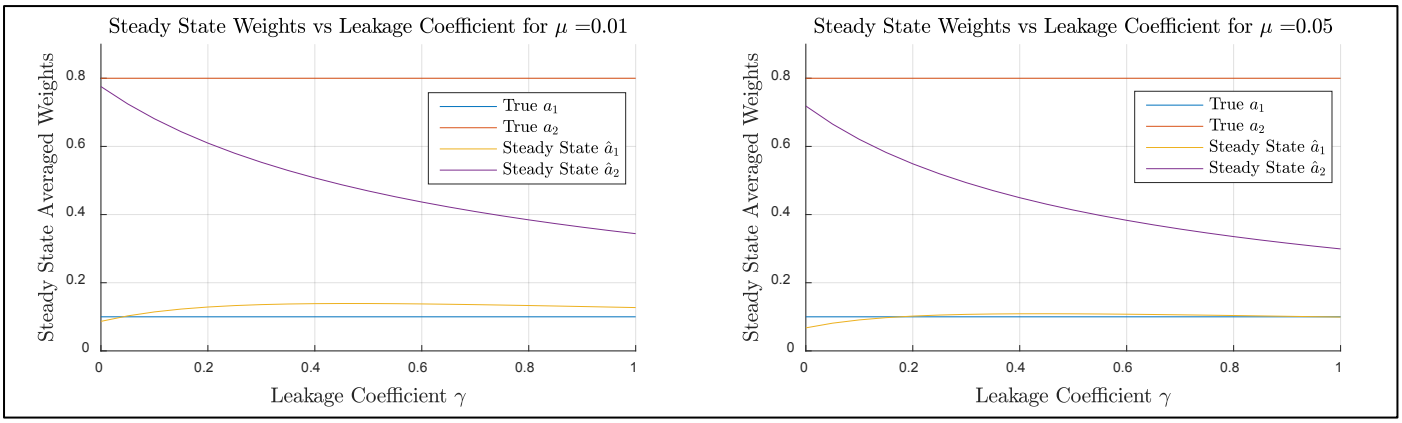

Figure 42: Steady state adaptive weights vs. leakage coefficient. The steady state values have been obtained by ensemble averaging 100 realisations each of length 2000, and then time averaging the last 1000 samples.

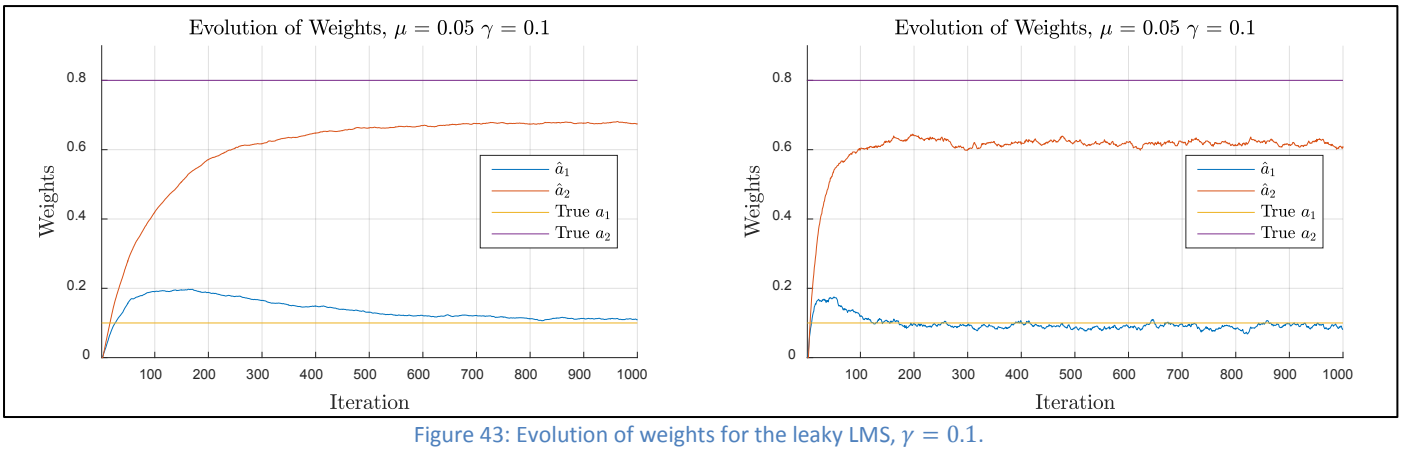

Figure 43: Evolution of weights for the leaky LMS, $\gamma = 0.1$.

3.2 Adaptive Step Sizes

3.2.a Gradient Adaptive Step-Size (GASS) Algorithms:

The standard LMS uses a fixed step size and hence there is a trade-off between speed of convergence and steady state error variance as previously discussed. Ideally we wish to have a large step when the error is large and then reduce the step size as the estimate approaches the true values, thus allowing for both fast convergence and small steady state error. Variable step size algorithms aim to simultaneously minimize the cost function with respect to the weights as well as minimising the cost function with respect to the step size. The update equation for the step size is given by equation 3.12 in which $\boldsymbol{\psi}(n)$ can take various forms as shown by equations 3.13 to 3.15.

Step size update equation: (ρ *is the step size learning rate*)	$u(n+1) = u(n) - \rho\nabla_{\mu}J(n)\big\|_{\mu=\mu(n-1)}$ $= u(n) - \rho e(n)\mathbf{x}(n)\boldsymbol{\psi}(n)$	3.12
Benveniste:	$\boldsymbol{\psi}(n) = \underbrace{[\mathbf{I} - (\mu(n-1) \times \mathbf{x}(n-1)\mathbf{x}^T(n-1))]}_{Low\ pass\ filtering\ term}\boldsymbol{\psi}(n-1) + e(n-1)\mathbf{x}(n-1)$	3.13
Ang & Farhang:	$\boldsymbol{\psi}(n) = \alpha\boldsymbol{\psi}(n-1) + e(n-1)\mathbf{x}(n-1), \quad \alpha \in (0,1)$	3.14
Matthews & Xie:	$\boldsymbol{\psi}(n) = e(n-1)\mathbf{x}(n-1)$	3.15

The term in the square brackets for the Benveniste algorithm represents a time varying low pass filter which smoothes the instantaneous gradient $e(n-1)\mathbf{x}(n-1)$ and hence increases robustness to statistical variations in the input. 3.14 is a simplification in which the low pass filter term is fixed and 3.15 is the same as 3.14 for the case when $\alpha = 0$, i.e. the filter time constant is zero and so there is no smoothing. This is the least robust meaning a small step (ρ) will be necessary but its advantage is in its low computational complexity.

Figure 44 below shows the performance each of the GASS algorithms with various parameters ρ and α. The initial adaptation gain $\mu(0)$ is set to 0 for all the algorithms in order to allow for a fair comparison and the average of 100 independent trials is used to obtain each plot (averaging reduces variance). Figure 44 shows that a larger ρ means faster convergence. This is because a larger ρ allows faster convergence of $\mu(n)$ to optimal values and hence faster convergence of the weights. However, a large value of ρ can result in non-convergence since it can cause $\mu(n)$ to be too large which causes instability. The third row of plots in Figure 44 below shows that a larger α results in faster convergence for the Ang & Farhang algorithm. This is because it allows $\mu(n)$ to increase faster from 0 to its optimal value. It was found that having a large α reduces the ability to have a large ρ so there is a trade-off.

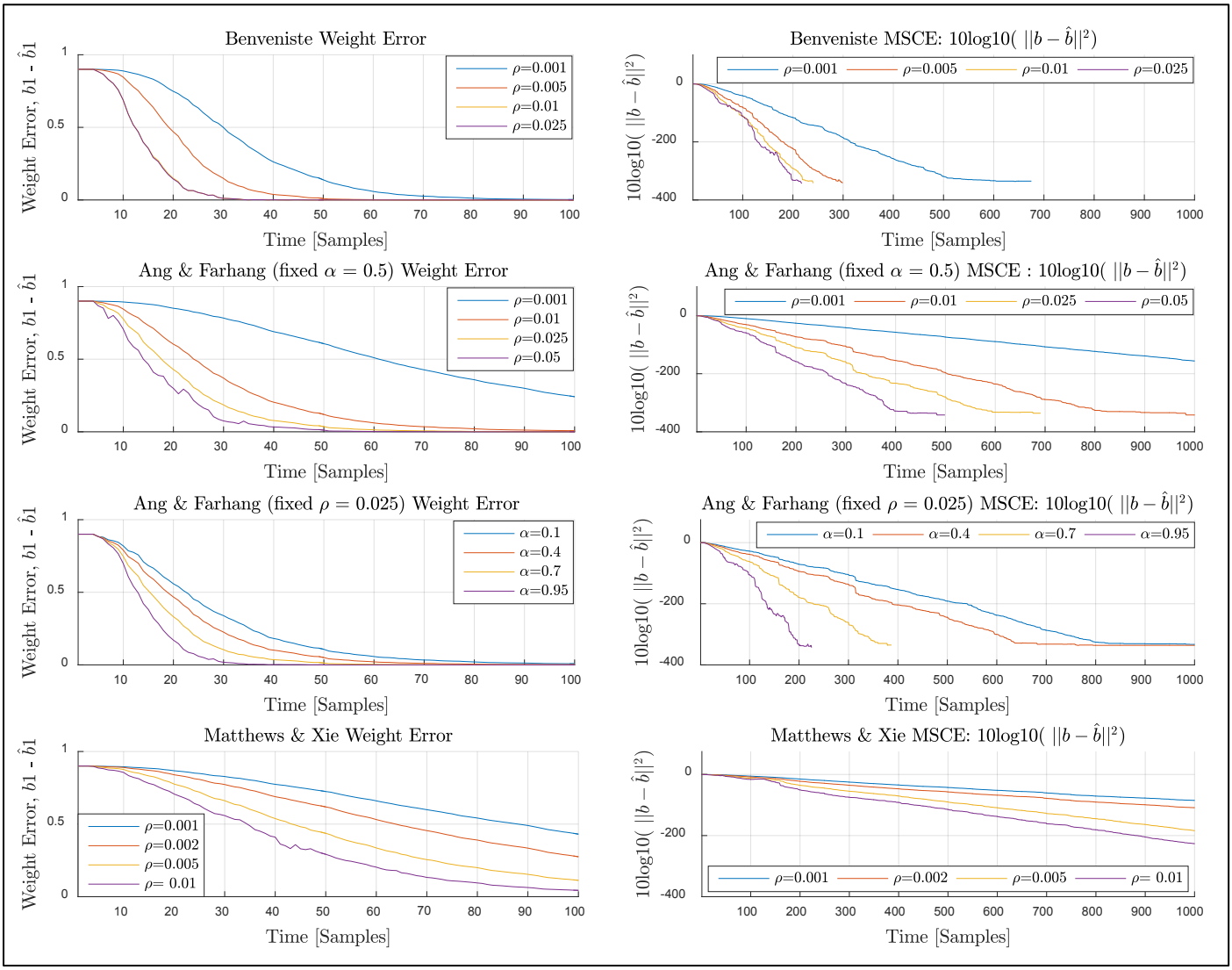

Figure 44: Performance measures of the three GASS algorithms with various parameters (α, ρ). The initial adaptation gain is initialised to 0 for all to allow a fair comparison and 100 independent trials are averaged.

In Figure 44, when the mean squared coefficient error (MSCE) reaches around $-350dB$ the plots cut out due to finite precision; i.e. the MSCE is essentially 0. Compared to the LMS in section 3.1, we can now achieve zero error in both the weights and the predicted output and this is because we are now identifying a $MA(1)$ process with no measurement noise. Figure 45 below compares the three GASS algorithms with their optimal parameters and the standard LMS for two different values of μ. It shows that Benveniste and Ang & Farhang algorithms to significantly outperform the standard LMS in terms of both convergence speed and steady state error as expected.

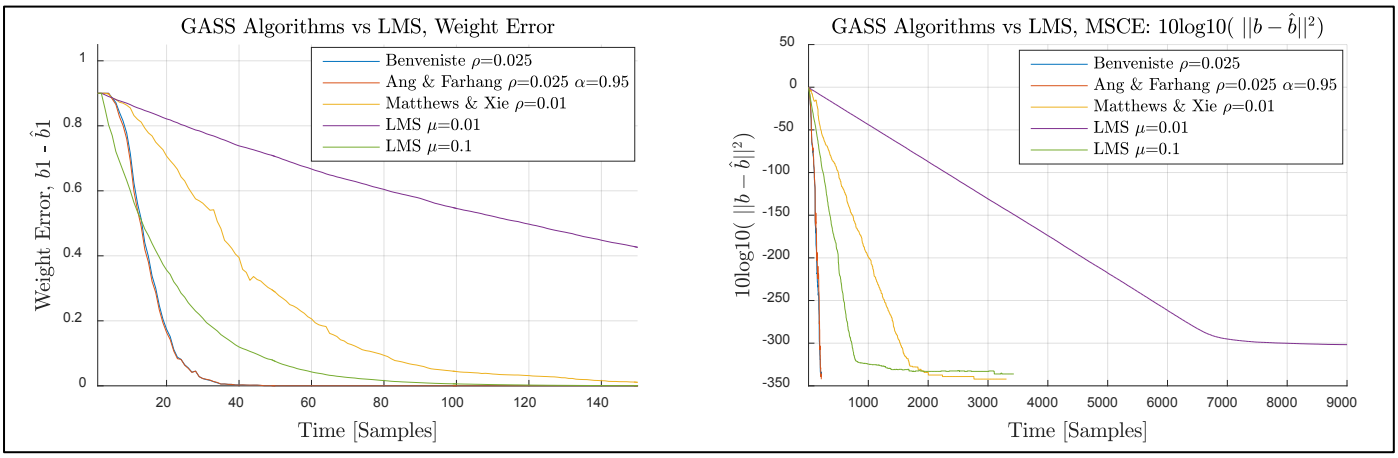

Figure 45: Comparing performance of the three GASS algorithms with optimal parameters and the standard LMS for two different values of μ.

3.2.b Normalised LMS (NLMS) Update Equation:

The NLMS algorithm is a variant of the LMS which basically normalises the step size based on the power of the signal in the filter memory in order to improve stability and convergence. The NLMS update equation is given by equation 3.16 below. We wish to shows that equation 3.17 is equivalent to 3.16 based on the a posteriori error which is given by 3.18. This is proved below. The final result of the proof is given by equation 3.20 and this is equivalent to 3.16 with $\beta = 1$ and $\epsilon = 1/\mu$.

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\beta}{\epsilon + \|\mathbf{x}(n)\|^2} e(n)\mathbf{x}(n) \tag{3.16}$$

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e_p(n)\mathbf{x}(n) \tag{3.17}$$

$$e_p(n) = d(n) - \mathbf{x}^T(n)\mathbf{w}(n+1) \tag{3.18}$$

Start with 3.17:	$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e_p(n)\mathbf{x}(n)$
Multiply by $\mathbf{x}^T(n)$ *and then add* $d(n)$:	$\mathrm{d(n)} + \mathbf{x}^T(n)\mathbf{w}(n+1) = \mathrm{d(n)} + \mathbf{x}^T(n)\mathbf{w}(n) + \mu e_p(n)\mathbf{x}^T(n)\mathbf{x}(n)$
$Re-arrange$ *and use* $\mathbf{x}^T(n)\mathbf{x}(n) = \|\mathbf{x}(n)\|^2$:	$\mathrm{d(n)} - \mathbf{x}^T(n)\boldsymbol{w}(n) = \mathrm{d(n)} + \mathbf{x}^T(n)\mathbf{w}(n+1) + \mu e_p(n)\|\mathbf{x}(n)\|^2$
Substitute error and a posteriori error terms:	$e(n) = \mathrm{e_p(n)} + \mu e_p(n)\|\mathbf{x}(n)\|^2$
$Re-arrange$:	$\mathrm{e_p(n)} = \frac{e(n)}{1+\mu\|\mathbf{x}(n)\|^2}$ (3.19)
Substitute 3.19 *into* 3.17:	$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu\left(\frac{e(n)}{1+\mu\|\mathbf{x}(n)\|^2}\right)\mathbf{x}(n)$
Re arrange to form final result: $c.f \;\; 3.16 \implies \beta = 1, \epsilon = 1/\mu$	$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{e(n)}{1/\mu + \|\mathbf{x}(n)\|^2}\mathbf{x}(n)$ (3.20)

3.2.c Generalised Normalised Gradient Decent (GNGD) Algorithm:

The purpose of ϵ in NLMS update equation is to prevent the step size from being too large (and hence causing instability) when the signal power in the filter memory ($\|\mathrm{x}(n)\|^2$) is small. It also prevents divide by zero errors when $\|\mathrm{x}(n)\|^2 = 0$. By adapting ϵ with time, we can make the gain of the NLMS adaptive and hence more optimal. We can choose an update equation for ϵ such that the cost function is minimized with respect to ϵ using the method of steepest decent, i.e. $\epsilon(n+1) = \epsilon(n) - \rho\nabla_\epsilon J(n)$. This results in the GNGD ϵ update equation as shown by equation 3.24.

Figure 46 below compares the performance of the Benveniste GASS algorithm with optimal parameters ($\rho = 0.025$) and the GNGD algorithm with optimal parameters ($\mu = 1, \beta = 1, \rho = 0.025$). It is worth noting that the parameters μ and ρ can actually be combined into a single parameter since they are simply multiplied together and never used separately. Both algorithms have essentially zero steady state error, but the GNGD reaches essentially zero error about 4 times faster than Benveniste as shown by the plot of MSCE. The coefficients converge with negligible error in fewer than 10 samples with GNGD compared to around 35 with Benveniste.

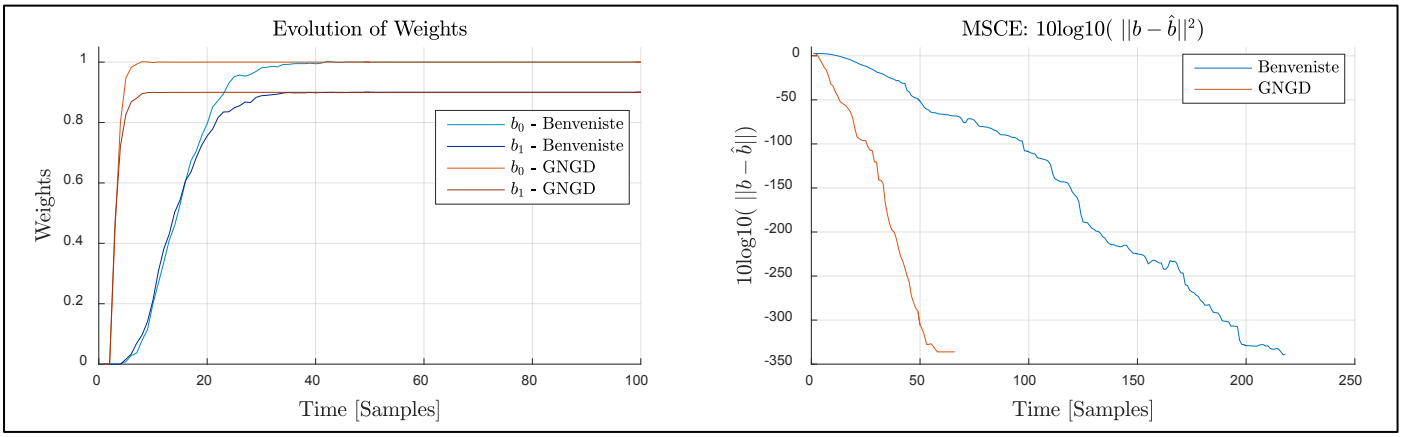

Figure 46: Optimal GNGD ($\mu = 0.1, \beta = 1, \rho = 0.01$) verses optimal Benveniste ($\rho = 0.025$). 100 independent trials are averaged.

The relative computational complexity of Benveniste GASS and GNGD can be compared by considering the complexity of equations 3.21 to 3.24. This is because the MATLAB implementations of these algorithms only differ by these equations and because these equations are the most computationally demanding parts of the algorithms. Table 6 compares the number of operations required for each equation. It assumes the order of operations is optimized.

Benveniste weight update:	$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) + \mu(n)e(n)\mathbf{x}(n)$	3.21
Benveniste μ update:	$\mu(n+1) = \mu(n) + \rho e(n)\mathbf{x}^T(n)\boldsymbol{\psi}(n), \quad where$ $\boldsymbol{\psi}(n) = \underbrace{\left(\boldsymbol{I} - (\mu(n-1) \times \mathbf{x}(n-1)\mathbf{x}^T(n-1))\right)}_{q \times q\ matrix}\boldsymbol{\psi}(n-1) + e(n-1)\mathbf{x}(n-1)$	3.22
GNGD weight update:	$\boldsymbol{w}(n+1) = \boldsymbol{w}(n) + \frac{1}{\epsilon(n) + \|\mathbf{x}(n)\|^2}e(n)\mathbf{x}(n)$	3.23
GNGD ϵ update:	$\epsilon(n+1) = \epsilon(n) - \rho\mu\frac{e(n)e(n-1)\mathbf{x}^T(n)\mathbf{x}(n-1)}{\epsilon(n-1) + \mathbf{x}^T(n-1)\mathbf{x}(n-1)}$	3.24

	Additions	*Multiplications*	*Divisions*
Benveniste weight update (eqn 3.21)	q	$2q$	0
Benveniste μ update (eqn 3.22)	$4q$	$2q^2 + 3q + 2$	0
GNGD weight update (eqn 3.23)	$2q$	$2q + 1$	1
GNGD ϵ update (eqn 3.24)	$2q$	$2q + 4$	1

Table 6: Comparing computational complexity at every time instant (n) for the Benveniste GASS and GNGD algorithms.

	Benveniste GASS/s	*GNGD/s*
$N = 1000, q = 2, 100$ *Averages*	3.84	3.13
$N = 1000, q = 8, 100$ *Averages*	4.72	3.60
$N = 2000, q = 2, 100$ *Averages*	7.70	6.21
$N = 2000, q = 8, 100$ *Averages*	9.45	7.26

Table 7: Run time in seconds for GNGD and Benveniste GASS obtained using `'tic'` and `'toc'`.

In summary, the advantage of the GNGD is that it does not have any matrix-vector multiplications; it only has vector-vector multiplications and hence the complexity GNGD is $O(q)$ compared to $O(q^2)$ for Benveniste GASS. The experimental runtime is compared in Table 7 above and this confirms the theoretical expectations for complexity in the sense that Benveniste has a stronger runtime dependence on model order.

3.3 Adaptive Noise Cancellation

3.3.a Minimum Delay (Δ) for the Adaptive Line Enhancer (ALE):

We wish to show that minimizing $E\{e^2(n)\} = E\{(s(n) - \hat{x}(n))^2\}$ of the ALE (see CW hand-out pg. 14 Figure 4) corresponds to minimising $E\{(x(n) - \hat{x}(n))^2\}$ provided that the noise in the signal $s(n) = x(n) + \eta(n)$ and noise in the signal at the predictor input $u(n) = s(n - \Delta)$ are uncorrelated. We also wish to show that this de-correlation can be achieved by a sufficiently large delay, Δ, for the noise given by $\eta(n) = v(n) + 0.5v(n-2)$, where $v(n)$ is white and has unit variance. The proof is as follows:

Start with mean squared error:	$E\{e^2(n)\} = E\left\{\left(x(n) + \eta(n) - \hat{x}(n)\right)^2\right\}$
Expand the mean squared error:	$= E\{\eta^2(n)\} + E\left\{\left(x(n) - \hat{x}(n)\right)^2\right\} + 2E\{\eta(n)x(n) - \eta(n)\hat{x}(n)\}$
Since $x(n)$ and $\eta(n)$ are uncorrelated:	$= \sigma_\eta^2 + E\left\{\left(x(n) - \hat{x}(n)\right)^2\right\} - 2E\{\eta(n)\hat{x}(n)\}$

Now it is clear that if we can show $E\{\eta(n)\hat{x}(n)\} = 0$ for a sufficiently large Δ, then minimizing $E\{e^2(n)\}$ is the same as minimizing $E\{(x(n) - \hat{x}(n))^2\}$ since σ_η^2 is constant. The proof that $E\{\eta(n)\hat{x}(n)\} = 0$ for some sufficiently large Δ is as follows:

Expand $\hat{x}(n)$:	$E\{\eta(n)\hat{x}(n)\} = E\left\{\eta(n)\left(\sum_{k=0}^{M} w_k(n)s(n-\Delta-k)\right)\right\}$
Expand $s(n-\Delta-k)$:	$= E\left\{\eta(n)\left(\sum_{k=0}^{M} w_k(n)(x(n-\Delta-k) + \eta(n-\Delta-k))\right)\right\}$
Since $x(n)$ and $\eta(n)$ are uncorrelated	$= E\left\{\eta(n)\left(\sum_{k=0}^{M} w_k(n)\eta(n-\Delta-k)\right)\right\}$
Expand noise terms and rearrange:	$= E\left\{\sum_{k=0}^{M} w_k(n)v(n)[v(n-\Delta-k) + 0.5v(n-\Delta-k-2)]\right\}$ $+ E\left\{\sum_{k=0}^{M} w_k(n)0.5v(n-2)[v(n-\Delta-k) + 0.5v(n-\Delta-k-2)]\right\}$
Since $E\{v(i)v(j)\} = 0 \quad \forall i \neq j$	$= 0\ for\ \Delta \geq 3$

The last result follows from the fact that $v(n)$ is white meaning its ACF is a Kronecker delta, i.e. $E\{v(i)v(j)\} = 0 \quad \forall \quad i \neq j$. Also note that last statement of the proof is valid for all filter lengths $(M) > 1$. In summary choosing delay $(\Delta) > 2$ means ensures only noise is suppressed. If $\Delta \leq 2$, then $E\{\eta(n)\hat{x}(n)\} \neq 0$ and so the linear predictor could also be trying to also minimize $\eta(n)\hat{x}(n)$. This means $\hat{x}(n)$ is suppressed since $\eta(n)$ is out of the linear predictors control. Note that a delay of Δ is introduced in the predicted output meaning the predicted output is 0 for the first Δ samples. Figure 47 below shows the prediction and the squared prediction error for various Δ. The time averaged squared prediction error (MSPE) is also computed and shown in Figure 47. This confirms the minimum Δ required for the ALE to be 3 since below this value the prediction error is high, but $\Delta > 3$ results in no noticeable improvement.

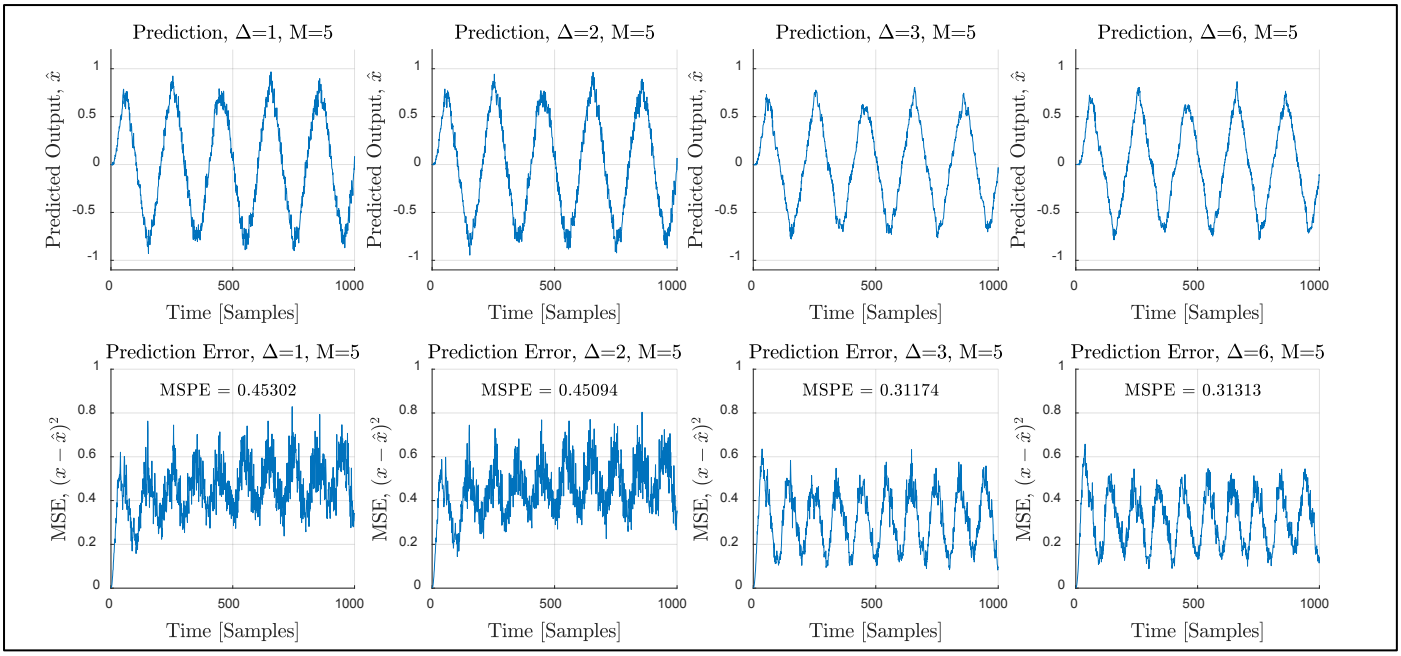

Figure 47: Predicted output and mean square prediction error (MSPE) of the ALE for various Δ. Obtained by averaging 100 independent trials. When there is no ALE (or equivalently when $\Delta = 0$) the MSPE is 1.25. The adaptation gain for the LMS is 0.01.

3.3.b Effect of M and Δ on the Mean Square Predication Error (MSPE) of the ALE:

Figure 48 below shows the predicted output and the squared prediction error averaged over 100 independent trials. It shows that increasing the filter order, M, (for a fixed $\Delta = 3$) results in a larger MSPE (defined by eqn 35, CW hand-out, pg. 15). The dependence of MSPE on Δ and M is investigated further in Figure 49. The left plot of Figure 49 shows how the MSPE varies with Δ for various values of M. The right plot of Figure 49 shows how the MSPE varies with M for various Δ. These plots are obtained by averaging 100 independent trials of length 1000. They show that one optimum value of Δ is 3, and the optimum filter order corresponding to $\Delta = 3$ is 4.

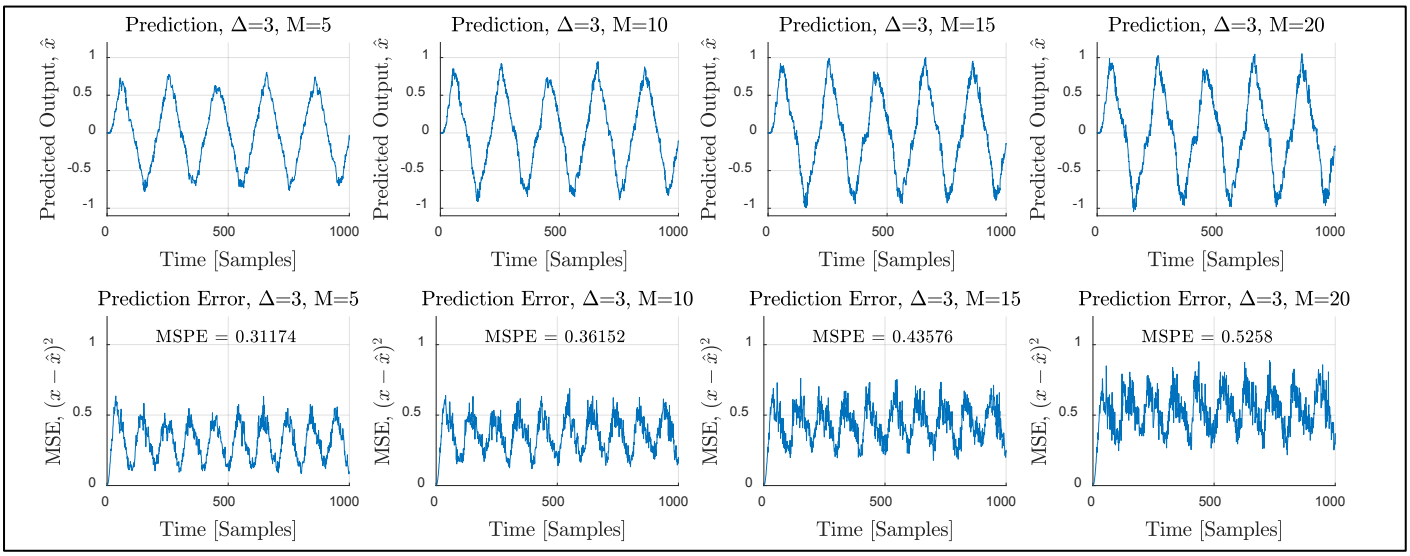

Figure 48: Effect of increasing M on $\hat{x}$ and on the MSPE. Obtained by averaging 100 independent trials. The adaptation gain for the LMS is 0.01

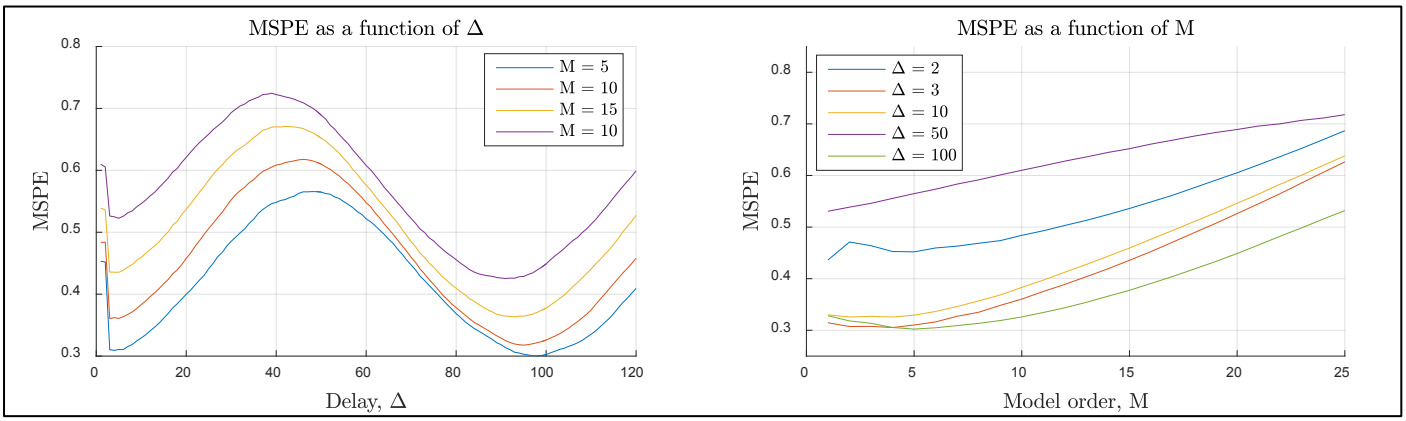

Figure 49: Left plot shows MSPE as a function of Δ for various M. Right plot shows MSPE as a function of M for various Δ. Plots are obtained by averaging 100 independent trials of length 1000. The adaptation gain for the LMS is 0.01.

The reason the MSPE as a function of Δ is periodic with a period half that of the input sinewave can be seen by considering $|E\{x(n)x(n-\Delta)\}|$, which is 0 when $\Delta = 50, 150, 250, etc$, and maximum when $\Delta = 0, 100, 200, etc$. This is because $x(n)$ is a sinewave with a period of 200 samples. In choosing an appropriate Δ we wish to have no correlation between the noise in $s(n)$ and $u(n)$, but a strong correlation of the signal component $x(n)$ in $s(n)$ and $u(n)$, thus ensuring only noise is suppressed. In general since the signal is not a sinewave of known frequency, we wish to select the smallest Δ such that the noise becomes uncorrelated but the signal component does not. In this particular case $\Delta = 3$ is suitable.

Generally a higher order filter is best in terms of accuracy since unnecessary coefficients simply converge to negligible values. However, the complexity of the LMS at each iteration is of $O(M)$ so there will be a tradeoff between computational complexity and predictor accuracy. Furthermore higher order adaptive filters are more unstable and more sensitive to small perturbations in coefficients; a small change in one coefficient can significantly affect the convergence of other coefficients. Therefore there can be an increase in MSPE as model order increases and this is evident in Figure 49. This can be overcome with a smaller step size, but this means slower convergence and hence reduced ability to track faster moving processes. In summary, $M = 4, \Delta = 3$ is suitable for this particular application since this minimizes the MSPE as illustrated by Figure 49.

3.3.c Adaptive Noise Cancellation (ANC) vs. ALE:

The primary input to the ANC configuration is the noise corrupted signal $s(n) = x(n) + \eta(n)$, and a secondary noise input which is correlated in some unknown way with the primary noise. For this exercise we use $v(n)$ as secondary the input. Therefore we expect excellent performance from this filter since $\eta(n)$ is directly generated from $v(n)$ via a MA(2) filter. We use the filter coefficients $b = [1, 0, 0.5]$ to generate $\eta(n)$ so we expect the adaptive filter coefficients to converge to $b = [1, 0, 0.5]$ as well.

Figure 50 compares the performance of the ANC with the ALE. The leftmost plot of Figure 50 shows the MSPE for the ANC and ALE as a function of model order. As expected the MSPE is minimised for the ANC when $M = 2$ since this is the order of the filter used to generate the noise $\eta(n)$. The middle plot of Figure 50 shows the filtered output averaged over 100 independent trials. Comparing this to Figure 48 we see significant improvements in terms of noise reduction. The rightmost plot of Figure 50 shows how the squared error varies over time, and it too shows the ANC to significantly outperform the ALE once the adaptive filter converges. In general the ANC better at removing the noise provided a well correlated secondary noise input is available. However, the advantage of the ALE is that it is more general and does not require a separate noise input.

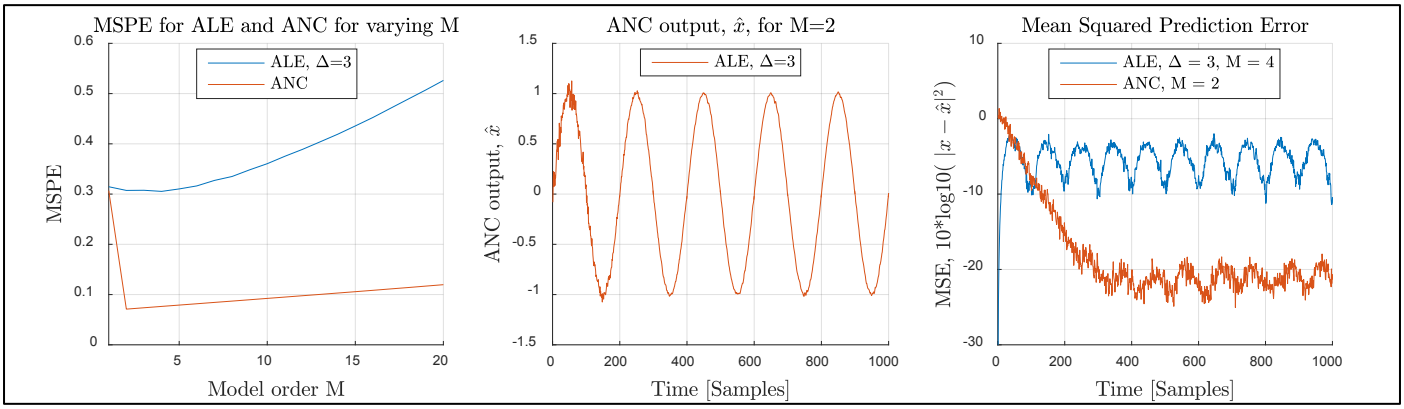

Figure 50: (Left) ANC vs. ALE MSPE for varying model order. (Middle) ANC denoised signal. (Right) comparing mean squared error vs. time for ANC and ALE. All plots have been obtained by ensemble averaging 100 independent trials.

3.3.d Applying the ANC to EEG data:

The ANC filter is applied to the POz EEG data with the secondary noise input being constructed from a 50 Hz sinusoid with AWGN of variance 0.1. The effects of adaptation gain, μ, and filter order, M, on filtering performance are investigated. The spectrograms in this section are constructed with the optimal parameters as determined from section 2.3.b. Figure 51 below shows the how different adaptation gains affect the filtered result. When μ is too small (0.0001), the adaptive filter is slow to adapt and hence not effective at removing the 50 Hz mains interference at every time instant. When μ is too big ($\mu \geq 0.05$), the adaptation is too erratic and hence additional harmonic components at 100 Hz are introduced.

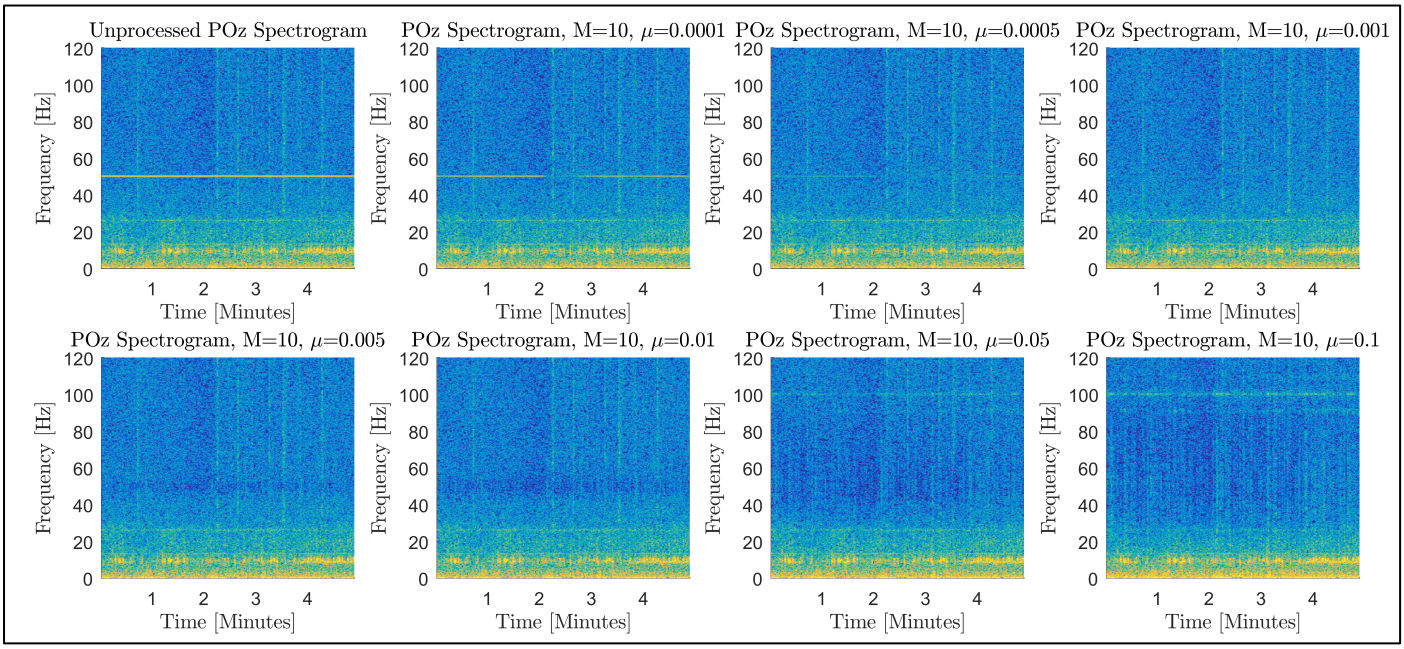

Figure 51: Unprocessed spectrogram for reference and filtered POz signals with ANC with different adaptation gains.

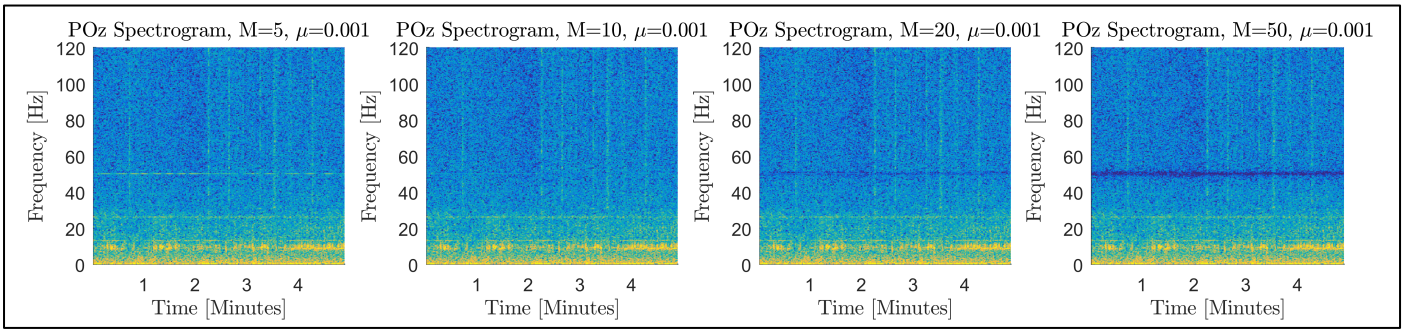

Figure 52: Effect of increasing adaptive filter model order. The second plot from the left shows the best achievable result.

Figure 52 above shows the effects of model order. When M is too small, the $50\ Hz$ component is not fully removed. When the order is too large i.e. 50, then too much spectral power in the $50\ Hz$ region is removed. In summary, the overall best result is obtained when $\mu = 0.01$ and $M = 10$ as shown by the second plot in Figure 52. This shows the mains interference to be removed without noticeably affecting the other frequencies.

4 Widely Linear Filtering and Adaptive Spectrum Estimation

4.1 Complex LMS and Widely Linear Modelling

4.1.a Complex LMS (CLMS) and Augmented CLMS (ACLMS):

This section investigates the performance of the CLMS and ACLMS for the identification of the widely linear moving average (WLMA) process given by equation 4.1, in which $x(n)$ is circular WGN. A complex random variable is said to be circular if its statistical properties are rotation invariant. Note that the process $y(n)$ cannot be circular since there is a weighted component of the conjugate of the past input. In fact, the circularity quotient (or coefficient) for $y(n)$ is $\rho = p/c = E(yy^*)/E(yy)$ is $0.7908 + 0.3255i$.

$$y(n) = x(n) + b_1 x(n-1) + b_2 x^*(n-1), \qquad b_1 = 1.5 + 1j, \qquad b_2 = 2.5 - 0.5j, \qquad x(n) \sim \mathcal{N}(0,1) \tag{4.1}$$

Figure 53 below shows the performance of the CLMS and ACLMS both set with an adaptation gain of 0.025. Since $y(n)$ is non-circular we expect the CLMS to struggle to identify the system. This is because strictly linear extensions of real valued estimators can only cater for circular data since they do not capture second order statistical relationships between input and output. This is confirmed by Figure 53 which shows the steady state MSE of the CLMS to be $8.3dB$. In contrast, the steady state MSE for the ACLMS is $-300dB \approx 0$.

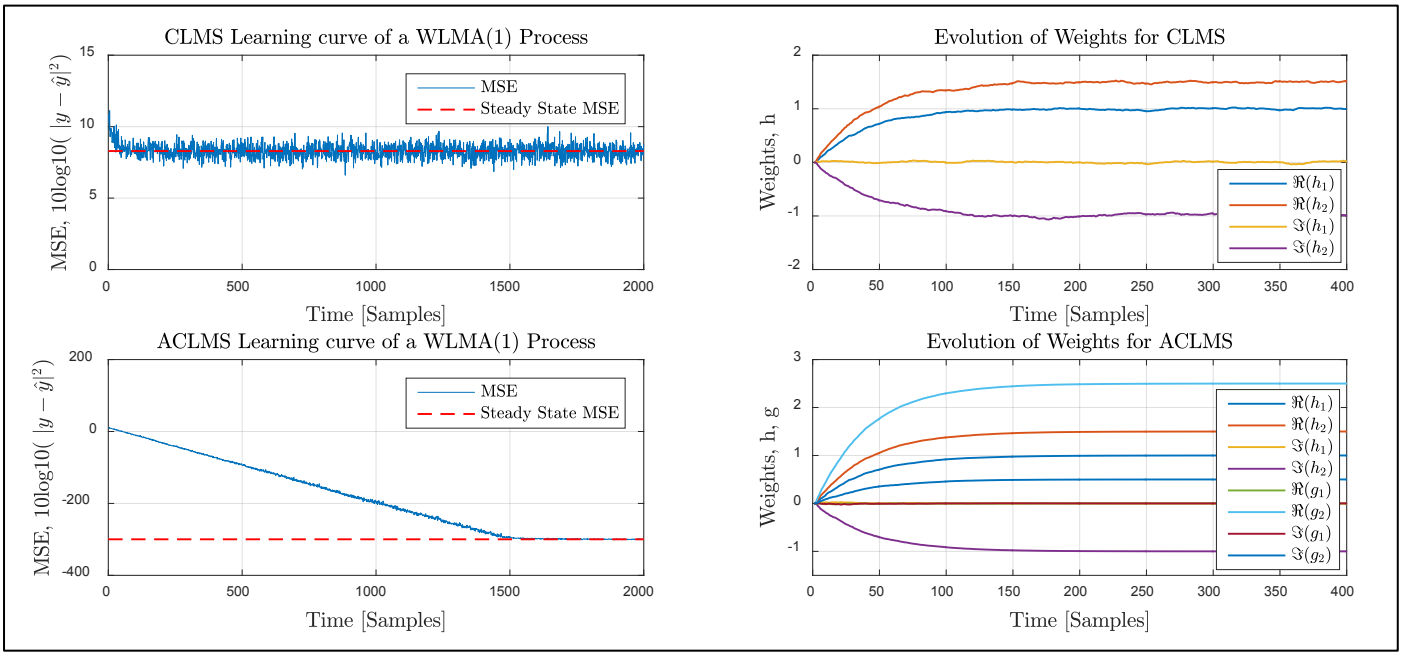

Figure 53: MSE and evolution of weights for the CLMS and ACLMS applied to $y(n)$ (eqn 4.1). 100 independent trials are averaged. $\mu = 0.025$

From Figure 53 we see that the weight vector $\mathbf{h}$ converges to $[1, b_1]^H$ and $\mathbf{g}$ converges to $[0, b_2]^H$. The reason for converging to the conjugates can be easily explained by considering the CLMS and the ACLMS formulae as shown by 4.2 below. Both of these algorithms minimize the cost function $|y(n) - \hat{y}(n)|^2$ by updating the weights in the direction of steepest decent. Because $\hat{y}(n)$ is constructed from the conjugates of the weights we expect $\mathbf{h}$ and $\mathbf{g}$ to converge to $[1, b_1]^H$ and $[0, b_2]^H$ respectively. These equations also show that the reason the CLMS is unsuitable for identifying $y(n)$ since there is no possible way for it to include the contribution to $y(n)$ from the $x^*(n-1)$ term.

CLMS:	*ACLMS*:
$\hat{y}(n) = \mathbf{h}^H(n)\mathbf{x}(n)$ $e(n) = y(n) - \hat{y}(n)$ $\mathbf{h}(n+1) = \mathbf{h}(n) + \mu e^*(n)\mathbf{x}(n)$	$\hat{y}(n) = \mathbf{h}^H(n)\mathbf{x}(n) + \mathbf{g}^H(n)\mathbf{x}^*(n)$ $e(n) = y(n) - \hat{y}(n)$ $\mathbf{h}(n+1) = \mathbf{h}(n) + \mu e^*(n)\mathbf{x}(n)$ $\mathbf{g}(n+1) = \mathbf{g}(n) + \mu e^*(n)\mathbf{x}^*(n)$

4.2

If we use the form of the CLMS and ACLMS described by 4.3 below, then we do not converge to the conjugates of the weights.

$CLMS-Alternate\ Form$:	$ACLMS-Alternate\ Form$:	
$\hat{y}(n) = \mathbf{h}^T(n)\mathbf{x}(n)$ $e(n) = y(n) - \hat{y}(n)$ $\mathbf{h}(n+1) = \mathbf{h}(n) + \mu e(n)^* \mathbf{x}(n)$	$\hat{y}(n) = \mathbf{h}^T(n)\mathbf{x}(n) + \mathbf{g}^T(n)\mathbf{x}^*(n)$ $e(n) = y(n) - \hat{y}(n)$ $\mathbf{h}(n+1) = \mathbf{h}(n) + \mu e(n)\mathbf{x}^*(n)$ $\mathbf{g}(n+1) = \mathbf{g}(n) + \mu e(n)\mathbf{x}(n)$	4.3

4.1.b Circularity and prediction of Bivariate Wind Data:

A complex valued wind signal is created as $v(n) = v_{east}(n) + jv_{north}(n)$. This is convenient since $|v(n)|$ is the wind speed magnitude, and $\angle v(n)$ is the wind direction. A complex random variable $Z = X + iY$ is said to be circular if its statistical properties are rotational independent, and this is the case if its probability density function (PDF) is a function of only $ZZ^* = |Z|$ and not of Z^*. If the wind data is circular, then the circularity quotient $\rho = p/c = 0$, or more specifically, $p = 0$. Since $p = \sigma_x^2 - \sigma_y^2 + 2j\rho_{xy}$, the wind speed data is circular *iff* there is equal wind power in the east and north directions and it is equally likely to be traveling in any direction ($\rho_{xy} = 0$).

The circularity plots are shown in Figure 54 below along with the circularity quotient. By visually comparing the spread of the data with the red concentric circles we can see that the medium wind speed data is most circular, and the low wind speed data is least circular. These observations agree with the magnitude of the circularity quotient which is smallest for the medium wind speed and largest for the low wind speed. Note that the data has been de-trended to remove the effects of the DC component. If the mean is not removed, then the high wind speed data is least circular because it has a large bias in the north easterly direction. This is intuitively expected since when wind is powerful we expect it to have a particular direction due to the natural phenomena which causes it.

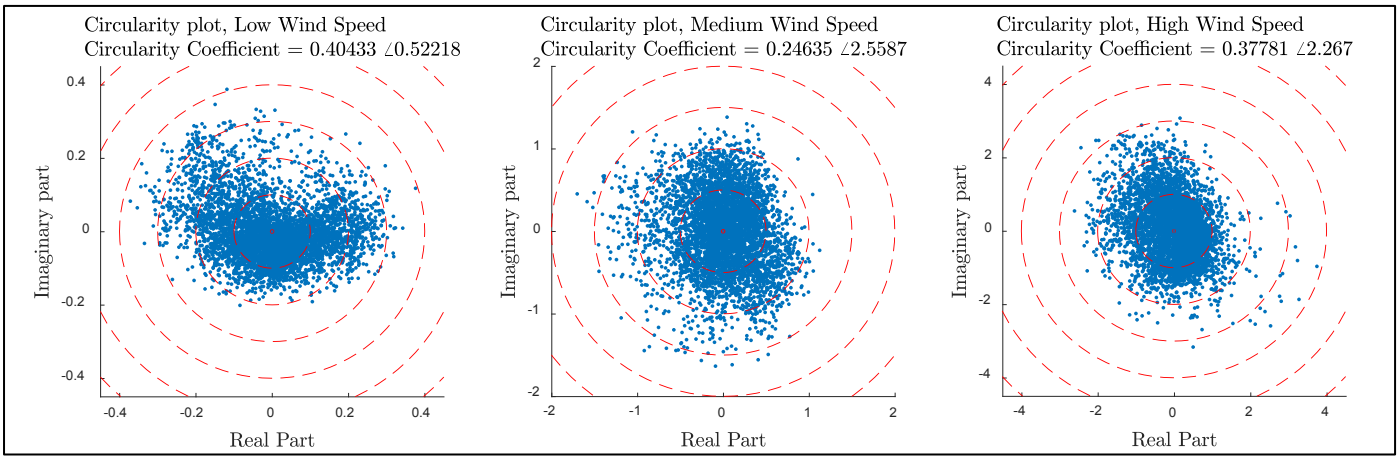

Figure 54: Circularity plots and circularity quotients of wind speed data. Concentric circles are included in red to help visually judge circularity.

The CLMS and ACLMS are configured in a prediction setting and used to perform one-step ahead prediction of the wind data for the three different wind regimes. The time averaged squared prediction error is plotted for different filter lengths (or model orders) as shown by Figure 55 below. From this we can see that the ACLMS outperforms the CLMS for the low and high wind speed data. For the medium wind speed data there is negligible difference. This is expected since the medium wind speed data is the most circular, whereas the low wind speed data is the least circular, and we know the CLMS is only suitable for circular data. Note that the axis scaling is the same in all plots in order to allow a better comparison. The difference between the ACLMS and the CLMS prediction error with a suitable model order is summarized in Table 8 below.

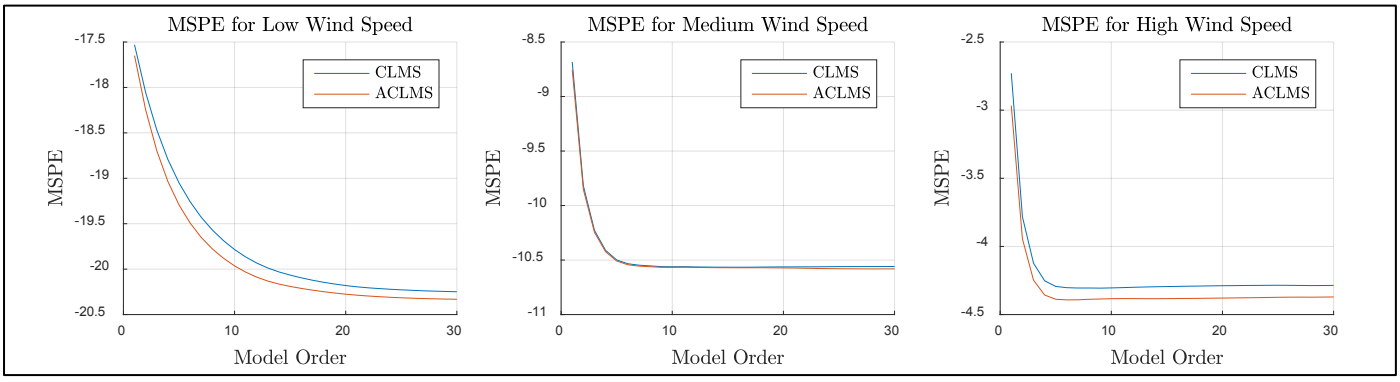

Figure 55: Time averaged squared prediction error (MSPE) vs. model order for the three wind regimes for both the CLMS and ACLMS in predictor configuration.

	Model Order:	*CLMS MSE (dB)*:	*ACLMS MSE (dB)*:	*ACLMS − CLMS (dB)*:
Low Wind Speed:	30	−20.25	−20.33	−0.07
Medium Wind Speed:	10	−10.56	−10.57	−0.01
High Wind Speed:	10	−4.304	−4.373	−0.069

Table 8: Comparing the MSE of the ACLMS and CLMS for the different wind regimes when a suitable model order is selected.

Table 8 confirms that the ACLMS outperforms the CLMS for the non-circular low wind speed data. It shows that the CLMS and the ACLMS perform similarly for the medium wind speed data which is relatively circular compared to the other regimes. It also shows that the prediction error is generally worse as the wind speed increases. This is simply because the higher wind speed is more erratic and thus more difficult to predict. For example, the standard deviation of the magnitude of the high wind speed data is 0.5242, whereas for the low wind speed data it is only 0.072.

4.1.c Clarke Voltage Circularity Plots:

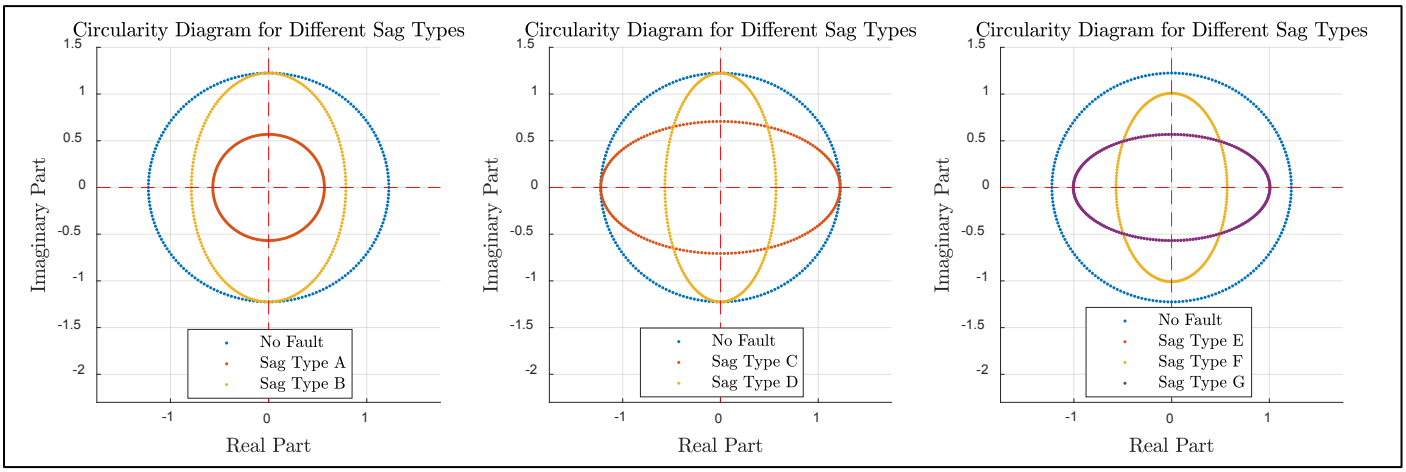

Figure 56: The circularity diagrams for the Clarke voltage for different sag types. Interestingly the circularity plot for type E and G sag is the exact same so the plot for sag type E is covered by the plot for sag type G. Refer to Table 9 for the definitions of the different sag types.

Figure 56 shows how the shape of the circularity diagram changes with different electrical fault conditions. A three phase fault results in all the three phase voltages decreasing and is known as Type A sag (refer to Table 9). This results in a smaller circle for the circularity plot as shown by the left plot in Figure 56, but since it is still balanced, the Clarke voltage is still circular. Type B sag is when only one phase voltage is reduced in amplitude, and this type of sag can only be caused by a single line to ground fault. From the circularity diagram we see that this results in the circle being compressed along only one axis. In this case it is compressed along the real axis since phase A is reduced. If instead phase B is reduced, it is compressed along the 45 degree line, or if phase C is reduced, then it is compressed along the -45 degree line.

Type C and D sag are when amplitudes of two phases are reduced and when phases are non-symmetrical, and they can be caused by either a line to line or single line to ground fault. Types E, F and G are only expected if the fault is line to line to ground. In conclusion whenever the system is unbalanced due to either amplitude or phase imbalance, we can detect this since the Clarke voltage is no longer circular. In addition, based on the shape of the circularity plot, we may be able to deduce the type of fault. The left plot in Figure 57 below shows the effects of only phase imbalance, and the middle plot shows the effect of only amplitude imbalance. The rightmost plot shows time varying amplitude.

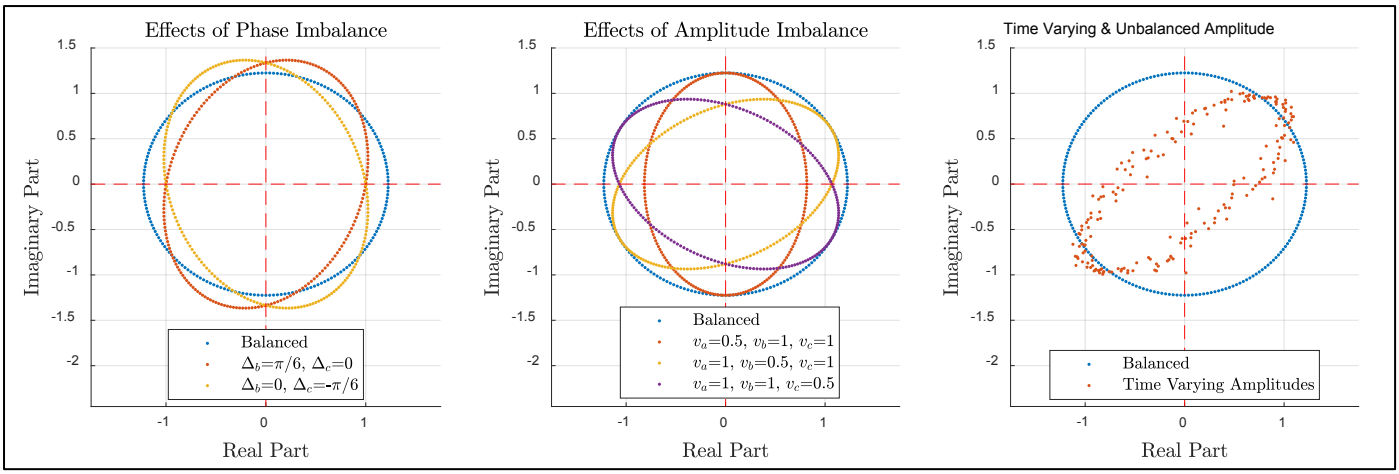

Figure 57: Showing additional Clarke voltages, this time isolating the effects of phase and amplitude imbalance. The rightmost plot shows the effect of time varying and unbalanced amplitude.

4.1.d Frequency Derivation:

For the balanced three phase system the Clarke voltage is given by equation 4.4. We can derive the frequency of the balanced system voltage from the coefficients $h(n)$ of the strictly linear AR model which is given by equation 4.5. The proof is as follows:

$Clarke\ Voltage\ (balanced\ system)$:	$v(n) = \sqrt{\frac{3}{2}} V e^{j\left(\frac{2\pi f_o}{f_s} n + \phi\right)}$	4.4
$Strictly\ Linear\ AR\ model$:	$v(n+1) = h^*(n) v(n)$	4.5
$Substitute\ 4.4\ into\ 4.5$: $(Valid\ in\ steady\ state)$	$\sqrt{\frac{3}{2}} V e^{j\left(\frac{2\pi f_o}{f_s}(n+1) + \phi\right)} = h^*(n) \sqrt{\frac{3}{2}} V e^{j\left(\frac{2\pi f_o}{f_s} n + \phi\right)}$	
$Divide\ 4.5\ by\ \sqrt{3/2} V e^{j\left(\frac{2\pi f_o}{f_s} n + \phi\right)}$:	$e^{j\left(\frac{2\pi f_o}{f_s}\right)} = h^*(n)$	
$Use\ arg\left(e^{j\theta}\right) = \theta = \arctan\left(\frac{\Im\left(e^{j\theta}\right)}{\Re\left(e^{j\theta}\right)}\right)$:	$f_o = \frac{f_s}{2\pi} \arctan\left(\frac{\Im\left(h^*(n)\right)}{\Re\left(h^*(n)\right)}\right)$	4.6

Note that $arctan(x) \approx x$ for small x so the final result given by equation 4.6 matches the required result in the CW hand out (equation 48 page 17). We can do a similar derivation for the frequency of unbalanced voltages. The complex valued voltage that arises from the Clarke transform in an unbalanced case is given by equation 4.7 below. Equation 4.8 is the same as 4.7 but with the index shifted by one sample and this result is used later.

$Clarke\ Voltage$ $(unbalanced\ system)$:	$v(n) = A(n) e^{j\left(\frac{2\pi f_o}{f_s} n + \phi\right)} + B(n) e^{-j\left(\frac{2\pi f_o}{f_s} n + \phi\right)}$ $where\ A(n) = \left(\sqrt{6}/6\right)\left(V_a(n) + V_b(n) e^{j\Delta_b} + V_c(n) e^{j\Delta_c}\right)$ $and\ B(n) = \left(\sqrt{6}/6\right)\left(V_a(n) + V_b(n) e^{-j(\Delta_b + 2\pi/3)} + V_c(n) e^{-j(\Delta_c - 2\pi/3)}\right)$	4.7
$Shift\ time\ index$ $and\ re-arrange$:	$v(n+1) = \left(e^{j\left(\frac{2\pi f_o}{f_s} n + \phi\right)}\right)\left[A(n+1) e^{j\left(\frac{2\pi f_o}{f_s}\right)}\right] + \left(e^{-j\left(\frac{2\pi f_o}{f_s} n + \phi\right)}\right)\left[B(n+1) e^{-j\left(\frac{2\pi f_o}{f_s}\right)}\right]$	4.8

The widely linear AR model of order 1 is given by equation 4.9 below. The derivation of determining frequency from the ACLMS coefficients $h(n)$and $g(n)$ is as follows:

$Widely\ Linear\ AR\ model$:	$v(n+1) = h^*(n) v(n) + g^*(n) v^*(n)$	4.9
$Substitute\ v(n)$ $from\ 4.7\ into\ 4.9$:	$v(n+1) = h^*(n) A(n) e^{j\left(\frac{2\pi f_o}{f_s} n + \phi\right)} + h^*(n) B(n) e^{-j\left(\frac{2\pi f_o}{f_s} n + \phi\right)}$ $+ g^*(n) A^*(n) e^{-j\left(\frac{2\pi f_o}{f_s} n + \phi\right)} + g^*(n) B^*(n) e^{j\left(\frac{2\pi f_o}{f_s} n + \phi\right)}$	4.10
$Factorise\ 4.10$:	$v(n+1) = \left(e^{j\left(\frac{2\pi f_o}{f_s} n + \phi\right)}\right)\left[h^*(n) A(n) + g^*(n) B^*(n)\right]$ $+ \left(e^{-j\left(\frac{2\pi f_o}{f_s} n + \phi\right)}\right)\left[h^*(n) B(n) + g^*(n) A^*(n)\right]$	4.11

Comparing equation 4.11 with 4.8 (the terms in the square brackets) results in equations 4.12 and 4.13. Note that this is valid only in the steady state since the widely linear model coefficients need to converge for 4.11 to be the comparable to 4.8.

$Compare\ 4.11\ with\ 4.8$:	$A(n+1) e^{j\left(\frac{2\pi f_o}{f_s}\right)} = h^*(n) A(n) + g^*(n) B^*(n)$	4.12
$Compare\ 4.11\ with\ 4.8$:	$B(n+1) e^{-j\left(\frac{2\pi f_o}{f_s}\right)} = h^*(n) B(n) + g^*(n) A^*(n)$	4.13
$Use\ A(n+1) \approx A(n)\ (since\ f_s \gg f_o)$ $in\ 4.12\ and\ re-arrange$:	$e^{j\left(\frac{2\pi f_o}{f_s}\right)} = h^*(n) + \frac{g^*(n) B^*(n)}{A(n)}$	4.14
$Use\ B(n+1) \approx B(n)\ (since\ f_s \gg f_o)$ $in\ 4.13\ and\ re-arrange$:	$e^{-j\left(\frac{2\pi f_o}{f_s}\right)} = h^*(n) + \frac{g^*(n) A^*(n)}{B(n)}$	4.15
$Conjugate\ 4.15$:	$e^{j\left(\frac{2\pi f_o}{f_s}\right)} = h(n) + \frac{g(n) A(n)}{B^*(n)}$	4.16

$a(n) \triangleq \frac{B^*(n)}{A(n)}$ *and put into* 4.14	$e^{j\left(\frac{2\pi f_o}{f_s}\right)} = h^*(n) + g^*(n)a(n)$	4.17
$a(n) \triangleq \frac{B^*(n)}{A(n)}$ *and put into* 4.16	$e^{j\left(\frac{2\pi f_o}{f_s}\right)} = h(n) + g(n)a^{-1}(n)$	4.18
Eliminate $e^{j\left(\frac{2\pi f_o}{f_s}\right)}$ *from* 4.17 *and* 4.18 *and re − arrange*:	$0 = g^*(n)a^2(n) + \left(h^*(n) - h(n)\right)a(n) - g(n)$	4.19
Solve quadratic equation 4.19:	$a_{1,2}(n) = \frac{-j\Im\left(h^*(n)\right) \pm \sqrt{-\left(\Im\left(h(n)\right)\right)^2 + \lvert g(n)\rvert^2}}{g^*(n)}$	4.20

The result shown by 4.20 can be substituted into 4.17 allowing us to solve for f_o. Before doing this we make some observations on the nature of the two solutions of 4.19. Firstly we know $a(n)$ is complex since $B^*(n)/A(n)$ is complex. Therefore the discriminant $(b^2 - 4ac)$ of equation 4.19 must be negative. Also we know that the imaginary part of 4.17 is positive since the imaginary part is given by $sin(2\pi f_o/f_s)$ and $f_s > f_o$. This means the only valid solution is given by 4.21 below. Using the other solution for $a(n)$ will result in a negative imaginary part for 4.17 and hence a negative f_o. The proof is continued below:

Only valid solution of 4.19:	$a_1(n) = \frac{-j\Im\left(h^*(n)\right) + j\sqrt{\left(\Im\left(h(n)\right)\right)^2 - \lvert g(n)\rvert^2}}{g^*(n)}$	4.21
Substitute 4.21 *into* 4.17:	$e^{j\left(\frac{2\pi f_o}{f_s}\right)} = h^*(n) - j\Im\left(h^*(n)\right) + j\sqrt{\left(\Im\left(h(n)\right)\right)^2 - \lvert g(n)\rvert^2}$	4.22
Use $arg\left(e^{j\theta}\right) = \theta = \arctan\left(\frac{\Im\left(e^{j\theta}\right)}{\Re\left(e^{j\theta}\right)}\right)$ *and* $\Re\left(h^*(n)\right) = \Re\left(\mathrm{h}(n)\right)$:	$f_o = \frac{f_s}{2\pi} arctan\left(\frac{\sqrt{\left(\Im\left(h(n)\right)\right)^2 - \lvert g(n)\rvert^2}}{\Re\left(\mathrm{h}(n)\right)}\right)$	4.23

4.1.e Frequency Estimation from CLMS and ACLMS:

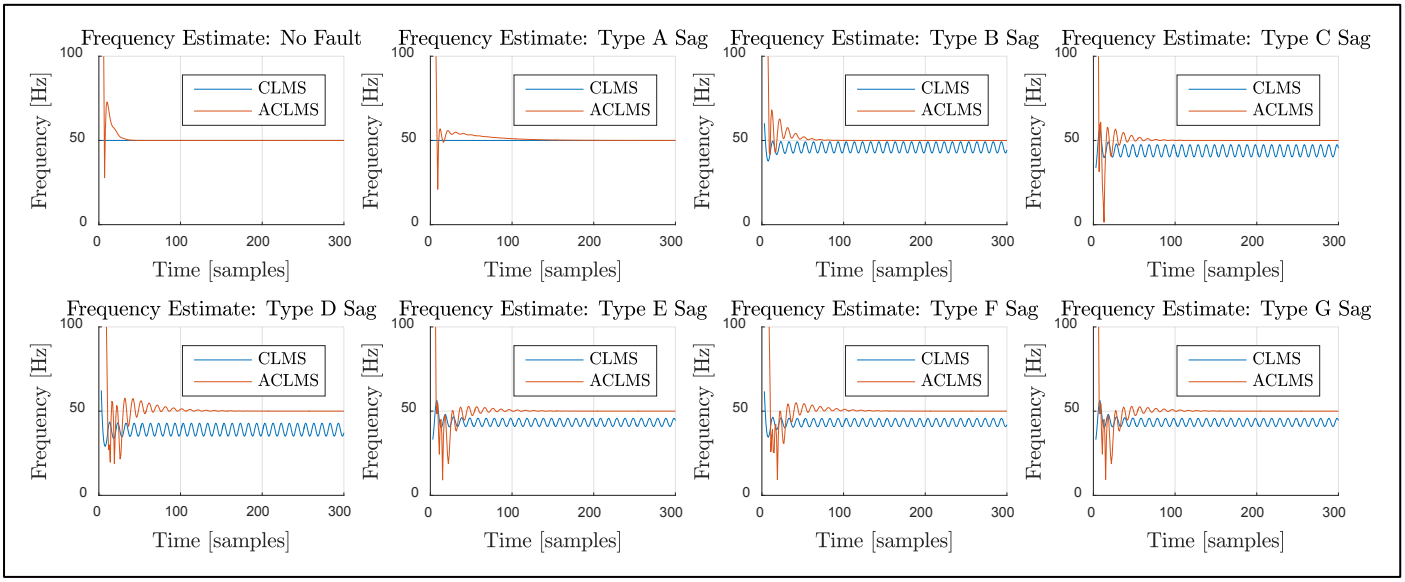

Figure 58: the frequency estimate for the three phase AC system under different sag types. Refer to Table 9 for the sag definitions.

Figure 58 shows the frequency estimate for the three phase AC system under different sag types (fault conditions). The seven types of voltage sag are explained by Table 9. We see that whenever the system is balanced, i.e. no fault or Type A sag, both the CLMS and ACLMS based frequency estimates converge to $50\ Hz$ exactly. The ACLMS takes a longer time to converge to the correct frequency and this is because the weighs **g** and **h** need to converge to their correct steady state values in order for the analysis in section 4.1.d to be valid. Regardless of the fault type, the ACLMS based frequency estimate always converges to exactly $50\ Hz$. In contrast, in imbalanced conditions, the CLMS based estimate does not converge to a single value and is on average lower than $50\ Hz$. Interestingly the oscillation frequency of the CLMS frequency estimate for all the imbalanced cases is exactly $100\ Hz$. Note that the Clarke voltage is sampled at $f_s = 1000\ Hz$.

As we would expect, the CLMS based method for estimating system frequency is not suitable when the system is imbalanced. This is because the Clarke voltage (eqn 4.7) is influenced by the conjugate component $(B(n) \neq 0)$ during imbalanced conditions, and it is therefore non-circular. As a consequence, the CLMS method is unsuitable for estimating the frequency since it assumes

a circular Clarke voltage. In summary, we must use the conjugate part when modelling the signal during fault conditions which is why the ACLMS outperforms the CLMS when determining system frequency under imbalanced conditions.

No Fault: $V_a = E$ $V_b = -0.5E - j0.5\sqrt{3}E$ $V_c == -0.5E + j0.5\sqrt{3}E$	Type A Sag: $V_a = V$ $V_b = -0.5V - j0.5\sqrt{3}V$ $V_c == -0.5V + j0.5\sqrt{3}V$
Type B Sag: $V_a = V$ $V_b = -0.5E - j0.5\sqrt{3}E$ $V_c == -0.5E + j0.5\sqrt{3}E$	Type C Sag: $V_a = E$ $V_b = -0.5E - j0.5\sqrt{3}V$ $V_c == -0.5E + j0.5\sqrt{3}V$
Type D Sag: $V_a = V$ $V_b = -0.5V - j0.5\sqrt{3}E$ $V_c == -0.5V + j0.5\sqrt{3}E$	Type E Sag: $V_a = E$ $V_b = -0.5V - j0.5\sqrt{3}V$ $V_c == -0.5V + j0.5\sqrt{3}V$
Type F Sag: $V_a = V$ $V_b = -0.5V - j(V\sqrt{3}/6 + E\sqrt{3}/3)$ $V_c == -0.5V + j(V\sqrt{3}/6 + E\sqrt{3}/3)$	Type G Sag: $V_a = 2E/3 + V/3$ $V_b = -E/3 - V/6 - j0.5\sqrt{3}V$ $V_c == -E/3 - V/6 + j0.5\sqrt{3}V$

Table 9: The seven types of voltage sag for unbalanced three phase systems according to the ABC classification. The pre event voltage magnitude is given by E, and the sagged voltage magnitude is given by V. The phasors V_a, V_b, V_c denote the phase vectors of the three phase system. For all the plots in this section the sag voltage V is taken as $0.5\ p.u.$

The left plot of Figure 59 below shows how the steady state time averaged frequency estimate varies with phase imbalance due to an additional phase angle of Δ_b in phase b. The right plot of Figure 59 shows how the magnitude imbalance affects the frequency estimate. These basically show that the ACLMS based method has no problem in determining the exact system frequency, whereas the CLMS based method always underestimates the frequency during imbalanced conditions.

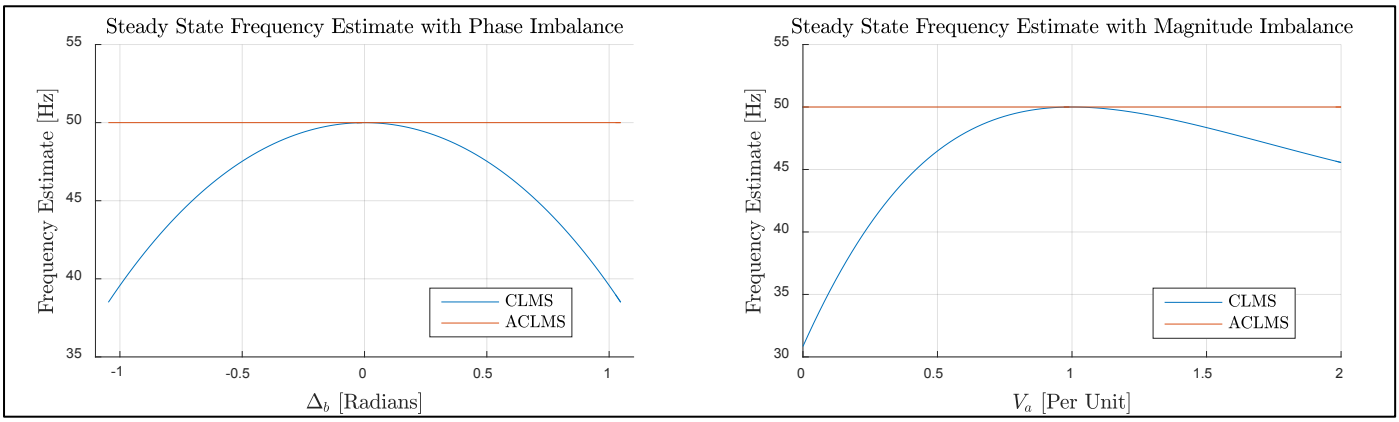

Figure 59: The left plot shows how the time averaged steady state frequency estimate for the ACLMS and CLMS varies with phase imbalance. The right plot shows how magnitude imbalance affects the frequency estimates.

4.2 Adaptive AR Model Based Time-Frequency Estimation

4.2.a Block Based AR Spectrum Estimation of a FM signal:

Noisy FM signal:	$y(n) = e^{(2\pi\phi(n)/fs)} + \eta(n),$	4.24
Instantaneous frequency:	$f(n) = \frac{d\phi(n)}{dn} = \begin{cases} 100, & 1 \le n \le 500 \\ n/2 - 150, & 501 \le n \le 1000 \\ n^2/625 - 3.2n + 1700 & 1001 \le n \le 1500 \end{cases}$	4.25

Equation 4.24 describes a FM signal with additive circular complex WGN $\eta(n)$ with $\mu = 0, \sigma^2 = 0.05$. The signal is sampled at a rate of $f_s = 1100\ Hz$, which just above the nyquist rate of $1000\ Hz$. Figure 60 below shows the instantaneous frequency and the real and imaginary parts of $y(n)$. Since the signal $y(n)$ is circular we can use a strictly linear model for estimation. A model order of 1 could be argued to be suitable considering that there is only one frequency at any time instant (albeit a time varying frequency). The aryule function is applied to the whole signal to estimate the AR(1) parameter and the corresponding PSD estimate is shown by the leftmost plot in Figure 61 below. This shows the estimated frequency to be $179\ Hz$. This is actually close to the average instantaneous frequency (the time average of equation 4.25) which is $186.3\ Hz$. As expected, using a block based spectrum estimation method is not suitable for estimating non-stationary signals since it does not capture the dynamics of the signal, i.e. in this case, it does not capture the time varying instantaneous frequency.

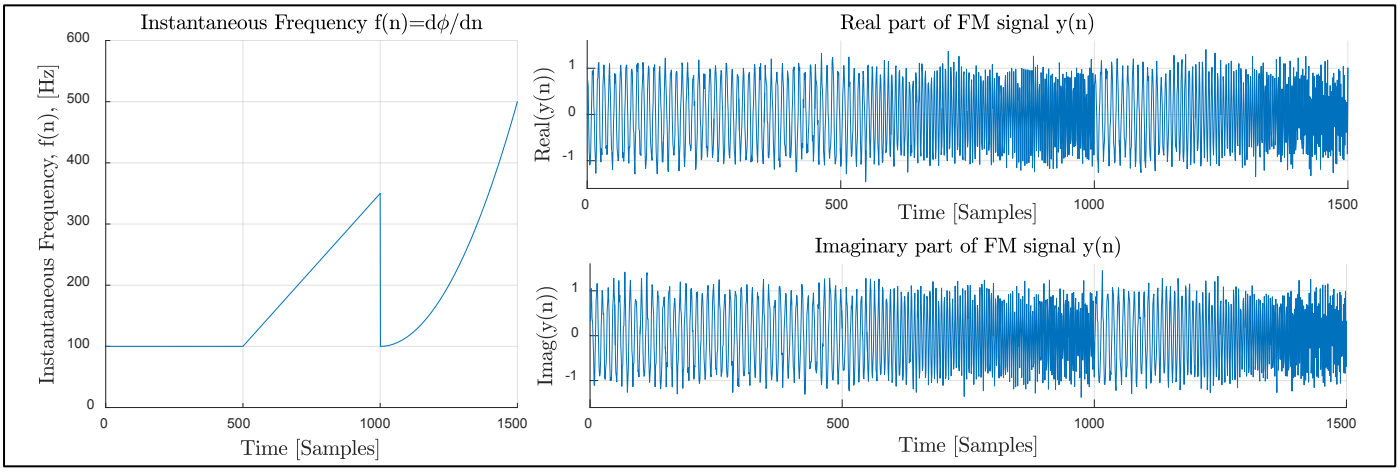

Figure 60: The left plot shows the instantaneous frequency ($f(n)$, eqn 4.25) of the noisy FM signal ($y(n)$, eqn 4.24). The right plots show the real and imaginary parts of the noisy FM signal.

The other plots in Figure 61 below show the PSD estimates with a higher model order. These capture more spectral information in the sense that they show evidence of the other frequencies known to exist in the FM signal, but the time varying nature is not captured. One suggestion for improvement could be to use overlapping smaller segments of the data (like the spectrogram) in order to track the non-stationary signal. In the next sub section an adaptive filter approach is considered.

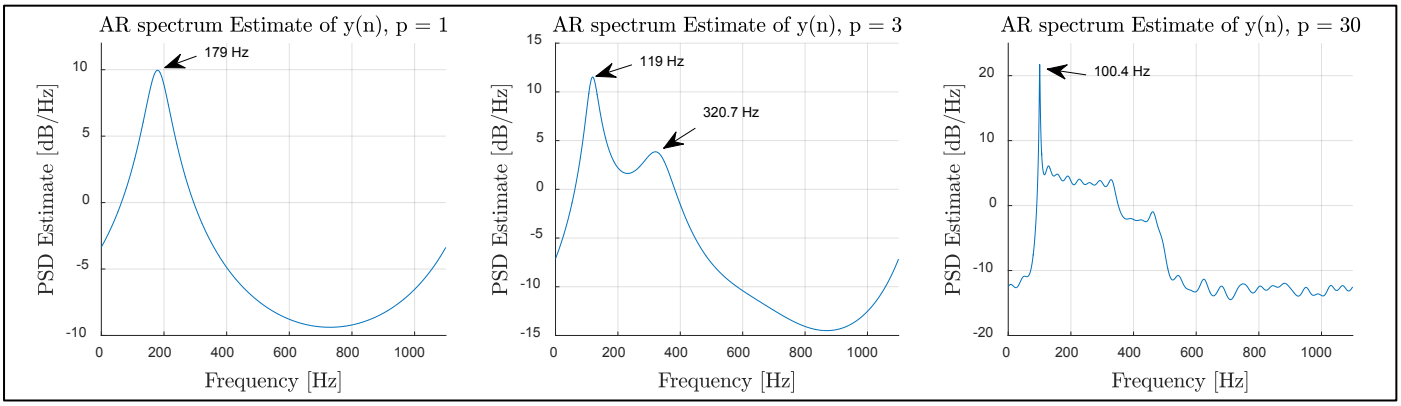

Figure 61: AR spectrum estimate of the whole FM signal with different model orders, p.

4.2.b Online (Adaptive) AR Spectrum Estimation of a FM signal:

The left plot of Figure 62 below shows the time frequency spectrum of the FM signal as described by equation 4.24. It has been obtained by estimating the time varying AR(1) coefficients using the CLMS algorithm with an adaptation gain of $\mu = 0.1$, and then computing the power spectrum at each time instant using the `freqz` function. A two dimensional heat map of this is then created using the `surf` function. The right plot of Figure 62 below shows the spectrogram of the FM signal to allow a comparison between the two time frequency estimation techniques.

Since the CLMS uses the gradient decent method to update the AR coefficients iteratively, it has the ability to more accurately estimate non-stationary signals compared to the block based approach. We can see that this method is able to successfully track the instantaneous frequency of the FM signal as it changes with time. The time frequency spectrum matches the plot of instantaneous frequency as shown by Figure 60, thus confirming the correct operation of this method. Comparing to the spectrogram, the AR CLMS method results in a more clear time frequency spectrum; the FM frequency peak is narrower and more clearly defined.

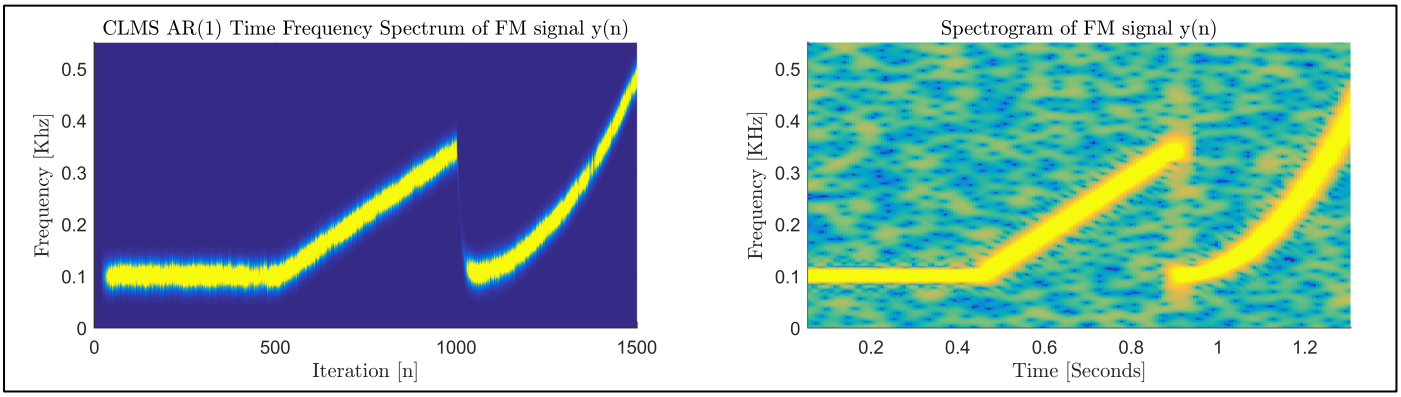

Figure 62: The left plot shows the time frequency spectrum of the FM signal given by equation 4.24, and it is obtained by the AR(1) CLMS algorithm. The right plot shows a spectrogram of the same signal for comparison. For the spectrogram a Hanning window of length 128 is used with 95% overlap and a FFT length of 1024.

4.3 A Real Time Spectrum Analyser Using Least Mean Square

4.3.a Least Squares Solution:

We wish to minimize the sum of squared errors between the estimated signal $\hat{y}(n) = \sum_{k=0}^{N-1} w(k)e^{j2\pi kn/N}$ and the signal $y(n)$, i.e. we wish to minimize the cost function given by equation 4.26. The derivation for this least mean square solution is below. Note that an upper case bold letter denotes a matrix, whereas a lowercase bold letter denotes a column vector.

Cost function to minimise:	$J(\mathbf{w}) = \|\mathbf{y} - \hat{\mathbf{y}}\|^2$	4.26
	$= \|\mathbf{y} - \mathbf{F}\mathbf{w}\|^2$	
	$= (\mathbf{y} - \mathbf{F}\mathbf{w})^H(\mathbf{y} - \mathbf{F}\mathbf{w})$	
Expand out multiplication:	$= \mathbf{y}\mathbf{y}^H - \mathbf{y}^H\mathbf{F}\mathbf{w} - \mathbf{w}^H\mathbf{F}^H\mathbf{y} + \mathbf{w}^H\mathbf{F}^H\mathbf{F}\mathbf{w}$	
Use $(\mathbf{w}^H\mathbf{F}^H\mathbf{y})^T = (\mathbf{w}^H\mathbf{F}^H\mathbf{y}) \in \mathbb{Z}^1$:	$= \mathbf{y}\mathbf{y}^H - 2\mathbf{w}^H\mathbf{F}^H\mathbf{y} + \mathbf{w}^H\mathbf{F}^H\mathbf{F}\mathbf{w}$	
Find gradient of $J(\mathbf{w})$: *use* $\nabla_{\mathbf{x}^*}\{\mathbf{x}^H\mathbf{M}\mathbf{x}\} = 2\mathbf{M}\mathbf{x}$, *and* $\nabla_{\mathbf{x}^*}\{\mathbf{x}^H\mathbf{v}\} = \mathbf{v}$	$\nabla_{\mathbf{w}^*}J(\mathbf{w}) = -2\mathbf{F}^H\mathbf{y} + 2\mathbf{F}^H\mathbf{F}\mathbf{w}$	4.27
Solve for $\nabla_{\mathbf{w}^*}J(\mathbf{w}) = 0$:	$2\mathbf{F}^H\mathbf{F}\mathbf{w} = 2\mathbf{F}^H\mathbf{y}$	
LMS Solution:	$\mathbf{w} = (\mathbf{F}^H\mathbf{F})^{-1}\mathbf{F}^H\mathbf{y}$	4.28

Note that in this derivation, the following theorem is used: If $f(\mathbf{z}, \mathbf{z}^*)$ is a real valued fucntion of the complex vectors $\mathbf{z}$ and $\mathbf{z}^*$, (where $\mathbf{z} = \mathbf{x} + j\mathbf{y}$), then the vector pointing in the direction of steepest decent of $f(\mathbf{z}, \mathbf{z}^*)$ is given by $\nabla_{\mathbf{z}^*}f(\mathbf{z}, \mathbf{z}^*)$, where $\nabla_{\mathbf{z}^*}f(\mathbf{z}, \mathbf{z}^*)$ is defined by 4.30. Thus stationary points are given by $\nabla_{\mathbf{z}^*}f(\mathbf{z}, \mathbf{z}^*) = 0$. (Hayes 1996)

$$\nabla_{\mathbf{z}^*}f(\mathbf{z}, \mathbf{z}^*) = \frac{1}{2}\begin{bmatrix} \frac{\partial}{\partial x_1} + j\frac{\partial}{\partial y_1} \\ \vdots \\ \frac{\partial}{\partial x_n} + j\frac{\partial}{\partial y_n} \end{bmatrix} \tag{4.29}$$

The matrix $(\mathbf{F}^H\mathbf{F})^{-1}\mathbf{F}^H = \mathbf{F}^{-1}$ is given by the equation 4.30 below. The DFT formula is $w(k) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi nk/N}$. Comparing the 4.30 and the DFT formula we can conclude that the matrix $(\mathbf{F}^H\mathbf{F})^{-1}\mathbf{F}^H = \mathbf{F}^{-1}$ is just the DFT matrix scaled by $1/N$.

$$(\mathbf{F}^H\mathbf{F})^{-1}\mathbf{F}^H = \mathbf{F}^{-1} = \left[\frac{1}{N}\right]\begin{bmatrix} 1 & 1 & 1 & 1 & \dots & 1 \\ 1 & W_N & W_N^2 & W_N^3 & \dots & W_N^{N-1} \\ 1 & W_N^2 & W_N^4 & W_N^6 & \dots & W_N^{2(N-1)} \\ 1 & W_N^3 & W_N^6 & W_N^9 & \dots & W_N^{3(N-1)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & W_N^{N-1} & W_N^{2(N-1)} & W_N^{3(N-1)} & \dots & W_N^{(N-1)(N-1)} \end{bmatrix}, \quad where \ W_N \triangleq e^{-j2\pi/N} \tag{4.30}$$

4.3.b Intuitive Fourier Transform Explanation:

The N point DFT is a method of expressing a time discrete function (of length N samples) in terms of the sum of its projections onto a set of N time discrete basis functions (also of length N), where the k^{th} basis function is given by $\left[1,\ e^{2k\pi/N},\ e^{4k\pi/N},\ \dots,\ e^{2(N-1)n\pi/N}\right]^T$. This projection is performed by the projection matrix given by 4.30, and this is such that the MSE between the signal $y(n)$ and the signal consisting of the weighted sum of the basis functions $\hat{y}(n) = \sum_{k=0}^{N-1} w(k)e^{j2\pi nk/N}$ is minimized. In fact the MSE is zero.

The Fourier transform is the continuous version of this same principle. In this case, there are an infinite number of basis vectors than span infinite time and are given by $e^{j\omega t}$. Since the function is continuous, the projection is given by the integral version of the inner product, so $\mathrm{X}(\omega) = \int_{t=-\infty}^{+\infty} y(t)e^{-j\omega t}\,dt$. The signal $y(t)$ can then be reconstructed from the sum (integral) of all of its projections $\hat{y}(t) = \frac{1}{2\pi}\int_{\omega=-\infty}^{+\infty} X(\omega)e^{j\omega t}\,d\omega$. Once again the MSE between $y(t)$ and $\hat{y}(t)$ is zero (assuming the Fourier transform exists and converges). The Fourier transform is essentially an extension of the Fourier series in the case when the period of the periodic signal limits to 0, and the Fourier series can actually be derived by minimizing the MSE between a periodic signal $y(t)$ and the sum of weighted orthogonal harmonic sinusoids (the basis functions) with the fundamental frequency the same as that of $y(t)$.

4.3.c DFT-CLMS Algorithm applied to FM signal:

The DFT-CLMS is applied to the FM signal as described by equation 4.24. The leftmost plot of Figure 63 below shows the magnitude of the weight vector at every time instant. Comparing this to Figure 62, we see that this method is not as good as the AR-CLMS method used in section 4.2.b. The spectrum obtained from the weights of the DFT-CLMS does not resemble the true power spectrum, and we see that the weights appear to be accumulating with each iteration rather than adapting. This accumulating effect can be explained by considering what the weights converge to. When the weights are initialized to zeros, it can be shown that the weight vector $\mathbf{w}(n) = [w_0(n),\ w_1(n), \dots,\ w_{N-1}(n)]$ at time instant n is given by equation 4.31 below:

$$\mathbf{w}(n) = \mu \sum_{m-0}^{n-1} y(m)\mathbf{x}(m), \qquad for \quad n = 0,1,\dots,N, \qquad where\ \ \mathbf{x}(m) = \frac{1}{N}\left[1,\ e^{\frac{j2m\pi}{N}},\ e^{\frac{j4m\pi}{N}},\dots,\ e^{\frac{j(N-1)m\pi}{N}}\right]^T \tag{4.31}$$

When n is equal to N (which is the number of weights) equation 4.31 is just the N point DFT scaled by μ/N, where the $1/N$ factor arises from the definition of $\mathbf{x}(n)$. This result is verified by Figure 64 which plots the scaled N point DFT and the weights of the DFT-CLMS at the N^{th} iteration.

Since by the N^{th} iteration, the DFT-CLMS converges to the N point DFT solution, we do not expect the time frequency spectrum to resemble the true power spectrum. This is because the DFT is a block based approach which is unsuitable for non-stationary signals (unless the spectrogram approach is taken), and the DFT-CLMS is essentially an alternate way to arrive at the DFT solution. One solution proposed is to use the leaky CLMS algorithm in order to counteract the cumulative effect of the weights. This actually yields promising results and the time frequency spectrum is shown by the right plot in Figure 63 below. This clearly shows the frequency modulation of the signal and matches the AR-CLMS method (Figure 62, left). In fact, the leaky DFT-CLMS results in power spectra with a narrower and clearer frequency peak corresponding to the instantaneous frequency compared to the LMS-AR method.

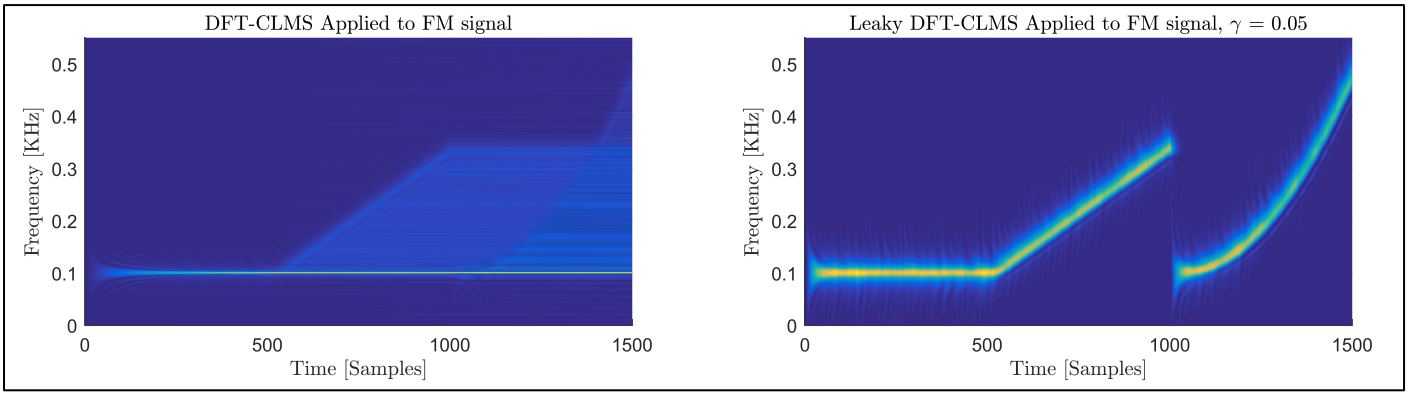

Figure 63: The left plot shows the time frequency spectrum obtained by the DFT-CLMS algorithm applied to FM signal. The right plot shows a leaky DFT-CLMS version. Both use an adaptation gain of μ=1

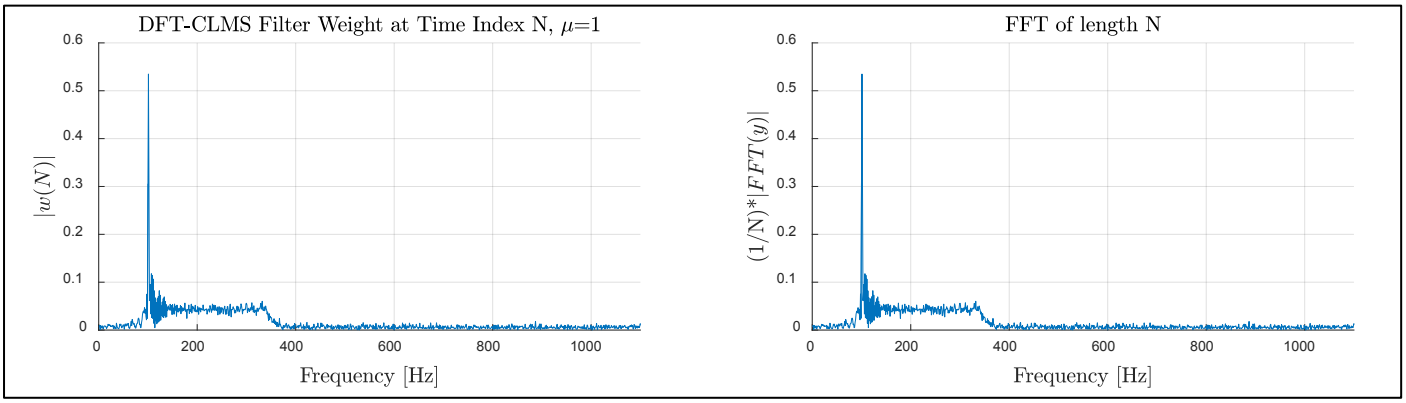

Figure 64: Shows that the N^{th} iteration of the weighs of the DFT-CLMS algorithm converges to the N point DFT solution. The DFT is scaled by $1/N$ due to the factor of $1/N$ in x(m) (equation 4.31). The mean square error between these two plots is only $3.8322 \times 10^{-29} \approx 0$.

In applying the DFT-ACLMS algorithm to the FM signal, a cumulative effect in the time evolution of the weights was observed, and this is now investigated further. The left plot of Figure 65 below shows the evolution of two weights corresponding to the $100\ Hz$ and $350\ Hz$ frequency bins. From this we see that while a particular frequency exists in the FM signal the corresponding weight increases in magnitude with each iteration. For example, considering the weight corresponding to the $100\ Hz$ frequency bin, we see that its magnitude increases from the 1^{st} to the 500^{th} iteration. Looking back at Figure 60, we know the frequency of the FM signal is $100\ Hz$ from the 1^{st}sample to the 500^{th}sample. Between samples 500 to 1000, there is no $100\ Hz$ component in the FM signal and so the $100\ Hz$ weight is fairly constant (it does not decrease to zero as we would like). At the 1001^{st} sample, the FM signal frequency goes back to $100\ Hz$ and we see the corresponding weight increase in magnitude again. These observations lead to another method to estimate the time frequency spectrum using the DFT-CLMS algorithm. The right plot of Figure 65 shows the time frequency spectrum obtained by time differentiating the time evolution of the DFT-CLMS weights. Although this is not as clear as Figure 63 (right), we are still able to track the instantaneous frequency.

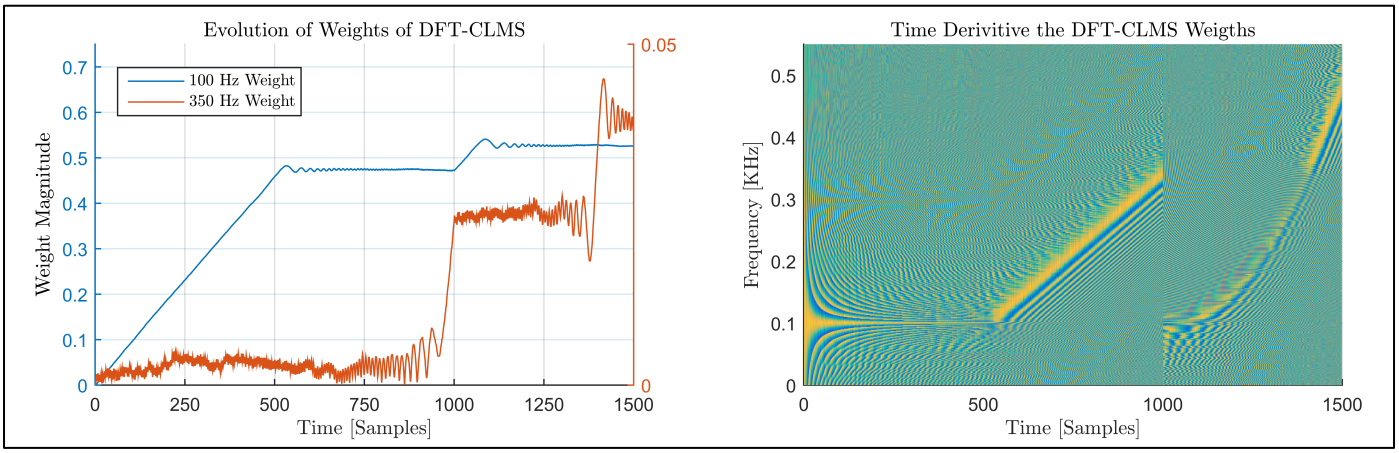

Figure 65: Investigating the time evolution of the weights, in particular the cumulative effect. The left plot shows the evolution of two weights corresponding to the $100\ Hz$ and $350\ Hz$ frequency bins. The right plot shows the time frequency spectrum of the time derivative of the weights of the DFT-CLMS algorithm. The time derivative is obtained by the `diff` function.

4.3.d DFT-CLMS Algorithm applied to Real World EEG Data:

The DFT-CLMS is applied to the real world POz Data and the time frequency spectrum is shown by Figure 66 below. This has been obtained by using only 1200 samples starting from the 1500^{th} sample. We must use only a few thousand samples due to the memory and computation requirements of this algorithm. Since each weight is stored at every time instant, the memory required is of $O(NL)$ where N is the number of weights and L is the signal length. Furthermore, the number of multiplications required is also of $O(NL)$. The 1500^{th}sample is chosen as the starting point since this segment of data has low power in the alpha bands, thus allowing for easier identification of the SSVEP at $13\ Hz$.

The rightmost time frequency diagram in Figure 66 has its colour scale adjusted such that the SSVEP can be more easily identified. From this we can identify the SSVEP at around $13\ Hz$ as well as the second harmonic at around $26\ Hz$. We can also see the $50\ Hz$ mains noise, and some alpha rhythm activity at around $10\ Hz$. These frequencies are only visible at after around 600 iterations, and this is because some number of iterations is necessary for the weights to converge and increase in amplitude before they are visible on the time frequency spectrum.

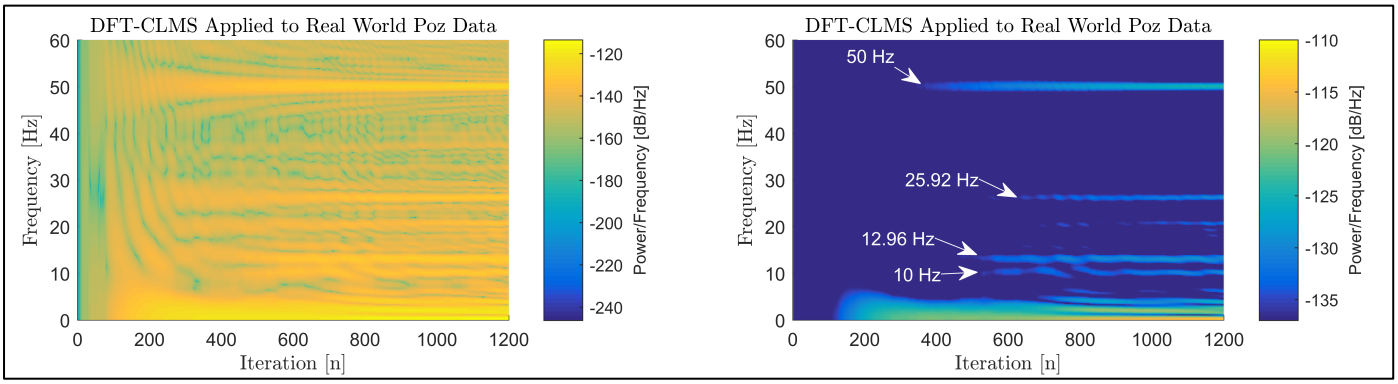

Figure 66: Time frequency spectrum obtained via the DFT-CLMS applied to a segment (samples 1500 to 2700) of POz data. The time frequency spectrum on the right shows an adjusted colour scale which enables us to identify the SSVEP at $\sim 13\ Hz$ and its second harmonic at $\sim 26\ Hz$.