

---

Advanced Digital Signal Processing Coursework  
Mireia Muñoz Rojo  
CID: 01572447

---

## Table of contents

1.	Random Signals and Stochastic Processes .....	3
1.1	Statistical Estimation .....	3
1.2.	Stochastic Processes .....	7
1.3.	Estimation of probability distribution.....	11
2.	Linear Stochastic Modelling.....	12
2.1	ACF of uncorrelated and correlated Sequences.....	12
2.2.	Cross-Correlation Function .....	14
2.3.	Autoregressive Modelling .....	15
2.4.	CRLB.....	18
2.5.	Real world signals: ECG experiment .....	21
3.	Spectral estimation and modelling .....	23
3.1.	Average Periodogram estimates .....	24
3.2.	Spectrum of Autoregressive Processes .....	25
3.3.	Least Squares Estimation (LSE) of AR coefficients .....	27
3.4.	Spectrogram for time-frequency analysis: dial tone pad .....	30
3.5.	Real world signals: Respiratory sinus arrhythmia from RR-Intervals.....	32
4.	Optimal filtering - fixed and adaptive .....	32
4.1.	Weiner Filter.....	32
4.2.	The Least Mean Square (LMS) algorithm.....	33
4.3.	Gear Shifting.....	34
4.4.	Identification of an AR process .....	35
4.5.	Speech recognition .....	36
4.6.	Dealing with computational complexity: Sign Algorithms .....	38
5.	MLE for the Frequency of a Signal .....	40

# 1. Random Signals and Stochastic Processes

## 1.1 Statistical Estimation

By using the Matlab command `rand`, a 1000- sample vector  $\mathbf{x} = [x[1], x[2], \dots, x[1000]]^T$  has been generated and has been plotted in figure 1.1, where  $x[n]$  is a realisation of a uniform random variable  $X \sim U(0, 1)$ :

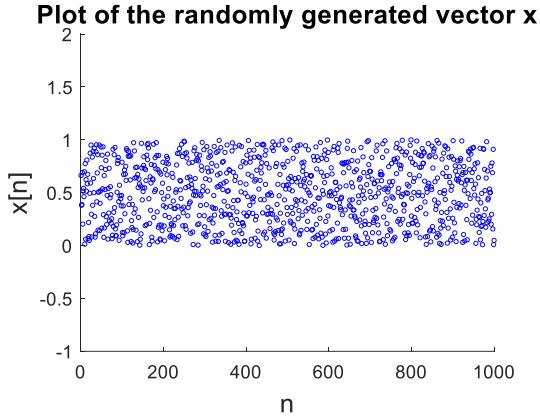


Figure 1.1: Plot of uniformly distributed 1000-sample random signal

### 1.1.1. Comparison between theoretical and sample mean.

The theoretical mean can be calculated using equation 1:

$$m_x = \mu_x = \mathbb{E}\{x\} = \int_{-\infty}^{\infty} x[n]p(x[n]) dx [n] \quad (1)$$

Since  $P(x[n]) = \frac{1}{b-a}$ , a and b being the lower and upper limits respectively:

$$m_x = \int_a^b \frac{z}{b-a} dz = \int_0^1 \frac{z}{1-0} dz = 0.5 \quad (2)$$

Therefore the theoretical mean has a value of 0.5. We can calculate the sample mean using the Matlab command `mean`, finding the value of  $\hat{m} = 0.4983$ . If we compare both values we can calculate the relative error by calculating the difference between the theoretical value and the estimated value and dividing this by the theoretical value. We obtain a relative error of 0.34%.

To make the estimator more accurate we desire a minimum variance. This can be obtained by increasing the number of samples considered, since as the number of samples tend to infinite the variance tends to zero.

$$\text{var}\{\hat{\theta}\} = \mathbb{E}\{[\hat{\theta} - \mathbb{E}\{\hat{\theta}\}]^2\} = \frac{1}{N^2} \sum_{n=0}^{N-1} \text{var}\{x[n]\} = \frac{\sigma^2}{N} \quad (3)$$

We can also demonstrate that the estimator is unbiased and therefore it is a good estimator:

$$\mathbb{E}\{\hat{m}\} = \mathbb{E}\left\{\frac{1}{N} \sum_{n=1}^N x[n]\right\} = \frac{1}{N} \sum_{n=1}^N \mathbb{E}\{x[n]\} = \frac{1}{N} Nm = m \quad (4)$$

### 1.1.2. Comparison between theoretical and sample standard deviation

In order to calculate the theoretical standard deviation, we use equation 5:

$$\sigma = \sqrt{\mathbb{E}\{x - \mathbb{E}\{x\}\}^2} = \sqrt{\mathbb{E}\{x - m\}^2} = \sqrt{\mathbb{E}\{x^2\} - m^2} \quad (5)$$

$$\begin{aligned}
&= \sqrt{\int_{-\infty}^{\infty} x[n]^2 p(x[n]) dx [n] - m^2} = \sqrt{\int_a^b \frac{z^2}{b-a} dz - m^2} = \\
&\sqrt{\int_0^1 \frac{z^2}{1-0} dz - (0.5)^2} = \sqrt{\left[\frac{z^3}{3}\right]_0^1 - (0.5)^2} = 0.28868
\end{aligned} \tag{6}$$

Therefore, the theoretical value of the standard deviation is  $\sigma = 0.28868$ . We can calculate the sample standard deviation by using the MATLAB command `std`, and we obtain  $\hat{\sigma} = 0.2830$ .

We do similar calculations as we did to determine the error in the estimate of the mean to obtain an error of 1.97% for the estimate of the standard derivation. Furthermore, an increase in the number of samples would decrease the variance of the estimator.

We can also derive the fact that this estimator is also unbiased and therefore, it is also a good estimator:

$$\begin{aligned}
E\{\hat{\sigma}\} &= E\left\{\sqrt{\frac{1}{N-1} \sum_{n=1}^N (x[n] - \hat{m})^2}\right\} = \sqrt{\frac{1}{N-1} E\left\{\sum_{n=1}^N x^2[n] - 2\hat{m} \sum_{n=1}^N x[n] + \sum_{n=1}^N \hat{m}^2\right\}} \\
&= \sqrt{\frac{1}{N-1} \left( N(\sigma^2 + \hat{m}^2) - N \left( \frac{\sigma^2}{N} + \hat{m}^2 \right) \right)} = \sqrt{\frac{1}{N-1} (N-1)\sigma^2} = \sigma
\end{aligned} \tag{7}$$

### 1.1.3. Bias of the mean and the standard deviation

An ensemble of ten 1000-sample realisations of  $X$  is generated, and the standard deviation and the mean for each sample realization are plotted in Figures 1.2 and 1.3 respectively.

The bias of an estimator is defined by:  $B = E\{\hat{\theta}_N\} - \theta$ . As we can see in figures 1.2 and 1.3, both estimates tend towards the theoretical values previously calculated. We can therefore deduce that the estimators are unbiased since the estimated values can be approximated to the theoretical value as the number of samples tends to infinity. We also know this due to the calculations derived in exercise 1.2 and 1.3.

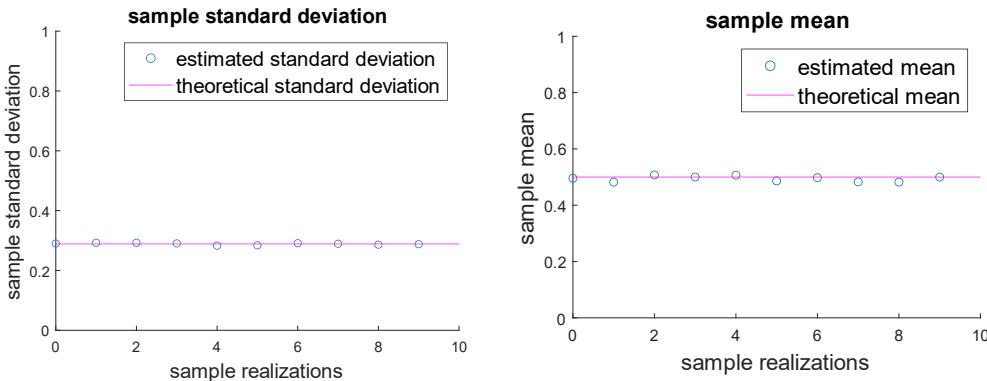


Figure 1.2: Plot showing the sample standard deviation for all the 10 different realizations of  $x$ . The straight line represents the theoretical value of the standard deviation.

Figure 1.3: Plot showing the sample mean for all the 10 different realizations of  $x$ . The straight line represents the theoretical value of the mean.

### 1.1.4. Probability density function

We compute the histogram using the Matlab command `hist` for different sample numbers in order to see the effect their increase cause. The histogram is normalized in order to make its area equal to 1. The theoretical probability density function (PDF) of  $x$  is defined by equation (8):

$$P(x[n]) = \frac{1}{b-a} \quad (8)$$

Where  $b = 1$  and  $a = 0$ , since the values range from 0 to 1.

From figure 1.4 we can deduce that an increase in the number of samples used makes the normalized histogram more accurate with respect to the PDF.

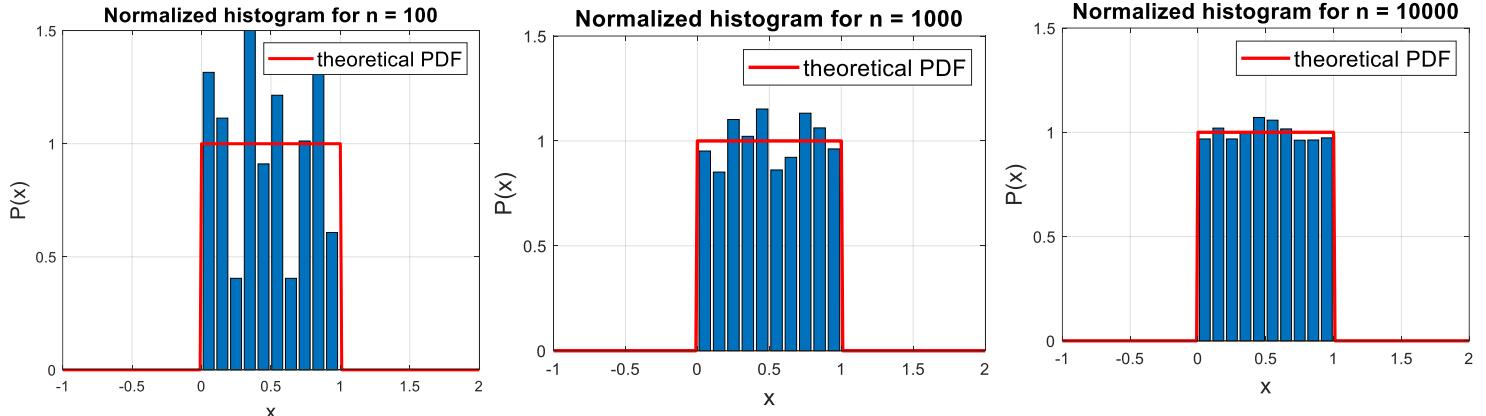


Figure 1.4: Plots of the normalized histogram for increasing number of samples ( $n$ ). The theoretical probability density function of the random process is shown in red. All the histograms have been plotted with a number of bins of 10.

If we increase the number of bins keeping the number of samples constant ( $n = 10000$ ), like in figure 1.5, we can appreciate that as they increase, the histogram resolution also increases. However, if they increase too much the histogram gets clustered.

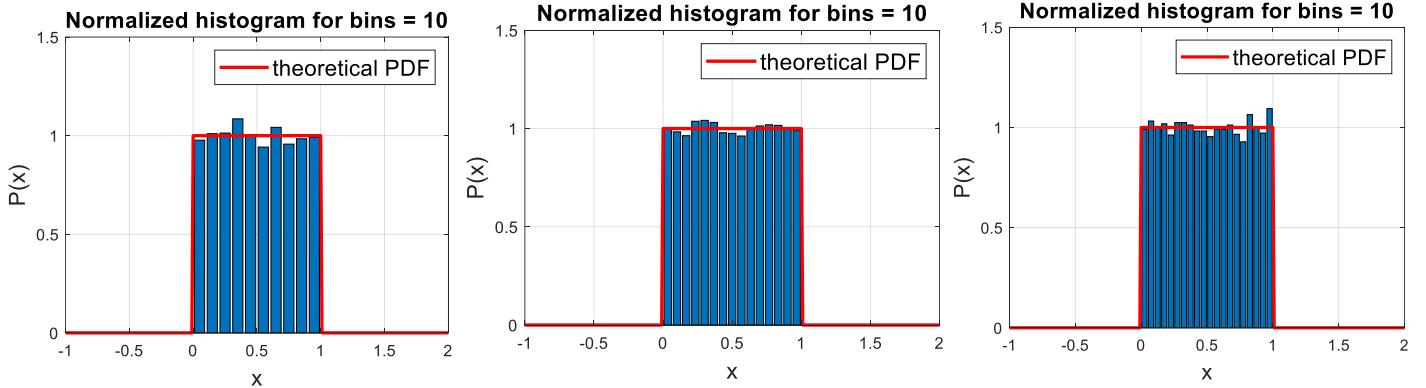


Figure 1.5: Plots of the normalized histograms with increasing numbers of bins. The theoretical probability density function can be seen in red. All the histograms were plotted for a random process of 1000 samples.

### 1.1.5. Zero-mean, unit standard deviation, Gaussian random variables

By using the Matlab command `rand`, a 1000- sample of Gaussian random variables is generated and plotted in figure 1.6.

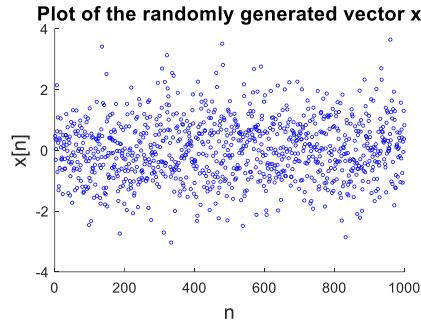


Figure 1.6: 1000-sample Gaussian Signal

Moreover, we will calculate the theoretical mean and standard deviation the same way we calculated them in Exercise 1.1 and 1.2.

#### Theoretical mean and standard deviation

The Matlab command `randn` returns a vector of normally distributed numbers therefore, the theoretical values of the mean and standard deviation of a normal distribution are  $m = 0$  and  $\sigma = 1$ . The sample mean and standard deviation were computed the same way they were in exercise 1.1 and 1.2 and they have values of  $\hat{m} = 0.0318$  and  $\hat{\sigma} = 1.0031$ .

#### Bias of the sample mean and sample standard deviation.

As we can see in the figures 1.7 and 1.8, both estimates tend towards the theoretical values. We can therefore deduce that the estimators are unbiased since the estimated values can be approximated to the theoretical value.

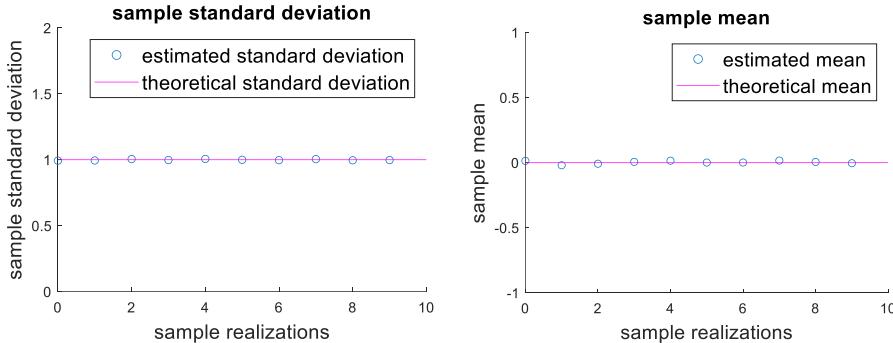


Figure 1.7: Plot showing the sample standard deviation for all the 10 different realizations of  $x$ . The straight line represents the theoretical value of the standard deviation.

Figure 1.8: Plot showing the sample mean for all the 10 different realizations of  $x$ . The straight line represents the theoretical value of the standard deviation.

#### Probability density function

The probability density function of a gaussian distribution ( $\sigma_x = 1$  and  $\mu_x = 0$ ) can be calculated using equation 9:

$$P(x[n]) = \frac{1}{\sqrt{2\pi\sigma_x^2}} e^{-\frac{(x[n]-\mu_x)^2}{2\sigma_x^2}} = \frac{1}{\sqrt{2\pi}} e^{-\frac{(x[n])^2}{2}} \quad (9)$$

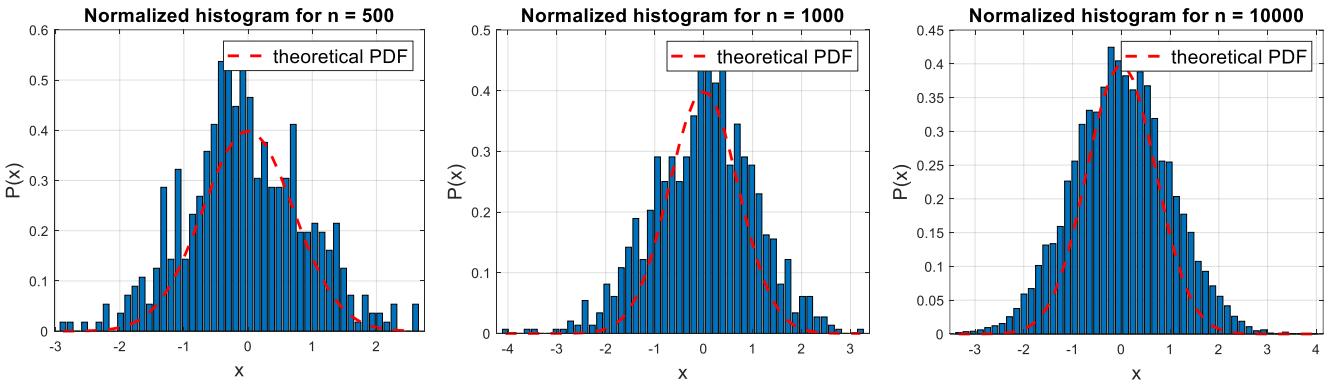


Figure 1.9: Plots of the normalized histogram for increasing number of samples ( $n$ ). The theoretical probability density function of the random process is shown in red. All the histograms have been plotted with a number of bins of 50.

Observing figure 1.9 we can conclude that the increase of number of samples in a normal distribution also increases the accuracy of the histogram. This is because the shape of the histogram matches more accurately the theoretical PDF for larger number of samples.

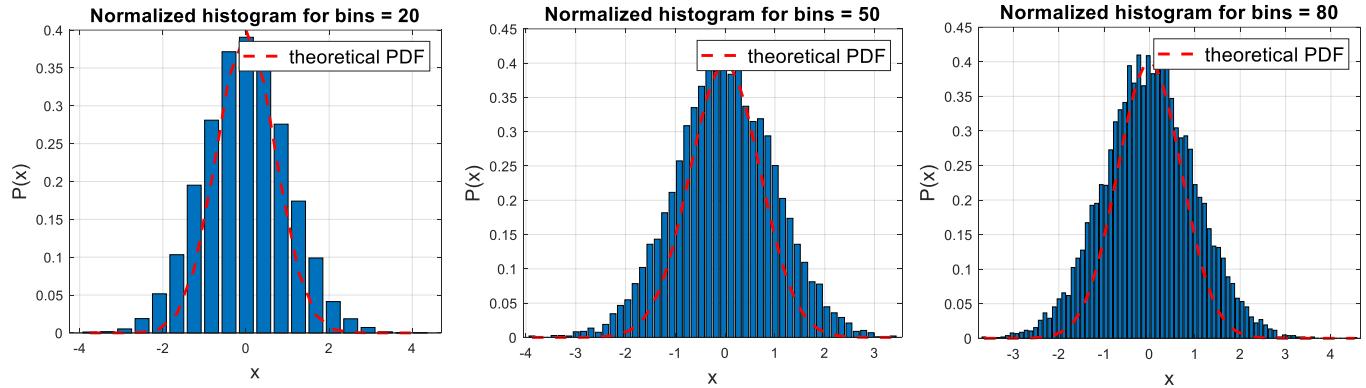


Figure 1.10: Plots of the normalized histograms with increasing numbers of bins. The theoretical probability density function can be seen in red. All the histograms were plotted for a random process of 1000 samples.

As the number of bins increases the histogram converges more accurately towards the theoretical value of the PDF.

## 1.2. Stochastic Processes

### 1.2.1. Ensemble mean and standard deviation of different random processes.

A stochastic process is stationary if its probability distribution is invariant under time shifts and if its other statistical properties such as the mean do not depend on time shifts.

#### *Random Process 1*

The ensemble mean and standard deviation are plotted in figure 1.11 with their respective theoretical value. From figure 1.11, we can see that for the first random process the ensemble mean varies over time as well as the standard deviation, indicating that the process is non-stationary.

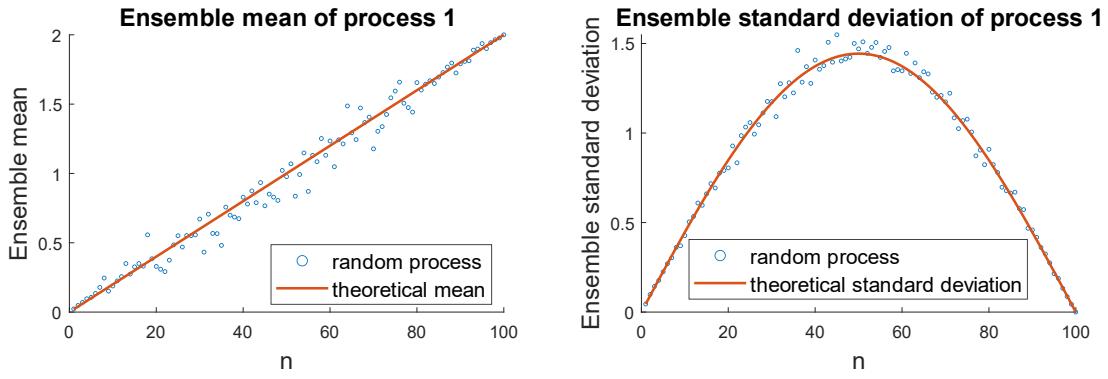


Figure 1.11: Plots of the ensemble mean and standard deviation of the first process compared to their calculated theoretical values.

#### Random process 2

For the second random process, the ensemble mean is time invariant since it converges into a fixed value, therefore we can conclude that the process is stationary.

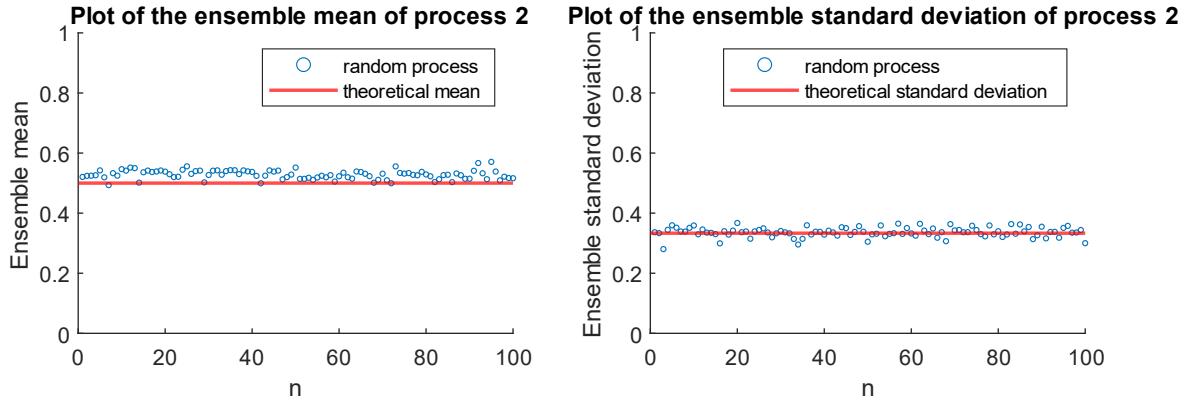


Figure 1.12: Plots of the ensemble mean and standard deviation of the second process compared to their calculated theoretical values.

#### Random process 3

For the third random process, the ensemble mean also converges to a fix value and therefore it is time invariant. Due to this property, we can conclude that the process is stationary.

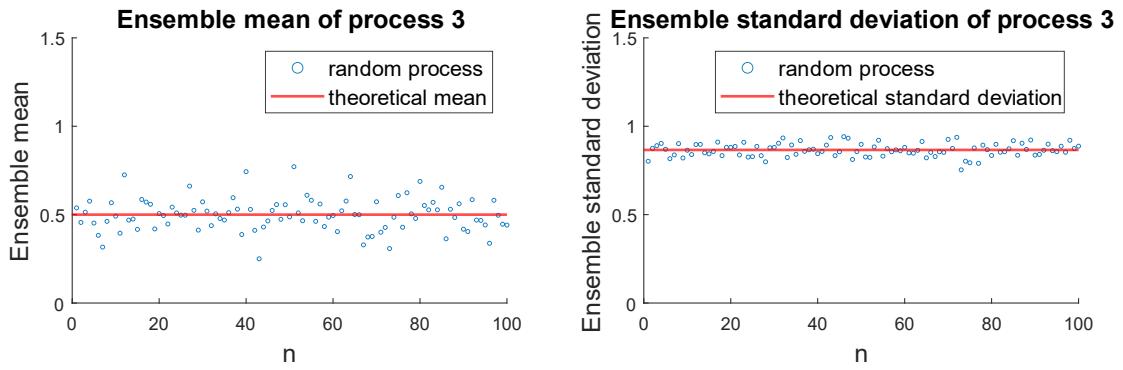


Figure 1.13: Plots of the ensemble mean and standard deviation of the third process compared to their calculated theoretical values.

#### 1.2.2. Mean and standard deviation of 4 realisations for each process

The difference between the time average mean and the ensemble average is that the time average is calculated by averaging in time, however the ensemble mean is calculated by averaging various realizations of the process. A stochastic process is ergodic if the average statistical properties of the process can be represented by a limited number of random samples. Therefore, if ergodic, its statistical expectation can be computed

using time averages on a single realization. This means that for an ergodic process, the time averages must match the ensemble average.

#### *Random process 1*

The time averages for the mean and the standard deviation were computed and reported in table 1 for each realization. When comparing the time averages of the mean with figure 1.11 and the values obtained there, we conclude that the process is not ergodic.

Realisation Number	Mean	Standard deviation
1	9.9972	5.8568
2	9.9995	5.8508
3	10.0287	5.8589
4	9.9861	5.8910

Table 1: time average mean and standard deviation measurements for the first random process of each realization number.

#### *Random process 2*

For the second random process, the values of the time average of the mean range from 0.3 to 0.7. This does not correspond to the values we observed in the ensemble mean in figure 1.12, this indicates that the process is non-ergodic.

Realisation number	mean	Standard deviation
1	0.3057	0.1503
2	0.3241	0.0077
3	0.7247	0.0342
4	0.4974	0.1556

Table 2: time average mean and standard deviation measurements for the second random process of each realization number.

#### *Random process 3*

For the last process, the time average values converge approximately to a value of 0.5 and so does the values of its ensemble mean as can be seen in figure 1.13. This means that the process is ergodic.

Realisation number	mean	Standard deviation
1	0.4581	0.8736
2	0.5136	0.8901
3	0.5324	0.8632
4	0.4684	0.8727

Table 3: time average mean and standard deviation measurements for the third random process of each realization number.

### 1.2.3 Theoretical mean and standard deviation for each process

All the theoretical values can be seen in figures 1.11, 1.12 and 1.13 in order to compare them with the estimated values.

#### *Random process 1:*

##### Theoretical mean calculation

The theoretical mean is calculated using equation (10):

$$m_x = \mu_x = \mathbb{E}\{x\} \quad (10)$$

And the first random process is defined as:

$$rp1_m[n] = 5 \sin\left(\frac{n\pi}{N}\right) x_m[n] + 0.02n \quad (11)$$

Where m represents the realization number and n is the discrete time instant. Therefore, the theoretical mean can be calculated by using equation 10 and equation 11:

$$\mu_x = \mathbb{E}\{r_{Pm}[n]\} = \mathbb{E}\left\{5 \sin\left(\frac{n\pi}{N}\right)x_m[n] + 0.02n\right\}$$

Since the only random part is  $x_m[n]$  and  $\mathbb{E}\{x_m[n]\} = 0$ , we can do the following:

$$5 \sin\left(\frac{n\pi}{N}\right)\mathbb{E}\{x_m[n]\} + 0.02n = 0.02n \quad (12)$$

#### Theoretical standard deviation calculation

In order to calculate its theoretical value we use equation 13:

$$\begin{aligned} \sigma &= \sqrt{\mathbb{E}\{x - E\{x\}\}^2} = \sqrt{\mathbb{E}\{x - m\}^2} = \sqrt{\mathbb{E}\{x^2\} - m^2} \\ \sigma &= \sqrt{\mathbb{E}\{(rp1_m[n])^2\} - \mathbb{E}^2\{rp1_m[n]\}} = \sqrt{\mathbb{E}\left\{\left(5 \sin\left(\frac{n\pi}{N}\right)x_m[n] + 0.02n\right)^2\right\} - (0.02n)^2} = \\ &\sqrt{\mathbb{E}\left\{25 \sin^2\left(\frac{n\pi}{N}\right)x_m^2[n] + 0.02n5 \sin\left(\frac{n\pi}{N}\right)x_m[n] + (0.02n)^2\right\} - 0.0004n^2} \end{aligned} \quad (13)$$

Since  $E\{x_m[n]\} = 0$  we can therefore conclude that:

$$\sigma = \sqrt{25 \sin^2\left(\frac{n\pi}{N}\right)E\{x_m^2[n]\}} \quad (14)$$

To calculate  $\mathbb{E}\{x_m^2[n]\}$  we use equation 15:

$$\mathbb{E}\{x_m^2[n]\} = \int_{-\infty}^{\infty} x_m^2[n]P(x_m[n])dx_m[n] \quad (15)$$

We calculate  $P(x_m[n])$  in the same way we did for exercise 1.1.2, using equation 9, and since the sine wave is multiplied by a random process that has values in the range -0.5 to 0.5, we obtain:

$$\int_{-0.5}^{0.5} x^2 dx = \left| \frac{x^3}{3} \right|_{-0.5}^{0.5} = \frac{1}{12}$$

Therefore:

$$\sigma = \frac{5}{\sqrt{12}} \sin\left(\frac{n\pi}{N}\right) \quad (16)$$

#### Random Process 2:

The second random process is defined by the following equation:

$$rp2_m[n] = x_m[n]Mr_m[M] + Ar_m[A] \quad (17)$$

#### Theoretical mean calculation:

We calculate the theoretical value of the mean the same way used for the first random process:

$$\mu = \mathbb{E}\{rp2_m[n]\} = \mathbb{E}\{x_m[n]Mr_m[M] + Ar_m[A]\} = \mathbb{E}\{x_m[n]\}\mathbb{E}\{Mr_m[M]\} + \mathbb{E}\{Ar_m[A]\} \quad (18)$$

We use equation 1 to obtain:

$$\int_{-0.5}^{0.5} x_m[n] dx_m[n] \int_0^1 M y_m[M] dM y_m[M] + \int_0^1 A y_m[A] dA y_m[A] = \left| \frac{x^2}{2} \right|_{-0.5}^{0.5} \left| \frac{y^2}{2} \right|_0^1 + \left| \frac{x^2}{2} \right|_0^1 = 0.5 \quad (19)$$

#### Theoretical standard deviation:

We calculate the theoretical value of the standard deviation the same way used for the first random process:

$$\begin{aligned}\sigma &= \sqrt{\mathbb{E}\{(rp2_m[n])^2\} - \mathbb{E}^2\{rp2_m[n]\}} = \sqrt{\mathbb{E}\{x_m^2[n]Mr_m^2[M]\} + 2\mathbb{E}\{x_m[n]Mr_m[M]Ar_m[A]\} + \mathbb{E}\{Ar_m^2[A]\} - \mu^2} = \\ &= \sqrt{\left|\frac{x^3}{3}\right|_{-0.5}^{0.5} \left|\frac{x^3}{3}\right|_0^1 + \left|\frac{x^3}{3}\right|_0^1 + 0.5^2} = \sqrt{\frac{1}{9}} = 0.333\end{aligned}\quad (20)$$

*Random process 3:*

The third random process is defined the following way:

$$rp3_m[n] = 3x_m[n] - 1 \quad (21)$$

Theoretical mean calculation

$$\mu = \mathbb{E}\{rp3_m[n]\} = \mathbb{E}\{3x_m[n] - 1\} = 3\mathbb{E}\{x_m[n]\} - 1 = 3 \int_0^1 x_m[n] dx m[n] - 1 = 3 \left|\frac{x^2}{2}\right|_0^1 - 1 = 0.5 \quad (22)$$

Theoretical standard deviation calculation

$$\begin{aligned}\sigma &= \sqrt{\mathbb{E}\{(rp3_m[n])^2\} - \mathbb{E}^2\{rp3_m[n]\}} = \sqrt{\mathbb{E}\{(3x_m[n] - 1)^2\} - \mu^2} = \sqrt{\mathbb{E}\{qx_m^2[n] - 6x_m[n] + 1\} - (0.5)^2} \\ &= \sqrt{9 \left|\frac{x^3}{3}\right|_0^1 - 6(0.5) - 0.25 + 1} = \sqrt{9 \frac{1}{3} - 3 + 1 - 0.25} = 0.866\end{aligned}\quad (23)$$

### 1.3. Estimation of probability distribution

#### 1.3.1 Estimate of PDF for normal distribution

The PDF estimate in figure 1.14 has 10,000 samples and 50 bins. Since it is a normal distribution its standard deviation is 1 and its mean is 0.

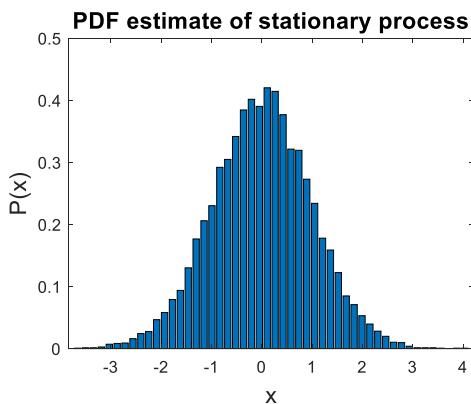


Figure 1.14: Plot of the estimated probability density function of a normal distribution

#### 1.3.2 Estimate of PDF for process number 3

The only stationary and ergodic process was process number 3. We plotted the estimated probability density function using 30 bins and a value of M = 100 for different values of N (number of samples considered). We can calculate the theoretical pdf to be 0.333 by using the formula previously used in exercise 1.1.4.

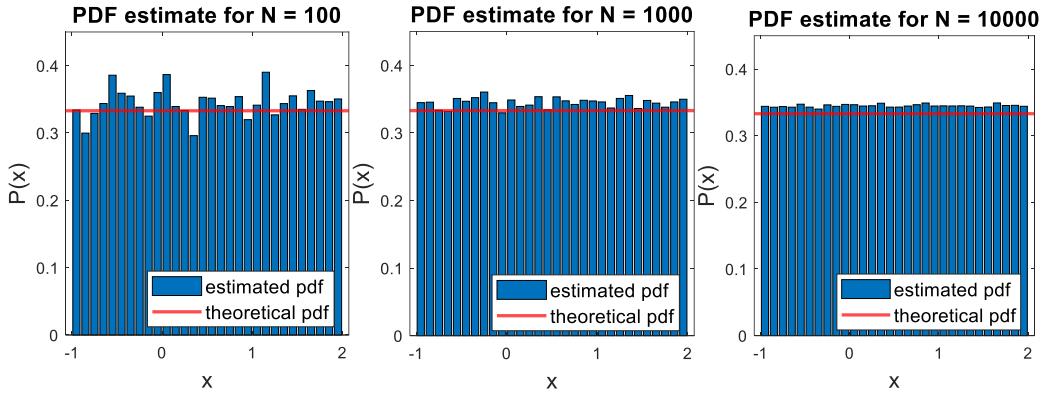


Figure 1.15: Plots of the estimated pdf with increasing number of samples  $N$ . The theoretical pdf is also plotted in red.

By observing figure 1.15, we can notice that as the number of samples  $N$  increase, the estimated PDF approximates its values more accurately to its theoretical value previously calculated as the difference between them decreases.

### 1.3.3 Non-stationary PDF estimation

We cannot estimate the PDF of a non-stationary process using the same function we used to estimate the PDF of a stationary process since, when we normalize the histogram, we are assuming time invariance.

Now we will consider a case where the mean of a 1000-sample-long signal varies from 0 to 1, after sample point  $N = 500$ . Due to the fact that the mean is not time invariant, since its value changes depending on time, we can not assume stationarity in order to compute the PDF. An alternative way we could do this is by separating the distribution into two uniform distributions (from  $N = 0$  to  $N = 500$  and from  $N = 500$  to  $N = 1000$ ) (figure 1.16).

Therefore, we could compute the theoretical PDF of each section separately. Both results yield to the same

value, this is therefore the correct theoretical PDF for the complete process. We can prove this using the following equation 8. For the first distribution, the theoretical PDF would be:

$$P(x[n]) = \frac{1}{b-a} = \frac{1}{0.5+0.5} = 1 \quad (24)$$

Similarly, the second's distribution PDF would be:

$$P(x[n]) = \frac{1}{b-a} = \frac{1}{1.5-0.5} = 1 \quad (25)$$

## 2. Linear Stochastic Modelling

### 2.1 ACF of uncorrelated and correlated Sequences

The autocorrelation function is the cross-correlation of a signal with itself and it represents the correlation between samples separated by a time lag, therefore it determines how correlated past samples are to future samples. Hence, it is used for finding repeating patterns in the data.

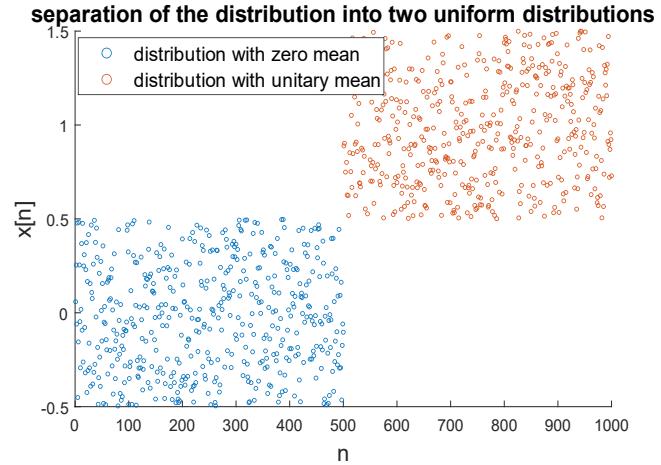


Figure 1.16: Plot of the separation of the original distribution into two uniform distributions. The first one has a mean in zero and the second one has a mean in one.

### 2.1.1. Unbiased estimate of ACF

Using the Matlab command `xcorr` we plot the unbiased estimate of the ACF of a 1000-sample realisation of white gaussian noise (WGN) generated using the Matlab command `randn` (figure 2.1).

As expected, the ACF shown in figure 2.1, has a spike at  $\tau = 0$  with a unitary amplitude, this is because the ideal ACF of white noise is represented by a single delta function.

This spike is the centre of symmetry of the function. It is symmetric due to the ACF property:  $R(\tau) = R(-\tau)$ . As mentioned before, the ACF is the cross-correlation of a signal with itself causing it to be symmetric.

If we use the Matlab command `zoom`, we can focus on the spike only (figure 2.2). In this figure the ideal spike can be seen better showing a stronger correlation with less noise, since the noise in the ACF increases as the time lag increases. The unbiased estimate of the ACF is given by:

$$\hat{R}_x(\tau) = \frac{1}{N-|\tau|} \sum_{n=0}^{N-|\tau|-1} x[n]x[n+\tau] \quad (26)$$

As the time lag increases, the number of samples decreases, increasing the noise and variance. Hence, the lags that correspond to the last two thirds of the plot shouldn't be considered since they are not good estimates and not reliable. The key features of the ACF can be seen for values of  $|\tau| < 50$ , therefore we could consider that as an empirical bound.

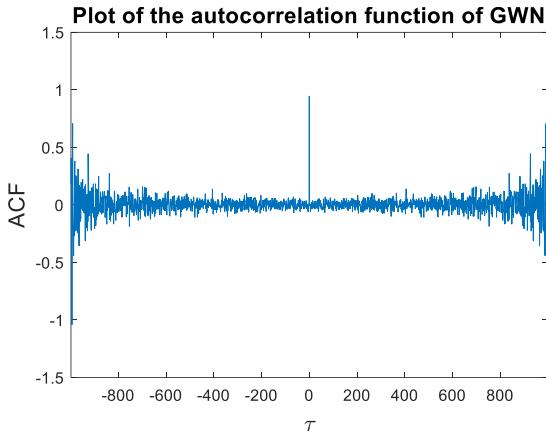


Figure 2.1: Plot of the ACF of a 1000-sample realisation of WGN.

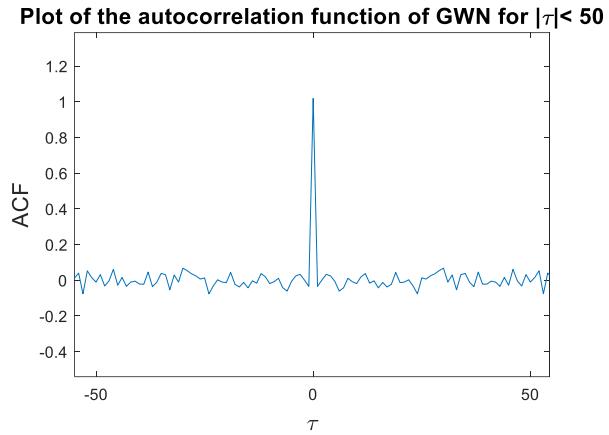


Figure 2.2: Zoomed in plot of the ACF for values of  $|\tau| < 50$ .

### 2.1.2 ACF of a WGN signal filtered by a Moving Average filter

A 1000-sample realisation of WGN is filtered by a Moving Average filter (MA) with coefficients of order 9 and its ACF is plotted in figure 2.3. A MA filter with coefficients of order 9 creates the average between the previous 9 samples and the current sample for each sample of the process, this is described by equation 27. By doing this, it smoothes the original signal.

$$y[n] = \frac{1}{N} \sum_{k=0}^{N-1} x[n-k] \quad (27)$$

Figures 2.3 and 2.4 show the estimated ACF of a WGN signal filtered by a MA filter of order 9 and 4 respectively.

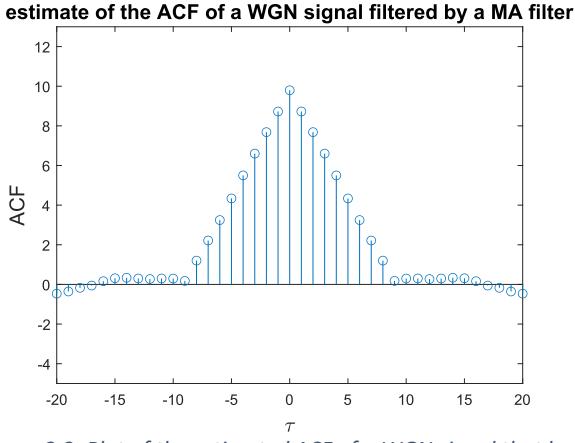


Figure 2.3: Plot of the estimated ACF of a WGN signal that has been filtered by a MA filter with coefficients of order 9.

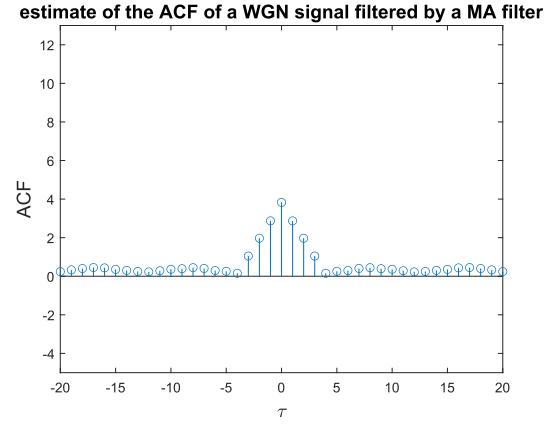


Figure 2.4: Plot of the estimated ACF of a WGN signal that has been filtered by a MA filter with coefficients of order 4.

As expected, the ACF has a symmetric shape described by a triangular wave in the centre. This characteristic shape is due to the fact that the correlation decreases as  $\tau$  increases. The width of this triangular shape will be defined by the order of the coefficients of the filter since it will double the width of the filter itself. Comparing figure 2.3 and 2.4 we can see that for lower orders the triangular function is smaller than for higher orders.

For the case considered, since the coefficients are of order 9, the correlation will be maximal when  $\tau = 0$  since there will be no distance between the samples. However, for  $|\tau| = 8$  the correlation will reach a minimum value before becoming zero for  $|\tau| > 8$ , this is the ideal case. In reality, when  $\tau > 8$  the ACF doesn't stay at a constant value of 0 due to the noise introduced by the limited number of samples available.

### 2.1.3 Uncorrelated Processes

If the ACF of the uncorrelated process is given by  $R_x(\tau)$  and the ACF of the impulse response of the filter is represented by  $R_h(\tau)$  making  $R_y(\tau)$  the output that represents the ACF of a filtered stochastic process:

$$R_y(\tau) = R_x(\tau) * R_h(\tau) \quad (28)$$

Since the ACF of WGN signal is denoted by the discrete Dirac function we can therefore say that:  $R_x(\tau) = \delta(\tau)$ . Moreover, the correlation of a signal with the delta function is equal to the signal itself meaning that  $R_y(\tau) = R_h(\tau)$  and therefore it is equal to the ACF of the impulse response of the filter.

## 2.2. Cross-Correlation Function

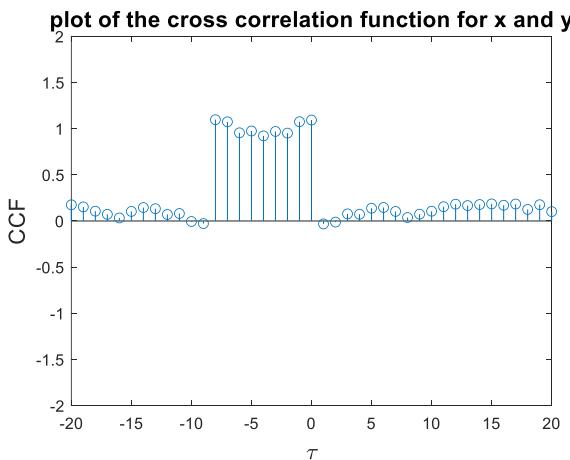


Figure 2.5: cross-correlation between a WGN signal and this signal filtered by a MA filter.

The cross-correlation between two signals represents how similar they are to each other with respect to the time lag and for the input and output of a filter it is given by:

$$R_{xy}(n, s) = \mathbb{E}\{X_n Y_s\} \quad (29)$$

However, this assumes an infinitely long signal, therefore, for ergodic data, the estimated cross-correlation function is given by:

$$R_{xy}(\tau) = \frac{1}{N-|\tau|} \sum_{n=0}^{N-|\tau|-1} x[n]y[n+\tau] \quad (30)$$

This shows that for the estimated cross-correlation each sample is related to the  $N$  previous number of samples. From figure 2.5 we can therefore conclude that there is a positive cross-correlation between the two signals between  $\tau = -8$  and  $\tau = 0$ .

As we have seen before, the ACF of WGN is given by a Dirac delta function, therefore the shape of the cross-correlation function for an uncorrelated stochastic process will be determined by the impulse response of the filter:

$$R_{xy}(\tau) = h(\tau) * R_x(\tau) = h(\tau) * \delta(\tau) = h(\tau) = \sum_{i=0}^N \delta(\tau - i) \quad (31)$$

This is confirmed by the plot in figure 2.5 since a square wave shape can be appreciated. This wave is not perfectly shaped due to the noise caused by the nonidealities of the WGN that have been assumed.

These qualities can be used for system identification since an uncorrelated process has an ACF defined by the Dirac delta function, therefore, when processed by an LTI system gives the impulse response of the system allowing its identification. Moreover, the number of spikes shown in the cross-correlation define the order of the filter coefficients.

## 2.3. Autoregressive Modelling

### 2.3.1. Stability based on the AR coefficients.

An Autoregressive (AR) model of order two is described as:

$$x[n] = a_1 x[n-1] + a_2 x[n-2] + w[n] \quad (32)$$

Where  $a_1$  and  $a_2$  are the model's coefficients. To study the stability of this model, the values of the coefficients have to produce a transfer function that doesn't have any poles outside the unitary circle and therefore doesn't diverge. The coefficient values must follow 3 stability conditions in order to meet these criteria. In figure 2.6 and 2.7 we can see the values of the coefficients ( $a_1, a_2$ ) for which the above condition is met.

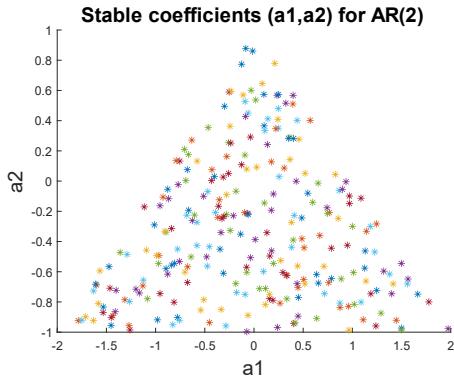


Figure 2.6: Pairs of coefficients  $(a_1, a_2)$  for which  $x[n]$  converges. The plot shows 1000 samples of  $(a_1, a_2)$ .

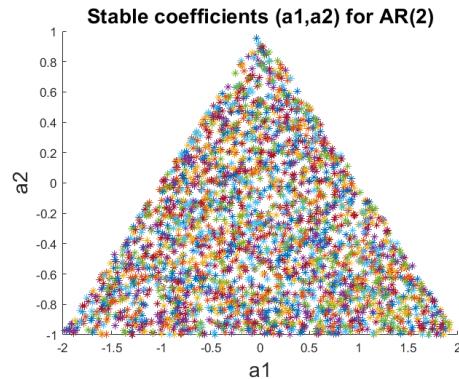


Figure 2.7: Pairs of coefficients  $(a_1, a_2)$  for which  $x[n]$  converges. The plot shows 10000 samples of  $(a_1, a_2)$ .

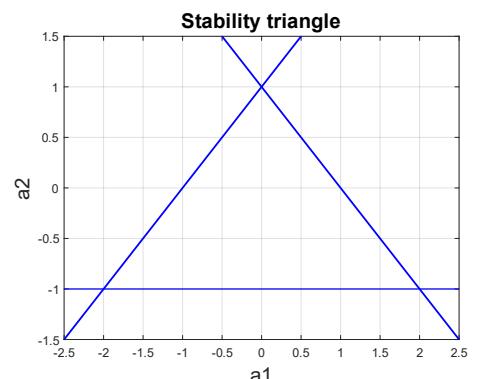


Figure 2.8: Plot of the stability triangle.

From figures 2.6 and 2.7 we can see that the shape of the plot converges into a triangle. This triangle has specific bounds described by the stability conditions, shown in figure 2.8, and it is known as the stability triangle.

By calculating the transfer function, we can derive the stability conditions. We do this by doing the Z-transform of the equation above:

$$H(z) = \frac{x(z)}{w(z)} = \frac{1}{1-a_1 z^{-1}-a_2 z^{-2}} \quad (33)$$

The poles are therefore given by equation 34:

$$z_{1,2} = \frac{a_1 \pm \sqrt{a_1^2 + 4a_2}}{2} \quad (34)$$

Since they must be inside the unitary circle:

$$\left| \frac{a_1 \pm \sqrt{a_1^2 + 4a_2}}{2} \right| < 1 \quad (35)$$

This gives two equations:

$$\begin{aligned} \frac{a_1 \pm \sqrt{a_1^2 + 4a_2}}{2} &> -1 \rightarrow a_2 - a_1 < 1 \\ \frac{a_1 \pm \sqrt{a_1^2 + 4a_2}}{2} &< 1 \rightarrow a_2 + a_1 < 1 \end{aligned} \quad (36)$$

By multiplying the two poles:

$$(z - \lambda_1)(z - \lambda_2) = z^2 - (\lambda_1 + \lambda_2)z + \lambda_1\lambda_2 \quad (37)$$

Therefore, if  $a_1 = (\lambda_1 + \lambda_2)$  and  $a_2 = \lambda_1\lambda_2$  and since both  $\lambda_1$  and  $\lambda_2$  must be smaller than 1:

$$|a_2| < 1$$

These three equations describe the boundaries of the stability triangle shown in figure 2.8.

### 2.3.2. ACF for the Sunspots series

The ACF of the sunspot series is plotted for different lengths ( $N = [5, 20, 250]$ ) alongside its zero mean ACF to study the difference between them in figure 2.9.

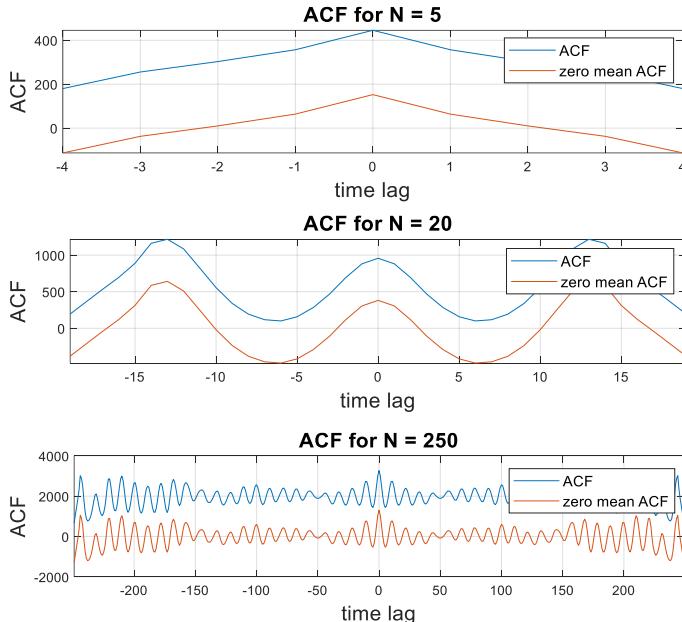


Figure 2.9: ACF of original and zero mean data of the sunspot series for different sample lengths.

As the value of  $N$  increases, the estimation of the ACF becomes more accurate in the representation of sunspots. When it is very small, the ACF doesn't show any characteristic pattern, however as it increases an oscillating pattern can be appreciated with a period of approximately 13-14 years. This behaviour was expected as the ACF of an AR(2) model is a mixture of damped exponentials, due to its real roots, and has a pseudo-periodic behaviour, due to its complex roots

However, for higher values of  $N$  the noise in the signal increases due to the decrease in samples considered. Hence, higher values of  $N$  should not be considered when trying to accurately represent the behaviour of sunspots.

The data has been centred, as we do not wish to model the DC offset (deterministic component), but the stochastic component (AR model driven by white noise). Hence, the zero-mean data provides a more accurate representation.

### 2.3.3. Yule-Walker Equations

The Yule Walker equations are used to calculate the coefficients of an AR model of order  $p$ . However, since the ideal order is unknown a confidence interval is needed. This threshold will indicate which coefficients are irrelevant and therefore will identify the correct model order. A 95% confidence interval is used where  $\text{PAC} \approx 0$  and has the value of  $\frac{\pm 1.96}{\sqrt{N}}$ .

The coefficient values obtained for a 10<sup>th</sup> order model are:

$$\boldsymbol{a}_{10} = [ 0.8212, -0.6783, -0.1223, 0.0473, -0.0156, 0.1623, 0.1751, 0.2276, 0.1766, 0.0037 ]$$

From figure 2.10 and from the values obtained, we can see that only the first 2 coefficients are significant and therefore enough to model the sunspot data. Hence, we conclude that it is an autoregressive model of order 2.

Figure 2.10 shows the PAC for both the original sunspot data and the standardised data, the latter one provides a more accurate representation since, as mentioned before, it removes the DC offset.

### 2.3.4. minimum description length (MDL) and the Akaike information criterion (AIC)

To determine what is the optimal model order we have to establish a trade-off between computational complexity and model accuracy, introducing a “penalty” for a high model order. Therefore, to select the optimal model order we follow the MDL and AIC, described by the following equations:

$$\begin{aligned} \text{MDL} &= \log(E_p) + \frac{p * \log(N)}{N} \\ \text{AIC} &= \log(E_p) + \frac{2p}{N} \\ \text{AICc} &= \log(E_p) + \frac{2p}{N} + \frac{2p(p+1)}{N-p-1} \end{aligned} \quad (38)$$

### MDL, AIC and AICc for increasing model order

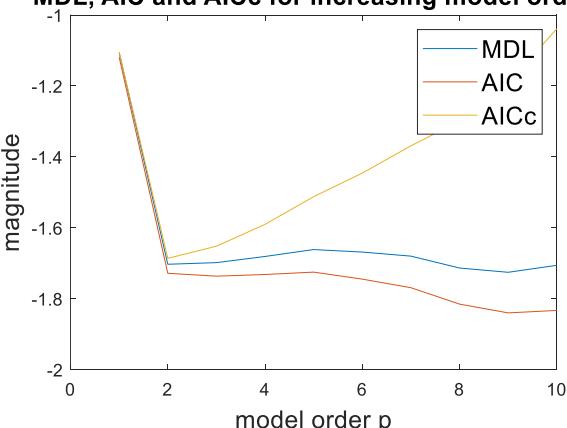


Figure 2.11: Plot of the MDL, AIC and AICc for model orders between 1 and 10.

with increasing model order will tend to zero, however a penalty term is introduced that increases as the model order increases in order to account for over-modelling in our model order selection.

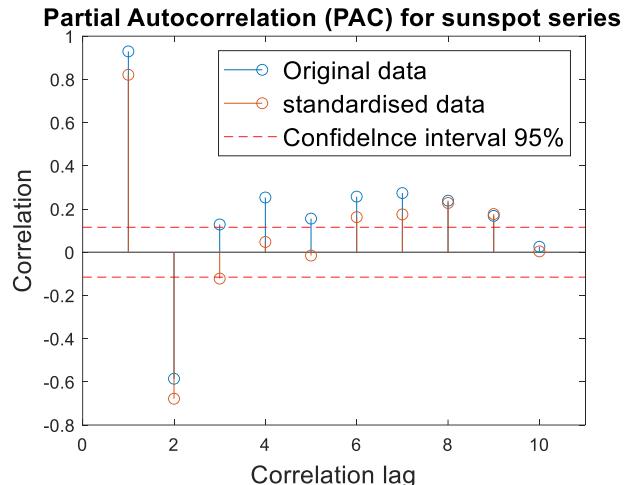


Figure 2.10: Partial Autocorrelation function (PAC) for the original and standardised data of the sunspot series.

Where  $E_p$  is the loss function,  $p$  is the model order and  $N$  is the number of samples. We also consider the corrected AIC (AICc) which is better suited for small sample sizes as it reduces the possibility of overfitting via an additional penalty term.

By observing figure 2.11, we observe a global minimum value of AICc at  $p = 2$  suggesting that the optimal model order of the AR process could be 2. However, it is also noticeable that the MDL and the AIC have a minimum around  $p = 9$ . These results are consistent with figure 2.10. Nevertheless, a model order of  $p = 9$  would increase the complexity of the model hence, a model order of  $p = 2$  could be the ideal one.

The loss function represents the error of the estimate that with increasing model order will tend to zero, however a penalty term is introduced that increases as the model order increases in order to account for over-modelling in our model order selection.

### 2.3.5. Prediction of the sunspot series for different model orders

Predictions for the sunspot data have been plotted in figure 2.12 using different AR model orders to compare their accuracy for different prediction horizons. For lower prediction horizons all model orders were able to accurately predict the data, however as the prediction horizon increases, AR(1) and AR(2) models show a prediction that compared to the prediction made with an AR(10) model, is less accurate. Therefore, the error in the prediction decreases as the model order increases.

However, using larger model orders causes over-fitting in the signal due to the increase in complexity causing a poor generalization of the data. Therefore, a trade-off has to be established between minimizing the prediction error and avoiding over-fitting in the signal.

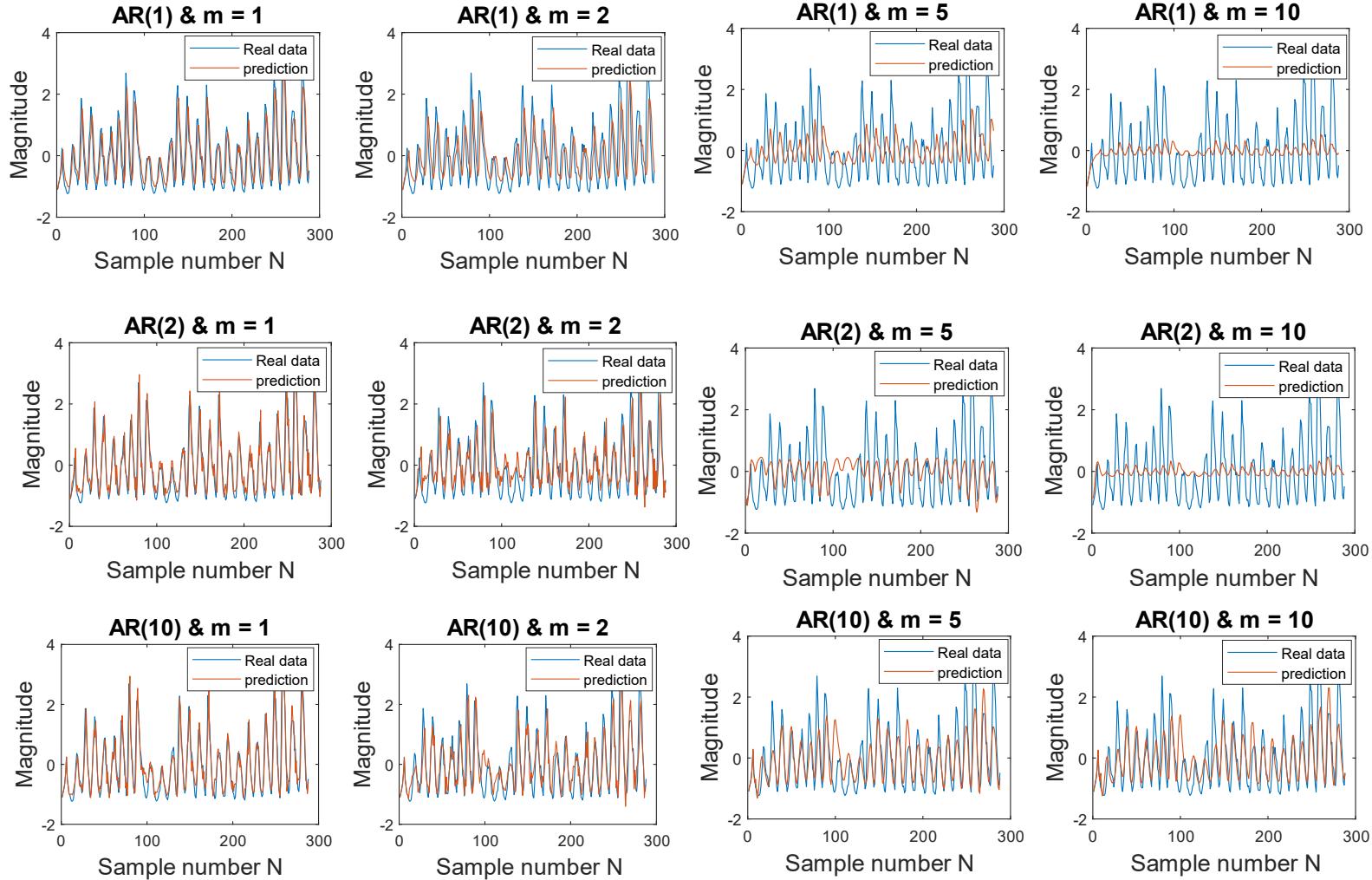


Figure 2.12: Plots showing the real and predicted data for different model orders (AR(1), AR(2) and AR(10)) and different prediction horizons  $m = [1, 2, 5, 10]$ .

## 2.4. CRLB

### 2.4.1. Ideal model order

The NASDAQ financial index has been studied in order to determine its optimal order. Looking at the partial autocorrelation function we can appreciate that only the first coefficient carries significant information since the rest are all below the threshold provided by the confidence interval.

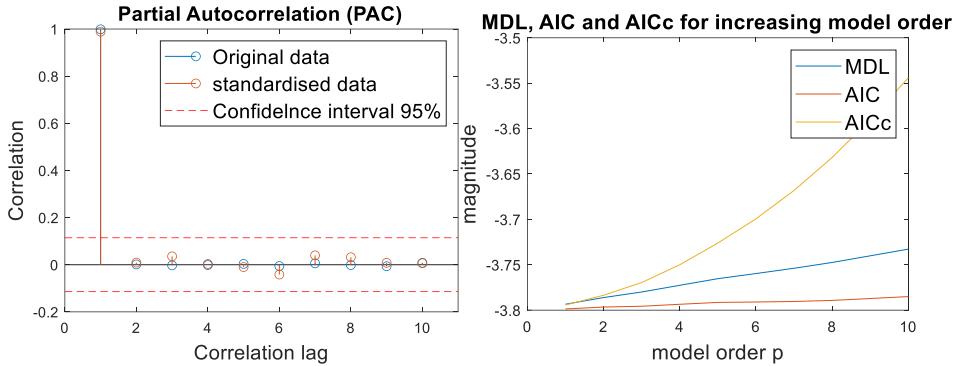


Figure 2.13: Plot of the Partial Autocorrelation (PAC) of the NASDAQ financial series. A threshold of 95% is also plotted.

Figure 2.14: Plot of the MDL, AIC and AICc for the NASDAQ financial series.

Furthermore, figure 2.14 shows the values of the MDL, AIC and AICc for increasing model order. All of them have a minimum at  $p = 1$ , since after that they only increase. Hence, we can conclude that the optimal model order is  $p = 1$ .

#### 2.4.2. Fisher information

For an autoregressive model of order 1, the fisher information matrix is given by equation 39:

$$[\mathbf{I}(\boldsymbol{\theta})]_{ij} = \frac{N}{2} \int_{-\frac{1}{2}}^{\frac{1}{2}} \frac{\partial \ln \hat{P}_x(f; \boldsymbol{\theta})}{\partial \theta_i} \frac{\partial \ln \hat{P}_x(f; \boldsymbol{\theta})}{\partial \theta_j} df \quad (39)$$

Since  $\ln \hat{P}_x(f; \boldsymbol{\theta}) = \ln \sigma^2 - \ln[1 - a_1 e^{-2\pi f j}] - \ln[1 - a_1 e^{2\pi f j}] = \ln \sigma^2 - j2\pi f \ln a_1 + j2\pi f \ln a_1 = \ln \sigma^2$

We also know the following:

$[\mathbf{I}(\boldsymbol{\theta})]_{11} = \frac{Nr_{xx}(0)}{\sigma^2}$  and  $[\mathbf{I}(\boldsymbol{\theta})]_{12} = [\mathbf{I}(\boldsymbol{\theta})]_{21} = 0$  and assuming that  $p = 1$  (from exercise 4.1) we can compute  $[\mathbf{I}(\boldsymbol{\theta})]_{22}$  following these derivations:

$$[\mathbf{I}(\boldsymbol{\theta})]_{22} = \frac{N}{2} \int_{-\frac{1}{2}}^{\frac{1}{2}} \frac{\partial(\ln \sigma^2)}{\partial \theta_i} \frac{\partial(\ln \sigma^2)}{\partial \theta_j} df = \frac{N}{2\sigma^4} \quad (40)$$

Therefore, the complete fisher information matrix is given by:

$$\mathbf{I}(\boldsymbol{\theta}) = \begin{pmatrix} \frac{Nr_{xx}(0)}{\sigma^2} & 0 \\ 0 & \frac{N}{2\sigma^4} \end{pmatrix} \quad (41)$$

From the CRLB theorem, the variance of the estimator is given by  $\sigma^2 \geq \frac{1}{\mathbf{I}(\boldsymbol{\theta})} = \mathbf{I}(\boldsymbol{\theta})^{-1}$

We can therefore calculate  $\mathbf{I}(\boldsymbol{\theta})^{-1}$ :

$$\mathbf{I}(\boldsymbol{\theta})^{-1} = \begin{pmatrix} \frac{\sigma^2}{Nr_{xx}(0)} & 0 \\ 0 & \frac{2\sigma^4}{N} \end{pmatrix} \text{ and therefore: } var(\hat{\sigma}^2) \geq \frac{2\sigma^4}{N} \text{ and } var(\hat{a}_1) \geq \frac{\sigma^2}{Nr_{xx}(0)}$$

And since the variance of an AR model of order 1 is given by:  $\sigma_x^2 = \frac{\sigma^2}{1-a_1^2} = r_{xx}(0)$ , we can therefore substitute in order to obtain:  $var(\hat{a}_1) \geq \frac{\sigma^2}{Nr_{xx}(0)} = \frac{1-a_1^2}{N}$

In figures 2.15 and 2.16 we can see the plot of the CRLB of the estimators  $\hat{\sigma}^2$  and  $\hat{a}_1$ .

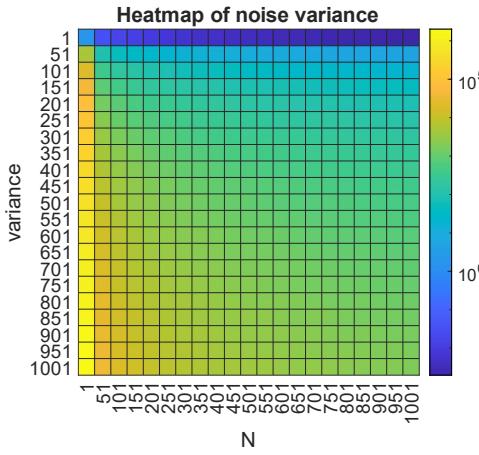


Figure 2.15: heatmap of  $\text{var}(\hat{\sigma}^2)$  as a function of  $\sigma^2$  and  $N$ .

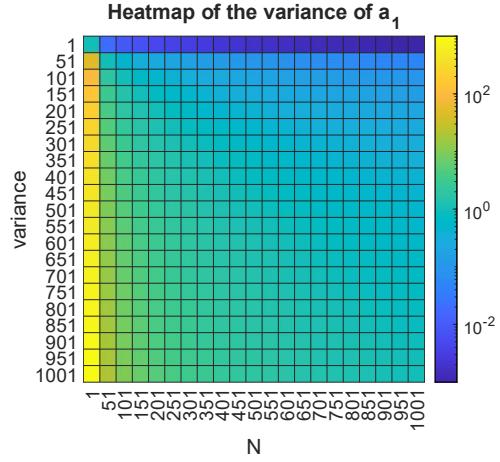


Figure 2.16: heatmap of  $\text{var}(\hat{a}_1)$  as a function of  $\sigma^2$  and  $N$ .

As expected,  $\text{var}(\hat{\sigma}^2)$  increases as  $N$  decreases and  $\sigma_x$  increases, the same occurs for  $\text{var}(\hat{a}_1)$ .

When we compute the coefficient of the AR(1) model using Matlab, we obtain a value of  $a = 0.9989$ . Therefore, in order to calculate  $\text{var}(\hat{a}_1)$  we do:

$$\text{var}(\hat{a}_1) = \frac{1-a_1^2}{N} = \frac{1-0.9989^2}{924} = 2.38 \times 10^{-6} \quad (42)$$

The value obtained is very small and can therefore be approximated to zero. If we examine the transfer function of an AR(1) model given by equation 43:

$$H(z) = \frac{1}{1-a_1z} = \frac{z}{z-a_1} \quad (43)$$

We notice the existence of a pole. As  $a_1$  tends more towards 1, the variance tends more towards zero, however if it reaches 1 the pole would be in the unitary circle ( $z = 1$ ) causing system instability. This is due to the fact that when  $a_1 = 1$  and  $z = 1$ ,  $z - a_1 = 0$  making the transfer function diverge.

#### 2.4.3. CRLB calculation

From the CRLB, it can be shown that:

$$\text{var}(\hat{P}_x(f; \boldsymbol{\theta})) \geq \frac{\partial \hat{P}_x(f; \boldsymbol{\theta})^T}{\partial \boldsymbol{\theta}} \mathbf{I}(\boldsymbol{\theta})^{-1} \frac{\partial \hat{P}_x(f; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \quad (44)$$

And we know that  $\boldsymbol{\theta} = [a_1, \sigma^2]$ ,  $\hat{P}_x(f; \boldsymbol{\theta}) = \frac{\hat{\sigma}^2}{|1-a_1e^{-j2\pi f}|^2}$  and that  $\mathbf{I}(\boldsymbol{\theta})^{-1} = \begin{pmatrix} \frac{\sigma^2}{Nr_{xx}(0)} & 0 \\ 0 & \frac{2\sigma^4}{N} \end{pmatrix}$ .

We can expand the following:  $\hat{P}_x(f; \boldsymbol{\theta}) = \frac{\hat{\sigma}^2}{|1-a_1e^{-j2\pi f}|^2} = \frac{\hat{\sigma}^2}{(1-a_1e^{-j2\pi f})(1-a_1e^{j2\pi f})}$ .

$$\frac{\partial \hat{P}_x(f; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \left[ \frac{\partial \hat{P}_x(f; \boldsymbol{\theta})}{\partial a_1}, \frac{\partial \hat{P}_x(f; \boldsymbol{\theta})}{\partial \sigma^2} \right]^T \quad (45)$$

$$= \left[ \frac{\hat{\sigma}^2 e^{-j2\pi f}}{(1-a_1e^{-j2\pi f})^2(1-a_1e^{j2\pi f})} + \frac{\hat{\sigma}^2 e^{j2\pi f}}{(1-a_1e^{j2\pi f})^2(1-a_1e^{-j2\pi f})}, \frac{1}{(1-a_1e^{-j2\pi f})(1-a_1e^{j2\pi f})} \right]^T$$

If  $A(f) = 1 - a_1e^{-j2\pi f}$ , we can therefore substitute in the equation 45:

$$\begin{aligned}\frac{\partial \hat{P}_x(f; \theta)}{\partial \theta} &= \left[ \frac{\hat{\sigma}^2(1 - A(f))}{a_1 A(f)^2 A(-f)} + \frac{\hat{\sigma}^2(1 - A(-f))}{a_1 A(-f)^2 A(f)}, \frac{1}{A(f) A(-f)} \right]^T = \left[ \frac{\hat{\sigma}^2(A(-f) + A(f) - 2|A(f)|^2)}{a_1 A(f)^2 A(-f)^2}, \frac{1}{A(f) A(-f)} \right]^T \\ &= \left[ \frac{\hat{\sigma}^2(A(-f) + A(f) - 2|A(f)|^2)}{a_1 |A(f)|^4}, \frac{1}{|A(f)|^2} \right]^T\end{aligned}\quad (46)$$

Going back to the equation 44:

$$var(\hat{P}_x(f; \theta)) \geq \left[ \frac{\hat{\sigma}^2(A(-f) + A(f) - 2|A(f)|^2)}{a_1 |A(f)|^4}, \frac{1}{|A(f)|^2} \right] \begin{pmatrix} \frac{\sigma^2}{Nr_{xx}(0)} & 0 \\ 0 & \frac{2\sigma^4}{N} \end{pmatrix} \begin{pmatrix} \frac{\hat{\sigma}^2(A(-f) + A(f) - 2|A(f)|^2)}{a_1 |A(f)|^4} \\ \frac{1}{|A(f)|^2} \end{pmatrix}$$

If we compute the equation above we obtain:

$$var(\hat{P}_x(f; \theta)) \geq \frac{2\sigma^4}{N|A(f)|^4} - \frac{\sigma^4(a_1^2 - 1)(A(-f) + A(f) - 2|A(f)|^2)^2}{Na_1^2|A(f)|^8} \quad (47)$$

## 2.5. Real world signals: ECG experiment

### 2.5.1. Probability density estimate

We plot the probability density estimates (PDE) for the original heart rate data for unconstraint breathing and the averaged data. In order to average the data we use two different weighting coefficients ( $\alpha$ ) in order to study their effect in the resulting PDE.

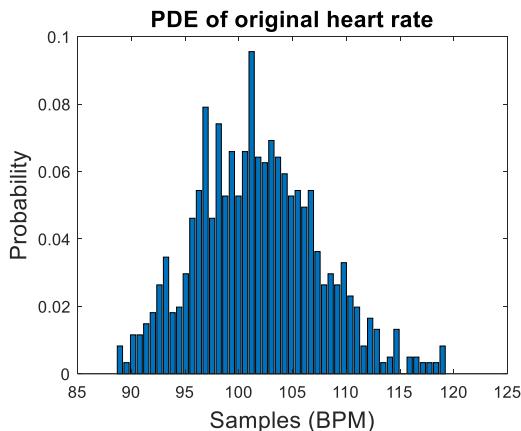


Figure 1.17: PDE of original heart rate (unconstrained breathing)

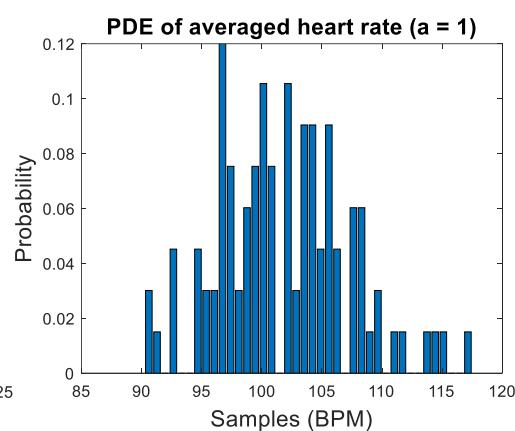


Figure 1.18: PDE of averaged heart rate for  $\alpha=1$ .

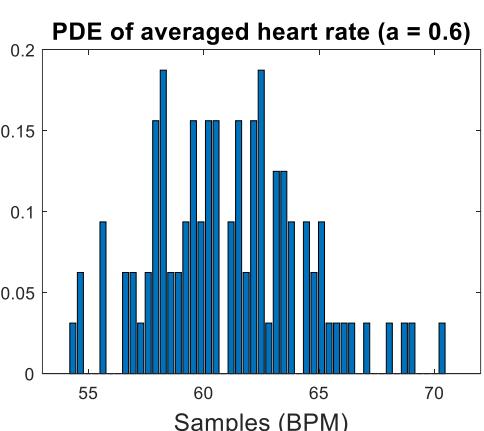


Figure 1.19: PDE of averaged heart rate for  $\alpha=0.6$ .

From figures 1.17, 1.18 and 1.19 we can see that averaging the data reduces the spread of the PDE in the x axis and therefore, reduces the variance in the estimate. Hence, the results will be more accurate.

By averaging the data with a weighting coefficient of  $\alpha = 1$  we obtain an average heart rate of 95-105 BPM. When resting the average heart rate is between 60-100 BPM<sup>1</sup> therefore our results indicate that the test subject has quite a high heart rate.

When the weighting coefficient used to average the data is reduced to  $\alpha = 0.6$  the result obtained, shown in figure 1.19, have been shifted to the left proportionally to 0.6 and have been reduced in width, reducing the variance of the estimate. However, in this case the average heart rate has been reduced to 58-62 BPM

---

<sup>1</sup> <https://www.bhf.org.uk/informationsupport/how-a-healthy-heart-works/your-heart-rate>

indicating that bias has been introduced in the estimate. The weighting coefficients just scale and shift the PDE.

### 2.5.2. AR Modelling of heart rate

The normalized ACF for each trial were plotted in figure 1.20.

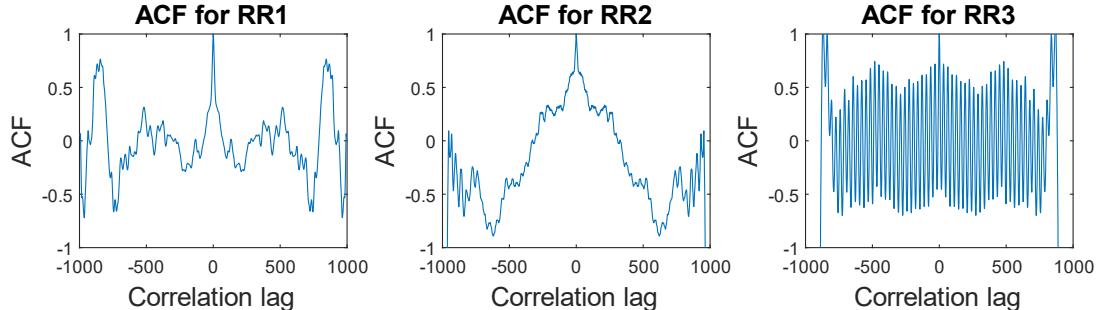


Figure 1.20: Normalised Autocorrelation Functions (ACF) that correspond to the three different trials

The ACF of an AR model is composed of damped sine waves, whereas for an MA model it has a finite length and the decay is determined by the order of the model. Knowing this, we conclude that the heart rate data can be modelled by an AR model.

Furthermore, by computing the PAC, MDL, AIC and AICc for each trial we can identify the optimal model order for each.

#### Trial 1 (RR1): unconstraint breathing

The PAC shows that only the 4 first coefficients are above the threshold indicating a model order of  $p = 4$ . This is confirmed by the MDL, AIC and AICc since they reach a minimum value at 4.

#### Trial 2 (RR2): constrained breathing at 50 beats per minute

The PAC shows that the first three coefficients are above the threshold, the 9<sup>th</sup> and the 10<sup>th</sup> are also above the threshold however, to avoid computational complexity, the optimum model order is  $p = 3$ . The MDL, AIC and AICc all have a local minimum at  $p = 3$ , and their values don't decrease until  $p = 8$ . Hence, the optimum model order is  $p = 3$ .

#### Trial 3 (RR3): constrained breathing at 15 beats per minute

The PAC suggests a model order of  $p = 5$ , however the last two coefficients have a smaller magnitude in comparison to the previous ones. Moreover, the MDL, AIC and AICc shows that from  $p = 4$  onwards their values either stay constant or increase and that from  $p = 3$  to  $p = 4$  the difference is minimal. Therefore, the optimal model order is  $p = 3$ .

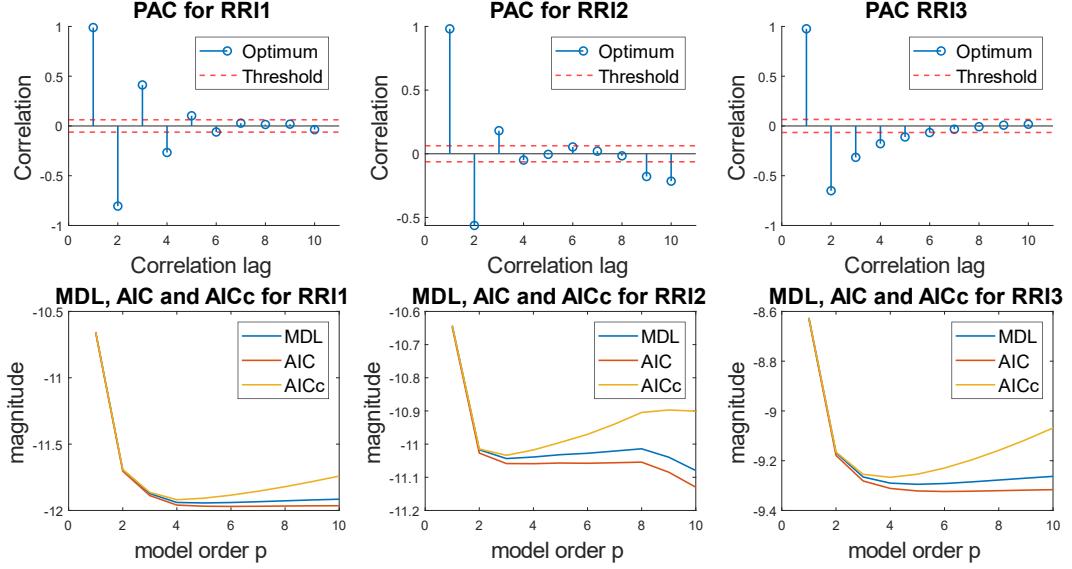


Figure 1.21: PAC, MDL, AIC and AICc of the RRI data for the three different trials.

### 3. Spectral estimation and modelling

The estimate of the PSD is known as the periodogram and it is calculated by the following equation based of the Fast Fourier Transform (FFT):

$$\hat{P}_x(f) = \frac{1}{N} \left| \sum_{n=0}^{N-1} x[n] e^{-j2\pi f \frac{n}{N}} \right|^2 \quad (48)$$

The periodogram for an N-sample realisation, where  $N = [128, 256, 512]$ , is calculated and plotted in figure 3.1.

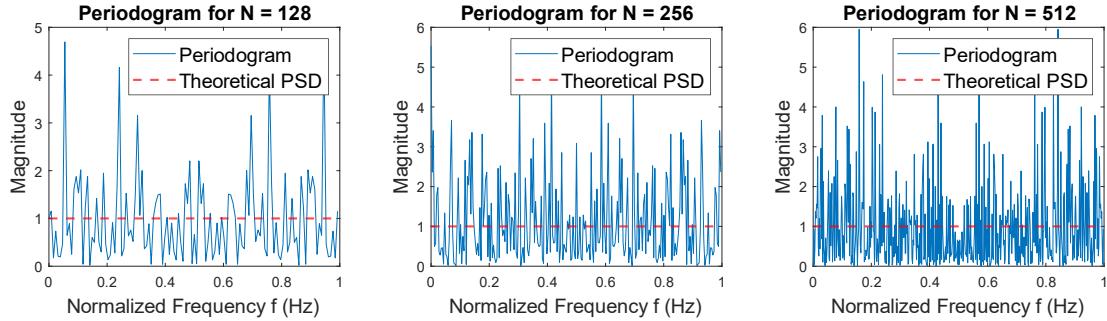


Figure 3.1: Periodogram of a WGN signal of different sample lengths  $N = [128, 256, 512]$

As we can see, all the periodograms plotted in figure 3.1 are symmetric with respect to the centre that is situated at 0.5 in the normalized frequency axis. It is also evident that even though the number of samples increase, the variance of the periodogram remains constant. The variance of the estimate is large, this is due to the fact that it is proportional to the square of the PSD.

The ideal PSD for WGN is constant and can be calculated using equation 49:

$$P_x(f) = \left| \sum_{r=-\infty}^{\infty} R_x(\tau) e^{-j2\pi f\tau} \right| \quad (49)$$

Since we are considering WGN,  $R_x(\tau)$  only has a value when  $\tau = 0$  and it has a value of  $R_x(0) = 1$ . Therefore, we conclude that ideally  $P_x(f) = 1$ . This is consistent with the plots since all of them tend to an approximate average of 1.

### 3.1. Average Periodogram estimates

#### 3.1.1. Filtering the periodogram

We use a zero-mean FIR filter to smooth out the periodograms previously calculated. In figure 3.2, we plot the original periodogram and the filtered periodogram.

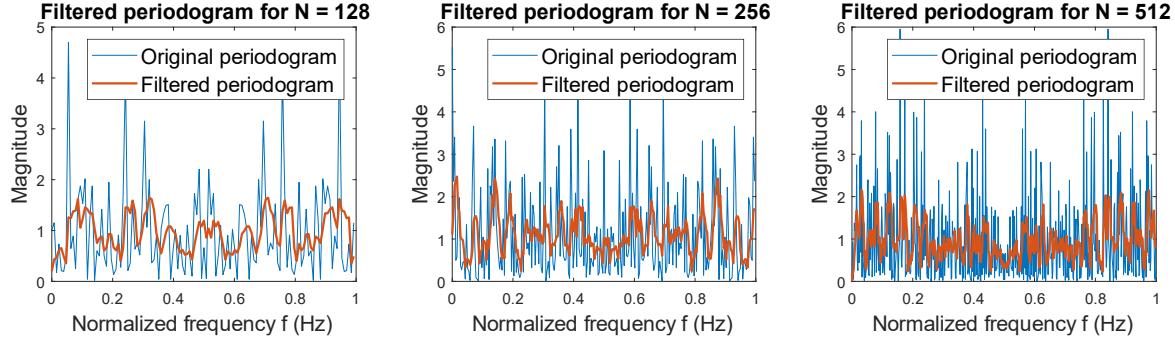


Figure 3.2: zero phase filtered periodogram by an FIR filter of a WGN signal of different sample lengths  $N = [128, 256, 512]$

When we filter the periodogram we notice a significant decrease in the variance of the estimate, however, this comes at a price since it also introduces bias in the estimate. The filtered estimate seems to represent better the theoretical value of the PSD previously calculated, and hence it improves the apparent PSD estimate.

#### 3.1.2. Periodogram of non-overlapping segments of WGN

We generate a 1024-sample realisation of WGN and we divide it into 8 equal and non-overlapping segments of 128 samples each. The periodogram for each segment has been plotted in figure 3.3.

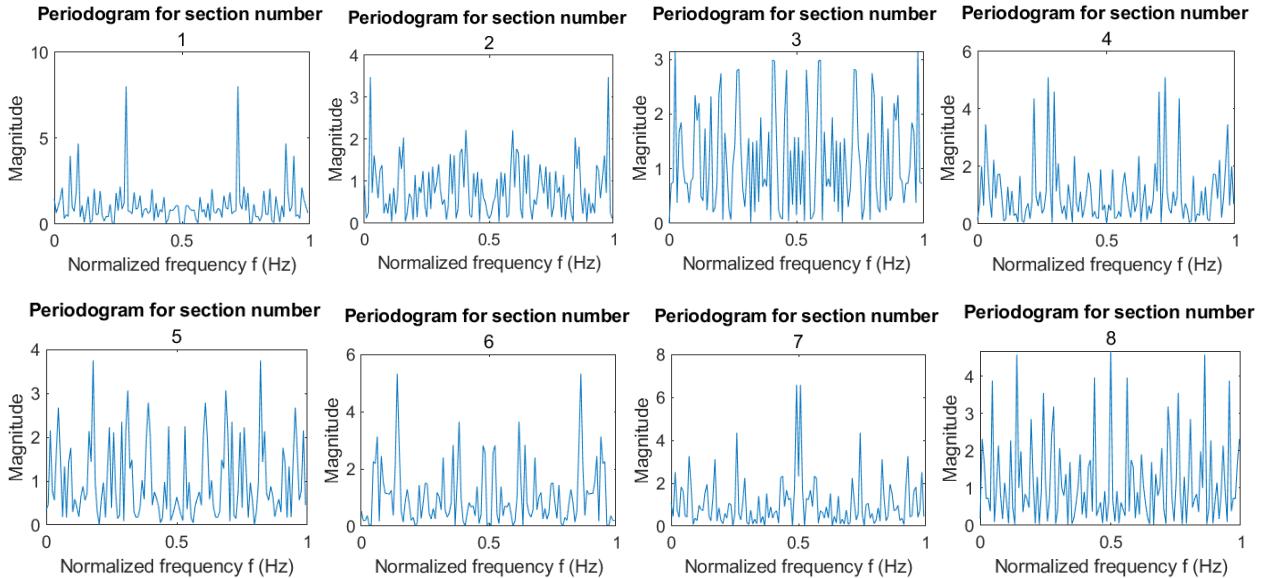


Figure 3.3: Periodograms of 128-segments of a 1024- sample long WGN signal.

Each of these segments are an unreliable estimate of the PSD since they present random spikes and each segment is uncorrelated with the others as there is no constant pattern between them.

#### 3.1.3. Averaged Periodogram

If we calculate the average periodogram using all the sections obtained in the previous exercise we obtain the plot shown in figure 3.4.

The averaged periodogram provides a more accurate estimate of the PSD, since its variance has decreased and it therefore approximates the theoretical value of the PSD better. However, the resolution of the periodogram has also decreased since the averaged periodogram resulted in a length of 128, whereas the original had a length of 1024 samples.

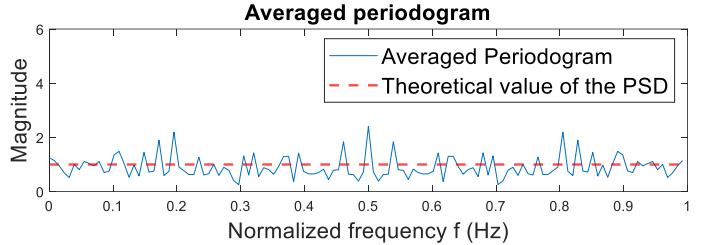


Figure 3.4: Plot of the averaged periodogram of the segments shown in figure 3.3.

### 3.2. Spectrum of Autoregressive Processes

Figure 3.5 shows a 1024-sampled realisation of WGN filtered and unfiltered. The filter has the following coefficients:  $\mathbf{b} = 1$  and  $\mathbf{a} = [1, 0.9]$ . We can say that the filter applied is a high pass filter since the high frequencies of the signal have been magnified whereas lower frequencies haven't. Notice that the first 40 values of the resulting output have been removed as these are affected by transient effects of the filter.

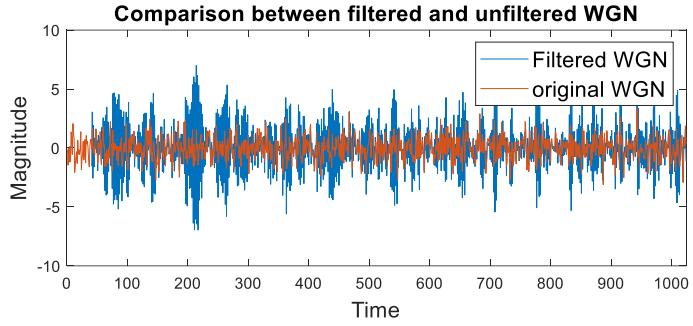


Figure 3.5: Plot of the input and output of the filter.

#### 3.2.1. Theoretical PSD and Periodogram

We compute the theoretical PSD using the Matlab function `freqz` and we plot it in figure 3.6. We can calculate the cut-off frequency of the filter by doubling the magnitude of the input signals power. By doing this we obtain a cut-off frequency of 0.405, this point is where the magnitude of the PSD starts increasing exponentially in figure 3.6.

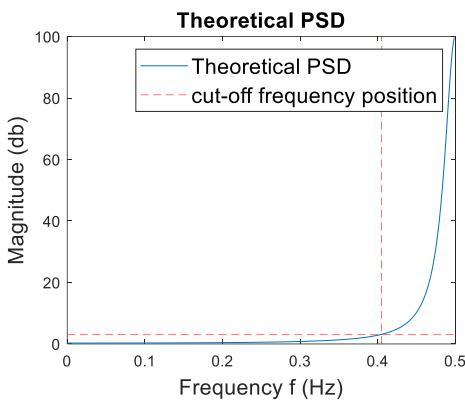


Figure 3.6: Plot of the ideal PSD of the process

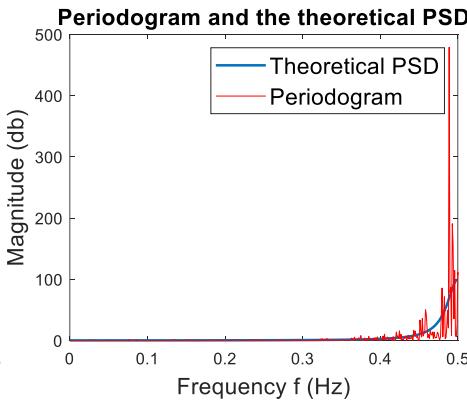


Figure 3.7: Plot of the ideal and estimated PSD of the process

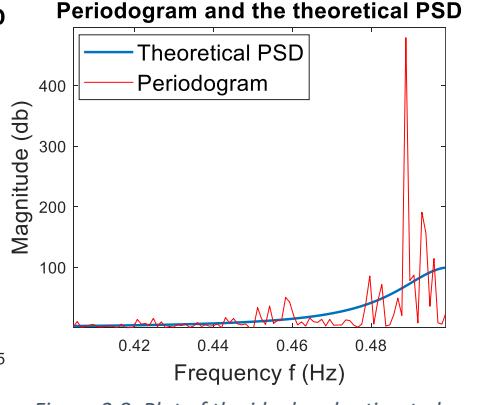


Figure 3.8: Plot of the ideal and estimated PSD of the process for  $0.4 < f > 0.5$

Moreover, we calculate the periodogram of the filtered WGN signal in order to compare it with its theoretical value. As we can see in figure 3.7, the periodogram follows the shape of the theoretical PSD however, it introduces variance. This variance can be better appreciated in figure 3.8, where we zoom in between the

frequencies 0.4-0.5 Hz and therefore we notice that the periodogram introduces oscillations that follow the theoretical value.

The presence of this oscillations is caused by the underlying rectangular windowing within the periodogram estimator. This is due to the fact that, the ideal Dirac delta function the signal is convolved with gets approximated by sinc waves in the frequency domain. These sinc waves are transformed into a rectangular window in the time domain by preforming a Fourier transform and multiplied by the original signal. The resulting oscillations are therefore introduced by the sinc function.

### 3.2.2. Model-based PSD

The model based PSD is generated by assuming that the output sequence is generated by an AR(1) model with two parameters ( $a_1, \sigma_x^2$ ). The estimates of these parameters depend on the estimated autocorrelation function of the output sequence ( $\hat{R}_y$ ):

$$\hat{a}_1 = \frac{-\hat{R}_y(1)}{\hat{R}_y(0)} \text{ and } \hat{\sigma}_x^2 = \hat{R}_y(0) + \hat{a}_1 \hat{R}_y(1) \quad (50)$$

The values obtained were  $\hat{a}_1 = 0.8739$  and  $\hat{\sigma}_x^2 = 1.0230$ . These values are very similar to the theoretical coefficients of the model (0.9 and 1).

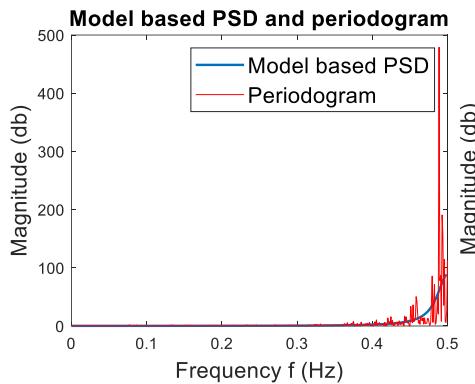


Figure 3.9: Plot of the model based and estimated PSD.

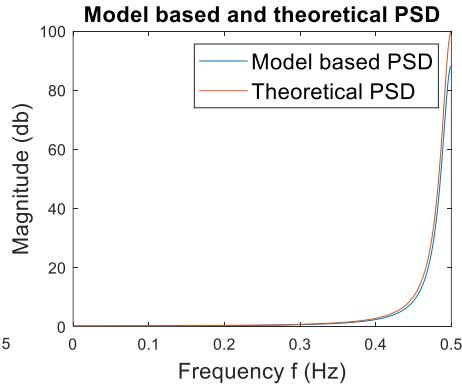


Figure 3.10: Plot of the model based and theoretical PSD.

Figures 3.9 and 3.10 show that the model-based PSD has a very similar shape to the theoretical PSD, however it does have a lower amplitude at the peak of the theoretical PSD. Similarly, the periodogram also follows the shape of the model-based PSD introducing variability in the form of oscillations as discussed before. We can therefore conclude that the model-based PSD is a more accurate estimate compared to the periodogram.

### 3.2.3. Sunspot Series

The periodogram has been calculated for the original sunspot data and for the zero-mean sunspot data and has been plotted with the model-based estimate for different model orders.

The model based PSD for the AR(1) model doesn't contain enough information to model the second peak of the periodogram and as you increase the model order a more accurate estimation of the shape of the periodogram is achieved, like in the AR(10) model.

As we can see in figure 3.11, the periodograms calculated contain fluctuations and as said in previous exercises it is not an optimal estimate. Hence, the fact that AR(10) models the periodogram more accurately doesn't mean it's the optimal model order, AR(2) however, follows the trend in the data without over-modelling. This conclusion is consistent with previous exercises.

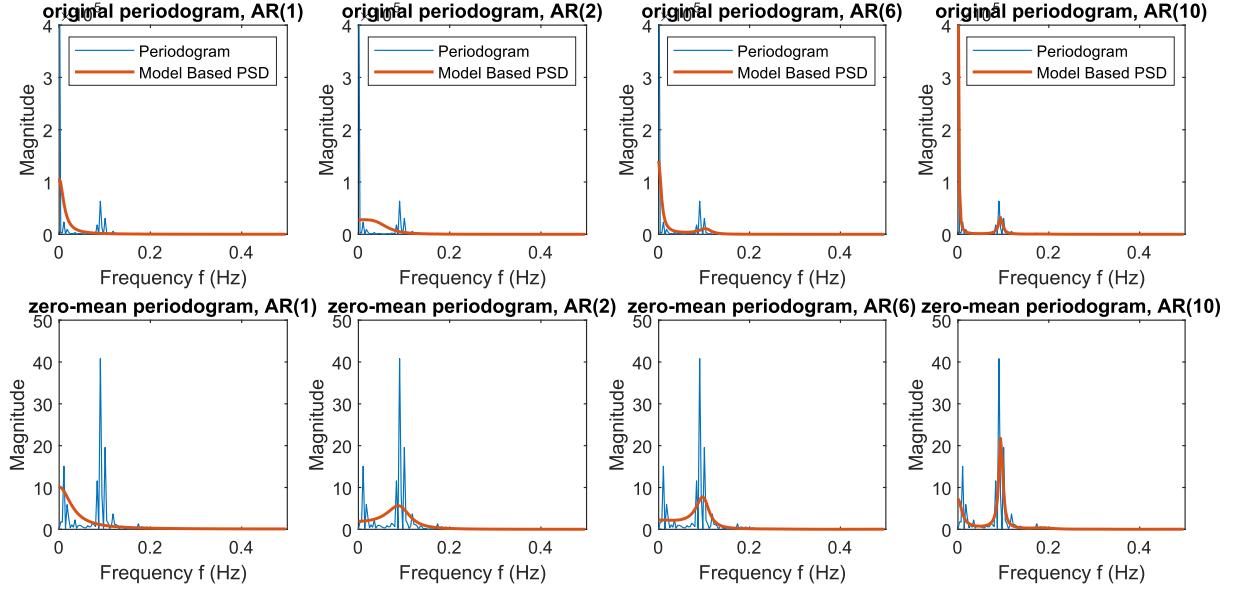


Figure 3.11: Plot of the model based and estimated PSD for different model orders and for original and zero-mean data.

### 3.3. Least Squares Estimation (LSE) of AR coefficients

#### 3.3.1. Cost Function calculation

The LS cost function for finding the unknown AR coefficients  $\mathbf{a} = [a_1, a_2, \dots, a_p]^T$  is given by:

$$J = \sum_{k=1}^M [\hat{r}_{xx}[k] - \sum_{i=1}^p a_i \hat{r}_{xx}[k-i]]^2 \quad (51)$$

And the autocorrelation function is given by:

$$\hat{r}_{xx}[k] = \sum_{i=1}^p a_i \hat{r}_{xx}[k-i] + \epsilon[k] \quad (52)$$

Substituting the equation 52 in equation 51, we obtain:

$$J = \sum_{k=1}^M \left[ \sum_{i=1}^p a_i \hat{r}_{xx}[k-i] + \epsilon[k] - \sum_{i=1}^p a_i \hat{r}_{xx}[k-i] \right]^2 = \sum_{k=1}^M [\epsilon[k]]^2 = \sum_{k=1}^M [x[k] - s[k]]^2 = (\mathbf{x} - \mathbf{s})^T (\mathbf{x} - \mathbf{s})$$

Where  $\mathbf{x}$  is the original signal and  $\mathbf{s}$  is the predicted signal and they are given by:

$$\mathbf{x} = \begin{bmatrix} \hat{r}_{xx}[1] \\ \hat{r}_{xx}[2] \\ \vdots \\ \hat{r}_{xx}[M] \end{bmatrix} \text{ and } \mathbf{s} = \begin{bmatrix} \sum_{i=1}^p a_i \hat{r}_{xx}[1-i] \\ \sum_{i=1}^p a_i \hat{r}_{xx}[2-i] \\ \vdots \\ \sum_{i=1}^p a_i \hat{r}_{xx}[M-i] \end{bmatrix} \quad (53)$$

Since  $\mathbf{a} = [a_1, a_2, \dots, a_p]^T$ ,  $\mathbf{s}$  can be written as  $\mathbf{s} = \mathbf{aH}$  where  $\mathbf{H}$  is the observation matrix and it is given by:

$$\mathbf{H} = \begin{pmatrix} \hat{r}_{xx}[0] & \hat{r}_{xx}[-1] & \cdots & \hat{r}_{xx}[1-p] \\ \hat{r}_{xx}[1] & \hat{r}_{xx}[0] & \cdots & \hat{r}_{xx}[2-p] \\ \vdots & & \ddots & \vdots \\ \hat{r}_{xx}[M-1] & \hat{r}_{xx}[M-2] & \cdots & \hat{r}_{xx}[M-p] \end{pmatrix}$$

Therefore, the cost function is given by:  $J = (\mathbf{x} - \mathbf{aH})^T (\mathbf{x} - \mathbf{aH}) = \mathbf{x}^T \mathbf{x} - 2\mathbf{x}^T \mathbf{H} \mathbf{a} + \mathbf{H}^T \mathbf{a}^T \mathbf{H} \mathbf{a}$

In order to find the least square estimates for the coefficients, the cost function must be minimised:

$$\frac{\partial J}{\partial \alpha} = -2x^T H - 2HH^T \alpha = 0 \rightarrow \alpha = (HH^T)^{-1}H^T x \quad (54)$$

The Yule-Walker equations can be expressed in the following compact matrix:

$$r_{xx} = R_{xx} \alpha \quad (55)$$

Where,

$$r_{xx} = \begin{pmatrix} r_{xx}(1) \\ r_{xx}(2) \\ \vdots \\ r_{xx}(p) \end{pmatrix} \text{ and } R_{xx} = \begin{pmatrix} \hat{r}_{xx}[0] & \hat{r}_{xx}[-1] & \cdots & \hat{r}_{xx}[p-1] \\ \hat{r}_{xx}[1] & \hat{r}_{xx}[0] & \cdots & \hat{r}_{xx}[p-2] \\ \vdots & \ddots & \ddots & \vdots \\ \hat{r}_{xx}[p-1] & \hat{r}_{xx}[p-2] & \cdots & \hat{r}_{xx}[0] \end{pmatrix}$$

Therefore, to estimate the Yule-Walker coefficients we use:  $\alpha = r_{xx} R_{xx}^{-1}$

Comparing both estimates we realise that both of them require the calculation of the autocorrelation function of the signal. The presence of non-zero mean noise biases the LSE estimator, as the LS approach assumes that the observed data is composed of a deterministic signal and zero mean noise. Whereas the Yule-Walker estimator is unbiased. Moreover, the LS estimator requires the calculation of  $(HH^T)^{-1}H^T$  that is a very costly operation, whereas the Yule-Walker approach only requires the calculation of  $R_{xx}^{-1}$ .

### 3.3.2. Observation Matrix H.

The observation matrix  $H$  is a deterministic matrix. This is due to the fact that in the LSE approach, the observation matrix is composed of multiple ACF of a deterministic signal and zero mean noise. Since it is calculated using the expected value of the data, the zero mean noise will have no effect in the result, meaning that only the deterministic part is considered.

### 3.3.3. Calculating AR coefficients using the LSE approach

The AR coefficients were calculated using equation 55 derived in the previous exercise. For model orders from 1 to 10, we obtained:

$$\alpha_{p=1} = [0.8609]$$

$$\alpha_{p=2} = [1.6856 \ -0.9582]$$

$$\alpha_{p=3} = [2.4531 \ -2.3039 \ 0.7951]$$

$$\alpha_{p=4} = [2.2128 \ -1.6134 \ 0.0638 \ 0.2961]$$

$$\alpha_{p=5} = [2.1619 \ -1.6283 \ 0.3494 \ -0.0903 \ 0.1729]$$

$$\alpha_{p=6} = [2.1429 \ -1.6142 \ 0.2846 \ 0.1383 \ -0.1062 \ 0.1233]$$

$$\alpha_{p=7} = [2.1302 \ -1.6081 \ 0.2896 \ 0.0435 \ 0.1773 \ -0.2000 \ 0.1395]$$

$$\alpha_{p=8} = [2.0924 \ -1.5539 \ 0.2154 \ 0.1340 \ -0.1214 \ 0.5136 \ -0.6327 \ 0.3308]$$

$$\alpha_{p=9} = [2.1757 \ -1.6988 \ 0.3318 \ 0.0971 \ -0.1073 \ 0.7065 \ -1.2616 \ 1.0484 \ -0.3173]$$

$$\alpha_{p=10} = [2.1712 \ -1.6847 \ 0.3158 \ 0.1072 \ -0.1119 \ 0.7087 \ -1.2486 \ 1.0091 \ -0.2746 \ -0.0180]$$

As we can see the second coefficient is the last one that carries important information. This observation is reinforced in figure 3.11, where  $p = 2$  can be identified as the optimal model order.

### 3.3.4. Optimal AR model order

In order to determine the optimal model order, the approximation error of the AR coefficients previously derived will be calculated. It is calculated by computing the mean squared error between the model output  $s[k]$  and the observed data  $x[k]$ .

The normalized mean squared error is plotted in figure 3.12 for different model orders. From model order  $p = 2$  onwards, the mean squared error approximately remains at a constant value. However, the higher the model order is, the more computationally expensive it will become. Hence, a model order of 2 is optimal since it has a low mean squared error, meaning that it will be accurate, and it is not computationally expensive. The conclusion reached supports the results obtained in exercise 3.2.3 and 3.3.3.

### 3.3.5. Power Spectrum of Sunspot series

The power spectrum for different model orders were plotted alongside the periodogram of the sunspot time series in figure 3.13. Model order 2 follows the trend of the data without imitating the random oscillations of the periodogram, hence as in the previous exercise we conclude that  $p = 2$  is the optimal model order.

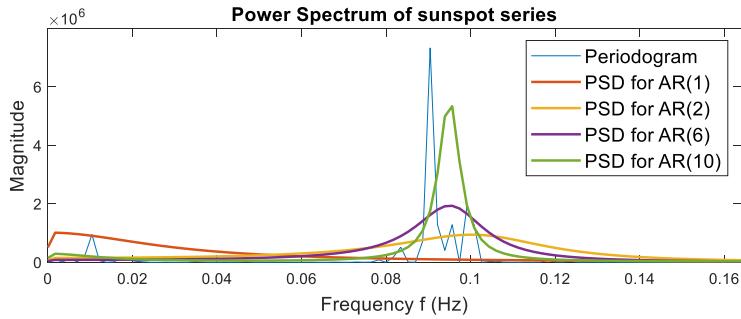


Figure 3.13: Plot of the periodogram and the PSD of different AR models calculated with the LSE method.

### 3.3.6. Approximation error for different data lengths

The approximation error for the AR(2) model for data lengths  $\mathbf{N} = [10 : 5 : 250]$  was plotted in order to identify the optimum data length.

As we can see in figure 3.15, the mean squared error generally decreases as the sample size increases. However, at approximately a sample size of  $N \approx 75$  the error increases slightly and it eventually starts decreasing progressively at a sample size of approximately  $N \approx 135$ . Therefore, we can conclude that the optimal sample size is  $N \approx 75$  since a minimum error is achieved with minimal computational complexity.

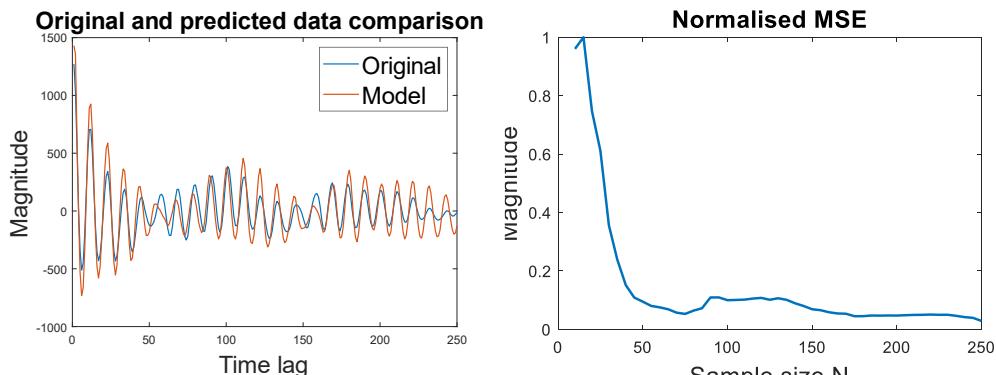


Figure 3.14: ACF comparison of the estimated AR(2) and original sunspot data.

Figure 3.15: Plot of the normalized MSE for different data lengths.

The AR(2) modelled data has been plotted in figure 3.14 alongside the original ACF data. The model order  $p=2$  was chosen since it was deemed to be optimal.

### 3.4. Spectrogram for time-frequency analysis: dial tone pad

A random London landline was generated (020 XXXX XXXX) with a sampling frequency of 32768 Hz. To establish an appropriate sampling rate we have to take into consideration the Nyquist criterion that establishes that the minimum sampling frequency that avoids aliasing is double the highest frequency of the signal. In this case, the highest frequency of the signal is 1477 Hz and since  $32768 \text{ Hz} > 2 \times 1477 \text{ Hz}$  the sampling frequency chosen is appropriate. The sequence  $y$  was then plotted with an idle time in between each number that was dialled and with the corresponding sampling frequency.

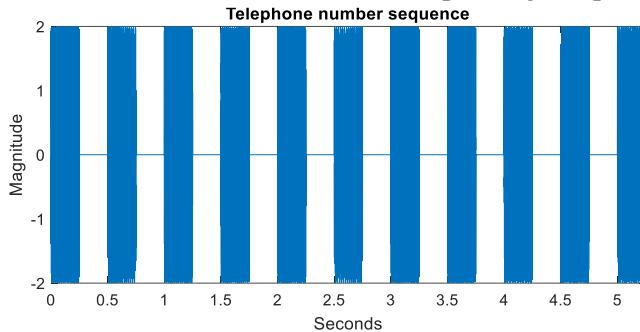


Figure 3.16: Plot of the signal generated when pressing a random phone number with an idle time of 0.25s between number.

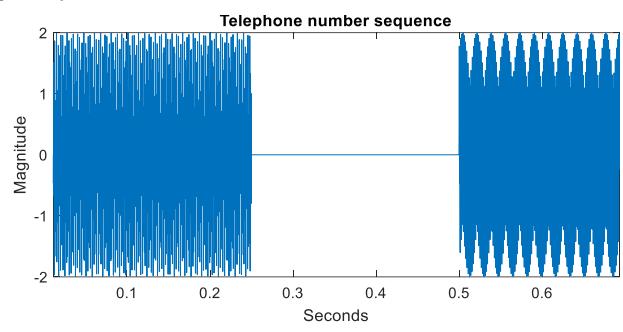


Figure 3.17: Plot of the signal generated when pressing 0 and 2 with an idle time of 0.25s between number.

If we zoom in into the signals, as done in figures 3.18 and 3.19 for numbers 0 and 2, we can better appreciate the patterns they present and the differences between each.

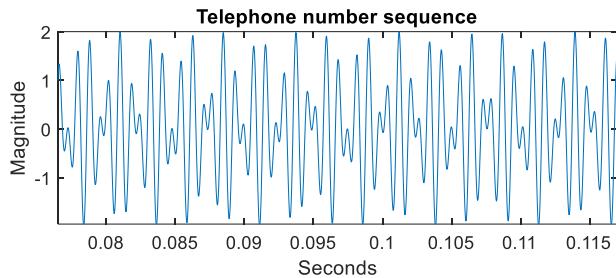


Figure 3.18: Plot of the signal generated for key 0.

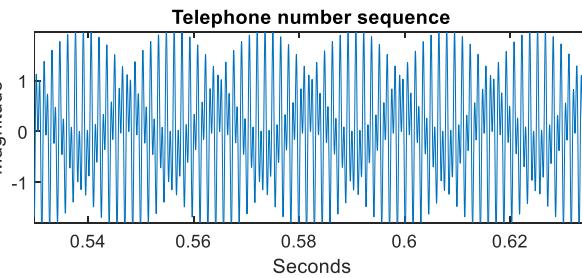


Figure 3.19: Plot of the signal generated for key 2.

#### 3.4.1. Spectrogram of sequence $y$

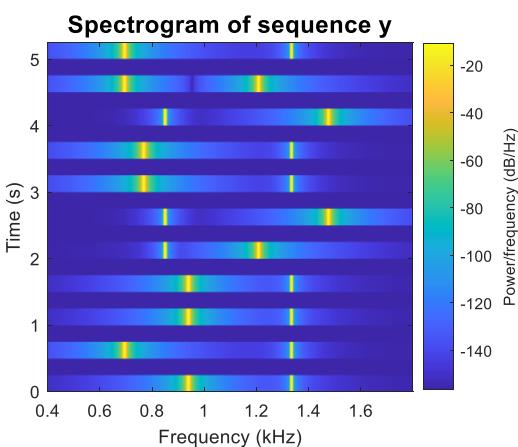


Figure 3.20: Spectrogram of the sequence  $y$  that represents the sequence of numbers that have been dialled.

The spectrogram of the sequence was plotted in figure 3.20 and was generated with the Matlab command `spectrogram` and by using a hanning window to separate the dialled numbers. The FFT of each number was plotted with no over-lapping segments in figure 3.21.

In both plots the idle bands present uniform and low frequency power bands. Whereas the telephone keys display two clear frequencies that correspond to those specified in the dial pad table. The key pressed can be identified by looking at the frequency values at which there is a spike in the FFT plot. This also can be done by identifying at which frequencies there is a yellow band in the spectrogram. By doing this and comparing the resulting frequencies with dial pad table the dialled number can be identified.

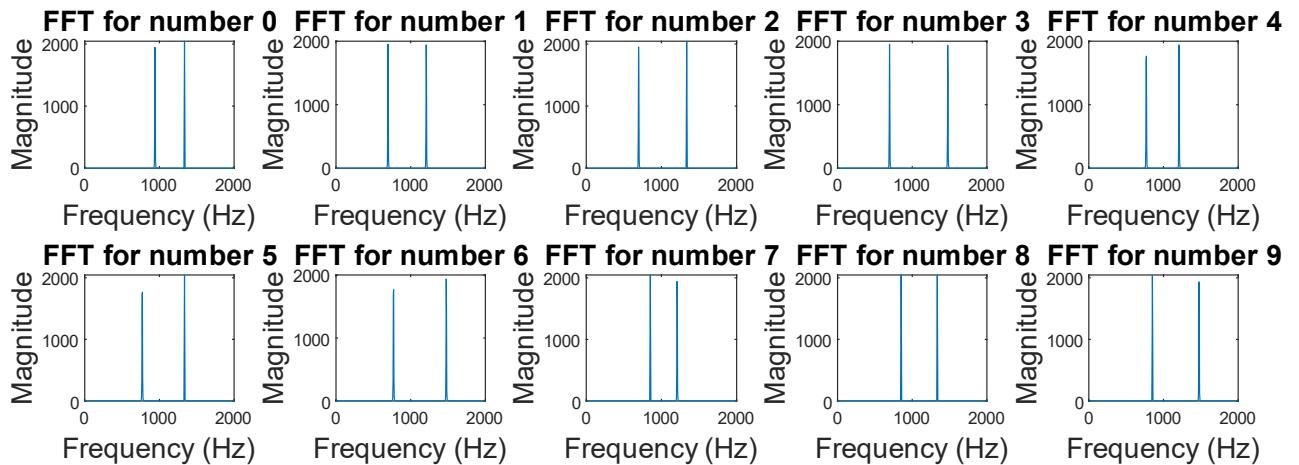


Figure 3.21: FFT of the tones that correspond to each number.

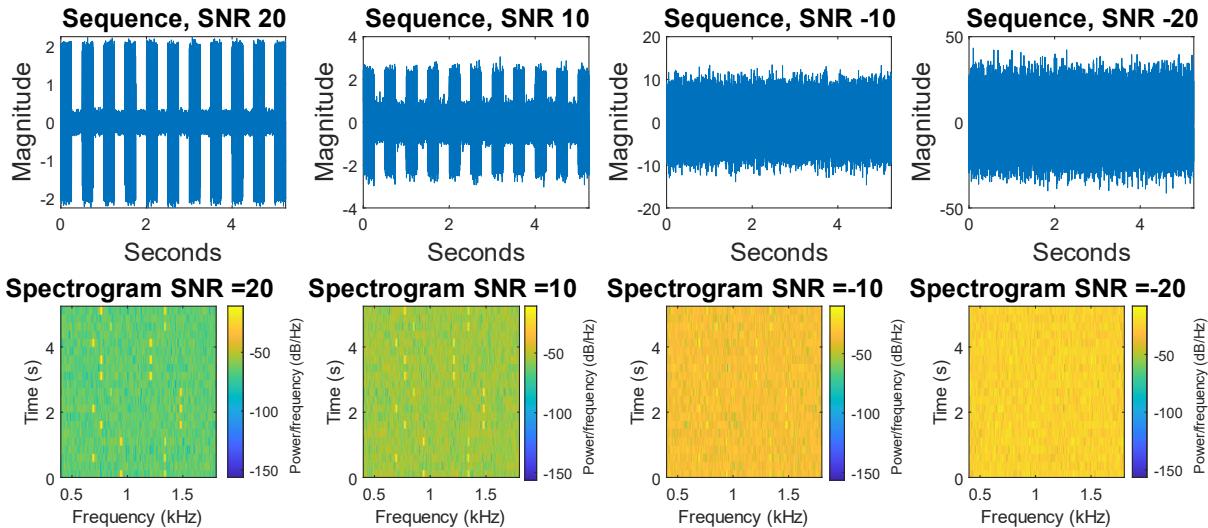
### 3.4.2. Noise Addition

The Matlab command `wgn` was used to add white noise to the signal considered. By varying the variance of the noise added, the signal to noise ratio also varies. The signal to noise ratio is described by:

$$SNR = 10 \times \log_{10} \left( \frac{\sigma_y^2}{\sigma_w^2} \right) \quad (56)$$

If we therefore consider 4 different signal to noise ratios of  $\text{SNR} = [20, 10, -10, -20]$ , the variance of the noise considered in each case will be 0.07, 0.23, 2.3 and 7.2 respectively.

Figure 3.22 shows sequence  $y$  with added white noise, its spectrogram and its FFT for increasing noise variance. As expected, as we increase the variance of the noise, the SNR increases. As we can see the signal can be better distinguished using the spectrogram representation, since the yellow bands are still present at the frequencies of interest. When analysing the FFT, if the variance increases too much the frequency peaks start to be less prominent compared to the noise, and therefore can't be recognised accurately, such as when the  $\text{SNR} = -20$ . Therefore, we can conclude that the frequency analysis of the signal can withstand a high amount of noise and it is therefore very useful for key classification in the presence of noise.



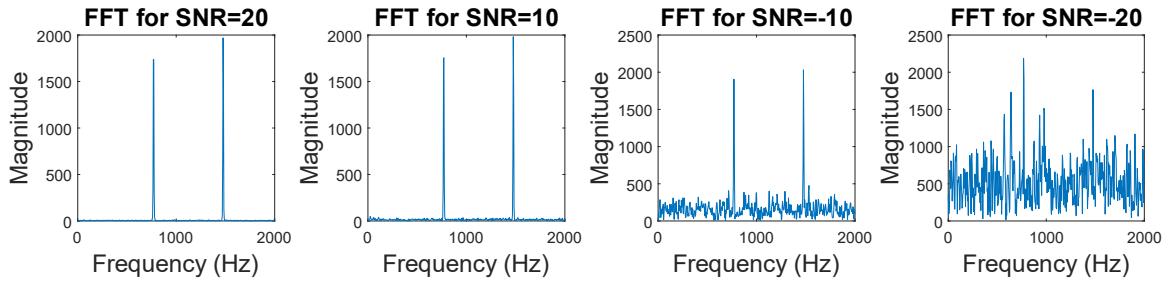


Figure 3.22: Spectrogram, FFT and time domain signal for different values of SNR.

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

The standard and average periodogram have been plotted in figure 3.23. The average periodogram has been plotted with window lengths of values: 50, 100 and 150 seconds.

By observing figure 3.23, clear spikes can be distinguished in the average periodogram for each trial, these are the frequencies of oscillation. As the window length increases, the variability of the estimate decreases since the width of the peaks decreases. However, the variability of the averaged periodograms is less than the variability of the standard periodogram, that shows an oscillating behaviour around the theoretical value.

A low heart rate corresponds to a large PSD in magnitude, whereas a higher heart rate corresponds to a smaller PSD. Therefore, the lowest heart rate occurs in trial 3 when constrained breathing at 15bpm takes place and the highest heart rate occurs in trial 2 constrained breathing at 50bpm takes place.

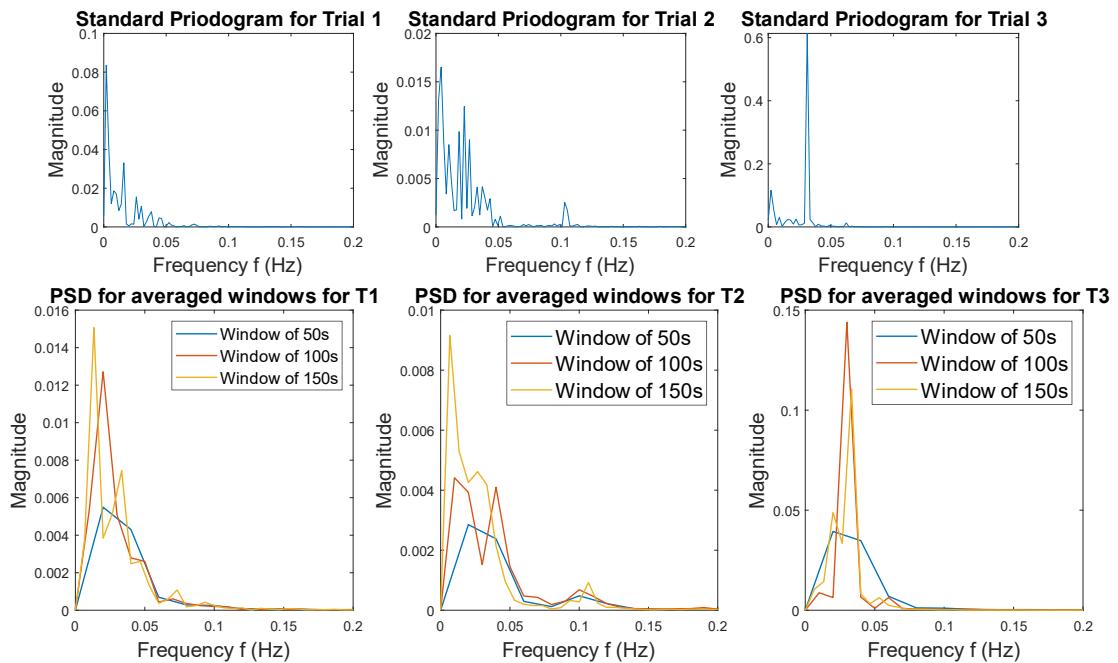


Figure 3.23: Plot of the standard and averaged periodograms for trial 1, 2 and 3. The window lengths of 50, 100 and 150 seconds have been used to calculate the average periodogram in each case.

## 4. Optimal filtering - fixed and adaptive

### 4.1. Weiner Filter

A white gaussian noise sequence was generated and was used as an input to a filter with coefficients  $\mathbf{b} = [1, 2, 3, 2, 1]$  and  $\mathbf{a} = [1]$ , in order to obtain an output  $\mathbf{y}$ . White gaussian noise with a standard deviation of 0.1 was then added to the output and the signal to noise ratio was calculated using the Matlab command `snr`. The results obtained were  $\text{SNR} \approx 20$ .

#### 4.1.1. Optimal coefficients of the Weiner filter

In order to calculate the optimal coefficients,  $\mathbf{R}_{xx}$  and  $\mathbf{p}_{zx}$  were also calculated and the following formula was used:  $\mathbf{w}_{opt} = \mathbf{R}_{xx}^{-1} \mathbf{p}_{zx}$

The values obtained for the scaled coefficients were:  $\mathbf{w}_{opt} = [1.0008, 1.9811, 2.9838, 2.0017, 1.0063]$ . These values are very close to the filter of the unknown system coefficient values.

The experiment was repeated for different variance values in the range of  $\sigma^2 \in [0.1, 10]$ . The results of the optimal weights and the SNR calculated for each variance level is shown in table 4.

variance	Filter coefficients					SNR
	1	2	3	4	5	
0.1	1.0136	1.9919	2.9983	2.0142	1.0170	20.2345
1	0.8997	2.1258	3.2665	2.0884	0.9116	-0.0620
3	1.2640	2.5419	3.2690	2.0302	1.2009	-9.3679
6	1.2811	1.4505	3.1152	3.4810	1.7376	-15.3579
8	0.3106	1.3217	1.8697	1.0494	-0.7326	-18.0682
10	3.2684	4.3521	1.8434	1.4911	0.1835	-19.6371

Table 4: Effect of varying noise on the optimal coefficients of the Weiner filter and the signal to noise ratio

Moreover, we increased the value of  $N_w$  to 4 and the results are shown in table 5.

$N_w$	Filter coefficients						SNR
	1	2	3	4	5	6	
5	1.0042	2.0148	3.0229	1.9709	0.9839	-0.0053	-0.0009
8	1.0050	2.0140	3.0219	1.9714	0.9836	-0.0051	0.0017

Table 5: Average variance of the weights for increasing noise.

If the value of  $N_w$  is above 4, the extra coefficients are small and do not carry important information. The accuracy of the estimators doesn't increase and the process becomes more computationally expensive, hence they shouldn't be included.

#### 4.1.2. Computational complexity of the Weiner solution

In order to calculate the Weiner solution we need to calculate  $\mathbf{R}_{xx}$  and  $\mathbf{p}_{zx}$ , and to calculate them we need to perform a number of multiplications and additions of  $N \times (N_w + 1)^2$  and  $N \times (N_w + 1)$  respectively. The inverse of  $\mathbf{R}_{xx}$  requires  $(N_w + 1)^3$  calculations and the multiplication between  $\mathbf{p}_{zx}$  and  $\mathbf{R}_{xx}^{-1}$  requires in total  $(N_w + 1)^2$  additions and multiplications since each element has to be taken into consideration. Hence, for the case considered of  $N = 1000$  and  $N_w = 4$ , we obtain:

$$(2 \times N \times (N_w + 1)^2) + (2 \times N \times (N_w + 1)) + (N_w + 1)^3 + (2 \times (N_w + 1)^2) \approx 60000$$

We approximately obtain 60000 operations proving the fact that the Weiner solution requires a high computational complexity.

#### 4.2. The Least Mean Square (LMS) algorithm

A Matlab function called `lms` was programmed in order to calculate the LMS estimate as well as the error between the estimate and the values of the optimal weights. Figure 4.1 shows the plot of the LMS weights for increasing step size ( $\mu$ ).

As we can see from figure 4.1, as the value of  $\mu$  increases, the time the coefficients take to converge towards the optimal value decreases. However, if  $\mu$  increases too much, the step size will be too large and therefore the system will diverge and will never converge towards the optimal values.

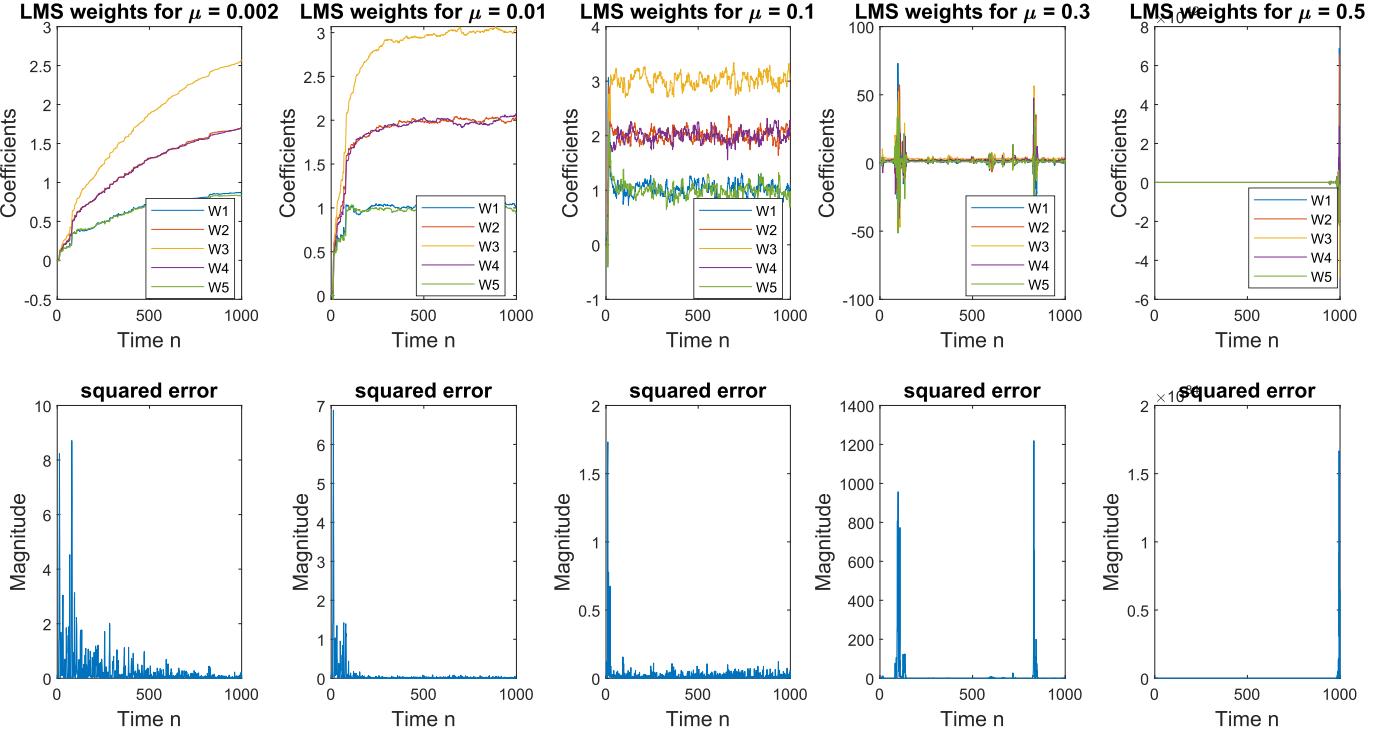


Figure 4.1: Plots of the LMS weights for different values of adaptive gain and the squared error of the estimation for each.

Moreover, looking at the squared error plots we can see that if the time of convergence is relatively small, the variance increases. Therefore, a trade-off exists between how fast the system converges and the variance of the estimate.

The critical gain defines the stability of the system, if  $\mu$  is smaller the system is stable, but if it surpasses the critical gain it becomes unstable. We can estimate the critical gain to be approximately 0.37.

#### 4.2.1. Computational complexity of the LMS solution

In order to determine the computational complexity of the LMS solution, the number of operations required to obtain  $\hat{y}[n], w[n]$  and  $e[n]$  have to be calculated. The calculation of  $\hat{y}[n]$  requires  $N_w + 1$  multiplications and  $N_w$  additions and to obtain  $w[n]$ ,  $2 \times (N_w + 1)$  multiplications and  $N_w + 1$  sums must be computed. To calculate  $e[n]$ , only one subtraction is needed. Therefore, the total number can be calculated:

$$4 \times (N_w + 1) + 1 + N_w \approx 25$$

if  $N_w = 4$ , the total number of operations required can be approximated to 25. Hence, the LMS solution is significantly less computationally complex with respect to the Weiner solution.

### 4.3. Gear Shifting

Gear shifting has been applied to reduce the adaptation gain in time in order to obtain improved estimates in the steady state. Since the adaptive gain is no longer constant for all time instants, even if the initial step size is too big, the system converges eventually as shown in figure 4.5. The maximum permissible overshoot for each coefficient has been assumed to be 20% of its true value.

If the error is too big → the adaptive gain is increased by 15% of its current value

If the error is too small → the adaptive gain is decreased by 15% of its current value

If the error is in the range considered  $\rightarrow$  the adaptive gain maintains a constant value.

The geared shifted version of the LMS variance presents less variance as the values of the weights converge to the optimal value as shown in figure 4.2. The squared error was plotted in figure 4.3, and as we can observe, the gear shifted version shows a higher squared error at the beginning but then it shows a slightly lower error as time goes by.

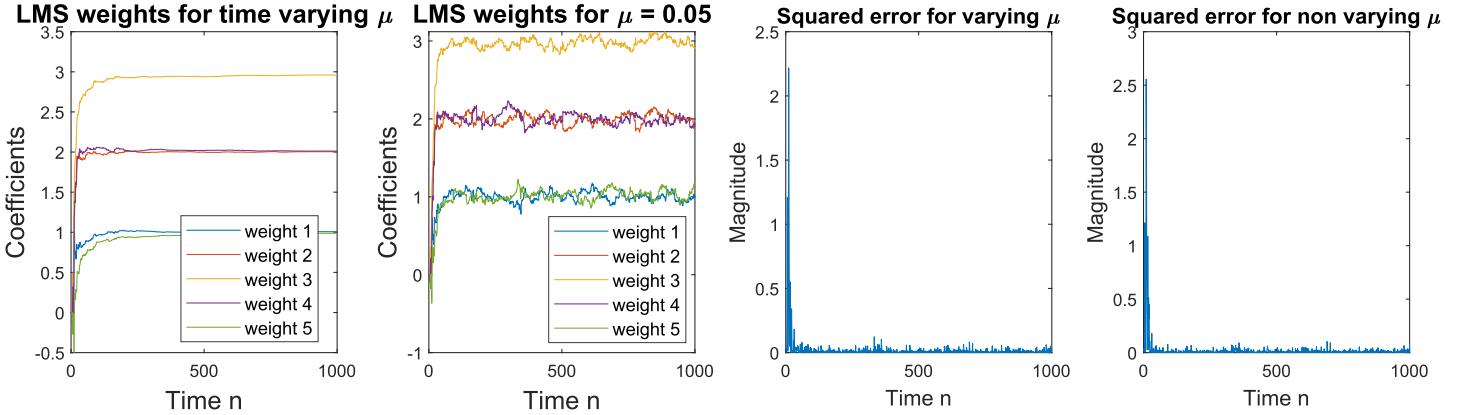


Figure 4.2: Plot of the LMS weights for constant ( $\mu = 0.05$ ) and varying  $\mu$ .

Figure 4.3: Plot of the squared error for constant and varying  $\mu$ .

For a step size of 0.3, without gear shifting the system would diverge since  $\mu$  would be bigger than the critical gain as shown in figure 4.5. However, when using gear shifting, since the step size is modified according to the error, the system manages to converge towards the optimal weight values, although with a lot of variance at the beginning.

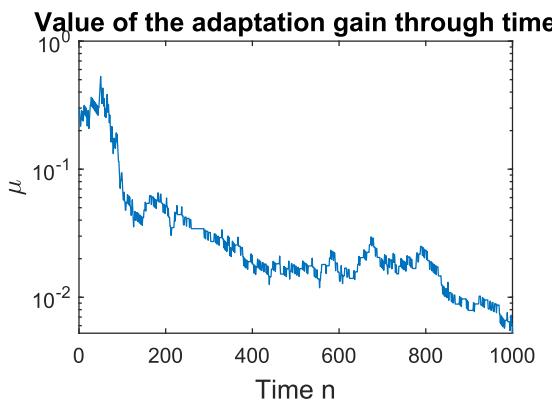


Figure 4.4: Plot of the evolution of the step size value in time.

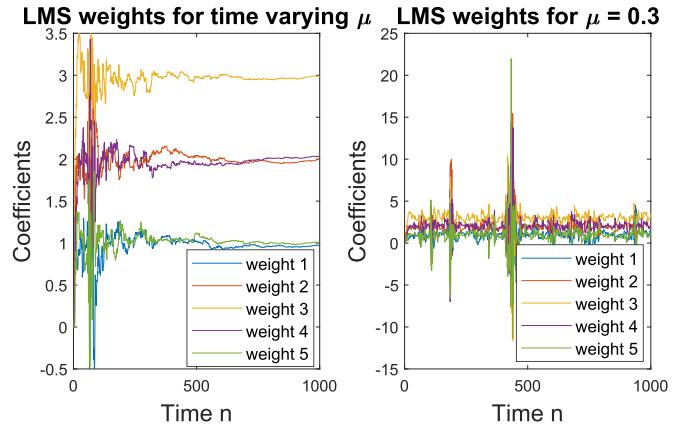


Figure 4.5: Plot of the LMS weights for constant ( $\mu = 0.3$ ) and varying  $\mu$ .

#### 4.4. Identification of an AR process

The parameters of the AR(2) filter used are  $\mathbf{a} = [1 \ 0.9 \ 0.2]$ , hence the expected value of convergence for the coefficients correspond to -0.9 and -0.2 respectively. Moreover, the convergence of the coefficients was plotted for 4 different step size values  $\mu = [0.01, 0.02, 0.05, 0.1]$ .

As we can see from figure 4.6, if the step size is small the values take longer to converge to their theoretical values, however there is less variability in the estimate. Hence, when the step size is larger the values converge faster but have a higher variability and therefore are less accurate. If the step size is too large, it leads to instability as we can observe for  $\mu = 0.1$ .

In conclusion, in order to find an optimal step size we want low variability in the estimate and a high rise time. From the step sizes investigated we can therefore identify  $\mu = 0.02$  as the optimal one.

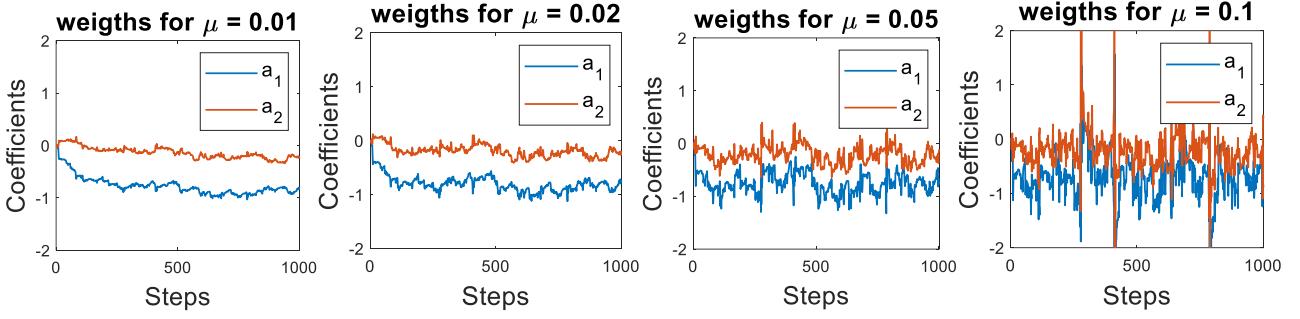


Figure 4.6: Plot of LMS coefficients for an AR(2) model and for different values of  $\mu = [0.01, 0.02, 0.05, 0.1]$

#### 4.5. Speech recognition

By taking a small number of samples from a speech recording we are able to analyse the data in a very short time frame that corresponds to 2.27ms. Hence, we can assume quasi-stationarity for all statistical properties. 5 different speech recordings that correspond to the letters ‘e’, ‘a’, ‘s’, ‘t’ and ‘x’ were considered and the recording function inside Matlab was used to obtain them. A sampling frequency of 44.1kHz was used and the Matlab command `getaudiodata` was used to obtain the data from the recordings that were later truncated to 1000 samples that were inputted to the predictor.

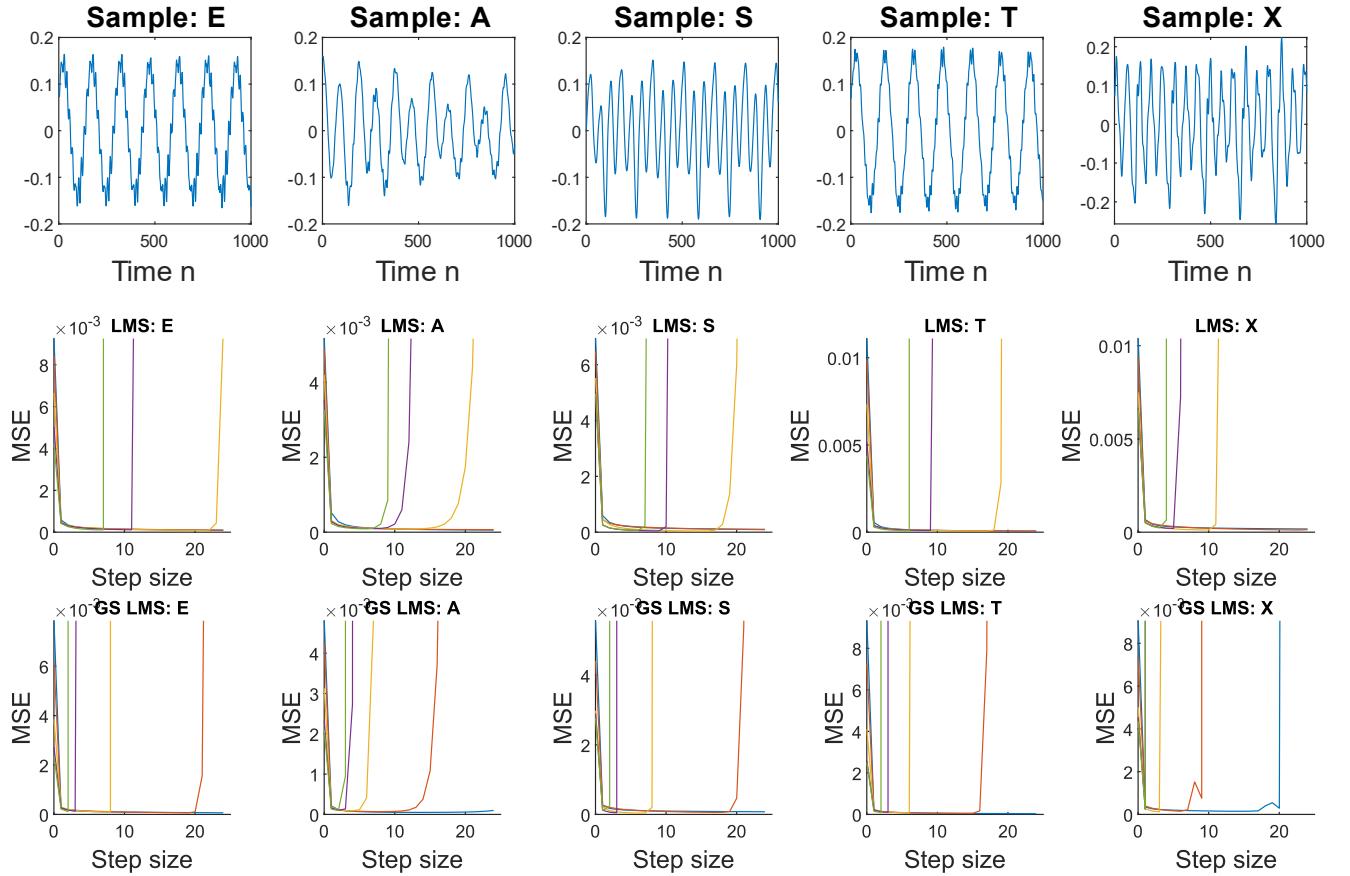


Figure 4.7: Plot of the time domain signal (row 1), plot of the MSE calculated using the LMS method (row 2) and plot of the MSE calculated using gear shifting for each letter.

The mean squared error was plotted for different model orders  $\mathbf{p} = [1, 2, 5, 10, 15]$  and step sizes in figure 4.7. Moreover, gear shifting was also considered. We observe that generally the error decreases with the step size until the critical step size is reached and hence the system diverges. Comparing lower with higher model orders we observe that generally the higher model orders produce better the prediction, however, they reach the critical step size sooner than lower model orders.

When gear shifting is applied, the error decreases in general and therefore a better prediction is achieved. However, a trade off exists between a good prediction and early divergence of the system since the critical step size for all orders will be reached much sooner. This behaviour could be due to the quasi-stationarity nature of the data since having a constant step size increases the tolerance for noise enabling it to predict slightly non-stationary signals such as the ones considered.

Furthermore, by looking at figure 4.7 we can identify the optimal step size for each letter by finding the maximum value of step size for which the MSE is minimal. These values are displayed in table 6.

<i>letter</i>	<i>e</i>	<i>a</i>	<i>s</i>	<i>t</i>	<i>x</i>
<i>Optimal <math>\mu</math></i>	5	5	4	3	3

Table 6: Optimal step size for each signal.

#### 4.5.1. Optimal Filter Length

The optimal filter length for each letter can be found assuming that the output of the predictor corresponds to an AR model. Hence, we can identify the ideal model order by calculating the PAC as well as the MDL, AIC and AICc. The optimal step size identified for each letter in the previous question was used and a 95% threshold was considered when analysing the PAC.

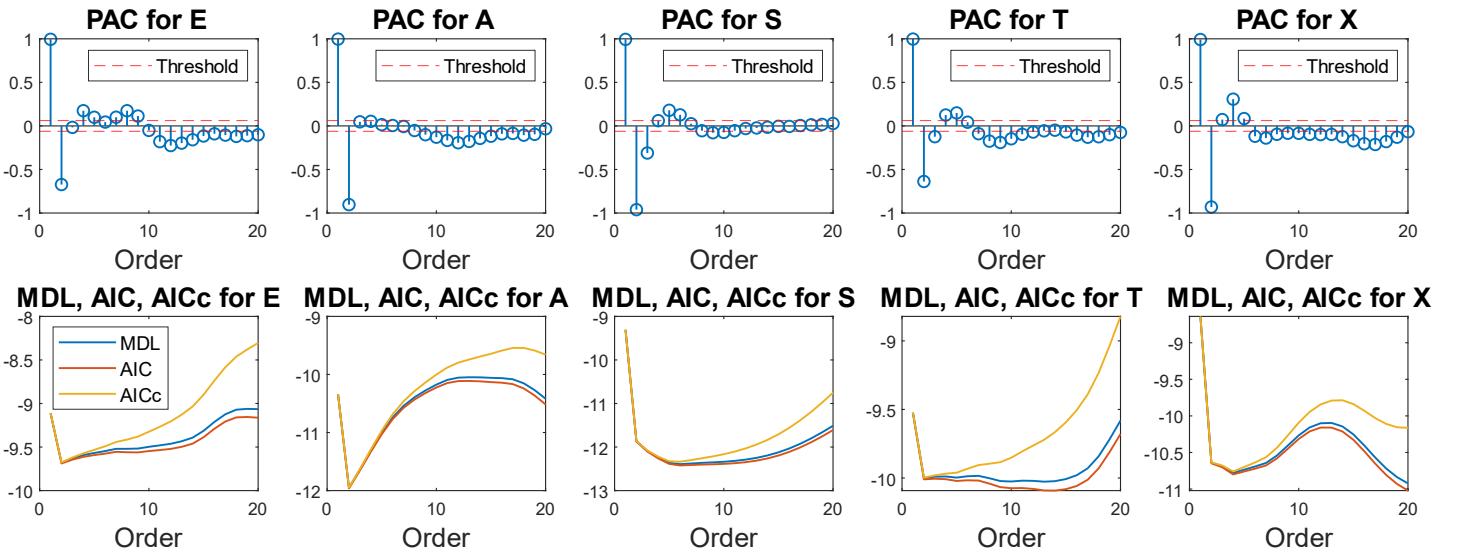


Figure 4.8: Plot of the PAC, MDL, AIC and AICc for each signal.

From figure 4.8 we can therefore identify the optimal model order for each letter. The results obtained can be seen in table 7 and although for some letters both methods do not yield to the same value they do not differ significantly. This is because the PAC especially takes into consideration the computational complexity of the model, whereas the MDL, AIC and AICc take more into consideration the prediction error.

<i>letter</i>	<i>e</i>	<i>a</i>	<i>s</i>	<i>t</i>	<i>x</i>
<i>PAC</i>	4	2	3	5	4
<i>MDL, AIC, AICc</i>	4	2	5	4	4

Table 7: Optimal filter length according to the PAC, MDL, AIC and AICc for each letter.

#### 4.5.2. Prediction Gain

The performance of each predictor can be assessed by computing its predictor gain that is given by:

$$R_p = 10 \times \log \left( \frac{\sigma_x^2}{\sigma_e^2} \right)$$

Where  $\sigma_x^2$  and  $\sigma_e^2$  denote the variance of the input and the error signals respectively. The prediction gain was plotted for all signals using two different sampling frequencies (44.1kHz and 16kHz) in figures 4.9 and 4.10.

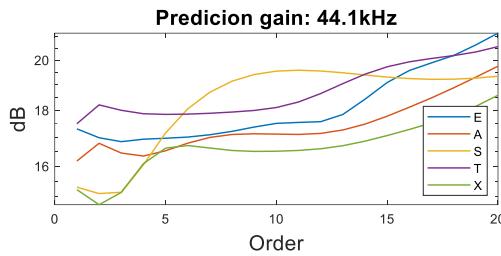


Figure 4.9: Plot of the prediction gain for each of the signals with a sampling frequency of 44.1kHz.

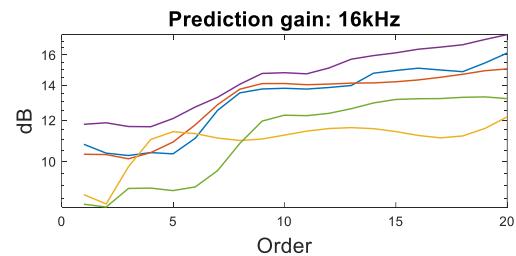


Figure 4.10: Plot of the prediction gain for each of the signals with a sampling frequency of 16kHz.

FFT of all letters

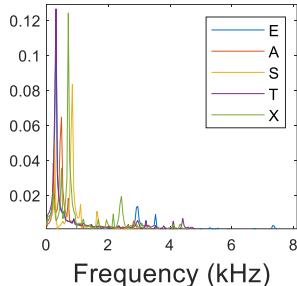


Figure 4.11: Plot of the FFT of all the signals.

From figures 4.9 and 4.10 we can observe that a decrease in the sampling rate will cause a decrease in the prediction gain and therefore in the accuracy of the predictor as well. Approximately all signals present very similar prediction gains since they are all in the range of 15-20 db before being downsampled.

Looking at figure 4.11 we can appreciate that letters ‘S’ and ‘X’ are located at the highest frequency values. This is why, when the signals are downsampled, these two letters lower their prediction gain more than the others.

#### 4.6. Dealing with computational complexity: Sign Algorithms

A Matlab function was programmed to compute a simplified version of the LMS algorithm, called sign LMS algorithms. In order to compare the standard LMS algorithm with the sign algorithms, the filter coefficients of exercise 4.4 were estimated for different step sizes in figure 4.12. The filter coefficients correspond to 0.9 and 0.2, hence  $a_1$  and  $a_2$  should converge to -0.9 and -0.2 respectively.

The best coefficient estimates are achieved with the standard LMS and sign-error algorithms, the others present less variability in the estimate, however the values they converge to differ more from the theoretical expectations. Furthermore, when the step size is increased to 0.1 the standard LMS algorithm diverges whereas the sign algorithms manage to converge to a value, although it slightly differs from the theoretical one.

In conclusion, the sign algorithms have very fast computations but, the update mechanism is degraded compared to standard LMS algorithm due to the quantization of gradient estimates. This is why, their steady state error increases at the cost of decreasing the convergence rate.

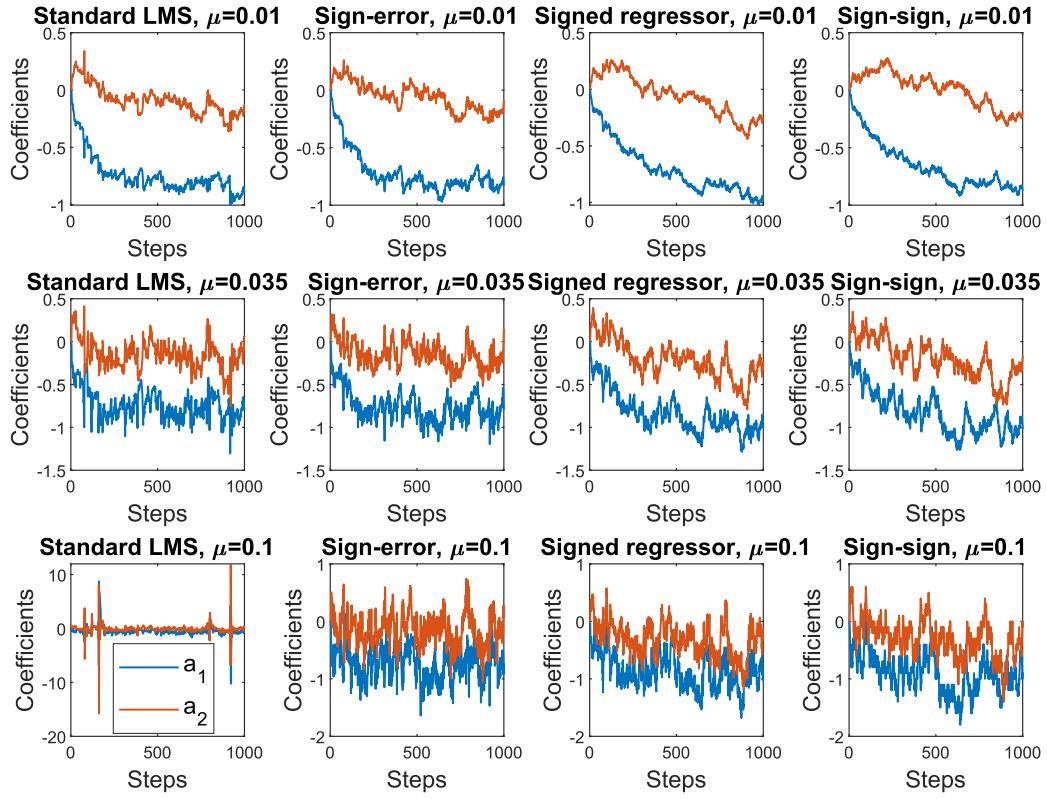


Figure 4.12: Plots of the filter coefficients ( $a_1, a_2$ ) estimated with the standard LMS algorithm and different sign algorithms for different step sizes.

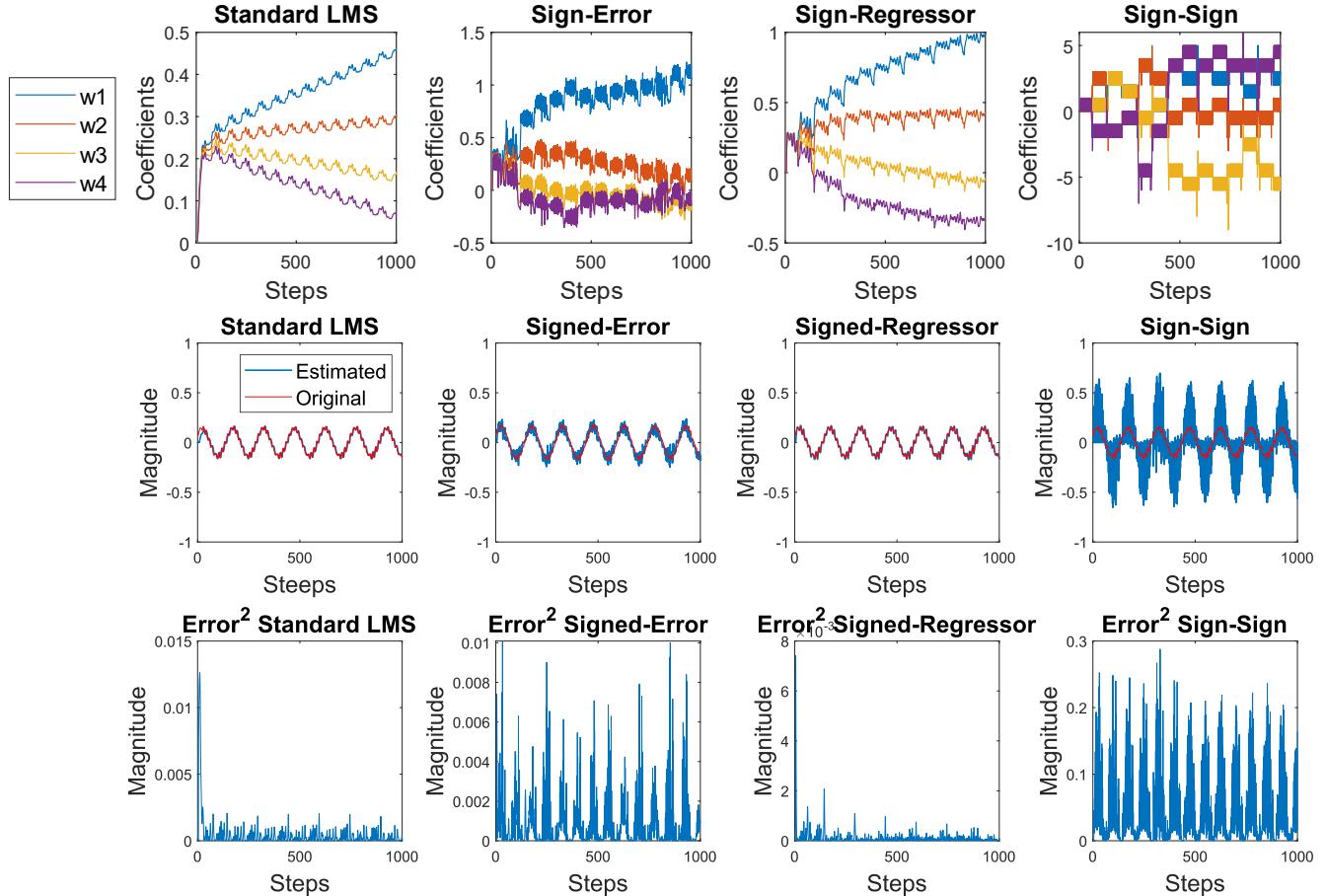


Figure 4.13: Plot of the prediction and weights computed with different algorithms and of the original signal 'E' alongside their respective squared errors.

Moreover, the prediction for the signal corresponding to the letter ‘E’ was computed and plotted alongside the signal in figure 4.13 for a step size of  $\mu = 1$  and a model order of  $p = 4$ . The squared error and the estimated weights for each prediction were also plotted.

The best prediction algorithm is the signed-regressor as its squared error never exceeds 0.008 and the estimated weights converge towards steady state with low variability. Moreover, although with a high initial error, the standard LMS algorithm also achieves an accurate prediction, however the estimated weights take longer to stabilize. Both of these algorithms produce a prediction that closely matches the original signal.

The signed-error and the sign-sign algorithms present a constant error, although the latter one achieves higher error values of approximately 0.29. Both of these algorithms produce a prediction that differs from the original signal and their estimated weights have a high variability and diverge slightly.

## 5. MLE for the Frequency of a Signal

The Maximum Likelihood Estimate (MLE) estimates an unknown parameter such that for this value the probability of obtaining an actually observed sample is as large as possible. The MLE can be found by minimizing:

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

Since  $c = [1, \cos(2\pi f_0), \dots, \cos(2\pi f_0(N-1))]^T$  and  $s = [1, \sin(2\pi f_0), \dots, \sin(2\pi f_0(N-1))]^T$  we can therefore simplify equation X further:

$$J(\boldsymbol{\theta}) = (\mathbf{x} - A \cos(\Phi) \mathbf{c} - A \sin(\Phi) \mathbf{s})^T (\mathbf{x} - A \cos(\Phi) \mathbf{c} - A \sin(\Phi) \mathbf{s}) \quad (57)$$

Let  $\alpha_1 = \cos(\Phi)$ ,  $\alpha_2 = \sin(\Phi)$ ,  $\mathbf{H} = [\mathbf{c}, \mathbf{s}]$  and  $\boldsymbol{\alpha} = [\alpha_1, \alpha_2]^T$ :

$$J(\boldsymbol{\theta}) = J(\alpha_1, \alpha_2, f_0) = (\mathbf{x} - \alpha_1 \mathbf{c} - \alpha_2 \mathbf{s})^T (\mathbf{x} - \alpha_1 \mathbf{c} - \alpha_2 \mathbf{s}) = (\mathbf{x} - \mathbf{H}\boldsymbol{\alpha})^T (\mathbf{x} - \mathbf{H}\boldsymbol{\alpha}) = J'(\boldsymbol{\alpha}, f_0) \quad (58)$$

In order to minimize  $J(\theta)$  and obtain  $\hat{\boldsymbol{\alpha}}$ , we set its derivative with respect to  $\alpha$  to zero:

$$\frac{\partial J}{\partial \alpha} \Big|_{\alpha=\hat{\boldsymbol{\alpha}}} = \frac{\partial}{\partial \alpha} (\mathbf{x}\mathbf{x}^T - 2\mathbf{x}^T \mathbf{H}\hat{\boldsymbol{\alpha}} + \hat{\boldsymbol{\alpha}}^T \mathbf{H}^T \mathbf{H}\hat{\boldsymbol{\alpha}}) = -2\mathbf{x}^T \mathbf{H} + 2\mathbf{H}^T \mathbf{H}\hat{\boldsymbol{\alpha}} = 0 \rightarrow \hat{\boldsymbol{\alpha}} = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{x} \quad (59)$$

Replacing  $\hat{\boldsymbol{\alpha}}$  into J:

$$\begin{aligned} J(\boldsymbol{\theta}) &= (\mathbf{x} - \mathbf{H}(\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{x})(\mathbf{x} - \mathbf{H}(\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{x})^T = ((\mathbf{I} - \mathbf{H}(\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T)\mathbf{x})((\mathbf{I} - \mathbf{H}(\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T)\mathbf{x})^T \\ &= \mathbf{x}(\mathbf{I} - \mathbf{H}(\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T)\mathbf{x}^T (\mathbf{I} - \mathbf{H}(\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T)^T = \mathbf{x}(\mathbf{I} - \mathbf{H}(\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T)\mathbf{x}^T = \mathbf{x}^T \mathbf{x} - \mathbf{x}^T \mathbf{x} \mathbf{H}(\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \end{aligned}$$

Therefore, when  $\mathbf{H}(\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{x} \mathbf{x}^T$  is maximized,  $J(\boldsymbol{\theta})$  is minimized.

Using the definition given of  $\mathbf{H}$ ,  $\mathbf{H} = [\mathbf{c}, \mathbf{s}]$ , it can be proven that  $\mathbf{H}(\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{x} \mathbf{x}^T$  is equal to:

$$\begin{pmatrix} \mathbf{c}^T \mathbf{x} \\ \mathbf{s}^T \mathbf{x} \end{pmatrix}^T \begin{pmatrix} \mathbf{c}^T \mathbf{c} & \mathbf{c}^T \mathbf{s} \\ \mathbf{s}^T \mathbf{c} & \mathbf{s}^T \mathbf{s} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{c}^T \mathbf{x} \\ \mathbf{s}^T \mathbf{x} \end{pmatrix} \quad (60)$$

Since:

$$\mathbf{x}^T \mathbf{H} = (x_0 \ x_1 \ \dots \ x_{N-1}) \begin{pmatrix} c_0 & s_0 \\ c_1 & s_1 \\ \vdots & \vdots \\ c_{N-1} & s_{N-1} \end{pmatrix} = \begin{pmatrix} \mathbf{c}^T \mathbf{x} \\ \mathbf{s}^T \mathbf{x} \end{pmatrix}^T \quad (61)$$

$$\mathbf{H}^T \mathbf{H} = \begin{pmatrix} c_0 & c_1 & \dots & c_{N-1} \\ s_0 & s_1 & \dots & s_{N-1} \end{pmatrix} \begin{pmatrix} c_0 & s_0 \\ c_1 & s_1 \\ \vdots & \vdots \\ c_{N-1} & s_{N-1} \end{pmatrix} = \begin{pmatrix} \mathbf{c}^T \mathbf{c} & \mathbf{c}^T \mathbf{s} \\ \mathbf{s}^T \mathbf{c} & \mathbf{s}^T \mathbf{s} \end{pmatrix} \quad (62)$$

$$\mathbf{H}^T \mathbf{x} = \begin{pmatrix} c_0 & c_1 & \dots & c_{N-1} \\ s_0 & s_1 & \dots & s_{N-1} \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{N-1} \end{pmatrix} = \begin{pmatrix} \mathbf{c}^T \mathbf{x} \\ \mathbf{s}^T \mathbf{x} \end{pmatrix} \quad (63)$$

Therefore, maximizing  $\mathbf{H}(\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{x} \mathbf{x}^T$  is equivalent to maximizing  $\begin{pmatrix} \mathbf{c}^T \mathbf{x} \\ \mathbf{s}^T \mathbf{x} \end{pmatrix}^T \begin{pmatrix} \mathbf{c}^T \mathbf{c} & \mathbf{c}^T \mathbf{s} \\ \mathbf{s}^T \mathbf{c} & \mathbf{s}^T \mathbf{s} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{c}^T \mathbf{x} \\ \mathbf{s}^T \mathbf{x} \end{pmatrix}$ .

This expression is equivalent to:  $\begin{pmatrix} \mathbf{c}^T \mathbf{x} \\ \mathbf{s}^T \mathbf{x} \end{pmatrix}^T \begin{pmatrix} \frac{N}{2} & 0 \\ 0 & \frac{N}{2} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{c}^T \mathbf{x} \\ \mathbf{s}^T \mathbf{x} \end{pmatrix}$ . This can be proven by computing the following expressions:

$$\mathbf{s}^T \mathbf{c} = \mathbf{c}^T \mathbf{s} = \frac{1}{2} \sum_{n=0}^{N-1} \sin(4\pi f_0 n) \quad (64)$$

$$\mathbf{c}^T \mathbf{c} = \frac{1}{2} \sum_{n=0}^{N-1} (\cos(4\pi f_0 n) + 1) \approx \frac{1}{2} \sum_{n=0}^{N-1} (0 + 1) = \frac{N}{2} \quad (65)$$

$$\mathbf{s}^T \mathbf{s} = \frac{1}{2} \sum_{n=0}^{N-1} (1 - \cos(4\pi f_0 n)) \approx \frac{1}{2} \sum_{n=0}^{N-1} (1 - 0) = \frac{N}{2} \quad (66)$$

This is only true if  $f_0 \neq \frac{1}{2}$  or  $f_0 \neq 0$ . If  $f_0 \approx \frac{1}{2}$  or 0,  $\cos(4\pi f_0 n) \approx \cos(0) \approx 1$  and or  $\cos(4\pi f_0 n) \approx \cos(2\pi n) \approx 1$ , these results would yield to  $\mathbf{c}^T \mathbf{c} = N$  and  $\mathbf{s}^T \mathbf{s} = 0$ .

The matrix seen before can be rewritten to:  $\begin{pmatrix} \mathbf{c}^T \mathbf{x} \\ \mathbf{s}^T \mathbf{x} \end{pmatrix}^T \begin{pmatrix} \frac{N}{2} & 0 \\ 0 & \frac{N}{2} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{c}^T \mathbf{x} \\ \mathbf{s}^T \mathbf{x} \end{pmatrix} = \begin{pmatrix} \mathbf{c}^T \mathbf{x} \\ \mathbf{s}^T \mathbf{x} \end{pmatrix}^T \begin{pmatrix} \frac{2}{N} & 0 \\ 0 & \frac{2}{N} \end{pmatrix} \begin{pmatrix} \mathbf{c}^T \mathbf{x} \\ \mathbf{s}^T \mathbf{x} \end{pmatrix}$ . This is equal to:

$$\begin{aligned} \frac{2}{N} (\mathbf{c}^T \mathbf{x})^2 + \frac{2}{N} (\mathbf{s}^T \mathbf{x})^2 &= \frac{2}{N} \left[ \left( \sum_{n=0}^{N-1} x[n] \cos(4\pi f_0 n) \right)^2 + \left( \sum_{n=0}^{N-1} x[n] \sin(4\pi f_0 n) \right)^2 \right] = \\ &= \frac{2}{N} \left[ \left( \sum_{n=0}^{N-1} x[n] (\cos(4\pi f_0 n) - j \sin(4\pi f_0 n)) \right) \left( \sum_{n=0}^{N-1} x[n] (\cos(4\pi f_0 n) + j \sin(4\pi f_0 n)) \right) \right] = \\ &= \frac{2}{N} [(\sum_{n=0}^{N-1} x[n] e^{-j2\pi f_0 n})(\sum_{n=0}^{N-1} x[n] e^{j2\pi f_0 n})] = \frac{2}{N} |X(f_0)|^2 \end{aligned} \quad (67)$$

Where  $X(f_0)$  corresponds to the Discrete-Time Fourier Transform (DTFT) of  $x[n]$ . Moreover,

$$\frac{2}{N} |X(f_0)|^2 = \frac{2}{N} |\sum_{n=0}^{N-1} x[n] e^{-j2\pi f_0 n}|^2 = 2 \hat{P}_x(f) \quad (68)$$

Hence,  $\mathbf{H}(\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{x} \mathbf{x}^T = 2 \hat{P}_x(f)$  and therefore the MLE ( $\hat{f}_0$ ) can be obtained by minimizing J or maximizing the periodogram.

The periodogram of the signal  $x[n] = \cos(4\pi f_0 n)$  was plotted for different frequency values as shown in figure 5.1. The frequencies chosen ( $\mathbf{f}_0 = [0.05, 0.1, 0.2, 0.3, 0.4, 0.45]$ ) range from 0 to 0.5 in order to analyse the behaviour of the MLE as  $f_0$  approaches these values. As predicted before, the value  $f_0$  approximately matches the maximum value of the periodogram that corresponds to its peak. As the value of  $f_0$

approaches either 0 or 0.5 the peak of the periodogram shifts no longer accurately matching the value of  $f_0$ . Moreover, the closer the value of  $f_0$  gets to the limits the higher the inaccuracy the plots show.

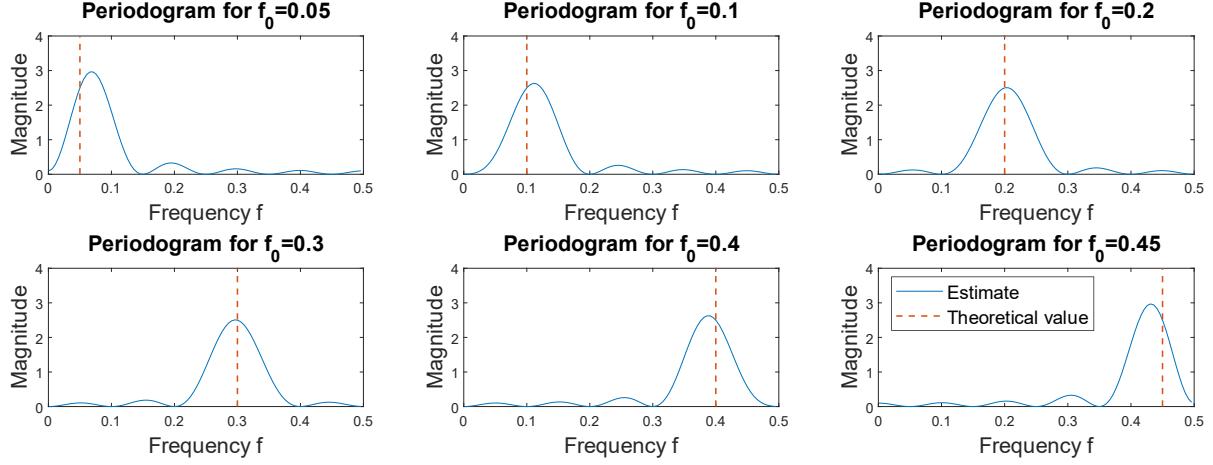


Figure 5.1: Plots of the periodogram of  $x[n]$  for  $f = [0.05, 0.1, 0.2, 0.3, 0.4, 0.45]$ .

Furthermore, the mean squared error was plotted in figure 5.2 to further analyse the accuracy of the estimation. As expected, we observe higher error values as  $f_0$  approaches the edges of the plot, the maximum error reached is approximately 0.8. For values of  $f_0$  in between 0.1 and 0.4 we observe an error ranging from 0 to approximately 0.2. oscillations due to harmonic

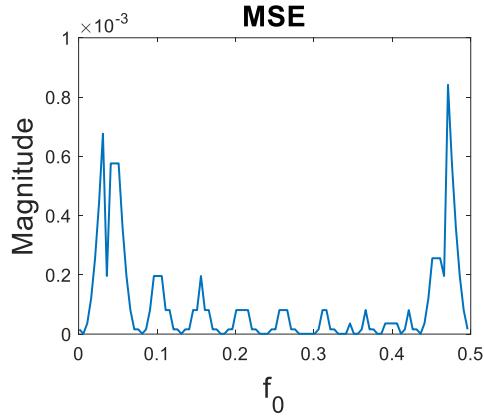


Figure 5.2: MSE of the maximum frequency of the periodogram compared to the corresponding value  $f_0$ .