

ELEC60002
- Statistical Signal Processing and Inference Coursework

Sakshi Singh - 01925165

14 March 2023

1 Random signals and Stochastic Processes

1.1 Statistical estimation

A vector \mathbf{x} containing 1000 random samples from a Uniform random variable $X \sim U(1,0)$ was generated. The results are plotted in Figure 1 below.

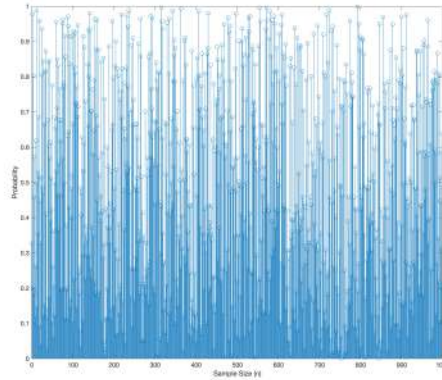


Figure 1: 1000 random samples from a Uniform random variable $X \sim U(1,0)$

1.1.1 Theoretical and Sample Mean

The expected value of X or the *theoretical mean* is 0.5 as the samples are from a uniform random distributions that spans from 0 to 1.

The *sample mean* of \mathbf{x} is given in the table below. The accuracy of sample mean as an estimator of the expected value depends on the sample size as well as the underlying distribution of the random variable. For large sample sizes the sample mean provides a close to accurate estimate of the expected value. From the table below we observe that the calculated results provide a good estimate for the expected value up to 2 decimal places.

| Sample Mean | | | |
|-------------|--------|--------|--------|
| 0.4992 | 0.4913 | 0.4857 | 0.5089 |

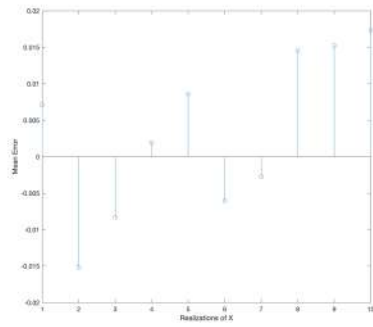
1.1.2 Theoretical and Sample Standard Deviation

The *theoretical standard deviation* of the uniform random variable is calculated to be 0.2887. The *sample standard deviation* of \mathbf{x} is given in the table below. Similar to the accuracy of sample mean, we observe that the sample standard deviations calculated provide a good estimate for the expected value up to 2 decimal places.

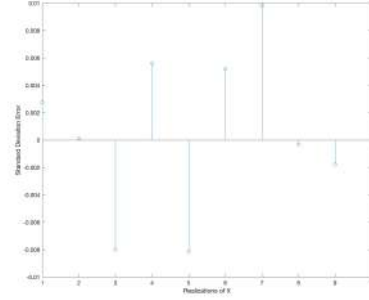
| Sample Standard Deviation | | | |
|---------------------------|--------|--------|--------|
| 0.2934 | 0.2995 | 0.2880 | 0.2815 |

1.1.3 Bias of the Sample Mean Estimation

The *bias* of sample mean and sample standard deviation estimation for ten 1000 sample realizations is shown in Figure 2(a) and 2(b) respectively.



(a) Plot of Mean Error vs. Realizations for ten 1000 sample realizations of X displays the bias of sample mean estimation.



(b) Plot of Standard Deviation Error vs. Realizations for ten 1000 sample realizations of X displays the bias of sample standard deviation estimation.

Figure 2: Bias of sample mean and standard deviation.

1.1.4 Approximating the Probability Density Function of X

Figure 3 below shows the estimated and theoretical PDF from the uniform random variable. As the number of samples increase the histogram estimate of the PDF of X will converge towards the theoretical PDF. This can be observed from the plot as each bin's probability density tends towards 0.1.

Therefore, for larger samples the estimate becomes more accurate.

The number of histogram bins affects the accuracy of the estimate of the PDF of X . A larger number of bins, therefore gives a more accurate estimate than smaller number of bins.

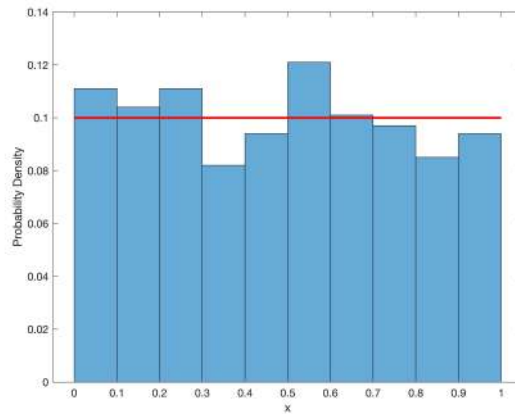


Figure 3: Probability Density Function for 1000 samples from $X \sim U(0,1)$

1.1.5 Statistical Analysis of Gaussian Random Variables

Statistical analysis in 1.11 to 1.14 is now done using Gaussian random variables with a *theoretical mean* of 0 and a *theoretical standard deviation* of 1.

A vector \mathbf{x} containing 1000 random samples from a Gaussian random variable $X \sim N(0,1)$ was generated. The results are plotted in Figure 4 below.

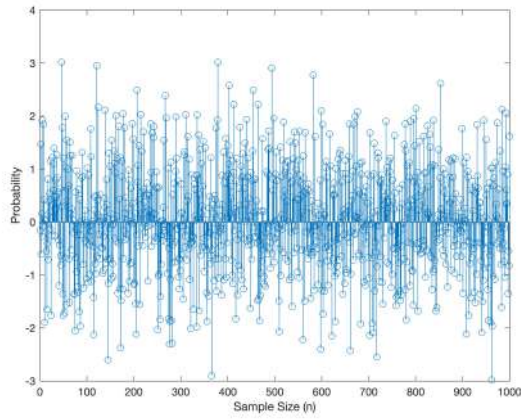


Figure 4: 1000 random samples from a Gaussian random variable $X \sim N(1,0)$

The *bias* of sample mean and sample standard deviation estimation for ten 1000 sample realizations depicted in Figure 5 show that the estimated values lie close to the theoretical value of the mean and standard deviation as the bias values are low.

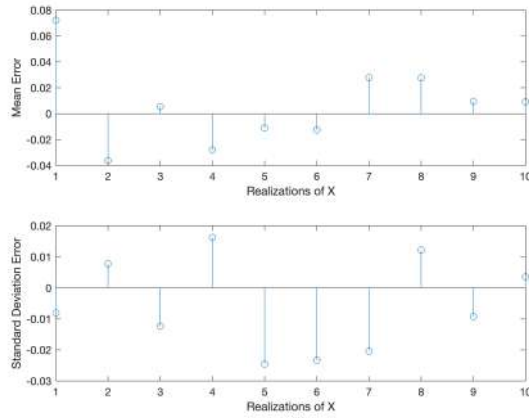


Figure 5: Bias in mean and standard deviation for ten realizations of 1000 sample Gaussian random variables.

The estimate of the Probability Density Function and the theoretical PDF for the Gaussian random variable is given in Figure 6 below.

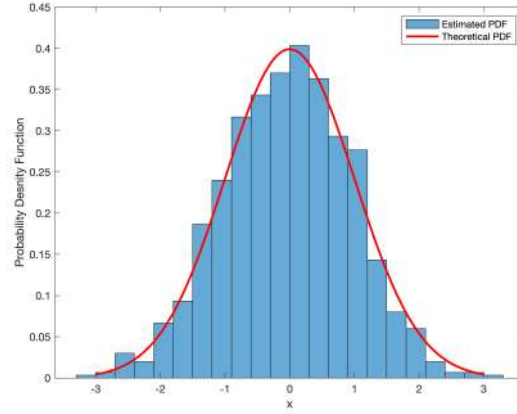


Figure 6: Probability Density Function for 1000 samples from $X \sim N(0,1)$

1.2 Stochastic Processes

1.2.1 Stationarity of Stochastic Processes

A *stationary* process is a stochastic process whose statistical properties such as mean and variance remain constant over time.

Figure 7 below illustrates that the stochastic process *rp1* is not stationary as its ensemble mean and standard deviation both vary with time. Ensemble mean is seen to increase linearly over time, whereas the ensemble standard deviation is seen to increase, reach a peak and then decrease over time.

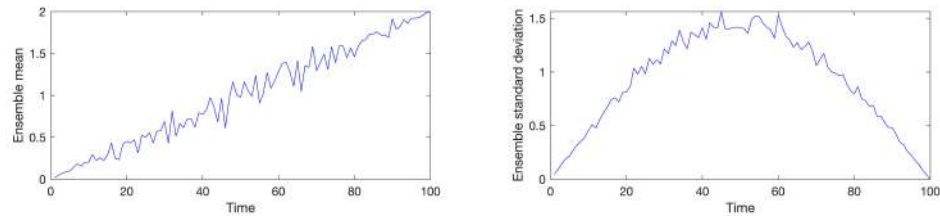


Figure 7: Ensemble mean and Standard deviation over time for *rp1* process

Figure 8 below shows that the stochastic process *rp2* is not stationary as its ensemble mean and standard deviation are not constant over time. This is checked by increasing M and N .

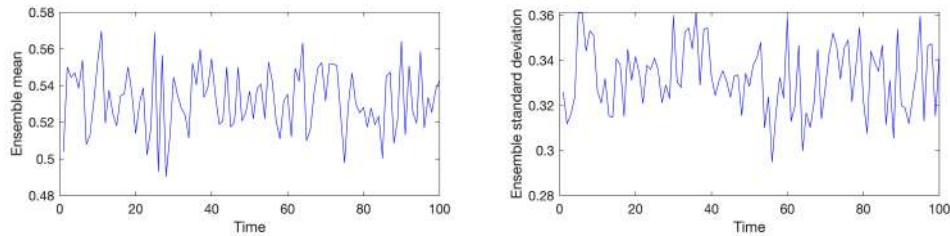


Figure 8: Ensemble mean and Standard deviation over time for *rp2* process

Figure 9 below shows that the stochastic process *rp3* is stationary with a constant mean around 0.50 and a constant standard deviation around 0.87.

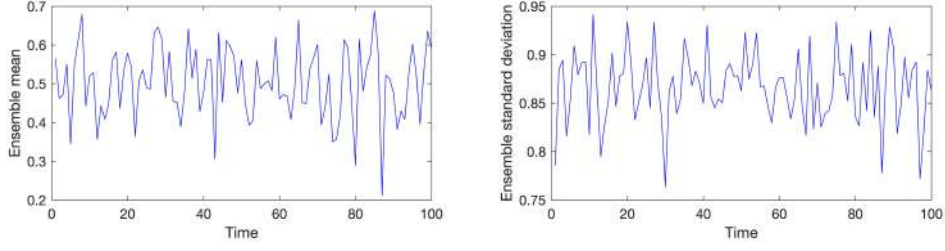


Figure 9: Ensemble mean and Standard deviation over time for rp3 process

1.2.2 Ergodicity of Stochastic Processes

An *ergodic* process is a stochastic process for which the time averages equal the ensemble averages. For an ergodic process, analyzing multiple ensembles at one instant in time provides accurate information about the entire random process.

For *rp1* process, even though it is non stationary, the table below shows that it is ergodic as the mean and standard deviation over time for each of the 4 realizations are similar.

| Realizations | Mean | Standard Deviations |
|--------------|-------|---------------------|
| 1 | 10.01 | 5.89 |
| 2 | 10.01 | 5.87 |
| 3 | 9.97 | 5.88 |
| 4 | 10.00 | 5.87 |

Process *rp2* is not ergodic as seen in the table below. The mean and standard deviation are not constant over time for each realization.

Process *rp3* is ergodic like *rp1* as the mean and standard deviation are constant for all time instants.

| Realizations | Mean | Standard Deviations |
|--------------|------|---------------------|
| 1 | 0.35 | 0.066 |
| 2 | 0.97 | 0.098 |
| 3 | 0.38 | 0.14 |
| 4 | 0.62 | 0.20 |

1.2.3 Mathematical Description of Stochastic process

For *rp1* process:

$$v_{ij} = (r_{ij} - 0.5) \cdot b \sin\left(\frac{\pi i}{N}\right) + ai \quad (1)$$

Where, v_{ij} is the output of a particular sample of random process, r_{ij} is a matrix of random values between -0.5 to 0.5, b and a are constants of 5 and 0.02 respectively and i ranges from 1 to M (All realisations), and j ranges from 1 to N (All samples).

We know that $(r_{ij} - 0.5) \sim U(-0.5, 0.5)$. Therefore, its $E[r_{ij} - 0.5]$ is 0 and $Var[r_{ij} - 0.5]$ is $\frac{1}{12}$. Therefore,

$$E[v_{ij}] = E[(r_{ij} - 0.5) \cdot b \sin\left(\frac{\pi i}{N}\right) + ai] = E[(r_{ij} - 0.5) \cdot b \sin\left(\frac{\pi i}{N}\right)] + E[ai] \quad (2)$$

$$E[v_{ij}] = 0 + E[ai] = 0.02i \quad (3)$$

$$\sigma_{v_{ij}}^2 = Var[v_{ij}] = Var[(r_{ij} - 0.5) \cdot b \sin\left(\frac{\pi i}{N}\right) + a] = \frac{b^2}{12} \sin^2\left(\frac{\pi i}{N}\right) \quad (4)$$

Hence, the theoretical results match with those obtained by sample averaging in section 1.2.1

For rp2 process:

$$v_{ij} = (r_{ij} - 0.5) \cdot Mr_{ij} + Ar_{ij} \quad (5)$$

Ar_{ij} and Mr_{ij} are randomly distributed numbers between 0 and 1, so $E[Ar_{ij}]$ and $E[Mr_{ij}]$ is 0.5 and $Var[Ar_{ij}]$ is $\frac{1}{12}$.

$$E[v_{ij}] = E[(r_{ij} - 0.5) \cdot Mr_{ij} + Ar_{ij}] = E[r_{ij} - 0.5] \cdot E[Mr_{ij}] + E[Ar_{ij}] \quad (6)$$

$$E[v_{ij}] = 0 + E[Ar_{ij}] = 0.5 \quad (7)$$

Hence, this shows that the theoretical mean of v_{ij} tends to a constant value of 0.5 when many realizations are considered.

$$\sigma_{v_{ij}}^2 = E[v_{ij}^2] - E[v_{ij}]^2 = E[(r_{ij} - 0.5)^2 \cdot Mr_{ij}^2 + 2(r_{ij} - 0.5) \cdot Mr_{ij} \cdot Ar_{ij} + Ar_{ij}^2] \quad (8)$$

We know, $E[r_{ij} - 0.5]$ is 0 and $Var[r_{ij} - 0.5]$ is $\frac{1}{12}$. $E[Ar_{ij}^2]$ is 0.333 and $E[Mr_{ij}^2]$ is 0.333. Therefore,

$$\sigma_{v_{ij}}^2 = 0.333 + (0.333 \cdot \frac{1}{12}) - 0.25 = 0.111 \quad (9)$$

Hence, both the theoretical results match with those obtained by sample averaging in Section 1.2.1.

For rp3 process:

$$v_{ij} = (r_{ij} - 0.5) \cdot m + a \quad (10)$$

We know that $(r_{ij} - 0.5) \sim U(-0.5, 0.5)$, m is 3 and a is 0.5.

$$E[v_{ij}] = E[(r_{ij} - 0.5) \cdot m + a] = E[r_{ij} - 0.5] \cdot m + a = a = 0.5 \quad (11)$$

$$\sigma_{v_{ij}}^2 = E[v_{ij}^2] - E[v_{ij}]^2 = E[m^2 x^2 + a^2 + 2 \cdot a m (r_{ij} - 0.5)] - 0.25 = 0.75 \quad (12)$$

Hence, both theoretical mean and variance match with those obtained by sample averaging in Section 1.2.1

1.3 Estimation of Probability Distributions

1.3.1 Estimating PDF for a stationary process

The MATLAB function to estimate the pdf using in-built `hist` function is as follows:

```
>> function [counts, bins] = pdf(v)
    N = length(v);
    [counts, bins] = hist(v, 100); % Compute histogram of data with 100 bins
    counts = counts / trapz(bins, counts); % Normalize the histogram by number of samples considered
    bar(bins, counts); % Plot histogram bars
    xlabel('x');
    ylabel('Probability');
end
```

Figure 10: MATLAB Code for estimating PDF

A large N (100000) is used as it allows the estimated PDF to converge towards the theoretical PDF, giving a more accurate PDF estimation. The PDF is normalized by the number of samples

considered. This is done by dividing the frequency or counts of each bin by N . The in-built `trapz` function allows us to do this by calculating the total area of each histogram bar. The estimated PDF is shown below in Figure 11.

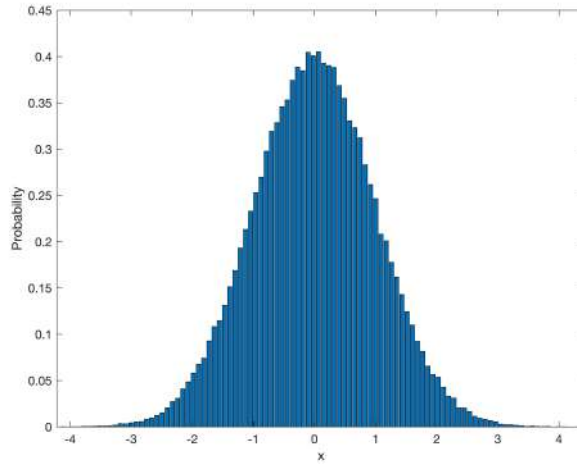


Figure 11: Estimated PDF for Stationary Process

1.3.2 Estimating PDF for a stationary and ergodic process

We analyze process $rp3$ as it is both stationary and ergodic. From Figure 12 below we observe that as data length N increases, the estimated PDF of $rp3$ process converge to the theoretical PDF.

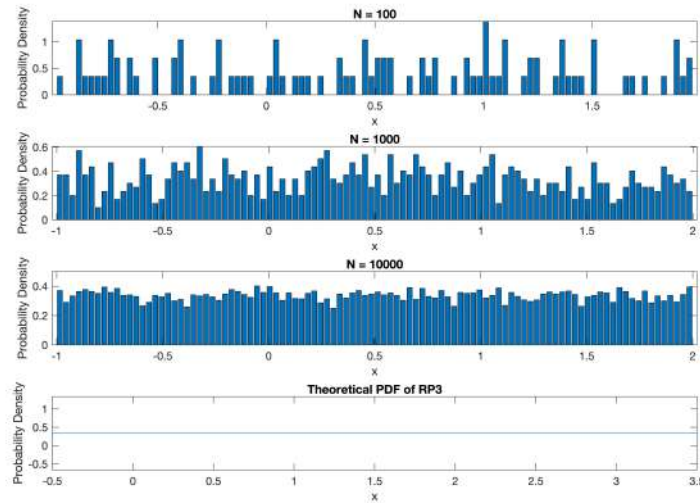


Figure 12: Estimated PDF of $rp3$ process for different N values compared to Theoretical PDF of $rp3$ process.

1.3.3 Estimating PDF for a non-stationary process

It is possible to estimate the PDF of non-stationary process with `pdf` function, however it may not correctly estimate the PDF of the entire signal. This is due to the mean and variance of the process being time varying.

Moreover, when the mean changes from 0 to 1 for a 1000 sample long signal at $N=500$, it can be difficult to determine the underlying distribution, thereby making it difficult to accurately estimate the PDF.

One method to overcome this difficulty is to divide the signal into 2 parts with 2 different means and estimate the PDF separately.

2 Linear stochastic modelling

2.1 Autocorrelation function (ACF) of uncorrelated and correlated sequences

2.1.1 Calculating the unbiased estimate of ACF for a 1000-sample realisation of WGN

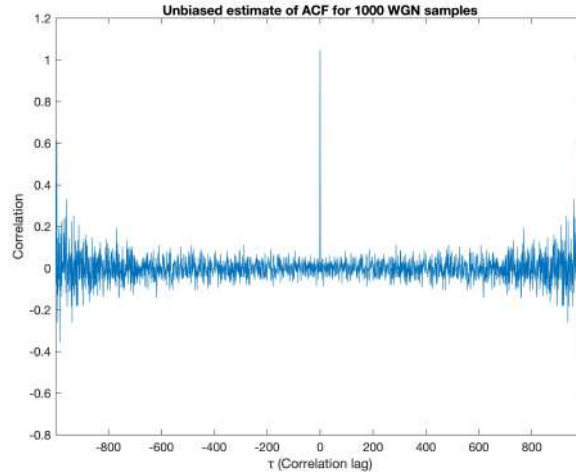


Figure 13: Unbiased estimate of ACF for WGN.

The unbiased estimate of the autocorrelation function of 1000 random White Gaussian Noise (WGN) is shown in Figure 13. The ACF would ideally have a Dirac delta function at $\tau = 0$ and be 0 everywhere else if the WGN samples were random and completely uncorrelated. Since the signal is perfectly correlated, there is a Dirac delta function at $\tau = 0$, but the ACF is not zero at other correlation lags. It is also observed that the ACF is symmetric about $\tau = 0$, as the WGN signal is real.

2.1.2 Using the `zoom` command to focus on $\tau = 0$

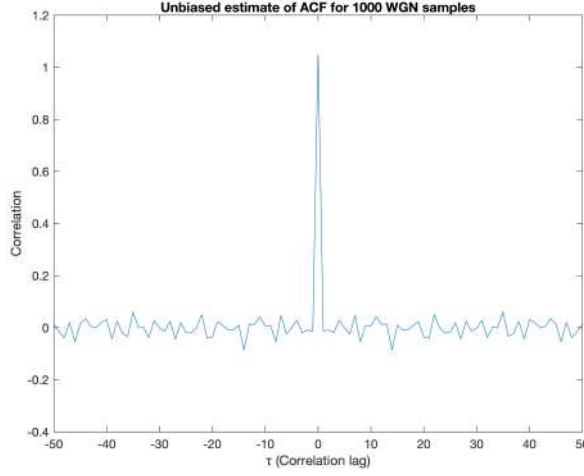


Figure 14: Zooming in at $\tau = 0$

Figure 14 shows that the ACF estimate has a Dirac delta function at $\tau = 0$ and close to 0 everywhere else. However, as τ increase, close to $\tau = \pm 999$, the ACF estimate is much larger and greater than the peak of 1.

2.1.3 Effects of a large autocorrelation lag τ

$$R_{x, \text{unbiased}}(\tau) = \frac{1}{N-|\tau|} \sum_{n=0}^{N-|\tau|-1} x[n]x[n+\tau] \quad (13)$$

From equation 13 above, we know that when the lag or τ values increase, it means that the lagged values get further apart from each other. As a result, the correlation between them becomes weaker. Therefore, the accuracy of ACF estimates decreases because the sample size decreases due to the reduction in the number of lags. Moreover, for large τ values, the ACF estimates become less statistically reliable because the assumption of independence between the observations is violated. As we saw from Figure 13 and Figure 14, the estimator begins diverging after $\tau = \pm 500$, a considerable empirical bound would be half of this value, that is $\tau = \pm 250$.

2.1.4 Using a Moving Average (MA) filter

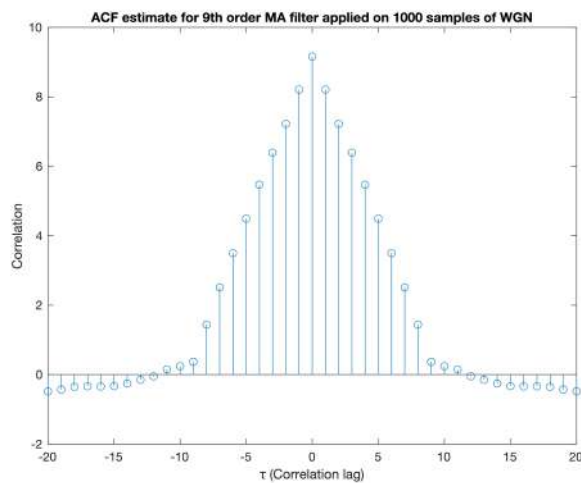


Figure 15: ACF of a 9th order MA Filter

Figure 15 displays the plot of the ACF estimate after applying a 9th order MA filter to 1000 realisations of WGN samples. It can be observed that the ACF estimate is a triangular function between $-8 < \tau < 8$ with a significant peak at $\tau = 0$. Ideally, ACF should be 0 outside of this range, however, we see values because we are using an estimate. The impulse response of this filter will also show a similar symmetrical shape.

Reducing the order of the filter will reduce the width and height of the triangular function observed.

2.1.5 Autocorrelation Function of Stochastic Process

Since X_n is an uncorrelated process, then its autocorrelation function can be represented as $R_X(\tau) = a\delta(\tau)$, where a is a constant and $\delta(\tau)$ is the Delta Dirac function.

Thus, the ACF of Y_n , $R_Y(\tau)$, can be represented as the convolution of the impulse response of the filter $h(n)$ with the Dirac delta function,

$$R_Y(\tau) = R_X \tau * R_h(\tau)$$

, which yields the impulse response itself shifted by τ . This implies that $R_Y(\tau)$ represents the autocorrelation function of the output process R_Y after being filtered by a linear time-invariant filter with an impulse response $h(n)$.

Therefore, the ACF of Y_n is equal to the scaled version of the ACF of the impulse response.

2.2 Cross-correlation Function (CCF)

2.2.1 Estimating the CCF

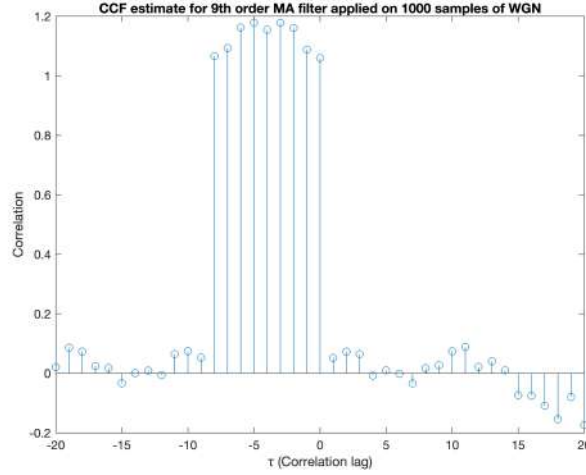


Figure 16: Cross correlation of input and output of a 9th order MA filter

Since a moving average filter of order 9 is used, the current and past 8 values of X are used to compute each value of Y , leading to a correlation between X and the current and last 8 values of Y . This is shown by the plot in Figure 16.

Since X_t is uncorrelated, $R_{XY}(\tau) = h(\tau) * R_X(\tau)$ becomes $h(\tau) * \delta(t) = h(\tau)$. Therefore, R_{XY} is an impulse train represented by $h(\tau) = \sum_{i=0}^N \delta(\tau - i)$.

2.2.2 Application of CCF for System Identification

Cross-correlation of an input signal with its output through a linear time-invariant (LTI) system gives us the impulse response of the system, assuming that the input signal is uncorrelated. The impulse response describes how the system responds to an impulse input, and its shape (number of delta peaks) can provide insight into the order and structure of the system.

If the system has a finite impulse response (FIR), the impulse response will be limited in duration, and the coefficients can be estimated directly from the impulse response. The amount of delta peaks in the impulse response indicates the order of the filter.

If the system has an infinite impulse response (IIR), the impulse response will be infinite in duration and may lack any distinct delta peaks. Other methods, such as the least squares method, may be used to determine the filter coefficients.

2.3 Autoregressive Modelling

2.3.1 Stability of AR(2) Coefficients - Stability Triangle

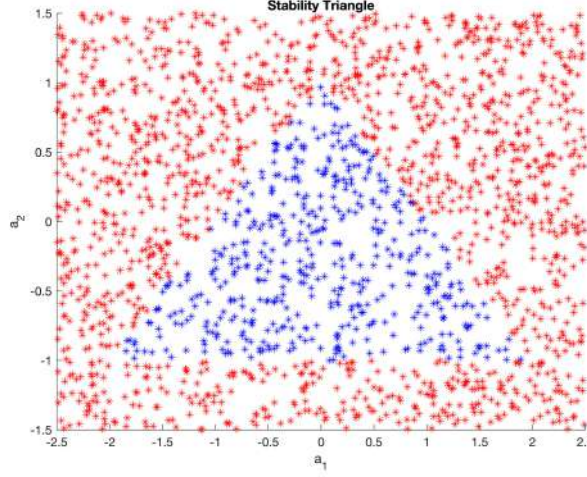


Figure 17: Stability of coefficients a1 and a2

Figure 17 displays the stable and unstable coefficient pairs (a_1, a_2) of an AR(2) process in blue and red respectively. We begin to look at the theoretical properties of AR(2) process to understand which values of the pairs (a_1, a_2) preserve the stability of this process.

The process is stable if its poles (the roots of the characteristic equation) are inside the unit circle in the complex plane. The characteristic equation of the AR(2) process is:

$$z^2 - a_1 z - a_2 = 0 \quad (14)$$

For the positive (real) root:

$$\frac{a_1 \pm \sqrt{a_1^2 + 4a_2}}{2} < 1$$

$$a_1^2 + 4a_2 < (2 - a_1)^2$$

$$a_1 + a_2 < 1$$

For the negative root we obtain $a_2 - a_1 < 1$. Since both roots are less than 1, we know $|a_2| < 1$.

Therefore, there are 3 stability conditions that can be visualised in the stability triangle in Figure 18:

$$a_1 + a_2 < 1 \quad (15)$$

$$a_1 - a_2 < 1 \quad (16)$$

$$|a_2| < 1 \quad (17)$$

The above expression defines the boundaries of the stability region in the (a_1, a_2) plane. If we plot these boundaries, we get two curves that intersect. The region enclosed by these curves is the stability region or convergence, which has a triangular shape.

2.3.2 ACF of Sunspot Time Series

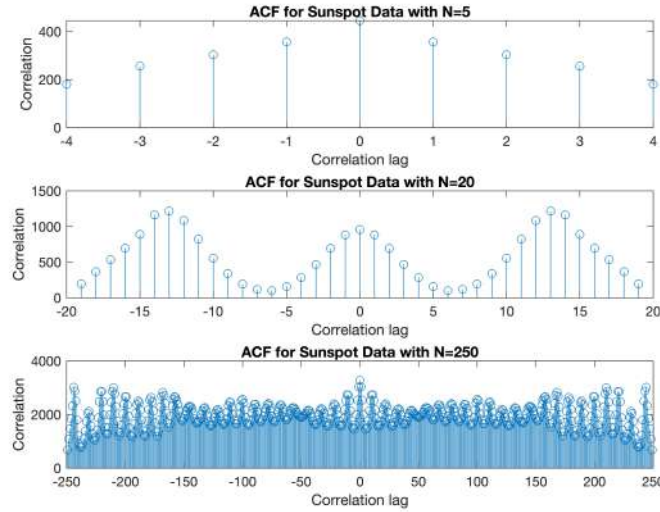


Figure 18: ACF of Sunspot Time Series for Different Data Length(N)

From Figure 18, we observe that for $N=5$, the ACF doesn't show any significant pattern because the data length is too short. For shorter data lengths, the ACF will have more variability. As we increase the data length to $N=20$, we observe a pattern that repeats every 13 years. For $N=250$, the plot shows an irregular pattern. This is because the sample number is greater than the empirical bound as discussed in section 2.1.3.

The mean of a time series can also affect the shape of the ACF, but only in terms of shifting the ACF up or down. This is because the ACF measures the degree of correlation between the time series and a lagged version of itself, and the shape of this correlation does not depend on the mean of the sunspot time series. When we compute the ACF of the zero-mean version of the sunspot series, we are essentially removing the mean from the data before calculating the ACF. This means that the resulting ACF will have the same shape as the original ACF, but the values will be shifted down by the mean of the series. This is shown in Figure 19 below.

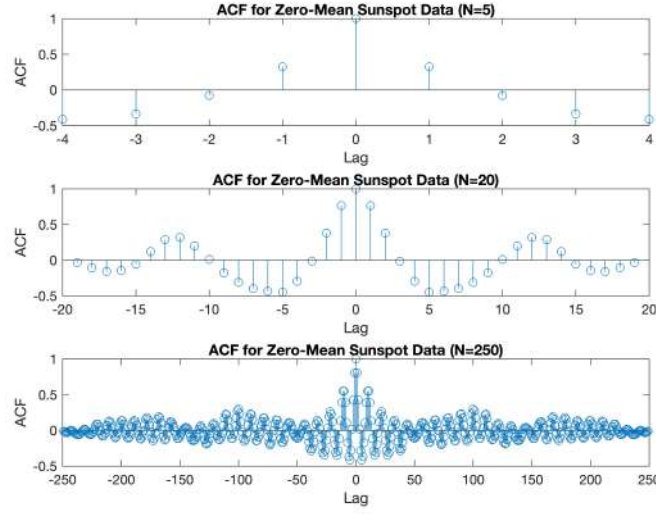


Figure 19: Zero-mean ACF of Sunspot Time Series for Different Data Length(N)

2.3.3 Partial Correlation Function (PCF) of the Sunspot Time Series

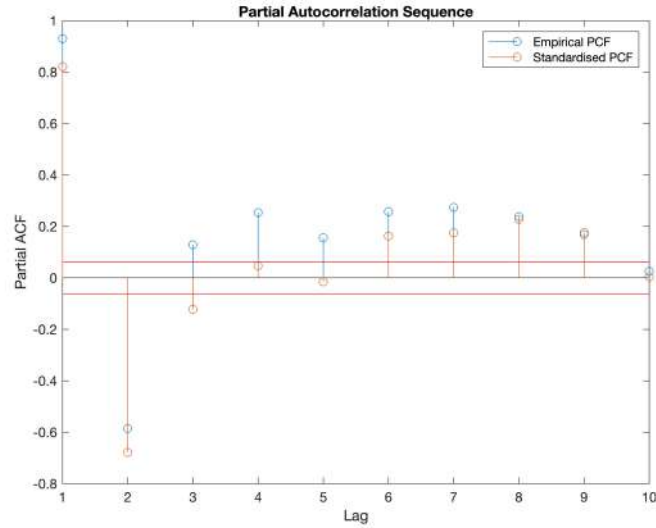


Figure 20: Partial Correlation Function up until the model order $p = 10$

Figure 20 shows the plot of PCF up until model order of 10 using the Yule-Walker equations, along with 95% confidence bounds.

The most likely model order can be determined by examining the partial autocorrelation function (PCF) of the sunspot time series. The only values of the PCF outside the 95% confidence bounds occur at lags 1 and 2 and after lag 2 the values converge to 0. This indicates that the correct model order for the AR process is 2.

By standardizing the series the main difference observed is that the partial correlation functions are different. The standardized PCF is more accurate since it removes any offsets.

2.3.4 Determining the correct model order using Minimum Description Length (MDL) and the Akaike Information Criterion (AIC) and Corrected AIC (AICc)

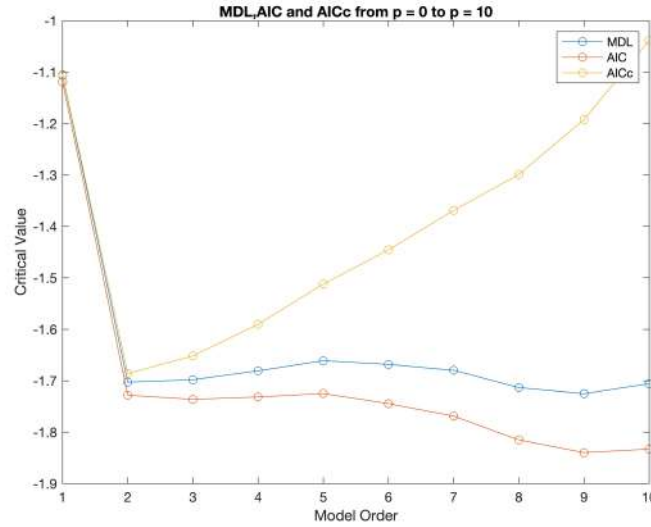
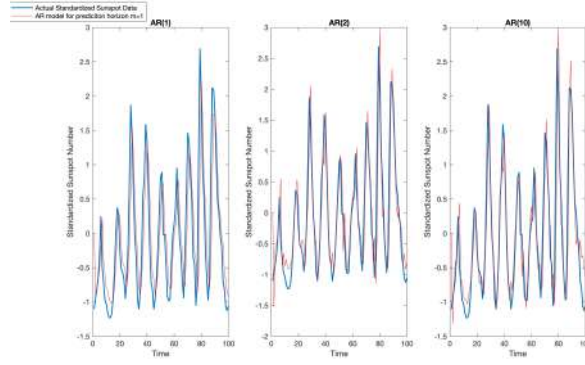


Figure 21: MDL, AIC, AICc from order $p=0$ to $p=10$

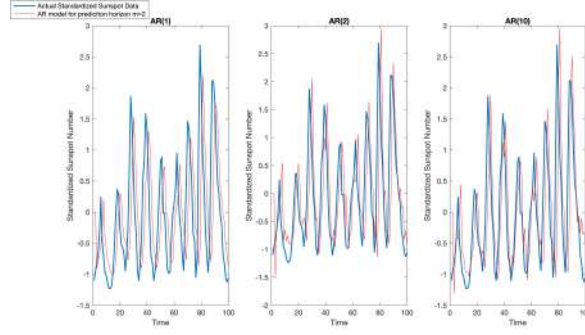
The 3 criteria for model order selection for standardised data are plotted on Figure 21. These establish a trade-off between computational complexity and error reduction, by introducing “penalty” for a high model order.

To determine the model order from plots of MDL, AIC, and AICc, we should look for the point at which the curves reach a minimum. This is seen at order $p=2$ in Figure 21. This point indicates the model order beyond which adding more parameters does not significantly improve the model fit, and may lead to overfitting.

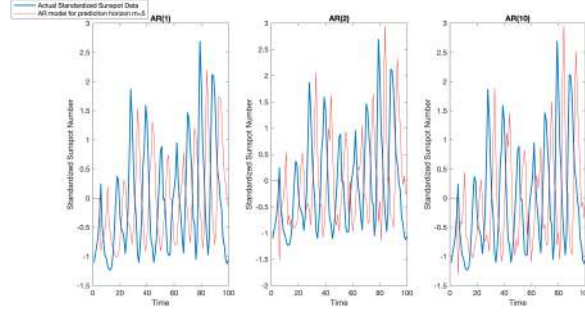
2.3.5 AR Modeling to Predict the Sunspot Time Series



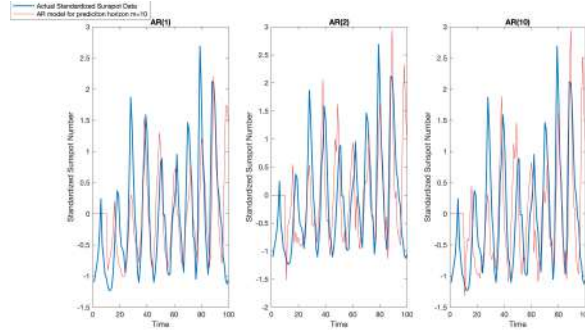
(a) Real and AR predicted sunspot data for $m = 1$



(b) Real and AR predicted sunspot data for $m = 2$



(c) Real and AR predicted sunspot data for $m = 5$



(d) Real and AR predicted sunspot data for $m = 10$

Figure 22: Real and AR predicted sunspot data for different model orders.

From Figure 22, we observe that increasing the model order from AR(1) to AR(10) provides us with a more accurate prediction of the real standardized sunspot data. Increase the prediction horizon leads to decrease in the amplitude of the signal and so, accuracy of the predicted data decreases. Therefore, for larger prediction horizons we can conclude that AR(1) and AR(2) are unsuitable for predicting the data.

Overall, increasing the model order may result in overfitting, resulting in greater prediction errors.

2.4 Cramer-Rao Lower Bound

2.4.1 Showing that an AR(1) model is sufficient to describe the daily returns of the index

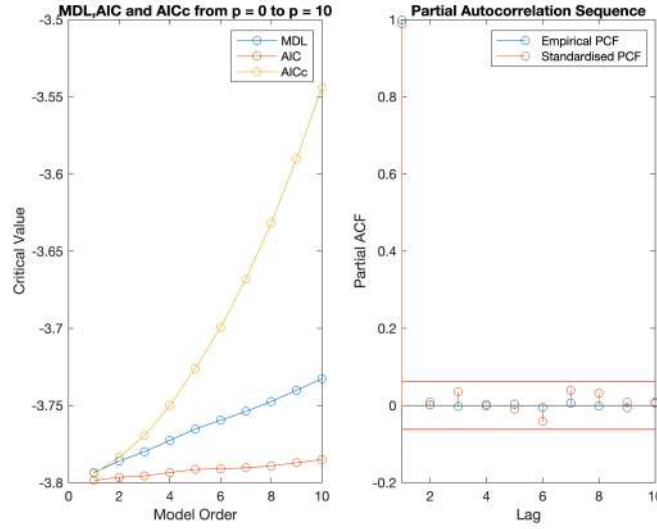


Figure 23: Information theoretic criteria and PCF for NASDAQ data

Figure 23 demonstrates that at $p=1$, the error for MDL, AIC is minimum. The only values of the PCF outside the 95% confidence bounds occur at lag 1, after which every value converges to 0. These metric indicate that the correct model order for the AR process is 2.

2.4.2 Computing the Fisher Information Matrix

To compute the fisher matrix we need to compute the partial derivatives of the log likelihood function

$$\frac{\partial \ln [\hat{P}_X(f; \theta)]}{\partial a_1} = \frac{2e^{-j2\pi f} [1 - \hat{a}_1 e^{-j2\pi f}]}{[1 - \hat{a}_1 e^{-j2\pi f}][1 - \hat{a}_1 e^{j2\pi f}]} \quad (18)$$

$$\frac{\partial \ln [\hat{P}_X(f; \theta)]}{\partial \sigma^2} = \frac{1}{\hat{\sigma}^2} \quad (19)$$

The fisher matrix is 2X2 matrix and the elements of the matrix can be calculated as

$$[I(\theta)]_{11} = \frac{1}{2} \int_{0.5}^{0.5} \left(\frac{\partial \ln [\hat{P}_X(f; \theta)]}{\partial a_1} \right)^2 df \quad (20)$$

$$[I(\theta)]_{12} = [I(\theta)]_{21} = \frac{1}{2} \int_{0.5}^{0.5} \frac{\partial \ln [\hat{P}_X(f; \theta)]}{\partial a_1} \frac{\partial \ln [\hat{P}_X(f; \theta)]}{\partial \sigma^2} df \quad (21)$$

$$[I(\theta)]_{22} = \frac{1}{2} \int_{0.5}^{0.5} \left(\frac{\partial \ln [\hat{P}_X(f; \theta)]}{\partial \sigma^2} \right)^2 df \quad (22)$$

Therefore, further simplifying gives us the fisher matrix elements as

$$[I(\theta)]_{11} = \frac{1}{2} \int_0^{\frac{1}{2}} \left(\frac{2e^{-j2\pi f} [1 - \hat{a}_1 e^{-j2\pi f}]}{[1 - \hat{a}_1 e^{-j2\pi f}][1 - \hat{a}_1 e^{j2\pi f}]} \right)^2 df \quad (23)$$

$$[I(\theta)]_{12} = [I(\theta)]_{21} = \frac{1}{2} \int_0^{\frac{1}{2}} \left(\frac{2e^{-j2\pi f} [1 - \hat{a}_1 e^{-j2\pi f}]}{[1 - \hat{a}_1 e^{-j2\pi f}][1 - \hat{a}_1 e^{j2\pi f}]} \right) \left(\frac{1}{\hat{\sigma}^2} \right) df \quad (24)$$

$$[I(\theta)]_{22} = \frac{1}{2} \int_0^{\frac{1}{2}} \left(\frac{1}{\hat{\sigma}^2} \right)^2 df \quad (25)$$

Where the fisher matrix is in the form

$$\begin{bmatrix} [I(\theta)]_{11} & [I(\theta)]_{12} \\ [I(\theta)]_{21} & [I(\theta)]_{22} \end{bmatrix} \quad (26)$$

2.4.3 Proving the CRLB Inequalities for the Variance

The variance inequality given by

$$\text{var}(\theta_i) \geq [I(\theta)^{-1}]_{ii} \quad (27)$$

For $\hat{\sigma}^2$,

$$\text{var}(\hat{\sigma}^2) \geq [I(\theta)^{-1}]_{22} \quad (28)$$

For \hat{a}_1 ,

$$\text{var}(\hat{a}_1) \geq [I(\theta)^{-1}]_{11} \quad (29)$$

The CRLB heatmap for $\hat{\sigma}^2$ and \hat{a}_1 is

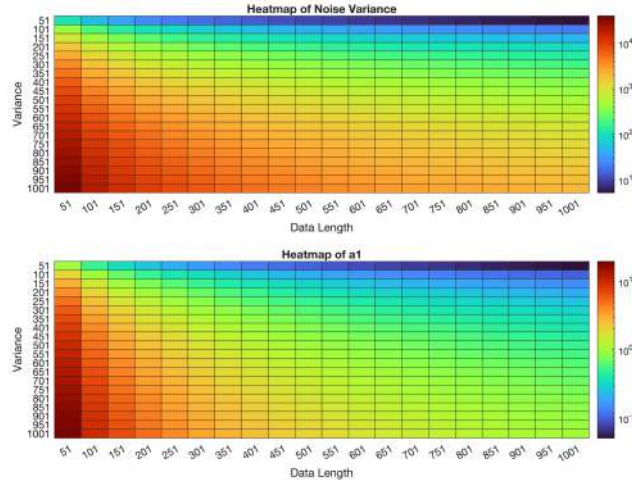


Figure 24: CRLB heatmaps for $\hat{\sigma}^2$ and \hat{a}_1

2.4.4 Computing the bound in terms of $A(f)$

We know,

$$\frac{\partial \ln [\hat{P}_X(f; \theta)]}{\partial a_1} = \frac{2e^{-j2\pi f} [1 - \hat{a}_1 e^{-j2\pi f}]}{[1 - \hat{a}_1 e^{-j2\pi f}][1 - \hat{a}_1 e^{j2\pi f}]} \quad (30)$$

$$\frac{\partial \ln [\hat{P}_X(f; \theta)]}{\partial \sigma^2} = \frac{1}{\hat{\sigma}^2} \quad (31)$$

The bounded terms for $A(f)$ can be found as, where $A(f)$ is $A(f) = 1 - \sum_{m=1}^p a[m]e^{-j2\pi f m}$,

$$\text{var}(\hat{P}_x(f; \theta)) \geq \begin{bmatrix} \frac{2e^{-j2\pi f} [1 - \hat{a}_1 e^{-j2\pi f}]}{[1 - \hat{a}_1 e^{-j2\pi f}][1 - \hat{a}_1 e^{j2\pi f}]} & \frac{1}{\hat{\sigma}^2} \end{bmatrix} \begin{bmatrix} [I(\theta)]_{11} & [I(\theta)]_{12} \\ [I(\theta)]_{21} & [I(\theta)]_{22} \end{bmatrix} \begin{bmatrix} \frac{2e^{-j2\pi f} [1 - \hat{a}_1 e^{-j2\pi f}]}{[1 - \hat{a}_1 e^{-j2\pi f}][1 - \hat{a}_1 e^{j2\pi f}]} \\ \frac{1}{\hat{\sigma}^2} \end{bmatrix} \quad (32)$$

2.5 Real world signals: ECG from iAmp experiment

2.5.1 Heart Rate Probability Density Estimate (PDE)

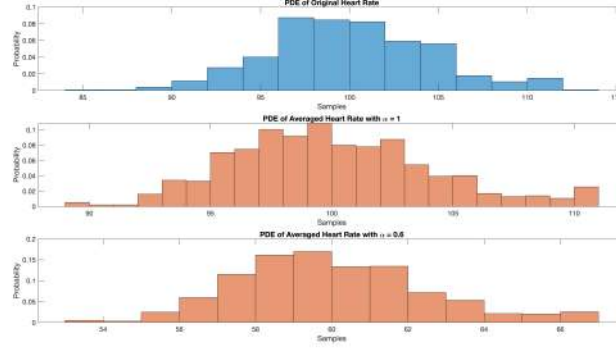


Figure 25: Heart Rate PDE

a) Figure 25 depicts the (PDE) of the original heart rates $h[n]$ and the averaged heart rates $\hat{h}[n]$ for $\alpha = 1$ and $\alpha = 0.6$.

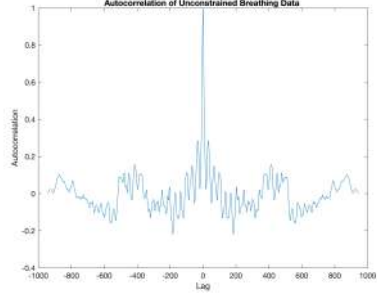
b) Overall, it can be seen that averaging every 10 samples of the heart rate lowers the variance of the data. This can be seen from the 2 plots of averaged heart rate which has a lower spread than the original heart rate plot.

The constant α affects the PDE by shifting the mean and changing the variance of the data. When the $\alpha = 1$, the second subplot has the same mean but a lower variance compared to the original PDE. When the $\alpha = 0.6$, the third subplot has even a lower mean and variance than the second subplot.

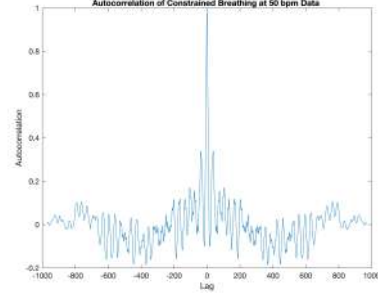
The constant α can be used to minimise effect on noise and has affects the PDE by smoothing the data. A smaller value of α as seen in the third subplot in Figure 28, leads to a smoother PDE as it reduces the noise and variations in the data.

2.5.2 AR Modelling of Heart Rate.

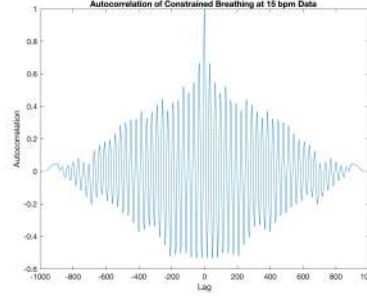
c) From Figure 26 shown below, we can infer that the RRI data is an AR process as it has a sinusoidal pattern and decays exponentially.



(a) ACF of Unconstrained Breathing Data



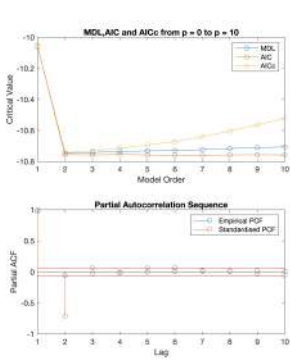
(b) ACF of constrained breathing at 50 bpm



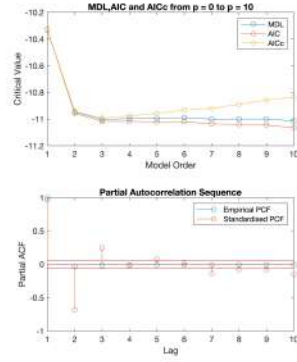
(c) ACF of constrained breathing at 15 bpm

Figure 26: ACF of the three RRI trials

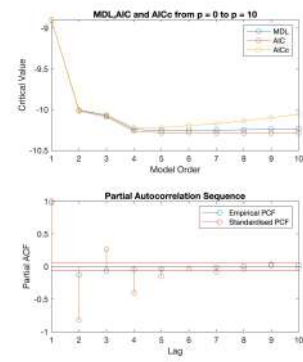
d) By looking at the PCF plot for each RRI trial in Figure 27 below, we can conclude that RRI signal for the three trials can be modelled by an AR(2) process. The only values of the PCF outside the 95% confidence bounds occur at lags 1 and 2 and after lag 2 the values converge to 0. This indicates that the correct model order for the AR process is 2.



(a) Model selection criteria and PCF for unconstrained breathing data



(b) Model selection criteria and PCF for constrained breathing at 50 bpm



(c) Model selection criteria and PCF for constrained breathing at 15 bpm

Figure 27: Model selection criteria and PCF for three RRI trials

3 Spectral Estimation and Modelling

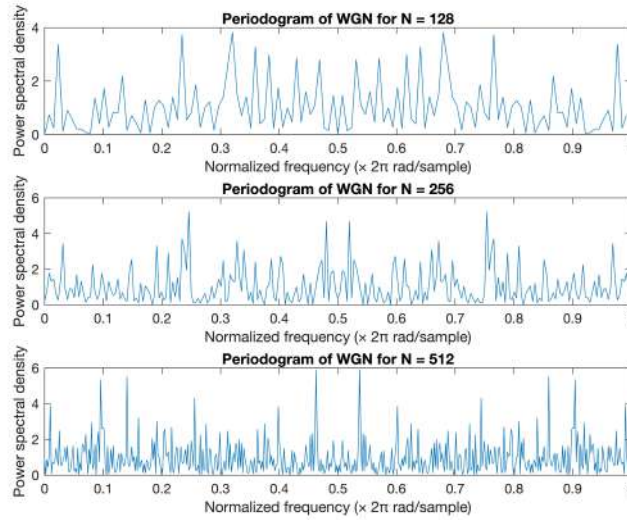


Figure 28: Estimated power spectral density for White Gaussian Noise

We use N -sample realisation of WGN as input of the function `pwm` which generates an estimated power spectral density (PSD) as an output using the periodogram method. The PSD is seen above in Figure 28.

We can also see that the periodogram is symmetrical around a normalized frequency of 0.5, due to the fact that the input signal is real.

The estimated PSD does not exactly match the ideal PSD for WGN, which is equal to 1 at all frequencies. This is due to the variance of the unbiased autocorrelation function (ACF) estimator used to calculate PSD, which increases at large lags. Since the periodogram is essentially a Fourier transform of the ACF, the periodogram estimator's variance also increases for large lags. As a result, the estimated PSD may not be consistent.

To address this issue, various methods can be used to reduce the variance of the periodogram estimator, such as windowing or averaging multiple periodograms.

3.1 Averaged periodogram estimates

3.1.1 Smoothing the Periodogram

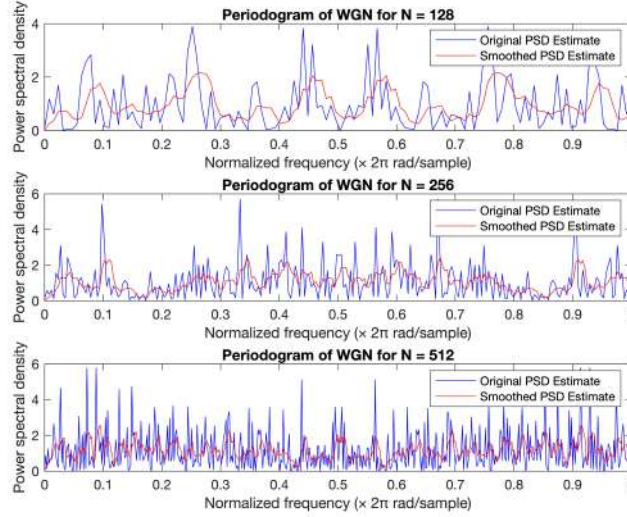


Figure 29: Smoothed estimated power spectral density for White Gaussian Noise

The resulting plot in Figure 29 shows that the filtered PSD estimate (in red) is smoother than the unfiltered estimate (in blue), as the noise-induced fluctuations are reduced. However, even though the variance of the periodogram estimate decreases after filtering, bias is introduced.

3.1.2 PSD Estimates for Eight Non-Overlapping Segments of WGN Sequence

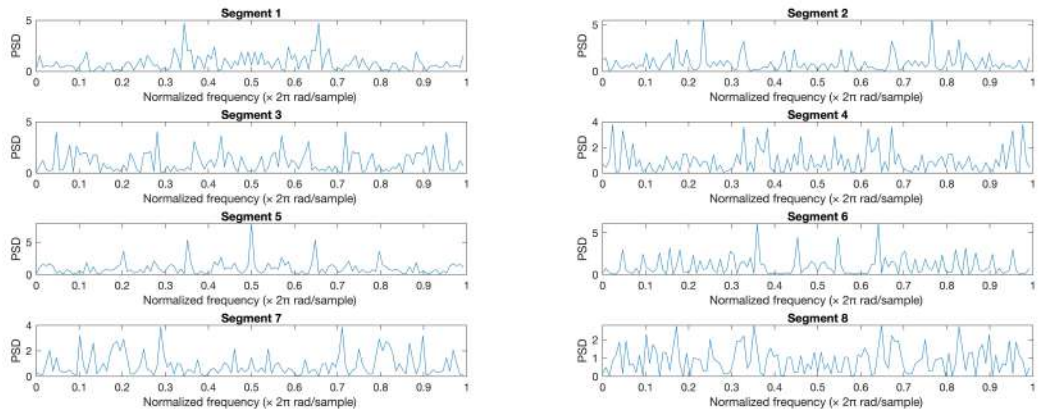


Figure 30: PSD estimate for 128 sample segments of WGN

The resulting plots in Figure 30 show the PSD estimates for each segment as a function of normalized frequency. Since each segment is a realization of WGN, the ideal PSD should be equal to 1 at all frequencies. However, due to the finite length of each segment and the bias and variance of the periodogram estimator, the PSD estimates exhibit some variation from segment to segment as well as from frequency to frequency.

3.1.3 Averaged Periodogram

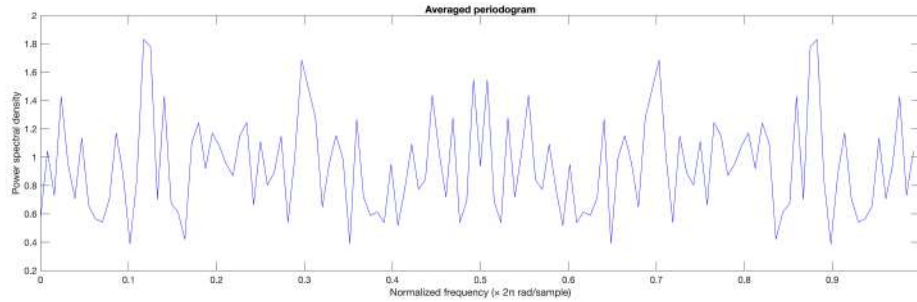


Figure 31: Averaged periodogram

Figure 31 displays the averaged periodogram obtained from averaging eight non-overlapping 128-sample segments of WGN.

As we can see from the plot, the averaged periodogram is smoother than the individual PSD estimates. The random fluctuations in the individual PSDs are smoothed out by the averaging process, resulting in a more accurate estimate of the underlying PSD.

The differences between the individual PSDs and the averaged periodogram can be attributed to the random nature of the WGN signal and the limited number of segments used for estimation. With more segments, the averaged periodogram would be even smoother and closer to the ideal PSD of 1 at all frequencies.

3.2 Spectrum of autoregressive processes

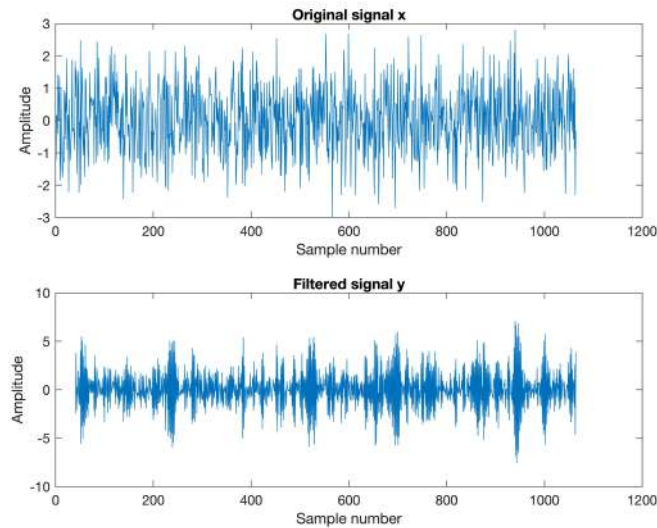


Figure 32: Signals \mathbf{x} and \mathbf{y} in time domain

Figure 32 displays how a low pass filter has attenuated the high frequency noise from the WGN sequence \mathbf{x} , providing us with signal \mathbf{y} , which is smoother and has less variation.

3.2.1 Ideal Power Spectral Density

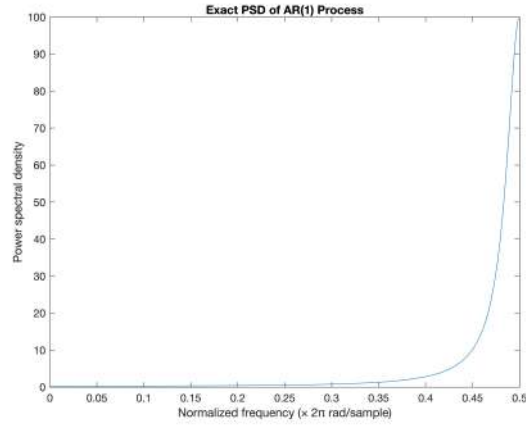


Figure 33: Ideal PSD

Figure 33 shows a bell shaped spectral estimate that is symmetric about 0.5 Hz. For clarity, the plot has only been shown up to 0.5 Hz. The cut-off frequency is around 0.38 Hz.

3.2.2 Periodogram Estimate of PSD

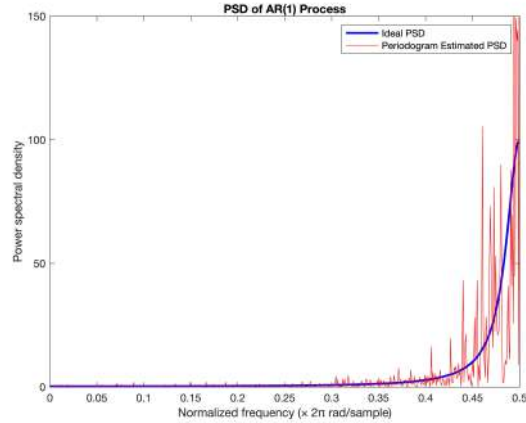


Figure 34: Ideal PSD and Periodogram Estimated PSD

The main difference observed is that the periodogram estimated PSD is much noisier when compared to the ideal PSD. The periodogram is more noisy as it is based on finite number of samples. As a result, the periodogram may have low resolution and spectral leakage.

3.2.3 Windowing within the Periodogram Estimator

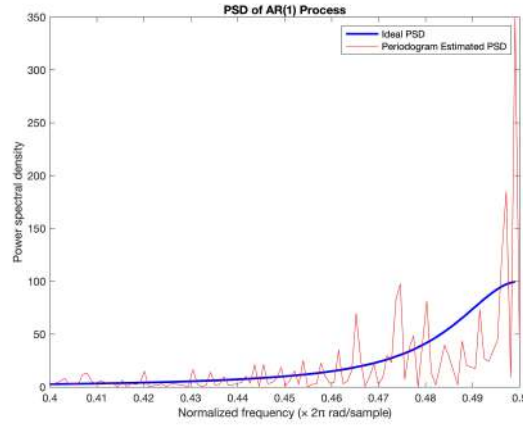


Figure 35: Zooming in on the interval $f = [0.4, 0.5]$

The effect of windowing in the periodogram estimator is largely responsible for the variations between the two plots in Figure 35 in this frequency range. The periodogram estimator's rectangle windowing function causes spectral leakage. This is especially evident when the frequency of interest is near to the window's edge, as in the interval $f = [0.4, 0.5]$. When compared to the ideal PSD, this results in more noise and lower resolution.

3.2.4 Model Based PSD

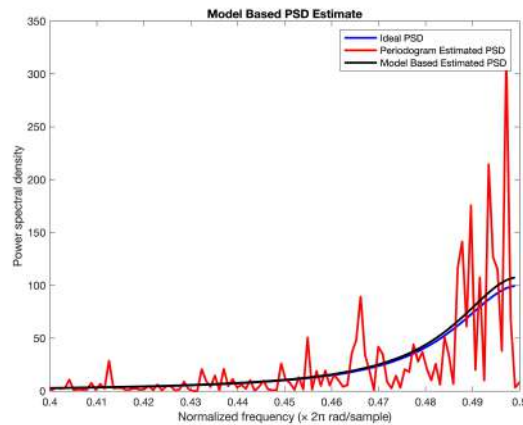


Figure 36: Model Based PSD estimate compared to Periodogram estimate

The model-based estimation assumes that the signal is generated by a linear time-invariant system and estimates the PSD based on the system's transfer function. Unlike the periodogram, it is more accurate as it does not suffer from spectral leakage and noise for finite-length signals.

3.2.5 Periodogram vs. Model Based PSD for Sunspot Time Series

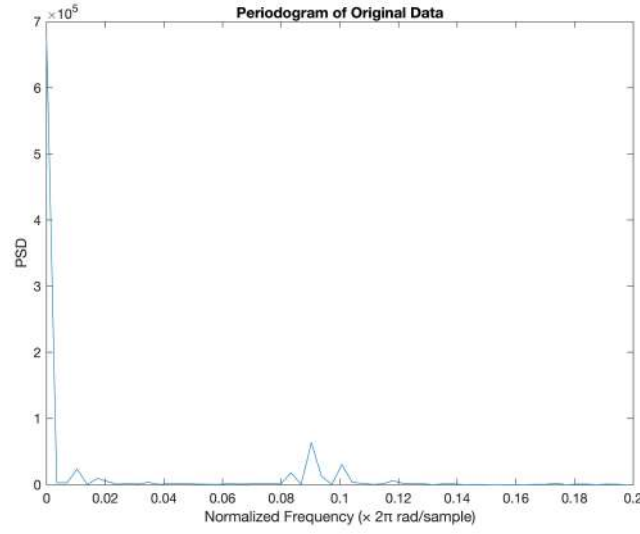


Figure 37: Periodogram of Original Sunspot Time Series Data

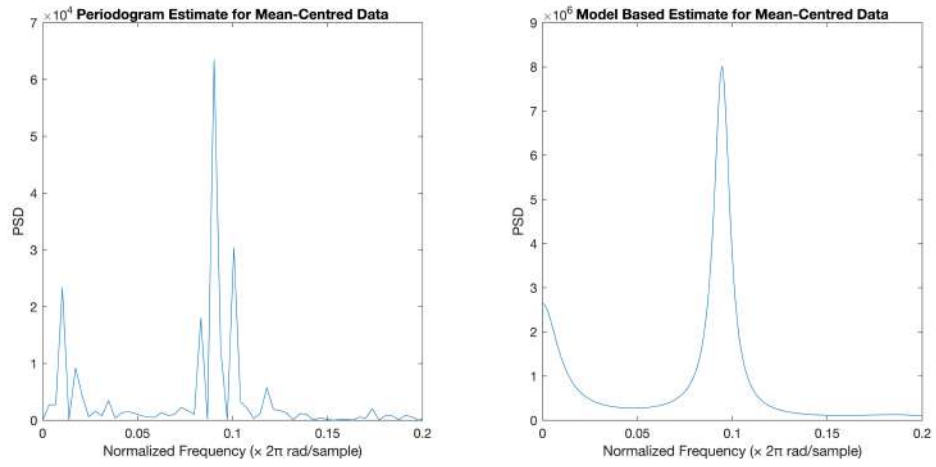


Figure 38: Periodogram vs. Model Based Estimate for mean centered sunspot data.

Figure 38 shows how the Model Based Estimate is overall, more accurate and less noisy than the Periodogram estimate, even though both plots show a peak at 0.1 Hz.

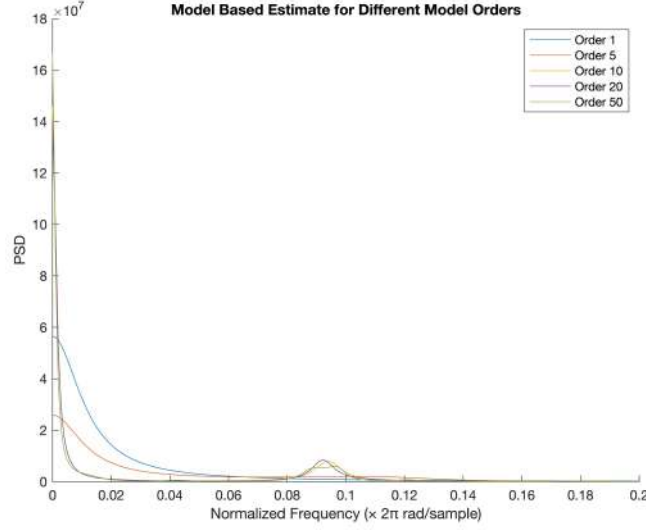


Figure 39: Periodogram vs. Model Based Estimate for mean centered sunspot data.

Figure 39 shows how both undermodelling and overmodelling can have an impact on the PSD estimate's accuracy. From section 2.3.3, we know that 2 is the optimal order to model the sunspot data. The above plot displays how order 1 under-models the data as it does not accurately reflect the ideal PSD and may have important frequency components that may be missing. On the other hand, order 10 and above, over-models the data.

3.3 The Least Squares Estimation (LSE) of AR Coefficients

3.3.1 The Cost Function

The given cost function for an AR model is:

$$J = \sum_{k=1}^M (r_{xx}[k] - \sum_{i=1}^p (a_i * \hat{r}_{xx}[k-i])), \quad \text{for } M \geq p \quad (33)$$

We define \mathbf{x} , \mathbf{H} , \mathbf{h}_i and \mathbf{a} as

$$\mathbf{x} = \begin{bmatrix} r_{xx}[p] \\ r_{xx}[p+1] \\ \vdots \\ r_{xx}[M] \end{bmatrix}, \mathbf{H} = [\mathbf{h}_1 \quad \mathbf{h}_2 \quad \cdots \quad \mathbf{h}_p], \mathbf{h}_i = \begin{bmatrix} r_{xx}[p-i] \\ r_{xx}[p+1-i] \\ \vdots \\ r_{xx}[M-i] \end{bmatrix}, \mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} \quad (34)$$

Substituting these values in Eq. 18 and simplifying gives us

$$J = (\mathbf{x} - \mathbf{H}\mathbf{a})^T (\mathbf{x} - \mathbf{H}\mathbf{a}) \quad (35)$$

To find the least squares estimate of unknown coefficients \mathbf{a} we need to begin by minimizing the cost function given in Eq. 18.

We can do this by partial differentiating the cost equation with respect to the unknown coefficients

$$\frac{\partial J}{\partial \mathbf{a}} = -2\mathbf{H}^T (\mathbf{x} - \mathbf{H}\mathbf{a}) = 0 \quad (36)$$

Rearranging for \mathbf{a} gives us the unknown coefficients shown in Eq. 22

$$\mathbf{a} = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{x} \quad (37)$$

Eq. 22 is also called the least squares modified Yule Walker equations. The LSE and the Yule-Walker estimates for the AR coefficients are similar when the data is stationary. LSE for AR coefficients is based on trying to minimize the squared error signal, while the Yule-Walker estimate based on the ACF of the signal.

3.3.2 The Nature of Observation Matrix \mathbf{H}

The observation matrix \mathbf{H} as shown in Eq. 19, is constructed using the ACF of data which is random in nature. A deterministic matrix is a matrix which contains no random values. Therefore, \mathbf{H} is considered to be a stochastic matrix.

3.3.3 LSE of a for Real Sunspot Data

The LSE approach to calculate the unknown coefficients of an AR(p) models of sunspot data was done and the coefficients are plotted in Figure 40.

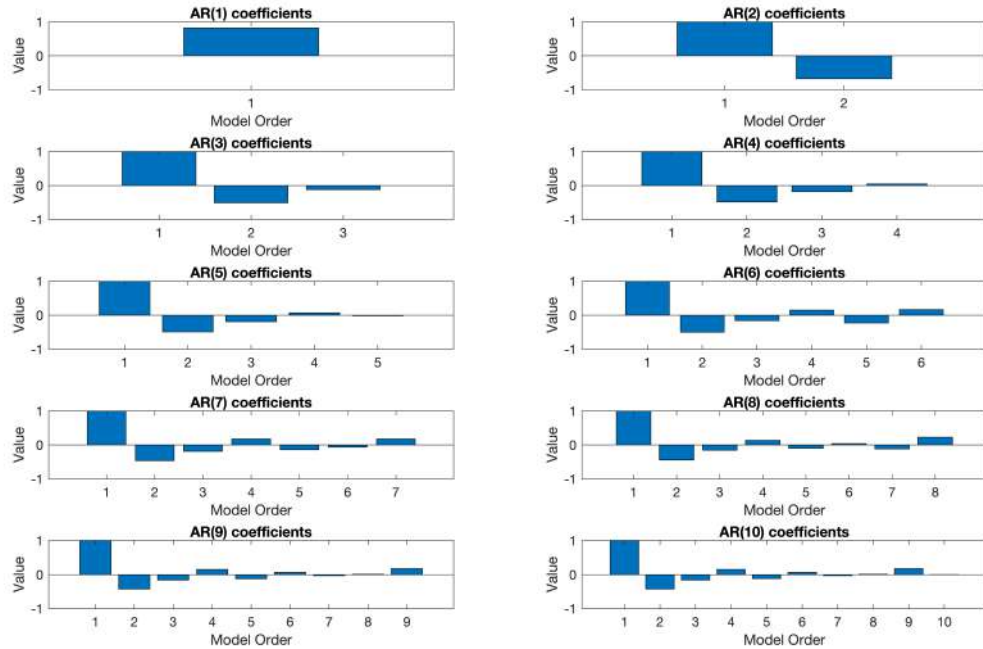


Figure 40: AR(p) coefficients using LSE approach

3.3.4 Finding the Suitable Order to Model Sunspot Data

From Figure 41, it shows that the lowest squared error is for AR(1) and AR(2) model. After model order of 2 the squared error begins to increase as we increase the model order. Thus, we can conclude that the most suitable order to model the real sunspot data is 2. This conclusion agrees with the analysis carried out in Section 2.3 question 3.

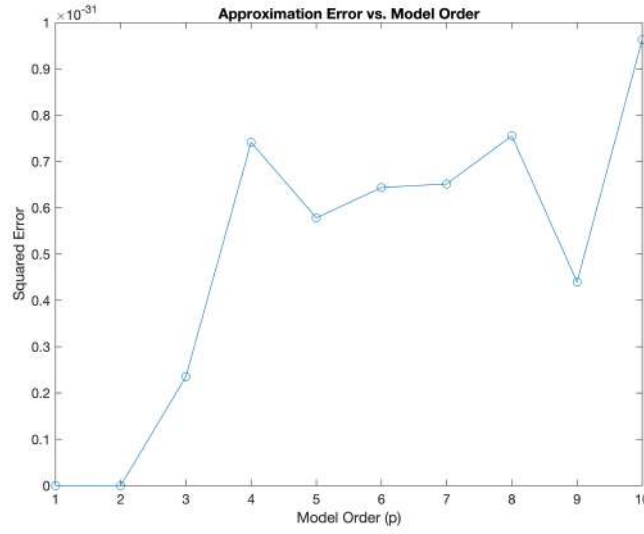


Figure 41: Squared approximation error vs. Model order for real sunspot data

3.3.5 Power Spectra Associated with the AR(p) Models of the Sunspot Time Series

The peaks shown in the power spectra for the AR(p) models show the periodicity of the sunspot time series - The sunspot time series has a 11 year cycle.

As we increase the model order, the periodicity of the data becomes increasingly apparent. Therefore, higher model order are better are capturing the data.

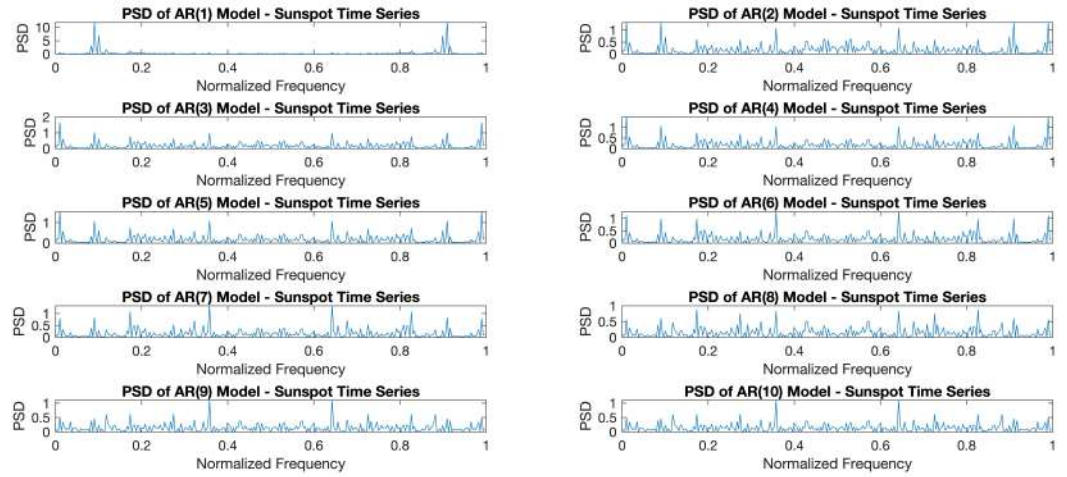


Figure 42: Power spectra for AR(p) models of sunspot data

3.3.6 Approximation error for the Optimal AR(p) Model Order

Figure 43 shows the squared error for increasing data length of the sunspot data. From the plot, it can be seen that the MSE increases as the data length increases and then it decreases again. For model order of 2, the optimum data length is the one which has the lowest MSE.

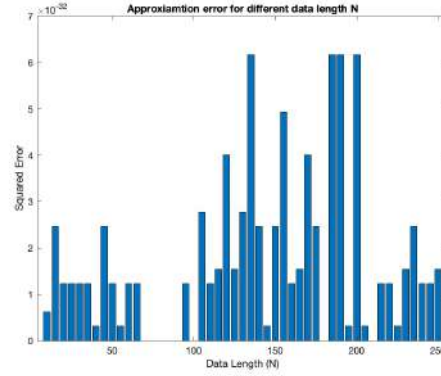


Figure 43: Approximated error for AR(2) models of sunspot data

3.4 Spectrogram for time frequency analysis: Dial tone pad

3.4.1 Computing the Discrete Time Sequence

The random London land-line number that was generated and used for the discrete time analysis was 020 4084 2872.

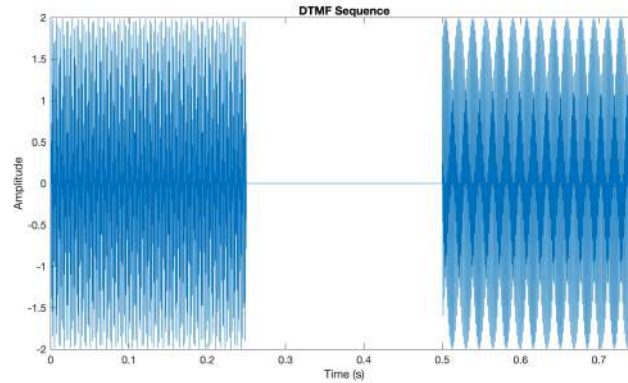


Figure 44: A two-tone signal for the digit 0 followed by idle time which is then followed by signal for digit 2

The proposed sampling rate of 32768 Hz is appropriate since it meets the Nyquist criterion of sampling rate to be at least twice the highest frequency component in the signal. In this case, the highest frequency component is 1477 Hz, so a sampling rate of 32768 Hz is more than twice the highest frequency component. The proposed sampling rate of prevents aliasing.

3.4.2 Spectrogram Analysis

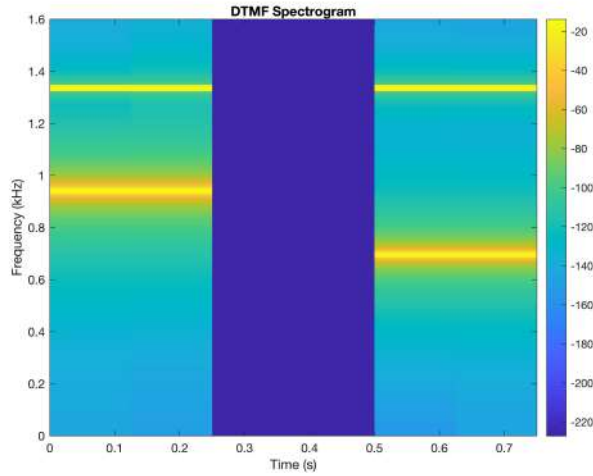


Figure 45: Spectrogram for a two-tone signal for the digit 0 followed by idle time which is then followed by signal for digit 2

The spectrogram displays the frequency content of the DTMF sequence, which can be useful for analysing and recognising individual tones.

The spectrogram clearly shows the individual tones that comprise the DTMF sequence, separated by idle time. Each tone is made up of two closely spaced sine waves, the frequencies of which correlate to the row and column of the key being pressed.

3.4.3 Performing Key Classification

Table 1 in the coursework question shows the frequencies associated with each DTMF key. We can see from the spectrogram estimates that there are distinct peaks at these frequencies for each key press, suggesting that the sequence generated by a key press can be identified.

For each DTMF key press, we can create a table that includes the associated frequencies as well as the distinct spectrogram peaks. We can then match these values with the spectrogram peaks of the sequence generated.

3.4.4 Adding Channel Noise

We introduce WGN of variances 5, 50 and 500 to sequence \mathbf{y} to display how noise affects the tone identification.

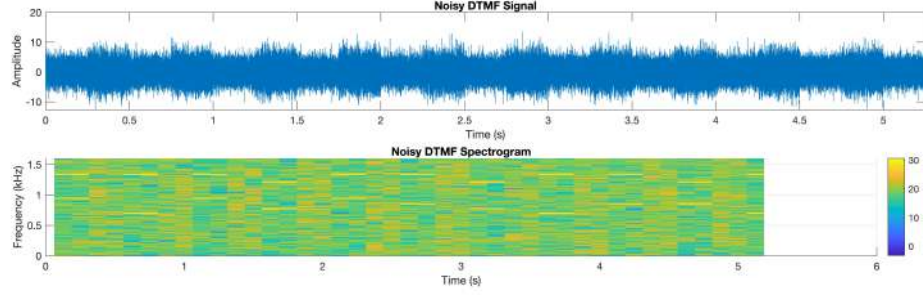


Figure 46: DTMF Sequence and Spectrogram with WGN $\sigma = 5$

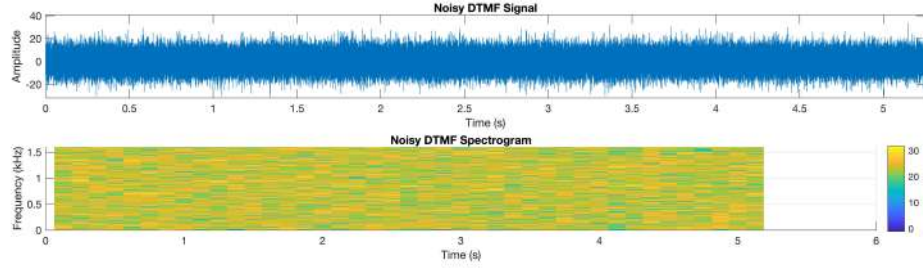


Figure 47: DTMF Sequence and Spectrogram with WGN $\sigma = 50$

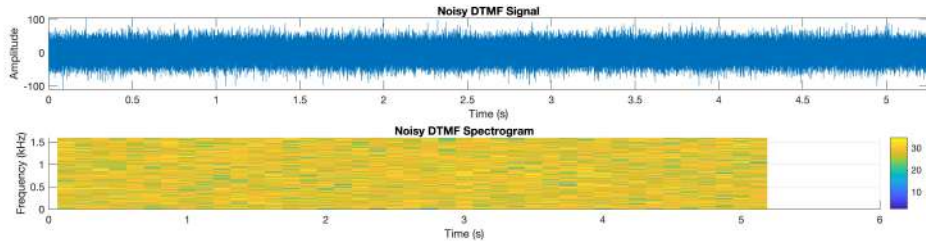
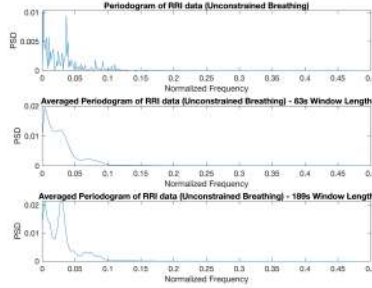


Figure 48: DTMF Sequence and Spectrogram with WGN $\sigma = 500$

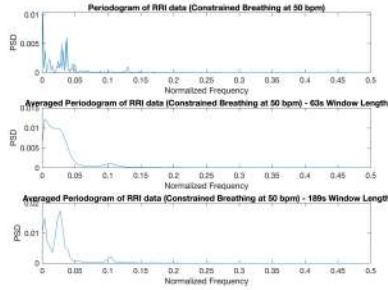
Overall, introduction of channel noise to the sequence can make it difficult to detect which keys are pressed. This can be observed from Figure 47 and 48, where it is difficult to distinguish between bands that belong to the noise frequency and bands that belong to the key pressed on the spectrogram. For low noise, as seen in Figure 46, key classification is still possible. This is because, from the spectrogram it is still possible to detect when the keys are pressed.

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

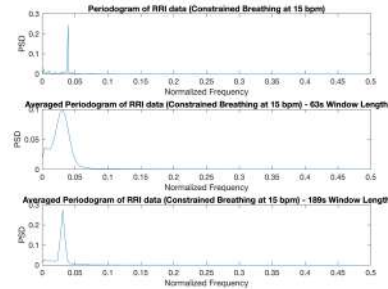
a) Figure 49 displays the standard periodogram, the average periodogram with 63 second window and the average periodogram with 189 second window for the three RRI intervals. The average periodogram for each window length was computed using Welch's PSD estimate.



(a) RRI Trial 1 periodogram



(b) RRI Trial 2 periodogram



(c) RRI Trial 3 periodogram

Figure 49: Periodograms for the 3 RRI trials

b) The main difference between the periodogram for each RRI trial is the location of the peaks which indicate the speeding up of heart rate. For trial 1, a peak is seen around the normalized frequency of 0.04. For trial 2 and 3, the peaks are visible at larger normalized frequency as well.

4 Optimal filtering - Fixed and Adaptive

4.1 Wiener Filter

4.1.1 Finding the Optimal Coefficients of the Wiener Filter

The ideal Signal-to- Noise Ratio (SNR) in dB for $z[n]$ is given by :

$$SNR(dB) = \frac{\sigma_{y[n]}^2}{\sigma_{\eta[n]}^2} = \frac{1^2}{0.1^2} = 20dB \quad (38)$$

We can find the optimal coefficients of the Wiener filter using the following relation :

$$R_{xx}^{-1} p_{zx} = w_{opt} \quad (39)$$

Without using normalized data, we obtain the optimal coefficients as $w_{opt} = [1.0054 \ 2.0012 \ 2.9932 \ 1.9943 \ 0.9986]$. These coefficients are similar to the unknown system's coefficients of $\mathbf{b} = [1 \ 2 \ 3 \ 2 \ 1]$. The SNR in this case is 33.11 dB.

4.1.2 Effects of Different Noise Powers on the Wiener Solution

| Noise Variance | SNR (dB) |
|----------------|----------|
| 0.1 | 23.22 |
| 0.25 | 18.27 |
| 1.96 | 9.28 |
| 6.25 | 4.16 |
| 9.6 | 2.11 |

The table above shows that as the noise variance increases from 0.1 to 9.6 the SNR decreases from 23.22 to 2.11. When the noise power is low (low noise variance), the wiener solution provides us with accurate coefficients.

Increasing the N_w beyond 4 would increase the complexity of the wiener solution and give us accurate coefficient values even at higher powers.

4.1.3 Computational Complexity of Wiener Solution

The computation of the autocorrelation matrix requires N_w^2 multiplications while the computation of the cross-correlation vector requires N_w multiplications.

Both autocorrelation and cross-correlation require $N(N_w - 1)$ additions.

The inversion of the autocorrelation matrix requires $O(N^3)$ operations. Finally, the computation of the estimated output signal requires N_w multiplications and $N(N_w - 1)$ additions.

Therefore, the total number of multiplications required for the Wiener solution is $N_w(N_w + 2N - 1)$ and the total number of additions is $O(N_w^3 + N^2 N_w)$.

4.2 The Least Mean Square (LMS) Algorithm

4.2.1 The LMS algorithm

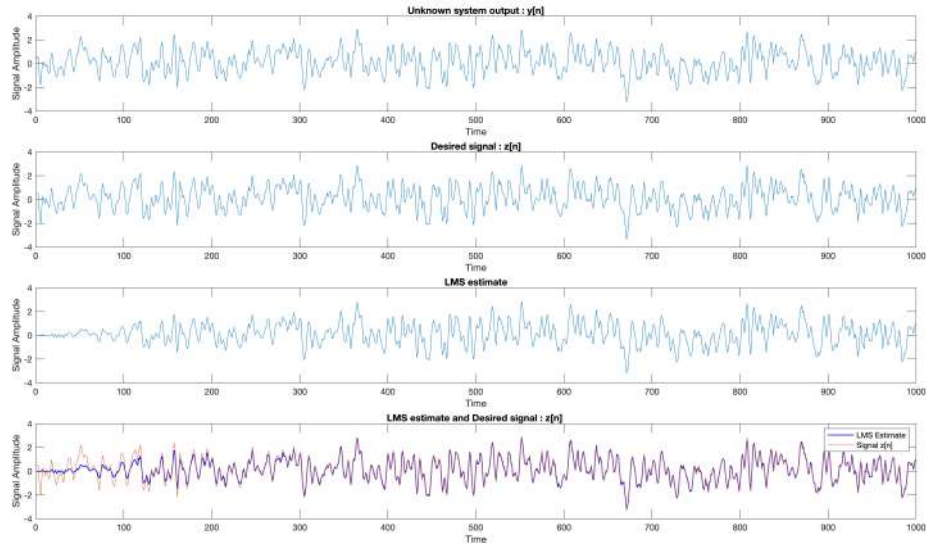


Figure 50: Comparing the LMS estimate with the desired signal $z[n]$

The third and fourth subplot in Figure 50 show the LMS estimate and how the LMS algorithm applied is able to accurately estimate the signal $z[n]$. From the figure above it can be seen that the LMS algorithm takes less than 100 seconds to estimate the desired signal accurately.

4.2.2 Time Evolution of the Coefficients

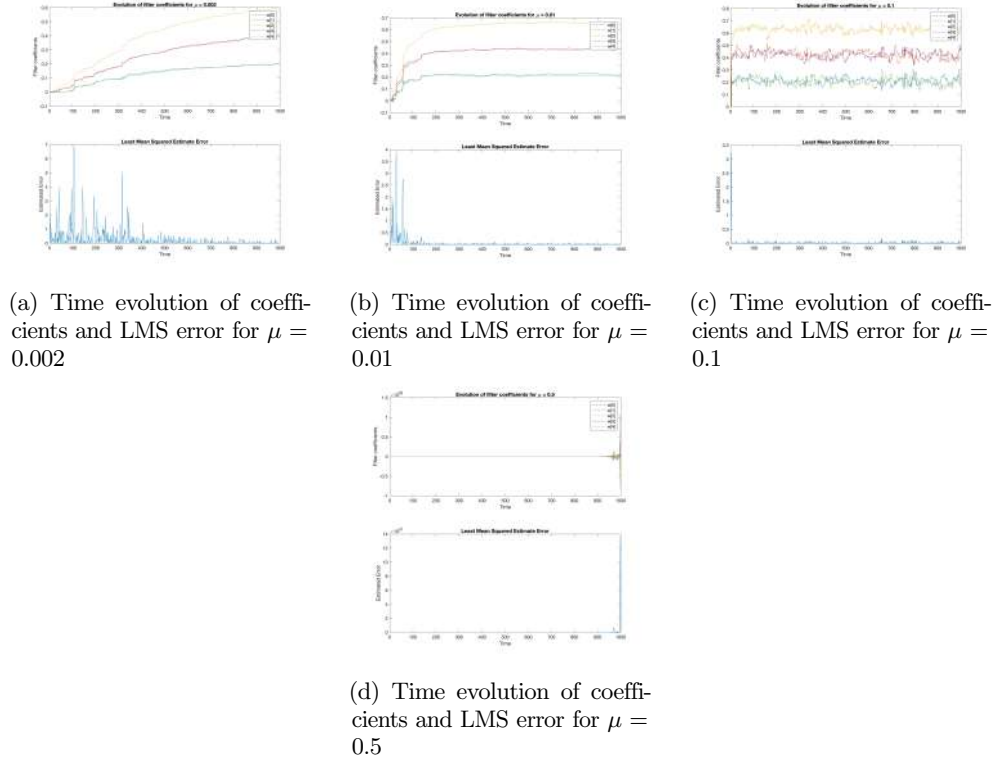


Figure 51: Time evolution of coefficients and LMS error for different adaption gain

The time evolution of coefficients plots above demonstrates that as μ rises, the filter coefficients converge to a steady-state value, and that the convergence is quicker at greater μ values. However, it can be seen that increasing μ beyond a certain value can cause the coefficients to diverge. Furthermore, as μ increases, the filter coefficients become more susceptible to noise. The LMS error plot demonstrates that as μ rises, the squared error decreases, suggesting that higher μ values result in improved filter performance.

4.2.3 Computational Complexity of LMS Algorithm

The number of multiplications and additions needed at each step of the algorithm determines the computational complexity of the LMS algorithm.

Overall, the total number of operations required at each iteration is:

$(2N_w + 6)$ multiplications and $(2N_w + 6)$ additions. Each iteration consists of 4 main steps:

1. Calculating filter output
2. Calculating error signal
3. Calculating the adaption gain
4. Calculating updated filter coefficients.

As the LMS algorithm iterates N times for a signal of length N , the LMS algorithm's overall processing complexity is:

$N(2N_w + 6)$ multiplications and $N(2N_w + 6)$

4.3 Gear Shifting

Figure 52 and 53 depict faster convergence of coefficients and lower LMS error for $\mu = 0.01$ compared to the standard LMS algorithm.

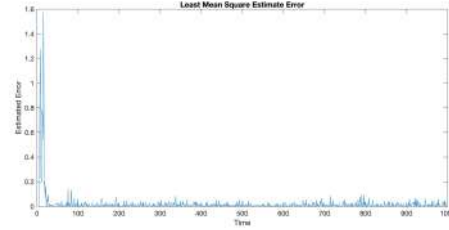


Figure 52: LMS error for $\mu = 0.01$

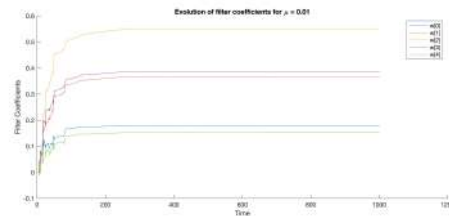


Figure 53: Time evolution of coefficients for $\mu = 0.01$

4.4 Identification of AR Processes

4.4.1 Estimating a_1 and a_2 with LMS algorithm

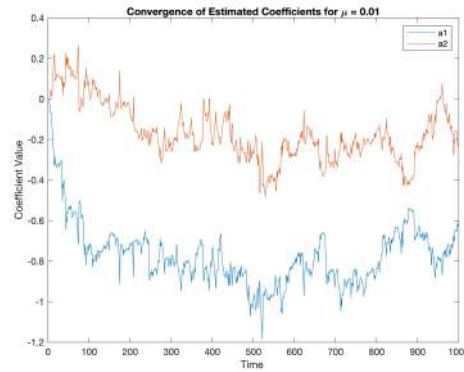


Figure 54: AR coefficients for $\mu = 0.01$

If the adaptive LMS algorithm is applied efficiently, the values of a_1 and a_2 converge to the initial AR model parameters. Figure 54 shows that the coefficients converge to $a_1 = -0.2$ and $a_2 = -0.9$.

4.4.2 Evolution of a_1 and a_2 for Different Adaptation Gains

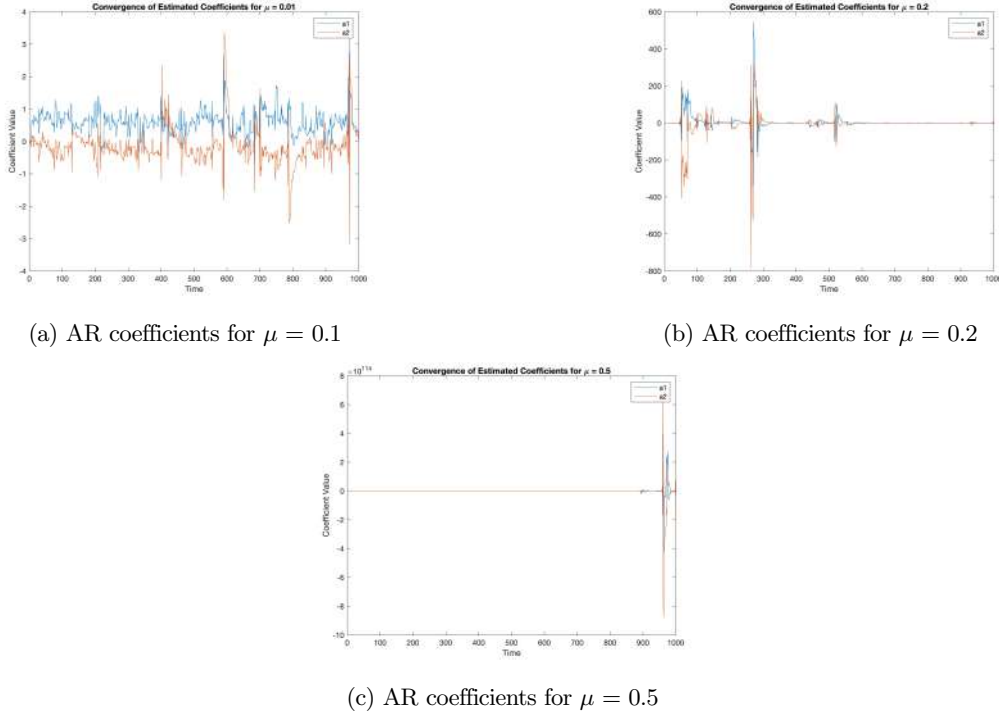


Figure 55: Convergence of AR coefficients for different adaption gains.

From the above figure as well as Figure 54 in previous section it can be seen that for low μ the convergence of the coefficients is slow. However, increasing the μ leads to a distorted coefficient estimate that begins to diverge away from the ideal AR coefficients.

4.5 Speech Recognition

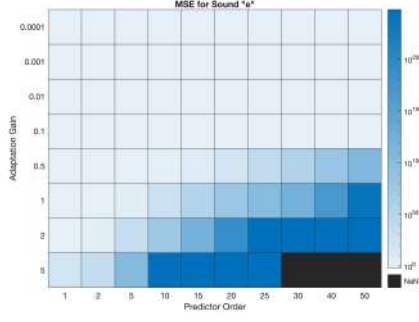
4.5.1 Testing the Performance of Predictor on Real World Audio Signals

A sampling frequency of $f_s = 44100\text{Hz}$ and $N = 1000$ samples was used on real world audio signals. These real world audio signals were used as an input into the adaptive LMS algorithm created in section 4.4.1.

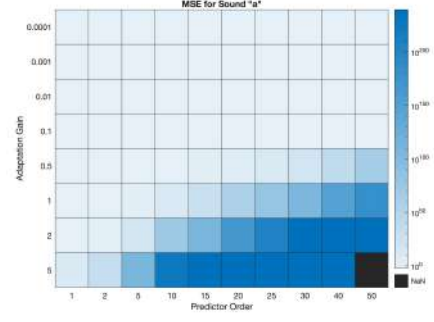
A heatmap of the mean squared error is produced to show the optimal selection of prediction order and adaption gain for each audio signal - "e", "a", "s", "t", and "x".

The LMS algorithm was edited to also output the best adaption gain and prediction order that leads to the lowest MSE overall.

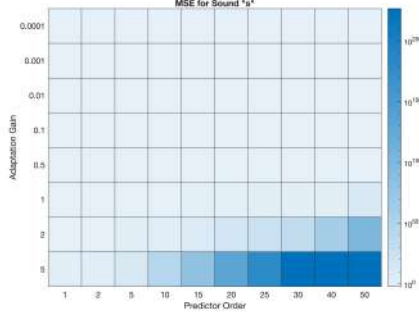
Moreover, gear shifting cannot be applied to these signals as real world signals are non-stationary in nature.



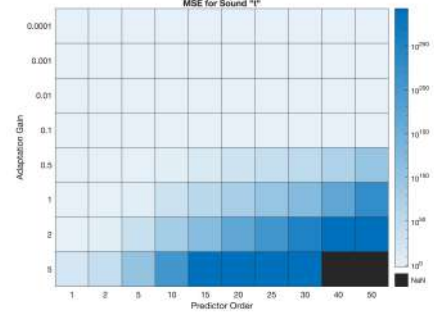
(a) MSE for sound "e"



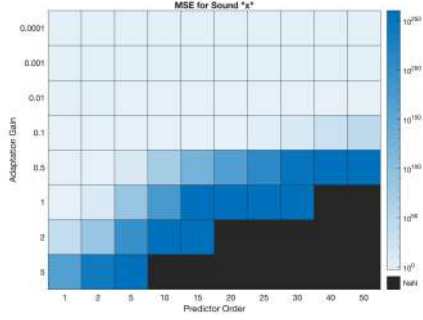
(b) MSE for sound "a"



(c) MSE for sound "s"



(d) MSE for sound "t"



(e) MSE for sound "x"

Figure 56: MSE for audio signals

The LMS algorithm was edited to also output the best adaption gain and prediction order that leads to the lowest MSE overall. These values are provided in the table below.

| Audio Signal | Adaption Gain | Prediction Order |
|--------------|---------------|------------------|
| e | 0.5 | 5 |
| a | 0.5 | 5 |
| s | 0.1 | 40 |
| t | 0.5 | 5 |
| x | 1.0 | 1 |

4.5.2 Finding Optimal Filter Length

The efficiency of adaptive filters depends significantly on determining the optimal filter length. An optimal filter length allows us to obtain an accurate prediction with minimum computational complexity.

One heuristic approach to find the optimal filter length is by the trial and error method, where we can choose the best combination of adaption gain and prediction order that gives the maximum R_p value or prediction gain value.

The calculation of prediction gain involves using the variance of input signal and the variance of error signal that we can obtain as an output by using the LMS algorithm.

One analytical approach to find the optimal filter length is looking at the eigenvalue spread for each adaption gain and prediction order. The eigenvalue spread helps determine the performance of the adaptive filter.

A large eigenvalue spread would mean that a larger filter length is required.¹

4.5.3 Assessing the Performance of the Predictor

Prediction gains of the predictor for 1000-sample recording of your own speech, as in Part 4.5.1 were calculated, but at a sampling frequency of $f_s = 16000$ Hz. The amplitude of the audio signal was normalized so that variations in amplitude would not impact the LMS algorithm.

The best adaption gain and prediction order were obtained for each signal and the prediction gain was calculated as follows

| Audio Signal | e | a | s | t | x |
|---------------------------|--------|---------|---------|---------|---------|
| Prediction Gain at 16 kHz | 2.3432 | 5.57838 | 3.52214 | 3.73286 | 19.9978 |

A higher prediction gain indicates a better performance of the predictor.

From the table above it can be observed that for different sounds, the prediction gain differs as each audio signal has different underlying characteristics making some signals easier to predict than the others.

I was able to obtain prediction gainsthat were comparable to those computed at 16 kHz by sampling my audio data at 44100 Hz. This shows that the effectiveness of the LMS predictor was not significantly different, indicating that the sampling frequency had minimal effect on the fundamental properties of my audio data.

Vowel sounds are considered quasi-stationary as they are stationary for a fixed period of time. Increasing the sampling frequency means more samples are obtained in a short period of time. Therefore, at a higher sampling frequency allows us to obtain are more detailed vowel audio signal which further allows the predictor to estimate the signal accurately.

The convergence of the learning curve is also impacted by the number of samples. More samples mean more data for the predictor to learn from, which leads to improved convergence. However, this can also increase the computational complexity.

¹Haykin, Simon S., and T. Prabhakar. Adaptive Filter Theory. Fifth edition, International edition. Upper Saddle River: Pearson, 2014. Print.

4.6 Dealing with Computational Complexity: Sign Algorithms

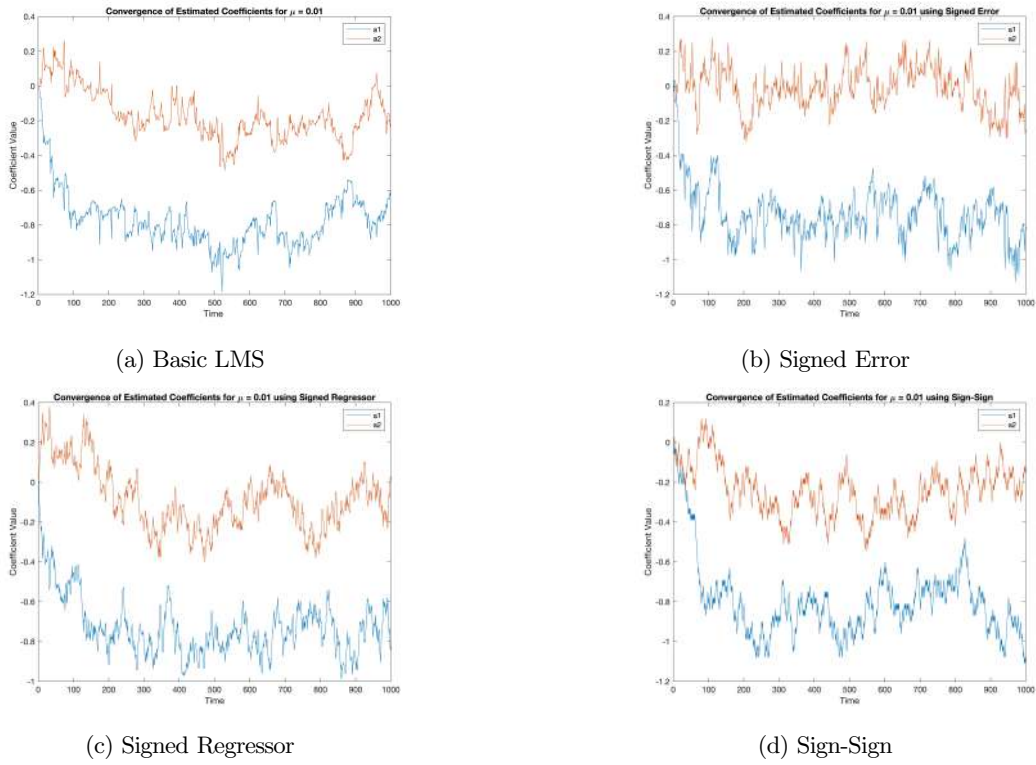


Figure 57: Sign Algorithm Plots

From the above figures we can observe that the sign LMS algorithm converge to the values of -0.2 and -0.9 faster than the basic LMS algorithm. However the sign algorithms show greater variance than the basic LMS algorithm and so, are less accurate.

Figure 58 shows the comparison of the basic LMS algorithm with the sign LMS algorithms for audio signal for "e". The prediction order used for 2 and the adaption gain was 0.01. It can be observed that for the audio signal the sign LMS algorithm converge quicker than basic LMS algorithm. The basic, signed error and signed regressor LMS converge to similar coefficients value, while the sign-sign LMS is seen to converge to 0 coefficients value.

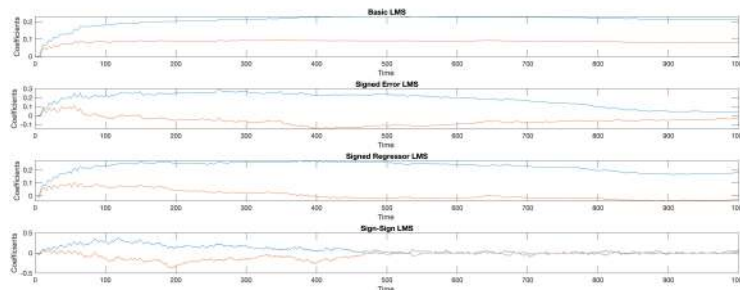


Figure 58: Performance of the sign algorithm for audio signal "e"