Imperial College London

COURSEWORK EE4-13

IMPERIAL COLLEGE LONDON

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

Adaptive Signal Processing and Machine Intelligence

Author:
Shamim Ahmed (CID: 00736898)

Date: April 11, 2019

Contents

1 Classic and Modern Spectrum Estimations **3**
1.1 Properties of Power Spectral Density (PSD) . . . 3
1.2 Periodogram-Based Applied to Real World Data . . . 3
1.3 Correlation Estimation . . . 5
1.4 Spectrum of Autoregressive Processes . . . 9
1.5 Real World Signals: Respiratory Sinus Arrhythmia from RR-Intervals . . . 10
1.6 Robust Regression . . . 13

2 Adaptive Signal Processing **15**
2.1 The Least Mean Square (LMS) Algorithm . . . 15
2.2 Adaptive Step Sizes . . . 19
2.3 Adaptive Noise Cancellation . . . 22

3 Widely Linear Filtering and Adaptive Spectrum Estimation **26**
3.1 Complex LMS and Widely Linear Modelling . . . 26
3.2 Adaptive AR Model Based Time-Frequency Estimation . . . 30
3.3 A Real Time Spectrum Analyser Using Least Mean Square . . . 32

4 From LMS to Deep Learning **36**
4.1 Non-Stationary Time Series . . . 36
4.2 Activation Function . . . 36
4.3 Scaling Activation Function . . . 36
4.4 Non-Zero Mean . . . 37
4.5 Epochs and Overfitting . . . 37
4.6 Backpropagation Algorithm . . . 38
4.7 Deep Neural Network Training . . . 38
4.8 Deep Neural Network Training - Change in Noise Power . . . 38

1 Classic and Modern Spectrum Estimations

1.1 Properties of Power Spectral Density (PSD)

1.1.1 Approximation in PSD

In order to equate both equation, we will need to begin from Power Spectral Density (PSD). After a couple of rearrangements, and after substituting $\tau = m - n$, and joining the 2 termed summed into one, it produces:

$$
\begin{aligned}
P(\omega) &= \lim_{N\to\infty} E\left\{\frac{1}{N}\left|\sum_{n=0}^{N-1} x(n)e^{-jn\omega}\right|^2\right\} = \lim_{N\to\infty} E\left\{\frac{1}{N}\sum_{m=0}^{N-1} x(m)e^{-jm\omega}\sum_{n=0}^{N-1} x^*(n)e^{jn\omega}\right\} \\
&= \lim_{N\to\infty}\frac{1}{N}\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} E\left\{x(m)x^*(n)\right\}e^{-jm\omega}e^{jn\omega} = \lim_{N\to\infty}\frac{1}{N}\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} r_{xx}(m-n)e^{-j(m-n)\omega} \\
&= \lim_{N\to\infty}\frac{1}{N}\sum_{\tau=-N+1}^{N-1}(N-|\tau|)r_{xx}(\tau)e^{-j\tau\omega} \\
&= \sum_{\tau=-\infty}^{\infty} r(\tau)e^{-j\tau\omega} - \lim_{N\to\infty}\frac{1}{N}\sum_{\tau=-N+1}^{N-1}|\tau|r(\tau)e^{j\tau\omega}
\end{aligned}
$$

Hence, the from the above, it could be observed that the continuous DTFT of the Autocovariance Function (ACF) is equivalent to the PSD under the small assumption that the covariance sequence $r(k)$ decays rapidly.

$$
P(\omega) = \sum_{\tau=-\infty}^{\infty} r(\tau)e^{-j\tau\omega} \tag{1}
$$

1.2 Periodogram-Based Applied to Real World Data

1.2.1 Spectral Estimation Techniques

The figures below, displays the sunspot time series and its associated periodogram, which uses a Chebyshev window; the raw sunspot time series is analysed and compared with two other time series, which are both pre-processed.

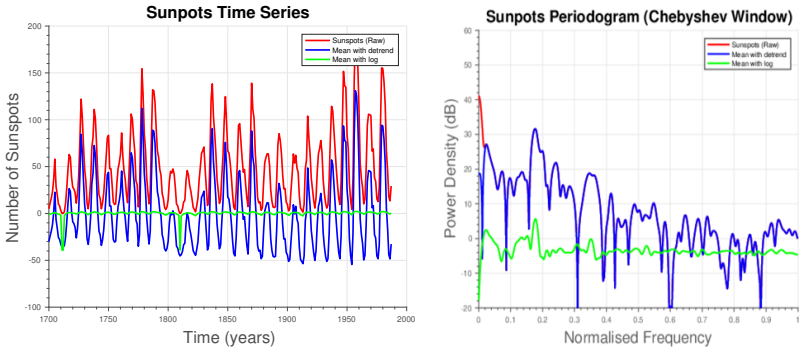

Figure 1: Sunspots Time Series: Raw, Pre-processed and Chebyshev-windowed periodograms

To remove the low frequency components, the function `detrend` is utilised, while to remove the DC components at f= 0Hz, subtraction of the `mean` is required. The lower frequency components of the centred and detrended time series spectral estimate are removed but mainly for frequencies $f \gtrapprox 0.02$ $(rad/sample)$ which the graph is nearly identical to the raw sunspot time series.

By summing a tiny constant, (MATLAB `eps` $= 2.2204e-16$), with raw sunspot time series and thereby applying a natural `log` and thus eliminating its mean, the logarithmic series can be produced along with escaping any logarithms of zero. The peaks greater than $0dB$, are of interest, and more apparent due the DC component being eliminated hence at the same frequencies, the peaks are seen.

1.2.2 EEG/POz Signals

Figure 2, displays the EEGs Standard and Bartlett method periodograms; the Bartlett-Periodogram has its peaks more emphasized hence it is less complicated to interpret. The main peaks of interest from the periodogram is the wide peak at $8-10\ Hz$ and the other peaks at frequencies $f = 13,\ 26,\ 39,\ 50\ Hz$ respectively.

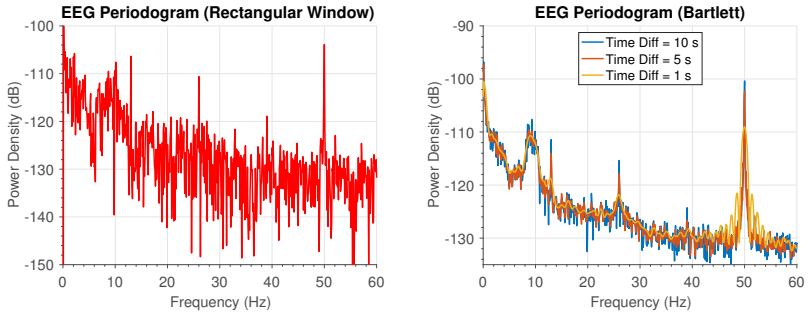

Figure 2: Sunspots Time Series: Raw, Pre-processed and Chebyshev-windowed periodograms

Irrespective of the window length Δt used, the components observed at $8-10\ Hz$ are greatly distinguished by the Bartlett method periodograms; this was caused by the subjects tiredness during the recording. At frequencies, $f_1^{SSEVP} = 26\ Hz$ and $f_2^{SSEVP} = 39\ Hz$, the harmonics SSEVP can be observed, whereas at the fundamental frequency can be observed from the peak at $f_0^{SSEVP} = 13\ Hz$. Due to the power-line interference, a strong component $f^{PLI} = 50\ Hz$ is observed which effects the observation of the third harmonic $f_3^{SSEVP} = 52\ Hz$ in both the standard periodogram and the Bartlett method periodogram, when $\Delta t = 10\ s$.

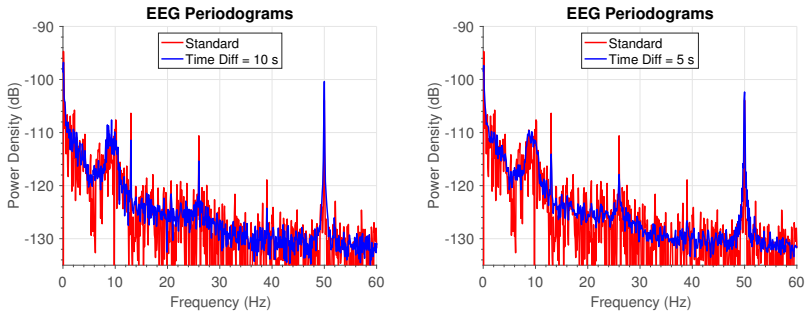

Figure 3: Sunspots Time Series: Raw, Pre-processed and Chebyshev-windowed periodograms

For varying averaged window length i.e when $\Delta t = 10\ s$ and $\Delta t = 1\ s$, figure 3, displays a comparison between the Standard and the Bartlett periodogram. Compared to the standard periodogram, the Barlett periodogram has a reduced variance but also reduced resolution when $\Delta t = 10\ s$. Regardless, the $8 - 10\ Hz$ band, the harmonics of SSEVP and power-line interference frequency are the important peaks which are distinguishable. However, only SSEVPs 3^{rd} harmonic is unable to be observed due to the low resolution associated when the $\Delta t = 1\ s$, even though the variance has reduced by 10 times. All in all, the variance-precision trade-off is observed: For $\Delta t = 1\ s$, the Bartlett method periodogram has the least variance but the worst precision while the standard periodogram has the largest variance with the best precision.

1.3 Correlation Estimation

1.3.1 Biased and Unbiased ACF

For the 3 specific signals: white Gaussian noise, filtered white Gaussian noise and a sine-wave, the corresponding the biased and unbiased estimations of the correlogram spectra and auto-correlation function (ACF) is displayed in Figure 4.

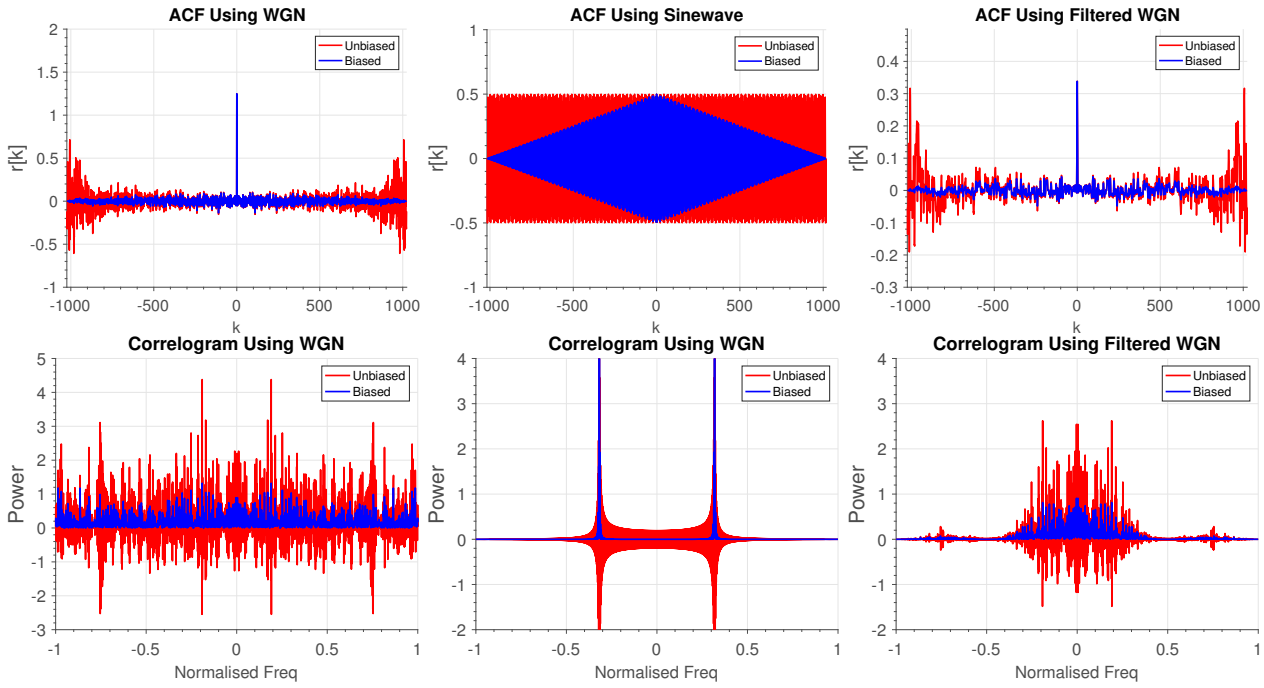

Figure 4: Biased and Unbiased Estimates of ACF and their Effect on the Correlogram

As seen above, the spectral window chosen is critical; the effects of an incorrect window is evident in correlogram's where negative power values are obtained from the unbiased ACFs spectral estimates, which is impossible. Moreover, it is further evident that when the lags are relatively small, the ACF estimates are comparable conversely when lag increases the estimates begin to reduce in similarity. Thus, a rectangular window is imposed on the ideal ACF by an unbiased estimator where as a Bartlett window is imposed on an ideal ACF by the biased estimator function.

Before the spectral estimates are computed, windowing in the time domain allows one to comprehend this problem more thoroughly. As seen from figure 4, for the spectral estimate, the positive semi-definitness is no longer retained when a window is applied to generate the unbiased estimate of ACF for a time-domain signal. However, as also observed from figure 4 when a rectangular window is applied to a time domain signal, the biased estimator of the ACF can be acquired. Moreover, for a rectangular window the power spectrum retains its positive semi-definiteness as can be seen in the calculated periodogram from the time domain signal, where the window magnitude spectrum is squared.

1.3.2 Correlogram Estimates

A hundred realizations of both the distinctive random processes are shown in figure 5, where shown in red, is the mean of the 100 realizations. 2 sinewaves are presented in each of the random processes are seen from the plots.

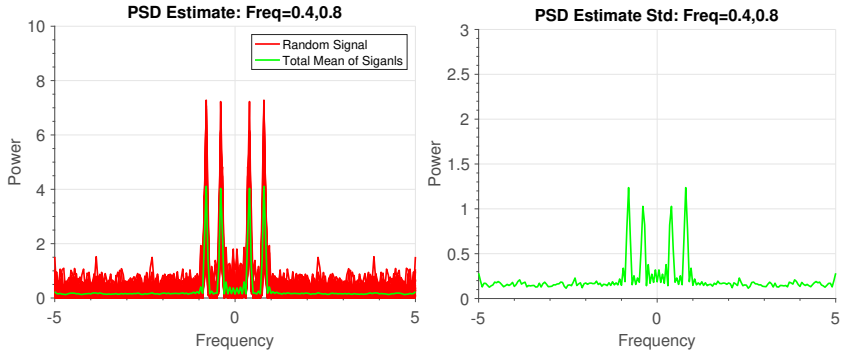

Figure 5: Spectral Estimates of Correlogram (Standard Deviation & Variance)

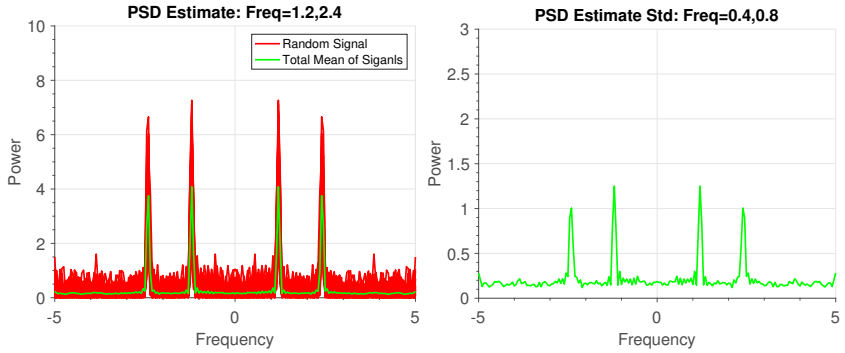

Figure 6: Spectral Estimates of Correlogram (Standard Deviation & Variance)

It could be observed from the figure above that a periodogram produces an inconsistent estimator, which is known to be one if its limitations; for the same frequencies, the standard deviation and periodogram presents the same peaks for the 100 spectral estimates. This can be evident from the frequencies at f=$0.4, 0.8$Hz & f=$1.2, 2.4$Hz in the standard deviations plots in figure 5. **Hence var**$(P_{per}) \to 0$ **as** $N \to \infty$ **is needed for the periodogram to become a consistent estimator**. Thus various other spectral estimation methods must be assessed to reduce the variance, such as the Blackman-Tukey technique; the variance will not diminish by merely increasing the number of samples.

1.3.3 Correlogram Estimates using dB Scale

Another interesting phenomena occurs when the dB scale is used to plot the above graphs which is that the effects described above become more distinct and apparent thus the windowing effects on sinwaves are also more apparent, hence the peaks at f=1.2, 2.4Hz display a sort of $sinc^2$ shape feature.

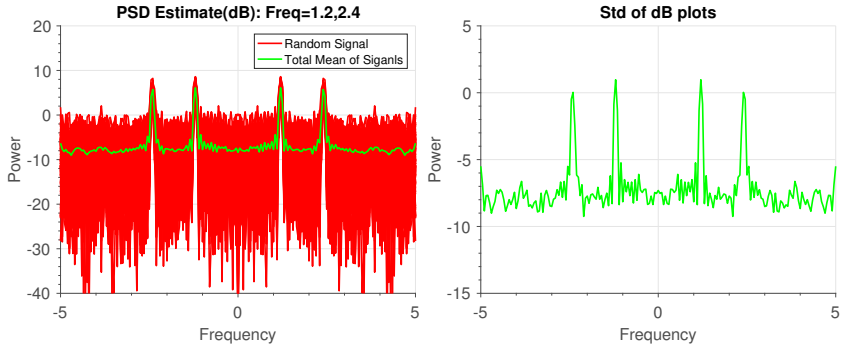

Figure 7: Spectral Estimates of Correlogram Effects from dB scale (Standard Deviation & Variance)

1.3.4 Periodogram Resolution

From figure 8 below, it can be concluded that when the sample size increases, the peaks become more apparent and distinguished. This is due to $\frac{1}{N}$ being proportional to a periodograms frequency resolution hence two exponential signals, both complex in nature, can be distinguished by having a resolution of 0.02 Hz; one signal of frequency 0.3Hz and the other having a frequency of 0.32Hz with approximately a sample size of 50. However, for complex functions, the periodograms are non-symmetric due their complex nature thus the one-sided periodogram is computed with the f_s, not $\frac{f_s}{2}$.

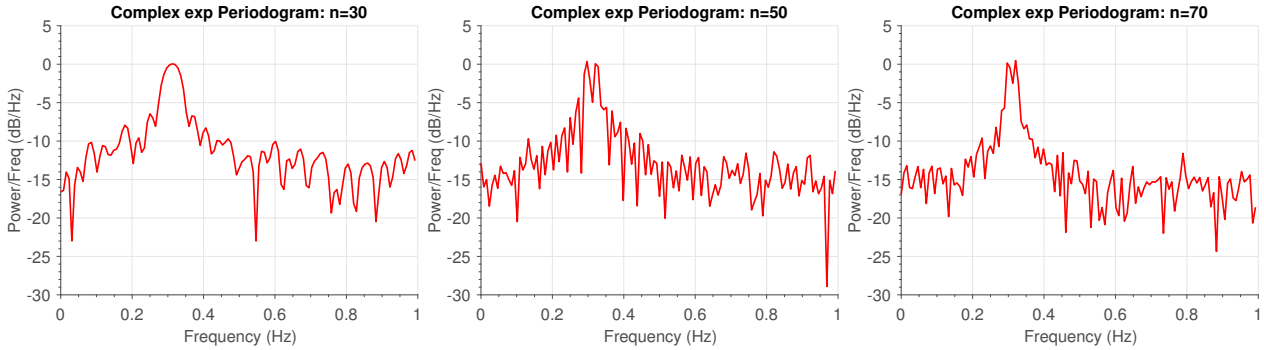

Figure 8: Resolution of Periodogram Spectral Estimates When Increasing Sample Number

1.3.5 MUSIC Frequency Estimation

To model the ACF matrix $\mathbf{R}_{xx}$ which can be set as the sum of ACF matrices, such as the signals ACF matrix $\mathbf{R}_s$ and the noises ACF matrix $\mathbf{R}_n$, a frequency estimation method can utilised, such as the MUSIC technique (MUltiple SIgnal Classification method). Thus equation 2, is the given form of a signal which possesses only a single complex exponential

$$x(n) = A_1 e^{jn\omega_1} + w(n) \therefore \mathbf{e}_i^H \mathbf{v} = \sum_{k=0}^{M-1} v(k) e^{-jk\omega_i} = 0,\ i = 1, 2, \ldots, p \tag{2}$$

The eigenvectors for the matrix $\mathbf{R}_{xx}$ are orthogonal as it is a Hermitian matrix, which is present in equation in form $\mathbf{e}_i^H \mathbf{v}$, where $\mathbf{e}_i$ and $\mathbf{v}$ represents an eigenvector from matrix $\mathbf{R}_s$ and $\mathbf{R}_n$ & $\mathbf{R}_{xx} \in \mathbb{R}^{M \times M}$ respectively, where also present in the signal are complex exponential represented as p. By using the formula above when $\mathbf{e} = [1, e^{j\omega}, e^{j2\omega}, \ldots, e^{j(M-1)\omega}]$, the power spectrum could also be estimated. Due to the periodic nature of complex exponentials, which is a requisite for the DFT (Discrete Fourier Transform), only a finite amount of unique complex exponentials are all comprised with the vector $\mathbf{e}$, for a discrete-time signal. As mentioned earlier, due to the periodic nature of complex exponential, the associated power spectrum can estimated using the equation, $\hat{P}_{MU}(e^{j\omega}) = \frac{1}{\sum_{i=p+1}^{M} |\mathbf{e}^H v_i|^2}$.

The complex exponential which are present in the signal, should present p roots which lie on the unit circle, which respectively correspond to their frequency which is due to the inner product of the eigenvectors $\mathbf{e}$ and v_i being orthogonal. Nevertheless, the inner product of the eigenvectors will have $M - p - 1$ number zeros on the z-plane due to $\mathbf{R}_{xx} \in \mathbb{R}^{M \times M}$ where $M > p + 1$. This can imply that a signals complex exponentials can be incorrectly identified as false peaks in the power spectrum if the corresponding zeros lie very near to the unit circle. By averaging all the $M - p - 1$ eigenvectors corresponding to the noise, the likelihood of the false peaks being present can be greatly reduced. The magnitude of each true peak can amplified by averaging hence this can make identifying complex exponential in the signal much simpler

It could be analyzed from the inner workings of the algorithm that the ACF matrix $\mathbf{R}_{xx} \in \mathbb{R}^{M \times M}$ is given by the first line. The value of M is chosen on quantity of trusted ACF values. The accuracy of the ACF is significantly reduced when the lag increases, which occurs when there are larger values of M, which is can be observed in figure 9. The spectral estimation is carried via utilising the MUSIC algorithm in second line of code where ACF matrix and the signals subspace dimensions are provided to first and second arguments respectively. Moreover, the FFT length and sampling period is provided in third and fourth arguments respectively whereas the in the fifth argument allows the function to operate on the first augment as a correlation matrix instead of a data matrix.

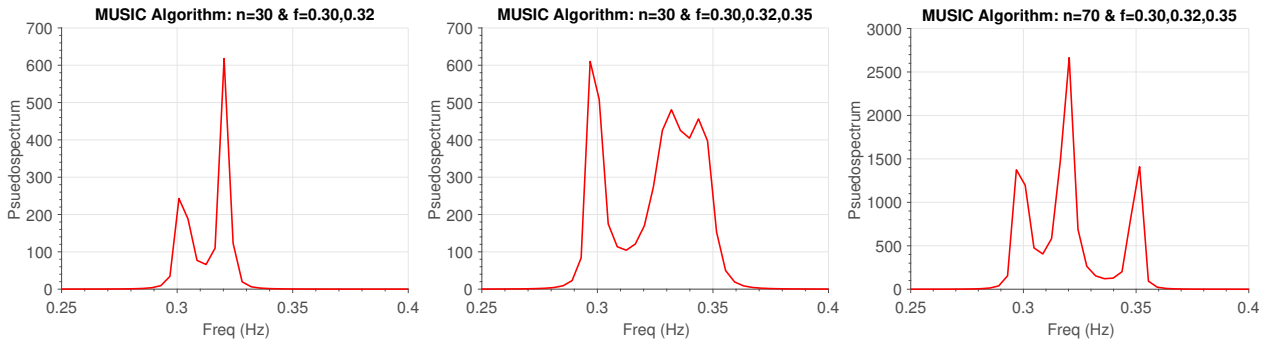

Figure 9: The effect on the resolution when increasing the sample number whilst using the MUSIC algorithm to detect sinewaves

As seen from the resulting figure 9, for a random signals x, a single realization is displayed in the first row, where as a 100 realizations are overplayed in the second row. Hence it can be

ascertained that the MUSIC algorithm, for 3 sinewaves very narrowly placed alongside each other, become very difficult to identify hence possessing a frequency resolution limit, similar to that of the periodogram is required. Therefore, it is possible to distinguish between the sinewaves by increasing the sample number which increases the frequency resolution. Hence a for the identification of complex exponentials when the availability of samples are very limited, the MUSIC algorithm can be utilised in such scenarios, which can be evident from the comparison between figure 9 and figure 10; with the MUSIC algorithm utilised for only 30 samples, the correct identification of the locations of the two peaks where made unlike the periodogram.

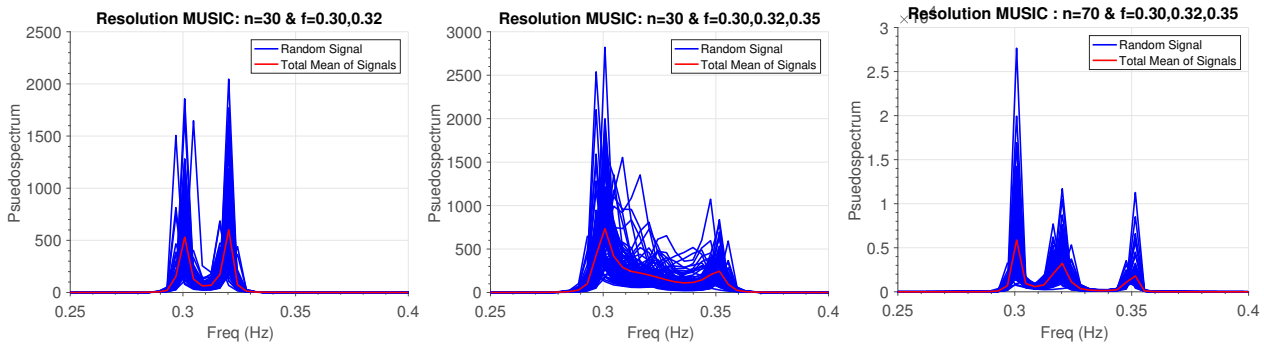

Figure 10: The effect on the mean resolution when increasing the sample number whilst using the MUSIC algorithm to detect sinewaves

However there are several disadvantages posed by these pseudo-spectrums. Firstly, the signals subspace dimension, which are not always known, are required as one of the algorithms inputs. Secondly the respective complex exponential magnitudes information are not provided; evident from signal x which was made with equal magnitudes for every respective complex exponential. Moreover, unfavourable effects such as eliminating noise spectra information can be utilised for the identification on non-white noise. The effect is a result, when averaging is implemented in order to eliminate any unwanted peaks from the spectra.

1.4 Spectrum of Autoregressive Processes

1.4.1 Autoregressive Parameter Estimations

By utilising the equations, $\mathbf{r}_{xx} = \mathbf{R}_{xx}a \implies a = \mathbf{R}_{xx}^{-1}\mathbf{r}_{xx}$, the Autoregressive (AR) parameter estimation can be applied. When using the biased estimator, the ACF can be calculated, where $\mathbf{R}_{xx}$, is positive definite hence the invertibility of matrix $\mathbf{R}_{xx}$ is essential. However, if the unbiased estimator is used, the positive definite nature of matrix $\mathbf{R}_{xx}$ is not guaranteed.

1.4.2 Autoregressive Models

When a signal $x(n)$ exhibits an AR process which have an order of 4, 8, 10 and 14, the resulting effects of the model order increasing becomes more apparent, as seen in figure 11 which displays the spectrum attained for the different order numbers, where the difference between an 4th order model and an 8th order model is obvious. As evident, the ideal spectrum has two distinct spike for the 8th order model, which are distinguished unlike the 4th order model. More variability of pole placements in the spectrum is achievable through increasing the model order above 8, which allows for a greater degree of freedom however

there is a trade off in accuracy as the subsequent spectrum is not as accurate. This is further proven when, incorrectly the same magnitude is reached by the both peaks when the number poles are increased. Thus, an accurate parameter estimation of an AR process, may not have sufficient information when even when 500 samples are utilised for an estimation.

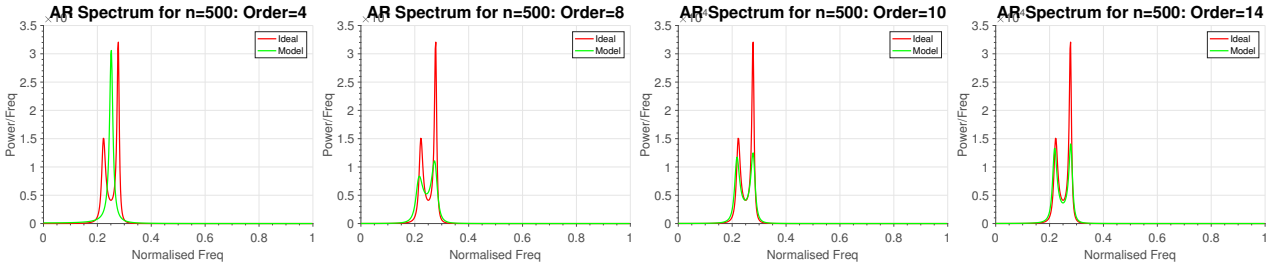

Figure 11: Autoregressive Model Power Spectrum Estimates

1.4.3 Over/Under Fitting Autoregressive Models

As shown in figure 12, a more accurate spectrum is generated when the sample number is increased. Further evidence is proving that the 500 samples do not hold sufficient AR process information which can be seen when the 4th order model estimate is not able to produce the two data spikes when the sample number is 10000. Nevertheless, both peaks to do not tend the same magnitude when the model order is increased to 12. However, overfitting the training data will start emerging when the model order is 12. The signal has become greatly periodic due the power of the signal being compress into a very small frequency band. Also, compared to the ideal spectrum peaks, the estimated spectrum peaks have been overestimated. Conversely, the reduction in the estimation error has been reduced by increasing the model order. Hence, the methods such as Minimum Description Length (MDL) or the Akaike Information Criterion (AIC) can be used to minimise the trade-off between the estimation error and generalization error for higher order models.

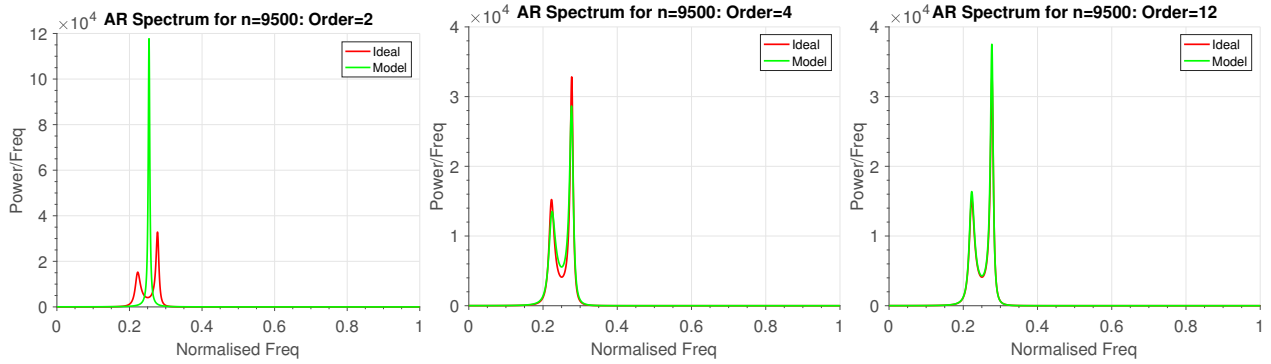

Figure 12: Autoregressive Models effects from Over/Under Fitting

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

1.5.1 Standard Periodogram of RRI trials

Figure 13, displays for 3 sets of RRI data , which are all unique, their respective standard periodogram. For each respective RRI trial, the averaged periodogram is obtained for various window lengths: 50, 150 & 200 samples as shown in figure 14 (without using overlapping

segments). For each segment, a hamming window, which can suppress the magnitude of the side-lobes, was used. In the occurrence of noise, to accurately identify the RSA harmonics, the Hamming window can minimize the extent of spectral leakage in the resulting spectrum.

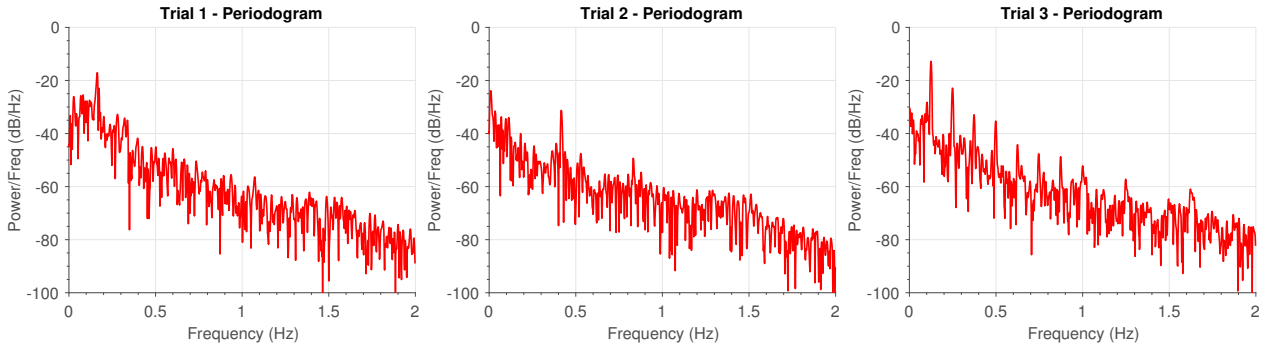

Figure 13: RRI Trial's Standard Periodograms

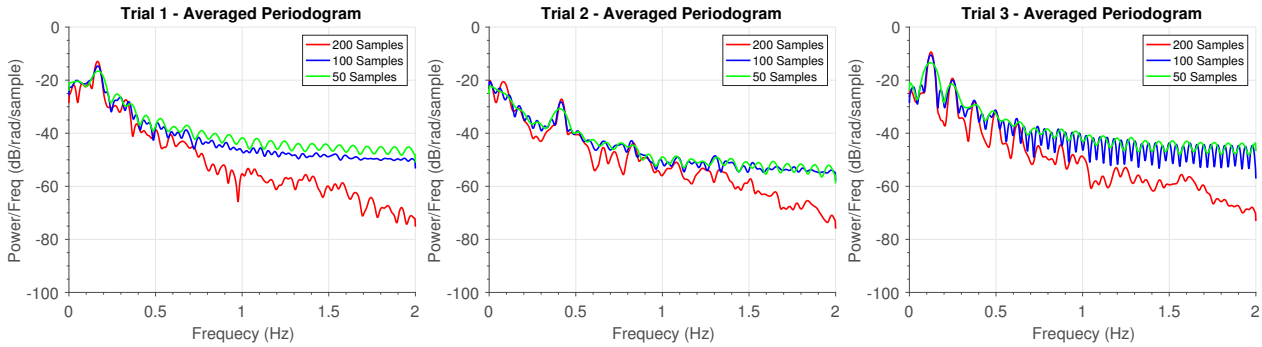

Figure 14: RRI Trial's Averaged Periodograms (Varying window lengths)

1.5.2 PSD Estimates Differences

The protocol for the 3 trials were such: **Trial 1** - Typical Ventilation Rate 20-45 Breaths Per Minute, BPM (Unconstrained Breathing) **Trial 2** - Ventilation Rate 50 BPM (Fast Breathing) **Trial 3** - Ventilation Rate 15 BPM (Slow Breathing). For each trial, from the Periodogram, the first harmonics of the RSA where analysed: **Trial 1** - breathing rate of $2{\times}60{\times}0.1641 = 19.69 \approx 20$ BPM due a peak at 0.1641 Hz. **Trial 2** - breathing rate of $2 \times 60 \times 0.416 = 49.92 \approx 50$ BPM due to a peak at 0.416 Hz. **Trial 3** - breathing rate of $2 \times 60 \times 0.125 = 15$ BPM due to a peak at 0.125 Hz

By increasing the sample number size, the estimates of the averaged periodograms are much smoother due to the reductions of the variance of the estimates. For trial 1, the second harmonic at f=0.332Hz, which is averaged using 200 samples, is very difficult to be distinguished from the other harmonics in the standard periodogram. Moreover, peaks were present at f=0.2773Hz and f=0.2949Hz, when the periodogram was averaged over 50 and 100 samples respectively; ideally the peaks should have been present at f=0.332Hz. Thus it can be concluded that there is a compromise between in the ability in attaining a higher resolution along with identifying the 2nd harmonic and trying to suppress unwanted noise by utilising smaller segments to average over. Moreover, when averaged over 50 and 100 samples, in other trials, the harmonics are shifted slightly but are more apparent in the averaged periodogram.

1.5.3 RRI Trials Using AR processes

When using the AR process to model the RRI data, the resulting figures are shown in figure 15, 16, 17. In Trial 1, at f=0.1641Hz the only significant periodic frequency component occurs hence it is not possible to use an AR model which can model any other harmonics except for the fundamental one. Moreover, to amplify the component at f=0.1641Hz an increase in model order is required however this will not increase the number of peaks. Whereas, for a periodogram which is averaged across 200 segments, it can model and characterize the harmonics more accurately. Moreover, the artifact which occurs at 1.5Hz has also disappeared, whereas it present the AR process figure.

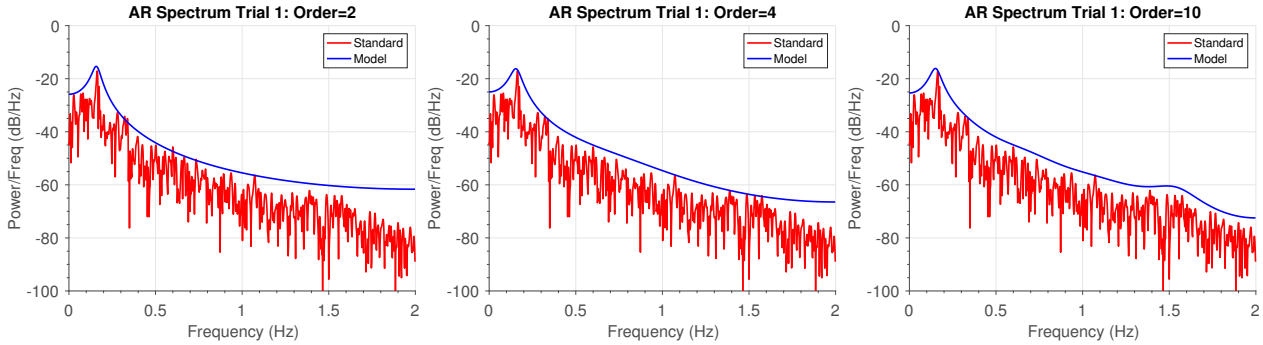

Figure 15: Trial 1: Autoregressive Spectrum Estimate

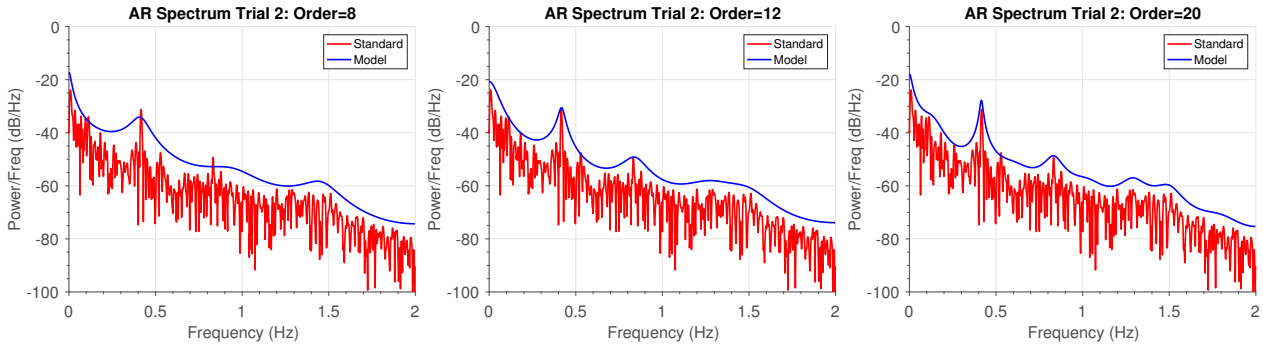

Figure 16: Trial 2: Autoregressive Spectrum Estimate

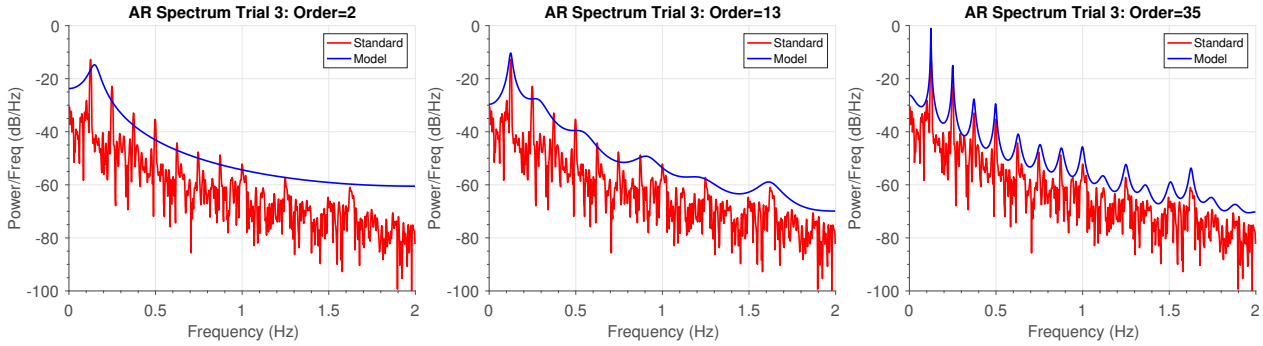

Figure 17: Trial 3: Autoregressive Spectrum Estimate

For Trial 2 &3, both the 2nd and, to some extent, the 3rd harmonics are modelled accurately due to the increase in model order, which allows for a greater degree of freedom to model the harmonics, hence all the peaks can be models correctly when using a model order of 35.

Real world process such as RRI, tend to have very limited memory thus it is discouraged to simulate the peaks present in the standard periodogram in Trial 3 using a very high model order. All in all, the AR model based spectrum estimates produces accurate and smooth data, with a few limited offsets and artifacts present; all of the AR processs limitations are discussed above.

1.6 Robust Regression

1.6.1 Matrix Rank

Figure 18 displays the $\mathbf{X}$ and $\mathbf{X}_{noise}$ respective singular values as well the squared error between each of the singular values. Due to the 3 non-zero singular values present in the noiseless input matrix, $\mathbf{X}$, the rank of the matrix is 3. On the other hand, 3 major singular values, present in the noise corrupted matrix, $\mathbf{X}_{noise}$, results in signal subspace being spanned by the corresponding eigenvectors while the remaining non-zero singular values are associated with the noise subspace. Due to the noise corruption, the singular values from the signal subspace are effected with an offset from their original singular values of $\mathbf{X}$. By simply thresholding, the eigenvalues corresponding to the signal subspaces can be made more apparent due to the eigenvalues of the noise subspace only being half the magnitude. However, the it will be difficult to distinguish the rank of $\mathbf{X}_{noise}$, if the noise power is amplified which allows for its singular value to be of similar magnitude from the eigenvalues in the signal subspace.

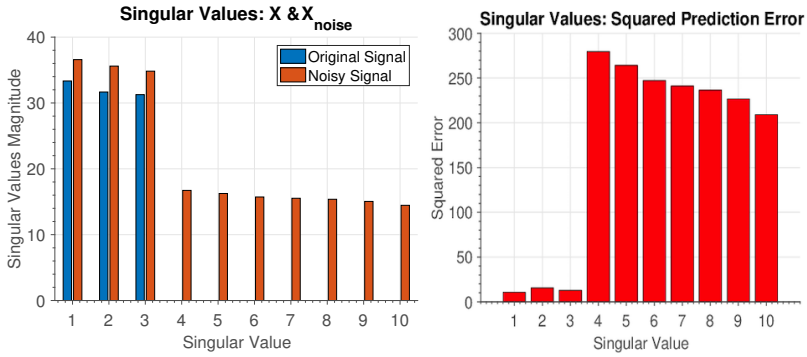

Figure 18: Squared Prediction Error for corrupted signals (Singular Value Decomposition)

1.6.2 Low-Rank Approximation

For varying values of r, the associated approximation error, $\|\mathbf{X} - \tilde{\mathbf{X}}_{noise}\|_F$, is shown in figure 19. By only possessing most principle components of r of the noisy data $\mathbf{X}_{noise}$, the Low-rank approximation was attained which was made possible from the assumption that most important components from r can describe the data, whereas the less significant components of r can be considered to be worthless noise. Thus, this is great ability of the Principle Component Analysis (PCA) possesses, as it had also reduced the error for $r = r_{true} = 3$.

1.6.3 OLS and PCR Comparison

Utilising the OLS and PCR methods, the estimation of the parameter matrix $\mathbf{B}$ can be attained. Figure 20, displays for the train and test data, the associated the estimation errors between $\mathbf{Y}$-$\mathbf{Y}_{OLS}$ and $\mathbf{Y}$-$\mathbf{Y}_{PCR}$ for varying values of r. The OLS outperforms the PCR by 0.4% in training

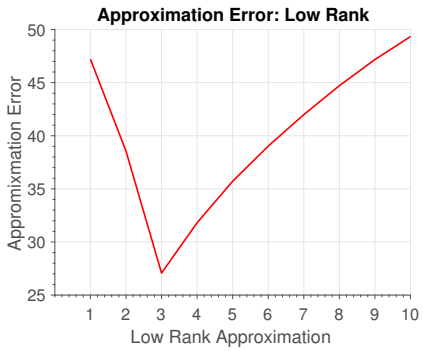

Figure 19: Low-Rank Approximation and squared prediction error for corrupted signal (SVD)

data, but PCR 0.7% greater than the OLS for the test data. All-in-all, the OLS and PCR methods, perform equally great for both test and training data, when $r \geq 3$.

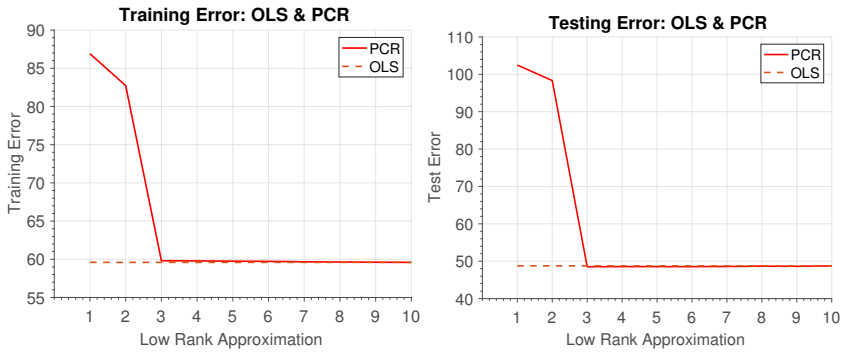

Figure 20: Train and Test Errors (SVD)

1.6.4 Models Evaluation

Across the ensemble of 100 test realizations from the stochastic process producing $\mathbf{X}_{noise}$, both the OLS and PCR models were evaluated. The PCR beats the OLS by 1.2%, which is seen from figure 21 which summarizes, for both models, its prediction errors, thus further ratifying the results from earlier parts.

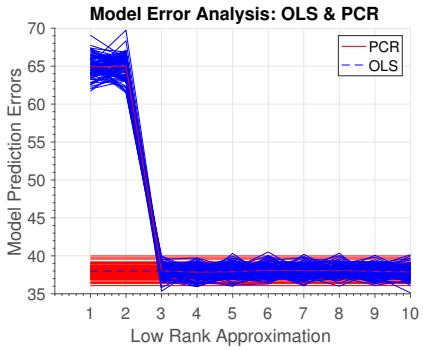

Figure 21: Evaluating both models (SVD)

2 Adaptive Signal Processing

2.1 The Least Mean Square (LMS) Algorithm

2.1.1 LMS Filter & Convergence

For this coursework, the general form of the AR(2) process is given as: $\boldsymbol{x(n) = a_1 x(n-1) + a_2 x(n-2) + \eta(n), where \eta(n) \sim \mathcal{N}(0, \sigma_\eta^2)}$. Since the correlation matrix $\mathbf{R}_x = \mathbb{E}\{\mathbf{x}(n)\mathbf{x}(n)^T\}$, where $\mathbf{x}(n) = [x(n-1), x(n-2)]^T$, the entries of the matrix are shown in equation 3 which is then multiplied with $x(n-k)$ required to determine the entries of $x(n)x(n-l) = a_1 x(n-1)x(n-k) + a_2 x(n-2)x(n-l) + \eta(n)x(n-k)$, who's expectation is given by: $\gamma(k) = \mathbb{E}\{x(n)x(n-k)\} = a_1 \mathbb{E}\{x(n-1)x(n-k)\} + a_2 \mathbb{E}\{x(n-2)x(n-k)\} + \mathbb{E}\{\eta(n)x(n-k)\}$

$$\mathbf{R}_x = \begin{bmatrix} \mathbb{E}\{x(n-1)x(n-1)\} & \mathbb{E}\{x(n-1)x(n-2)\} \\ \mathbb{E}\{x(n-2)x(n-1)\} & \mathbb{E}\{x(n-2)x(n-2)\} \end{bmatrix} \tag{3}$$

The inputted value for the correlation matrix can obtained from the 3 simplified, simultaneous equations below:

$$\gamma(0) = a_1\gamma(1) + a_2\gamma(2) + \sigma_\eta^2$$
$$\gamma(1) = a_1\gamma(0) + a_2\gamma(1)$$
$$\gamma(2) = a_1\gamma(1) + a_2\gamma(0)$$

The resulting values obtained from the ACF matrix, after solving the three simultaneous equation are: $a_1 = 0.1$, $a_2 = 0.8$ and $\sigma_\eta^2 = 0.25$

$$\mathbf{R}_x = \begin{bmatrix} \gamma(0) & \gamma(1) \\ \gamma(1) & \gamma(0) \end{bmatrix} = \begin{bmatrix} \frac{25}{27} & \frac{25}{54} \\ \frac{25}{54} & \frac{25}{27} \end{bmatrix} \tag{4}$$

The convergence inequality for μ, where $0 < \mu < \frac{2}{\lambda_{max}}$ will need to be applied and also be true, which will make it possible through determining the eigenvalues of $\mathbf{R}_x$ which are all needed to for the LMS algorithms mean to converge to determine the range of values for μ. The LMS algorithm, if and only if the inequality condition is met, will the converge to a particular solution named the Winer-Hopf. The LMS will converge given that that $\mathbf{R}_x$s maximum eigenvalue is 1.3889 and hence μ will need to range from $0 < \mu < 1.44$.

2.1.2 LMS Convergence Speed

For a sample size of $N = 1000$ of $x(n)$, a LMS adaptive predictor was utilised with results shown in figure 22 with $\mu = 0.01$ and $\mu = 0.05$m for a single realization and for a 100 realizations which are averaged, respectively. When the error prediction graph is squared, as shown in figure 22, it is conclusive that when the value of $\mu = 0.05$, a more rapid convergence is observed, which was predicted as the error would decease more rapidly and steeply for large values of μ.

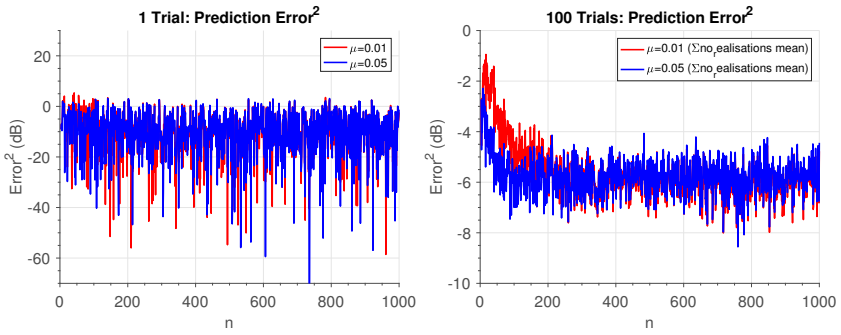

Figure 22: Convergence Speed of LMS Algorithm when varying step sizes

2.1.3 LMS Misadjusments

For a hundred trials of the experiment, all independent, the ensemble average learning curves steady-state graph is shown in figure 23. The EMSE (excess mean square error), the calculated discrepancies and the calculated theoretical discrepancies for 100 independent realizations are displayed in the table below. It is clearly evident that that a reduced discrepancies value is produced due to less EMSE error, only when a smaller step sizes is used. However, a trade-off is encountered as a slower rate of converge occurs when smaller step sizes are used. Moreover, the empirical discrepancies are marginally higher than the theoretical discrepancies.

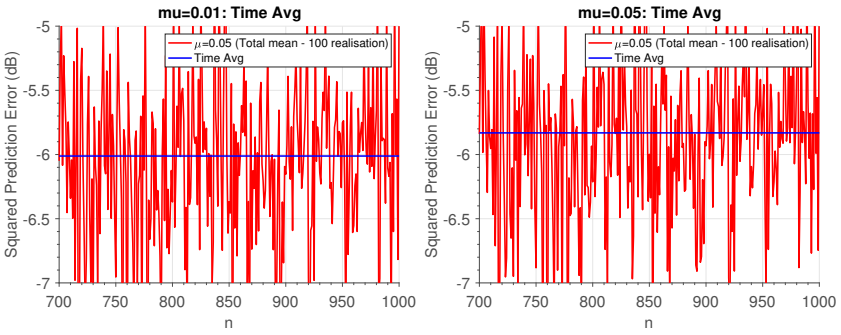

Figure 23: Corresponding Steady-State Error for $\mu = 0.01$ and $\mu = 0.05$ (Time-Average)

μ	EMSE	Empirical	Theoretical
0.01	0.0032	0.0128	0.0093
0.05	0.124	0.0496	0.0463

Table 1: Empirical and Theoretical Misadjustments

2.1.4 Evolution of Coefficients

Due to adaptive filters (real world) functioning only for random signals which have 1 realization, the figure 24 below displays how the coefficient changes with time, thus for $\mu = 0.05$ it evident that there is a trade-off between steady-state error and speed, in addition to the fact, that the ideal values have a substantial amount of disparity compared to the value obtained. It should be also noted, that majority of the graphs that are displayed from now, will be averaged across 100 realizations.

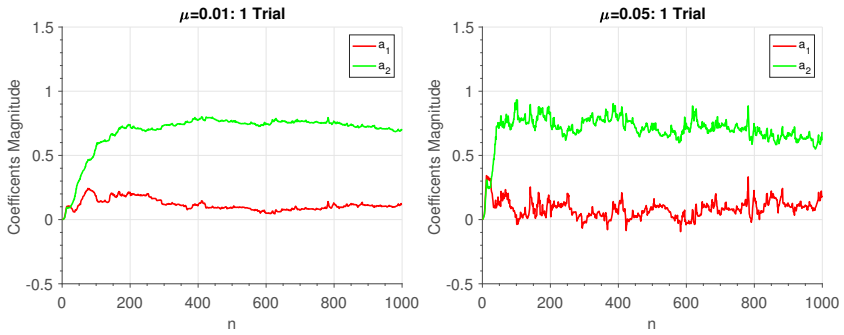

Figure 24: For $\mu = 0.01$ and $\mu = 0.05$, The corresponding Coefficient evolution for 1 Trial

It is also evident, as shown in figure 25, that for smaller values of μ, the steady-state values obtained are nearer $a_1 = 0.1$ & $a_2 = 0.8$ which are the ideal values; moreover the variance for the steady state value are also much smaller when μ is smaller

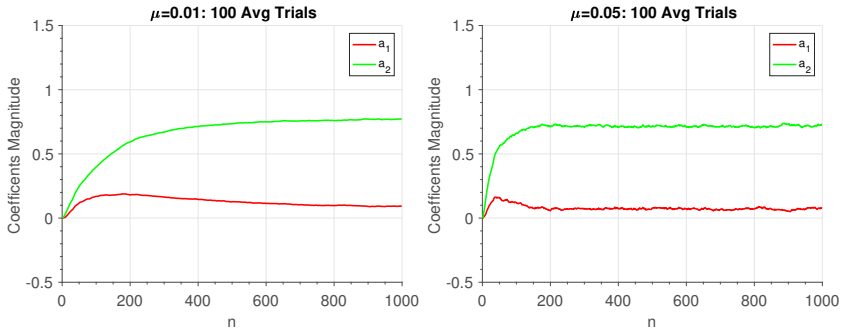

Figure 25: For $\mu = 0.01$ and $\mu = 0.05$, The corresponding Coefficient evolution for 100 Trials averaged

Nonetheless, it only after the 900 interations, the steady value converges when $\mu = 0.01$, and yet on the other hand the algorithm requires 200 iterations for convergence, as shown in figure 26.

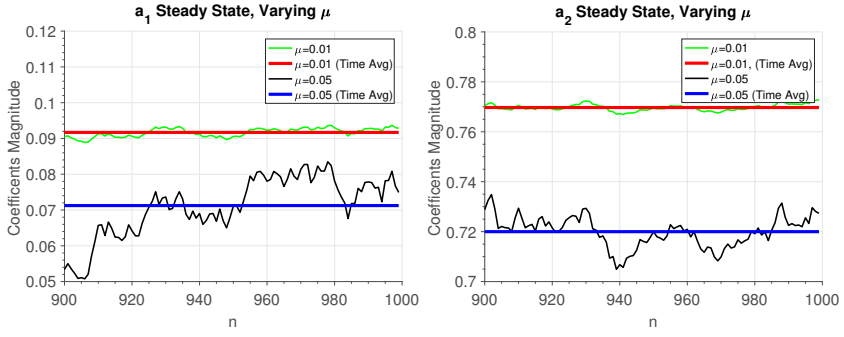

Figure 26: For $\mu = 0.01$ and $\mu = 0.05$, The corresponding Steady-State Values of Coefficients a_1 and a_2

2.1.5 Cost Function Minimisation

By differentiating the cost function, $J(n)\& = \frac{1}{2}\left(e^2(n)+\gamma\|\mathbf{w}(n)\|_2^2\right)$ with respect to $\mathbf{w}$ and thereby equating to zero, we able to minimise the cost function with respect to $\mathbf{w}$, as seen from the

mathematical derivations below:

$$\frac{\partial J}{\partial \mathbf{w}} = e(n)\frac{\partial e}{\partial \mathbf{w}} + \gamma \mathbf{w}(n) = 0$$

The error can be represented as $e(n) = \mathbf{x}(n) - \mathbf{w}^T(n)\mathbf{x}(n)$, thus after substituting followed by differentiation, results in:

$$\frac{\partial e}{\partial \mathbf{w}} = \frac{\partial}{\partial \mathbf{w}}\left(\mathbf{x}(n) - \mathbf{w}^T(n)\mathbf{x}(n)\right) = -\mathbf{x}(n)$$

$$\frac{\partial J}{\partial \mathbf{w}} = -e(n)\mathbf{x}(n) + \gamma \mathbf{w}(n)$$

$$\mathbf{w}(n) = \mathbf{w}(n) - \mu \nabla_w J(n) \tag{5}$$

To acquire the optimal value of w, the gradient decent method will need to be utilised to apply to revised equation above. After substituting the cost functions derivative, it results in:

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

Hence, it can be concluded that minimizing the cost function $J(n)\& = \frac{1}{2}\left(e^2(n) + \gamma \|\mathbf{w}(n)\|_2^2\right)$ is equal to the leaky LMS equation which was derived directly above.

2.1.6 Leaky LMS Algorithm

By implementing the leaky LMS algorithm, figure 27 displays how, for varying values of μ and γ, the coefficients evolve; its clear that the ideal values of a_1 and a_2 are not converged to by the steady-state values obtained. Actually, a larger steady-state bias is obtained when a great value of γ is used. Moreover, by utilising the cross correlation vector, $\mathbf{p}$ and the autocorrelation matrix, $\mathbf{R}$ to calculate the ideal weights using the Weiner-Hopf method, where $\mathbf{w}_{opt} = \mathbf{R}^{-1}\mathbf{p}$.

Thus, it can concluded that Weiner-Hopf solution can be obtained after a few iterations of the μ used with a few conditions of the LMS algorithm. Inhibiting the inversion of autocorrelation matrix is the main aim, however the Wiener-Hopf solution can be converged to by the leaky LMS algorithm. The cost function which is minimized by the cost function and converges to the solution $\lim_{k\to\infty} \mathbb{E}[\mathbf{w}_k] = (\mathbf{R} + \gamma \mathbf{I})^{-1}\mathbf{p}$, however, varies from the cost function minimized by the leaky LMS. Moreover, another reason for non-convergence to the Wiener-Hopf solution by the leaky LMS, is due the fact there is a bias produced by the $\gamma \mathbf{I}$. Mathematically, we can derive the convergence values of the leaky LMS algorithm by utilising the autocorrelation matrix $\mathbf{R}_x$ and $\mathbf{p} = [\gamma(1), \gamma(2)]^T$.

γ	Coefficient a_1	Coefficient a_2
0.1	0.1542	0.3919
0.5	0.1626	0.4992
0.9	0.1319	0.7076

The leaky LMS produces different values for different inputs of γ, as shown in the table above, which ratifies the theoretical outcomes. As discussed above, with respect to μ, identical properties are exhibited by the leaky LMS.

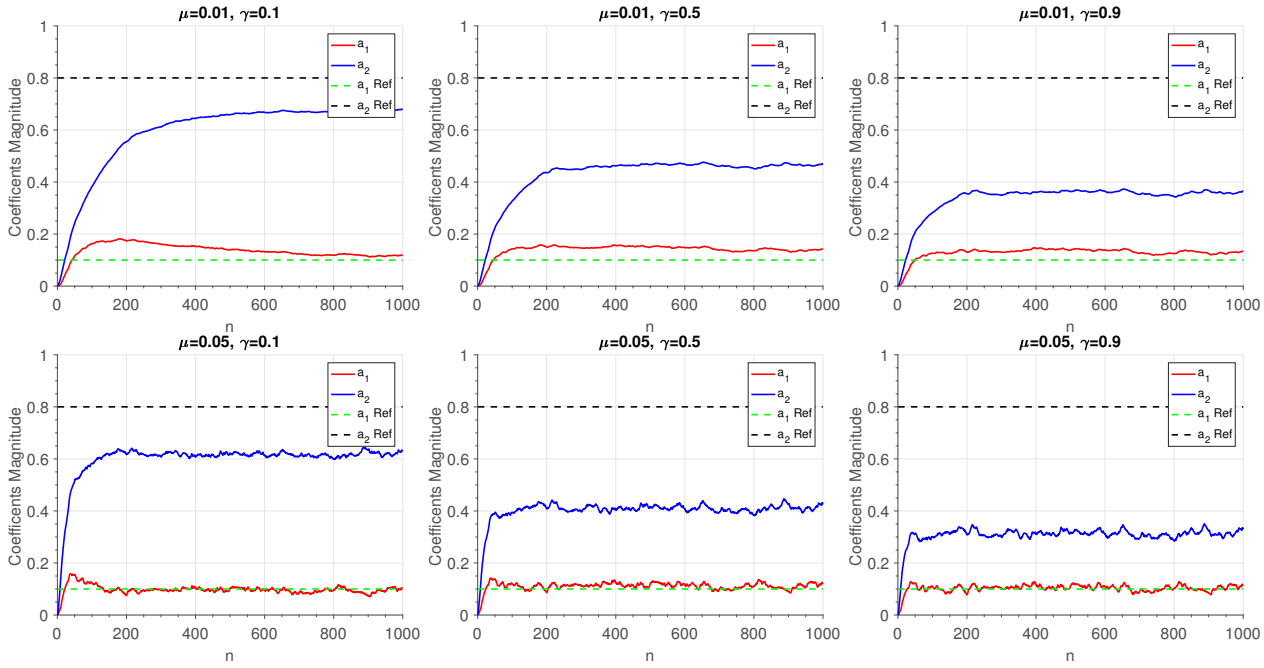

Figure 27: For $\mu = 0.01$ and $\mu = 0.05$ & Coefficients a_1 and a_2,the corresponding effects of increasing γ

2.2 Adaptive Step Sizes

2.2.1 Gradient Adaptive Step Size (GASS) Algorithm

In the previous section, we have seen the effects on the time of convergence and discrepancies produced by μ, when applying the LMS algorithm. Computing the LMS algorithm is inexpensive in memory, hence μ, which is non-varying, can be chosen for the application of constraints such as steady-state error, convergence time or overshooting. However, there are a few drawbacks of some values of μ (constant) such as divergence from a solution, rather than convergence when using the LMS algorithm.

There are methods which allow μ to vary with time, such as the Gradient Adaptive Step Size (GASS) algorithm. To provide convergence, the algorithms operate by the allowing the learning rates of the steady-state to approach zero. Correspondingly the error experienced, the step sizes can be adjusted to adapt μ appropriately. As shown below, all the step-sizes are adjusted corresponding to the equations which are imposed by the 3 GASS algorithms stipulated in this coursework as $\mu(n+1) = \mu(n) + \rho e(n)\mathbf{x}^T(n)\boldsymbol{\psi}(n)$. Using the equations below, $\boldsymbol{\psi}$ values (which are time-varying) can be adapted; ρ is a constant value used in the all algorithms below also.

$$\boldsymbol{\psi}(n) = [\mathbf{I} - \mu(n-1)\mathbf{x}(n-1)\mathbf{x}^T(n-1)]\boldsymbol{\psi}(n-1) + e(n-1)\boldsymbol{x}(n-1) \quad (Benveniste)$$
$$\boldsymbol{\psi}(n) = \alpha\boldsymbol{\psi}(n-1) + e(n-1)\boldsymbol{x}(n-1), \quad 0 < \alpha < 1 \quad (Ang\ \&\ Farhang)$$
$$\boldsymbol{\psi}(n) = e(n-1)\boldsymbol{x}(n-1) \quad (Matthew\ \&\ Xie)$$

The Benvensite Algorithm, which is able to handle gradient estimates with noise, has an update formula embedded for ψ, which has an instantaneous gradient equal to $e(n-1)x(n-1)$ and also functions as an adaptive low-pass filter. The Benveniste algorithms update equation can be transformed into a non-adaptive filter by the using the Ang & Farhang which has a trade-off between inferior convergence abilities for reduced complexity in computation. These two algorithms can be made much faster however at a trade-off expense for convergence abilities, by using the Matthew & Xie algorithm, which does not use low-pass filtering on the instantaneous gradient compared to the other two algorithms. Figure 28 displays, for a single trial, their weight error graph for the different algorithms; it is evident from the GASS algorithm expressed that compared to LMS algorithm which inhibits the adjustments of μ, the GASS algorithms performs with much more superiority.

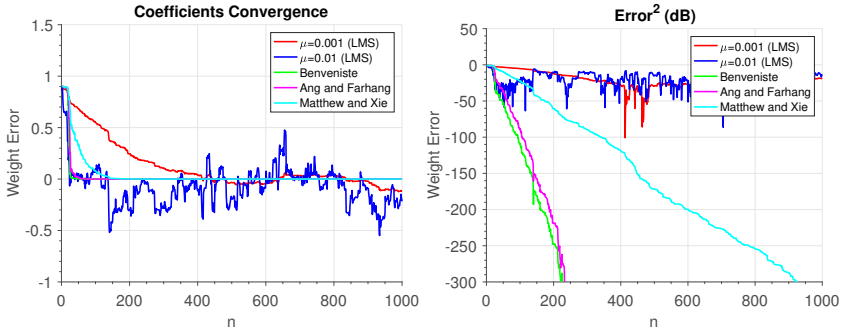

Figure 28: Adaptive Step Sizes and Standard LMS Algorithms Convergence Time Comparison

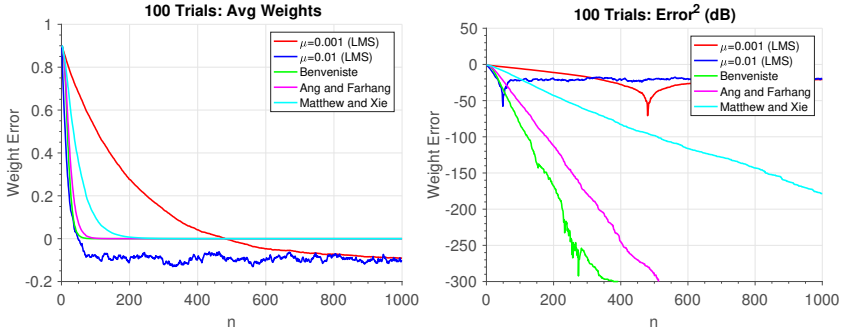

Figure 29: Analysis of the Averaging Weights over 100 Random Realizations and their Convergence Time

From figure 29 shown below, which displays the outcomes of a hundred trials averaged, clearly presents the non-convergence to the true weights when using a constant μ for the LMS algorithm. While the GASS algorithms results were predictable, the quickest to converge yet the costliest computationally was the Benveniste algorithm, but still better than Ang & Farhang algorithm. For LMS algorithms without step-size variability, the Matthew & Xie algorithm is superior however is the worst performance wise from algorithms. No offsets are presents and true weight convergence is consistent for all the GASS algorithms

2.2.2 Normalised LMS (NLMS)

The update equation given by, $\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\beta}{\epsilon+\|\mathbf{x}\|^2} e(n)\mathbf{x}(n)$, is ascribed to the normalised LMS algorithm (NLMS). In order to prove that this update equation in based upon the a

posteriori error $e_p(n) = d(n) - \mathbf{x}^T(n)\mathbf{w}(n+1)$, and thus equal to the NLMS algorithm, where thereafter, $\mathbf{x}^T(n)$ is multplied on both sides of $\mathbf{w}(n+1)\& = \mathbf{w}(n) + \mu e_p(n)\mathbf{x}(n)$ and then added with $\mathbf{d}(n)$ on both sides to give:

$$\mathbf{d}(n) - \mathbf{x}^T(n)\mathbf{w}(n) = \mathbf{d}(n) - \mathbf{x}^T(n)\mathbf{w}(n+1) + \mathbf{x}^T(n)\mu e_p(n)\mathbf{x}(n) \tag{6}$$

The *a posteriori* error, $e_p(n)$, and the error, $e(n)$, can then be simplified as the following:

$$e(n) = e_p(n) + \mathbf{x}^T(n)\mu e_p(n)x(n) = e_p(n)\left(1 + \mu\|\mathbf{x}\|^2\right)$$

$$e_p(n) = \frac{e(n)}{1 + \mu\|\mathbf{x}\|^2} \tag{7}$$

This can then be substituted into the original update equation and to produce the NLMS algorithm below in which can then be equaled to the update equation due its $\beta = 1$ and *a posteriori* error with $\epsilon = \frac{1}{\mu}$. Regardless of how small $\|\mathbf{x}\|^2$ is, the algorithm still maintains stability due to inversely proportionality relationship between ϵ and the step size μ.

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

2.2.3 Generalized Normalized Gradient Descent (GNGD)

From figure 30, compared to the Benveniste algorithm the quicker convergence observed from Generalized Normalized Gradient Descent algorithm (GNGD). When $\mu_{\text{Benveniste}} = 1$ it results in below figure, however previous test were applied to find the Benveniste algorithms optimal μ value. Yet, the GNGDs performance is immune to changes in initial values of μ, and performs exceptionally well when a large range μ values are used; for $\mu_{\text{GNGD}} = 10$. the figure is displayed below.

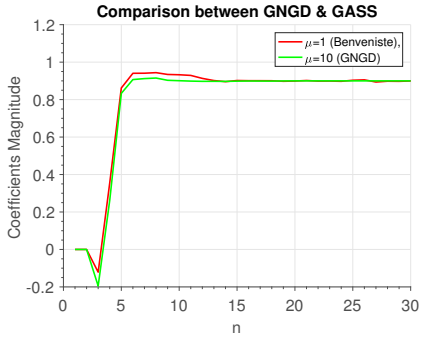

Figure 30: GNGD and GASS Algorithm's Convergence Speed Comparisons

Even though we have shown earlier, the greatness in terms of prediction error and speed, of the Benvenist GASS algorithm compared to the standard LMS and GASS algorithms, a cost of greater computational complexity is associated with the improved performance. The number of lags in AR or MA model processes can be denoted as M, which can also be consider the model order thus we have $\mathbf{x}(n) \in^M$ as the input vector. For each iteration of the update for the Benvenist GASS algorithm, the following calculations are required:

- $\mathbf{x}(n-1)\mathbf{x}(n-1)^T$ (Outer product)

- the $\Big[I - \mu(n-1)\mathbf{x}(n-1)\mathbf{x}(n-1)^T\Big]\psi(n-1)$ (Matrix Product)

Thus, $\mathcal{O}(\mathbf{M^2})$ is the complexity for the Benvenist GASS algorithm, as it possesses a quadratic complexity in M for both. However, the GNGD algorithm is bounded by the linear complexity in M, which is due to the fact that each update iteration only requires scalar operations, inner product calculations and additions of M-dimensional vectors hence the GNGD algorithm is only $\mathcal{O}(\mathbf{M})$. Remarkably, the GNGD algorithm computationally less expensive than the Benvenist GASS algorithm but also outperforms it in terms of reduced prediction error and greater convergence speed.

2.3 Adaptive Noise Cancellation

2.3.1 Adaptive Line Enhancement

We can configure the Adaptive Line Enhancement (ALE) to allow for non-correlation between $\mathbf{u}(n)$, which is the predicted input, $s(n)$, which is the signal and $\eta(n)$, which is the noise; all can be configured by selecting certain values of Δ. A signal which is uncorrupted by noise can produced by inhibting the influence of noise by ensuring that the predicted input $\mathbf{u}(n)$ and the noise $\eta(n)$ are uncorrelated. By working from the mean-squared error, The Δ optimal value can attained from $\mathbb{E}\left\{\left(s(n)-\hat{x}(n)\right)^2\right\} = \mathbb{E}\left\{\left(x(n)+\eta(n)-\hat{x}(n)\right)^2\right\}$ which means that it is also equal to $\mathbb{E}\{\eta^2(n)\} + \mathbb{E}\left\{\left(x(n)-\hat{x}(n)\right)^2\right\} + 2\mathbb{E}\left\{\eta(n)\Big(x(n)-\hat{x}(n)\Big)\right\}$

There are 3 main terms which cause the mean-squared error (as seen from after expanding). In the ideal case, having the noise power equivalent to the mean squared error is desired. Yet, only the adjustment of $2\mathbb{E}\left\{\eta(n)\Big(x(n)-\hat{x}(n)\Big)\right\}$ through δ values being changed, hence to minimize $2\mathbb{E}\left\{\eta(n)\Big(x(n)-\hat{x}(n)\Big)\right\}$, the corresponding value of δ will need to be selected. Moreover, the equation can be reduce to $\mathbb{E}\{\eta(n)\hat{x}(n)\}$ since $x(n)$ and $\eta(n)$ are uncorrelated, therefore $\eta(n) = v(n) + 0.5v(n-2)$:

$$\begin{aligned}
\mathbb{E}\{\eta(n)\hat{x}(n)\} &= \mathbb{E}\left\{\Big(v(n)+0.5v(n-2)\Big)\mathbf{w}^T\mathbf{u}(n)\right\} = \mathbb{E}\left\{\Big(v(n)+0.5v(n-2)\Big)\left(\sum_{i=0}^{M} w_i s(n-\Delta-i)\right)\right\} \\
&= \mathbb{E}\left\{\Big(v(n)+0.5v(n-2)\Big)\left(\sum_{i=0}^{M} w_i\Big(x(n-\Delta-i)+\eta(n-\Delta-i)\Big)\right)\right\} \\
&= \mathbb{E}\left\{\Big(v(n)+0.5v(n-2)\Big)\left(\sum_{i=0}^{M} w_i\eta(n-\Delta-i)\right)\right\} \\
&= \mathbb{E}\left\{\Big(v(n)+0.5v(n-2)\Big)\left(\sum_{i=0}^{M} w_i\Big(v(n-\Delta-i)+0.5v(n-\Delta-i-2)\Big)\right)\right\}
\end{aligned}$$

As seen from above figure 31, due to the uncorrelated property, the expectation operator can split into a sum of expectations, for various time instances of white noise only when the value of Δ greater than 2. 100 realizations are presented on the lower rows whereas 100

realizations averaged are presented on the top rows along with a reference ideal sinewave. These findings are substantiated due to the fact that when Δ increases from 2 to 3, the mean-squared error decreases hugely.

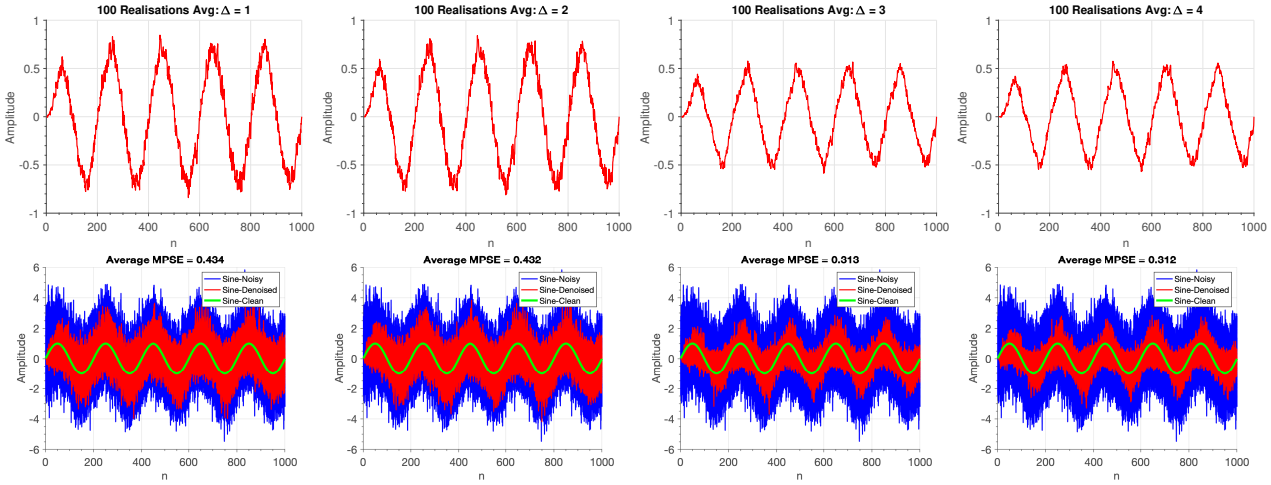

Figure 31: Adaptive Line Enhancement Algorithm - Ideal Δ to be utilised

2.3.2 Effect of Increasing Delays

The above findings are further verified from the figure 32 below. For each of the model orders, there is reduction in the mean-squared error due to having a Δ value larger than 2 however increasing the Δ value haphazardly is not recommended for the following reasons. $\mathbb{E}\left\{\left(x(n)-\hat{x}(n)\right)^2\right\}$ is a term which supplies $\mathbb{E}\left\{\left(s(n)-\hat{x}(n)\right)^2\right\}$, which the is mean squared error which can increase when there is a larger difference between $x(n)$ and $\hat{x}(n)$ caused by larger values of Δ. When Δ is equal to 25, compared to the ideal sinewave input, the sinewave which is produced in the output is shifted which causes a larger difference between $\mathbb{E}\left\{\left(x(n)-\hat{x}(n)\right)^2\right\}$, which consequently also increases the mean-square error.

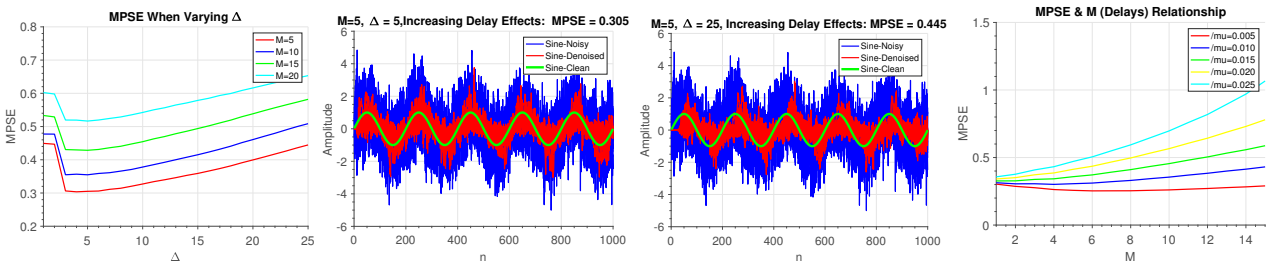

Figure 32: Affects on MPSE when increasing Delay

Ideally, a greatly periodic process can be described with huge accuracy when the model order is increased, however due to influential features such as computational complexity, stability and convergence this is not always the case. The mean-squared error can be enlarged, when increasing the model order whilst allowing μ to be constant which is evident when $M > 6$. However, it is possible, to have large model order with small μ value but will cause a slow

convergence. Thus, the most reasonable values which should be chosen are $M = 6$ and $\mu = 0.005$, from the analysis of figure 32 (rightmost graph).

2.3.3 Comparison of ALE and ANC

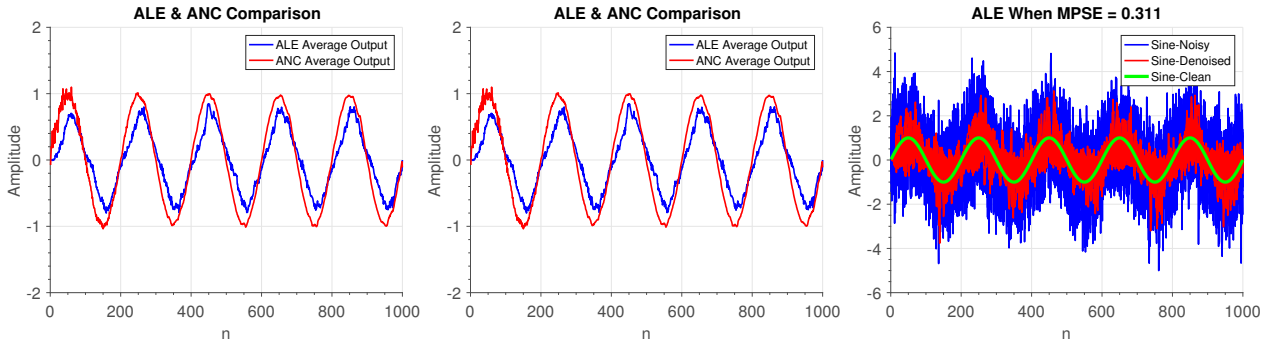

Figure 33: Denoising Sinewave - ALE and ANC Comparison

Above is the figure which displays the performance comparison between the Adaptive Noise Cancellation (ANC) configuration and the ALE configuration. Large errors are present for the ANC configuration when smaller values of n are used. However, it is overwhelmingly clear that, after convergence, ANC configuration is superior performance wise.

2.3.4 EEG Data

A reference spectrogram is shown in figure 34 which displays the initial EEG values which has a large component value at 50Hz. A Hamming window is used to window each segment which is of length 4096 along with each neighbouring segments being overlapped by 80%.

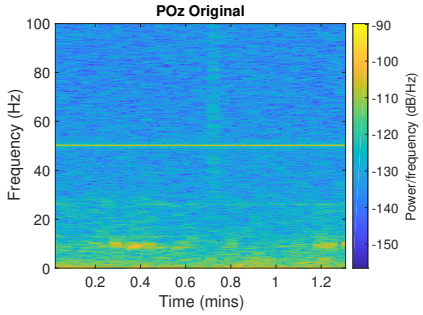

Figure 34: POz Location on the Scalp - Original EEG Data collected with Strong Component at 50 Hz

Firstly, eliminating the large component at 50Hz will require a specific value of M. Thus, when the model order is increased the effects are shown below in figure 35 and figure 36. The noise is removed more efficiently when the model order is increased because a filter effect is experienced, nevertheless some unwanted artifacts also become apparent. The large 50Hz component is removed by the filter along with reducing the components which range from 40 Hz to 60 Hz, only when $M = 25$ and $\mu = 0.01$

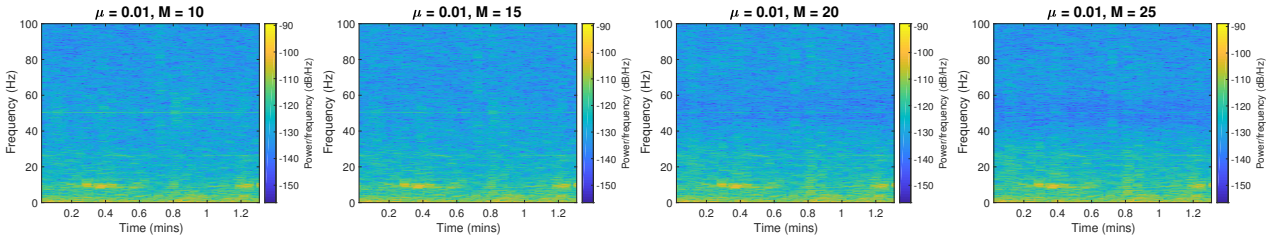

Figure 35: Spectrogram of EEG Data when increasing model order

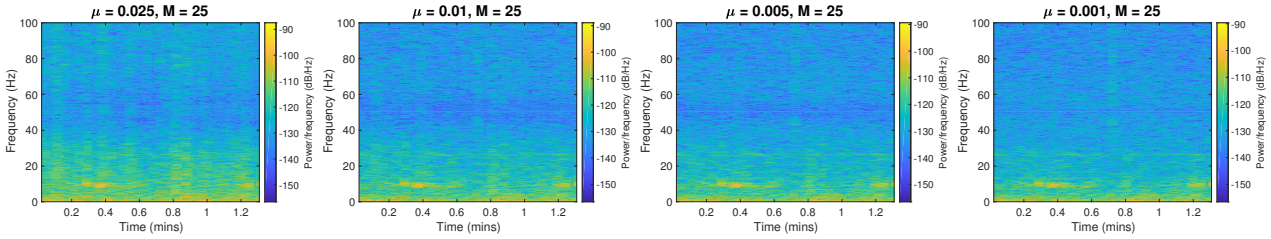

Figure 36: Spectrogram of Denoised EEG Data when varying μ on artifacts

By varying the value of μ, we can limit the range of the frequencies of the filter. As shown above, when μ is decreased, it produces improved performance results for frequencies where the ANC algorithms operates. It is very evident that the filter is now operating in a targeted and limited range of frequencies.

Figure 37 shows that, for the squared error plots and the periodogram also verify the result above. At low frequencies, size-able errors are produced when μ is large for instance when μ is equal to 0.025, however this phenomena is not produced when μ is equal to 0.001.

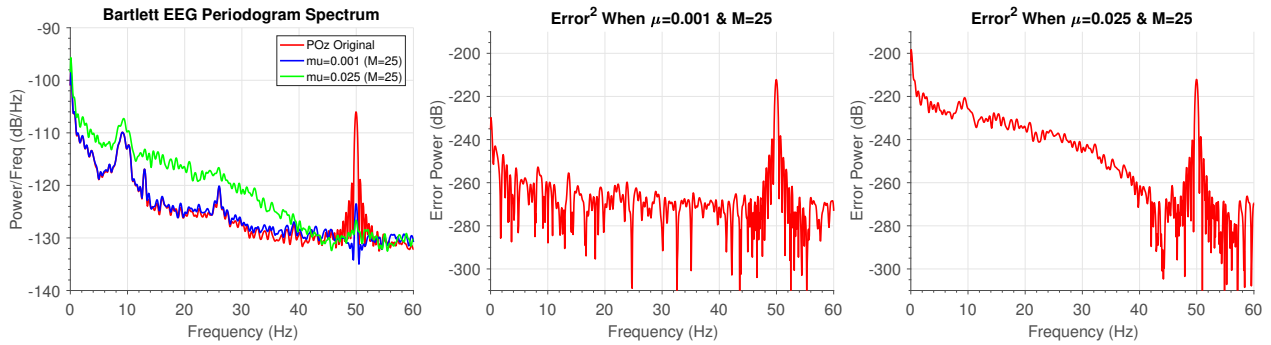

Figure 37: Bartlett Periodogram (2 second intervals averaged) and Squared Errors

3 Widely Linear Filtering and Adaptive Spectrum Estimation

3.1 Complex LMS and Widely Linear Modelling

3.1.1 Learning Curves

A Widely Linear Moving Average (WLMA(1)), of order 1, can be identified and concluded that it is non-circular using the Augmented Complex LMS (ACLMS) and the Complex LMS (CLMS). The complete non-circular process cannot be described due to the limited degrees of freedom available for the CLMS algorithm to utilise, as shown in the figure 38 which displays the learning curve. However, by using extra weights $\mathbf{g}(n)$, the process can described completely due to the extra degrees of freedom available to the ACLMS algorithms. By applying both $x(n)$ and $x^*(n)$, the second order data statistics can be manipulated using ACLMS the algorithm.

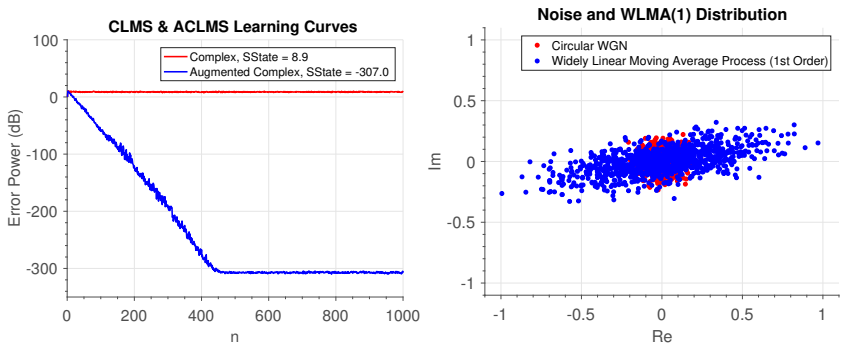

Figure 38: CLMS and ACLMS Learning Curves Comparison For a Non-Circular WLMA(1) Process

3.1.2 Circularity of Wind Speeds

If the probability distribution is constant in terms of rotation (independent of angle), then it can be said that the complex-valued random variable is circular, as seen from the 3 wind regime scatter plot shown in figure 39. To summarise, only the Euclidean distance measured from the complex domains origin should be the only dependent factor for the probability function distribution of a variable. **The less circular data is produced when the circularity quotient value is higher as seen from figure 39 below, which displays for each regime, the respective circularity coefficient.**

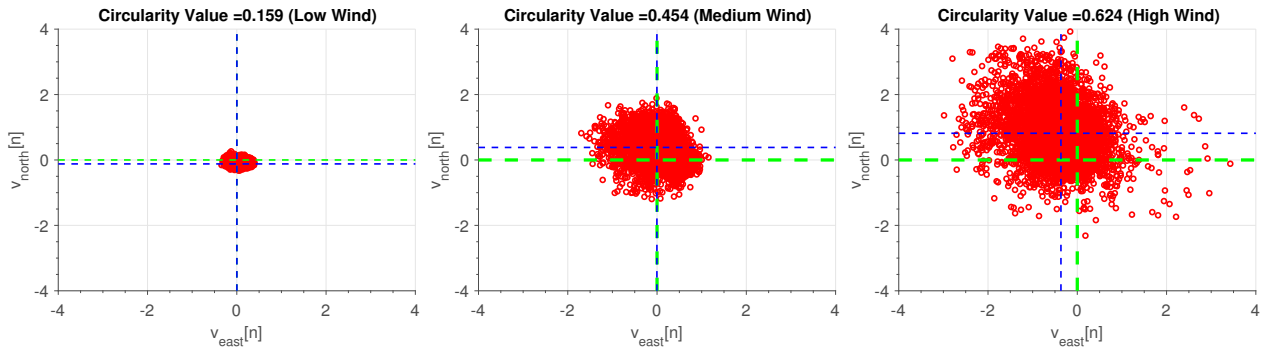

Figure 39: Varying Wind Speeds and their associated circularity plots

It can mistakenly be assumed that the least circular is the low wind regime, however when zoomed in, it is evident that the a huge mean offset is present for the high wind regime. A reference to the origin and the averaged values across both axes are denoted by the green and blue dotted lines respectively. The low wind regime can be approximated to be circular mathematically, due it being centred near the origin even though the plot looks more elliptical rather than circular. However, the high wind regime is not rotationally invariant due the scatter plot being off centred from the origin. Figure 40 below, graphs the CLMS and the ACLMS algorithms applied on both of the wind regimes:

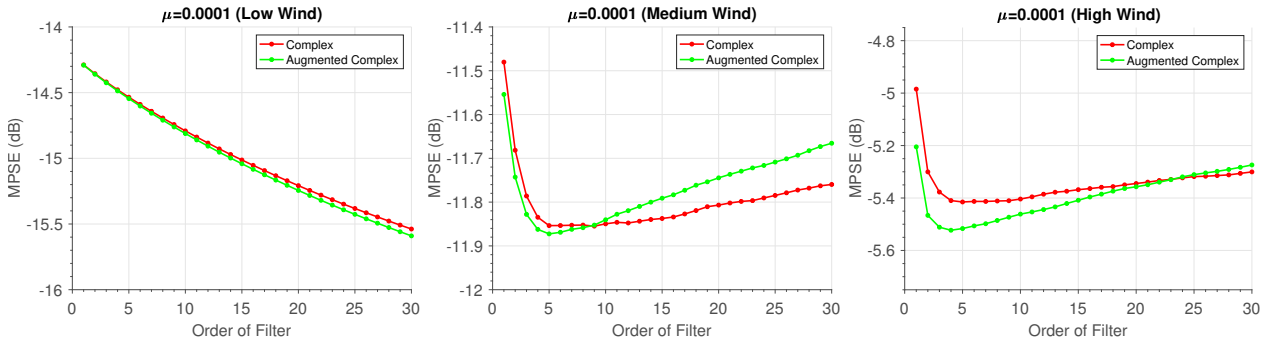

Figure 40: Positive Correlation between Filter order and MPSE

When the CLMS algorithm is applied, the one-step prediction is performed effectively for both medium and high wind regimes; which was predicted for smaller model orders for each respective regimes circularity coefficients, however data which is non-circular is not able to be apprehended by the algorithm. Using both ACLMS and CLMS algorithms, the performance from low wind regime is exceptional for both algorithms respectively.

Due to overfitting, both algorithms performances start deteriorating substantially when higher model orders are used. In comparison to the CLMS algorithm, ACLMS algorithm has many more parameters to satisfy hence its is much better at fitting the training data. Nevertheless, this can lead to overfitting which can cause the filter to adequately predict future values or classify data well. This can be observed in the CLMS algorithm, which has fewer parameters which need to be satisfied, but suffers from the data being overfitted because as the model order gets larger the prediction error begins to increase yet the overfitting rate is not as fast as the CLMS. Learning algorithms can be described statistically using the **Vapnik-Chervonenkis Theory**, hence can be applied to the simple logical deduction above, based on degrees of freedom available or the amount of free parameters available which are causing the overfitting.

3.1.3 Complex Voltages

By utilising the Clarke Transform, a compact expression of the 3-phase power system, which involve complex voltages, can be produced as $v(n) = A(n)e^{j\left(2\pi\frac{f_0}{f_s}n+\phi\right)} + B(n)e^{-j\left(2\pi\frac{f_0}{f_s}n+\phi\right)}$ where $A(n)$ and $B(n)$ are $A(n) = \frac{\sqrt{6}}{6}\left(V_a(n)+V_b(n)e^{j\Delta_b}+V_c(n)e^{j\Delta_c}\right)$ and $B(n) = \frac{\sqrt{6}}{6}\left(V_a(n)+V_b(n)e^{-j(\Delta_b+\frac{2\pi}{3})}+ V_c(n)e^{-j(\Delta_c-\frac{2\pi}{3})}\right)$ respectively.

$B(n)$ must equated to zero, in order for the system to balanced along with all the voltages having equal magnitudes. We also need $\Delta_b = \Delta_c = 0$, to prevent the 3 signals from deviating from their respective initial values from their relative phase difference, hence we have 2 methods for a system to be unbalanced. A balance signal is graphed below in figure 41, where each component have been adjusted relative magnitudes and varied phase difference between components.

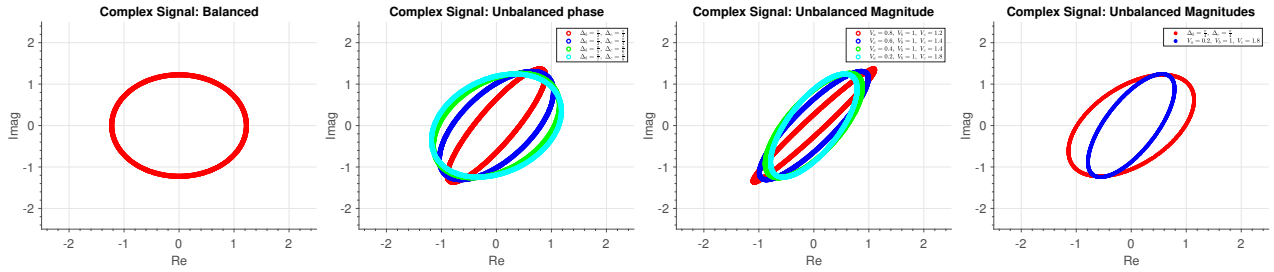

Figure 41: Complex Voltages - Balanced and Unbalanced

Due to circular plots and non-circular plots corresponding to balance and unbalanced system respectively, it is therefore obvious to determine a balance or unbalanced system visually. Furthermore, the plots below are also further effected from adjusting phase differences and magnitude differences.

3.1.4 Autoregressive Models for Complex Voltages

The form $v(n+1) = h^*(n)v(n)$ is a strictly autoregressive model, of first order. For a **balanced system**, the voltage can expressed as $v(n) = \sqrt{\frac{3}{2}} V e^{j\left(2\pi \frac{f_0}{f_s} n + \phi\right)}$. Therefore, the strictly autoregressive model can be rearranged to give:

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

The LHS can be simplified by substituting $h^*(n)$ for $h(n)$:

$$e^{j2\pi \frac{f_0}{f_s}} = h^*(n) = \Re\mathfrak{e}\{h(n)\} - \mathfrak{Im}\{h(n)\} = |h(n)| e^{-j\left(\arctan\left(\frac{\Re\mathfrak{e}\{h(n)\}}{\mathfrak{Im}\{h(n)\}}\right)\right)}$$

By equating the LHS and RHS phases only, the negative sign can brought inside the arctan by taking the reciprocal of the fraction, then rearranging to achieve the proof.

$$2\pi \frac{f_0}{f_s} = -\arctan\left(\frac{\Re\mathfrak{e}\{h(n)\}}{\mathfrak{Im}\{h(n)\}}\right) \therefore$$

$$f_0 = \frac{f_s}{2\pi} \arctan\left(\frac{\mathfrak{Im}\{h(n)\}}{\Re\mathfrak{e}\{h(n)\}}\right)$$

For an **unbalanced system**, the widely linear first order autoregressive model is given in the form of $v(n+1) = h^*(n)v(n) + g^*(n)v^*(n)$ hence we can therefore express the voltage of a unbalanced system using the Clarke Transforms: $v(n) = A(n)e^{j\left(2\pi \frac{f_0}{f_s} n + \phi\right)} + B(n)e^{-j\left(2\pi \frac{f_0}{f_s} n + \phi\right)}$. Thus, we will to need use the Clarke Transform and the associated conjugate to produce:

$$v(n+1) = h^*(n)A(n)e^{j\left(2\pi\frac{f_0}{f_s}n+\phi\right)} + h^*(n)B(n)e^{-j\left(2\pi\frac{f_0}{f_s}n+\phi\right)} + g^*(n)A^*(n)e^{-j\left(2\pi\frac{f_0}{f_s}n+\phi\right)} + g^*(n)B^*(n)e^{j\left(2\pi\frac{f_0}{f_s}n+\phi\right)}$$

There after, by adapting the index's of the Clarke Transform and then substituting $v(n+1)$, we can then simplify and group exponential terms to produce:

$$v(n+1) = A(n+1)e^{j\left(2\pi\frac{f_0}{f_s}(n+1)+\phi\right)} + B(n+1)e^{-j\left(2\pi\frac{f_0}{f_s}(n+1)+\phi\right)} \therefore$$

$$A(n+1)e^{j\left(2\pi\frac{f_0}{f_s}(n+1)+\phi\right)} = \left(h^*(n)A(n) + g^*(n)B^*(n)\right)e^{j\left(2\pi\frac{f_0}{f_s}n+\phi\right)}$$

$$B(n+1)e^{-j\left(2\pi\frac{f_0}{f_s}(n+1)+\phi\right)} = \left(h^*(n)B(n) + g^*(n)A^*(n)\right)e^{-j\left(2\pi\frac{f_0}{f_s}n+\phi\right)}$$

By assuming that $A(n+1) \approx A(n)$ and $B(n+1) \approx B(n)$, the above equation can be simplified as:

$$e^{j2\pi\frac{f_0}{f_s}} = \frac{h^*(n)A(n) + g^*(n)B^*(n)}{A(n+1)} \approx h^*(n) + g^*(n)\frac{B^*(n)}{A(n)} \tag{8}$$

$$e^{-j2\pi\frac{f_0}{f_s}} = \frac{h^*(n)B(n) + g^*(n)A^*(n)}{B(n+1)} \approx h^*(n) + g^*(n)\frac{A^*(n)}{B(n)} \tag{9}$$

When taking the conjugate of (9) and equating it to (8), it will produce:

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

By allowing for a change of variable, where $Y = \frac{B^*(n)}{A(n)}$, a quadratic equation in Y is formed: $g^*(n)Y^2 + \left(h^*(n) - h(n)\right)Y + g(n) = 0$, which can be solved for Y, to get:

$$\begin{aligned} Y &= \frac{-\left(h^*(n) - h(n)\right) \pm \sqrt{\left(h^*(n) - h(n)\right)^2 - 4g^*(n)g(n)}}{2g^*(n)} \\ &= \frac{\mathrm{Im}\{h(n)\}j \pm j\sqrt{\mathrm{Im}^2\{h(n)\} - |g(n)|^2}}{g^*(n)} \end{aligned}$$

The solution of Y into can be substituted in (8), we produce:

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

A solution, (where M is a constant), can be eliminated due $f_s > f_0 > 0$ to produce:

$$e^{j2\pi\frac{f_0}{f_s}} = \mathrm{Re}\{h(n)\} + j\sqrt{\mathrm{Im}^2\{h(n)\} - |g(n)|^2} = |M|e^{j\left(\arctan\left(\frac{\sqrt{\mathrm{Im}^2\{h(n)\} - |g(n)|^2}}{\mathrm{Re}\{h(n)\}}\right)\right)}$$

By equating the LHS and RHS phases only, followed by rearranging, proof is completed

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

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

3.1.5 Frequency Estimates

To equal the UK power transmission operating frequency, the frequency f_0 was made to equal 50 Hz. As seen below, a balanced system has been operated on and tested by two algorithms, the CLMS and ACLMS. 50Hz, which is the nominal frequency, is the frequency which both algorithms converge to, however the CLMS converges at a much faster rate compared to ACLMS. Throughout the first two iterations, compared to the CLMS, the ACLMS exhibits a large overshoot.

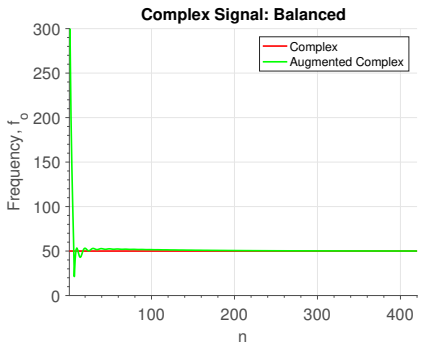

Figure 42: Balanced Complex Voltages CLMS and ACLMS Comparison

For an unbalanced system, each algorithm was applied as shown in figure 43 and again the fastest to converge was the CLMS compared the ACLMS algorithm which also exhibited a huge overshoot. The big difference however, is that a bias is not used for the convergence to the 50Hz nominal frequency by the ACLMS algorithm, but this frequency is not converged to by the CLMS, which is due to the limited ability of the CLMS algorithm only being a able to plot circular data and, in the case when the system is unbalanced, a non-circular plot is produced. Thus, it could be said that characterising an ellipse from a circle is the equivalent as for describing an unbalanced system when utilising the CLMS algorithm.

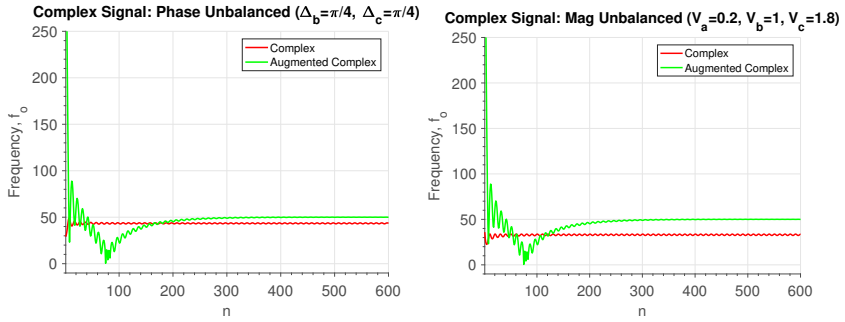

Figure 43: Unbalanced Complex Voltages CLMS and ACLMS Comparison

3.2 Adaptive AR Model Based Time-Frequency Estimation

3.2.1 FM Signal

A non-stationary frequency modulated (FM) signal is shown below in figure 44, where the frequency was produced from circular white Gaussian noise, which has the mean and variance equal to 0 and 0.05 respectively. An AR process is also modelled from the signal in the frequency domain, as shown in the same figure, however it is not recommended to model

the full signal into an AR process. For the frequency modulated signal, the input signal is non-stationary, however in the function used in MATLAB, `aryule`, assumes that a stationary input signal is used. The three unique stages of the signals frequency is to be increasing in linear or quadratic or constant manner.

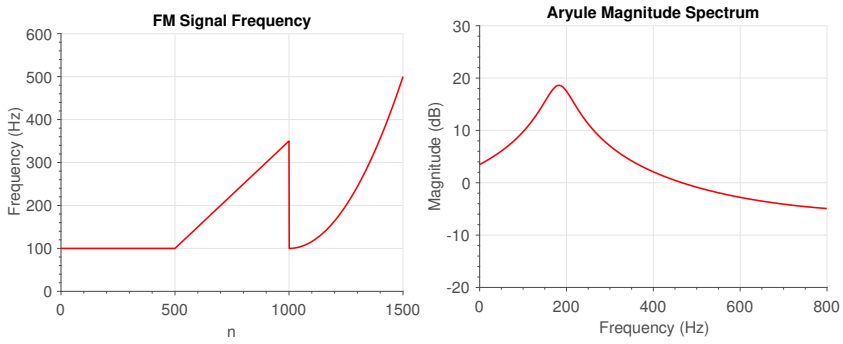

Figure 44: FM Signal Frequency and Power Spectrum from Block Based Estimate of AR Coefficients

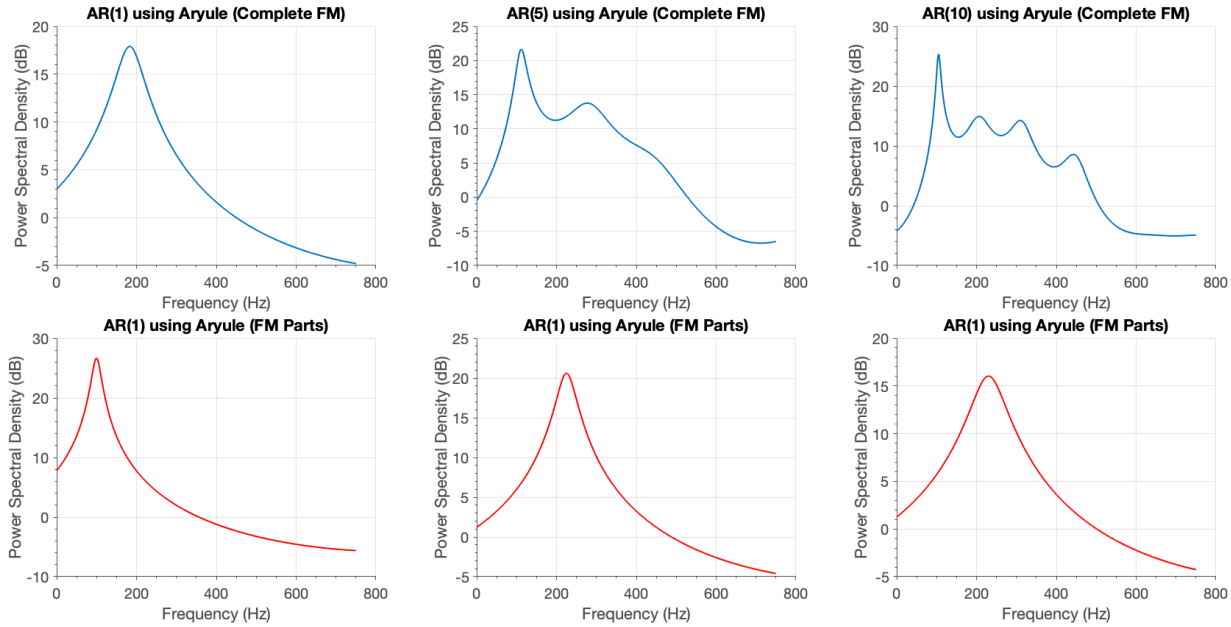

Figure 45: Aryule Method: Power spectrum of FM segments

Nevertheless, by utilising block sizes which are smaller, we are still able to carry out autoregressive modelling based on these blocks. The AR(1) process can modelled from the first block, which also possesses a constant frequency; in total we would required three blocks, each carrying 500 samples, and inputted in the `aryule` function for the modelling to occur. Due to the question requesting for AR(1) process to be modelled from the frequency modulated signal, the block type AR process is still unable to seize the non-stationarity characteristics with the two other blocks. Using various model orders, figure 45, displays the power spectrum of block-based AR estimation with 500 samples for with in each block. The number of poles in a frequency spectrum can be increased, by simply increasing the model order, however due to the linear and quadratic frequency regions increasing without displaying stationarity, a poor model estimate is obtained.

3.2.2 Frequency Evolution

As seen from above, an AR(1) process can modelled from frequency regions which are constant in frequency modulated signal which present in small block sizes, for instance, the 100Hz frequency which is constant. It can be observed that there is very little improvements when the model order is increased. Regardless, of the model order, the frequency modulated signals non-stationary regions are poorly modelled.

We able to determine the non-stationarity parts of the data by deducing their weights by utilising the CLMS algorithm thus instead of processing the whole dataset in one go, we have deduced the AR coefficients values in an adaptive manner as show in the figure 46, which is very clear in presenting the effect produced when using a range of values of μ (rate of convergence and other features impact when changing the μ value were discuss in thoroughly in previous sections). Real world scenarios such as signal frequency tracking is possible through fast convergence hence the figure below validates the requirement of fast convergence. When there is a big uncertainty in the precision of a signals frequency, a thick line is used to indicate this, which are due to large values of μ being used. This emphasizes the compromise between error variation from the steady state and the speed of convergence.

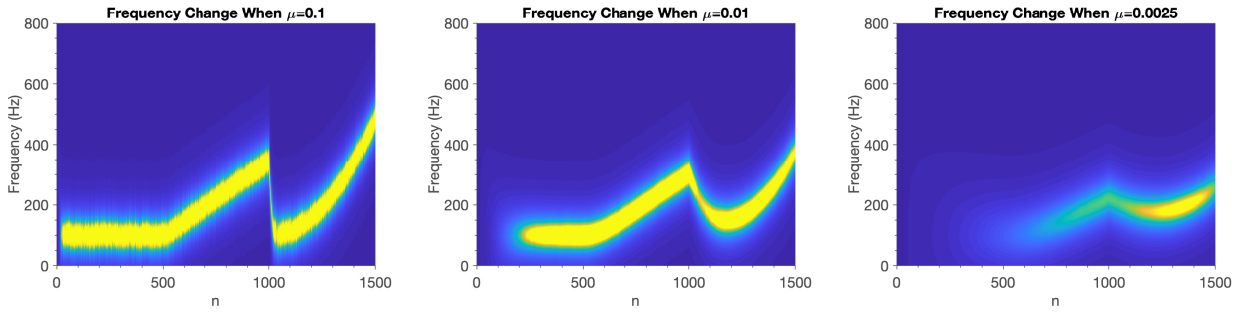

Figure 46: Time-Frequency Estimation using Complex LMS Algorithm with Different Values of μ

3.3 A Real Time Spectrum Analyser Using Least Mean Square

3.3.1 Least Squares Solution

The Least squares problem which is present in many real world scenarios, can be solved by utilising the cost function $\min_{\mathbf{w}} \|\mathbf{y}-\hat{\mathbf{y}}\|^2 = \min_{\mathbf{w}} \sum_{n=0}^{N-1} |y(n)-\hat{y}(n)|^2$, where $\|\mathbf{y}-\hat{\mathbf{y}}\|^2$ can be expressed as:

$$\begin{aligned}\|\mathbf{y}-\hat{\mathbf{y}}\|^2 &= (\mathbf{y}-\hat{\mathbf{y}})^H(\mathbf{y}-\hat{\mathbf{y}}) = (\mathbf{y}-\mathbf{F}\mathbf{w})^H(\mathbf{y}-\mathbf{F}\mathbf{w}) \\ &= \mathbf{y}^H\mathbf{y} - \mathbf{w}^H\mathbf{F}^H\mathbf{y} - \mathbf{y}^H\mathbf{F}\mathbf{w} + \mathbf{w}^w\Big(\mathbf{F}^H\mathbf{F}\Big)\mathbf{w}\end{aligned}$$

In order to minimize, the above equation can be differentiated with respect to $\mathbf{w}$, then setting it equal to 0, whilst removing the constant multipliers and then performing a transpose on both sides. Thereafter, by simply inverting the matrix $\mathbf{F}^H\mathbf{F}$ accomplishes the proof:

$$\begin{aligned}-\mathbf{y}^H\mathbf{F} - \mathbf{y}^H\mathbf{F} + \mathbf{w}\Big(\mathbf{F}^H\mathbf{F} + \mathbf{F}^H\mathbf{F}\Big) &= 0 \\ 2\mathbf{w}^H\mathbf{F}^H\mathbf{F} &= 2\mathbf{y}^H\mathbf{F}^H\end{aligned}$$

$$\mathbf{F}^H\mathbf{F}\mathbf{w} = \mathbf{F}\mathbf{y} \therefore$$
$$\mathbf{w} = \left(\mathbf{F}^H\mathbf{F}\right)^{-1}\mathbf{F}\mathbf{y}$$

For a signal $\hat{x}(n)$,the Inverse Discrete Fourier Transform (DFT), is given by the equation: $\hat{x}(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(K)e^{j2\pi nk/N}$ which can also be expressed in a matrix vector notation, where $W_{n,k} = e^{j2\pi nk/N}$. This can simply also be expressed as $\hat{\mathbf{x}} = \mathbf{WX}$

$$\begin{bmatrix} \hat{x}(0) \\ \hat{x}(1) \\ \vdots \\ \hat{x}(N-1) \end{bmatrix} = \begin{bmatrix} W_{0,0} & W_{0,1} & \cdots & W_{0,N-1} \\ W_{1,0} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ W_{N-1,0} & \cdots & \cdots & W_{N-1,N-1} \end{bmatrix} \begin{bmatrix} X(0) \\ X(1) \\ \vdots \\ X(N-1) \end{bmatrix}$$

The Fourier coefficients **X**, from the Inverse DFT in a matrix vector form, can be proven using the fact that $\mathbf{w} = \left(\mathbf{F}^H\mathbf{F}\right)^{-1}\mathbf{F}\mathbf{y}$

$$\mathbf{X} = \left(\mathbf{W}^H\mathbf{W}\right)^{-1}\mathbf{W}\mathbf{x} \tag{10}$$

The squared error between **x** and $\hat{\mathbf{x}}$ can be reduced to a minimal by using the Fourier coefficients which are produced, **X**, the final equation enables us to produce a signal $\hat{\mathbf{x}}$

3.3.2 Fourier Transform

The signal **x** approximate can be formed by using the Inverse DFT in matrix vector form to produce $\hat{\mathbf{x}} = \mathbf{WX}$. By projecting **x** on the column space of **W**, the approximation **X** can be generated. An orthonormal basis containing complex exponentials are produced from the columns of **W**. The DFT requires a limited set vectors which form the basis, more specially the N for a signal $x(n)$ which possess components which are N non-zero, compared the CTFT (Continuous-Time Fourier Transform). It is implicitly assumed that N is the period of the signal $x(n)$. Because the basis of $x(n)$ is projected on an basis which is orthonormal, the Parsevals theorem can be applied.

3.3.3 DFT-CLMS

Figure 47 below, displays the results of the frequency modulated signal when the DT-CLMS algorithm is applied to it, and it can be concluded that the spectrum obtained somewhat resembles the ideal frequency spectrum however an issue is present such that the frequency components go to infinity. This because a frequency components resides in the spectrum indefinitely, when the algorithm obtains a frequency component which is due to its adaptive manner, the weights are updated instead of being repeatedly calculated for each instance of time. Hence reverse error propagation is required to erase a spectrums frequency component therefore the frequency components appear to be going on for infinity, but this is due to the propagation of error which requires some more time.

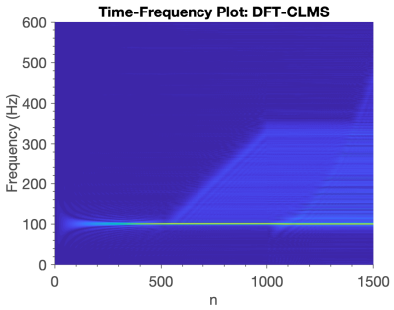

Figure 47: DFT-CLMS: Time-Frequency Estimation

We can use the Leaky-LMS algorithm to eradicate the back-propagation of the error as this enables weights to be disregarded which allows for greater accuracy when predicting weights thus when γ was equal to a very small value like 0.05, the results achieved were vastly more accurate. However, when γ increased above 0.05, the outcomes were not as accurate. The frequency components which are most dominate are unable to be identified correctly which is due to the spectrum being estimated from a sample of the signal when $\gamma > 0.05$.

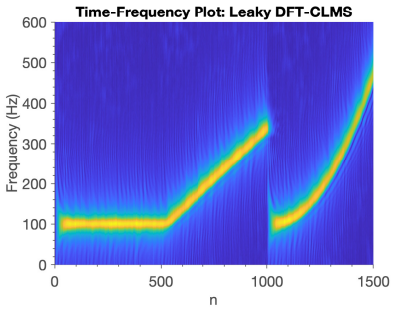

Figure 48: Leaky DFT-CLMS: Time-Frequency Estimation

3.3.4 EEG Signals and DFT-CLMS

The POz which is located on scalp, is where the EEGs spectrum is acquired using both the Leaky DFT-CLMS and DFT-CLMS algorithms, which both display a clear frequency components when the frequency is equal to 50 Hz. As shown in figure 49, at 13Hz and 26Hz, the SSEVPs first and second harmonics are located there respectively, but at 39Hz, the third harmonic un-decipherable

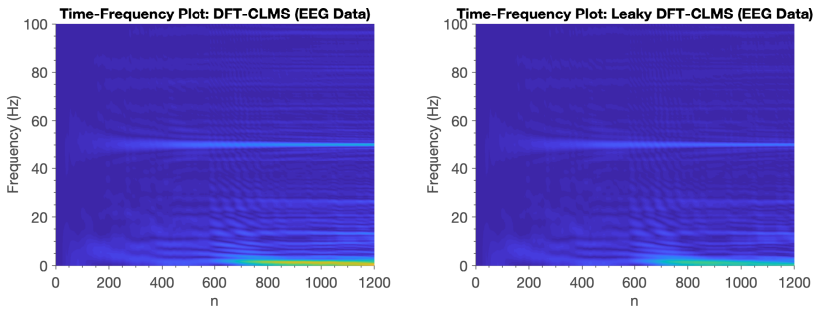

Figure 49: EEG Data: Time-Frequency Estimation

The DFT-CLMS performs much better in this situation compared to the Leaky DFT-CLMS, due to EEGs signal being of a stationary type signal. Due to the Leaky DFT-CLMS algorithm disregarding some its learned weights, it inadvertently intensifies the effect for the latest samples rather than earliest observed samples therefore, when signal is stationary, a lengthier time is needed to converge to the correct solution. However, by utilising backward error propagation, the DFT-CLMS algorithm only disregards weights during this mechanism which is why a converges happens much faster even though the mechanism itself is slow. As seen from the figure above, the resonant frequency components at f= 13Hz, 26Hz, 50Hz are able to be distinguished much more clearly due to the spectrograms brightness at these respective frequencies.

4 From LMS to Deep Learning

4.1 Non-Stationary Time Series

Figure 50 displays two figures: On the left, the original time-series with it's mean plot in orange, and on the right is the zero-mean, standard LMS approximation, $\hat{y}[n]$, plotted on top of the original $y[n]$. With regards to the speed of convergence, $\hat{y}[n]$, begins to converge approximately when the sample number, $n = 180$. For the whole signal: the mean-squared error, $MSE = 40.25$ and Prediction Gain, $Rp = 5.20$. From the point at which signal converges: $MSE_{conv} = 16.66$ and the $Rp_{conv} = 9.39$

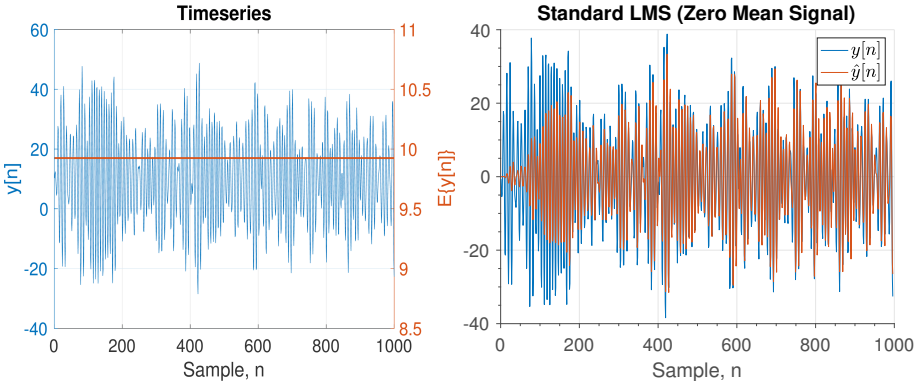

Figure 50: Time-series and Standard LMS

4.2 Activation Function

As seen from figure 51 below, Tan(x) only ranges from -1 to +1 however the data, $y[n]$, ranges from -45 to 45 approximately. Thus, the range of the activation function should meet the range of the data, which could be achieved in two ways: 1) Change the amplitude of the activation function 2) Normalizing the data. For the whole signal: $MSE = 197.52$, $Rp = -23.19$. For the convergence section: $MSE_{conv} = 197.32$, $Rp_{conv} = -23.11$

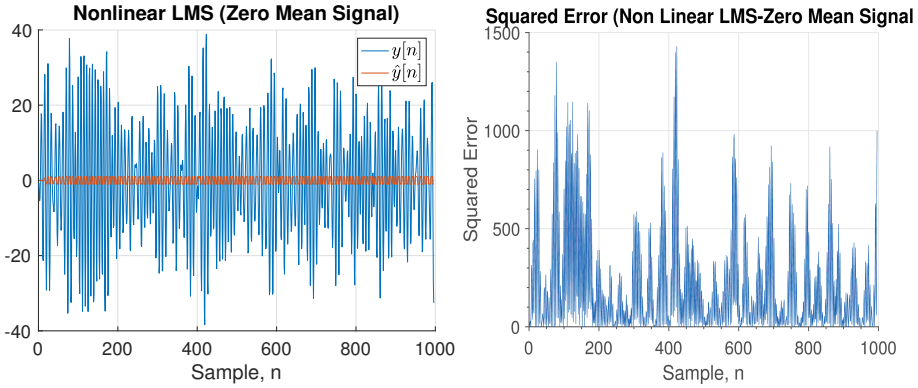

Figure 51: Activation Function Analysis

4.3 Scaling Activation Function

The multiplier, $a = 45$, is a suitable scaling factor as it captures all of the data as seen from figure 52. Also the algorithm converges much faster than the standard LMS. For the whole signal: $MSE = 7.23$, $Rp = 14.64$. For the convergence section: $MSE_{conv} = 6.08$, $Rp_{conv} = 15.44$

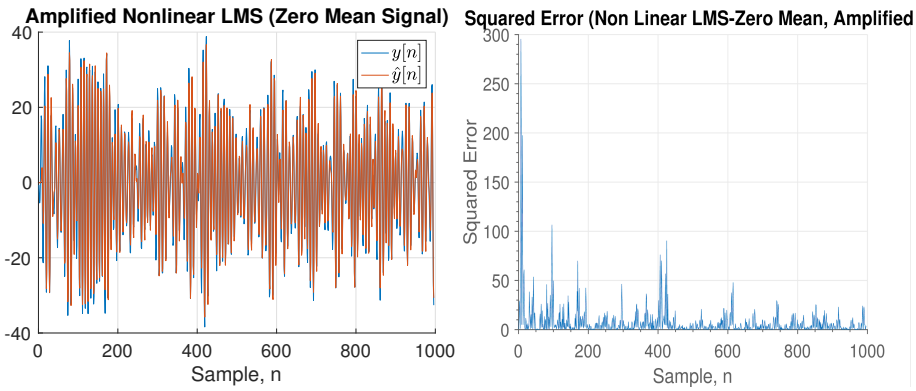

Figure 52: Scaling Activation function to capture all the data

4.4 Non-Zero Mean

When the data has a non-zero mean, the convergence of the algorithm slows down as seen from figure 53. From the error plot, it can be seen that $\hat{y}[n]$ converges approximately when $n = 180$. For the whole signal: $\boldsymbol{MSE = 14.53}$, $\boldsymbol{Rp = 6.22}$. For the convergence section: $\boldsymbol{MSE_{conv} = 11.997}$, $\boldsymbol{Rp_{conv} = 14.87}$

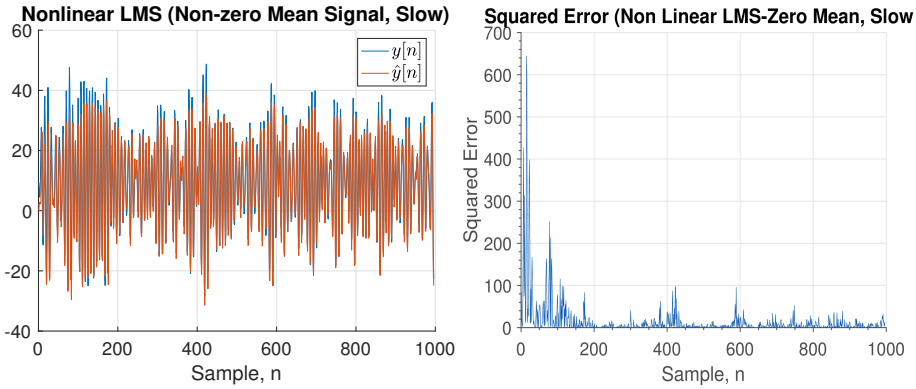

Figure 53: Non-zero mean data

4.5 Epochs and Overfitting

To counteract the slow convergence in the previous part, we can over-fit to the first few samples, by training the weights. From the error plot, it can be seen that $\hat{y}[n]$ converges approximately when $n = 10$. For the whole signal: $\boldsymbol{MSE = 7.75}$, $\boldsymbol{Rp = 14.67}$. For the convergence section: $\boldsymbol{MSE_{conv} = 6.12}$, $\boldsymbol{Rp_{conv} = 14.96}$

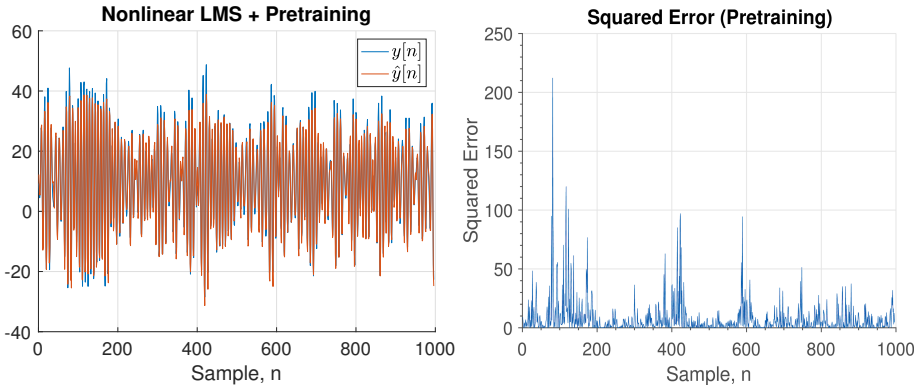

Figure 54: Overfitting by training the weights

4.6 Backpropagation Algorithm

In artificial neural networks **backpropagation**, also known as **the backward propagation of errors**, is the technique used to compute the gradient which is required to calculate the weights required in a network. It also **calculates the error at the output and which is then propagated backwards across the layers of the network**; backpropagation is mainly utilised to train deep neural networks. By applying the chain-rule iteratively in order to calculate the gradient for each layer, the Backpropagation method can be seen as a simplified delta-rule to feedforward networks which are multi-layered and is very similar to the GaussNewton algorithm. From amongst the general automatic differentiation techniques, backpropagation is considered to be unique as it is mainly utilised by gradient descent optimization algorithms to vary certain neurons weights which is achieved by computing the loss functions gradient.

4.7 Deep Neural Network Training

As seen from figure 55, compared with a simple dynamical preceptron, from around 5000 epoch, the networks on the data starts to deteriorate due to overfitting of the training data; this also verified from the loss curve in figure 56 (3rd sub-figure)

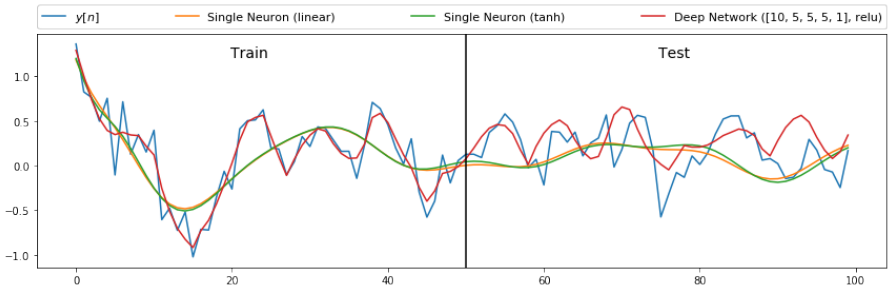

Figure 55: Training and Test Data

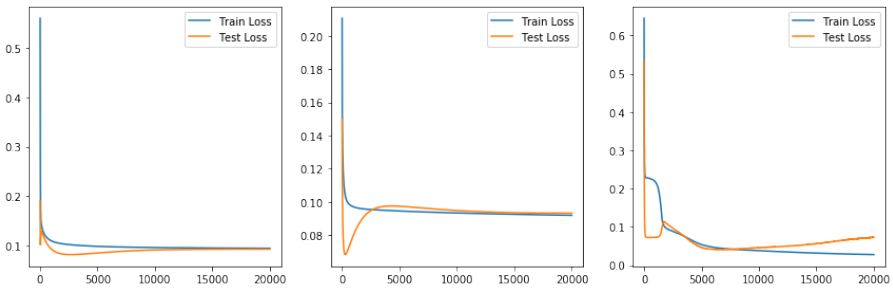

Figure 56: Training and Test Loss Curves

4.8 Deep Neural Network Training - Change in Noise Power

When the noise power is amplified from 0.05 to 5, it could be stated that deep learning is not suitable for high noise environments as it start's to fit the noise, as seen from figure 57, this also verified from the loss curve in figure 58 (3rd sub-figure)

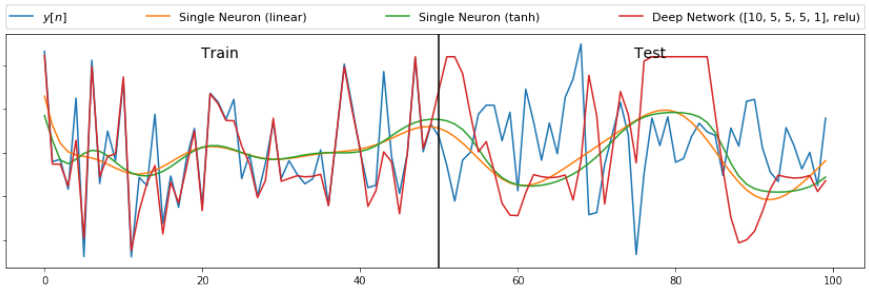

Figure 57: Increased Noise Power - Training and Test Data

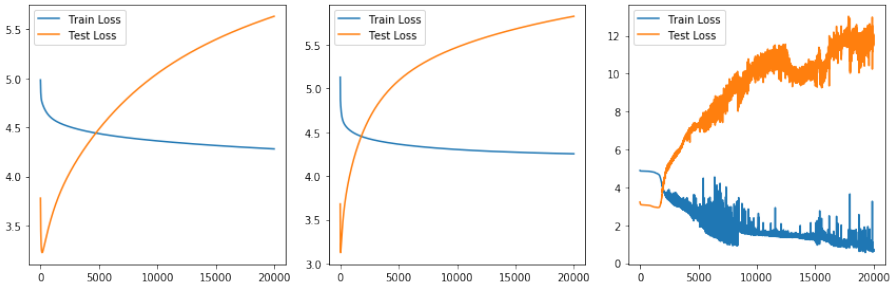

Figure 58: Increased Noise Power - Training and Test Loss Curves