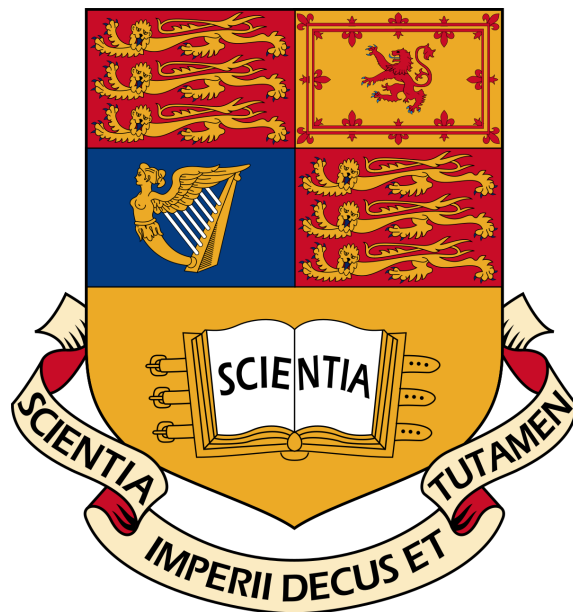


IMPERIAL COLLEGE LONDON  
DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

---

# Advanced Signal Processing

---



*Author*  
RICARDO VERA

*CID*  
01061547

# Contents

<b>1</b>	<b>Random Signals and Stochastic Processes</b>	<b>2</b>
1.1	Statistical Estimation . . . . .	2
1.2	Stochastic Processes . . . . .	4
1.3	Estimation of probability distributions . . . . .	6
<b>2</b>	<b>Linear Stochastic Modelling</b>	<b>8</b>
2.1	ACF of uncorrelated and correlated sequences . . . . .	8
2.2	Cross-correlation function . . . . .	9
2.3	Autoregressive modelling . . . . .	10
2.4	Cramer-Rao Lower Bound . . . . .	14
2.5	Real world signals: ECG from iAmp experiment . . . . .	17
<b>3</b>	<b>Spectral Estimation Modelling</b>	<b>20</b>
3.1	Averaged Periodogram Estimates . . . . .	21
3.2	Spectrum of Autoregressive Processes . . . . .	21
3.3	The Least Squares Estimation (LSE) of AR Coefficients . . . . .	23
3.4	Spectrogram for time-frequency analysis: dial tone pad . . . . .	25
3.5	Real world signals: Respiratory sinus arrhythmia from RR-Intervals . . . . .	27
<b>4</b>	<b>Optimal Filtering - fixed and adaptive</b>	<b>29</b>
4.1	Wiener filter . . . . .	29
4.2	The least mean square (LMS) algorithm . . . . .	29
4.3	Gear shifting . . . . .	31
4.4	Identification of AR processes . . . . .	32
4.5	Speech recognition . . . . .	33
4.6	Dealing with computational complexity: sign algorithms . . . . .	35
<b>5</b>	<b>MLE for the Frequency of a Signal</b>	<b>37</b>

# 1 Random Signals and Stochastic Processes

Aims:

- To become acquainted with the generation of random signals in MATLAB.
- To investigate the effect of a linear system upon a random signal.
- To calculate and understand auto- and cross-correlation functions.

## 1.1 Statistical Estimation

### Statistical Estimation for Uniform Distribution

The signal under study is a 1000 sample vector  $x$  each being a realization of  $X \sim \mathcal{U}(0,1), \forall n$ . Due to their time-invariant statistical properties, they are called statistically stationary, i.e. their statistical properties do not change over time.

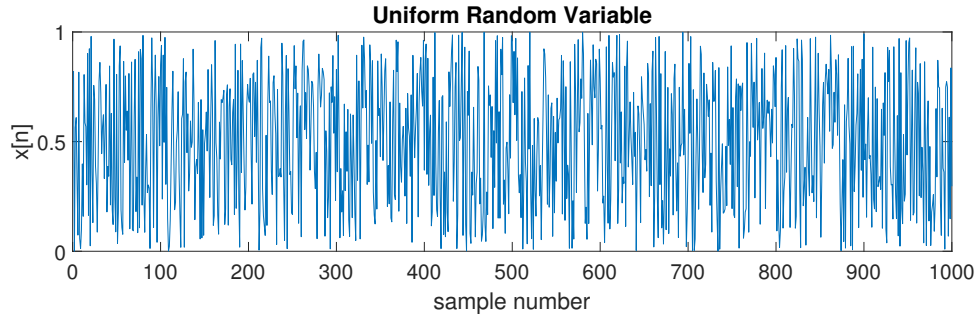


Figure 1: 1000 sample realization of uniform random variable

Theoretical mean and Standard Deviation:

$$\mu = \mathbb{E}\{X\} = \int_0^1 x dx = 0.5 \quad (1)$$

$$\sigma^2 = \mathbb{E}\{(X - \mu)^2\} = \mathbb{E}\{X^2\} - \mu^2 = \int_0^1 x^2 dx - \mu^2 = \frac{1}{12} \quad (2)$$

$$\text{Hence} \quad \sigma = \frac{1}{\sqrt{12}} \quad (3)$$

Calculation of the vector's mean and standard deviation give:  $\mu = 0.5044$  and  $\sigma = 0.2900$ . Where *mean* and *std* function as

$$\hat{m} = \frac{1}{N} \sum_{n=1}^N x[n] \quad \hat{\sigma} = \sqrt{\frac{1}{N-1} \sum_{n=1}^N (x[n] - \hat{m})^2} \quad (4)$$

The calculated values give an error of 0.0044 and 0.0013 respectively (0.88% and 0.45%) with respect to the theoretical values, with the error value decreasing as the number of samples increases.

The bias of the sample mean estimation is given by  $B = E\{X\} - \hat{m}$ . By generating an ensemble of ten 1000-sample realizations of  $X$ , it is possible to get a range of values for the distribution's mean and standard deviation, as seen in the following graphs.

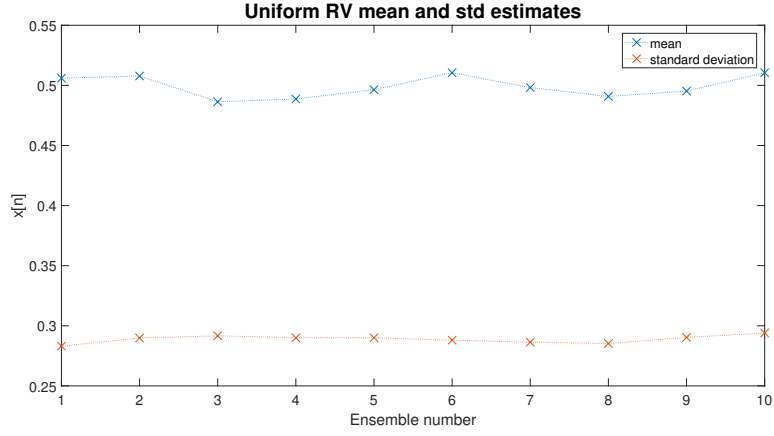
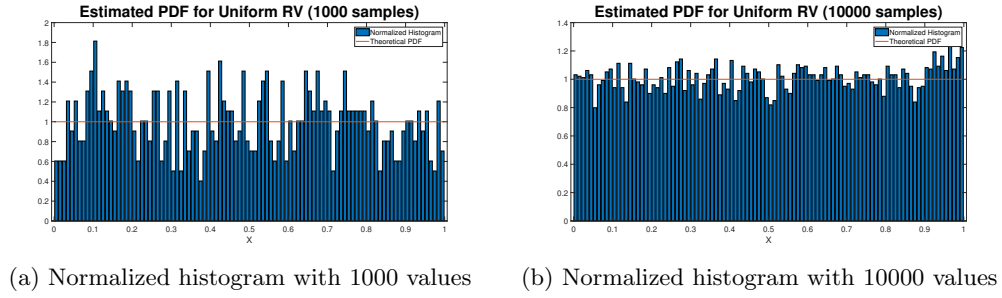


Figure 2: Mean and STD estimates for Uniform RV ensemble

As can be seen from Figure 2 the mean for the RV locates around 0.5 and the standard deviation around 0.28, matching with the theoretical values.

To obtain a complete statistical description it is needed to examine the probability density function. An approximation can be done by plotting a normalized histogram of the data.



(a) Normalized histogram with 1000 values (b) Normalized histogram with 10000 values

Figure 3: Uniform RV pdf estimation

As can be seen from the figures the normalized histogram converges to the pdf as the number of samples increases. Additionally the number of bins makes the estimation better as samples are more evenly distributed into more bins, hence reducing the error.

### Statistical Estimation for Normal Distribution

The same analysis will also be done for a vector composed of 1000 samples of Normal Gaussian distribution  $X \sim \mathcal{N}(0, 1)$ .

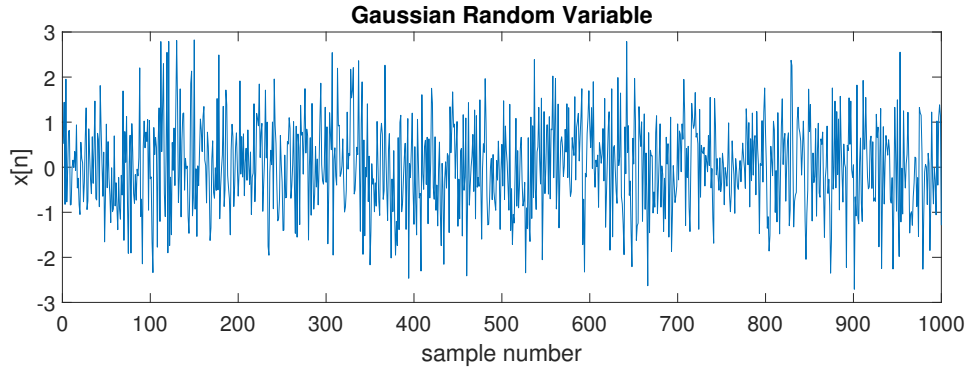


Figure 4: 1000 sample realization of a Gaussian RV

The theoretical mean and standard deviation are  $\mathbb{E}\{X\} = 0$  and  $\sigma = \sqrt{\mathbb{E}\{(X - \mu)^2\}} = 1$ . The equation for the probability density function is:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma^2} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (5)$$

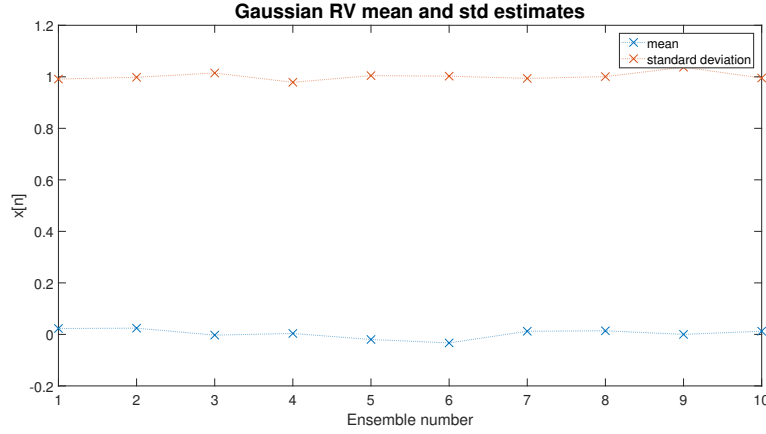
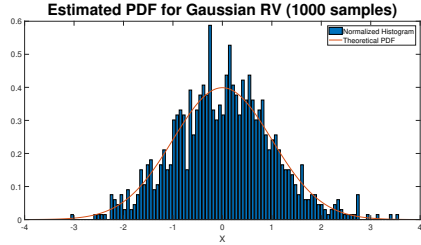


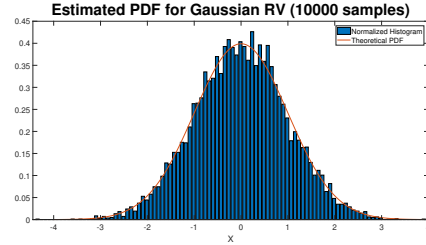
Figure 5: mean and standard deviation estimates

As can be seen again ensemble mean and standard deviation locate around 0 and 1, hence matching the theoretical values.

The functions *mean* and *std* give calculated  $\mu = -0.0125$  and  $\sigma = 1.0051$ . These give an error of 0.0125 and 0.0051 with respect to the theoretical values. Again an increase in the number of samples will improve the error, and by creating an ensemble of ten 1000-sample realizations, it is possible to get a range of values for the distribution's mean and standard deviation, as seen in Figure 5.



(a) Normalized histogram with 1000 values



(b) Normalized histogram with 10000 values

Figure 6: Gaussian RV pdf estimation

Finally, the estimation of the distribution's pdf ( $X \sim \mathcal{N}(0,1)$ ) via a normalized histogram shows how the realizations converge to the theory as the number of samples increases, as can be seen from the graphs.

## 1.2 Stochastic Processes

### Ensemble Mean and Standard Deviation as a function of time

Real world signals are normally modeled as stochastic processes, ergodic processes can be properly estimated using time averages, however non-ergodic processes don't necessarily match averages hence giving incorrect estimations. Signals can be distorted by random signals such as noise,  $y[n] = x[n] + w[n]$ , by taking averages of multiple realizations the aim is to achieve a high SNR

such that the estimation is as close to the original signal. For these  $SNR = \frac{\sigma_w^2}{\sigma_v^2} M$ , such that the number of realizations,  $M$ , is proportional to the SNR.

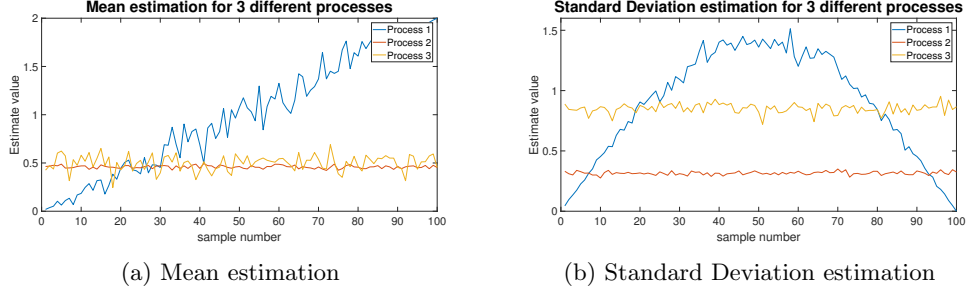


Figure 7: Ensemble estimations for the 3 different process

The figures above show the mean and standard deviation for the ensembles of the different processes ( $rp1$ ,  $rp2$ ,  $rp3$ ) for 100 realizations. From the graphs it can be seen how clearly the first process is not stationary as both its mean and standard deviation vary with time. The second and third processes the ensemble means and standard deviations are relatively constant. Both processes show a close to constant behavior around a value, with the expected distortion caused by noise.

### Mean and Standard Deviation Calculation for different processes

In order to value the ergodicity of each process a generation of 4 realizations of 1000 samples each is processed in order to obtain the mean and standard deviation for each realization.

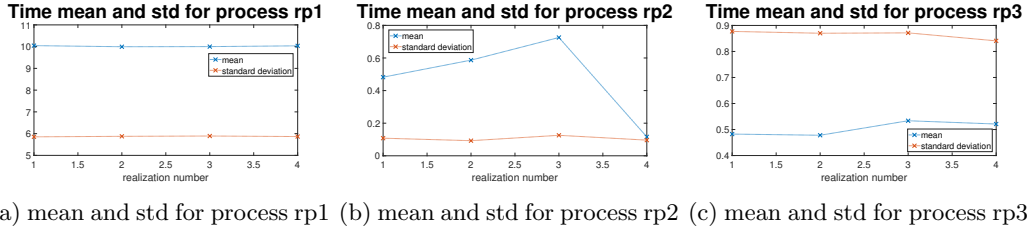


Figure 8: Ensemble estimations for the 3 different process

As can be seen from the figures, process 1 is clearly non-ergodic, as mentioned before the process is non-stationary and hence is not ergodic either, as the averages will vary with time. For processes 2 and 3 the figures somehow resemble the values obtained in the ensemble calculations (Figure 7). A comparison between process 2 and 3 shows how clearly process 3 varies significantly less than process 2 and hence would be considered more ergodic in comparison. This however is only relative to each other as an appropriate evaluation of the ergodicity of the processes would need a proper reference to quantify this. Additionally the use of only 4 realizations makes the calculation prone to larger error.

### Mathematical Description of different processes

For the derivation of the following calculations, it is known that for  $x \sim \mathcal{U}(0, 1)$ :

$$\mathbb{E}\{X\} = 0.5 \quad \text{Var}\{X\} = \frac{1}{12} \quad \mathbb{E}\{X^2\} = \mathbb{E}\{X\}\mathbb{E}\{X\} = \frac{1}{3} \quad (6)$$

Process 1: The expression is  $v[n] = an + b(X - 0.5) \sin(\frac{n\pi}{N})$ , where  $a = 0.02, b = 5$  and  $X \sim \mathcal{U}(0, 1)$

$$\begin{aligned}
\mathbb{E}\{v[n]\} &= \mathbb{E}\{an\} + \mathbb{E}\{b(X - 0.5) \sin(\frac{n\pi}{N})\} \\
&= an + b \sin(\frac{n\pi}{N}) \mathbb{E}\{X - 0.5\} \\
&= an \\
&= 0.02n
\end{aligned} \tag{7}$$

$$\begin{aligned}
Var\{v[n]\} &= \mathbb{E}\{v^2[n]\} - \mathbb{E}^2\{v[n]\} \\
&= \mathbb{E}\{a^2n^2 + b^2(X - 0.5)^2 \sin^2(\frac{n\pi}{N}) + 2abn(X - 0.5) \sin(\frac{n\pi}{N})\} - (an)^2 \\
&= \mathbb{E}\{b^2 \sin^2(\frac{n\pi}{N}) (X - 0.5)^2\} + \mathbb{E}\{2abn \sin(\frac{n\pi}{N}) (X - 0.5)\} \\
&= b^2 \sin^2(\frac{n\pi}{N}) \mathbb{E}\{X^2 - X + 0.25\} \\
&= (5)^2 \sin^2(\frac{n\pi}{N}) \left(\frac{1}{3} - 0.50 + 0.25\right) = \frac{25}{12} \sin^2(\frac{n\pi}{N})
\end{aligned} \tag{8}$$

As can be seen the theoretical values confirm that this process is not stationary as both the mean and standard deviation depend on  $n$ . It can be seen that additionally it matches with our plots giving a ramp and a sinusoidal-type curve.

Process 2: The expression is  $v[n] = X_1 + X_3X_2 - 0.5X_2$  where  $X_{1,2,3} \sim \mathcal{U}(0, 1)$

$$\begin{aligned}
\mathbb{E}\{v[n]\} &= \mathbb{E}\{X_1\} + \mathbb{E}\{X_3X_2\} - 0.5\mathbb{E}\{X_2\} \\
&= \mathbb{E}\{X\} + \mathbb{E}\{X\}\mathbb{E}\{X\} - 0.5\mathbb{E}\{X\} \\
&= 0.5 + 0.25 - 0.25 = 0.5
\end{aligned} \tag{9}$$

$$\begin{aligned}
Var\{v[n]\} &= \mathbb{E}\{v^2[n]\} - \mathbb{E}^2\{v[n]\} \\
&= \mathbb{E}\{X_1^2 + (X_3X_2)^2 + \frac{X_2^2}{4} + 2X_1X_2X_3 - X_1X_2 - X_3X_2^2\} \\
&= \mathbb{E}\{X_1^2 + (X_3 - 0.5)^2X_2^2 + 2X_1X_2(X_3 - 0.5)\} - 0.25 \\
&= \mathbb{E}\{X_1^2\} + \mathbb{E}\{(X_3^2 - X_3 + 0.25)X_2^2\} + 2\mathbb{E}\{X_1X_2X_3\} - \mathbb{E}\{X_1X_2\} - 0.25 \\
&= \frac{1}{3} + \mathbb{E}\{X_2^2\}(\mathbb{E}\{X_3^2\} + 0.25 - \mathbb{E}\{X_3\}) + 2(0.5)^3 - 0.25 - 0.25 = \frac{1}{9}
\end{aligned} \tag{10}$$

As can be seen the theoretical values show that the process is stationary as it doesn't vary with  $n$ . The standard deviation is similar to our plot which exhibits an output around 0.3.

Process 3: The expression is  $v[n] = (X - 0.5)m + a$ , where  $a = 0.5, m = 3$  and  $X \sim \mathcal{U}(0, 1)$

$$\mathbb{E}\{v[n]\} = m\mathbb{E}\{X\} - 0.5m + a = 0.5 \tag{11}$$

$$\begin{aligned}
Var\{v[n]\} &= \mathbb{E}\{v^2[n]\} - \mathbb{E}^2\{v[n]\} \\
&= \mathbb{E}\{m^2(X - 0.5)^2 + a^2 + 2am(X - 0.5)\} \\
&= m^2\mathbb{E}\{X^2 - X + 0.25\} + 2am\mathbb{E}\{X\} - am \\
&= 3^2\mathbb{E}\{X^2\} + (-3^2 + 3)\mathbb{E}\{X\} + (3^2(0.25) - 1.5) \\
&= 0.75
\end{aligned} \tag{12}$$

As can be seen again the theoretical values match the plots and show that the process is stationary.

### 1.3 Estimation of probability distributions

A stochastic process' characteristics can be fully described by its probability density function (*pdf*). This can be estimated from data, with the function *hist*. By plotting a normalized histogram, dividing bins values by total histogram area, a approximate representation of the *pdf* is obtained,

with an increase in accuracy as the number of samples increases. The approximation of the *pdf* for a  $X \sim \mathcal{N}(0, 1)$  Gaussian process is shown in Figure 9.

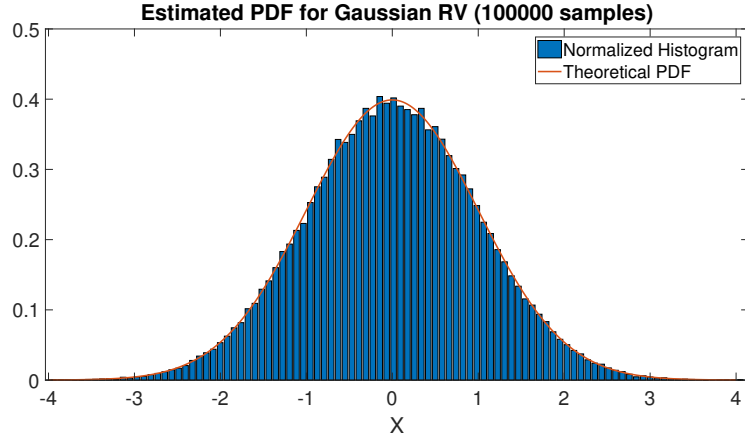


Figure 9: Gaussian PDF estimation using a normalized histogram

Figure 10 shows *pdf* estimation for the third process (stationary and ergodic), for different sample sizes ( $100, 1000, 10000$ ). As can be seen the *pdf* estimation improves as the number of samples increases, with the 10000 realization giving the best approximation to  $X \sim \mathcal{U}(0, 1)$ .

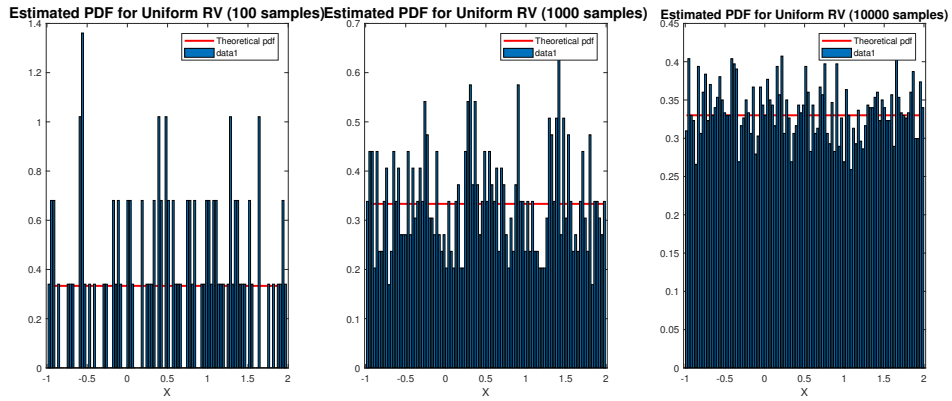


Figure 10: Normal distribution estimation with different sample values

For an non-stationary process, the *pdf* would change depending on the number of samples taken and hence would not be possible to estimate. A 1000-sample-long signal whose mean changes from 0 to 1 after 500 samples would be an example of this, as the *pdf* would be different for the 1<sup>st</sup> 500 samples than for the 2<sup>nd</sup> 500.



## 2 Linear Stochastic Modelling

Aims:

- To introduce practical estimators of the autocorrelation and cross-correlation functions.
- To give insight into linear stochastic modelling and its stability.
- To understand linear prediction models and optimal model order selection.
- To introduce the CRLB of an unbiased estimator and make judgement on proposed estimators.

### 2.1 ACF of uncorrelated and correlated sequences

The autocorrelation function (ACF) of a stochastic process  $X_n$  is determined by  $R_X(n, s) = \mathbb{E}\{X_n X_s\}$ . For real-world applications, however, only a finite number of samples is available and so the so-called unbiased estimate of the autocorrelation function is used:

$$\hat{R}_X(\tau) = \frac{1}{N - |\tau|} \sum_{n=0}^{N-|\tau|-1} x[n]x[n + \tau], \quad \tau = -N + 1, \dots, N - 1 \quad (13)$$

As knowledge of one samples doesn't give any information on the rest, for independent stochastic processes the autocorrelation has the form of a Dirac function.

Figures 11a and 11b show the autocorrelation function for a 1000-sample WGN signal for  $\tau \in [-999 : 999]$  and  $\tau < |50|$  respectively.

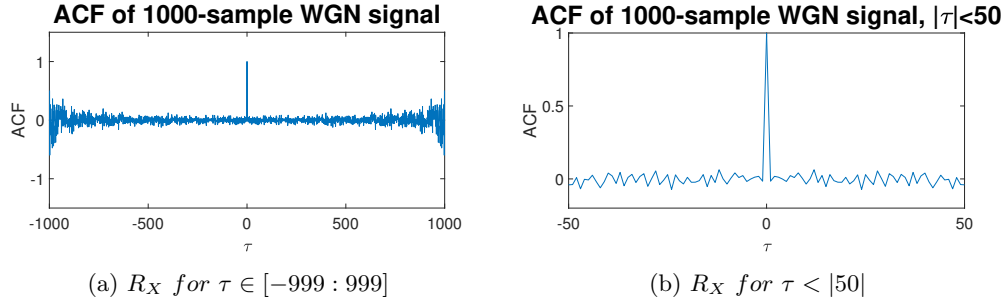


Figure 11: Autocorrelation function of 1000- sample WGN signal

As can be seen from the figures the autocorrelation estimation of a WGN signal matches closely the theoretical Dirac function. The result of substituting  $n$  for  $n = \hat{n} - \tau$ , yields the exact same result and hence varies not the autocorrelation equation resulting in a symmetric function, this is true for all real functions. From comparison between both plots it is clear  $\hat{R}_X$  increase with  $\tau$ , this is due to the number of samples being used to compute  $\hat{R}_X$  decreasing as  $\tau$  approaches  $\pm 1000$  and so becoming less reliable. The number of samples used in the calculation determines reliability and so any empirical bound would have to be done according to number of samples, a 90% reliable calculation would involve 90% of the samples of  $0.9 \times 1000 = 900$  samples hence giving  $N - |\tau| = 900$ ,  $\tau = 100$ .

Now we will filter the WGN signal with a moving average of 9 samples weighed equivalently, Figure 12 shows the autocorrelation function of the result.

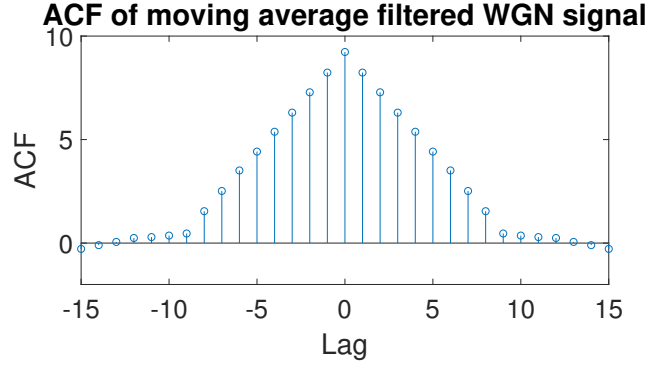


Figure 12: ACF of filtered 1000-sample WGN signal

With the filter being effectively a rectangular function, the autocorrelation function of the filtered signal is the equivalent of a convolution of two rectangular functions, that is a triangular function. This can also be considered as  $Y(f) = |H(f)|^2 X(f)$ , with  $H(f)$  being the filter transfer function (in this case a *sinc* function), resulting in  $Y(f)$  being a constant times  $\text{sinc}^2$ . Considering  $R_y = FT^{-1}\{Y(f)\}$  clearly yields the triangular function as before. If the order of the filter is increased it gradually accumulates more and more samples, eventually when the order reaches  $N-1$  it will be accumulating all the samples and hence by dividing over  $N$ , the sample mean can be obtained.

If  $X_n$  is an uncorrelated process its autocorrelation will be zero except for  $R_X(0) = 1$ . The ACF of  $Y_n$  denoted by  $R_Y$ , is thus given by

$$\begin{aligned} R_Y(\tau) &= R_X(\tau) * R_h(\tau) \\ &= \sum_{n=-\infty}^{\infty} R_X(n) R_h(\tau - n) = R_X(0) R_h(\tau) \\ &= R_h(\tau) \end{aligned} \quad (14)$$

Thus the result would be the autocorrelation of the filter response.

## 2.2 Cross-correlation function

The cross-correlation function (CCF) is defined as the expectation of a product of delayed samples of two different stochastic processes,  $R_{XY}(n, s) = \mathbb{E}\{X_n Y_s\}$ . For real world applications it can be used:

$$\hat{R}_{XY}(\tau) = \frac{1}{N - |\tau|} \sum_{n=0}^{N-|\tau|-1} x[n] y[n + \tau], \quad \tau = -N + 1, \dots, N - 1 \quad (15)$$

When estimating the CCF between the sequences  $\mathbf{x}$  and  $\mathbf{y}$  from before, Figure 13, it is clear  $\mathbf{x}$  is correlated to the last 8 samples of  $\mathbf{y}$ . This is expected due to the nature of the filter of order 9 applied. The approximations used when calculating the CCF result in the values outside the range being non-zero.

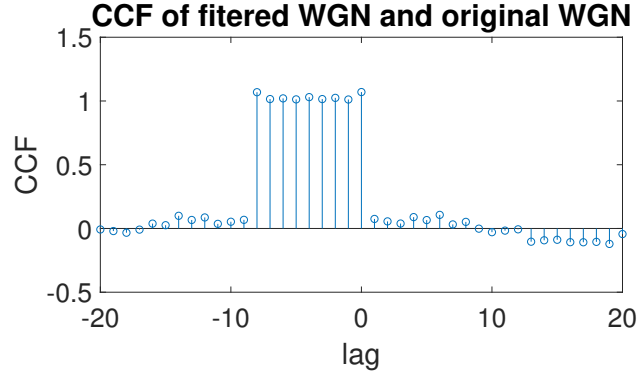


Figure 13: ACF of filtered 1000-sample WGN signal

The cross-correlation function between the input and the output of a filter is given by  $R_{XY}(\tau) = h(\tau) * R_X(\tau)$ . If  $X_t$  is an uncorrelated stochastic process the ACF would again be 0 except for  $R_X(0) = 1$ . With a similar calculation as to Equation 14 and hence  $R_{XY} = h(\tau)$ . The result of inputting an uncorrelated stochastic process to the filter can be clearly used to predict the model order as it will effectively be the impulse's response. Thus plotting the CCF will allow to identify the system and the order according too the relative magnitude of the lags.

### 2.3 Autoregressive modelling

An second order autoregressive model can be described as:

$$x[n] = a_1x[n-1] + a_2x[n-2] + w[n], \quad w[n] \sim \mathcal{N}(0, 1). \quad (16)$$

With values  $a_1 \in [-2.5, 2.5]$  and  $a_2 \in [-1.5, 1.5]$ , the convergence of the signal is tested. Figure 14 shows a plot of the signals which converge.

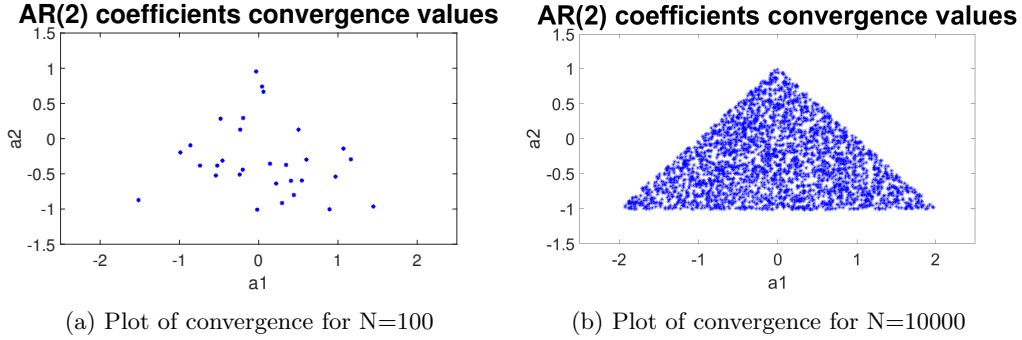


Figure 14

The autoregressive model is stable if the following conditions are met.

$$a_1 + a_2 < 1 \quad (17)$$

$$a_1 - a_2 < 1 \quad (18)$$

$$|a_2| < 1 \quad (19)$$

This 3 equations picture the triangle shown in Figure 14. This is obtained by performing the spectrum analysis of the process in the search of unstable roots. The z-transform of the autoregressive model is:

$$X(z) = \frac{W(z)}{1 - a_1z^{-1} - a_2z^{-2}} \quad (20)$$

The magnitude of the roots have to be less than one for the roots to be stable, that is lie inside the unit circle. To look for instability the denominator is set to zero such that  $1 - a_1 z^{-1} - a_2 z^{-2} = z^2 - a_1 z - a_2 = 0$ . Hence the roots can be expressed as:

$$|z| = \left| \frac{a_1 \pm \sqrt{a_1^2 + 4a_2}}{2} \right| \quad (21)$$

Setting the constraint of the magnitude being less than one:

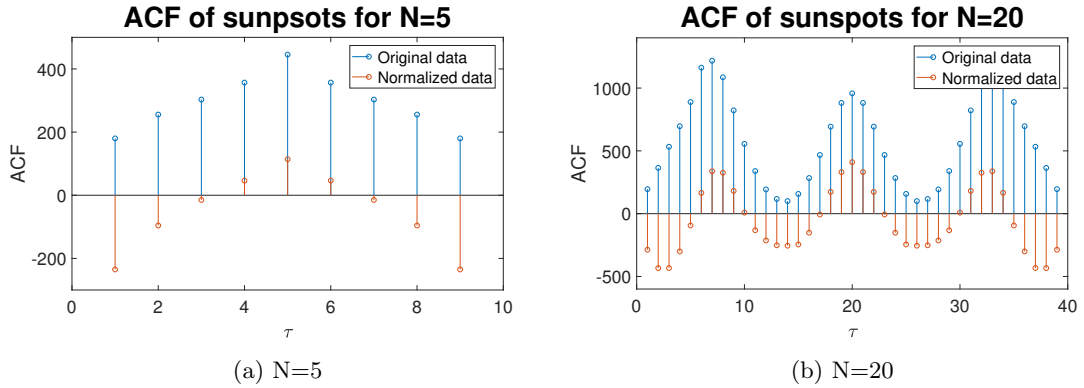
$$\begin{aligned} \frac{a_1 + \sqrt{a_1^2 + 4a_2}}{2} &< 1 \\ a_1^2 + 4a_2 &< (2 - a_1)^2 \\ a_1^2 + 4a_2 &< 4 - 4a_1 + a_1^2 \\ a_1 + a_2 &< 1 \end{aligned} \quad (22)$$

$$\begin{aligned} \frac{a_1 - \sqrt{a_1^2 + 4a_2}}{2} &< 1 \\ a_1^2 + 4a_2 &< (2 + a_1)^2 \\ a_1^2 + 4a_2 &< 4 + 4a_1 + a_1^2 \\ a_1 - a_2 &< 1 \end{aligned} \quad (23)$$

Finally, as  $|z_1| < 1$  and  $|z_2| < 1$ , we can further develop into  $|z_1||z_2| = |z_1 z_2| < 1$ , which as a result gives the 3<sup>rd</sup> condition:

$$\begin{aligned} \left| \left( \frac{a_1 + \sqrt{a_1^2 + 4a_2}}{2} \right) \left( \frac{a_1 - \sqrt{a_1^2 + 4a_2}}{2} \right) \right| &< 1 \\ \left| \frac{a_1^2}{4} - \frac{a_1^2 + 4a_2}{4} \right| &< 1 \\ |a_2| &< 1 \end{aligned} \quad (24)$$

The following figures show the plots for the unbiased ACF for sunspot data for both the empirical data and the zero-mean version of such. The plots are for  $N = 5, 20$  and  $250$ , and clearly the first plot gives little information as not enough samples are used to observe any trend, for  $N=20$  and  $250$  however, a sinusoidal-like trend is observed. As discussed before the ACF becomes less reliable as it approaches  $N$ , this is clearly seen in Figure 15 where the value of such increases considerably. By observing the zero-mean version of the data it can be seen how the ACF has a behavior hard to visualize in the biased ACF, due to the effect of the bias.



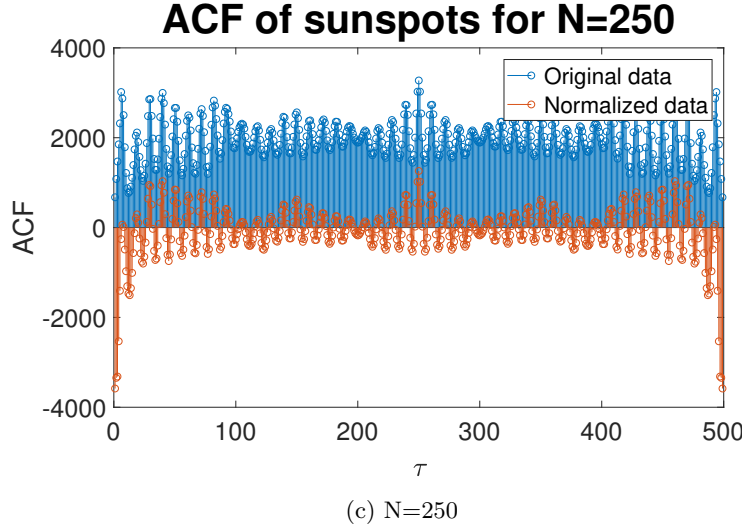


Figure 15: ACF of sunspot data

In order to do a model estimation of the process a Yule-Walker analysis is performed to obtain the Partial Autocorrelation Function (PACF), plotted in Figure 16. From this figure it can be seen how the PACF decreases considerably after order 2, with the standardized data giving a better result than the empirical data due to the removal of biases. This clearly suggests the process is AR(2). In theory PACF=0, for  $\tau > \text{model order}$ , however for real data PACF just decreases to small values, hence introducing the need for a threshold to decide the model order.

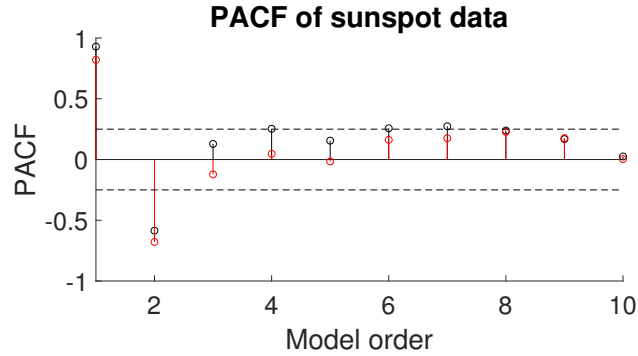


Figure 16: ACF of filtered 1000-sample WGN signal

The PACF gives a good estimation for the model order, however the threshold is chosen arbitrary and hence guarantees no real reliability. Instead the Minimum Description Length (MDL) and the Akaike Information Criterion (AIC) (and its corrected version AICc) can be used to determine the correct model order of the data.

$$MDL = \log(E_p) + \frac{p \log(N)}{N} \quad (25)$$

$$AIC = \log(E_p) + \frac{2p}{N} \quad (26)$$

$$AICc = AIC + \frac{2p(p+1)}{N-p-1} \quad (27)$$

As can be seen these functions combine the error achieved plus the complexity of the model, introducing a penalty proportional to order  $p$  which prevents over-fitting of model order. This way it is achieved a trade-off between minimizing error and model order to correctly estimate the process' order.

Figures 17a and 17b show the MDL, AIC and AICc for the sunspot data. The correct model order for the data will be the one that minimizes these functions, hence giving minimum error without over-fitting. From the plots it can be clearly seen how AR(2) minimizes these functions best and hence should be considered the model order. It is worth to note the differences between the functions, particularly AICc. The AICc has an additional penalty for order in a squared manner, this makes the plot very low order based, as high order models will have to have incredibly low error to compensate for the large penalty.

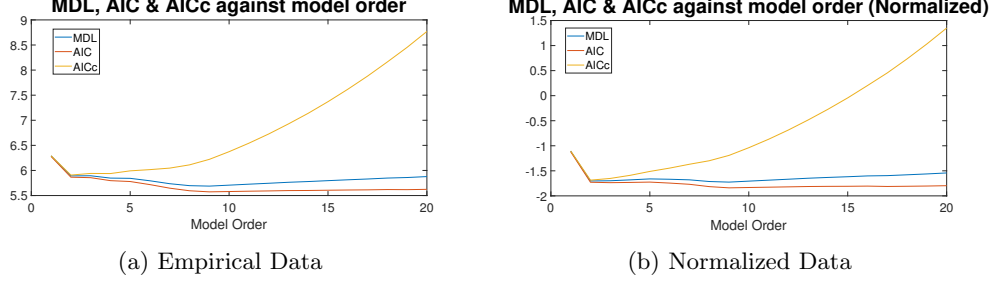


Figure 17: MDL, AIC and AICc for empirical and normalized sunspot data

Using the AR models AR(1), AR(2) and AR(10) calculated before we will now use the AR model to predict the sunspot  $m$  steps ahead, with  $m = 1, 2, 5, 10$ :

$$\hat{x}[n + m] = a_1x[n - 1] + \dots + a_px[n + p] \quad (28)$$

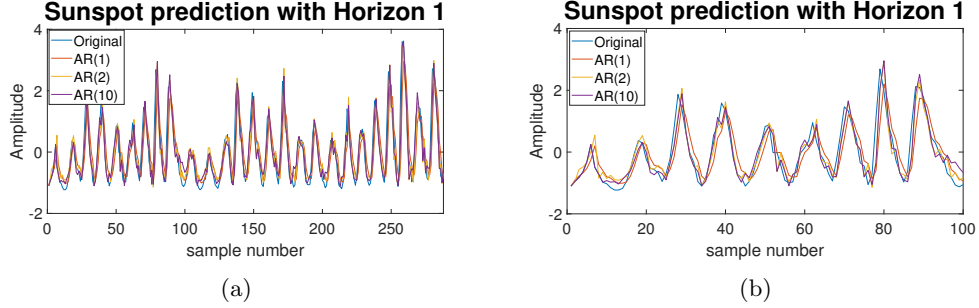


Figure 18: Sunspot signal prediction, Horizon 1

For  $m = 1$  it is clear that all models predict very well the signal. All models follow nicely the data and only the AR(1) which is just copying the previous sample would struggle if the signal was to change faster.

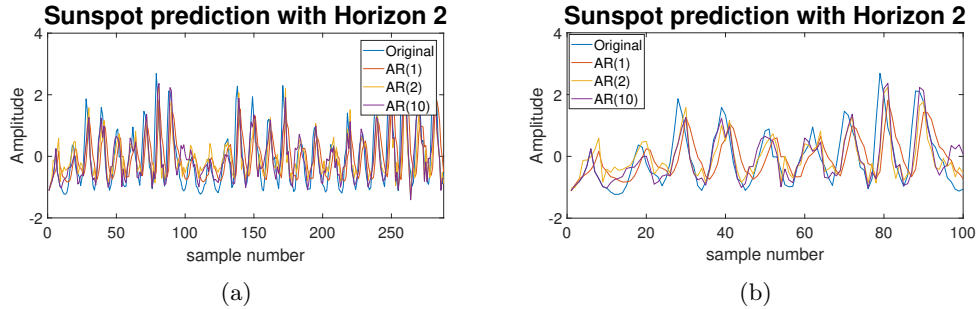


Figure 19: Sunspot signal prediction, Horizon 2

When  $m = 2$  the models start to diverge. Clearly AR(1) can no longer follow the signal nicely

and hence is inappropriate for this case. AR(2) and AR(10) still follow the data smoothly however the deviation from the data is more considerable than with  $m = 1$ .

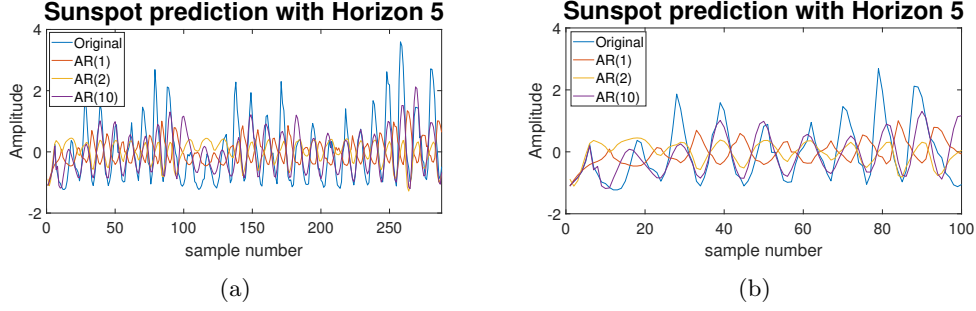


Figure 20: Sunspot signal prediction, Horizon 5

When  $m = 5$ , AR(1) and AR(2) no longer follow nicely the data. AR(2) somehow follows the data with a similar trend and simultaneous peaks, however the shape is completely different and is of no use for the prediction of the original data. On the other hand, AR(10) follows the data considerably well although with slight attenuation and shifting.

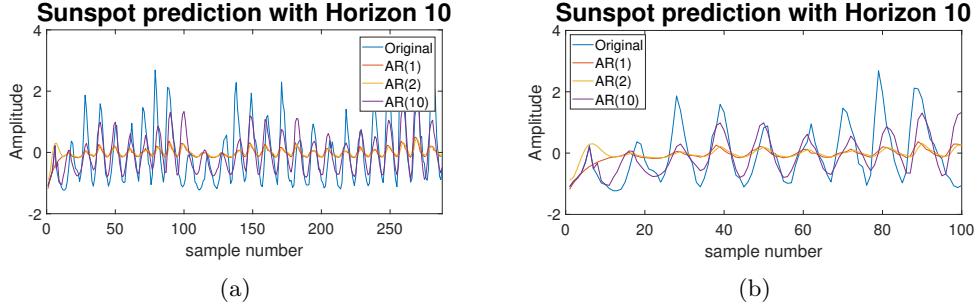


Figure 21: Sunspot signal prediction, Horizon 10

Finally for  $m = 10$  it is clear that AR(1) and AR(2) are of no use to predict the original data, giving negligible results in comparison with the original data. AR(10) follows the data to a certain extent however its deviation from the data is very remarkable than in other cases.

After the analysis of the different order models on predicting  $m$  samples ahead of the process, it is clear that low-order models only perform when subject to low values of  $m$  (ie, not very far in the future values), and that higher order models can only perform up to a certain point. It can then be hypothesized that in order to predict the  $x[n + m]$  sample, an AR( $m$ ) model is needed to obtain reasonably good predictions. Hence, even though higher order models introduce a cost in complexity, a clear evaluation of the prediction error (considering how high it is in cases such as  $m = 10$ ) should be done to properly evaluate the order of the model.

## 2.4 Cramer-Rao Lower Bound

The Cramer-Rao lower bound (CRLB) determines a lower bound on the variance of any unbiased estimator. The lower bound for an unbiased estimator is at least as high as the inverse of the Fisher information, if an estimator achieves this bound it is said to be efficient.

Figure 22 shows the PACF and MDL and AIC calculations for the NASDAQ closing prices, June 2003 to Feb 2007. The PACF clearly suggests an AR(1) model, being able to set a really small threshold at 0.1 and still determine it is AR(1). Additionally the MDL and AIC plots show an increasing behavior beyond modelorder=1, hence reinforcing the suggestion that the returns of the index can be modelled as an AR(1).

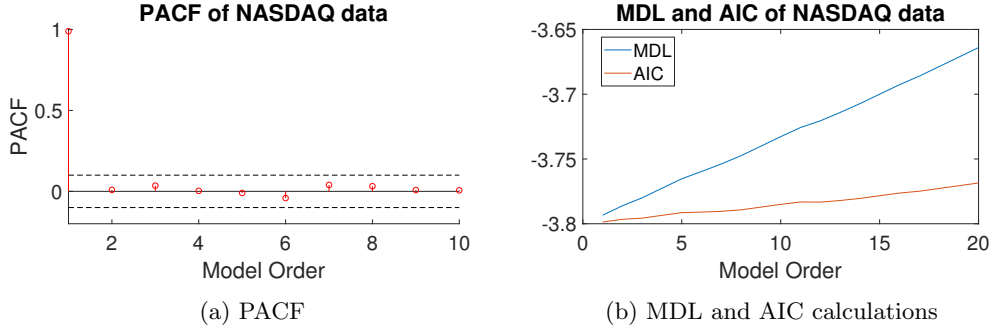


Figure 22: PACF and MDL and AIC calculations for NASDAQ closing prices

The derivation of the CRLB is hard to compute due to the complexity of inverting the covariance matrix of its estimators vector. We therefore use the standard log likelihood function of the power spectrum of an AR( $p$ ) model, Equation 29.

$$\ln[\hat{P}_X(f; \theta)] = \ln[\hat{\sigma}^2] - 2 \ln \left[ 1 - \sum_{m=1}^p \hat{a}_m \exp^{-j2\pi f m} \right] \quad (29)$$

Where  $m=1$  in this case and the Fisher information is obtained by Equation 30.

$$[I(\theta)]_{ij} = \frac{N}{2} \int_{-\frac{1}{2}}^{\frac{1}{2}} \frac{\delta \ln[\hat{P}_X(f; \theta)]}{\delta \theta_i} \frac{\delta \ln[\hat{P}_X(f; \theta)]}{\delta \theta_j} df \quad (30)$$

Assuming  $p = 1$  such that  $\theta = [a_1, \sigma^2]$ ,  $[I(\theta)]_{22}$  is computed where  $\frac{\delta \ln[\hat{P}_X(f; \theta)]}{\delta \sigma^2} = \frac{1}{\sigma^2}$ .

$$\begin{aligned} [I(\theta)]_{22} &= \frac{N}{2} \int_{-\frac{1}{2}}^{\frac{1}{2}} \frac{1}{\sigma^2} \frac{1}{\sigma^2} df \\ &= \frac{N}{2\sigma^4} \end{aligned} \quad (31)$$

$$I(\theta) = \begin{bmatrix} \frac{Nr_{xx}(0)}{\sigma^2} & 0 \\ 0 & \frac{N}{2\sigma^4} \end{bmatrix} \quad (32)$$

Hence  $I(\theta)$  can be obtained, Equation 32, given  $I(\theta)_{11} = \frac{Nr_{xx}(0)}{\sigma^2}$  and  $I(\theta)_{12} = I(\theta)_{21} = 0$ .

Given  $\text{var}(\hat{\theta}_i) \geq [I^{-1}(\theta)]_{ii}$ , the variance for both estimator can be obtained as  $\text{var}(\hat{\sigma}^2) \geq \frac{1}{N/2\sigma^4} = \frac{2\sigma^4}{N}$  and similarly  $\text{var}(\hat{a}_1) \geq \frac{\sigma^2}{Nr_{xx}(0)}$ :

$$\begin{aligned} r_{xx}(0) &= \frac{1}{N} \sum_{n=0}^{N-1} x[n]x[n] = \frac{1}{N} \sum_{n=0}^{N-1} (a_1 x[n-1] + w[n])^2 \quad w[n] \sim \mathcal{N}(0, \sigma^2) \\ &= \frac{1}{N} \sum_{n=0}^{N-1} a_1^2 x^2[n-1] + w^2[n] \quad \text{as } \sum_{n=0}^{N-1} w[n] = 0 \quad \text{where } \frac{1}{N} \sum_{n=0}^{N-1} w^2[n] = \sigma^2 \\ &= \sigma^2(1 + a_1^2 + a_1^4 + a_1^6 + \dots) + \frac{a_1^{2N}}{N} \sum_{n=0}^{N-1} x^2[n-N] \\ &= \frac{\sigma^2}{1 - a_1^2} \quad \text{as } a_1 < 1 \end{aligned} \quad (33)$$

For Equation 33 the final result is dependent on the constraint  $a_1 < 1$  as  $N \rightarrow \infty$ , which is obtained by using the equations in section 2.3 and setting  $a_2 = 0$ , from where the geometric approximation can be used to obtain the solution,  $\frac{a(1-r^n)}{1-r}$ , where  $a = 1$ ,  $r = a_1^2$ , hence  $r^n = 0$  for  $n \rightarrow \infty$ . Substitute this result into the equation for  $\text{var}(\hat{a}_1)$  to get the expected  $\text{var}(\hat{a}_1) \geq \frac{1-a_1^2}{N}$ .



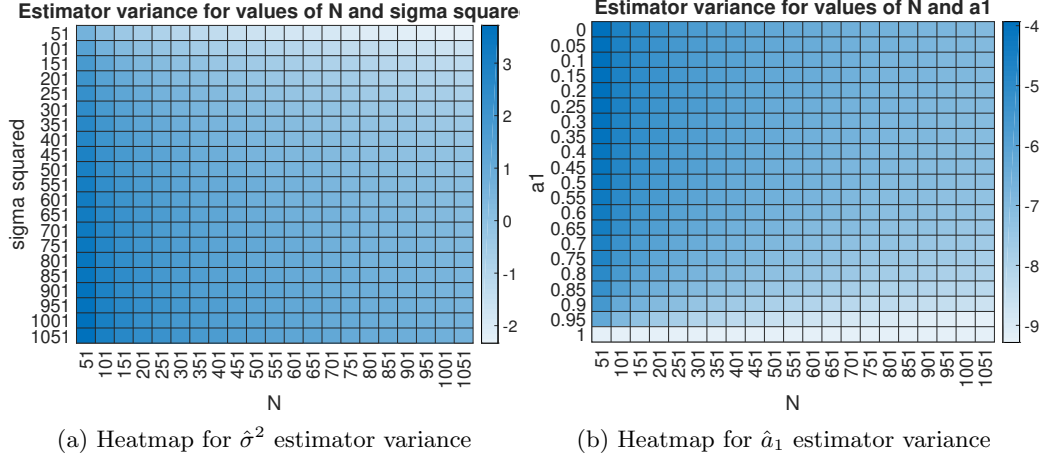


Figure 23: Heatmaps for  $\sigma^2$  and  $a_1$

Figure 23 shows the heatmaps for the variance of  $\hat{\sigma}^2$  and  $\hat{a}_1$ , the results were compressed with the log function to give a more progressive gradient (hence the -inf values for  $a_1 = 1$ ). Additionally  $N$  was shifted to 51 : 50 : 1051, in order to properly see the gradient, as  $N = 1$  restricted the plot too much. As expected bigger values of  $N$  decrease the variance, for  $\hat{\sigma}^2$  a lower actual value gives a lower variance on the estimate, whilst for  $a_1$  the closer the actual value is to 1 the lower the estimate variance.

From the yule-walker equations, the approximate value of  $a_1$  for the NASDAQ financial data is 0.9887, from the calculation it is clear further order models just provide negligible values of  $a_p$  for  $p \geq 2$ . As  $\hat{a}_1$  approaches 1 the variance of the estimator tends to be greater 0, which provides no information on the estimator. This is due to the nature of the spectrum derived upon the position of the pole. As  $a_1 \rightarrow 1$  the system becomes unstable and hence the estimator can diverge to anything, thus having no information on the estimator performance as it becomes unsuitable.

From the CRLB it can be shown:

$$\text{var}(\hat{P}_X(f; \theta)) \geq \frac{\delta \hat{P}_X(f; \theta)^T}{\delta \theta} I^{-1}(\theta) \frac{\delta \hat{P}_X(f; \theta)}{\delta \theta} \quad (34)$$

where  $\frac{\delta \hat{P}_X(f; \theta)}{\delta \theta} = \left[ \frac{\delta \hat{P}_X(f; \theta)}{\delta \theta_1}, \frac{\delta \hat{P}_X(f; \theta)}{\delta \theta_2} \right]^T$ , where  $\theta_1 = a_1$  and  $\theta_2 = \sigma^2$  and the bound is to be computed in terms of  $A(f) = 1 - a_1 e^{-j2\pi f}$ .

$$\begin{aligned} \frac{\delta \hat{P}_X(f; \theta)}{a_1} &= \frac{\sigma^2(A(f)e^{j2\pi f} + A^*(f)e^{-j2\pi f})}{|A(f)|^4} \\ &= \frac{2\sigma^2 \Re\{A(f)e^{j2\pi f}\}}{|A(f)|^4} \end{aligned} \quad (35)$$

$$\frac{\delta \hat{P}_X(f; \theta)}{\sigma^2} = \frac{1}{|A(f)|^2} \quad (36)$$

$$I(\theta)^{-1} = \begin{bmatrix} \frac{1-a_1^2}{N} & 0 \\ 0 & \frac{2\sigma^4}{N} \end{bmatrix} \quad (37)$$

$$\begin{aligned} \text{var}(\hat{P}_X(f; \theta)) &\geq \left[ \frac{\sigma^2(A(f)e^{j2\pi f} + A^*(f)e^{-j2\pi f})}{|A(f)|^4} \right] \begin{bmatrix} \frac{1-a_1^2}{N} & 0 \\ 0 & \frac{2\sigma^4}{N} \end{bmatrix} \left[ \frac{\sigma^2(A(f)e^{j2\pi f} + A^*(f)e^{-j2\pi f})}{|A(f)|^4} \right] \left[ \frac{1}{|A(f)|^2} \right] \\ &\geq \frac{1-a_1^2}{N} \left( \frac{\sigma^2(A(f)e^{j2\pi f} + A^*(f)e^{-j2\pi f})}{|A(f)|^4} \right)^2 + \frac{2\sigma^4}{N} \left( \frac{1}{|A(f)|^2} \right)^2 \\ &\geq \frac{2\sigma^4}{N|A(f)|^4} \left( \frac{(1-a_1^2) \Re\{A(f)e^{j2\pi f}\}^2}{|A(f)|^4} + 1 \right) \end{aligned} \quad (38)$$

## 2.5 Real world signals: ECG from iAmp experiment

The RRI signal can be used to obtain the heart rate by the relationship,  $h[n] = \frac{60}{rr[n]}$ . Using the pdf estimate with the function used in Part 1, we can see how the data is distributed.

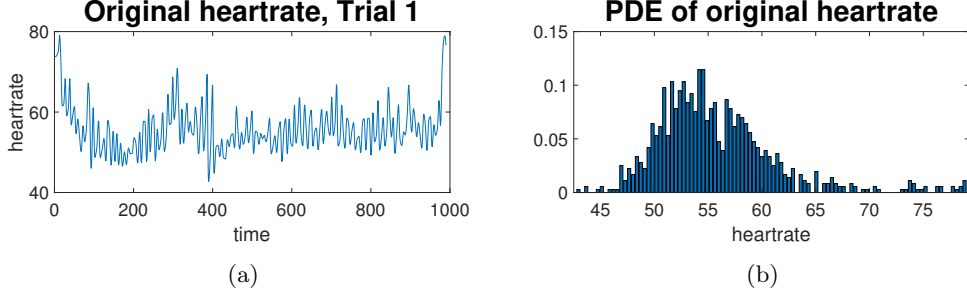


Figure 24: Heart rate and estimated pdf of original heart rate

Additionally, a smoother estimate can be obtained by averaging every 10 samples, Equation 39. Setting  $\alpha = 1$  and 0.6, the effect of alpha can be seen on the estimated pdf, assuming all signals are stationary.

$$\hat{h}[n] = \frac{1}{10} \sum_{i=10n-9}^{10n} \alpha h[i] \quad (39)$$

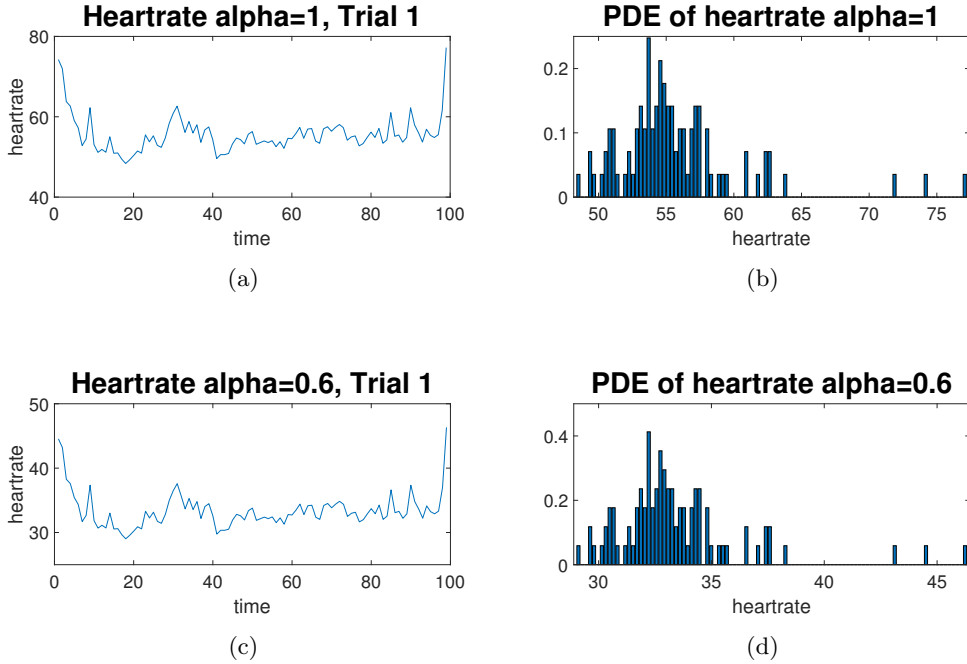


Figure 25: Heart rate and estimated pdf of heart rate for  $\alpha = 1$  and 0.6

The averaged estimated pdfs clearly show less variance with the result being much more reliable about one point. The variance is affected as  $Var(\hat{h}[n]) = Var(\frac{1}{10} \sum_{i=10n-9}^{10n} h[i]) = \frac{1}{10} Var(h[n])$ . Hence the variance of the averaged signal is reduced by a factor of 10 and the estimated pdf is more compressed, as can be seen comparing Figures 24b and 25b.

Furthermore the effect of  $\alpha$  on the estimated pdf is clear between Figures 25b and 25d. The value of  $\alpha$  shifts the PDE, effectively introducing a bias to the estimated pdf. Additionally, because

$\alpha \leq 1$ , the variance will be reduced further as  $Var(\alpha \hat{h}[n]) = \alpha^2 Var(\hat{h}[n])$ , giving an attenuation of 0.36 for  $\alpha = 0.6$ .

The conclusion is that the effect of *alpha* and averaging the signal results in a reduction in variance, and an introduction of bias assuming  $\alpha \neq 1$ .

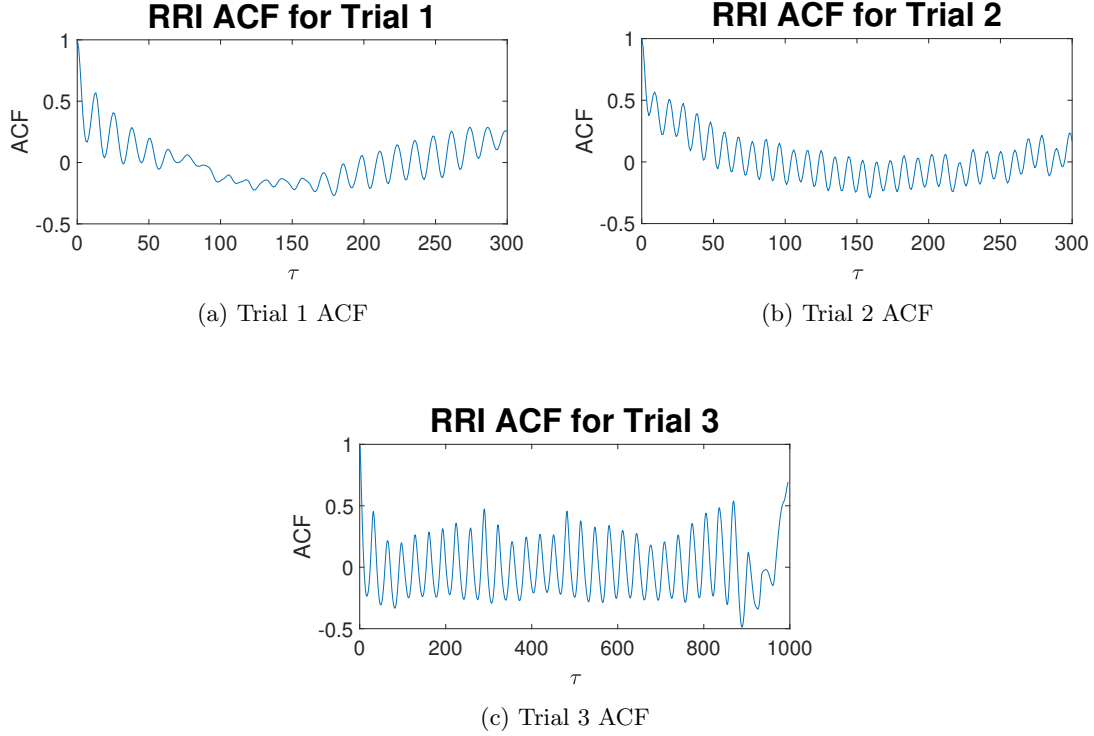


Figure 26: Autocorrelation of Trials 1-3

Figure 26 shows the unbiased autocorrelation function estimation for Trials 1-3. From the plots it can be deduced the processes are not correlated by their shifted version as the ACF decays from  $R_{xx}(0)$ , however the ACF doesn't settle to zero like in a MA process and thus it is modeled as an AR process.

Finally the methods in section 2.3 will be used to predict the AR model order of the different trials. The PACF will be done and the behavior under the MDL, AIC and AICc measures evaluated.

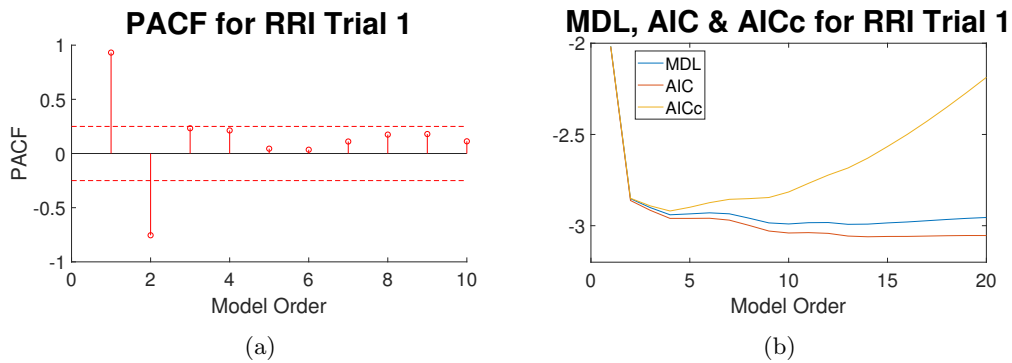


Figure 27: PACF and complexity measures for Trial 1

Figure 27 shows the PACF and complexity measures for Trial 1. Setting the arbitrary threshold at 0.25, the PACF suggests an AR(2) model, however the MDL and AIC predictions suggest an AR(4) model might also fit the data. A trade-off between the arbitrary threshold of the PACF

and the relative difference between AR(2) and AR(4) in the MDL and AIC needs to be evaluated as AR(3) gives no significant trade-off and just behaves like AR(4). It is clear that the AR(2) is a more significant representation which would also prevent over-fitting and hence should be selected. As a result it is suggested Trial 1 is an AR(2) model.

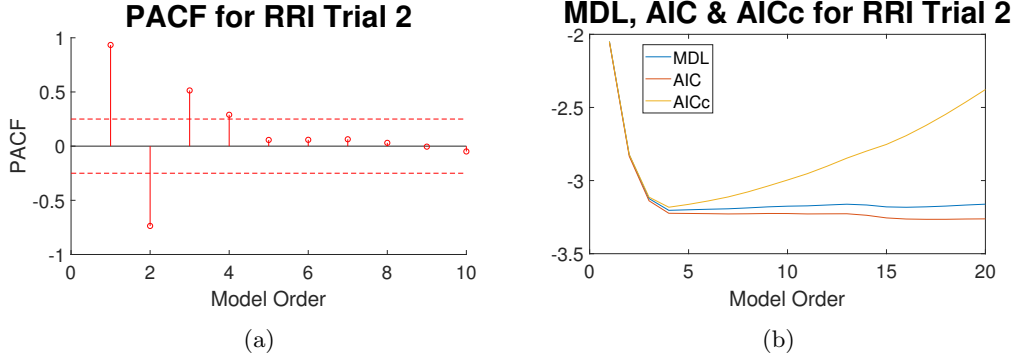


Figure 28: PACF and complexity measures for Trial 2

Unlike Trial 1, Trial 2 shows a more clear suggestion towards an AR(4) model. A threshold of 0.25 gives an AR(4) suggestion clearly and MDL and AIC give minimum error at AR(4). Even though the MDL and AIC error seems to settle past AR(4) at minimum values, the PACF helps us decide as values greater than 4 decay greatly to negligible values.

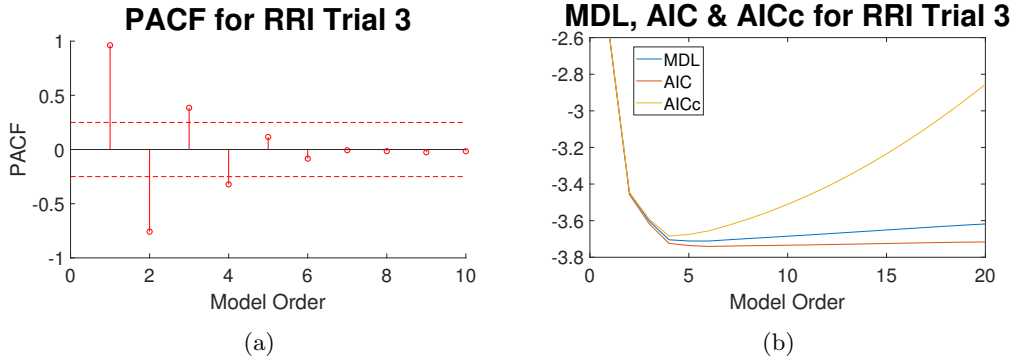


Figure 29: PACF and complexity measures for Trial 3

Trial 3, Figure 29, has a similar behavior to Trial 2, giving a clear suggestion towards AR(4) using the MDL and AIC errors backed by the PACF with an arbitrary threshold of 0.25.

### 3 Spectral Estimation Modelling

Aims:

- To introduce the Power Spectral Density (PSD) of a random signal.
- To study classical methods for PSD estimation.
- To investigate model-based methods for PSD estimation.
- To understand ties between spectrum estimation and MLE and LSE methods.
- To gain experience with practical applications of spectrum estimation.

The estimated PSD of noise is done on N-samples realizations of WGN (N=128, 256, 512), Figure 30 shows the different results:

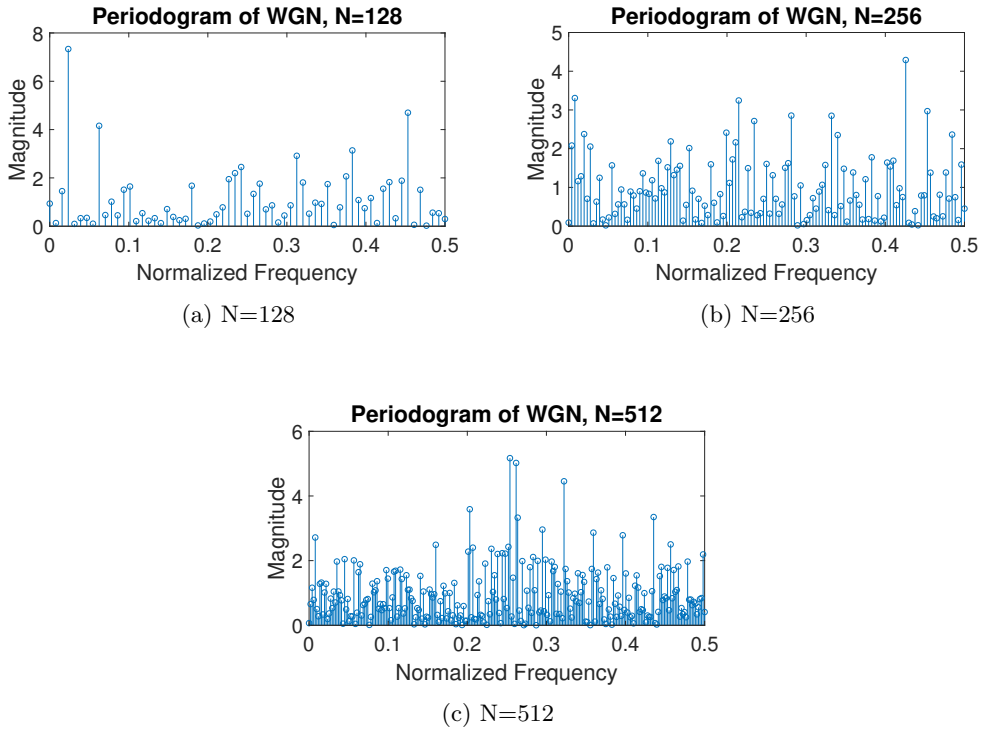


Figure 30: PSD estimation for WGN

The formula used for PSD estimation is the so called periodogram defined by:

$$\hat{P}_X(f) = \frac{1}{N} \left| \sum_{n=0}^N x[n] e^{-j2\pi f \frac{n}{N}} \right|^2 \quad (40)$$

This method clearly does not estimate accordingly the PSD of WGN which should be a constant value. This is because the periodogram does not converge to the PSD with  $N \rightarrow \infty$  as shown in the plots, Figure 30c is not objectively better than Figure 30a. The periodogram definition can be written as the Fourier Transform of the Autocorrelation function, hence the estimator used will be unbiased as  $\mathbb{E}\{\hat{P}_X\} = P_X$  as  $N$  increases. However,  $\text{Var}\{\hat{P}_X\}$  does not reduce to 0 as  $N$  increases and thus the estimate is not appropriate as it has a great variance even for large  $N$ , hence giving an inconsistent estimate.

### 3.1 Averaged Periodogram Estimates

To smooth out the PSD estimate a zero-phase FIR filter is used:

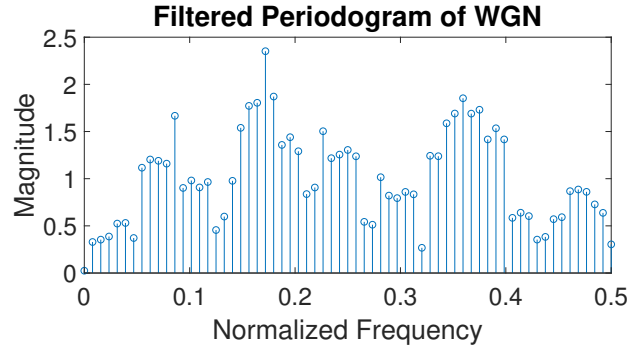


Figure 31: Periodogram of filtered WGN

Figure 31 shows how the filtering clearly smoothens the result by reducing variance about the mean. By filtering the signal the estimate improves as  $Var\{\frac{1}{5} \sum_{n=1}^5 \hat{P}_X[n-i]\} = \frac{1}{5} Var\{\hat{P}_X[n]\}$ , thus reducing the variance by 5.

By dividing a 1024-sample realization into 8 128-sample realization we can get 8 different periodograms of the same sample of noise. These turn out to have no clear trend with clear differences in between them. The large variance and small sample realization make the result very different to the ideal flat result.

By calculating an average of these 8 results an average result can be obtained.

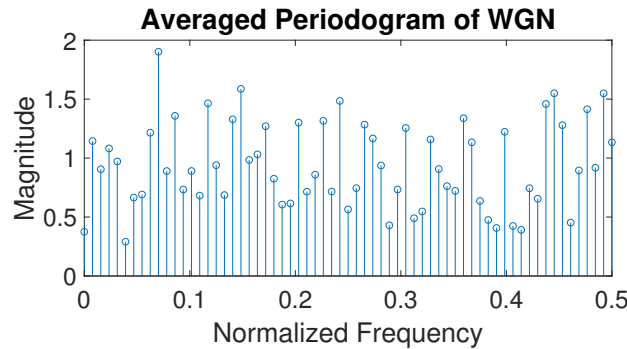


Figure 32: Average Periodogram of filtered WGN

The averaged version of the signal clearly gives a more appropriate estimate for the PSD of noise. The averaging reduces the variance by  $\frac{1}{M}$ , where M is the number of segments. This segmentation however reduces frequency resolution as the number of samples is effectively reduced from  $N$  to  $\frac{N}{M}$ , thus a decrease in resolution is obtained, the process of averaging the different segments is equivalent to the Section 1.2 performing ensemble averages to signals.

### 3.2 Spectrum of Autoregressive Processes

Figure 33 shows the original and the filtered signals for a WGN signal, in the time domain. Clearly the filtered signal varies more and has higher amplitude, this is because the high-pass filter will amplify the high frequency components while attenuating the low frequency ones, and as a result the signal will vary more aggressively.

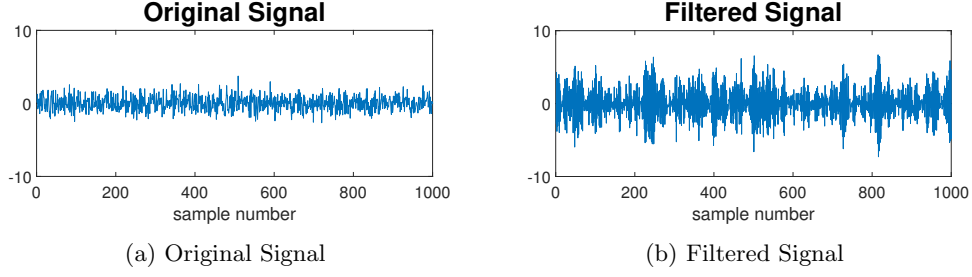


Figure 33

By filtering noise under certain criteria an AutoRegressive Model (AR(1)) is obtained. From the theory of AR models and the selected filtering parameters the PSD of the obtained model is:

$$P_Y(f) = \frac{1}{|1 + 0.9e^{-j2\pi f}|^2} \quad (41)$$

Figure 34 shows the spectrum of the signal which clearly shows a high-pass behavior. The theoretical response however show a much more stable response than the periodogram of the signal which oscillates considerably more.

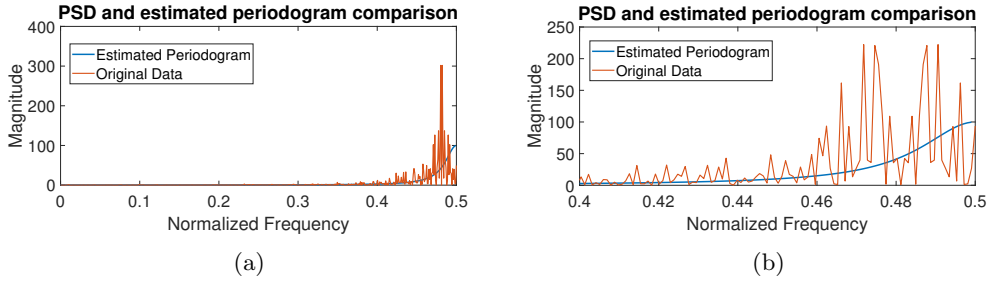


Figure 34: Periodogram for AutoRegressive AR(1) process

The estimated PSD doesn't follow the theoretical response, the variance of the periodogram is not zero and hence oscillates around the real value. Furthermore, because we are using a finite number of samples for the WGN realization we are effectively windowing an infinite-sample realization of WGN, or the equivalent of convolving with a sinc function. As the PSD is small for low frequencies the impact is negligible, until frequencies are considerably high when the convolution with the sinc function results in such significant oscillations.

By assuming  $\mathbf{y}$  is generated by an AR(1) model its PSD estimation is reduced to the estimation of the parameters  $\hat{a}_1$  and  $\hat{\sigma}_X^2$ , and its use in the equation:

$$\hat{P}_y(f) = \frac{\hat{\sigma}_X^2}{|1 + \hat{a}_1 e^{-j2\pi f}|^2} \quad (42)$$

These parameters can be in turn estimated from the autocorrelation of  $\mathbf{y}$ ,  $\hat{R}_Y$ .

$$\hat{a}_1 = -\frac{\hat{R}_Y(1)}{\hat{R}_Y(0)} \quad (43)$$

$$\hat{\sigma}_X^2 = \hat{R}_Y(0) + \hat{a}_1 \hat{R}_Y(1) \quad (44)$$

This gives  $\hat{a}_1 = 0.9039$  and  $\hat{\sigma}_X^2 = 1.0791$ , from which an estimate of the PSD can be obtained, see Figure 35.

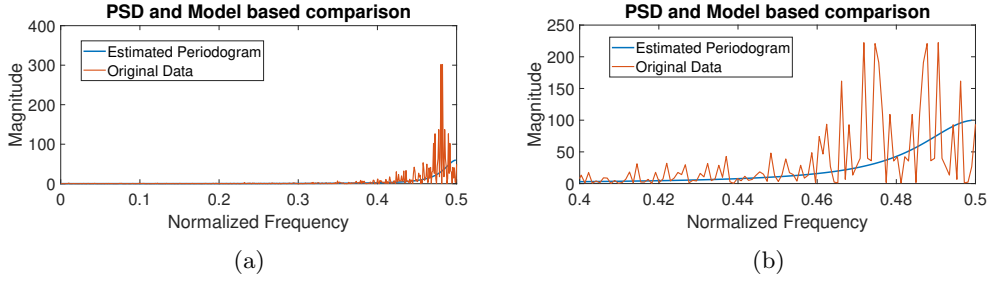


Figure 35: Model based PSD estimate and periodogram comparison

The same analysis and comparison can be done for sunspot data comparing the model based estimated PSD and periodogram result. In previous analysis it was discussed how this data corresponds to an AR(2) process.

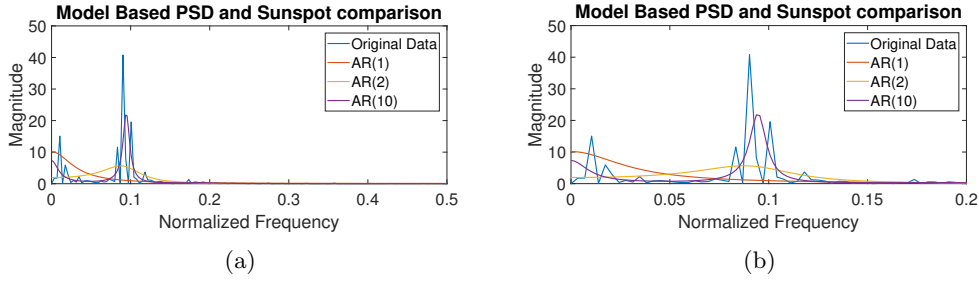


Figure 36: Model based PSD estimate for sunspot data vs periodogram

It is clear from Figure 36 that the AR(1) is unsuitable for the PSD estimation as the estimation doesn't manage to follow the periodogram and hence is not a good PSD estimate. Comparing AR(2) and AR(10), AR(2) provides a smoother response whilst AR(10) provides a more complex response. However, when comparing with the periodogram AR(10) performs best in simulating the periodogram, thus suggesting AR(10) is a better estimate for the PSD. Despite this, it is worth to notice that the periodogram is not a fully reliable estimate of the PSD as discussed above. As a result even though AR(10) performs better than AR(2) in following the periodogram's behavior, the statement that AR(10) is a better estimate of the PSD is unreliable and should only be considered as a better estimate of the periodogram, which in turn is an estimate of the PSD.

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

For an AR(p) process the LS cost function of finding the unknown AR coefficients is given by:

$$J = \sum_{k=1}^N \left[ \hat{r}_{xx}[k] - \sum_{i=1}^p a_i \hat{r}_{xx}[k-i] \right]^2 \quad \text{for } M \geq p \quad (45)$$

This cost function can be expressed in matrix form in the form  $(\mathbf{x} - \mathbf{H}\mathbf{a})^T(\mathbf{x} - \mathbf{H}\mathbf{a})$  by mapping Equation 45 into the geometric form such that  $(\mathbf{x} - \mathbf{H}\mathbf{a})$  equals:

$$\begin{bmatrix} \hat{r}_{xx}[1] \\ \hat{r}_{xx}[2] \\ \vdots \\ \hat{r}_{xx}[N] \end{bmatrix} - \begin{bmatrix} \hat{r}_{xx}[0] & \hat{r}_{xx}[-1] & \cdots & \hat{r}_{xx}[1-p] \\ \hat{r}_{xx}[1] & \hat{r}_{xx}[0] & \cdots & \hat{r}_{xx}[2-p] \\ \vdots & \vdots & \ddots & \vdots \\ \hat{r}_{xx}[N-1] & \hat{r}_{xx}[N-2] & \cdots & \hat{r}_{xx}[N-p] \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} \quad (46)$$

From Equation 46 it can be seen how  $\mathbf{H}$  is the matrix corresponding to the equation  $\mathbf{H} = \sum_{k=1}^M \sum_{i=1}^p \hat{r}_{xx}[k-i]$ , whilst the signal model  $\mathbf{s} = [\mathbf{H}\mathbf{a}]$  which equals  $\mathbf{H} = \sum_{k=1}^M \sum_{i=1}^p a_i \hat{r}_{xx}[k-i]$ . The Least Squares estimates for the unknown coefficients  $\mathbf{a}$  is done by differentiating the cost function and setting it to zero to get the minimum, giving  $\hat{\mathbf{a}} = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{x}$ .



The observation matrix  $\mathbf{H}$  is computed using  $\hat{\mathbf{r}}_{xx}$ , Equation 47, from this equation it is clear it contains  $\epsilon[k]$  accounting for the ACF estimate errors due to the noise,  $w[n] \sim \mathcal{N}(0, 1)$ , of the AR process. Hence as this noise is stochastic, the observation matrix  $\mathbf{H}$  which is eventually derived from this noise is also stochastic.

$$\hat{r}_{xx}[k] = \sum_{i=1}^p a_i \hat{r}_{xx}[k-i] + \epsilon[k] \quad \text{for } i \geq 1 \quad (47)$$

In order to obtain the AR( $p$ ) coefficients for the sunspot time series the LSE approach is taken for  $p = 1, \dots, 10$ . The results are given in Figure 37a where it can be seen how the optimum value of  $p$  is 3, which minimizes the approximation error without having too much model complexity. Figure 37b shows the estimation of the sunspots with the AR(3) coefficients showing how it clearly follows the sunspot data very closely.

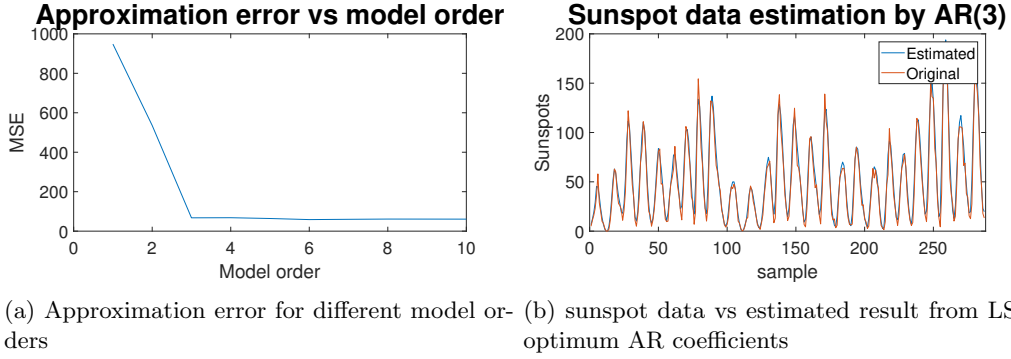


Figure 37: LSE approach for different model orders and data estimation

Figure 37 suggests an AR(3) model which differs from the AR(2) from section 2.3, however section 2.3 suggests the model to be modeled as order 2 considering the trade-off of complexity versus error is not beneficial. The LSE approach here gives a bigger importance to error and less to model complexity, thus pushing the estimation to higher model orders.

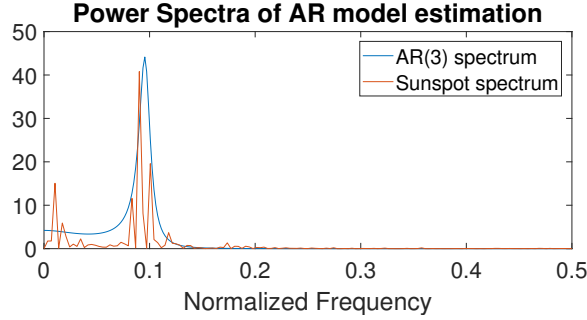


Figure 38: AR(3) spectrum vs sunspot spectrum

Figure 38 shows the comparison between the spectrum of the sunspot data and that of the AR(3) model which was found to be the best estimate. The AR(3) clearly estimates the sunspot nicely with both plots having a resemblance in shape.

Figure 39 shows the plot of approximation error vs number of data points for the optimum AR model with  $N$  varying between 10 and 250. It is clear the AR model estimates best the sunspot data for lower values of  $N$ . This suggests that the model is not suitable for this estimation, the sudden increase in error around  $N = 70$  can be explained by a change in model parameters for which the set AR coefficients would no longer be properly estimating the data. The data is assumed to be stationary, as the coefficients don't change through time, however the sunspot model may

in reality be non-stationary, hence making approximation error to increase the moment the model parameters change. This suggests  $N$  to be a lower value in the region 20 to 70.

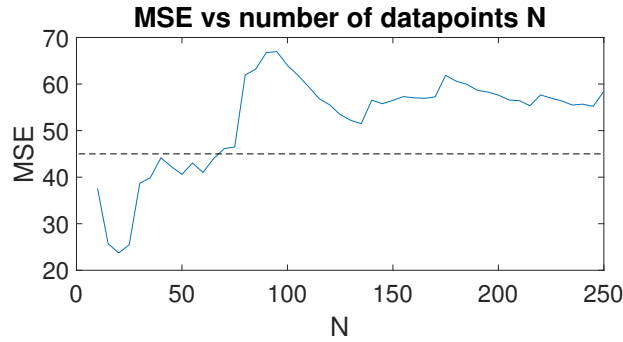


Figure 39: Approximation error vs number of data points

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

Figure 40 shows the signal  $y$  when numbers 0 and 2 are pressed.

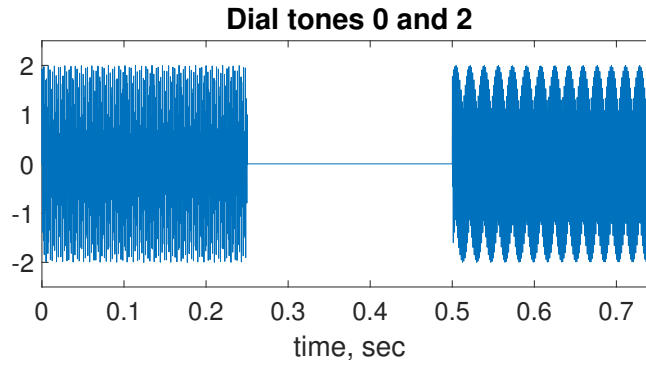


Figure 40: Pressing of dials 0 and 2

It is clear the difference between the two dials with the idle time delimiting them. The sampling frequency of  $32768Hz$  allows frequencies to avoid aliasing up to  $16384Hz$  which provides an appropriate coverage of any frequencies used (max frequency of dials =  $1477Hz$ ). In addition this results in each dial pressing accounting for  $0.25 * 32768 = 8192$  samples, which as a power of 2 is ideal for FFT operation as no zero-extension is needed.

Figure 41 shows the spectrogram of the number 020XXXXXXX, along with the FFT of the dials 0 and 2. From the spectrogram it is clear that the different dials can be distinguished by determining which frequencies are being detected and then comparing with the original table. Similarly, the FFT shows clear peaks around the corresponding frequencies of the dial. Clearly both in the spectrogram and in the FFT plot the frequencies are not perfectly pointed, with the power gradually decreasing away from the selected frequency, and the FFT not showing  $\delta$  functions but peaks. This effect is due to the use of the Hamming window in the spectrogram function which results in frequencies around the result obtaining some power in the analysis.

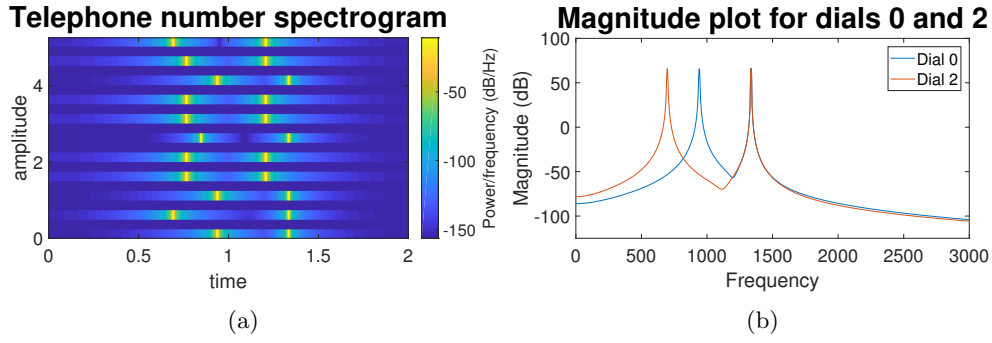


Figure 41: Spectrogram and FFT plot of dial number

From the figures it is clear how the dial number can be estimated with both plots. The process would involve setting a threshold in power in the spectrogram for a brief period of time, and if the set frequency is above that threshold a simple look-back to the original frequencies table would identify the dial. For the FFT plot a similar analysis would be done with peak detection of frequencies.

In real world situations, the signal is corrupted by noise. This makes signal identification harder due to the corruption introduced by noise, the higher the noise power the more corruption and the harder the identification. Figure 42 shows the noisy versions of Figure 40 for  $\sigma = 2, 5, 15$ , thus clearly showing the effect of noise on the signal.

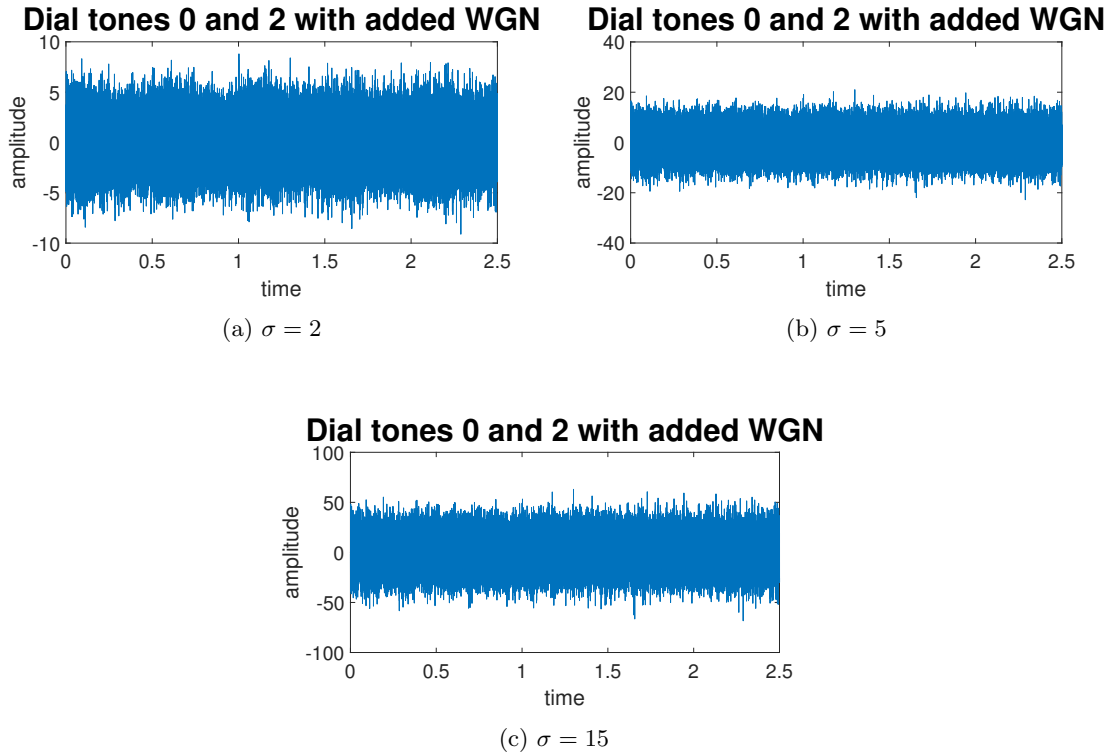


Figure 42: Signal corrupted with different noise powers

For  $\sigma = 2$  the noise power is considerable and hence the signal is corrupted and harder to identify. Dialing and idle periods are unclear and a real effort is needed to distinguish between dial and idle periods, the dialing signal is corrupted enough so that its identification is impossible, at least visually. From the plot of the spectrogram it is clearly possible to identify the different frequencies of the corresponding dial. Noise has clearly affected the result but the frequencies are still identifiable according to its  $\frac{\text{power}}{\text{frequency}}$ .

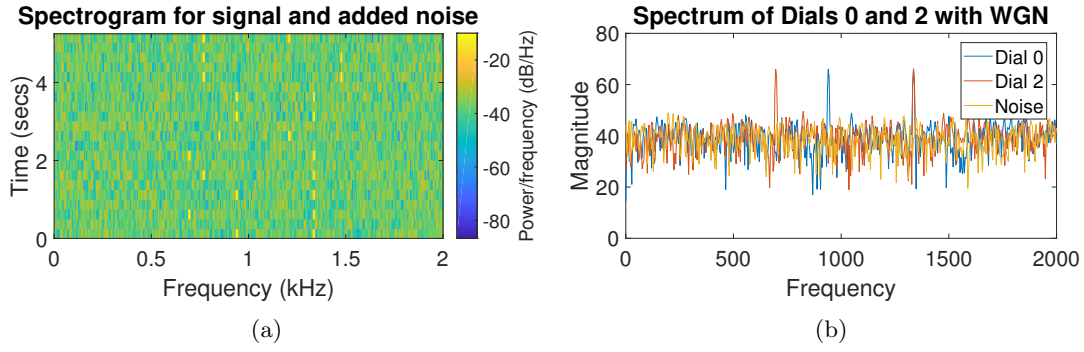


Figure 43: Spectrogram and FFT plot of dial number with  $\sigma = 2$

As  $\sigma$  is increased to 5, the signal is further corrupted and harder to identify. The spectrogram is still able to identify the different frequency however the threshold to be considered should be higher as the relative power against other frequencies is much lower. The signal is completely corrupted in time domain and impossible to identify as can be seen in Figure 42b, however using the spectrogram results this considerably high noise power is avoided in signal identification.

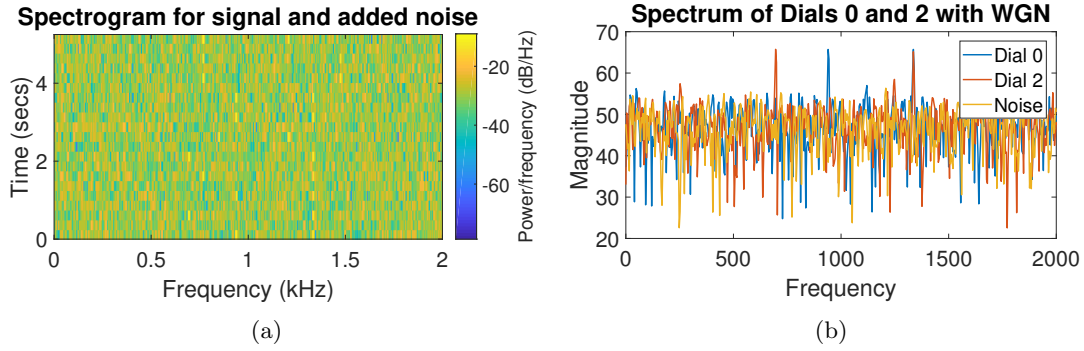


Figure 44: Spectrogram and FFT plot of dial number with  $\sigma = 5$

Finally, for  $\sigma = 15$  the signal is completely immersed in noise and no identification is possible. At this point noise power is too high and not even analysis using the spectrogram is able to identify the signal. Thus, under this noise corruption the signal is impossible to identify with the resources available.

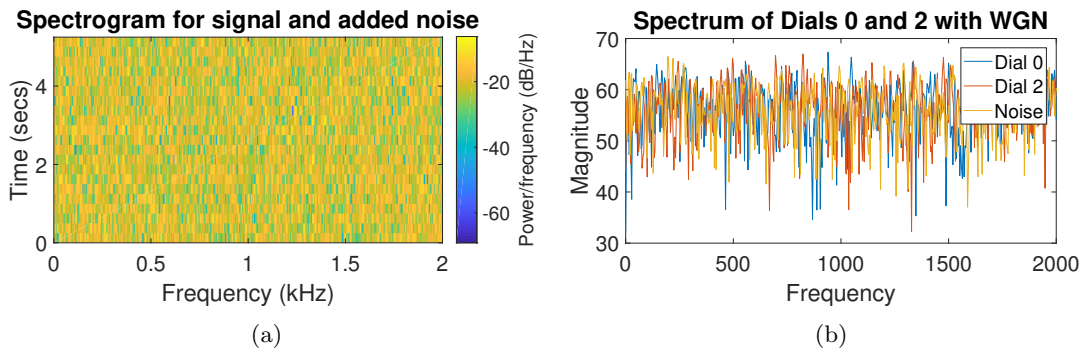


Figure 45: Spectrogram and FFT plot of dial number with  $\sigma = 15$

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

Respiratory sinus arrhythmia (RSA) refers to the modulation of cardiac function by respiratory effort. A periodogram analysis on the RRI data obtained from the recorded ECG data can help

point out conditions like this on individuals. Figure 46 shows the estimated PSD of the RRI data by means of a periodogram. The analysis was done on the original data and averaged data. Average data was done taking 5 segments of 400 samples, after padding the signal from 970 samples to 2000 samples, as further padding showed no clear improvement.

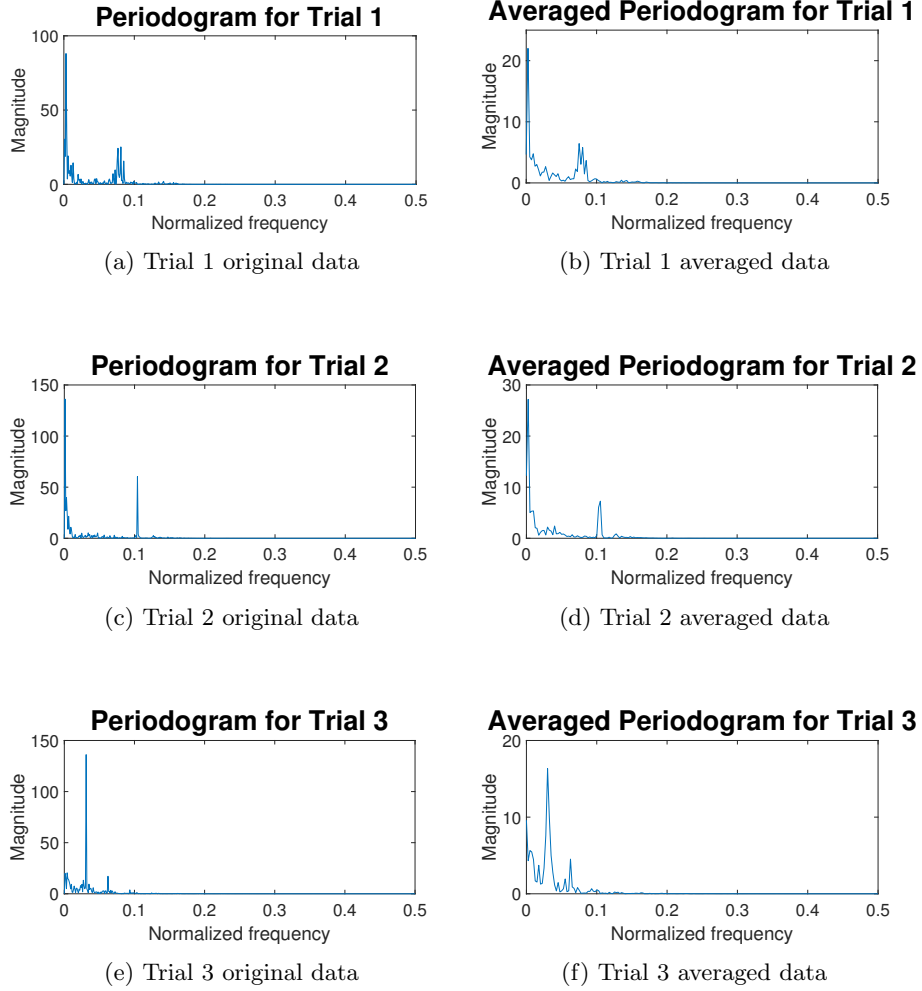


Figure 46: Periodograms of original and averaged RRI data

From Figure 46 it is clear how the averaged signal provides cleaner version of the estimated PSD of the signal. The PSD of the signal should indicate the rate of breathing and hence peaks at higher frequencies would correspond to short breaths whilst low frequency peaks would correspond to long breath periods.

Trial 1's PSD shows a peak at frequency  $0.09(\times 2\pi)$ , but with no huge difference in magnitude with respect to other frequencies. With the unconstrained breathing environment, the person just relaxed and naturally breathed in a somehow periodic manner at this frequency. Taking this into consideration the PSD shows an expected result with in clear frequency dominating.

Trial 2 & 3 however do show a clear dominance by one frequency. Trial 2 shows a clear dominance of frequency  $0.1(\times 2\pi)$  representing the higher constrained breathing at 50 beats per minute. On the contrary Trial 3 shows a peak at frequency  $0.03(\times 2\pi)$  which is even lower than the peak recorded for unconstrained breathing, this clearly shows the nature of the slow breathing constrained at 15 beats per minute.

## 4 Optimal Filtering - fixed and adaptive

Aims:

- To review adaptive system identification.
- To introduce the Wiener filter as the optimal least squares filter for stationary signals.
- To consider online estimation of optimal filter coefficients based upon adaptive filtering in system identification and prediction setting.
- To illustrate the utility of adaptive filtering on real world speech (your own voice).

### 4.1 Wiener filter

In order to obtain the optimum Wiener filter solution and hence estimate the unknown system the equation  $\mathbf{w}_{opt} = \mathbf{R}_{xx}^{-1} \mathbf{p}_{zx}$  is used. Table 47 shows how the obtained optimum weights vary according to noise power ( $\sigma^2$ ) with its SNR also calculated. Results are the average of 100 executions.

Noise Power		$\mathbf{w}_{opt}$				
$\sigma^2$	SNR(dB)	$w_1$	$w_2$	$w_3$	$w_4$	$w_5$
0.1	10	0.9976	2.0170	3.0131	1.9914	1.0108
0.2	6.99	0.9966	2.0240	3.0191	1.9879	1.0152
0.5	3.01	0.9947	2.0378	3.0311	1.9809	1.0240
1	0	0.9926	2.0535	3.0445	1.9730	1.0338
5	-6.99	0.9832	2.1194	3.1013	1.9398	1.0754
10	-10	0.9763	2.1688	3.1439	1.9149	1.1066
30	-14.77	0.9589	2.2924	3.2502	1.8528	1.1845
<b>b</b>		1	2	3	2	1

Figure 47: Optimal weights for different noise power values

The calculated weights follow closely the real values for different values of noise power, showing the accuracy of the Wiener filter to estimate a process. However, it is clearly seen how as noise power is increased the estimated weights become less accurate, clearly noise power has a direct impact on the accuracy of the estimation. Something worth noting is that an observation of the signal suggests most of it to lie within  $\pm 10$  hence suggesting  $3\sigma = 98\%$  of the signal is  $\approx 10$ , and as a result  $\sigma \approx 3$ . This means any noise power which is below this will not affect greatly the estimation as the signal will still dominate.

Additionally, it is observed the effect of increasing  $N_w$  (the maximum lag) on data doesn't change the model prediction. The squared error doesn't change as increasing  $N_w > 4$  results in the additional values being extremely small and hence not making any considerable impact to the estimated system.

One of the main drawbacks of the Wiener Filter is its computational complexity. The need of computing the autocorrelation and cross-correlation functions involves for  $N$  samples,  $N$  multiplications and additions for each of the  $(N+1)$  delay values  $\tau$ . This involves a complexity involving  $N(N+1)$  or  $O(N^2)$  which adds to the predominant  $O(N^3)$  needed for the inversion of the matrix  $\mathbf{R}_{xx}$ .

### 4.2 The least mean square (LMS) algorithm

In real-world situations the studied system may be time-varying, hence producing a nonstationary output, in these cases the Wiener filter is not appropriate and filters such as the Least Mean Square algorithm are used:

$$w[n+1] = w[n] + \mu e[n]x[n] \quad n = 0, 1, \dots \quad (48)$$

When the algorithm is applied to the signals  $x$  and  $z$ , from section 4.1, a set of new weights are obtained. The algorithms applied are different and thus the weights estimates are different, however they both give accurate estimates of the filter coefficients. In this particular case, the system is statistical stationary and hence the Wiener filter performs slightly better than the LMS algorithm, this is due to the fact that the Wiener filter uses all the data, whilst the LMS algorithm only uses the a moving window of samples. However in a time-varying system the Wiener filter would not be able to perform correctly and hence the LMS algorithm would perform significantly better.

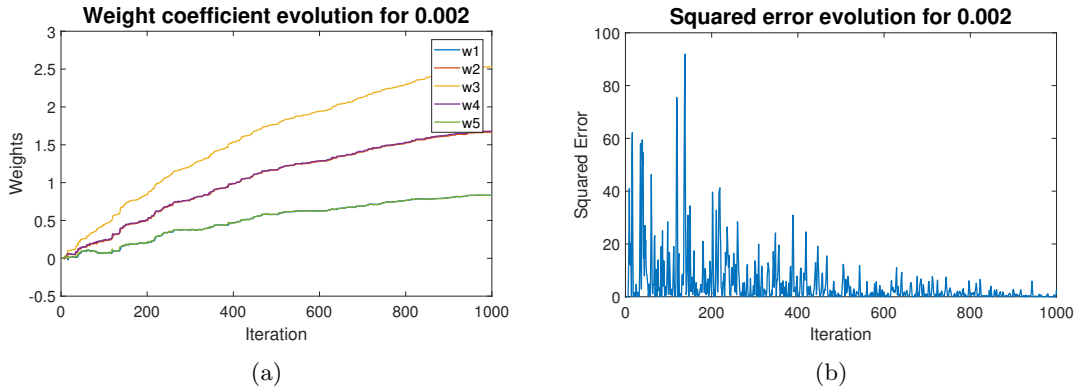


Figure 48: Evolution of coefficient estimates and squared error for  $\mu = 0.002$

Figure 48 shows the coefficient evolution and MSE for  $\mu = 0.002$ . From 48 it can be seen how the estimates do not reach the actual value, as the learning rate  $\mu$  is too small. This is backed up by the plotting of the MSE, that gradually decays as the coefficient estimates approach the filter coefficient values. This suggest the learning rate is too slow and hence it does not converge to the real values.

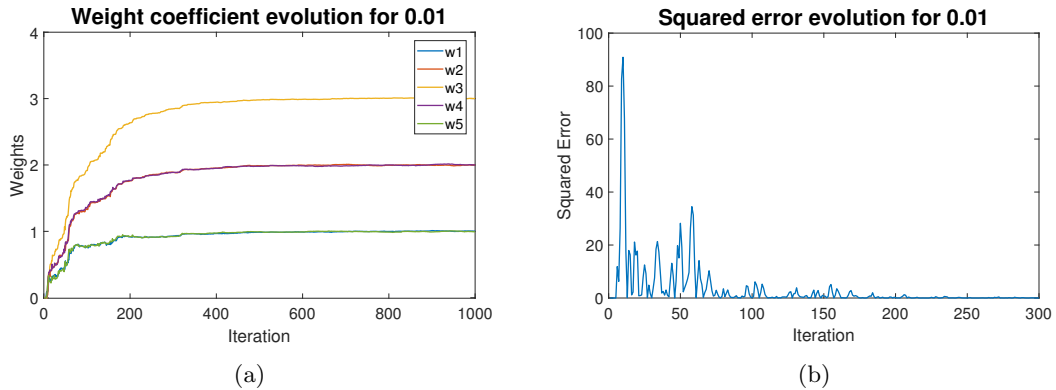


Figure 49: Evolution of coefficient estimates and squared error for  $\mu = 0.01$

For  $\mu = 0.01$  the coefficient estimates converge much faster and reach the actual values within 400 updates. The MSE also converges to zero much faster as can be seen in Figure 49b, showing only the first 300 updates. This suggests a value of  $\mu = 0.01$  is an appropriate value as it converges considerably quickly reaching its value with little error.

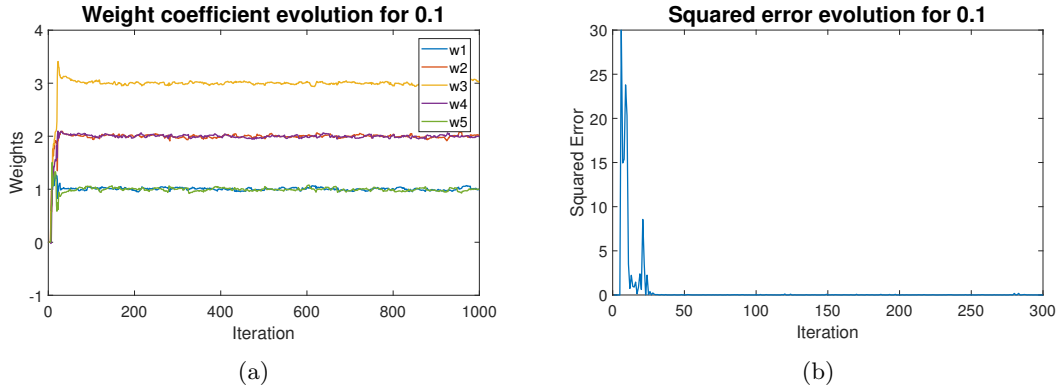


Figure 50: Evolution of coefficient estimates and squared error for  $\mu = 0.1$

$\mu = 0.1$  shows a similar behavior to  $\mu = 0.01$  but with a change in trade-off. The values of the coefficient estimates are within their actual values in less than 100 updates, however the estimates do not converge exactly to the filter coefficients and instead vary around this value, hence not giving zero error after certain amount of updates. The quick convergence can be seen on the MSE graph where the error is only considerable for less than 30 updates and then decays very rapidly. From this analysis the use of  $\mu = 0.1$  can be considered to be slightly high, as the learning rate is very fast, however it is too fast and the step size is too big for it to reach to the optimal values.

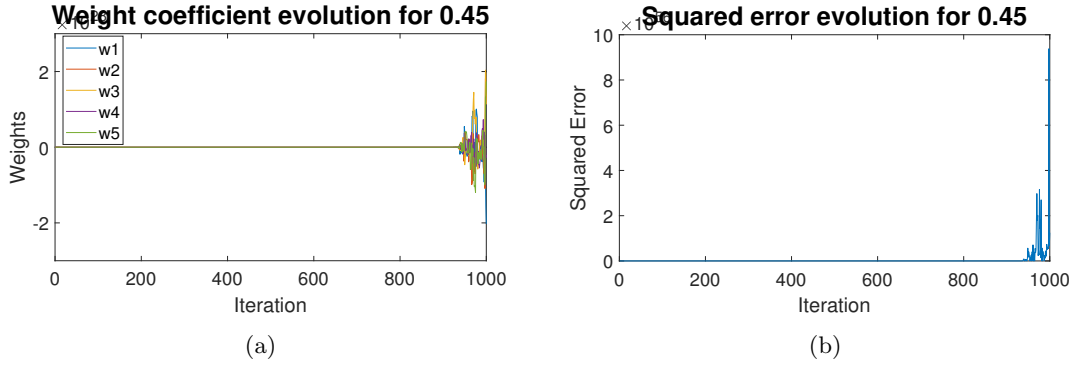


Figure 51: Evolution of coefficient estimates and squared error for  $\mu = 0.45$

For  $\mu = 0.5$  a whole different picture is presented, the learning rate is too big and the step size prevents the values from converging. The MSE shows a similar behavior without any trend towards convergence, showing the step size is too big and hence every step the weights are overestimated and overpassing the local minimum of the cost function.

From the analysis for different values of  $\mu$  it is clear a trade-off is present between speed of convergence and accurateness of the result. This sets the optimum learning rate to be a middle ground between 0.002 and 0.5 (eg,  $\mu = 0.02$ ), thus attempting to achieve a good compromise that provides the benefits of both extremes, whilst avoiding its disadvantages.

The computational complexity of this algorithm is considerably better than the Wiener filter. On each update it is required  $(N_w + 1)$  multiplications for the dot product to produce the estimate of  $\mathbf{y}$ ,  $\hat{\mathbf{y}}$ , and the last  $(N_w + 1)$  values of  $\mathbf{x}$ , on top of this  $N_w$  multiplications and  $N$  additions are needed to update the weight vector. Over  $N$  iterations this gives a computational complexity of  $O(N(N_w + 1))$ , considerably less than the  $O(N^3)$  of the Wiener filter.

### 4.3 Gear shifting

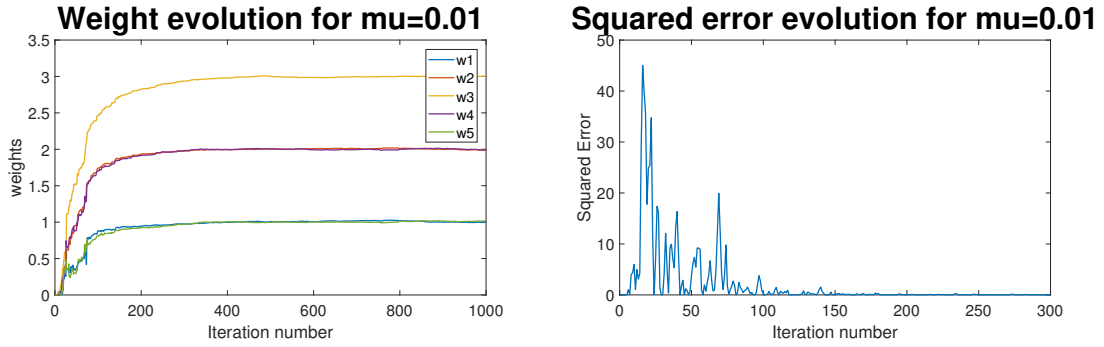
For statistical stationary environments it would be beneficial to use gear shifting, that is to vary the learning rate accordingly to converge quicker to the solution. By relating the adaptation rate to the



gradient of the error function, a progressive reduction in adaptation rate occurs until the objective value is reached. This method manages to converge towards the objective values at a fast rate, due to large rate at the beginning, whilst still being able to reach to a final solution within a very small margin, due to the slow adaptation rate towards the end. In order to relate the adaptation to the error gradient we will implement the LMS algorithm such that if  $error(n) \leq error(n-1)$  then the adaptation rate is set to:

$$\mu(n) = \mu(n-1) - \rho(e(n) - e(n-1)) \quad (49)$$

This way the adaptation rate is set to match the gradient of the error function and hence change accordingly, the bigger the difference the larger the step size. Figure 52a shows the convergence of the coefficients showing how the step sizes gradually decrease as approaching the objective value, eventually reaching the coefficient value within a negligible error. In addition Figure 52b shows the squared error of the coefficient estimates with every iteration, a direct comparison with Figure 49b shows how the gear shifting modification performs much better converging a much faster rate. Both plots reinforce the usefulness of the Gear Shifting algorithm, not only converging quickly towards the coefficient value, but also approaching the values very accurately.



(a) Coefficient estimates for Gear Shifting Algorithm (b) Squared Error for Gear Shifting Algorithm

Figure 52

#### 4.4 Identification of AR processes

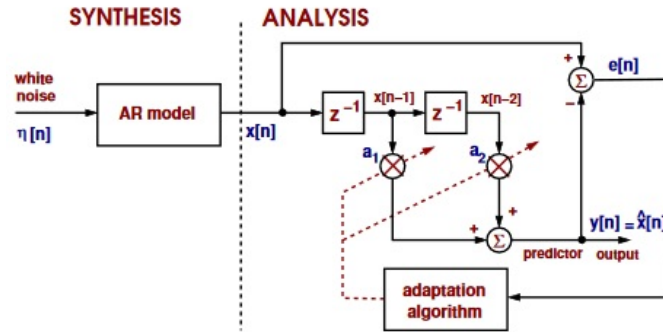


Figure 53: Synthesis and analysis structure for an AR model

Figure 53 is now implemented to use the LMS algorithm to estimate an AR model with coefficients  $a = [1 \ 0.9 \ 0.2]$ . For this values of  $a$  the algorithm should converge to  $-0.9$  and  $-0.2$ . The implementation will now be tested by varying the adaptation gain for four different values, for a WGN signal of 1000 samples. Figure 54a shows how for  $\mu = 0.01$  the filter does converge to the AR coefficients, however there is considerable variations making it imprecise. When  $\mu = 0.001$  the

system does not have time to converge and hence gives completely wrong estimates, as can be seen in Figure 54b.

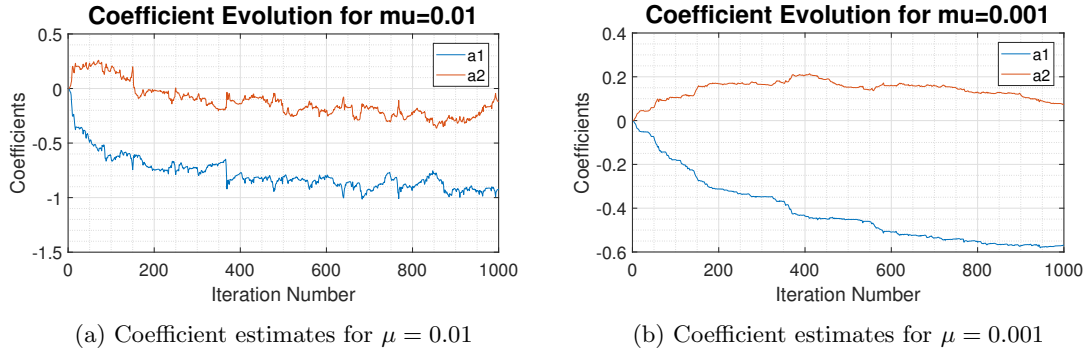


Figure 54

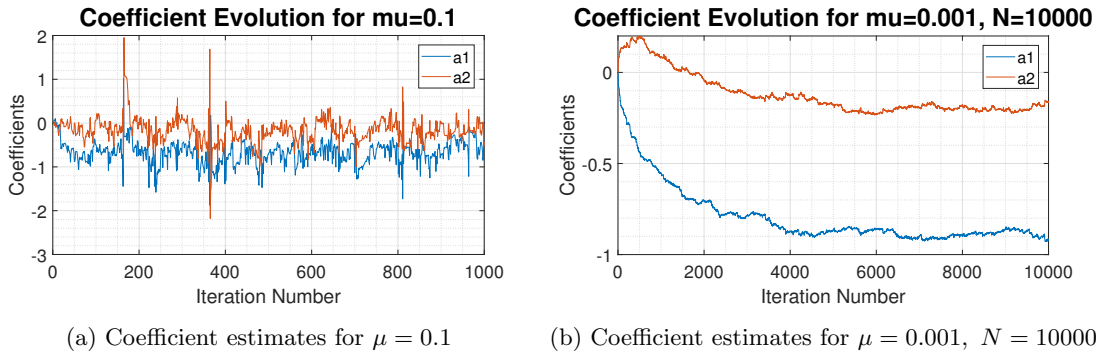


Figure 55

Figure 55a shows the implementation for an adaptation rate of 0.1, which clearly shows how given a certain value the step size is too big and gives increased variations between iterations, the model doesn't manage to converge to the values and the variation through the process is noticeable, similarly for bigger values of  $\mu$  such as 0.5 the system no longer performed correctly and gave inadequate estimates. Finally it was observed how if the number of samples is increased to 10000, and  $\mu = 0.001$ , the system converges without any huge variations as the step size is small and has enough time to converge, Figure 55b.

## 4.5 Speech recognition

The algorithm used in the previous section was used this time to predict an AR model that would fit the spoken letters 'e', 'a', 's', 't' and 'x', for a sampling frequency  $f_s = 44100$  and  $N = 1000$ . As can be seen in Table 54, the different letters varied greatly and hence the ideal parameters do not follow any trend. Gear shifting improved the result considerably as  $\mu$  could change accordingly.

Letter	Initial $\mu$	Order
e	0.5	26
a	20	50
s	1000	44
t	5	30
x	0.5	9

$\mu$  was found empirically and follows no general trend. Model order however did depend greatly on the letter, vowels especially would tend towards the highest order as this would follow the data

the best, Figure 56b. However as can be seen in Figure 56a higher order models over-fit the data and just follow the trend but don't replicate the details. Figure 56a and 56b show the prediction models and mean error vs order for letter 'a'.

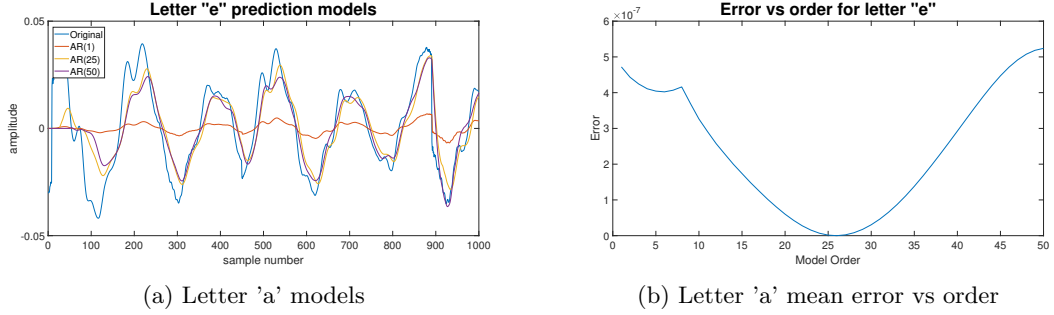
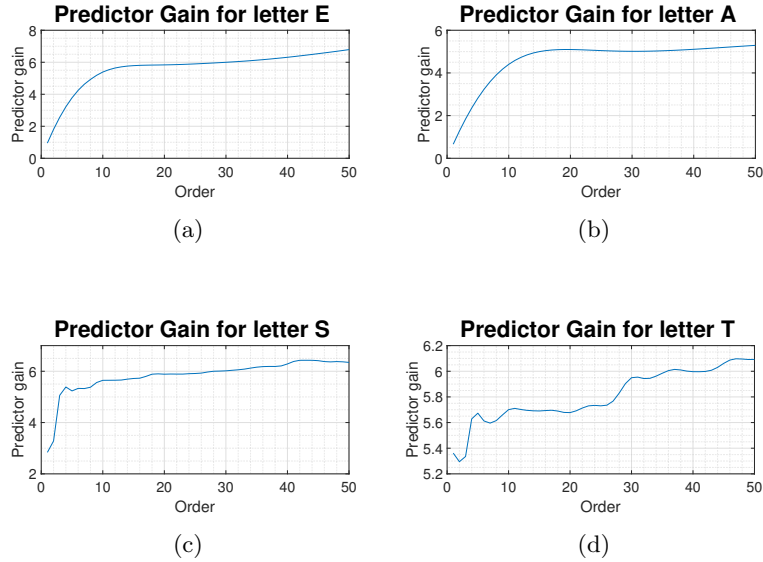
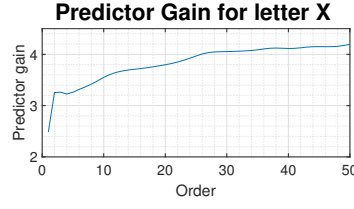


Figure 56

It is clear that finding the optimal model order (length) is a complicated task as higher order models fit the trend best but tend to over-fit the data. Analytical approaches such as MDL and AIC analysis, would provide a measure of the optimal filter length, by obtaining a trade-off between error and model order (over-fitting).

The predictor gain is defined as  $R_P = 10 \log_{10} \left( \frac{\sigma_s^2}{\sigma_e^2} \right)$  for AR models, on the input audio signal of length 1000 samples and  $f_s = 44100\text{Hz}$ . For the vowels 'e' and 'a' the predictor gain is very smooth against order, and becomes stable after model order 12, suggesting this to be the model order as higher order models will not make much difference. Consonants all follow a general trend stabilizing after order 6. 's' and 'x' show a very similar plot, since their pronouncing is similar, and gradually increases from order 6 at a slow rate. Letter 't' however has two 'jumps' at orders 6 and 28, suggesting the final order 28 is the ideal order as after that no significant changes occur.





(e)

Figure 57: Predictor gains for letter 'e', 'a', 's', 't', 'x'

If a similar analysis is done but this time with a sampling frequency  $f_s = 16000\text{Hz}$ , the following results are obtained for letters 'e' and 'a', Figure 58. Consonants are non-stationary and hence reducing the sampling frequency gives worse results so none of them are plotted. The results clearly give smaller values of prediction gain. By reducing the sampling frequency the resolution of the signal is being reduced and hence noise is harder to identify than for  $f_s = 44100\text{Hz}$ . This is a challenge for AR modelling and as a result results are significantly worse. To avoid this signals with more samples could be used, however this would not really be suitable for on-line learning as letters would have to last longer than normal speaking.

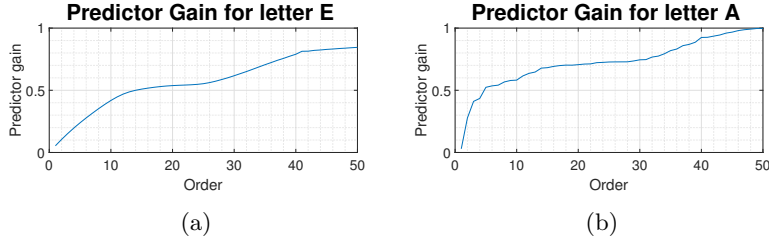


Figure 58: Predictor gain for letters 'e' and 'a',  $f_s = 16000\text{Hz}$

## 4.6 Dealing with computational complexity: sign algorithms

The above algorithm is considerably computational expensive and hence other algorithms involving just the sign of the components are considered to achieve efficiency.

Different sign algorithms were used on the AR filtered signal from section 4.4, in order to estimate the filter coefficients. Figure 59 shows the coefficient evolution for all three different algorithms and the original one. The original one is the best one as it converges quicker, this is because it is more computational expensive, thus performing the task better. The sign-sign algorithm performs as expected the worse, with the low complexity of only using the signs it is computationally efficient but the result is obviously not as good. By comparison it can be seen how sign-error converges faster than sign-regressor, meaning the information in the error vector has more useful information for the weight updates than the regressor vector.

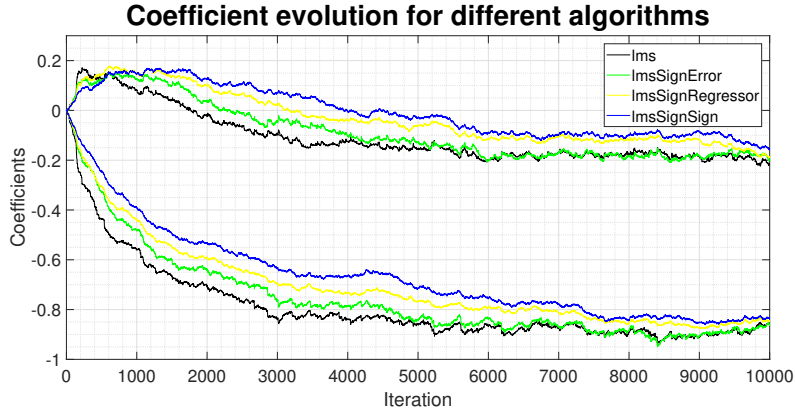


Figure 59: Coefficient evolution for different sign algorithms

The same algorithms were now used for AR order identification on the audio signal for the vowel 'e'. The sign-sign algorithm performed the worst out of all the different algorithms and only gave a smooth result for very small values of  $\mu$ . Only using the sign of the error and regressor vectors proves not to be a good way of fighting computational complexity as the results differ considerably from the real ones. For the Predictor Gain unlike for coefficient evolution, the sign-regressor performed better than the sign-error, suggesting that maybe it has a better result even if it converges slower. None of the three algorithms performed satisfactory to simulate the original algorithm in terms of prediction gain, all three were scaled factors of the original and  $\mu$  had to be modified accordingly to obtain a clear plot, Figure 60. It is clear though that all algorithms suggest an order of 10, by when the predictor gain doesn't change much for higher orders.

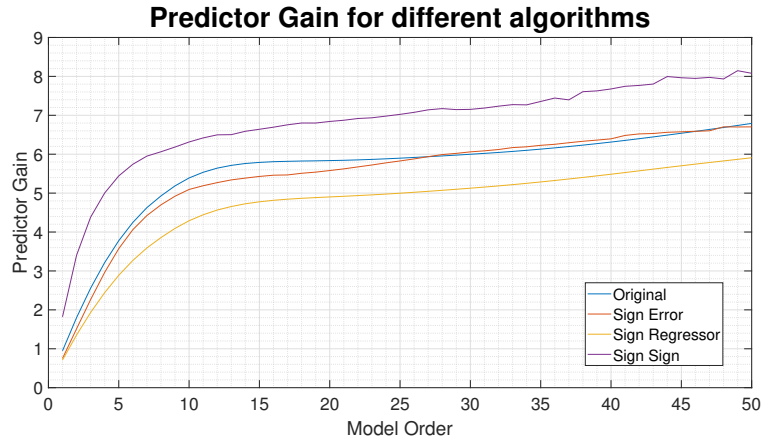


Figure 60: Prediction Gain for different sign algorithms

## 5 MLE for the Frequency of a Signal

The Maximum Likelihood Estimator (MLE),  $\hat{f}_0$  of the frequency  $f_0$ , can be obtained from the pdf of a signal, by minimizing the cost function:

$$J(\theta) = \sum_{n=0}^{N-1} (x[n] - A \cos(2\pi f_0 n + \phi))^2 \quad (50)$$

By expanding  $A \cos(2\pi f_0 n + \phi)$  as in Equation 51, setting  $\alpha_1 = A \cos \phi$  and  $\alpha_2 = -A \sin \phi$ , and similarly for  $\mathbf{c}$  and  $\mathbf{s}$ ,  $\mathbf{c} = [1, \cos(2\pi f_0), \dots, \cos(2\pi f_0(N-1))]^T$  and  $\mathbf{s} = [0, \sin(2\pi f_0), \dots, \sin(2\pi f_0(N-1))]^T$ . The cost function can be expressed as  $J'(\alpha_1, \alpha_2, f_0) = (\mathbf{x} - \alpha_1 \mathbf{c} - \alpha_2 \mathbf{s})^T (\mathbf{x} - \alpha_1 \mathbf{c} - \alpha_2 \mathbf{s})$  which is equivalent to the form  $J'(\alpha, f_0) = (\mathbf{x} - \mathbf{H}\alpha)^T (\mathbf{x} - \mathbf{H}\alpha)$ .

$$\begin{aligned} A \cos(2\pi f_0 n + \phi) &= A \cos(2\pi f_0 n) \cos(\phi) - A \sin(2\pi f_0 n) (-\sin(\phi)) \\ &= \alpha_1 \cos(2\pi f_0 n) + \alpha_2 \sin(2\pi f_0 n) \end{aligned} \quad (51)$$

Therefore,  $J(\theta)$  can be mapped to a form  $(\mathbf{x} - \mathbf{H}\alpha)^T (\mathbf{x} - \mathbf{H}\alpha)$  where  $\mathbf{H} = [\mathbf{c}, \mathbf{s}]$  and  $\alpha = [\alpha_1, \alpha_2]$ . Such that if we express it as before in terms of  $\alpha_1$  and  $\alpha_2$ ,  $(x[n] - A \cos(2\pi f_0 n + \phi))$  is mapped to:

$$\begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[N-1] \end{bmatrix} - \alpha_1 \begin{bmatrix} \cos(0) \\ \cos(2\pi f_0) \\ \vdots \\ \cos(2\pi f_0(N-1)) \end{bmatrix} - \alpha_2 \begin{bmatrix} \sin(0) \\ \sin(2\pi f_0) \\ \vdots \\ \sin(2\pi f_0(N-1)) \end{bmatrix} \quad (52)$$

In order to find the estimators that minimize the cost function, the derivative of the function is done and set to zero. This procedure gives an optimum estimator of  $\alpha = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{x}$ , by expanding the cost function and putting this value in we clearly see how:

$$\begin{aligned} J'(\alpha, f_0) &= (\mathbf{x} - \mathbf{H}\alpha)^T (\mathbf{x} - \mathbf{H}\alpha) = \mathbf{x}^T \mathbf{x} - 2\mathbf{x}^T \mathbf{H}\alpha + \alpha^T \mathbf{H}^T \mathbf{H}\alpha \\ &= \mathbf{x}^T \mathbf{x} - 2\mathbf{x}^T \mathbf{H}(\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{x} + ((\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{x})^T \mathbf{H}^T \mathbf{H}(\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{x} \\ &= \mathbf{x}^T \mathbf{x} - 2\mathbf{x}^T \mathbf{H}(\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{x} + \mathbf{x}^T \mathbf{H}(\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{x} \\ &= \text{constant} - \mathbf{x}^T \mathbf{H}(\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{x} \end{aligned} \quad (53)$$

The cost function converges to the result in Equation 53, from this it can be seen how the cost is minimized if  $\mathbf{x}^T \mathbf{H}(\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{x}$  is maximized, as it will tend to turn the cost function negative. Knowing  $\mathbf{H} = [\mathbf{c}, \mathbf{s}]$  it can be seen, Equation 54, how the cost function also minimizes by maximizing the last line in Equation 54.

$$\begin{aligned} \mathbf{x}^T \mathbf{H} &= [x(0) \quad x(1) \quad \dots \quad x(N-1)] [\mathbf{c} \quad \mathbf{s}] = \begin{bmatrix} \mathbf{c}^T \mathbf{x} \\ \mathbf{s}^T \mathbf{x} \end{bmatrix} \\ \mathbf{H}^T \mathbf{H} &= \begin{bmatrix} \mathbf{c} \\ \mathbf{s} \end{bmatrix} [\mathbf{c} \quad \mathbf{s}] = \begin{bmatrix} \mathbf{c}^T \mathbf{c} & \mathbf{c}^T \mathbf{s} \\ \mathbf{s}^T \mathbf{c} & \mathbf{s}^T \mathbf{s} \end{bmatrix} \\ \mathbf{x}^T \mathbf{H}(\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{x} &= \begin{bmatrix} \mathbf{c}^T \mathbf{x} \\ \mathbf{s}^T \mathbf{x} \end{bmatrix}^T \begin{bmatrix} \mathbf{c}^T \mathbf{c} & \mathbf{c}^T \mathbf{s} \\ \mathbf{s}^T \mathbf{c} & \mathbf{s}^T \mathbf{s} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{c}^T \mathbf{x} \\ \mathbf{s}^T \mathbf{x} \end{bmatrix} \end{aligned} \quad (54)$$

The final line in Equation 54 shows the value to maximize to minimize the cost function, however this value is subject to  $f_0$  not being close to 0 or  $\frac{1}{2}$ , this is because for those values all the sine components in the calculation would go to 0, hence meaning  $\mathbf{H}^T \mathbf{H}$  is singular and non-invertible, meaning the operation is not possible to perform. Additionally, if we multiply out the approximation that  $\mathbf{H}^T \mathbf{H}$ 's values become  $[\mathbf{H}^T \mathbf{H}]_{11} = [\mathbf{H}^T \mathbf{H}]_{22} = N/2$  and  $[\mathbf{H}^T \mathbf{H}]_{12} = [\mathbf{H}^T \mathbf{H}]_{21} =$

0, we can see how the result is:

$$\begin{aligned}
\begin{bmatrix} \mathbf{c}^T \mathbf{x} \\ \mathbf{s}^T \mathbf{x} \end{bmatrix}^T \begin{bmatrix} N/2 & 0 \\ 0 & N/2 \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{c}^T \mathbf{x} \\ \mathbf{s}^T \mathbf{x} \end{bmatrix} &= (2/N)(\mathbf{c}^T \mathbf{x})^T \mathbf{c}^T \mathbf{x} + (2/N)(\mathbf{s}^T \mathbf{x})^T \mathbf{s}^T \mathbf{x} \\
&= \frac{2}{N} \sum_{n=0}^{N-1} (x[n] \cos(2\pi f_0 n))^2 + \frac{2}{N} \sum_{n=0}^{N-1} (x[n] \sin(2\pi f_0 n))^2 \quad (55) \\
&= \frac{2}{N} \left| \sum_{n=0}^{N-1} x[n] e^{-j2\pi f_0 n} \right|^2
\end{aligned}$$

This corresponds to maximizing the periodogram, the  $\frac{2}{N}$  factor is just a constant and has no effect. However it is different for the missing  $N$ . In the periodogram we have discrete values as  $P(f_k)$  is the periodogram just with the complex exponential being  $e^{-j2\pi f_k n}$ .  $2\pi f_k$  are just  $N$  uniformly spaced samples around the unit circle,  $f_k = \frac{k}{N}$ . Hence  $P(f_k) = P(\frac{k}{N}) = P(k)$ . If we consider  $f_k = f_0$  and thus substitute this result into the result of Equation 55, we can see how maximizing the result is the same as maximizing the periodogram.

Using the noiseless data  $x[n] = \cos(2\pi f_0 n)$  where  $n = 1, 2, \dots, 9$ , a direct comparison can be made between the periodogram and the calculations done above. Figure 61 shows the periodogram and the plotting of  $\mathbf{x}^T \mathbf{H}(\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{x}$  for various values of  $f$ .

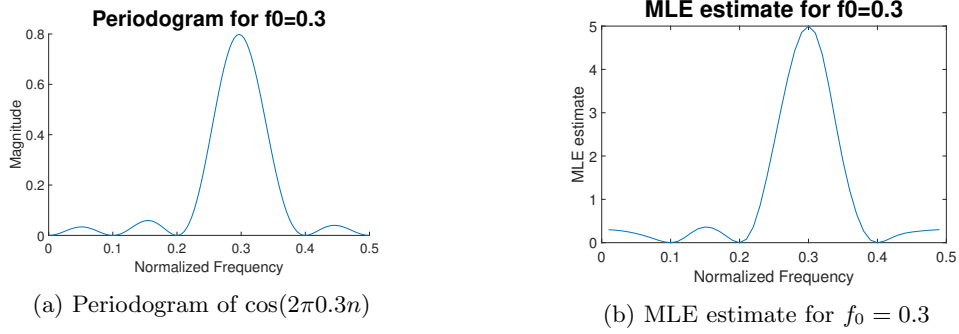


Figure 61

As can be seen both the periodogram and the MLE estimate match in the maximum value at  $f = 0.3$ , thus matching the value of  $f_0$ , and proving the usefulness of the periodogram when finding the MLE solution. As  $f_0$  approaches 0 or  $\frac{1}{2}$  as mentioned above the sine values in  $\mathbf{s}$  go to zero and  $\mathbf{H}^T \mathbf{H}$  becomes singular, hence the MLE estimate becomes unsuitable. Figure 62 shows the periodogram and MLE estimate for  $f_0 = 0.0001$ , and it can be observed how the MLE estimate settles around this value however doesn't show a clear maximum and diverges in magnitude (tends to 10 for  $f_0 = 0.001$ , and to 5 for  $f_0 = 0.3$ ), which makes it unsuitable for frequency estimation. This is clearly seen in Figure 62b.

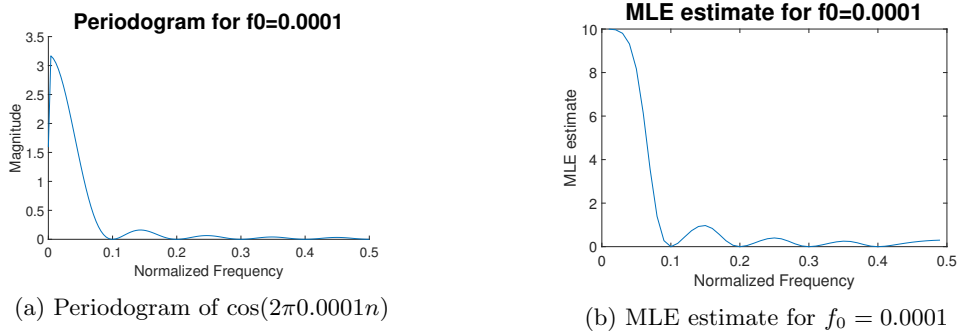


Figure 62