

IMPERIAL COLLEGE LONDON

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

Advanced Signal Processing Report

Author: Patrick McCarthy

CID: 01353165

Lecturer: Professor Danilo Mandic

Submitted in partial fulfillment of the requirements for the degree of
MEng in Biomedical Engineering of Imperial College London

April 12, 2020

Contents

1 Random Signals and Stochastic Processes	1
1.1 Statistical Estimation	1
1.2 Stochastic processes	4
1.3 Estimation of probability distributions	7
2 Linear Stochastic Modelling	9
2.1 ACF of uncorrelated and correlated sequences	9
2.2 Cross-correlation function	10
2.3 Autoregressive modelling	11
2.4 Cramer-Rao Lower Bound	15
2.5 Real world signals: ECG from iAmp experiment	17
3 Spectral Estimation and Modelling	20
3.1 Averaged periodogram estimates	20
3.2 Spectrum of autoregressive processes	22
3.3 The Least Squares Estimation (LSE) of AR Coefficients	24
3.4 Spectrogram for time-frequency analysis: dial tone pad	25
3.5 Real world signals: Respiratory sinus arrhythmia from RR-Intervals	27
4 Fixed and Adaptive Optimal Filtering	29
4.1 Wiener filter	29
4.2 The least mean square (LMS) algorithm	30
4.3 Gear shifting	31
4.4 Identification of AR processes	32
4.5 Speech recognition	33
5 MLE for the Frequency of a Signal	34
6 Nomenclature	35

1 Random Signals and Stochastic Processes

1.1 Statistical Estimation

A 1000-sample vector \mathbf{x} where each sample was a realisation of a uniform random variable $X \sim U[0, 1]$ was created and plotted, obtaining the plot in Figure 1.

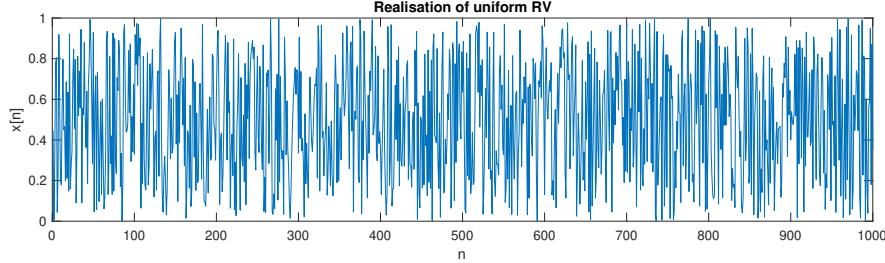


Figure 1: 1000-sample realisation of uniformly distributed random process \mathbf{x}

1.1.1 Mean

The theoretical mean m can be calculated using using Equation 1.

$$m = \int_{-\infty}^{\infty} xp(x)dx \quad (1)$$

Since $p(x) \sim U[0, 1]$ is a uniform normal distribution, x has a constant value of 1 in limits $[0,1]$ and 0 elsewhere. With this insight, the following result is obtained:

$$m = \int_0^1 x dx = 0.5 \quad (2)$$

Using MATLAB's `mean()` function, the sample mean \hat{m} was calculated with slight variation each time. Table 1 shows the results obtained for 5 instances of this random process.

Sample mean	Error
0.5045	-0.0045
0.4971	0.0029
0.5038	-0.0038
0.4977	0.0023
0.4928	0.0058

Table 1: Sample means and error.

It is clear that the sample mean \hat{m} is an accurate estimator for the theoretical/true mean m up to 2 decimal places.

1.1.2 Standard Deviation

A similar process can used to compare the theoretical and sample SD. The theoretical SD σ can be calculated using Equation (3).

$$\sigma = \sqrt{E(X^2) - (E(X))^2} \quad (3)$$

$$= \sqrt{\int_0^1 x^2 dx - (\frac{1}{2})^2} \quad (4)$$

$$= \sqrt{\frac{1}{3} - \frac{1}{4}} \quad (5)$$

$$= \sqrt{\frac{1}{12}} \quad (6)$$

$$= 0.2887 \quad (7)$$

Using MATLAB's `std()` function, the sample SD $\hat{\sigma}$ was calculated with slight variation each time. Table 2 shows the results obtained for 5 instances of this random process.

Sample SD	Error
0.2834	0.0053
0.2845	0.0041
0.2908	-0.0021
0.2840	0.0046
0.2909	-0.0023

Table 2: Sample SDs and error.

Like the mean, it is clear that the sample SD $\hat{\sigma}$ is an accurate estimator for the theoretical/true SD σ up to 2 decimal places. As the number of samples is increased, the sample statistics will converge to the theoretical statistics.

1.1.3 Bias

An ensemble of 10 realisations of \mathbf{x} was generated and then the procedure outlined above was repeated to calculate the sample means and SDs for each realisation. Plots illustrating the theoretical values in red are shown in Figure 2.

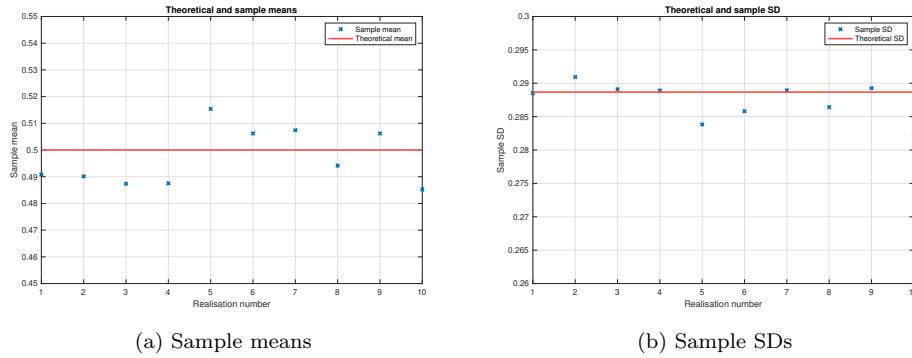


Figure 2: Sample statistics for $\mathbf{x}_{1:10}$ with uniform RV

The sample statistics clearly have a tendency to cluster around the theoretical statistics, implying that the sample mean \hat{m} and SD $\hat{\sigma}$ are both unbiased estimators for the theoretical statistics. However, the sample size of 10 is too small to make any meaningful inferences, and this would need to be repeated with more samples to be sure.

1.1.4 PDF Estimation

The histogram of a realisation of x acts as an approximation of the theoretical pdf since the y-axis represents frequency and therefore when normalised by total number of counts is analogous to probability.

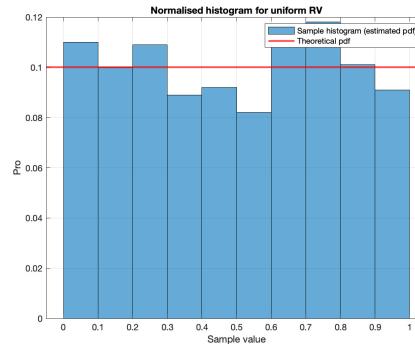


Figure 3: 1000-sample realisation of uniformly distributed random process $X \sim U[0, 1]$

As the number of samples is increased, the accuracy of the approximation improves, and the histogram tends towards the uniform distribution, which has been overlaid in red. Since the total theoretical probability is 1 and there are 10

bins, the value of each bin must tend to 0.1, such that the area under the distribution is 1. If the number of bins is increased, the value that each tends towards changes according to the relationship given by Equation 8.

$$P_{bin} = \frac{1}{n_{bins}} \quad (8)$$

1.1.5 Using a Gaussian random variable

This analysis was repeated drawing the 1000 samples of the random variable from a Gaussian distribution, i.e. $X \sim N[0, 1]$.

By definition, the theoretical mean m of the standard normal distribution is 0. It was found that the sample mean \hat{m} is an accurate estimator of the theoretical mean m to 1 decimal place.

By definition, the theoretical SD σ of a standard normal distribution is 1. It was found that the sample SD $\hat{\sigma}$ is, on average, an accurate estimator of the theoretical SD σ to 1 decimal place.

The bias of the estimated values for mean and SD can be estimated using the same approach as before, as demonstrated by Figure 4.

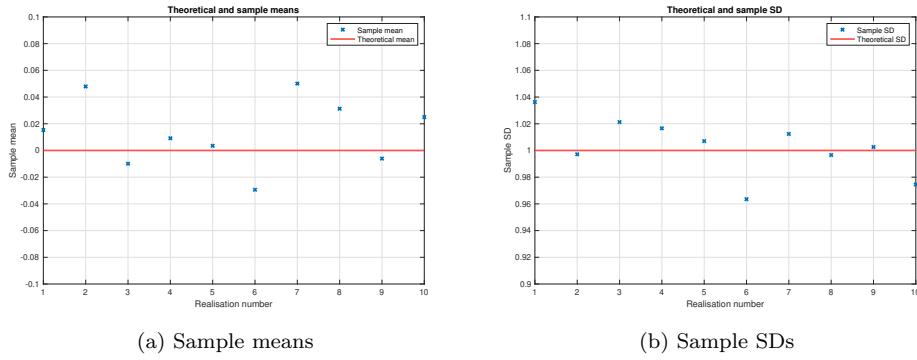


Figure 4: Sample statistics for $x_{1:10}$ with Gaussian RV

There is no clear bias in the estimate for mean or SD in this case. However, it is noteworthy that the sample mean \hat{m} appears to be a less accurate estimator for this Gaussian normal distribution than for the uniform distribution, with a maximum error of ≈ 0.01 for the uniformly distributed process but ≈ 0.05 for this Gaussian process.

An estimate of the PDF can be obtained from the normalised histogram in the same manner as before and is displayed in Figure 5. (This time, the `bar()` function was used rather than the `histogram()` function, hence the slightly different appearance).

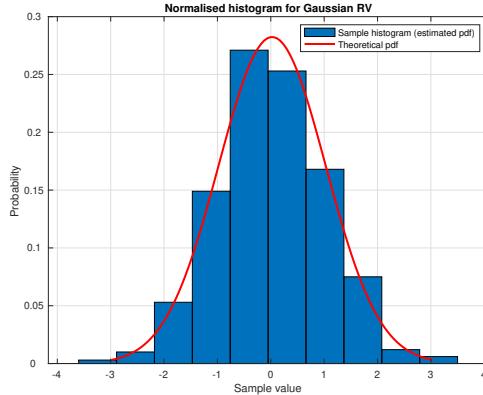


Figure 5: 1000-sample realisation of Gaussian distributed random process $x \sim N[0, 1]$

As before, as the number of samples is increased, the accuracy of the approximation improves, and the histogram tends towards the Gaussian standard normal distribution, which has been overlaid in red.

1.2 Stochastic processes

Stochasticity, stationarity and ergodicity are three of the most important concepts in statistical signal processing, and are therefore defined below for clarity.

- A *stochastic* process is as a collection of random variables.
- A *stationary* process is a random process whose statistical properties are constant in time.
- An *ergodic* process is one in which the time average is equal to the ensemble average. The same result is obtained for statistical properties of one process realisation in time as from all realisations at a single time instant.

We were given 3 random processes to analysed, which I will refer to as 'RP1', 'RP2', and 'RP3'. Ensembles of 100 realisations of these random processes, each of length 100 samples were generated.

1.2.1 Stationarity

To analyse stationarity, ensemble means and SDs of each process were plotted against time, as shown in Figure 8.

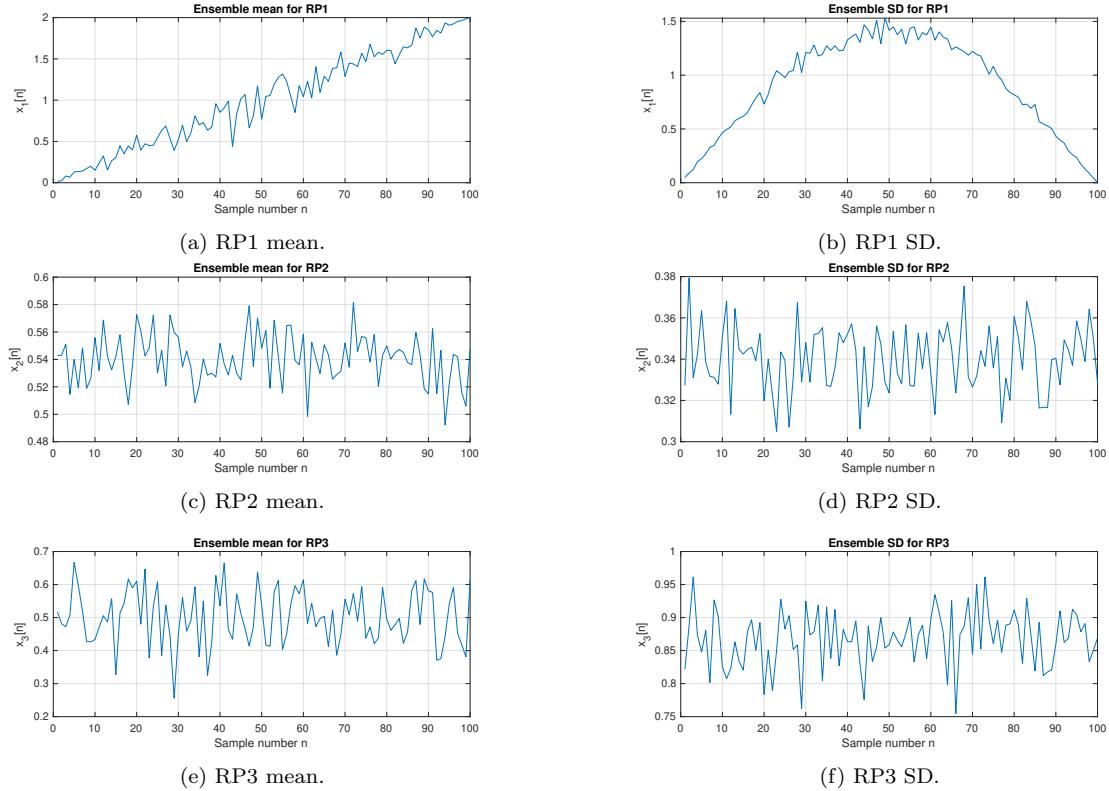


Figure 6: Ensemble means and SDs for random processes

It is immediately obvious that RP1 is non-stationary, since the ensemble mean $m_1[n]$ linearly increases from a value of 0 to a value of 2 and can be approximated by Equation 9.

$$m_1[n] = \frac{n}{50} \quad (9)$$

The ensemble SD $\sigma_1[n]$ also changes and can be approximated by Equation 10.

$$\sigma_1[n] = 1.4\sin\left(\frac{n\pi}{N}\right) \quad (10)$$

where N is the total number of samples (in this case N = 100).

It is also clear that RP2 and RP3 are stationary, since there is no clear change in the mean and SD of either process. RP2 appears to have a mean m_2 of 0.54 and a SD σ_2 of 0.34. RP3 appears to have a mean m_3 of 0.50 and a SD σ_3 of 0.88.

1.2.2 Ergodicity for M=4, N=1000

1000-sample realisations of the three random processes are created and the mean and SD are calculated for each realisation and displayed in Table 3.

Process	Realisation	Mean	Standard Deviation
RP1	1	9.2900	5.9025
	2	9.9936	5.8944
	3	9.9995	5.8690
	4	10.0452	5.8515
RP2	1	0.5756	0.0809
	2	0.4340	0.1515
	3	0.9109	0.1420
	4	0.5539	0.0444
RP3	1	0.5324	0.8535
	2	0.5171	0.8567
	3	0.4860	0.8702
	4	0.4757	0.8655

Table 3: Time means and SDs for different process realisations

The mean and SD of RP1 are roughly constant for each realisation. However, since this process is non-stationary, it cannot be ergodic since its statistical properties cannot be inferred from a single realisation.

The mean and SD of RP2 vary considerably between realisations, implying that the statistical properties of the process cannot be inferred from a single realisation, regardless of the number of samples/ signal length and therefore implying that the process is not ergodic.

The mean and SD of RP3 are roughly constant for each realisation and are also equal to the ensemble mean and SD values calculated in the previous part, implying that the process is ergodic.

1.2.3 Mathematical Description for Process Means and SD

Based on the MATLAB code provided, I was able to devise mathematical descriptions of each of the three stochastic processes.

RP1

RP1 is defined by Equation 11

$$x_1[n] = z b \times \sin\left(\frac{n\pi}{N}\right) + a n \quad (11)$$

where $z \sim U(-0.5, 0.5)$, $b = 5$, $a = 0.02$ and N = total number of samples. Since $z \sim U(-0.5, 0.5)$, $E(z) = 0$ and $var(z) = \frac{1}{12}$.

The theoretical mean can be calculated by applying the expectation operator, as below.

$$E(x_1[n]) = E(ub \sin\left(\frac{n\pi}{N}\right) + a n) \quad (12)$$

$$= E(z) \times E(b \times \sin\left(\frac{n\pi}{N}\right)) + E(a n) \quad (13)$$

$$= E(a n) \quad (14)$$

$$= a \times E(n) \quad (15)$$

$$= a \times n \quad (16)$$

$$m_1 = \frac{n}{50} \quad (17)$$

The theoretical SD can be calculated by applying the identity in Equation 18.

$$\text{var}(y) = E(y^2) - (E(y))^2 \quad (18)$$

$$\text{var}(x_1[n]) = E(x_1[n]^2) - (E(x_1[n]))^2 \quad (19)$$

$$= E(z^2 b^2 \sin^2(\frac{n\pi}{N}) + a^2 n^2 + 2zbsin(\frac{n\pi}{N})an) - a^2 n^2 \quad (20)$$

$$= E(z^2 b^2 \sin^2(\frac{n\pi}{N})) + E(a^2 n^2) - a^2 n^2 \quad (21)$$

$$= E(z^2) b^2 \sin^2(\frac{n\pi}{N}) \quad (22)$$

$$= \frac{1}{\sqrt{12}} b^2 \sin^2(\frac{n\pi}{N}) \quad (23)$$

$$\therefore \sigma_1 = \sqrt{\text{var}(x_1[n])} \quad (24)$$

$$= \frac{1}{\sqrt{12}} b \sin(\frac{n\pi}{N}) \quad (25)$$

$$\sigma_1 = 1.44 \times \sin\left(\frac{n\pi}{N}\right) \quad (26)$$

It is clear that both the calculated mean and SD agree with the observations in the previous section.

RP2

RP2 is defined by Equation 27.

$$x_2[n] = zM_r + A_r \quad (27)$$

where $M_r, A_r \sim U[0, 1]$ and $z \sim U[-0.5, 0.5]$. Immediately, we can say $E(M_r) = E(A_r) = 0.5$ and $E(z) = 0$, whilst the SD for all variables is $\frac{1}{\sqrt{12}}$.

Again, the theoretical mean can be calculated by applying the expectation operator, as below.

$$E(x_2[n]) \quad (28)$$

$$= E(z)E(M_r) + E(A_r) \quad (29)$$

$$m_2 = 0.5 \quad (30)$$

The theoretical SD can again be calculated using the identity in Equation 18.

$$\text{var}(x_2[n]) = E(x_2[n]^2) - (E(x_2[n]))^2 \quad (31)$$

$$= E(A_r^2 + 2A_r M_r z + M_r^2 z^2) - A_r^2 \quad (32)$$

$$= E(A_r^2) + E(M_r^2)E(z^2) - A_r^2 \quad (33)$$

$$= \frac{1}{3} \times \frac{1}{36} - \frac{1}{4} = \frac{1}{9} \quad (34)$$

$$\therefore \sigma_2 = \sqrt{\frac{1}{9}} = \frac{1}{3} = 0.333 \quad (35)$$

Again, it is clear that both the calculated mean and SD agree with the observations in the previous section.

RP3

RP3 is defined by Equation 36.

$$x_3[n] = zm + a \quad (36)$$

where $m = 3$, $a = 0.5$ and $z \sim U[-0.5, 0.5]$. Therefore, clearly $E(z) = 0$. Again, the theoretical mean can be calculated by applying the expectation operator, as below.

$$E(x_3[n]) = mE(z) + a \quad (37)$$

$$m_3 = 0.5 \quad (38)$$

The theoretical SD can again be calculated using the identity in Equation 18.

$$\text{var}(x_3[n]) = E(x_3[n]^2) - (E(x_3[n]))^2 \quad (39)$$

$$= E(m^2 z^2 + a^2 + 2amz) - 0.2 \quad (40)$$

$$= 0.75 \quad (41)$$

$$\therefore \sigma_3 = 0.867 \quad (42)$$

Again, it is clear that both the calculated mean and SD agree with the observations in the previous section.

1.3 Estimation of probability distributions

1.3.1 PDF estimator function

I wrote a MATLAB function to estimate the probability distribution function of a random process from its histogram. The code used in this function is shown below.

```
function pdf_est(samples)
    histogram(samples,'BinWidth',0.05,'Normalization','pdf');grid on;
    xlabel('Random Variable');
    ylabel('Probability');
    title('PDF Estimate for RV');
end
```

Testing the code on a 100-sample stationary Gaussian random process obtains the plot shown in Figure 7. The PDF has been normalised such that its integral is 1.

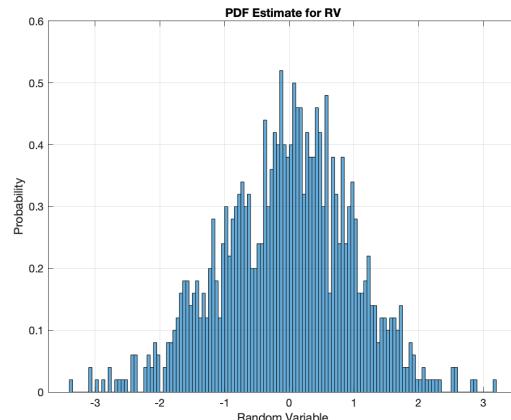


Figure 7: PDF estimator function applied to 1000-sample Gaussian process

1.3.2 Effect of data length N

The function was then tested on RP3 since it is both stationary and ergodic, obtaining the following plots for signal lengths of 100, 1000 and 10,000.

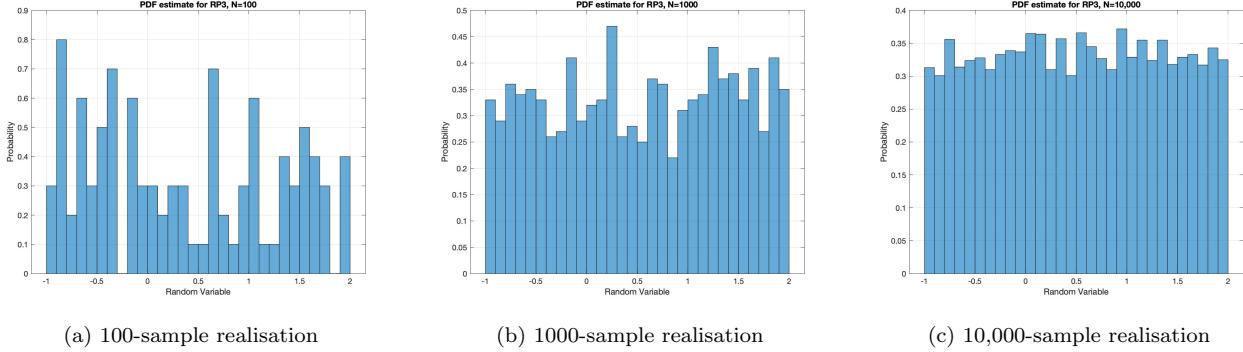


Figure 8: Estimated PDFs of RP3

It is clear that as the data length increases, the function better approximates the PDF of the process (which is a uniform distribution $U[-1,2]$ with constant value 0.333).

1.3.3 Non-stationary processes

As defined at the beginning of this section, a process is stationary if its PDF is constant in time. For this reason, the function written here could not accurately estimate the PDF of a non-stationary signal. A non-stationary process whose mean changes from 0 to 1 at $N = 500$ would essentially have two separate PDFS- one for $0 \leq N < 500$ and one for $500 < N \leq 1000$. The only way to usefully apply the function written here would be to split the signal accordingly and apply the function to the two separate PDFs.

2 Linear Stochastic Modelling

2.1 ACF of uncorrelated and correlated sequences

2.1.1 Using MATLAB `xcorr()` function

Using the `randn()` function in MATLAB, a 1000-sample WGN process was generated, and then its ACF was estimated using the `xcorr()` function. The result is displayed in Figure 9.

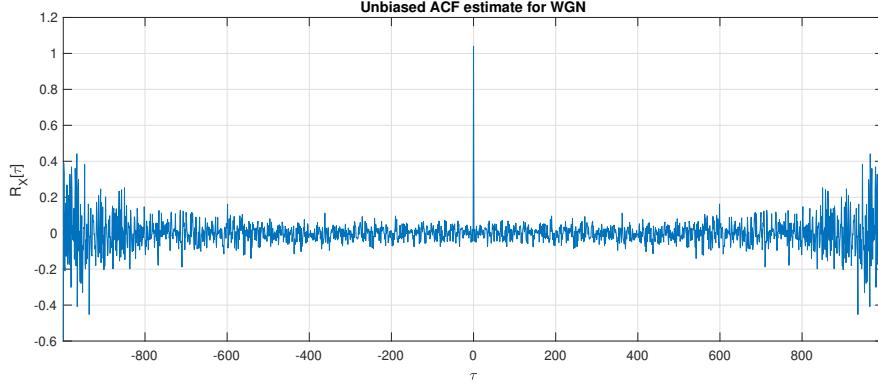


Figure 9: Unbiased estimate of ACF of 1000-sample WGN process.

The ACF estimate is notably symmetric about $\tau = 0$. The true ACF of a WGN process would be a Dirac delta function at $\tau = 0$ and zero for all other $\tau \neq 0$. Here, there is indeed a Dirac delta function at $\tau = 0$ but the signal is clearly not zero elsewhere. The ACF remains close to zero for $0 < |\tau| < 500$, but increases for $500 < |\tau| < 1000$. This can be explained by the fact that the number of samples included in the ACF calculation for comparison decreases (less than half of the signal is compared with itself), therefore the likelihood that the signal is self-similar increases as $|\tau| \rightarrow 1000$.

2.1.2 ACF for small lag τ

Using MATLAB's `zoom()` function to focus in on $|\tau| < 50$, I obtained the plot in Figure 10.

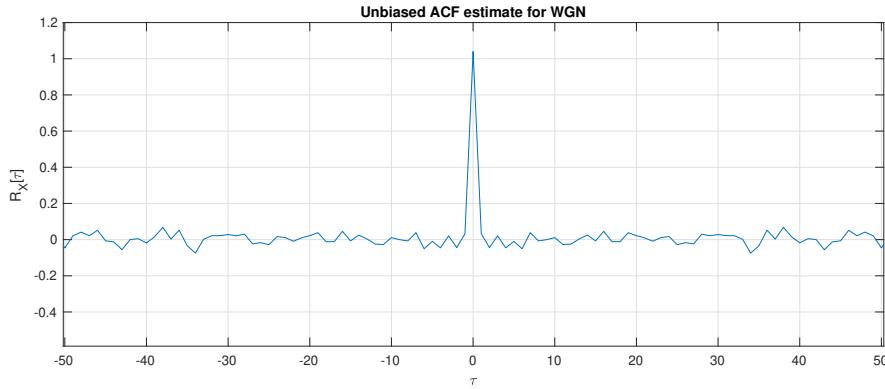


Figure 10: Estimate of ACF of 1000-sample WGN process.

Zooming in for small τ , it is clear that the ACF estimate is a very good approximation to the ideal ACF, verifying the trend identified in the previous section.

2.1.3 ACF for large lag τ

The unbiased ACF estimate is given by Equation 43.

$$\hat{R}_X(\tau) = \frac{1}{N - |\tau|} \sum_{n=0}^{N-|\tau|-1} x[n]x[n+\tau], \quad \tau = -N+1, \dots, N-1 \quad (43)$$

It is evident from this equation that the ACF estimate $\hat{R}_X(\tau)$ increases with increasing $|\tau|$ since fewer samples are included in the calculation for comparison. An empirical bound at which ACF estimates are statistically reliable would be $|\tau| < 500$. Generalising this, the bound would be $|\tau| < \frac{N}{4}$ (where N is the number of samples in the ACF, not the original signal).

2.1.4 Applying MA filter

The 1000-sample WGN signal was passed through a 9th order MA filter, as well as a 5th order MA filter for comparison. The ACFs of the resultant signals are shown in Figure 11.

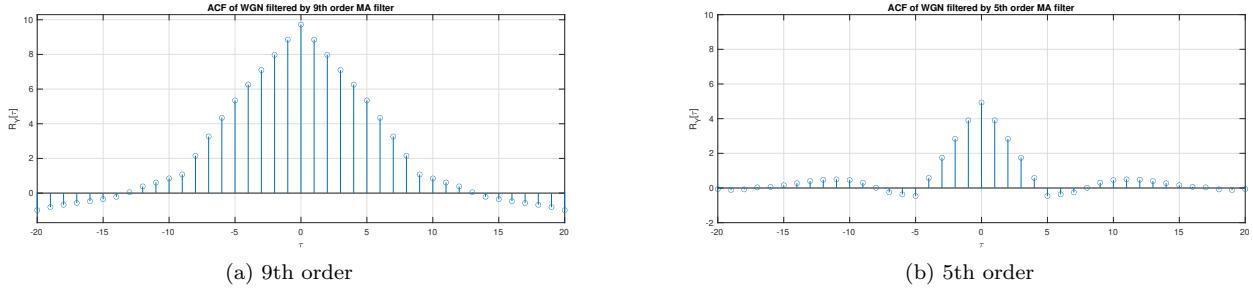


Figure 11: ACFs of MA-filtered 1000-sample WGN process.

The ideal ACF of a MA-filtered process is $R_T(\tau) = N - |\tau|$ for $|\tau| < N$ and $R_T(\tau) = 0$ otherwise (in other words, a triangular function). Figure 11(a) is indeed a triangular function with height ≈ 9 and width ≈ 18 , but we see non-zero values for $|\tau| > 9$. This is because an estimate of the ACF is being used, resulting in some error. Reducing the order of the MA filter to 5, as in 11(b), the signal clearly has more oscillation as $|\tau|$ increases. This can be explained by the way a MA filter works, defined by Equation 44.

$$y[n] = \sum_{i=0}^9 b_i x[n-i] \quad (44)$$

It works by taking the mean of the 9 previous samples and assigning this value to the current index, then shifting to the next index and repeating for all values. Through this process, an MA filter has a smoothing effect, essentially acting as a LPF. As the filter order is decreased, fewer samples are included in the averaging and therefore the signal is smoothed to a lesser extent.

2.1.5 Uncorrelated stochastic processes

It is given that \mathbf{y} is a realisation of the stochastic process Y_n (a filtered version of the uncorrelated process X_n) whose ACF $R_Y(\tau)$ is given by Equation 45.

$$R_Y(\tau) = R_X \tau * R_h(\tau) \quad (45)$$

Since X_n is uncorrelated, its ACF R_X must be an arbitrarily scaled Dirac delta-function $\alpha\delta(\tau)$ centered on zero. By the identity property of convolution, convolving a signal with a delta-function does not change the signal, therefore $R_Y(\tau) = \alpha R_h(\tau)$.

2.2 Cross-correlation function

Similarly to ACF (Equation 9), the unbiased estimate of the CCF between two signals is given by Equation 46.

$$\hat{R}_{XY}(\tau) = \frac{1}{N - |\tau|} \sum_{n=0}^{N-|\tau|-1} x[n]y[n+\tau], \quad \tau = -N+1, \dots, N-1 \quad (46)$$

The CCF was calculated for the 1000-sample WGN process used in the previous section and the 9th order MA-filtered version. The result is shown in Figure 12.

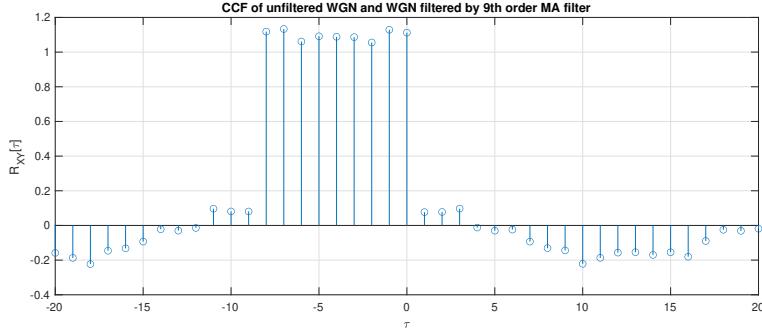


Figure 12: Estimate of CCF of 1000-sample WGN process and MA-filtered version.

Following from the result found in the previous section, the CCF is simply the impulse response of a 9th order MA filter, which is a train of 8 delta-functions.

2.2.1 Application to System Identification

Generalising the relationship found in the previous section, if an uncorrelated signal is passed through an LTI system and the CCF is taken between input and outputs, the result will simply be the impulse response of the system. This is because the output of an LTI system can be found by convolving the ACF of an input with the ACF of its impulse response. If the input is simply a Dirac-delta function, the result will be the system impulse response. From LTI systems theory, it is known that the impulse response of an LTI system provides a complete characterisation of it, therefore the system can be identified.

2.3 Autoregressive modelling

An AR process model of order 2, denoted by AR(2), is defined by Equation 47.

$$x[n] = a_1 x[n-1] + a_2 x[n-2] + w[n], \text{ where } w[n] \sim N(0, 1) \quad (47)$$

The coefficients $a_1 \in [-2.5, 2.5]$, $a_2 \in [-1.5, 1.5]$ determine the stability of the model. The region of convergence of a_1 and a_2 (in which the model is WSS) forms a triangle known as the 'stability triangle'. This is evident in Figure X, in which values of a_1 and a_2 for which the system is stable have been marked with an asterisk *.

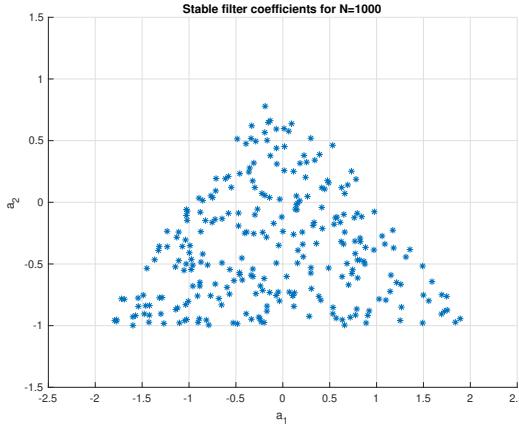


Figure 13: Stable coefficients for AR(2) process defined in Equation 47.

This triangular shape can be explained by the fact that the system poles must lie within the unit circle for stability. The characteristic equation of the system is given by Equation 48.

$$C(z) = z^2 - a_1 z - a_2 \quad (48)$$

Therefore, the system poles are given by Equation 49.

$$z = \frac{a_1 \pm \sqrt{a_1^2 + 4a_2}}{2} \quad (49)$$

Applying the condition that the system poles must be less than zero for stability, we obtain Equations 50-52, which define the triangle seen in Figure 13.

$$a_1 + a_2 < 1 \quad (50)$$

$$a_1 - a_2 < 1 \quad (51)$$

$$|a_2| < 1 \quad (52)$$

2.3.1 Sunspot Time Series

The Sunspot time series is a dataset that has been recorded for over 300 years which indexes the number of sunspots (Wolf number) observed every year. This was loaded from MATLAB and the raw data is shown in Figure 14

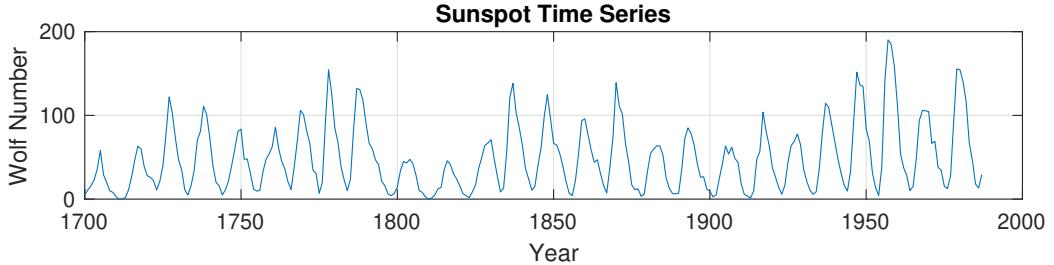


Figure 14: Raw Sunspot Time Series data.

The ACFs of the Sunspot data were calculated for both the raw Sunspot data and the standardised Sunspot data for data lengths $N = 5$, $N = 20$ and $N = 250$. These are shown in Figure 15.

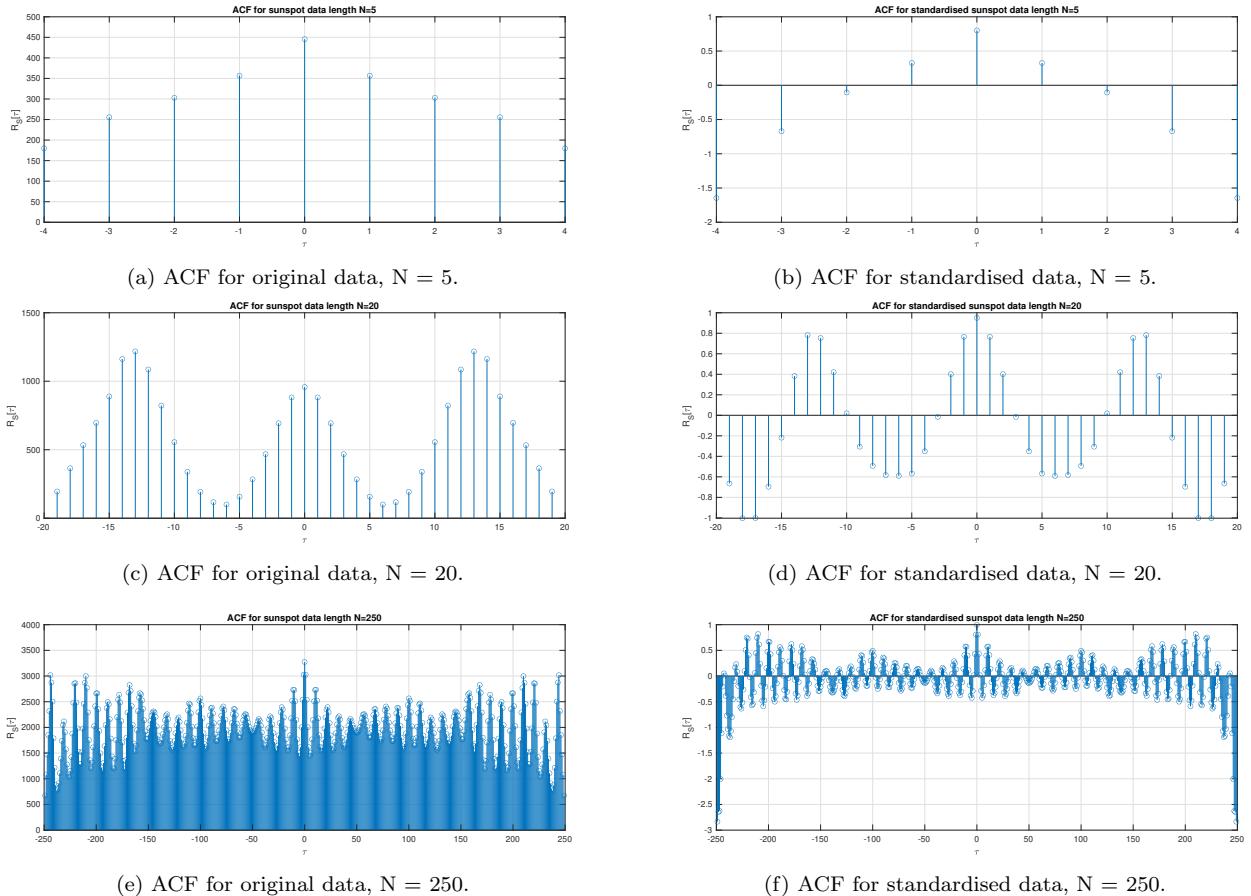
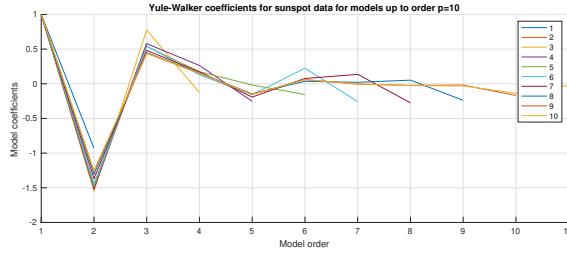


Figure 15: ACFs for original and standardised Sunspot data for varying data lengths.

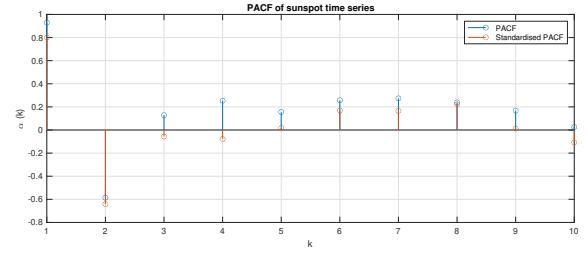
For data length $N = 5$, the ACF appears to be a triangular function. However, for data length $N = 25$, the ACF is clearly periodic, with a period of roughly 14 years. For data length $N = 250$, the ACF still appears to be sinusoidal in nature but is obscured as $|\tau|$ increases. However, within the empirical bound of statistically reliable results $|\tau| < \frac{N}{4}$ identified earlier, the ACF is more clear and appears to be the summation of a sinusoid and a triangular function. There are several irregularities in the ACFs of the original data. Notably, the maximum values are not found at $\tau = 0$. Standardising the data fixes this problem and reveals the self-similarity of the signal more clearly.

2.3.2 Yule-Walker Equations and PACFs

Using MATLAB, model coefficients were calculated for a range of model orders using Yule-Walker equations, and are shown in Figure 16(a). The PACF of the Sunspot data was also calculated for both the original and standardised values, and is shown in Figure 16(b).



(g) Yule-Walker model coefficients.



(h) PACFs for range of model orders

Figure 16: Statistics for Sunspot data.

Figure 16(a) clearly shows that the correct model order is 2, since the coefficients converge to zero for orders above this. Furthermore, the PACF values for orders above 2 are significantly lower, implying that there is little correlation between a given data sample and samples from 3 or more time-steps ago, thus confirming that the correct model order is 2.

2.3.3 Determining model order

Minimum description length (MDL) and Akaike information criterion (AIC) are useful for determining the correct model order when applied to the standardised data. As model order is increased, the loss function decreases but the complexity of the model increases so this does not necessarily imply a better model. MDL , AIC and corrected AIC (AIC_c), defined in Equations 53-55 increase in magnitude as order increases, thus their minima can be used to identify the correct model order.

$$MDL = \log E_p + \frac{p \log N}{N} \quad (53)$$

$$AIC = \log E_p + \frac{2p}{N} \quad (54)$$

$$AIC_c = AIC + \frac{2p(p+1)}{N-p-1} \quad (55)$$

where E_p = loss function, p = number of parameters/ model order and N = number of estimated data points. Plots of these statistics for a range of model orders are displayed in Figure 17.



Figure 17: Statistics for Sunspot time series models.

The global minima for both MDL and AIC_c occurs at $p = 2$, confirming that the correct model order is 2. Whilst AIC has a local minimum at $p = 2$, the global minimum is at $p = 9$. However, this is corrected for by using AIC_c so can be ignored.

2.3.4 AR Model for Sunspot Time Series

A comparison was made of AR models for a range of model orders p and prediction horizons m . The prediction horizon is defined as the number of time-steps into the future the model is used to predict. Results of this comparison are shown in Figure 18.

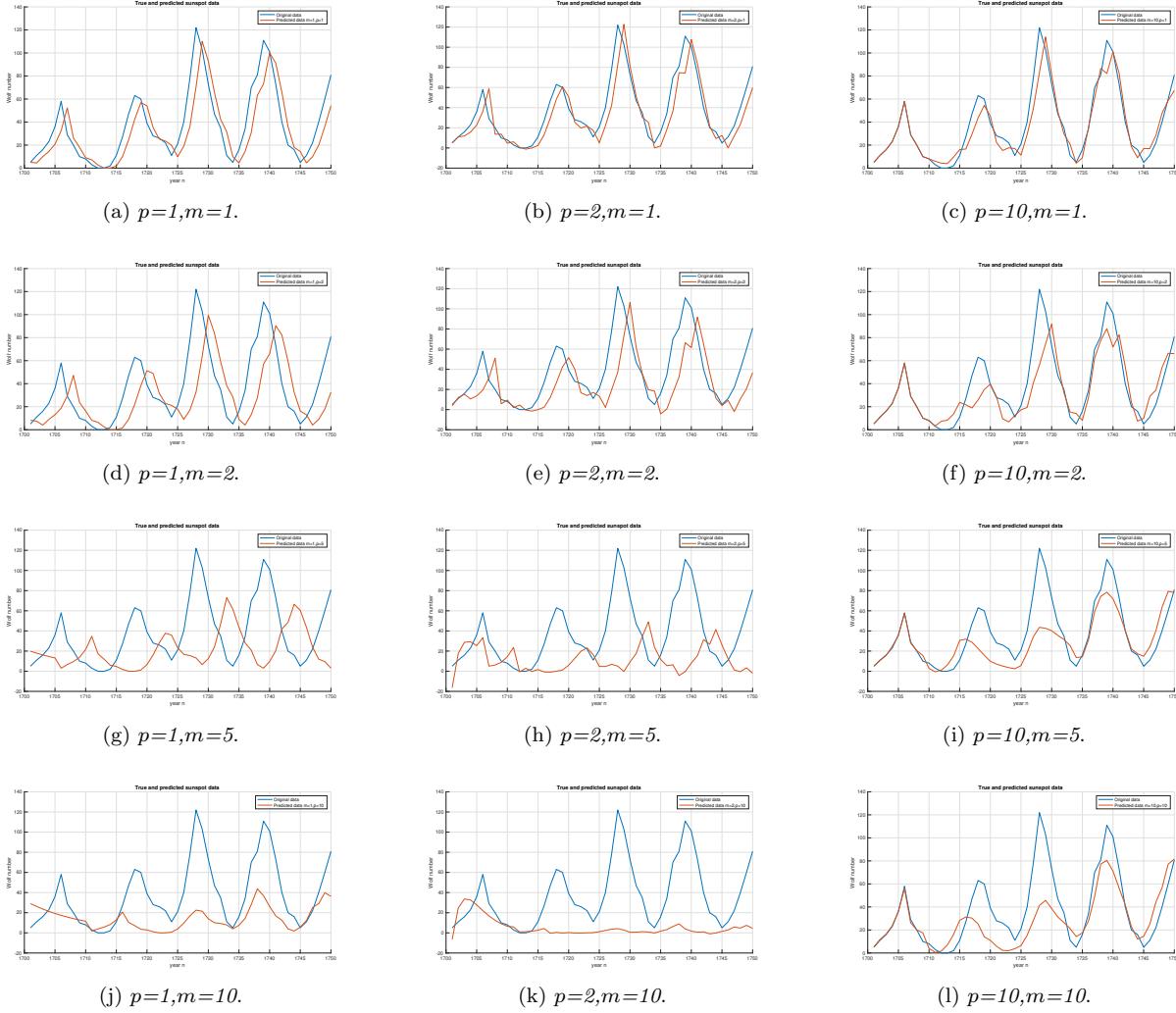


Figure 18: True and predicted data for a range of model orders p and prediction horizons m .

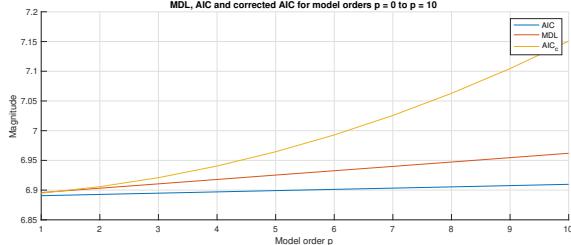
It is clear from this comparison that increasing the model order increases the temporal precision of the prediction as the time shift between the real and predicted data is significantly less. Increasing the prediction horizon significantly decreases the amplitude of the prediction signal, especially for low order models. This implies that the lower order models AR(1) and AR(2) are not capable of making predictions relatively far in the future (i.e. for a large prediction horizon). This comparison would imply that an AR(10) model is capable of making such predictions accurately, but the computational complexity of the model increases significantly with order and this apparent success in prediction may be an example of model overmodelling. Overmodelling is when more parameters (in this case AR model coefficients) are used for a model than truly describe the process. This results in very low model error, and therefore an apparently accurate model, but will likely result in high prediction error when the model is used for extrapolation. On the contrary, if a model order lower than the true order is chosen, the model will have both high model error and high prediction error, since it cannot make accurate predictions. Minimum prediction error will be found for the true model order, whereas model error will continue to decrease as the model order is increased. This highlights the importance of not using model error as a strict measure of model accuracy, but merely as a rough guide. Testing a model on new data is always necessary in order to build an accurate model, and prediction error should be the primary measure of model accuracy.

2.4 Cramer-Rao Lower Bound

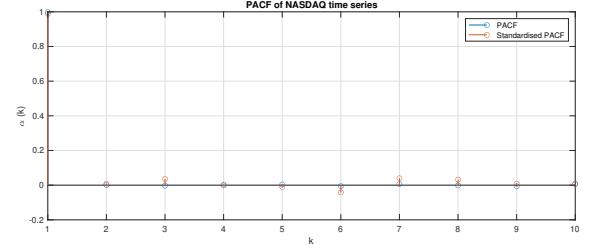
Closing prices for the NASDAQ Financial Index from June 2003 to February 2007 were loaded into MATLAB. Analysis was then performed to find the CRLB, a fundamental measure in estimation theory which represents the lower bound of the variance of unbiased estimators of a deterministic parameter.

2.4.1 AR(1) model

Similarly to the Sunspot time series, MDL , AIC , AIC_c and PACF were used to evaluate the sufficiency of AR models of different orders for modelling the NASDAQ time series. The results are displayed in Figure 19.



(a) MDL , AIC and AIC_c for NASDAQ data.



(b) PACFs for range of model orders

Figure 19: Statistics for NASDAQ data.

It is immediately evident from Figure 19(a) that the global minima for all statistics is at $p = 1$, implying that the correct model order is 1. This is confirmed by the PACF in Figure 19(b), which shows by far the highest magnitude for $p = 1$. This suggests that there is no correlation between samples separated by more than 1 time step, thus an AR(1) model is in fact sufficient to model the NASDAQ time series.

2.4.2 Fisher Information Matrix

The asymptotic CRLB of an AR(p) process is given by Equation 56.

$$\hat{P}_X(f; \boldsymbol{\theta}) = \frac{\hat{\sigma}^2}{|1 - \sum_{m=1}^p \hat{a}_m e^{-j2\pi f m}|^2} \quad (56)$$

where $\hat{\sigma}^2$ is the estimated value of the driving noise variance and p is the order. The elements of the Fisher Information Matrix, $\mathbf{I}(\boldsymbol{\theta})$ are given by Equation 57 where $\boldsymbol{\theta} = [a_1, \sigma^2]$, whilst $\ln [\hat{P}_X(f; \boldsymbol{\theta})]$ is given by Equation 58.

$$[\mathbf{I}(\boldsymbol{\theta})]_{ij} = \frac{N}{2} \int_{-\frac{1}{2}}^{\frac{1}{2}} \frac{\partial \ln [\hat{P}_X(f; \boldsymbol{\theta})]}{\partial \theta_i} \frac{\partial \ln [\hat{P}_X(f; \boldsymbol{\theta})]}{\partial \theta_j} df \quad (57)$$

$$\ln [\hat{P}_X(f; \boldsymbol{\theta})] = \ln [\hat{\sigma}^2] - \ln \left[1 - \sum_{m=1}^p \hat{a}_m e^{-j2\pi f m} \right] - \ln \left[1 - \sum_{m=1}^p \hat{a}_m e^{j2\pi f m} \right] \quad (58)$$

In order to find $[\mathbf{I}(\boldsymbol{\theta})]_{22}$, $p=1$ is assumed and Equation 58 is substituted into Equation 57, along with $\boldsymbol{\theta}$ as below.

$$[\mathbf{I}(\boldsymbol{\theta})]_{22} = \frac{N}{2} \int_{-\frac{1}{2}}^{\frac{1}{2}} \frac{\partial \ln [\hat{P}_X(f; \boldsymbol{\theta})]}{\partial \sigma^2} \frac{\partial \ln [\hat{P}_X(f; \boldsymbol{\theta})]}{\partial \sigma^2} df \quad (59)$$

$$\frac{N}{2} \int_{-\frac{1}{2}}^{\frac{1}{2}} \frac{1}{\sigma^4} df \quad (60)$$

$$\frac{N}{2} \left[\frac{1}{\sigma^4} \right]_{-\frac{1}{2}}^{\frac{1}{2}} \quad (61)$$

$$\frac{N}{2} \left[\frac{1}{2\sigma^4} - \left(-\frac{1}{2\sigma^4} \right) \right] \quad (62)$$

$$[\mathbf{I}(\boldsymbol{\theta})]_{22} = \frac{N}{2\sigma^4} \quad (63)$$

Since $[I(\theta)]_{11}$, $[I(\theta)]_{12}$ and $[I(\theta)]_{21}$ were given, we can conclude that the Fisher Information Matrix is given by Equation 64.

$$[I(\theta)] = \begin{pmatrix} \frac{Nr_{xx}(0)}{\sigma^2} & 0 \\ 0 & \frac{N}{2\sigma^4} \end{pmatrix} \quad (64)$$

2.4.3 CRLB of variance

The CRLB is given by Equation 65.

$$\text{var}(\hat{\theta}_i) \geq [I^{-1}(\theta)]_{ii} \quad (65)$$

Therefore, the CRLB for the given matrix is calculated as below.

$$\text{var}(\hat{\sigma}^2) = \text{var}(\hat{\theta}_2) \geq [I^{-1}(\theta)]_{22} \quad (66)$$

$$\text{var}(\hat{\sigma}^2) \geq \left(\frac{N}{2\sigma^4} \right)^{-1} \quad (67)$$

$$\text{var}(\hat{\sigma}^2) \geq \frac{2\sigma^4}{N} \quad (68)$$

The other bound to be proved is given by 69.

$$\text{var}(\hat{a}_1) = \text{var}(\hat{\theta}_1) \geq [I^{-1}(\theta)]_{11} \quad (69)$$

Using the Fisher Information Matrix, this can be rewritten as Equation 70.

$$\text{var}(\hat{a}_1) \geq \frac{\sigma^2}{Nr_{xx}[0]} \quad (70)$$

$r_{xx}[0] = \frac{\sigma_w^2}{1-a_1^2}$, therefore Equation 70 can be rewritten to obtain the desired result below.

$$\text{var}(\hat{a}_1) \geq \frac{\sigma^2}{N} \frac{1-a_1^2}{\sigma^2} \quad (71)$$

$$\text{var}(\hat{a}_1) \geq \frac{1}{N} (1-a_1^2) \quad (72)$$

Letting N = number of data points and σ^2 = driving noise variance, heatmaps were plotted ranging between 1 and 1001 in increments of 50, and are shown in Figure 20.

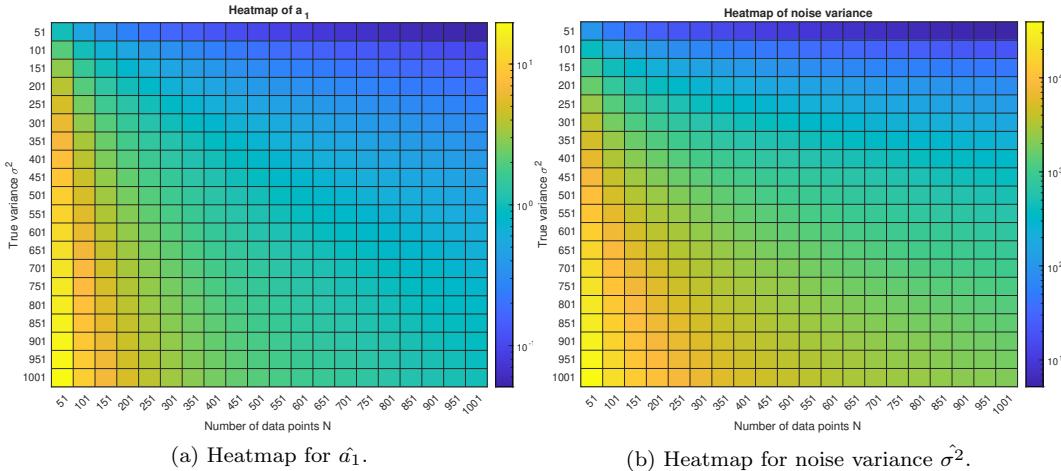


Figure 20: Heatmaps illustrating variation in CRLB.

The largest CRLB for both \hat{a}_1 and $\hat{\sigma}^2$ is found where the noise variance is highest and the number of data points used for calculation is lowest. This can be explained by the fact that $\text{var}(\hat{\sigma}^2) \propto \sigma^4$, $\text{var}(\hat{\sigma}^2) \propto \frac{1}{N}$ and $\text{var}(\hat{a}_1) \propto \frac{1}{N}$. Using the result derived above, $\text{var}(\hat{a}_1)$ can be calculated. From the NASDAQ data, the coefficient a_1 of the AR(1) model is -0.999. Substituting this into Equation 72, the result is obtained $\text{var}(\hat{a}_1) = 2.16 \times 10^6$.

2.4.4 Bound in terms of $\mathbf{A}(f)$

From the CRLB, it can be shown that

$$\text{var}(\hat{P}_X(f; \boldsymbol{\theta})) \geq \frac{\partial \hat{P}_X(f; \boldsymbol{\theta})^T}{\partial \boldsymbol{\theta}} \mathbf{I}^{-1}(\boldsymbol{\theta}) \frac{\partial \hat{P}_X(f; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \quad (73)$$

Letting $A(f) = 1 - a_1 e^{-j2\pi f}$ and defining $\frac{\partial P_X(f; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$ as in Equation 74.

$$\frac{\partial \hat{P}_X(f; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \left(\frac{2e^{-j2\pi f} \sigma^2}{|A(f)|^3} \frac{1}{|A(f)|^2} \right)^T \quad (74)$$

The CRLB defined in Equation 73 therefore simplifies to Equation 75.

$$\text{var}(\hat{P}_X(f; \boldsymbol{\theta})) \geq \frac{4\sigma^6 e^{-4\pi f j}}{N \mathbf{r}_{xx}[\mathbf{0}] |A(f)|^6} + \frac{2\sigma^4}{N |A(f)|^4} \quad (75)$$

2.5 Real world signals: ECG from iAmp experiment

The ECG data recorded during our iAmp experiment was split into 3 segments, representing the ECG for unconstrained breathing, breathing constrained to 15 bpm and breathing constrained to 50 bpm. This was then converted to an RR interval signal which using an algorithm that finds the time between successive R-peaks, giving a measure of heart rate.

2.5.1 PDE of heart rates

Probability density estimates were produced using both the original signal and an averaged version of the RRI signal for the Trial 1 (unconstrained breathing), based on Equation 76, for values of $\alpha = 0.6$ and $\alpha = 1$.

$$\hat{h}[1] = \frac{1}{10} \sum_{i=1}^{10} \alpha h[i], \hat{h}[2] = \frac{1}{10} \sum_{i=11}^{20} \alpha h[i], \dots \quad (76)$$

The results are shown in Figure 21.

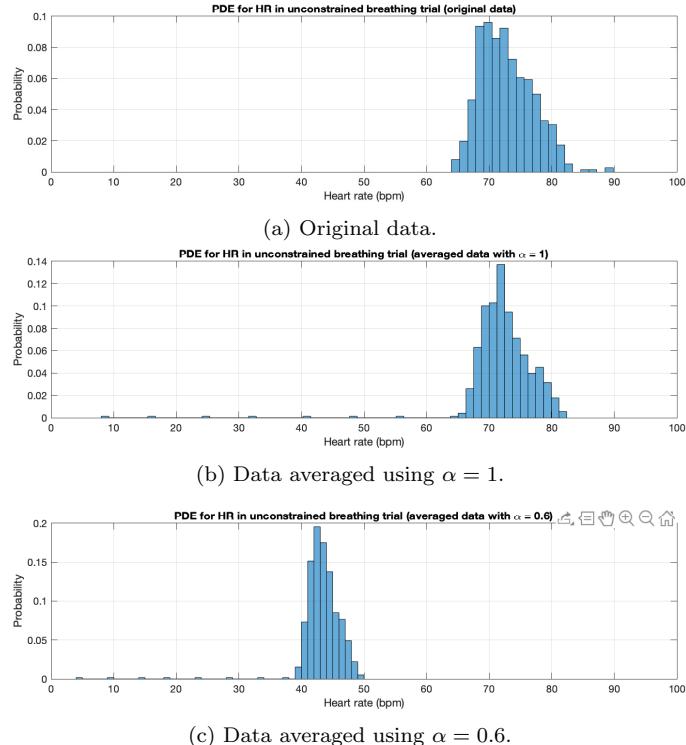


Figure 21: PDEs for Trial 1 (unconstrained breathing).

The mean of the original signal is 72.8 and the SD is 4.24. From Figure 21(21b), averaging the signal appears to have a 'sharpening' effect on one side of the distribution, removing high frequency outliers whilst low frequency outliers remain. This is because averaging has a similar effect to a LPF of the data. The mean of the averaged distribution with $\alpha = 1$ is 72.2, whilst the variance ie 5.85. Whilst there is negligible impact on the mean, the low frequency harmonics that are included have the effect of increasing SD. Reducing α to 0.6, as in 21c reveals that α shifts the mean of the data from to 43.3 and the SD to 3.51. Such an effect on the mean is to be expected based on Equation 76, since α acts as a scaling coefficient for all values. The reduction in SD can be explained by the 'sharpening' effect of low-pass filtering mentioned above.

2.5.2 AR modelling of heart beat

The ACFs for the RRI signals of each trial were calculated and are displayed in Figure 22.

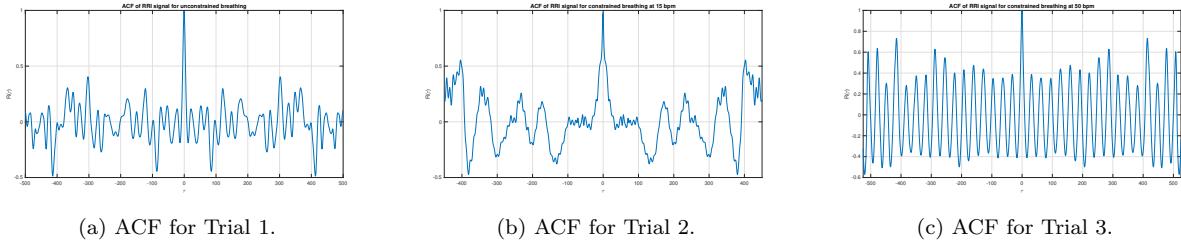


Figure 22: ACFs for Trial 1 (unconstrained breathing), 2 (constrained at 15 bpm) and 3 (constrained at 50 bpm).

The ACFs for all trials are sinusoidal and of infinite length, suggesting that the RRI signal is an AR process. This implies that the current value of the signal has some dependence on previous values. To determine how many previous values this is (i.e. the model order), further investigation is required.

In order to determine the correct model order, PACFs were calculated for the RRI signals for each trial. These are shown in in Figure 23.

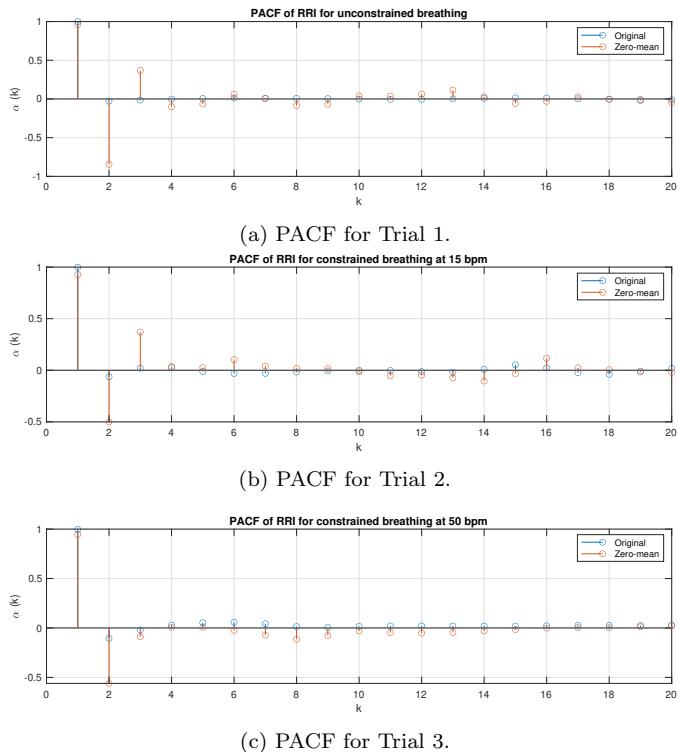


Figure 23: PACFs used to investigate AR model order for RRI signals.

The zero-mean PACF for Trial 1 in Figure 23a appears to converge to zero for shifts above 2 or 3, implying that the

correct model order is 2 or 3. The same is true for Trial 2 in Figure 23b. However, the PACF of Trial 3 in Figure 23c clearly converges to zero after shifts of 2, implying that the correct model order is 2. In order to confirm this, further statistical analysis is required.

As in the previous section, the MDL , AIC and AIC_C can be used to determine the correct model order and resolve the uncertainty of the PACF-based estimate.

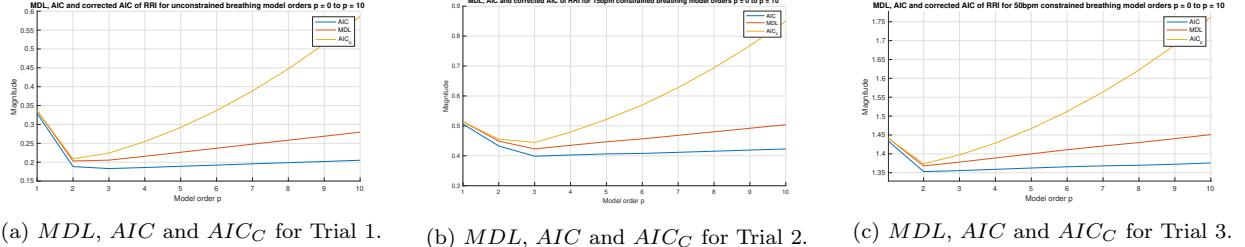


Figure 24: Statistical measures to investigate AR model order for RRI signals.

For Trial 1 in Figure 24a, the AIC global minimum is at 3 but the MDL global minimum is at 2, reflecting the uncertainty in model order from the PACFs. However, the corrected AIC global minimum is at 2, implying that Trial 1 can be modelled as an AR(2) process. For Trial 2 in Figure 24b, the global minima for all measures are at 3, agreeing with what was seen in the PACFs and implying that Trial 2 can be modelled as an AR(3) process. Finally, for Trial 3 in Figure 24c, the global minima for all measures are at 2, again agreeing with what was seen in the PACFs and implying that Trial 3 can be modelled as an AR(2) process.

3 Spectral Estimation and Modelling

The PSD of an ergodic stochastic process \mathbf{X}_n is the power distribution of its constituent frequency components. The PSD can be approximated by a signal known as the periodogram which is calculated using the FFT algorithm as in Equation 77.

$$\hat{P}_X(f) = \frac{1}{N} \left| \sum_{n=0}^{N-1} x[n] e^{-j f n 2\pi} \right|^2 \quad (77)$$

where f = frequency, n = sample number and N = total number of samples.

I implemented this algorithm in a MATLAB function named `pgm.m` which is shown below.

```
function [pgm_out] = pgm(x)
    N = length(x);
    pgm_out = (1/N)*abs(fft(x)).^2;
end
```

This function was applied to WGN processes of lengths $N = 128$, 256 , and 512 obtaining the results shown in Figure 25. The x-axis was also manually rescaled to show zero-center the periodograms and show normalised frequency.

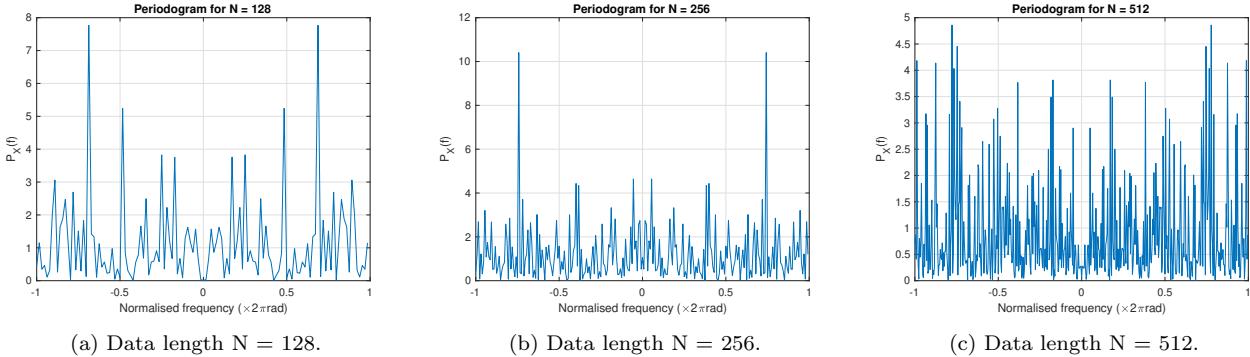


Figure 25: Periodograms for WGN processes of variable lengths.

By rescaling the axis, the expected symmetry of the periodogram becomes evident. The ideal PSD would be 1 for all frequencies since a defining feature WGN is that it is 'white', meaning that it contains all frequencies in equal proportions. Whilst the periodograms do not match this, the mean does appear to be roughly 1, which agrees with our expectations.

3.1 Averaged periodogram estimates

3.1.1 Smoothing periodograms

Applying a zero-phase FIR filter to the periodogram has a smoothing effect. A filter with the impulse response $h = [0.2 \ 0.2 \ 0.2 \ 0.2 \ 0.2]$ was applied to the periodograms. The results are shown in Figure 26.

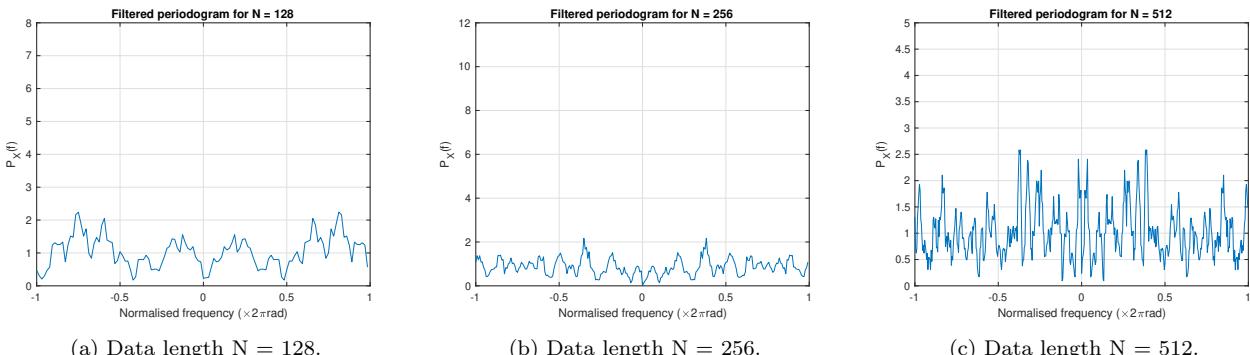


Figure 26: Filtered periodograms for WGN processes of variable lengths with same axis scale.

Comparing the plots in Figure 26 with those in Figure 25, the smoothing effect of the applied filter is evident. By filtering out higher frequencies, it enables us to see the general shape of the signals and thus they more interpretable than the unfiltered versions.

3.1.2 Periodograms of non-overlapping segments

The 1024-sample WGN signal was divided into 8 128-sample segments whose periodograms were obtained using the `pgm.m` function. The results are shown in Figure 27.

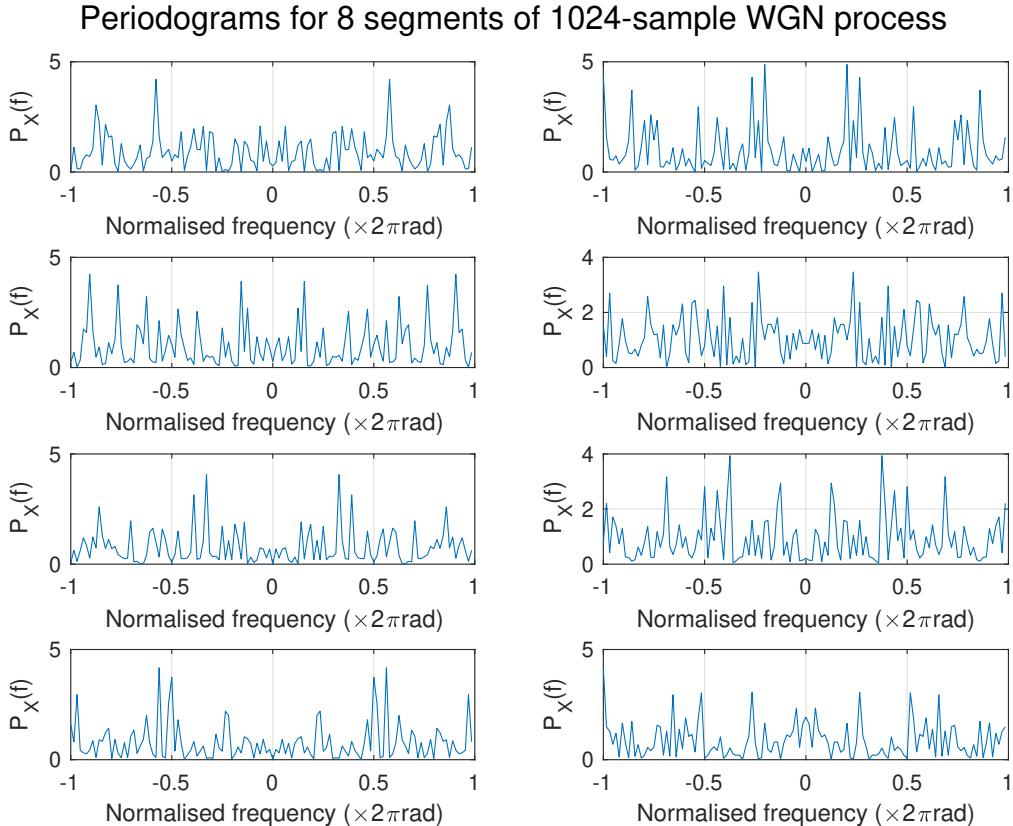


Figure 27: Periodograms obtained for 8 segments of WGN process

These periodograms appear to have the same effect as generating individual 128-sample WGN processes. This can be explained by the fact that WGN is a stationary process, therefore the mean and variance are constant in time so the theoretical PSD of any given segment is the same.

3.1.3 Average periodogram

The 8 periodograms in Figure 27 were averaged using MATLAB's `mean()` function to obtain an average periodogram, which is shown in Figure 28.

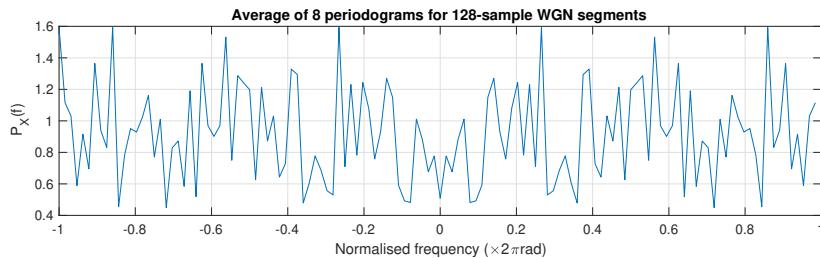


Figure 28: Average periodogram obtained from 8 segments.

In this averaged periodogram, the variance of the signal is significantly lower and the signal mean of 1 is much more clear. This is because averaging has the effect of making the statistical properties of the process clearer as the signal tends towards the ideal theoretical PSD (which, as previously mentioned, is constant at 1). This implies that averaging the periodogram across segments provides a better estimate than generating a single periodogram.

3.2 Spectrum of autoregressive processes

A 1024-sample WGN signal was generated and subsequently filtered using MATLAB's `filter()` function. The filter used has the transfer function given by Equation 78.

$$H(z) = \frac{1}{1 + 0.9e^{-z}} \quad (78)$$

The effect of filtering is shown in Figure 29.

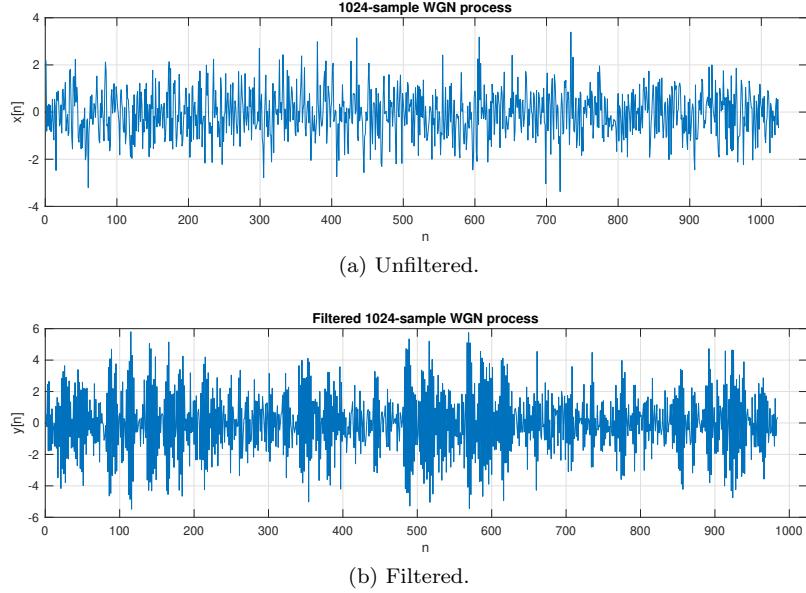


Figure 29: Effect of filtering on WGN process.

Filtering clearly has the effect of increasing the variance of the signal. The signal also appears to have more high frequency oscillations, and has acted as a HPF.

3.2.1 The periodogram and the PSD

The PSD can be calculated using Equation 79.

$$P_Y(f) = \frac{\sigma_X^2}{|1 + \sum_{k=1}^p a_k e^{-j2kh\pi}|^2} \quad (79)$$

This equation was applied to the WGN process in question, obtaining the result shown in Figure 30.

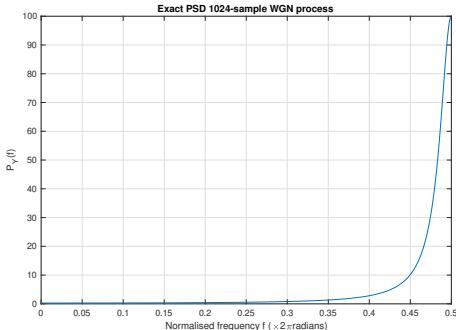


Figure 30: Ideal PSD calculated using Equation 79.

Subsequently, the `pgm.m` function was used to approximate the PSD with a periodogram. The comparison of these is shown in Figure 31.

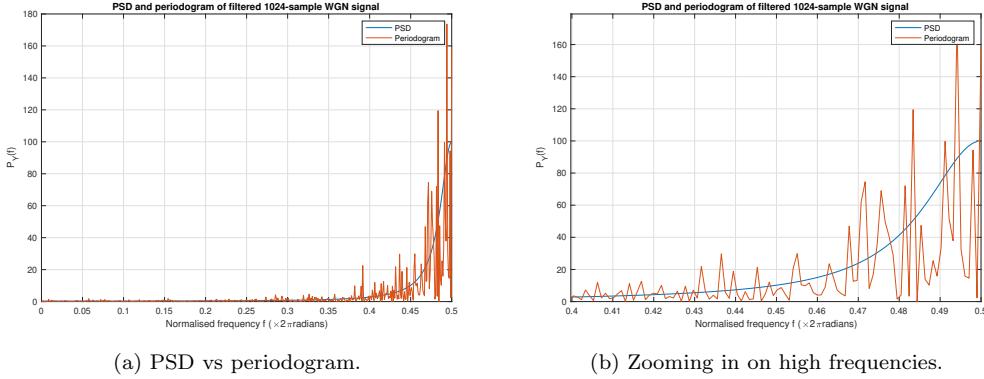


Figure 31: Comparison of PSD and periodogram.

From Figure 31a, the periodogram appears to be a reasonable approximation of the PSD, since the signal power is concentrated at higher frequencies and the periodogram follows the same shape as the PSD. Figure 31b zooms in on (a) for frequencies between 0.4Hz and 0.5 Hz. It is evident from this that, despite following the same general shape as the PSD, the periodogram oscillated with increasing variance. This arises due to the fact that the periodogram is generated using a rectangular window function at each time instant. In the frequency domain, multiplication with a rectangular window is equivalent to convolution with a sinc() function (when the sampling frequency is above the Nyquist-frequency). The resultant signal is therefore oscillatory. The signal could be smoothed by averaging across periodograms of segments of the original signal, reducing the variance of the signal and better approximating the ideal PSD.

3.2.2 Model-based PSD estimation

Assuming that the sequence $y[n]$ is generated by an AR(1) model with two parameters (a_1, σ_X^2) , the calculation of the PSD simplifies into the estimation of \hat{a}_1 and input variance $\hat{\sigma}_X^2$. These estimates are calculated from the estimate of the correlation function of Y , \hat{R}_Y , according to Equations 80 and 81.

$$\hat{a}_1 = -\hat{R}_Y(1)/\hat{R}_Y(0) \quad (80)$$

$$\hat{\sigma}_X^2 = \hat{R}_Y(0) + \hat{a}_1 \hat{R}_Y(1) \quad (81)$$

Using MATLAB's `xcorr()` function, \hat{a}_1 and $\hat{\sigma}_X^2$ were estimated according to above, and then used to calculate a model-based PSD according to Equation 82.

$$\hat{P}_y(f) = \frac{\hat{\sigma}_X^2}{|1 + \hat{a}_1 e^{-j2\pi f}|^2} \quad (82)$$

This model-based PSD was plotted and is compared to the periodogram generated with the `pgm.m` function in Figure 32.

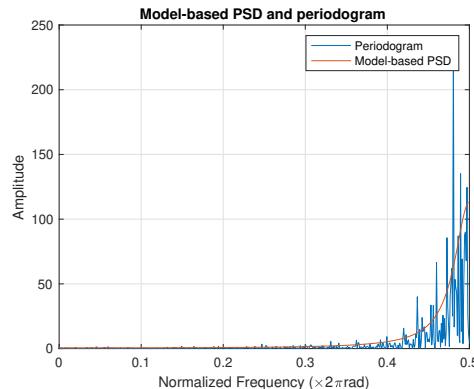


Figure 32: Model-based PSD estimate vs periodogram.

This process was repeated using data from the Sunspot time series, both with the original and standardised data, varying model order. The results of this are shown in Figure 33.

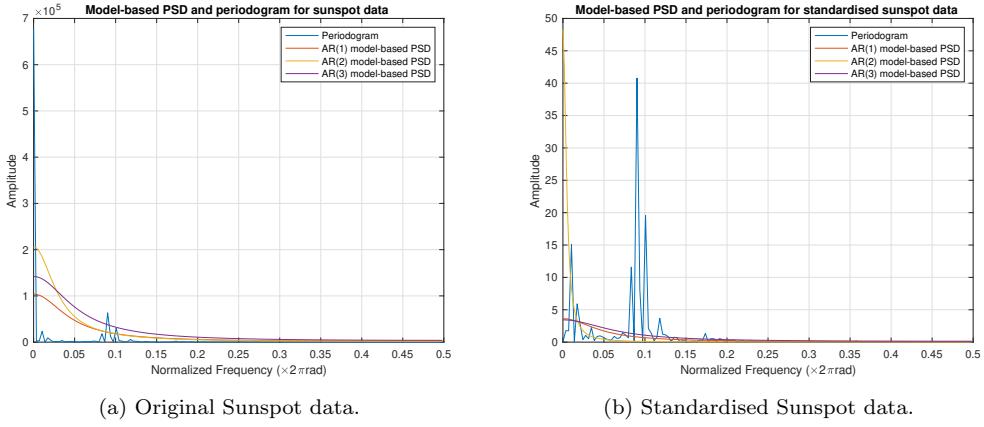


Figure 33: Periodogram and model-based PSDs for Sunspot data.

The model appears to be more accurate for the original Sunspot time series data, though none of the models appear to approximate the PSD well.

3.3 The Least Squares Estimation (LSE) of AR Coefficients

For an AR process given by Equation 83, the biased ACF can be calculated by Equation 84.

$$x[n] = a_1 x[n-1] + a_2 x[n-2] + \dots + a_p x[n-p] + w[n], \quad w[n] \sim \mathcal{N}(0, 1) \quad (83)$$

$$\hat{r}_{xx}[k] = \frac{1}{N} \sum_{n=0}^{N-1-k} x[n]x[n+k] \quad (84)$$

For the process in Equation 83, this can be written as in Equation 85.

$$\hat{r}_{xx}[k] = \sum_{i=1}^p a_i \hat{r}_{xx}[k-i] + \epsilon[k], \quad \text{for } i \geq 1 \quad (85)$$

where $\epsilon[k]$ is the error due to the effect of errors on ACF estimate.

3.3.1 Cost function

The LS cost function for finding the unknown AR coefficients is given by Equation process 86.

$$J = \sum_{k=1}^M \left[\hat{r}_{xx}[k] - \sum_{i=1}^p a_i \hat{r}_{xx}[k-i] \right]^2, \quad \text{for } M \geq p \quad (86)$$

Substituting Equation 85 into 86, Equation 87 is obtained.

$$J = \sum_{k=1}^M \epsilon[k]^2 = \epsilon^T \epsilon \quad (87)$$

This can be rewritten as below to obtain Equation 88.

$$J = (x - s)^T (x - s) \quad (88)$$

The signal model s is calculated as $s = \mathbf{H}a$, therefore Equation 88 can be rewritten as 89.

$$J = (x - \mathbf{H}a)^T (x - \mathbf{H}a) \quad (89)$$

where the observation matrix \mathbf{H} is given by Equation 90.

$$\mathbf{H} = \begin{bmatrix} x[n-1] & x[n-2] & \dots & x[n-p] \\ x[n] & x[n-1] & \dots & x[n+1-p] \\ \vdots & \vdots & \vdots & \ddots \\ x[N-1] & x[N-2] & \dots & x[N-p] \end{bmatrix} \quad (90)$$

3.3.2 The observation matrix \mathbf{H}

Since the signal $x[n]$ is constant across realisations, \mathbf{H} does not display any random behaviour and is therefore deterministic.

3.4 Spectrogram for time-frequency analysis: dial tone pad

3.4.1 Randomly generated London landline number

The underlying concept of touch-tone telephone dialling is the Dual Tone Multi-Frequency (DTMF) system which assigns a signal composed of two sinusoids to each button of the keypad. This is described by Equation 91.

$$y[n] = \sin(2\pi f_1 n) + \sin(2\pi f_2 n) \quad (91)$$

where f_1, f_2 are the frequencies in Hz, and n is the time index. The frequencies corresponding to each digit are shown in Table 4.

	1209Hz	1336Hz	1477Hz
697Hz	1	2	3
770Hz	4	5	6
852Hz	7	8	9
941Hz	*	0	#

Table 4: DTMF frequencies.

Using the DTMF system outlined above, I generated a signal for the dial-tone sequence dialing randomly generated London land-line number. The number generated was '02049726813'.

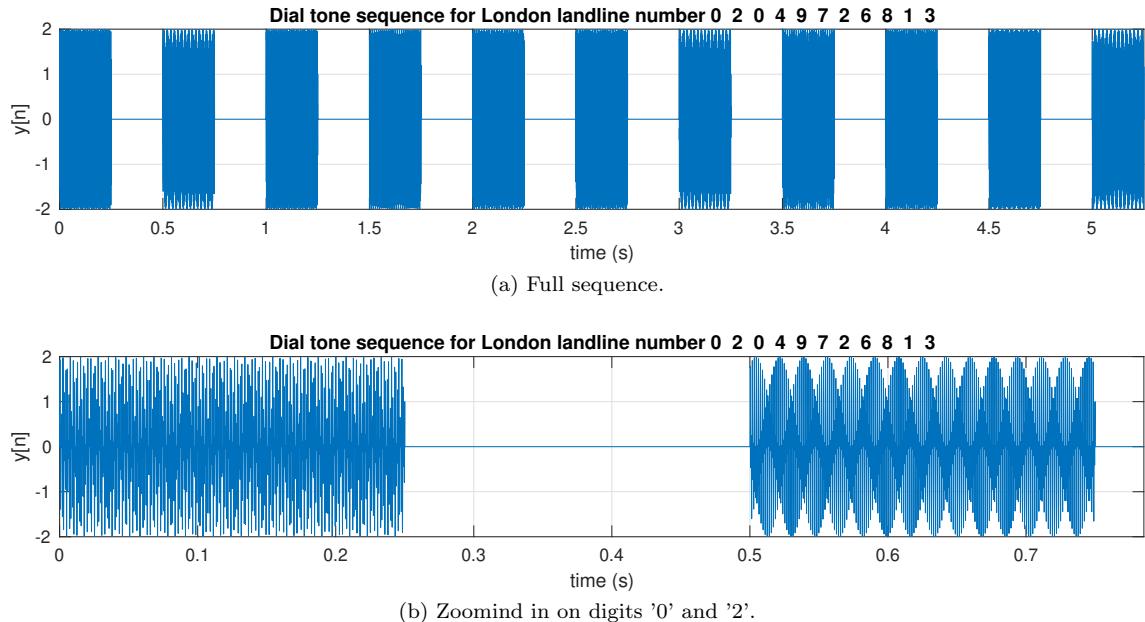


Figure 34: Signal for dial-tone sequence '02049726813'.

The Nyquist frequency is 2954Hz since the maximum frequency used in the DTMF system is 1477Hz. A sampling frequency of 32768Hz ensures that there will be no aliasing for signals up to a frequency 16384Hz. This is significantly higher than required but oversampling increases the SNR so should always be done if possible.

3.4.2 Spectrogram and magnitude spectra

A spectrogram was generated for this dial-tone sequence using MATLAB's `spectrogram()` function, and it shown in Figure 35.

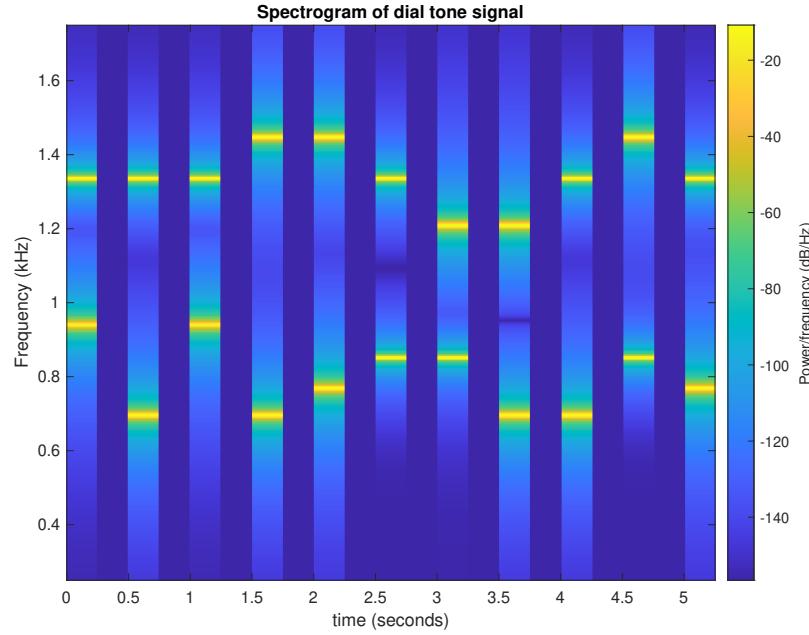


Figure 35: Spectrogram of dial-tone sequence.

This spectrogram clearly shows peaks in signal power for two distinct frequencies for each digit dialed. Checking these against the values in the table provided that describes the DTMF system, the peaks correspond to the expected values. These peaks can be revealed more clear by magnitude spectra, as shown in Figure 36.

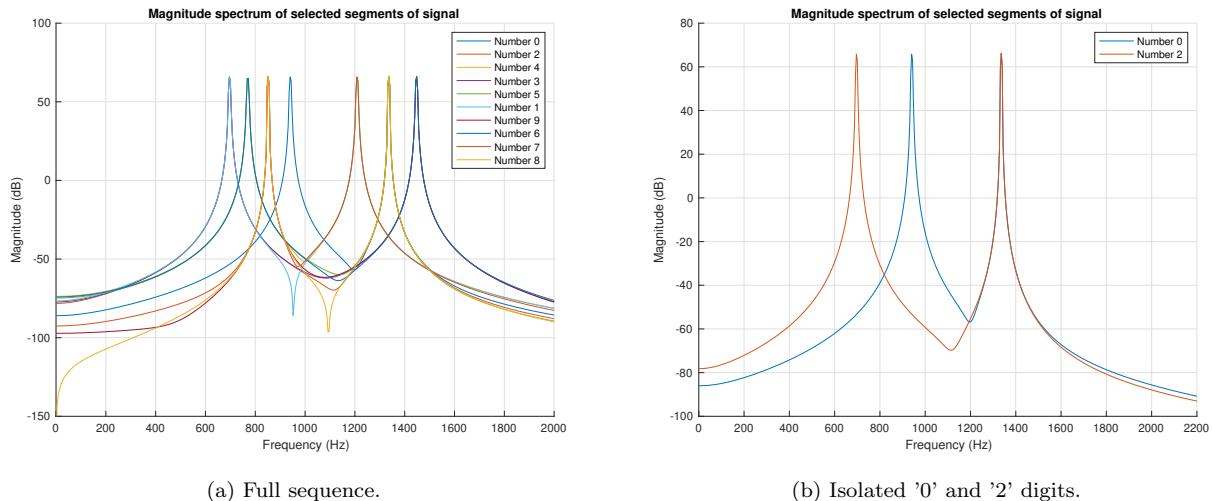


Figure 36: Magnitude spectra for dial sequence signal.

In Figure 36a, there are 7 distinct peaks which correspond to the seven frequencies which make up the tones for all digits in the DTMF system. Isolating the digits '0' and '2' as in Figure 36b, there are 3 distinct peaks which correspond the three distinct frequencies that make up the tones for these two digits.

3.4.3 Key Classification

Looking at the spectrogram in Figure 35 and Table 4, it is possible to identify the keys which have been pressed as well as their order. If the magnitude spectra in Figure 36 are available, but not the spectrogram, then it is possible

to identify the keys which have been pressed, but not the order in which they have been pressed. This is because a magnitude spectrum encapsulates no time-domain information.

3.4.4 Introducing channel noise

In order to better simulate real-world conditions, WGN noise of increasing variance was introduced to corrupt the dial-tone sequence signal. The corrupted signals are shown in Figure 37, along with their spectrograms and magnitude spectra.

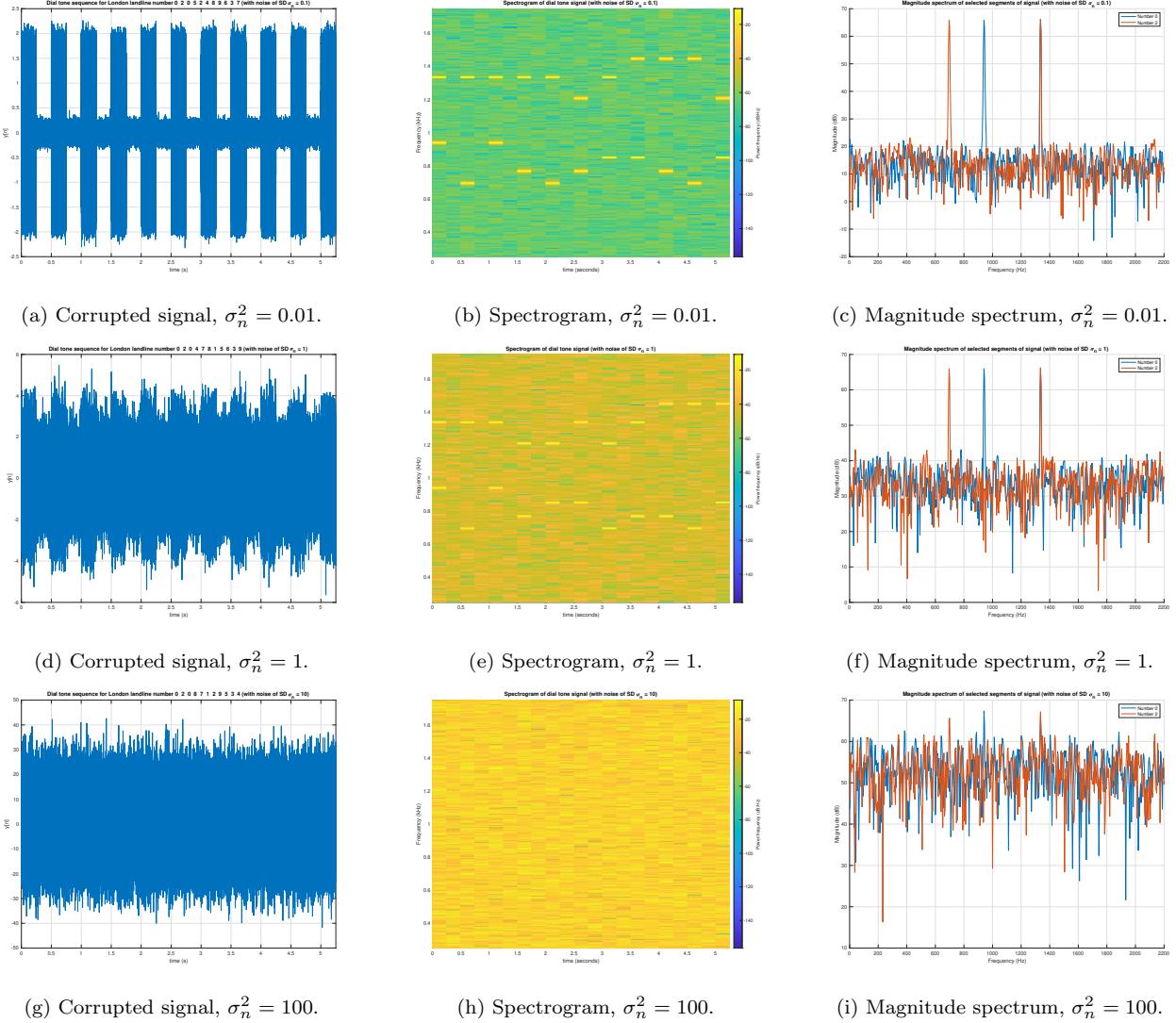


Figure 37: Corrupted dial-tone sequence signals with their spectrograms and magnitude spectra.

It is qualitatively obvious that introducing noise corrupts the signal from the left column of Figure 37. To gain a more quantitative insight into the nature of this corruption, the spectrograms in the middle column are useful. When the noise variance $\sigma_n^2 = 0.01$, the power peaks are still easily distinguishable. For $\sigma_n^2 = 1$, they are less clear but still identifiable. However, when $\sigma_n^2 = 100$, the SNR is so low that the noise power is roughly equivalent to the signal power and the peaks of the tone frequencies are no longer distinguishable. The magnitude spectra prove more useful for identifying tone frequencies in the noisy signal. The tone frequencies are identifiable from the maxima of the magnitude spectra even for noise variance $\sigma_n^2 = 100$ (Figure 37i). It is important to note that these magnitude spectra are only for the digits '0' and '2', so whilst the three frequencies for these digits alone are identifiable, the magnitude spectra for the full sequence may be less clear and it may not be possible to identify all tone frequencies.

3.5 Real world signals: Respiratory sinus arrhythmia from RR-Intervals

The periodogram was calculated for the RRI signals for the three trials from the iAmp experiment. This was both for the original data and for data windowed with window lengths of 50 and 150 and averaged across periodograms.

The results are shown in Figure 38.

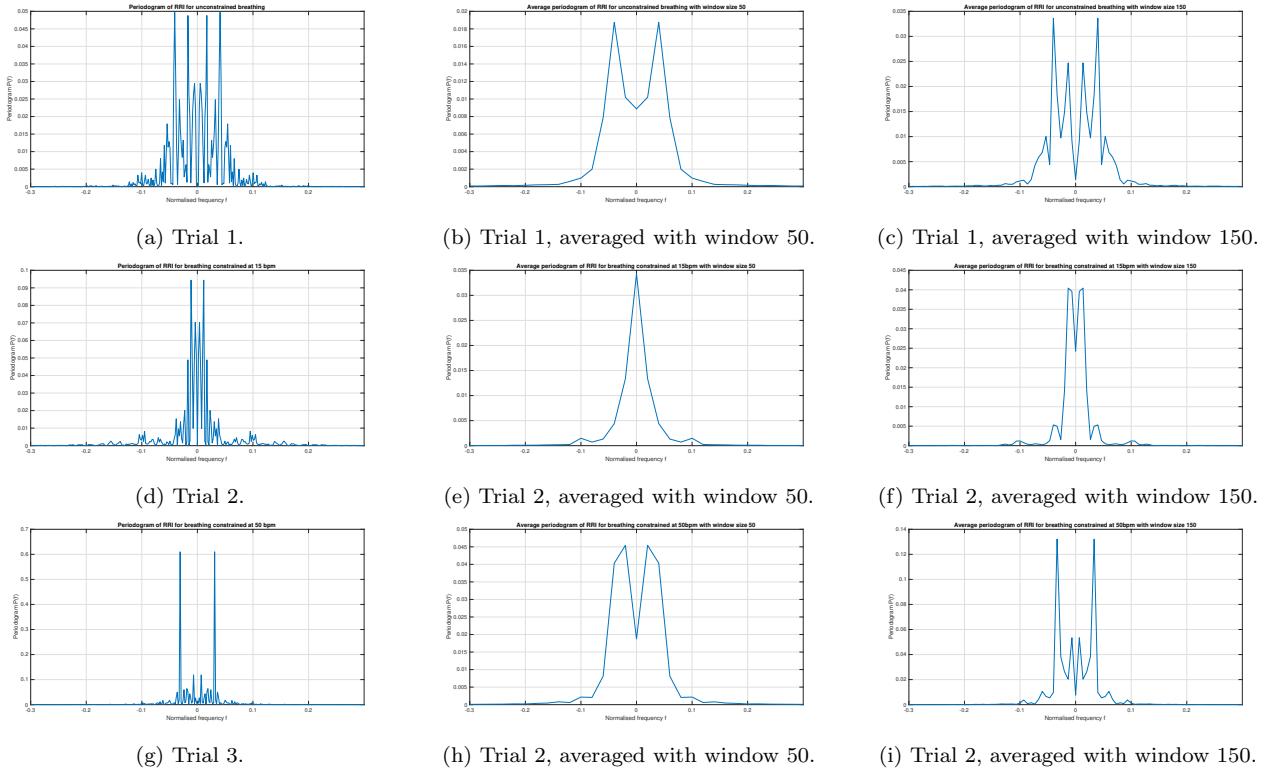


Figure 38: Periodograms for all trials of RRI signals with different windowing applied.

It is apparent from the left column of Figure 38 that each of the trials have peaks at different normalised frequencies. This shows that the dominant frequencies in each signal (which should be indicative of heart rate) are different, suggesting that RSA (whereby altering breathing rate has an effect on heart rate) does take place. Windowing and averaging has the effect of smoothing the periodogram, with more prominent smoothing taking place for the shorter window size of 50 (middle column of Figure 38). This can be explained by the inverse relationship of time and frequency. The signals averaged using window length 150 (right column of Figure 38) are the most useful for deducing the dominant frequencies in each trial. Trial 1 has a dominant peak at a normalised frequency of 0.04, Trial 2 at 0.02 and Trial 3 at 0.03. Trial 1 has a second peak at 0.02, Trial 2 at 0.04 and Trial 3 at 0.01. During RSA, the heart rate increases slightly during inspiration and decreases slightly during exhalation. Therefore, the two peaks seen in each spectrogram correspond to the heart rates during inspiration and exhalation for each exercise.

4 Fixed and Adaptive Optimal Filtering

4.1 Wiener filter

4.1.1 Optimal filter coefficients

The coefficients given for the unknown system are $\mathbf{b} = [1 \ 2 \ 3 \ 2 \ 1]$ and $\mathbf{a} = [1]$. The SNR with normalised data is given by Equation 92.

$$SNR = \frac{\sigma_{y[n]}^2}{\sigma_{\eta[n]}^2} = \frac{1^2}{0.1^2} = 100 = 20dB \quad (92)$$

The optimum Wiener filter coefficients are calculated by Equations 93-95.

$$\mathbf{p}_{zx} = E\{z[n]\mathbf{x}[n]\} = \begin{bmatrix} E\{z[n]x[n]\} \\ E\{z[n]x[n-1]\} \\ \vdots \\ E\{z[n]x[n-N_w]\} \end{bmatrix} = \begin{bmatrix} r_{zx}(0) \\ r_{zx}(-1) \\ \vdots \\ r_{zx}(-N_w) \end{bmatrix} \quad (93)$$

$$\mathbf{R}_{xx} = E\{\mathbf{x}[n]\mathbf{x}^T[n]\} = \begin{bmatrix} r_{xx}(0) & r_{xx}(-1) & \cdots & r_{xx}(-N_w) \\ r_{xx}(1) & r_{xx}(0) & \cdots & r_{xx}(-N_w+1) \\ \vdots & \vdots & \ddots & \vdots \\ r_{xx}(N_w) & r_{xx}(N_w-1) & \cdots & r_{xx}(0) \end{bmatrix} \quad (94)$$

$$\mathbf{w}_{opt} = \mathbf{R}_{xx}^{-1} \cdot \mathbf{p}_{zx} \quad (95)$$

In order to determine the optimal Wiener filter coefficients, R_{xx} and p_{zx} were first calculated in MATLAB and then substituted into Equation 95. I determined the coefficients to be $[0.2122 \ 0.4236 \ 0.6333 \ 0.4237 \ 0.2122]$. These are not the same as the coefficients of the unknown system, but appear to be scaled down by a factor of ≈ 0.21 . This is due to the normalisation of $y[n]$. Removing this normalisation, I obtained the coefficients $[0.9989 \ 1.9869 \ 2.9649 \ 1.9655 \ 0.9944]$ which are very similar to the original coefficients \mathbf{b} .

4.1.2 Effect of noise power

The noise variance was incrementally increased from $\sigma_n^2 = 0.1$ to 10, and its effect on the SNR was investigated. The results are shown in Table 5.

Noise variance	0.1	1	2	4	8	10
SNR (dB)	52.13	12.83	6.35	5.29	3.70	2.47

Table 5: SNR values for a range of noise variances.

Subsequently, the effect of noise variance on models of filters orders was investigated. The results of this are shown in Table 6.

Optimum Wiener Coefficient Index	Noise variance					
	0.1	1	2	4	8	10
1	1.000	0.984	1.000	0.935	0.920	1.046
2	2.000	2.025	1.901	2.127	2.045	2.005
3	3.000	3.005	3.041	3.079	2.786	3.070
4	2.004	1.947	2.025	2.173	1.911	1.952
5	1.004	1.011	0.962	0.889	1.003	1.014

Table 6: Wiener coefficient values for different noise variance.

It is clear from these values that increasing the noise variance increases the error of the Wiener coefficients, which is in line with what would intuitively be expected.

4.1.3 Computational complexity

The computational cost of the calculation of optimal coefficients can be analysed by evaluating Equations 93-95. The matrices in Equations 93 and 94 can be calculated using symmetry properties and are therefore of complexity $\mathcal{O}(N_w)$ (using Big-O notation). The matrix inverse in Equation 95 is of complexity $\mathcal{O}(N_w^3)$. Finally, the matrix in Equation W is of complexity $2\mathcal{O}(N_w^2)$. Summing the complexities of these individual steps, we obtain $\mathcal{O}(N_w^3) + 2\mathcal{O}(N_w^2) + \mathcal{O}(N_w) + \mathcal{O}(N_w) = \mathcal{O}(N_w^3 + 2N_w^2 + 2N_w) \approx \mathcal{O}(N_w^3)$.

4.2 The least mean square (LMS) algorithm

The LMS algorithm is a simple adaptive filter which adapts the Wiener coefficients to adapt to non-stationary signals. It is defined by Equation 96.

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e[n]x(n), \quad n = 0, 1, \dots \quad (96)$$

where μ is the adaptation gain which controls the stability of the algorithm, and $\mathbf{w}(0) = 0$. The error $e[n]$ is calculated as the difference between $z[n]$ and the output of the adaptive filter $\hat{y}[n]$, given by Equations 97 and 98.

$$\hat{y}[n] = \mathbf{w}^T(n)\mathbf{x}(n) = \sum_{m=0}^{N_w} w_m(n)x(n-m) \quad (97)$$

$$e[n] = z[n] - \hat{y}[n] \quad (98)$$

4.2.1 The `lms()` function

In MATLAB, I wrote a function `lms()` which changes Wiener filter coefficients to adapt to non-stationary signals over time by approximating the LMS estimate $\hat{y}[n]$ for the signal $z[n]$, as well as outputting the estimation error $e[n]$ and a matrix containing the evolution of the adaptive weights over time. The code for this function is shown below.

```
function [y_estimate,error,coeffs] = lms(x,z,mu,order)
    N = length(x); % signal length
    coeffs = zeros(order, N-1); % each row represents coefficient for 1 input variable
    y_estimate = zeros(N, 1);
    error = zeros(N, 1);
    for i = order+1:N
        a = coeffs(:,i-order); % AR coefficient
        b = x(i:-1:i-order+1); % MA coefficient
        y_estimate(i) = a'*b;
        error(i) = z(i) - y_estimate(i); % error
        coeffs(:,i-order+1) = coeffs(:,i-order)+mu*error(i)*b;
    end
end
```

Applying this function to the \mathbf{x} and \mathbf{z} vectors in the previous section, the results shown in Figure 39 are obtained.

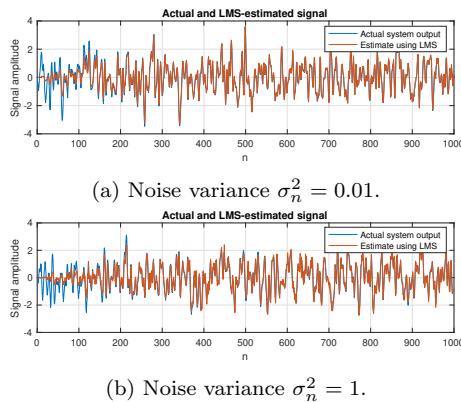


Figure 39: Effect of `lms()` for \mathbf{x} and \mathbf{z} with variable noise variance.

It is evident that the function generates coefficients which accurately estimate the system signal, but that the coefficients take longer to adapt and are less accurate for a signal with higher noise variance.

4.2.2 Time evolution of coefficients

Figure 40 shows the evolution of the filter coefficients in time for a range of values for adaptation gain μ .

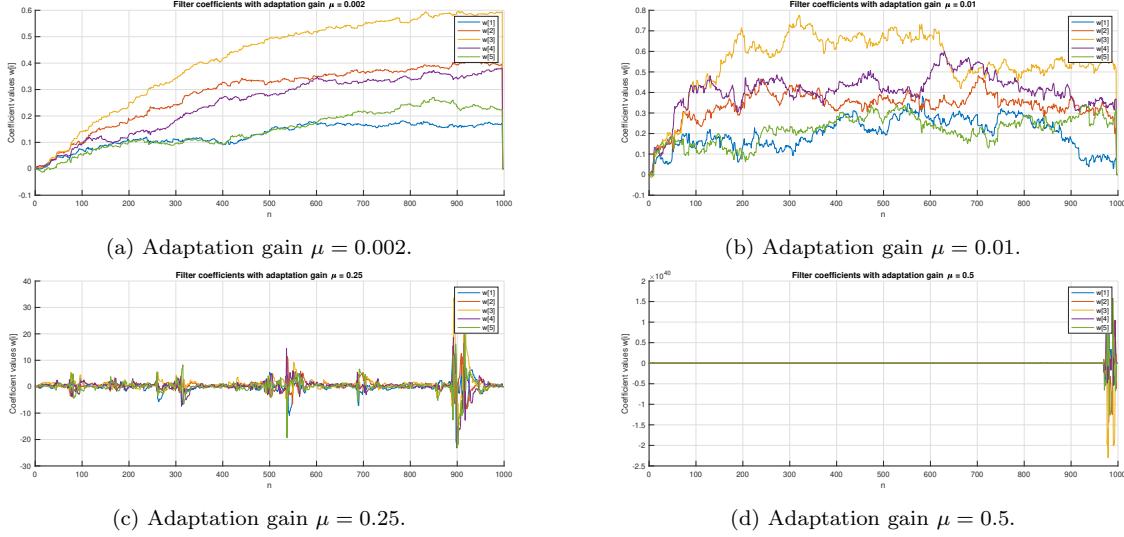


Figure 40: Coefficient evolution using `lms()` for variable adaptation gain μ .

Increasing adaptation gain μ causes the Wiener filter coefficients to converge more quickly up to a point. However, when μ is increased beyond a certain value, the coefficients actually diverge in an oscillatory and unstable manner, implying that there exists an optimum value for μ which results in fast but stable convergence.

4.2.3 Computational complexity

The complexity of the LMS algorithm can be analysed by evaluating Equations 96-98. Equation 96 has complexity $\mathcal{O}(1)$. Equation 98 requires N_w multiplications and N_w+1 additions, therefore having complexity $\mathcal{O}(2N_w+1)$. Finally, Equation 97 has complexity $\mathcal{O}(2N_w + 2)$. Therefore, the overall computational complexity is $\mathcal{O}(5N_w + 1) \approx \mathcal{O}(N_w)$. This algorithm has relatively low computational complexity and therefore can be utilised in adaptive filtering.

4.3 Gear shifting

Since lower adaptation gains have been found to be more effective at making coefficients converge in a stable manner, 'gear shifting' which implements time-varying adaptation gain is a sensible improvement. Gear shifting was implemented by extending the `lms()` function such that μ is increased by 10% if the error signal decreases and decreased by 10% if the error signal increases. The effect of this is shown using an initial adaptation gain $\mu = 0.002$ in Figure 41.

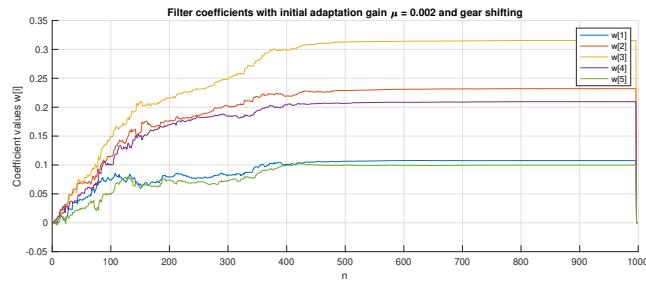


Figure 41: Evolution of filter coefficients with gear shifting.

This causes μ to quickly converge to an optimum value, in turn causing the filter coefficients to quickly converge to stable values as well.

4.4 Identification of AR processes

4.4.1 AR model implementation

The system shown in Figure 42 is an AR(2) process, therefore the LMS function needs to estimate two AR filter coefficients, a_1 and a_2 .

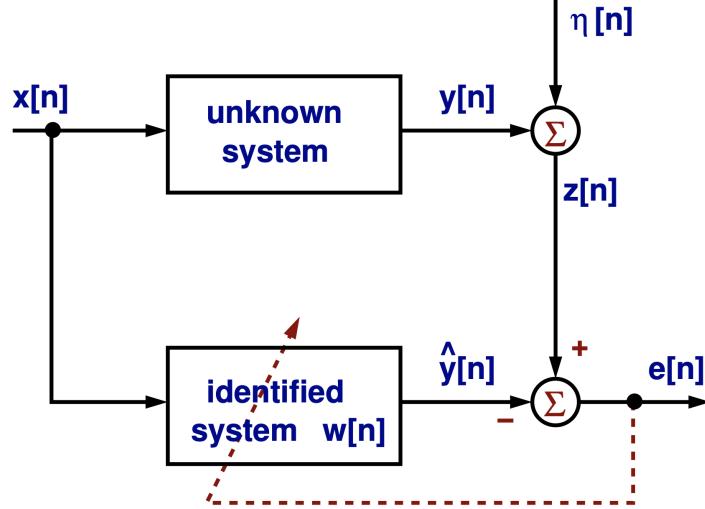


Figure 42: System given in coursework instructions.

The given system is described by Equation 99.

$$y(n) = \sum_{i=0}^M b_i x(n-i) - \sum_{j=0}^N a_j x(n-j) = -0.9x(n-1) - 0.2x(n-2) \quad (99)$$

Therefore, the theoretical values that we expect the coefficients to converge to are $a_1 = -0.9$ and $a_2 = -0.2$.

4.4.2 Evolution of model coefficients

Figure 43 shows the values of the model coefficients a_1 and a_2 using the `lms()` function with a range of values for adaptation gain μ .

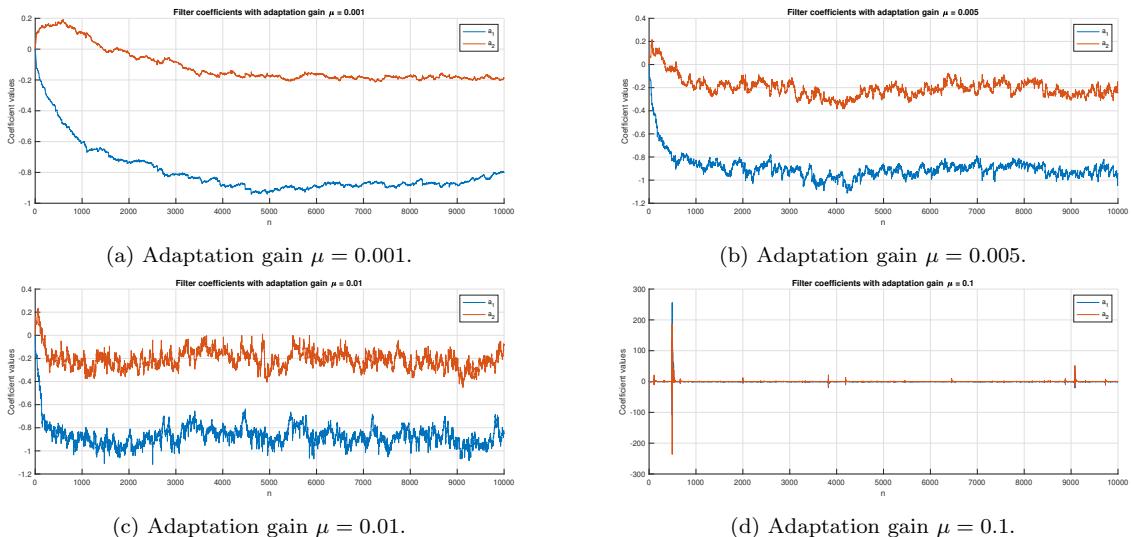


Figure 43: Coefficient evolution for given system using `lms()` and variable adaptation gain μ .

An adaptation gain of $0.001 \leq \mu \leq 0.01$ appears to strike a balance between fast convergence but minimal distortion and therefore more stability.

4.5 Speech recognition

The AR system given can be used to estimate the model order for best prediction performance for 5 audio snippets. These snippets contain my voice saying the letters 'E', 'A', 'S', 'T' and 'X'. Figure 44 shows the original signal for the letter 'E'.

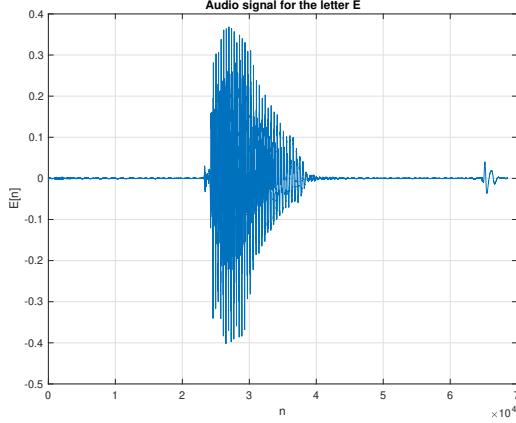


Figure 44: Audio signal of my voice saying 'E'.

In order to determine the model order of each signal, the *AIC* and *MDL* were calculated for each and their global minima were found. The results are displayed in Table 7.

Measure	Letter				
	E	A	S	T	X
AIC	3	2	11	6	12
MDL	2	2	7	5	10

Table 7: Statistics for different audio signals.

The Wiener coefficients were calculated using the same method before, and their evolution is shown in Figure 45. Gear shifting was not used because the signals are non-stationary.

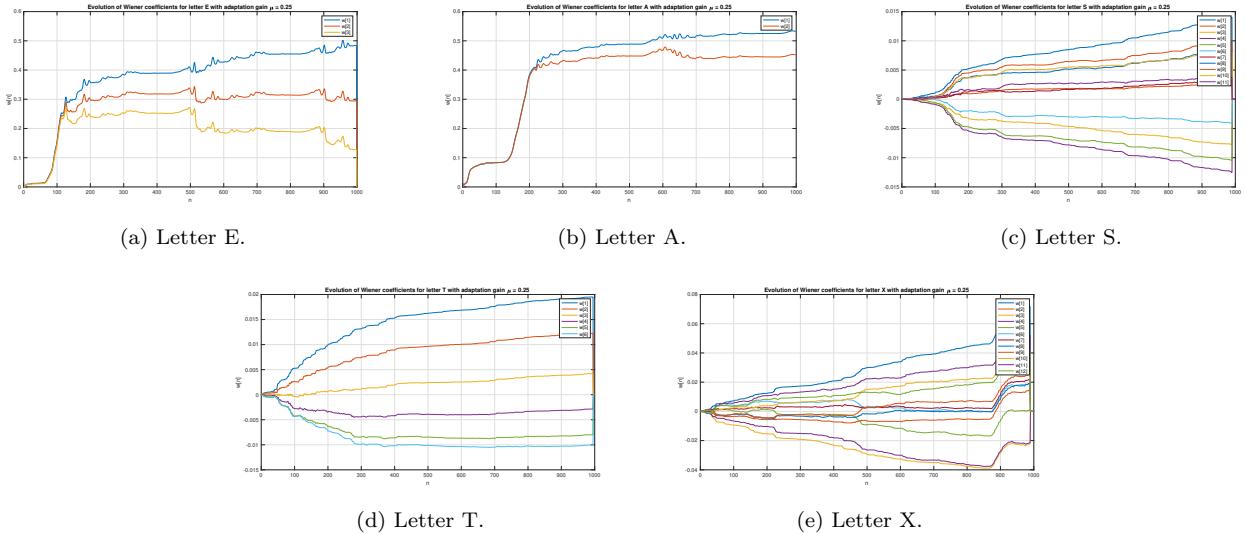


Figure 45: Evolution of coefficients for different audio signals.

5 MLE for the Frequency of a Signal

1. We start with Equation 100

$$J'(\alpha_1, \alpha_2, f_0) = (\mathbf{x} - \alpha_1 \mathbf{c} - \alpha_2 \mathbf{s})^T (\mathbf{x} - \alpha_1 \mathbf{c} - \alpha_2 \mathbf{s}) \quad (100)$$

where $\alpha_1 = A \cos(\phi)$, $\alpha_2 = -A \sin(\phi)$, $\mathbf{c} = [1, \cos(2\pi f_0), \dots, \cos(2\pi f_0(B-1))]^T$ and $\mathbf{s} = [0, \sin(2\pi f_0), \dots, \sin(2\pi f_0(B-1))]^T$.

$(\mathbf{x} - \alpha_1 \mathbf{c} - \alpha_2 \mathbf{s})$ can be rewritten as in Equation 101

$$(\mathbf{x} - \alpha_1 \mathbf{c} - \alpha_2 \mathbf{s}) = \begin{pmatrix} x[0] - A \cos(\phi) \cos(0) + A \sin(\phi) \sin(0) \\ x[1] - A \cos(\phi) \cos(2\pi f_0) + A \sin(\phi) \sin(2\pi f_0) \\ \vdots \\ x[N-1] - A \cos(\phi) \cos(2\pi f_0(N-1)) + A \sin(\phi) \sin(2\pi f_0(N-1)) \end{pmatrix} \quad (101)$$

Using the trigonometric identity $\cos(a+b) = \cos(a)\cos(b) + \sin(a)\sin(b)$, this can be rewritten again as in Equation 102.

$$(\mathbf{x} - \alpha_1 \mathbf{c} - \alpha_2 \mathbf{s}) = \begin{pmatrix} x[0] - A \cos(0 + \phi) \\ x[1] - A \cos(2\pi f_0 + \phi) \\ \vdots \\ x[N-1] - A \cos(2\pi f_0(N-1) + \phi) \end{pmatrix} \quad (102)$$

Therefore, Equation 100 can be rewritten as below to obtain the desired result in Equation 106.

$$J'(\alpha_1, \alpha_2, f_0) = \begin{pmatrix} x[0] - A \cos(0 + \phi) \\ x[1] - A \cos(2\pi f_0 + \phi) \\ \vdots \\ x[N-1] - A \cos(2\pi f_0(N-1) + \phi) \end{pmatrix}^T \begin{pmatrix} x[0] - A \cos(0 + \phi) \\ x[1] - A \cos(2\pi f_0 + \phi) \\ \vdots \\ x[N-1] - A \cos(2\pi f_0(N-1) + \phi) \end{pmatrix} \quad (103)$$

$$= (x[0] - A \cos(0 + \phi) \dots x[N-1] - A \cos(2\pi f_0(N-1) + \phi))^2 \quad (104)$$

$$= \sum_{n=0}^{N-1} (x[n] - A \cos(2\pi f_0 n + \phi))^2 \quad (105)$$

$$J'(\alpha_1, \alpha_2, f_0) = J(\boldsymbol{\theta}) \quad (106)$$

2. Equation 100 can be rewritten $J'(\boldsymbol{\alpha}, f_0) = (\mathbf{x} - \mathbf{H}\boldsymbol{\alpha})^T (\mathbf{x} - \mathbf{H}\boldsymbol{\alpha}) = \mathbf{x}\mathbf{x}^T - 2\mathbf{x}^T \mathbf{H}\boldsymbol{\alpha} + \boldsymbol{\alpha}^T \mathbf{H}^T \mathbf{H}\boldsymbol{\alpha}$, and the minimising solution can be found as below, obtaining the result in Equation 111.

$$\frac{\partial J(\boldsymbol{\alpha})}{\partial \boldsymbol{\alpha}} = -2\mathbf{H}^T \mathbf{x} + 2\mathbf{H}^T \mathbf{H}\boldsymbol{\alpha} = 0 \quad (107)$$

$$\hat{\boldsymbol{\alpha}} = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{x} \quad (108)$$

$$J(\hat{\boldsymbol{\alpha}}, f_0) = \mathbf{x}\mathbf{x}^T - 2\mathbf{x}^T \mathbf{H}\hat{\boldsymbol{\alpha}} + \hat{\boldsymbol{\alpha}}^T \mathbf{H}^T \mathbf{H}\hat{\boldsymbol{\alpha}} \quad (109)$$

$$= \mathbf{x}\mathbf{x}^T - 2\mathbf{x}^T \mathbf{H} (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{x} + ((\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{x})^T \mathbf{H}^T \mathbf{H} (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{x} \quad (110)$$

$$J(\hat{\boldsymbol{\alpha}}, f_0) = \mathbf{x}\mathbf{x}^T - \mathbf{x}^T \mathbf{H} (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{x} \quad (111)$$

3. Letting $\mathbf{H} = [\mathbf{c}, \mathbf{s}]$ the MLE of the frequency, f_0 can be found by verifying that $\mathbf{x}^T \mathbf{H} (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{x}$ can be rewritten as in Equation 112, since it is known that maximising $\mathbf{x}^T \mathbf{H} (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{x}$ minimises $J'(\boldsymbol{\alpha}, f_0)$.

$$\begin{bmatrix} \mathbf{c}^T \mathbf{x} \\ \mathbf{s}^T \mathbf{x} \end{bmatrix}^T = \begin{bmatrix} \mathbf{c}^T \mathbf{c} & \mathbf{c}^T \mathbf{s} \\ \mathbf{s}^T \mathbf{c} & \mathbf{s}^T \mathbf{s} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{c}^T \mathbf{x} \\ \mathbf{s}^T \mathbf{x} \end{bmatrix} \quad (112)$$

$$\mathbf{H}^T \mathbf{x} = \begin{bmatrix} \mathbf{c}^T \mathbf{x} \\ \mathbf{s}^T \mathbf{x} \end{bmatrix} \Leftrightarrow (\mathbf{H}^T \mathbf{x})^T = \mathbf{x}^T \mathbf{H} = \begin{bmatrix} \mathbf{c}^T \mathbf{x} \\ \mathbf{s}^T \mathbf{x} \end{bmatrix}^T \quad (113)$$

Therefore, the desired condition is proved as shown in Equation 114.

$$\mathbf{x}^T \mathbf{H} (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{x} = \begin{bmatrix} \mathbf{c}^T \mathbf{x} \\ \mathbf{s}^T \mathbf{x} \end{bmatrix}^T \begin{bmatrix} \mathbf{c}^T \mathbf{c} & \mathbf{c}^T \mathbf{s} \\ \mathbf{s}^T \mathbf{c} & \mathbf{s}^T \mathbf{s} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{c}^T \mathbf{x} \\ \mathbf{s}^T \mathbf{x} \end{bmatrix} \quad (114)$$

6 Nomenclature

<i>PDF</i>	Probability Density Function
<i>PDE</i>	Probability Density Estimate
<i>WGN</i>	White Gaussian Noise
<i>ACF</i>	Autocorrelation Function
<i>CCF</i>	Cross-Correlation Function
<i>PACF</i>	Partial Autocorrelation Function
<i>MDL</i>	Minimum Description Length
<i>AIC</i>	Akaike Information Criterion
<i>PH</i>	Prediction Horizon
<i>MA</i>	Moving Average
<i>AR</i>	Auto-Regressive
<i>LPF</i>	Low-Pass Filter
<i>HPF</i>	High-Pass Filter
<i>WSS</i>	Wide-Sense Stationary
<i>CRLB</i>	Cramer-Rao Lower Bound
<i>ECG</i>	Electrocardiogram
<i>FIR</i>	Finite Impulse Response
<i>RRI</i>	RR Interval
<i>DFT</i>	Discrete Fourier Transform
<i>FFT</i>	Fast Fourier Transform
<i>MLE</i>	Maximum Likelihood Estimate