

# Advanced signal processing

Coursework, due on 9<sup>th</sup> April 2015

# 1. Random signals and stochastic processes

## 1.1. Statistical estimation

### *Estimation of the Uniform distribution law statistical properties*

The vector  $X$  contains 1000 samples of the uniform random variable  $X \sim U(0,1)$ . Each sample corresponds to the realisation of the random variable  $X$ . The samples indicate a uniformity, which is due to the stationary characteristic of the signal. To show this, the expectation and the standard deviation of the signal will be calculated in the following.

For an infinite number of samples, the theoretical expectation of the signal is given by:

$$E(X) = \int_{-\infty}^{+\infty} X \cdot pdf(X) dX \Rightarrow E(X) = \int_0^1 X dX = \left[ \frac{X^2}{2} \right]_0^1 = \frac{1}{2}$$

Its theoretical standard deviation is as shown below.

$$\begin{aligned} \sigma_X &= \sqrt{E\left((X - E(X))^2\right)} = \sqrt{E\left(\left(X - \frac{1}{2}\right)^2\right)} = \sqrt{\int_{-\infty}^{+\infty} \left(X - \frac{1}{2}\right)^2 \cdot pdf(X) dX} = \sqrt{\int_0^1 \left(X - \frac{1}{2}\right)^2 dX} \\ &= \sqrt{\int_0^1 \left(X^2 - X + \frac{1}{4}\right) dX} = \sqrt{\left[\frac{X^3}{3} - \frac{X^2}{2} + \frac{X}{4}\right]_0^1} = \sqrt{\left(\frac{1}{3} - \frac{1}{2} + \frac{1}{4}\right)} = \sqrt{0.08333333} \approx 0.28867 \end{aligned}$$

The theoretical expectation and standard deviation are thus constant across the realisations. As a consequence, the signal is stationary. On the other hand, the number of samples is practically finite so approximations must be used.

1) The expectation is approximated by the sample mean given by

$$\hat{m} = \frac{1}{N} \sum_{n=1}^N x[n],$$

This estimator is computed in MatLab with the *mean* command.

2) The standard deviation is approximated by the sample estimator given by

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

This last one is calculated with the *std* command in MatLab.

The estimates  $\hat{m} = 0.4891$  and  $\hat{\sigma} = 0.2838$  result from these commands in MatLab.

This shows that  $\hat{m}$  and  $\hat{\sigma}$  are accurate estimators, when compared to the theoretical ones.

3) The matrix  $X_{1:10}$  contains 10 realisations of a 1000 samples. The mean and the standard deviation are calculated for each of the ten realisation. As shown below, their values do cluster around the theoretical value and their bias estimated by  $E(X) - \hat{m}$  equals to zero.

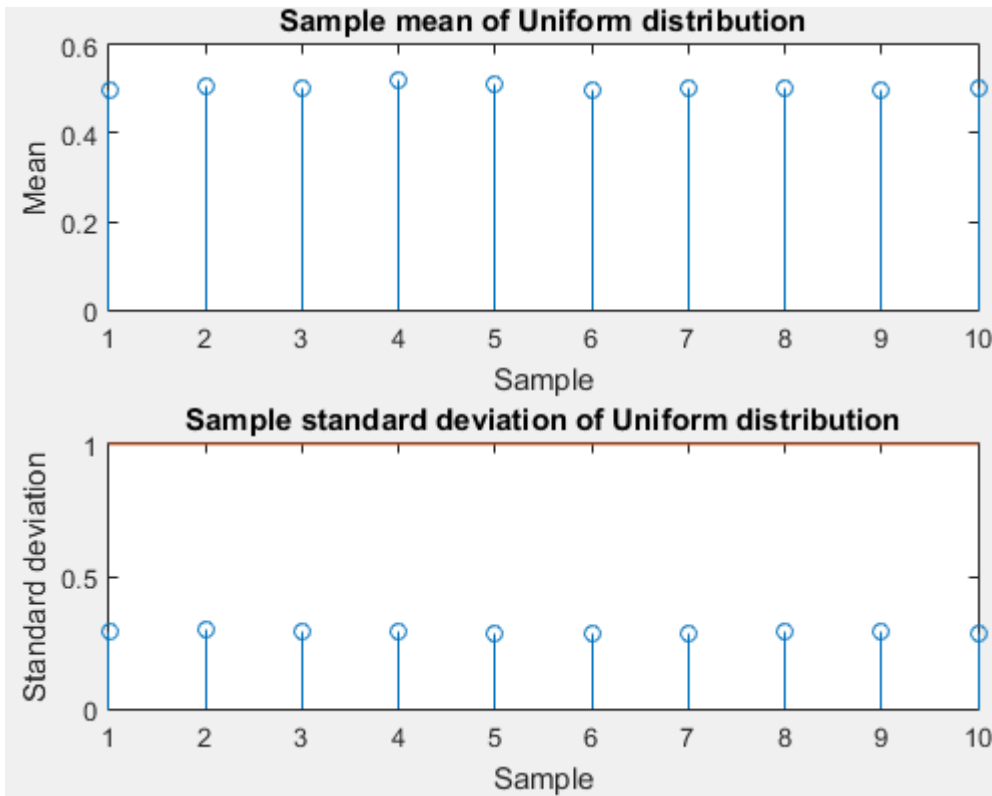


Figure 1: Mean and standard deviation for each realisation

4) The probability density function (PDF) of the signal can be calculated with the command *hist*. The function returns a vector which is then normalized by being divided by its area (using *trapz*). The result is thus a measurement of the density of each realisation.

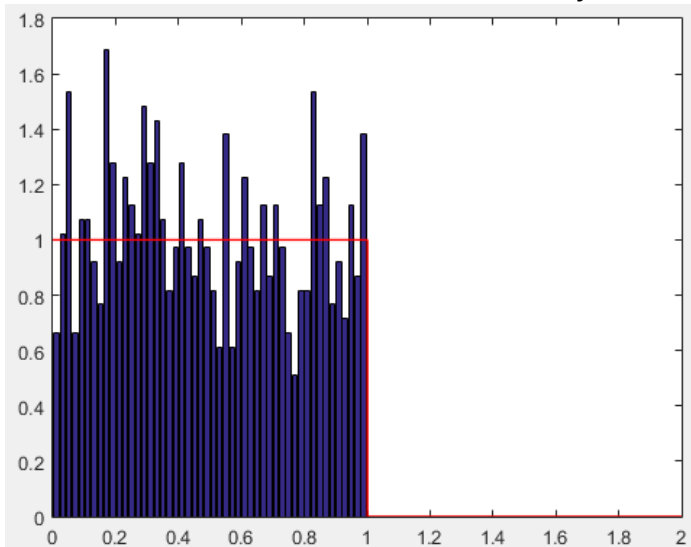


Figure 2: Plot of the PDF for a uniform distribution,  $N=1000$ , bins=50

This is the resulting plot of the results from the *hist* instruction, for a 1000 samples random variable.

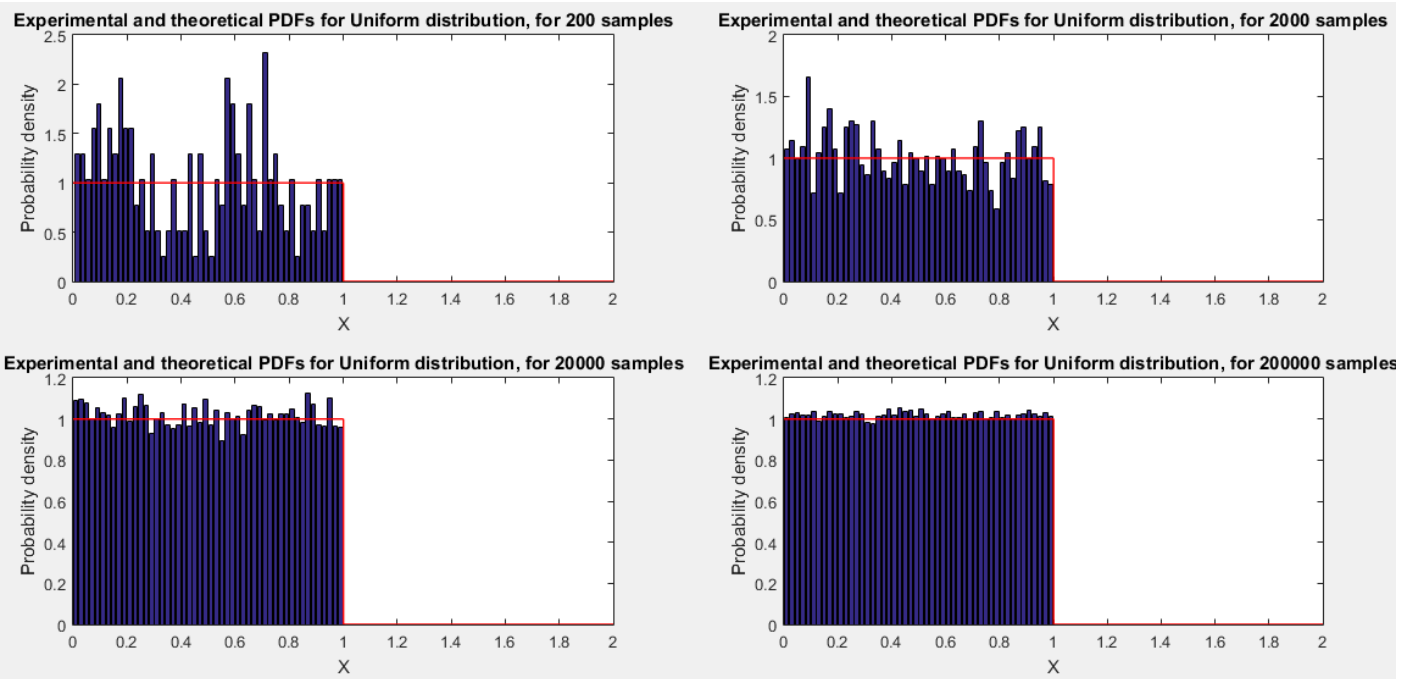


Figure 3: PDF of X, uniform distribution, various N (samples) values: 200, 2000, 20000 and 200000

Figure 3 shows the estimated PDF converges towards the theoretical PDF as the number of sample increases. This is because the theoretical PDF corresponds to signal with an infinity of realisations, and having more samples brings the experimental PDF closer to this target theoretical one.

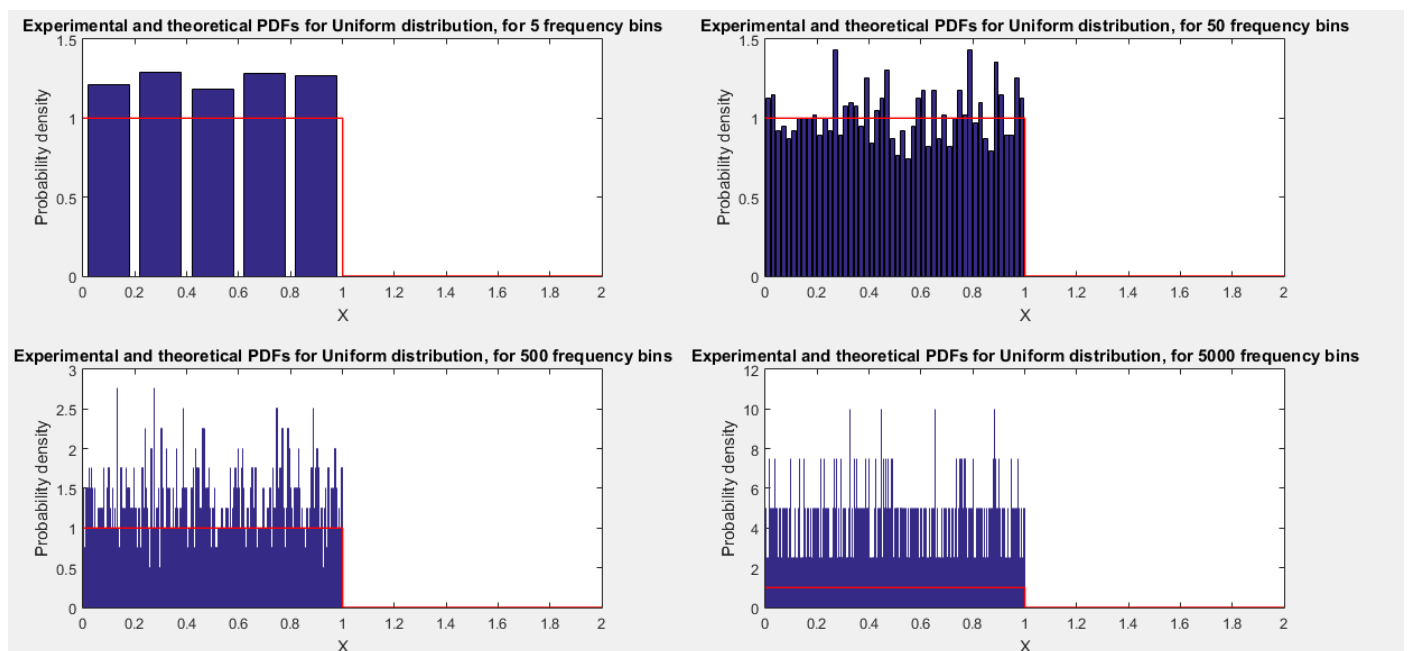


Figure 4: PDF of X, uniform distribution, N=2000 samples, for different frequency bins values

This last figure highlights that above 50 frequency bins, the estimated PDF is spikier with a lower uniformity. The precision of the bin values is enhanced as the total number of frequency bins increases. This also lowers the difference between a frequency bin value and the next one's value. Now, as there is a fixed and finite number of samples (2000) produced, for a high level of precision chosen, the samples will differ a lot between each other. The distribution will no longer be uniform. On the other hand, it is interesting to note that increasing the number of samples N will improve the uniformity of the estimated PDF.

## Estimation of the Normal distribution law statistical properties

5) The vector column  $X$  now contains  $N=1000$  realisation of the zero-mean and unit-variance Gaussian variable  $X \sim N(0,1)$  (using command `randn`). The mean and the standard deviation are computed as in the section before, with `mean` and `std`. The resulting estimates are  $\hat{m} = -0.0689$  and  $\hat{\sigma} = 0.9937$ .

The mean and the standard deviation are then computed for each of the 10 realisations of 1000-samples of random Gaussian noise.

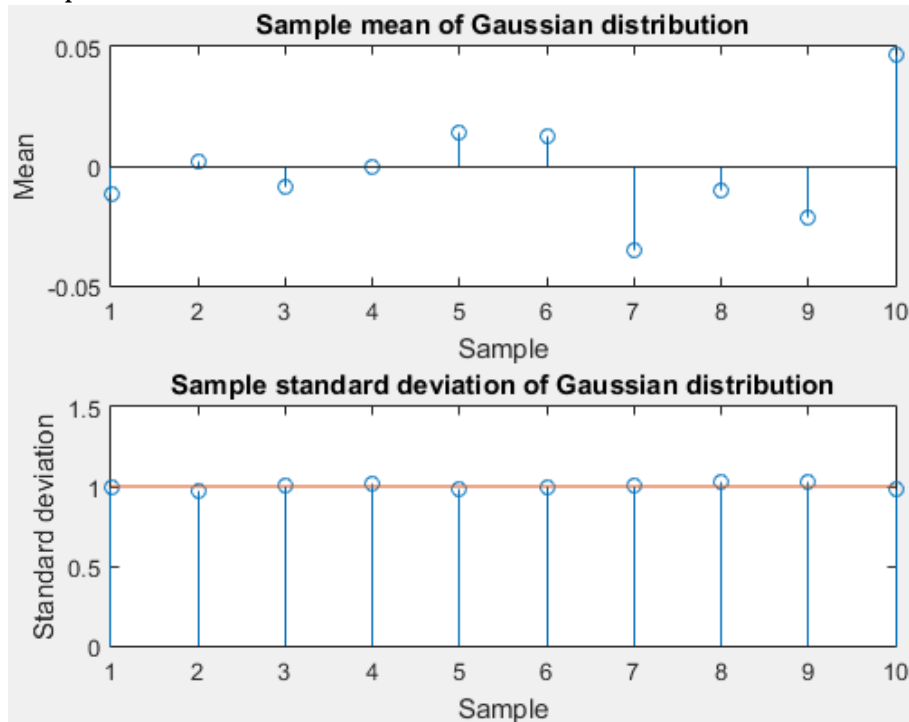


Figure 5: Sample mean and standard deviation of the PDF of  $X$  (Gaussian distribution),  $N=200000$ , bins=50

The estimations of the mean and standard deviation both cluster again to the theoretical values of 0 and 1.0 respectively. Their respective bias is thus equal to zero.

The PDF of the Gaussian variable  $X$  is estimated as for the uniformly distributed variable, using the command `hist`. The following plot results from a MatLab estimation of the PDF for  $N=200000$  samples and BINS=50 frequency bins.

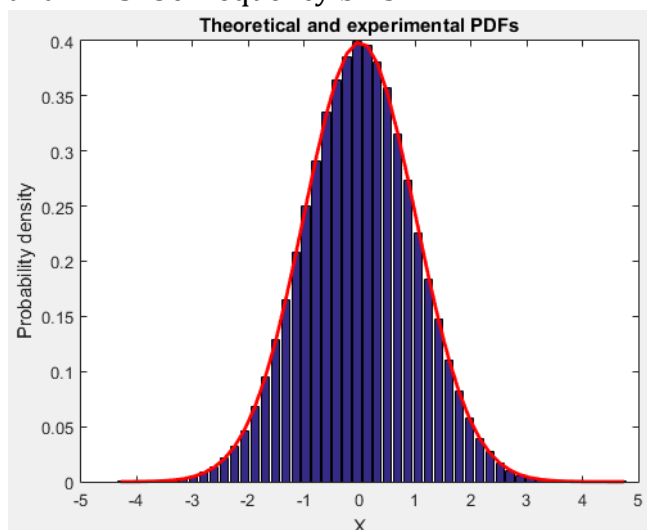


Figure 6: Sample mean and standard deviation of the PDF of  $X$  (Gaussian distribution),  $N=200000$ , bins=50

The following figure describes how the estimated PDF varies with the number of samples  $N$ .

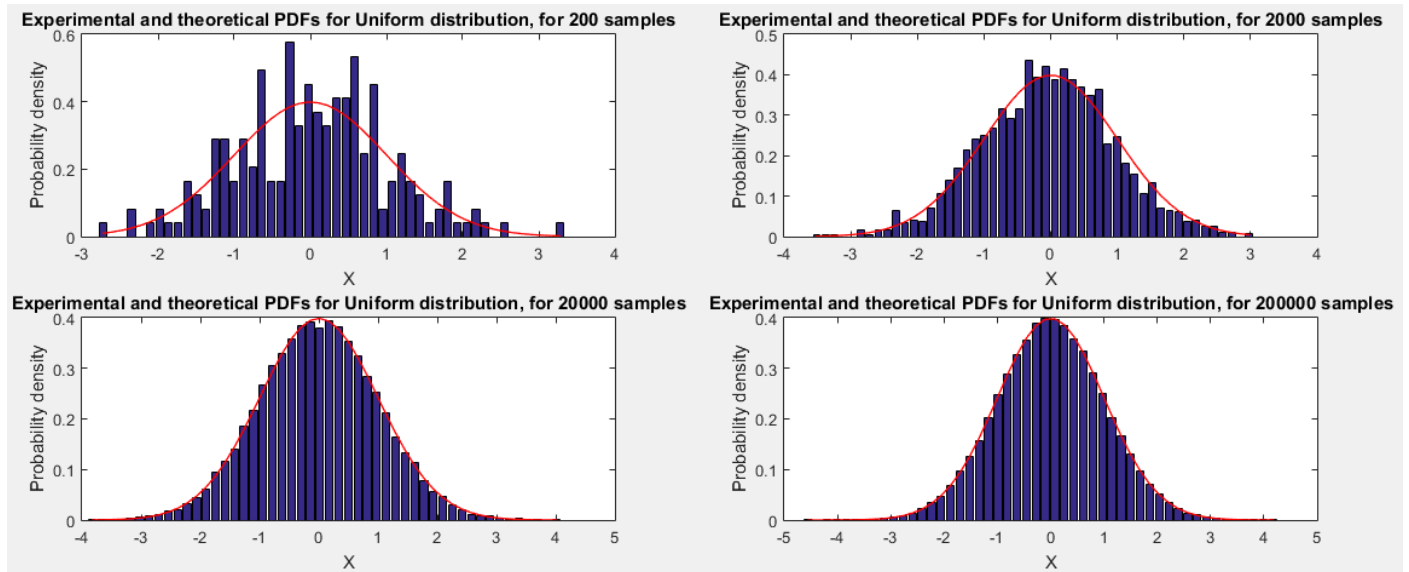


Figure 7: PDF of  $X$ , Gaussian distribution, various  $N$  (samples) values: 200, 2000, 20000 and 200000

An increase in the number of samples will make the PDF converging towards the theoretical PDF. In the next figure, several values of frequency bins are tested. This reveals the same effects as for the Uniform distribution (see 1.1.4).

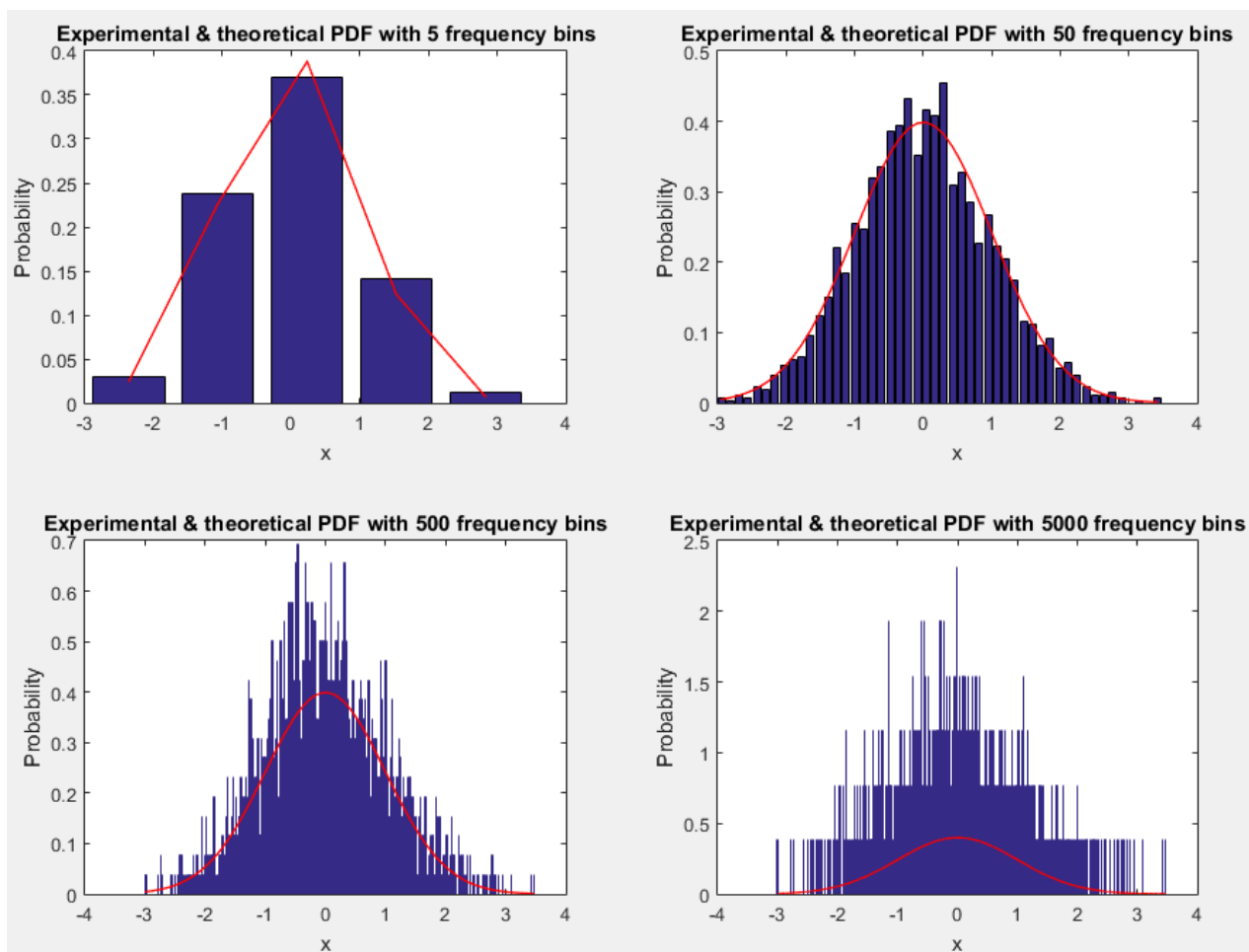


Figure 8: PDF of  $X$ , Gaussian distribution,  $N=2000$  samples, for different frequency bins values

## 1.2. Stochastic processes

The following three random process are generated with the following given Matlab code.

```
function v=rp1(M,N);
a=0.02;
b=5;
Mc=ones(M,1)*b*sin((1:N)*pi/N);
%Mc is a M x N matrix (sine wave value, sample number).
Ac=a*ones(M,1)*[1:N];
%Ac is a M x N matrix where each element is equal to
%the sample number multiplied by a.
v=(rand(M,N)-0.5).*Mc+Ac;
%Each element of Mc is multiplied by the outcome
%of the uniform distribution U(0,1).
%The result is then added to the element of Ac.

function v=rp2(M,N);
Ar=rand(M,1)*ones(1,N);
Mr=rand(M,1)*ones(1,N);
v=(rand(M,N)-0.5).*Mr+Ar;

function v=rp3(M,N);
a=0.5;
m=3;
v=(rand(M,N)-0.5)*m + a;
```

1) The mean and standard deviation are calculated for each process rp1, rp2 and rp3 with the number of realizations  $M=100$  and with the number of samples in each realization given by  $N=100$ .

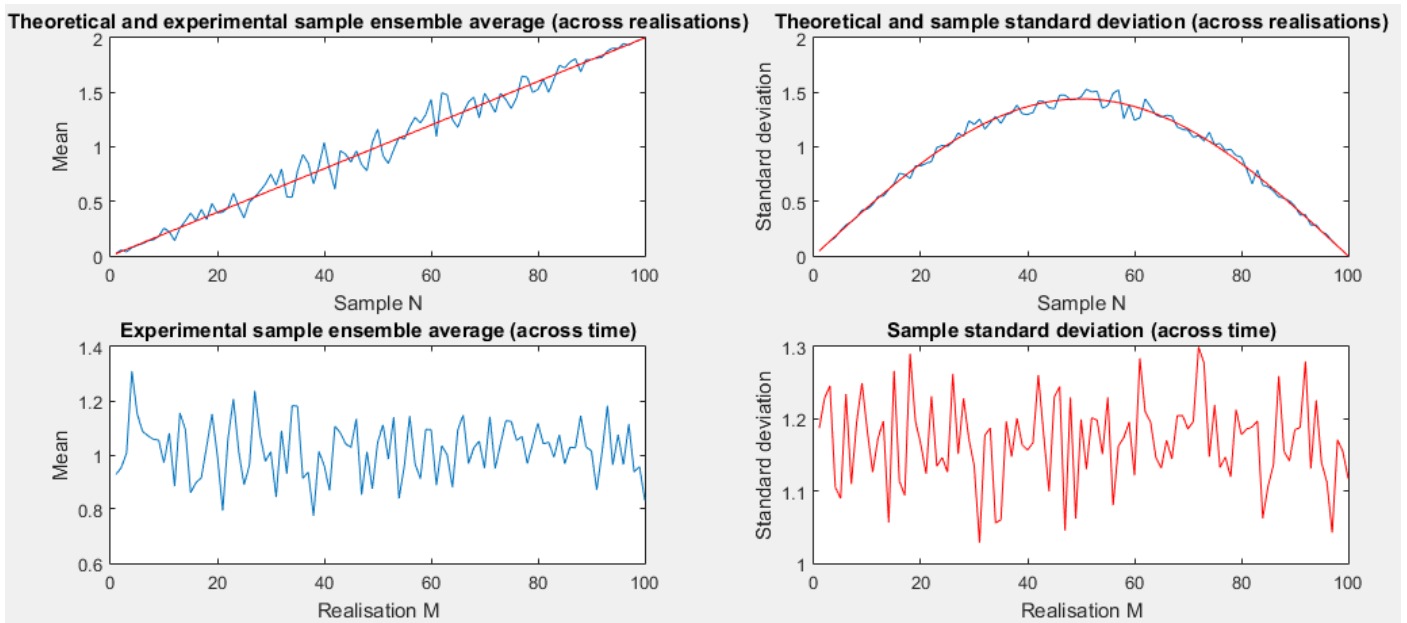


Figure 9: Sample ensemble average and sample standard deviation across realisations and time for rp1.  $N=100$  and  $M=100$

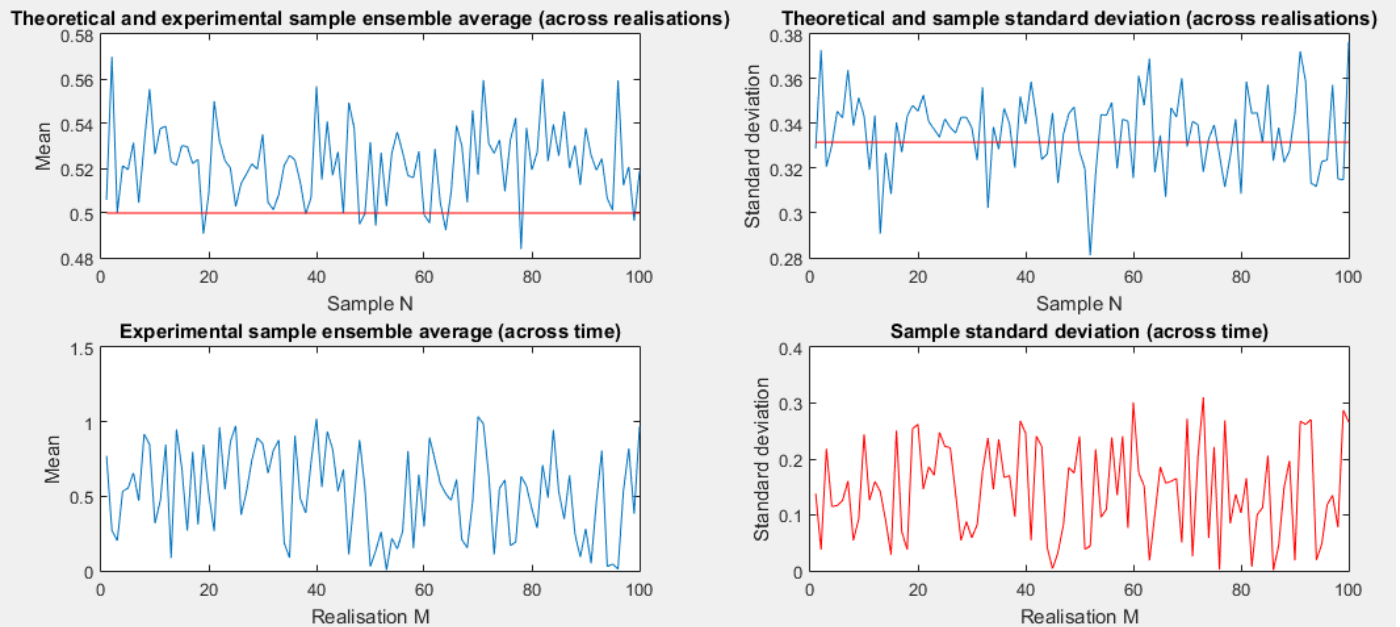


Figure 10: Sample ensemble average and sample standard deviation across realisations and time for rp2.  $N=100$  and  $M=100$

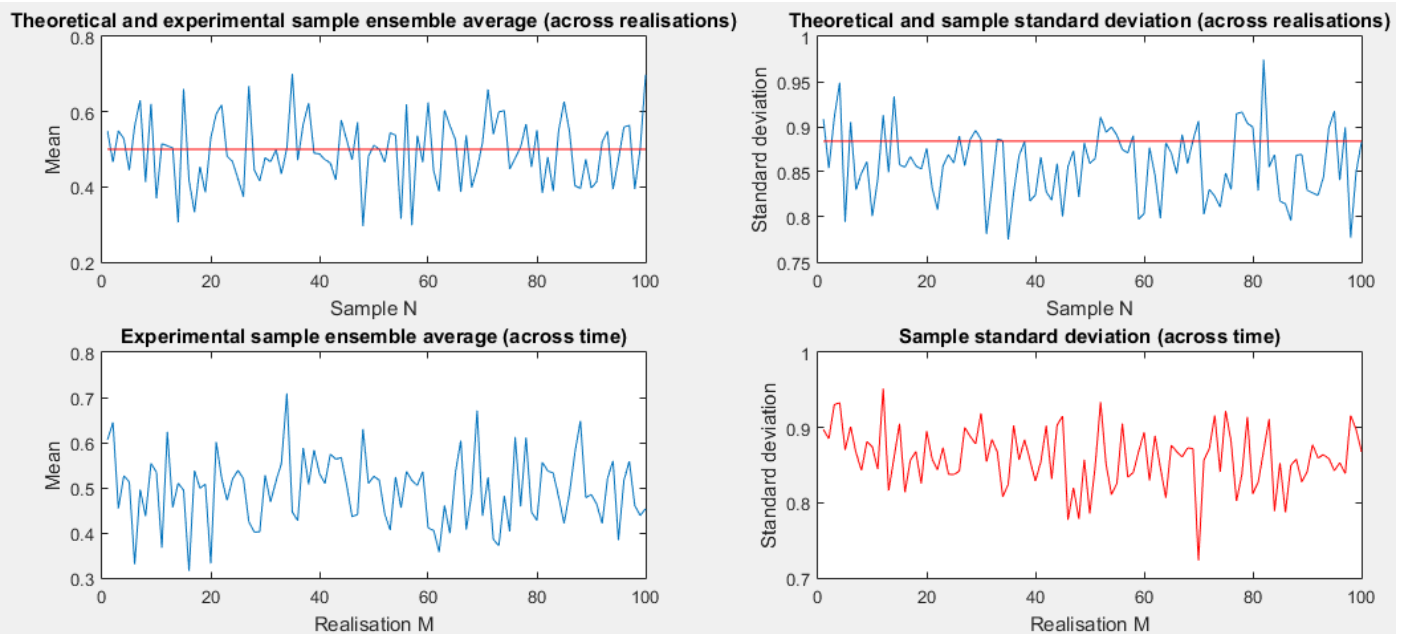


Figure 11: Sample ensemble average and sample standard deviation across realisations and time for rp3.  $N=100$  and  $M=100$

The mean of rp1 varies from 0 to 2 with time (sample), as does its standard deviation between 0 and 1.45. Its statistical parameters all vary with time. As a consequence, rp1 cannot be stationary. The mean of rp2 changes in the range from 0.49 to 0.57, and its standard deviation varies between 0.28 and 0.37. These relatively small variations are due to the finite number of samples (as explained before). Finally, the mean of rp3 varies between 0.3 and 0.7, and its standard deviation varies between 0.77 and 0.95. From these results, we can conclude that the process rp2 and rp3 are stationary.



3) The statistical properties of each of these processes can be used to verify their stationary status. This is done by deriving the theoretical mean and standard deviation for each process.

### Statistical properties of rp1

The following mathematical expression describes the signal  $v$  (rp1):

$$v[n] = 5A \sin\left(\frac{n\pi}{N}\right) + 0.02n \quad \text{where } A = X - 0.5, \text{ with } X \sim U(0,1)$$

The theoretical mean is  $E(v[n]) = E\left(5A \sin\left(\frac{n\pi}{N}\right) + 0.02n\right) = 0.02n + 5 \sin\left(\frac{n\pi}{N}\right) + E(A) = 0.02n$

The mean of rp1 is a straight line with a slope of 0.02.

$$\begin{aligned} \text{The standard deviation is } \sigma_v &= \sqrt{E\left(\left(v[n] - E(v[n])\right)^2\right)} = \sqrt{E\left(\left(5A \sin\left(\frac{n\pi}{N}\right) + 0.02n - 0.02n\right)^2\right)} \\ &= \sqrt{E\left(\left(5A \sin\left(\frac{n\pi}{N}\right)\right)^2\right)} = \sqrt{25 \sin^2\left(\frac{n\pi}{N}\right) E(A^2)}, \quad \text{where } E(A^2) = E((X - 0.5)^2) = \sigma_X^2 = 0.0833333 \\ &\Rightarrow \sigma_v = \sqrt{2.0833 \sin^2\left(\frac{n\pi}{N}\right)} \approx 1.443 \sin\left(\frac{n\pi}{N}\right) \end{aligned}$$

### Statistical properties of rp2

The signal  $v$  is now described by:

$$v[n] = x_0[n] \cdot x_1[n] + x_2[n] \quad \begin{aligned} \text{where } x_0[n] &\sim U(0,1) - 0.5 \\ x_1[n] &\sim U(0,1) \\ x_2[n] &\sim U(0,1) \end{aligned}$$

hence, we have the mean:  $E(v[n]) = E(x_0[n] \cdot x_1[n]) + E(x_2[n])$  where  $x_0[n] \cdot x_1[n]$  are independent  
 $\Rightarrow E(v[n]) = E(x_0[n]) \cdot E(x_1[n]) + E(x_2[n])$  where  $E(x_0[n]) = 0.5 - 0.5 = 0.0$   
 $\Rightarrow E(v[n]) = 0 + 0.5 = 0.5$

$$\begin{aligned} \text{The standard deviation is } \sigma_v &= \sqrt{E\left(\left(v[n] - 0.5\right)^2\right)} = \sqrt{E\left(v[n]^2 - v[n] + 0.25\right)} \\ &= \sqrt{E\left(\left(x_0[n] \cdot x_1[n] + x_2[n]\right)^2 - \left(x_0[n] \cdot x_1[n] + x_2[n]\right) + 0.25\right)} \\ &= \sqrt{E\left(x_0^2[n]x_1^2[n] + 2x_0[n]x_1[n]x_2[n] + x_2^2[n] - x_0[n] \cdot x_1[n] - x_2[n] + 0.25\right)} \\ &= \sqrt{E\left(x_0^2[n]x_1^2[n]\right) + E\left(2x_0[n]x_1[n]x_2[n]\right) + E\left(x_2^2[n]\right) - E\left(x_0[n] \cdot x_1[n]\right) - E\left(x_2[n]\right) + 0.25} \end{aligned}$$

Since  $x_0[n]$ ,  $x_1[n]$  and  $x_2[n]$  are independent, they are also uncorrelated.

so  $E(2x_0[n]x_1[n]x_2[n]) = 2E(x_0[n])E(x_1[n])E(x_2[n]) = 0$ , because  $E(x_0[n]) = 0$   
and thus  $E(x_0[n] \cdot x_1[n]) = E(x_0[n])E(x_1[n]) = 0$

$$\text{So we have: } \sigma_v = \sqrt{E\left(x_0^2[n]x_1^2[n]\right) + E\left(x_2^2[n]\right) - E\left(x_2[n]\right) + 0.25}$$

If  $x_0[n]$  and  $x_1[n]$  are independent, then  $x_0^2[n]$  and  $x_1^2[n]$  are also independent so:

$$\sigma_v = \sqrt{E\left(x_0^2[n]\right)E\left(x_1^2[n]\right) + E\left(x_2^2[n]\right) - 0.5 + 0.25} = \sqrt{-0.25 + \frac{0.083}{3} + \frac{1}{3}} \approx 0.33317$$

### Statistical properties of rp3

The signal  $v$  is given by the expression  $v[n] = 3A + 0.5$ , where  $A \sim U(0,1) - 0.5$

The mean is  $E(v[n]) = E(3A + 0.5) = 3E(A) + 0.5 = 0.5$

$$\begin{aligned} \text{and the standard deviation is } \sigma_v &= \sqrt{E\left(\left(v[n] - E(v[n])\right)^2\right)} = \sqrt{E\left(\left(3A + 0.5 - 0.5\right)^2\right)} \\ &= \sqrt{9E\left(A^2\right)} = 3\sigma_X \approx 0.867 \end{aligned}$$

2) The mean and standard deviation are calculated for each process rp1, rp2 and rp3 with the number of realizations  $M=1000$  and with the number of samples in each realization given by  $N=4$ . The following results are obtained:

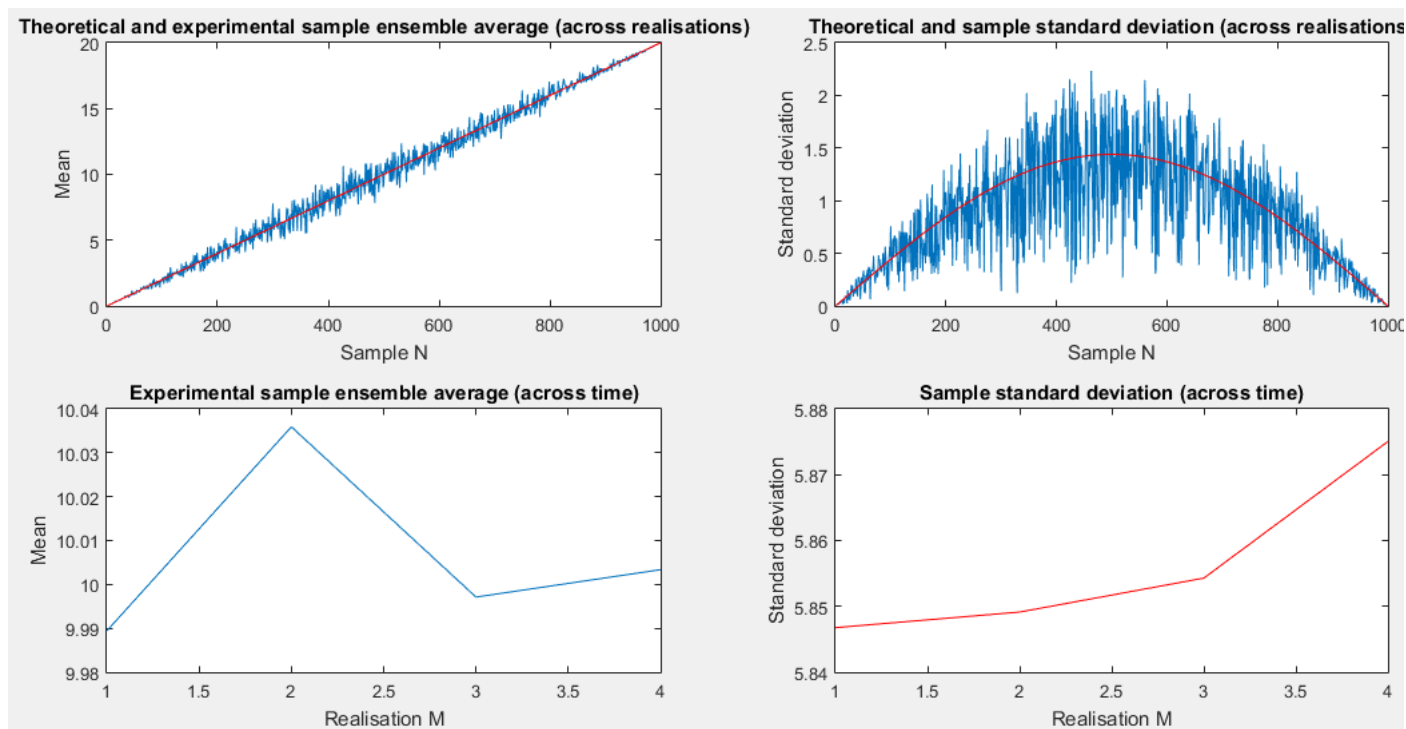


Figure 12: Sample ensemble average and sample standard deviation across realisations and time for rp1.  $N=1000$  and  $M=4$

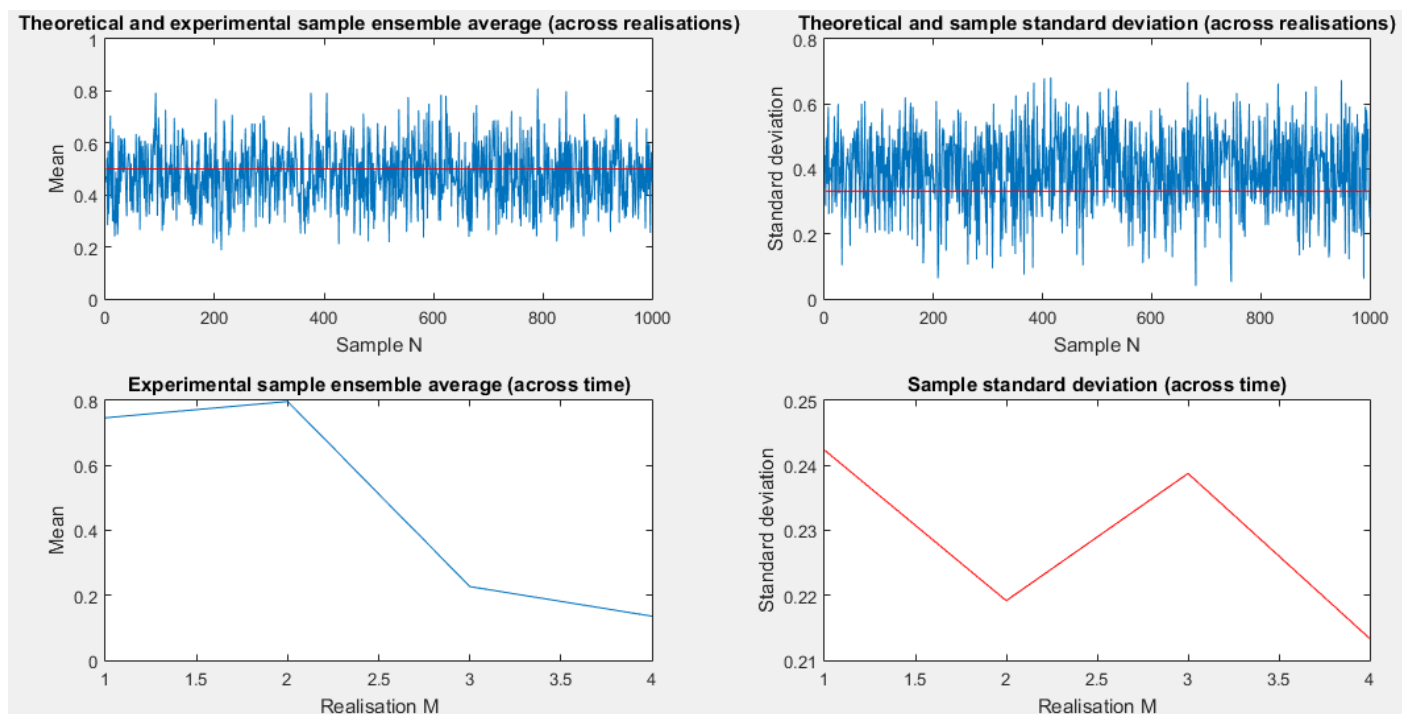


Figure 13: Sample ensemble average and sample standard deviation across realisations and time for rp2.  $N=1000$  and  $M=4$

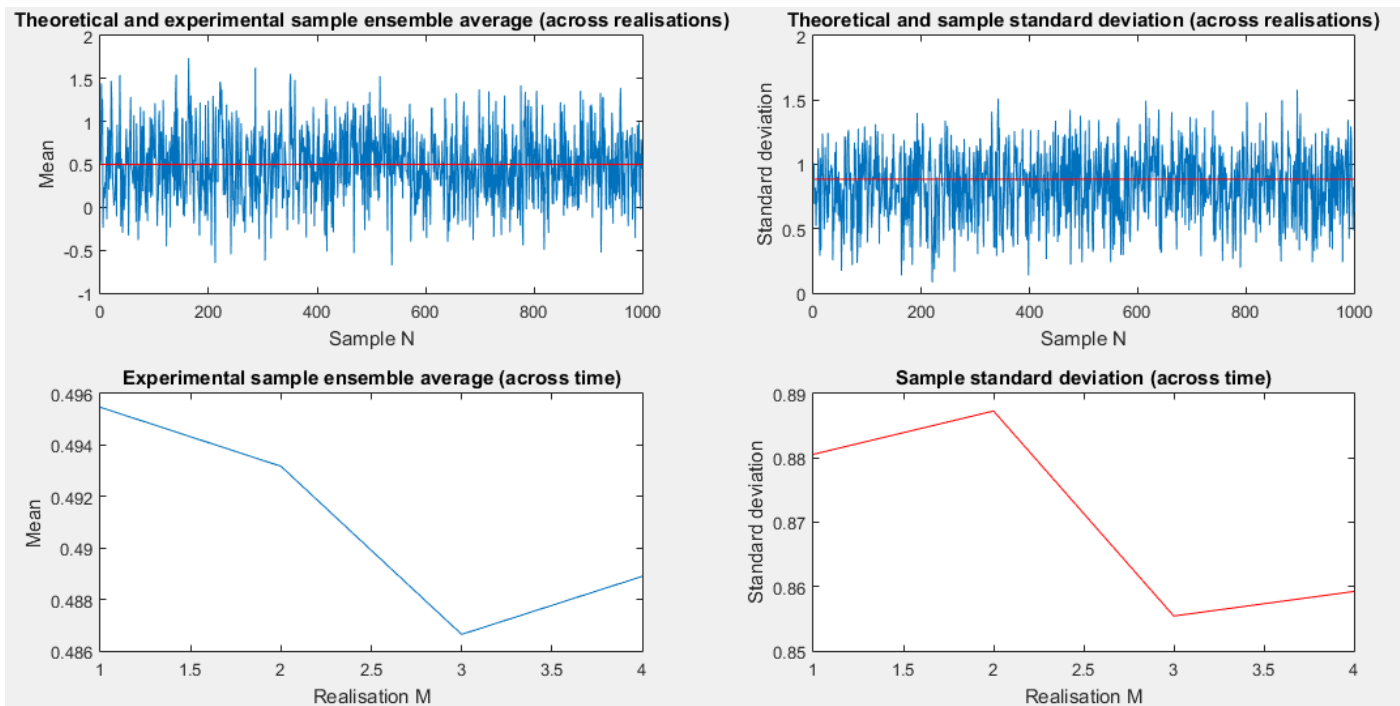


Figure 14: Sample ensemble average and sample standard deviation across realisations and time for rp3. N=1000 and M=4

The process rp3 has its time average of its 4 realisations which matches its expectation. The process rp3 is thus ergodic. On the other hand, rp1 and rp2 are not ergodic since their time average vary for each of their realisations.

### 1.3. Estimation of probability distributions

1) The function *pdf* is based on the MatLab *hist* command. It requires the process to be stationary and with time-invariant statistical properties. The function is defined as shown below.

```
function [PDF_sample, m] = pdf(X, BINS)
    if BINS == 0 %by default
        BINS = 50;
    end
    [PDF_sample, m]=hist(X,BINS);
end
```

The function returns the array *PDF\_sample* and the array of bins, *m*.

This output is then divided by the area (using *trapz* which requires *m*) to obtain the density function.

```
bar(m,PDF_sample/trapz(m,PDF_sample));
%The histogram is normalized by the integral of the PDF
%estimated by a trapezoidal numerical integration (trapz).
```

The function is tested with the stationary process `X=randn(1,N); %Gaussian distribution` where `N=10000; %number of samples` and `BINS=100; %frequency bins`. The following figure shows the theoretical and estimated PDFs of X.

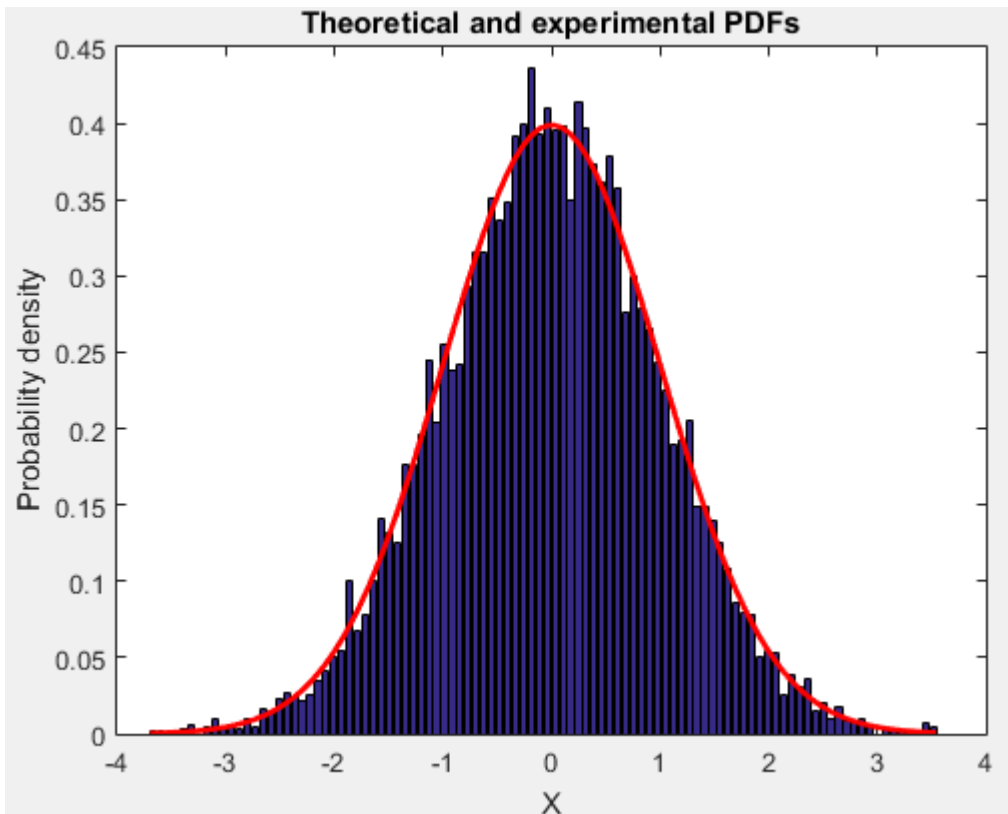


Figure 15: Theoretical and experimental PDFs for X, obtained with pdf.m, with  $N=10000$  and  $BINS=100$

2) From the results of part 1.2, only rp3 is ergodic and stationary. The following figure is the result of a MatLab script comparing rp3's theoretical and estimated PDFs for different samples  $N$  (100, 1000, 10000). As before, the red line represents the theoretical PDF of the process for its associated  $N$  value.

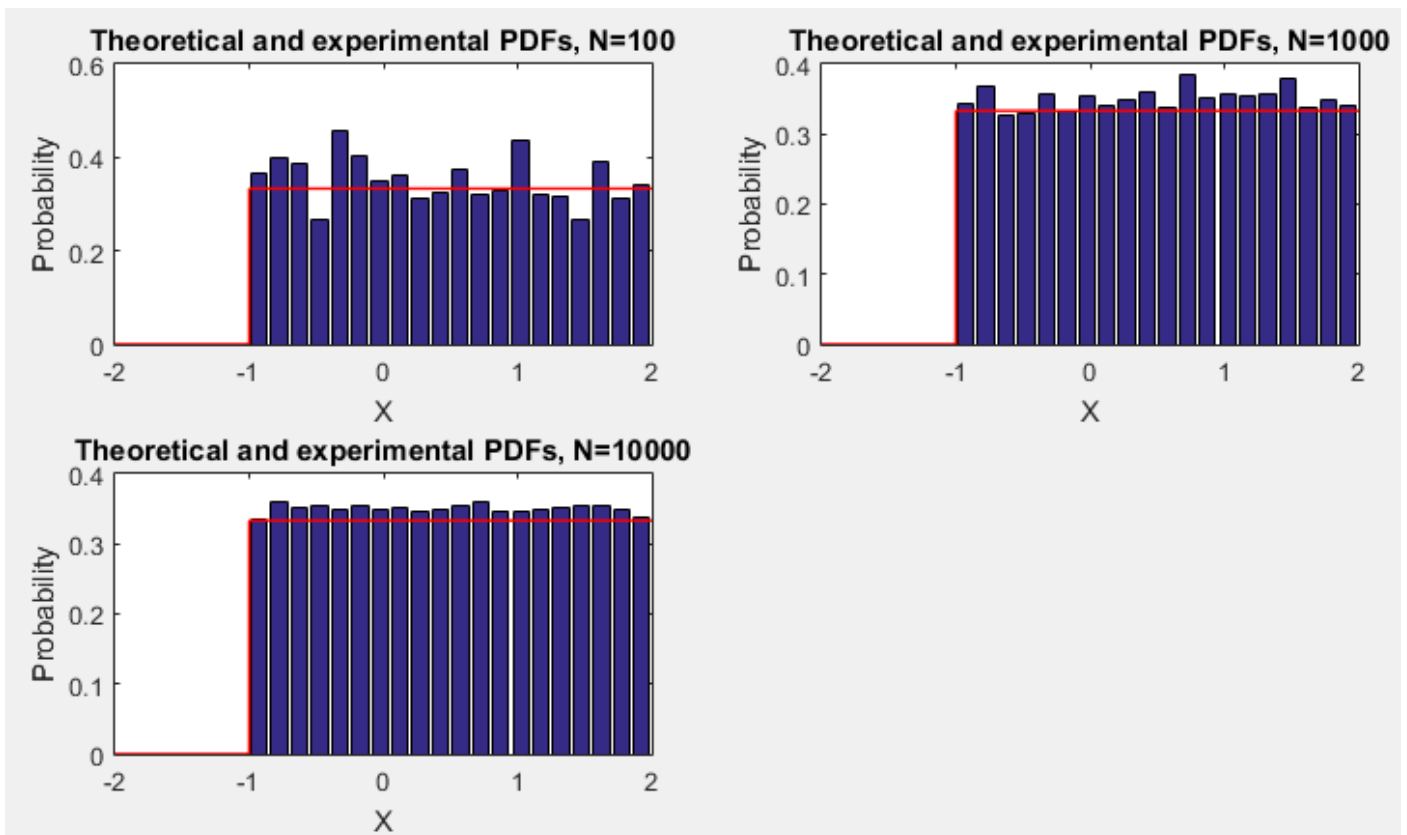


Figure 16: Theoretical and experimental PDFs for rp3, with  $N=100$ ,  $1000$  and  $10000$  ( $BINS=20$ )

As for the Gaussian distributed random variable, an increase in the number of samples makes the estimated PDF converging to the theoretical PDF (which is the PDF of  $x=3A+0.5$  where  $A$  is a uniform random variable with zero-mean). It can hence be deduced that the amplitude of **the PDF is 1/3, as** the amplitude of  $X$  varies between -1 and 2.

3. The *pdf* function cannot give an accurate estimates for non-stationary process. In the case of process  $rp3$ , the mean value increases with time. It would is impossible to bound the amplitude of the signal. The correct pdf could be estimated by computing the time average for each realizations and computing the pdf of the averaged data.

## 2. Linear stochastic modelling

The autocorrelation function (ACF) of a signal is given by

$$R_X(n, s) = \mathbb{E}\{X_n X_s\}.$$

where  $X_n$  and  $X_n$  are the realisations of a random variable at different times.

For ergodic processes, a time average estimates the ACF, with the following estimator:

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

### 2.1. ACF of uncorrelated sequences

1) A 1000 samples white Gaussian noise (WGN)  $X$  is generated using the command `X=randn(1,N)` ;

The autocorrelation of the signal  $X$  is then calculated with:

```
[values,range]=xcorr(X,'unbiased');
%range contains the values of t for which the
%autocorrelation has been calculated.
```

The results are then plotted and the following is obtained.

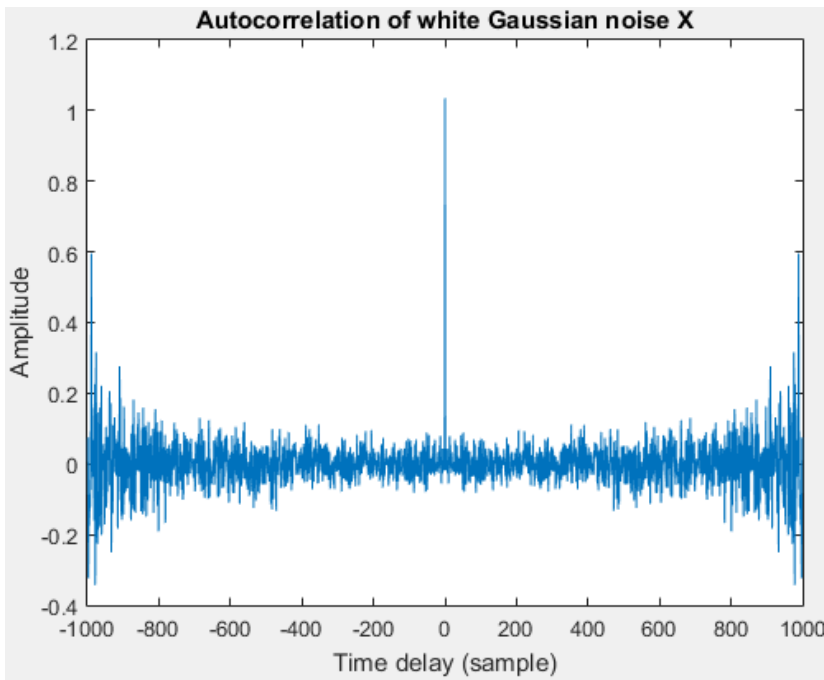


Figure 17: Autocorrelation function of white Gaussian noise (WGN) for delays in [-1000, 1000]

First of all, the plot from figure 17 describes a symmetry. This can be explained because points with equal distance from the zero time delay will have the same absolute delay and as a consequence also the same autocorrelation. This plot also describes a peak of amplitude for a zero time delay. With a time delay of zero, the signal is basically correlated with itself. This implies that the highest similarity is reached for a delay of zero.

2) 3) The following figure shows the ACF of the white Gaussian noise zoomed for delays smaller than 50.

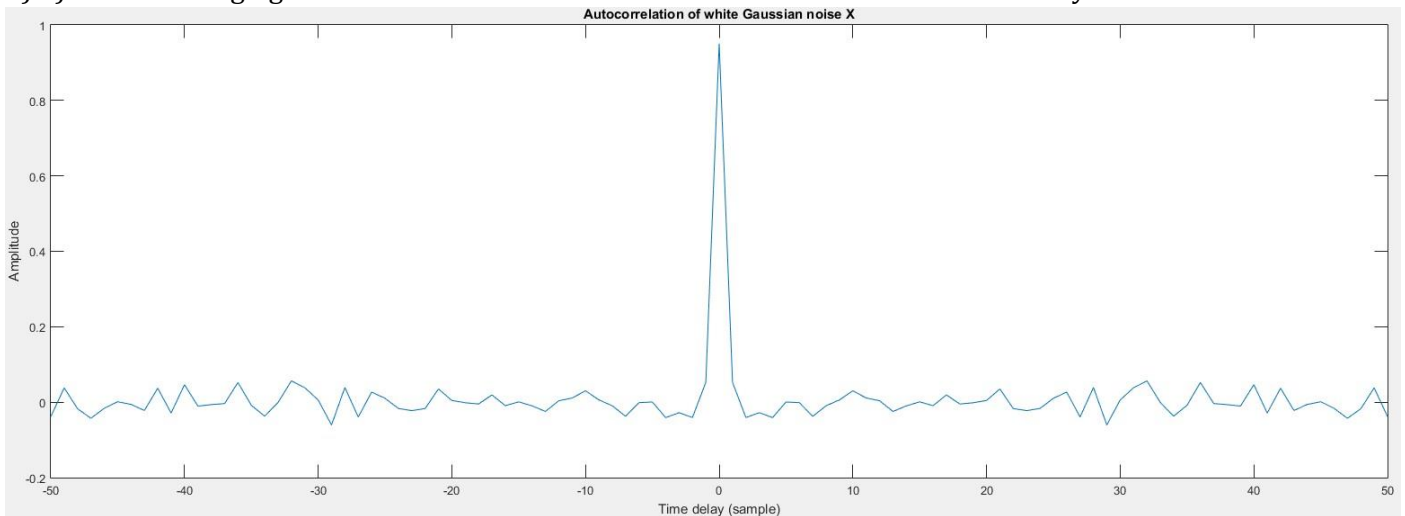


Figure 18: ACF of WGN for delays from -50 to 50.

The ACF is now apparent to the Dirac function. The spike in the amplitude which was found in figure 17 for large delays (up to 1000) is not present in this view. In fact, as the delay decreases, the samples available for comparison are more numerous. Samples cancelling each other in the averaging sum will be fewer, so the total sum will be different than zero. As a consequence, the estimator will not be accurate from a certain delay. A good empirical bound would be a delay of  $\frac{N_{samples}}{2}$ .

## 2.2. ACF of correlated sequences

1) The 1000-samples WGN vector  $x$  is now filtered by a moving average filter of order 9 with unit coefficients using the `filter` command. The following piece of code is used to do this filtering.

```
N=1000; %sample realisation of the WGN
order=9; %order of the moving average filter
X=randn(1,N); %white Gaussian noise
y=filter(ones(order,1),[1],X);
[values,range]=xcorr(y,'unbiased');
%range contains the values of t for which the
%autocorrelation has been calculated.
```

The following plots are also obtained (with `plot` and `stem`).

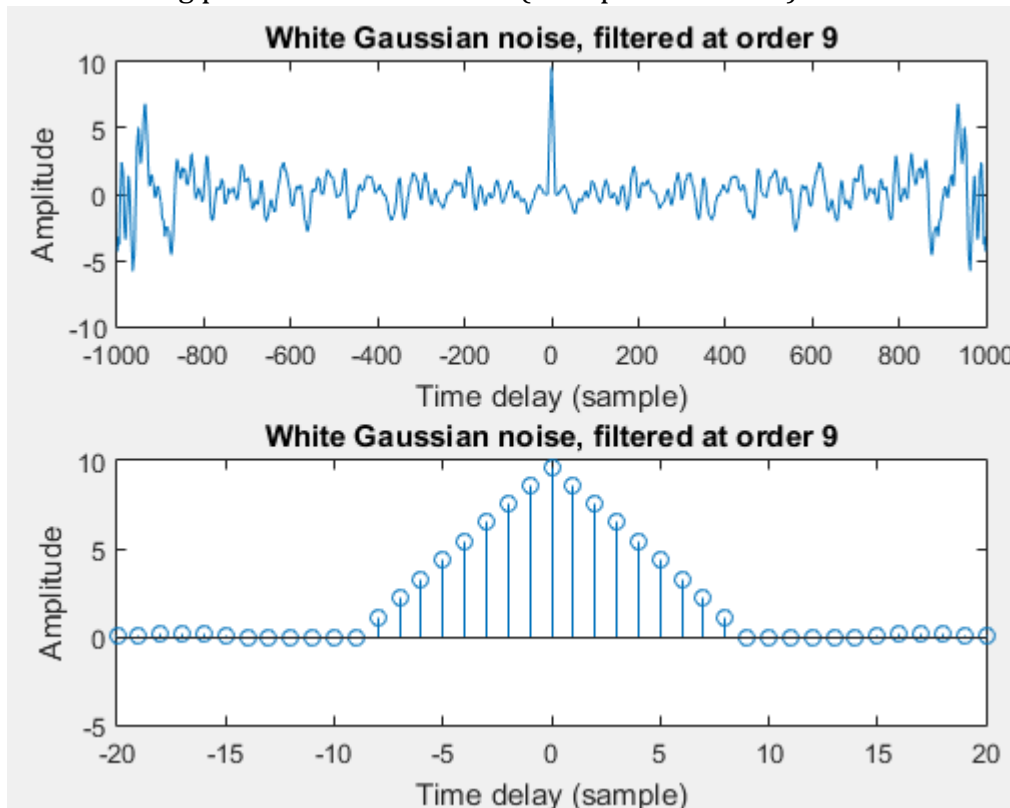


Figure 19: ACF of filtered WGN

The moving average filter acts as a low pass filter on the white Gaussian noise  $x$ . Averaging the shifted data before executing the correlation function produces a smoother signal. Because the samples are better correlated, the amplitude of the autocorrelation is higher for small delays and reaches zero for high delays. This can be explained because the high frequency variations for large delays are filtered out before computation. In addition, as the order of the filter increases, the triangular waveform of the ACF is more wide and visible.

2) The ACF of the filtered output is theoretically the convolution of the ACF of the input with the ACF of the impulse response. For an uncorrelated process  $X$ , its ACF is a Dirac function centred at zero. Now, the convolution of a zero-centred delta with a signal gives the signal itself. So we have:

$$R_y(\tau) = R_h(\tau)$$

The ACF of the impulse response of the moving average filter is a triangular waveform with base and amplitude equal to the order of the filter (as shown in figure 2 of figure 19, for order 9).

### 2.3. Cross correlation function

The cross correlation of two signals X and Y is theoretically:

$$R_{XY}(n, s) = \mathbb{E}\{X_n Y_s\}$$

This can be estimated through the following time average:

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

1) The signal X is the white Gaussian noise and the signal y is the filtered noise (as in 2.2). The figure below shows the cross correlation of the two signals for delays between -1000 and 1000, and between -20 and 20 using stem.

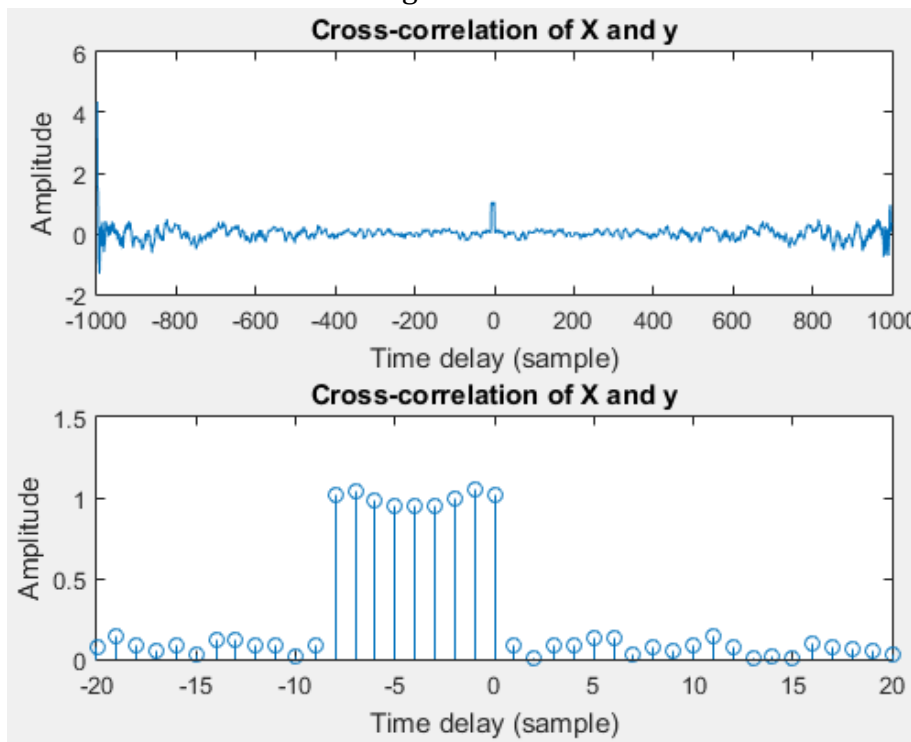


Figure 20: Cross correlation of X and y, for delays between -1000 and 1000

The cross correlation between the input and the output of a filter is given by,

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

Therefore if X is uncorrelated  $R_{XY}(\tau) = h$ .

2) Executing the cross correlation of the uncorrelated (process) signal, such as WGN, with the unknown impulse response of the system allows to identify the transfer function of the system. This result can therefore enable system identification.



### 2.3. Auto-regressive modelling

The autoregressive model used is generated with the following equation:

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

1)  $a_1$  and  $a_2$  are uniformly distributed random variables, where  $-2.5 \leq a_1 \leq 2.5$  and  $-1.5 \leq a_2 \leq 1.5$ . The aim is to determine stable and unstable values of  $a_1$  and  $a_2$  for which the filter will converge. Through MatLab,  $a_1$  is plotted in function of  $a_2$  and the color of the plotted points gives an indication if the pair of values is stable or unstable. The figure 21 below shows the triangle of stability, in which red dots indicates possible stable combinations of  $a_1$  and  $a_2$ .

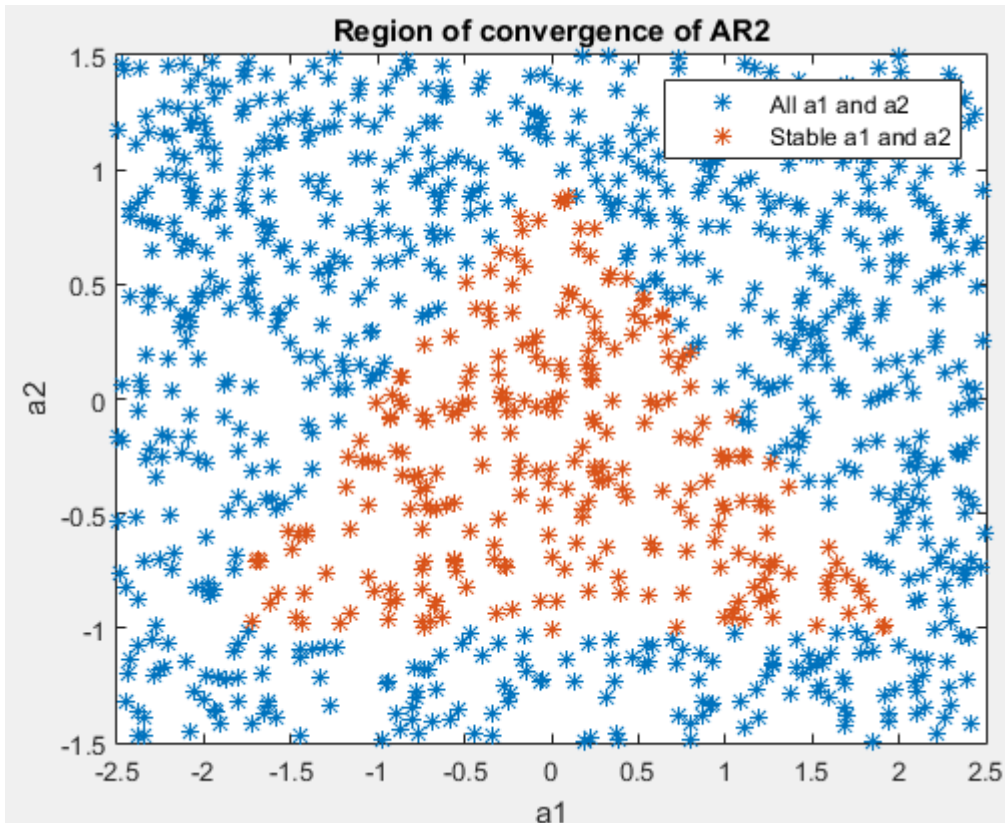


Figure 21: Region of convergence for AR

According to the Yule-Walker equations,

we have  $\rho_1 = a_1 + a_2\rho_1$

$\rho_2 = a_1\rho_1 + a_2$  where  $\rho_1$  and  $\rho_2$  are the correlation coefficients of order 1 and 2 and are both in the range  $[-1,1]$ .

The z-transform of the autoregressive model time equation gives the following condition:

$$z^2 - a_1z - a_2 = 0$$

For the filter to converge, the magnitude of  $z_1$  and  $z_2$  must be smaller than one.

In that case,  $-a_2 = z_1z_2$  and the magnitude of  $a_2$  is smaller than 1.

We then have,  $\rho_1 = \frac{a_1}{1-a_2}$  hence  $-1 < \frac{a_1}{1-a_2} < 1 \rightarrow a_2 - a_1 < 1$  &  $a_1 + a_2 < 1$ .

These two resulting equations combined with the restrictive ranges of  $a_1$  and  $a_2$  explains the triangular shape of the region of convergence.

2) The ACF of the sunspot series is plotted for  $N=5$ , 20 and 250 in the figure below.

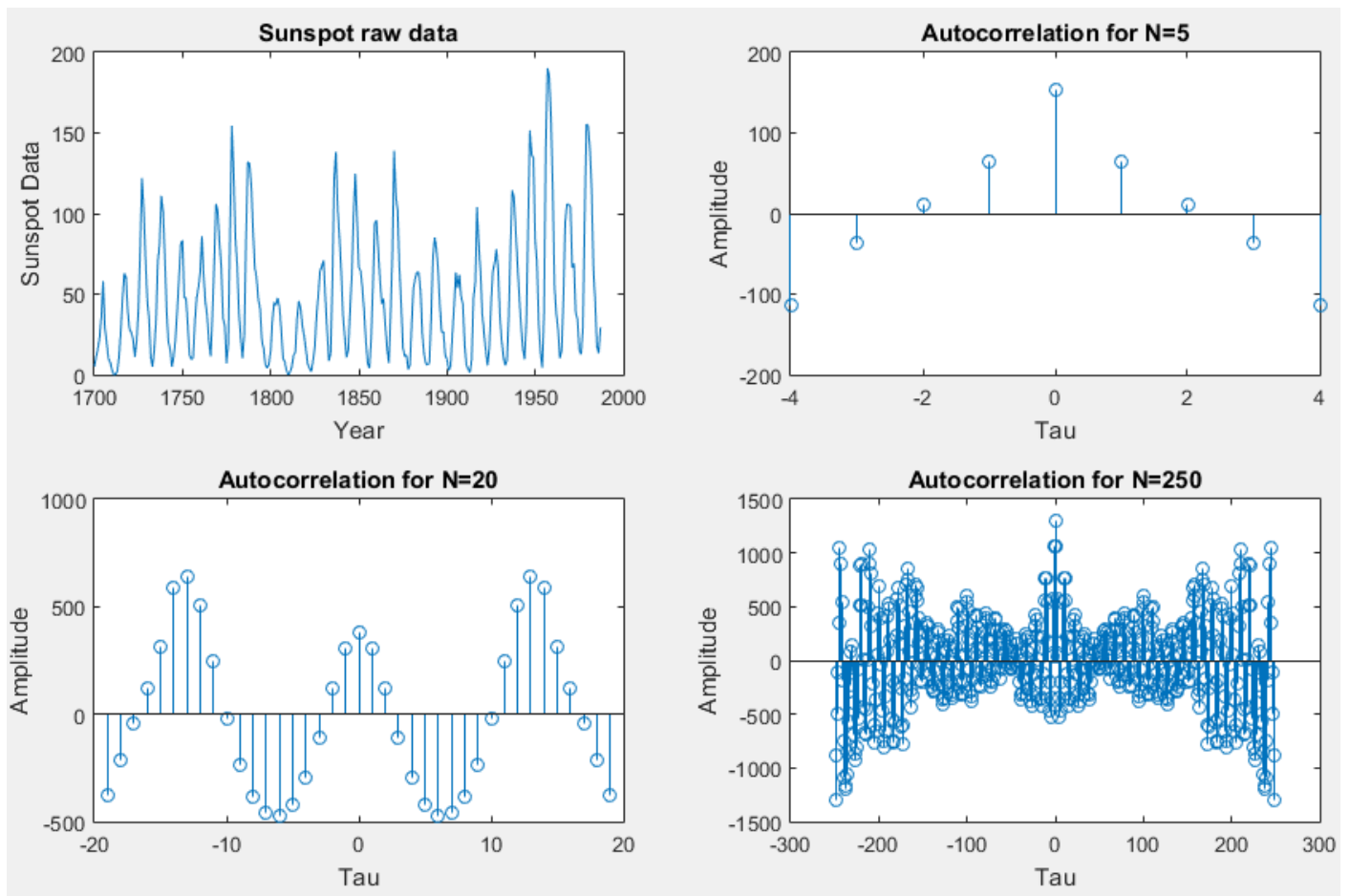


Figure 22: ACF of sunspot series for  $N=5$ , 20 and 250

For  $N=5$  samples, the repetitive patterns of the data can't be determined as there is not enough samples. The 20 samples ACF shows clearly the repetition of the data every 13 years. For  $N=250$ , the excess of samples introduce new undesired peaks and the data repetition pattern disappears. This is due to the delay being larger than  $N/2$ , implying the ACF estimator to be inaccurate.

When the ACF is computed for the data which undertook a mean subtraction, the peaks are no longer higher than 300. The global amplitude of the ACF decreases. Due to the DC, the data is more similar to its shifted versions.

3) To find out the most probable order of the data, the Yule Walker partial correlation coefficients are plotted on the same figure up to the 10<sup>th</sup> order.

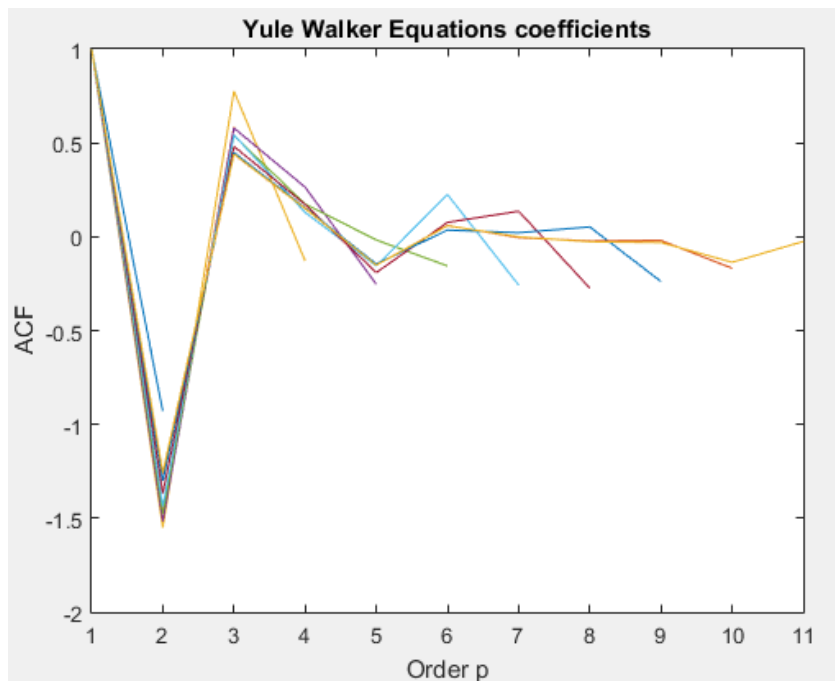


Figure 23: Yule-Walker coefficients, for order up to 10 (11).

This plot describes that the amplitude of the coefficients becomes smaller as the order increases. The order of 2 is also a threshold order for which the coefficients can be smaller. This second order is therefore the most likely order of a system.

In order to compute the zero mean unit variance data, the mean is subtracted to the data, which is then divided by the variance. The following line executes this:  $x = (x - \text{mean}(x)) / \text{std}(x)$  ;

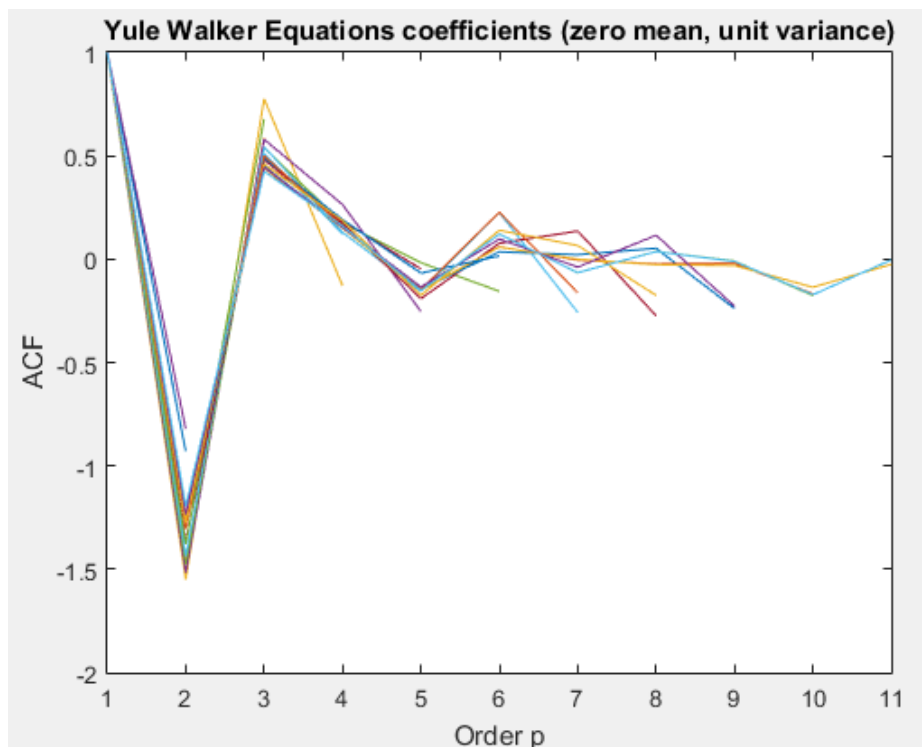


Figure 24: Yule-Walker coefficients, for order up to 10 (11) with zero mean, unit variance

In this case, the coefficients for orders higher than 2 become even more smaller, and the peak at order 2 increases. This confirms the result obtained with the original sunspot data.

4) The alternative ways to determine the maximum order of the coefficients are the Aikake Information Criteria (AIC) and the Minimum Description Length (MLD) criteria, given by:

$$\text{MDL} = \log E_p + \frac{p \log N}{N}$$

$$\text{AIC} = \log E_p + \frac{2p}{N},$$

Where  $E_p$  is the loss function,  $p$  is the model order and  $N$  is the number of estimated samples. The following figure 24 shows the error for each filter order, from 1 to 40. The order minimising the loss error is 8 for both MLD and AIC. On the other hand, it is important to note that the error has already been considerably reduced for order 2. A trade-off must be done between the accuracy and the computing complexity of the filter, hence making order 2 a generally best choice.

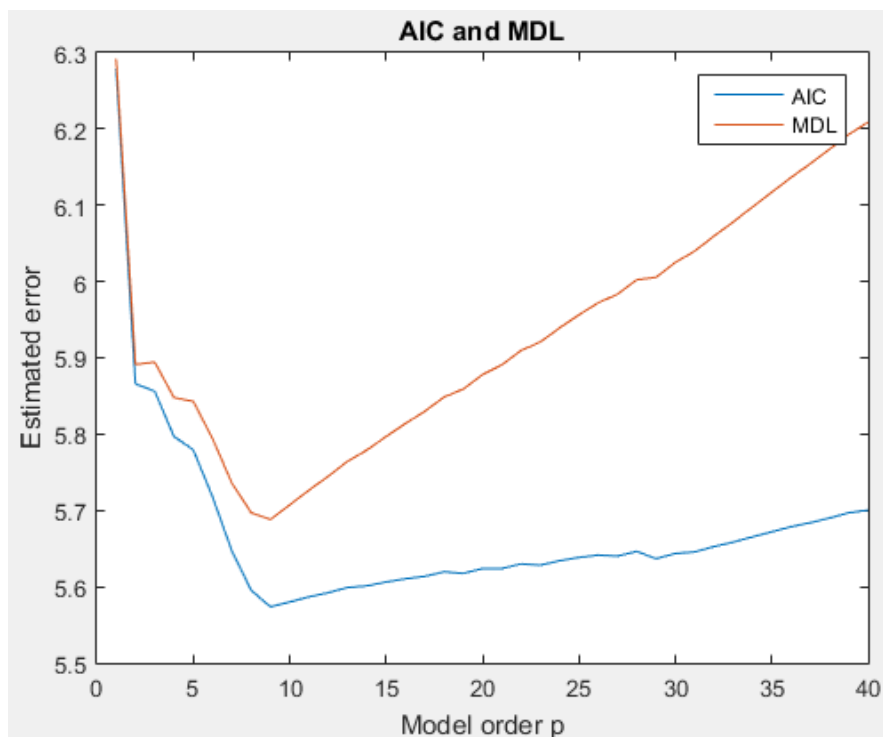


Figure 25: Estimated error by MLD and AIC, up to order 40

5) An auto-regressive (AR) model is used to predict the sunspot data with some steps ahead. The figure 26 below shows the predicted data for prediction horizons 1,2, 5 and 10 and different model orders.

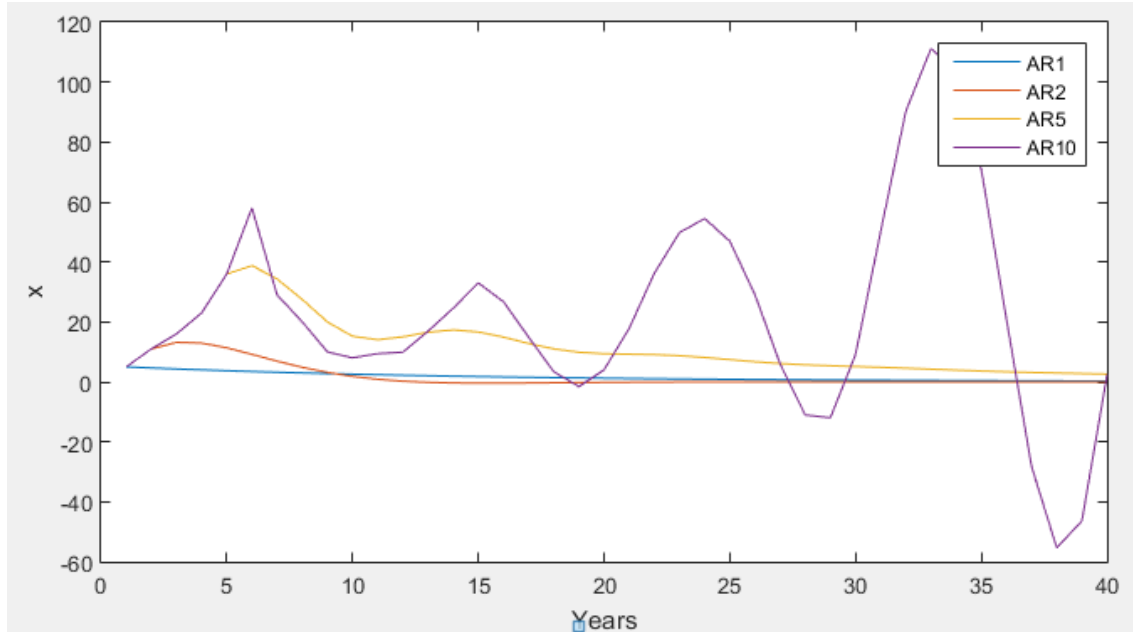


Figure 26: Predicted data over 40 years

This plot shows that the model quickly converges to zero if the order of the filter is not high enough. This is the case for the ar1 and ar2 process. The ar10 model oscillates longer but is out of phase with the data. If the order is very high, the model will overestimate small fluctuations due to noise. In the end, it seems that the model of order 5 is a good trade-off between accuracy and long-lasting predictions.

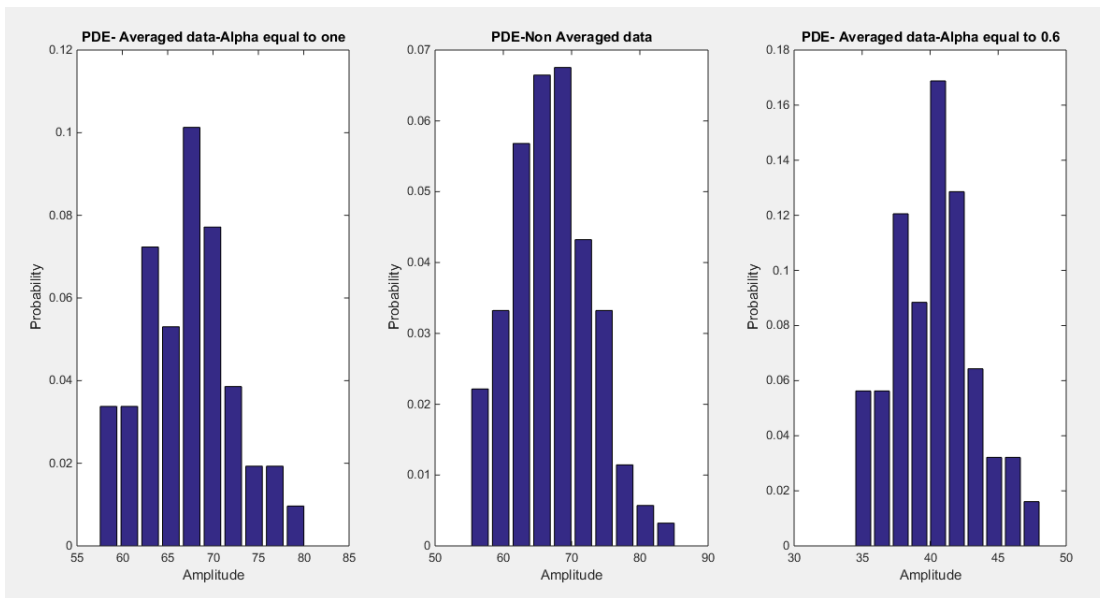
## 2.5. Real World Signal: ECG for iAmp experiment

The heart rate can be obtained using the following formula.

$$h[n] = \frac{60}{rr[n]}$$

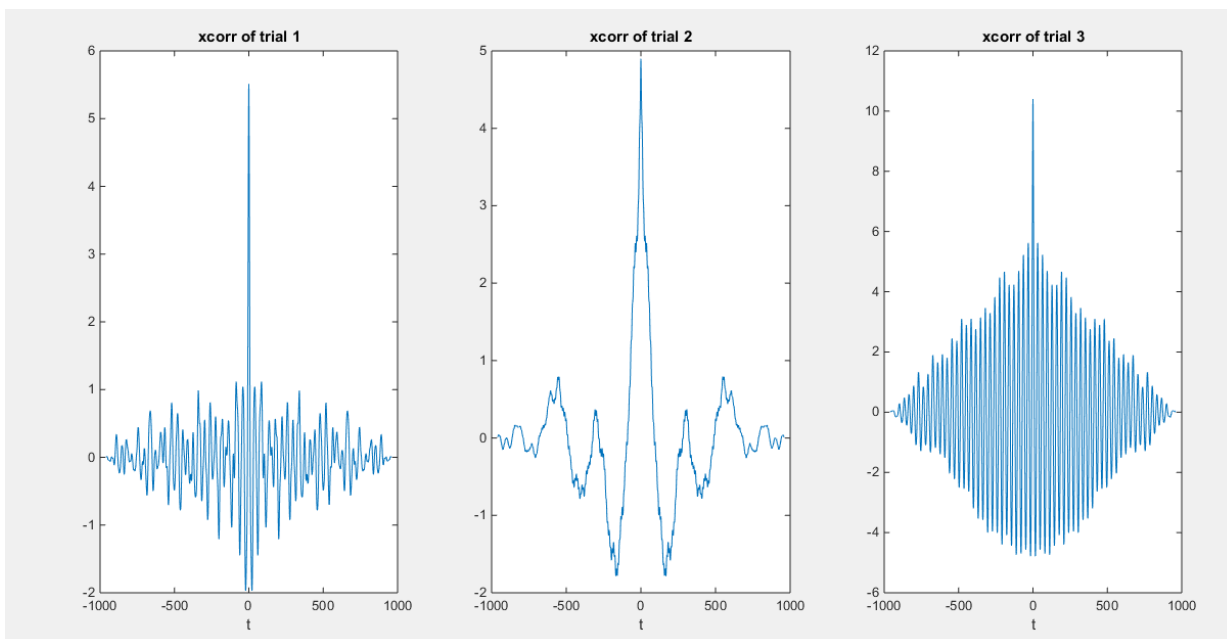
By averaging every 10 samples of the heart rate, a smoother estimate can be obtained.

a) b) The probability density estimate has been plotted for alpha equal to 1 and 0.6. Compared to the original, the averaged data is smoother. As we decrease alpha, the variation in the amplitude of the data becomes smaller.



**Figure 27: Probability Density estimate (PDE) of averaged and non-averaged data**

c) The ACF of the data presents sharp peaks, so the transfer function must present double poles and cannot be a moving average filter. It must then be an autoregressive filter.



**Figure 28: ACF for trials 1,2 and 3**

d) In this section, the coefficients  $Ar(p)$  of the autoregressive model characterising the ECG data are plotted.

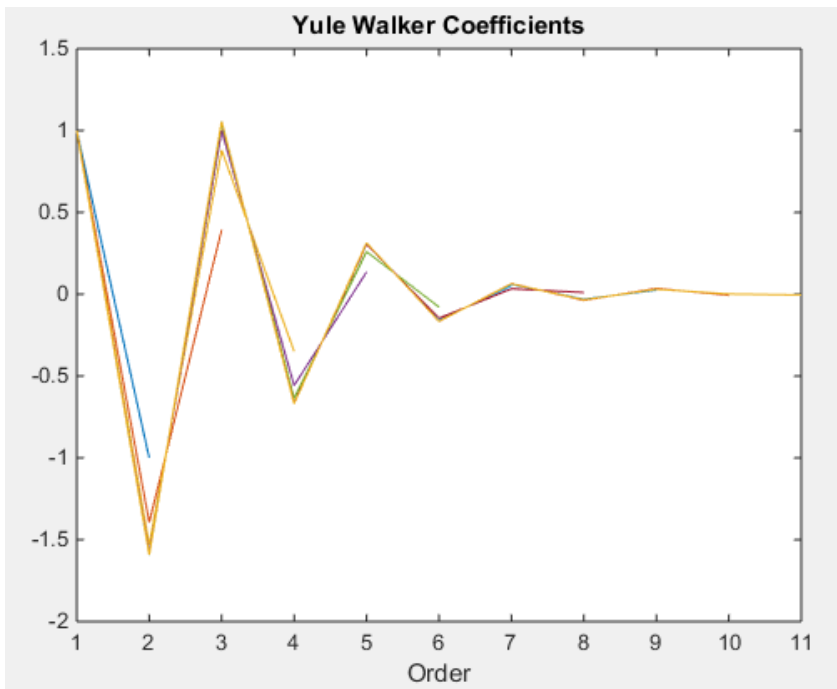


Figure 29: Yule Walker coefficients

The optimal order is again the second one, as coefficients for higher orders are smaller than 0.5.

### 3. Spectral estimation and modelling

The power spectral density (PSD) of a signal  $X$  is given theoretically by:

$$P_X(f) = \left| \sum_{\tau=-\infty}^{\infty} R_X(\tau) e^{-j2\pi f\tau} \right|, \quad f \in [0, 1]$$

The periodogram is an estimate of the PSD based on the Fast Fourier transform (FFT) determined by:

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

The function `pgm` takes as an input a signal  $x$  and outputs its periodogram  $Px$ . It is defined as follow.

```
function Px=pgm(x)
N=length(x);
e=zeros(1,N);
for n=0:N-1
    e=exp((-i*2*pi)*(n)).*(0:1:N-1)'./N';
    Px(n+1)=(abs(sum(x.*e))).^2./N;
end
end
```

The function `pgm` is tested with an  $N$ -sample realisation of WGN, where  $N=128, 256$  and  $512$ . Theoretically, the PSD of the signal is the Fourier transform of the ACF. Now, as explained before, since the ACF of the WGN is a Dirac function with an amplitude determined by the variance, its PSD should therefore be uniform with an amplitude given also by the variance. The figure below shows that the PSD becomes uniform as the number of samples increases, hence getting closer to the theoretical model.

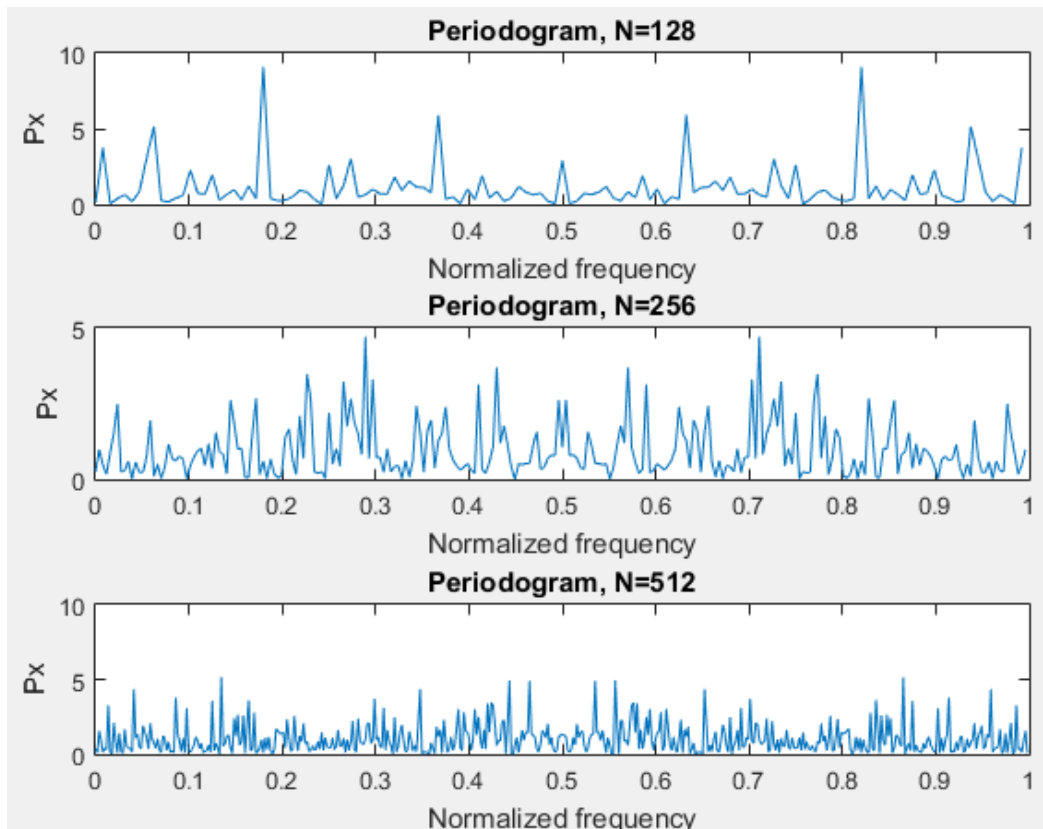


Figure 30: Periodogram of WGN for  $N=128, 256$  and  $512$



### 3.1. Averaged periodogram estimates

1) The PSD estimate is now filtered by an moving average filter (FIR filter, low pass). The following code is used:

```
px=pgm(x); %periodogram calculation  
px_smooth=filter(b,a,px);
```

where we have:

```
a=1;  
b=0.2*[1 1 1 1 1];
```

This gives the following periodograms for the different values of N previously used. This figure shows that the smoothness of the estimated PSD is improved.

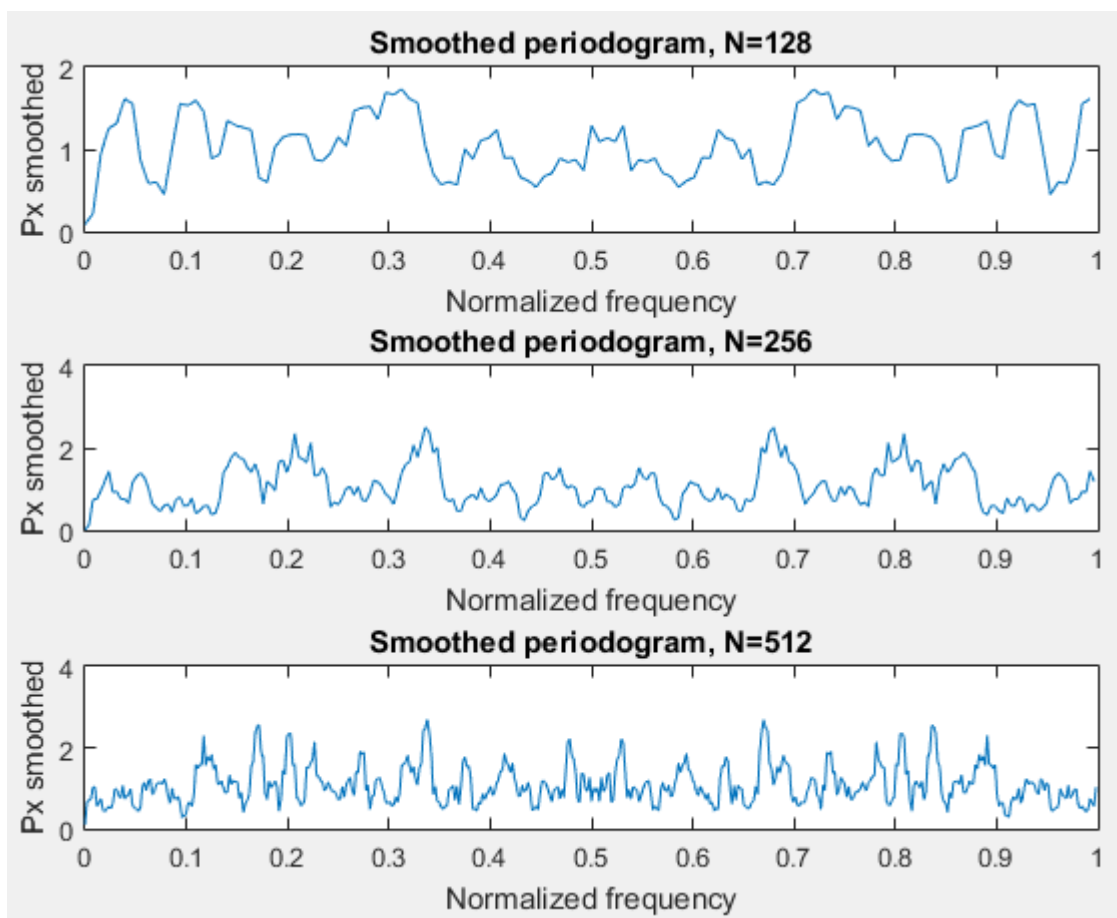


Figure 31: Filtered periodogram of WGN for N=128,256 and 512 (smoother)

2) 3) The white Gaussian noise MatLab vector  $x$  of 1024 samples is now split in 8 non-overlapping 128 samples segments. As this process is stationary, the statistical properties of each segment will be the same. The figure below shows the averaged periodogram obtained by computing the mean across the 8 different split segments. For this averaged estimated PSD, the amplitude variation is smaller and it is centred at 1 as expected.

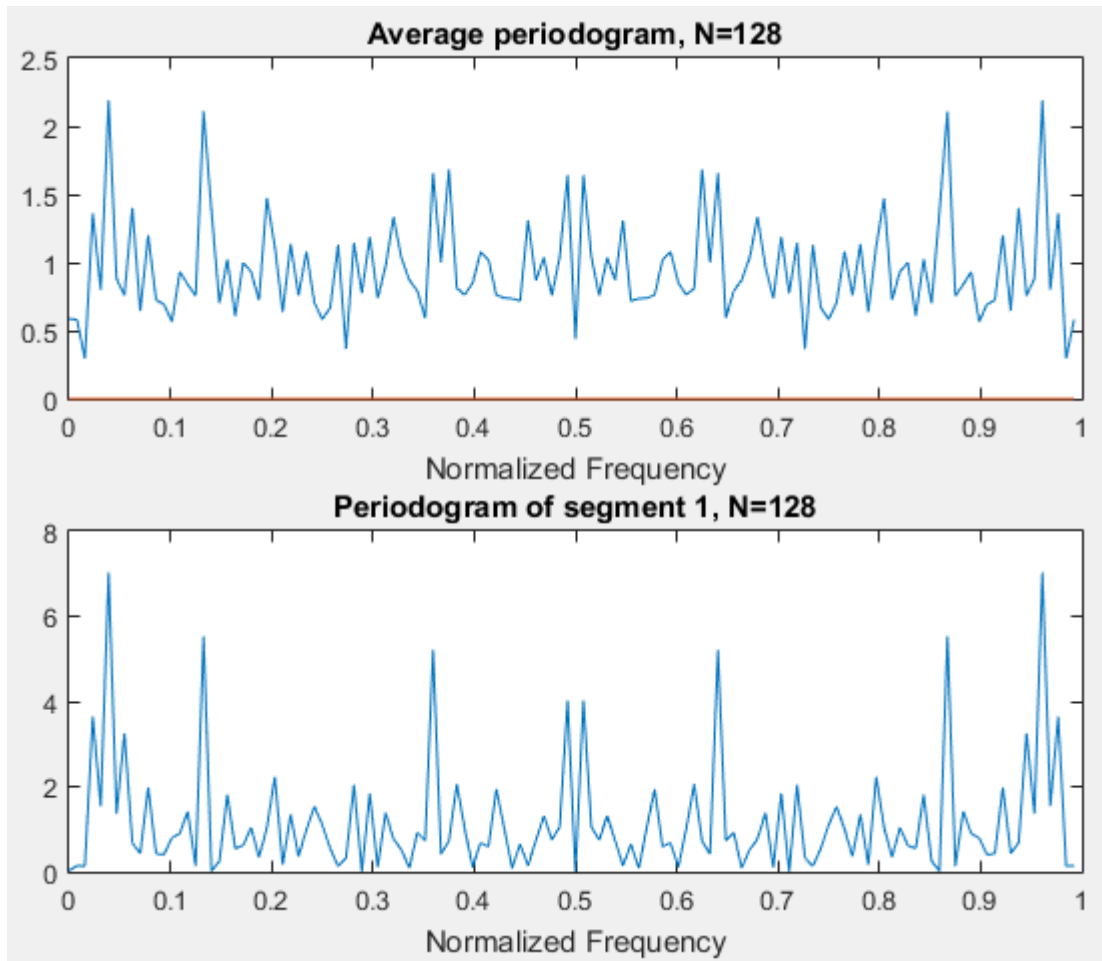


Figure 32: Average periodogram of WGN for N=1024 (split in 128 samples segments)

### 3.2. Spectrum of autoregressive process

The theoretical PSD of an autoregressive process  $Ar(p)$  with input  $X$  and output  $Y$  is:

$$P_Y(f) = \frac{\sigma_X^2}{|1 + \sum_{k=1}^p a_k e^{-j2\pi kf}|^2}$$

The autoregressive process used is of order one and has  $a_1 = 0.9$ .

The input signal considered is a white Gaussian noise (WGN) vector  $x$  of 1064 samples. The figure below shows the time sequences of the input  $x$  (WGN) and of the output  $y$  which is a filtered version of  $x$ . This plot confirms that the filter acts as a high pass filter on the input as it removes most of the variations and spikes.

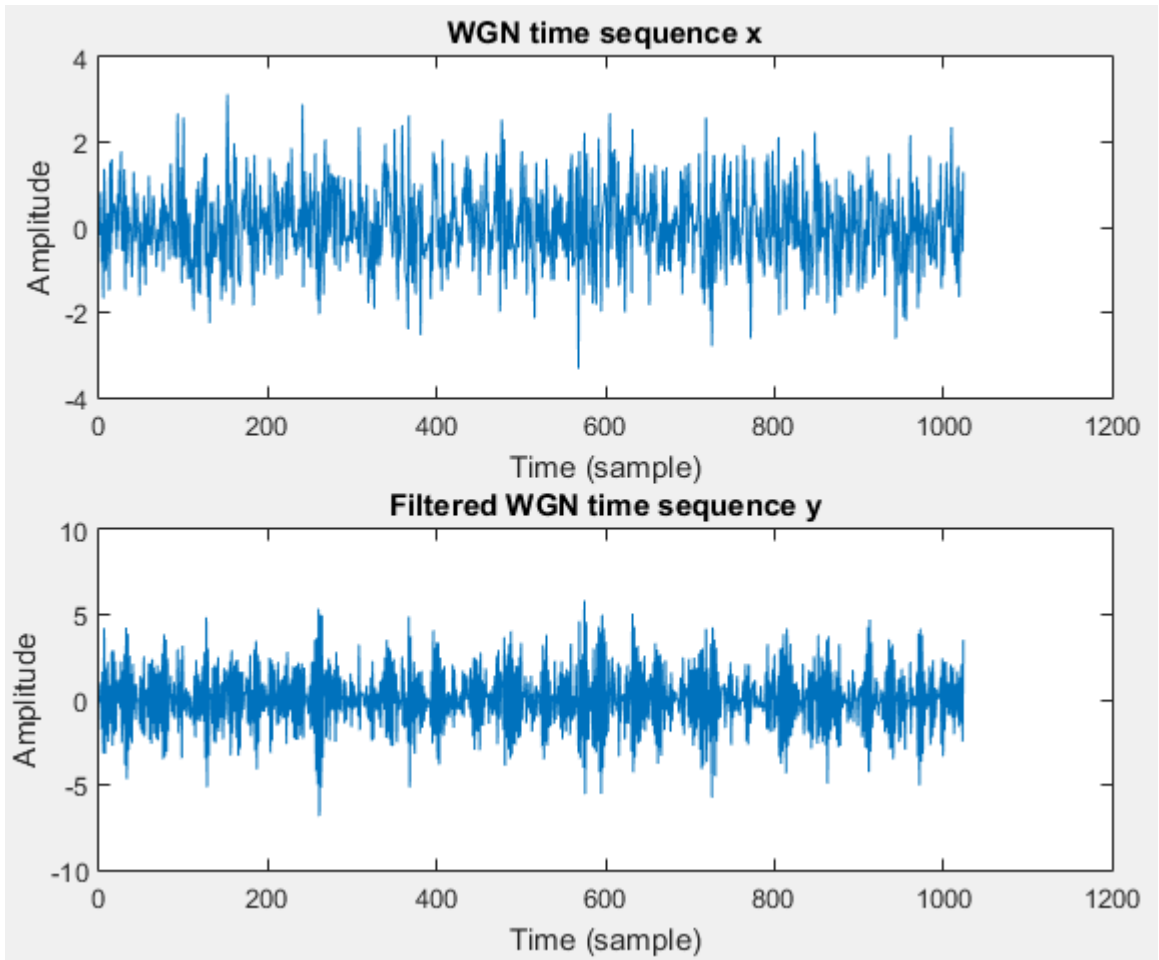


Figure 33: Average periodogram of WGN for  $N=1024$  (split in 128 samples segments)

1)2) To plot the theoretical PSD of  $y$ , the filter transfer function  $h$  is determined with the function *freqz*.

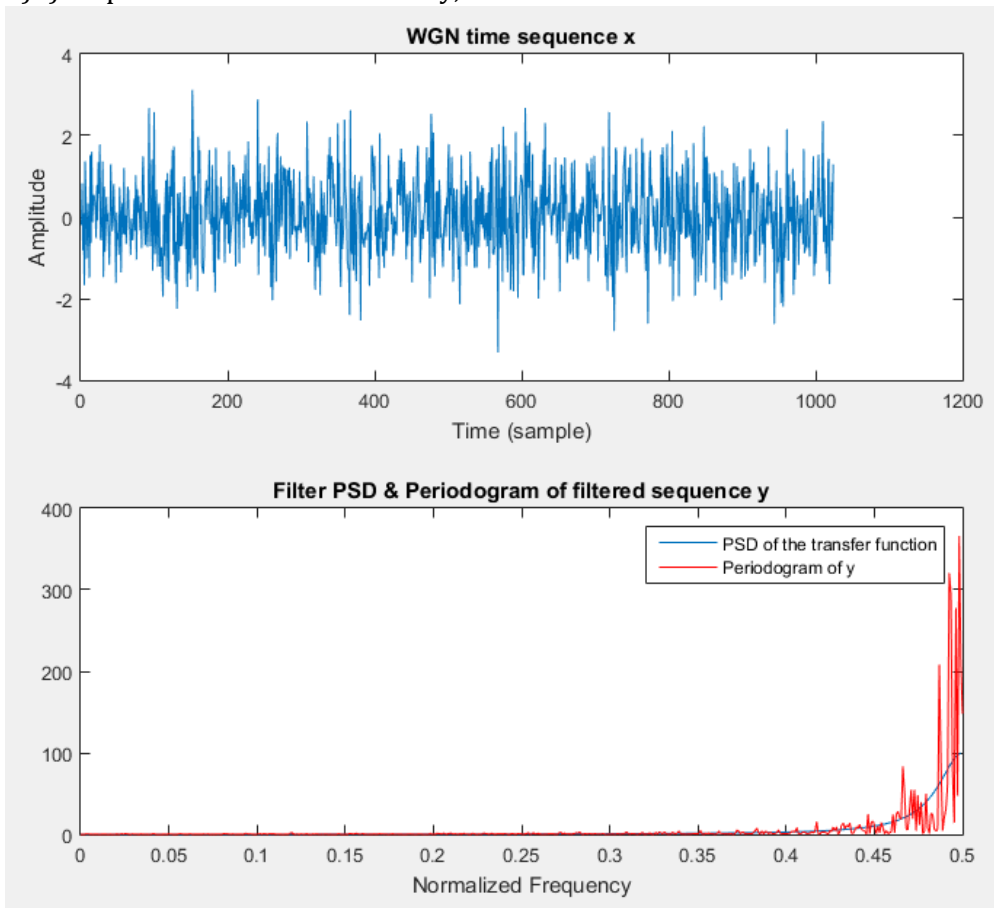


Figure 34: Filtered PSD and periodogram of filtered sequence  $y$

The cut-off frequency of the system is at 0.44 Hz normalized. For smaller frequencies, the power distribution equals to zero, hence proving the filter's high pass abilities. The periodogram is calculated with the function *pgm* and is shown in red on the same plot in Figure 34. Even if the periodogram filters out low frequencies, peaks are observed at low and high frequencies. This is because the periodogram is an estimate considering only a finite number of samples.

3) The figure 35 below shows the zooming in the region for *f* in the range [0.4,0.5]. Large peaks between 0.48 and 0.5 are observable. This is mostly due to the effects of windowing. In deeper details, when a finite number of samples of WGN is processed, this one is multiplied by a rectangular window. In the frequency domain, this is a convolution between the data and a sinc function, hence introducing peaks in a wider transition band.

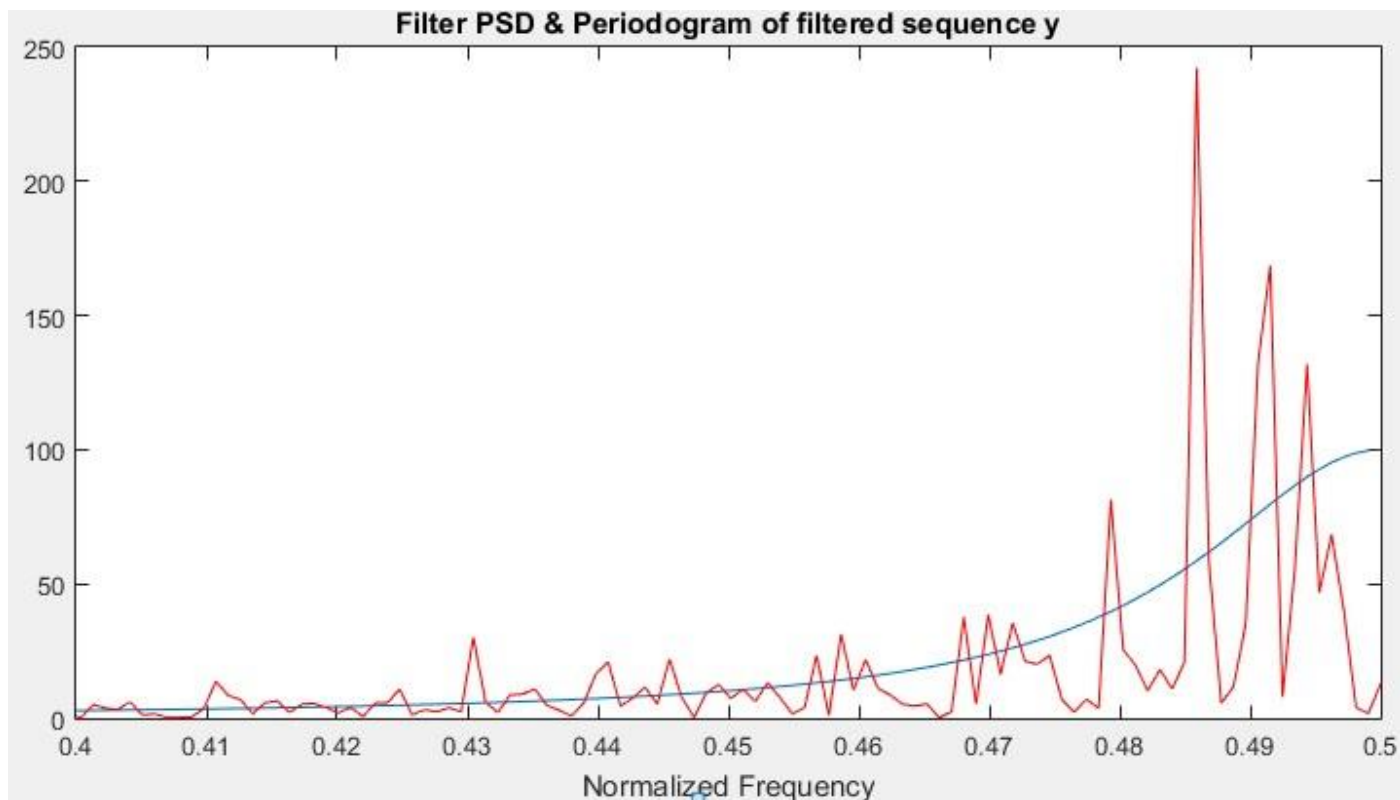


Figure 35: Zoom of figure 34 for *f* in the range [0.4, 0.5]

## Model Based PSD (MBPSD)

The MBPSD is an estimate based on the estimations of the parameters  $a_1$  and  $\sigma_x^2$  of the autoregressive process (ARP). These are calculated according to the estimated ACF of the filter output  $y$  with:

$$\begin{aligned}\hat{a}_1 &= -\hat{R}_Y(1)/\hat{R}_Y(0) \\ \hat{\sigma}_X^2 &= \hat{R}_Y(0) + \hat{a}_1\hat{R}_Y(1).\end{aligned}$$

4) The model based estimate is determined by the equation:

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

The following figure shows the MBPSD. The parameters previously discussed have been determined in the MatLab script using the `xcorr` command. The MBPSD was found by squaring the amplitude of the transfer function of the AR filter which is returned by the function `freqz`.

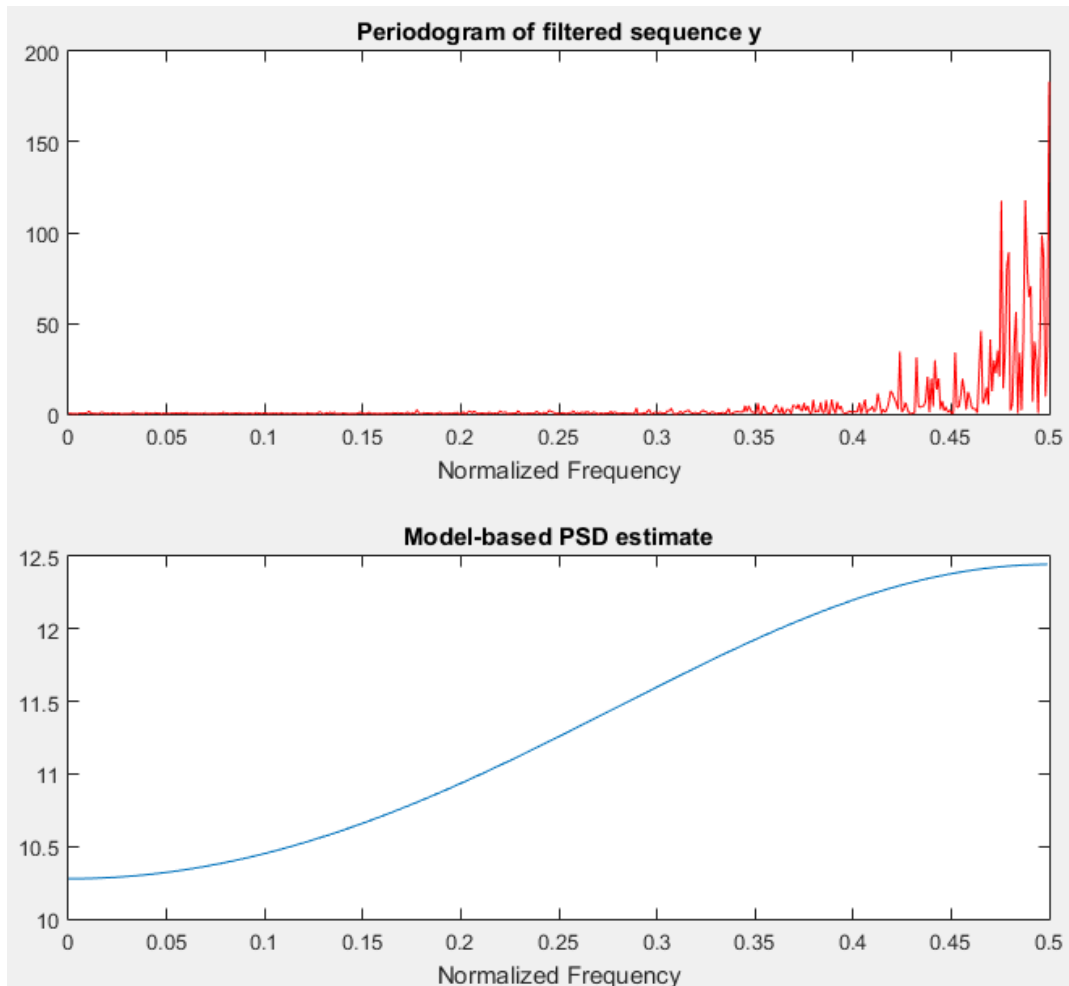


Figure 36: Model-based PSD estimate and periodogram

The MBPSD gives clearer results than the periodogram. It is smooth and its shape corresponds to the theoretical PSD. On the other hand, its amplitude is very small relatively to the periodogram. Indeed, the amplitude of the MBPSD is dependent on the variance, which can vary a lot, depending on the data. We can also deduce that an increase in the size of the data will increase the magnitude of the PSD.

5) We now set the sunspot series data as the input to the filter. This data is an AR(2) process. The figure 37 below shows the periodograms and the MBPSDs for the original and the centred data.

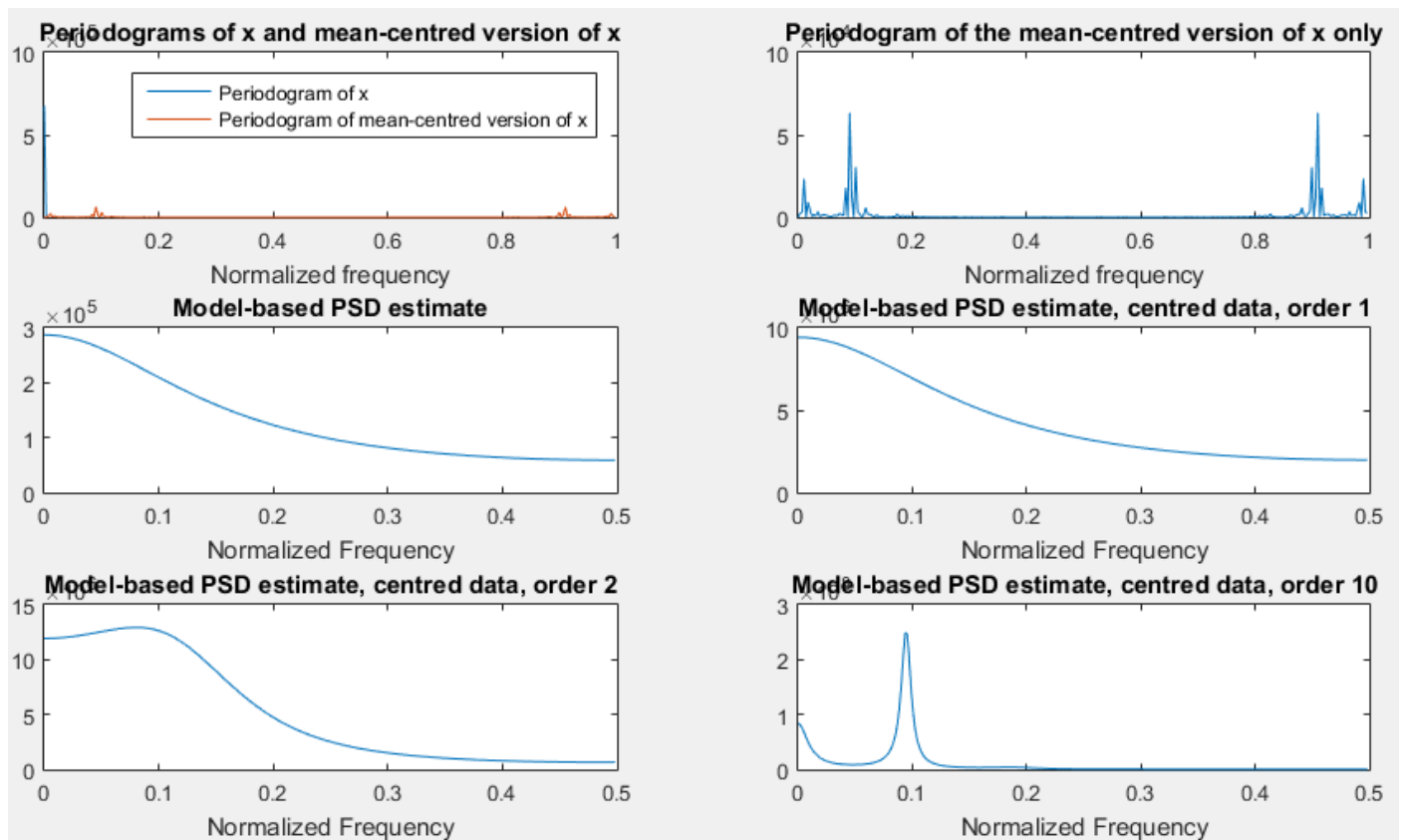


Figure 37: Model-based PSD estimates and periodograms for sunspot data

If the data is centred at zero, the amplitude of the PSD increases. The variance increases when the mean goes to zero, and hence increases the MBPSD. In more details, because  $\sigma_x^2 = E(x^2) - E(x)^2$ , if  $E(x) = 0$ , the variance will increase. The periodogram of the original data shows a huge peak of  $7 \times 10^5$  around DC. When the mean is set to zero, the peak disappears from the periodogram but remains in the MBPSD. This is because the periodogram is highly connected to the sample data whilst the MBPSD is more distant reflection of the real data.

The MBPSD is computed for order 1, 2 and 10. The estimate for order 10 is not the theoretical model at all. This is due to over-modelling, which basically means that the filter is 'ultra-sensitive' and exaggerates small fluctuations due to noise, as described before.

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

In a dual tone multi-frequency (DTMF) system, each keypad is associated to two frequencies  $f_1$  and  $f_2$  which determines the frequency of the keypad tone.

$$y[n] = \sin(2\pi f_1 n) + \sin(2\pi f_2 n),$$

The values for  $f_1$  and  $f_2$  are attributed according to the following table:

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

Table 1: Dial pad frequencies

1) A MatLab script has been coded to generate a random London landline number (020XXXXXXXXXX), which is then stored in the signal  $x$  (vector). The assumption is that a digit is pressed during 0.25 second and is released (idle) for 0.25 second as well. The program produces a sample of  $y$  at the sampling frequency of 32168 Hz. This value is appropriate as it is higher than twice the maximum value that  $f_1$  or  $f_2$  can take (1477 Hz). The following figure 38 shows the signal  $y$  for 2 random digits.

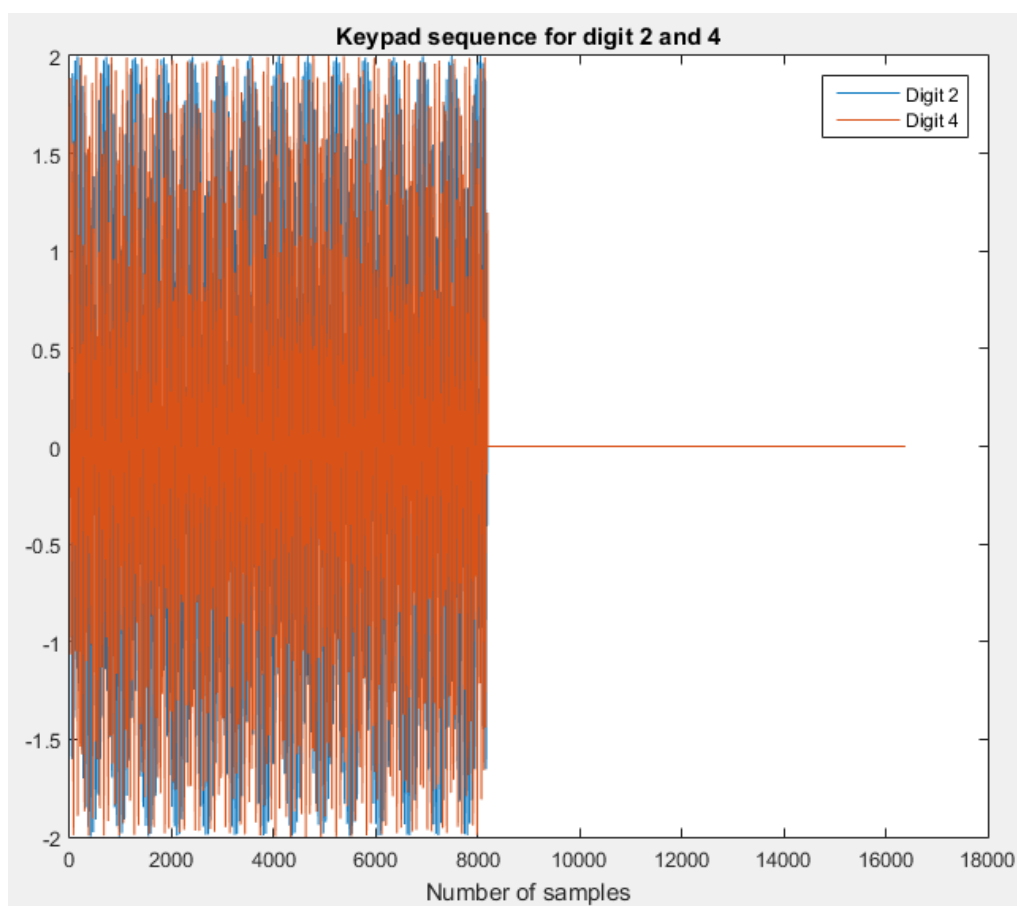


Figure 38: Signal  $y$  for 2 random digits (2 and 4 here)

2) The function spectrogram is used to analyse the spectral component of  $y$ . A Hanning window with a length of 0.25 second is chosen (8192 samples). This will prevent the windows from overlapping, since  $y$  is discontinuous.

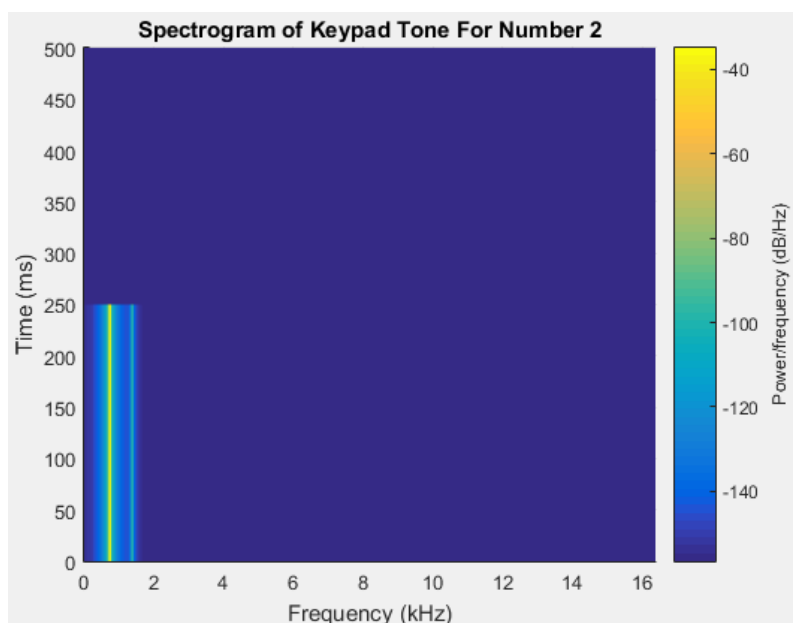


Figure 39: Spectrogram of digit 2

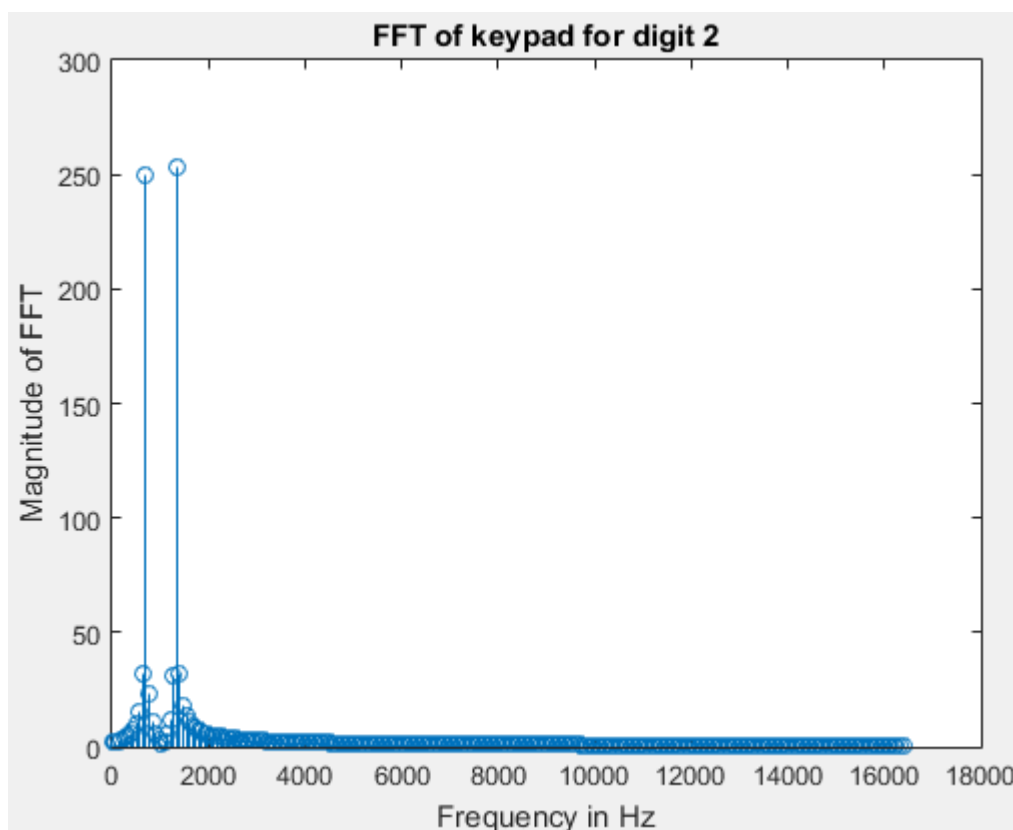


Figure 40: FFT of digit 2

3) If the plots on figure 38 (red and blue) were separate, it would be easy to recognize/learn which sequence corresponds to which keypad. Moreover, several tones can be represented on the spectrogram, so each would be easily identified.



4) The previous experiments are now repeated with white Gaussian noise added to the keypad tone, hence corrupting the signal. The following expression can be used here:

$$y[n] = \sin(2\pi f_1 n) + \sin(2\pi f_2 n) + w(n)$$

The variance of the noise will vary between 0.1 and 2, by multiplying the generated WGN vector by the constant variance. The following figures illustrate the results.

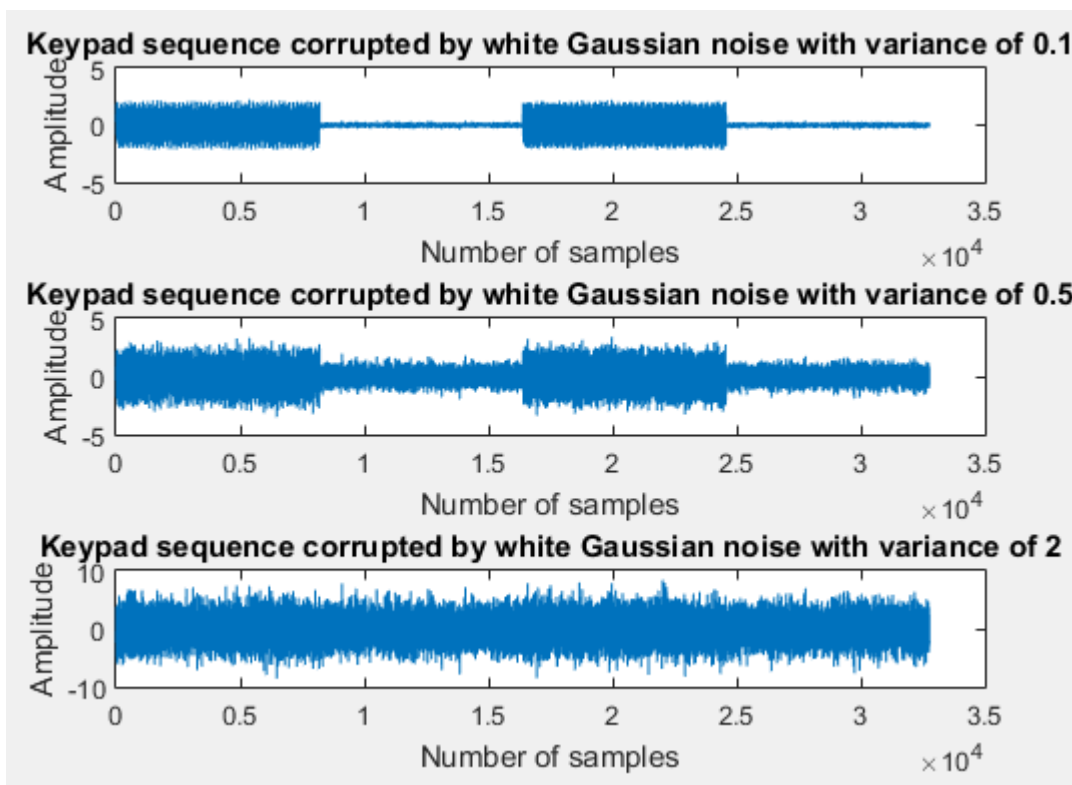


Figure 41: Keypad sequence (y) corrupted with WGN with different variances

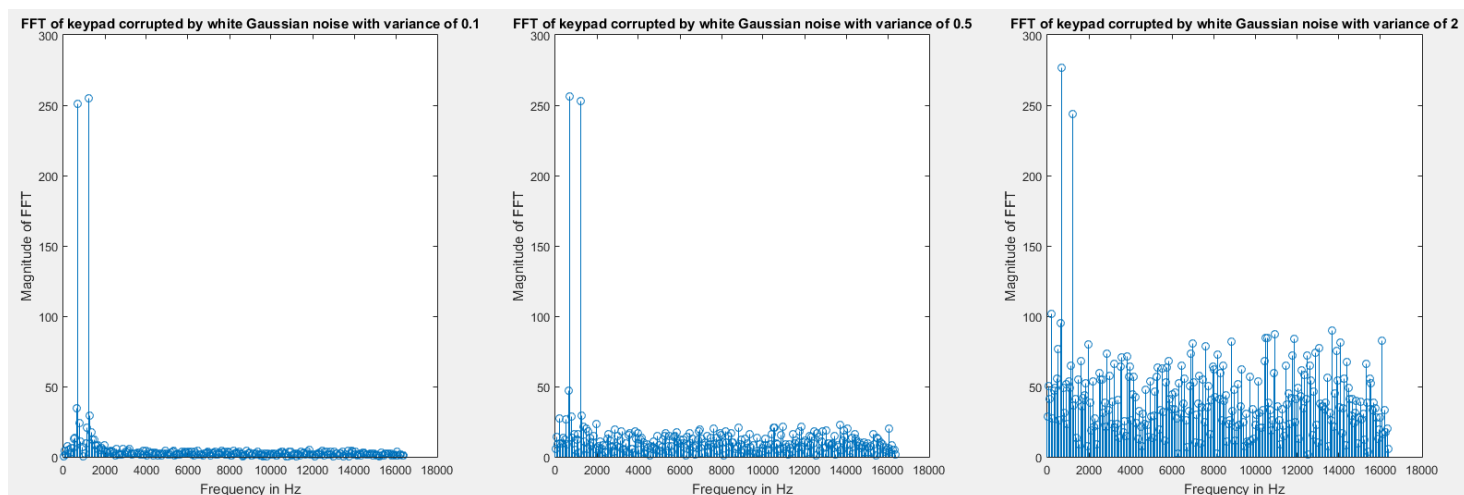
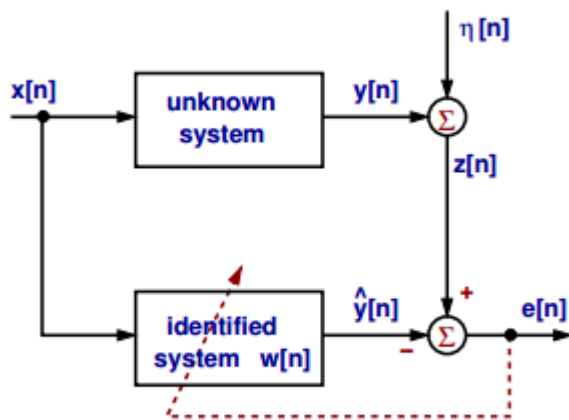


Figure 42: FFT of Keypad sequence (y) corrupted with WGN with different variances

For a variance of 0.1, it is still possible to identify the two keys and the silent tone. As the variance increases, it becomes impossible to detect the silent tone.

## 4. Optimal Filtering – Fixed and adaptive

In this section, the process of adaptive filtering is introduced. The parameters of the filters are triggered in function of the error  $e$ .



The optimum Wiener filter solution for the coefficient  $w$  is determined by minimizing the average squared error  $e[n]$ ,

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

### 4.1 Wiener Filter

1) The coefficients for the Wiener filter are determined using by calculating the autocorrelation  $R_{xx}$  and cross correlation  $p_{zx}$  between  $z$  and  $x$ . The following values are found for the coefficients  $w_{opt}$ , 1.0844, 1.9760, 2.9463, 2.0304 and 0.9945. These are similar to the ones of the unknown system.

2) We repeat the experiment by varying the value of the variance, we obtain the following results:

Variance	SNR dB	W1	W2	W3	W4	W5
0.5	20	1.0334	1.9900	2.9943	2.0030	0.9923
1.1	18	1.0322	1.9960	2.9925	2.0010	1.0032
1	17	1.0844	1.9900	2.9463	2.0304	0.9945
2.3	12	1.0822	1.9809	2.9233	2.0505	0.9098
5	5	1.0913	1.9800	2.9024	1.9705	0.90086
10	-5	1.0944	1.9760	2.9031	1.9605	1.0345

As the noise increase, the value of the coefficients fluctuate from the real coefficients.

3) The multiplication and additions due to the correlation functions of the routine have complexity  $O(n^2)$ .

## 4.2 Least Mean Square algorithm

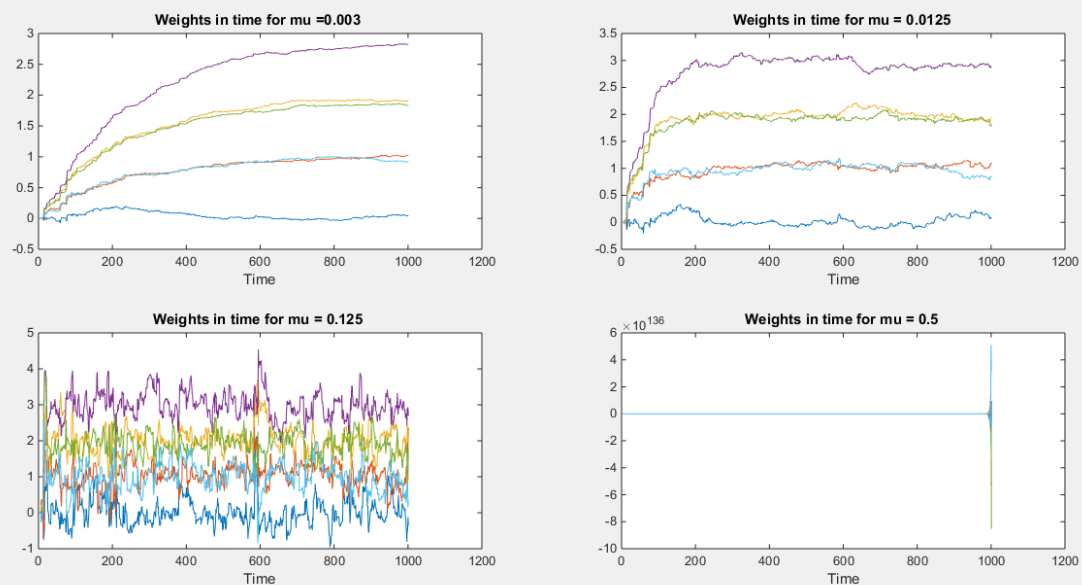


Figure 43: Evolution of the coefficients with  $\mu = 0.03, 0.0125, 0.125$  and  $0.5$

If  $\mu$  is too small then the convergence towards the final value will be slow. Increasing the value of  $\mu$  beyond 0.2 will accelerate but destabilize the response.