Advanced Signal Processing Coursework

ELEC60002

Adam Horsler, ah2719, 01738592

March 27th, 2022

Contents

1 Random signals and stochastic processes

1.1 Statistical estimation

1.1.1

The theoretical mean of $\mathbf{x}$, $m = (0+1)/2 = 0.5$.
The sample mean, $\hat{m} = \frac{1}{N}\Sigma_{n=1}^{N} x[n] = 0.489$.
The sample mean is accurate as 1000 is a large sample size. As sample size tends to infinity, the sample mean will equal the theoretical mean as there is no biasness.

1.1.2

The theoretical standard deviation, $\sigma = (1-0)/\sqrt{12} = 0.2887$.
The sample standard deviation, $\hat{\sigma} = \sqrt{\frac{1}{N-1}\Sigma_{n=1}^{N}(x[n]-\hat{m})^2} = 0.2832$.
The sample standard deviation is accurate as 1000 is a large sample size. As sample size tends to infinity, the sample standard deviation will equal the theoretical standard deviation, due in part to the Bessel correction as it results in an unbiased estimator.

1.1.3

Figure 1 shows how the estimates of mean and standard deviation cluster around their theoretical values (as calculated in Part 1.1.1 and Part 1.1.2). The clustering is both above and below the theoretical values at roughly equal levels, showing that there is no bias in the sample mean.

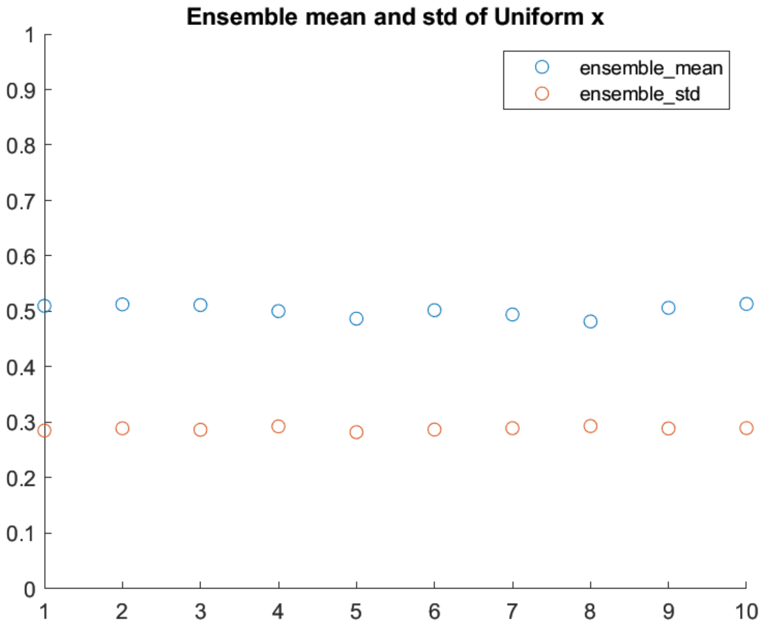


Figure 1: Graph showing sample means and variances against ensembles of $\mathbf{x}$

1.1.4

Figure 2 shows the approximated pdf of $\mathbf{x}$ and the theoretical pdf with a 1000-sample realisation. Increasing the sample size results in the estimate being closer to the theoretical value, thus converging.

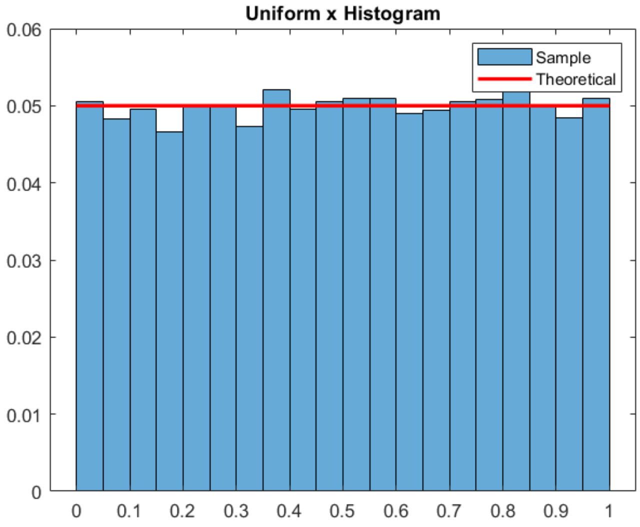


Figure 2: Histogram of approximate and theoretical pdf of **x**

1.1.5

Let **x** be a 1000-sample, zero mean, unit standard deviation Gaussian random process.
Then, the theoretical mean of **x**, $m = 0$.
The calculated sample mean, $\hat{m} = 0.0071$.

The theoretical standard deviation, $\sigma = 1$.
The calculated standard deviation, $\hat{\sigma} = 0.958$.

Figure 3 shows how the estimates of mean and standard deviation cluster around their theoretical values. The clustering is both above and below the theoretical values at roughly equal levels, showing that there is no bias in the sample mean or standard deviation.

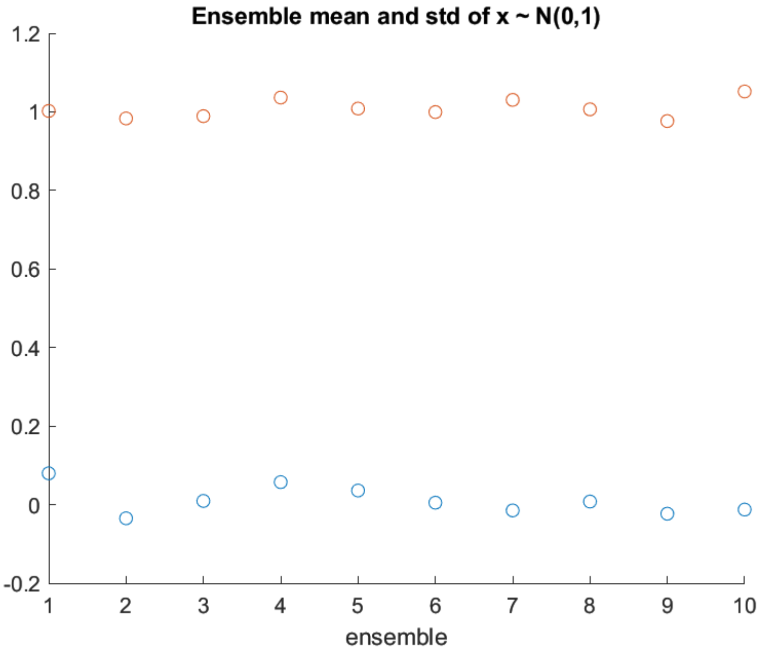


Figure 3: Ensemble mean and std's of **x**

Figure 4 shows the normalised histogram plot of **x**. It has a similar shape to the theoretical standard normal Gaussian process. Given more sample realisations, the histogram will converge to the theoretical process. The more bins there are, the greater it converges. This is due to the increased precision from higher bins.

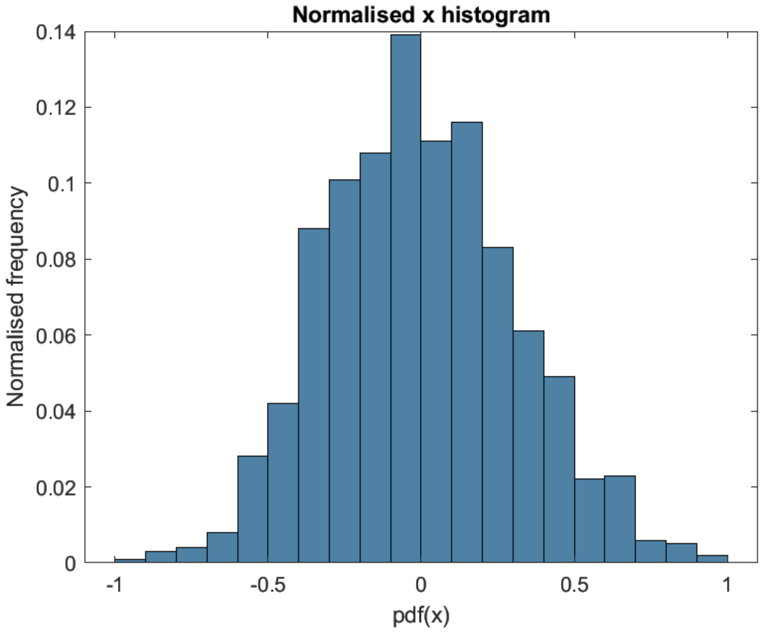
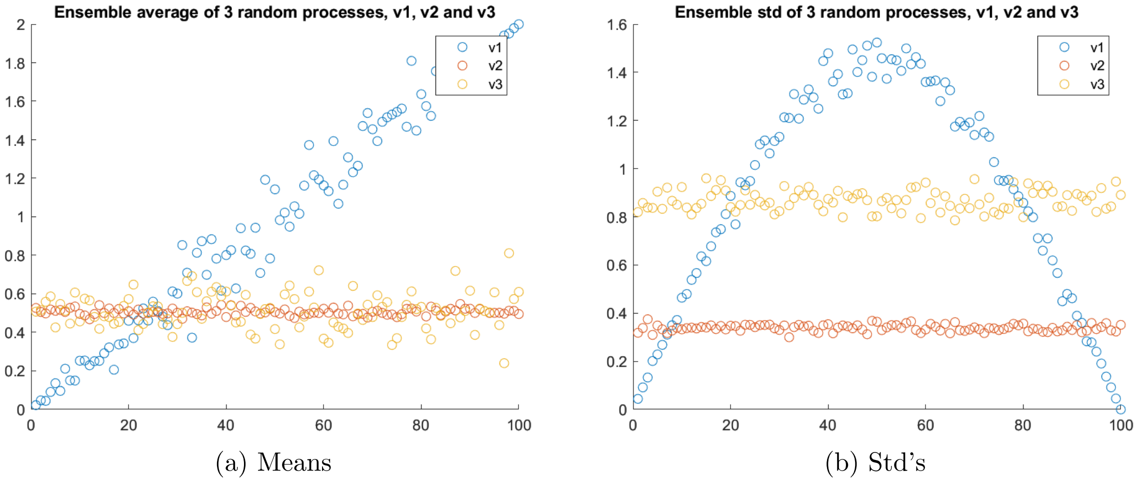


Figure 4: Normalised histogram of **x**

1.2 Stochastic Processes

Figure 5 show the means and standard deviations of 3 random processes, **rp1**, **rp2** and **rp3**, as a function of time. **v1** corresponds to rp1, **v2** corresponds to rp2 and $\vec{v}3$ corresponds to rp3. The figure shows that **v1** is not stationary as its mean is not constant, while **v2** and **v3** are stationary.



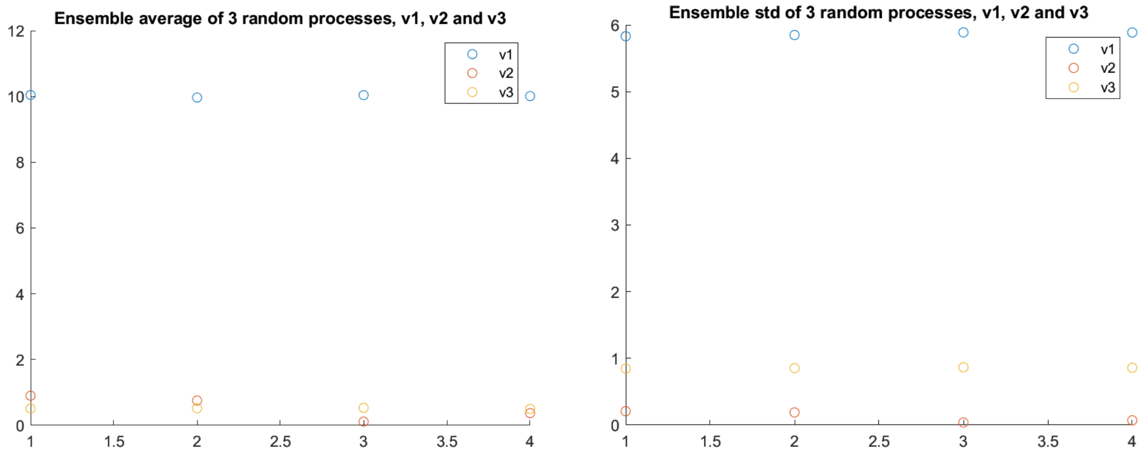
(a) Means

(b) Std's

Figure 5: Mean and std against time

1.2.1

Figure 6 shows the ensemble mean and standard deviations of 4 realisations of the processes mentioned in Part 1.2.1. The figure shows that all 3 processes are ergodic as each realisation has the same mean and standard deviation to a small degree of error, i.e. less than 5%.



(a) Means

(b) Std's

Figure 6: Ensemble means and std's against time (4 realisations)

1.2.2

rp1: Theoretical mean calculation:
$E[v] = E[(rand(M,N) - 0.5).*Mc + Ac] = E[(rand(M,N) - 0.5)].*E[Mc] + E[Ac]$.
$E[(rand(M,N) - 0.5)] = 0.E[Mc] = 0$. Therefore, $E[v] = E[Ac] = Ac$, where we have used the fact that rand(M,N), Mc and Ac are independent of one another, thus we can split the expectation function up. But Ac is a range from 1:N, thus it is variable.
Theoretical variance calculation: $var[v] = E[rp3^2] - E[rp3]^2$
$= E[((rand(M,N) - 0.5).*Mc + Ac)^2] - Ac^2$
$= E[(rand(M,N) - 0.5).*Mc)^2 + 2*(rand(M,N) - 0.5).*Mc)*Ac + Ac^2] - Ac^2$.
$E[(rand(M,N) - 0.5).*Mc)^2] = 0$.
$E[2*(rand(M,N) - 0.5).*Mc)*Ac + Ac^2] = sin(Ac)$.

rp2: Theoretical mean calculation:
$E[v] = E[(rand(M,N) - 0.5).*Mr + Ar]$
$= E[(rand(M,N) - 0.5)].*E[Mr] + E[Ar]$.
$E[(rand(M,N) - 0.5)] = 0$.
$E[Mr] = 0.5$, $E[Ar] = 0.5$.
Therefore, $E[v] = E[Ar] = 0.5$.
Theoretical variance calculation: $var[v] = E[rp2^2] - E[rp2]^2$
$= E[((rand(M,N) - 0.5).*Mr + Ar)^2] - 0.25 = 0.680625$.
Therefore, $var[v] = 0.681$.

rp3: Theoretical mean calculation:
$E[v] = E[(rand(M,N) - 0.5)m + a]$
$= E[(rand(M,N) - 0.5)].*E[m] + E[a]$.
$E[(rand(M,N) - 0.5)] = 0$.
$E[m] = 3$, $E[a] = 0.5$.
$E[v] = E[a] = 0.5$.

Theoretical variance: $var[v] = E[((rand(M,N) - 0.5)m + a)^2] - 0.25$
$= 0.375.$

Therefore, $var[v] = 0.375$

Process	Mean	Variance
rp1	1:N	$sin(Ac)$
rp2	0.5	0.681
rp3	0.5	0.375

Table 1: Mean and Variance of random processes

1.3 Estimation of Probability Distribution

1.3.1

Figure 7 shows a histogram estimate of the pdf of **v**, a standard normal Gaussian random process, using the pdf function. The histogram closely resembles the theoretical pdf of a standard normal Gaussian process.

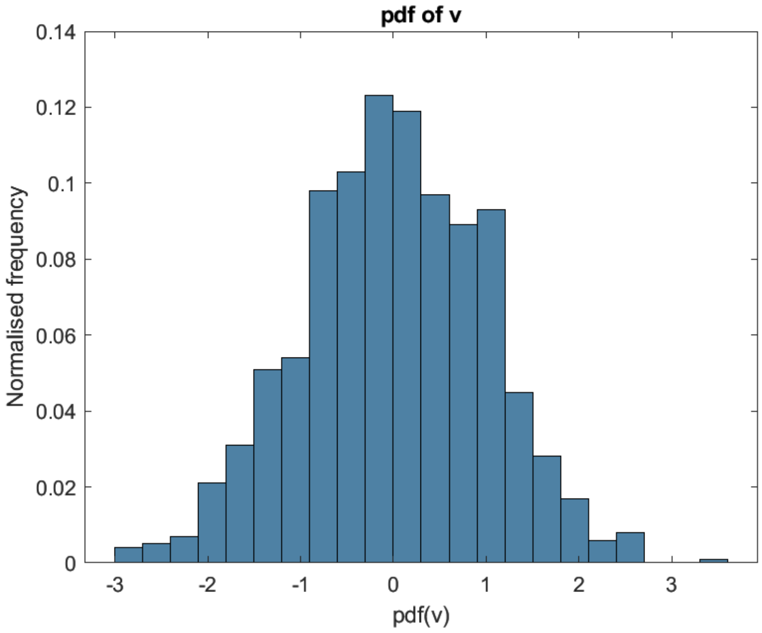


Figure 7: pdf of **v**

1.3.2

The random processes that are stationary and ergodic are **v2** and **v3**. The histogram plots for $N \in \{100, 1000, 10000\}$ are shown in Figure 8. As N increases, the histogram plots show that the pdf of the processes become closer to their theoretical pdf's - Uniform.

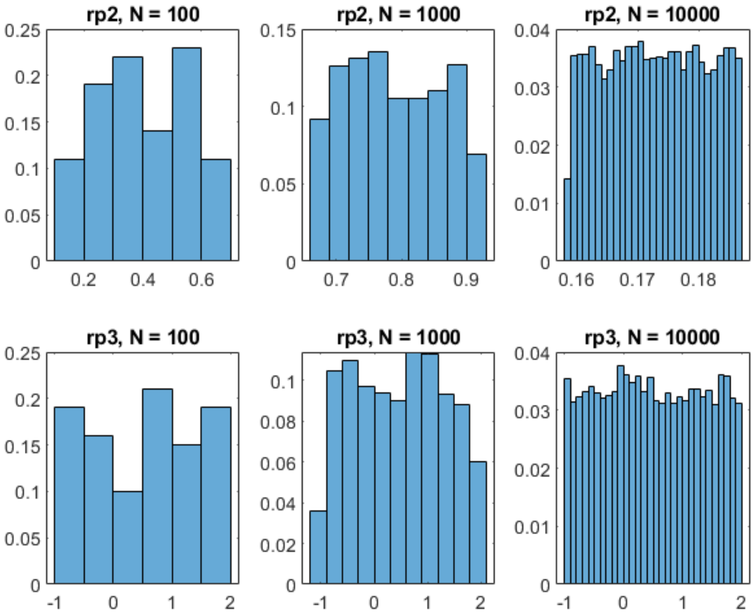


Figure 8: **v2** and **v3** random process histogram plots for different N.

1.3.3

It is not possible to estimate the pdf of a nonstationary process with the pdf function because it assumes a constant mean.

To compute the correct pdf, it will require the joining of 2 seperate pdf calculations. As the mean changes after N = 500, we can calculate the pdf of the process from N = 1:500, then seperately for N = 501:1000. There are various methods to calculate an overall pdf, for example summing the 2 pdf's together.

2 Linear stochastic modelling

2.1 ACF of uncorrelated and correlated sequences

2.1.1

Figure 8 shows the ACF of WGN, denoted by **x**. The figure shows that the ACF is symmetric about 0 (τ = 1000) in the figure where it has a peak. This is because for WGN, there is no information gained from past signal values on the one. However, there are increasing fluctuations around $|\tau| < 50$.

This is due to using a finite number of samples, whereas the theory has $N \to \infty$.

There is symmetry about $\tau = 1000$ as it is an even function.

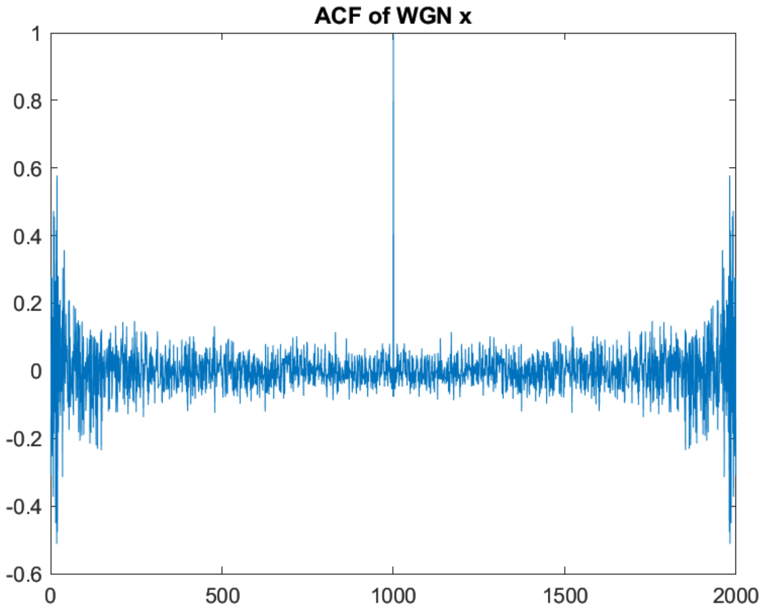


Figure 9: pdf of $\vec{v}$

2.1.2

Figure 9 shows the ACF of $\mathbf{x}$, with $|\tau| < 50$. There are increased fluctuations around this period which suggests that there is some correlation between past and present time lags. We expect a peak at $\tau = 0$ and $\tau = 2000$ as this is where the WGN function is repeated (N = 1000), but these fluctuations prior to the aforementioned τ values should not be present.

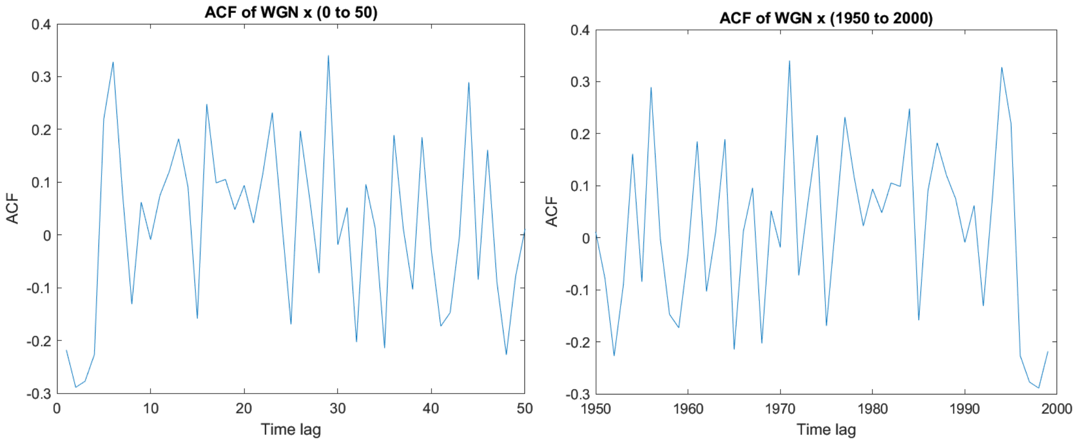


Figure 10: ACF's of WGN with $|\tau| < 50$

2.1.3

τ is defined as $\tau = -N+1, ..., N-1$ according to equation (8). A large $|\tau|$ therefore implies a large N. Equation (8) uses $\frac{1}{N-|\tau|}$ in its calculation. As $N - |\tau|$ is on the denominator, this will increase the result of $\hat{R}_x(\tau)$, resulting in larger ACF values from large $|\tau|$. Therefore, these ACF estimates are not statistically reliable.

An example empirical bound on τ could be 20% of N, which in this case would limit τ to $|N| < 200$.

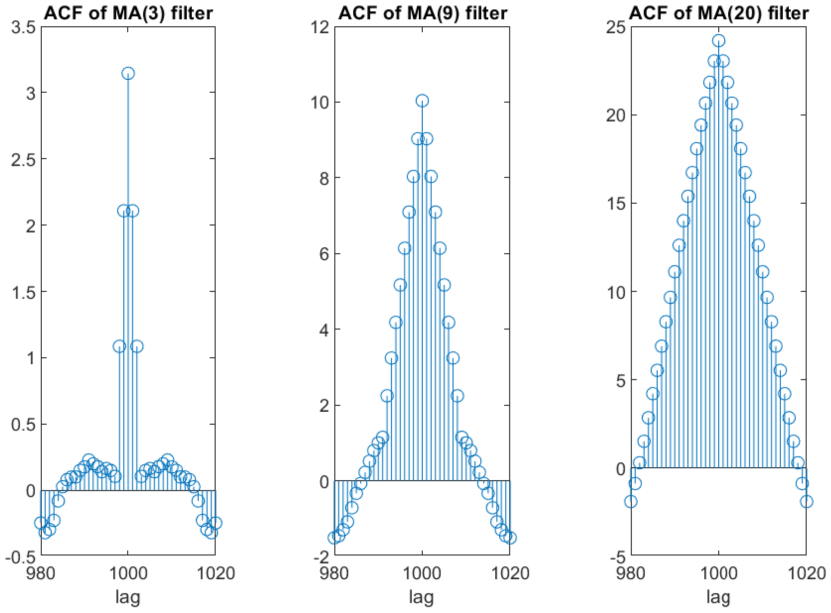
2.1.4

Figure 11 shows the ACF of three moving average (MA) filters with different coefficients, q. The first one has q = 3, second has q=9 and third has q=20. This allows for comparisons to be made between the ACF and MA model order.

As expected, the ACF shows a drop-off for time lags ¿ model order. For example, in the MA(3) case, the ACF value of $\tau = 3$ was 1, whereas the ACF value of $\tau = 4$ was 0.3.

The ACF's also showed consistent gradual decreases from $\tau = 0$, illustrating that the higher order coefficients are not as influential on the model as the lower order coefficients.

The variance of the MA(q) process, $c_0 = (1 + b_1^2 + ...b_q^2)\sigma_w^2$. Using the coefficients found from the ACF calculations, we can find the variance of each MA process. The variance can then be used to calculate the sample mean.



(a) MA(3) (b) MA(9) (c) MA(20)

Figure 11: ACF of different coefficient MA filters

2.1.5

If X_n is an uncorrelated process, then it's ACF will be a Dirac delta function at $\tau = 0$. A convolution operation with a Dirac delta function results in the identity, thus $R_x(\tau) * R_h(\tau) = R_h(\tau) = R_y(\tau)$.

2.2 Cross-correlation function

2.2.1

The figure shows that there is large correlation (value 0.8) between $\mathbf{x}$ and $\mathbf{y}$ for $|\tau| \leq 9$, and a small correlation, value ≈ 0.1, at $|\tau| > 9$. This is because $\mathbf{y}$ is an MA(9) filter of $\mathbf{x}$, and a property of an MA process is that there is a cutoff after q.

Given $R_{XY}(\tau) = h(\tau) * R_X(\tau)$, where X_t is an uncorrelated stochastic process, then $R_{XY}(\tau) = h(\tau)$ due to the reasoning described in Section 2.2.1.

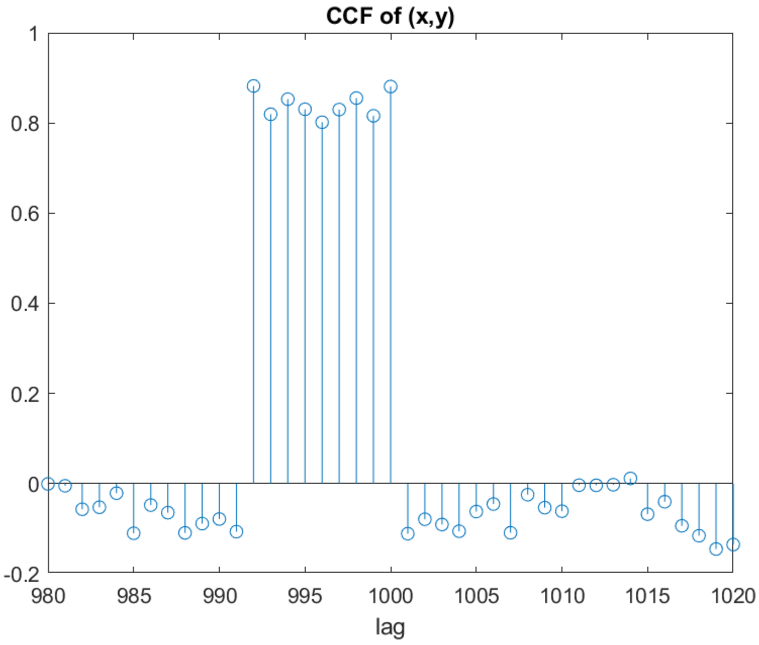


Figure 12: pdf of $\vec{v}$

2.2.2

Let $\epsilon = \mathbf{y} - \mathbf{h}$, the error between the function $\mathbf{h}$, which we are changing, and $\mathbf{y}$, the process we are estimating. This is possible due to X_t being an uncorrelated stochastic process as discussed in section 2.2.1. Thus, we can vary $\mathbf{h}$ seeking to minimise ϵ, which will allow us to identify $\mathbf{y}$ when $\epsilon \approx 0$.

2.3 Autoregressive modelling

2.3.1

The values of the pairs which preserve stability can be defined as follows:
Condition 1: $a_1 + a_2 < 1$.
Condition 2: $a_2 - a_1 < 1$.
Condition 3: $-1 < a_2 < 1$.
The stability conditions can also be visualised using the stability triangle as described in the lectures. Given the definitions of $\mathbf{a}_1$ and $\mathbf{a}_2$, the probability that their values will result in the AR(2) process being outside the stability triangle is high. Thus, the figure shows no patterns in the oscillations of the AR(2) process.
The canvas illustrates the pair values of (a_1, a_2).

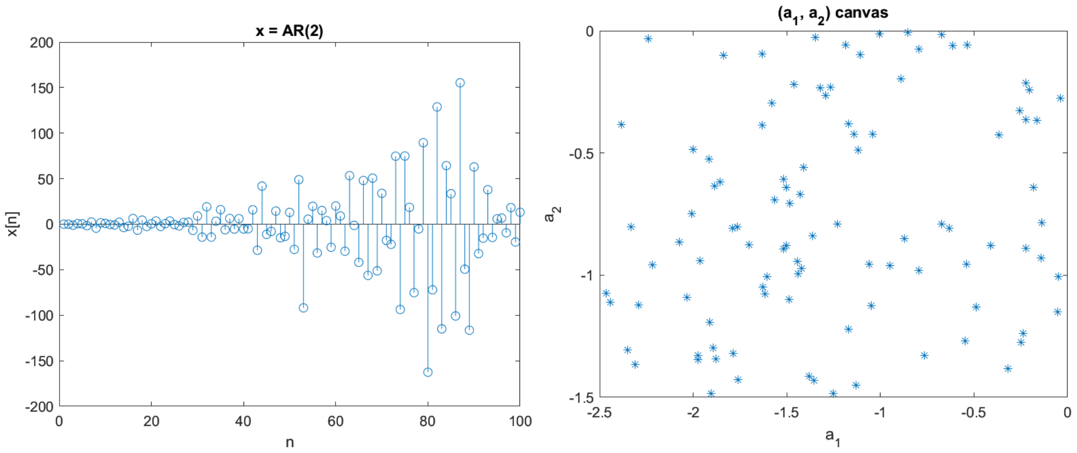


Figure 13: $\mathbf{x} = AR(2)$ and AR coefficients canvas

2.3.2

The ACF for N = 5 shows gradually decreasing values. For N = 20, the values are decreasing and increasing at intervals of 5. For N = 250, the ACF model is similar to N = 20, in the sense that it experiences peaks and troughs.

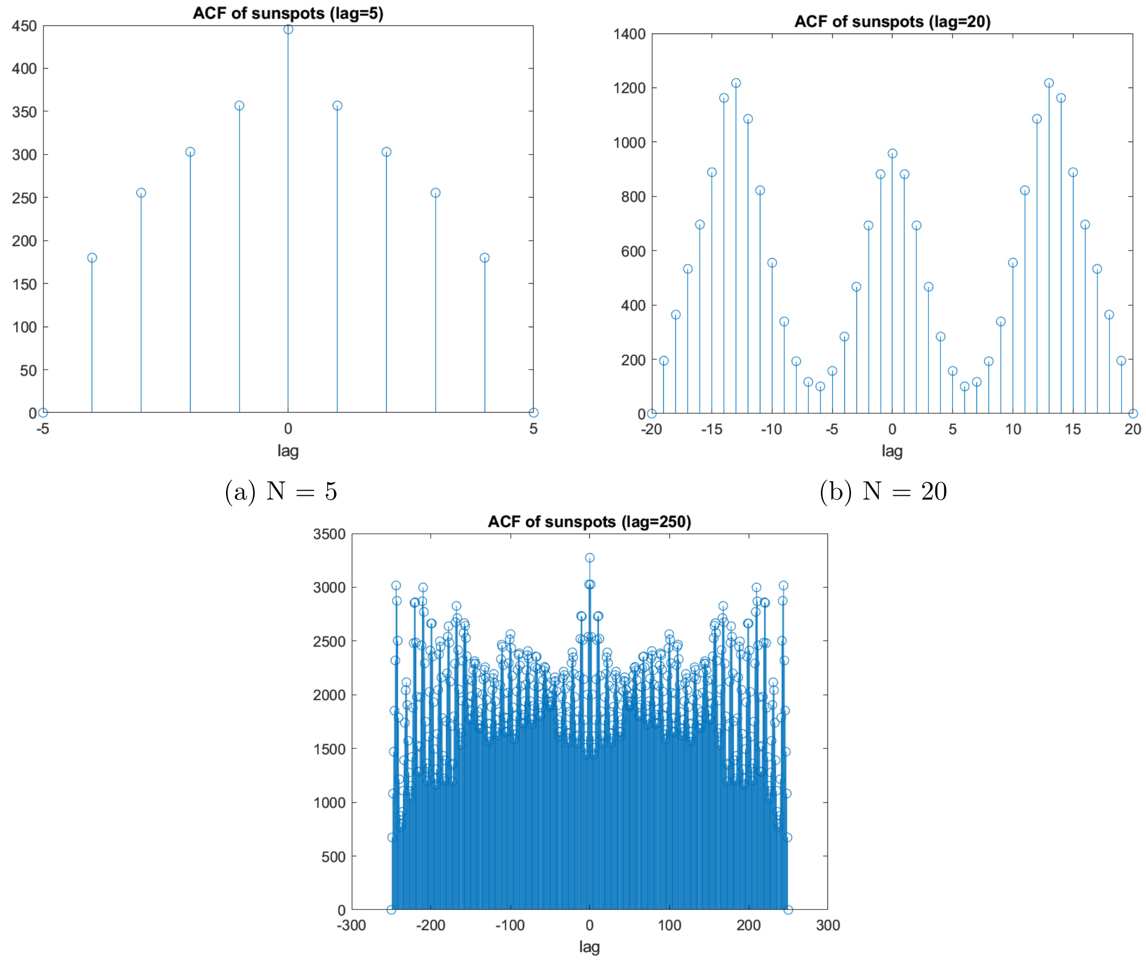
The value ranges of the ACF increase as the data length increases. For example, the means of the 3 data lengths are described in the table. This occurs because the ACF has cumulative correlations between time lags. This is different to the partial-ACF, which removes the relationships of intervening observations.

Data Length	ACF Mean Value
5	293
20	548
250	1960

Table 2: Mean ACF Value's of sunspot numbers at different N

Without zero-meaning the sunspot series, the ACF values are vertically translated such that they are non-negative, as the mean of the sunspot series is high enough to shift the ACF values such that they do not cross the x-axis. The zero-mean sunspot ACF shown in the Figure illustrates that the ACF has positive and negative values with psuedo-periodicity.

However, the vertical translation value is not equal to the mean, as the ACF is influenced by intervening time lags as described earlier.



(a) N = 5

(b) N = 20

(c) N = 250

Figure 14: ACF of sunspots with different data lengths

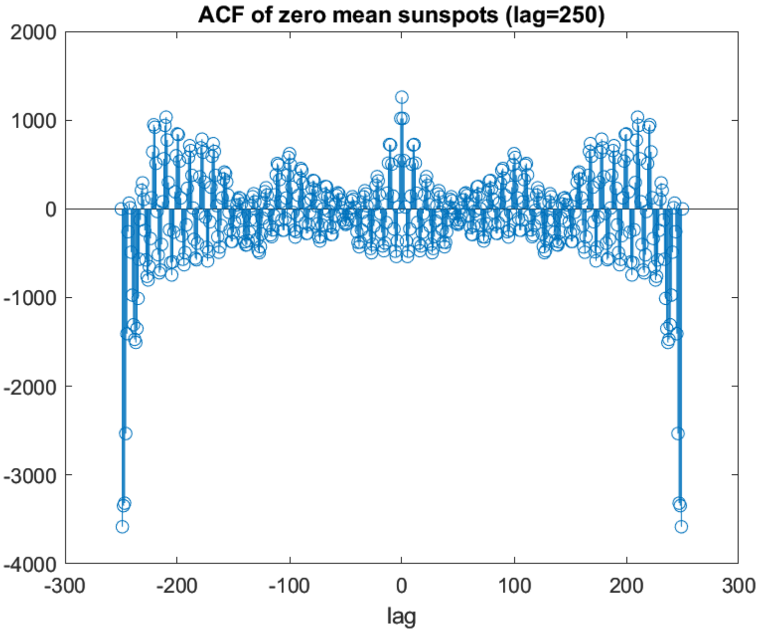


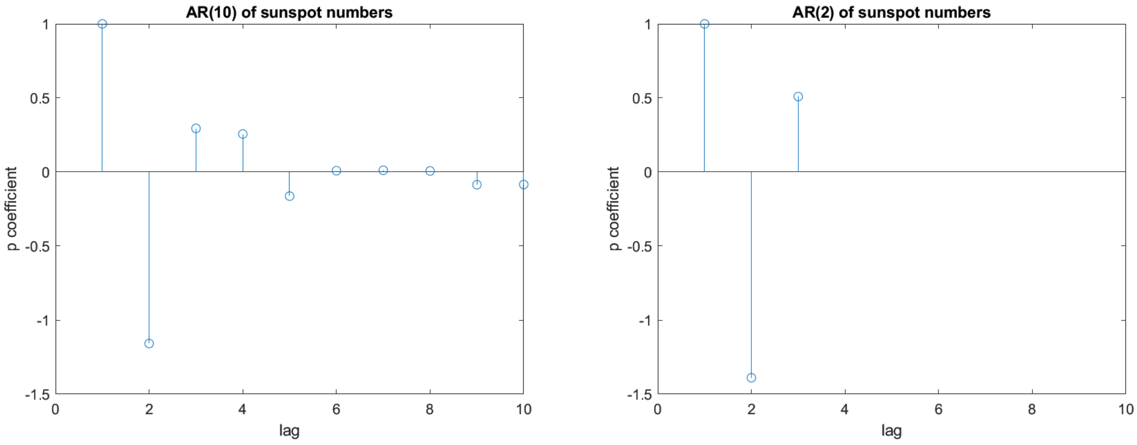
Figure 15: ACF of zero mean sunspots, N = 250

2.3.3

As the *aryule* MATLAB function was used, the value at $\tau = 1$ is always 1. The AR(10) plot shows significant values at $\tau = 1$ and $\tau = 2$. The values beyond $\tau = 3$ are relatively small in comparison. The AR(2) plot follows a similar pattern, thus implying that the sunspot numbers, for N = 250, can be modelled accurately using an AR(2) process.

The other models, AR(3) to AR(9), have not been shown for conciseness but follow a similar pattern.

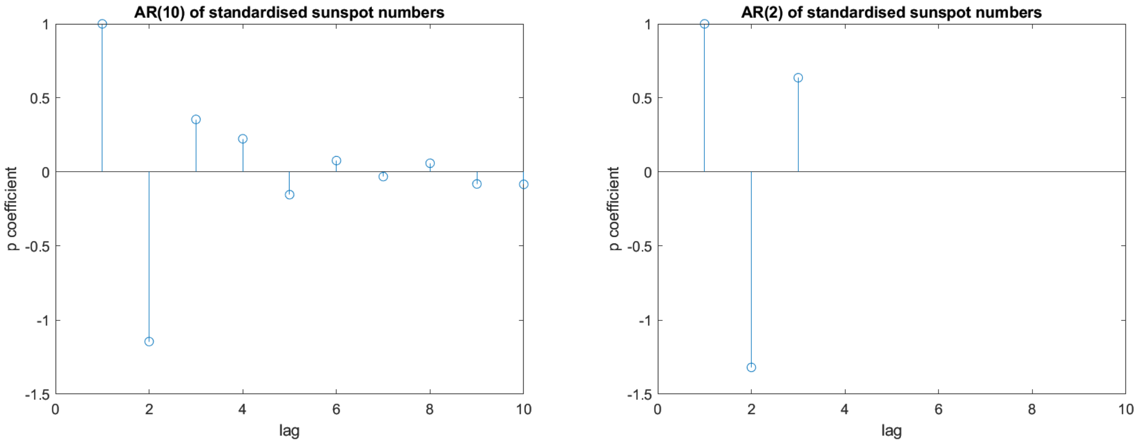
The standardised sunspot number AR plots also showed the same structure and values. This is because the Yule-Walker equations are the same and thus the calculated AR coefficients are the same.



(a) AR(10)

(b) AR(2)

Figure 16: AR(10) and AR(2) of sunspot numbers



(a) AR(10)

(b) AR(2)

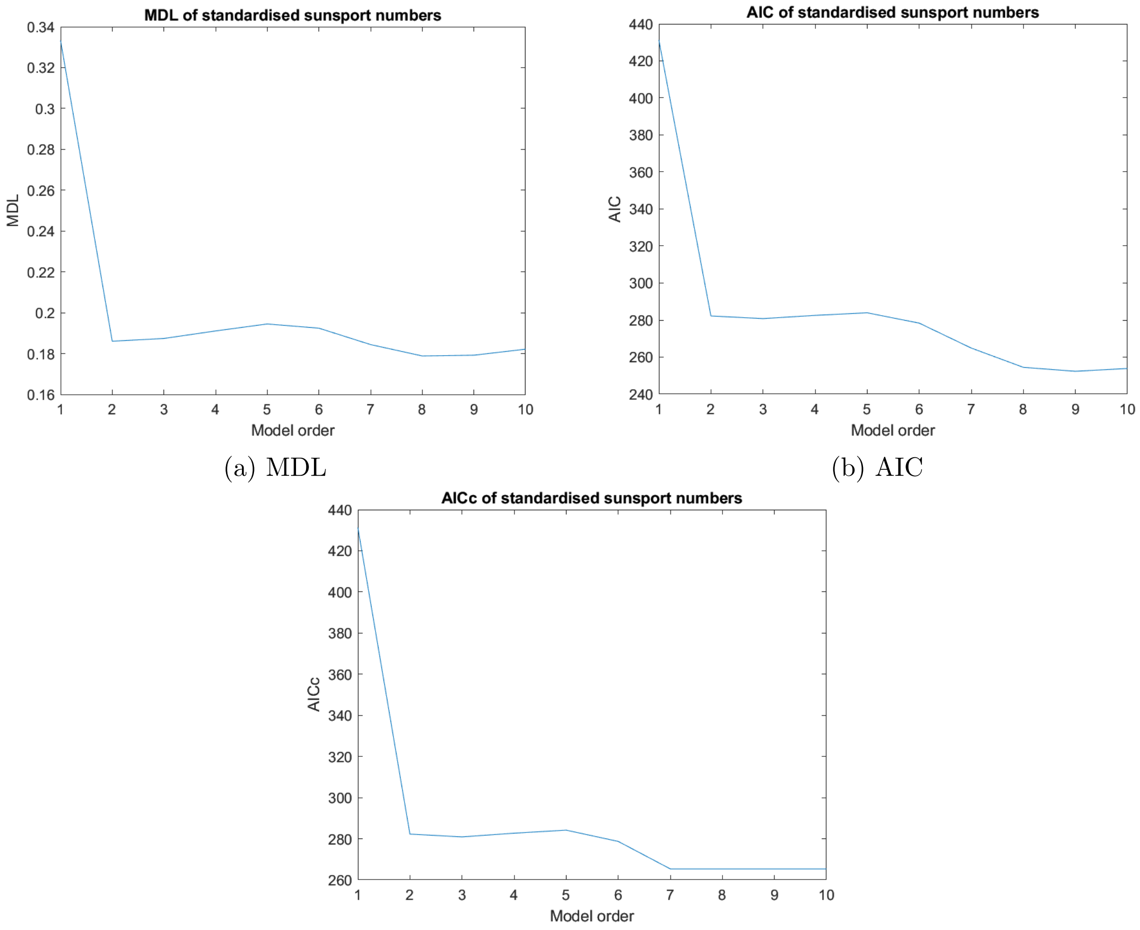
Figure 17: AR(10) and AR(2) of standardised sunspot numbers

2.3.4

The figure shows the MDL, AIC and AICc of the standardised sunspot numbers. They all exhibit a significant drop from 1 to 2, for example the MDL decreasing by nearly 50% and the AIC by nearly 40%.

They also show another, smaller drop from model order 5 to 7 which suggests that an AR(7) model order may also be beneficial, however this could lead to overfitting and an increase in computational complexity that is not worth it for the likely additional gain in accuracy.

Thus, a model order of 2 is likely to be optimal given these criterion.



(a) MDL

(b) AIC

(c) AICc

Figure 18: MDL, AIC and AICc of standardised sunspot numbers

2.3.5

The AR(10) model is overfitting as it is fitting its statistical properties to the noise, rather than the underlying data. This is evident as the predicted values for the AR(10) model have a higher prediction error than the AR(2) model.
The AR(1) model is underfitting as it doesn't have enough degrees of freedom to accurately model the sunspot numbers, thus it also suffers from prediction error.
On the other hand, the model error in AR(10) is low as it has high degrees of freedom (10 coefficients) to accurately model the data.
Overall, the AR(2) model has the best trade-off between model and prediction error.

2.4 Cramer-Rao Lower Bound

2.4.1 Part a

The MDL and AIC of the AR processes (using Yule-Walker method) is shown in the figures. The plots show an increasing MDL and AIC compared to model order. Thus, an AR(1) model is sufficient in modelling the daily returns of the index.

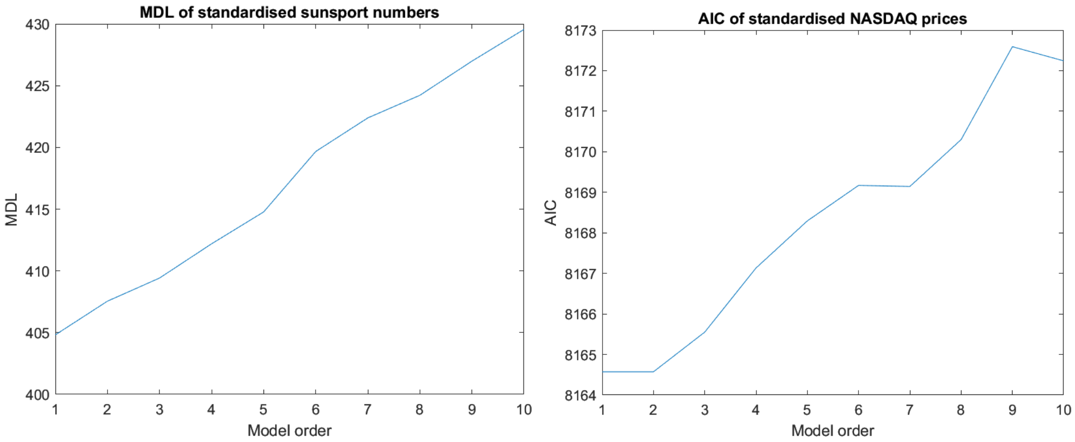


Figure 19: MDL and AIC of AR processes of standardised NASDAQ Prices

2.4.2 Part b

We know the driving variance of an AR(1) model to be 1 due to the White Gaussian Noise property.
First, we calculate the power spectrum of the AR(2) model using *pyulear* method in MATLAB: $P_X(f;\boldsymbol{\theta})$.
We then take the logarithm: $ln[P_X(f;\boldsymbol{\theta})]$.
We then take the partial derivatives: $\frac{\partial ln[P_X(f;\boldsymbol{\theta})]}{\partial \theta_i} = \frac{\partial ln[P_X(f;\boldsymbol{\theta})]}{\partial a_1} = \frac{e^{-2\pi i f}}{\hat{a}_1 e^{-2\pi i f}} - \frac{e^{2\pi i f}}{\hat{a}_1 e^{2\pi i f}}$.
Similarly, $\frac{\partial ln[P_X(f;\boldsymbol{\theta})]}{\partial \sigma^2} = \frac{2\sigma}{\sigma^2} = \frac{2}{\sigma}$
Thus, $\boldsymbol{I}(\boldsymbol{\theta})_{22} = \frac{2}{\sigma}(\frac{e^{-2\pi i f}}{\hat{a}_1 e^{-2\pi i f}}) - \frac{e^{2\pi i f}}{\hat{a}_1 e^{2\pi i f}})$
Thus, $\boldsymbol{I}(\boldsymbol{\theta}) = \begin{bmatrix} \frac{Nr_{xx}(0)}{\sigma^2} & 0 \\ 0 & \boldsymbol{I}(\boldsymbol{\theta})_{22} \end{bmatrix}$

2.4.3 Part c

From the lecture notes, the minimum variance of an unbiased estimator, $\hat{\theta}$ can be defined as: $var(\hat{\theta}) \geq \frac{1}{\frac{-E[\partial^2 lnp(\boldsymbol{x};\theta)}{\partial \theta^2}}$.
Linking this definition back to part b, we have already calculated $\frac{-E[\partial^2 lnp(\boldsymbol{x};\theta)}{\partial \theta^2}$, it is the Fisher Information Matrix approximation.
First solving for $var(\hat{\sigma}^2)$:
We use the $\boldsymbol{I}(\boldsymbol{\theta})_{11}$ part of the Fisher Information matrix, which equals $\frac{Nr_{xx}(0)}{\sigma^2}$.
$r_{xx}(0) = \frac{1}{2\sigma^2}$. Therefore, $var(\hat{\sigma}^2) \geq \frac{2\sigma^4}{N}$.
We then solve for $\delta(\hat{a}_1)$:
We use the $\boldsymbol{I}(\boldsymbol{\theta})_{11}$ part of the Fisher Information matrix, which equals $\frac{e^{-2\pi i f}}{\hat{a}_1 e^{-2\pi i f}} - \frac{e^{2\pi i f}}{\hat{a}_1 e^{2\pi i f}}$.

The following tables show the CRLB against various data lengths. For the $\hat{\sigma}^2$ estimator, the noise variance σ was also altered. As expected, σ^2 has a much higher impact on the CRLB than data length due to it being squared.

	1	50	1001
1	2	5000	2004002
50	0.04	100	40080
1001	0.002	1960	2002

Table 3: CRLB of different σ^2 and N. Horiztonal = σ^2, Vertical = N

From the MATLAB code and *aryule* output information, $a_1 = 1$, which we can then set to calculate the $\hat{a_1}$ CRLB for varying N.

2.4.4 Part d

We wish to calculate $var((PX(f;\boldsymbol{theta}))$:
First, we find the inverse of the Fisher Information Matrix, which was calculated in Part b.

2.5 Real world signals: ECG from iAmp experiment

2.5.1 Part a

The figures show the probability density estimate's (PDE's) of the original RRI1 data and various smoothing functions.

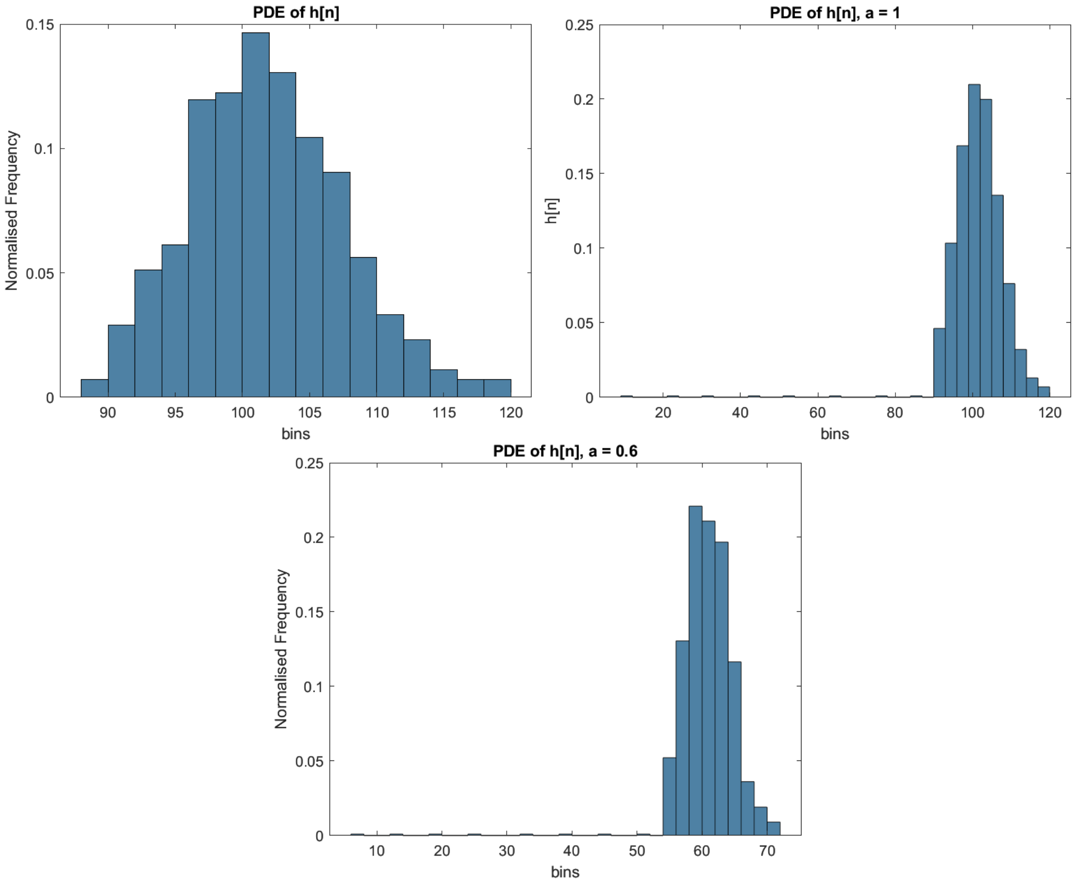


Figure 20: RRI1 PDE estimates of different smoothing rates

2.5.2 Part b

The figures shown in Part a of this section show that all 3 functions have a Guassian distribution, denoted by the bell-curve like shape.
However, without a smoothing function (i.e. the original heart rate), the variance is significantly higher than with a smoothing function as the data has greater fluctuations, and so the frequency in each bin is higher. On the other hand, the smoothing functions result in less variance, and so the frequencies are concentrated on a smaller number of bins, namely those centred around 100 for $\alpha = 1$ and 60 for$\alpha = 0.6$.
The constants therefore result in shifting the mean value while keeping the variance equal.

2.5.3 Part c

The figures show the ACF plots of the 3 RRI data trials, centred to zero mean. RRI1 and RRI2 ACF's exhibit similar patterns in that they have peaks near $\tau = 0$ (denoted by lag = 1000 in the figures). These peaks in the data infer that they can be modelled by AR processes.

However, the RRI3 ACF is considerably different to the ACF's of RRI1 and RRI2 as it doesn't have any peaks as it is oscilliating around an ACF value of 0, thus every lag value has equal weighting in the current value. Thus, this process is likely to be modelled by an MA process.

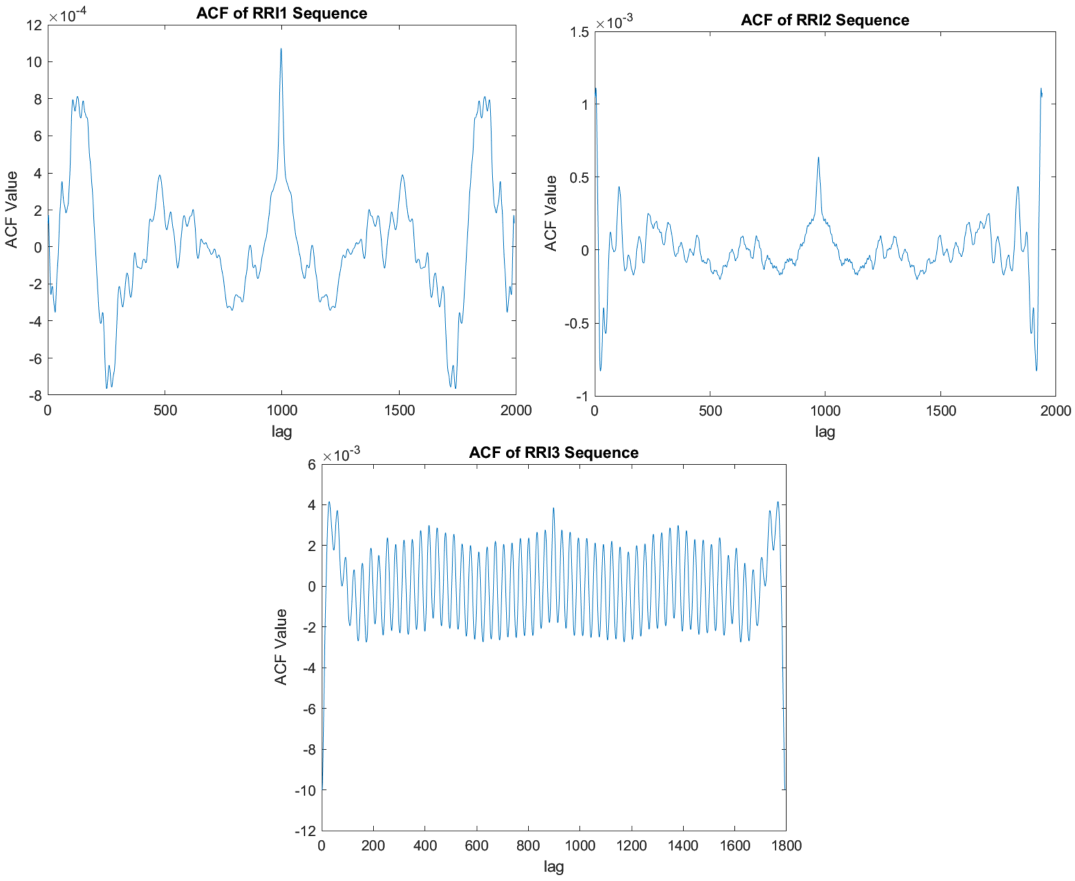


Figure 21: ACFs' of 3 RRI data trials

2.5.4 Part d

The figures show the MDL and AIC of AR processes of the RRI1 and RRI3 data trials. From them, it can be inferred that an optimal model order for the RRI1 data trial is 3 as this combines both a drop in AIC and MDL values as well as limiting the computational complexity and overfitting from a higher order model.

Both trials share similar plots, and for this reason a suitable model order for RRI3 could also be 3. Although not shown in the figures, the optimal model order according to the criteria as described earlier for RRI2 data trial is also 3.

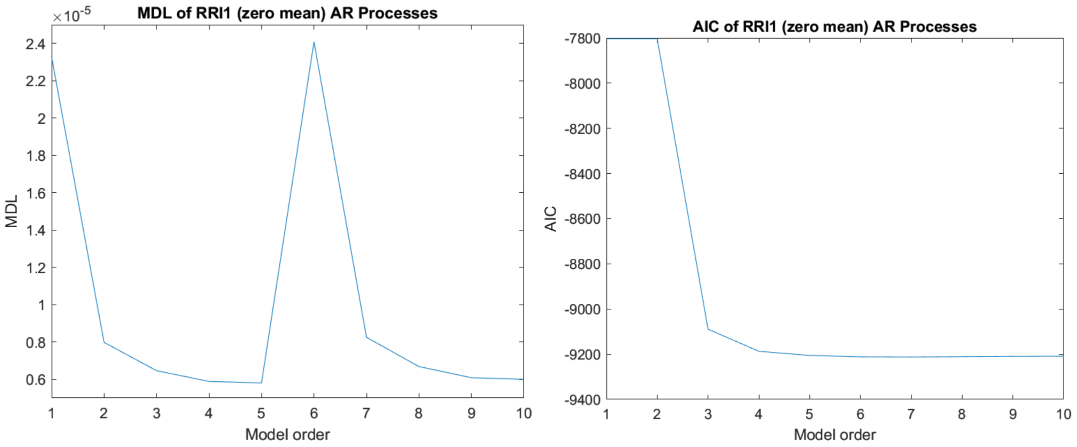


Figure 22: MDL and AIC of RRI1 data trial AR processes

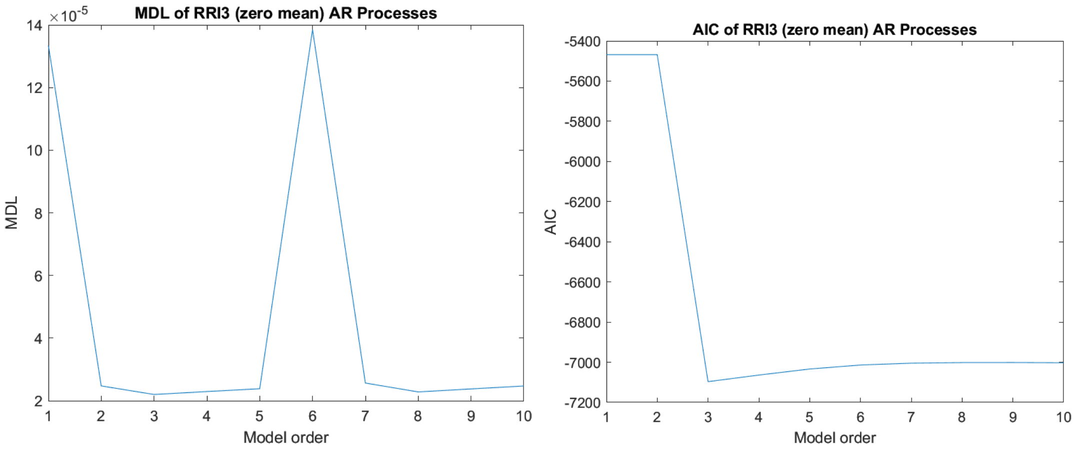
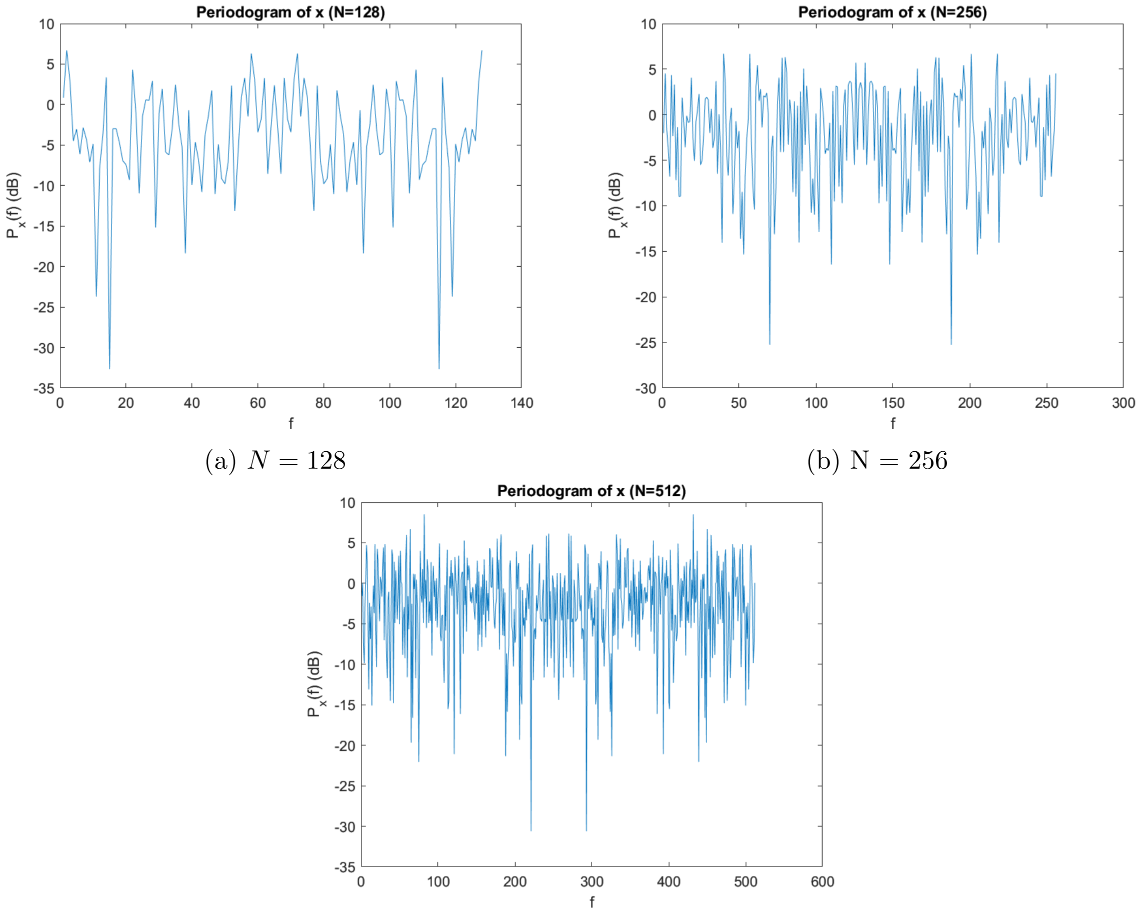


Figure 23: MDL and AIC of RRI3 data trial AR processes

3 Spectral estimation and modelling

The periodogram plot's show a roughly constant PSD of 0dB. As **x** is a WGN process, it has constant power at all frequencies, centered around 0d.

However, the plots show considerable fluctuations. For example, some $P_x(f)$ values were less than -30dB. The variance of the N = 512 periodogram was found to be 25dB.



(a) $N = 128$

(b) N = 256

(c) N = 512

Figure 24: Periodogram Plots of WGN with varying lengths

3.1 Averaged periodogram estimates

3.1.1

The FIR filter acts as a moving average filter, which uses a sliding window size of size 5 to compute the average of the data. Thus, it acts to reduce fluctuations in the data as larger fluctuations are averaged out by the other values. This is evident in the plot, where the smoothed periodogram values lie between 5db and -11dB, compared to 10dB and -35dB in the non-smoothed version.

The variance was calculated to be 4dB, much less than that found in the non-smoothed version.

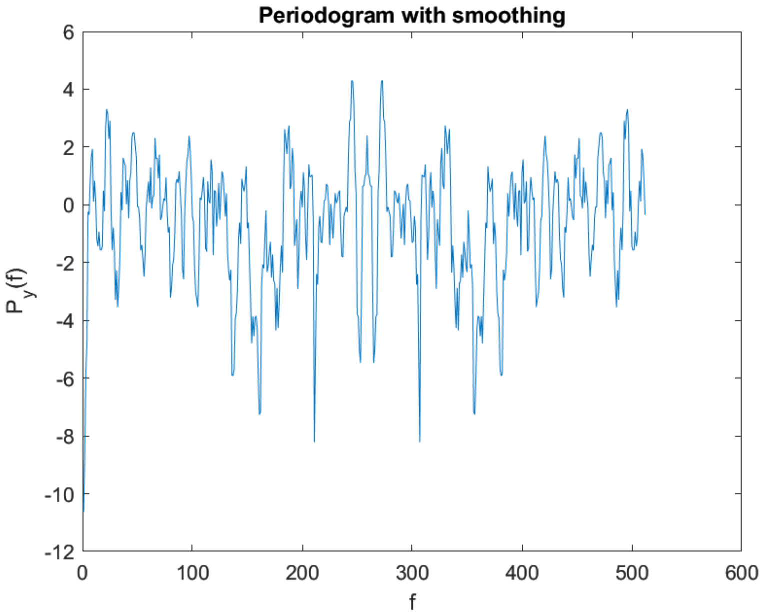


Figure 25: Smoothed Periodogram, N = 512

3.1.2

The figure shows 4 of the 8 segments at intervals of 2, of which can be found in their titles. The PSD segments have the same pattern, centred around 0dB. However, they have considerably different fluctuations at certain frequencies. For example, segment 4 experiences a fluctuation of -30dB for one frequency, whereas segment 2 has values all above -20dB. The 4 segments shown all have variances within 5dB of each other.

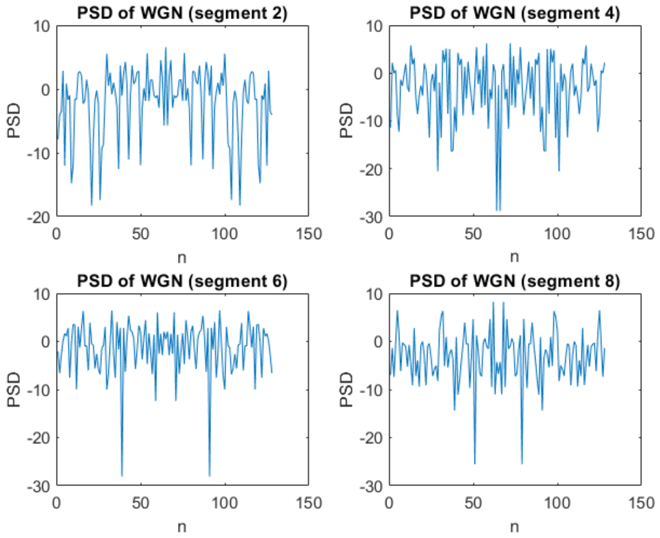


Figure 26: Periodogram Segments, interval of 2.

3.1.3

The mean PSD has much less fluctuations and the values are centred much more around 0dB than the individual segments. For example, the lowest value in the mean PSD is $\approx -6dB$ and highest is $\approx 4dB$.

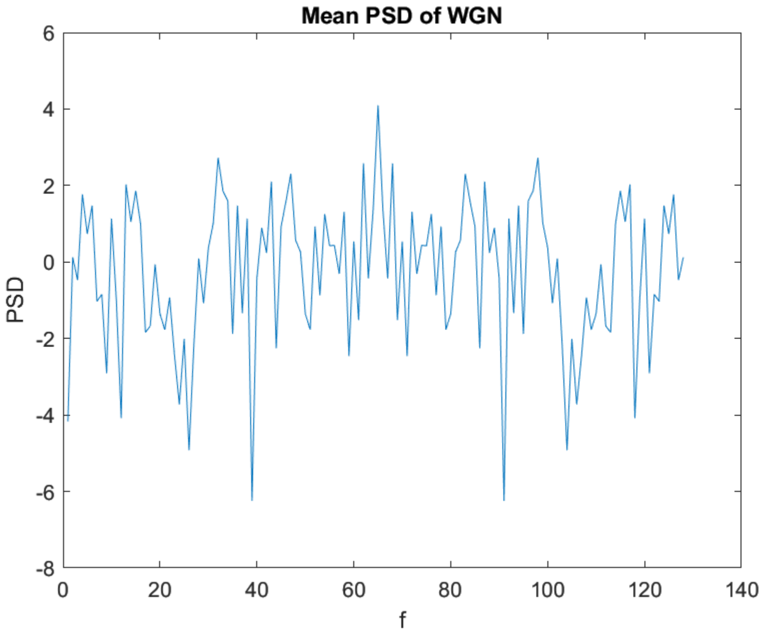


Figure 27: Mean PSD across all segments

3.2 Spectrum of autoregressive processes

3.2.1

The figure shows the exact PSD and $P_y(f)$ of the process $\mathbf{y}$. The AR(1) process is a high pass filter as MATLAB treats the AR coefficients as [1 -**a**]. Thus, [1 0.9] is treated as [1 -0.9]. An AR(1) process is a high pass filter when its' AR coefficient is less than 0.
Cutoff frequency is defined as the frequency at which the output equals 70.7% of the input - when the PSD crosses the -3dB line in the plot.
Thus, the cutoff frequency is at $f_c \approx 2.5$

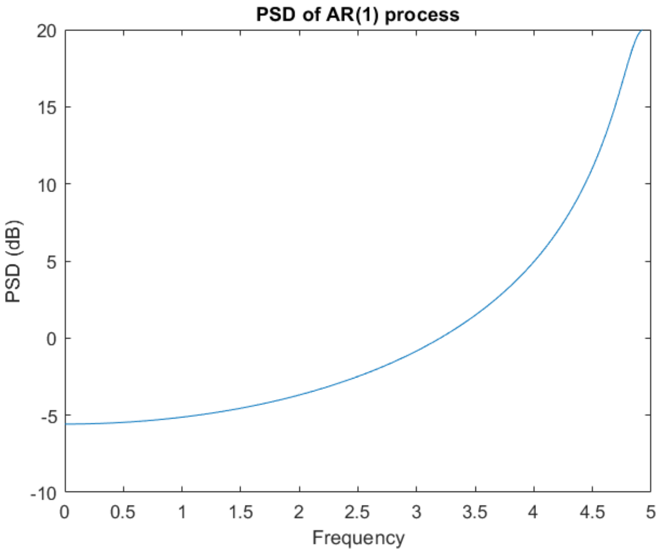


Figure 28: Exact PSD of process $\mathbf{y}$

3.2.2

The periodogram is a fluctuating version of the exact PSD, however many of the periodogram values are below the exact PSD line.

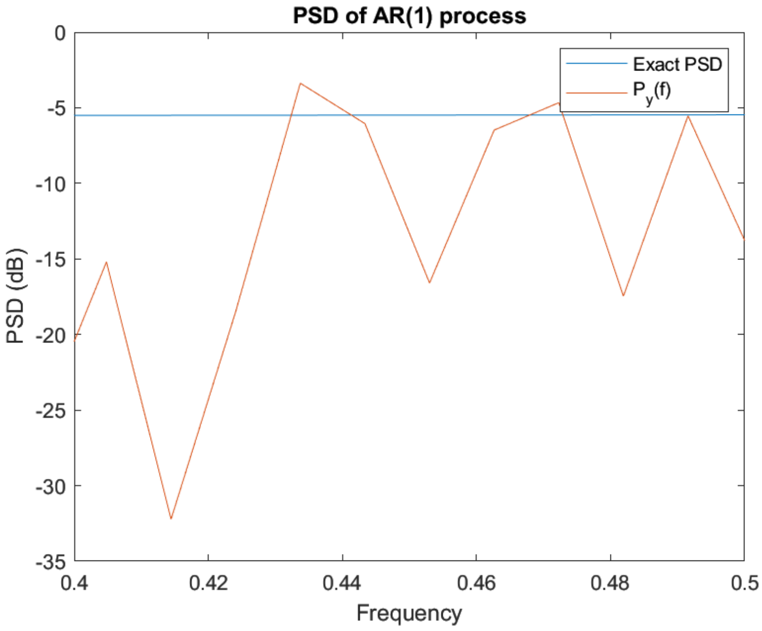


Figure 29: Exact PSD and Periodogram of process $\mathbf{y}$

3.2.3

The plots in the interval $f = [0.4, 0.5]$ show a straight line at ≈ -5dB for the exact PSD, however the $P_y(f)$ shows considerable fluctuations, most of which appear below the -5dB line.

3.2.4

The estimated AR(2) model parameters are:
$\hat{a}_1 = 0.8871$
$\hat{\sigma}_X^2 = 0.97.$
From this, the model based PSD plot is shown in the figure. It is very similar to the exact PSD shown earlier as the model parameter estimation using correlation are a good approximation to the actual parameters used.

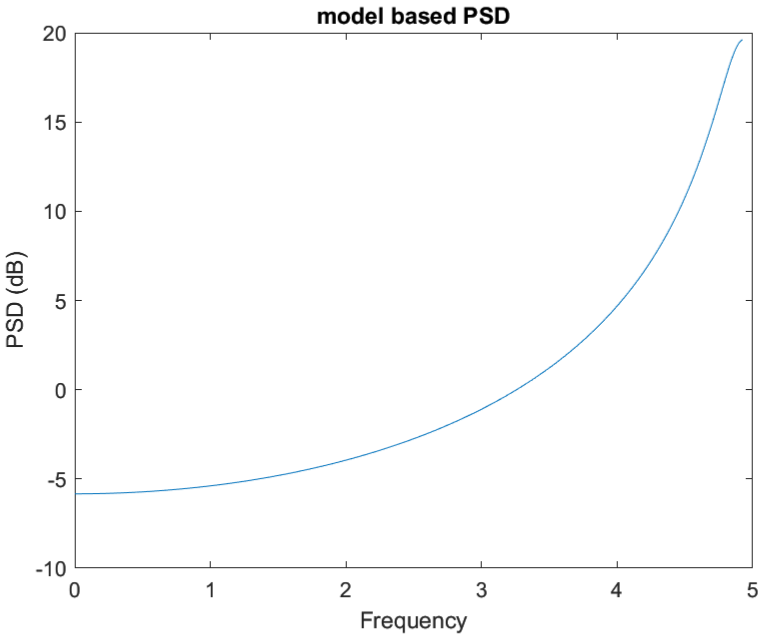


Figure 30: Model based PSD

3.2.5

Under modelling in the sunspot numbers example implies a model order less than 2, as we know this is the optimal order to model the data using an AR process. Under modelling results in the PSD's not having enough enough peaks to model the data accurately. For example, the AR(1) model-based PSD is decreasing from f = 0 whereas the AR(2) model shows a peak. This is shown in the figure.

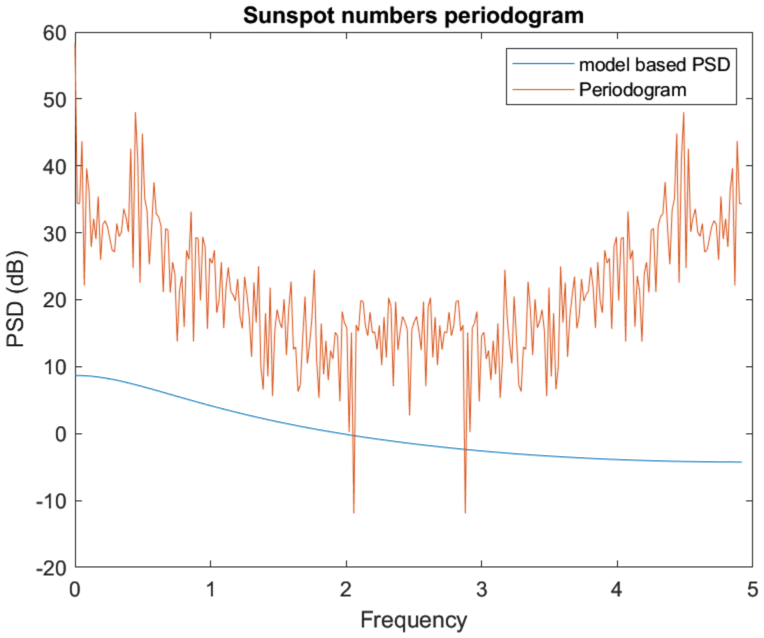


Figure 31: Model based PSD AR(1) and the periodogram

3.3 Least Squares Estimation (LSE) of AR Coefficients

3.3.1

Let $\hat{r}_{xx}[k] = x$ and $\mathbf{H} = \Sigma_{i=1}^{p} \hat{r}_{xx}[k-i]$. Then, $J = (x - \mathbf{Ha})^T(x - \mathbf{Ha})$.
The signal model, $\mathbf{s} = \Sigma_{i=1}^{p} = a_i x[n-i]$. $\mathbf{H} = \Sigma_{i=1}^{p} = x[n-1]$
The LSE can be found by minimising the cost function, i.e. finding the $\mathbf{a}$ coefficients where $\frac{\partial J(\mathbf{a})}{\partial \mathbf{a}} = 0$:
$J(\mathbf{a}) = x^T x - 2x^T \mathbf{Ha} + \mathbf{aH}^T\mathbf{Ha}$.
Therefore, $\frac{\partial J(\mathbf{a})}{\partial \mathbf{a}} = -2\mathbf{H}^T x + 2\mathbf{H}^T\mathbf{Ha} = 0$.
The LSE estimator is then $\hat{\mathbf{a}} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T x$.
The $\boldsymbol{a}$ coefficients for different model orders are shown in the table.

AR model order	$\boldsymbol{a}$
1	[1, -0.83]
2	[1,-1.38, 0.68]
10	[1, -1.218, 0.4843, 0.1207, -0.1655, 0.1502, -0.04028, 0.0312, -0.08247, -0.0822, -0.03904]

Table 4: AR coefficients of various model orders using LS method

3.3.2

H is deterministic as it only contains a deterministic component of $\hat{r}_{xx}[k]$. The cost function seeks to minimise the mean squared error of the measured signal. In that sense, **Ha** is the non-noise induced part of the measured signal **x** from (31).

3.3.3

The figure shows the AR(2) and AR(10) models of the sunspot numbers using the LSE method.

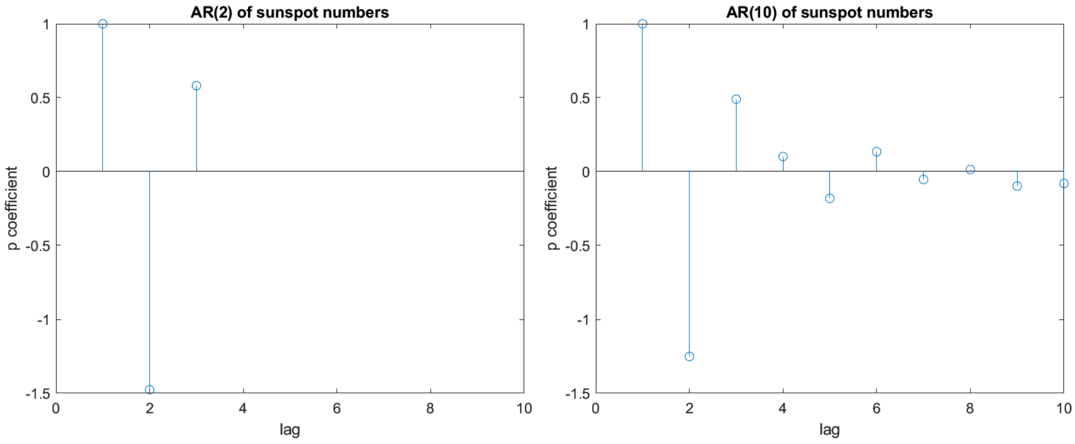


Figure 32: AR(2) and AR(10) using LSE method

3.3.4

The optimal model order is AR(2) as this has the best trade-off between model complexity and MSE reduction. The bar chart shows a big drop of MSE between the AR(1) model and AR(2). While the MSE continues to reduce (as expected), the reductions are small in comparison to the drop from AR(1) and AR(2). For example, the drop from 1 to 2 was $\approx$ 200, whereas the drop from 2 to 3 was $\approx$ 5 This is the same result obtained from the analysis conducted in section 2.3.

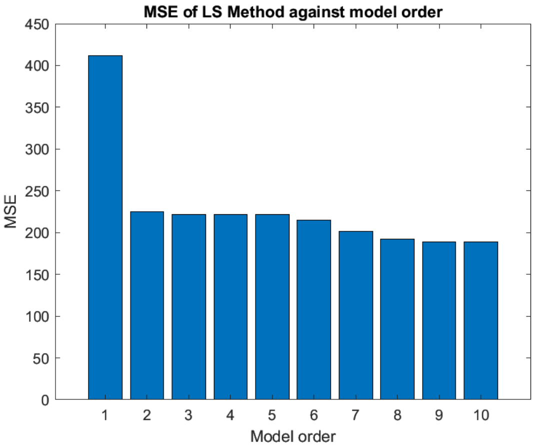


Figure 33: MSE against model order

3.3.5

The AR(1), AR(2) and AR(10) power spectrum's are shown in the figures. The model order and power spectrum are relation as shown in the shape of the plots - a higher model order results in more changes of gradients, and therefore more peaks, due to the AR process corresponding to poles in a transfer function.

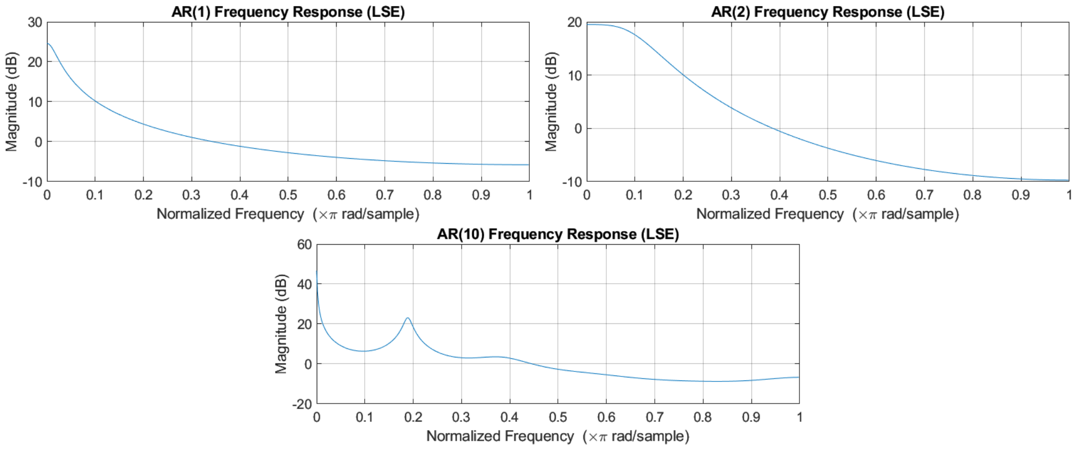


Figure 34: AR(1), AR(2) and AR(10) PSD (LSE method)

3.3.6

A table showing the MSE against sunspot data lengths is shown for an AR(2) process.

Data Length	MSE
5	$9.3e^{-25}$
10	$5.7e^{-26}$
250	293

Table 5: MSE of AR(2) process on different data lengths

3.4 Spectogram for time-frequency analysis: dial tone pad

3.4.1

The figure shows the different time responses for each key symbol as an overview.
The generated discrete time sequence $\boldsymbol{y}$ is shown in the figure.
The sampling rate is appropriate as it is much higher than the Nyquist rate, which is the minimum sampling frequency to ensure no information loss.

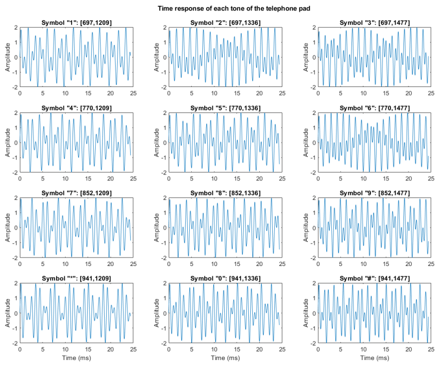


Figure 35: Time response of different key symbols

3.4.2

The spectral components of $\boldsymbol{y}$ are shown in the figure. It shows the power of the signal over time at various frequencies. As expected, the frequencies used in the DTMF show the highest signal power. The idle time corresponds to no power.
The FFT segments plots are also shown in a separate figure. They show the frequencies used in DTMF as peaks.

3.4.3

It is possible to identify a sequence generated by a key press.
We can use the Fourier Transform (FT) to convert the key press from time domain to frequency domain.
The specific key press frequency from the FT will show as a spike in a frequency plot. As we know the specific frequencies used in DTMF and their combinations, given a key press, we expect a spike in those 2 frequencies. For example, a key press of 1 will cause a spike at $f_1 = 697$ and at $f_2 = 1209$. These spikes together will then be classified as a key press = 1. We can set a threshold for this spike to be considered a legitimate key press in the presence of noise.

3.4.4

As more noise was added, the frequency spectrum became less distinguishable in terms of the frequency spikes associated with the DTMF as discussed in part 3. White noise has the property that it is made up of all frequencies by an equal amount. This will make it harder for key classification as the threshold for the classifier may be crossed even when keys aren't pressed, i.e. from just the noise.

3.5 Real world signals: RSA from RRI-Intervals

3.5.1 Part a

The figures below show the periodogram's of the RRI data with different window lengths.

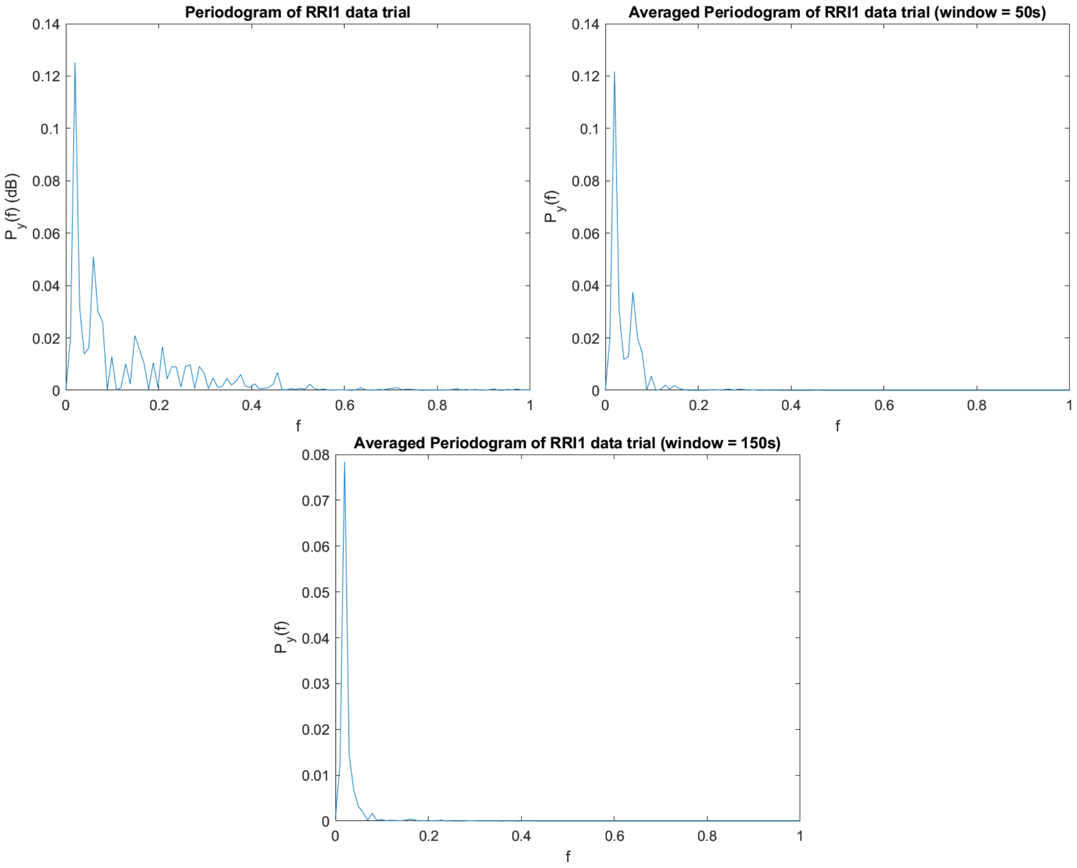


Figure 36: PSD of RRI1 data trial with different window lengths

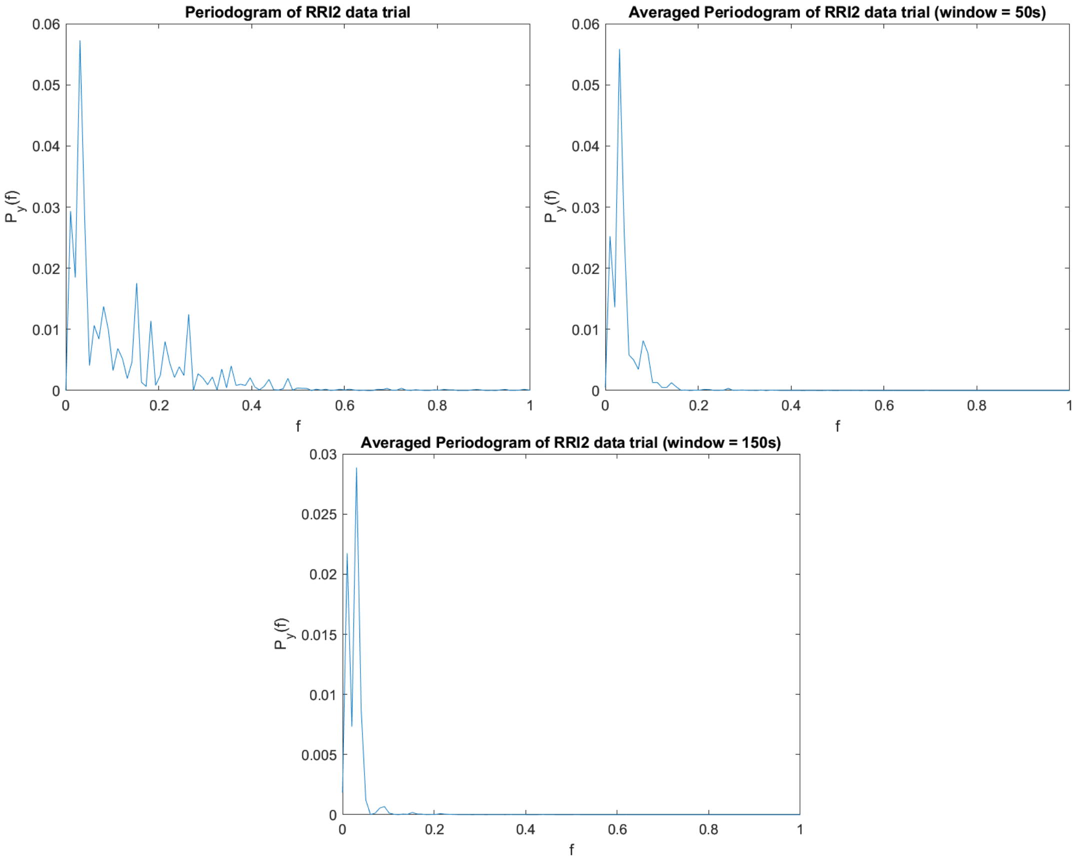


Figure 37: PSD of RRI2 data trial with different window lengths

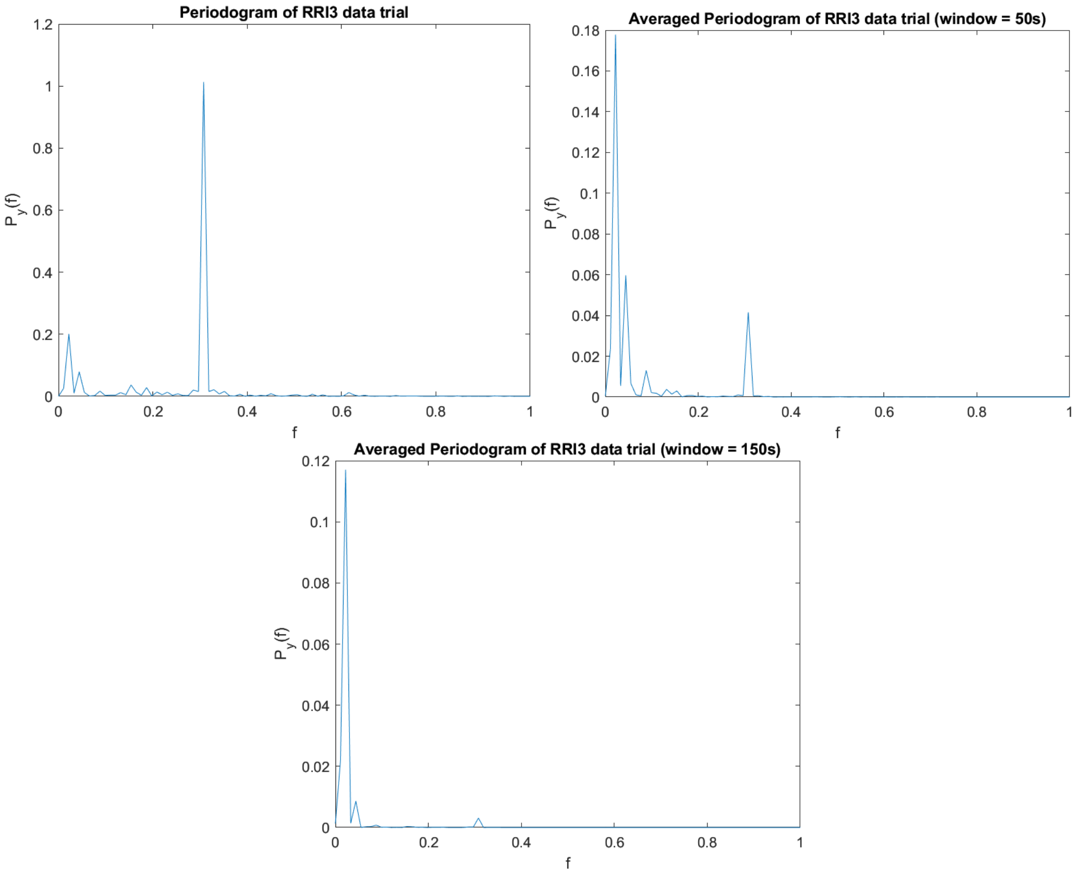


Figure 38: PSD of RRI3 data trial with different window lengths

3.5.2 Part b

The PSD data of the RRI1 and RRI2 data trial are relatively similar in their peak location, however RRI1's peak PSD is approximately double that of RRI2.
The location of the RRI3 data trial peak PSD is very different to RRI1 and RRI2 as it is centred around 0.3, whereas the other 2 trials' peak is very close to 0. This is due to the different breathing rate of RRI3 compared to the other 2 trials.
The peak PSD corresponds to the heart rate of the data trial.

4 Optimal filtering - fixed and adaptive

4.1 Wiener filter

4.1.1

The $\mathbf{R}_{xx}$ statistic:

$$\begin{bmatrix} -0.7116 & -1.1235 & 4.8861 & -3.0658 & 1.4355 \\ -1.1235 & -0.7116 & -1.1235 & 4.8861 & -3.0658 \\ 4.8861 & -1.1235 & -0.7116 & -1.1235 & 4.8861 \\ -3.0658 & 4.8861 & -1.1235 & -0.7116 & -1.1235 \\ 1.4355 & -3.0658 & 4.8861 & -1.1235 & -0.7116 \end{bmatrix}$$

The $\mathbf{R}_{xx}$ is symmetric and square due to the toeplitz method to ensure matrix multiplication compatibility. The $\mathbf{p}_{zx}$ statistic:

$$\begin{bmatrix} -0.0409 \\ -0.2312 \\ -0.2411 \\ -0.6019 \\ 1.7198 \end{bmatrix}$$

And finally the calculated $\mathbf{w}_{opt}$:

$$\begin{bmatrix} 0.7303 \\ 0.4770 \\ 0.3049 \\ -0.1547 \\ -0.6611 \end{bmatrix}$$

The $\mathbf{w}_{opt}$ is similar to the unknown system in the sense that they are within the unknown filter coefficients by -2.7 in the worst case (3-0.3) and 0.27 in the best case (1-0.73).

4.1.2

The table shows the SNR and Weiner solution coefficients as a result of changes in the variance of the additive noise. As variance increases, the SNR decreases. The Weiner solution coefficients have values further from the unkown signal coefficients as the variance increases. This is due to the effect of the stochasticity of the noise which cannot be modelled by the adaptive filter.

Variance	SNR (dB)	$\mathbf{w}_{opt}$
0.1	40.0	[0.7967, 0.5619, 0.3603, -0.0773, -0.6032]
0.5	12.3	[0.8246, 0.5027, 0.2816, -0.3054, -0.8601]
1	-0.194	[1.4284, 0.2136, -0.3032, -1.0899, -1.1576]
2	-11.6	[2.6631, -0.0244, -0.5753, -0.2344, 0.7269]
5	-27.67	[-10.3997, -15.9719, -7.4995, -0.6090, 6.9688]
10	-39.69	[-122.0334, -73.4583, 9.7463, 61.4273, 53.5206]

Table 6: SNR (dB) and Weiner solution coefficients of additive noise with different variances

4.1.3

$\mathbf{w}_{opt} = \mathbf{R}_{xx}^{-1}\mathbf{p}_{xx}$.
To calculate $\mathbf{r}_{xx}$ requires $O(N^2)$ calculations using the convolution method.
To convert $\mathbf{r}_{xx}^{-1}$ to $\mathbf{R}_{xx}$ requires copying the calculations to form an N by N matrix.
To then find the inverse of this matrix requires $O(N^3)$ time.
To calculate $\mathbf{p}_{xx}$ requires $O(N^2)$ calculations using the convolution method.
To then multiply $\mathbf{R}_{xx}^{-1}$ by $\mathbf{p}_{xx}$ requires $O(N^3)$ time.
The total time would be the previously mentioned time complexities summed together, however according to Big-O notation rules summations are ignored and the biggest complexity is chosen, thus resulting in a final time complexity of $O(N^3)$.

4.2 Least Mean Square (LMS) algorithm

4.2.1

The $\mathbf{w}$ coefficients will converge to the unknown system filter coefficients by employing instantaneous estimates of the statistics described in Section 4.1.1 of the Weiner filter. The results of the adaptive filter are shown in the Figure. The adaptive filter follows the general pattern of the desired signal. The derivation of the formulation of the LMS method is shown below:
We start from the full statistics steepest descent formulation:
$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu(\mathbf{r}_{dx} - \mathbf{R}\mathbf{w}(n))$.
The instantaneous estimates of the $\mathbf{r}_{dx}$ and $\mathbf{Rw}$ are:
$\hat{\mathbf{R}}(n) = \mathbf{x}(n)\mathbf{x}^T(n)$ and $\hat{\mathbf{r}}_{dx}(n) = d(n)\mathbf{x}(n)$.
Substituting these into the method of steepest descent above:
$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu[d(n)\mathbf{x}(n) - \mathbf{x}(n)\mathbf{x}^T(n)\mathbf{w}(n)$
$= \mathbf{w}(n) + \mu[d(n) - y(n)]\mathbf{x}(n)$
And finally:
$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e(n)\mathbf{x}(n)$.

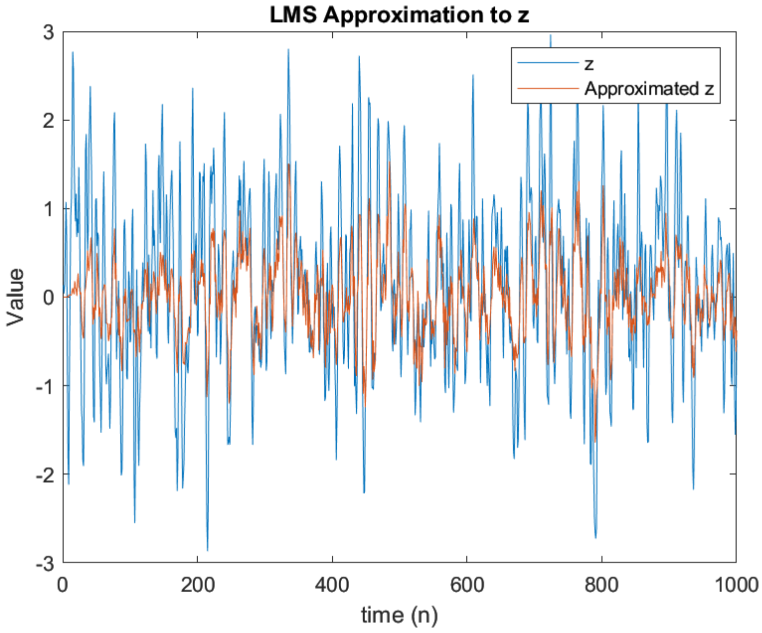


Figure 39: LMS Approximation Plot

4.2.2

The figure shows the squared errors as a result of different adaption gains. The highest adaption gain tested, 0.5, has not been included in the figure as it's error was too large to be plotted with the other gains.

As shown, the higher adaption gains proved to have the highest error's, with the 0.2 gain dominating, reaching a peak value of 15 in some cases, compared to the next highest 0.1 reaching a peak gain of 5.

The worsening performance of the filters is exponential in the adaption gain, as the difference from 0.1 to 0.2 was exponentially worse than from 0.2 to 0.5.

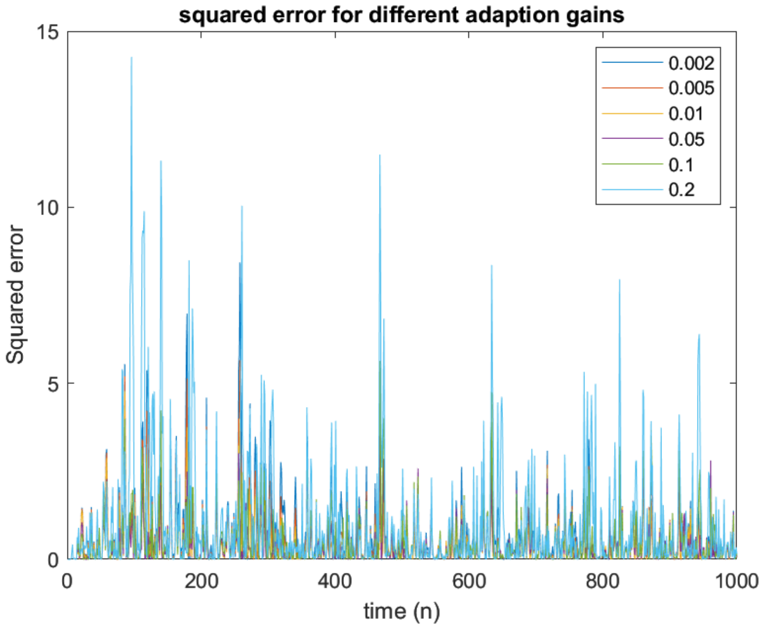


Figure 40: Squared errors for different adaption gains

4.2.3

At each iteration:

$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e[n]\mathbf{x}(n) \rightarrow$ 1 addition, 1 scalar multiplication between $\mu e[n]$ and the vector $\mathbf{x}(n)$. As the order of the adaptive filter has been defined to be N_w, this results in 1

addition and N_w multiplications.
$\hat{y}[n] = \mathbf{w}^T(n)\mathbf{x}(n) \rightarrow O(N_w^2)$ from the property of matrix multiplication.
$e[n] = z[n] - \hat{y}[n] \rightarrow 1$ subtraction of scalar quantities.
From Big-O notation properties, the final time complexity is $O(N_w^2)$.

4.3 Gear shifting

The values of the adaption gain that would reduce a high error would be a high adaption gain, As the estimator tends towards the steady state, i.e. the error is gradually decreasing, then the adaption gain should decrease proportionally.
The adaption gain criteria was chosen such that the adaption gain would halve when e[n] approached within 1/10 of the minimum MSE achieved.

4.4 Identification of AR processes

4.4.1

a_1 and a_2 should converge to 0.9 and 0.2 if the algorithm is successful. This is because the synthesis AR model is as follows: $x[n] = a_1x[n-1] + a_2x[n-2] + w[n]$, where "w" is defined as White Gaussian Noise.
Similarly, the predicted $y[n] = w_1x[n-1] + w_2x[n-2] + wgn$.
Thus, w_1 should converge to $a_1 = 0.9$, while w_2 should converge to $a_2 = 0.2$. The table shows the final weight values along with a plot of $x[n]$ (output) against $y[n]$ (predictor).

4.4.2

The table shows the final a_1 and a_2 for different adaption gains. When the adaption gain increased beyond 0.05, the training became very unstable and the $\boldsymbol{a}$ coefficients were varying heavily between times.
An adaption gain below 0.5 showed much more stable training, as the coefficient variance between iterations was much lower, leading to better convergence.

Adaption gain	$a_1 and a_2$
0.01	[0.016, -0.138]
0.02	[0.36, -0.09]
0.05	[0.37, -0.38]
0.1	[-1.05, 1.21]
0.2	[-4.62, -3.18]
0.5	$[-2.93e^8, -3.15e^8]$

Table 7: MSE of AR(2) process on different data lengths

4.5 Speech recognition

4.5.1

An adaption gain of 0.01 was used initially to test different model orders on prediction performance.

4.5.2

One way to analyse the optimal filter length could be to concatenate the predicted sounds with the input and analyse the results using human intuition - i.e. qualitatively assess by listening to the result.
A quantitative approach could be to analyse model complexity against performance, for example visualising the error of each model against the computational complexity of applying the filter (e.g. CPU cycles per 10 samples). The best model would be dependent on the goal: i.e. trade-off between complexity and performance.

4.5.3

A higher sampling frequency increases the number of data samples. This increases the ability to treat the vowel sounds as quasi-stationary, which allows us to treat the signal as stationary for a certain time period. This increased time period will help the learning curves to converge due to the advantageous statistical properties of stationarity, which was analysed in Section 1.

4.6 Dealing with computational complexity: sign algorithms

The signed-error and signed-regressor algorithms had similar performance with each other but both performed less well compared to the basic LMS algorithm when analysis final a_1 and a_2 coefficients from Part 4.4.
The sign-sign algorithm performed the worst for both Part 4.4 and Part 4.5. This is because it is more simplified than the other algorithms, using 2 signed operations instead of one for the other sign algorithms and 0 for the basic LMS algorithm.