

IMPERIAL COLLEGE LONDON

ADVANCED SIGNAL PROCESSING

*Pranav Malhotra, CID: 00823617*

January 21, 2017

# Contents

<b>1 Random signals and stochastic processes</b>	<b>2</b>
1.1 Statistical Estimation . . . . .	2
1.2 Stochastic Processes . . . . .	5
1.3 Estimation of probability distributions . . . . .	8
<b>2 Linear Stochastic Modelling</b>	<b>9</b>
2.1 ACF of Uncorrelated Sequences . . . . .	9
2.2 ACF of Correleated Sequences . . . . .	10
2.3 Cross-Correlation Function . . . . .	10
2.4 Autoregressive Modelling . . . . .	11
2.5 Real World Signals: ECG from iAmp Experiment . . . . .	14
<b>3 Spectral Estimation and Modelling</b>	<b>18</b>
3.1 Averaged Periodogram Estimate . . . . .	19
3.2 Spectrum of Autoregressive Process . . . . .	20
3.3 Spectrogram for Time-Frequency Analysis: Dial Tone Pad . . . . .	22
3.4 Real World Signals: Respirator Sinus Arrhythmia from RR-Intervals . . . . .	25
<b>4 Optimal Filtering - Fixed and Adaptive</b>	<b>27</b>
4.1 Wiener Filter . . . . .	27
4.2 Least Mean Square (LMS) Algorithm . . . . .	28
4.3 Gear Shifting . . . . .	30
4.4 Identification of AR Process . . . . .	31
4.5 Dealing With Computational Complexity: Sign Algorithms . . . . .	32
<b>5 A Real World Case Study: Vinyl Denoising</b>	<b>33</b>

# 1 Random signals and stochastic processes

## 1.1 Statistical Estimation

To explore random signals and stochastic processes, it is of great importance that we learn how to generate random signals. Below, is a plot of a vector that consists of a 1000 samples, each of which is an independent realisations of the random variable  $X \sim \mathcal{U}(0, 1)$ . The vector is first graphed using the `plot` function, which returns a continuous-time description of the signal; it is formed by linear interpolation between the samples. If we wish to interpret the samples as a discrete-time signal, we use the `stem` function.

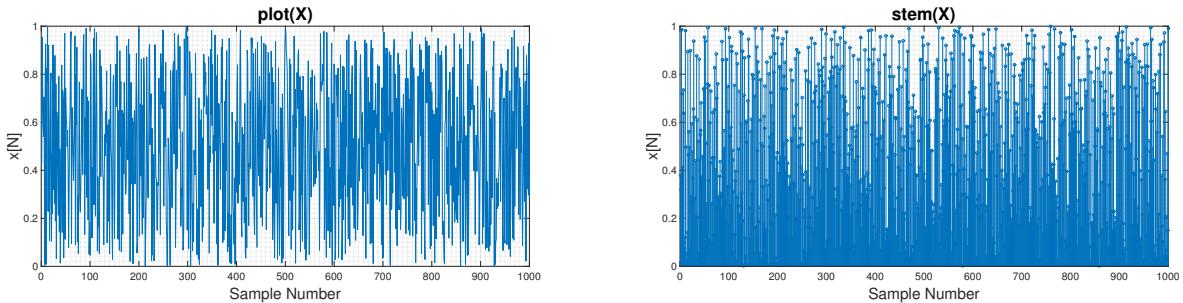


Figure 1: Plot and stem of a 1000 samples drawn from  $X \sim \mathcal{U}(0, 1)$

From the sequence, it is clear that there is some sort of uniformity although the samples are completely independent. This uniformity stems from the fact that the samples are independent realisations of the same random variable and thus the mean and variance of the signal is time-invariant.

1. To calculate the theoretical mean of the random variable  $X \sim \mathcal{U}(0, 1)$ , the statistical expectation operator is employed.

$$\text{Theoretical Mean} = \mathbb{E}\{X\} = \int_{-\infty}^{\infty} x f_X(x) dx = 0.5$$

In MATLAB, the `mean` function returned a value of 0.4888. The calculated sample mean,  $\hat{m}$  is a function of the random samples; we can represent the sample mean as a random variable,  $\hat{M}$  whose distribution depends on the underlying distribution of the individual samples. The calculated sample mean,  $\hat{m}$  is a single realisation of the random variable  $\hat{M}$ . If the samples are drawn from a distribution with a mean,  $\mu$  and a variance,  $\sigma^2$ , then the sample mean will have a mean,  $\mu$  and a variance,  $\sigma^2/N$ , where  $N$  is the number of samples. It is clear that the sample mean is an unbiased estimator of the theoretical mean and that its accuracy increases with the number of samples. Also, it is interesting to note that by the *central limit theorem* (CLT), if a sufficiently large number of samples is used, the random variable  $\hat{M}$  will follow a Gaussian distribution such that  $\hat{M} \sim \mathcal{N}(\mu, \sigma^2/N)$ .

2. To calculate the theoretical variance of the random variable  $X \sim \mathcal{U}(0, 1)$ , the statistical expectation operator is once again employed. The argument of the operator is slightly different.

$$\text{Theoretical Variance} = \mathbb{E}\{(X - \mu)^2\} = \mathbb{E}\{X^2\} - \mathbb{E}\{X\}^2 = \int_{-\infty}^{\infty} x^2 f_X(x) dx - \mathbb{E}\{X\}^2 = \frac{1}{12}$$

$$\text{Theoretical Standard Deviation, } \sigma = \sqrt{1/12} \approx 0.2887$$

In MATLAB, the `std` function returned a value of 0.2832. The calculated sample standard deviation,  $\hat{\sigma}$  is also a function of the individual samples and thus the sample standard deviation can also be represented by a random variable. The calculations are, however a lot more involved than the calculations for the sample mean. It is important to note that the sample standard deviation is also an unbiased estimator of the theoretical variance. Just like the sample mean, the accuracy of the estimator increases with the number of samples.

3. The sample mean and sample standard deviation are graphed for each member of the ensemble. The plots clearly show that the two estimators are both unbiased as the means and standard deviations hover around their theoretical values.

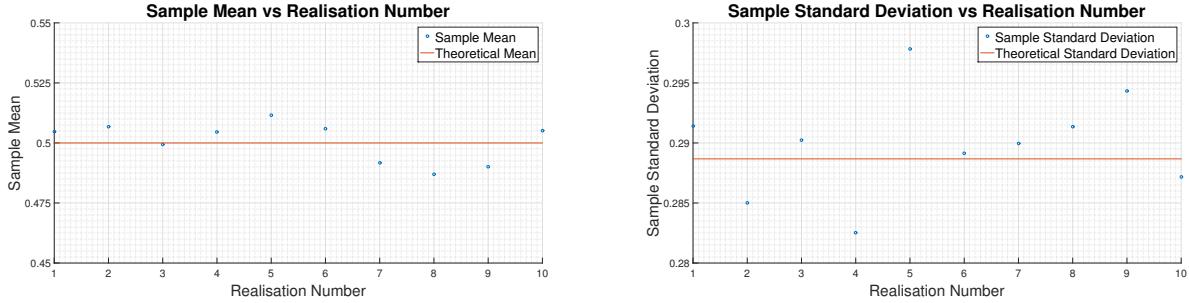


Figure 2: Sample mean and sample standard deviation for each member of the ensemble, samples considered are uniformly distributed

4. Estimating the *probability density function* (PDF) is of great importance in statistical signal processing. The PDF underlying the random samples is estimated using the `hist` function in MATLAB and is normalised using the `trapz` function. To measure the performance of our PDF estimator, a reference line is included that marks the theoretical PDF of the uniform distribution.

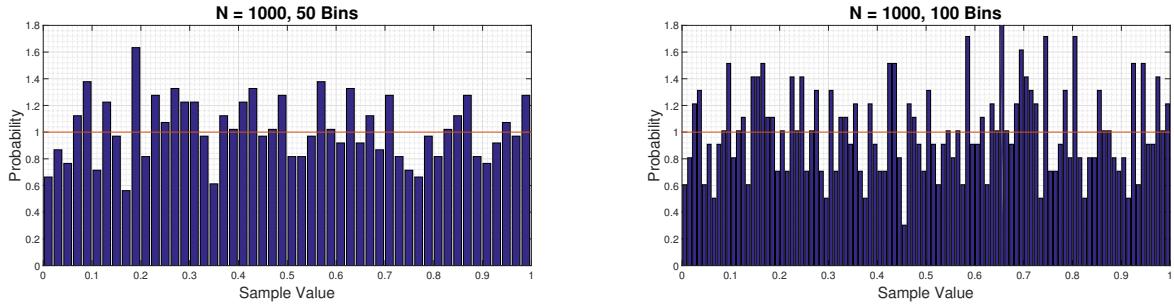


Figure 3: Approximate PDF of random samples drawn from  $X \sim \mathcal{U}(0, 1)$ , small sample size

Increasing the bin size will result in better resolution of the estimated PDF, however arbitrarily increasing the bin size is not wise and will result in unclear results. Increasing the sample size significantly improves the estimate and the result approaches the theoretical PDF.

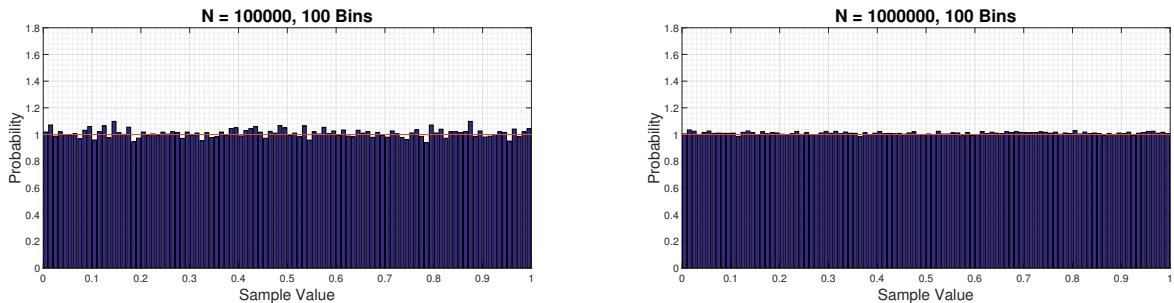


Figure 4: Approximate PDF of random samples drawn from  $X \sim \mathcal{U}(0, 1)$ , large sample size

5. The PDF of a Gaussian random variable  $X \sim \mathcal{N}(\mu, \sigma^2)$ , where  $\mu$  and  $\sigma$  are the mean and standard deviation of the random variable respectively, is defined as

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

When we use the `randn` function in MATLAB, we are drawing samples from a Gaussian distribution with a theoretical mean of 0 and a theoretical standard deviation of 1. A 10-member ensemble of 1000-sample realisations of  $X \sim \mathcal{N}(0, 1)$  is generated. The sample mean and sample standard deviation is calculated for each member of the ensemble and the results are graphed.

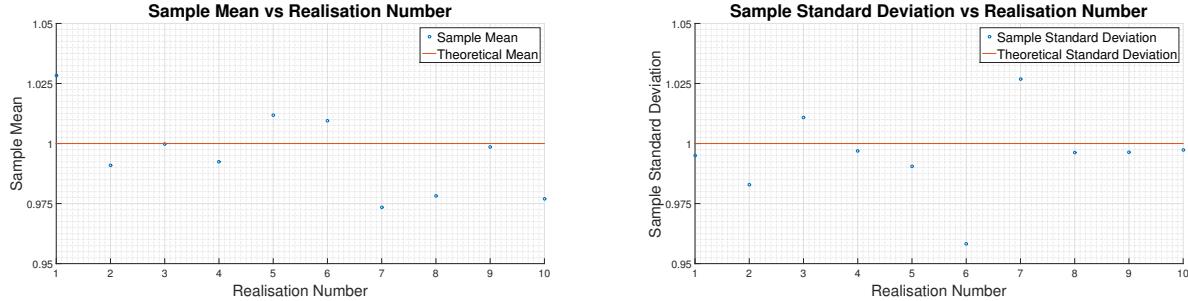


Figure 5: Sample mean and sample standard deviation for each member of the ensemble, samples considered have a Gaussian distribution

Similar to the results obtained when the samples were drawn from a uniform distribution, the means and standard deviations hover around their theoretical values. This further corroborates the fact that the sample mean and the sample standard deviation are unbiased estimators.

Next, the PDF of the underlying distribution that our samples are drawn from, which in this case is the Gaussian distribution, is estimated. Again as a performance measure, a plot of the PDF of a zero-mean unit standard deviation Gaussian distribution is included.

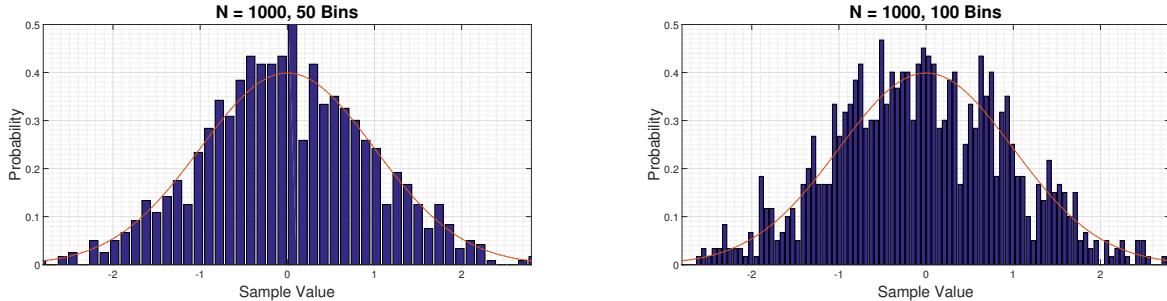


Figure 6: PDF underlying random samples using `hist`, varying bin size

The number of bins is clearly related to the resolution of the estimate whereas the number of samples will determine the estimator's accuracy.

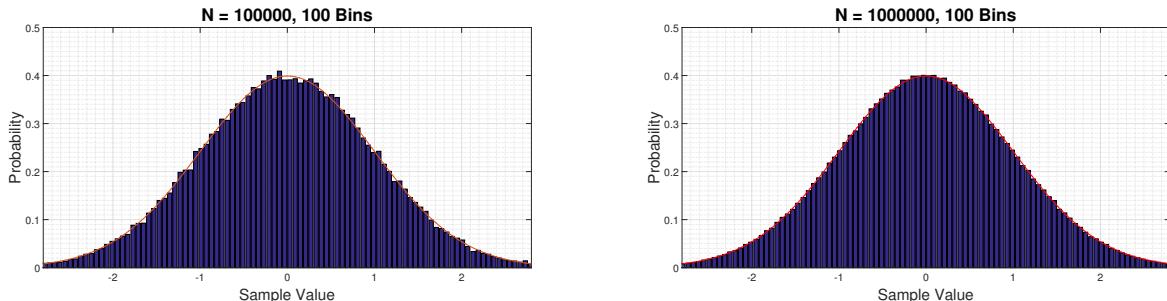


Figure 7: PDF underlying random samples using `hist`, varying sample size

## 1.2 Stochastic Processes

A random process that has time-invariant statistical properties is referred to as statistically stationary. White Gaussian noise,  $w[n]$  is an example of a stationary process whereas a sine wave of the form  $w[n]\sin(2\pi Fn)$  is not stationary as the expected value of the process depends on the time instance at which the process is being observed.

A random process processes ergodicity if a single realisation of the process of sufficient length is enough to determine the statistical properties of the process. For an ergodic signal, the time average will approach the ensemble average as the number of samples are increased. To fully grasp the idea of ergodicity, we consider a process, that once initialised at a random value  $x_i$ , which is a single realisation of the random variable  $X \sim \mathcal{U}(0, 1)$ , will retain its value for all time. The time average of this signal will be  $x_i$ , regardless of the number of samples that are considered. The time average however does not reveal the randomness inherent in the process. This process is thus, not ergodic. To formalise the concept of ergodicity, a mathematical description of a ergodic process is provided in equation (2).

$$\lim_{N \rightarrow \infty} \hat{m}_x(N) = m_x \quad (2)$$

In equation (2),  $\hat{m}_x$  is the sample mean of the signal, which can be thought of as the time-average of the random process, and  $m_x$  is the ensemble average which is calculated by averaging across realisations of the random process. The ensemble average is mathematically formalised in equation (3), where  $x_i[n]$  is the  $n^{\text{th}}$  sample of the  $i^{\text{th}}$  ensemble.

$$m_x = \frac{1}{M} \sum_{i=1}^M x_i[n] \quad (3)$$

1. The ensemble mean and standard deviations are graphed for each of the three process as a function of the sample number.

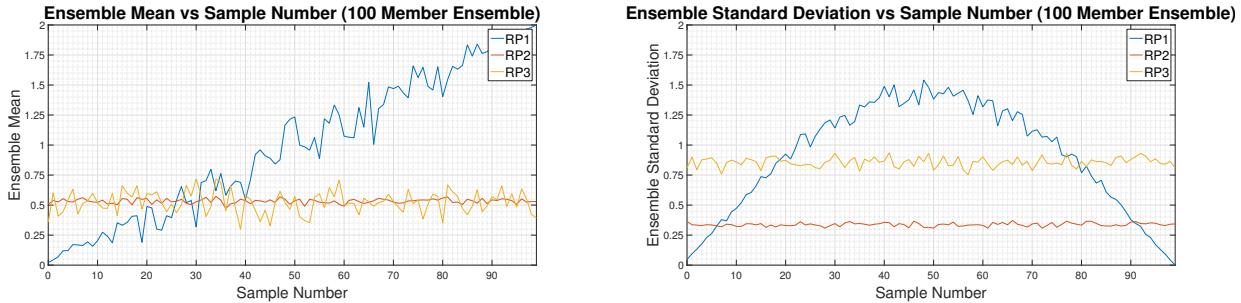


Figure 8: Ensemble means and standard deviations of each of the three random processes

It is clear that for random process 1, the mean and standard deviation are not time-invariant. There is an increasing trend in process's mean and a parabolic trend in its standard deviation. Thus, it is not a stationary process. Random process 2 and 3 are stationary as their ensemble means and standard deviations do not follow any trend and hover around a single value, making them time-invariant.

2. A 4-member ensemble is generated for each of the random processes and the sample mean and sample standard deviation is graphed for each member of the ensemble.

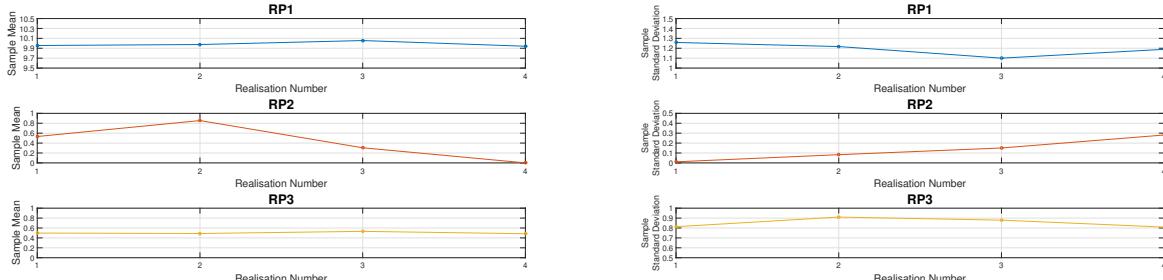


Figure 9: Sample means and standard deviations of each realisation in the 4-member ensemble

As discussed above, for an ergodic process, the sample mean should approach the ensemble mean. This implies that multiple realisations, of infinite length, of an ergodic process should have the same sample mean. The plots above show that the sample means for different realisations of random process 2 vary by significantly more than that of other processes<sup>1</sup>. It is hard to conclude, at this point, if this large variation is due to the fact that we are dealing with finite length sequences or that random process 2 does not possess ergodicity. We can however conclude that since random process 1 is not stationary, the ensemble average is time variant and thus it cannot possess ergodicity. In addition, since the time-averages of multiple realisations of random process 3 are approximately the same, it is highly likely that random process 3 is ergodic. Ergodicity will be conclusive proven once mathematical descriptions of the processes are considered.

3. Mathematical descriptions of the 3 random processes are provided below.

i. Random Process 1

Random Process 1 is defined by equation 4, where  $Z$  is a random variable such that  $Z \sim \mathcal{U}(0, 1)$ ,

$$x[n] = 5(Z - 0.5)\sin\left(\frac{\pi n}{N}\right) + 0.02n \quad (4)$$

To find the theoretical mean of the random process,

$$\begin{aligned} \mathbb{E}\{x[n]\} &= \mathbb{E}\{5(Z - 0.5)\sin\left(\frac{\pi n}{N}\right) + 0.02n\} \\ &= 5\mathbb{E}\{Z - 0.5\}\sin\left(\frac{\pi n}{N}\right) + \mathbb{E}\{0.02n\} \\ &= 5(0.5 - 0.5)\sin\left(\frac{\pi n}{N}\right) + 0.02n \\ &= 0.02n \end{aligned} \quad (5)$$

To find the theoretical variance of the random process,

$$\begin{aligned} Var(x[n]) &= \mathbb{E}\{x[n]^2\} - \mathbb{E}\{x[n]\}^2 \\ &= \mathbb{E}\{(5(Z - 0.5)\sin\left(\frac{\pi n}{N}\right) + 0.02n)^2\} - (0.02n)^2 \\ &= \mathbb{E}\{25(Z - 0.5)^2\sin^2\left(\frac{\pi n}{N}\right)\} + \mathbb{E}\{0.2n(Z - 0.5)\sin\left(\frac{\pi n}{N}\right)\} + \mathbb{E}\{(0.02n)^2\} - (0.02n)^2 \\ &= 25\mathbb{E}\{(Z - 0.5)^2\}\sin^2\left(\frac{\pi n}{N}\right) + 0.2n(0.5 - 0.5)\sin\left(\frac{\pi n}{N}\right) + (0.02n)^2 - (0.02n)^2 \\ &= 25\sin^2\left(\frac{\pi n}{N}\right)(\mathbb{E}\{Z^2\} - \mathbb{E}\{Z\} + (0.5)^2) \\ &= 25\sin^2\left(\frac{\pi n}{N}\right)(1/3 - 0.5 + 0.25) \\ &= \frac{25}{12}\sin^2\left(\frac{\pi n}{N}\right) \end{aligned}$$

It is important to note that the square of a uniformly distributed random variable  $Z \sim \mathcal{U}(0, 1)$  is another random variable  $Y$  that has the PDF which is defined as  $\frac{1}{2\sqrt{Z}}$ . The expectation of  $Y$  is equal to  $\frac{1}{3}$ . As such the standard deviation of random process 1 is,

$$\sigma_{RP1} = \frac{5}{\sqrt{12}}\sin\left(\frac{\pi n}{N}\right) \quad (6)$$

From equations (5) and (6), it is clear that random process 1 is not stationary. Both its mean and its standard deviation are time variant. The mathematical description corroborates the earlier conjecture that random process 1 is not stationary.

ii. Random Process 2

Random Process 2 is defined by the equation (7), where  $M$ ,  $A$  and  $Z$  are random variables that are uniformly distributed.  $M$  and  $A$  are time-invariant but change with each realisation of the random process whereas  $Z$  varies both with time and with realisation.

---

<sup>1</sup>in Figure 9 all graphs have the same scale

$$x[n] = M(\mathcal{U}(0, 1) - 0.5) + A \quad (7)$$

To find the theoretical mean of the random process,

$$\begin{aligned} \mathbb{E}\{x[n]\} &= \mathbb{E}\{M(Z - 0.5) + A\} \\ &= \mathbb{E}\{M\}\mathbb{E}\{Z - 0.5\} + \mathbb{E}\{A\} \\ &= (0.5)(0.5 - 0.5) + (0.5) \\ &= 0.5 \end{aligned} \quad (8)$$

To find the theoretical variance of the random process,

$$\begin{aligned} Var(x[n]) &= \mathbb{E}\{(M(Z - 0.5) + A)^2\} - \mathbb{E}\{M(Z - 0.5) + A\}^2 \\ &= \mathbb{E}\{M^2(Z - 0.5)^2 + 2MA((Z - 0.5) + A^2)\} - (0.5)^2 \\ &= \mathbb{E}\{M^2\}\mathbb{E}\{(Z - 0.5)^2\} + 2\mathbb{E}\{M\}\mathbb{E}\{A\}\mathbb{E}\{Z - 0.5\} + \mathbb{E}\{A^2\} - 0.25 \\ &= \left(\frac{1}{3}\right)\left(\frac{1}{12}\right) + 2(0.5)(0.5)(0.5 - 0.5) + \left(\frac{1}{3}\right) - 0.25 \\ &= \frac{1}{9} \end{aligned}$$

As such, the standard deviation of random process 2 is,

$$\sigma_{RP2} = \frac{1}{3} \quad (9)$$

From equations (8) and (9), it is clear that random process 2 is stationary. However, equation (7) shows that the process is not ergodic. The sample mean of one realisation of the random process approach  $A_i$ , which is just one realisation of the random variable  $A$ . No matter how many samples we take of one realisation of random process 2, we will not be able to determine the random behavior inherent in  $A$  and thus calculate the mean of the random process.

### iii. Random Process 3

Random Process 3 is defined by the following equation, where  $Z$  is a random variable such that  $Z \sim \mathcal{U}(0, 1)$ ,

$$x[n] = 3(Z - 0.5) + 0.5 \quad (10)$$

To find the theoretical mean of the random process,

$$\begin{aligned} \mathbb{E}\{x[n]\} &= \mathbb{E}\{3(Z - 0.5) + 0.5\} \\ &= 3\mathbb{E}\{(Z - 0.5)\} + \mathbb{E}\{0.5\} \\ &= 3(0.5 - 0.5) + 0.5 \\ &= 0.5 \end{aligned} \quad (11)$$

To find the theoretical variance of the random process,

$$\begin{aligned} Var(x[n]) &= \mathbb{E}\{(3(Z - 0.5) + 0.5)^2\} - \mathbb{E}\{3(Z - 0.5) + 0.5\}^2 \\ &= 9\mathbb{E}\{(Z - 0.5)^2\} + 3\mathbb{E}\{(Z - 0.5)\} + \mathbb{E}\{(0.5)^2\} - (0.5)^2 \\ &= \frac{9}{12} + 3(0.5 - 0.5) + (0.5)^2 - (0.5)^2 \\ &= \frac{9}{12} \end{aligned}$$

As such, the standard deviation of random process 3 is,

$$\sigma_{RP3} = \frac{\sqrt{3}}{2} \quad (12)$$

From equations (11) and (12), it is clear that random process 3 is stationary. Equation (10) shows that random process 3 is ergodic and the mean and standard deviation can be calculated from one sufficiently long realisation of the random process.

### 1.3 Estimation of probability distributions

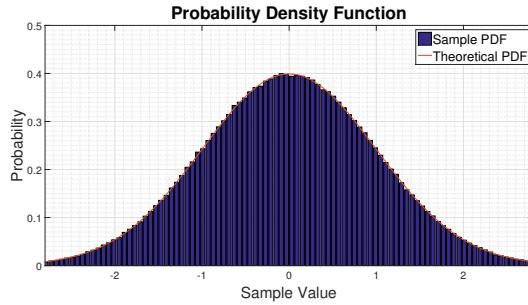


Figure 10: Estimated PDF of Gaussian random variable

1. Figure 10 shows that the file `pdf.m` works as expected. As a performance measure, a plot of the PDF of a zero-mean, unit standard deviation, Gaussian distribution is included.
2. The only random process that is both stationary and ergodic is random process 3. Random process 3 can be represented by a single random variable  $X$  that has a uniform distribution. More specifically  $X \sim \mathcal{U}(-1, 2)$ . Figure 11 shows the estimated PDF approximated using `pdf.m`. Increasing the sample size significantly improves the estimate. As a performance measure, the theoretical PDF has been included.

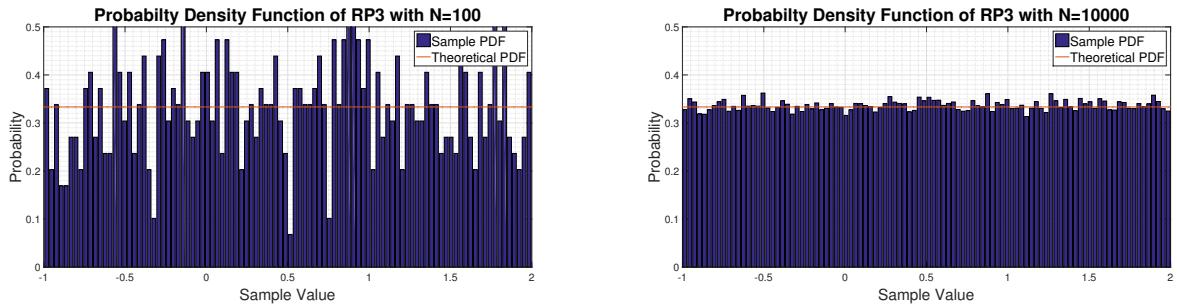


Figure 11: Estimated PDF of random process 3

## 2 Linear Stochastic Modelling

### 2.1 ACF of Uncorrelated Sequences

1. The unbiased estimate of the ACF is graphed in figure 12. All real signals have symmetric ACF, i.e, the ACF is an even function if the input signal is real. If the signal is complex valued, the ACF will be a Hermitian function; a Hermitian Function is defined as one in which  $R(\tau) = R^*(-\tau)$ . The ideal ACF of white noise should consist of a dirac delta function at  $\tau = 0$  and should be 0 elsewhere. This would indicate that the samples are completely uncorrelated and thus, random. The peak at  $\tau = 0$  is present in the unbiased estimate, however the estimate is not 0 elsewhere.

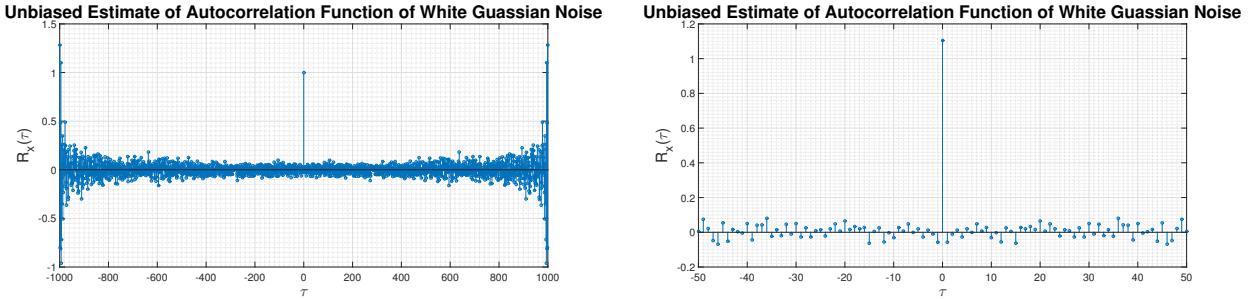


Figure 12: Unbiased estimate of ACF of uncorrelated sequences

2. Observing the zoomed plot, it is clear that the estimate of the ACF is very close to the ideal ACF described above. There is a large peak at  $\tau = 0$  and for other values of  $\tau$ , the estimate is approximately 0. This is however not the case for large values of  $\tau$  where the estimate is extremely large, often larger in magnitude than the peak at  $\tau = 0$ .

$$R_{X, \text{unbiased}}(\tau) = \frac{1}{(N - |\tau|)} \sum_{n=0}^{N-|\tau|-1} x[n]x[n + \tau] \quad (13)$$

$$\text{Var}(R_{X, \text{unbiased}}(\tau)) = \frac{1}{(N - |\tau|)^2} \text{Var}\left(\sum_{n=0}^{N-|\tau|-1} x[n]x[n + \tau]\right) \quad (14)$$

3. The unbiased estimate of the ACF has the form defined in equation (13) and the estimator's variance is defined in equation (14). Notice that the estimator has a scaling factor of  $\frac{1}{N-|\tau|}$  to ensures that the estimator is unbiased; the mean of the estimator is independent of  $\tau$ . Although the estimator is unbiased, the variance of the estimator increases with  $\tau$ . As such, the shape of the ACF for large  $\tau$  is unreliable.

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

$$\text{Var}(\hat{R}_{X, \text{biased}}(\tau)) = \frac{1}{N^2} \text{Var}\left(\sum_{n=0}^{N-|\tau|-1} x[n]x[n + \tau]\right) \quad (16)$$

The form of the biased estimator is presented in equation (15). Observing the form of the biased estimator, it is clear that the values calculated at large  $\tau$  are given lower weightings. The ACF for large  $\tau$  is calculated with a smaller number of samples; the limits of the summation decreases as  $\tau$  increases. As such, the result of the summation is bound to be smaller. Scaling all the values by the same amount, which is equal to  $\frac{1}{N}$ , is equivalent to giving lower weighting to values which correspond to large delays. In theory, it makes sense to give a lower weighting to these values as they were calculated from fewer data points and thus are less reliable. In practice, giving a lower weighting to values corresponding to large  $\tau$  makes the estimator biased.

In conclusion, the unbiased estimator has its variance dependent on  $\tau$  whereas the biased estimator has its mean dependent on  $\tau$ ; the type of estimator to be used should be carefully considered. If an unbiased estimator is used, it is wise to discard values of the ACF for  $\tau$  larger than  $\frac{N}{10}$ .

## 2.2 ACF of Correleated Sequences

- The unbiased estimate of the ACF of filtered white noise is graphed in figure 13. The ACF has a triangular shape between  $-8 < \tau < 8$ . Equation (17) shows the form of the ACF of a filter signal, where  $h(\tau)$  is the impulse response of the filter; this equation is based on the Wiener–Khinchin theorem.

$$R_Y(\tau) = h(\tau) * h^*(-\tau) * R_X(\tau) \quad (17)$$

The impulse response of the moving average (MA) filter is rectangular. When convolved with itself, a triangular function is obtained. Although the ideal ACF of the filter signal should be 0 outside the range  $-8 < \tau < 8$ , non-zero values are observed. This is due to the fact that the ACF graphed in figure 13 is only an estimate.

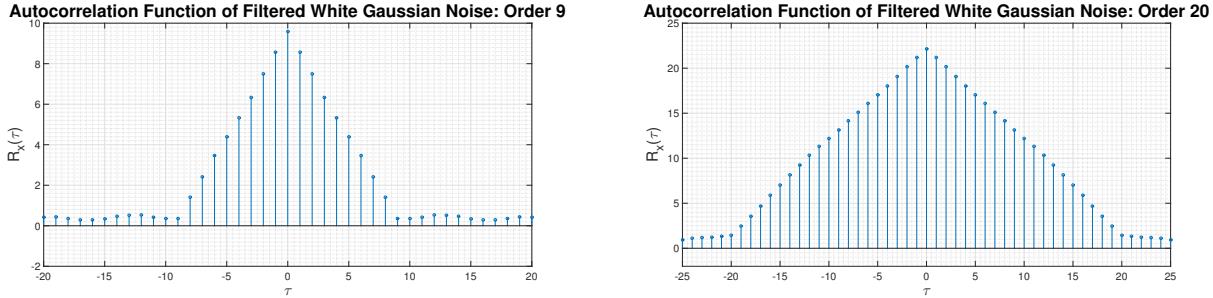


Figure 13: Unbiased estimate of ACF of correlated sequences

Increasing the order of the filter increased the width of the triangle; this is expected. The magnitude of the ACF obtained is also scaled. Ideally, ACF should be measured within the normalised range  $[-1, 1]$ . The (local) sample mean can be calculated using the equation  $\hat{m} = y[n] = \sum_{i=0}^{N-1} b_i x[n-i]$ , where  $N$  is the order of the MA filter. To achieve a good estimate, the sample size should be at least as large as the order of the filter. In addition, all the coefficients of the MA filter should be equal to  $\frac{1}{N}$ , so that  $\hat{m} = y[n] = \frac{1}{N} \sum_{i=0}^{N-1} x[n-i]$

- If  $X_n$  is an uncorrelated, wide-sense stationary process, then its ACF will be  $R_X(\tau) = \alpha\delta(\tau)$  where  $\alpha$  is a constant.  $R_Y(\tau)$  will then represent the ACF of the filter scaled by the constant  $\alpha$ . From the ACF of the filter, both the impulse response and the Power Spectral Density (PSD) can be determined.

## 2.3 Cross-Correlation Function

- The cross-correlation of  $X$  and  $Y$  is graphed in figure 14. It is clear that  $X$  is correlated to the current value and past 8 values of  $Y$ . This is expected as the MA filter has order 9. Again, the non-zero values for other values of  $\tau$  are due to the fact that the cross-correlation that has been graphed is an estimate and thus not ideal.

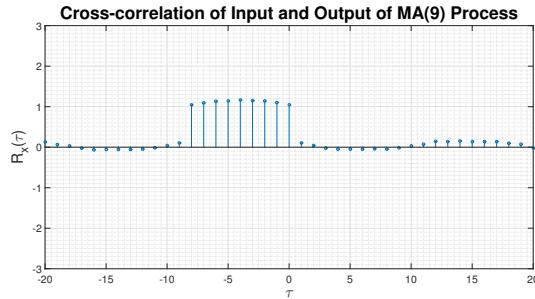


Figure 14: Cross-correlation of  $X$  and  $Y$  for 1000 samples

If  $X_t$  is an uncorrelated, wide-sense stationary process, the cross-correlation function can be represented as  $R_{XY}(\tau) = h_Y(\tau) * \alpha\delta(\tau) = \alpha h_Y(\tau)$ . As such, the cross-correlation function is equivalent to the impulse response of the filter. Similar to the assertion in section 2.2, the ACF and the PSD of the filter can be found from the impulse response.

- System identification be conducted by identifying any characteristic property of the system. For example, the system can be identified based on its impulse response, ACF or PSD. Driving the filter with Additive White

Gaussian Noise (AWGN) and observing the cross-correlation function, the order of the filter can be obtained. The relative magnitudes of the coefficients can also be determined.

## 2.4 Autoregressive Modelling

1. The results obtained from testing for convergence are plotted figure 15. The suggested sample size of 100 is not enough to clearly observe the very regular pattern of convergence and thus the exercises was repeated with 10000 samples.

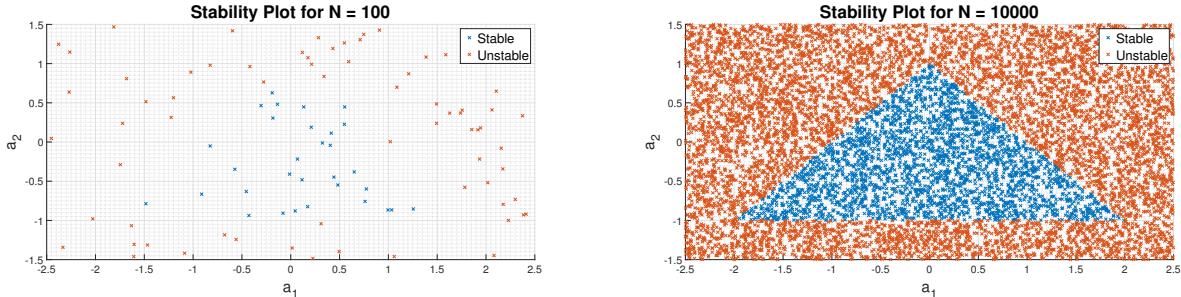


Figure 15: Stability plots for AR(2) process

The general form of an AR(2) process has been defined in the coursework handout. For the process to be stable, the three inequalities listed below have to be met.

$$\begin{aligned} a_1 + a_2 &< 1 \\ a_1 - a_2 &< 1 \\ |a_2| &< 1 \end{aligned}$$

The three inequalities define a triangular region on the  $a_1a_2$  plane. Within the defined region, the response of all AR(2) process will be bounded; they will be wide-sense stationary. Outside this region, the response of the processes will diverge; they will not be wide-sense stationary as the mean of the process is changing.

The inequalities listed above are obtained by noting that the process will only be stable if the poles of the system lie within the unit circle. To determine the position of the poles, the characteristic equation  $C(z) = z^2 - a_1z - a_2$  is studied. The roots of the equation can be found using the equation  $z = \frac{a_1 + \sqrt{a_1^2 + 4a_2}}{2}$ . If  $a_1 + 4a_2 \geq 0$ , the roots will be real. Otherwise the roots will be complex conjugates.

If the roots are real, the larger root has to be less than 1,

$$\begin{aligned} \frac{a_1 + \sqrt{a_1^2 + 4a_2}}{2} &< 1 \\ a_1^2 + 4a_2 &< a_1^2 - 4a_1 + 4 \\ a_1 + a_2 &< 1 \end{aligned} \tag{18}$$

and the smaller root has to be greater than -1,

$$\begin{aligned} \frac{a_1 - \sqrt{a_1^2 + 4a_2}}{2} &> -1 \\ a_1^2 - \sqrt{a_1^2 + 4a_2} &< -2 \\ a_2 - a_1 &< 1 \end{aligned} \tag{19}$$

Lastly, if the process has 2 real roots then the characteristic equation can be expressed as,  $C(z) = (z - \lambda_1)(z - \lambda_2) = 0$ . Expanding, we get  $C(z) = z^2 - (\lambda_1 + \lambda_2)z + \lambda_1\lambda_2 = 0$ . For the random process to be stationary,  $|\lambda_i| < 1$  for  $i = 1, 2$ . Thus,  $a_1 = (\lambda_1 + \lambda_2)$  and  $a_2 = \lambda_1\lambda_2$ . Since  $|\lambda_i| < 1$ ,  $|a_2| < 1$ .

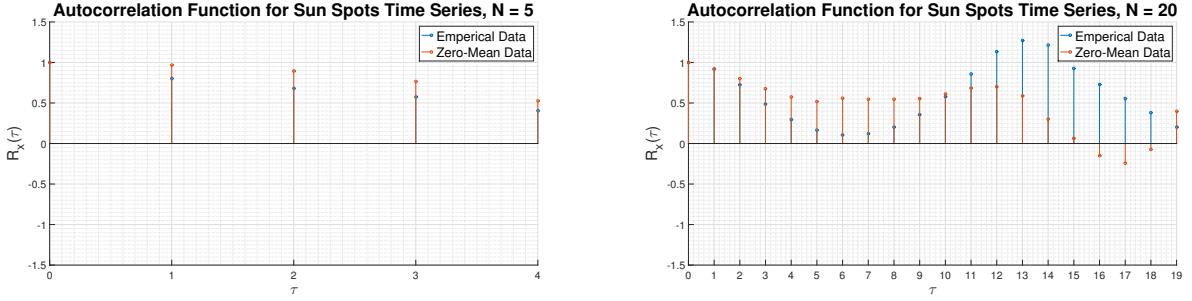


Figure 16: ACF of sunspot time series for data lengths N=5 and N=20

2. The ACF of the sunspot time series is graphed for data lengths of 5 and 20 in figure 16. With data length N=5, trends cannot be clearly observed; a longer data series is required. When data length N=20, a sinusoidal trend starts to appear.

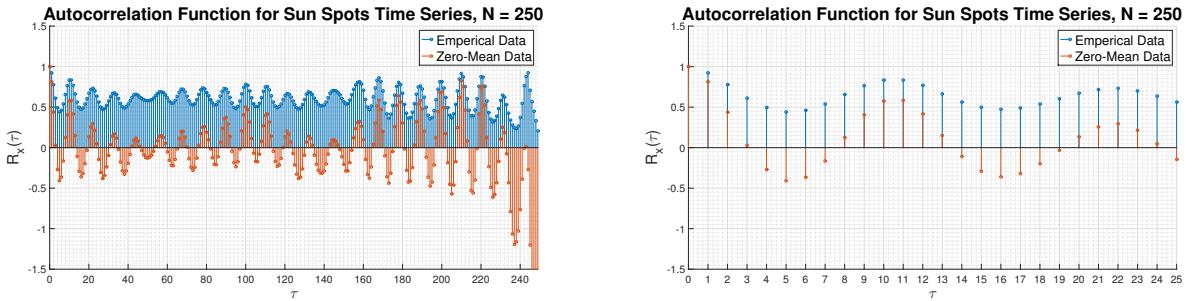


Figure 17: ACF of sunspot time series for data length N=250

With data length N=250, the best sinusoidal trend is observed. The ACF is not a perfect sinusoid and irregularities occur at  $\tau=50$  and  $\tau=100$ . These irregularities are most likely due to the fact that the data length is N=250. An empirical bound was placed on the ACF function in section 2.1 beyond which the estimate of the ACF becomes unreliable. The empirical bound stated above was  $N/10$ ; as such, the ACF function should only be studied for  $\tau < 25$ . Figure 17 shows a zoom of the ACF of the zero-mean sunspot series for N=250. In this region, the ACF resembles the shape of the ACF of an AR(2) system with  $a_1 > 0$  and  $a_2 < 0$ . Lastly, the bias inherent in the empirical data hides trends that can be easily observed with the zero-mean data.

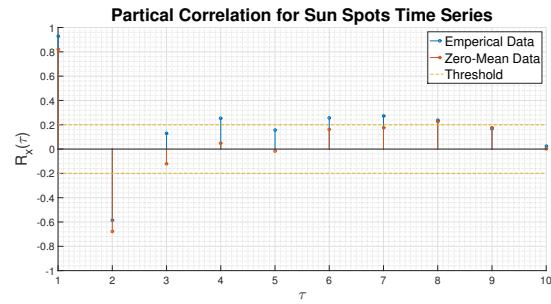


Figure 18: PACF for sunspot time series

3. The partial autocorrelation function (PACF) of the sunspot time series is graphed in figure 18. It is important to note that the standardised plot yields a result that is slightly different from the empirical data. Using zero-mean unit-variance data removes all biases and thus produces a more accurate answer. Ideally, the PACF should be 0 for  $\tau > p$ , where  $p$  is the model order; however for real data, a small threshold has to be introduced. As seen in the graph, the threshold has been set to 0.2. It is evident that the PACF function decreases significantly after order 2. This suggests that the current sample at any point in time is not strongly correlated to the sample three time instances ago. The PACF correlation function corroborates the findings above; it is most likely that sunspot time series is an AR(2) process.

4. The Minimum Description Length (MDL) and the Akaike Information Criterion (AIC) introduce a "penalty" for overestimating the order number. Both criteria were first introduced in the field of information theory; the

main motivation behind the criteria is to balance the trade-off between the error incurred and the computational complexity involved in utilising a certain model. In practice, increasing the order of the model almost always reduces the error and thus a penalty is needed to limit the order number; without this penalty, a higher model order will usually be preferred.

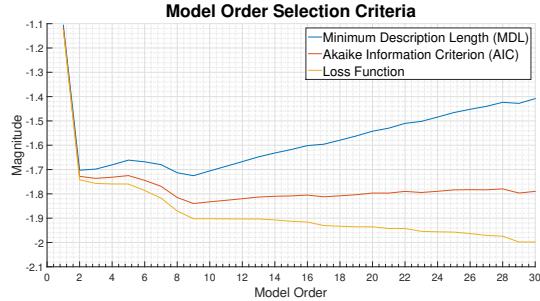


Figure 19: Model selection criteria for sunspot time series: MLD, AIC, Loss Function

Figure 19 shows the MDL, AIC and the loss function for the sunspot time series. Note that the loss function is calculated by taking the log of the estimated variance as calculated by the `aryule` function. The estimated variance is a scaled version of the cumulative squared error, where the scaling factor is the length of the time series. The loss function confirms the assertion made above; increasing the order of the estimated model decreases the error. However, an optimal model is one that minimises the MDL and AIC function. Based on that criteria, it is clear that the sunspot time series should be modelled as either an AR(2) or an AR(9) process.

It should be noted that over-modelling can lead to spectral line splitting. While this may reduce the error incurred, the model will not be representative of real data. As such, the sunspot time series should be modelled as an AR(2) process.

5. Using the AR models calculated above, future values of the sunspot time series can be predicted. The MATLAB function `predict` is used and the results obtained for different combinations of model orders and prediction horizons have been graphed in figure 20.

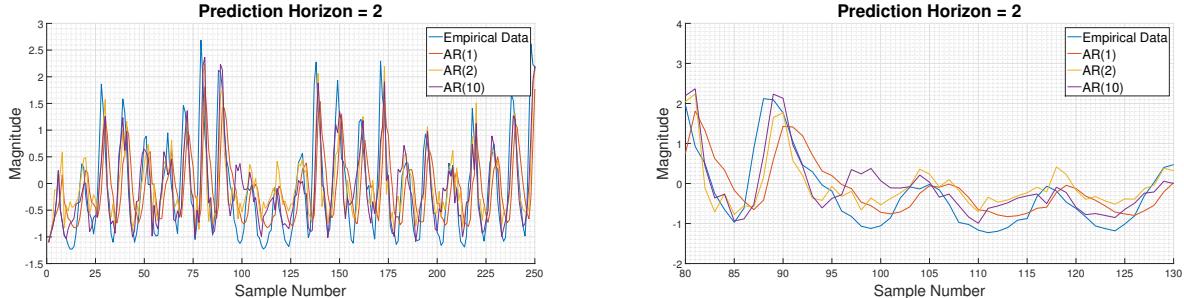


Figure 20: Sunspot time series prediction: horizon = 2

From figure 20, the first thing to notice is that using the AR(1) model, the prediction seems to be lagging the real data. The AR(1) model provides too few degrees of freedom to correctly represent the data. The current sample seems to depend on more than just the previous sample. Observe the prediction made using the AR(2) model; the lag has been reduced. Both the AR(1) and AR(2) model seem to be reacting to changes rather than exploiting the periodic nature of the sunspot series. The AR(10) model seems to predict peaks in the time series before they occur. This is evident near sample number 95. There is actually a dip in the number of sunspots however the AR(10) model predicts a peak. The AR(1) and AR(2) models predicted the trough well.

From figure 21, it is clear that using a small model order yields bad predictions for large horizons; using AR(1) and AR(2) and predicting 10 years into the future, little information about the number of sunspots can be obtained. Although the peaks in the predicted data correspond strongly to the peaks in the empirical data, the amplitude of the prediction is significantly lower. The decrease in amplitude of the prediction as the prediction horizon increases is a trend that can be observed across all three models.

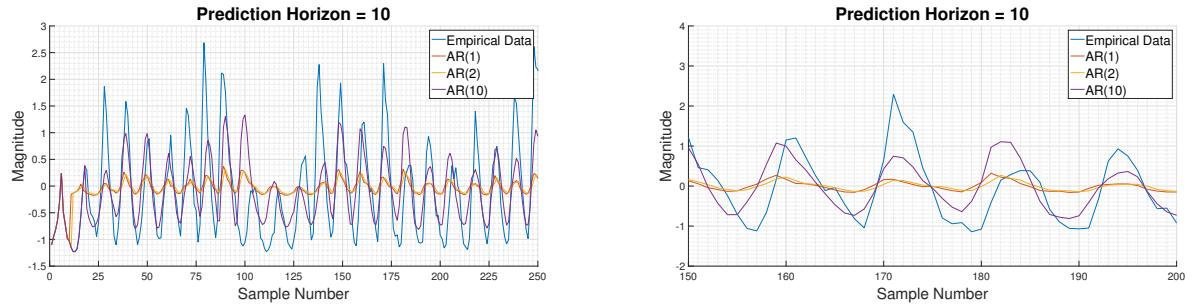


Figure 21: Sunspot time series prediction: horizon = 10

In both figure 20 and 21, the expected value of the process has been graphed. When the expectation is taken, the input AWGN goes to zero. As such, it is equivalent to studying a process in which the input has been suddenly removed. Analysing such systems is a very involved process; careful analysis of the AR coefficients is required [1]. For the sunspot time series, it is clear that if the input is cut, the system will gradually approach zero. Using an AR(10) model to predict 10 years into the future, the decrease in amplitude is not that significant. As such, it is intuitive to conclude that a large model order allows better prediction for long horizons.

Figure 22 shows the results obtained when predictions are made for horizons 1 and 5. The observations corroborate the findings listed above. For a prediction horizon of 5, the AR(1) and AR(2) model provide invalid data whereas the AR(10) model provides somewhat reliable data. Again, it can be concluded that using a small order model to predict for large horizons produces inaccurate results. For a prediction horizon of 1, all models produce similar results.

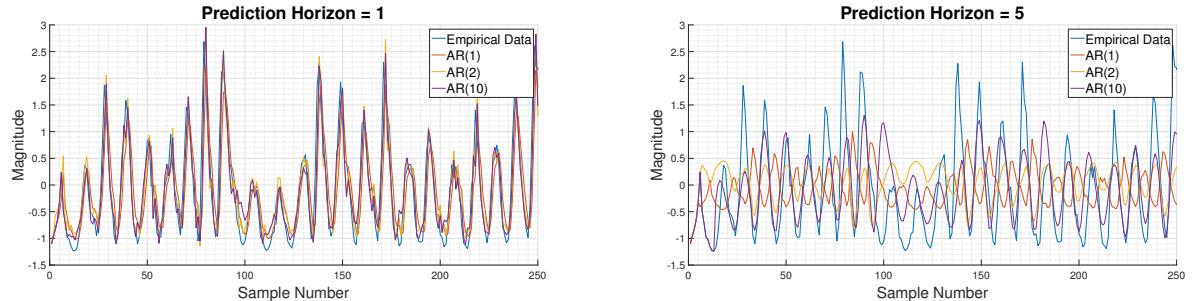


Figure 22: Sunspot time series prediction: horizon = 1 and horizon = 5

## 2.5 Real World Signals: ECG from iAmp Experiment

a) The probability density estimates of the original and averaged heart rate, with  $\alpha=1.0$  is graphed in figure 23. The pdf function was utilised to obtain the estimates.

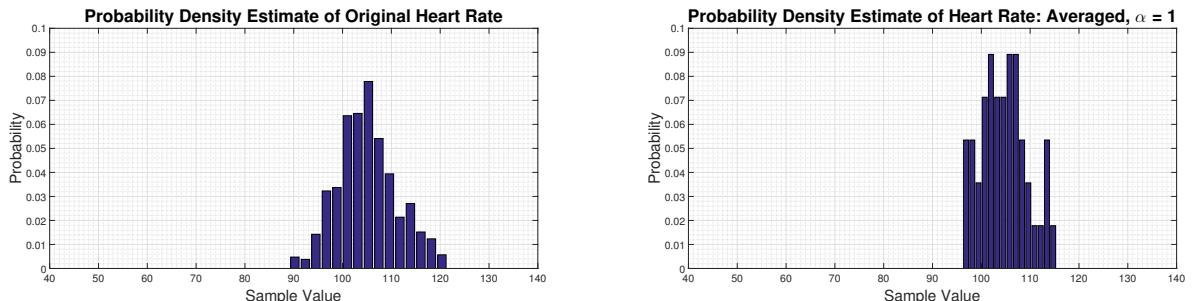


Figure 23: Probability density for original heart rate and averaged heart rate, with  $\alpha=1.0$

The probability density estimates of the averaged heart rate, with  $\alpha=0.6$  is graphed in figure 24.

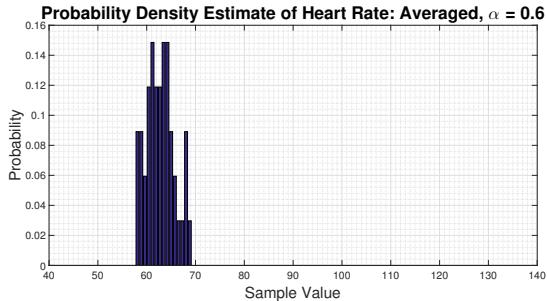


Figure 24: Probability density for averaged heart rate, with  $\alpha=0.6$

b) Firstly, it is important to note that the heart rate for each trial has been modelled as a stationary process. As discussed in section 1.3, obtaining the pdf of a non-stationary process using the pdf.m is not possible. As such, each sample of the heart rate is 1 realisation of a random variable  $H$ .

The averaging process, as described in equation (15) of the Advanced Signal Processing coursework, will have the effect of reducing the variance of the probability density estimate. The averaged data are no longer unique realisations of the random variable  $H$ ; they are unique realisations of the random variable  $\hat{H}$ . If the original heart rate,  $H$  has a variance of  $\sigma_H^2$ , averaging every  $L$  samples will reduce the variance of the new heart rate,  $\hat{H}$ , by a factor of  $L$ , to  $\frac{\sigma_H^2}{L}$ . This is clearly observed in figure 23. The original estimate is a lot wider than the averaged estimate.

Multiplying by a constant  $\alpha$  has two effects. Firstly, it shifts the mean of the estimator. Shifting the mean of the estimator might introduce a bias into the estimate. Alternatively, if the way in which the data was obtained is biased, multiplication by  $\alpha$  might be deliberately introduced to remove the bias and obtain an unbiased estimate; in the case of the unbiased estimate of the ACF, a scaling factor of  $\frac{1}{N-|\tau|}$  was introduced to remove the inherent bias in the estimate.

Multiplication by a constant will also have the effect of changing the variance of the estimate. When the variance of a random signal is to be computed, constant terms are taken out of the expectation operator and are squared. Multiplying by 0.6 will have the effect of reducing the variance to 0.36 of its original value. The estimate in figure 24 has a variance that is smaller than the averaged estimate in figure 23. This is exactly as expected.

c) Figures 25 and 26 show the ACF for the three RRI time series. The MATLAB function `detrend` has been used to remove linear biases from the data.

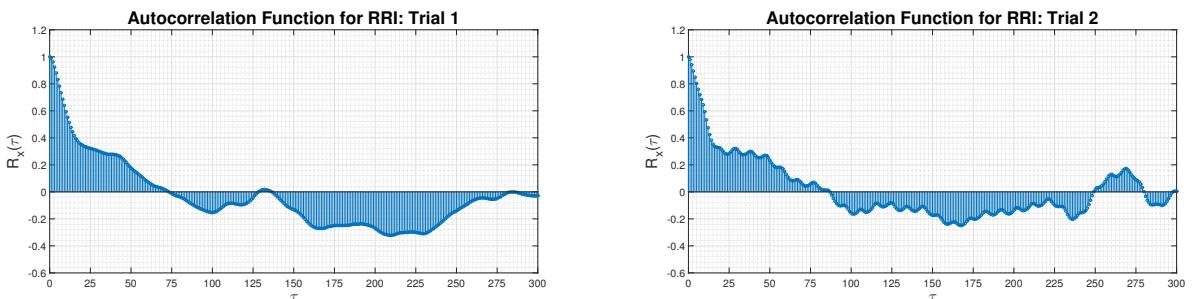


Figure 25: ACF for RRI time series: trial 1 and trial 2

It is evident that physical process used compute the RRI is autoregressive. The ACF of an MA process is finite in length and goes to 0 after a certain value of  $\tau$ . The amount of delay after which the ACF goes to 0 represents the order of the MA process; the ACF of MA process has a similar shape to the graph presented in figure 14. The ACF of an AR process is infinite in length. It has a sinusoidal shape. As such, it is clear that the RRI data should be modelled as an AR model rather than a MA process.

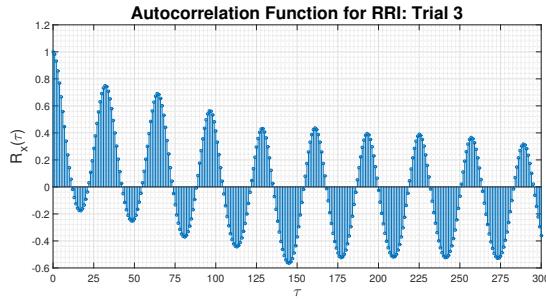


Figure 26: ACF for RRI time series: trial 3

4. The three model selection criteria, namely the MDL, AIC and loss function are graphed for each trial. In addition, the partial correlation function for each trial is graphed as well. The threshold introduced to account for the non-idealities has been reduced from 0.2 to 0.1.

For trial 1, the model order selection process is rather simple. The data should be modelled as an AR(4) model. Both the MDL and AIC functions have a minimum at 4 and the partial correlation function does not exceed the threshold of 0.1 after 4.

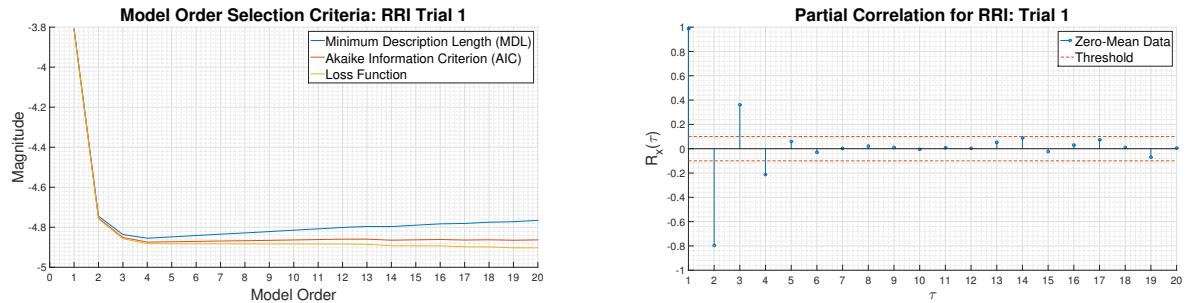


Figure 27: Model selection criteria and partial correlation function for RRI Trial 1

For trial 2, the model selection process is involved. There are multiple local minima for both the MDL and AIC function. The global minimum of the MDL function occurs at order 16, whereas the global minimum for the AIC function occurs at order 17. For order 17, the partial correlation is below the threshold of 0.1. As such, trial 2 should be modelled as an AR(16) model.

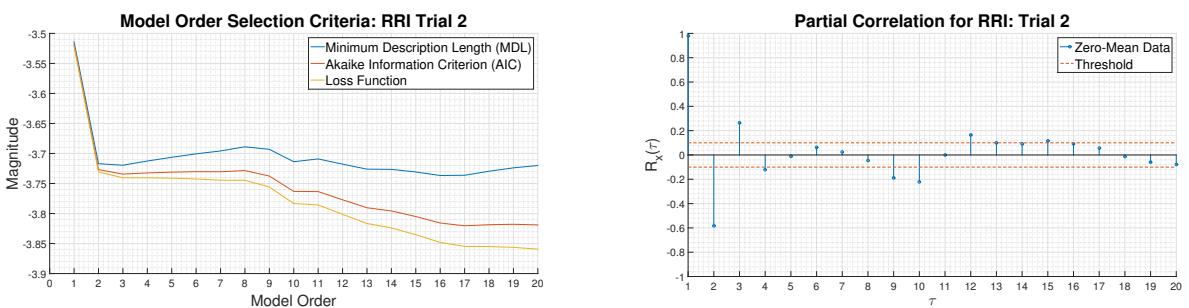


Figure 28: Model selection criteria and partial correlation function for RRI Trial 2

Similar to trial 1, the model order selection process for trial 3 is straight forward. The data should be modelled as an AR(2) model. Both the MDL and AIC functions have a minimum at 2 and the partial correlation function does not exceed the threshold of 0.1 after 4.

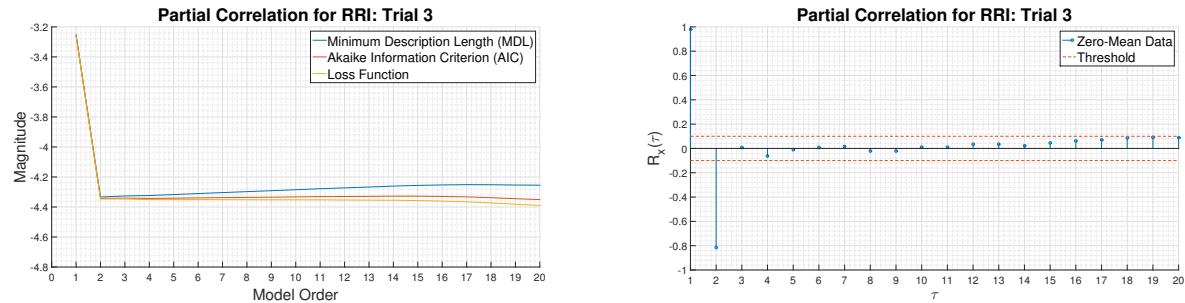


Figure 29: Model selection criteria and partial correlation function for RRI Trial 3

### 3 Spectral Estimation and Modelling

1. Since WGN is a real signal, the PSD of the signal is symmetric about  $\pi$ . As such, only half of the spectrum is graphed.

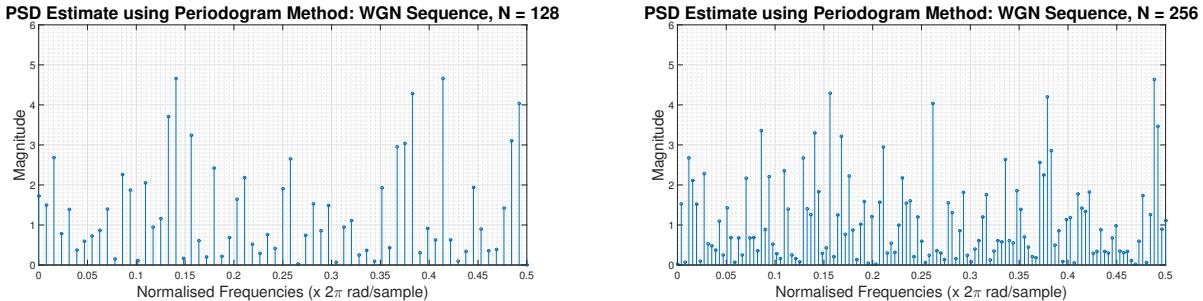


Figure 30: Periodogram of WGN:  $N = 128$  and  $N = 256$

The ideal PSD of WGN is unity at all frequencies; the plots in figure 30 do not resemble the ideal PSD. A longer sequence consisting of 512 samples is graphed in figure 31.

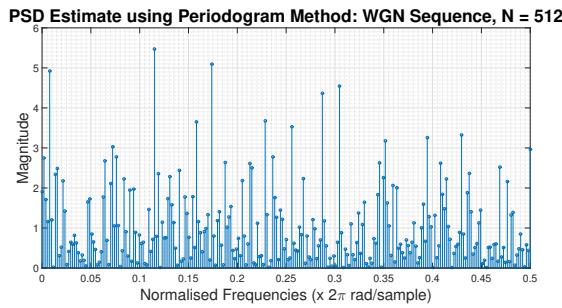


Figure 31: Periodogram of WGN for data length of 512

The PSD is evaluated at more unique frequencies since the sequence is longer however the shape is still not ideal. Notice that increasing the number of sample does not cause the estimated PSD to approach the ideal PSD of WGN.

Equation (17) from the Advanced Signal Processing coursework can be rearranged to show that the periodogram is actually the Fourier transform of the ACF. The ACF can be calculated through 2 estimators, one that is asymptotically biased while one is asymptotically unbiased. Using the biased estimator, the estimate of PSD may be negative at certain frequencies. To circumvent this, the unbiased estimate of ACF is used. However, as discussed in section 2.1, the unbiased estimator has large variances at large values of  $\tau$ . Since every single point in the ACF is used to calculate the PSD estimate, the PSD estimate will also have a large variance. The estimator is inconsistent; the variance of the estimator does not decrease when the number of samples is increased. The bias inherent in the periodogram based estimate of the PSD depends on the window that is applied to obtain a finite length sequence of an infinitely long random process. If a rectangular window is used, the periodogram is asymptotically unbiased.

As such, the periodogram based estimates presented in figures 30 and 31 are asymptotically unbiased but inconsistent; as the number of samples increases, the mean converges to its true value of 1 however the variance does not converge to 0.

### 3.1 Averaged Periodogram Estimate

1. Filtering the WGN with a MA filter certainly improves the PSD estimate. The filter smooths out the PSD, removing the large peaks. Mathematically, smoothening the PSD is equivalent to reducing the variance of the estimate. The variance is scaled by a factor which is approximately equal to  $\frac{1}{2M+1}$ , where  $M$  is the order of the MA filter. The decrease in variance comes at a cost; filtering causes the PSD estimate to become biased. Smoothening the PSD also causes the frequency resolution to decrease. By filtering with an MA filter, frequency resolution and estimator bias is traded for reduced variance.

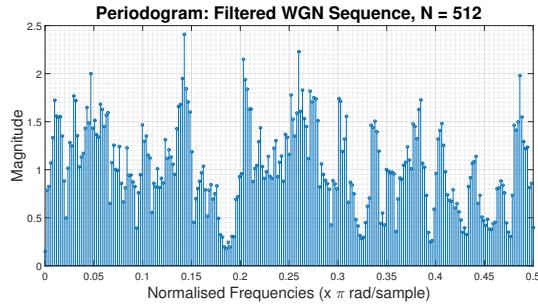


Figure 32: Periodogram of filtered WGN

2. The 8 periodograms obtained were completely random. The peaks occurred at different frequencies in each estimate. The lack of a distinct pattern in the position of the spectral peaks is attributed to the fact that the ideal spectral is flat and the nonidealities are due to the very large variance rather than a bias at certain frequencies.

3. The 8 segments are averaged and the resulting PSD estimate called the averaged periodogram is graphed in figure 33. An additional graph showing the effects of averaging 64 segments is also shown. Averaging the periodogram certainly reduces the variance in the PSD estimate. The larger the number of segments, the greater the decrease in variance; the decrease in variance is actually  $1/K$  where  $K$  is the number of segments.

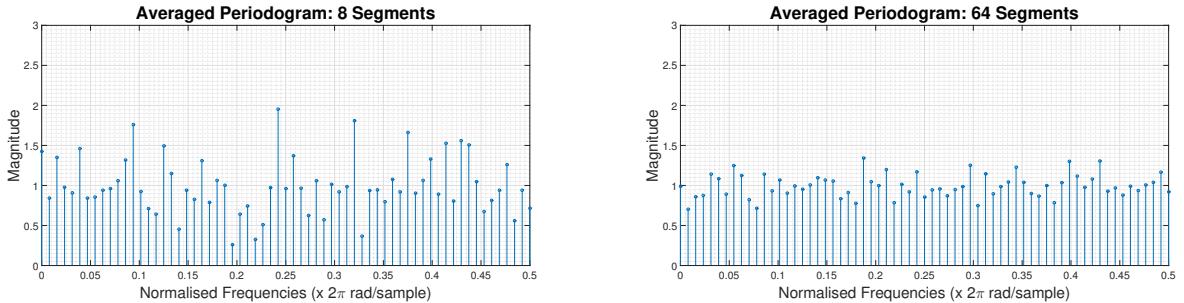


Figure 33: Averaged periodogram: 8 and 64 segments

If the length of the sequence is limited, the number of samples in each segment can be reduced; this will lower the frequency resolution of the periodogram. Alternatively, to maintain the same level of frequency resolution, overlapping segments can be used. Overlapping segments may introduce bias into the estimate. As such, obtaining an averaged periodogram is a trade-off between frequency resolution, variance of the estimate and bias of the estimate.

### 3.2 Spectrum of Autoregressive Process

- Figure 34 shows the exact and estimated PSD of the filter defined in the Advanced Signal Processing coursework. The normalised cut-off frequency of the filter is approximately  $f = 0.3799\text{Hz}$ . If the PSD estimate is considered as a digital filter itself, it will be a 2<sup>nd</sup> order filter with a cut-off frequency of  $f = 0.3555\text{Hz}$

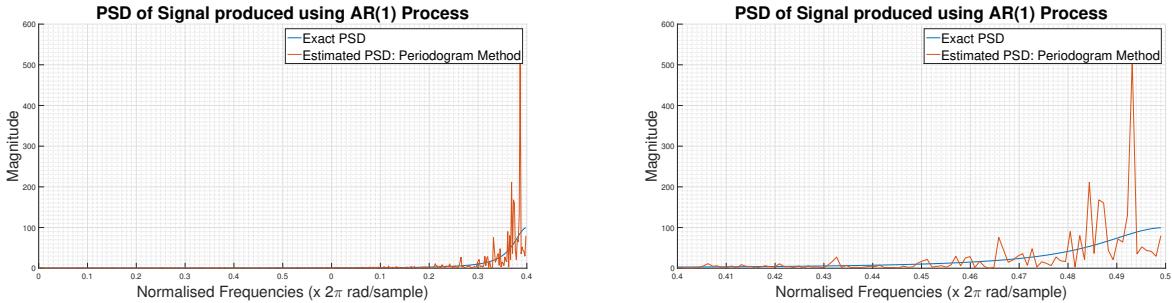


Figure 34: Periodogram of AR(1) process

- The signal used to generate the periodogram is of finite length. Mathematically, a finite length signal is obtained by multiplying the infinitely long signal with a rectangular window. Multiplying by a rectangular window in the time domain is equivalent to convolving the PSD of the signal with a sinc function. The effects of the convolution are not as pronounced at low frequencies because the PSD is almost zero in that frequency range. At high frequencies, the PSD is non-zero and thus the effects of the convolution are obvious.

Often the rectangular window is believed to introduce high frequency components into the signal's power spectrum. This is fallacious. Considering the periodogram of the low-pass filter graphed in figure 35. It is clear that the effects of the rectangular window will be pronounced in whichever frequency range the PSD is non-zero.

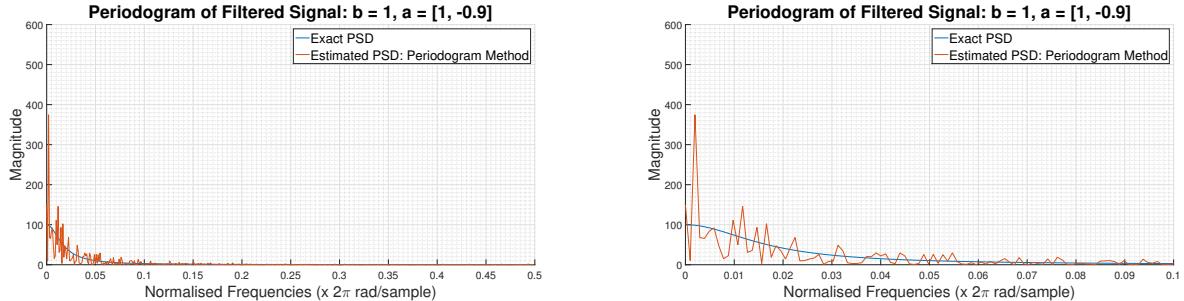


Figure 35: Periodogram of low pass filter

- Periodogram based methods of estimating the PSD do not require a priori knowledge about the way the random samples are generated; these methods do not incorporate knowledge about the type of data. On the other hand, model based estimates take this information into account when generating the estimate. Since the model based estimates are made under the assumptions that the time series follows certain model, if the assumption is correct, the model based estimate should be more accurate than the periodogram based estimate. Figure 36 graphs the model-based estimate of the AR(1) process.

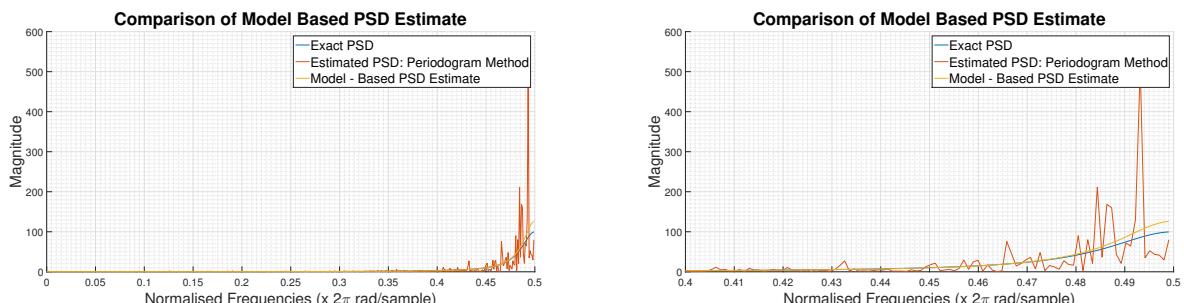


Figure 36: Comparison of model based estimate of PSD to periodogram based estimate of PSD

Equations (20) and (21) from the Advanced Signal Processing coursework are used to estimate the model parameters. Using the equations,  $\hat{a}_1 = 0.9128$  and  $\hat{\sigma}^2 = 0.9637$ ; the estimates are very close to the ideal values of  $a_1 = 0.9$  and  $\sigma^2 = 1$ . It is clear from figure 36 that the model based estimate is much better than that obtained using the periodogram. This is exactly as expected; the model based estimate of PSD should certainly be closer to the ideal PSD.

Most notably, the effects of having a finite length sequence rather than an infinitely long sequence manifest by introducing errors in the parameters directly rather than in the PSD estimate; the model-based estimate does not suffer the large errors the periodogram based estimate suffer due to the convolution with the sinc function.

5. Figure 37 shows the PSD estimates of the sunspot time series using both the periodogram and the model-based estimation methods.

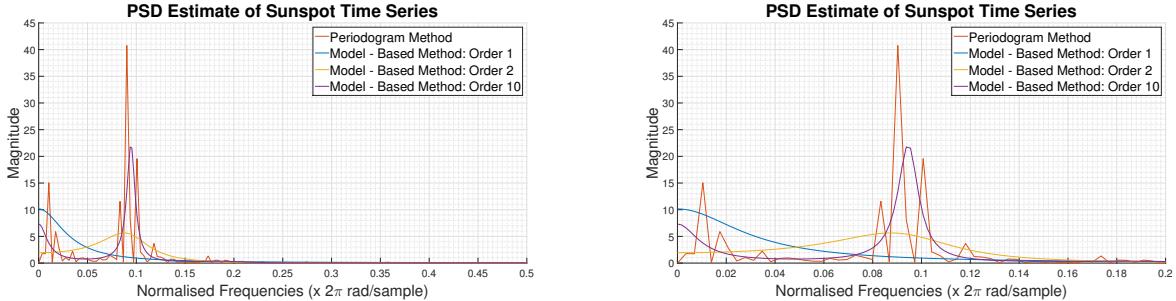


Figure 37: PSD estimates for sunspot time series

In section 2.4, it was concluded that the sunspot time series should be modelled as an AR(2) process. Estimating the PSD using an AR(1) model results in a completely invalid result; the model does not have enough degrees of freedom to represent the periodic nature of the time series. Using an AR(10) model to estimate the PSD, a very large peak is observed. It is clear that the AR(10) model based estimate is closer to the periodogram than the AR(2) model based estimate. However this does not mean that the AR(10) model best represents the data; the periodogram is not the ideal estimate and it suffers multiple shortcomings that have already been discussed. If the model based estimate in figure 36 followed the periodogram exactly, it would be a horrible estimate.

Moreover, the AR(10) model compresses the power of the time series into a very small frequency range. This suggests that the sunspot series is extremely regular and highly periodic. Looking at 20, the effects of modelling the sunspot time series as an AR(10) process are clear. The AR(10) model often pre-empts the existence of peaks. The 2<sup>nd</sup> order model provides a good balance. The 1<sup>st</sup> order model severely under models the time series and the 10<sup>th</sup> order model over models it.

### 3.3 Spectrogram for Time-Frequency Analysis: Dial Tone Pad

- The signal  $y$  is graphed in figure 38. The axis has been adjusted to show the signals generated when the numbers 0 and 2 are pressed.

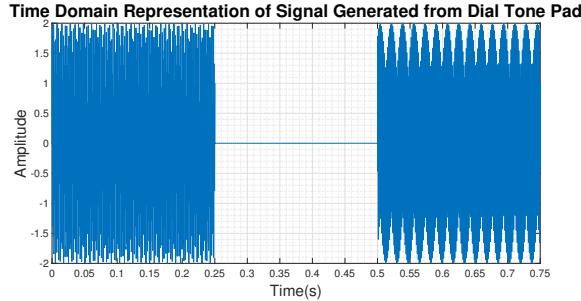


Figure 38: Signal generated when 0 and 2 are pressed

Note that the sampling frequency of 32768 will allow all frequency components up to the  $f = 16384\text{Hz}$  to be clearly distinguished without aliasing. This will allow the 10<sup>th</sup> octave to be captured. It is fallacious to think that sampling at such a high rate will increase the frequency resolution. Since we are dividing 1 second of data into 4 non-overlapping segments, the Fast Fourier Transform (FFT) will only be evaluated at frequencies that are multiples of 4.

- Figure 39 shows the spectrogram of the signal produced when a random phone number is dialed. The graph is generated using the MATLAB function `spectrogram`; the syntax is `spectrogram(y, hann(8192), 0, 8192, 32768)`. The signal is split into segments of length 8192 with no overlap. Each segment is windowed using the Hanning window. For testing, a seed is used to generate the same random sequence. The phone number that will be used is 0200891960025.

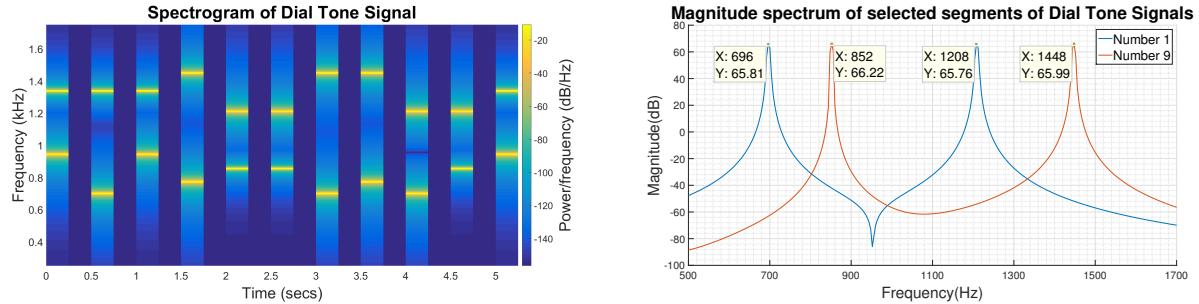


Figure 39: Spectrogram and magnitude spectrum of dial tone signal

From the spectrogram, the distinct spectral components of each number are extremely clear. The spectrogram provides the aerial view of the spectrum. The magnitude spectrum provides the side view of the spectrum; the magnitude spectrum for number 1 and number 9 have been graphed. Firstly, the effects of the Hanning window are clear. The ideal spectrum should contain two peaks at two distinct frequencies. The spreading of the spectrum, referred to as spectral leakage, is attributed to the windowing effect. Note that the slight discrepancies in the position of the peaks in the magnitude spectrum are due to the fact that the frequency resolution of the spectrum is poor. As discussed above, the sampling frequency is 32768 however since each segment is 8192 samples long, the FFT is only evaluated at 8192 frequencies in the range [0,32768]. As such, the peak that should occur at 697 and 1477 occur at 696 and 1448 respectively.

- Each segment of the FFT can be analysed and the two peaks can be located; a threshold should be set, below which the FFT is assumed to not contain any spectral peaks. The spectral peaks can be matched with the frequencies for each number using the table provided in the Advanced Signal Processing coursework.
- Figure 40 shows the time domain signal that has been corrupted by AGWN with  $\sigma = 2$ . It is interesting to note that the time domain signal is almost nothing like the signal presented in figure 38. The middle region of the graph, the time in which no number is pressed, is not identifiable. Also note that the signal in figure 38 had an amplitude of 2, whereas the signal in figure 40 has a maximum amplitude of approximately 6. Even though the time domain signal looks nothing like the ideal signal, the spectrogram is still extremely clear. Frequency domain analysis can still be used for key classification.

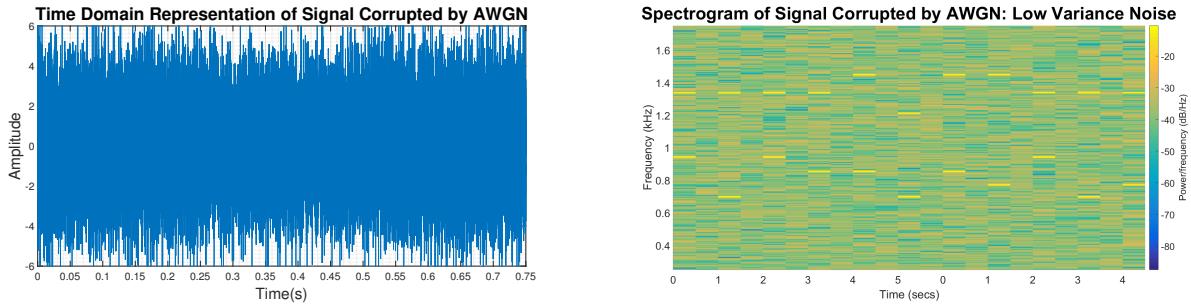


Figure 40: Spectrogram and magnitude spectrum of dial tone signal corrupted with low noise

As mentioned, the spectrogram provides the aerial view of the FFT. For the key classification process described above the magnitude spectrum of each segment has to be studied. As such, the magnitude spectra of numbers 1 and 9 are graphed in figure 41. In addition, the magnitude spectrum of a time segment during which no number is dialled is also graphed.

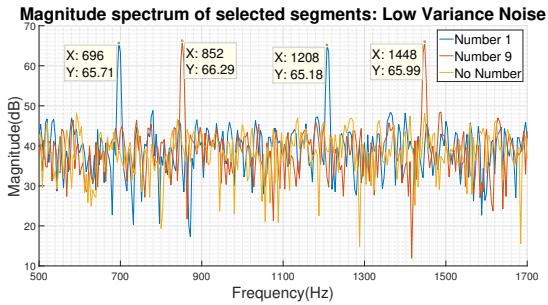


Figure 41: Magnitude spectrum of selected FFT segments

From 41, it is clear that adding WGN with  $\sigma = 2$  does not significantly hinder the key classification process although the time domain signal is almost unrecognizable. Next, noise with  $\sigma = 8$  is added to the signal. The spectrogram and the FFT of selected segments are graphed in 42.

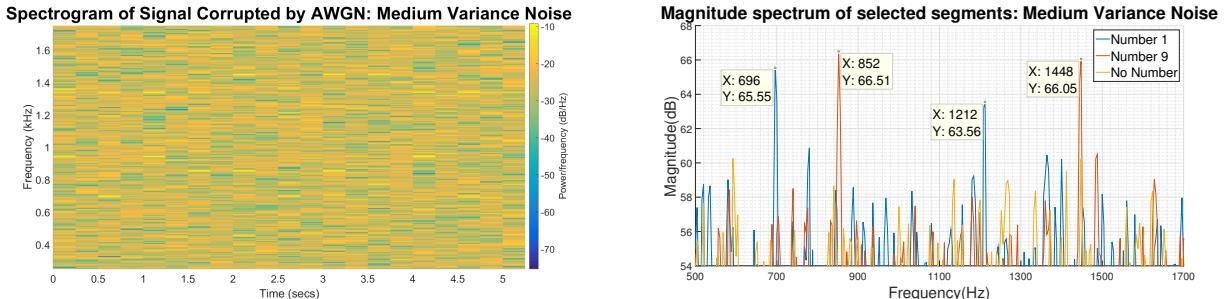


Figure 42: Spectrogram and magnitude spectrum of dial tone signal corrupted with medium noise

With the standard deviation of the noise equal to 8, key classification is still possible, however, the threshold set below which the signal is considered to not have any spectral peaks needs to be adjusted. If the threshold is not adjusted, a segment during which a number is not dialled might be interpreted as a segment in which a number is pressed. A smart algorithm can be developed to dynamically determine the value of the threshold based on the input power of the signal. Finally, the spectrogram and FFT of selected segments of a signal that has been corrupted with AGWN with  $\sigma = 15$  are graphed in figure 43. The key classification process can no longer be performed because the spectral peaks due to the sinusoidal tones are not distinguishable from the spectral peaks due to the noise.

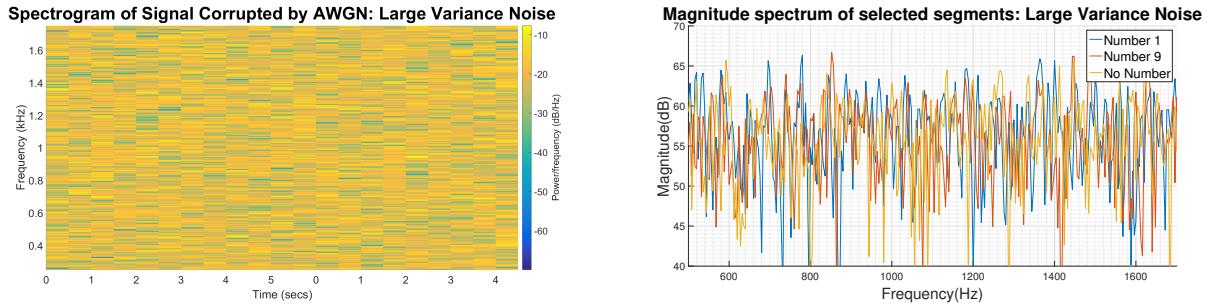


Figure 43: Spectrogram and magnitude spectrum of dial tone signal corrupted with medium noise

Again the time domain signal produced when medium and large noise were added is graphed in figure 44. It is amazing to see that with medium noise, the maximum amplitude of the signal has been increased more than 15 fold; even with this significant increase the key classification process can be conducted with reasonable certainty. Also, note that no sophisticated signal processing has been performed. Neither has a priori knowledge of the signal has been taken into account. This reveals the immense power of analysing signals in the frequency domain.

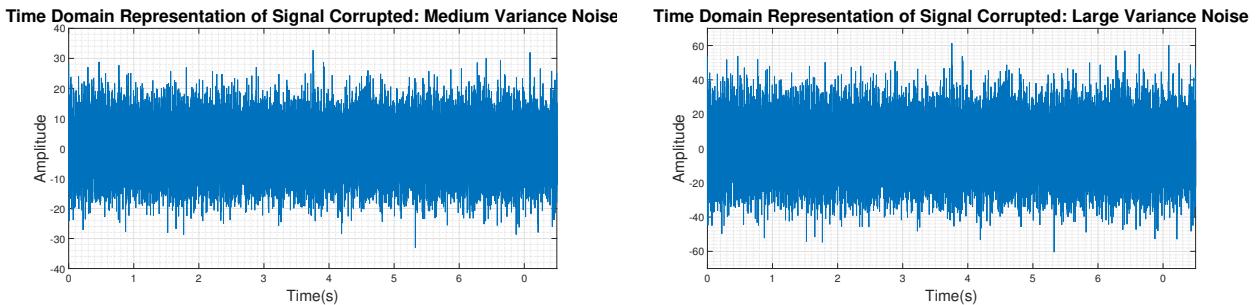


Figure 44: Spectrogram and magnitude spectrum of dial tone signal corrupted with medium noise

### 3.4 Real World Signals: Respirator Sinus Arrhythmia from RR-Intervals

a) Figure 45 shows the estimate of the PSD for the RRI time series using the periodogram method for trial 1 and 2.

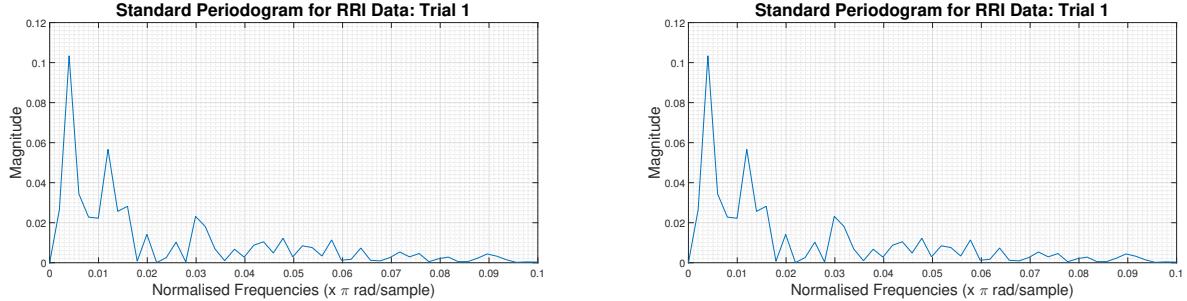


Figure 45: Periodogram based estimate of PSD for RRI time series: trial 1 and trial 2

Figure 46 shows the estimate of the PSD for the RRI time series using the periodogram method for trial 3.

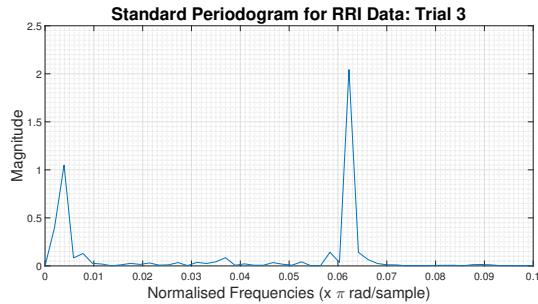


Figure 46: Periodogram based estimate of PSD for RRI time series: trial 3

Next, the averaged periodograms are obtained. The averaging is done by using 5, 150 sample long, segments. Since the FFT would only have been evaluated at 150 unique frequencies, the signal was zero-padded. The averaged periodogram for trial 1 and trial 2 is presented in figure 47.

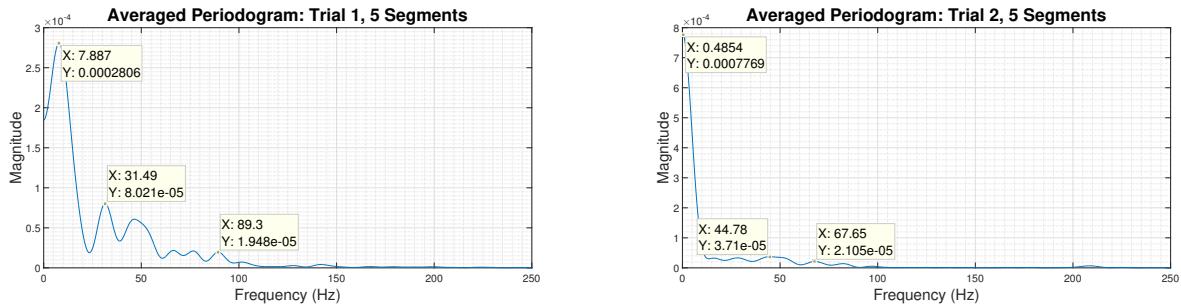


Figure 47: Averaged periodogram for RRI time series: trial 1 and trial 2

Lastly, the averaged periodogram for trial 3 is graphed in figure 48.

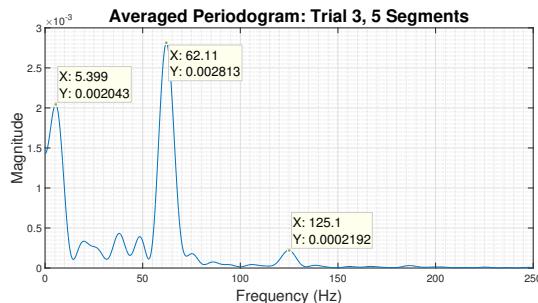


Figure 48: Averaged periodogram for RRI time series: trial 3

b) Large spectral components indicate that the test subject's heart rate is high whereas low spectral components indicate the test subject's heart rate is low. Trial 2 and trial 3 provides the RRI data when the test subject was asked to constraint his breathing. It is evident that the pattern of breathing, taking faster shallower breaths, in trial 2 cause the test subject's heart rate to slow down in comparison to the resting heart rate measured in trial 1. The pattern of breathing, taking long deep breaths, in trial 3 cause the test subject's heart rate to increase significantly.

## 4 Optimal Filtering - Fixed and Adaptive

### 4.1 Wiener Filter

- The signal-to-noise (SNR) ratio for  $z[n]$  is equal to  $1/0.1^2 = 100$ ; in decibels the SNR is  $20dB$ . Using the statistics  $R_{xx}$  and  $p_{zx}$ , the optimal coefficients calculated are presented in table 1.

b	1	2	3	4	5
$W_{opt}$ - Empirical Data	0.9990	1.9995	2.9963	2.0005	0.9973
$W_{opt}$ - Normalised Data	0.2168	0.4367	0.6563	0.4379	0.2155

Table 1: Optimal coefficients of wiener filter

The coefficients obtained are extremely close to the ideal values. Using normalised data, the power of the signals  $x[n]$  and  $\hat{y}[n]$  will be identical. However, when normalised data is used, the coefficients obtained are a scaled version of the original.

Noise Variance	b				
	1	2	3	4	5
0.2	0.9954	1.9944	2.9924	1.0000	0.9895
0.5	0.9927	1.9906	2.9895	1.9996	0.9836
1	0.9896	1.9864	2.9862	1.9992	0.9769
3	0.9820	1.9757	2.9780	1.9982	0.9604
7	0.9724	1.9624	2.9678	1.9969	0.9397
10	0.9670	1.9549	2.9620	1.9962	0.9280

Table 2: Effects of increasing noise variance on estimated coefficients of wiener filter

Increasing the noise variance causes the error incurred in the estimation of the coefficients to increase. The coefficients seem to be decreasing in magnitude; this is expected. As noise variance is increased, the signal  $z[n]$ , which is used as the reference signal in the estimation process starts to look more like white noise and less like the filtered signal  $y[n]$ . White noise is characterised by samples that are uncorrelated. The unknown system  $w[n]$  is a moving average filter and has the effect of introducing dependence into the independent white noise signal. If the reference signal has a large proportion of noise and a low proportion of the correlated outputs from the unknown system, then the identified system, which aims to produce a signal that is statistically similar to  $z[n]$ , should introduce lesser dependence into the signal  $\hat{y}[n]$ . By decreasing the magnitude of the variables, less dependence is introduced.

The mean-squared-error in the estimated signal  $\hat{y}[n]$  is graphed for different values of  $N_w$  in figure 49. Notice that for  $N_w > 4$ , the mean-squared-error does not decrease. This shows that over-modelling the system does not result in a better model. The estimated coefficients are extremely small and have almost no effect in determining the current sample.

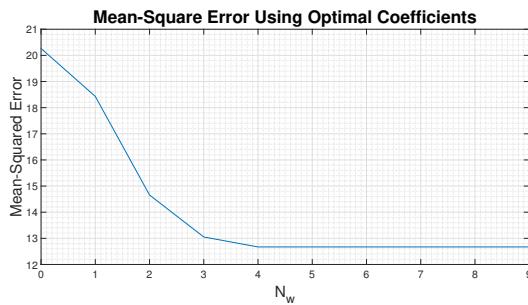


Figure 49: Mean-squared-error in the estimated signal  $\hat{y}[n]$

- The number of multiplications and additions can be determined by analysing each step of the algorithm. To estimate the complexity of computing the ACF, the number of multiplications and addition at each value of  $\tau$  is considered. Assuming that the data length,  $N$ , is significantly large than the number of coefficients,  $N_w + 1$ , the number of multiplications at each value of  $\tau$  is approximately,  $N$  whereas the number of additions is also  $N$ . The ACF has to be calculated for  $N_w + 1$  delays. As such, the ACF takes  $(N_w + 1)N$  multiplications and

$(N_w + 1)N$  additions. The cross correlation also takes  $(N_w + 1)N$  multiplications and  $(N_w + 1)N$  additions and the inverting of the matrix takes approximately,  $(N_w + 1)^3$  operations.

## 4.2 Least Mean Square (LMS) Algorithm

1. The LMS algorithm is used for system identification. The results obtained after the LMS algorithm has run through the entire data set are presented in table 3. The signal  $y$  was not normalised to have unity variance; this allows a clearer comparison to the ideal values.

Noise Variance	b									
	Wiener Solution					LMS Solution, $\mu = 0.01$				
	1	2	3	4	5	1	2	3	4	5
0.2	0.9954	1.9944	2.9924	1.0000	0.9895	0.9942	2.0516	2.9536	2.0113	0.9897
0.5	0.9927	1.9906	2.9895	1.9996	0.9836	0.9873	2.1154	2.8954	2.0251	0.9755
1	0.9896	1.9864	2.9862	1.9992	0.9769	0.9824	2.1623	2.8522	2.0363	0.9652
3	0.9820	1.9757	2.9780	1.9982	0.9604	0.9685	2.2814	2.7434	2.0621	0.9401
7	0.9724	1.9624	2.9678	1.9969	0.9397	0.9517	2.4295	2.6084	2.0958	0.9089
10	0.9670	1.9549	2.9620	1.9962	0.9280	0.9420	2.5126	2.5313	2.1137	0.8903

Table 3: Comparison of wiener solution and solution obtained using LMS algorith

Both the algorithms have utilised the same amount of information however the estimated coefficients are slightly different. The main short-coming of the Wiener solution is that stationarity was assumed and in real-world applications, stationarity is not always guaranteed. However, in this case, the signal  $z$  is stationary and thus the wiener solution is expected to be closer to the ideal solution than the LMS algorithm. Moreover, the LMS algorithm does not have the luxury of processing all the data at once; it processes data with a moving window and thus the results obtained are expected to be less accurate. This is observed in table 3. For every value of noise variance, the LMS solution produces a worse answer than the Wiener solution.

2. The LMS algorithm is used to estimate the signal  $\hat{y}[n]$ . The output using the LMS is graphed in figure 50 for two values of the adaptive gain,  $\mu$ . For small sample numbers, the error signal is both scenarios is large; as the number of samples increases and the amount of information on which the estimated signal is generated increases, the error decreases. The adaptive gain  $\mu$  is an extremely important parameter in the LMS algorithm. Observing the results obtained using  $\mu = 0.1$ , it is clear that the error signal settles to around 0 by the 50<sup>th</sup> sample. The error signal obtained using  $\mu = 0.01$  settles to around 0 at approximately the 300<sup>th</sup> sample. Although using a smaller value of  $\mu$  translates to slower convergence, the estimated signal is bounded. Notice that using  $\mu = 0.1$ , a very large overshoot appears in the estimated signal; in many applications, such overshoots can cause complete failure and thus a large value of  $\mu$  is not always preferred. Note that if the value of  $\mu$  is large, convergence is not guaranteed.

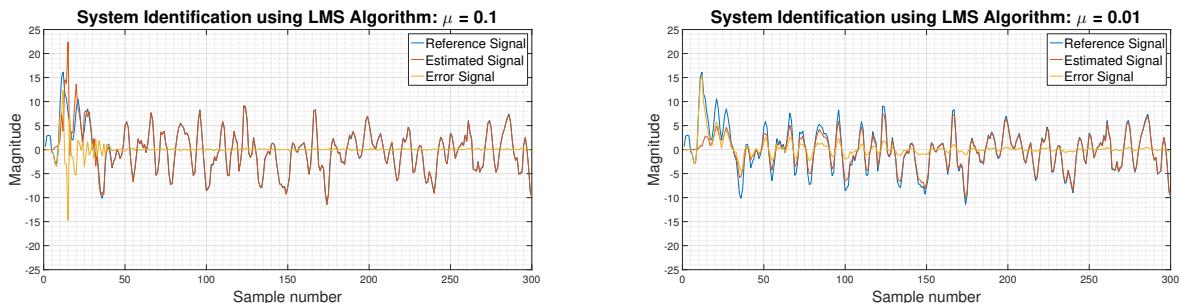


Figure 50: System identification using LMS algorithm

Next, the evolution of the coefficients is analysed for 4 distinct values of  $\mu$ . The results obtained for  $\mu = 0.002$  and  $\mu = 0.01$  are graphed in figure 51.

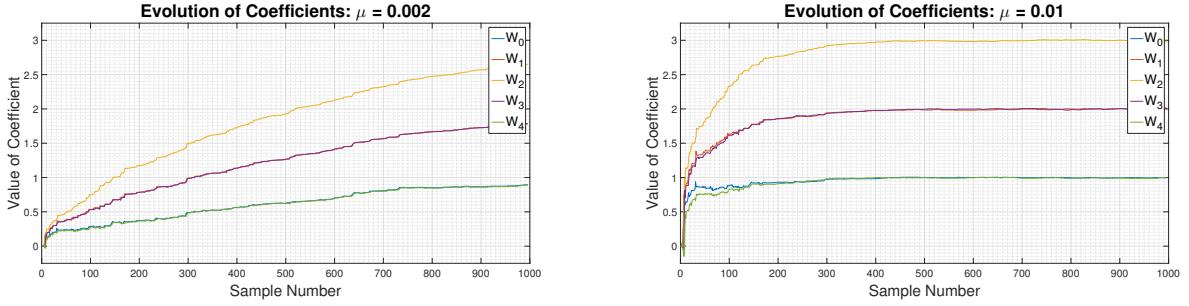


Figure 51: Evolution of coefficients for small  $\mu$

For small values of  $\mu$  the coefficients gradually approach their asymptotic values. The gradual approach towards the asymptotic values accounts for the lack of an overshoot. Notice that very small values of  $\mu$  will cause the convergence to be extremely slow. The results obtained for  $\mu = 0.1$  and  $\mu = 0.4$  are graphed in figure 52.

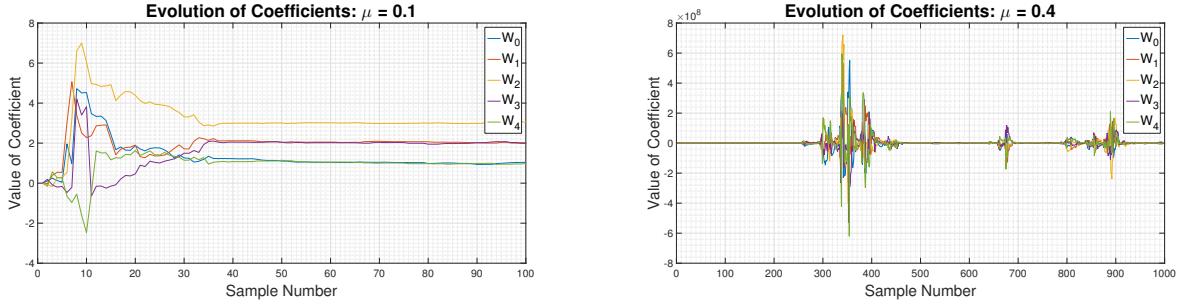


Figure 52: Evolution of coefficients for small  $\mu$

For  $\mu = 0.1$ , the approach towards the asymptotic values is not gradual. The large adaptive step size causes the coefficients to overshoot from their ideal values. The overshoot in the coefficients translates into an overshoot in the estimated signal as observed in figure 50. However, notice that the coefficients converge by the 100<sup>th</sup> sample. The rate of convergence is extremely fast. Arbitrarily increasing the step size does not result in increased convergence. Notice when  $\mu = 0.4$ , the coefficients do not converge.

### 4.3 Gear Shifting

Gear shifting enables the algorithm to change its learning rate,  $\mu$  based on certain parameters. The gear shifting algorithm that is experimented with is presented in equation (20). The algorithm was first suggested by Matthews & Xie [2]. The learning rate is increased when the power of the input signal increases. This is a valid proposition because having increased input power means that the signal's amplitude has increasing. If the learning rate is increased, the estimate is likely to follow the rise in amplitude. Next, the learning rate also depends on the value of the error incurred. Again this is wise; a large error requires a large correction whereas a small error requires a small correction. The learning rate will adapt to the error incurred.

$$\mu(n+1) = \mu(n) + \rho e(n)\mathbf{x}^T(n)e(n-1)\mathbf{x}(n-1) \quad (20)$$

Figure 53 is used to compare the two algorithms based on their rise time.

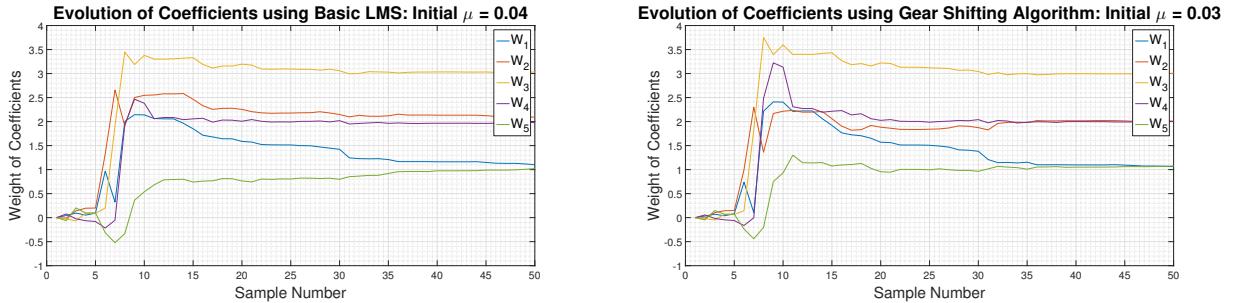


Figure 53: Comparison of gear shifting algorithm to basic LMS algorithm

It is clear that the improvement is not very significant. In section 5, another gear-shifting algorithm called the Normalised LMS will be discussed. The improvement observed in normalised LMS is significant.

#### 4.4 Identification of AR Process

1. Using the LMS algorithm, the coefficients of the AR(2) process are estimated. The results obtained for small values of  $\mu$  are graphed in figure 54.

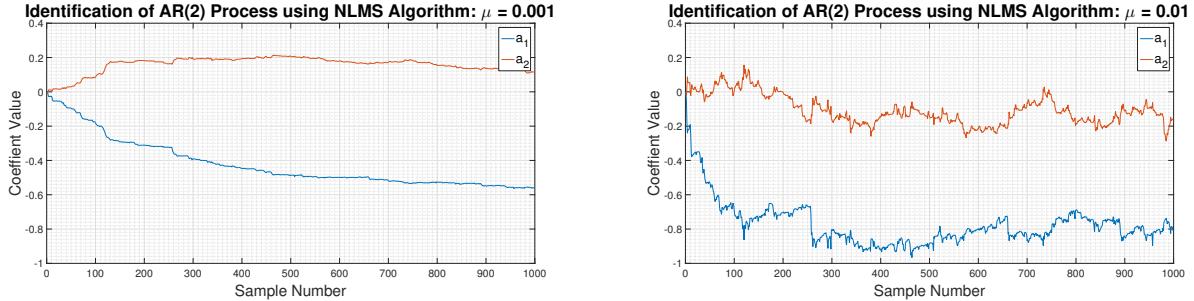


Figure 54: Identification of AR(2) process using the LMS: small values of  $\mu$

Although the rate of converge is slow, the function works well and is correctly identifying the model parameters. In figure 55, the results for large values of  $\mu$  are presented. Again the results are exactly as expected; beyond a certain threshold value of  $\mu$ , the LMS results obtained are not accurate. The system becomes unstable.

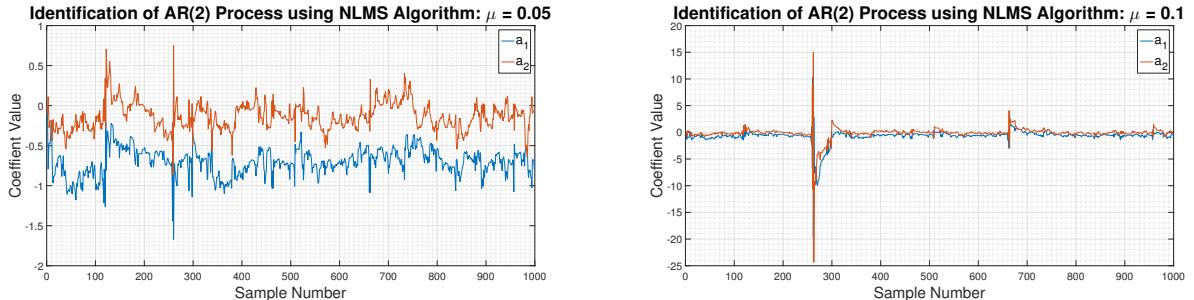


Figure 55: Identification of AR(2) process using the LMS: large values of  $\mu$

## 4.5 Dealing With Computational Complexity: Sign Algorithms

Figure 56 shows how the signed-error and signed-regressor algorithms identify the AR parameters in section 4.4.

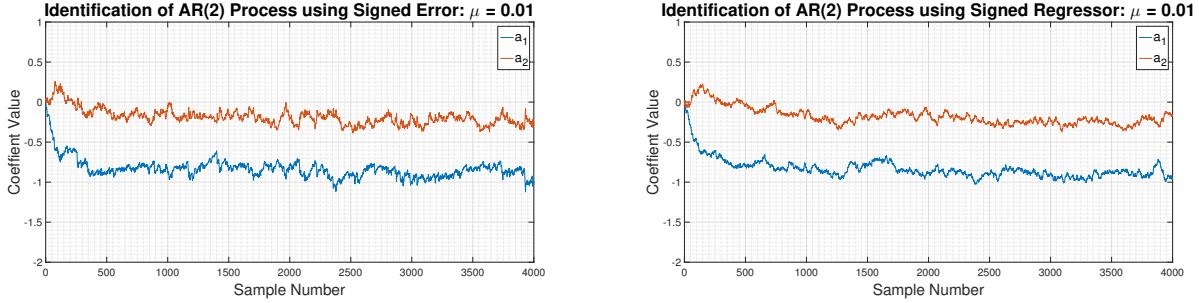


Figure 56: Identification of AR(2) process using the signed-error and signed-regressor algorithms

Figure 57 shows how the sign-sign algorithm identifies the AR parameters in section 4.4.

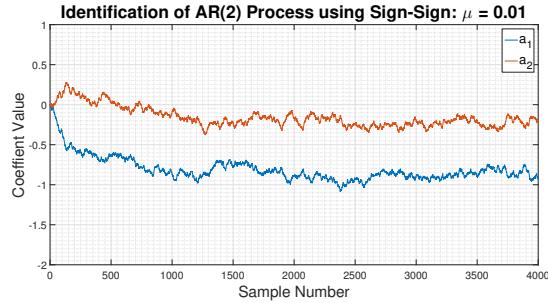


Figure 57: Identification of AR(2) process using the sign-sign algorithm

All the algorithms presented have their pros and cons. The signed-error algorithm seems to converge the fastest out it has the largest variance around the steady state value. Both the signed-regressor and sign-sign algorithms have slower convergence ranges but have smaller steady state error. The signed-regressor and sign-sign algorithms seem almost identical however notice the small difference near sample number 3800. It seems as if the signed-regressor algorithm is poorer at dealing with shocks at steady state as compared to the sign-sign algorithm.

## 5 A Real World Case Study: Vinyl Denoising

1. Listening to both tracks, the ticking sound can be clearly distinguished. The tick is of a very regular nature; it is periodic. Thus it is expected that the noise will appear as a sharp peak in the frequency spectrum. The periodograms of both channels are graphed in figure 58. Note that the frequency axis of the periodogram represents normalised digital frequencies. To obtain the equivalent frequency of the analogue signal, the sampling frequency needs to be known. In this case, the signal was sampled at  $44.1\text{kHz}$ . The spectral components corresponding to the ticks cannot be fully distinguished just by considering figure 58, however it is highly likely that the ticks are due to the large peaks at  $\omega = 0.033$ .

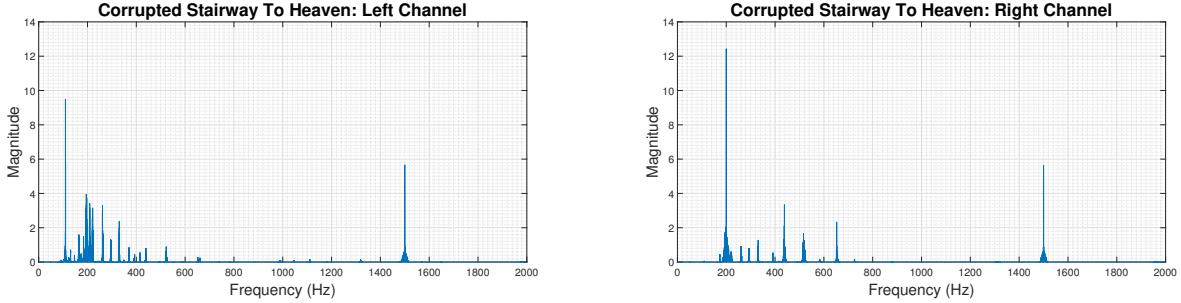


Figure 58: Periodogram of corrupted stairway to heaven tracks

2. The periodogram of the clean signal is graphed in figure 59.

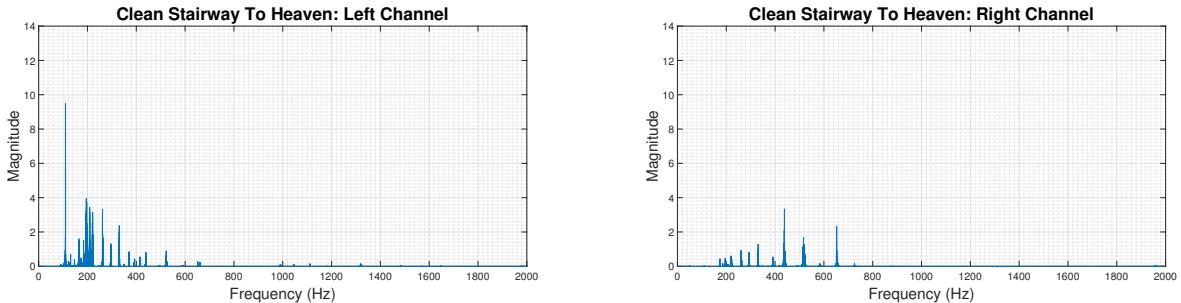


Figure 59: Periodogram of clean stairway to heaven tracks

Notice that the axis of the clean and corrupted periodograms are the same. Comparing the periodograms, it is clear that the initial conjecture that the ticks in the signal are due to the large peak at  $\omega = 0.033$  is confirmed. Both the left and right tracks have a peak at  $\omega = 0.033$  whereas the clean signals do not. The right channel has another peak at approximately  $\omega = 0.005$ . To locate the exact position of the peaks, the difference between the periodogram of the clean signal and the periodogram of the corrupted signal is graphed in figure 60.

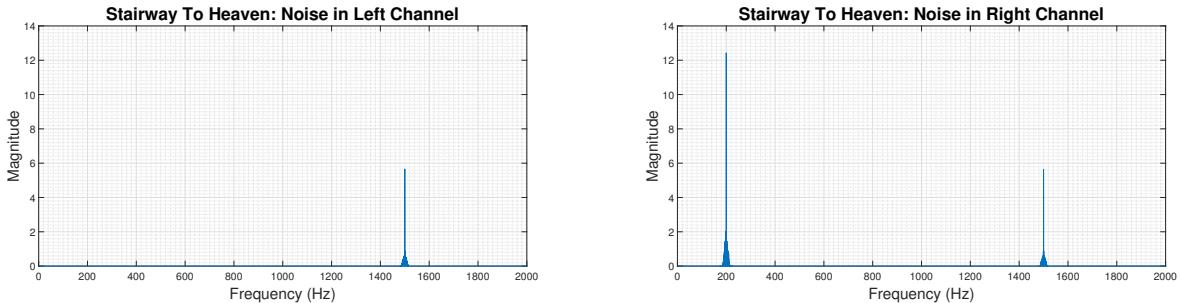


Figure 60: Difference between periodograms of clean signals and periodograms of noisy signals

By considering the periodograms graphed in figure 60, the exact position of the spectral components causing the ticking sound can be determined.

3. To remove the two spectral peaks observed in figure 60, band stop filters need to be designed. To implement the filtering process, multiple design methodologies were considered. Firstly, a Chebyshev Type 1 bandstop

filter was designed using the MATLAB command `cheby1`. The range of frequencies that have to be stopped are [180, 200] and [1465, 1525]; the edge frequencies were specified appropriately. The order of the filter was chosen to be 8; for an 8<sup>th</sup> order stopband filter to be designed using the MATLAB `cheby1` function, the argument of the MATLAB function corresponding to order has to be specified as 4. The magnitude response of the filter was not ideal. The noise was not sufficiently attenuated. When the filtered signal was played, the ticking noise was still obvious. Next, a notch filter was considered. The MATLAB function `irrnotch` was utilised however again the desired stopband attenuation could not be obtained.

Lastly, the MATLAB command `designfilt` was used to design a digital stopband filter. The order of the filter was chosen to be 20. The butterworth algorithm was used by the `designfilt` function to meet the specifications. The design produced using the `designfilt` function was actually a cascade of 10 2<sup>nd</sup> order filters. The magnitude response of the 2 filters, designed using `desginfilt`, are shown in figure 61.

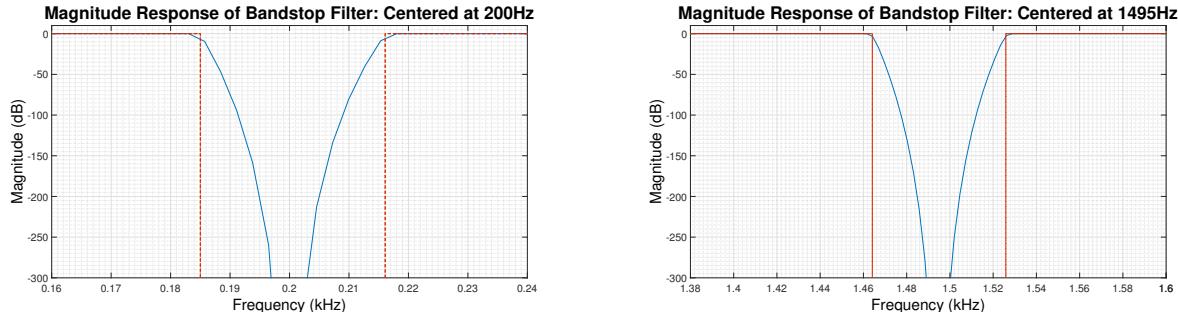


Figure 61: Frequency response of stopband filters designed using `designfilt`

The filters were extremely effective at removing the peak at occurred at 1495Hz; after filtering, no ticks could be heard in the left channel. The peak occurring at 200Hz proved harder to remove; after filtering the ticking sound could still be heard however it is greatly attenuated.

4. Figure 62 shows the periodogram of the 3 signals. The periodogram has been plotted in terms of decibels rather than in terms of the absolute magnitude. The decibel scale accentuates the differences between the three signals.

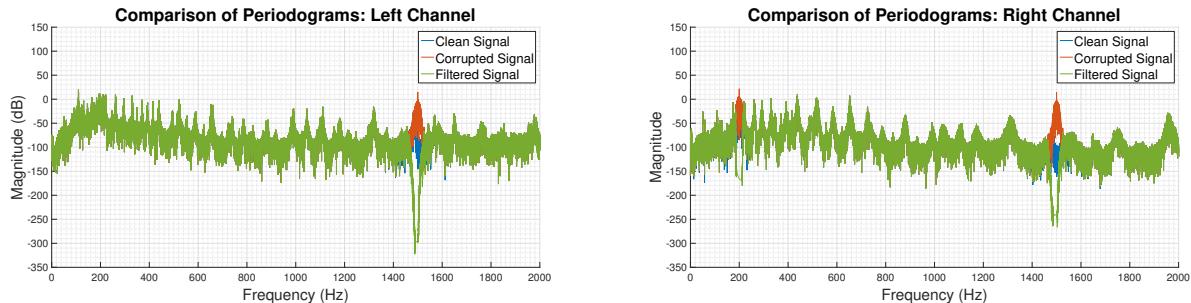


Figure 62: Periodogram of clean, corrupted and filtered signal

It is clear that the filter is performing its operation. The periodogram of the filtered signal looks very similar to the periodogram of the clean signal; the large red peaks present in the corrupted signal have been removed. Notice that the filter has also attenuated some spectral components that were in the clean signal; this is inevitable, however the spectral components of the clean signal that have been attenuated contribute very little power to the signal and thus attenuating these components does not introduce a great deal of distortion. Lastly, note that the stopband filter that acts in the frequency range [180, 220] has not attenuated the noise as much as the filter that acts in the frequency range [1465, 1525]. This is expected as the filter has a much smaller frequency range to act on.

Channel	Relative Error Estimate	
	Corrupted	Filtered
Left	0.22754	0.00721
Right	1.32961	0.09198

Table 4: Relative Error Before and after Filtering

The relative error of the corrupted signal and the filtered estimate with respect to the *ground truth* recording is presented in table 4. The filtering has certainly enhanced the quality of the corrupted signal; the relative error is approximately 30 times lower for the left estimate and 15 times lower for the right estimate. It should be noted although the relative error has been significantly reduced, it is often important to identify the source of the error. From figure 62, it is clear that the error in the corrupted signal is due to the large peaks whereas in the filtered signal, the error incurred is due to the over-attenuation at the stopband frequencies. This explains why the filtered estimate for the right channel has a greater relative error; two stopband filters have been used and thus the error due to over-attenuation is incurred in 2 unique frequency bands.

The relative error does not provide information about the difference in phase between the clean and filtered signals.

5. The normalised LMS (NLMS) algorithm is used to filter the corrupted stairway to heaven signal. The algorithm has 2 parameters,  $\mu$  and  $N_w$ . The effects of changing both these parameters is studied. In addition, the corrupted stairway to heaven track was filtered using the basic LMS algorithm. The findings are presented in figure 63.

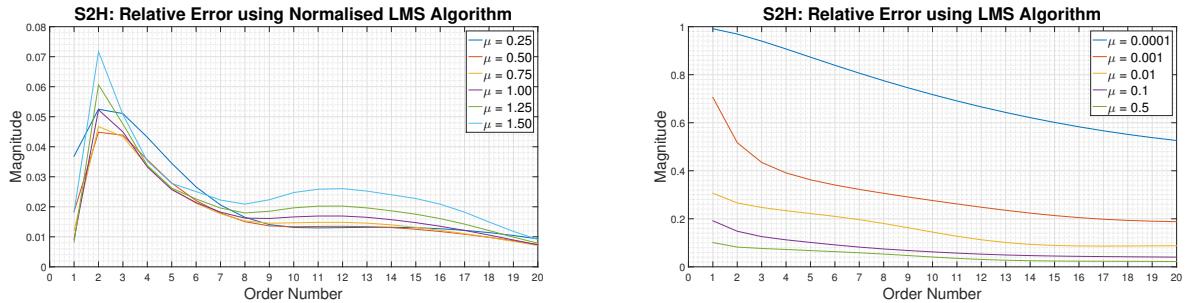


Figure 63: Relative error analysis for Stairway to Heaven soundtrack for both NLMS and basic LMS algorithms

Firstly, by observing the y-axis of both graphs, it is clear that the NLMS algorithm returns a much lower relative error as compared to the basic LMS algorithm. The value of  $\mu$  for which the basic algorithm performs the best is 0.5; note that even then the performance of the basic LMS algorithm only approaches that of the NLMS, it does not surpass it. Secondly, choosing a value of  $\mu = 0.5$  was a matter of trial and error. Slightly increasing the value of  $\mu$  beyond 0.5 will cause the basic LMS to lose its convergence property. In addition, the value of  $\mu = 0.5$  does not represent a universal bound below which convergence is guaranteed. Convergence properties of the basic LMS algorithm are dependent on multiple factors, one of which is the input signal. This highlights the most important benefit of the NLMS algorithm. Convergence properties for the NLMS algorithm are a lot more predictable and independent of the input signal.

It should be noted that for the LMS algorithm, a high learning rate is preferred for this data set. The noise in the time series is extremely non-stationary; it is highly periodic and thus the PDF of the noise is constantly changing. High learning rates are preferred for signals with highly non-stationary noise. If the noise is stationary, a small learning rate might decrease the overall error in the estimate since it has a lower steady state error.

Next, the squared error using the optimal value of  $\mu$  for this data set is studied. The model order is set at 20,  $\mu_{NLMS} = 1$  and  $\mu_{LMS} = 0.5$ . Figure 64 shows how the squared error in the estimate varies as a function of time.

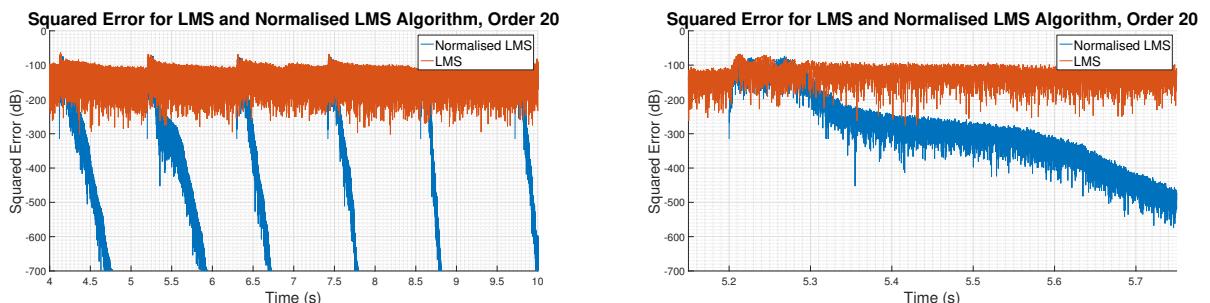


Figure 64: Analysis of convergence rates for LMS and NLMS algorithms

The graph reveals two things about the LMS and NLMS algorithms. Firstly, it is clear that the LMS algorithm has a much higher steady state error than the NLMS algorithm. The rate at which the ticking happens is high and thus the NLMS algorithm did not have time to settle to a steady state. Secondly, it is clear that the NLMS algorithm has a much higher rate of convergence than the LMS algorithm. This is expected since the learning rate of the NLMS is adaptive. When the input power is high, the learning rate increases and the overshoot in the estimate is reduced.

Both graphs shown in figure 63 indicate that increasing the model order will decrease the error incurred in the estimate. This was discussed in section 2.4 as well. The MDL and AIC models are necessary to put an upper bound on the model order; the model selection criterion place a penalty that quantifies the computational complexity involved in increasing the model order. Without considering computational complexity for the time being, even with the model order being set at 20, the ticking sound can still be heard.

Figure 65 shows the corrupted signal, the filtered estimate and the clean reference signal for a short duration of the soundtrack. Even when the NLMS algorithm is used with an order of 20, the ticks are still not completely attenuated; in fact, the ticks are still quite significant. Note that the value of  $\mu$  was set to 1. As observed in figure 63, the value of  $\mu$  does not matter for large model orders.

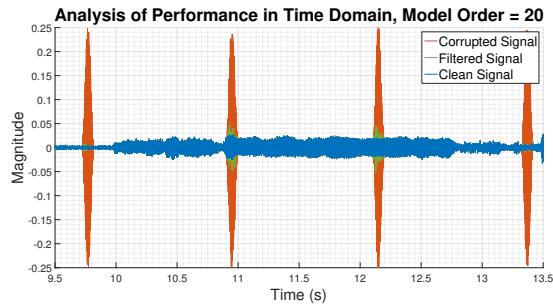


Figure 65: Time domain analysis of filtered estimate made with model order 20

To improve the quality of the sound such that a similar performance to that obtained by utilising stopband filters is achieved, the model order is set to 100. Once again the time domain signal is analysed. The stark improvement in the quality of the signal is clear; the clean signal graphed in blue almost completely overlaps the green signal. The green peaks that could be observed in figure 65 are not present in figure 66.

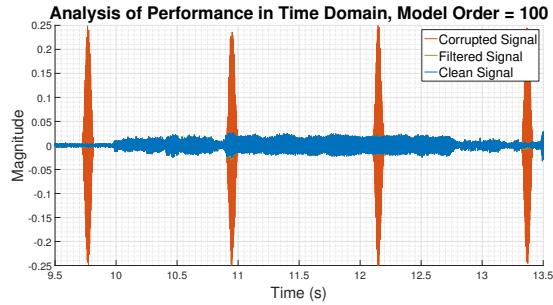


Figure 66: Time domain analysis of filtered estimate made with model order 100

Estimates made using model order 20 and model order 100 are compared in the frequency domain. The graphs are presented in figure 67. As seen in the time domain, the periodogram of the estimate produced using a model order of 20 does not match the periodogram of the clean signal; using a model order of 100, a much better estimate is produced. It is important to note that the NLMS filter is not indiscriminate in the frequencies that it removes from the corrupted signal. Unlike the stopband filters, the NLMS filter does not over-attenuate at the frequencies which contain noise.

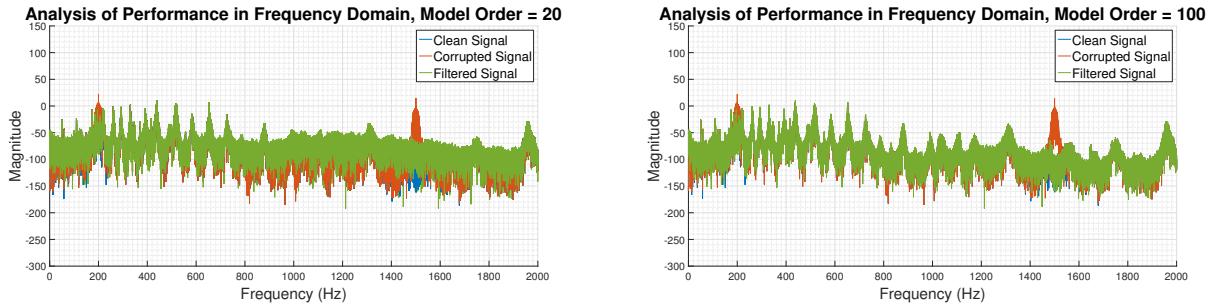


Figure 67: Frequency domain analysis of Stairway to Heaven soundtrack

6. Firstly, before the LMS and NLMS algorithms are used to filter the Liquid Tension Experiment soundtrack, its periodogram is graphed in figure 68. Again, to accentuate the ticking noise a decibel scale is used.

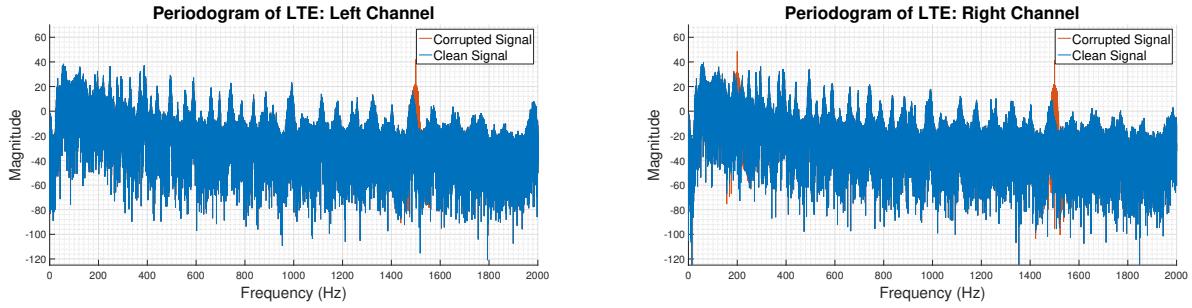


Figure 68: Periodogram of clean and corrupted Liquid Tension Experiment soundtrack

Notice that the corrupted Liquid Tension Experiment soundtrack has spectral peaks at the same frequency as the corrupted Stairway to Heaven soundtrack. However, the periodogram of the clean Liquid Tension Experiment soundtrack has significant spectral components at the frequencies at which the noise occurs. As such, using an indiscriminate filter such as the stopband filters designed above would cause the sound to be distorted. As discussed above, the NLMS algorithm does not over-attenuate the noise and thus it is a much better filter to use for such applications.

Similar analysis to that performed in section 5.5 is performed below. The relative error incurred using both the NLMS and the LMS algorithm is studied. The results obtained are graphed in figure 69.

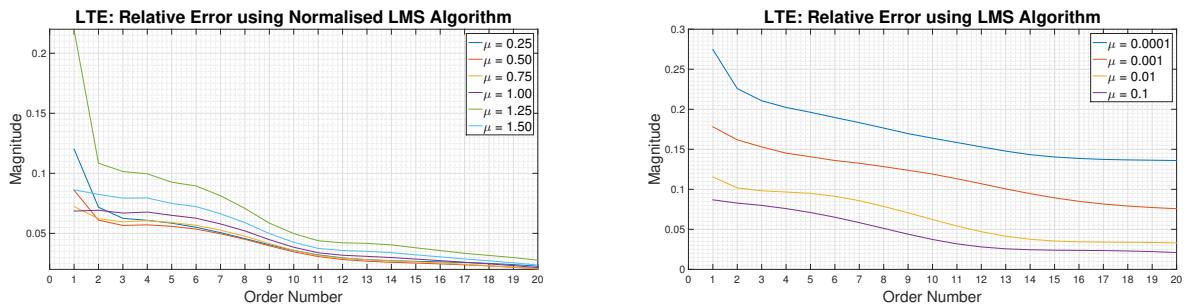


Figure 69: Relative error analysis for Liquid Tension Experiment soundtrack for both NLMS and basic LMS algorithms

The first point to note is that the LMS algorithm was only stable up to a learning rate of  $\mu = 0.1$ . This is in contrast to the 0.5 value that was obtained for the Stairway to Heaven soundtrack. The discrepancy corroborates the assertion made that the stability of the LMS algorithm is heavily dependent on the input signal that is used.

Once again it is clear that the NLMS algorithm out performs the LMS algorithm in terms of the error incurred. Next, by observing the errors obtained using the NLMS algorithm, patterns start to emerge. Firstly, having a learning rate of higher than 1 causes the error incurred to increase. Secondly, decreasing the learning rate such that  $\mu < 1$  typically decreases the error incurred. The learning rate controls how fast the algorithms corrects its output. As mentioned above, large learning rates are ideal for signals that are highly non-stationary. This

ensures that the coefficients of the model converge to the ideal value before the ideal value changes. Low learning rates typically trade the rate of convergence for a decrease in the steady state error. The optimal learning rate for the NLMS and LMS algorithms differs. This is expected because the NLMS algorithm can afford to have a smaller initial learning rate and still converge faster than the LMS algorithm.

The periodograms obtained by filtering using the NLMS algorithm with model order equal to 20 and model order equal to 100 are presented in figure 70.

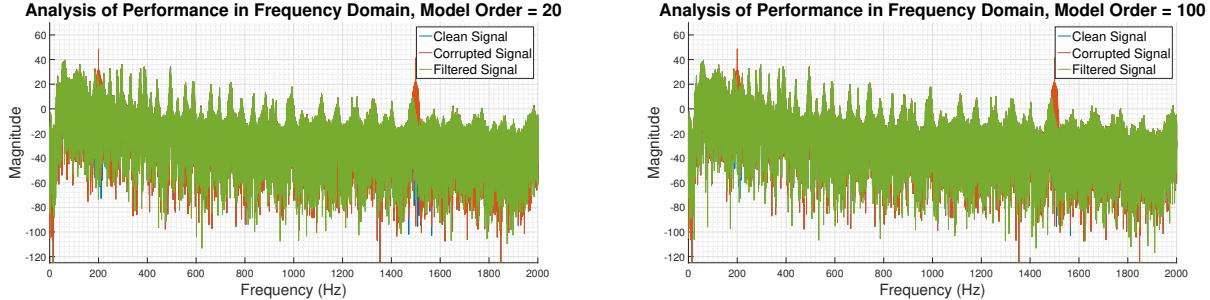


Figure 70: Frequency domain analysis of Liquid Tension Experiment soundtrack

Unlike in the case of the Stairway to Heaven soundtrack, increasing the model order does not have a significant effect on the periodogram of the Liquid Tension Experiment soundtrack. This is observed not only in the periodogram but also when the filtered sound is played. The ticks are almost indistinguishable when an order 20 filter is used. Increasing the filter order does not have a significant effect in improving sound quality. The Liquid Tension Experiment soundtrack has significant spectral contents at the frequencies at which the noise occurs and thus attenuating the ticks even slightly has a significant impact on the Signal-to-Noise ratio.

## References

- [1] D. P. Mandic and J. A. Chambers, "On stability of relaxive systems described by polynomials with time-variant coefficients," in IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, vol. 47, no. 10, pp. 1534-1537, Oct 2000.
- [2] V. J. Mathews and Z. Xie, "A stochastic gradient adaptive filter with gradient adaptive step size," in IEEE Transactions on Signal Processing, vol. 41, no. 6, pp. 2075-2087, Jun 1993.