

# EE3-08 Advanced Signal Processing Report

by Agni Iyer

March 29, 2018

# 1 Random Signals and Stochastic Processes

## 1.1 Statistical Estimation

We generate a vector  $\mathbf{x}$  containing 1000 samples from a Uniform Distribution  $U(0, 1)$ , and observe its plot shown in Figure 1.

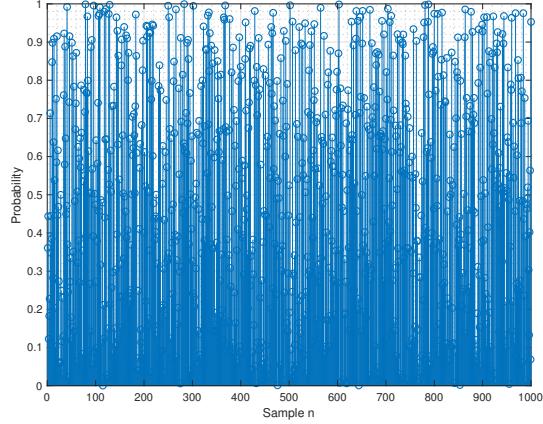


Figure 1: 1000 samples of a Uniform Distribution  $U(0, 1)$

### 1.1.1 Sample and Theoretical Mean

We know that the *theoretical mean* is 0.5 since we are given a uniform probability distribution between 0 and 1. We calculate the *sample mean* using  $\mathbf{x}$ . Since the accuracy of the estimator increases with more samples, there will be a certain error in the mean calculated by this method.

Sample Mean	Error
0.49683	0.0031684
0.49296	0.0070372
0.49018	0.0098166
0.50179	-0.001792
0.50605	-0.0060459

Thus we see that the sample mean matches the theoretical mean up to 2 decimal places.

### 1.1.2 Sample and Theoretical Standard Deviation

The *theoretical standard deviation* converges to 0.28868. Multiple iterations give the following values for *sample standard deviation*:

Sample Standard Deviation	Error
0.29143	-0.0027575
0.2938	-0.005121
0.28912	-0.00044826
0.28221	0.0064643
0.28569	0.002981

The sample standard deviation shows the same level of accuracy as the mean, with theoretical and sample values being identical up to 2 decimal places.

### 1.1.3 Bias Estimation

The estimator bias for the mean and standard deviation for 10 realizations of  $\mathbf{x}$ , each having 1000 samples, are shown in Figure 2.

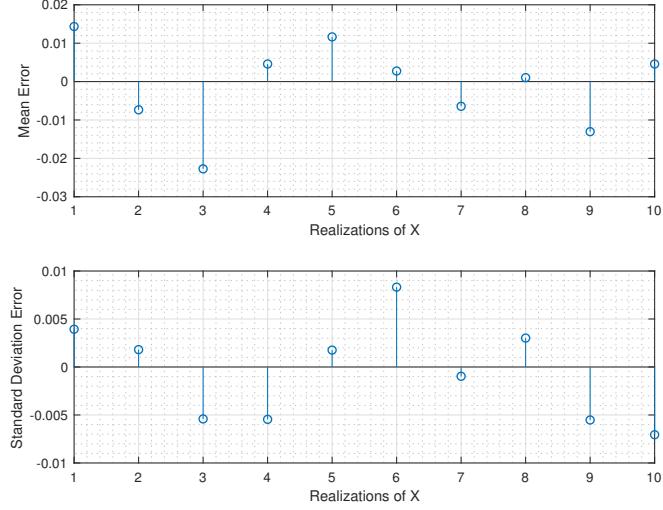


Figure 2: Bias of mean and standard deviation. Bias given by  $B = E[X] - m$

### 1.1.4 Estimating the Probability Density Function

Since the histogram in Figure 3 has 10 bins, each bin's probability must tend to 0.1 since the total theoretical probability is 1. The accuracy of the estimation increases as you increase the number of samples, i.e., the probability of each bin becomes closer to 0.1.

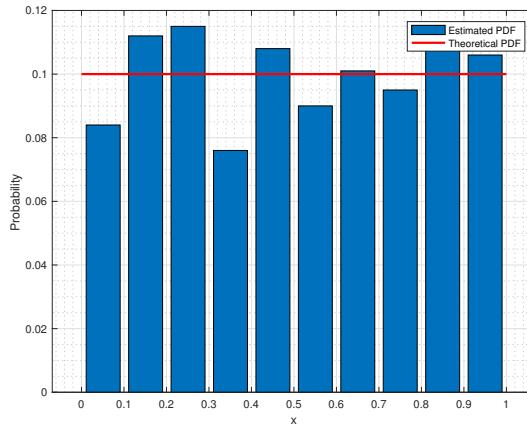


Figure 3: Estimated PDF for 1000 samples from  $U(0, 1)$

### 1.1.5 Using a Gaussian Random Variable

We now use Gaussian random variables instead of Uniform ones, and repeat our analysis.

The *theoretical mean* of the new Gaussian distribution is 0 and the *theoretical standard deviation* is 1. The estimated values of the mean and standard deviation lie close to their theoretical values, and we can observe the bias for 10 iterations of the random variable in Figure 4.

An estimate of the probability density function can be derived using the same program as before. The output for 10,000 samples is shown in Figure 5.

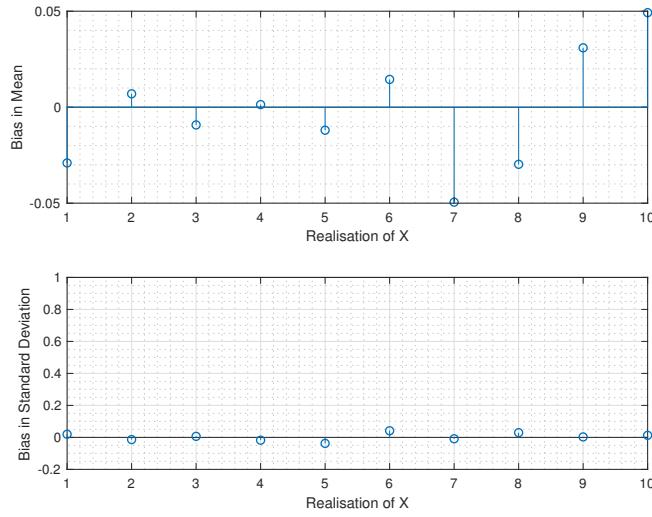


Figure 4: Bias in mean and standard deviation for 10 Gaussian random variables

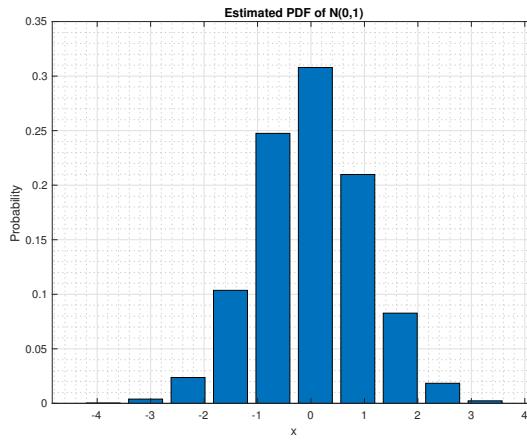


Figure 5: Estimated PDF for 10,000 samples of  $N(0, 1)$

## 1.2 Stochastic Processes

- A *stochastic* or random process is defined as a collection of random variables.
- A *stationary* process is a random process whose statistical properties - such as its mean, moments and variance - remain constant over time.
- An *ergodic* process is one in which the *time average equals the ensemble average*. This means that you get the same result if you measure statistical properties of one ensemble member over the entire time period or of all ensemble members at one time instant.

We are given 3 random processes to analyze - *rp1*, *rp2* and *rp3*. Each of the 3 Matlab functions that generate these processes outputs an  $M$  by  $N$  matrix, where  $M$  is the number of ensemble members and  $N$  is the number of samples in each ensemble. Using the *plot* function graphs the data points in a column against the column number.

### 1.2.1 Ensemble Mean and Standard Deviation

Figure 6 shows that  $rp1$  is not stationary since both its mean and standard deviation vary over time. The mean for the given signal is linearly increasing and can be approximated by the equation  $mean = \frac{t\pi}{50}$ , where  $t$  is the sample number. The standard deviation can be described by the equation

$$SD = 1.44 \sin\left(\frac{t\pi}{T}\right)$$

where  $t$  is as before and  $T$  is the total number of samples.

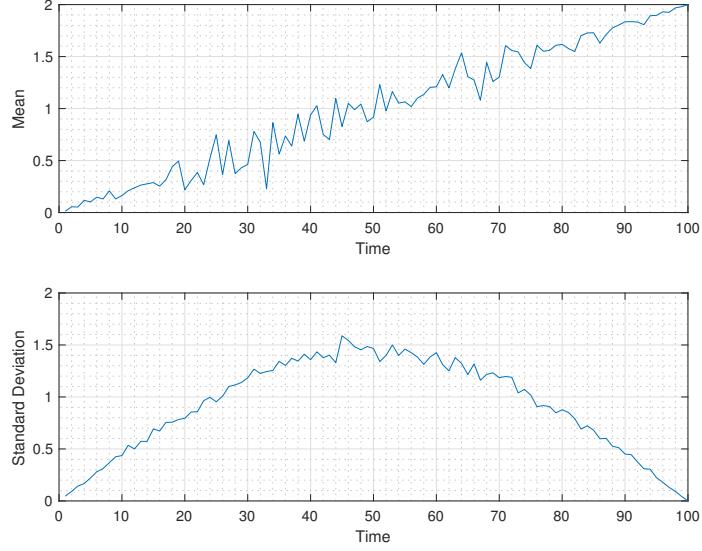


Figure 6: Mean and standard deviation of  $rp1$  with respect to time

Figure 7 and Figure 8 show that  $rp2$  and  $rp3$  respectively are both stationary.  $rp2$  has a constant mean of 0.53 and standard deviation of 0.34.  $rp3$  has a constant mean of 0.5 and 0.87. These can be checked by further increasing  $M$  and  $N$ .

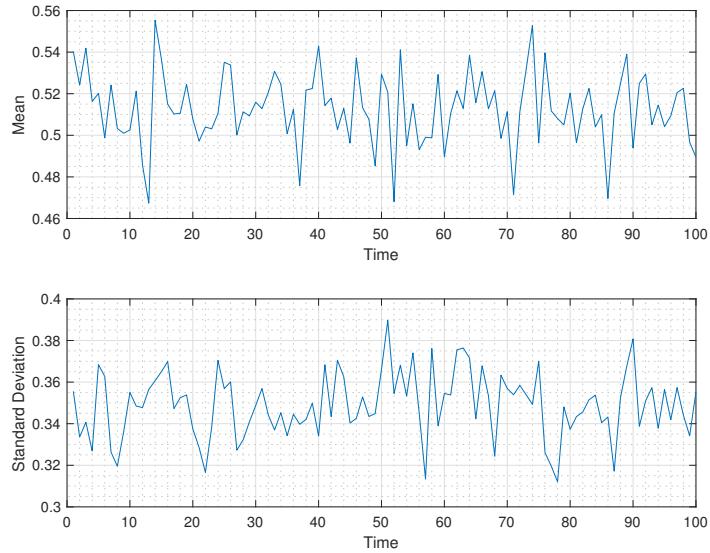


Figure 7: Mean and standard deviation of  $rp2$  with respect to time

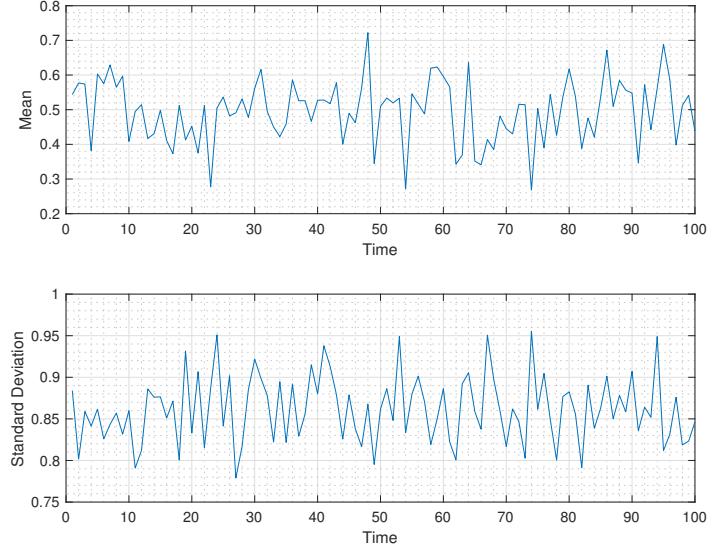


Figure 8: Mean and standard deviation of  $rp3$  with respect to time

### 1.2.2 Ergodicity when $M = 4$ and $N = 1000$

We now have 4 samples at 1000 different moments of time. In other words, 1000 realizations of 4 samples.

$rp1$  cannot be ergodic because its ensemble mean is linearly increasing with time, and therefore the ensemble average cannot equal the time average for all time. This means that examining multiple ensembles at one point in time will not reveal the properties of the signal completely.

$rp2$  is not ergodic either. Each set of 4 samples for every time point provides different values of mean and standard deviation, which can be observed in Table 1. This means that the system's properties cannot be gleaned from analyzing all ensembles at a single point in time.

Trial	Expectation of Mean	Expectation of Standard Deviation
1	0.2630	0.3411
2	0.7350	0.2902
3	0.5625	0.2404
4	0.5309	0.2292
5	0.6480	0.3091

Table 1: Variations of mean and standard deviation of  $rp2$

$rp3$  is ergodic since it has a constant mean and standard deviation over all time. Therefore analyzing all ensembles at one point in time can provide accurate information about the entire process.

### 1.2.3 A Mathematical Description of the Mean and Standard Deviation

#### 1.2.3.1 rp1

*rp1* is defined as

$$v_t = cb \times \sin\left(\frac{t\pi}{T}\right) + at \quad (1)$$

where  $a = 0.02$ ,  $b = 5$  and  $c \sim U(-0.5, 0.5)$ . Therefore  $E[c] = 0$  and  $Var(c) = \frac{1}{12} \Rightarrow \sigma_c = \frac{1}{\sqrt{12}}$ .  $t$  represents the particular sample and  $T$  is the total number of samples.

$$E[v_t] = E[cb \times \sin\left(\frac{t\pi}{T}\right) + at] = E[c] \times E[b \times \sin\left(\frac{t\pi}{T}\right)] + E[at] = 0 + E[at] = aE[t] = at = \frac{t}{50} \quad (2)$$

$$Var(v_t) = E[v_t^2] - E[v_t]^2 = E[c^2 b^2 \sin^2\left(\frac{t\pi}{T}\right) + a^2 t^2 + 2cb \sin\left(\frac{t\pi}{T}\right)at] - a^2 t^2 \quad (3)$$

$$= E[c^2 b^2 \sin^2\left(\frac{t\pi}{T}\right)] + E[a^2 t^2] - a^2 t^2 \quad (4)$$

$$= E[c^2] \times b^2 \sin^2\left(\frac{t\pi}{T}\right) \quad (5)$$

$$= \frac{1}{12} \times b^2 \sin^2\left(\frac{t\pi}{T}\right) \quad (6)$$

(7)

Note that we have found  $E[c^2]$  using  $Var(c) = E[c^2] - E[c]^2$ . We now have

$$\sigma_{v_t} = \sqrt{Var(v_t)} = \frac{1}{\sqrt{12}} b \sin\left(\frac{t\pi}{T}\right) = 1.44 \times \sin\left(\frac{t\pi}{T}\right) \quad (8)$$

Thus both the mean and standard deviation of *rp1* match what was observed in Section 1.2.1.

#### 1.2.3.2 rp2

*rp2* is defined by Equation 9. Note that the random variable  $Z$  is derived from a matrix with  $M \times N$  distinct random numbers, and therefore depends on both the ensemble members as well as the sample instance.  $X$  and  $Y$  on the other hand only depend on the sample instance.

$$v_t = X_n + Y_n \times Z_{n,t} \quad (9)$$

$X$  and  $Y$  are simply randomly distributed numbers between 0 and 1, so  $E[X_n]$  and  $E[Y_n]$  both equal 0.5. The standard deviation is  $\sigma_{X_n} = \sqrt{Var(X_n)} = \frac{1}{\sqrt{12}}$ .

Since  $Z_n$  is randomly distributed between -0.5 and 0.5,  $E[Z_n] = 0$ . Its standard deviation is also  $\frac{1}{\sqrt{12}}$  since it is a uniform distribution spanning a length of 1. Therefore

$$E[v_t] = E[X_n] + E[Y_n] \times E[Z_n] = E[X_n] + E[Y_n] \times 0 = E[X_n] \Rightarrow X_n \rightarrow 0.5 \quad (10)$$

This means that although a single realization of *rp2* has a random mean, the mean tends to 0.5 for over many realizations.

The standard deviation of *rp2* is

$$Var(v_t) = E[v_t^2] - E[v_t]^2 = E[X_n^2 + 2X_n Y_n \times Z_{n,t} + Y_n^2 Z_{n,t}^2] - X_n^2 \quad (11)$$

$$= E[X_n^2] + E[Y_n^2] E[Z_{n,t}^2] - X_n^2 \quad (12)$$

since  $E[Z_{n,t}] = 0$ .  $E[X_n^2]$  can be found from  $Var(X_n) = E[X_n^2] - E[X_n]^2$ , which gives  $E[X_n^2] = \frac{1}{12} + 0.25 = 0.333$ . Similarly  $E[Y_n] = 0.333$  and  $E[Z_{n,t}^2] = \frac{1}{12}$ .

$$\sigma_{v_t} = \sqrt{Var(v_t)} = \sqrt{0.333 + (0.333 \times \frac{1}{12}) - 0.25} = 0.333 \quad (13)$$

Thus both the mean and standard deviation of *rp2* match what was observed in Section 1.2.1.

### 1.2.3.3 rp3

*rp3* is defined as

$$v_t = m \cdot X + a, \text{ where } X = U(-0.5, 0.5), m = 3, \text{ and } a = 0.5 \quad (14)$$

Therefore its expected value is

$$E(v_t) = m \cdot E(X) + a = 0 + a = a = 0.5 \quad (15)$$

and its standard deviation is

$$Var(v_t) = E[v_t^2] - E[v_t]^2 = E[m^2x^2 + a^2 + 2 \cdot amX] - 0.25 = 0.75 \quad (16)$$

$$\sigma_{v_t} = \sqrt{Var(v_t)} = \sqrt{0.75} \approx 0.87 \quad (17)$$

Thus both the mean and standard deviation of *rp3* match what was observed in Section 1.2.1.

## 1.3 Estimation of Probability Distributions

### 1.3.1 For a stationary process

The following code has been used to implement the PDF estimator:

```

1 N = 10000;
2 v = randn(1,N);
3 [freq,bin] = hist(v,100); % Returns frequency and centre of each bin.
4 freq = freq/trapz(bin,freq); % trapz calculates the integral by the trapezoidal method
5 bar(bin,freq);

```

$N$  is set to a minimum of 100 as required. A larger  $N$  (more samples) will give greater matching between theoretical and estimated values since more data is available for describing the curve.

We need to scale the PDF so that its total integral is 1. Dividing the frequency of each bin in the histogram by  $N$ , as we did in Section 1.1.4 is only accurate if the distribution is uniform or the bars are small. We now use an improved technique where we divide by the integral of the curve calculated by the trapezoidal method. The estimated PDF is shown in Figure 9.

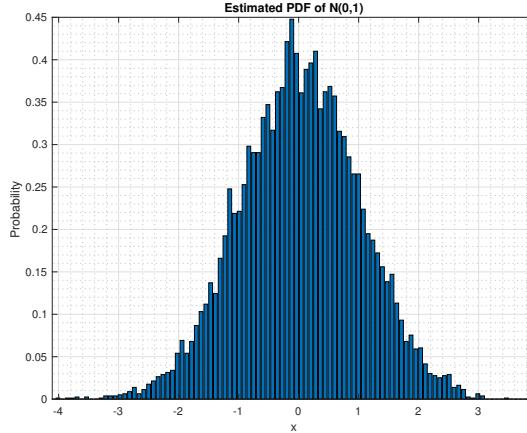


Figure 9: Estimated PDF for 10,000 samples of  $N(0, 1)$  using improved technique

### 1.3.2 For a stationary and ergodic process

We analyze  $rp3$  since it is the only signal that is both stationary and ergodic. We can clearly see in Figure 10 that the matching between theoretical and estimated values increases with a greater number of samples.

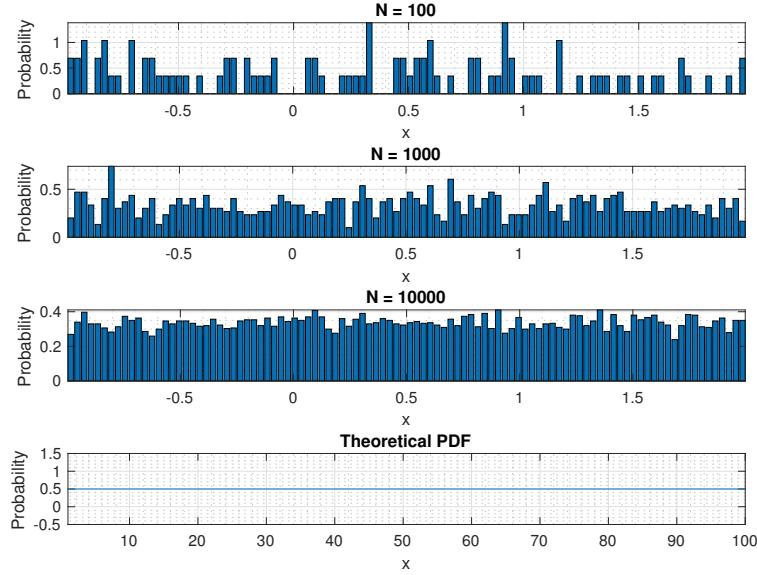


Figure 10: Comparing theoretical and estimated PDF of  $rp3$

### 1.3.3 For a non-stationary process

The estimated PDF, shown in Figure 11, incorrectly tells us that  $rp2$  is a random number between 1 and 20. We know that  $rp2$  is not stationary since it is linearly increasing with time, and therefore our program causes loss of critical information about the signal. It is worth noting that stationary processes would not contain this kind of data.

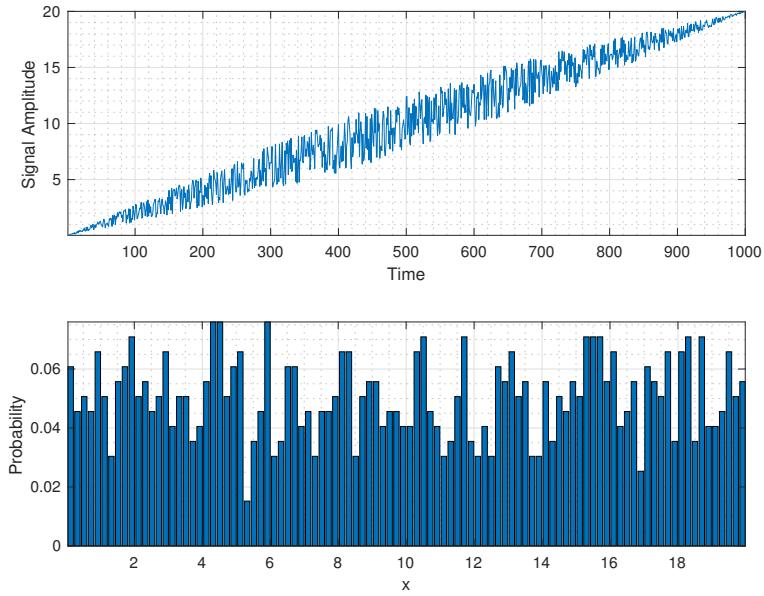


Figure 11: The signal  $rp2$  and its estimated PDF

## 2 Linear Stochastic Modeling

### 2.1 ACF of uncorrelated and correlated sequences

#### 2.1.1 Using xcorr in Matlab

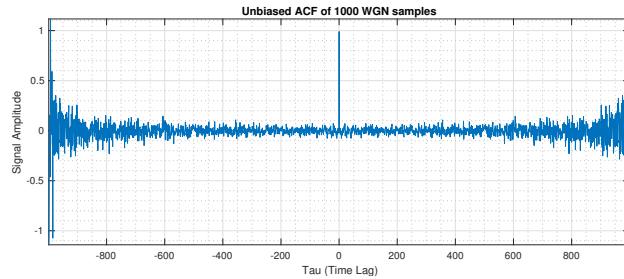


Figure 1: Unbiased autocorrelation of White Gaussian Noise

Figure 1 shows the autocorrelation function of 1000 random White Gaussian Noise (WGN) samples. The function is symmetric about  $\tau = 0$  since the signal is real. Ideally, the WGN samples would be random and hence completely uncorrelated, so the ACF would have a Dirac delta function at  $\tau = 0$  and be 0 everywhere else. There is a Dirac delta signal at  $\tau = 0$  since the signal is perfectly correlated, but the ACF is not zero everywhere else. The cross correlation estimate is close to 0 around  $\tau = 0$  and increases after  $\tau = 500$ . This is because less than half the signal (of length 1000) is available for comparison and there is a greater error in the estimate. Therefore, the amplitude of the estimate increases as the signal length compared decreases towards  $\tau = 1000$ .

#### 2.1.2 Zooming in on $\tau = 0$

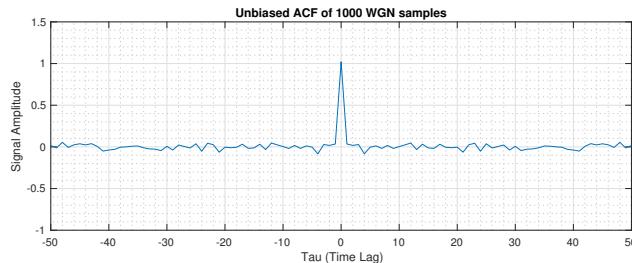


Figure 2: Zooming in to observe the ACF near  $\tau = 0$

Figure 2 shows the cross correlation estimate is almost identical to the ideal case around  $\tau = 0$ . However, for large  $\tau$ , the ACF estimate is much larger and becomes higher than the peak of 1 close to  $\tau = \pm 999$ .

#### 2.1.3 Effects of large lag $\tau$

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

We know from Equation 1 that the autocorrelation estimator's output increases when the number of samples available for comparison decreases. This happens when  $\tau$  increases. This makes sense intuitively since the chances for compared samples to be identical increases when the number of samples decreases. We said earlier that the estimator begins diverging roughly after  $\tau = \pm 500$ , so a conservative empirical bound would be half this value, i.e.,  $\tau = 250$ . In the general case, the bound would be  $\frac{N}{4}$ .

#### 2.1.4 Using a Moving Average Filter

Figure 3 shows the effects of applying an  $N^{\text{th}}$  order Moving Average Filter to 1000 WGN samples. The ideal ACF output for an  $N^{\text{th}}$  order MA filter should be  $ACF(\tau) = N \cdot \Delta(\frac{\tau}{N})$ , where  $\Delta(t)$  is the triangle function. Figure 3a shows that the ACF of a 9<sup>th</sup> order filter is the triangular function between  $-8 < \tau < 8$ . While

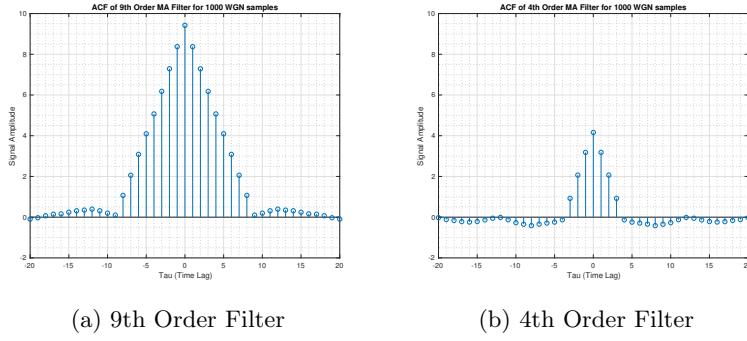


Figure 3: Comparing the ACF of Moving Average Filter outputs for different orders

the ACF should ideally be 0 for  $\tau$  outside this range, we see non zero values since we are using an estimate. Decreasing the filter order to 4 decreases the width and height of the triangle, and we can expect a scaling up if the order is increased.

An FIR filter can be described by the equation  $y[n] = \sum_{i=0}^{N-1} b_i x[n - i]$ , where  $N$  is the filter order. If all the coefficients are  $\frac{1}{N}$ , then  $y[n] = \frac{1}{N} \sum_{i=0}^{N-1} x[n - i]$ . This is equal to the local sample  $\hat{m}$ . Note that the number of samples must be at least as large as the filter order for a good estimate.

### 2.1.5 Correlation of a stochastic process

$$R_Y(\tau) = R_X \tau * R_h(\tau) \quad (2)$$

where  $X_n$  is an uncorrelated process and  $Y_n$  is a filtered version of  $X_n$ . Since  $X_n$  is an uncorrelated process,  $R_X(\tau) = \alpha \delta(\tau) \Rightarrow R_Y(\tau) = \alpha R_h(\tau)$  since the convolution of any function with the delta function is equal to itself. Thus the ACF of Y is equal to a scaled version of the ACF of the impulse response, which in this case is  $N \cdot \Delta(\frac{\tau}{N})$ .

## 2.2 Cross correlation function

### 2.2.1 Cross correlation of X and Y from 2.2

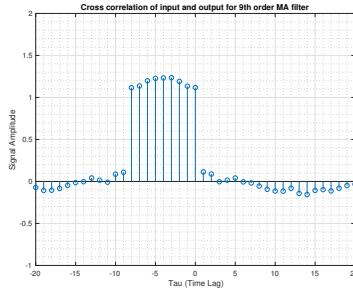


Figure 4: Cross correlation of input and output for 9th order MA filter

Figure 4 shows that X is correlated to the current and last 8 values of Y. This is expected since the filter order is 9. Since  $X_n$  is uncorrelated,  $R_{XY}(\tau) = h(\tau) * R_X(\tau) = h(\tau) * \delta(t) = h(\tau)$ . In this case,  $h(\tau)$  is the pulse train represented by  $\sum_{i=0}^N \delta(\tau - i)$ .

### 2.2.2 Application to System Identification

If we cross correlate a function with its output through any LTI system, we can obtain the impulse response of the system as long as the original function is uncorrelated. The order of the filter can then be deduced by the number of delta peaks in the impulse response. The coefficients can also be estimated if the system is an FIR. Thus system identification can be conducted by observing any of these characteristic properties.

## 2.3 Autoregressive Modelling

### 2.3.1 Stability of coefficients for AR(2)

AR(2) is defined as:

$$x[n] = a_1 x[n-1] + a_2 x[n-2] + w[n], \text{ where } w[n] \sim N(0, 1) \quad (3)$$

Figure 5 shows the stable and unstable coefficient pairs  $(a_1, a_2)$  in the blue and red regions respectively. The response will be bounded (i.e WSS) within the blue region. The roots will diverge outside the stable region and the mean will vary (i.e not WSS).

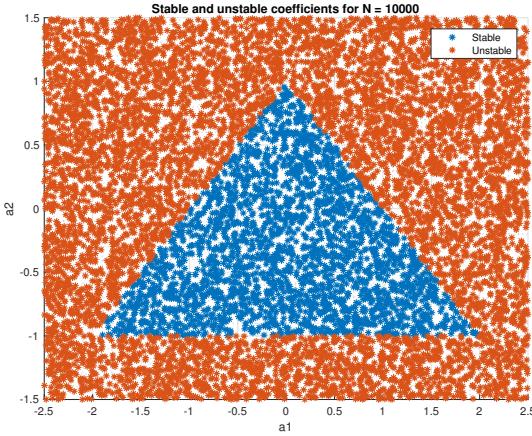


Figure 5: Stability of coefficients  $a_1$  and  $a_2$

The system's poles must lie within the unit circle for stability. The roots of the characteristic equation  $C(z) = z^2 - a_1 z - a_2$  are  $z = \frac{a_1 \pm \sqrt{a_1^2 + 4a_2}}{2}$ . For the positive root, we have:

$$\begin{aligned} \frac{a_1 \pm \sqrt{a_1^2 + 4a_2}}{2} &< 1 \\ a_1^2 + 4a_2 &< (2 - a_1)^2 \\ a_1^2 + 4a_2 &< 4 + a_1^2 - 4a_1 \\ a_1 + a_2 &< 1 \end{aligned}$$

Similar calculations for the negative root give  $a_2 - a_1 < 1$ .

The characteristic equation can now be written as:

$$\begin{aligned} C(z) &= (z - \lambda_1)(z - \lambda_2) = 0 \\ &= z^2 - (\lambda_1 + \lambda_2)z + \lambda_1 \lambda_2 = 0 \\ \therefore a_1 &= (\lambda_1 + \lambda_2) \\ a_2 &= \lambda_1 \lambda_2 \end{aligned}$$

Since both roots are less than 1,  $|a_2| < 1$ .

Thus the triangular plane of stability is defined by following inequalities:

$$a_1 + a_2 < 1 \quad (4)$$

$$a_1 - a_2 < 1 \quad (5)$$

$$|a_2| < 1 \quad (6)$$

### 2.3.2 Sunspot Time Series

We cannot see any recurrence in the ACF of sunspot data for  $N=5$  (see Figure 6), but can observe the pattern repeating about every 13 years for  $N=20$  (See Figure 7). The ACF with 250 samples has irregularities (see Figure 8) due to the number of samples being greater than the empirical bound of  $N/4$  discussed earlier. If we stay within this bound, we see the perfect sinusoid pattern shown in Figure 9.

The zero mean version provides equal representation for positive and negative elements and prevents a build up of the product of shifted elements. This reveals a clearer picture of the underlying trends in the sunspot data. For example, the ACF of sunspot data for  $N=20$  would suggest that the signal is more similar to itself when it is shifted, which is obviously incorrect. Standardizing the data removes this problem. It also makes the periodicity of data for  $N=5$  and  $N=250$  more clear.

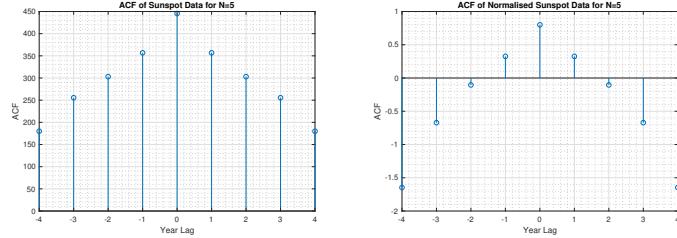


Figure 6: ACF of sunspot data with data length  $N=5$

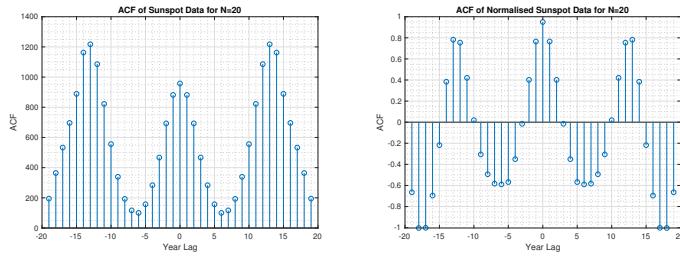


Figure 7: ACF of sunspot data with data length  $N=20$

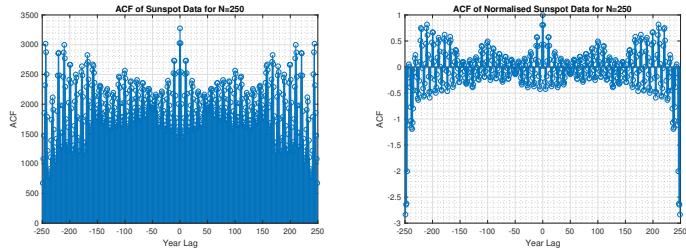


Figure 8: ACF of sunspot data with data length  $N=250$

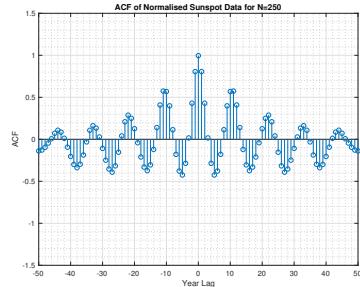


Figure 9: Zooming into the normalized ACF with  $N=250$  shows a sinusoidal pattern

### 2.3.3 The Yule Walker Equations and Partial Correlation

Figure 10a clearly shows that the best model order is 2 since the coefficients for all others converge to 0. Figure 10b illustrates the difference between the regular and standardized partial correlation function (PCF). The standardized version is more accurate since it removes any offsets that obscure underlying trends. The significant decrease in the PCF after order 2 indicates that the current sunspot sample has weak correlation with the sample 3 time units ago. This property defines the AR(2) process.

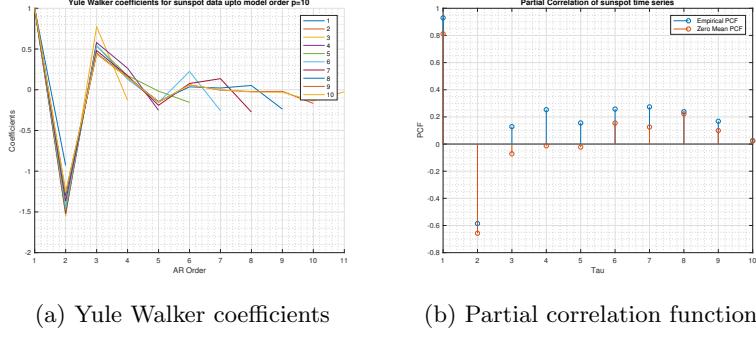


Figure 10

### 2.3.4 Determining the correct model order

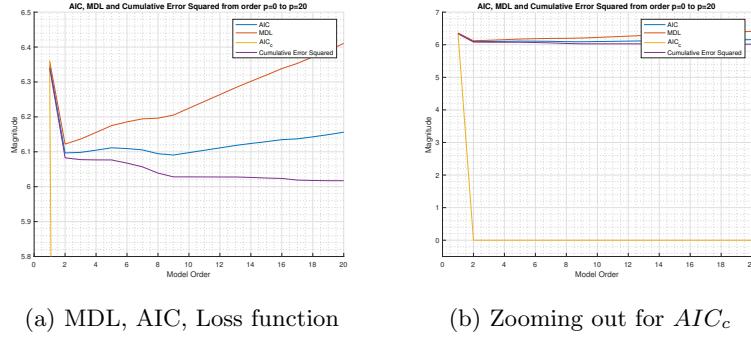


Figure 11: Selection criteria for model order

The purpose of using the Minimum Description Length (MDL), Akaike Information Criterion (AIC) and cumulative error squared is to balance the reduction in error with the increase in complexity. Since the loss function will always decrease when the order is increased, the MDL and AIC increase their respective magnitudes for higher orders, thus providing a guide to prevent unnecessary increases in the applied model order.

While the global minimum in Figure 11b occurs at  $p=9$ , the first minimum occurs at  $p=2$ . Since the drop in error between the two orders is only about 0.055, the sunspot time series should be modeled as a second order autoregressive process. AR(1) clearly follows the real sunspot data for one horizon (i.e  $m=1$ ) but deviates for  $m=2$ ,  $m=5$  and  $m=10$ .

### 2.3.5 AR Modeling to predict the sunspot time series

We now compare the the AR(1),AR(2) and AR(10) models to see which one can predict the sunspot time series best. Figure 12 clearly shows that increasing the model order allows for a more accurate prediction. All 3 orders display a fall in amplitude with increase in horizon, which means AR(1) and AR(2) are unsuitable for estimating the sunspot data over a long horizon. Increasing the order from AR(1) to AR(2) decreases the lag between the estimate and the real data. However, as discussed earlier, this increases the number of variables required for computation and hence raises the complexity of calculations involved.

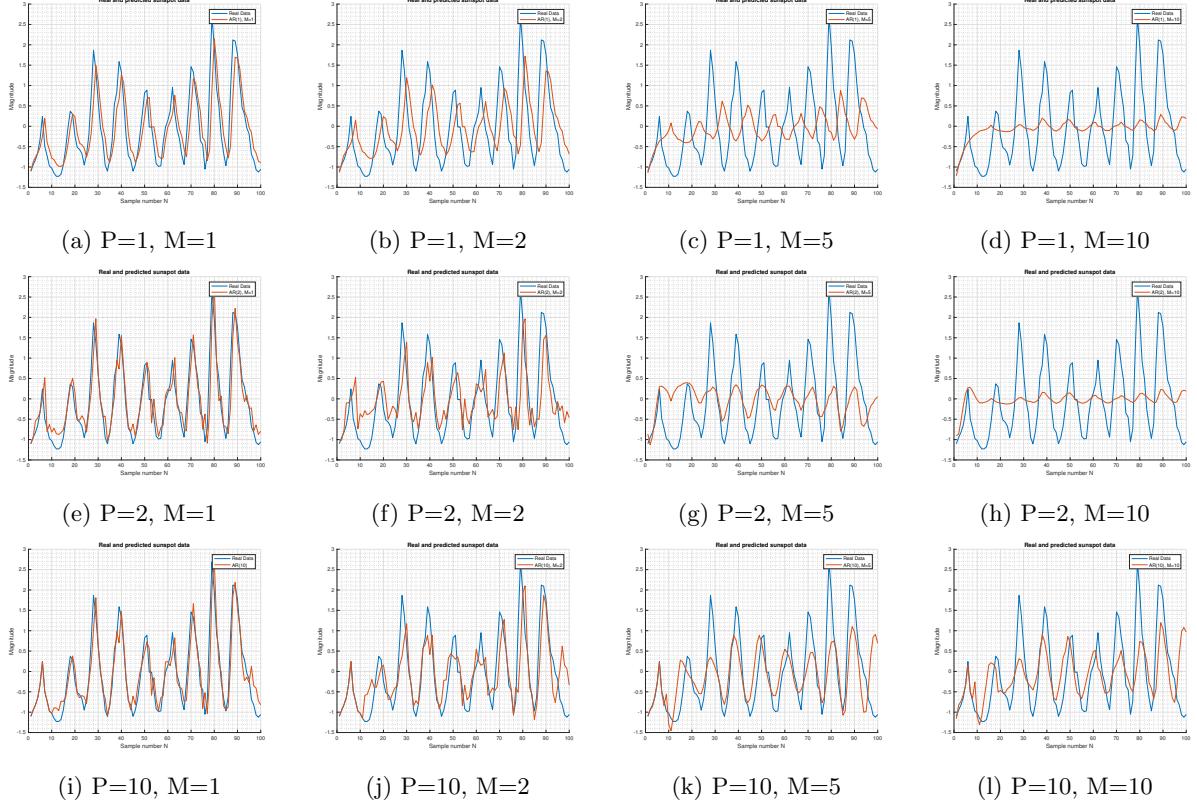


Figure 12: Comparing trends across model order and prediction horizon for  $N=100$  samples

## 2.4 Cramer-Rao Lower Bound

### 2.4.1 Sufficiency of an AR(1) model

Figure 13 shows that the error is minimum for MDL, AIC and the loss function at  $p=1$ . Since the NASDAQ data's sample size is no longer small (it is almost 3 times the size of the sunspot data), we can ignore  $AIC_c$ . The drastic decrease in PCF after order 1 indicates that the NASDAQ data has almost no correlation with the sample more than 1 time unit ago. Thus an AR(1) process is sufficient to model the NASDAQ data.

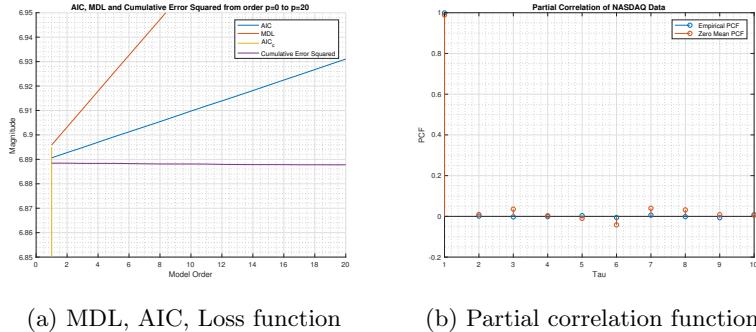


Figure 13: Selection criteria for model order of NASDAQ data

#### 2.4.2 Computing the Fisher Information Matrix

The following derivations have been sourced from Stephen Kay's 'Fundamentals of Signal Processing: Estimation Theory'. The PSD implied by the AR model is:

$$P_{xx}(f; \theta) = \frac{\sigma_u^2}{|A(f)|^2} \quad (7)$$

where  $\theta = [a[1]a[a]] \cdots a[p]\sigma_u^2]^T$  and  $A(f) = \sum_{m=1}^p a[m]exp(-j2\pi fm)$ . The partial derivatives are:

$$\frac{\partial \ln P_{xx}(f; \theta)}{\partial a[k]} = -\frac{\partial \ln |A(f)|^2}{\partial a[k]} \quad (8)$$

$$= -\frac{1}{|A(f)|^2} [A(f)exp(j2\pi fk) + A^*(f)exp(-j2\pi fk)] \quad (9)$$

$$\frac{\partial \ln P_{xx}(f; \theta)}{\partial \sigma_u^2} = \frac{1}{\sigma_u^2} \quad (10)$$

For  $k = 1, 2, \dots, p$  and  $l = 1, 2, \dots, p$ , we have:

$$\begin{aligned} I[\theta]_{kl} &= \frac{N}{2} \int_{-0.5}^{0.5} \frac{1}{|A(f)|^4} [A(f)exp(j2\pi fk) + A^*(f)exp(-j2\pi fk)] \times [A(f)exp(j2\pi fl) + A^*(f)exp(-j2\pi fl)] df \\ &= \frac{N}{2} \int_{-0.5}^{0.5} \frac{1}{A^*(f)^2} exp(j2\pi f(k+l)) + \frac{1}{|A(f)|^2} exp(j2\pi f(k-l)) + \frac{1}{|A(f)|^2} exp(j2\pi f(l-k)) + \frac{1}{A^2(f)} exp(-j2\pi f(l+k)) df \end{aligned} \quad (11)$$

Using the Hermitian property of the integrand ( $A(-f) = A^*(f)$ ), we have

$$\int_{-0.5}^{0.5} \frac{1}{A^*(f)^2} exp(j2\pi f(k+l)) df = \int_{-0.5}^{0.5} \frac{1}{A^2(f)} exp(-j2\pi f(l+k)) df \quad (12)$$

$$\int_{-0.5}^{0.5} \frac{1}{|A(f)|^2} exp(j2\pi f(k-l)) df = \int_{-0.5}^{0.5} \frac{1}{|A(f)|^2} exp(j2\pi f(l-k)) df \quad (13)$$

$$\therefore [I(\theta)]_{kl} = N \int_{-0.5}^{0.5} \frac{1}{A^*(f)^2} exp(j2\pi f(k+l)) + N \int_{-0.5}^{0.5} \frac{1}{|A(f)|^2} exp(j2\pi f(k-l)) df \quad (14)$$

The second integral is the inverse Fourier transform of  $\frac{1}{A^*(f)^2}$  evaluated at  $n = k + l > 0$ . This term is 0 since the sequence is the convolution of 2 anti-causal sequences, i.e,

$$F^{-1}\left[\frac{1}{A(f)}\right] = \begin{cases} h[n], & n \geq 0 \\ 0, & n < 0 \end{cases} \quad (15)$$

$$F^{-1}\left[\frac{1}{A^*(f)^2}\right] = \begin{cases} h[-n] \star h[-n] \\ 0, \end{cases} \quad n > 0 \quad (16)$$

$$\text{therefore } [I(\theta)]_{kl} = \frac{N}{\sigma_u^2} r_{xx}(k-l) \quad (17)$$

We are given  $I_{11}$ ,  $I_{12}$  and  $I_{21}$ . For  $k = p + 1$  and  $l = p + 1$ :

$$[I(\theta)]_{kl} = \frac{N}{2} \int_{-0.5}^{0.5} \frac{1}{\sigma_u^4} df = \frac{N}{2\sigma_u^4} \quad (18)$$

$$\therefore I(\theta) = \begin{bmatrix} \frac{N}{\sigma_u^2} R_{xx} & 0 \\ 0^T & \frac{N}{2\sigma_u^4} \end{bmatrix} \quad (19)$$

where  $[R_{xx}]_{ij} = r_{xx}(i-j)$  is a  $p \times p$  Toeplitz autocorrelation matrix and 0 is a  $p \times 1$  vector of zeros.

### 2.4.3 Computing the lower bound of the variance

Upon inverting the Fisher Information matrix, we get:

$$var(a[\hat{k}]) >= \frac{\sigma_u^2}{N} [R_{xx}^{-1}]_{kk} \text{ for } k = 1, 2, \dots, p \quad (20)$$

$$var(\hat{\sigma}_u^2) >= \frac{2\sigma_u^4}{N} \quad (21)$$

$$\text{Using } p = 1, (\hat{a}[1]) >= \frac{\sigma_u^2}{Nr_{xx}[0]} \quad (22)$$

$$\text{But } r_{xx}[0] = \frac{\sigma_u^2}{1 - a^2[1]} \quad (23)$$

$$\text{So that } var(\hat{a}[1]) >= \frac{1}{N}(1 - a^2[1]) \quad (24)$$

This shows that it is easier to estimate the filter parameter when  $a[1]$  is closer to 1 than to 0. Since the pole of the filter is at  $-a[1]$ , the filter parameters of processes with PSDs having sharp peaks are more easily estimated.

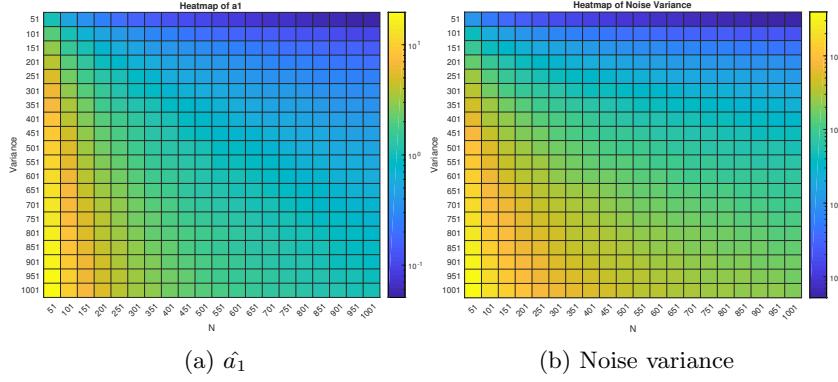


Figure 14: Cramer Rao lower bound heatmaps for  $\hat{\sigma}^2$  and  $\hat{a}_1$

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

For  $p = 1$ ,

$$\frac{P_x(\hat{f}; \theta)}{\partial a_1} = \frac{\sigma^2 [\exp(j2\pi f)A(f) + \exp(-j2\pi f)A(-f)]}{[A(f)A(-f)]^2} \quad (25)$$

$$= \frac{\sigma^2 [\exp(-j2\pi f) - a_1 + \exp(j2\pi f) - a_1]}{|A(f)|^4} \quad (26)$$

$$= \frac{2\sigma^2 [\cos(2\pi f) - a_1]}{|A(f)|^4} \quad (27)$$

$$\text{and } \frac{P_x(\hat{f}; \theta)}{\partial \sigma^2} = \frac{1}{|A(f)|^2} \quad (28)$$

$$\therefore var(P_x(\hat{f}; \theta)) <= \left[ \frac{2\sigma^2 [\cos(2\pi f) - a_1]}{|A(f)|^4} \quad \frac{1}{|A(f)|^2} \right] \begin{bmatrix} \frac{1-a_1^2}{N} & 0 \\ 0 & \frac{2\sigma^4}{N} \end{bmatrix} \begin{bmatrix} \frac{2\sigma^2 [\cos(2\pi f) - a_1]}{|A(f)|^4} \\ \frac{1}{|A(f)|^2} \end{bmatrix} \quad (29)$$

$$= \left[ \frac{1-a_1^2}{N} \frac{2\sigma^2 [\cos(2\pi f) - a_1]}{|A(f)|^4} \quad \frac{2\sigma^4}{N|A(f)|^2} \right] \begin{bmatrix} \frac{2\sigma^2 [\cos(2\pi f) - a_1]}{|A(f)|^4} \\ \frac{1}{|A(f)|^2} \end{bmatrix} \quad (30)$$

$$= \frac{1}{N} \left[ \frac{4\sigma^4 (\cos(2\pi f) - a_1)^2 (1 - a_1^2)}{|A(f)|^8} + \frac{2\sigma^4}{|A(f)|^4} \right] \quad (31)$$

## 2.5 Real world signals: ECG from iAMP experiment

### 2.5.1 Heart rate probability density estimate (PDE)

All sub-figures in Figure 15 have been drawn to a scale of 200 samples for easy comparison. Note that the original heart rate's PDE has almost all its data points between 20 and 100, with one around 190. The data points for the averaged PDE lies between 50 and 85 for  $\alpha = 1$  and between 30 and 50 for  $\alpha = 0.6$ .

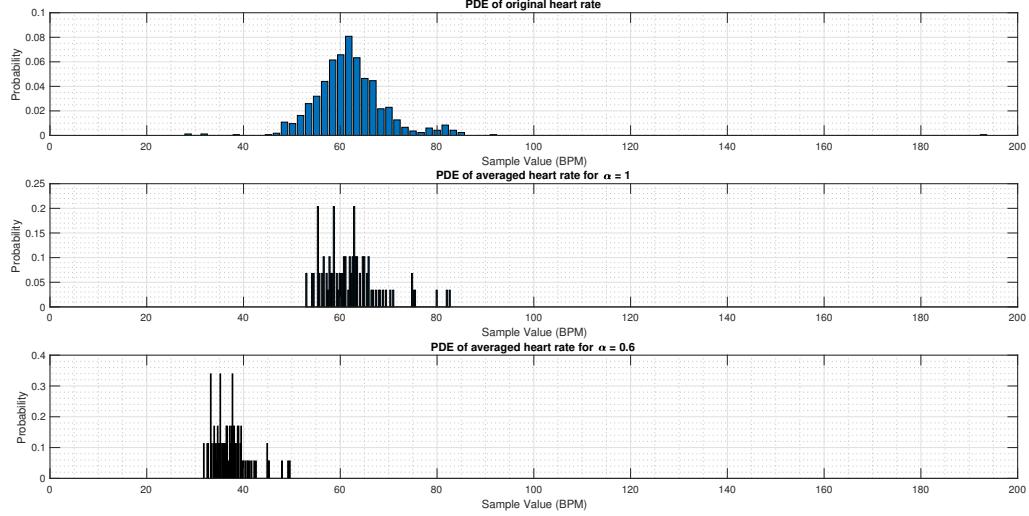


Figure 15: Comparing the PDE of the original heart rate with that of the averaged heart rate for  $\alpha = 1, 0.6$

Since the `pdf.m` function from Part 1 cannot model processes that are not stationary, the heartbeat has been modeled as a stationary process. Averaging the data in blocks of  $B$  samples (in this case 10) will reduce the variance of the data to  $\frac{\sigma}{B}$ , which can be seen in Figure 15.

Multiplying by  $\alpha$  can be used to eliminate noise bias since it shifts the mean of the averaged heart rate  $\hat{h}$ . The factor of  $\alpha$  also changes the variance of the data to  $\alpha^2$ .

Therefore, the mean of the  $\alpha = 1$  case is the same as that of the original data, but it has a lower variance. The  $\alpha = 0.6$  case has a lower mean and variance than  $\alpha = 1$  case as expected.

### 2.5.2 AR Modeling of heart rate

Figure 16 clearly shows that the heart rate data is autoregressive since it is sinusoidal and of infinite length. By contrast, Moving Average (MA) processes are of finite length and their order determines the rate of decay to 0 after some  $\tau$ . Note that the part of the ACF near  $\tau = 1000$  is particularly distorted due to the deliberately induced noise.

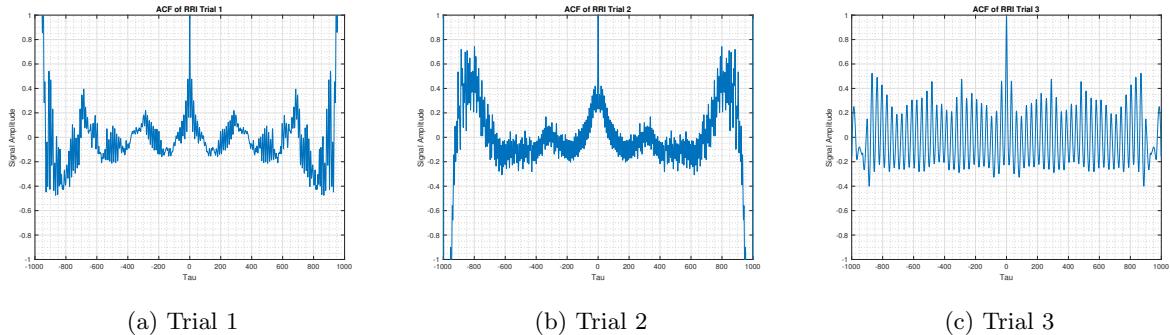


Figure 16: Autocorrelation function of RRI Trials

Trial 1 has a global minimum for MDL at order 2, AIC at order 9, and loss function at 15. Since the PCF of trial 1 is bounded and decreasing after order 2, this process can be modeled as an AR(2) system.

This is also supported by the  $AIC_c$  measure.

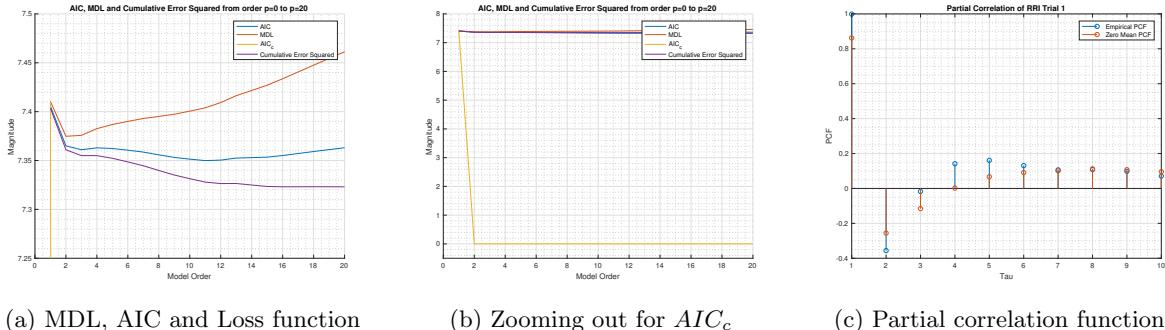


Figure 17: Model selection criteria and partial correlation function for RRI trial 1

Trial 2 has a global minimum for MDL at order 2, AIC at order 6, and loss function at 16. Since the PCF of trial 2 is bounded and decreasing after order 2, this process can also be modeled as an AR(2) system. Again, this is supported by the  $AIC_c$  measure.

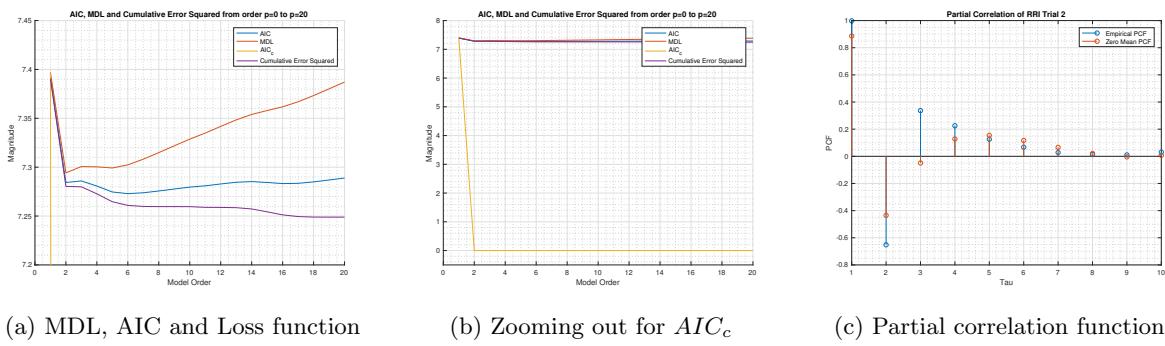


Figure 18: Model selection criteria and partial correlation function for RRI trial 2

Trial 3 has a global minimum for MDL at order 4, AIC at order 4, and loss function at 6. However, the  $AIC_c$  measure has a global minimum at order 2. Since the PCF of trial 1 is bounded and decreasing after order 2, this process can be modeled as an AR(2) system.

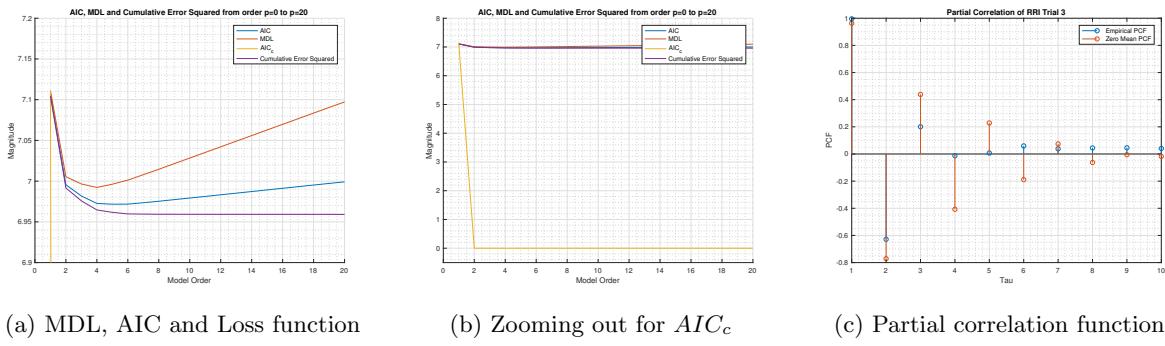


Figure 19: Model selection criteria and partial correlation function for RRI trial 3

Thus a second order AR process is best suited to model the measured heart rate data. The PCF stays bounded after this point and the drop in the loss function is negligible for higher orders.

### 3 Spectral Estimation and Modeling

We provide White Gaussian Noise as input to the `pgm.m` function which generates the estimated power spectral density (PSD) by the periodogram method. Since the WGN input is real, the PSD will be symmetric about  $\pi$  in the frequency domain, which corresponds to 0.5 on the normalized frequency scale. This is clearly shown in Figure 1.

However, the periodogram's estimated PSD does not match the ideal PSD for WGN, which is equal to 1 at all frequencies. The PSD is the Fourier Transform of the ACF, and since the unbiased ACF estimator's variance increases for large  $\tau$ , the periodogram estimator's variance also increases for large  $\tau$ . Increasing the number of samples makes the mean of the data points tend to the ideal case of 1 but does not make the variance converge to 0. Thus the estimator is inconsistent.

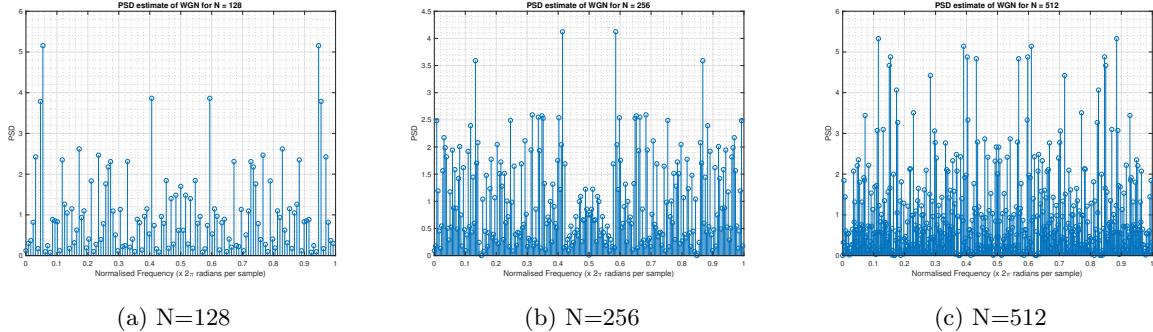


Figure 1: Estimated power spectral density for White Gaussian Noise

#### 3.1 Averaged periodogram estimates

##### 3.1.1 Smoothing the PSD estimate

Smoothing creates an approximate function that retains important trends and patterns in the data while removing noise and other fine phenomena. While this reduces the variance of the data set, it decreases the resolution of frequency (0 to 0.5 instead of 0 to 1) and introduces a bias in the estimate. Figure 2 clearly shows that the smoothed estimate removes the large peaks from the original estimate.

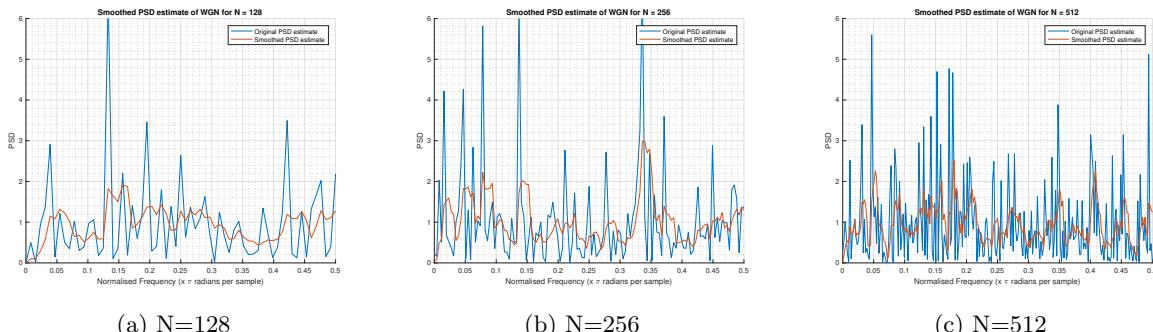


Figure 2: Smoothed estimated power spectral density for White Gaussian Noise

##### 3.1.2 PSD estimates for non-overlapping WGN segments

The 8 non-overlapping segments are parts of a WGN signal and therefore completely random. The position of the peaks follows no pattern and their existence is attributed to the large variance of the data rather than a bias for specific frequencies. Thus each segment is an unreliable estimate for the ideal unity frequency PSD.

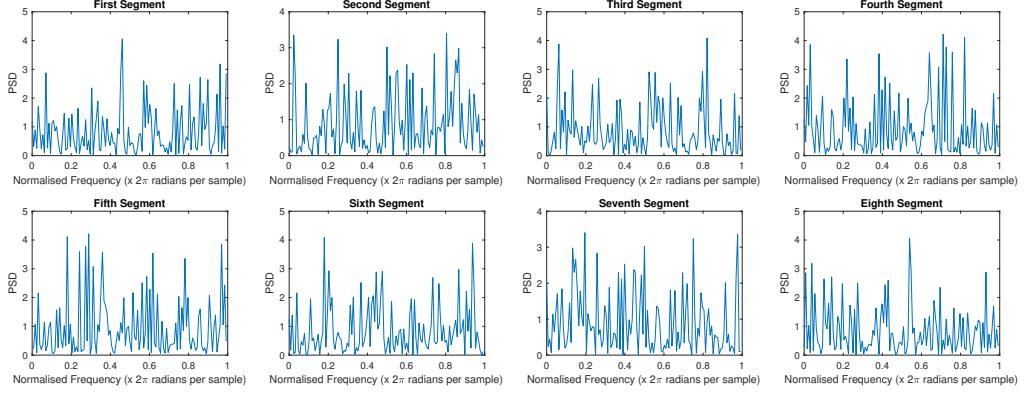


Figure 3: PSD for 1024 samples split into 8 segments

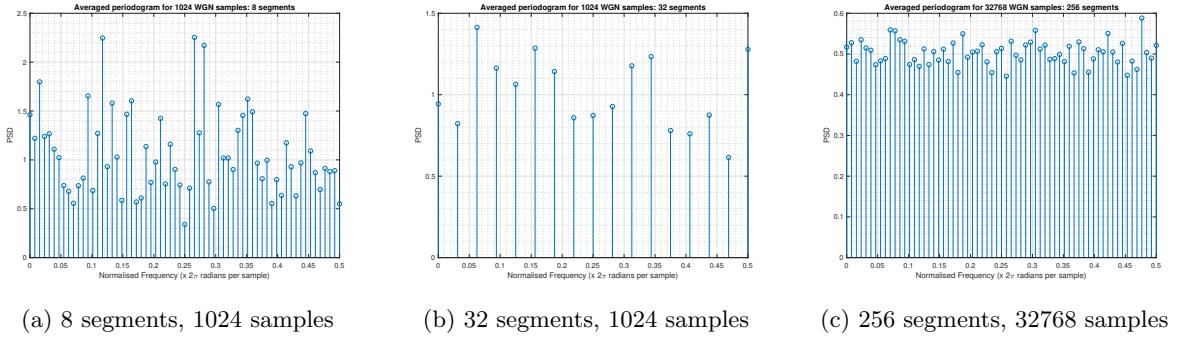


Figure 4: Averaged periodogram

### 3.1.3 Averaged Periodogram

Figure 4a shows the average periodogram of 1024 WGN samples where averaging has been carried out in segment lengths of 8. Increasing the number of segments is linearly proportional to the decrease in variance, but we lose frequency resolution if the total number of samples is constant (shown in Figure 4b). However, if we keep the samples per segment constant and simply increase the number of segments, we get a much more accurate estimate of the PSD (shown in Figure 4c).

## 3.2 Spectrum of autoregressive processes

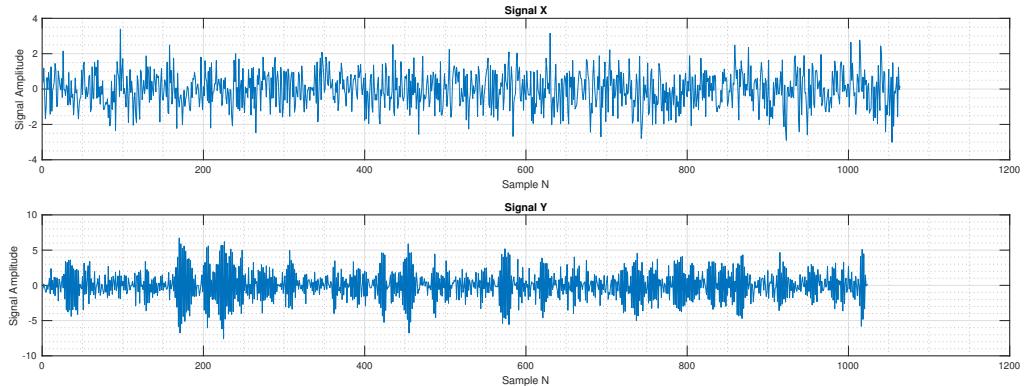


Figure 5: Signals X and Y in time domain

We can see in Figure 5 that the WGN sequence  $\mathbf{x}$  has been high pass filtered into  $\mathbf{y}$  since the filter has attenuated the low frequency elements and left the high frequency components.

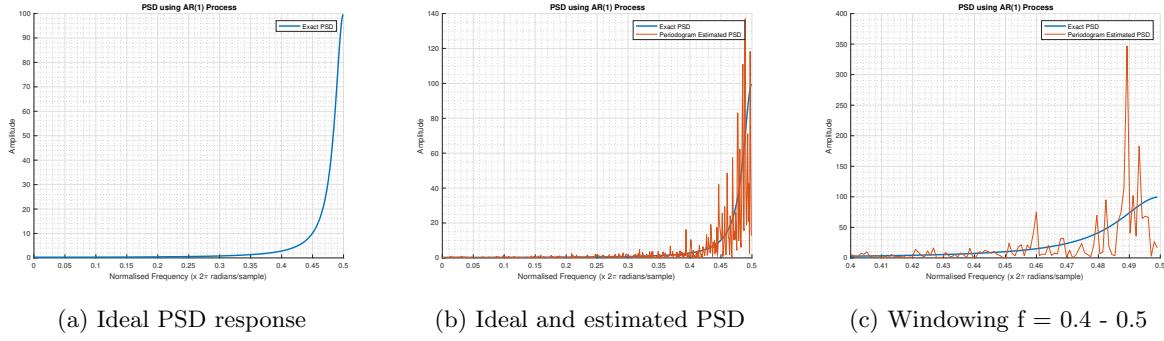


Figure 6: Ideal PSD, periodogram estimate of PSD, and windowing the PSD

### 3.2.1 Ideal PSD response

Figure 6a shows the exact PSD of the output signal  $y$ . The normalized cut-off frequency is approximately 0.39 Hz and the maximum PSD value is at 0.5 Hz. Note that the plots in Figure 6 have been shown only up to  $f = 0.5$  Hz since the figures will be symmetric about  $f = 0.5$  Hz.

### 3.2.2 Periodogram estimate of PSD response

The periodogram estimate is based on a finite number of samples and hence has some error in its estimation. However, the periodogram successfully suppresses lower frequencies and allows high frequencies to pass.

### 3.2.3 Effect of windowing

The finite length from 0.4 to 0.5 is obtained by multiplying the long signal by a window in the time domain, which corresponds to convolution with a sinc function in the frequency domain.

$$\mathcal{F}_{rect(\frac{t}{a})}(f) = a \cdot sinc(af)$$

This has an insignificant effect at lower frequencies where the PSD is almost 0 but distorts the PSD at higher frequencies.

### 3.2.4 Model based PSD

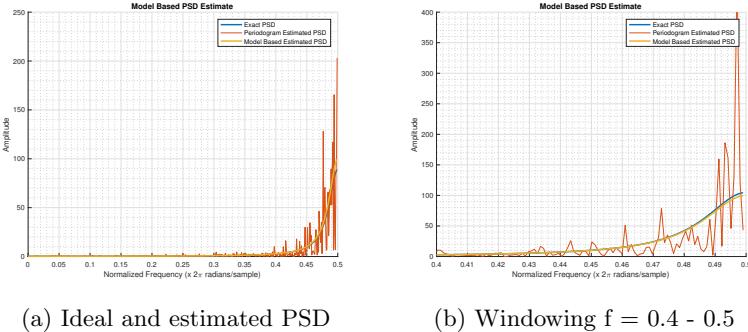


Figure 7: Comparing the PSD estimate of the model based method and the periodogram

Since we have seen how unreliable the periodogram is, we now explicitly model the signal as an AR(1) process defined by  $a_1$  and  $\sigma_x^2$ . This approach caters to the specific type of data we are using and hence is more accurate than the periodogram which uses the same approach for every data type. It is also immune to the error introduced because of the convolution with the sinc function due to windowing.

### 3.2.5 Application to sunspot time series

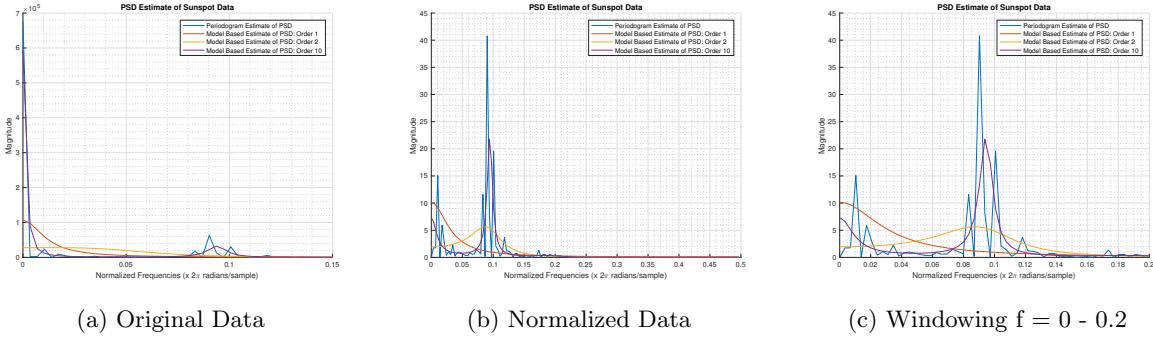


Figure 8: PSD estimate using the model based method and the periodogram for sunspot data

We know from our earlier analysis using the MDL and AIC that the optimal order for modeling the sunspot data is 2. Order 1 is not flexible enough to show the periodicity properly and order 10 is more accurate but computationally more complex. AR(2) strikes the right balance, while AR(1) under-models the data and AR(10) over-models it.

### 3.3 The Least Squares Estimation (LSE) of AR Coefficients

### 3.3.1 General Form of Cost Function

An autoregressive moving average (ARMA) model assumes a PSD of:

$$P_{xx}(f) = \frac{\sigma_u^2 |B(f)|^2}{|A(f)|^2} \quad (1)$$

where

$$B(f) = 1 + \sum_{k=1}^q b[k] \exp(-j2\pi fk) \quad (2)$$

$$A(f) = 1 + \sum_{k=1}^p a[k] \exp(-j2\pi fk) \quad (3)$$

Here the  $b[k]$  terms are the MA filter parameters and the  $a[k]$  terms are the AR filter parameters. The LSE approach only focuses on the AR parameters. We now take the inverse z transform of the PSD to find the ACF:

$$P_{xx}(z) = \frac{\sigma^2 B(z)B(z^{-1})}{A(z)A(z^{-1})} \quad (4)$$

where  $B(f) = B(\exp(j2\pi f))$  and  $A(f) = A(\exp(j2\pi f))$ .

$$Z^{-1}[A(z)P_{xx}(z)] = Z^{-1}[\sigma^2 \frac{B(z^{-1})}{A(z)^{-1}}] \quad (5)$$

Since the filter impulse response is causal,

$$h[n] = Z^{-1} \frac{B(z)}{A(z)} = 0 \text{ for } n < 0 \quad (6)$$

$$\text{and } h[-n] = Z^{-1} \frac{B(z^{-1})}{A(z^{-1})} = 0 \text{ for } n > 0 \quad (7)$$

Thus the system is anticausal and we have:

$$Z^{-1}\sigma^2 B(z) \frac{B(z^{-1})}{A(z^{-1})} = \sigma^2 b[n] \times h[-n] \quad (8)$$

$$Z^{-1}\sigma^2 B(z) \frac{B(z^{-1})}{A(z^{-1})} = \begin{cases} \sigma^2 b[n] \times h[-n] \star h[-n] \\ 0, \end{cases} \quad n > q \quad (9)$$

$$\therefore Z^{-1}[A(z)P_{xx}(z)] = Z^{-1}[\sigma^2 B(z) \frac{B(z^{-1})}{A(z^{-1})}] = 0 \text{ for } n > q \quad (10)$$

The difference equation of the ACF for  $n > q$  can now be written as:

$$\sum_{k=0}^p a[k]r_{xx}[n-k] = 0 \text{ for } n > q \quad (11)$$

where  $a[0]=1$ . These equations are known as the modified Yule Walker equations since they are identical to the original Yule Walker equations except for the fact that they hold for  $n > 0$ . Assuming  $x[n]$  is available from  $0, 1, \dots, (N-1)$ , and the ACF is estimated for lags  $n = 0, 1, \dots, M$ , where  $M \leq N - 1$ , then the LSE of  $a[k]$  will minimize:

$$J = \sum_{n=q+1}^M [r_{xx}^{\hat{}}[n] - (-\sum_{k=1}^p a[k]r_{xx}^{\hat{}}[n-k])]^2 \quad (12)$$

$$= (x - H\theta)^T(x - H\theta) \quad (13)$$

where

$$x = \begin{bmatrix} r_{xx}^{\hat{}}[q+1] \\ r_{xx}^{\hat{}}[q+2] \\ \vdots \\ r_{xx}^{\hat{}}[M] \end{bmatrix} \quad (14)$$

$$\theta = \begin{bmatrix} a[1] \\ a[2] \\ \vdots \\ a[p] \end{bmatrix} \quad (15)$$

$$H = \begin{bmatrix} r_{xx}^{\hat{}}[q] & r_{xx}^{\hat{}}[q-1] & \cdots & r_{xx}^{\hat{}}[q-p+1] \\ r_{xx}^{\hat{}}[q+1] & r_{xx}^{\hat{}}[q] & \cdots & r_{xx}^{\hat{}}[q-p+2] \\ \vdots & \vdots & \vdots & \vdots \\ r_{xx}^{\hat{}}[M] & r_{xx}^{\hat{}}[M-2] & \cdots & r_{xx}^{\hat{}}[M-p] \end{bmatrix} \quad (16)$$

### 3.3.2 Examining the observation matrix H

$$LSE_{\theta} = (H^T H)^{-1} H^T x \quad (17)$$

This is termed as the least squares modified Yule Walker equations, where the LSE is being used for the ACF estimate instead of actual data. While the observation matrix H is usually deterministic, it is now completely random. M should be small since the estimate is prone to higher error at larger lags due to the averaging of  $(N-k)$  products in the calculation of  $r_{xx}[k]$ .

### 3.4 Spectrogram for time frequency analysis: Dial tone pad

#### 3.4.1 The London land-line number

A random London land-line number of the form 020 XXXX XXXX is generated, and the signal for the first 2 digits (blank included) is shown in Figure 9. The number used for all the following analysis is 02044081677.

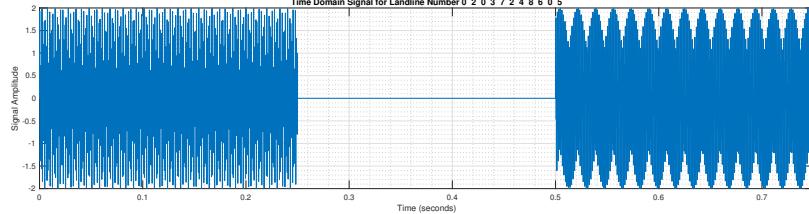


Figure 9: Signal for 0 and 2 separated by a blank

The minimum sampling frequency (Nyquist frequency) must be 2954 Hz since the maximum frequency on the given look up table is 1477 Hz. A sampling frequency of 32768 Hz ensures that there will be no aliasing for signals up to 16384 Hz, which is well above our requirement. Sampling at a higher frequency than what is required gives more effective number of bits (ENOB) which helps boost the SNR.

#### 3.4.2 Analyzing the spectrogram

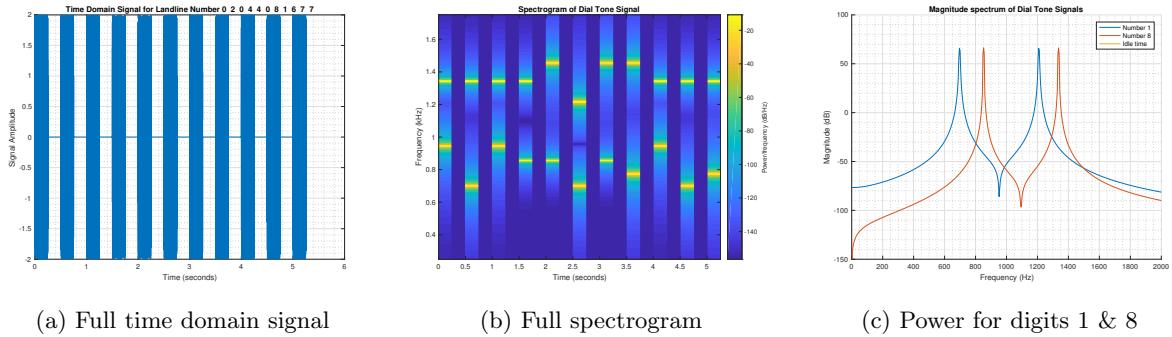


Figure 10: The spectrogram shows how the signal's power varies with time and frequency

The spectrogram in Figure 10b clearly shows the power peaks at specific frequencies for each key as time passes. The vertical variation (power vs frequency) is further examined in Figure 10c where we can see two distinct peaks for numbers 1 and 8. Note that there is no idle time signal here since we have not introduced any noise yet.

The spreading of the spectrum in the spectrogram is due to the side lobes introduced by the Hanning window. Since we need 32768 samples per second, and each key has a duration of 0.25 seconds, we split each key's spectrum into 8192 samples. This is a finite number, and hence leads to a slight displacement in the position of the peaks from the ideal.

#### 3.4.3 Key Classification

We can examine each block of 8192 samples separately to identify the two frequency peaks for the key pressed. Allowing for some tolerance from the maximum and minimum values in the look up table, we know there should be no frequency components below 675 Hz or above 1500 Hz. Since every possible frequency is well separated, we can set up bands around the table values (for example  $\pm 30$  Hz) to check for the target frequencies. This will tell us exactly which key was pressed.

### 3.4.4 Introducing channel noise

We introduce WGN of variance 2, 7 and 25 to illustrate the effect of low, medium and high noise respectively.

For low noise, the average amplitude rises from 2 to 6. The time domain signal has no visible distinction during the break between keys, but the spectrogram clearly shows which keys are pressed and hence can be used for key classification. For example, the power magnitude peaks for keys 1 and 8 are clearly visible above the noise in Figure 11c.

For medium noise, the spectrogram is still useful but only a small part of the power peaks are visible. Therefore the lower limit for detecting the keys needs to be raised. For large noise, the spectrogram cannot distinguish between noise and the keys. There is no distinction between the power peaks for the noise and the keys, and thus we cannot use this data for key classification.

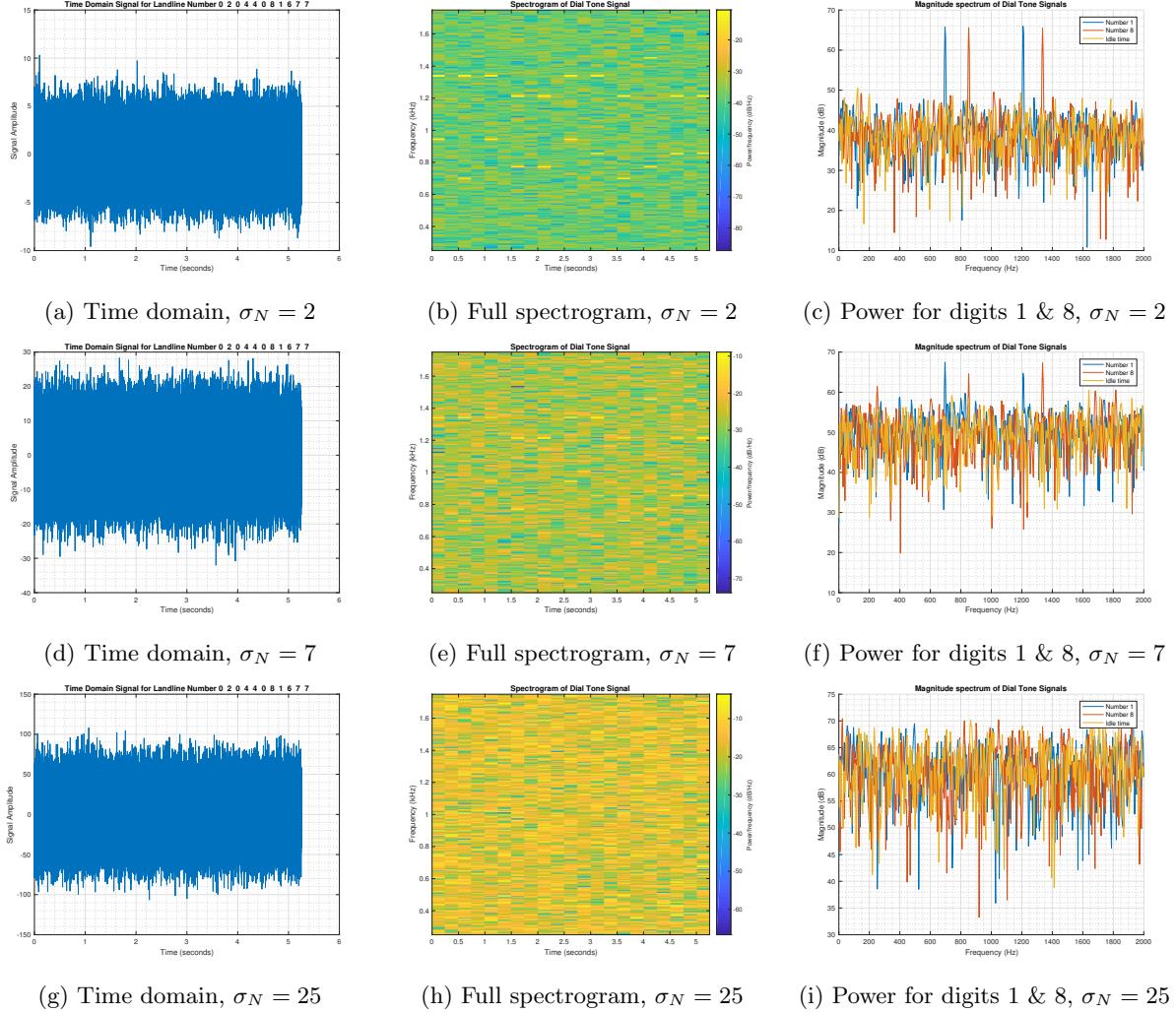


Figure 11: Comparing trends across the time domain signal and the spectrogram for numbers 1 and 8 for low, medium and high noise levels

### 3.5 Respiratory sinus arrhythmia from RR intervals

Figure 12 shows the standard periodogram, an average periodogram with a 50 second window and an average periodogram with a 150 second window for all 3 trials.

A PSD spike indicates a high heart rate at the corresponding frequency, and the main difference between the 3 trials is the position of these peaks. Trial 1 involved normal breathing and therefore has a PSD peak at a higher frequency than Trial 2 and 3, which were for constrained breathing. Trial 3 induced a lower bpm (beats per minute) and hence has a peak at a larger frequency.

If we zoom into the average periodogram with a window of 50 seconds, we see that Trial 1 has a peak at a normalized frequency of 0.06 (with harmonics at 0.12), Trial 2 has a peak at 0.95 (with harmonics at 0.145 and 0.19), and Trial 3 has a peak at 0.16 (with harmonics at 0.25 and 0.33).

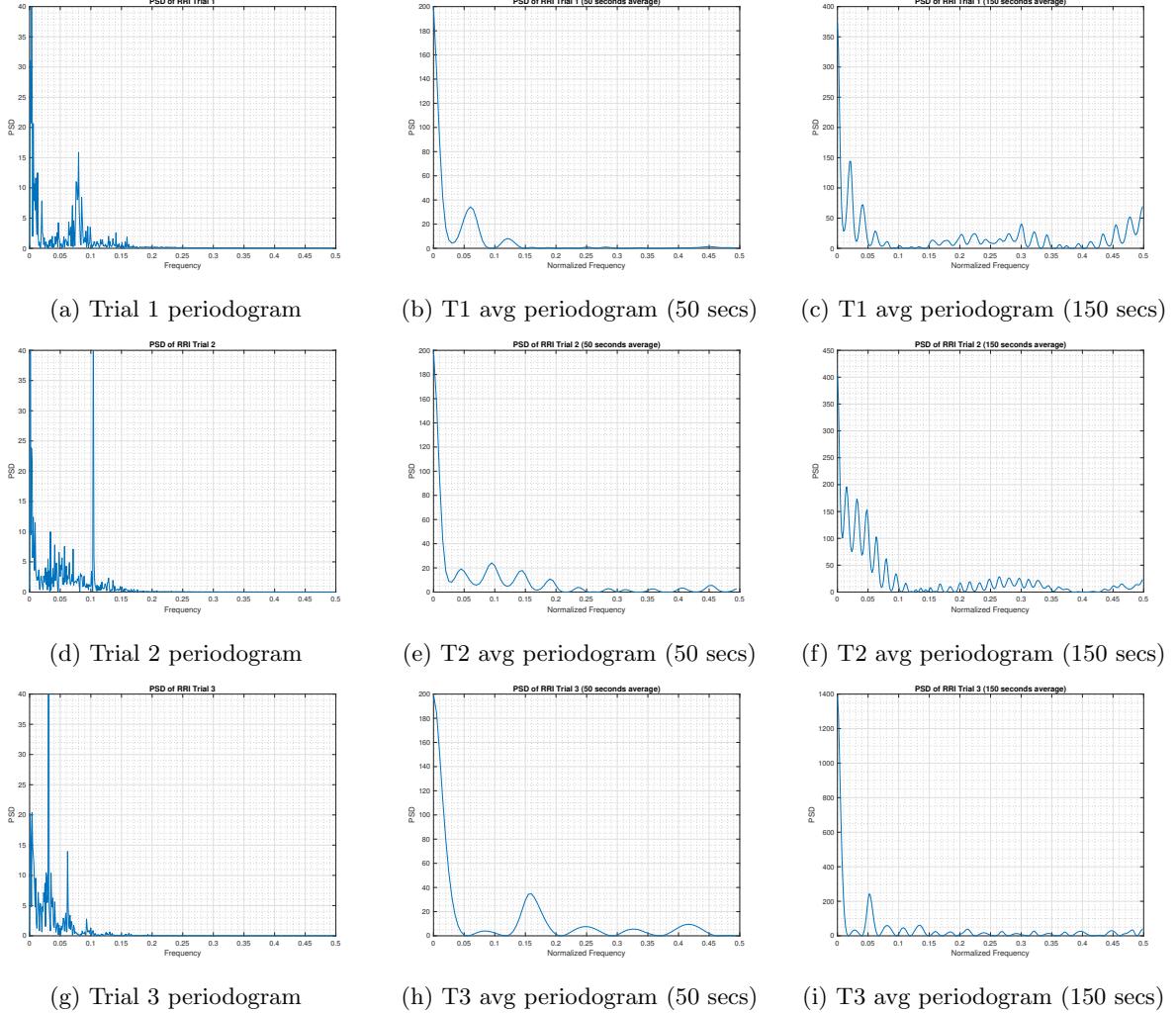


Figure 12: Comparing the PSD estimate for RRI Trials 1,2,3 from the standard periodogram, 50 second average periodogram and 150 second average periodogram

## 4 Optimal filtering - fixed and adaptive

### 4.1 Wiener Filter

#### 4.1.1 Optimal coefficients of the Wiener Filter

The given filter coefficients for the unknown system are  $\mathbf{b} = [1 \ 2 \ 3 \ 2 \ 1]$ . The ideal SNR using normalized data is:

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

We use  $R_{xx}$  and  $p_{zx}$  to find the optimal weight coefficients  $w_{opt} = R_{xx}^{-1} \cdot p_{zx}$ . The normalized optimal coefficients for noise with standard deviation of 0.1 are  $\mathbf{w} = [0.2168 \ 0.4371 \ 0.6575 \ 0.4392 \ 0.2163]$ . While normalizing the filter output ensures power matching between the input and output, it scales the data. Hence the coefficients do not resemble those of the given unknown system.

Without normalization, the coefficients are  $\mathbf{w} = [0.9990 \ 2.0011 \ 3.0020 \ 2.0064 \ 1.0013]$ , which resemble the given coefficients. The SNR in this case is 32.8895 dB. Increasing the sample size and/or decreasing the noise variance give a better estimate of the coefficients.

#### 4.1.2 Examining the effect of different noise powers

Since the variance needs to be in the range 0.1 - 10, the standard deviation lies in the range 0.01 - 3.16. Table 1 shows that both the accuracy of the estimate for the coefficients and the SNR decrease with increasing noise variance. If we increase the order beyond 5, we have more coefficients than that of the given unknown system, and the extra coefficients are effectively 0. Increasing the noise variance causes the estimate to deviate further from 0.

Noise Standard Deviation	b					
	w[1]	w[2]	w[3]	w[4]	w[5]	SNR (dB)
0.01	1.0002	2.0005	3.0049	2.0032	1.0024	52.7863
0.1	1.0041	2.0091	2.9992	1.9796	0.9824	32.4898
0.5	0.9804	1.9746	2.9597	2.0066	1.0096	18.2664
1.4	1.0425	1.9846	2.9582	2.0057	0.9108	9.1680
2.5	1.1666	2.1163	3.0370	1.8805	1.0565	4.3505
3.1	1.1971	2.0309	3.0611	1.9085	0.9377	2.4010

Table 1: The effect of varying noise standard deviation on the optimal coefficients

#### 4.1.3 Computational complexity of the Wiener solution

We need to find the ACF of  $\mathbf{x}$  and the cross-correlation of  $\mathbf{x}$  and  $\mathbf{z}$  to calculate  $R_{xx}$  and  $p_{zx}$ .  $N_w + 1$  correlations are needed to calculate each of these statistics. If we denote the complexity of calculating one correlation as  $O(N)$ , the complexity for  $N$  samples for each statistic is  $O(N(N_w + 1))$ .

This means that both the ACF and the cross correlation need  $N(N_w + 1)$  multiplications and  $N(N_w + 1)$  additions. According to the rules of Big O notation, the total for both statistics is  $O(N(N_w + 1)) + O(N(N_w + 1)) \approx O(N(N_w + 1))$ . Assuming  $N \gg N_w$ , the complexity is approximately  $O(NN_w)$ .

We know that inverting an  $N$  by  $N$  matrix has complexity  $O(N^3)$ . Therefore the inverse in  $w_{opt} = R_{xx}^{-1} \cdot p_{zx}$  will take  $O((N_w + 1)^3)$  operations to compute. Assuming  $N_w \gg 1$ , the complexity is approximately  $O(N_w^3)$ .

## 4.2 The least mean square (LMS) algorithm

### 4.2.1 The lms routine

The LMS algorithm recursively generates coefficients to give  $\hat{y}$ , and these coefficients are continuously updated on the basis of the error vector. Figure 1 illustrates how the estimate takes approximately 100 seconds to adapt to the target signal.

Although the Wiener solution assumes that  $\mathbf{z}$  is always stationary, real world signals are not always stationary. The LMS estimate analyzes the data in blocks instead of all at once which allows it to handle non stationary processes as well. This process, however, makes it less robust to noise and gives lower accuracy. In the given case,  $\mathbf{z}$  is stationary and the Wiener solution is more accurate. Using the same range limits of noise as we did earlier, we can see that the accuracy of the estimate degrades with increasing noise variance.

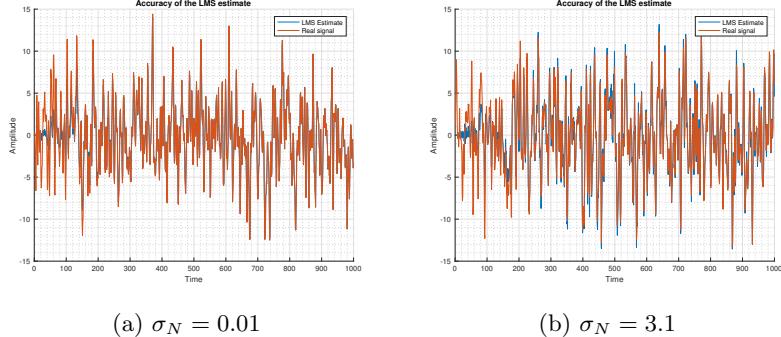


Figure 1: Comparing the input signal  $\mathbf{x}$  with the LMS estimate

### 4.2.2 Time evolution of coefficients

Figure 2a shows the time evolution of the estimated error squared for adaptive gain  $\mu = 0.01$ , which becomes almost negligible after the 250th sample. Figure 2c shows the time evolution of the optimal coefficients for  $\mu = 0.01$ . The rest of Figure 2 shows the time evolution of optimal coefficients for different values of  $\mu$ .

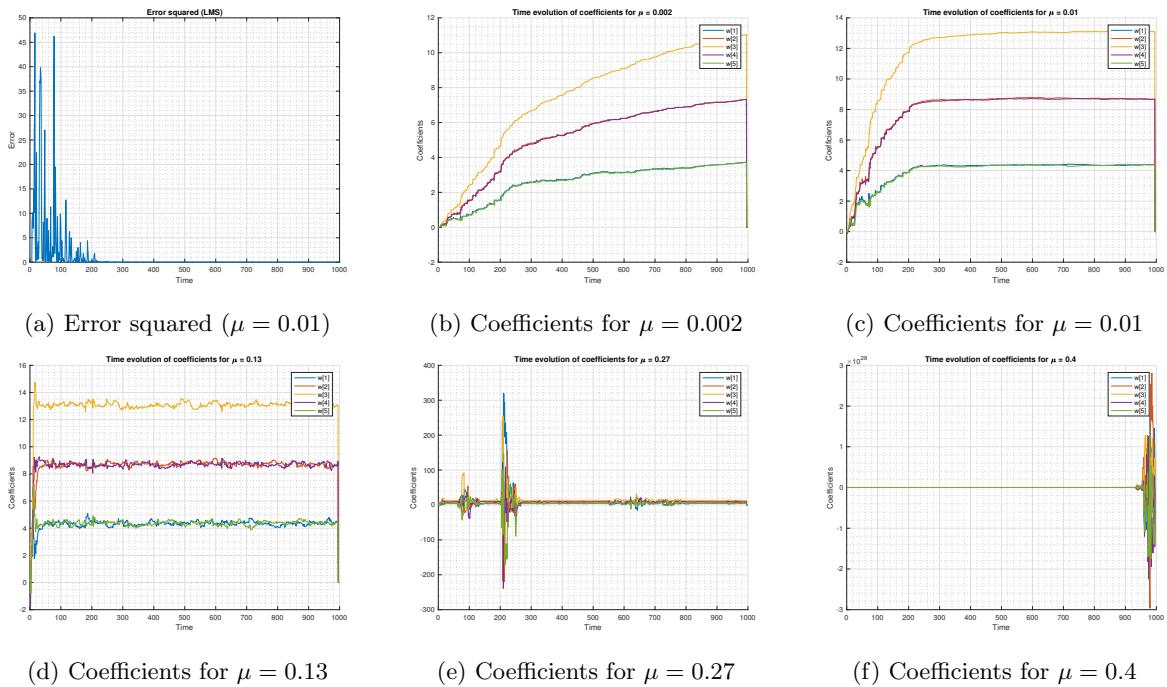


Figure 2: Comparing the time evolution of optimal coefficients for different values of adaptive gain  $\mu$

Increasing  $\mu$  causes the coefficients to converge to their ideal values faster, which in turn causes the error to converge to 0 more quickly. Reducing  $\mu$  delays this convergence (see Figure 2b). For low  $\mu$ , the gradual convergence of the coefficients to their ideal values would make the estimate smoothly converge to the ideal signal. However, increasing  $\mu$  beyond a certain point (see Figure 2d) causes the coefficients to diverge from the ideal. The error increases and the estimated coefficients (and hence the estimated signal) completely diverges from the ideal (see Figure 2e).

#### 4.2.3 Computational complexity of the LMS algorithm

The complexity can be determined by analyzing the 3 equations that make the LMS algorithm.  $w(n+1) = w(n) + \mu e[n]x(n)$  requires  $N_w + 1$  additions. This is denoted as  $O(N_w + 1)$ . The transpose in  $\hat{y}[n] = w^T(n)x(n)$  would be approximately  $O(N_w)$ . The product of the  $N_w$  by  $N_w + 1$  matrix with an  $N_w + 1$  by 1 column would be  $O(N_w + 1) + O(N_w)$ . Calculating the error is a simple subtraction of scalar values and hence does not influence the complexity. Assuming  $N_w \gg 1$ , the total complexity is of the order  $4O(N_w) \approx O(N_w)$ .

### 4.3 Gear Shifting

Gear shifting is a natural extension to the previous exercise.  $\mu$  is decreased by 20% if the new error is lower than the old error, and increased by 20% in the opposite case. This avoids divergence of the coefficients and hence overshoot of the estimated signal  $\hat{y}$ . The initial gain is kept small in order to avoid divergence. Figure 3 shows the faster convergence of coefficients and error for  $\mu = 0.1$  compared to the standard LMS algorithm.

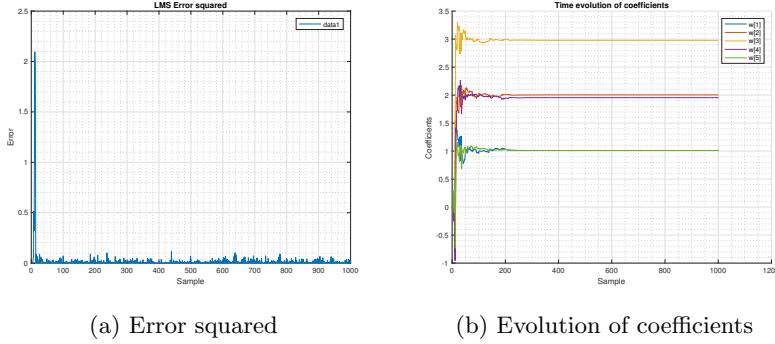


Figure 3: Faster convergence using gear shifted LMS algorithm for  $\mu = 0.1$

## 4.4 Identification of AR processes

### 4.4.1 Implementing the given AR model

Since the given system is an AR(2) process, the LMS algorithm needs to estimate  $a_1$  and  $a_2$ . Figure 4b shows that the coefficients converge to -0.2 and -0.9. This is because the filter output is defined by:

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

### 4.4.2 Evolution of AR coefficients for different gains

The plots in Figure 4 have been extended to 5000 samples so that the convergence is clear. The convergence is too slow for  $\mu = 0.001$  and the estimate is too distorted for  $\mu = 0.1$ . There is a good balance between accuracy and speed of convergence at  $\mu = 0.01$ .

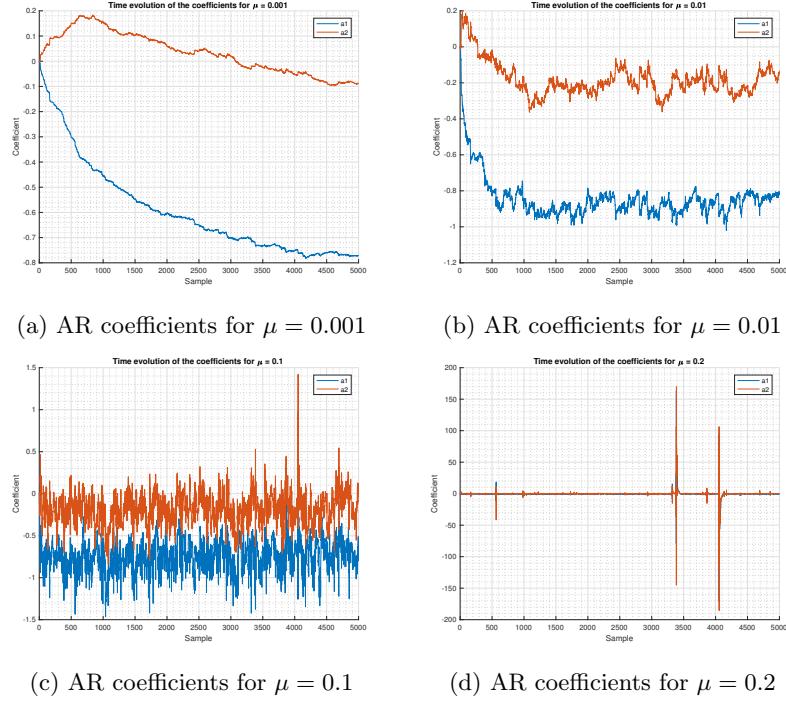


Figure 4: Evolution of AR coefficients  $a_1$  and  $a_2$  for different values of  $\mu$

## 4.5 Speech Recognition

### 4.5.1 Testing the predictor performance

The model proposed in the earlier section is now used to estimate the best prediction order for 5 pieces of audio data - "e", "a", "s", "t", and "x". We use the MDL and AIC to determine the orders, and the results are tabulated in Table 2.

Criterion	Recommended Order				
	e	a	s	t	x
MDL	4	7	5	4	5
AIC	5	23	6	4	8

Table 2: Recommended model order for each letter's audio signal

The calculations above use 1000 samples, a sampling frequency of 441 kHz, an adaption gain of 0.5, and a filter order of 20. The low  $\mu$  and high order have been chosen to ensure accuracy. Gear shifting is not recommended in this case since the speech signal is non-stationary, and the gear shifting algorithm would introduce errors in the estimates since it is strictly meant for stationary processes. Increasing the adaptation gain beyond 0.8 causes diversion in the estimates.

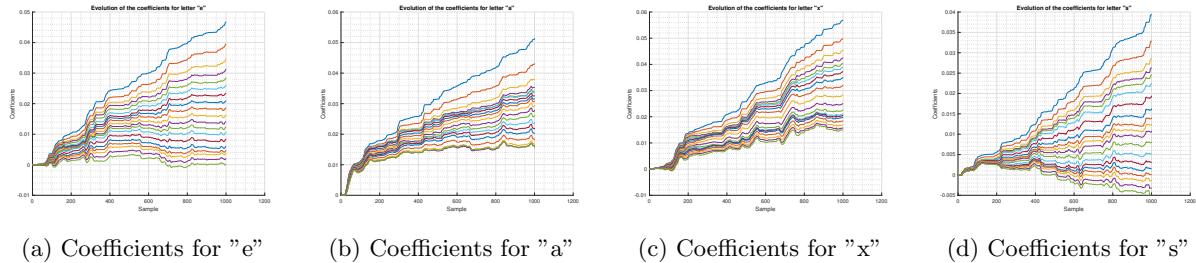


Figure 5: Convergence of coefficients for the audio signals

### 4.5.2 The optimal filter length

The optimal filter length could be found by looking for the combination of  $\mu$  and  $p$  that gives the maximum  $R_p$ . This must also be balanced with the constraint on computational complexity, since higher orders require more mathematical operations. Table 3 shows the variation in prediction gain for the audio signal "e" when adaptation gain and filter order are changed.

Every person's speech can be modeled uniquely depending on the specific frequency components in their voice.  $R_p$  peaks at  $p=5$  for  $\mu = 0.1$ , which implies that a 5th order system is optimal for modeling my voice. The estimated orders become distorted for  $\mu > 0.8$ , so the trends for  $\mu = 1, 2$  may be ignored for the purpose of this analysis.

	p=20	p=15	p=10	p=5	p=1
$\mu = 0.01$	25.9189	25.9321	25.9413	25.9484	25.9654
$\mu = 0.1$	25.9940	26.1359	26.2291	26.2565	26.2603
$\mu = 1$	27.6064	27.6103	27.5602	27.1570	26.0351
$\mu = 2$	28.4507	28.4430	28.4166	28.0057	26.0278

Table 3: Trends in prediction gain for "e" across filter order and adaptation gain

#### 4.5.3 Assessing the predictor performance at $F_s = 16$ kHz

Tables 4 and 5 illustrate the difference in performance for sampling at 44.1 kHz and 16 kHz. There is a higher prediction gain at 44.1 kHz and lower recommended model orders for "e", "s", and "x". This is indicative of better performance. However, the lower sampling rate causes oscillations in the recommended model orders.

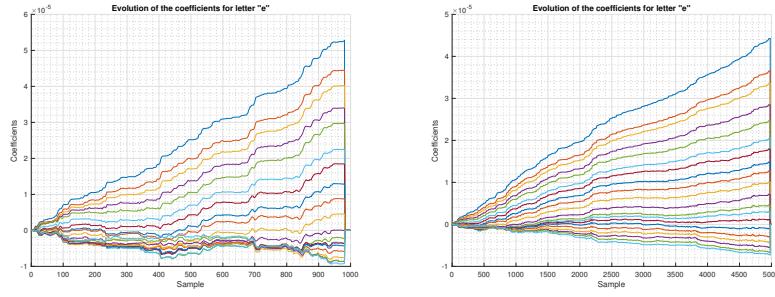
Higher sampling frequency reduces the quantization error since each sample covers a smaller time frame. For a lower sampling rate, the AR process would have to adapt to a wider range of coefficients since the audio signal is not stationary. Increasing the number of samples would compensate for these errors and would allow a lower value of adaptation gain for stable AR coefficients.

Sampling Frequency	e	a	s	t	x
44.1 kHz	25.841873	23.636714	25.282723	20.336425	20.182098
16 kHz	25.480873	24.956100	21.369334	21.895127	18.847481

Table 4: Comparing the prediction gain for sampling at 44.1 kHz and 16 kHz

Criterion	Recommended Order				
	e	a	s	t	x
MDL	8	4	4	21	4
AIC	22	4	21	21	4

Table 5: Recommended model order for each letter's audio signal using  $F_s = 16$  kHz



(a)  $F_s=16$  kHz,  $N = 1000$ ,  $\mu = 0.5$       (b)  $F_s=16$  kHz,  $N = 5000$ ,  $\mu = 0.1$

Figure 6: Improved convergence and stability for coefficients of "e" by increasing number of samples

## 4.6 Computational complexity of sign algorithms

Figure 7 shows that while the signed LMS algorithms converge to the necessary values (-0.2 and -0.9) faster than the standard LMS, they are much more oscillatory after convergence. Of the signed algorithms, the signed regressor is the slowest to converge but shows the least variance. The sign-sign algorithms shows the largest variance and converges slower than the signed error algorithm.

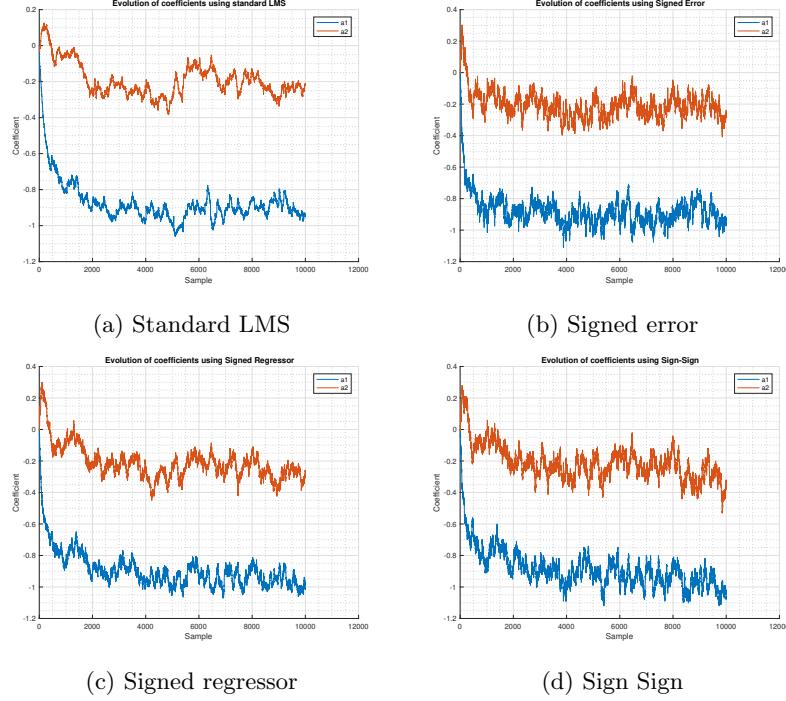


Figure 7: Comparing the performance of signed LMS algorithms for AR prediction

Figure 8 shows the performance of the signed algorithms on the audio for letter "a". Since the LMS algorithm cannot be applied to speech which is non-stationary, the model orders recommended would be inaccurate. Nevertheless, it is worth noting that the rate of convergence for the standard LMS is lower than that of the signed error and signed regressor. The LMS, signed error and signed regressor recommend identical model orders, while the sign-sign algorithm has skewed predictions.

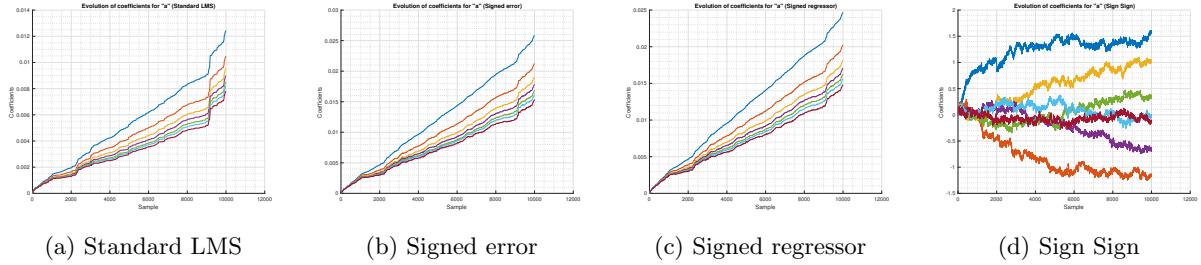


Figure 8: Comparing the performance of signed LMS algorithms for audio signal "a"