

Statistical Signal Processing and Inference Coursework

James Liu

CID: 01871216

Contents

1	Random Signals and Stochastic Processes	3
1.1	Statistical Estimation	3
1.2	Stochastic Processes	6
1.3	Estimation of probability distribution	9
2	Linear Stochastic Modelling	11
2.1	Autocorrelation of uncorrelated and correlated sequences	11
2.2	Cross-correlation function	13
2.3	Autoregressive Modelling	14
2.4	Cramer-Rao Lower Bound	16
2.5	Real world signals: ECG from iAmp experiment	17
3	Spectral Estimation	19
3.1	Average periodogram estimates	19
3.2	Spectrum of autoregressive processes	20
3.3	The Least Squares Estimation (LSE) of AR Coefficients	22
3.4	Spectrogram for time-frequency analysis: dial tone pad	24
3.5	Respiratory sinus arrhythmia from RR-Intervals	26
4	Optimal filtering – fixed and adaptive	27
4.1	Wiener filter	27
4.2	The least mean square (LMS) algorithm	28
4.3	Gear Shifting	29
4.4	Identification of AR Process	30
4.5	Speech recognition	31
4.6	Dealing with computational complexity, sign algorithms	34

1 Random Signals and Stochastic Processes

1.1 Statistical Estimation

Below, a boxplot of a 1000-sample vector generated from the uniform random variable $X \sim U(0, 1)$ was plotted.

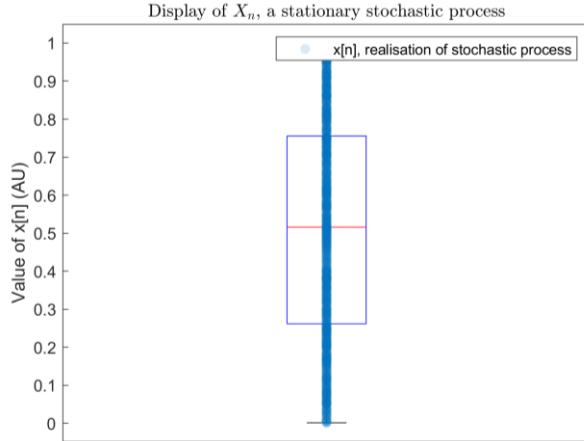


Figure 1: Boxplot of realisations $x[n]$, overlayed with actual data points.

The probability density function or pdf, $f(x)$, assumes the form:

$$f(x) = \begin{cases} 1 & \text{if } 0 < x < 1 \\ 0 & \text{otherwise} \end{cases}$$

1.1.1 Theoretical and sample mean

The theoretical mean can be calculated using the expectation operator,

$$E(X) = \int_{-\infty}^{+\infty} \alpha f(\alpha) d\alpha \quad (1)$$

where $f(x)$ is the pdf of the random variable. Since X is uniformly distributed between 0 and 1, and the pdf has a constant value of 1 in that interval, and zero elsewhere so we can calculate:

$$E(X) = \int_0^1 \alpha d\alpha = \frac{1}{2} \quad (2)$$

The calculated sample mean, \hat{m} was 0.5084, which is a 1.68% error from the true value of 0.5. Since the sample mean is taken from a finite number of samples it can only approximate the true expected value of X , and there is uncertainty associated with this estimate. Due to the central limit theorem, since we have 1000 samples, the sample mean will tend towards being normally distributed, $\bar{X} \sim N(\mu_X, \sigma_X^2/n)$, with the variance of the sample mean proportional to the reciprocal of n , the number of samples. Therefore, as the number of samples increases, the uncertainty of the estimator decreases.

1.1.2 Theoretical and Sample Standard Deviation

To calculate the theoretical standard deviation, we use the definition for variance:

$$Var[X] = E[(X - E(X))^2] \quad (3)$$

$$\begin{aligned} Var[X] &= E[X^2] - E[X]^2 \\ E[X^2] &= \int_0^1 x^2 dx = \left[\frac{x^3}{3} \right]_0^1 = \frac{1}{3} \\ Var[X] &= \frac{1}{3} - 0.5^2 = \frac{1}{12} \\ \sigma &= \sqrt{\frac{1}{12}} \end{aligned}$$

The sample standard deviation, $\hat{\sigma}$ was 0.2920, a 1.15% error on the theoretical value. This aligns with our expectations that the sample standard deviation is an unbiased estimator of the true standard deviation, after having applied Bessel's correction.

1.1.3 Bias of estimations

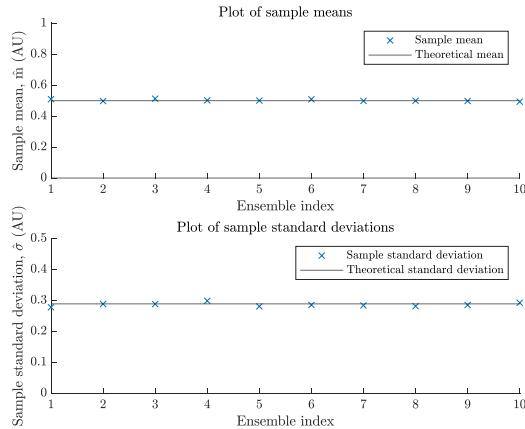


Figure 2: Scatter displaying the clustering of sample statistics around their theoretical values.

The sample mean and standard deviation are both unbiased estimators, meaning that $E(X - \hat{m}) = 0$. As a result, we should expect those statistics to be clustered around their respective theoretical values. As seen in *Figure 2*, this is shown in our observations, with the sample means and standard deviations being centred on their respective theoretical values of 0.5 and $1/\sqrt{12}$. The maximum error was ± 0.0115 for \hat{m} and ± 0.0099 for $\hat{\sigma}$.

1.1.4 Approximating the pdf

By plotting a histogram and normalising the frequencies by the total number of samples, we are able to generate an approximation of the pdf for the random variable, X . In *Figure 3*, the theoretical pdf for the standard uniform distribution is also displayed for comparison.

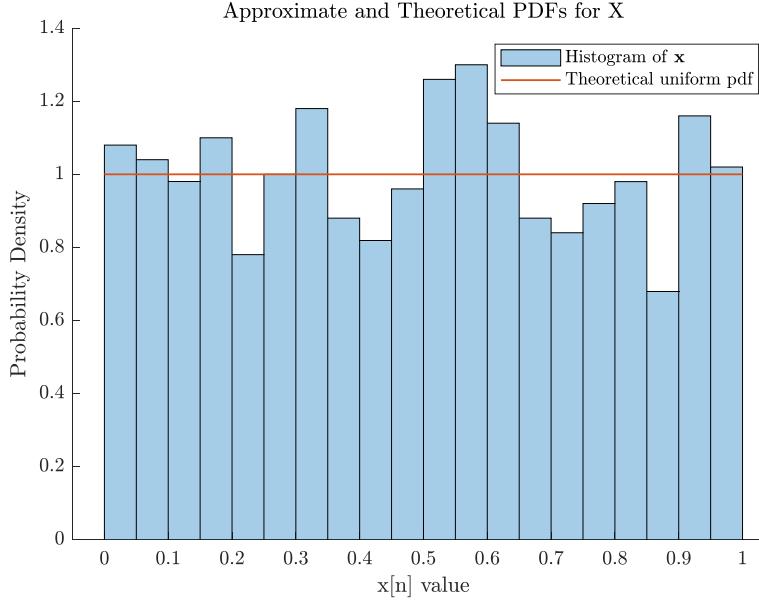


Figure 3: Histogram plot for a realisation of the stochastic process, X .

As the number of generated samples increases, the approximate pdf tends towards the theoretical pdf of a horizontal line. This is because the single realisation, $x[n]$, is better able to represent the entire stochastic process. For example, since this process is ergodic, as the number of samples increases, the time average becomes closer to averaging over an infinite number of samples.

1.1.5 Gaussian random variables

Now, using the `randn` function to generate samples from a zero-mean, unit-variance Gaussian distribution, we may repeat the analysis carried out above.

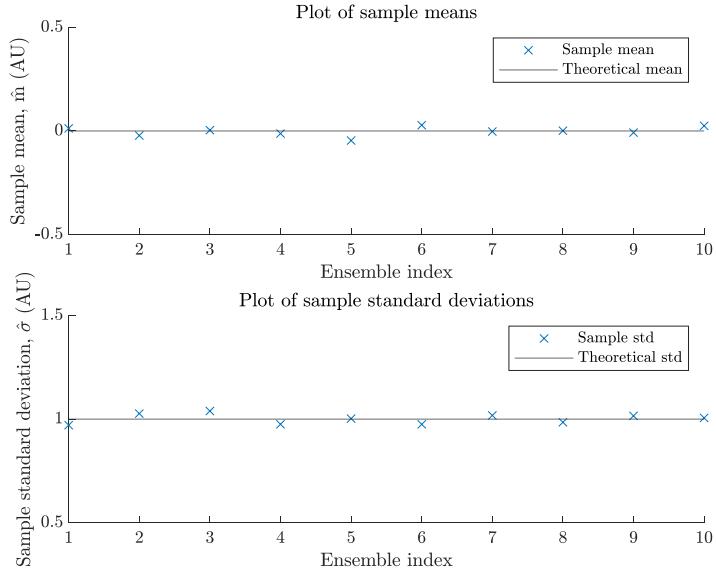


Figure 4: Scatter displaying the bias of the sample means and standard deviations.

After generating a 1000-sample realisation of this new stochastic process, the sample mean and standard deviation were 0.0319 and 0.9675 respectively. This corresponds to ± 0.0319 error on the theoretical mean value of 0, and ± 0.0325 error on the theoretical standard deviation of 1. These errors again align with our expectations that the estimators are unbiased. This is further supported by *Figure 4*, with the tight clustering of \hat{m} and $\hat{\sigma}$ to their respective theoretical values.

Finally, the approximate pdf was then plotted in *Figure 5*. We observe again that histogram approximates the theoretical Gaussian pdf, and converges as the number of samples increases.

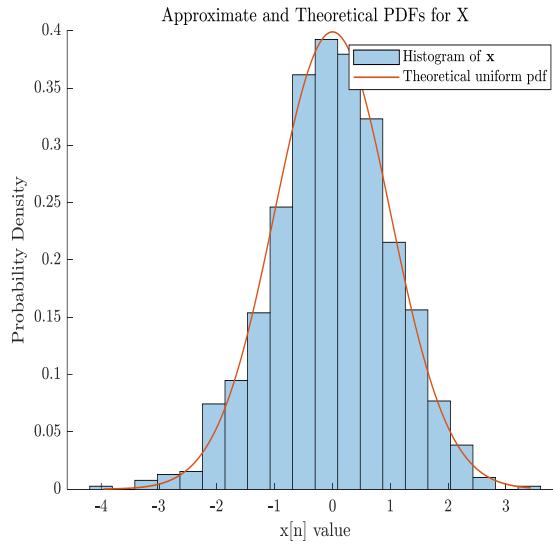


Figure 5: Histogram displaying the approximation of the theoretical Gaussian PDF.

1.2 Stochastic Processes

1.2.1 Stationarity

On the next page, plots have been generated to display the ensemble means and standard deviations over time for ensembles from each of the three random processes. For each random process, an ensemble of $M=100$ members, each with $N=100$ samples have been generated.

In *Figure 6*, we observe that the ensemble mean increases approximately linearly over time, and the standard deviation has a quadratic dependence. As a result, we conclude that random process 1 (rp1) is not stationary, since these second order moments are not constant over time.

In both *Figure 7* and *Figure 8*, the ensemble statistics appear to be uniformly distributed over time. They are not however exactly constant over time. We may still hypothesise that these random processes are stationary as we should expect a degree of uncertainty in ensemble statistics since we are considering a finite number of members in the ensembles.

Ensemble average and standard deviation over time for rp1

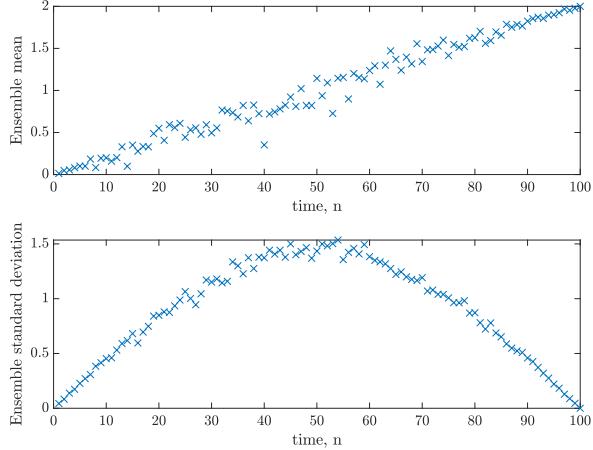


Figure 6: rp1 ensemble statistics over time.

Ensemble average and standard deviation over time for rp2

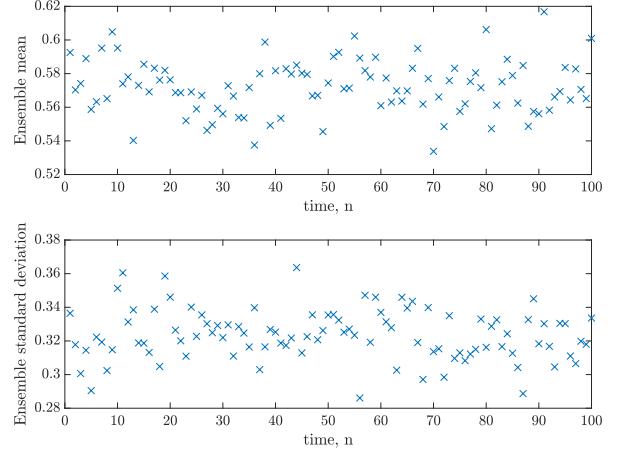


Figure 6: rp2 ensemble statistics over time.

Ensemble average and standard deviation over time for rp3

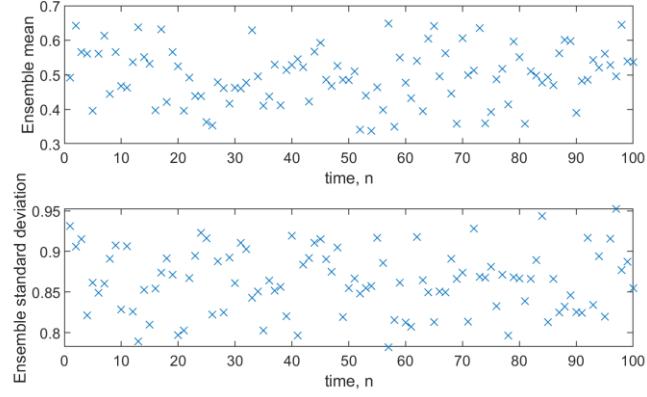


Figure 8: rp3 ensemble statistics over time

1.2.2 Ergodicity

Now, we consider $M = 4$ realisations of length $N = 1000$ for each of the three random processes. The following mean and standard deviations for each realisation were obtained.

We previously established that random process 1 was not stationary, this necessarily implies that this process is not ergodic, since ergodicity is defined only for stationary processes. This can also be seen as the time average is drastically different from the ensemble averages calculated previously. Since $M = 1000$ is 10 times longer than the previous realisation, and the ensemble mean was increasing linearly, it explains why the time average is higher for the longer realisation.

In Table 2, the time averages are significantly different to the ensemble averages, with a large degree of variability, even though the time averaged standard deviation values were low. However, time averages are centred around 0.5, which agrees with the ensemble average. Since the ensemble mean can be approximated well by the time average, the process is ergodic.

Finally in *Table 3*, we observe that time averages of the realisations are equal approximately to 0.5 at 1 s.f. This agrees with the ensemble average values which are approximately centred around 0.5 over time. Since this process seems stationary, we may also hypothesise that this process is ergodic in the mean, since the time average is able to approximate the ensemble average. Generally, for a process to be ergodic in the mean, the sample mean $\widehat{m}_x(N)$ is asymptotically unbiased towards the ensemble mean, m_x :

$$\lim_{N \rightarrow \infty} E(\widehat{m}_x(N)) = m_x$$

Realisation	Time average	Standard Dev.
1	10.0319	5.8650
2	10.0317	5.8774
3	9.9959	5.8953
4	10.0107	5.8576

Table 1: Time average and standard deviation for each realisation of rp1

Realisation	Time average	Standard Dev.
1	0.5378	0.1371
2	0.7956	0.1714
3	0.9768	0.1244
4	0.0741	0.2484

Table 2: Time average and standard deviation for each realisation of rp2

Realisation	Time average	Standard Dev.
1	0.5156	0.8559
2	0.4782	0.8640
3	0.5257	0.8744
4	0.4582	0.8871

Table 3: Time average and standard deviation for each realisations of rp3

1.2.3 Mathematical description

Random process 1 is an amplified sinusoid that has been biased with a term linearly increasing with the sample index, n.

$$RP1 = U(-0.5, 0.5) * 5 * \sin\left(\frac{n\pi}{N}\right) + 0.02n$$

The theoretical mean will be the bias term, $E[RP1] = 0.02n$. This agrees with the ensemble data from Section 1.2.1, with linearly increasing mean. Random process 2 takes the form

$$RP2 = U(-0.5, 0.5) * U(0,1) + U(0,1)$$

with $E[U(-0.5, 0.5)] = 0$, such that $E[RP2] = E[U(0,1)] = 0.5$. This again agrees with the ensemble means seen in Section 1.2.1. Random variables from random process 3 are uniformly distributed between -1 and 2.

$$RP3 \sim U(-1,2),$$

meaning $E[RP3] = 0.5$ and the $Var[RP3] = \frac{(b-a)^2}{12} = \frac{3}{4}$, using similar variance calculations as seen in Section 1.1. This again agrees with the observed ensemble data.

1.3 Estimation of probability distribution

1.3.1 pdf generation file

By utilising the **pdf** function that was written, we are able to obtain an estimate of the pdf of a collection of samples taken from a random process.

The following histogram was obtained after applying the **pdf** function to white gaussian noise or WGN, with a data length of $N=500$.

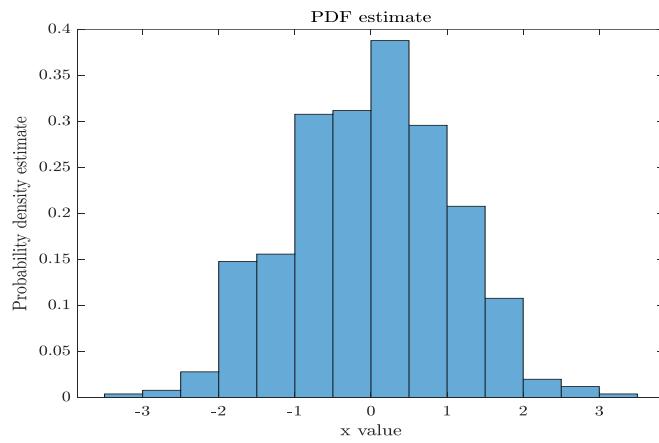


Figure 9: PDF estimation from a collection of $N=500$ samples taken from a Gaussian pdf.

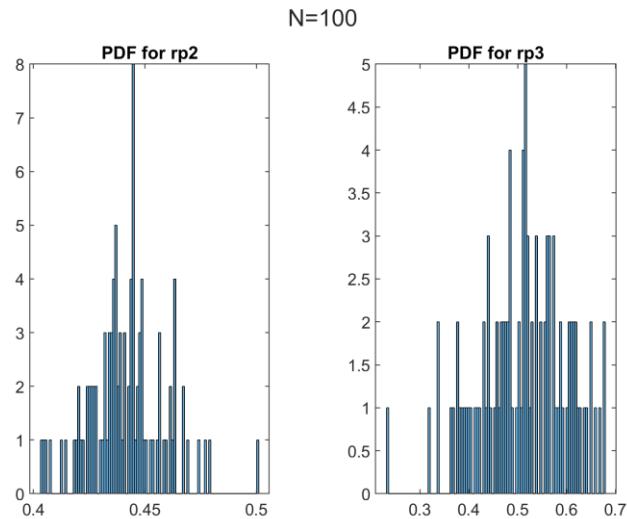


Figure 10: PDF estimate for $N=100$

1.3.2 Using custom function to approximate pdfs for rp2 and rp3

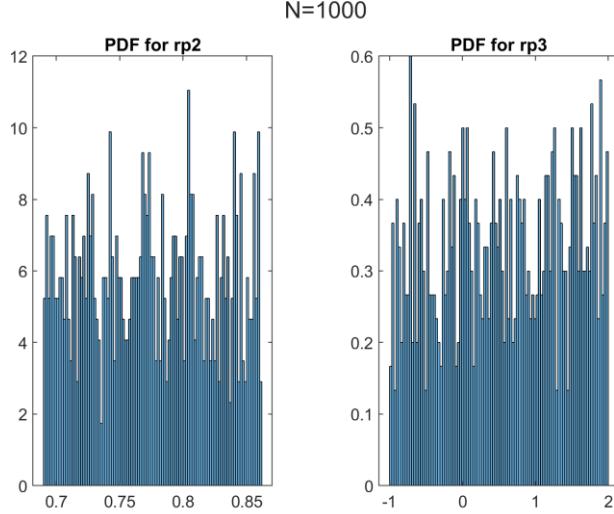


Figure 11: PDF estimates for $N=1000$

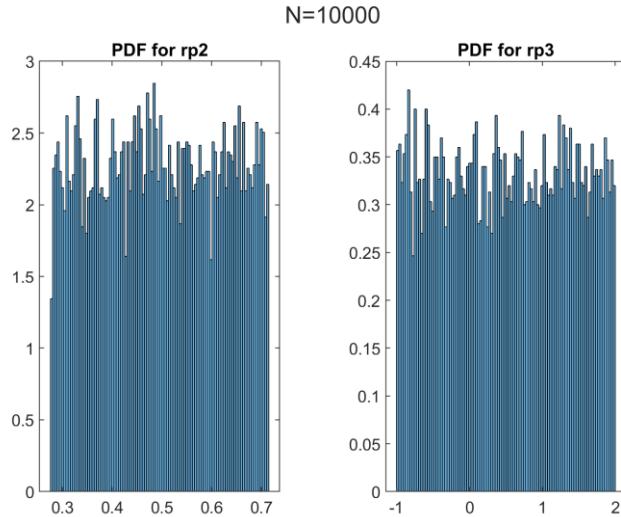


Figure 8: PDF estimates for $N=10000$

As data length N increases, the pdfs tend towards the theoretical predictions for rp1 and rp2, with both distributions approaching a uniform distribution.

1.3.3 Difficulties with non-stationary processes

It is not possible to estimate the pdf of a nonstationary process with this current function. This is because the pdf will change in time, meaning a new pdf will have to be generated to reflect the changes. For the given example, you could tackle this challenge by splitting the signal into 2 sections, from $N = 1$ to 500 before the pdf changes, and then $N = 501$ to 1000. By generating separate pdfs for each section, you'll more accurately model the nonstationary process. A similar approach will be used in Section 3.4 to generate spectrograms.

2 Linear Stochastic Modelling

2.1 Autocorrelation of uncorrelated and correlated sequences

2.1.1 Autocorrelation of White Gaussian Noise

The function `xcorr` was used to plot the autocorrelation function (ACF) for a 1000-sample realisation of WGN with zero mean and unit variance. This function utilises an unbiased estimator given by

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

Theoretically, an independent stochastic process will have an ACF in the form of a discrete Dirac function, since samples at different times will not be correlated with one another. We observe a peak value of 1 at the autocorrelation lag $\tau = 0$ since this value corresponds to its variance for a zero-mean process.

$$R_X(0) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{n=0}^{N-1} x[n]x[n] = E[X^2] = \text{Var}(X)$$

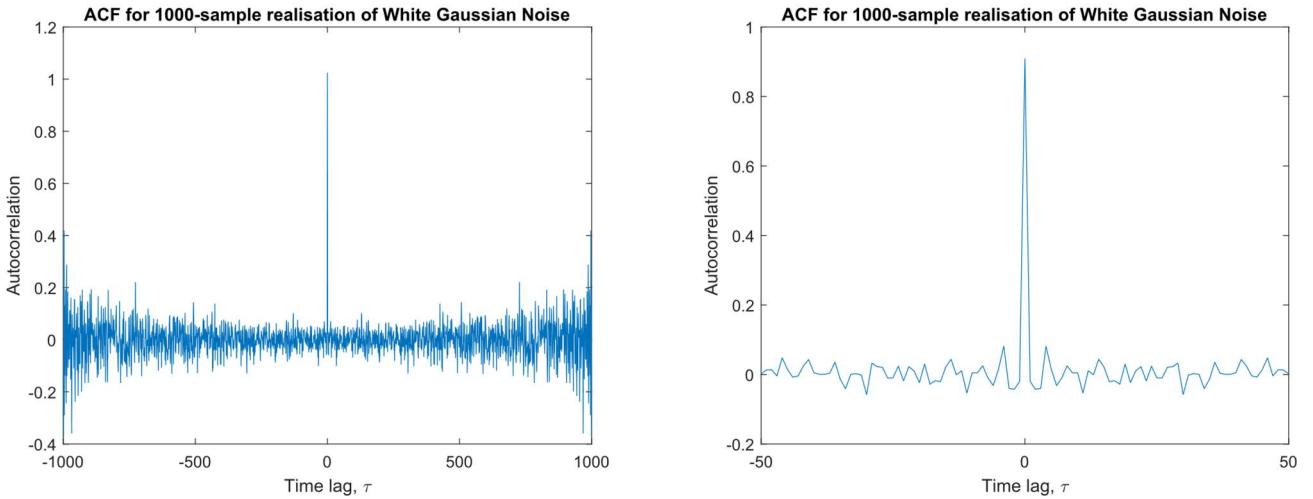


Figure 1: ACF of WGN realisation for large (left) and small (right) autocorrelation lags.

We also observe that the ACF is an even function, since we are considering a stationary process, so the ACF is purely a function of the size of the separation lag. Hence, $\hat{R}_X(\tau) \approx \hat{R}_X(-\tau)$.

2.1.2 Large vs small τ

As autocorrelation lag $|\tau|$ increases, there are less samples to average over, so the variability of the estimate increases. This can be seen in the noise appearing in Figure x (left). For $|\tau| < 50$, the estimate has less variability around the theoretical value of 0, since the estimation is averaged over more samples.

2.1.3 Accuracy of ACF estimates for varying autocorrelation lag

As seen in Eqn (x), as $|\tau|$ increases, $N - |\tau| - 1$ decreases, meaning the summation is over less terms and the average is over less samples. This therefore increases the variance of the estimator, introducing more noise. As a result, ACF estimates for large $|\tau|$ values are not

statistically reliable. By allowing a 10% deviation from the true value of 0, an empirical bound of $\tau \approx 300$ would be suitable.

2.1.4 ACF for MA Process

By filtering a WGN realisation \mathbf{x} through an 8th order Moving Average (MA) filter, we obtain \mathbf{y} , a MA(8) process.

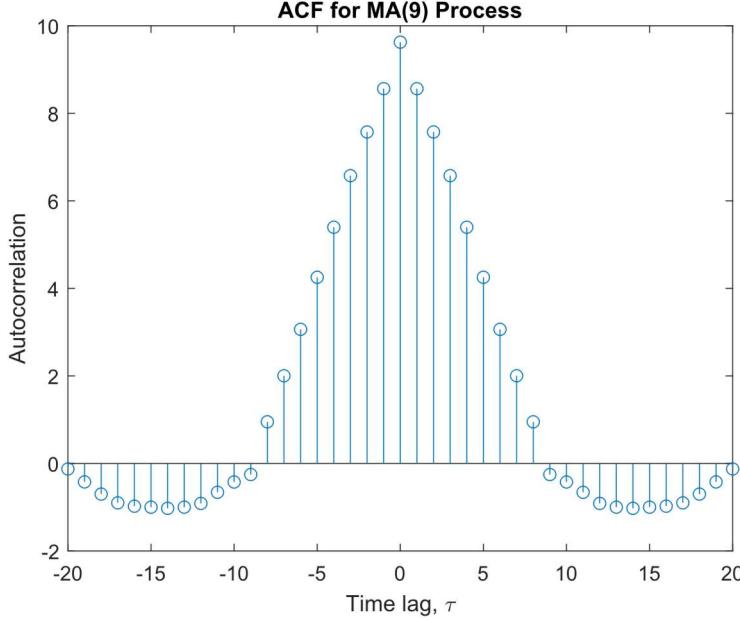


Figure 2: ACF for realisation of MA(8) process

A MA(q) process driven by zero-mean WGN can be shown as:

$$y[n] = \sum_{i=0}^q b_i x[n-i] + w[n], \quad w[n] \sim N(0, \sigma_w^2).$$

We can therefore calculate the theoretical ACF:

$$R_Y(\tau) = E[y[n]y[n+\tau]]$$

$$R_Y(\tau) = \begin{cases} \sum_{i=0}^{q-|\tau|} b_i b_{i+|\tau|}, & \text{if } |\tau| \leq q, \\ 0, & \text{if } |\tau| > q \end{cases}$$

where, in our case, all $b_i = 1$. These coefficients also correspond to the impulse response of the MA filter, therefore making the impulse response a constant delta train of length q . We therefore expect a triangle function with a peak ACF centered at $\tau = 0$, which we observed after producing a realisation of this MA process in the figure above. Notice the cut off in the ACF after autocorrelation lag $|\tau| = q$, which is due to the fact that the moving average signal is only dependent on samples up to q lags previously. This was also seen in the plotted ACF, with only spurious values present due to a finite sample number for the realisation. The resulting signal, \mathbf{y} , is the local sample mean at each time point in the input signal, \mathbf{x} .

2.1.5 Autocorrelation and Linear Systems

Y_n is a filtered version of X_n , meaning that the ACF of Y_n , denoted by $R_Y(\tau)$, is given by

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

where $R_h(\tau)$ is the ACF of the impulse response. Since X_n is uncorrelated,

$$R_X(\tau) = \sigma^2 \delta(\tau)$$

$$R_Y(\tau) = \sigma^2 \delta(\tau) * R_h(\tau)$$

$$R_Y(\tau) = \sigma^2 R_h(\tau)$$

where σ^2 and $\delta(\tau)$ denote the variance of the process X_n and the Kronecker delta function. By utilising the convolution property of $\delta(\tau)$, we see that the ACF of the output process is the ACF of the impulse response scaled by the variance of the input process, in this case the driving WGN. Since the WGN had a variance of unity, $R_Y(\tau)$ should theoretically be equivalent to $R_h(\tau)$.

2.2 Cross-correlation function

2.2.1 CCF for MA filter

We now compute the cross-correlation function (CCF) between the sequences \mathbf{x} and \mathbf{y} generated in Section 2.1, using a similar unbiased estimator.

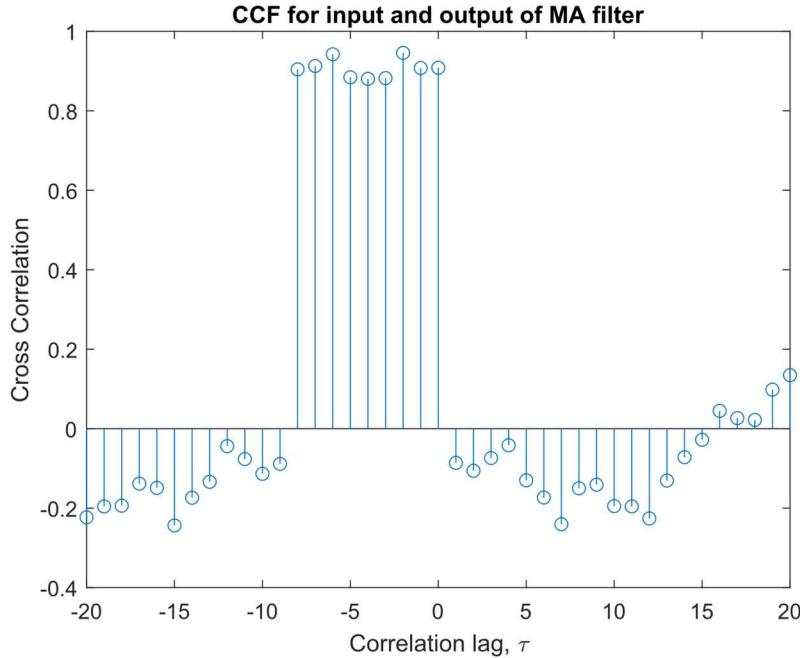


Figure 3: CCF between input and output of moving average filter.

The cross-correlation function, R_{XY} is given by

$$R_{XY}(\tau) = h(\tau) * R_X(\tau),$$

where $h(\tau)$ is the impulse response of the filter. Since X is an uncorrelated stochastic process, it's ACF again assumes the form of a scaled Kronecker delta function, meaning that R_{XY} assumes the form of the impulse response.

$$R_{XY}(\tau) = h(\tau) * \sigma^2 \delta(\tau)$$

$$R_{XY}(\tau) = h(\tau),$$

where the variance of X is again taken as unity. This is exactly what we observe in Figure 3, where the ACF has taken the form of a constant delta train of length $p + 1 = 9$. The value of each delta spike is also approximately 1, agreeing with the impulse response coefficients.

2.2.2 System identification

By estimating R_{XY} from realisations of a stochastic process, we are able to attain an estimate for the impulse response, which contains all the information needed to describe the system. By increasing the number of available samples and realisations, we can attain better estimates of R_{XY} and so accurately identify the system. Higher order MA filters cause longer cross-correlations, since this increases the number of lags until the ACF cuts off.

2.3 Autoregressive Modelling

2.3.1 Stability criteria

An AR process of order p driven by noise $w[n]$ can be described as follows:

$$x[n] = \sum_{i=1}^p a_i x[n-i] + w[n].$$

where a_i denotes the model coefficients. By sampling pairs of autoregressive (AR) coefficients, (a_1, a_2) from uniform distributions $U(-2.5, 2.5)$ and $U(-1.5, 1.5)$ respectively, 100 realisations of AR(2) processes were generated. Pairs (a_1, a_2) that resulted in stable processes were plotted in Figure 4.

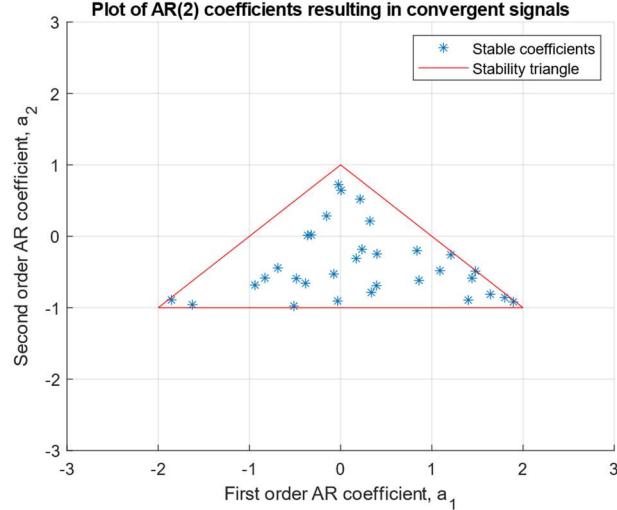


Figure 4: Stability triangle and stable coefficients from simulation.

The pairs of coefficients (a_1, a_2) that result in convergence fall within a triangular region. This triangular shape results from the stability criteria imposed by the equation for the variance of an AR(2) process.

$$\sigma_x^2 = \frac{\sigma_w^2}{1 - \rho_1 a_1 - \rho_2 a_2},$$

where σ_x^2 , σ_w^2 , ρ_i denote the variance of the AR process, the variance of the driving noise and the normalised autocorrelation coefficients respectively. By solving for the Yule-Walker equations for an AR(2) process,

$$\rho_1 = a_1 + a_2 \rho_2,$$

$$\rho_2 = a_1 \rho_1 + a_2,$$

and substituting into the equation for the variance, we attain the following:

$$\sigma_x^2 = \left(\frac{1 - a_2}{1 + a_2} \right) \frac{\sigma_w^2}{(1 - a_2)^2 - a_1^2}.$$

Since variance must be real and positive, we must satisfy $(1 - a_2)^2 > a_1^2$, and $\frac{1-a_2}{1+a_2} > 0$. This leads us to the 3 conditions defining the stability triangle as seen in Figure 4:

Condition 1: $a_1 + a_2 < 1$,

Condition 2: $a_2 - a_1 < 1$,

Condition 3: $-1 < a_2 < 1$.

2.3.2 Sunspot ACF

Real world sunspot time series were then analysed to produce the following ACF plots, with raw data being utilised in Figure 5, and standardised data in Figure 6.

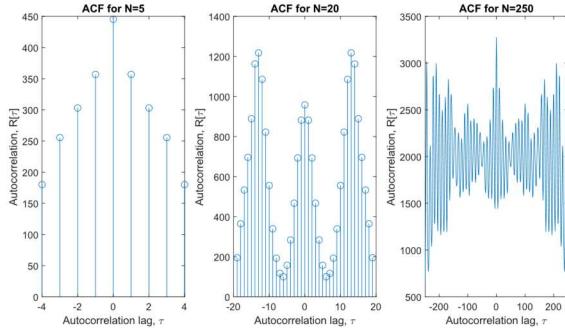


Figure 6 ACFs for varying data length (Raw sunspot data).

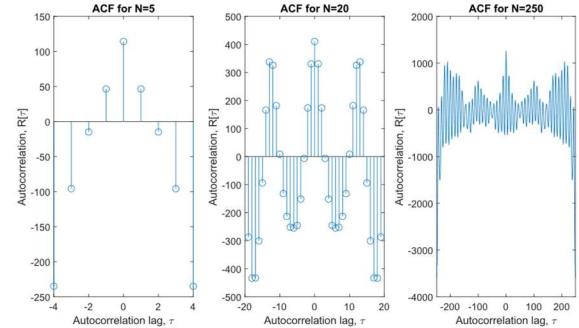


Figure 5: ACFs for varying data length (Standardised data).

Before standardisation, the ACF was corrupted by a DC level due to the mean value of the time series. After removing the mean and scaling to unit variance, the ACF plots display their true pseudo-periodic behaviour. For the standardised data, the maximum autocorrelation value is centred at $\tau = 0$, which is expected. The autocorrelation values then oscillate, with the amplitude of oscillations decaying exponentially with autocorrelation lag, suggesting complex roots for the characteristic equation governing the AR(2) process, corresponding to Region IV in the stability triangle.

2.3.3 Yule-Walker Equations and Partial Correlation Functions

By solving the Yule-Walker equations for model orders $p = 1, 2, \dots, 10$ partial correlation functions were generated and are displayed below in Figure 7. The raw sunspot data had more spurious PACF values above the 95% confidence intervals, whereas the standardised data had only the first 2 orders as significant values. The pseudo-periodicity was again more evident in the standardised data. The 95% confidence intervals correspond to $1.96/\sqrt{N}$, where N is the number of data points.

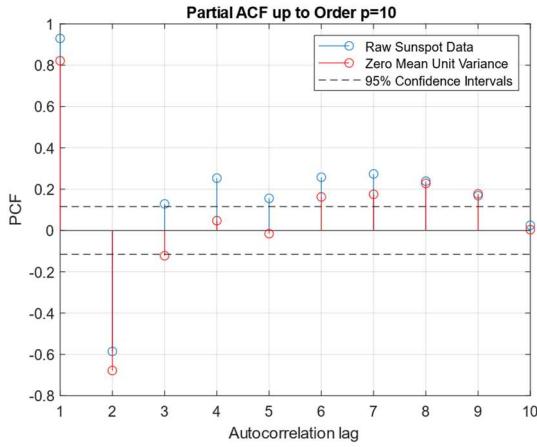


Figure 7: PACF up to order 10.

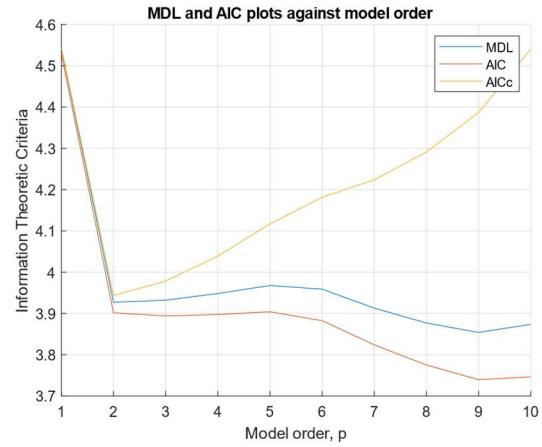


Figure 8: Information theoretic criteria plots.

2.3.4 Information theoretic criterion

Minimum description length (MDL) and Akaike information criteria (AIC) were used to determine the correct model order. By examining the minima and elbows in the plots for these criteria (Figure 8), we observe that the optimal model order is $p = 2$.

2.4 Cramer-Rao Lower Bound

2.4.1 AR(1) model for NASDAQ financial data

By utilising PACF and information theoretic criteria again, we can determine that an AR(1) model is sufficient to model the close prices in the NASDAQ dataset.

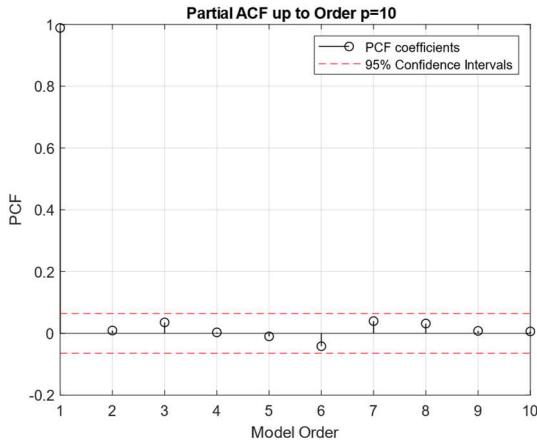


Figure 10: PACF for NASDAQ data



Figure 9: Information theoretic criteria plots.

2.4.2 Cramer Rao Lower Bound

When finding the Cramer Rao Lower Bound (CRLB) for the estimation of the parameters $\boldsymbol{\theta} = [a_1, \dots, a_p, \sigma^2]^T$ of an AR(p), it is difficult to invert its covariance matrix. Instead we can use the asymptotic CRLB based on the power spectrum of the model. The standard log likelihood function can be replaced by

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

and the elements of the Fisher information matrix become

$$[\mathbf{I}(\boldsymbol{\theta})]_{ij} = \frac{N}{2} \int_{-0.5}^{0.5} \frac{\partial \ln[\hat{P}_X(f; \boldsymbol{\theta})]}{\partial \theta_i} \frac{\partial \ln[\hat{P}_X(f; \boldsymbol{\theta})]}{\partial \theta_j} df. \quad (2)$$

By using Equations (1) and (2), and taking partial derivatives of the approximate log likelihood function, we obtain the Fisher information matrix:

$$\begin{aligned} [\mathbf{I}(\boldsymbol{\theta})]_{22} &= \frac{N}{2\hat{\sigma}^4} \\ \mathbf{I}(\boldsymbol{\theta}) &= \begin{bmatrix} \frac{Nr_{xx}(0)}{\sigma^2} & 0 \\ 0 & \frac{N}{2\hat{\sigma}^4} \end{bmatrix}. \end{aligned} \quad (3)$$

2.4.3 Proof of CRLB

The Covariance matrix, $\mathbf{C}_{\boldsymbol{\theta}}$ is the inverse of the Fisher information matrix. Thus by inverting the matrix in Equation (3), and realising that $r_{xx}(0) = \frac{\sigma^2}{1-a_1^2}$ for an AR(1) process, we obtain the CRLBs.

$$\begin{aligned} \text{var}(\hat{\sigma}^2) &\geq \frac{2\sigma^4}{N} \\ \text{var}(\hat{a}_1) &\geq \frac{\sigma^2}{Nr_{xx}[0]} = \frac{1}{N} (1 - a_1^2) \end{aligned}$$

2.4.4 CRLB heatmaps for varying data length and variance of driving noise

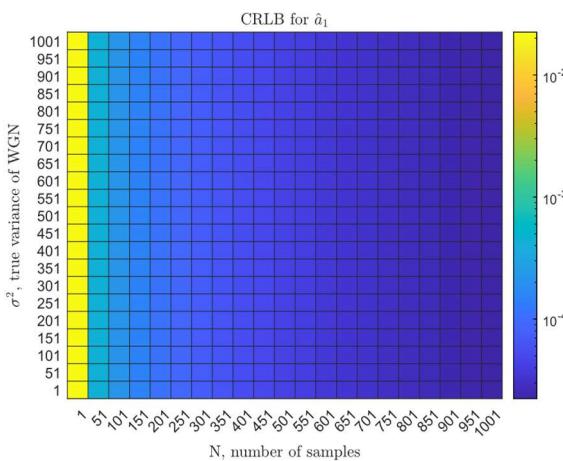


Figure 11: Heatmap depicting CRLB for \hat{a}_1

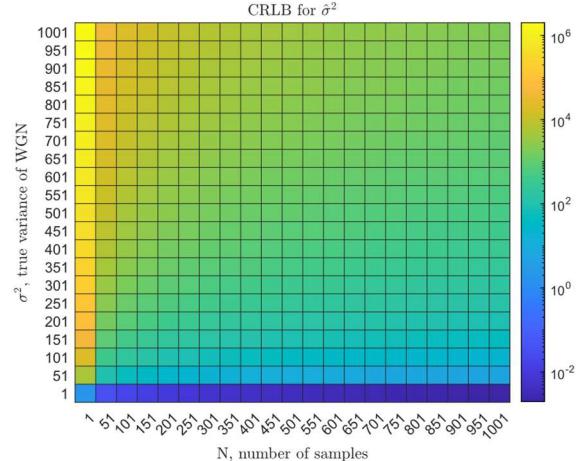


Figure 12: Heatmap depicting CRLB for $\hat{\sigma}^2$

As seen in the figures above, where colour represents the log magnitude of the CRLB, the CRLB decreases for increasing N , and increases with the true variance of the driving noise. From the solutions of the CRLB above, we observe that the CRLB for $\text{var}(\hat{a}_1)$ tends to zero as a_1 tends to unity. This is because a_1 is a pole of the system, so as the pole approaches the unit circle, the system tends towards being more unstable. Therefore less variance is tolerable for a stable estimator.

2.5 Real world signals: ECG from iAmp experiment

2.5.1 Heart rate probability density estimate (PDE)

Below are probability density estimates (PDEs) for original and averaged heart rates, $h[n]$ and $\hat{h}[n]$, at $\alpha = 1$ and $\alpha = 0.6$.

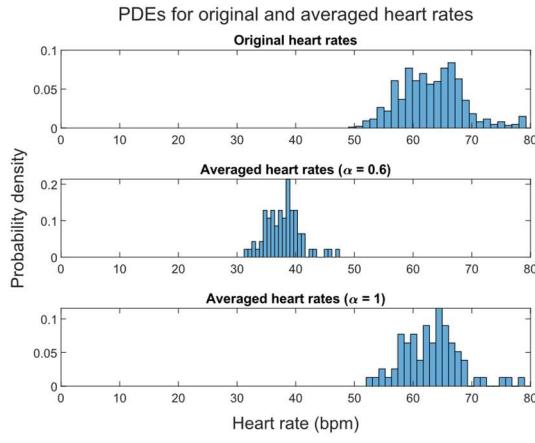


Figure 13: PDEs for heart rates from RRI data.

2.5.2 AR modelling of heart rate

The following ACF plots were obtained for the RRI data in the three trials consisting of unconstrained breathing, constrained breathing at 50 bpm and constrained breathing at 15 bpm.

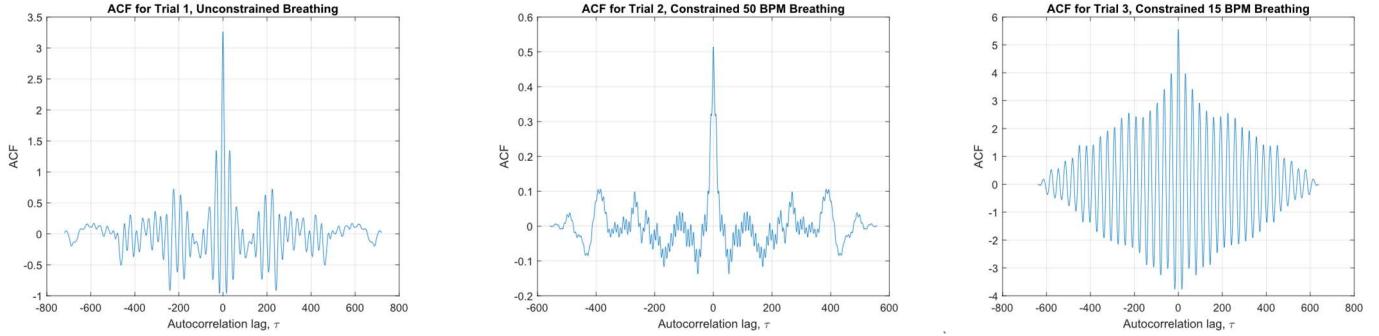


Figure 14: ACF for each trial in iAmp experiment

By noticing that the ACFs exhibit pseudo-periodic behaviour and extend to large autocorrelation lags, we can conclude that the RRI data cannot be MA and must be an AR process.

The PACF was then used to determine the optimal AR model order. Again examining the 95% confidence intervals, we can see that the most likely optimal AR model order is 2 for each trial. There are anomalous spikes in the PACF for Trial 2, suggesting that more filtering needs to be performed before further analysis.

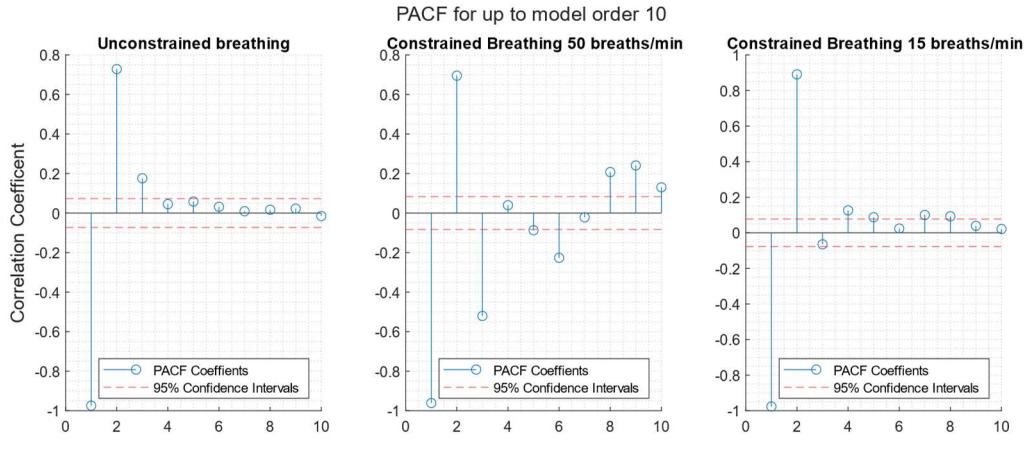


Figure 15: PACF to determine AR model order.

The PDEs have similar shapes with an approximate gaussian distribution. The means however are located at different heart rates, with the value of $\alpha = 0.6$ decreasing the mean, since this estimator descales the input signal, whereas $\alpha = 1$ does not affect the mean heart rate. The estimator with $\alpha = 1$ is therefore unbiased whereas the other is biased.

3 Spectral Estimation

Results of the `pgm` function show that the periodogram estimates are incredibly noisy. White gaussian noise should theoretically have a constant PSD with the value equal to the WGN variance. This is because the PSD is the Fourier Transform of the ACF, so $FT[\sigma^2\delta(\tau)] = \sigma^2$.

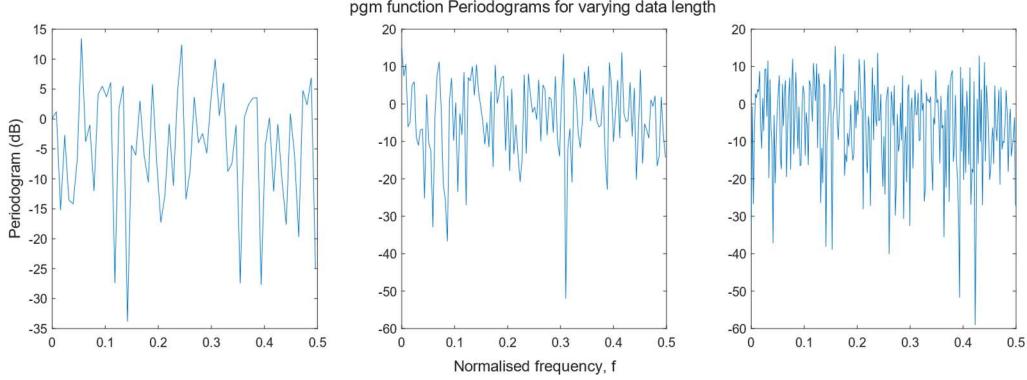


Figure 1: `pgm` periodogram estimates

3.1 Average periodogram estimates

3.1.1 FIR smoothing of periodogram

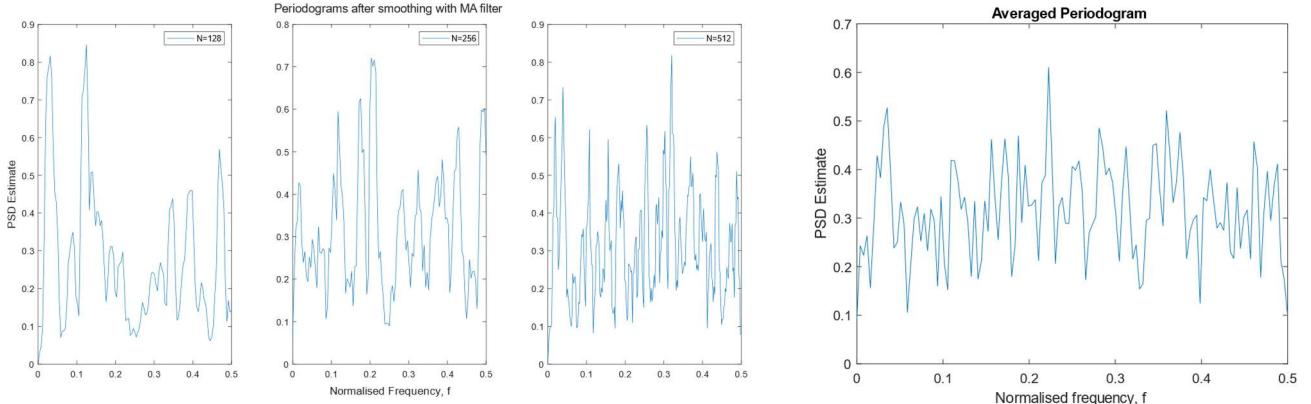


Figure 2: FIR smoothed periodogram (left) and averaged periodogram (right)

Applying a FIR filter does improve the PSD estimate as high frequency noises are filtered, resulting in a smoother PSD. However, resolution in the peak frequencies are decreased.

3.1.2 Averaged periodogram (Bartlett's method)

By subdividing the WGN into 8 non overlapping windows and calculating PSDs for each, we observe that each individual PSD has higher variability than previously, due to the lower sample size.

3.1.3 Averaged periodogram vs individual PSDs

We see in Figure 2 that the averaged periodogram has much lower variability than that of the individual PSDs. With a finite number of samples, we approximate the expectation operator required when calculating the PSD. The individual periodogram is therefore an estimator. The variance of the periodogram is proportional to the square of the PSD value, and constant over the signal length. Therefore by segmenting and averaging over the periodograms, the variance of each estimator is not affected, but resulting estimate is now a mean value. The variance of the averaged periodogram thus is inversely proportional to the number of segments.

3.2 Spectrum of autoregressive processes

A WGN sequence \mathbf{x} , was passed through an AR(1) filter. In Figure 1, we observe clear autocorrelation in the resulting signal, \mathbf{y} .

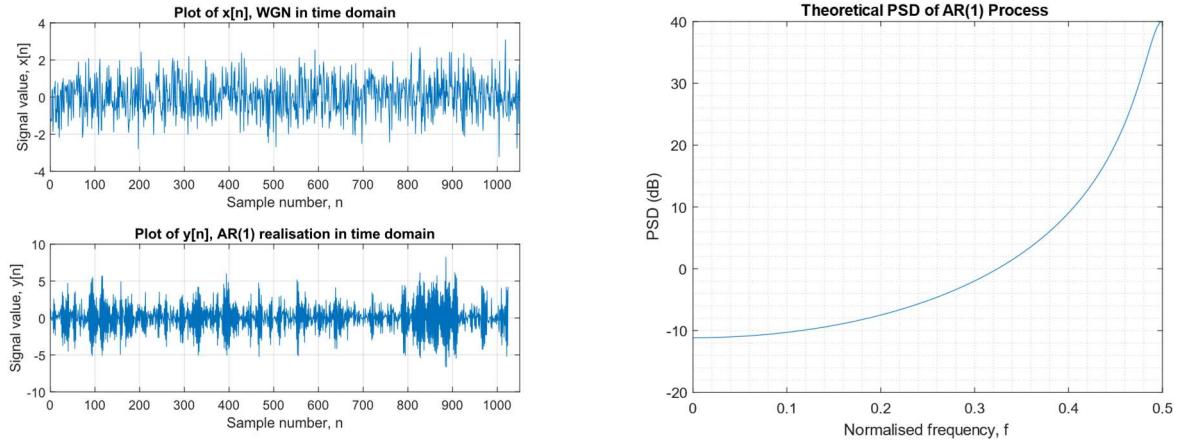


Figure 3: Plot of WGN and AR(1) process realisations (left) and Theoretical PSD (right)

3.2.1 Theoretical PSD

The theoretical PSD for \mathbf{y} is a high pass filter as seen above. This is due to the negative AR coefficient, $a_1 = -0.9$, which means that adjacent samples are negatively correlated, encouraging high frequency oscillations in the signal. The PSD value of -3dB corresponds to the normalised cut off frequency at 0.28 cycle per sample.

3.2.2 Periodogram against theoretical PSD

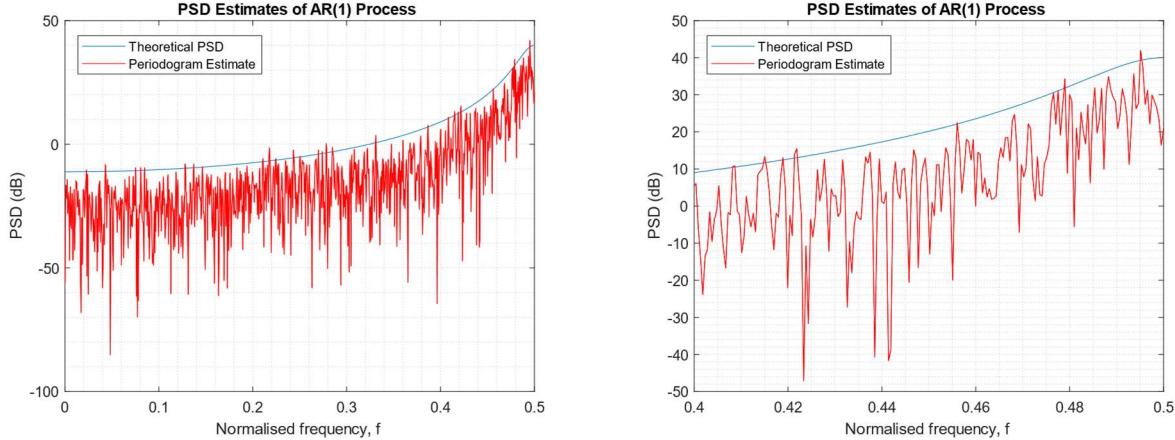


Figure 4: Full (left) theoretical PSD and periodogram and zoomed (right)

3.2.3 Periodogram vs theoretical PSD

We see that the periodogram estimate approximately follows the high pass shape of the theoretical PSD, however there is a DC offset and variability around the true value. Due to the windowing occurring by using a finite length sample, we approximate the expectation operator as mentioned earlier when calculating the PSD. The periodogram is a RV that takes the Chi-Squared distribution, explaining the variability.

3.2.4 Estimating parameters and model-based PSD

To avoid noise in the PSD estimates, we can use a model-based PSD. This can be achieved by estimating auto correlations from measured data, and solving the Yule Walker equations to obtain estimates for the AR model parameters.

$R_Y(0) = 5.6540$; $R_Y(1) = -5.1140$ were the autocorrelation values calculated from data. We then attain the estimates for the AR model parameters.

$$\hat{a}_1 = -\frac{\widehat{R}_Y(1)}{\widehat{R}_Y(0)} = 0.9045$$

$$\widehat{\sigma}_X^2 = \widehat{R}_Y(0) + \hat{a}_1 \widehat{R}_Y(1) = 1.0284$$

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

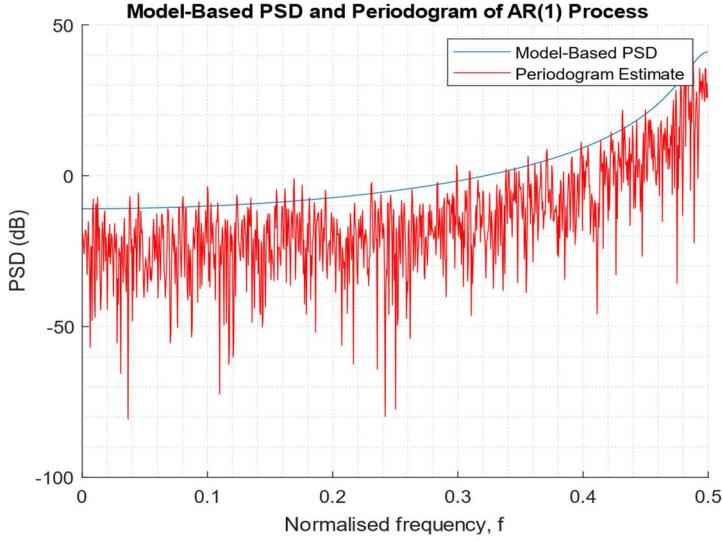


Figure 5: Model-Based PSD for an AR(1) process

3.2.5 Sunspot spectrum modelling

The unstandardised data results in worse fits to the model-based PSDs. Standardising the sunspot series improves these fits. Undermodelling occurs for model order $p=1$, where the model is not complex enough to capture the peak at the normalised frequency of 0.1. Increasing the model order to $p=2$ allows for the agreement of the model-based PSD to the periodogram. Further increasing the model order to $p=10$ results in overmodelling, where the model is overfit to the input data, and captures spurious peaks. Overmodelling may lead to good interpolation error but will fail to generalise when extrapolating outside of seen data.

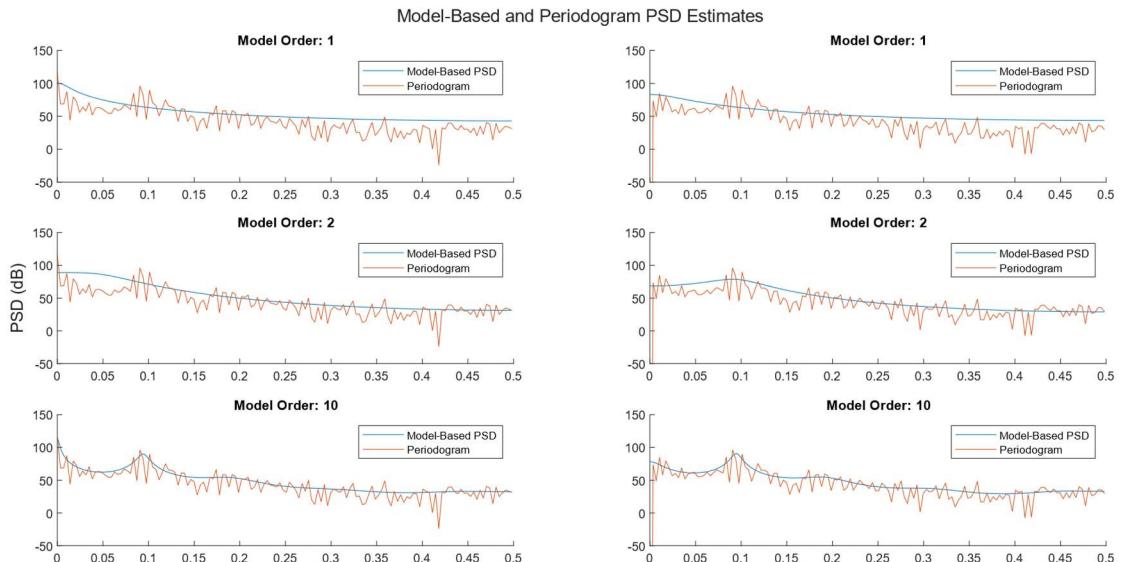


Figure 6: Model-Based PSDs for raw (left) and standardised (right) sunspot series

3.3 The Least Squares Estimation (LSE) of AR Coefficients

3.3.1 Least Squares Estimates for coefficients, \mathbf{a}

$$\begin{aligned}
 J &= \sum_{k=1}^M \epsilon^2(k) = [123, 123, \dots, 123] - \left[\begin{array}{c} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_M \end{array} \right] \\
 \epsilon &= \hat{\mathbf{r}}_{xx}(k) - \sum_{i=1}^p a_i \hat{\mathbf{r}}_{xx}(k-i) \\
 \underline{\epsilon} &= \underline{x} - \underline{H}\underline{\theta} \\
 \left[\begin{array}{c} \epsilon(1) \\ \vdots \\ \epsilon(M) \end{array} \right] &= \left[\begin{array}{c} \hat{\mathbf{r}}_{xx}(1) \\ \vdots \\ \hat{\mathbf{r}}_{xx}(M) \end{array} \right] - \left[\begin{array}{c} r_{xx}(1-p) \\ \vdots \\ r_{xx}(M-p) \end{array} \right] \left[\begin{array}{c} a_1 \\ a_2 \\ \vdots \\ a_p \end{array} \right] \\
 \underline{\epsilon} &= \underline{H}\underline{\theta} = \left[\begin{array}{c} \hat{\mathbf{r}}_{xx}(1) & \hat{\mathbf{r}}_{xx}(2) & \cdots & \hat{\mathbf{r}}_{xx}(p) \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\mathbf{r}}_{xx}(M-1) & \hat{\mathbf{r}}_{xx}(M-2) & \cdots & \hat{\mathbf{r}}_{xx}(M-p) \end{array} \right] \left[\begin{array}{c} a_1 \\ a_2 \\ \vdots \\ a_p \end{array} \right]
 \end{aligned}$$

Figure 7: Demonstration for the form of the Least Squares cost function

The form of least squares estimator solution is similar to form of solution to Yule-Walker equations, where the LSE and Yule Walker parameters are given by,

$$\hat{\mathbf{\Theta}} = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{x},$$

$$\mathbf{a} = \mathbf{R}_{xx}^{-1} \mathbf{r}_{xx},$$

respectively. The parameters are the matrix multiplication between the inverse of square matrices containing autocorrelations, multiplied by a vector of autocorrelations.

3.3.2 Stochasticity of observation matrix, \mathbf{H}

The observation matrix is composed of estimates for autocorrelations, therefore making it stochastic. The observation matrix will not be deterministic values but will instead be an estimate derived from data.

3.3.3 Sunspot series LSE approach

The LSE approach was used to determine the parameters \mathbf{a} for different AR(p) model orders $p=1,2,\dots,10$. The table below displays the estimated AR coefficients.

Coefficient	a1	a2	a3	a4	a5	a6	a7	a8	a9	a10
Order, p										
1	0.8219	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
2	1.3864	-0.6853	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
3	1.3076	-0.5272	-0.1145	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
4	1.3126	-0.4989	-0.1833	0.0515	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
5	1.3135	-0.5020	-0.1918	0.0740	-0.0171	0.0000	0.0000	0.0000	0.0000	0.0000
6	1.3172	-0.5140	-0.1613	0.1574	-0.2342	0.1665	0.0000	0.0000	0.0000	0.0000
7	1.2876	-0.4702	-0.1972	0.1918	-0.1211	-0.1194	0.2123	0.0000	0.0000	0.0000
8	1.2412	-0.4468	-0.1669	0.1492	-0.0798	-0.0101	-0.0841	0.2331	0.0000	0.0000
9	1.1936	-0.4310	-0.1628	0.1660	-0.1127	0.0243	0.0067	-0.0211	0.2085	0.0000
10	1.1883	-0.4298	-0.1654	0.1683	-0.1110	0.0191	0.0112	-0.0105	0.1787	0.0249

Table 1: Estimated AR coefficients for varying model order using the LSE approach.

3.3.4 Model Order Selection

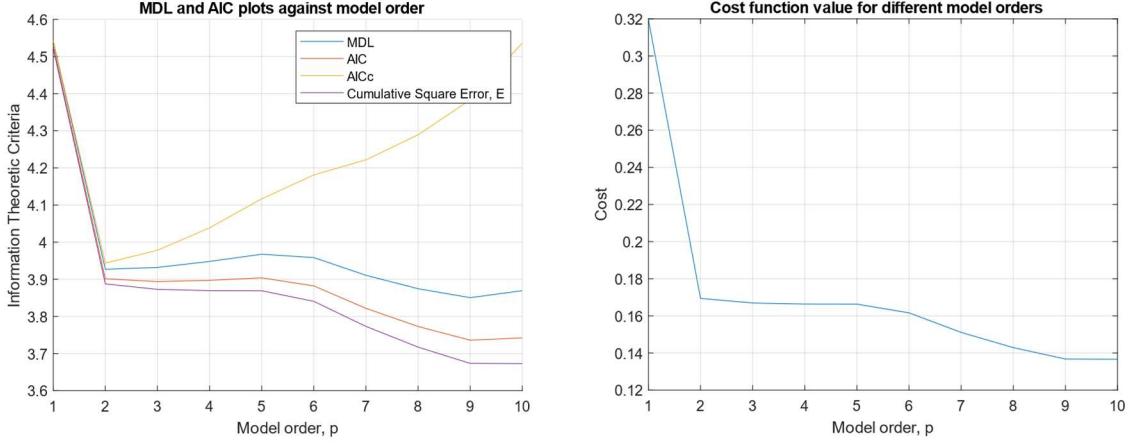


Figure 8

In Figure 7 (left) we see the plots of the information theoretic criteria as seen in Section 2.3. On the left, we also have the approximation error, E over model order. The error continually decreases with p , however at a decreasing rate after $p=2$, forming an elbow. This, together with the minima in information theoretic criteria suggests an optimal model order of 2. On the right we see the cost function varied over model order.

3.3.5 Power Spectra of AR(p) models

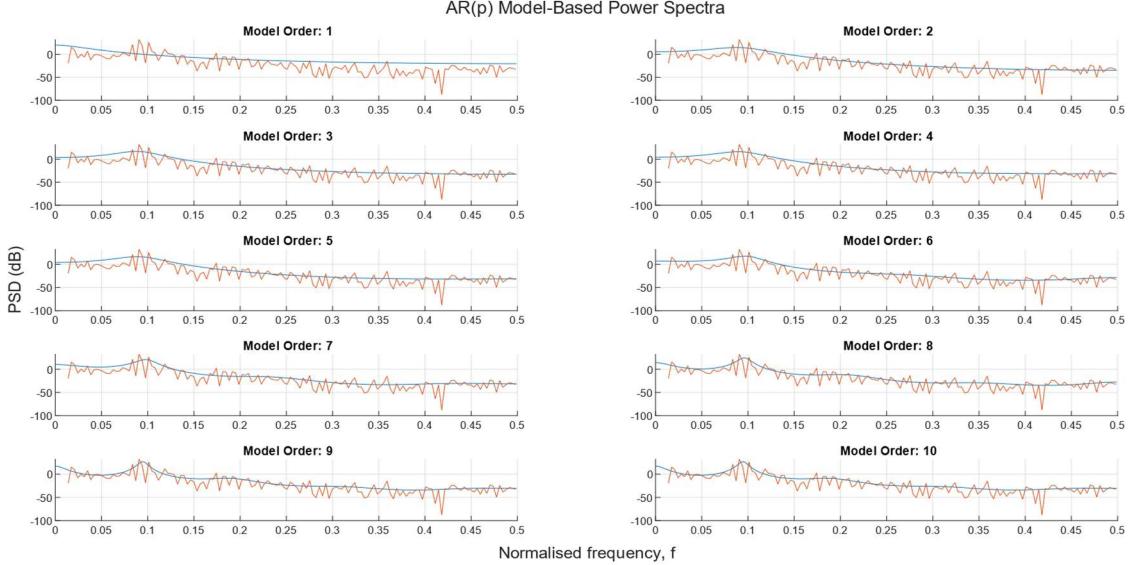


Figure 9: Power spectra for varying model order.

Plotted above are the power spectra associated with varying model orders. As seen previously, the higher model orders overmodel the data, and produce spurious peaks in the power spectra. The model order $p=1$ is insufficient to capture the complexity of the data again, which is to be expected. The order $p=2$ models the data well, and further increases in p do not improve performance significantly, supporting the claim for $p=2$ being the optimal model order.

3.3.6 Approximation Error for Optimal AR(p) model with varying data length

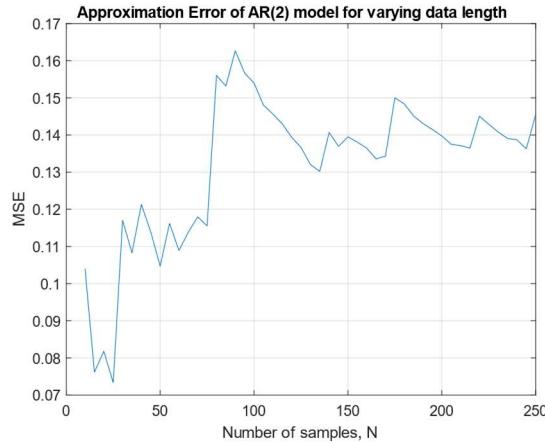


Figure 10: Approximation Error vs data length

Figure 9 shows the approximation error against data length. Initially, more samples decreased the MSE, since the autocorrelation estimates required to calculate the LSE solution are more accurate. Increasing the data length further increases the rows in the observation matrix with large autocorrelation lags. This therefore increases the variance and decreases performance.

3.4 Spectrogram for time-frequency analysis: dial tone pad

3.4.1 Generating a random DTMF signal

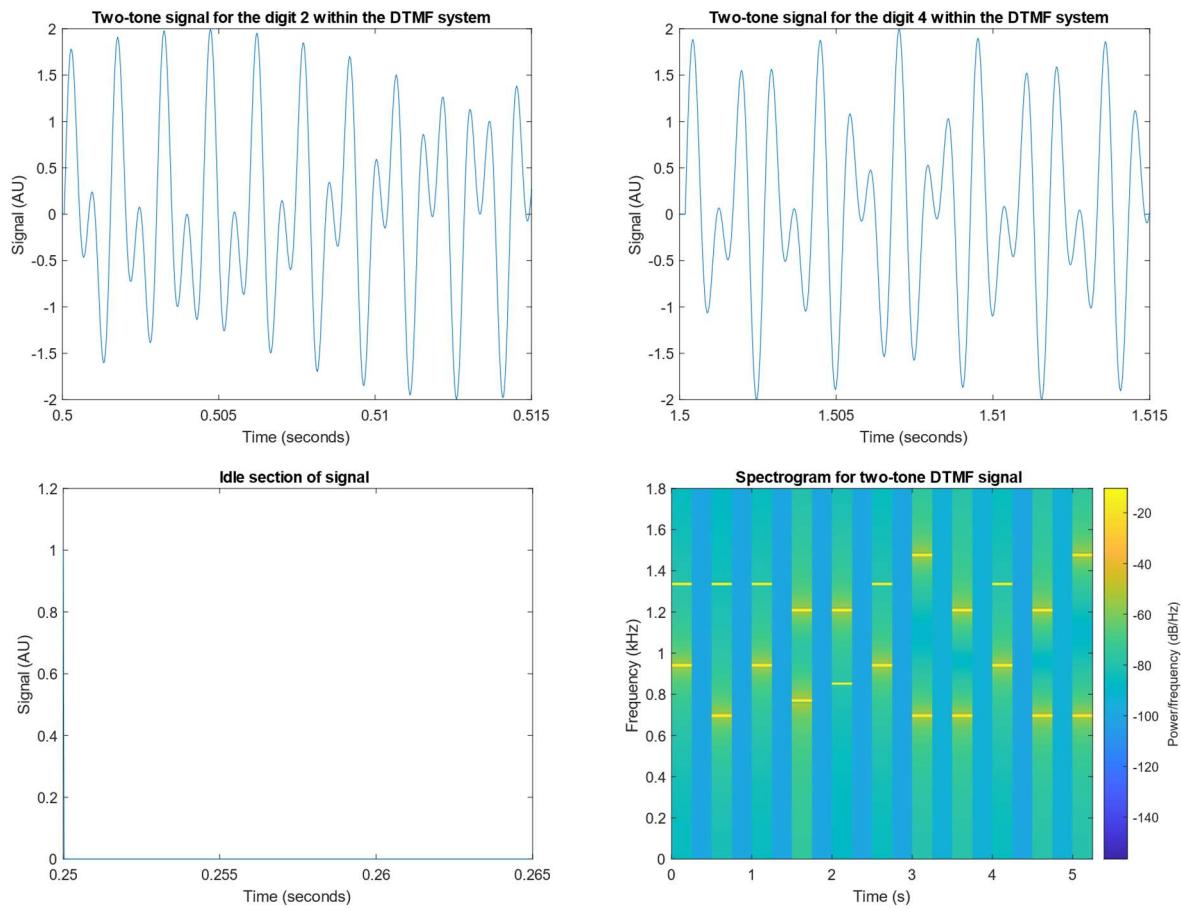


Figure 11: Spectrogram for DTMF signal (bottom right) and Visualisations of digit "2", "4" and idle sections (others)

The sampling frequency of 32678 Hz was appropriate since the bandwidth of the two-tone DTMF signal was 1.5 kHz, meaning that the Nyquist rate was 3 kHz. Sampling above this rate meant that there was no aliasing, and all information was preserved.

3.4.2 Spectrogram estimate

The spectrogram for the two-tone DTMF signal is shown in Figure 10. This was achieved by segmenting the signal into 0.25s windows, corresponding to the windows of time when either

a certain digit was “pressed” or there was an idle section. This ensured that the segments did not overlap and that the frequency content of each digit was separated.

3.4.3 Key classification

As seen in Figure 10, there are clear peaks in the power spectra, corresponding to the dial pad two-tone frequencies. The signal has been separated into 21 sections of 0.25s durations, alternating between key presses and idleness. The bars with no apparent peaks in power spectra correspond to the idle time between key presses.

Due to the clear peak frequencies present it is possible to identify the key pressed by comparing the peak frequencies to the dial pad frequencies table. Due to the Hanning window used, there is a slight decrease in frequency resolution, meaning spreading of the peak intensities, so an error margin of around $\pm 20\text{Hz}$ should be introduced when comparing to the look up table. Also, the finite window length means that the PSDs are not ideal.

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

Table 2: Dial pad frequencies

3.4.4 Corruption by channel noise

In real world scenarios, the signal will be corrupted by noise. Spectrograms were thus obtained for signals having been introduced to varying levels of noise.

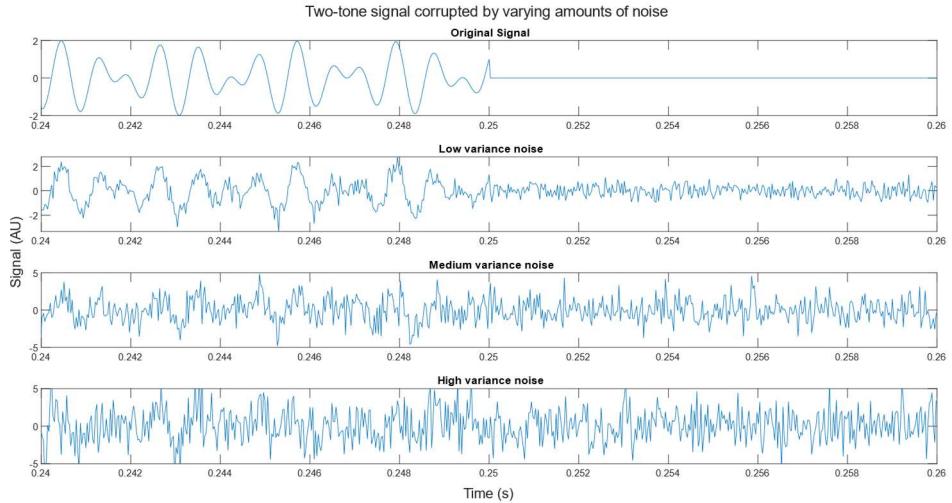


Figure 11: Visualisation of signal with varying levels of noise added.

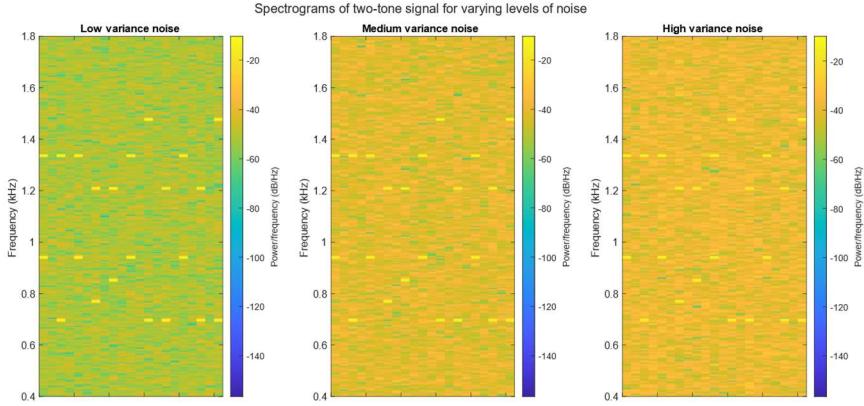


Figure 12: Spectrograms with varying levels of noise added.

As the variance of the noise increases, a larger DC level is present in the spectrogram. Since the PSD is the Fourier Transform of the ACF, and the ACF for white noise is a Kronecker delta function scaled by the noise's variance, the Fourier Transform of a higher amplitude Kronecker delta will result in a higher DC level in the PSD. Since the Fourier Transform is linear, the addition of the noise to the signal means that their Fourier Transforms will add, thus explaining the increased DC levels in the spectrograms. This makes tone identification more difficult, since thresholding the true spectral peaks against the background noise is now harder. Upon visual inspection of the spectrograms, we can see that the spectral peaks are harder to distinguish. For low variance though, peaks are still distinguishable.

3.5 Respiratory sinus arrhythmia from RR-Intervals

Displayed below are the standard and averaged periodograms for each trial in the RRI data from Section 2.5.

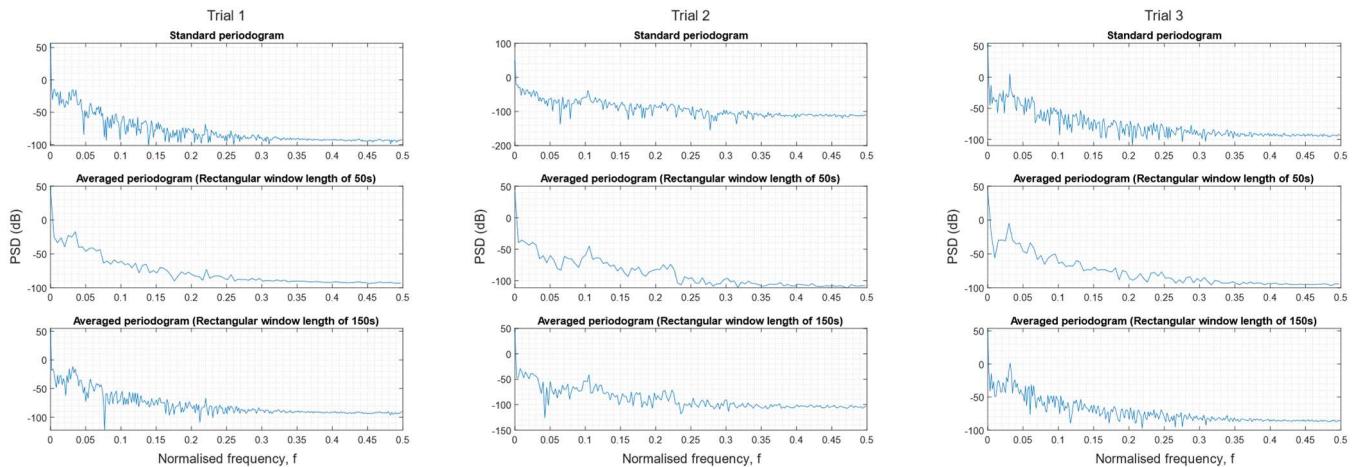


Figure 12: Standard and averaged periodograms for window lengths of 50s (middle) and 150s (bottom).

There are peaks corresponding to different frequencies for each trial. For Trial 1, we see a peak at 0.03 in normalised frequency, which corresponds to 0.12 Hz or 7.2 breaths per minute after scaling by the sampling frequency, 4 Hz. In trial 2, for 50 breaths/min, we would expect a peak at $f = 0.2$, and there does seem to be a secondary peak here corresponding to this breathing rate. The PSD is very noisy however so it is difficult to draw conclusions. For Trial 3 we would expect a peak for 15 breaths/min or $f=0.0625$, and again there is a corresponding secondary peak at this normalised frequency. The other peaks in the spectra may be due to other electrical disturbances recorded by the ECG, such as heart sounds or motor movements.

4 Optimal filtering – fixed and adaptive

4.1 Wiener filter

A 1000-sample WGN sequence \mathbf{x} was filtered to a FIR system with transfer function coefficients corresponding to $\mathbf{b} = [1, 2, 3, 2, 1]$ and $\mathbf{a} = [1]$, to produce an output sequence \mathbf{y} . A noisy signal $z[n]$ was then obtained by adding a WGN sequence with a variance of 0.1 to the filter output $y[n]$. The signal-to-noise ratio for $z[n]$ is then given by:

$$SNR = 20 \log_{10} \frac{\text{Signal Power}}{\text{Noise Power}} = 20 \log_{10} \frac{1}{0.1} = 20dB,$$

Since the power for uncorrelated processes is equal to the process' variance.

4.1.1 Statistics and optimal coefficients of the Wiener filter

By calculating statistics \mathbf{R}_{xx} and \mathbf{p}_{zx} , we are able to calculate the solution for the optimum Wiener filter:

$$\mathbf{w}_{opt} = [1.0481, 1.9376, 2.9651, 1.9020, 0.9700].$$

These correspond to the true coefficients of the system, and their accuracy is shown below.

4.1.2 Differing observation variance

The above section was repeated for varying levels of noise, and model orders.

Variance	w0	w1	w2	w3	w4	SNR (dB)	MSE
0.1000	0.9832	2.1158	3.0864	2.1135	1.0620	20	0.0076
0.2512	1.0482	2.2192	3.1862	2.0764	1.0919	12	0.0199
0.6310	0.8420	2.1479	2.9283	2.1459	0.9952	4	0.0147
1.5849	1.3584	1.9129	2.9432	1.8544	0.9265	-4	0.0332
3.9811	1.2764	2.1504	3.1888	2.1490	1.3317	-12	0.0534
10.0000	1.0768	1.7313	3.8192	2.7395	1.2496	-20	0.2716

Table 1: Optimal Wiener filter coefficients with their SNR and MSE for varying levels of noise.

Order	w0	w1	w2	w3	w4	w5	w6	w7	w8	w9	w10
4	1.0510	2.1057	3.0770	2.1380	1.0300	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
5	1.0874	2.0124	3.1643	2.0153	0.9682	0.0569	0.0000	0.0000	0.0000	0.0000	0.0000
6	1.0952	2.0213	3.0654	2.0412	0.9996	-0.0080	-0.0239	0.0000	0.0000	0.0000	0.0000
10	1.0996	2.0973	3.1207	2.0074	1.0394	0.0487	0.0061	0.0033	-0.0068	0.0090	0.0290

Table 2: Wiener coefficients for assuming Nw greater than 4.

Order	4	5	6	10
MSE	0.007934	0.007206	0.003096	0.007113

Table 3: MSE for different model orders.

As the variance of the added noise increased, the MSE of the parameters increased, which agrees with theoretical expectations. Similarly, the SNR decreased with increasing noise variance, since the power of the noise was increasing. It is evident that increasing model order

beyond 4 did not improve performance, since the higher order coefficients took negligible values and did not contribute to the predictions. The MSE remained constant as model order increased as well, showing that the extra computation for higher orders are not desirable.

4.1.3 Computational complexity of Wiener solution

$$w_{opt} = R_{xx}^{-1} p_{zx}$$

where R_{xx} is a $N_{w+1} \times N_{w+1}$ matrix and p_{zx} is a $(N_w + 1) \times 1$ vector.

The number of operations to find the inverse of an arbitrary $N \times N$ matrix is of the order of magnitude $O(N^3)$. Therefore, to compute the inverse of \mathbf{R}_{xx} , the number of operations is $O((N_w + 1)^3) \approx O(N_w^3)$. The resulting matrix \mathbf{R}_{xx}^{-1} , is also of dimension $(N_w + 1) \times (N_w + 1)$ so the resulting vector will be of dimension $(N_w + 1) \times 1$. To compute each element in Wiener solution vector, there is a dot product from the row of the matrix \mathbf{R}_{xx}^{-1} and the vector \mathbf{p}_{zx} , meaning $(N_w + 1)$ scalar multiplications and N_w scalar additions. As a result, to compute the entire solution vector, you require $O(N_w^3)$ operations + $(N_w + 1) \cdot (N_w + 1)$ scalar multiplications and $(N_w + 1) \cdot (N_w)$ scalar additions.

4.2 The least mean square (LMS) algorithm

4.2.1 LMS algorithm implementation

The following adaptive weight values were obtained after applying the `lms` routine to the process above.

w_0	w_1	w_2	w_3	w_4
0.9782	2.0349	2.9804	2.0734	0.9652

Table 4: Adaptive weight values

4.2.2 Time evolution of estimated coefficients

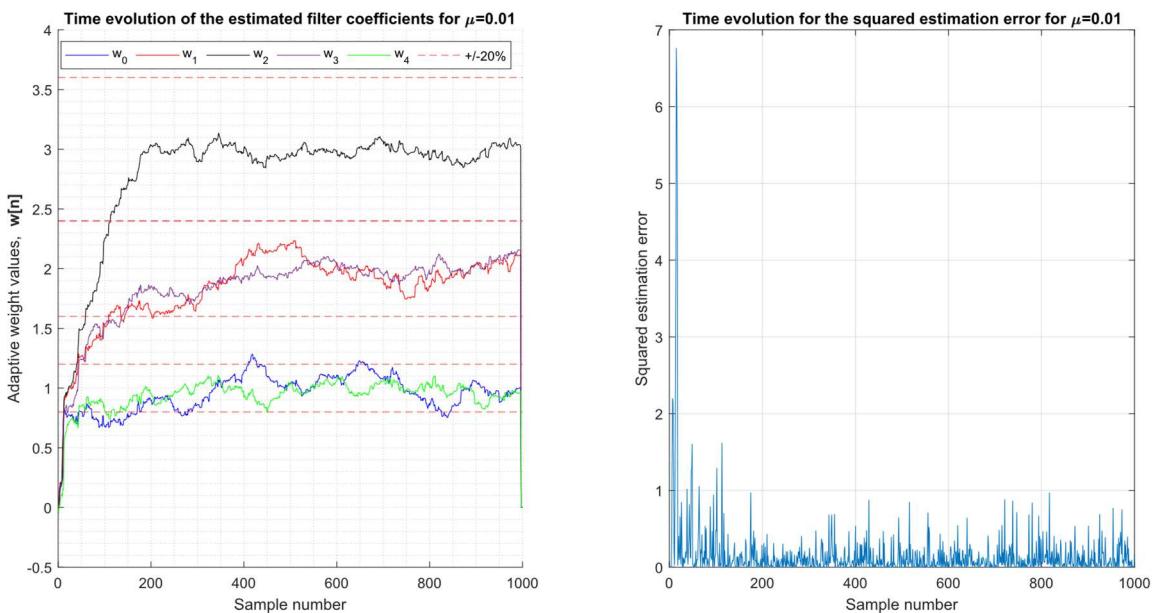


Figure 1: (Left) Time evolution of adaptive weights, with 20% error margins from theoretic values.
(Right) Squared estimation error over time.

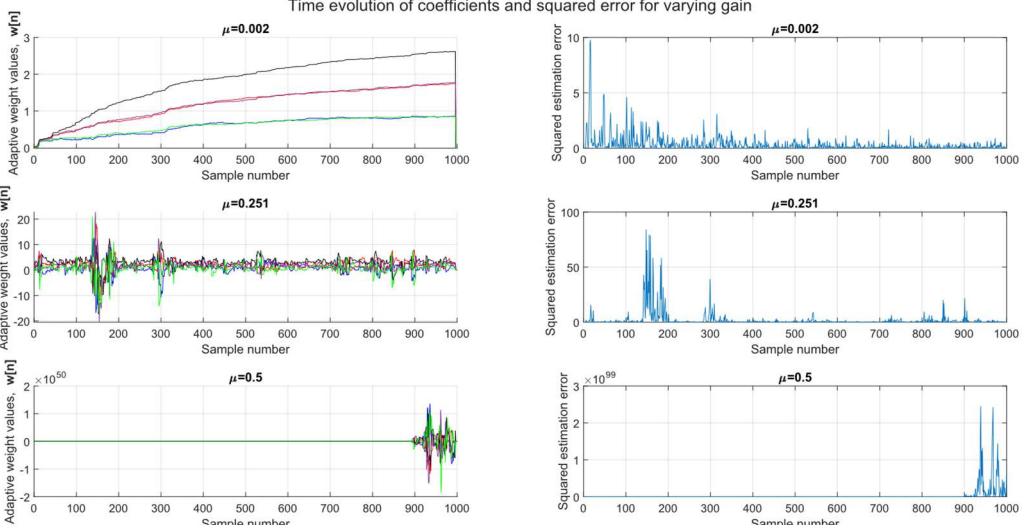


Figure 2: Time evolution of LMS algorithm for varying adaptation gains.

Adaptive weights initialised at 0 begin with large estimation errors and gradually converge to optimal filter coefficients, with squared estimation error decreasing over time. For low values of μ , the coefficients converged slowly but accurately. For an appropriate adaptive gain, $\mu=0.01$, the coefficients converged faster whilst maintaining accuracy. For large μ ($=0.5$), the coefficients did not converge and error diverged.

4.2.3 Computational complexity of LMS algorithm

The LMS algorithm is determined by the following three equations:

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

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

$$e[n] = z[n] - \hat{y}[n]$$

For one time step, the first equation would have operations on the order $O(N_w)$, the second equation would be $O(N_w)$, as well as the last equation. For N time steps we would have a total computational complexity of $O(N_w N)$.

4.3 Gear shifting

By increasing the adaptive gain when error is large and decreasing when the error is small, the coefficient rapidly deviates from poor estimates, and converges to the optimum value once error is small. This can be achieved by setting threshold for the absolute error value of < 0.05 to be considered small, and multiplying gain by 0.9 to reduce it, and setting a threshold of > 0.5 to be considered large, where the gain is multiplied by 1.1. As shown in Figure 3, this policy decreases rise time compared to the static value $\mu=0.01$, whilst still ensuring the learned filter coefficients converge to optimal values.

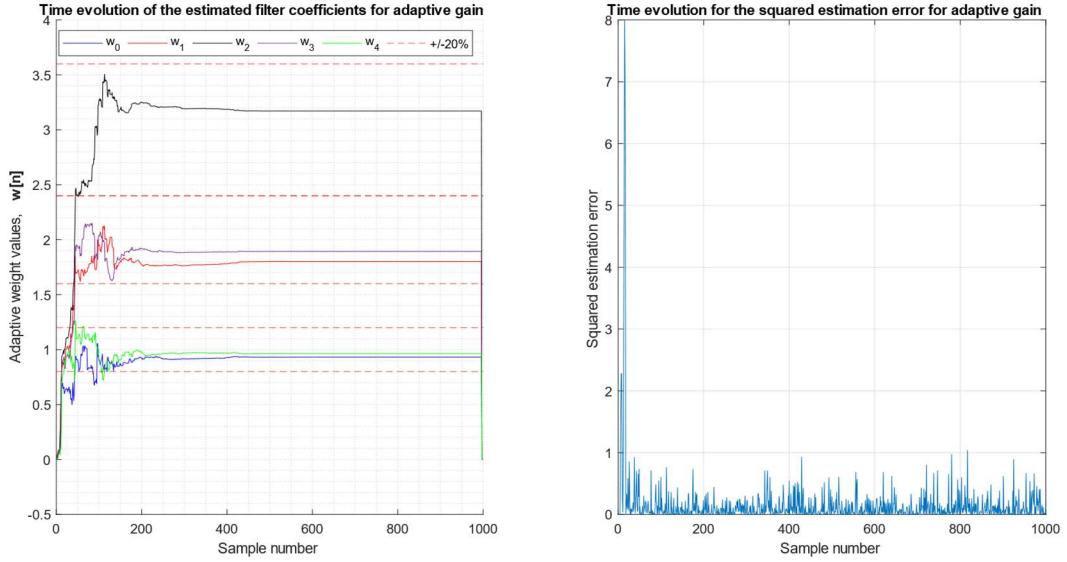


Figure 3: Demonstration of gear shifting to decrease rise time.

4.4 Identification of AR Process

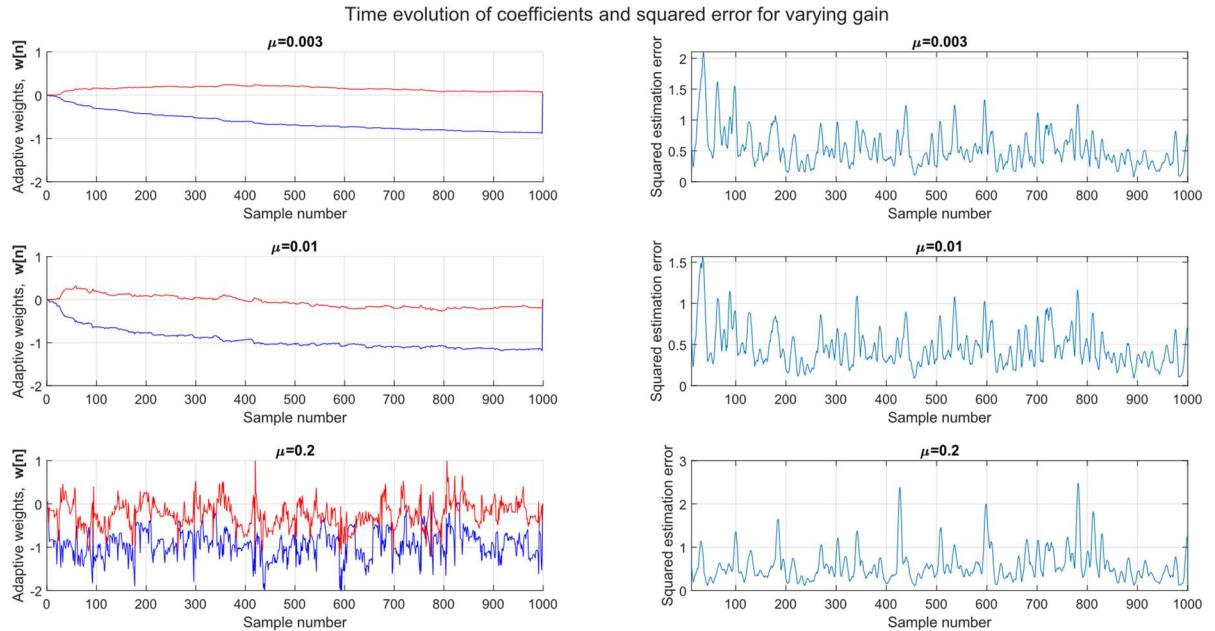


Figure 4: Adaptive LMS for AR process identification for varying gains.

The values of a_1 and a_2 should converge to that of the AR model parameters, -0.2 and -0.9 respectively. This can be seen in the case of $\mu=0.01$, whereas the lower gain value meant convergence was too slow, and the larger lead to diverging coefficients.

4.5 Speech recognition

4.5.1 Performance of p^{th} order predictor

Voice recordings corresponding to the sounds “e”, “a”, “s”, “t”, “x” were sampled at a frequency of $f_s = 44100\text{Hz}$, with a length of $N = 1000$ samples. By providing the adaptive AR filter with a delayed version of the input sound, the estimator served as a predictor. Below are plots displaying the predictions alongside the recorded signals for varying predictor orders.

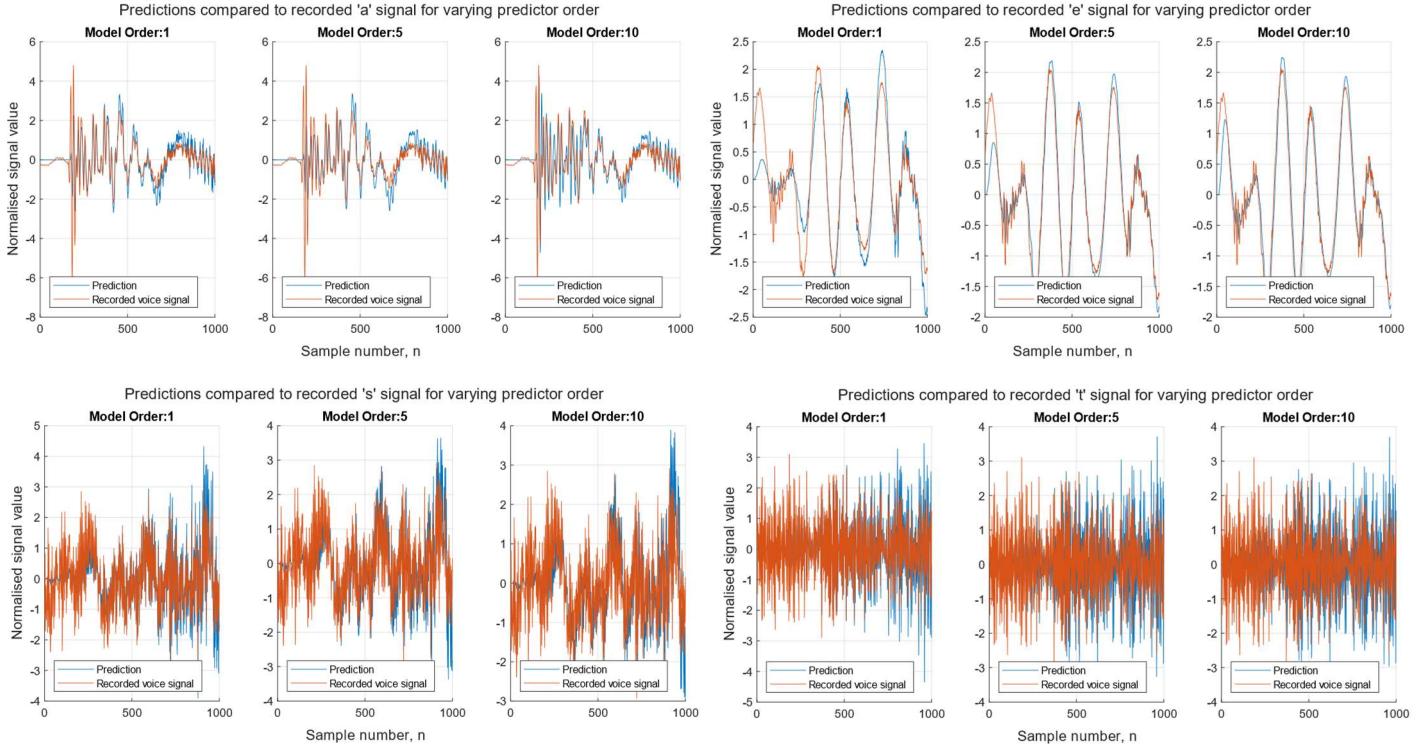


Figure 5: Speech predictions with LMS p^{th} order predictor.

An adaptation gain of 0.1 was used to attain the plots above, after trialling a selection of values ranging from [0.002, 1.5]. Small adaptation gains lead to poor estimates since the non-stationarity of the signals meant that the filter parameters could not be updated quickly enough. Adaptation gains larger than 1 lead to diverging parameter estimates. Similarly, gear shifting was not appropriate since it is designed for use with stationary signals, whereas voice recordings are non-stationary. Increasing the predictor order did not seem to improve performance significantly, suggesting that the computational effort required to compute higher orders is not justified.

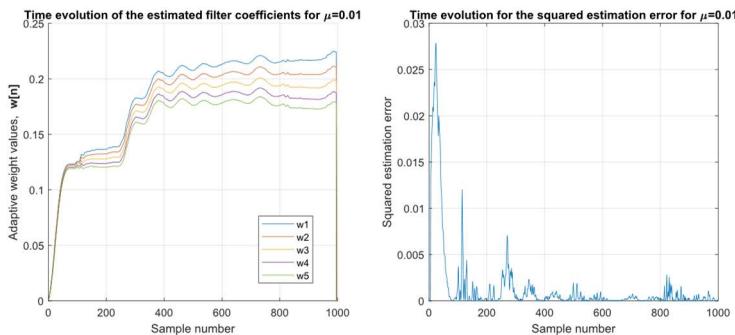


Figure 6: Convergence of adaptive weights for correct adaptation gain

4.5.2 Optimal filter length

A standard measure for the performance of a predictor we can use is the predictor gain

$$R_p = 10 \log_{10} \frac{\sigma_x^2}{\sigma_e^2},$$

where σ_x^2 and σ_e^2 denote the variance of the input and error signals respectively. Thus we can select an optimal predictor order by maximising R_p . By examining the peaks in the plots below, we observe that the optimal predictor orders for the “e”, “a”, “s”, “t”, “x” signals are $p = 7, 1, 5, 2$ and 1 respectively.

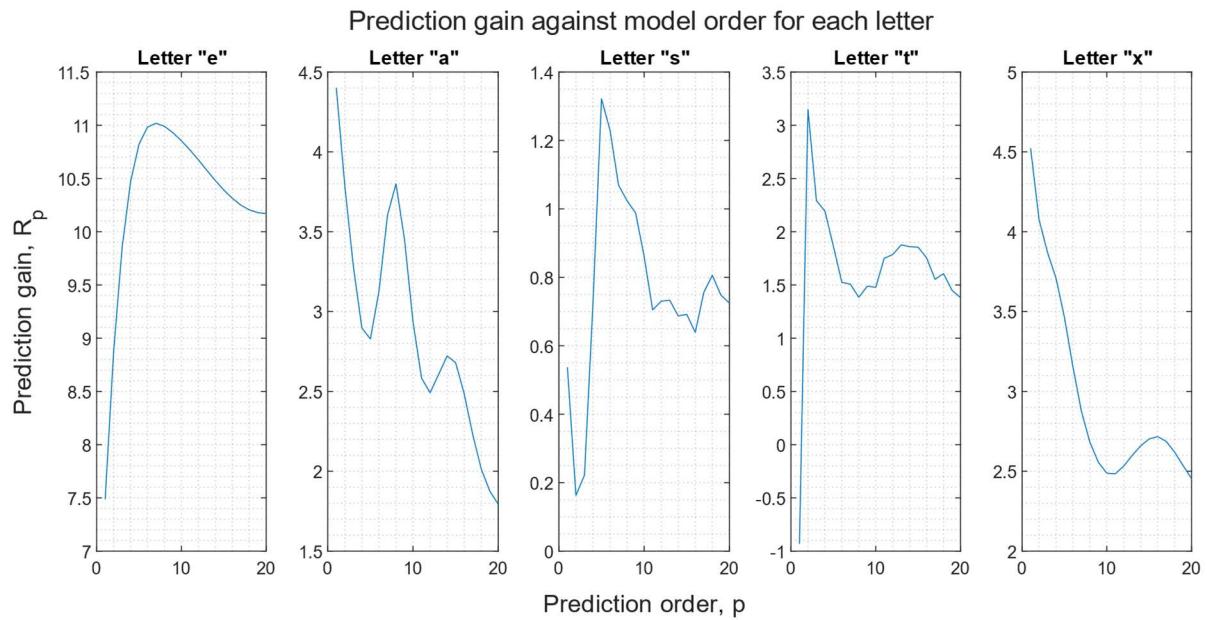


Figure 7: Prediction gains for different prediction orders.

Information theoretic criterion such as MDL or AIC could also be utilised, where the order corresponding to the minima would be considered optimal.

4.5.3 Performance for each recording and effect of sampling frequency

It is worth noting that the prediction gain is significantly higher for “e” than any other letter. This may be explained by the plots in Section 4.5.1, where the “e” signal appears to be quasi-stationary, allowing it to be more effectively modelled by an AR filter. For the other vowel, “a”, there’s also a secondary peak at $p = 8$, similar to that of “e”, suggesting that vowels may be more quasi-stationary than the consonants, and so more appropriately modelled by this filter.

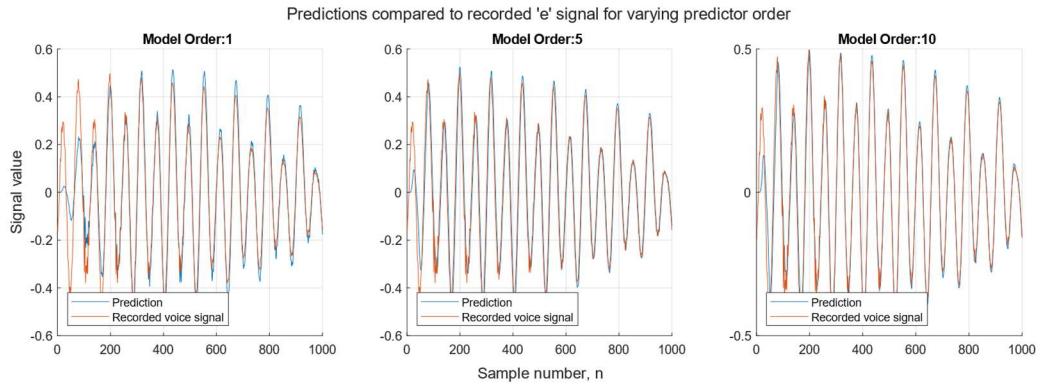


Figure 8: Predictions for "e" signal whilst varying predictor order

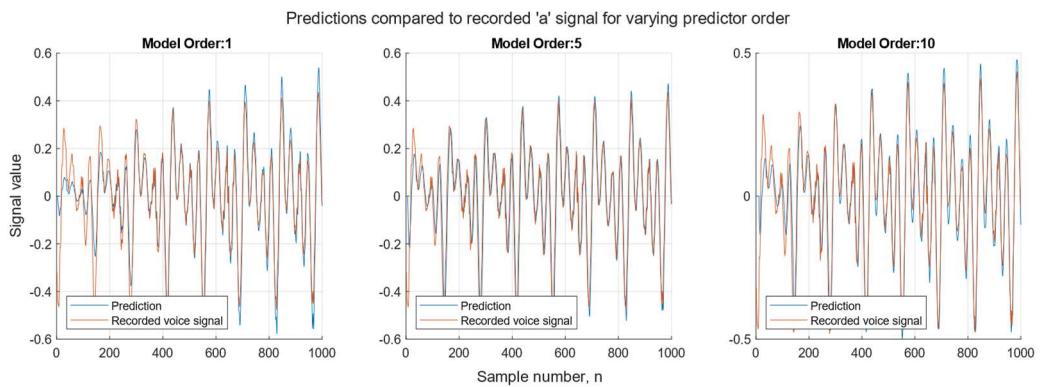


Figure 9: Predictions for "a" signal whilst varying predictor order

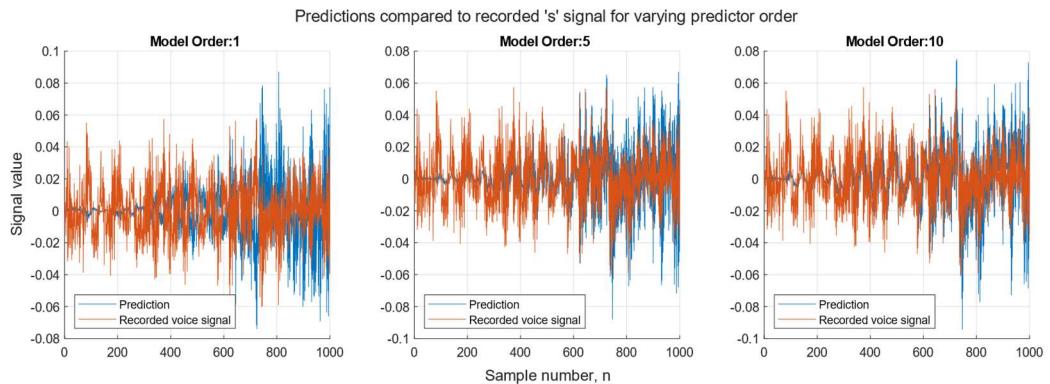


Figure 10: Predictions for "s" signal whilst varying predictor order

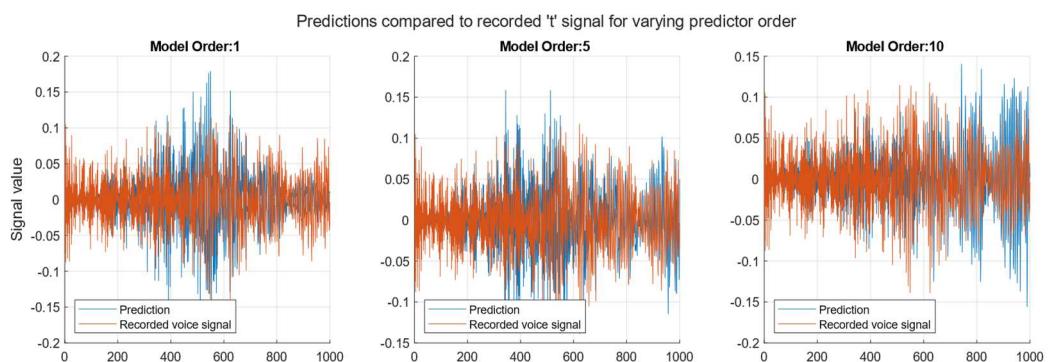


Figure 11: Predictions for "t" signal whilst varying predictor order

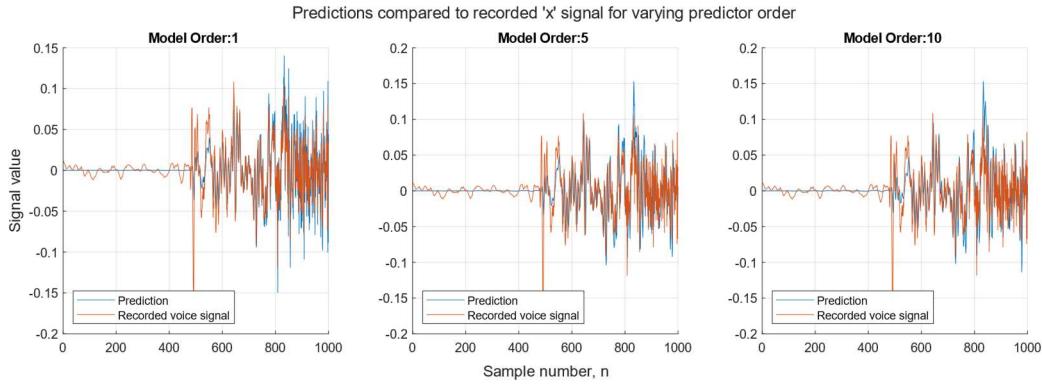


Figure 8: Predictions for "x" signal whilst varying predictor order

When using the lower sampling frequency of $f_s = 16000$ Hz, vowels tended to remain with higher prediction gains, whereas the consonant's prediction gains decreased by 1-4 dB, or approximately halving the signal to error power ratio. This could be explained by a consonant having a wider bandwidth, B , than a vowel, such that the new sampling frequency is less than the Nyquist rate, or $f_s < 2B$. Therefore for a given period of continuous time, Δt , a minimum number of samples, $N = f_{Nyquist}\Delta t = 2Bt$, is needed to avoid information loss and degradation in estimator performance. This also applies to the bandwidth of the vowels, where a minimum number of samples is required to ensure quasi-stationarity and the convergence of learning curves.

4.6 Dealing with computational complexity, sign algorithms

4.6.1 AR parameter identification

To compare the performance of the given LMS algorithms, we first use the AR parameter identification task in Section 4.4 as a benchmark. The following results were obtained.

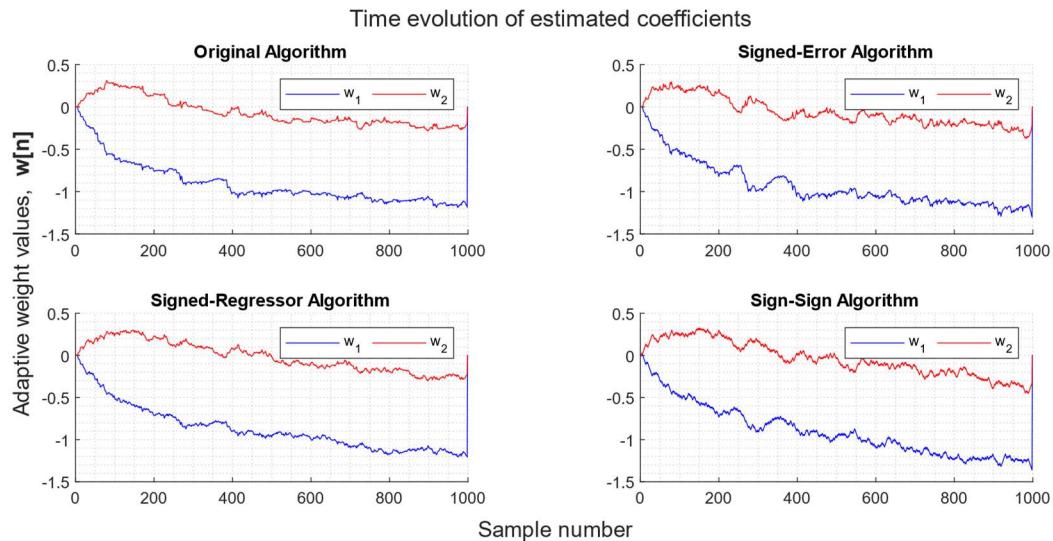


Figure 9: Sign LMS algorithms for AR parameter identification

For this task, the signed-regressor and signed-error algorithms performed well, with convergence to the same weight values as those learned by the standard LMS algorithm. The sign-sign algorithm had worse performance, with deviation from the ideal values of -0.9 and -0.2 of approximately 30%.

4.6.2 Speech recognition task

Another benchmark for the performance of sign LMS algorithms is the speech recognition task from Section 4.5. The time evolution of the adaptive weights is again shown to display the convergence properties.

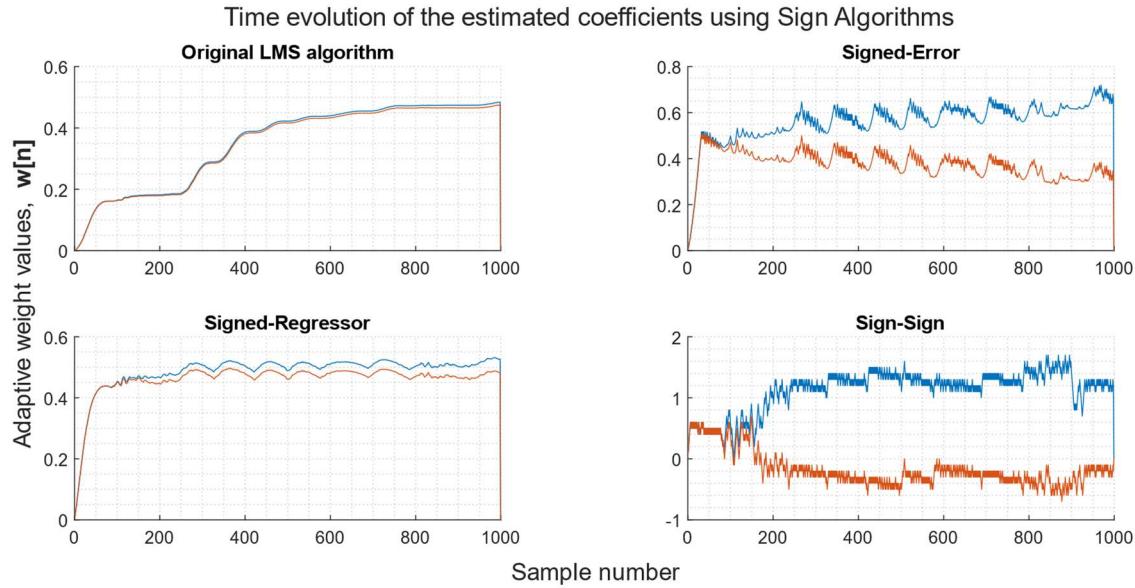


Figure 10: Adaptive weight trajectories over time (raw).

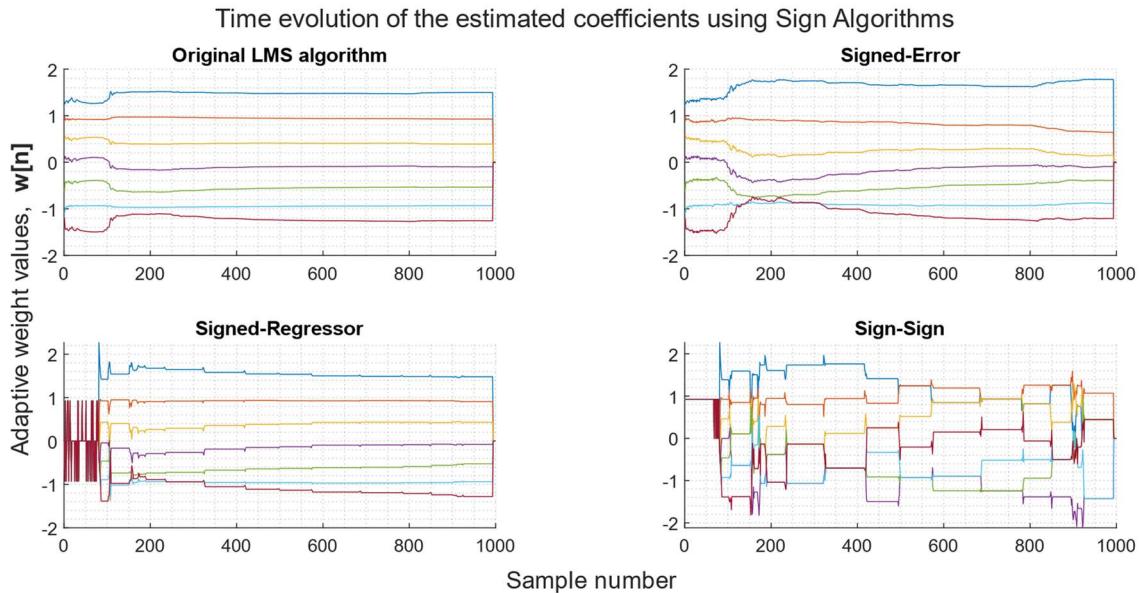


Figure 11: Adaptive weight trajectories over time (standardised for comparison)

Figure 13 shows that the sign LMS algorithms had much shorter rise times, with rapid initial increase. We can see that the signed-regressor algorithm most closely follows the trajectory of the standard LMS algorithm, with the signed-error algorithm diverging towards the end. Again the sign-sign algorithm again diverged, since not enough information was carried without the magnitude of the regressor or error. Figure 14 shows that when normalised, the weight values between the standard, signed-error and signed-regressor agreed.

R_p	Original	Signed-Error	Signed-Regressor	SS
p=2	7.8984	16.1148	14.9329	16.1658
p=7	10.9670	15.6348	15.0593	7.4938

Table 5: Prediction gain for varying model orders and LMS algorithms.

In the table above, we observe that the Signed-Error and Signed Regressor algorithms have competitive performance against the original LMS algorithm, with converging coefficients but lower rise times. The predictor gains were actually higher for these two signed algorithms, suggesting that the original estimator was not converging in time, and the faster rise times allowed for convergence to optimum coefficients. The Signed-Signed algorithm did not lead to convergence, since too much information was lost once removing the magnitude of the error. Given a longer data length, the SS algorithm could converge.