

Advanced Signal Processing Coursework Assignment Report

Kleanthis Marios Papadopoulos

CID:01527402

4 April 2021

Contents

1	Random Signals and Stochastic Processes	3
1.1	Statistical Estimation	3
1.2	Stochastic Process	5
1.3	Estimation of Probability Distributions	8
2	Linear Stochastic Modelling	11
2.1	ACF of Uncorrelated and Correlated Sequences	11
2.2	Cross-Correlation Function	12
2.3	Autoregressive Modelling	13
2.4	Cramer-Rao Lower Bound	15
2.5	ECG from iAmp Experiment	17
3	Spectral Estimation and Modelling	19
3.1	Averaged Periodiogram Estimates	20
3.2	Spectrum of Autoregressive Processes	21
3.3	Least Squares Estimation of AR Coefficients	23
3.4	Spectrogram for Time-Frequency Analysis	25
3.5	Respiratory Sinus Arrhythmia	27
4	Optimal Filtering- Fixed and Adaptive	28
4.1	Wiener Filter	28
4.2	Least Mean Square Algorithm	30
4.3	Gear Shifting	32
4.4	Identification of AR Processes	32
4.5	Speech Recognition	33
4.6	Sign Algorithms	35
5	MLE for the Frequency of a Signal	35
5.1	Mapping $J(\theta)$	35
5.2	Minimising Solution	36
5.3	MLE of f_0	36
5.4	Periodogram and MLE Estimate	37

1 Random Signals and Stochastic Processes

1.1 Statistical Estimation

A uniform random process consisting of 1000 samples stored in a row vector \mathbf{x} is generated using the MATLAB command `rand`. Each sample $x[n]$ has a uniform distribution on the interval $[0,1]$. The plot below indicates the process is stationary, which can be attributed to its time-invariant statistical properties.

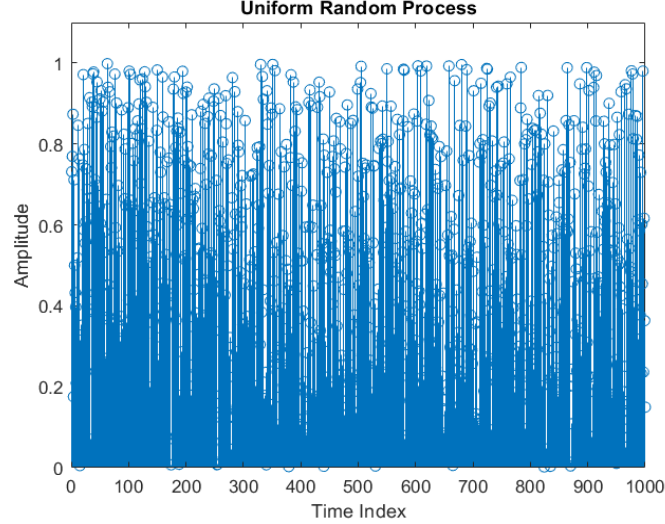


Fig. 1: Uniform random process

1.1.1 Mean

The theoretical mean of a uniform distribution equals the average of the endpoints, which yields an expected value of 0.5 in this case. Using the function `mean`, the *sample mean* of the process can be estimated. The sample mean is an accurate estimator because its estimates have a maximum deviation of less than 0.05, as seen in Table 1.

1.1.2 Standard Deviation

The theoretical standard deviation of a uniform distribution is 0.2887. The function `std` is used to obtain the sample standard deviation estimate. It can be viewed as an accurate estimator given that the difference between sample and theoretical values is negligible, as indicated in Table 1.

Sample Mean	Absolute Sample Mean Error	Sample Std. Dev.	Absolute Std. Dev. Error
0.4935	0.0065	0.2843	0.0044
0.4965	0.0035	0.2860	0.027
0.4896	0.104	0.2928	0.0041
0.5033	0.0033	0.2841	0.0046

Table 1: Sample mean and standard deviation error for uniform random process

1.1.3 Bias

Figure 2 shows how the mean and standard deviation of each sample compare with the predicted results. Clearly, the estimates of the mean and standard deviation have minimum bias as their deviation from the theoretical values is minimal.

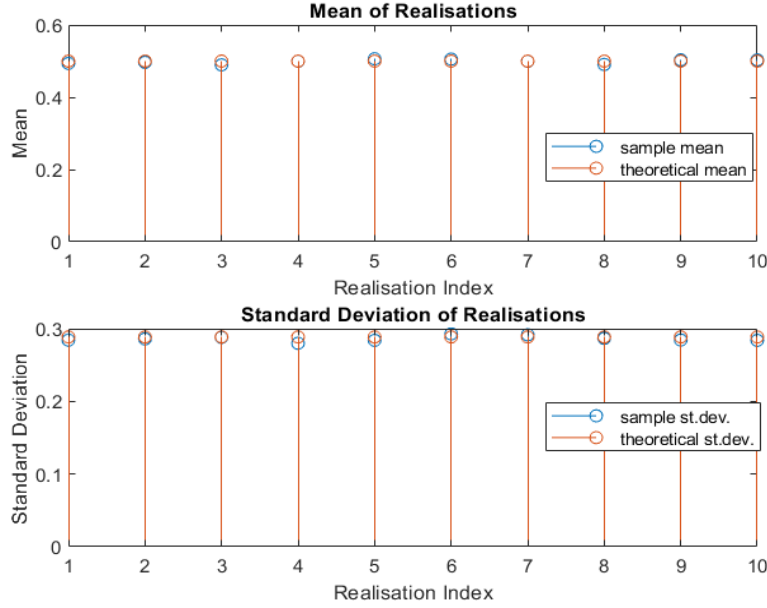


Fig. 2: Mean and standard deviation

1.1.4 PDF Approximation

The MATLAB function `histogram` has been used to obtain the histogram of the random process normalised by the number of samples. Figure 3 has been plotted with the sample size set to 1000, while, in figure 4, the size has been set to 10000. By comparing the two images, it can be deduced that the estimate converges as the number of samples increases. Increasing the number of bins enhances the histogram's resolution. However, if more bins than necessary are used, information about the distribution is distorted, leading to inaccurate inferences.

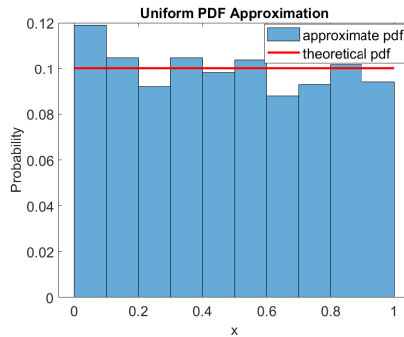


Fig. 3: Histogram and theoretical pdf for 1000 samples

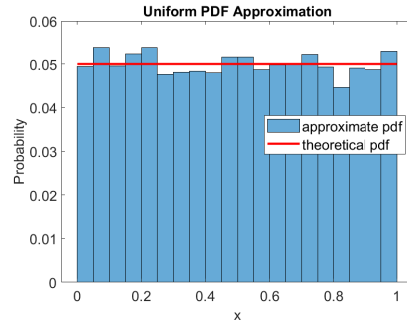


Fig. 4: Histogram and theoretical pdf for 10000 samples

1.1.5 Gaussian Random Process

A process with Standard normal random samples is created using the `randn` function and depicted in Figure 5. The sample mean and standard deviation are accurate estimators in this case too. As seen in Figure 6, the bias is consistently smaller than 0.1. The normalised histogram of the 1000 sample process is displayed in Figure 7. The same conclusions mentioned in section 1.1.4 apply in this case.

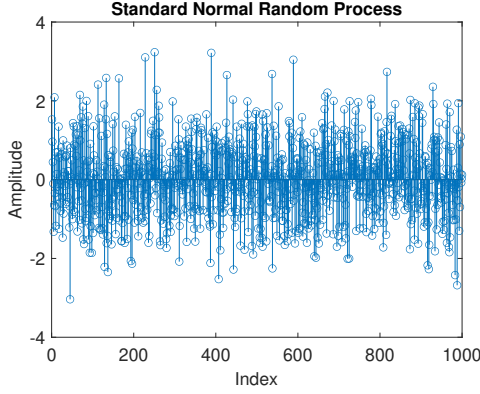


Fig. 5: Standard normal random process

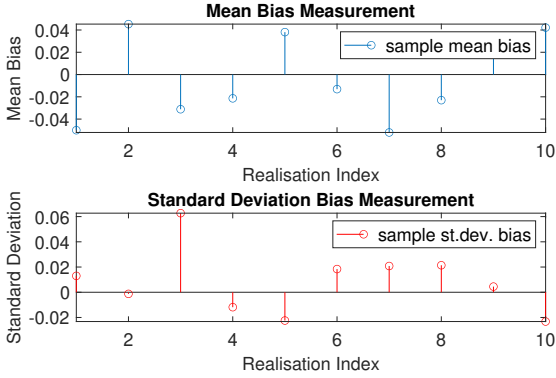


Fig. 6: Mean and standard deviation

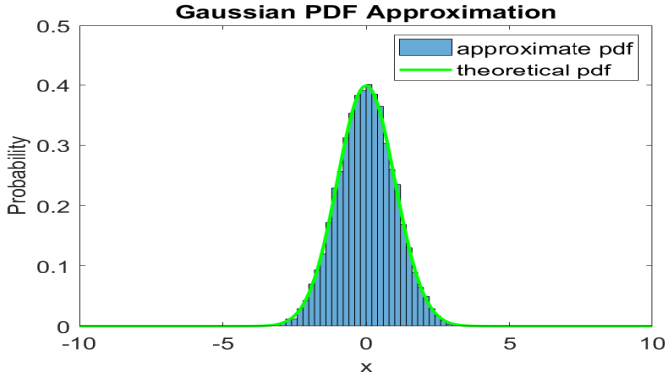


Fig. 7: Histogram of Standard normal distribution

1.2 Stochastic Process

1.2.1 Stationarity

Figure 8 shows the mean and standard deviation of *rp1*. The process is not stationary as both ensemble statistical moments vary with time.

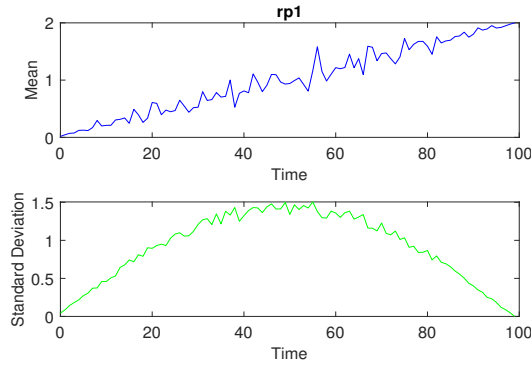


Fig. 8: Mean and standard deviation of $rp1$

In contrast, $rp2$ and $rp3$ are stationary and their ensemble average and standard deviation are depicted in Figures 9 and 10. The process $rp2$ has an ensemble mean of 0.5 and standard deviation of 0.31. The corresponding values for $rp3$ are 0.5 and 0.87 respectively. The stationarity of $rp2$ and $rp3$ is evident when higher values of the parameters M and N , which refer to the number and length of realisations, are used.

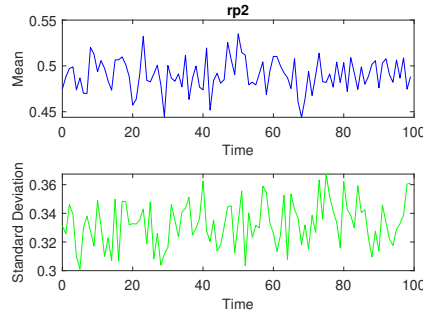


Fig. 9: Mean and standard deviation of $rp2$

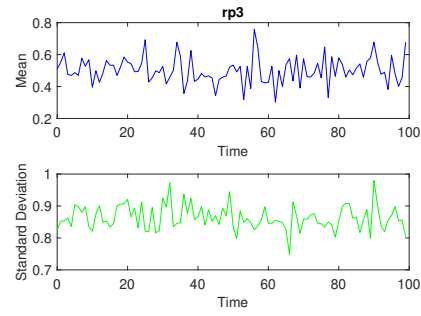


Fig. 10: Mean and standard deviation of $rp3$

1.2.2 Ergodicity

A random process is characterised as ergodic if the time statistics of a sufficiently long realisation converges to the ensemble's statistics. In this section, 4 realisations of length 1000 are generated for each process. Given that $rp1$ has a time-varying ensemble mean and standard deviation, it cannot be ergodic. In addition, the time statistics of $rp2$ do not converge to the ensemble's statistics, as shown in Table 2, even for higher M and N . Hence, the process $rp2$ is also not ergodic. The process $rp3$ is ergodic as its time average and standard deviation converge to the ensemble mean and standard deviation, with an error no larger than 0.1, as indicated in Table 3.

Realisation	Time Average	Time Std. Dev.
1	0.8127	0.1569
2	0.4889	0.1624
3	0.4505	0.2633
4	0.2640	0.0544

Table 2: Time values of statistics of $rp2$

Realisation	Time Average	Time Std. Dev.
1	0.4596	0.8831
2	0.5236	0.8711
3	0.4963	0.8354
4	0.5361	0.8990

Table 3: Time values of statistics of $rp3$

1.2.3 Mathematical Description

The discrete-time index of each realisation is denoted by n and the sample length by N . The process $rp1$ can be described as:

$$R_1[n] = U \cdot 5 \cdot \sin((n \cdot \pi)/N) + 0.02 \cdot n \quad (1)$$

U is a uniform distribution defined on the interval $[-0.5, 0.5]$ and has a theoretical mean of 0 and a variance of 0.0833. It has been generated using the `rand` function, which accepts as input arguments `M` and `N`, as previously defined. Recalling the linearity of the expectation operator, the theoretical mean and variance of $rp1$ are computed as follows:

$$E[R_1[n]] = E[U \cdot 5 \cdot \sin((n \cdot \pi)/N)] + E[0.02 \cdot n] = E[U] \cdot E[\sin((n \cdot \pi)/N)] + 0.02 \cdot n = 0.02 \cdot n$$

$$\begin{aligned} E[R_1^2[n]] &= E[(U \cdot 5 \cdot \sin((n \cdot \pi)/N))^2] + 2 \cdot E[U \cdot 5 \cdot \sin((n \cdot \pi)/N)] \cdot E[0.02 \cdot n] + E[(0.02 \cdot n)^2] \\ &= \frac{1}{12} \cdot 25 \cdot \sin^2((n \cdot \pi)/N) + 0.0004 \cdot n^2 \end{aligned}$$

$$Var[R_1[n]] = E[R_1^2[n]] - E^2[R_1[n]] = \frac{25}{12} \cdot \sin^2((n \cdot \pi)/N) \Rightarrow \sigma_{R_1} = \sqrt{Var[R_1[n]]} = 1.44 \cdot \sin((n \cdot \pi)/N)$$

The expressions obtained match the results obtained in Figure 8 of section 1.2.1.

The random process $rp2$ is described by the following equation:

$$R_2[n] = U \cdot X + Z \quad (2)$$

U is defined in the same manner as in Equation (1). Both X and Z are uniform random variables with support on $[0, 1]$, a mean of 0.5 and a variance of 0.0833. Noting that X, U are independent:

$$E[R_2[n]] = E[U \cdot X] + E[Z] = E[Z] = 0.5, \text{ as } E[[U \cdot X]] = E[U] \cdot E[X] = 0$$

$$E[R_2^2[n]] = E[(U \cdot X)^2] + 2 \cdot E[U \cdot X] \cdot E[Z] + E[Z^2] = E[U^2] \cdot E[X^2] + E[Z^2] = 0.0833 \cdot \int_0^1 x^2 dx + \int_0^1 z^2 dz = 0.36$$

$$Var[R_2[n]] = E[R_2^2[n]] - E[R_2[n]]^2 = 0.11 \Rightarrow \sigma_{R_2} = \sqrt{Var[R_2[n]]} = 0.33$$

The theoretical mean matches the experimental mean, while the theoretical standard deviation is approximately equal to the value that the ensemble varies around.

The process $rp3$ is defined mathematically as:

$$R_3[n] = 3 \cdot U + 0.5 \quad (3)$$

$$E[R_3[n]] = 3 \cdot E[U] + 0.5 = 0.5$$

$$Var[R_3[n]] = 9 \cdot Var[U] = 9 \cdot \frac{1}{12} = 0.75 \Rightarrow \sigma_{R_3} = \sqrt{Var[R_3[n]]} = 0.87$$

The theoretical results for $rp3$ match the experimental findings in section 1.2.1.

1.3 Estimation of Probability Distributions

1.3.1 PDF Estimate

The pdf of a collection of samples can be estimated by plotting the histogram. Listing 1 shows the MATLAB implementation. The code extracts the endpoints and the relative frequency of each bin. In order to ensure dimensional consistency between the matrix of sample values and the probabilities matrix, the average of each bin's edges was computed and used for the approximate pdf plot.

```
1 hist=histogram(v, 'Normalization', 'Probability');
2 hold on;
3 probs=hist.Values;
4 rv_values=hist.BinEdges;
5 intervals=zeros(length(rv_values)-1,2);
6 for index=1:(length(rv_values)-1)
7     intervals(index,1)= rv_values(1,index);
8     intervals(index,2)= rv_values(1,index+1);
9 end
10 x_ax=(mean(intervals,2))';
11 plot(x_ax,probs, 'Color','Red');
```

The method's accuracy improves as the number of samples increases. The effect can be observed in the figures below.

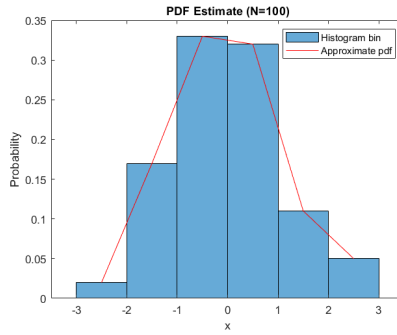


Fig. 11: approximate pdf for N=100 samples

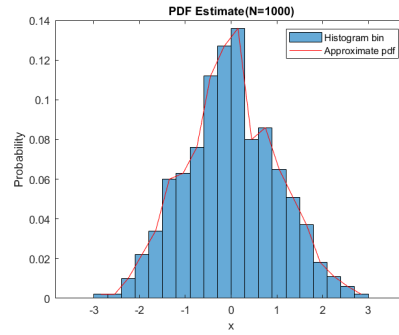


Fig. 12: approximate pdf for N=1000 samples

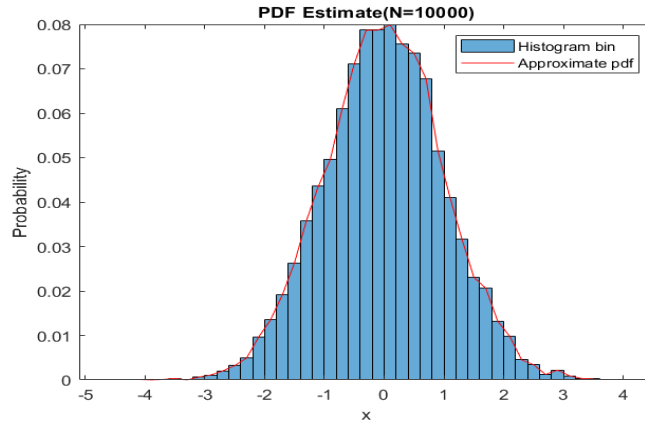


Fig. 13: approximate pdf for N=1000 samples

1.3.2 Approximate PDF of Stationary Ergodic Processes

$rp3$ is the only process which is both stationary and ergodic. The pdf estimates along with the theoretical distributions for sample sizes of 100, 1000, and 10000 samples are plotted in the following figures. The theoretical pdf of this uniform distribution is a constant line $y=1/3 \approx 0.33$. Increasing the number of samples results in the estimate converging to the theoretical constant line. The maximum error observed for a sample size of 10000 is approximately 0.01.

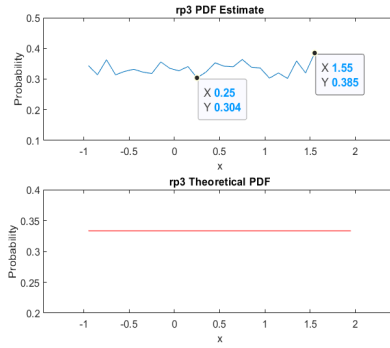


Fig. 14: approximate pdf of $rp3$ for N=100 samples

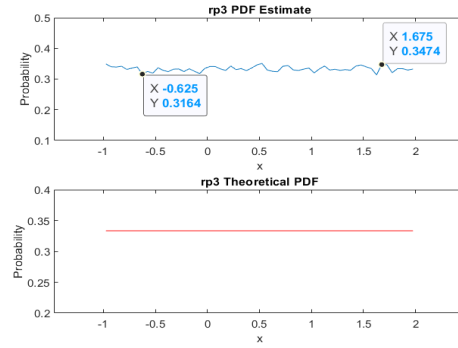


Fig. 15: approximate pdf of $rp3$ for N=1000 samples

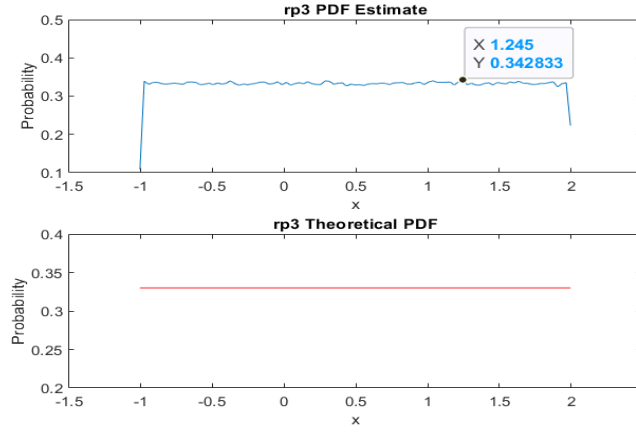


Fig. 16: approximate pdf of $rp3$ for $N=10000$ samples

1.3.3 Approximate PDF of Non-stationary Processes

The pdf of a non-stationary process cannot be computed with the function `pdf` because it depends on the relative frequencies of a single realisation's amplitudes. If stationarity cannot be assumed, the histogram built from this realisation is not representative of the ensemble as the MATLAB functions `histogram` and `hist` use time averaging, which requires stationarity.

In this case, the non-stationary signal shown in Figure 17, whose mean changes from 0 to 1 after the first 500 samples, can be decomposed into a sum of stationary parts. However, this is not practical in real-world applications, where knowledge of a change in statistics will not be available a priori.

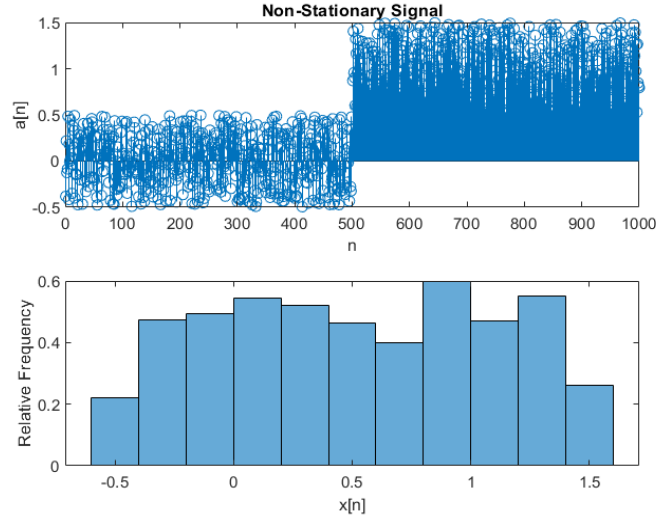


Fig. 17: Plot and histogram of non-stationary signal

2 Linear Stochastic Modelling

2.1 ACF of Uncorrelated and Correlated Sequences

2.1.1 ACF of 1000 Sample White Gaussian Noise Realisation

The Autocorrelation Function (ACF) of the 1000-sample white Gaussian noise realisation \mathbf{x} is shown in Figure 18. It is symmetric with respect to the y-axis, which is in line with the theoretical properties of an ACF. As the samples of the process are uncorrelated, the resulting ACF consists of single spikes at each time lag. The maximum value appears at time lag $\tau=0$, where the autocorrelation of the waveform is computed with a copy of itself.

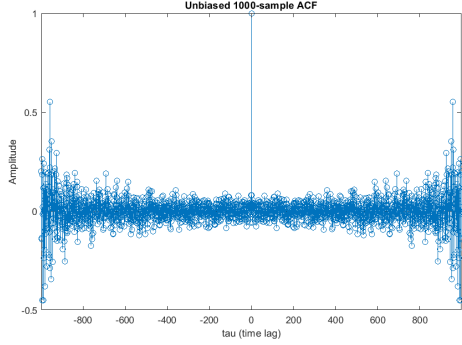


Fig. 18: ACF of 1000-sample white Gaussian noise realisation

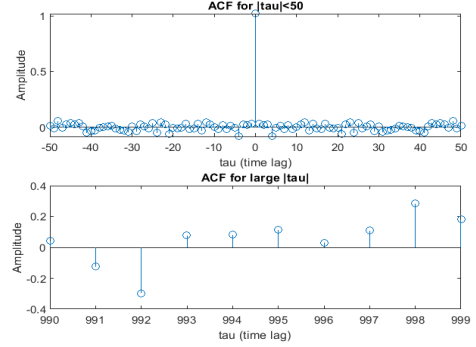


Fig. 19: ACF comparison for different τ values

2.1.2 Zooming in on $|\tau| < 50$

Figure 19 depicts the shape of the ACF for $|\tau| < 50$ and for large τ . In the first region, the function exhibits close to ideal behaviour, as its values tend to 0 at the interval's endpoints. In contrast, the values for large τ tend to oscillate, significantly deviating from 0.

2.1.3 Differences Due to Varying Time Lags

A large autocorrelation lag τ limits the accuracy of the ACF estimate. The unbiased estimate for the ACF is given by the formula:

$$R_x(\tau) = \frac{1}{N - |\tau|} \sum_{n=0}^{N-|\tau|-1} x[n] \cdot x[n + \tau]$$

As $|\tau|$ grows and approaches $N=1000$, the number of terms over which the summation is performed decreases, while the scaling factor increases. Hence, as the autocorrelation is computed over fewer input terms, the ACF estimates for large $|\tau|$ are unreliable. By inspection of the plot, an empirical bound on τ is $N/4$, where N denotes the number of samples the sequence has.

2.1.4 ACF of MA Filter

The implementation of the Moving Average (MA) filter of order 9 is plotted in Figure 20. The plot is symmetric with respect to the y axis and the amplitudes beyond $\tau = 9$ vary insignificantly in comparison to the value at $\tau = 9$. This suggests that using more than 9 coefficients would result in an over-fitted model.

A Finite Impulse Response (FIR) filter is described by the equation $y[n] = \sum_{k=1}^{k=N} b[k] \cdot x[n - k]$, where N denotes the filter order. By setting $b[k] = \frac{1}{N}$, $\forall k = 1, \dots, N$, the sample mean over a window of samples is obtained. An accurate estimate requires N to be greater or equal than the number of filter coefficients.

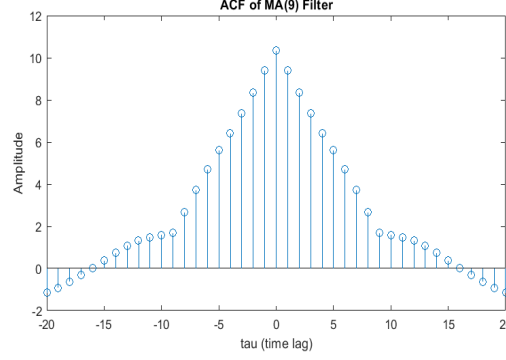


Fig. 20: ACF of MA(9) filter

2.1.5 ACF of Filter Output

The ACF of \mathbf{x} $R_x(\tau)$ will consist of a series of impulses as the samples of the process are independent from each other. The output \mathbf{y} is obtained through the convolution of the filter's impulse response with the input \mathbf{x} . Consequently, R_y equals the filter's coefficients scaled by the amplitude of the impulses at each time index in the ACF of \mathbf{x} , given that the Dirac delta function is the neutral element of convolution.

2.2 Cross-Correlation Function

2.2.1 Cross-Correlation Estimate

The Cross-Correlation Function (CCF) of the filter input \mathbf{x} and the output \mathbf{y} has large values $\forall \tau \in \{-8, -7, \dots, -1, 0\}$, as indicated in Figure 21. This means \mathbf{y} will strongly depend on the present and past 8 samples of \mathbf{x} , which is justified given that the MA filter has order 9. As X_t is an uncorrelated process, its ACF is represented by a series of impulses and hence $R_{x_y}(\tau) = h(\tau)$.

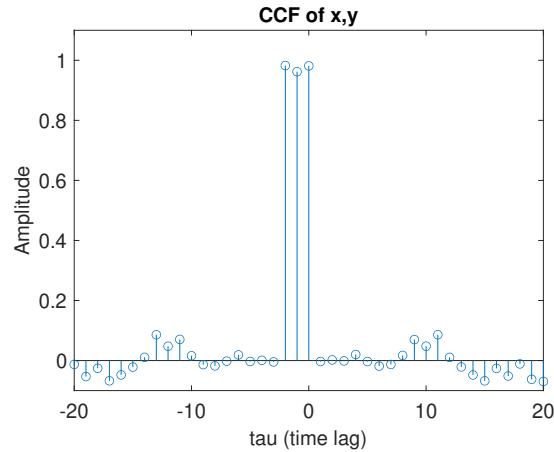


Fig. 21: Cross correlation plot of \mathbf{x} and \mathbf{y}

2.2.2 System Identification

The above-mentioned result shows that the cross-correlation of an uncorrelated input signal \mathbf{x} with the output \mathbf{y} of a Linear Time Invariant(LTI) system yields the system's impulse response. This is significant because the impulse response uniquely identifies a system.

2.3 Autoregressive Modelling

2.3.1 Stability Region

The given AR(2) process \mathbf{x} is described by the equation: $x[n] = a_1 \cdot x[n-1] + a_2 \cdot x[n-2] + w[n]$, where \mathbf{w} is a noise process with samples $w[n]$ obtained from a standard normal distribution. By applying the Z-transform and rearranging, the following expression is obtained:

$$\frac{X(z)}{W(z)} = \frac{1}{1 - a_1 \cdot z^{-1} - a_2 \cdot z^{-2}}$$

The denominator polynomial is a second order characteristic equation, which can be solved to yield two roots:

$$z_{1,2} = \frac{a_1 \pm \sqrt{a_1^2 + 4 \cdot a_2}}{2}$$

The stability condition is $|z| < 1$, as the process will become unstable if its poles lie outside the unit circle in the z-domain. The following conditions can be derived for the larger (z_1) and smaller root (z_2):

$$a_1 + \sqrt{a_1^2 + 4 \cdot a_2} < 2 \Rightarrow \sqrt{a_1^2 + 4 \cdot a_2} < 2 - a_1 \Rightarrow (\sqrt{a_1^2 + 4 \cdot a_2})^2 < (2 - a_1)^2 \Rightarrow a_1 + a_2 < 1, \text{ assuming } a_1 \leq 2.$$

$$a_1 - \sqrt{a_1^2 + 4 \cdot a_2} > -2 \Rightarrow (\sqrt{a_1^2 + 4 \cdot a_2})^2 < (2 + a_1)^2 \Rightarrow a_2 - a_1 < 1, \text{ assuming } a_1 \geq -2.$$

The characteristic equation is written in terms of its roots λ_1 and λ_2 as follows:

$$Q(z) = (z - \lambda_1) \cdot (z - \lambda_2) = z^2 - (\lambda_1 + \lambda_2) \cdot z + \lambda_1 \cdot \lambda_2 = 0$$

For the process to be stationary, $\lambda_1, \lambda_2 < 1$, which leads to the condition $a_2 < 1$. The stability region forms a triangle and matches the one that can be obtained using the conditions derived above.

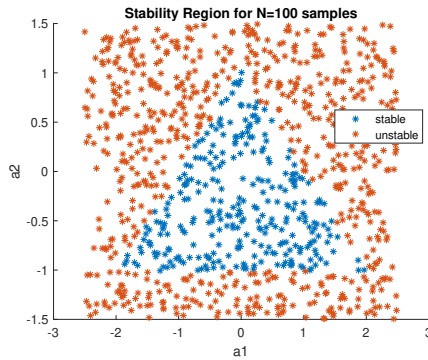


Fig. 22: Stability triangle for N=100 samples

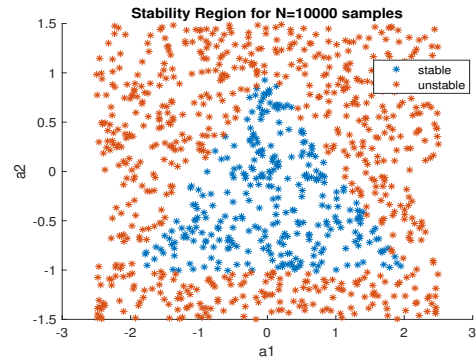


Fig. 23: Stability triangle for N=1000 samples

2.3.2 ACF of Sunspot Time Series

Figure 24 depicts the sunspot ACF for $N=5$ samples. Due to the small number of samples, no pattern can be observed. When increasing the sequence length to 20, as seen in Figure 25, the sunspot amplitude is the same every 13 time instants. The mean biases the ACF, which explains why the original data's ACF plots have no negative amplitudes. It can also be seen that zero-mean ACF graphs present their maximum value at lag $\tau = 0$, which is consistent with the theoretical properties of an ACF, but does not hold for the non-zero mean plots. Further increasing N to 250 renders the ACF more periodic. As mentioned in 2.1.3, the ACF values at large time lags are less accurate due to the smaller number of samples used in the summation.

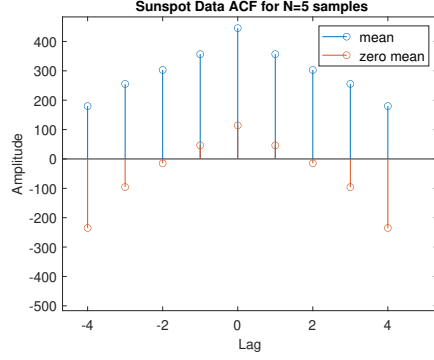


Fig. 24: Sunspot ACF for $N=5$ samples

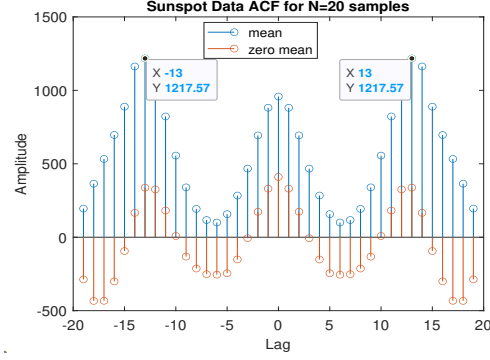


Fig. 25: Sunspot ACF for $N=20$ samples

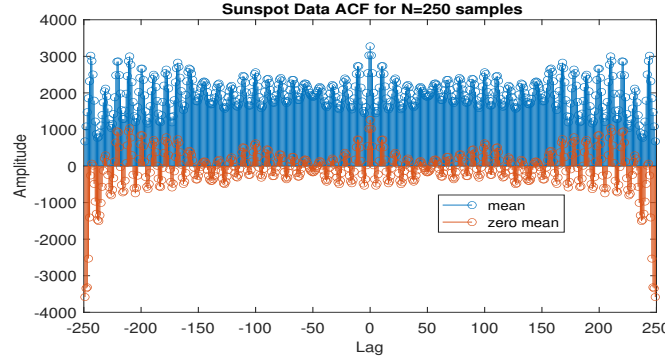


Fig. 26: Sunspot ACF for $N=250$ samples

2.3.3 Partial ACF for Different Model Orders

The partial autocorrelation coefficients for the regular and standardised sunspot series are plotted in Figure 27. The partial ACF at lag $\tau = 3$ is significantly smaller than the value at $\tau = 2$, which suggests $p=2$ is the most likely order for this series. The slight difference in correlation values between the original data and the normalised data is due to the bias introduced by the mean in the original series.

2.3.4 Model Order Selection

The Minimum Description Length (MDL) and Akaike Information Criterion (AIC) suggest that the possible model orders are $p=2$ and $p=9$. The corrected AIC, which minimises the probability of overfitting through its

additional penalty term, demonstrates that order 2 is the appropriate choice. This result confirms the findings in the previous part.

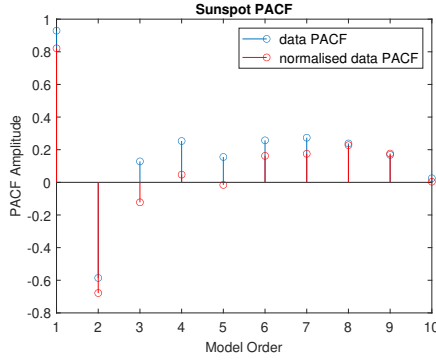


Fig. 27: Sunspot series PACF

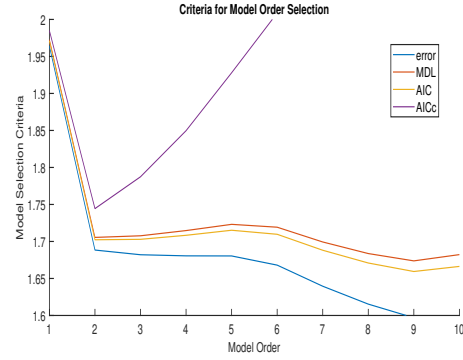


Fig. 28: Sunspot series model order selection

2.3.5 AR Model Predictions

A higher AR model order predicts future sunspot series values more accurately. For a horizon $m = 1$, the AR(1) model lags the sunspot data by 2 samples. The performance of the AR(1) model degrades significantly for larger prediction steps. In contrast, AR(10) produces the most accurate results for all prediction horizons under consideration. The disadvantage of higher orders is the increase in computational complexity given the increased number of variables being used.

While overfitted models exhibit minimal error, their results cannot be generalised to larger prediction windows, which means their extrapolation error is higher. The plots for prediction steps $m = 1, 5$ are shown below.

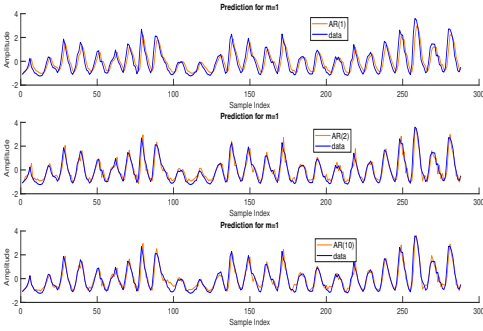


Fig. 29: Predictions for m=1

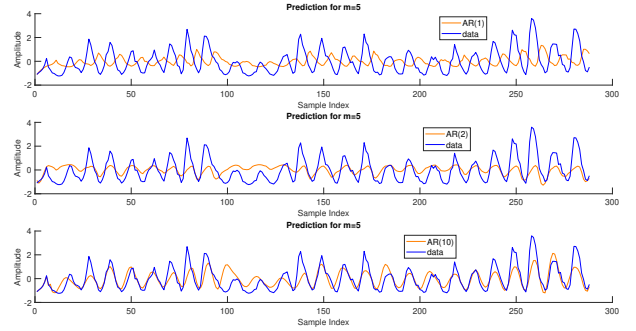


Fig. 30: Predictions for m=5

2.4 Cramer-Rao Lower Bound

2.4.1 Model Order for NASDAQ Closing Prices

Figure 31 shows that the partial ACF of the NASDAQ data decreases rapidly after lag $\tau = 1$, which suggests that an AR(1) model is sufficient. Given the large number of data samples, the corrected AIC does not have to be considered. As indicated in Figure 32, both the MDL and AIC have a minimum at model order $p=1$, thus confirming the observation from the partial ACF plot.

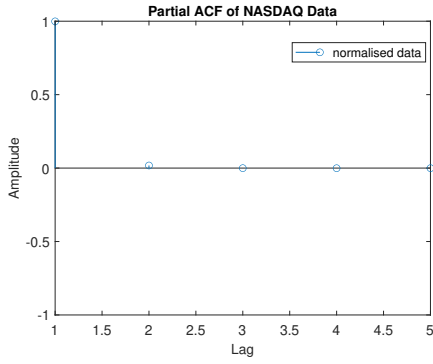


Fig. 31: Partial ACF of NASDAQ data

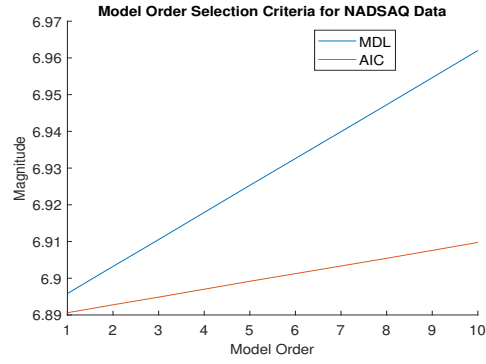


Fig. 32: Model order selection criteria for NASDAQ data

2.4.2 Fisher Information Matrix Derivation

The log-likelihood function of the spectrum of an AR(p) process is given by:

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

For p=1, evaluating equation 19 in the coursework specification requires differentiating the equation above with respect to $\hat{\sigma}^2$:

$$\frac{\partial \ln[\hat{P}(f; \theta)]}{\partial \hat{\sigma}^2} = \frac{1}{\sigma^2} \quad \text{and} \quad \frac{\partial^2 \ln[\hat{P}(f; \theta)]}{\partial \hat{\sigma}^2} = -\frac{1}{\sigma^4}$$

We then substitute the first partial derivative into equation 19 in order to evaluate the element $[\mathbf{I}(\theta)]_{22}$ of the Fisher Information matrix :

$$[\mathbf{I}(\theta)]_{22} = \frac{N}{2} \cdot \int_{-\frac{1}{2}}^{\frac{1}{2}} \frac{1}{\sigma^2} \cdot \frac{1}{\sigma^2} df = \frac{N}{2} \cdot \frac{1}{\sigma^4}$$

Therefore, the Fisher Information matrix is written as:

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

2.4.3 CRLB Derivation and Heatmap

For the parameter matrix $\theta = [a_1, \hat{\sigma}^2]$, the CRLB is given by $\text{var}(\hat{\theta}_i) \geq [I^{-1}(\theta)]_{ii}$. By inverting the diagonal Toeplitz Fisher matrix and using relation (15) from the specification, the following inequalities are derived:

$$\text{var}(\hat{a}_1) \geq \frac{1}{N} \cdot \frac{\sigma^2}{r_{xx}(0)} = \frac{1}{N} \cdot (1 - a_1^2), \text{ since } \sigma_x^2 = r_{xx}(0) = \frac{\sigma^2}{1 - \rho_1 a_1} = \frac{\sigma^2}{1 - a_1^2}, \text{ where } \sigma_x \text{ is the signal power}$$

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

The value of $\text{var}(\hat{a}_1)$ is $2.3796 \cdot 10^{-6}$, since $\hat{a}_1 = 0.9989$. Therefore, as the pole of the transfer function is at -a[1], this parameter is easier to estimate when the magnitude of the pole approaches unity, which corresponds to the unit circle in the z-domain. The heatmaps for \hat{a}_1 and $\hat{\sigma}$ are plotted below.

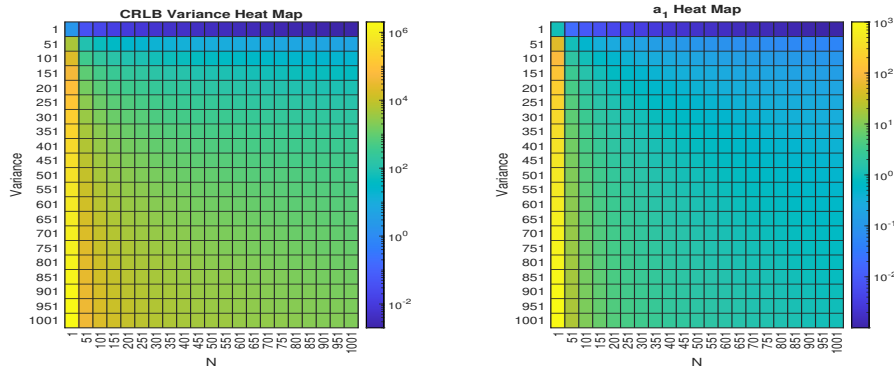


Fig. 33: CRLB heatmaps for $\hat{\sigma}^2$ and \hat{a}_1

2.4.4 Computing CRLB Using $\mathbf{A}(f)$

For $p=1$, the AR(1) power spectrum is written as:

$$P_x(f; \theta) = \frac{\sigma^2}{|1 - a_1 \cdot e^{-j2\pi f}|^2} \quad (4)$$

Differentiating with respect to a_1 and σ^2 yields:

$$\frac{\partial P_x(f; \theta)}{\partial a_1} = 2\sigma^2 \cdot \frac{e^{-j2\pi f}}{(1 - a_1 e^{-j2\pi f})^3} = 2\sigma^2 \cdot \frac{e^{-j2\pi f}}{|A(f)|^3} \quad (5)$$

$$\frac{\partial P_x(f; \theta)}{\partial \sigma} = \frac{1}{|A(f)|^2} \quad (6)$$

We therefore obtain:

$$\begin{aligned} \text{var}(\hat{P}_x(f; \theta)) &\geq \begin{bmatrix} 2\sigma^2 \frac{e^{-j2\pi f}}{|A(f)|^3} & \frac{1}{|A(f)|^2} \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{N}(1 - a_1^2) & 0 \\ 0 & \frac{N}{2\sigma^4} \end{bmatrix} \cdot \begin{bmatrix} 2\sigma^2 \cdot \frac{e^{-j2\pi f}}{|A(f)|^3} \\ \frac{1}{|A(f)|^2} \end{bmatrix} \\ &= \frac{2\sigma^4}{N} \left[\frac{2(1 - a_1^2)e^{-j4\pi f}}{|A(f)|^6} + \frac{1}{|A(f)|^4} \right] \end{aligned} \quad (7)$$

2.5 ECG from iAmp Experiment

2.5.1 Heart Rate Probability Density Estimate

Every 10 samples of the heart rate data $h[n]$ are averaged so that a smoother estimate of the heart rate is obtained as follows:

$$\hat{h}[k] = \frac{1}{10} \sum_{i=1}^{10} \alpha h[i] \quad \forall k = 1, \dots, 10 \quad (8)$$

The PDE of the original heart rates and the averaged heart rates for $\alpha = 1$ and $\alpha = 0.6$ is plotted in Figure 34. Averaging yields a signal estimate with a variance lower by a factor of 100 compared to the original signal, as $\text{var}(\hat{h}[k]) = \text{var}(\frac{1}{10} \sum_{i=1}^{10} \alpha h[i]) = \frac{1}{100} \cdot \text{var}(h[k])$. The α parameter shifts the PDE to the left when it is set as less than 1 and further reduces the variance since $\text{var}(\alpha \hat{h}[k]) = \alpha^2 \cdot \text{var}(\hat{h}[k])$.

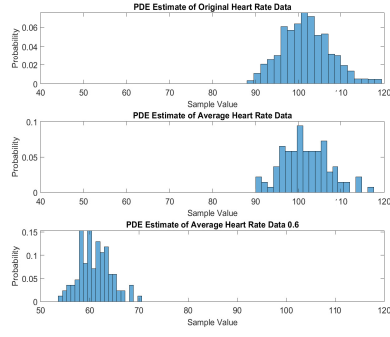


Fig. 34: PDEs of original and averaged heart rates for Trial 1

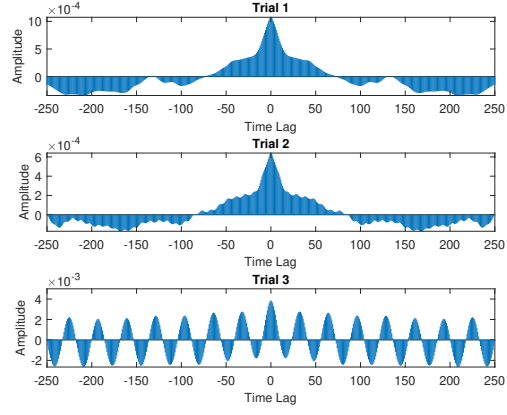


Fig. 35: ACF for the 3 RRI trials

2.5.2 AR Modelling of Heart Rate

Figure 35 shows the auto-correlation sequences of the RRI data for the 3 trials. All 3 trials can be modelled as AR processes because the length of their ACFs is infinite and sinusoidal. Trial 1 can be modelled as an AR(4) process since the partial ACF coefficients for orders 5 and higher are insignificant compared to the previous ones. This is confirmed by the MDL, AIC and AICc as they show a minimum at $p = 4$ and the AICc thanks to its characteristic term shows that higher orders would result in overfitting. Trial 2 can be modelled as an AR(3) process, considering that all three criteria have a local minimum at that order. Although the partial ACF coefficient at order $p = 9$ is comparable to that at $p = 3$, the AICc shows that a ninth order model would lead to overfitting. An AR(3) model should be used for Trial 3 because all order selection criteria present a minimum at $p = 3$ and the partial ACF values for orders greater than 3 are negligible.

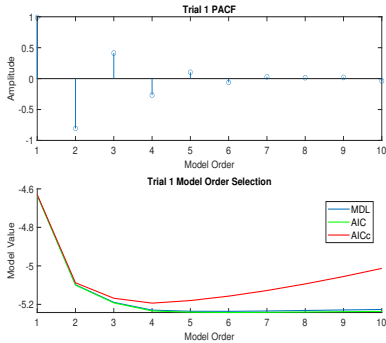


Fig. 36: Trial 1 model order selection

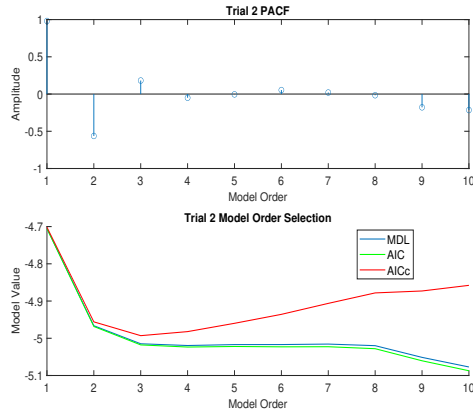


Fig. 37: Trial 2 model order selection

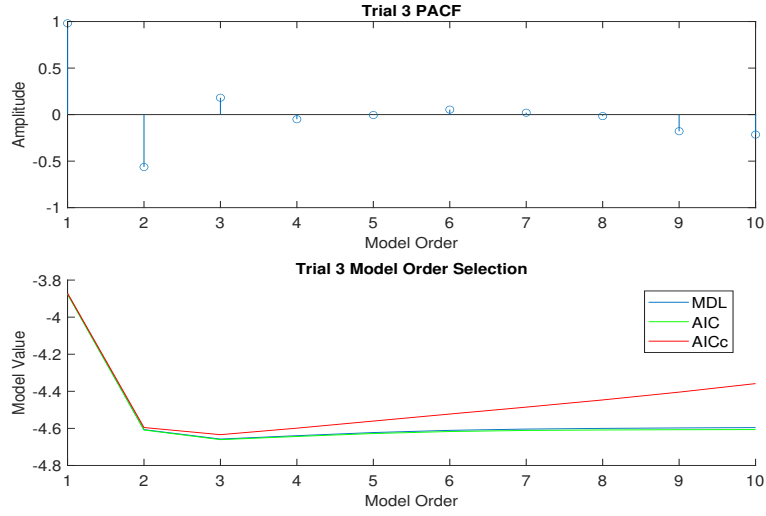


Fig. 38: Trial 3 model order selection

3 Spectral Estimation and Modelling

The PSD of a signal can be estimated using the periodogram, which is based on the DFT and is mathematically defined as:

$$\hat{P}_x(f) = \frac{1}{N} \left| \sum_{n=0}^{N-1} x[n] \cdot e^{-j2\pi f \frac{n}{N}} \right|^2 \quad (9)$$

In theory, white Gaussian noise has a flat PSD of unity magnitude. The periodogram does not match the ideal PSD graph, as indicated by the figure below. Based on the Wiener-Khinchine theorem, the PSD estimate uses the unbiased ACF of the random process, which exhibits increasing variance at delays τ that are as large as the signal length. The plot is symmetric about the normalised frequency 0.5 since the white noise samples are purely real and will hence have a conjugate symmetric PSD.

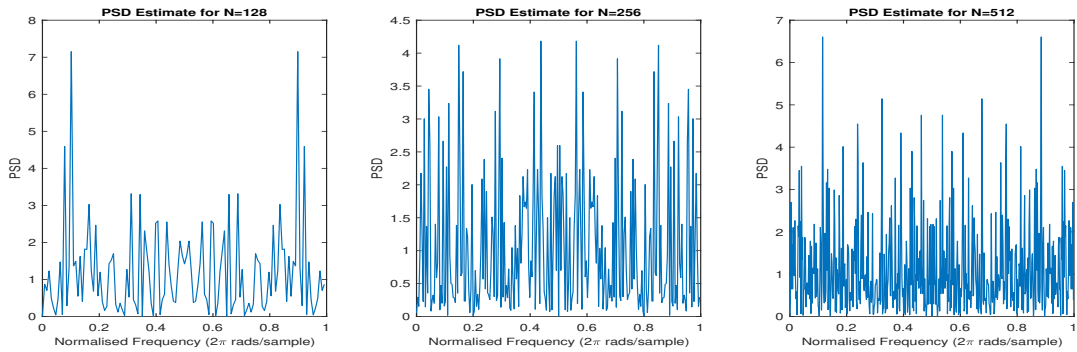


Fig. 39: PSD plots using "pgm.m" for N=128, 256, 512 samples

3.1 Averaged Periodiogram Estimates

3.1.1 PSD Smoothing

The PSD estimates for unfiltered and filtered white Gaussian noise are plotted in Figure 40. The FIR filter smooths the data by removing peaks resulting from the ACF's variance, while retaining the general shape of the original waveform.

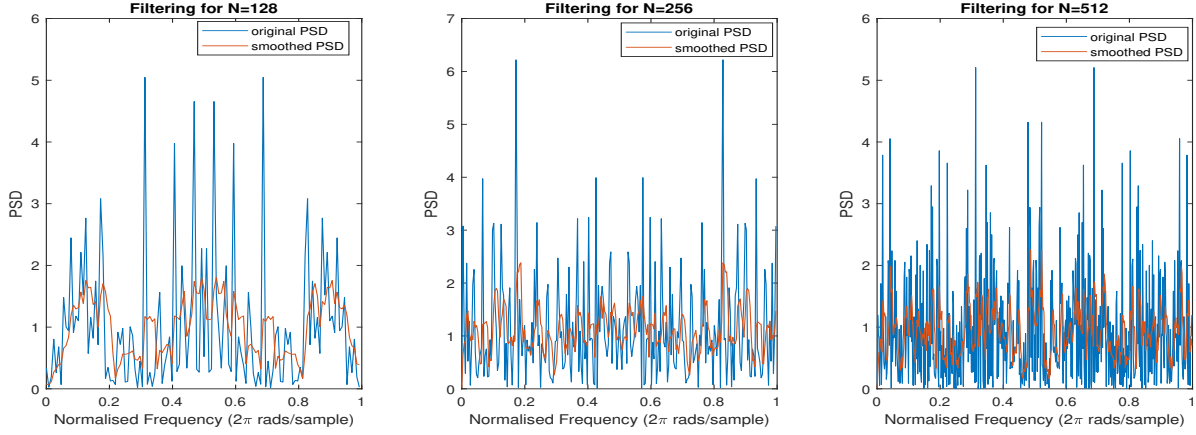


Fig. 40: Smoothed PSD plots for N=128, 256, 512 samples

3.1.2 Multiple PSDs

The 1024-sample noise process is divided into 8 non-overlapping segments. Each PSD exhibits peaks at different frequencies, which suggests there exists no clear pattern. Therefore, each segment is an unreliable estimate of the theoretical white Gaussian noise PSD.

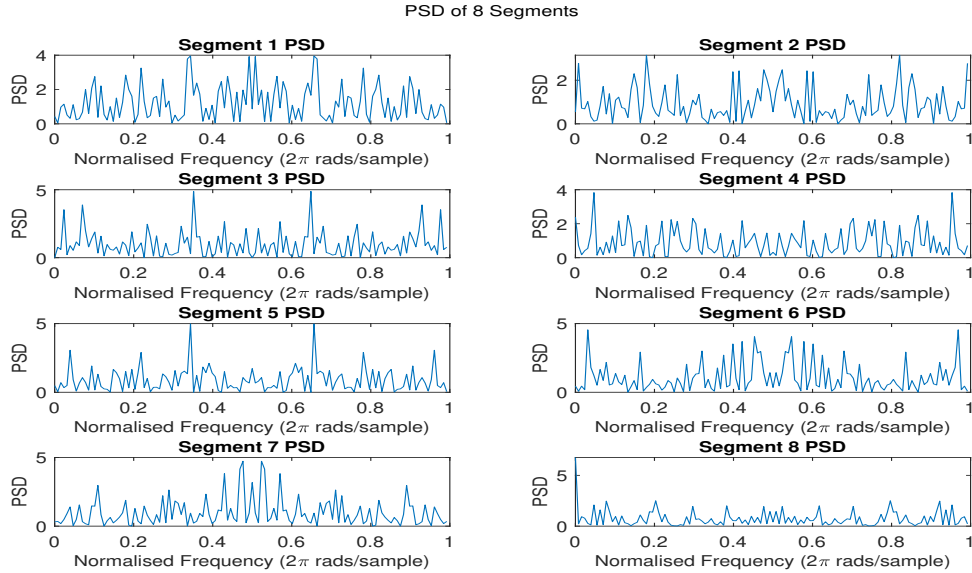


Fig. 41: PSD of each white Gaussian noise segment

3.1.3 Averaged Periodogram

The *averaged periodogram* is obtained by averaging the PSDs of the 8 segments. This estimate is a significant improvement because the variance is noticeably lower. In fact, the plot resembles the uniform frequency spectrum that one would expect for a white Gaussian noise process.

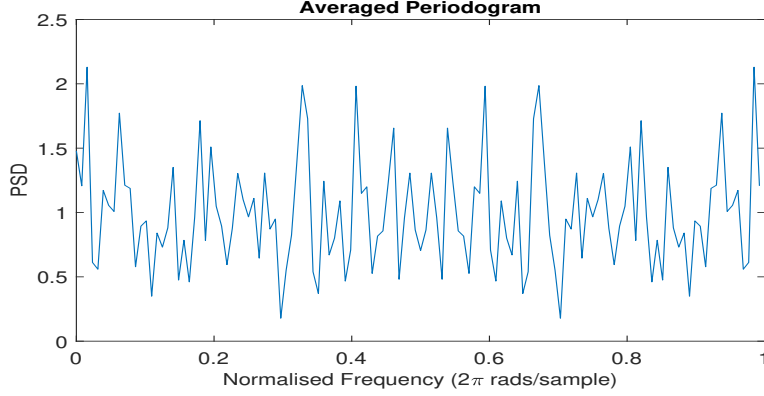


Fig. 42: Averaged Periodogram PSD

3.2 Spectrum of Autoregressive Processes

3.2.1 Exact PSD and Periodogram

The 1024-sample WGN realisation \mathbf{x} and the filtered output signal \mathbf{y} are depicted in Figure 43. The exact PSD of \mathbf{y} is given by the following expression and plotted below:

$$P_y(f) = \frac{1}{|1 + 0.9 \cdot e^{-j2\pi f}|^2} \quad (10)$$

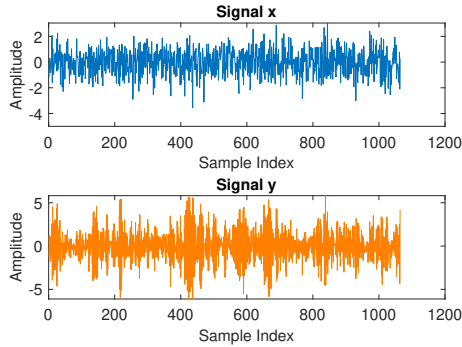


Fig. 43: Sequence \mathbf{x} , \mathbf{y}

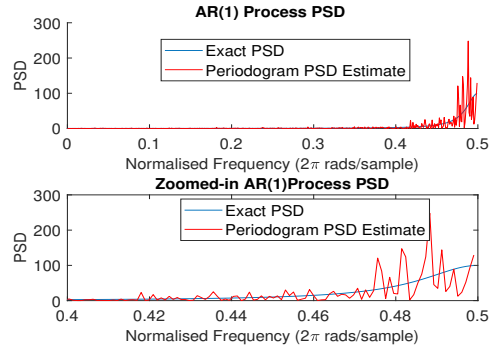


Fig. 44: PSD of \mathbf{y} and zoomed-in plot

By comparing the periodogram and exact PSD curves, the periodogram is evidently an inaccurate estimator of the PSD because it exhibits oscillatory behaviour at the $[0.4, 0.5]$ frequency range. Both curves follow the trend of a high-pass filter and the cutoff frequency is 0.41 Hz. The periodogram shows high variance in the $[0.4, 0.5]$ range because it is computed through the FFT, which uses a rectangular window. A rectangular function is transformed to a *sinc* in the Fourier domain. Therefore, the multiplication of the original periodogram estimate with the the window yields a signal which exhibits oscillations at high frequencies.

3.2.2 Model Based PSD

A model-based PSD for \mathbf{y} can be estimated through the parameters \hat{a}_1 and $\hat{\sigma}$ as follows:

$$\hat{a}_1 = -\hat{R}_y(1)/\hat{R}_y(0) \quad (11)$$

$$\hat{\sigma} = \hat{R}_y(0) + \hat{a}_1 \cdot \hat{R}_y(1) \quad (12)$$

Evaluating the equations above gives $\hat{a}_1 = 0.9130$ and $\hat{\sigma} = 0.9337$. Figure 45 illustrates how the exact, model-based, and periodogram estimated PSD compare. The model-based PSD's accurately predicts the exact curve because it is constructed with parameters heavily dependent on \mathbf{y} and does not suffer from windowing effects, unlike the periodogram. However, its accuracy requires a-priori knowledge of the signal, which is not guaranteed in real-world applications.

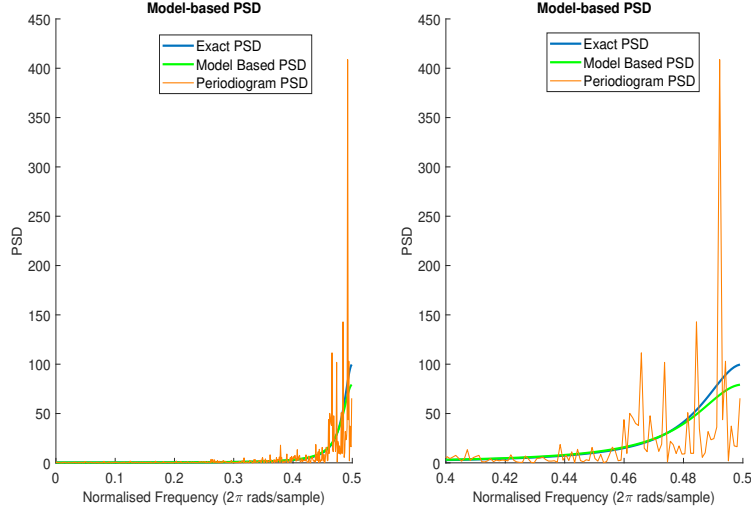


Fig. 45: Model-based, periodogram-based and exact PSDs

3.2.3 PSD Analysis for Sunspot Series

The periodogram and model-based PSD estimates for various orders of the original and normalised sunspot series are plotted in Figure 46. The difference between the two PSDs is the fact that the one on the left has a significant DC component, which is expected given the large mean of the sunspot data. Normalising the data allows for a meaningful comparison of the effects of different orders. The AR(1) model clearly underfits the data since it does not include the peak at $f=0.1$. The AR(2) model captures the main peak at $f=0.1$, but fails to capture the ones close to $f=0$. However, as mentioned in section 2.3.4, the order $p = 2$ is the best fit for the sunspot data.

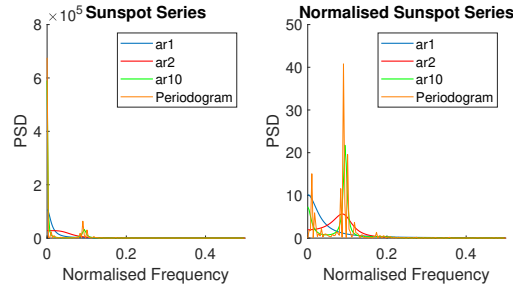


Fig. 46: PSDs of original and normalised sunspot data

3.3 Least Squares Estimation of AR Coefficients

3.3.1 Mathematical Expressions for Cost Function

The Least Squares(LS) cost function for unknown AR coefficients $\mathbf{a} = [a_1, a_2, \dots, a_p]$ is given by:

$$J = \sum_{k=1}^M [\hat{r}_{xx}[k] - \sum_{i=1}^p a_i \cdot \hat{r}_{xx}[k-i]]^2 \quad (13)$$

This equation can be rewritten in matrix form as $(\mathbf{x} - \mathbf{s})^T(\mathbf{x} - \mathbf{s}) = (\mathbf{x} - H\mathbf{a})^T(\mathbf{x} - H\mathbf{a})$, where $\mathbf{x} = [\hat{r}_{xx}(1), \dots, \hat{r}_{xx}(M)]^T$. By comparison of the forms, \mathbf{s} and H are:

$$H = \begin{bmatrix} \hat{r}_{xx}[0] & \hat{r}_{xx}[-1] & \cdots & \hat{r}_{xx}[-p+1] \\ \hat{r}_{xx}[1] & \hat{r}_{xx}[0] & \cdots & \hat{r}_{xx}[-p+2] \\ \vdots & \vdots & & \vdots \\ \hat{r}_{xx}[M-1] & \hat{r}_{xx}[M-2] & \cdots & \hat{r}_{xx}[M-p] \end{bmatrix} \quad (14)$$

$$\mathbf{s} = H\mathbf{a} = \begin{bmatrix} \sum_{i=1}^p a[i] \hat{r}_{xx}[\hat{1}-i] \\ \sum_{i=1}^p a[i] \hat{r}_{xx}[\hat{2}-i] \\ \vdots \\ \sum_{i=1}^p a[i] \hat{r}_{xx}[\hat{M}-i] \end{bmatrix} \quad (15)$$

By projecting the signal $\mathbf{x} \in R^N$ onto a subspace S^p , where $p < N$, $\mathbf{s} = H\mathbf{a}$ is obtained. The error term ϵ of the orthogonal projection yields the least-squares estimate a_{ls} :

$$\epsilon^T H = 0^T \Rightarrow H^T \epsilon = 0 \Rightarrow H^T(\mathbf{x} - H\mathbf{a}) = 0 \Rightarrow \mathbf{a} = \mathbf{a}_{ls} = (H^T H)^{-1} H^T \mathbf{x} \quad (16)$$

The Yule-Walker estimates are given by $\mathbf{a} = R_{xx}^{-1} \mathbf{r}_{xx}$, where $\mathbf{r}_{xx} = [r_{xx}(1), \dots, r_{xx}(p)]^T$ and R_{xx} is a Toeplitz matrix whose first column is $[r_{xx}(0), r_{xx}(1), \dots, r_{xx}(p-1)]^T$. If $M = p$, then the LS estimate is equivalent to the Yule-Walker estimate.

3.3.2 Observation Matrix H

The ACF values of \mathbf{x} form the matrix H . If \mathbf{x} is a deterministic signal, its ACF and hence H is also deterministic. If \mathbf{x} is a stochastic process or deterministic and corrupted by stochastic noise, then H is also stochastic.

3.3.3 LSE Coefficients

The values of the LS estimator \mathbf{a} for orders $p = 1, 2, \dots, 10$ are provided in Table 5.

Model Order	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}
1	0.7946									
2	1.6151	-0.9095								
3	2.3431	-2.1262	0.7018							
4	2.3096	-2.0331	0.6060	0.0377						
5	2.3015	-2.1623	0.9978	-0.3765	0.1650					
6	2.2316	-2.0045	0.5667	0.4992	-0.7039	0.3479				
7	2.1537	-1.8559	0.4736	0.3522	-0.2794	-0.0839	0.1767			
8	2.1229	-1.8459	0.5311	0.2885	-0.3827	0.2539	-0.1786	0.1487		
9	2.1259	-1.8497	0.5364	0.2806	-0.3773	0.2675	-0.2179	0.1892	-0.0169	
10	2.1276	-1.8582	0.5460	0.2681	-0.3597	0.2560	-0.2466	0.2721	-0.1023	0.0354

Table 4: LSE Coefficients for Sunspot Series

3.3.4 Approximation Error

The approximation error of the obtained AR models against the original data is plotted in Figure 47. By inspection of the graph for the model order selection criteria, it is evident that all three criteria have a minimum at $p = 3$, which is thus the most suitable order. This result is confirmed in the Mean Square Error (MSE) plot, as the error decreases significantly for models of order 3 or greater.

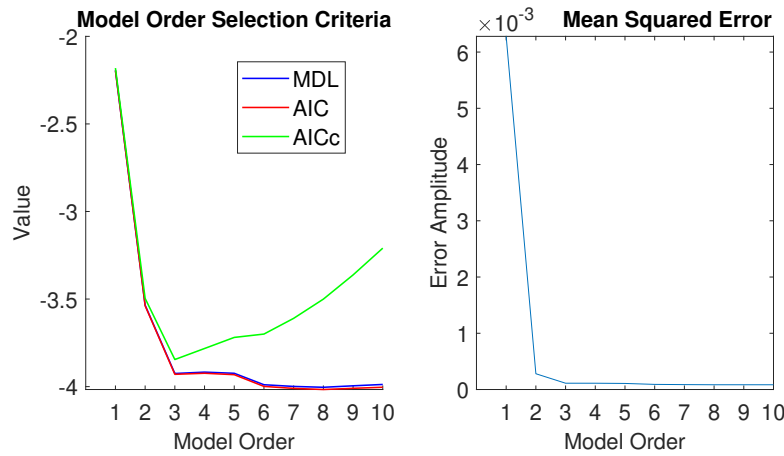


Fig. 47: Model order selection criteria and MSE plot for LSE sunspot models

3.3.5 AR(p) Power Spectra

The power spectra of the normalised sunspot series are plotted in Figure 48. The AR(1) model and periodogram perform poorly as they are unable to track the frequency peak. As expected, the AR(3) model is able to capture the peak. AR(10) also exhibits strong performance although it is not a suitable order as it leads to overfitting.

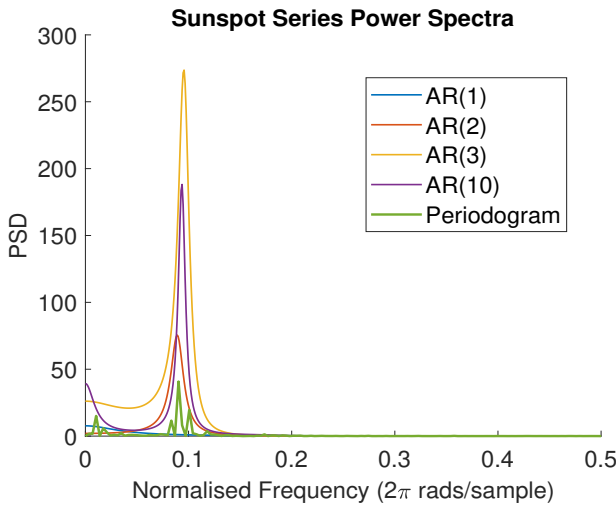


Fig. 48: Normalised Sunspot Series PSDs

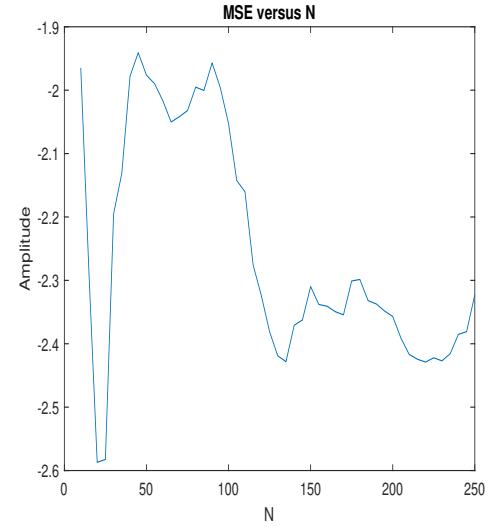


Fig. 49: MSE versus N

3.3.6 MSE Behaviour

Figure 49 depicts the logarithm of the approximation error as a function of data lengths $N \in [10 : 5 : 250]$. The MSE has a minimum at $N=20$, which is surprising given that the MSE is typically expected to decrease at large N . Data lengths in the range $[50, 100]$ result in a high approximation error, which subsequently decreases for larger N . The high error can be attributed to the inherent bias of the biased ACF, which is computed for every data length.

3.4 Spectrogram for Time-Frequency Analysis

3.4.1 London Landline Number

The Dual Tone Multi-Frequency system assigns the signal $y[n] = \sin(2\pi f_1 n) + \sin(2\pi f_2 n)$ to each keypad button, where f_1 and f_2 are each key's frequencies in Hz and n the discrete time index. The analysis below has been performed for the London landline number 020 6477 0733. Considering that the maximum frequency of all keys is 1477 Hz, the minimum Nyquist sampling frequency that should be used to avoid aliasing is 2954 Hz. Therefore, a sampling rate of 32768 guarantees that no aliasing can occur. In practice, oversampling is essential because critical sampling can still lead to aliasing. The entire signal y along with the waveforms for keys 0 and 2 is plotted in Figure 50.

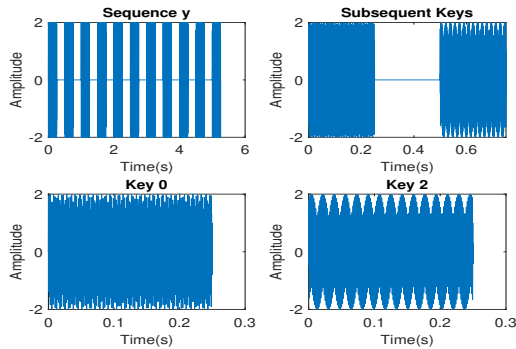


Fig. 50: Spectrogram of y

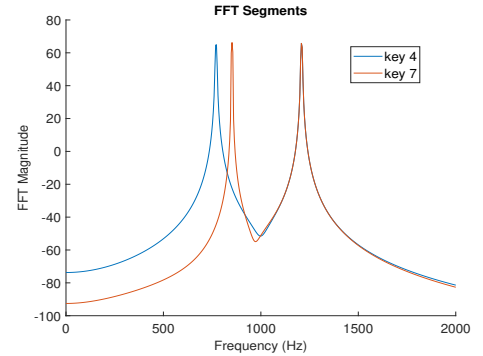


Fig. 51: FFT plots for keys 4,7

3.4.2 Spectrogram Analysis

The spectrogram of y is plotted in Figure 52. The spectrogram has been plotted using a Hamming window of 8192 samples, 8192 FFT points and no overlap between frames. The two horizontal yellow frequency bins contained in each 8192 sample-frame indicate which digit corresponds to each key press. Frames in time that only have dark blue bins correspond to idle time between key presses.

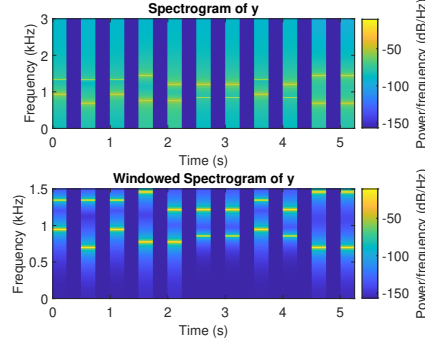


Fig. 52: Spectrogram of y

3.4.3 Identifying Keys

Identifying the two spectral peaks of each key and matching it to the data in Table 1 of the specification allows us to determine the sequence. For instance, the spectrum of keys 4 and 7 is plotted in Figure 51. The frequencies at which the Discrete Fourier Transform has peaks are approximately equal to the corresponding to the ones listed in the table.

3.4.4 Analysis with Channel Noise

The same analysis as above has been performed for channel variance of variance 0.01, 1 and 10. Due to the presence of noise, the idle time between key presses cannot be discerned in any of the three plots. Key classification is still possible in the low and medium variance cases because the characteristic yellow frequency bands can be seen in the spectrogram in each case. However, that is not the case in the high variance case since the yellow bands can be clearly distinguished from noise.

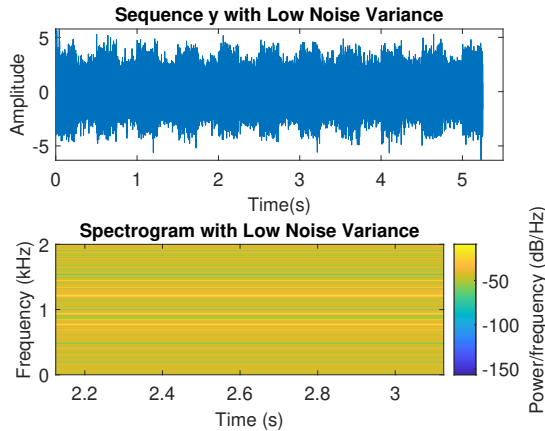


Fig. 53: Plots for noise variance $\sigma^2 = 0.01$

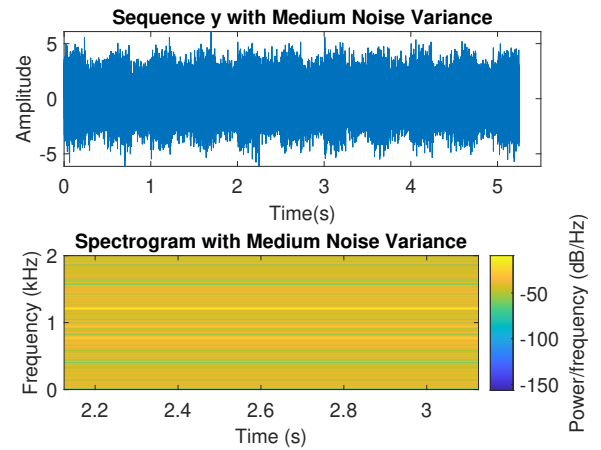


Fig. 54: Plots for noise variance $\sigma^2 = 1$

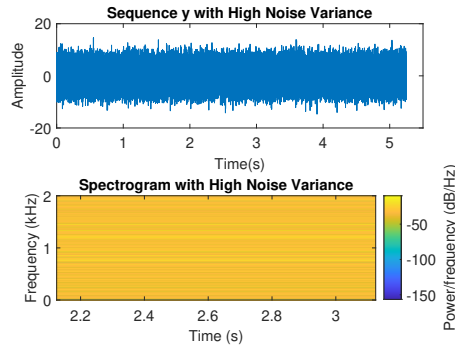


Fig. 55: Plots for noise variance $\sigma^2 = 10$

3.5 Respiratory Sinus Arrhythmia

Respiratory Sinus Arrhythmia (RSA) describes the modulation of cardiac function through respiration and is used to evaluate cardiovascular health. Figures 56-58 show the standard periodogram and the averaged periodogram with window lengths of 50s and 150s for all three trials. The data has been sampled at a frequency $f_s = 4Hz$.

The breathing rate in Trial 1 is unconstrained. The Trial 1 PSD has peaks at DC and at normalised frequency $f = 0.006Hz$. The 50s and 150s windows render the average periodogram smoother by removing noise since the windowed plots have significantly fewer oscillations. The periodogram of this trial exhibits multiple peaks with no clear pattern, which is attributed to the unconstrained nature of the breathing. The breathing rate in Trial 2 is 25 breaths per minute. Beyond the DC peak, Trial 2 PSD also peaks at $f = 0.015$ and $f = 0.026Hz$. The high number of oscillations in the frequency range $[0, 0.1]$ is explained by the fact that the breathing rate of Trial 2 is overly quick. Therefore, more respiratory effort is needed, which requires the heart rate to adjust appropriately. The breathing rate for Trial 3 was constrained to 7.5 breaths per minute. This PSD shows a distinct peak at $f = 0.030 Hz$, which is more clearly shown in the 50s window segment. Considering the breathing rate in this case is the lowest out of all three trials, it is expected that a peak will be observed at a higher frequency, which corresponds to longer frequencies of inhaling and exhaling. Therefore, the main difference between the PSDs is the location of the frequencies at which spectral peaks occur, which is attributed to the difference in breathing rates.

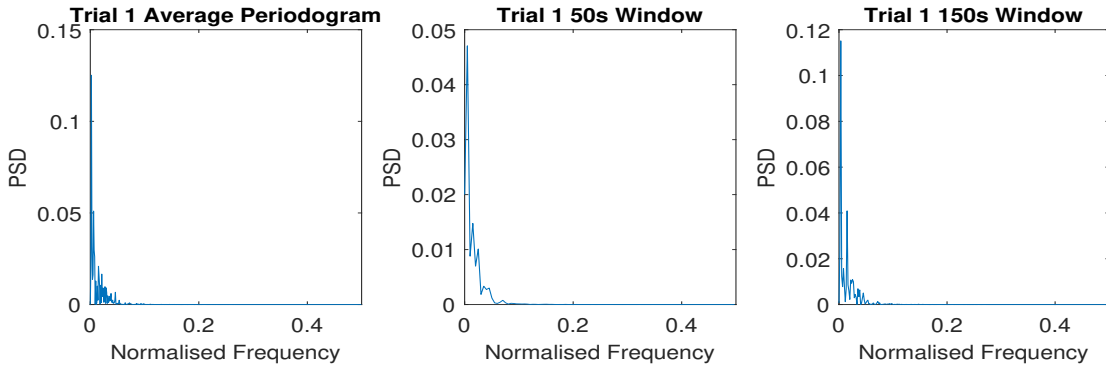


Fig. 56: xRRI1 periodogram and windowed periodograms

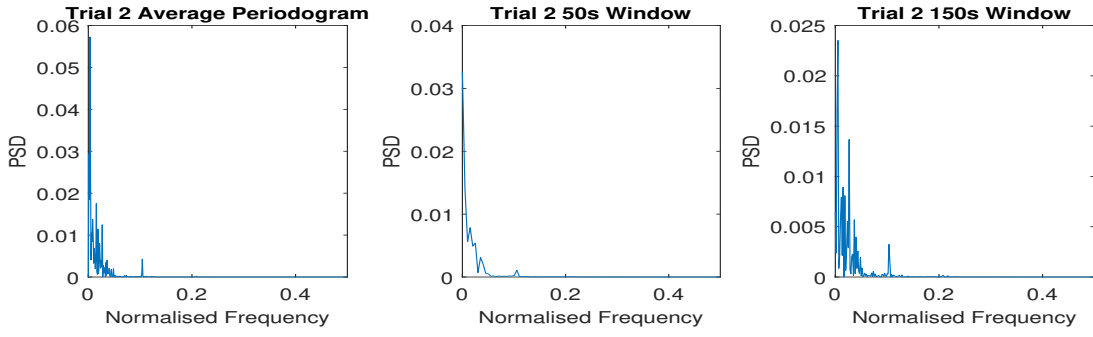


Fig. 57: xRRI2 periodogram and windowed periodograms

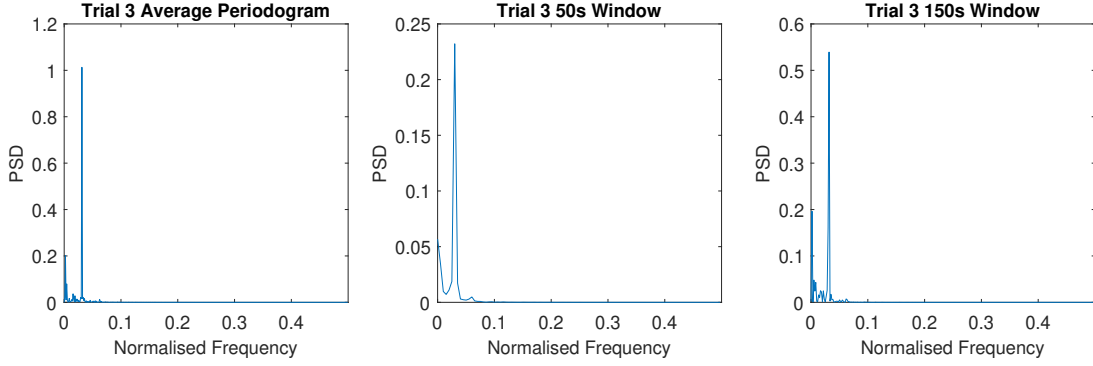


Fig. 58: xRRI3 periodogram and windowed periodograms

4 Optimal Filtering- Fixed and Adaptive

4.1 Wiener Filter

The 1000-sample random process \mathbf{x} is passed through an unknown system, which is modelled as a MA system with coefficients $b = [1, 2, 3, 2, 1]$ and $a = [1]$ and output \mathbf{y} . The output of the unidentified system corrupted by noise is denoted by \mathbf{z} . The output of the optimum filter, whose coefficients are found using the Wiener solution:

$$\mathbf{w}_{\text{opt}} = \mathbf{R}_{\mathbf{xx}}^{-1} \cdot \mathbf{p}_{\mathbf{zx}} \quad (17)$$

The cross-correlation and auto-correlation matrices are defined as follows:

$$\mathbf{p}_{\mathbf{zx}} = [r_{zx}(0), r_{zx}(1), \dots, r_{zx}(N_w)]^T \quad (18)$$

$$\mathbf{R}_{\mathbf{xx}} = \begin{bmatrix} r_{xx}(0) & r_{xx}(-1) & \dots & r_{xx}(N_w) \\ \vdots & \vdots & \dots & \vdots \\ r_{xx}(N_w) & r_{xx}(N_w - 1) & \dots & r_{xx}(0) \end{bmatrix} \quad (19)$$

4.1.1 Optimal Wiener Filter Coefficients

The optimal coefficients of the Wiener filter are provided in Table 5 for \mathbf{y} and \mathbf{y} after normalisation. The coefficients in the non-normalised case are quite similar to those of the unknown system.

Case	w[1]	w[2]	w[3]	w[4]	w[5]
Non normalised	0.9975	1.9871	2.9639	1.9443	0.9804
Normalised	0.2307	0.4446	0.6685	0.4500	0.2242

Table 5: Wiener filter solution coefficients

The ideal SNR for $z[n]$ after normalising $y[n]$ is calculated as:

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

The experimental value is $SNR = 20.4620$ db, which is close to the theoretical calculation. The autocorrelation and cross-correlation matrices have the following values:

$$\mathbf{R}_{xx} = \frac{1}{5} \cdot \begin{bmatrix} 0.8798 & 0.0027 & -0.060 & 0.0265 & -0.003 \\ 0.0027 & 0.8798 & 0.0027 & -0.060 & 0.0265 \\ -0.060 & 0.0027 & 0.8798 & 0.0027 & -0.060 \\ 0.0265 & -0.060 & 0.0027 & 0.8798 & 0.0027 \\ -0.003 & 0.0265 & -0.060 & 0.0027 & 0.8798 \end{bmatrix} \quad (21)$$

$$\mathbf{p}_{zx} = \frac{1}{5} \cdot [0.7514, 1.6703, 2.5266, 1.6743, 0.7544]^T \quad (22)$$

4.1.2 Examining Effect of Varying Noise Power

σ^2	SNR (dB)	1	2	3	2	1
0.1	10.3254	0.2363	0.4858	0.7100	0.4625	0.2118
1	2.7711	0.2479	0.4455	0.6926	0.4237	0.2177
3	1.1265	0.1460	0.5015	0.7031	0.57586	0.2803
6	0.7709	0.0259	0.4950	0.7418	0.6117	0.3633
9	-4.77	0.0796	0.4036	0.6670	0.5275	0.2838
10	-9.75	0.04914	0.4036	0.8440	0.3677	0.2455

Table 6: SNRs and Wiener solutions for different noise variances

Table 6 summarises the effects of varying the noise variance in the range $[0.1, 10]$. The calculations have been performed with \mathbf{y} normalised, which means the $w[i]$ coefficients must be scaled by 5. The Wiener coefficients' accuracy and the SNR decrease as the variance of the noise grows, as expected. Using $N_w > 4$, the Wiener solution grows in size as it has additional $N_w - 5$ terms, which are insignificant as they tend to 0. For instance, the choice of $N_w = 7$ yields $w = [0.98494, 2.0059, 3.0057, 1.9937, 1.0126, -0.0123, 0.0063]$. As it can be observed, the last 2 coefficients are negligible.

4.1.3 Wiener Solution Complexity

N denotes the number of samples of $x[n]$ and N_w the number of samples in the Wiener matrix w . Constructing r_{zx} requires $(N-1)$ additions and N multiplications to be performed N_w times, leading to a complexity $O(N_w N(N-1)) = O(N_w N^2)$. Creating the R_{xx} matrix requires also $O(N_w N^2)$ operations, since the only operation being done is re-ordering the r_{zx} to obtain a Toeplitz matrix. The matrix product and the R_{xx}^{-1}

computation have a complexity of $O(N_w^2)$ and $O(N_w^3)$ respectively.

Hence, the total complexity is $O(N_w N^2) + O(N_w N^2) + O(N_w^2) + O(N_w^3) = O(N_w N^2) + O(N_w^3)$. If $N_w \ll N$, then this can be stated as $O(N^2)$.

4.2 Least Mean Square Algorithm

The following listing shows the implementation of the "lms.m" routine.

```

1 function [y_hat, error, evolution] = lms(x, z, mu, ord)
2     x=x'; z=z';
3     N=length(x);
4     w=zeros(N, ord);
5     y_hat=zeros(1, N);
6     error=zeros(1, N);
7     for n=ord+1:N
8         x_hat=x(n:-1:n-(ord-1));
9         y_hat(1, n) = dot(w(n-ord, :), x_hat);
10        error(1, n) = z(1, n) - y_hat(1, n);
11        temp = mu * error(1, n) * x_hat;
12        w(n+1, :) = w(n, :) + temp;
13    end
14    w_out = w(height(w), :); %display final coefficients
15    evolution=w;
16 end

```

4.2.1 LMS Routine and Learning Rate

Adaptive filters are used to approximate non-stationary signals. The simplest adaptive filter is implemented through the Least Mean Square(LMS) algorithm, which is defined by $\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e[n]\mathbf{x}(n)$, where μ is the adaptation gain and $e[n]$ the error between $z[n]$ and $\hat{y}[n]$.

The plots below show the evolution of the Wiener coefficients and the LMS error for $\mu = 0.01$. The LMS error appears large at the first 50 time iterations because the algorithm requires time to converge to the appropriate solution, given that the initial coefficients are initialised to 0. As the coefficients begin to converge to their final values, the LMS error decreases significantly, eventually tending to 0. The solution converges approximately after the 600th iteration.

Increasing μ helps the algorithm converge faster to a solution, but sacrifices accuracy and smoothness. In contrast, decreasing μ leads to a gradual smooth convergence to the ideal values. If μ is chosen to be too large, the coefficients will oscillate rapidly in the final iterations, which suggests they may diverge from the optimal solution.

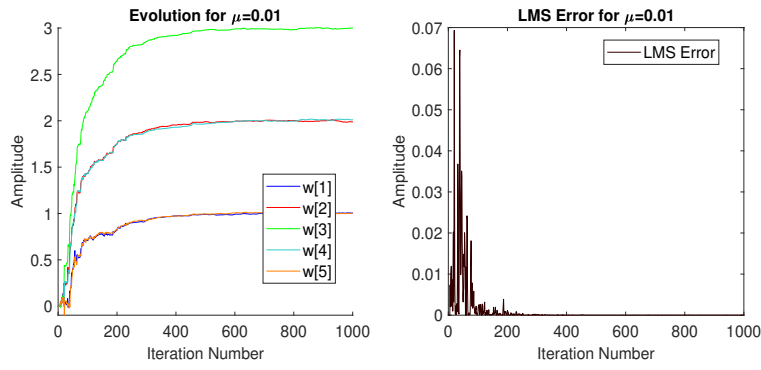


Fig. 59: Wiener coefficients for $\mu = 0.01$

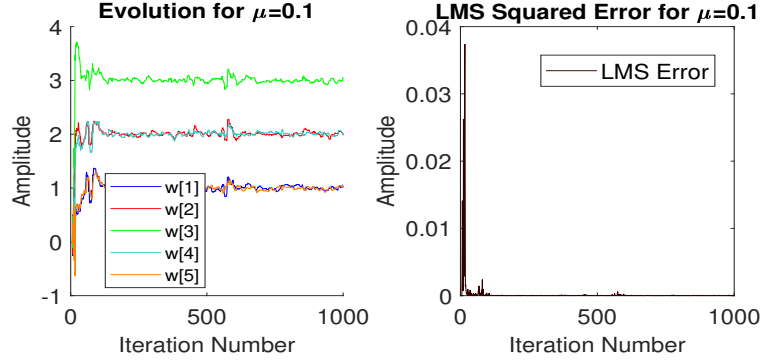


Fig. 60: Wiener coefficients for $\mu = 0.1$

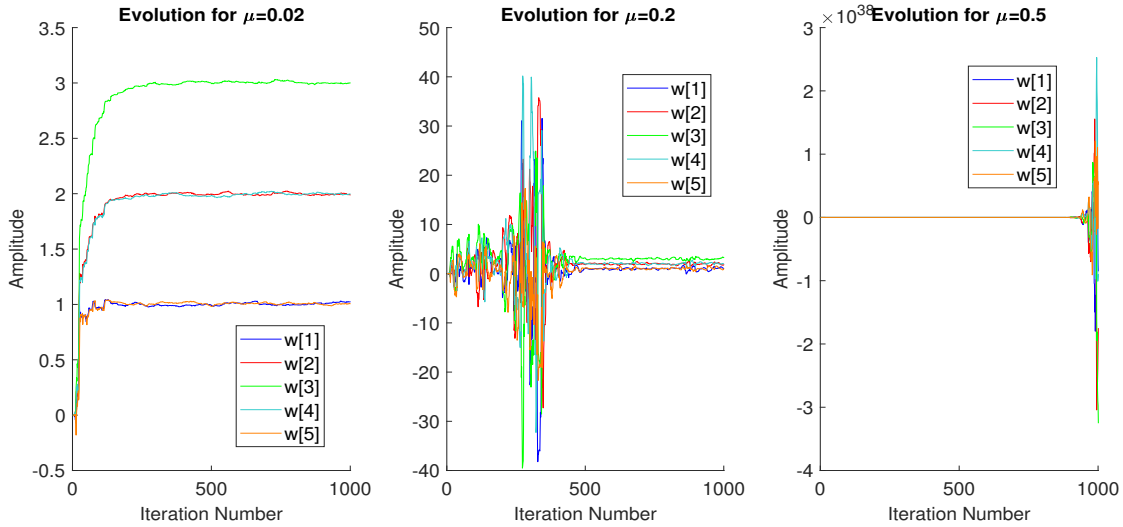


Fig. 61: Plots for $\mu \in [0.02, 0.5]$

4.2.2 Computational Complexity of LMS

For the LMS algorithm, the output of the unknown system is given by $y[n] = \mathbf{w}_n^T \mathbf{x}[n]$. Considering that $\mathbf{w}[n] = [w[n], w[n-1], \dots, w[n-N_w]]^T$ and $\mathbf{x}[n] = [x[n], x[n-1], \dots, x[n-N_w]]^T$, where N_w is the Wiener

solution length, the dot product computation has a complexity of $O(N_w)$. For one iteration, the coefficient update requires $(N_w + 1)$ operations, which translates into $O(N_w)$ complexity. As the process is repeated N times, the total LMS complexity is $O(NN_w)$.

4.3 Gear Shifting

Gear shifting is employed in order to obtain improved steady-state estimates for stationary signals. The adaptation gain μ is updated every time based on the error $e[n]$. Specifically, if $e[n] > e[n - 1]$, the adaptation gain is increased by 20% so that the convergence is accelerated. Conversely, if $e[n] < e[n - 1]$, μ is decreased by 20%, since decreasing error indicates the algorithm is closer to converging. Hence, it is preferable to lower μ to ensure accuracy. When comparing Figure 60 and 58, it can be seen that the gear-shifting algorithm converges to a solution at iteration 150, while the standard LMS requires more than 200 iterations to converge. Consequently, the gear shifting method chosen achieves faster convergence.

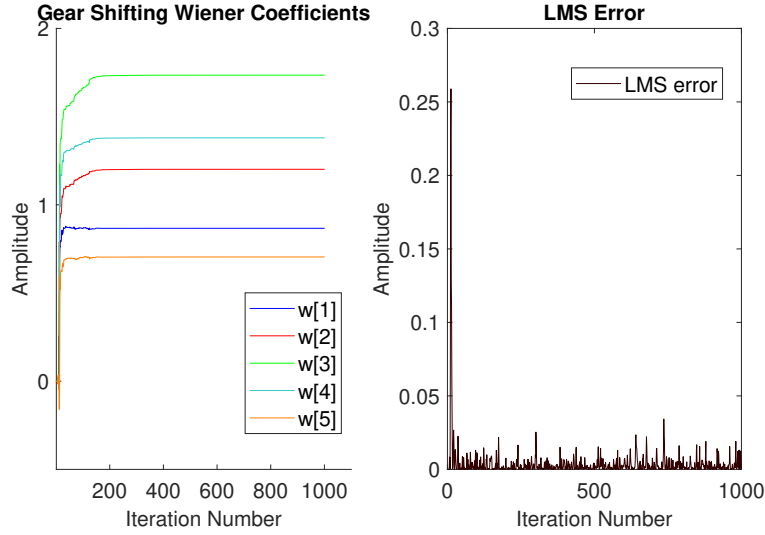


Fig. 62: Wiener coefficients evolution

4.4 Identification of AR Processes

The second order predictor shown in Figure 5 of the coursework specification is constructed using an AR(2) model with coefficients $a = [1, 0.9, 0.2]^T$. A successful LMS algorithm will yield coefficients $a_1 = -0.9$ and $a_2 = -0.2$, since MATLAB's `filter` routine inverts the sign of the mathematically defined coefficients. The evolution of a_1 and a_2 for $\mu = 0.01, 0.05, 0.1, 0.3$ is plotted in Figure 63. Choosing an adaptation gain greater or equal than 0.1 results in significantly higher error, but not in faster convergence. Therefore, choosing $\mu = 0.01$ is the best choice in terms of accuracy out of the learning rates used.

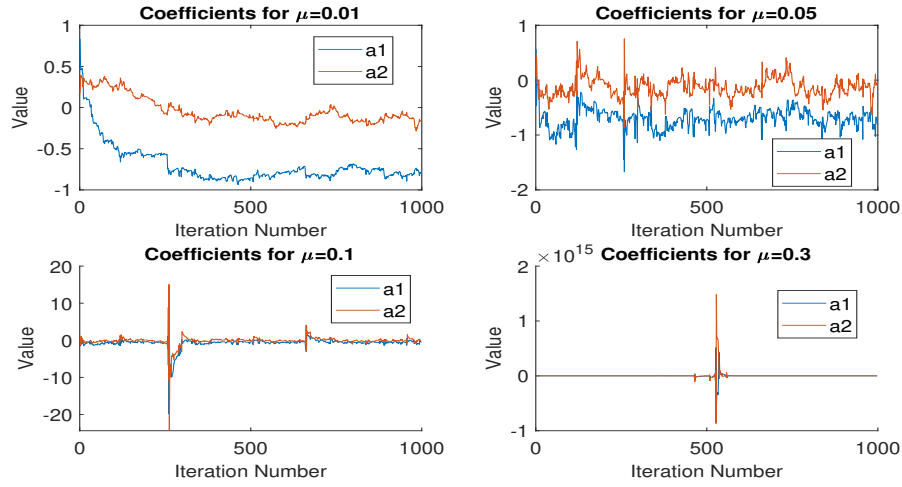


Fig. 63: Evolution of a_1 and a_2 for 4 different adaptation gains

4.5 Speech Recognition

4.5.1 Selection of Adaptation Gain and Predictor Order

A p^{th} order predictor is used to calculate the adaptive filter coefficients. I have recorded my voice pronouncing the letters "a", "e", "s", "t", and "x" and each file will be used as an input to test the predictor's performance. Table 7 summarises the choices of μ that result in the LMS algorithm converging for order 10, which balances computational complexity with accuracy. The high sampling rate of $f_s = 44.1$ kHz suggests that it is difficult to predict accurately with a sample size of 1000. Therefore, it is possible that a recording will require an adaptation gain close or higher than 1, which is not usually preferred.

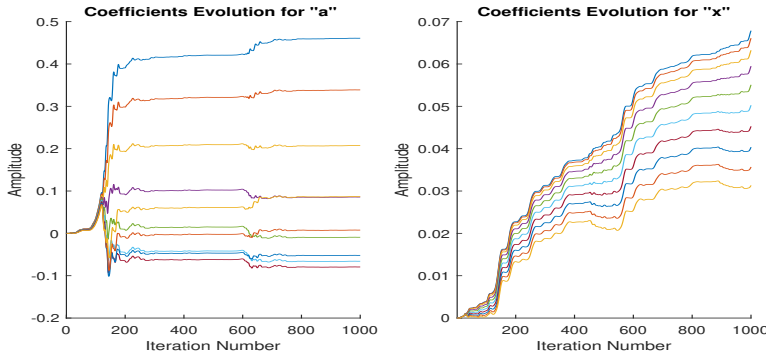


Fig. 64: Coefficients evolution of a and x for respective μ values

Letter	μ
a	0.6
e	5
s	0.2
t	0.1
x	0.05

Table 7: Adaptation gain choices

For instance, "e" requires a learning rate close to 5 in order for the algorithm to converge, while the sample file for "a" requires a μ of 0.6. The remaining letters require common learning rate values that are in the range $[0.001, 0.2]$.

4.5.2 Optimal Filter Length

A heuristic approach to obtaining the optimal filter length is varying the predictor order and the adaptation gain. However, real-time applications often have specific timing constraints, which effectively set an upper

bound on the order chosen. Therefore, the analytical approach of evaluating the MDL and AIC for a range of model orders identifies the minimum possible order. The order for which both criteria present a minimum is chosen. The possibilities for all letters are listed in Table 8.

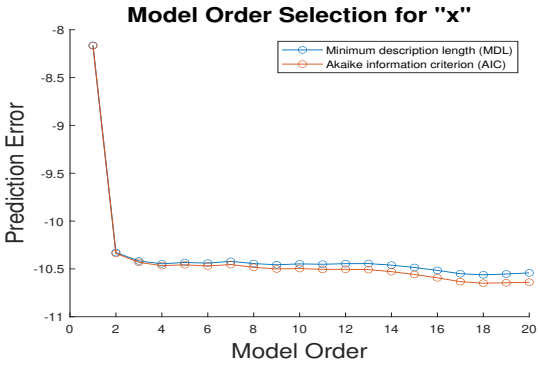


Fig. 65: MDL and AIC plot for "x"

Letter	Order
a	3
e	5
s	6
t	10
x	9

Table 8: Minimum order selection from AIC and MDL for all letters

The order p that is being chosen must be a minimum for both MDL and AIC. In the case of letter x, the absolute global minimum for both is observed at order $p = 9$. This selection process is repeated for every letter. The prediction gain, which depends on μ and the model order is another quantitative selection method. It is defined as $R_p = 10\log_{10}(\frac{\sigma_x^2}{\sigma_e^2})$.

4.5.3 Switching Sampling Frequency

Changing the sampling frequency to $f_s = 16$ kHz introduces higher quantisation error because every frame is composed of fewer samples. Due to the performance degradation, higher model orders will be required. Table 9 lists the minimum recommended orders based on the MDL and the AIC.

Letter	Minimum Order
a	13
e	9
s	16
t	15
x	11

Table 9: Minimum order selection based on MDL, AIC for $f_s = 16kHz$

Another effect of reducing the sampling frequency is lower prediction gain. Table 10 illustrates the degradation of this parameter for the letter "e" by comparing the prediction gain values for specific orders.

	p=3	p=5	p=9
fs=44.1 kHz	7.5008	8.5863	8.8855
fs=16 kHz	3.1899	3.8969	4.0839

Table 10: Prediction gain values comparison

4.6 Sign Algorithms

The following sign LMS algorithms constitute a simplified implementation of LMS and are defined as:

- signed-error: $\mathbf{w}(n+1) = \mathbf{w}(n) + \mu \text{sign}(e[n])\mathbf{x}(n)$
- signed-regressor: $\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e[n] \text{sign}(\mathbf{x}(n))$
- sign-sign: $\mathbf{w}(n+1) = \mathbf{w}(n) + \mu \text{sign}(e[n]) \text{sign}(\mathbf{x}(n))$

Figure 65 shows how the sign LMS algorithms compare to the standard LMS when used in the AR(2) system in section 4.4. Although the standard algorithm performs better than the sign versions, they all converge to the correct values. This suggests that sacrificing accuracy for lower computational complexity is a worthwhile compromise for this system.

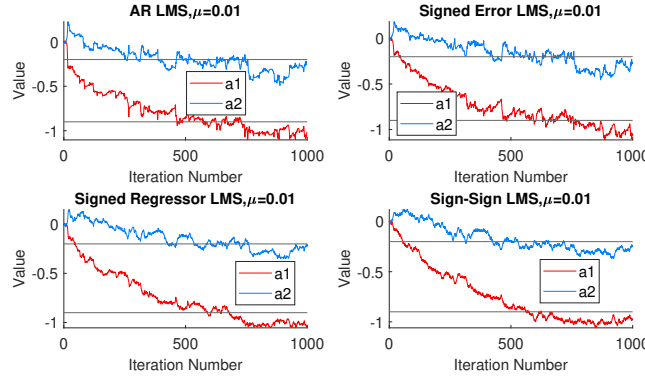


Fig. 66: Comparison of LMS algorithms for AR(2) Predictor

5 MLE for the Frequency of a Signal

5.1 Mapping $\mathbf{J}(\theta)$

The Maximum Likelihood Estimator (MLE) of the parameter vector $\theta = [A, f_0, \phi]^T$ is obtained by finding the θ which maximises the log-likelihood function. An alternative approach is to find θ such that the cost function J is minimised. J is defined as follows:

$$J(\theta) = \sum_{n=0}^{N-1} (x[n] - A \cos(2\pi f_0 n + \phi))^2 \quad (23)$$

The parametric pdf of \mathbf{x} is stated in equation (44) of the specification. Using the trigonometric expansion $\cos(k+m) = \cos(k)\cos(m) - \sin(k)\sin(m)$ and letting $\alpha_1 = A \cos(\phi)$ and $\alpha_2 = -A \sin(\phi)$ with $\alpha = [\alpha_1, \alpha_2]^T$, J can be rewritten as:

$$J(\theta) = \sum_{n=0}^{N-1} (x[n] - A \cos(2\pi f_0 n + \phi))^2 = \sum_{n=0}^{N-1} (x[n] - A \cos(2\pi f_0 n) \cos(\phi) + A \sin(2\pi f_0 n) \sin(\phi))^2 \quad (24)$$

$$= \sum_{n=0}^{N-1} (x[n] - \alpha_1 \cos(2\pi f_0 n) - \alpha_2 \sin(2\pi f_0 n))^2 \quad (25)$$

The inverse transformation pair is $A = \sqrt{a_1^2 + a_2^2}$ and $\phi = \tan^{-1}(-\frac{a_2}{a_1})$.

By defining $c = [1, \cos(2\pi f_0), \dots, \cos(2\pi f_0(N-1))]^T$ and $s = [1, \sin(2\pi f_0), \dots, \sin(2\pi f_0(N-1))]^T$, \mathbf{J} can be expressed as $J(\theta) = (x - \alpha_1 \mathbf{c} - \alpha_2 \mathbf{s})(x - a_1 \mathbf{c} - a_2 \mathbf{s})^T$. \mathbf{H} is formed by using \mathbf{c} and \mathbf{s} as columns:

$$H = \begin{bmatrix} 1 & 0 \\ \cos(2\pi f_0) & \sin(2\pi f_0) \\ \vdots & \vdots \\ \cos(2\pi f_0(N-1)) & \sin(2\pi f_0(N-1)) \end{bmatrix} \quad (26)$$

Combining all the above equations, we obtain:

$$J(\theta) = (\mathbf{x} - \mathbf{H}\alpha)^T(\mathbf{x} - \mathbf{H}\alpha) = J'(\hat{\alpha}, f_0) \quad (27)$$

5.2 Minimising Solution

If \mathbf{x} is described by the linear model $x = H\theta + w$, the MLE estimator is $\hat{\theta}_{mle} = (\mathbf{H}^T \mathbf{c}^{-1} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{c}^{-1} \mathbf{x}$. Using the result of the previous subsection, the minimising α is $\hat{\alpha} = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{x}$.

\mathbf{J}' is rewritten as follows when $\hat{\alpha}$ is optimal:

$$\mathbf{J}'(\hat{\alpha}_1, \hat{\alpha}_2, f_0) = (\mathbf{x} - \mathbf{H}\alpha)^T(\mathbf{x} - \mathbf{H}\alpha) = (\mathbf{x} - \mathbf{H}(\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{x})^T (\mathbf{x} - \mathbf{H}(\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{x}) \quad (28)$$

$$= (\mathbf{I}\mathbf{x} - \mathbf{H}(\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{x})^T (\mathbf{I}\mathbf{x} - \mathbf{H}(\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{x}) \quad (29)$$

$$= \mathbf{x}^T (\mathbf{I} - \mathbf{H}(\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T)^T (\mathbf{I} - \mathbf{H}(\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T) \mathbf{x} \quad (30)$$

Using the fact that $P = (\mathbf{I} - \mathbf{H}(\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T) = P^2$, which means P is idempotent, \mathbf{J}' can be re-written as:

$$\mathbf{J}'(\hat{\alpha}_1, \hat{\alpha}_2, f_0) = \mathbf{x}^T \mathbf{H}(\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{x} \quad (31)$$

Consequently, minimising equation 31 requires maximising the matrix expression.

5.3 MLE of f_0

The problem has been simplified to finding f_0 that maximises $J(f_0) = \mathbf{x}^T \mathbf{H}(\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{x}$. Using the definition of $\mathbf{H} = [\mathbf{c}, \mathbf{s}]$, the components of the matrix formula can be written as:

$$\mathbf{x}^T \mathbf{H} = (\mathbf{x} \mathbf{H}^T)^T = \begin{bmatrix} \mathbf{c}^T \mathbf{x} \\ \mathbf{s}^T \mathbf{x} \end{bmatrix} \quad (32)$$

$$\mathbf{H}^T \mathbf{H} = \begin{bmatrix} \mathbf{c}^T \\ \mathbf{s}^T \end{bmatrix} \begin{bmatrix} \mathbf{c}^T \mathbf{s}^T \end{bmatrix} = \begin{bmatrix} \mathbf{c}^T \mathbf{c} & \mathbf{c}^T \mathbf{s}^T \\ \mathbf{s}^T \mathbf{c} & \mathbf{s}^T \mathbf{s} \end{bmatrix} \quad (33)$$

$$\mathbf{H}^T \mathbf{x} = \begin{bmatrix} \mathbf{c}^T \\ \mathbf{s}^T \end{bmatrix} \mathbf{x} = \begin{bmatrix} \mathbf{c}^T \mathbf{x} \\ \mathbf{s}^T \mathbf{x} \end{bmatrix} \quad (34)$$

Hence $J(f_0)$ is re-expressed:

$$J(f_0) = \begin{bmatrix} \mathbf{c}^T \mathbf{x} \\ \mathbf{s}^T \mathbf{x} \end{bmatrix}^T \begin{bmatrix} \mathbf{c}^T \mathbf{c} & \mathbf{c}^T \mathbf{s} \\ \mathbf{s}^T \mathbf{c} & \mathbf{s}^T \mathbf{s} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{c}^T \mathbf{x} & \mathbf{s}^T \mathbf{x} \end{bmatrix}^T \approx \begin{bmatrix} \mathbf{c}^T \mathbf{x} \\ \mathbf{s}^T \mathbf{x} \end{bmatrix}^T \begin{bmatrix} \frac{N}{2} & 0 \\ 0 & \frac{N}{2} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{c}^T \mathbf{x} \\ \mathbf{s}^T \mathbf{x} \end{bmatrix} \quad (35)$$

The approximation used in the last step holds as long as f_0 does not approach either 0 or 1/2. If f_0 approaches these values, the sine and cosine components of \mathbf{H} will tend to 0, which will render \mathbf{H} non-invertible and the computation impossible. Inverting the matrix in the above formulation yields:

$$\begin{bmatrix} \mathbf{c}^T \mathbf{c} & \mathbf{c}^T \mathbf{s} \\ \mathbf{s}^T \mathbf{c} & \mathbf{s}^T \mathbf{s} \end{bmatrix}^{-1} = \begin{bmatrix} \frac{N}{2} & 0 \\ 0 & \frac{N}{2} \end{bmatrix}^{-1} = \begin{bmatrix} \frac{2}{N} & 0 \\ 0 & \frac{2}{N} \end{bmatrix} \quad (36)$$

By combining equations 35 and 36, the following result is obtained:

$$\begin{bmatrix} \mathbf{c}^T \mathbf{x} \\ \mathbf{s}^T \mathbf{x} \end{bmatrix}^T \begin{bmatrix} \frac{2}{N} & 0 \\ 0 & \frac{2}{N} \end{bmatrix} \begin{bmatrix} \mathbf{c}^T \mathbf{x} \\ \mathbf{s}^T \mathbf{x} \end{bmatrix} \quad (37)$$

5.4 Periodogram and MLE Estimate

$J(f_0)$ can be expressed as follows by expansion of the matrix expression:

$$J(f_0) = \frac{2}{N} \left[\left(\sum_{n=0}^{N-1} x[n] \cos(2\pi f_0 n) \right)^2 + \left(\sum_{n=0}^{N-1} x[n] \sin(2\pi f_0 n) \right)^2 \right] \quad (38)$$

Equation 38 is a sum of sine and cosine terms, which can be rewritten as a complex exponential using Euler's identity ($e^{j\theta} = \cos(\theta) + j\sin(\theta)$):

$$J(f_0) = \frac{2}{N} \left| \sum_{n=0}^{N-1} x[n] e^{-j2\pi \frac{f_0 n}{N}} \right|^2 \quad (39)$$

Therefore, the MLE estimate \hat{f}_0 is obtained by maximising the periodogram estimate $\hat{P}_x(f)$.