Master International E3A

PRACTICAL REPORT

FOR

STATISTICAL SIGNAL PROCESSING LABORATORY

PRATICALS

BY:

ABODE DANIEL

SUBMITTED TO:

PROF. ALEXANDRE RENAUX

DATE: 19/05/2020

ABSTRACT

The purpose of the laboratory practical was to gain knowledge on the basics of MATLAB and its application to Statistical Signal Processing. The experiment was divided into two parts; the first part was on basics of MATLAB and its application to signal processing problems, the second part was on Speech processing using MATLAB. In the first chapter, after getting acquainted with basic matlab programming concepts, we proceeded to generating noise using the rand() and randn() functions. The concept of linear estimation was modelled to estimate the message in a signal corrupted by noise and the designed estimators were compared and justified to satisfy the Cramer-Rao Lower Bound theorem. In addition, a non linear estimation techniques; Least-Square criterion was developed to estimate the variables in a sum of sinusoidal signal corrupted by noise. The second chapter covered parameter estimation of an AR process. A sound signal was analyzed and a sample of the sound signal was synthesized and compared with the recorded sound signal.

Contents

Chapter 1: Matlab 101 and Basic Signal Processing

1.1 Introduction

MATLAB is an integrated development environment that combines the use of toolboxes and a programming language to facilitate computational analysis. This chapter taught the basic of matlab programming language and its application to signal processing. The task covered involved generation of noise, study of the central limit theorem, linear estimation, and non-linear estimation.

1.2 Objectives

The objectives of this experiment is as follows;

- To learn the basic of matlab programming language
- To generate noise for signal processing using matlab programming language
- To implement the Central Limit Theorem
- To implement a Linear Estimator
- To implement a non-linear Estimator

1.3 Practical Procedures

1.3.1 Practical Procedure on Basics of Matlab

a) Defining a variable "a" and assigning value "5" to it > a =5;

b) Vector Definition

- Defining a vector "t", as a column vector and as a row vector

t = [2;6;-5]; %%column vector declaration

t = [4 7 -1]; %%row vector declaration

- Finding the transpose of a vector "v" > v'
- Finding the modulus of a vector v > abs(v)
- Finding the square of each elements of a vector "x" > x.^2
- Adding the value "-5" to a vector t = [4 7 -1] > t = [t -5];
- finding the length of a vector > length(t)

c) Drawing of Plots

- Plotting a vector “t” as a function of vector “x” > plot(t,x)
- Use command “hold on” to plot several curves on the same plot
- Use command “Subplot” to create axes in tiled positions

d) Use “for loops”

1.3.2 Application to Signal Processing Problems

A) Noises

- Use command “rand” to create a vector of random variables of size 10 and plot the vector.
- Plot an approximate probability distribution function of the random variable using command “hist”.
- Increase the size of the vector to 100 and plot its approximate probability distribution function.
- Use command “mean()” and “var()” to find the mean and variance
- Repeat above steps but using command “randn” instead of “rand”

b) Central limit theorem

- Generate several values of the random variable $Sj = \sum_{I=1}^{w} n\,i$ (ni is uniform from [-1,1]), j from 1 to 1000
- Plot the probability density function of Sj for different values of N

Observation: When N becomes large, more of the values of S align in the center of the pdf

C) Linear estimation

Observation model: $y\,(t) = m + n(t), \qquad t = 1,\ldots,N,$

- We want to estimate m, n(t) is a Gaussian random process with mean = 0 and variance = σ^2.
- m = 5 was chosen and vector y $= [\; y\,(1) \quad y\,(2) \quad \cdots \quad y\,(N) \;]^T$ was generated for variance, var = 1

- Vector y was plotted with different values of var;

```
plot(y)
```

- A maximum likelihood estimator $\hat{m}_{ML} = \frac{1}{N}\sum_{t=1}^{N} y(t)$ was modelled

```
est_m = (1/N)*sum(y);
```

- This was applied to y(t) with several realization of the noise n(t) for a fixed value of σ^2 to estimate m
- This was repeated to different increments of variance σ^2

  Observation: As the variance increases the estimation, error increases
- The bias of the estimator was developed
- To show that the estimator in unbiased for high N, a plot of the bias against N was made

  Observation: It was observed that N increases, bias tends to zero
- The estimator $\hat{m} = \frac{1}{N-1}\sum_{t=1}^{N} y(t)$ was modelled and the above operations was performed to calculate the bias.
- To show that the estimator is biased with mc high and N low, a plot of the bias was made for this condition.
- To show that the estimator is asymptotically unbiased, a plot was made to show that as N becomes large, bias b becomes 0.
- A plot of the variance of the estimator $m\hat{}_{ML}$: $var(m\hat{}_{ML}) = E\,(m\hat{}_{ML} - E[m\hat{}_{ML}])$ was made as a function of N for fixed σ^2 and as a function of σ^2 for fixed N.
- The result was compared with Cramer-Rao bound CRLB = σ^2 /N for different values of σ^2.

  Observation: The Variance of the expectation is greater or equal to CRLB for different values of σ^2

  Conclusion: The estimator is unbiased and efficient since it satisfies the Cramer-Rao bound rule

D) Non-Linear Estimation: Spectral analysis

Observation model: $y(t) = a_1 \sin(2\pi f_0 t) + a_2 \sin(2\pi f_1 t) + n(t), \quad t = 1,\ldots,N,$ where a_1 and a_2 are known amplitudes and f_0 and f_1 are unknown frequency (between 0 and 1/2) that is to be estimated. n(t) is a Gaussian random process with mean=0, and variance = σ^2

- $a_1 = 2$, $a_2 = 4$, $f_0 = 0.25$ and $f_1 = 0.4$ were chosen arbitrarily and

  y $= [\, y(1) \quad y(2) \quad \cdots \quad y(N) \,]^T$ was generated.

- An algorithm was developed to compute the Least-Square criterion

$$J(f_0, f_1) = \sum_{t=1}^{N} (y(t) - a_1 \sin(2\pi f_0 t) - a_2 \sin(2\pi f_1 t))^2 .$$

- $a_1 = 1$, $a_2 = 0$ were chosen, meaning the second sine function will be zero. The criterion was plotted for several values of f_0 (0:0.01:0.5) for a fixed variance = 1
- $a_1 = 1$, $a_2 = 0$ were chosen, meaning the second sine function will be zero. The criterion was plotted for several values of variance, with f_0 kept constant.
- $a_1 = 1$, $a_2 = 1$ were chosen. The criterion was plotted for several values of f_0, f_1 (0:0.01:0.5) for a fixed variance = 1
- $a_1 = 1$, $a_2 = 1$ were chosen. The criterion was plotted for several values of variance, with f_0, f_1 kept constant.

1.4 Result and Discussion

1.4.1 Result on Noises

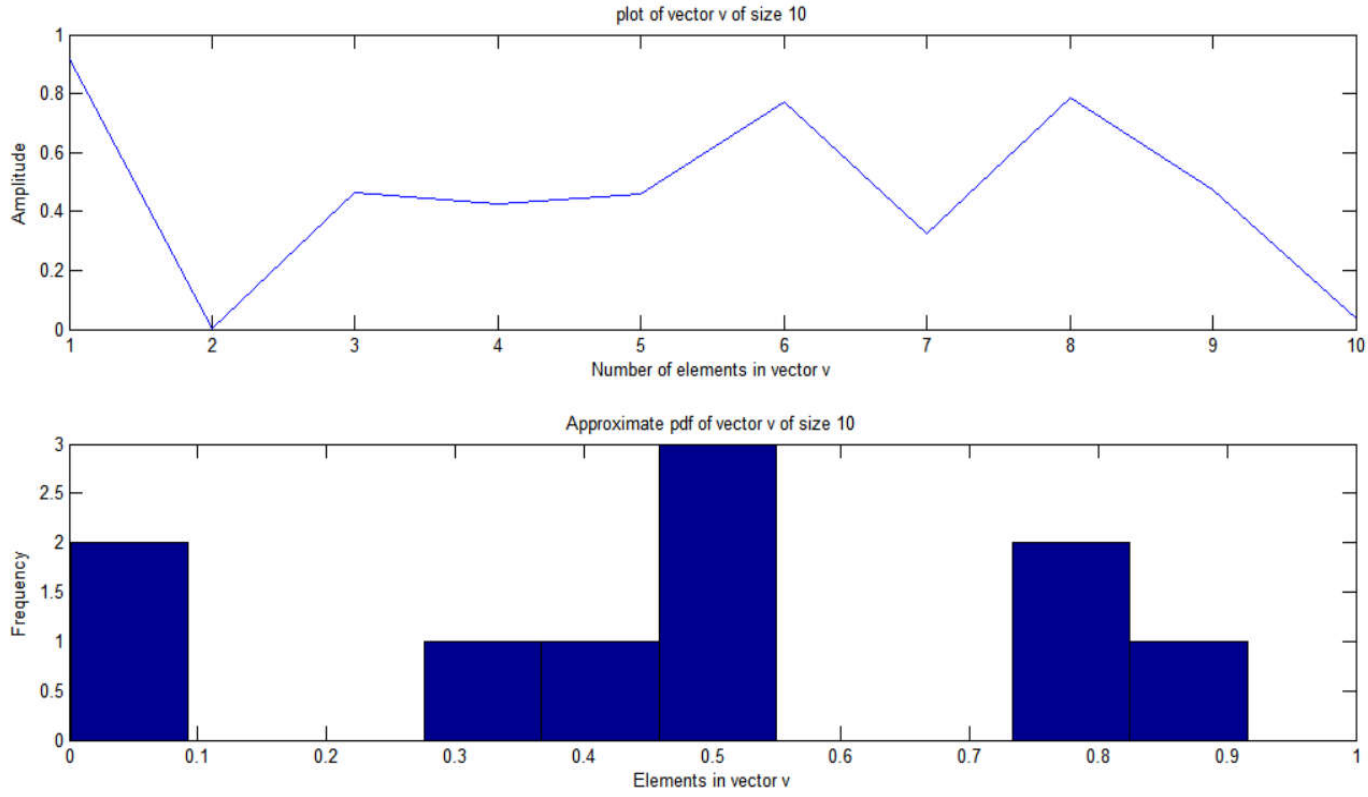

Figure 1.1 Plot realized for the vector and its approximate pdf

After the command rand was used to create a vector of random variables with size 10, the plot realized for the vector and its approximate pdf is shown in figure 1.1

The size of this vector was increased to 10000, and its plot and approximate is shown below in figure 1.2.

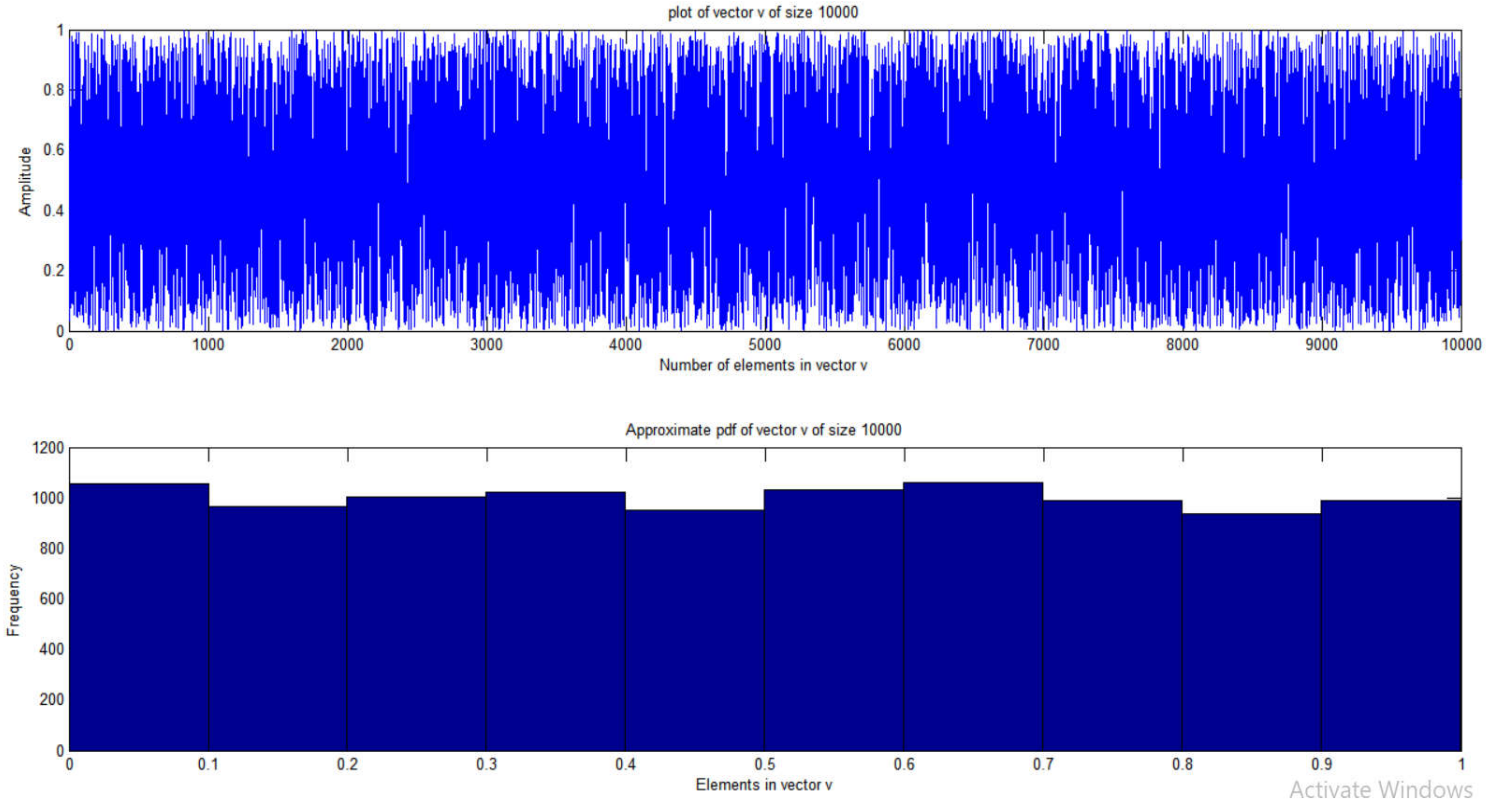

Figure 1.2 Plot realized for the vector for size 10000 and its approximate pdf

Observation: It was observed that as we increase the size of vector V, the pdf becomes more compact together and was becoming close to uniform over the values. This means that if V is large enough, then the PDF will become uniform.

The mean and the variance of the vector v with size 10 (v_mean_1, v_var_1) and the mean and variance for size 10000 are shown in figure 1.3.

```
v_mean_1 =      v_var_1 =      v_mean_2 =      v_var_2 =

    0.5261          0.0874          0.5020          0.0848
```

Figure 1.3 Result for the variance and mean when size = 10 and when size = 1000

Observation: It was observed that as the size of the vector increase from 10 to 10000, the mean becomes closer to the center value (0.5) as seen in v_mean_2. The variance also becomes smaller from 0.087 to 0.085.

This process was repeated for command randn(), which generates a random normal variable. The plot of the vector v realized and its pdf is shown in figure 1.4.

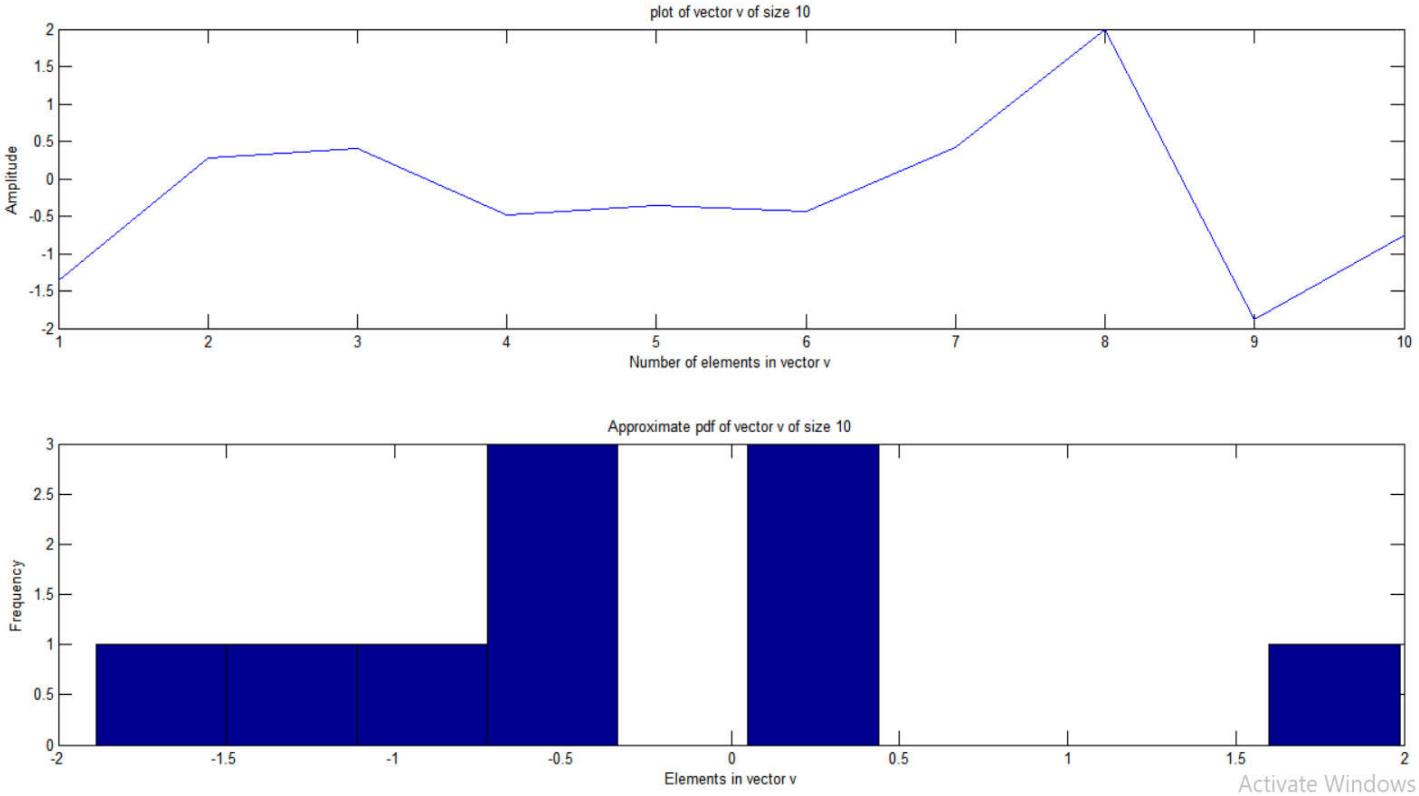

Figure 1.4 Plot of vector V and its pdf for randn()

Figure 1.5 shows the result for size 10000

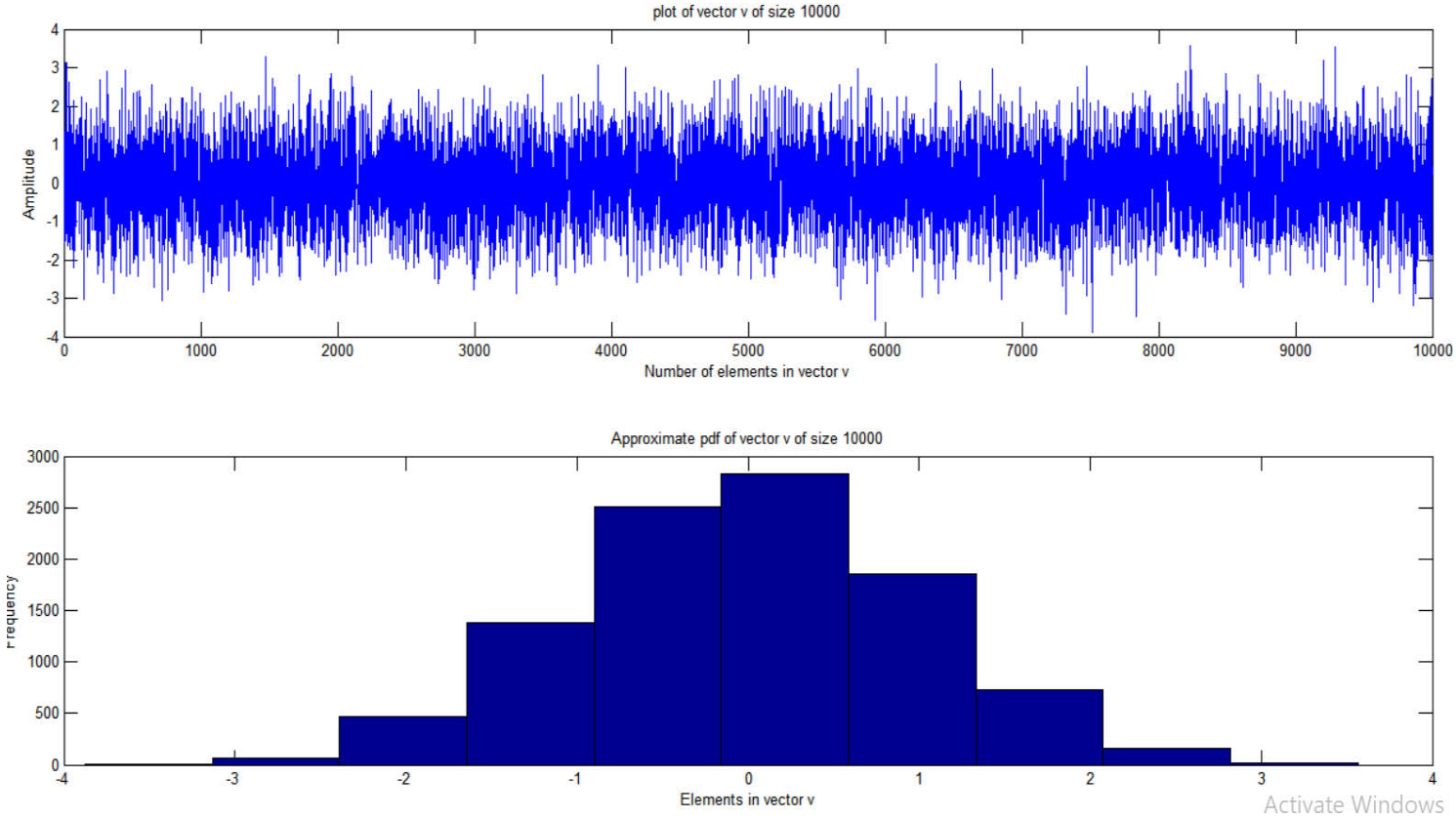

Figure 1.5 Plot realized for the vector V for size 10000 and its approximate pdf for randn()

Observation: It was observed that as the size of v increased to 10000, the normal distribution nature of the distribution becomes more evident, increasing the value further will see higher frequency concentrating at the center of the distribution. This is also evident in the values of the mean and the variance of the distributions shown in figure 1.6.

v_mean_1 =	v_var_1 =	v_mean_2 =	v_var_2 =
0.2029	1.3760	-0.0072	1.0369

Figure 1.6 Result for the variance and mean when size = 10 and when size = 1000

Observation: As can be seen from the figure, the mean is closer to zero (v_mean_2 = -0.0072) for size 10000 compare to (v_mean_1=0.2029) for size 10. It was also observed that the variance becomes closer to 1 as well as the size of the distribution increases.

1.4.2 Result on Central Limit Theorem

The values of random variable Sj which is a sum of N size of uniform random variable over [-1,1] was plotted and the pdf was shown below in figure 1.7 for N = 10, 1000, and 100000;

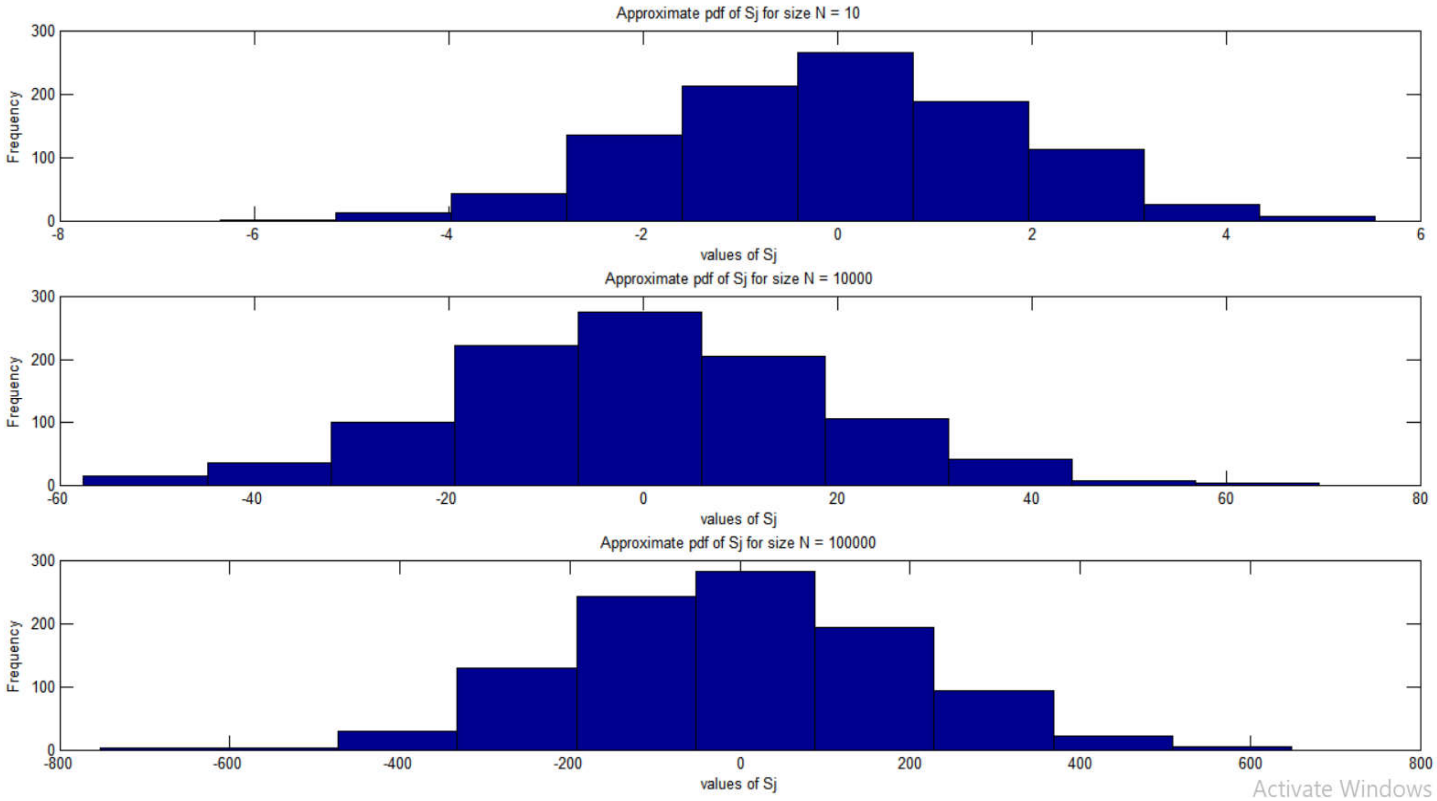

Figure 1.7 Result for the pdf for varying size of N

Observation: It was observed that the mean of the distribution approaches 0 as N was increased from 10 to 100000.

1.4.3 Result on Linear Estimation

A plot of the vector y was made with m = 5 and various values of σ^2 , the plot is shown below in figure 1.8.

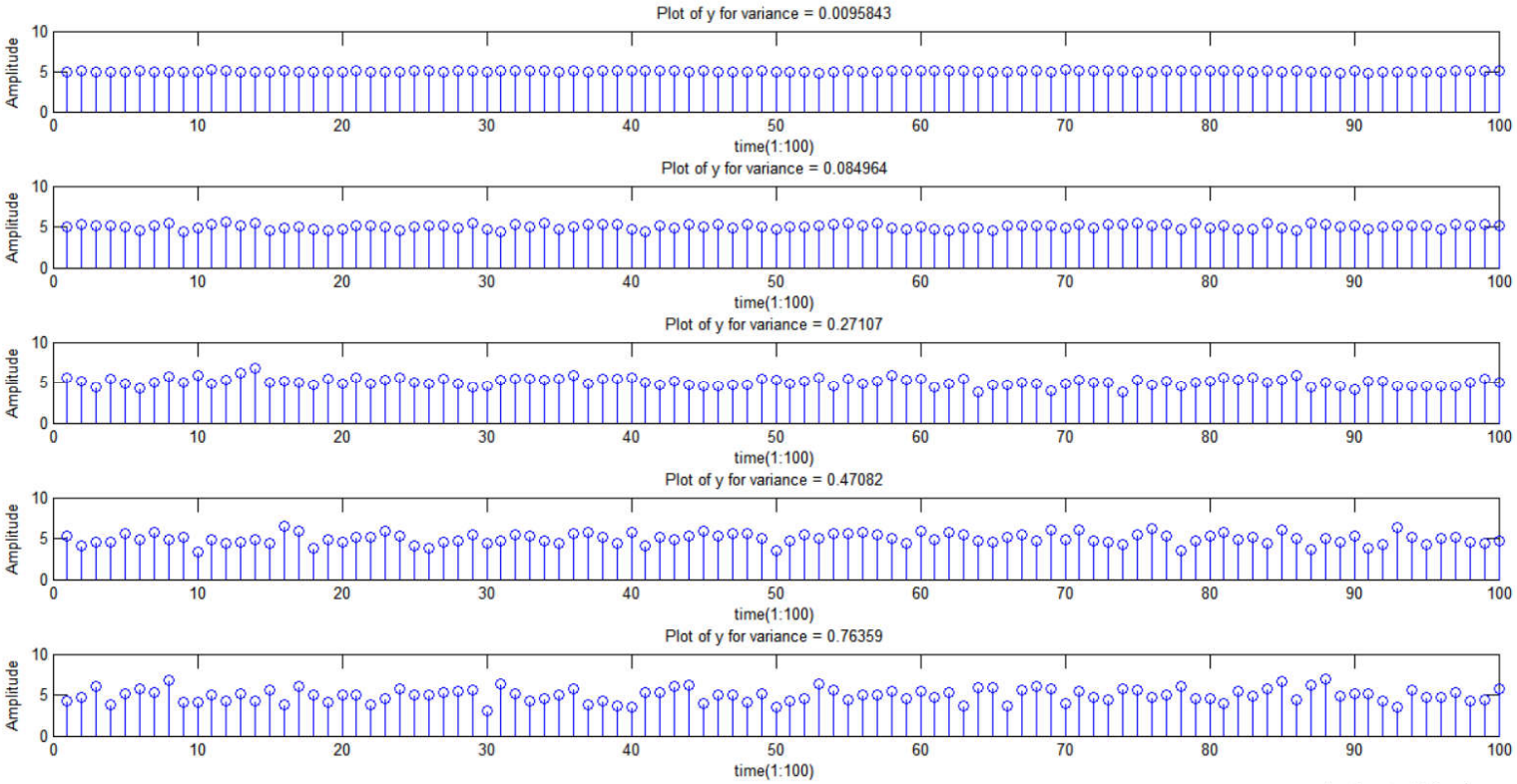

Figure 1.8 Plot of y for different values of variance

Observation: It can be observed that as the variance increases from 0.0095843 to 0.76359 the Amplitude of the individual components of y becomes more distinct.

The Maximum Likelihood Estimator was realized and a it value for different values of N and fixed variance, the result is shown in below

```
Estimated value for N = 10 is 5.173
Estimated value for N = 100 is 5.1589
Estimated value for N = 1000 is 4.9827
Estimated value for N = 10000 is 4.9818
Estimated value for N = 100000 is 4.9946
```

From the result, it can be observed that as N increases, the estimator performance in estimating the value of m improves (recall that actual value of m = 5).

The Maximum Likelihood Estimator was realized and it value for different values of variance with fixed N = 1000 is shown below

```
Estimated value for variance = 4.0888 is 4.9988
Estimated value for variance = 16.6993 is 5.0877
Estimated value for variance = 34.3202 is 4.977
Estimated value for variance = 60.086 is 4.8445
Estimated value for variance = 105.3161 is 4.5269
```

From the result, it can be observed that as the value of variance increases the performance of the estimator becomes worst.

The bias of the estimator was realized and the values for several estimates of N for a fixed variance, the plot is shown in figure 1.9.

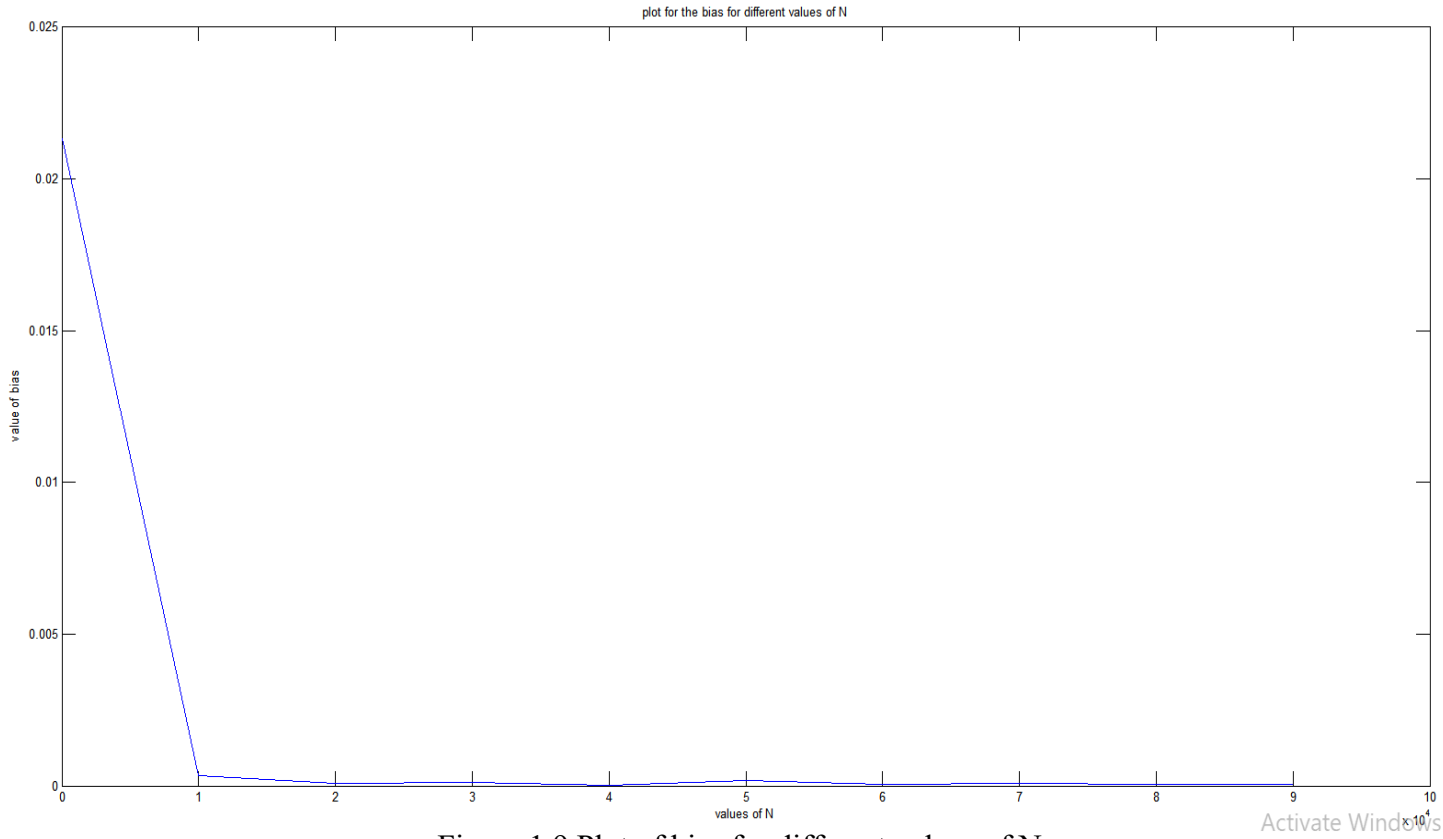

Figure 1.9 Plot of bias for different values of N

Observation: the result shows that as the N increases, the bias tends to 0, this means that at high value of N, the estimator is unbiased.

A second estimator was realized, the estimator was bias with m high and N low.

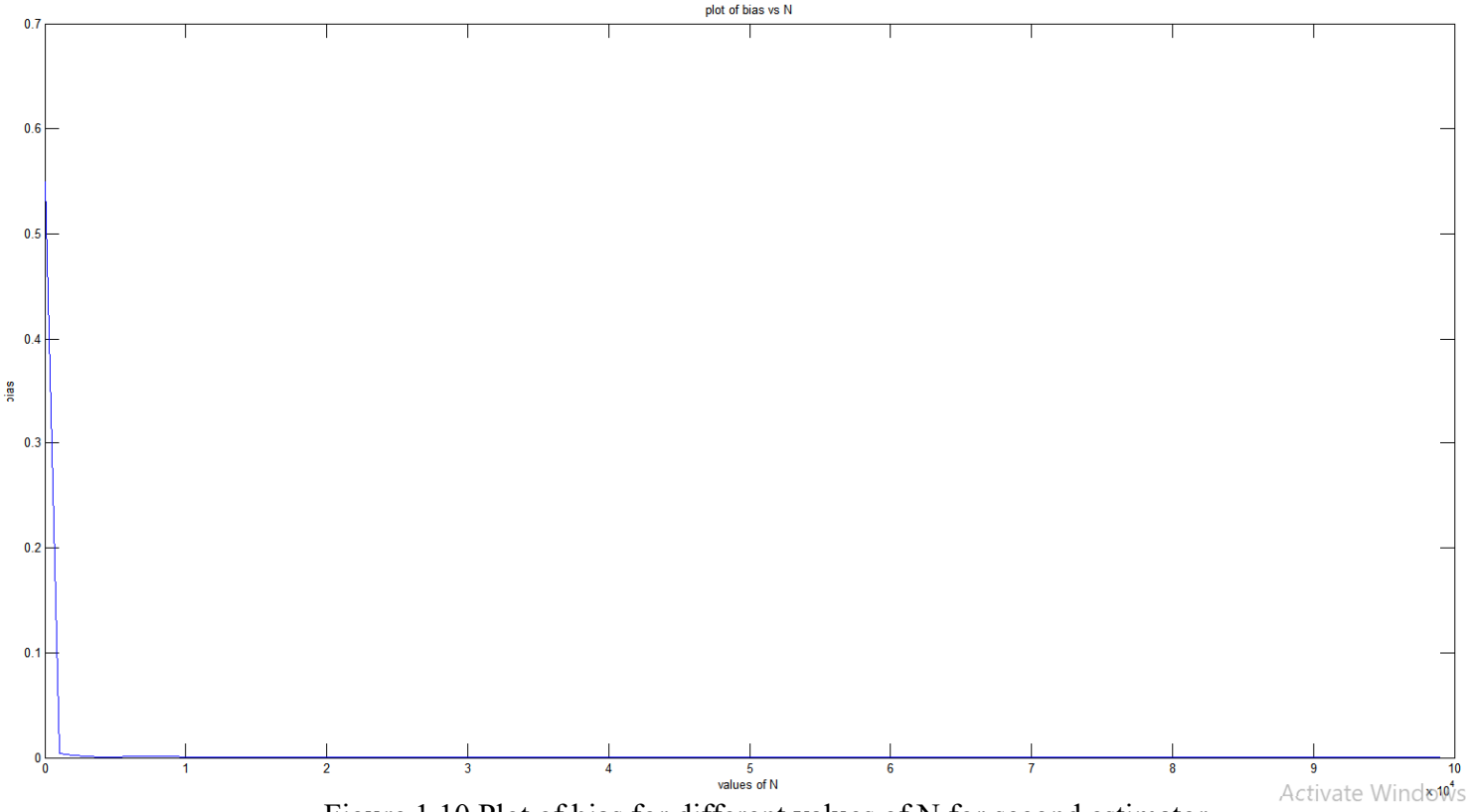

Figure 1.10 Plot of bias for different values of N for second estimator

Observation: The estimator was asymptotically biased because at high values of N the bias became 0. And it has a better performance than the previous estimator as shown in the plot below in figure 1.10.

A plot of the variance of the estimator as a function of N was made with fixed σ^2, and also for fixed N and varying σ^2 as shown in figure 1.11.

Observation: It was observed that the variance of the estimator reduces as N increases, but on the other hand, it increases as σ^2 increases.

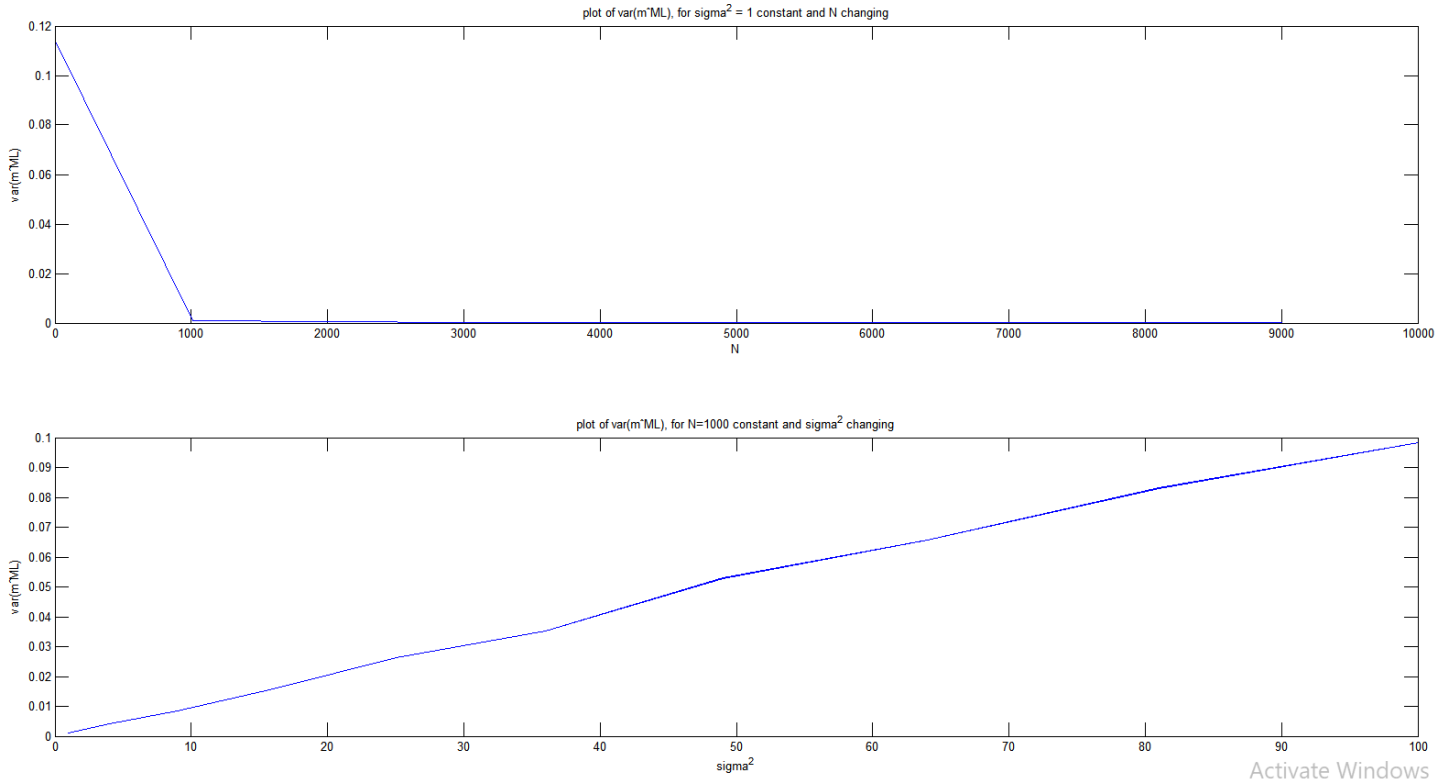

Figure 1.11 Plot of bias variance with sigma2 constant and N varying and vice-versa

The comparison with the Cramer Rao Lower bound was made in the plot in figure 1.12.

Observation: The variance of the estimator is equal or greater than the CRLB for different values of σ^2, hence it is an efficient estimator.

Conclusion: This Estimator is an efficient estimator because it is unbiased for large enough values of N and it satisfies the CRLB.

1.4.4 Result on Non-Linear Estimation

A plot of y was made for various values of σ^2 as shown in the plot in figure 1.13.

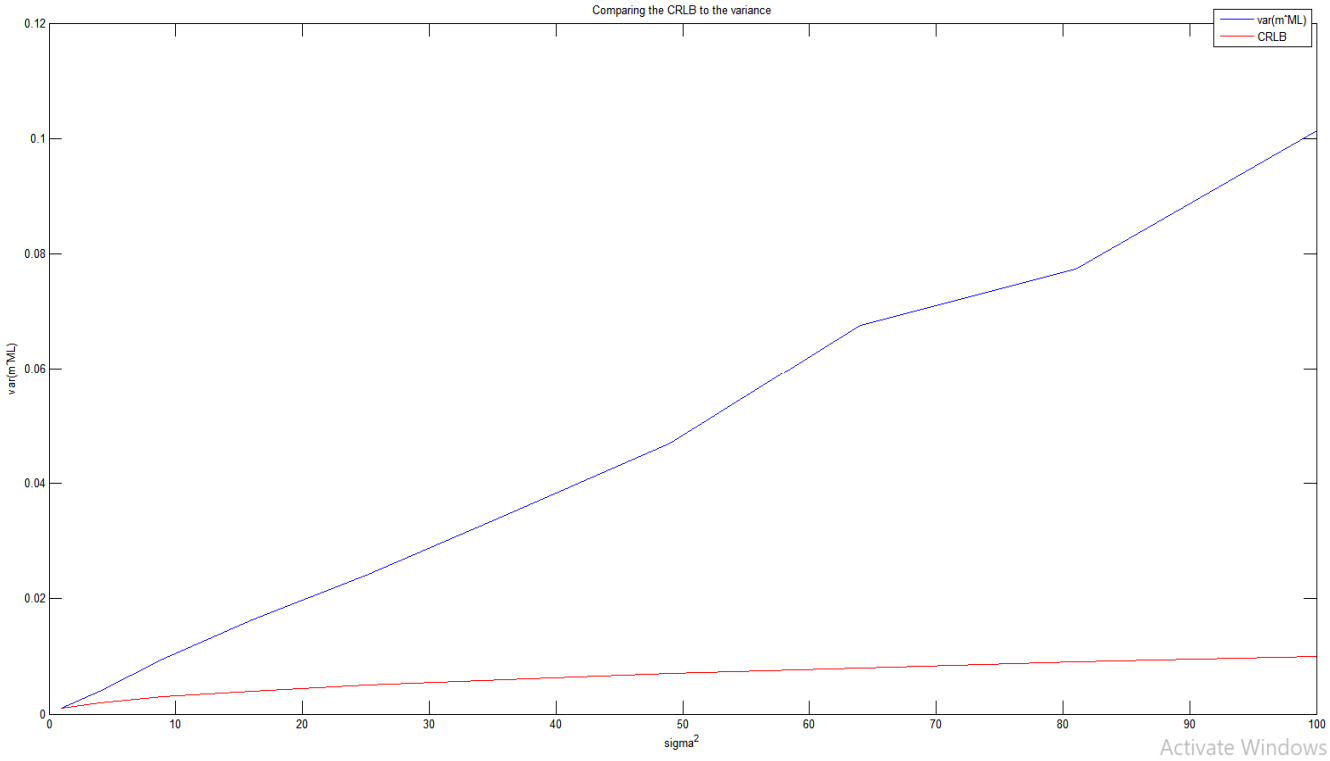

Figure 1.12 Comparing CRLB to the variance vs sigma^2

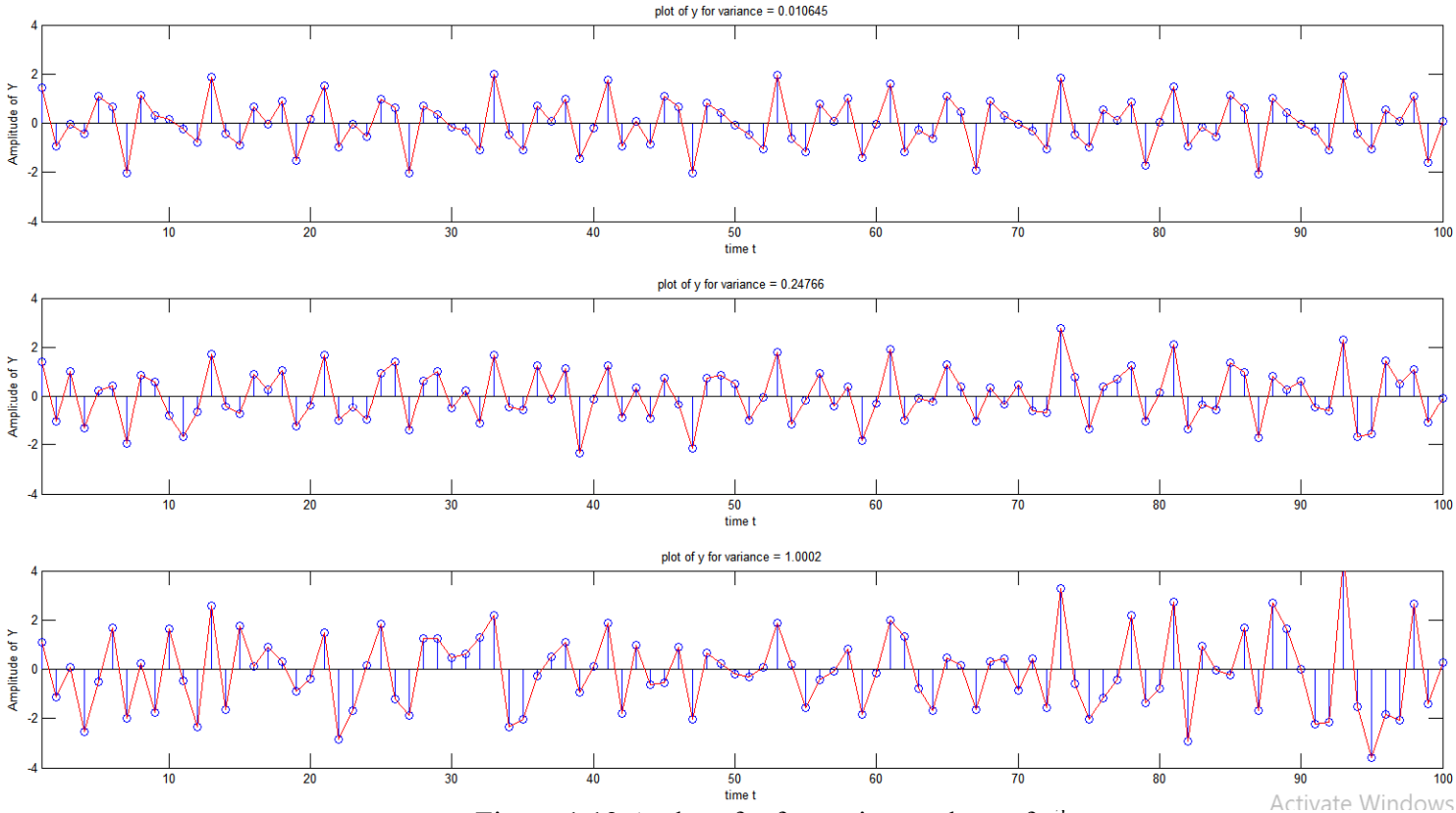

Figure 1.13 A plot of y for various values of σ^2

Observation: It was observed that as variance increases the difference in amplitude of the various elements of y also increases.

We developed an Algorithm for Least Square Criterion, and we plot the criterion as shown in figure 1.14 for a1 = 1 and a2 = 0, for several values of f_0 (0:0.01:0.5) for a fixed variance.

The value of fo that correspond to minimum of J was picked by ginput() and it is shown below

```
minimum coordinate of J as picked by manually by ginput() is
value for f0 = 0.24942
at J = 1001.7544
```

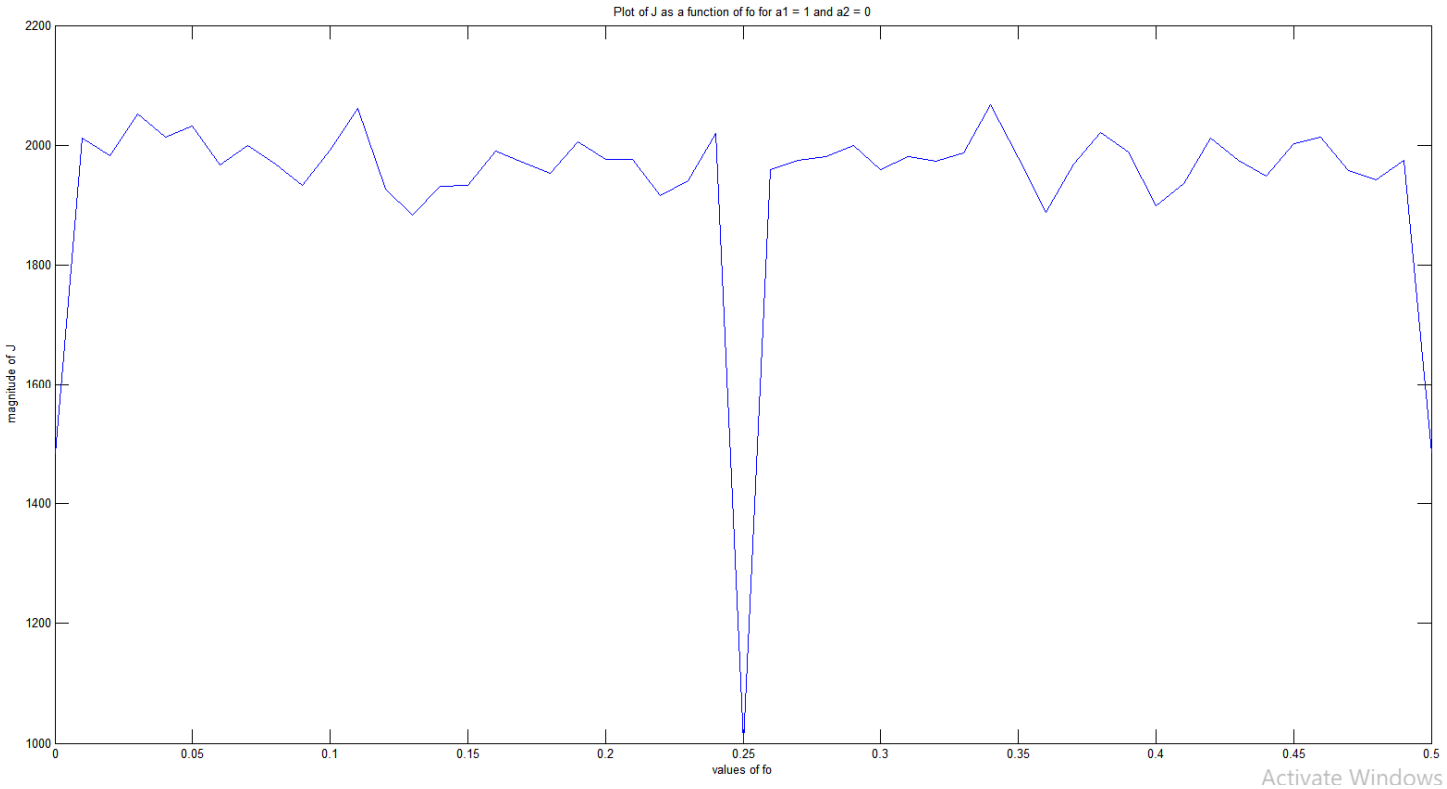

Figure 1.14 plot of the criterion for a1 = 1 and a2 = 0, for several values of f_0

A plot was also made to visualize how J varies with σ^2 as shown in figure 1.15

Observation: It was observed that as σ^2 increases the value of Amplitude of J also increases.

A plot of variation in J was made for a1 = 1 and a2 = 1, for changing values of fo, and fixed value of f1 and σ^2 as shown in figure 1.16.

This is similar to the previous except that the function J has a peak value at a possible estimation of f1. Meanwhile its estimation of fo shown by its minimum is close to 0.25.

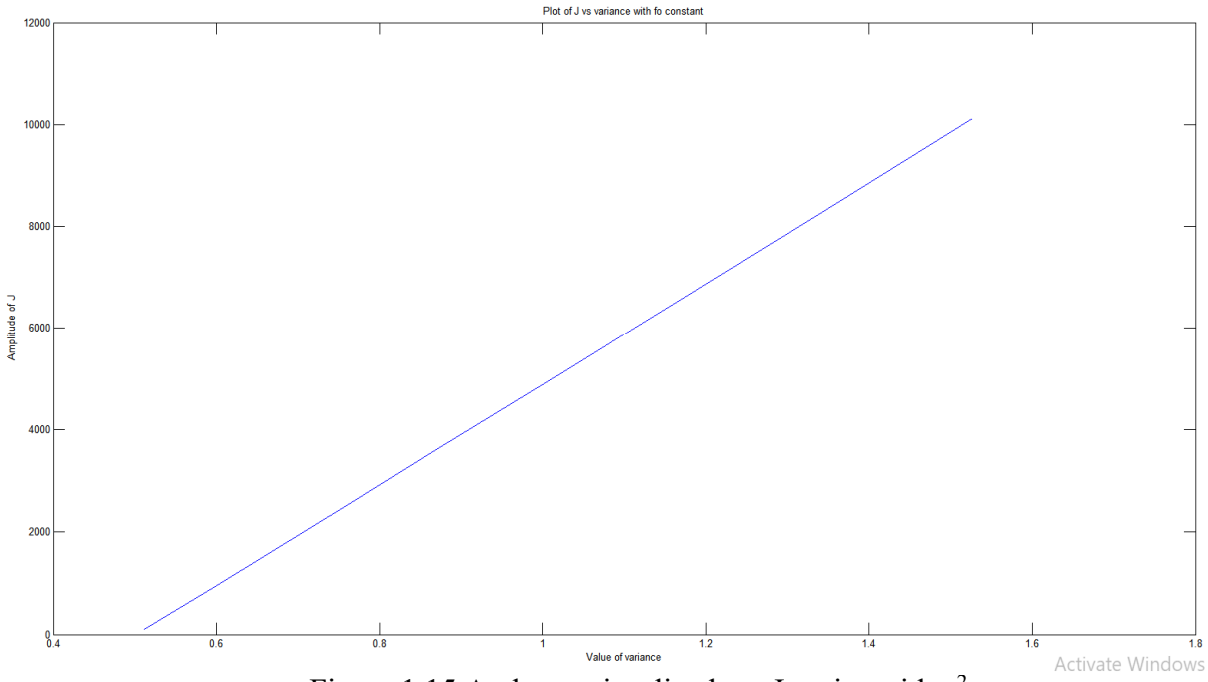

Figure 1.15 A plot to visualize how J varies with σ^2

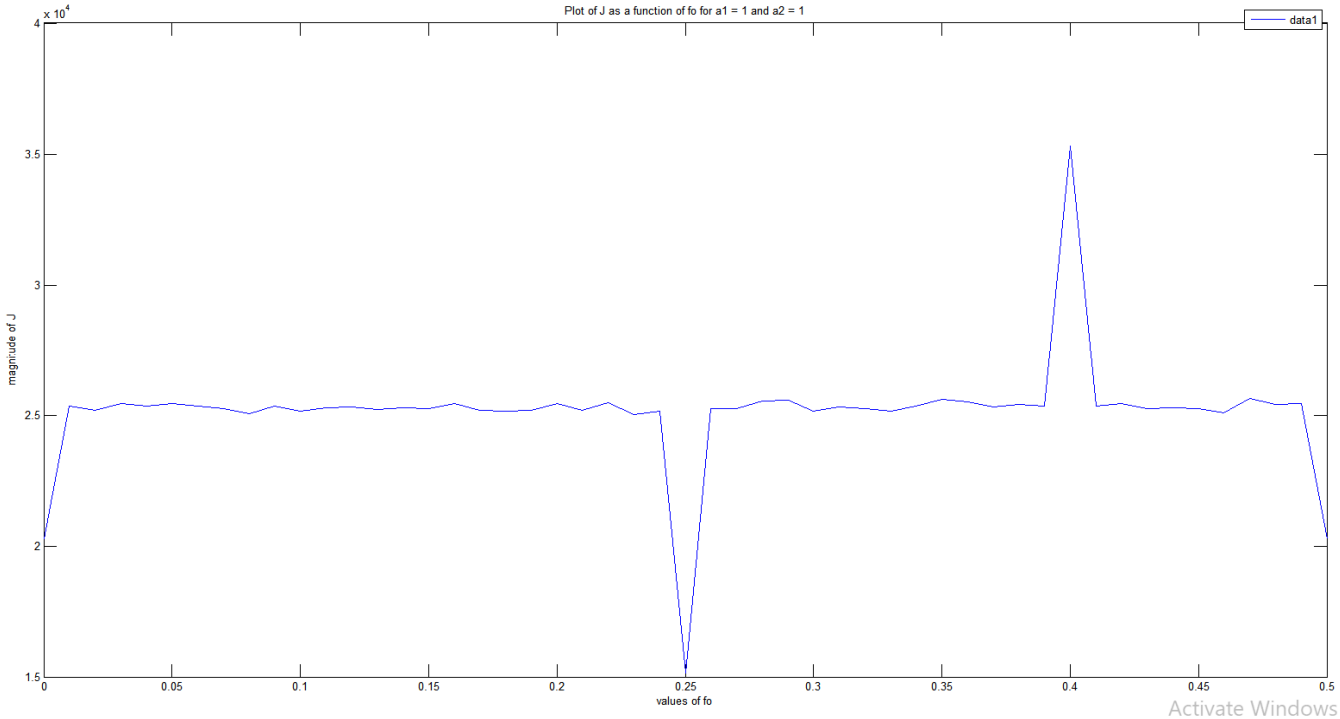

Figure 1.16 A plot of variation in J with a1 = 1 and a2 = 1, for changing values of fo, and fixed value of f1 and σ^2

A plot of J vs variance was also performed with a1=a2 = 1 with fo, f1 constant and it is shown in figure 1.17.

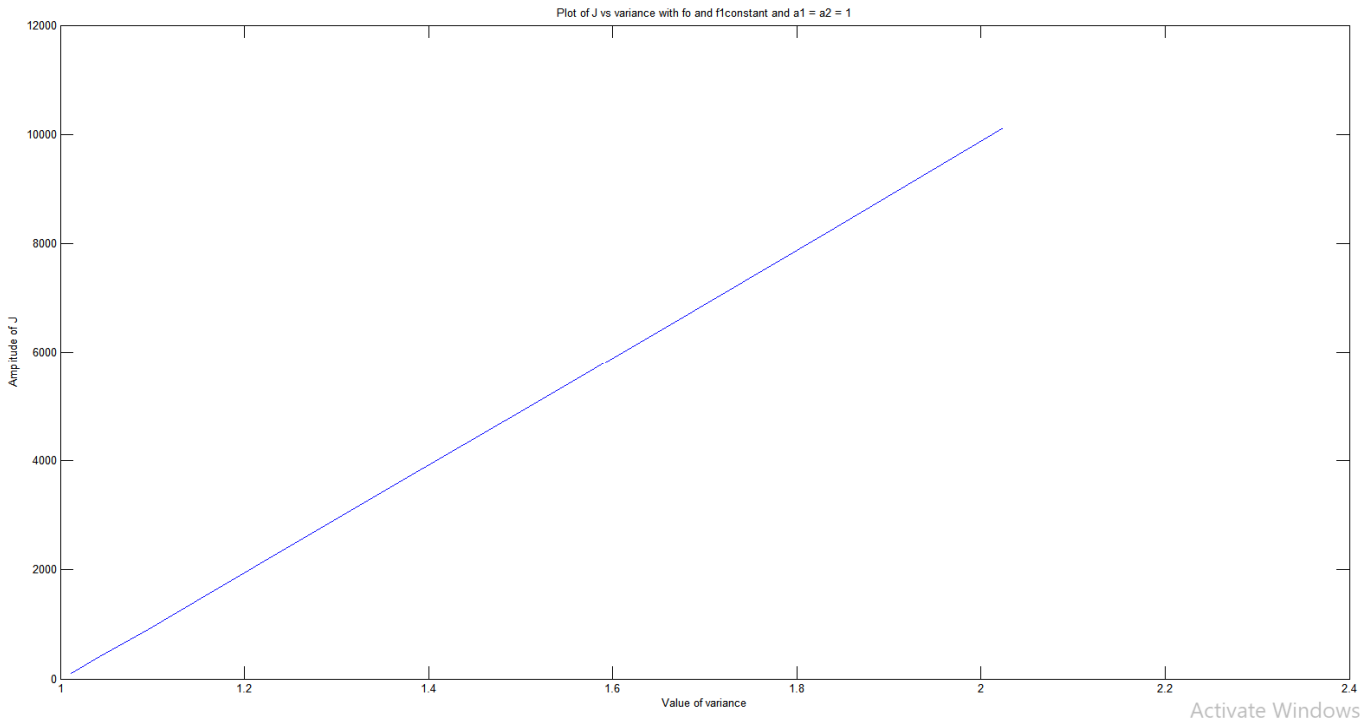

Figure 1.17 A plot of J vs variance with a1=a2 = 1 with fo, f1 constant

Finally, a plot of J(fo,f1) vs varying values of fo and f1 for a fixed value of σ^2 was made and shown in figure 1.18.

A better view of it was done by rotating the 3d plot as shown in figure 1.19, the darkest blue point shows the lowest value of J and the darkest red show the highest values.

It is seen from the plot picked that the minimum value of J(fo,f1) falls close to fo=0.25, f1 = 0.4 or vice versa, which is the exact value that was chosen initially.

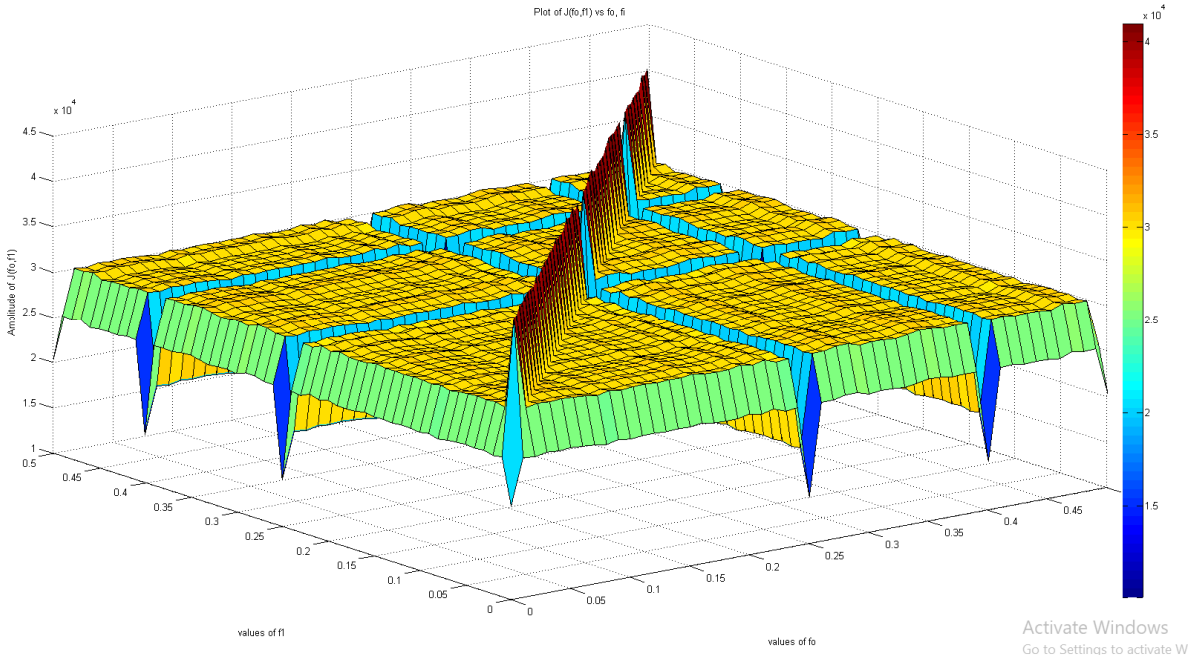

Figure 1.18 A plot of J(fo,f1) vs varying values of fo and f1 for a fixed value of σ^2

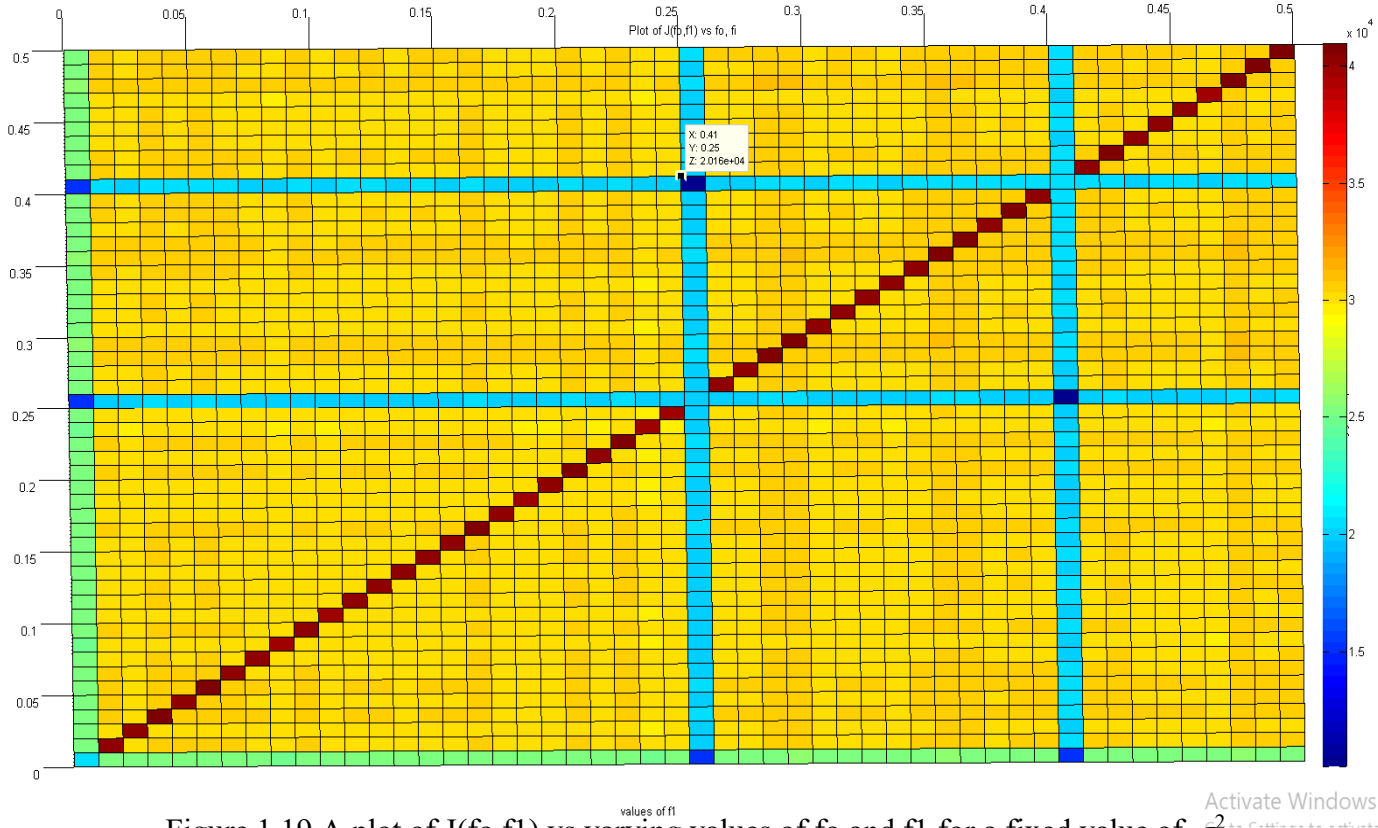

Figure 1.19 A plot of J(fo,f1) vs varying values of fo and f1 for a fixed value of σ^2

1.5 Conclusion

After the part on introduction to matlab programming, we generated noise with rand() and randn(), it was learnt that randn() generates white Gaussian noise. An analysis of the central theorem was done and it was observed that as N becomes large for the signal Sj generated, more of the values of S aligns in the center of the approximate pdf plot, forming a better Gaussian shape. Two linear estimation model was developed to estimate the message in a noise corrupted signal, the second estimator performs better even though it is asymptotically unbias. The variance of the expectation of the estimator is greater or equal to CRLB for different values of variance in data. Hence, we concluded that the estimator is unbiased and efficient since it satisfies the Cramer-Rao lower bound rule. A least square algorithm was developed to estimate the unknown variable in an observation model. The algorithm was analyzed manually by plotting it and picking the points with where it has its minimum value. It successfully estimated the unknown variables with good accuracy.

Chapter 2 Signal Processing

2.1 Introduction

The goal of this practical is to analyze and synthesize some particular sounds. The recording can be modelled by the repetition of K elementary functions similar to weighted sine functions with a repetition period T.

A weighted sinus function can be easily modelled by an Autoregressive process (AR). These processes are very useful since they need few parameters (approximatively 20 for a sound). However, in order to fit with the signal, we have to estimate these coefficients of the AR process in order to characterize the sound. The synthesis is the inverse process. If we have the AR coefficients, the number of repetition *K* and the period of repetition *T*, how to recover the sound?

2.2 Objectives

1. To analyze a sound

2. To synthesize the sound

2.3 Practical Procedure

A) Parameter Estimation of an AR Process

We see that Sn is the output of a digital filter of frequency response

$$H(z) = \frac{1}{1 - \sum_{i=1}^{d} a_i z^{-i}},$$

- Num = 1
- Den = $z^d + a.z^{d-1} + a^2.z^{d-2} + \ldots\ldots + a^{d-1}.z + a^d$
- $a = \gamma.M^{-1}$, if γ is known, since M is a matrix of k values of γ, by finding the inverse of M and multiplying by γ we can deduce a.
- Firstly, from the equation of H(z), we determine that the numerator = 1 and the denominator is the coefficient of z^{-I} which is a vector $a_{i.}$

- With the command load synthe22, the files of sound in synthe22.mat file was downloaded
- The data for letter “a” was plotted, as a function of real time, and the useful part was isolated and saved in a new vector “A”.
- The MATLAB function iden(s,d) provided as part of the practical script was used to generate the denominator of H(z) of “A”, using d = 20
- The roots of the denominator of H(z) was derived using the roots() function, saved in variable rt and it was plotted on the complex plane.
- The modulus and the angle of rt was derived by using functions abs() and angle() respectively and it was confirmed that all values of abs(rt) is less than 1.
- The frequencies Fi was computed using $F_i = \frac{1}{2\pi} \arg(p_i) F_e$ whew pi is the derived roots rt.
- The impulse response and frequency response of H(z) was plotted using the command impz() and freqz().

B) Analysis of the sound

- Q1: What is the value of T : T = 5ms, it is a woman voice.
- Q2: What is the value of K : K = 786 (considering “A” the useful part of “a”).
- A program was developed to calculate the Fourier transform with a sampling frequency of Fe and it was plotted as a function of the real frequency axis.
- Q3: The value of T from the plot of the FFT is 5.02ms. Which confirms the fact that it is a woman voice
- Q4: Why is the fourier transform of a signal constituted of peaks? The point with the highest peak is the main frequency of the signal, the other peaks are other frequency present in the signal, which can be due to noise during the recording or change in intonation of the speaker.

C) Synthesis of the Sound

- Q1: What is the time between two impulses? 4.5ms
- A program was created to generate an input 'en' of the filter
- A program was created to give the output of the filter when the input is 'en'.
- Plots were made to compare the synthesized signal to the recorded one in time domain and in frequency domain.

2.4 Result and Discussion

2.4.1 Results from the part on Parameters Estimation of an AR process

Figure 2.1 shows the plot of the data of "a", and the useful part that was picked and saved as a vector "A"

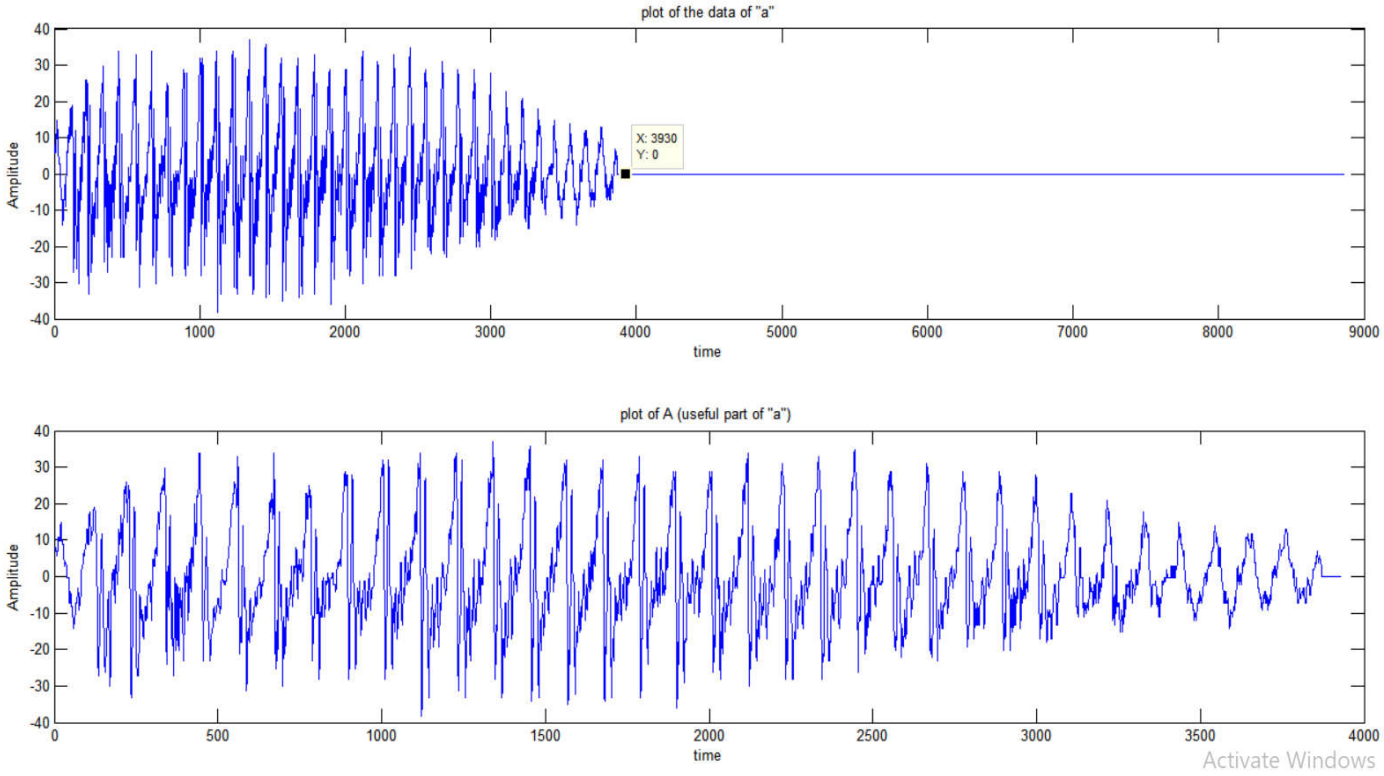

Figure 2.1 A plot of the data of "a", and the useful part "A"

The result of the coefficient of H(z) of the useful signal was derived and it is presented below

```
>> iden(A,20)

ans =

  Columns 1 through 9

    1.0000   -1.9881    1.2916   -0.2109    0.0246   -0.1708   -0.0165    0.0556    0.1042

  Columns 10 through 18

    0.1011   -0.1328   -0.0492   -0.0089    0.0958   -0.1089   -0.0167    0.0198    0.0522

  Columns 19 through 21

    0.0626   -0.1008    0.0251
```

The roots of the denominator of H(z) is shown as below as;

```
rt =

  -0.8084 + 0.1214i
  -0.8084 - 0.1214i
  -0.7293 + 0.4351i
  -0.7293 - 0.4351i
  -0.5268 + 0.6595i
  -0.5268 - 0.6595i
  -0.2781 + 0.7761i
  -0.2781 - 0.7761i
  -0.0145 + 0.8899i
  -0.0145 - 0.8899i
   0.3956 + 0.7859i
   0.3956 - 0.7859i
   0.6509 + 0.7352i
   0.6509 - 0.7352i
   0.8790 + 0.3261i
   0.8790 - 0.3261i
   0.9305 + 0.0000i
   0.7874 + 0.1542i
   0.7874 - 0.1542i
   0.3459 + 0.0000i
```

A plot of this in the complex plane is shown in figure 2.2 below

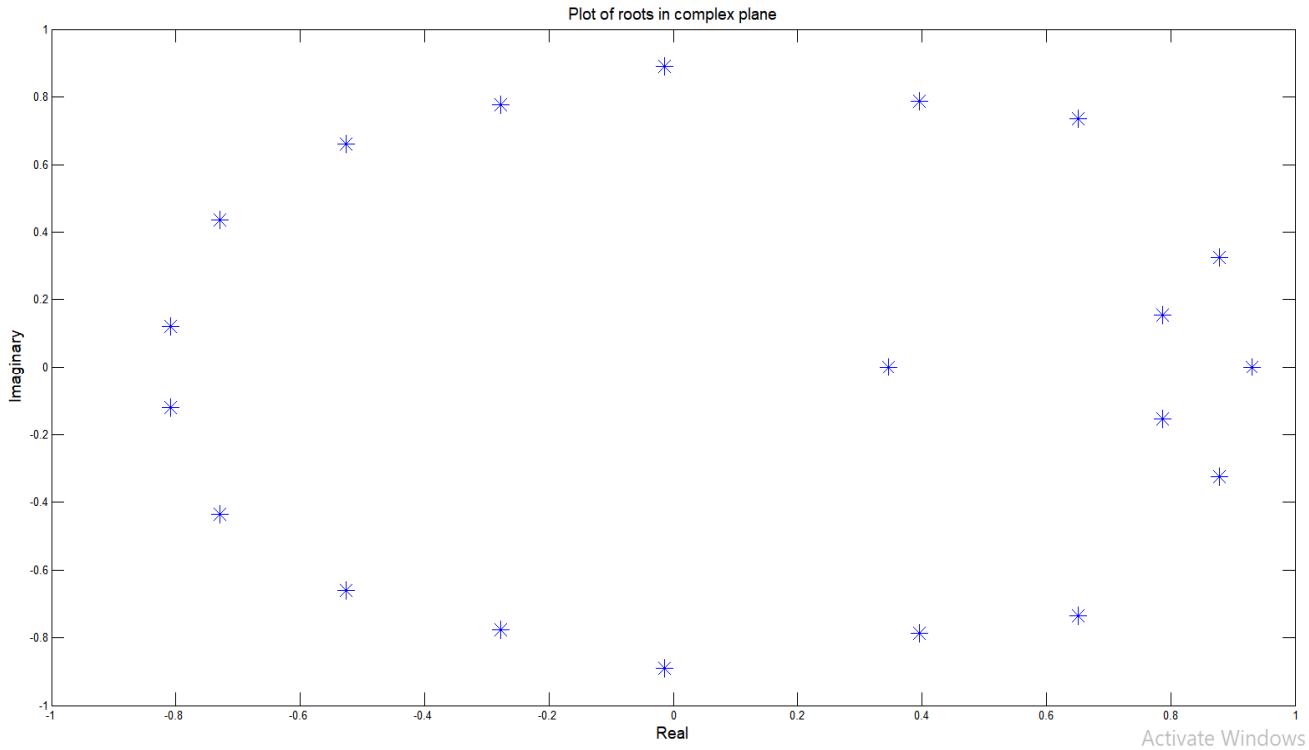

Figure 2.2 A plot of the root of the denominator of H(z) in complex plane

The modulus and angle of rt was derived and it is shown below, it can be observed that the modulus of rt is less than 1. The corresponding frequencies is also shown in below as well.

```
modulus =

    0.8175
    0.8175
    0.8493
    0.8493
    0.8441
    0.8441
    0.8244
    0.8244
    0.8900
    0.8900
    0.8799
    0.8799
    0.9820
    0.9820
    0.9376
    0.9376
    0.9305
    0.8024
    0.8024
    0.3459
```

```
ang =

    2.9925
   -2.9925
    2.6037
   -2.6037
    2.2448
   -2.2448
    1.9148
   -1.9148
    1.5871
   -1.5871
    1.1044
   -1.1044
    0.8461
   -0.8461
    0.3553
   -0.3553
         0
    0.1934
   -0.1934
         0
```

```
Frequencies =
   1.0e+04 *

    1.0502
   -1.0502
    0.9137
   -0.9137
    0.7878
   -0.7878
    0.6720
   -0.6720
    0.5570
   -0.5570
    0.3876
   -0.3876
    0.2969
   -0.2969
    0.1247
   -0.1247
         0
    0.0679
   -0.0679
         0
```

The plot of the frequency response and the impulse response of H(z) is shown below in figure 2.3.

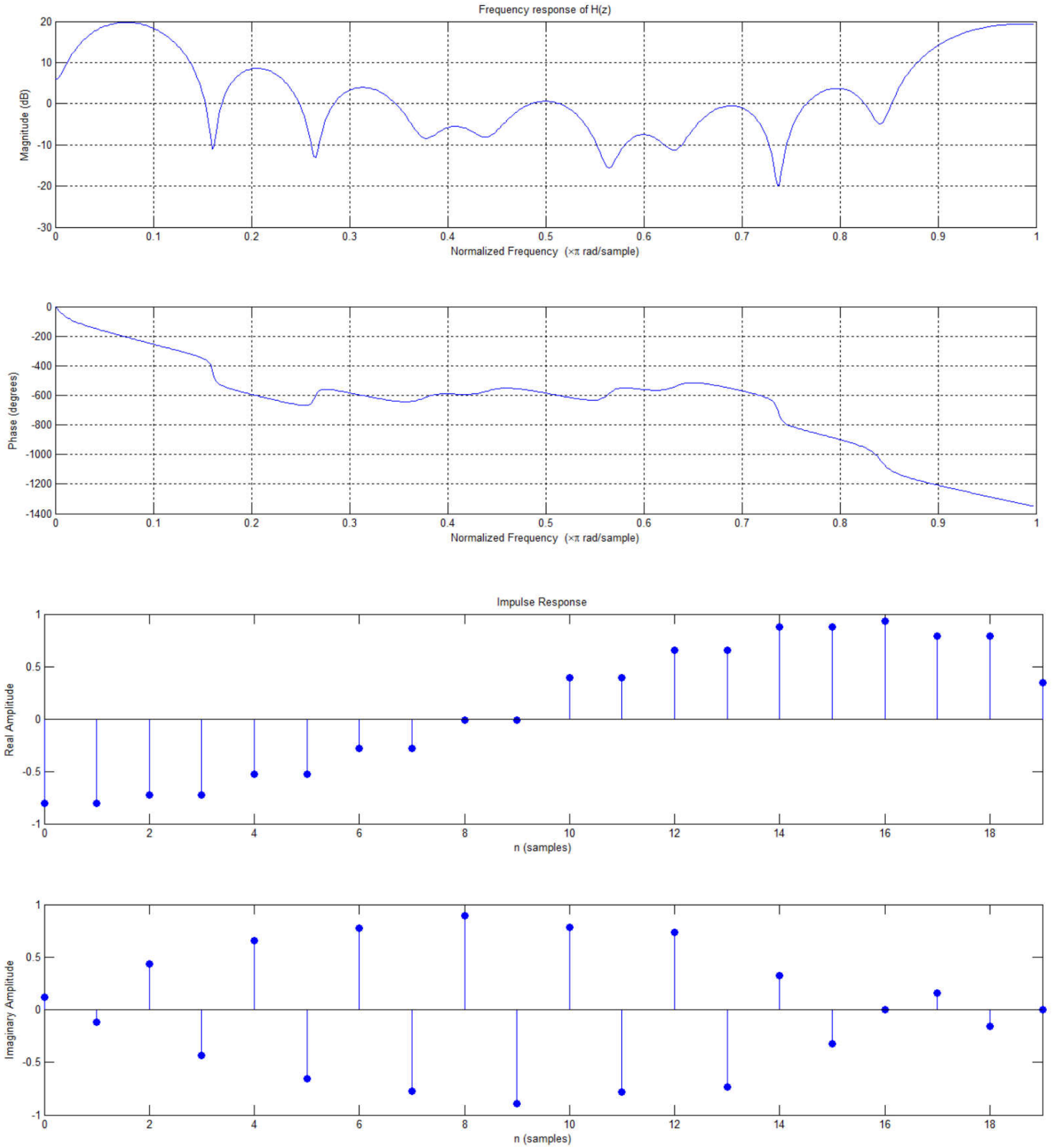

Figure 2.3 The plot of the frequency response and the impulse response of H(z)

The plot of the FFT of our signal ‘A’ is shown in figure 2.4 below, the point picked is the frequency 199.2Hz, if we convert it to time we have 5.02ms, which is the value of T.

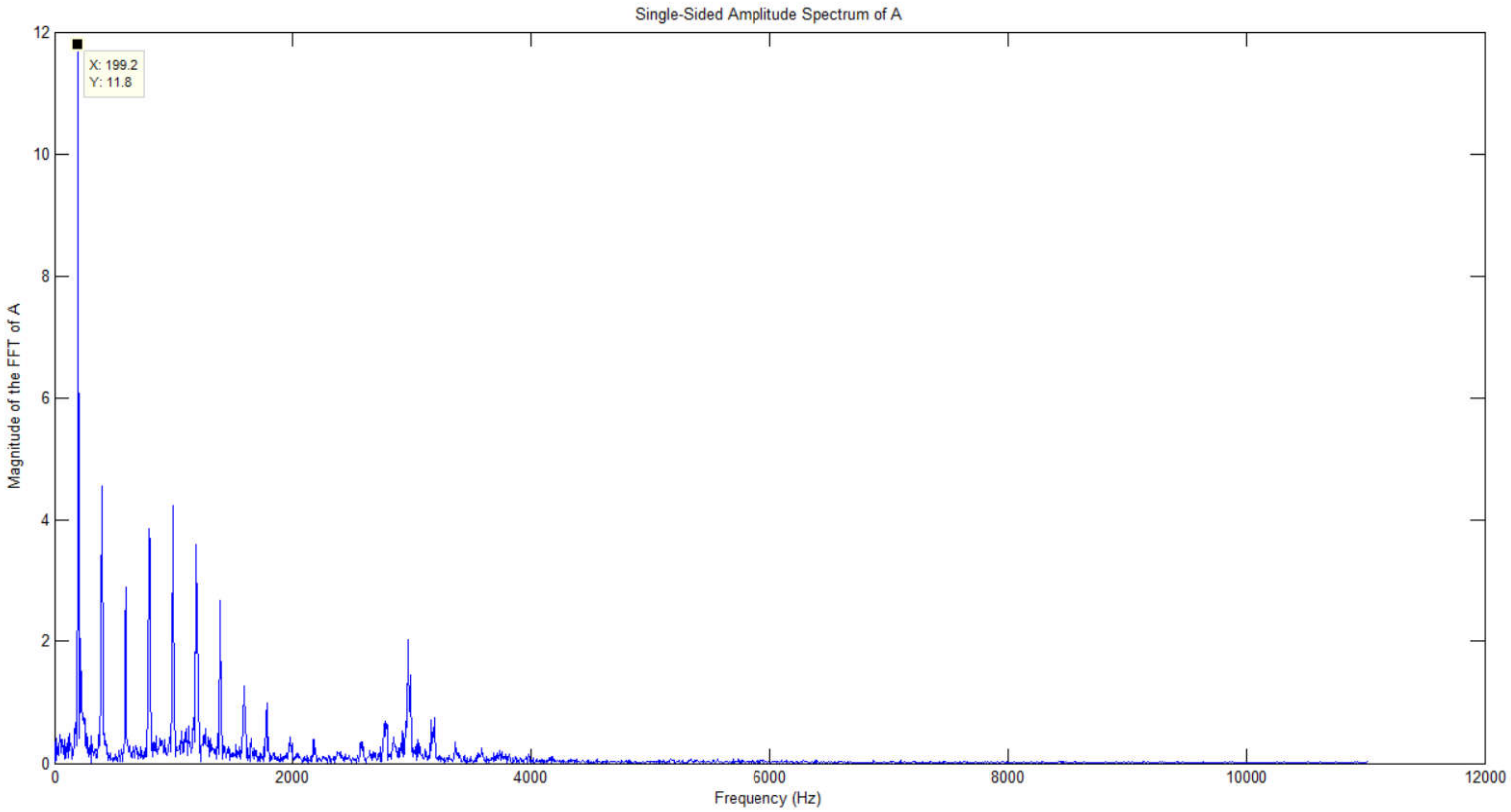

Figure 2.4 The plot of the FFT of our signal ‘A’

2.4.1 Results from the part on Analysis of the sound

After a program was created to give an ‘input’ to the filter, a plot showing the input and our signal “A” is shown below in figure 2.5.

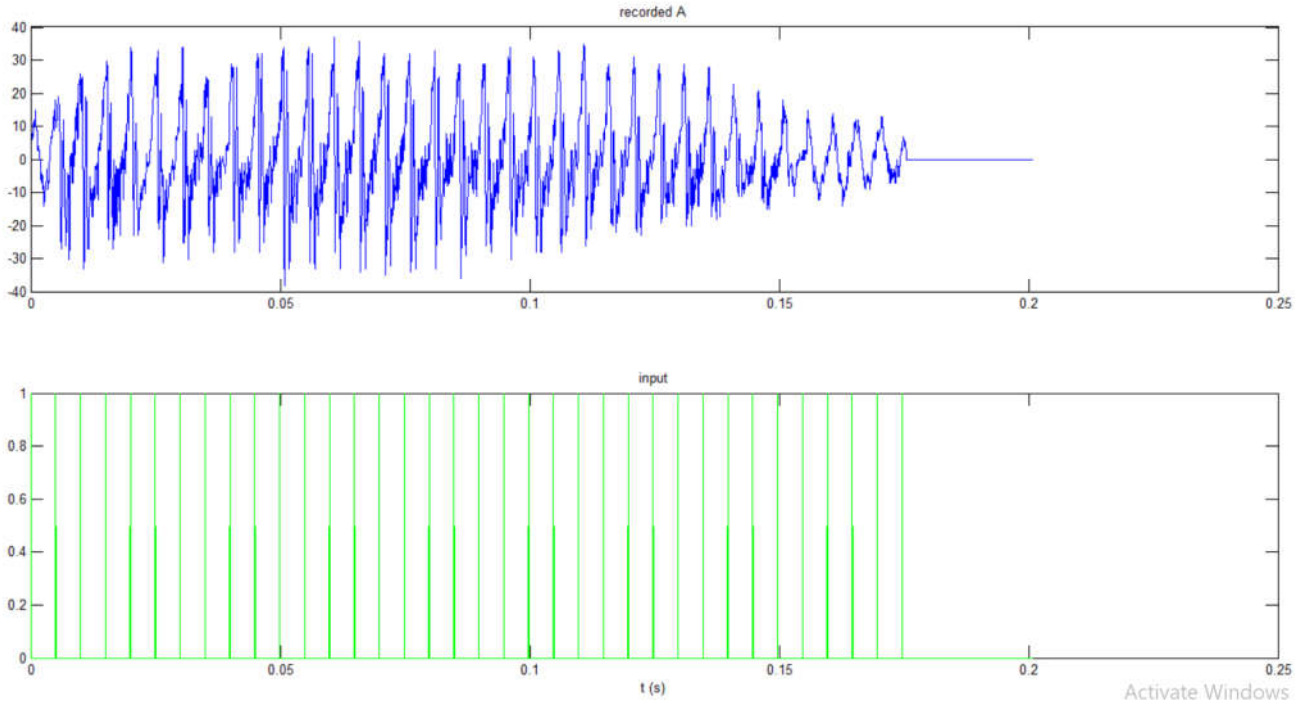

Figure 2.5 A plot showing the input and our signal “A”

Figure 2.6 below shows the ‘input’ and the resulting synthesized signal

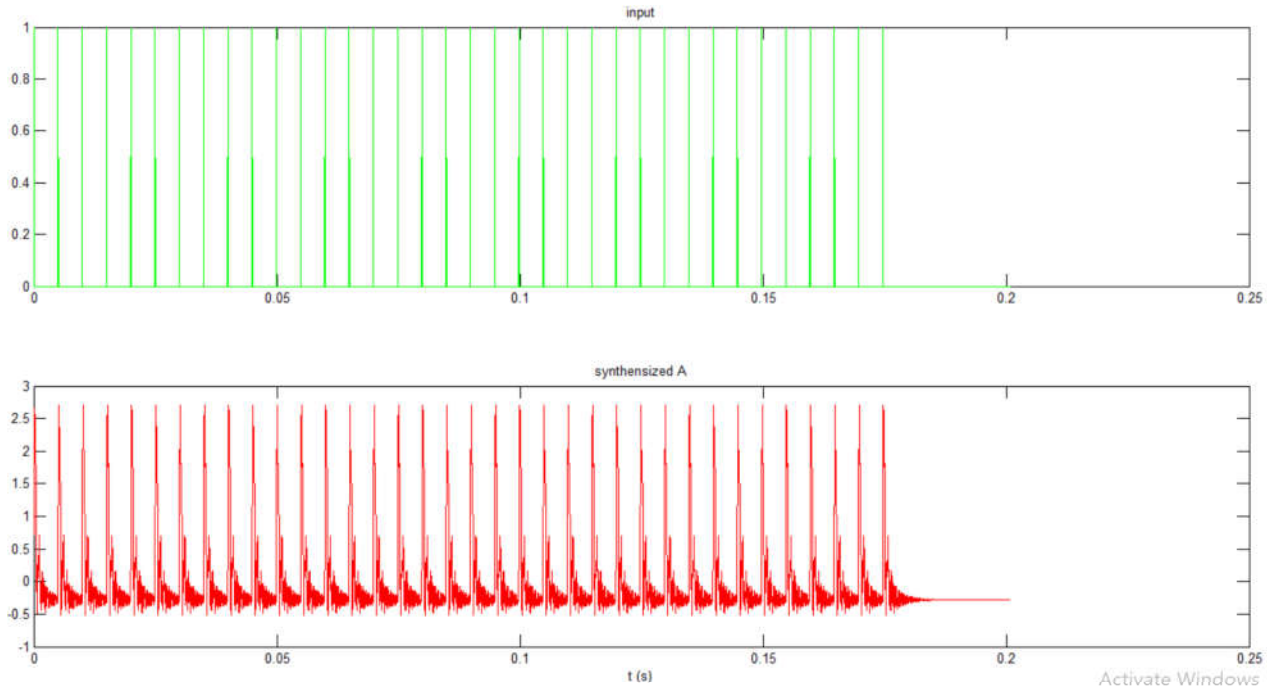

Figure 2.6 A plot shows the ‘input’ and the resulting synthesized signal

Figure 2.7 shows the plot of our recorded “A” and the synthesized “A”

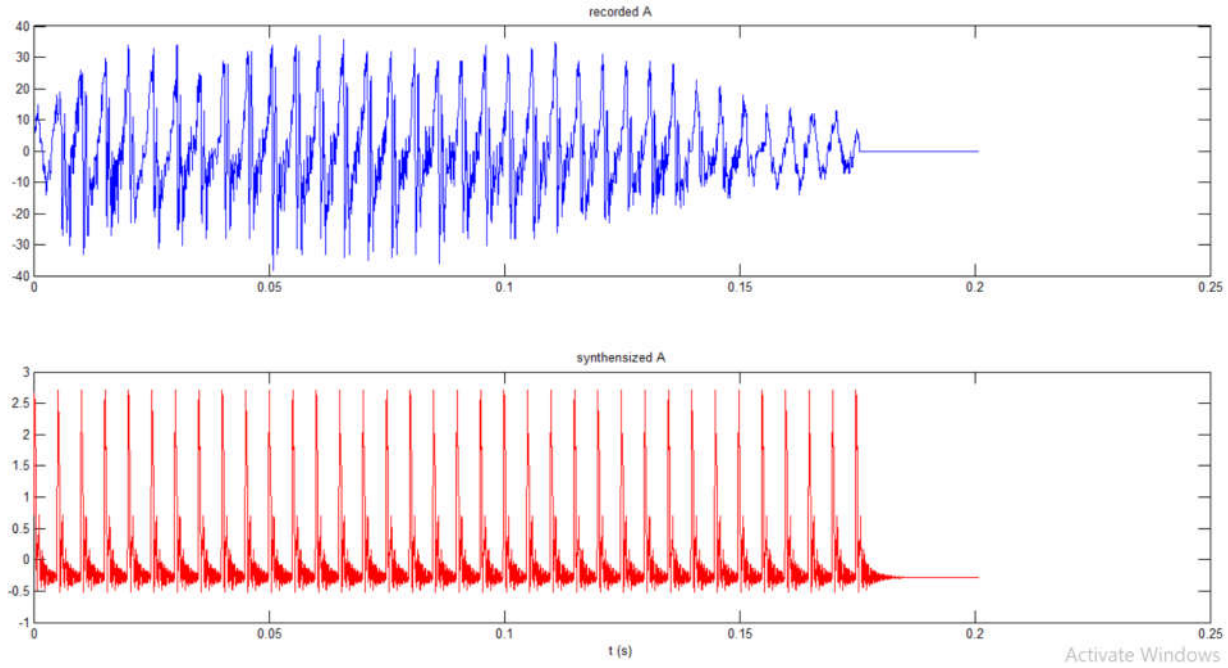

Figure 2.7 the plot of our recorded “A” and the synthesized “A” in time domain

A closer look from for better comparison is shown below in figure 2.8

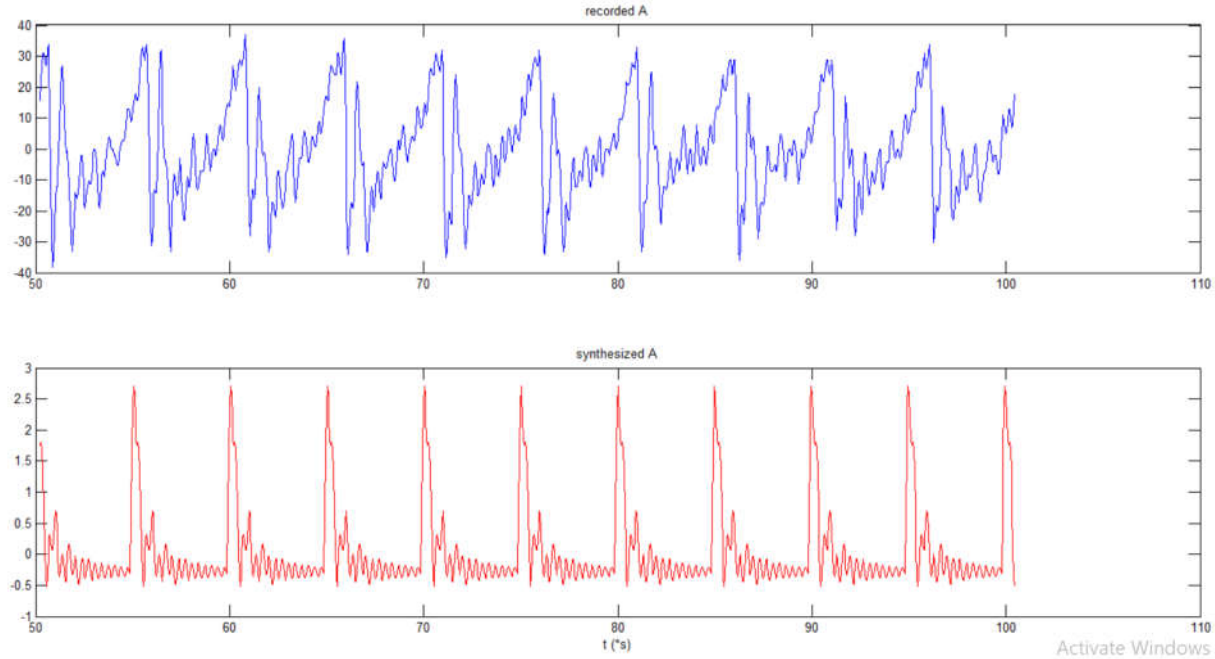

Figure 2.8 a zoomed plot of our recorded “A” and the synthesized “A” in time domain

The comparison between the recorded and the synthesized signal in the frequency domain can be seen below in figure 2.9.

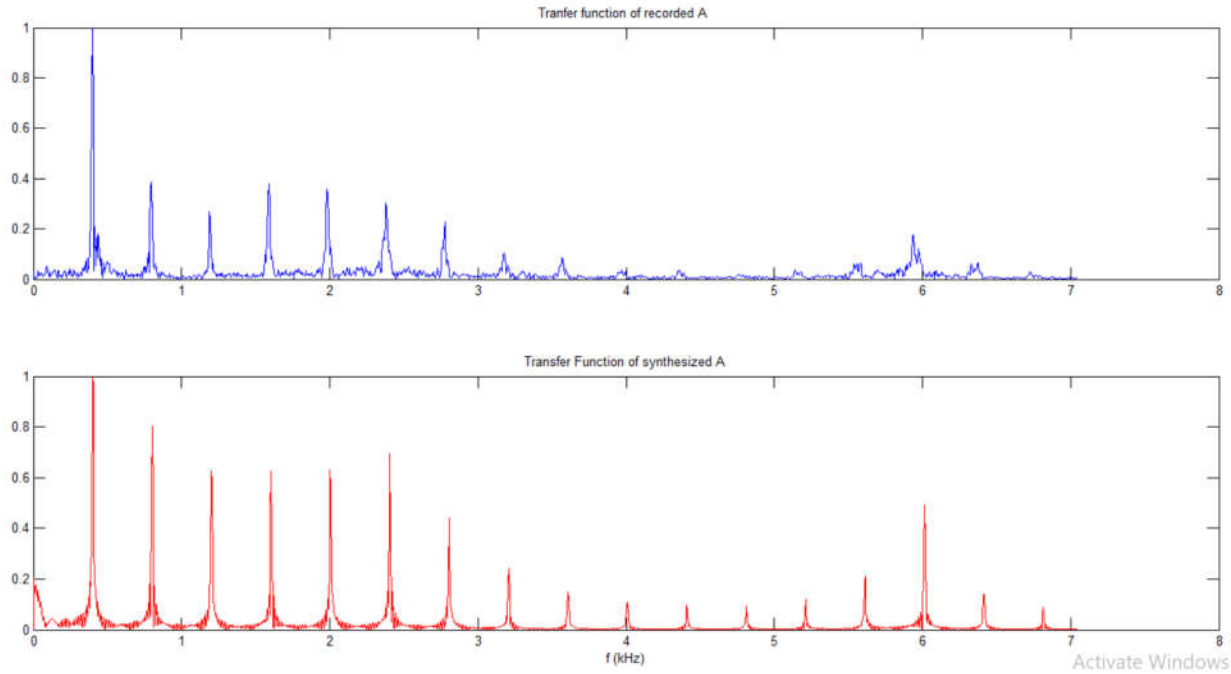

Figure 2.9 the plot of our recorded “A” and the synthesized “A” in frequency domain

It can be observed that the main component of the two signals are found in the same frequency.

2.5 Conclusion

An analysis of the parameter estimation of an AR process was carried out. A sound of alphabet ‘a’ was analyzed. From analyzing the sound we were able to derive the repetition period of the sound which correspond to the repetition period of a sound generated by a woman. We were able to conclude that the sound was recorded from a female voice. From the previous analysis, we were able to synthesize a new sound and compare it to the recorded voice. We were able to conclude that the synthetized sound bears resemblance with the recorded sound as shown in the frequency spectrum.

Appendix 1

Appendix 1.1: The code for the part on Noise generation

```
clc
clear all
close all
%%Noises - Using rand to create a vector of random variables
v = rand(1,10);   %vector v of size 10
figure
subplot(2,1,1);   %creating a subplot
plot(v);          %plot vector
title('plot of vector v of size 10')
ylabel('Amplitude')
xlabel('Number of elements in vector v')
subplot(2,1,2);
hist(v);         %use hist to plot approximate pdf
title('Approximate pdf of vector v of size 10')
ylabel('Frequency')
xlabel('Elements in vector v')
v_mean_1 = mean(v)   %mean of vector v
v_var_1 = var(v)     %variance of vector v

%repeating above vector v of size 10000
v = rand(1,10000);
figure
subplot(2,1,1);
plot(v);
title('plot of vector v of size 10000')
ylabel('Amplitude')
xlabel('Number of elements in vector v')
subplot(2,1,2);
hist(v);         %use hist to plot approximate pdf
title('Approximate pdf of vector v of size 10000')
ylabel('Frequency')
xlabel('Elements in vector v')
v_mean_2 = mean(v)    %mean of vector v
v_var_2 = var(v)      %variance of vector v

%% Using randn() to generate v for size 10
v = randn(1,10);
figure
subplot(2,1,1)
plot(v);
title('plot of vector v of size 10')
ylabel('Amplitude')
xlabel('Number of elements in vector v')
subplot(2,1,2)
hist(v);
title('Approximate pdf of vector v of size 10')
ylabel('Frequency')
xlabel('Elements in vector v')
v_mean_1 = mean(v)  %mean of vector v
v_var_1 = var(v)    %variance of vector v

%% Repeating using randn() to generate v for size 10000
```

```
v = randn(1,10000);
figure
subplot(2,1,1)
plot(v);
title('plot of vector v of size 10000')
ylabel('Amplitude')
xlabel('Number of elements in vector v')
subplot(2,1,2)
hist(v);
title('Approximate pdf of vector v of size 10000')
ylabel('Frequency')
xlabel('Elements in vector v')
v_mean_2 = mean(v)  %mean of vector v
v_var_2 = var(v)    %variance of vector v
```

Appendix 1.2: The code for the part on Central Limit Theorem

```
clc
clear all
close all
%%Central Limit Theorem
S = zeros(1,1000);   %%declaring vector S
for j = 1:1000
    N = 10;       %%value of N
    a = -1;
    b = 1;
    ni = a+(b-a)*rand(1,N);  %%generating ni for size N
    S(j) = sum(ni);          %%summing ni and storing in S
end
subplot(3,1,1)
hist(S);
title('Approximate pdf of Sj for size N = 10');
ylabel('Frequency')
xlabel('values of Sj')

%% Repeated for N = 1000
for j = 1:1000
    N = 1000;       %%value of N
    a = -1;
    b = 1;
    ni = a+(b-a)*rand(1,N);  %%generating ni for size N
    S(j) = sum(ni);          %%summing ni and storing in S
end
subplot(3,1,2)
hist(S);
title('Approximate pdf of Sj for size N = 10000');
ylabel('Frequency')
xlabel('values of Sj')

%% Repeated for size N = 100000
for j = 1:1000
    N = 100000;       %%value of N
    a = -1;
    b = 1;
    ni = a+(b-a)*rand(1,N);  %%generating ni for size N
    S(j) = sum(ni);          %%summing ni and storing in S
```

```
end
subplot(3,1,3)
hist(S);
title('Approximate pdf of Sj for size N = 100000');
ylabel('Frequency')
xlabel('values of Sj')
```

Appendix 1.3 The code for the part on Linear Estimator

```
%% generating y for various values of variance
%% y(t) = m + n(t),  t = 1.....N
m = 5;
N= 1000;
t = 1:1:N
y = zeros(N,5);
n = zeros(N,1);
i = 1;   %%counter for filling y for different values of variance
for std_d = 0.1:0.2:1
    n = std_d * randn(N,1);
    y(:,i) = m + n;
    subplot(5,1,i);
    stem(t(1:100),y(1:100,i));
    title(['Plot of y for variance = ' num2str(var(y(:,i)))])
    xlabel('time(1:100)')
    ylabel('Amplitude')
    i = i+1;
end


%% Realizing the estimator performance for different values of N
clear all
close all
clc
%% y(t) = m + n(t),  t = 1.....N
m = 5;
y = [];
n = [];
est_m = [];
std_d = 1;
i = 1; %%counter for filling y for different values of variance
for N = [10, 100, 1000, 10000, 100000]
    n = std_d * randn(N,1);
    y = m + n;
    est_m(i) = sum(y)/N; %realizing the estimator
    disp(['Estimated value for N = ' num2str(N) ' is ' num2str(est_m(i))])
    i = i+1;
end


clear all
close all
clc
%% realization of the performance of the estimator for various value of
variance
```

```
%% y(t) = m + n(t),  t = 1.....N
m = 5;
y = [];
n = [];
est_m = [];
N = 1000;
i = 1; %%counter for filling y for different values of variance
for std_d = [2, 4, 6, 8, 10]
    n = std_d * randn(N,1);
    y = m + n;
    est_m(i) = sum(y)/N;
    disp(['Estimated value for variance = ' num2str(var(y)) ' is '
num2str(est_m(i))])
    i = i+1;
end

clc
clear all
close all
%% Realizing the bias of the estimator for different values of N
%% y(t) = m + n(t) t= 1,....,N
est_m = zeros(1,1000);
b = zeros(1,10);
i = 1;
for N = 10:10000:100000
for mc = 1:1000
m = 5; %%choose m = 5
n = randn(1,N);
y = m + n;
y = y';
%plot(y)
%% Creating a Maximum Likelihood Estimator
est_m(mc) = (1/N)*sum(y);
end
exp_est_m = sum(est_m)/mc;
%%calculating the bias
b(i) = exp_est_m - m;
b(i) = abs(b(i));
i = i+1;
end
plot(10:10000:100000,b);
title('plot for the bias for different values of N')
xlabel('values of N')
ylabel('value of bias')

clc
clear all
close all
%% y(t) = m + n(t) t= 1,....,N
est_m = zeros(1,1000);
b = zeros(1,100);
i = 1;
for N = 10:1000:100000
for mc = 1:1000
m = 5; %%choose m = 5
n = randn(1,N);
```

```
y = m + n;
y = y';
%plot(y)
%% Creating a Maximum Likelihood Estimator
est_m(mc) = (1/(N-1))*sum(y);
end
exp_est_m = sum(est_m)/mc;
%%calculating the bias
b(i) = exp_est_m - m;
b(i) = abs(b(i));
i = i+1;
end
plot(10:1000:100000,b);
title('plot of bias vs N')
xlabel('values of N')
ylabel('bias')

clc
clear all
close all
%% realizing the variance of the estimator as a function of N
%% y(t) = m + n(t) t= 1,....,N
est_m = zeros(1,1000);
var_est_m = zeros(1,10);
i = 1;
for N = 10:1000:10000
for mc = 1:1000
m = 5; %%choose m = 5
n = randn(1,N);
y = m + n;
y = y';
%plot(y)
%% Creating a Maximum Likelihood Estimator
est_m(mc) = (1/(N-1))*sum(y);
end
exp_est_m = sum(est_m)/mc;
%%calculating the variance of
var_est_m(i) = (sum((est_m - exp_est_m).^2))/mc;
i = i+1;
end
subplot(2,1,1)
plot(10:1000:10000,var_est_m);
title('plot of var(m^ML), for sigma^2 = 1 constant and N changing')
ylabel('var(m^ML)')
xlabel('N')

clc
%% realizing the variance of the estimator as a function of N
%% y(t) = m + n(t) t= 1,....,N
est_m = zeros(1,1000);
p = zeros(1,10);
var_est_m = zeros(1,10);
i = 1;
N = 1000;
for v = 1:1:10
for mc = 1:1000
```

```
m = 5; %%choose m = 5
n = v .* randn(1,N);
y = m + n;
y = y';
%plot(y)
%% Creating a Maximum Likelihood Estimator
est_m(mc) = (1/(N-1))*sum(y);
end
exp_est_m = sum(est_m)/mc;
%%calculating the variance of
var_est_m(i) = (sum((est_m - exp_est_m).^2))/mc;
p(i) = var(est_m);
i = i+1;
end
subplot(2,1,2)
plot((1:1:10).^2,var_est_m);
title('plot of var(m^ML), for N=1000 constant and sigma^2 changing')
ylabel('var(m^ML)')
xlabel('sigma^2')
hold on
plot((1:1:10).^2,p);


clc
clear all
close all
%% y(t) = m + n(t) t= 1,....,N
est_m = zeros(1,1000);
var_est_m = zeros(1,10);
i = 1;
N = 1000;
CRLB = zeros(1,10);
for v = 1:1:10
for mc = 1:1000
m = 5; %%choose m = 5
n = v .* randn(1,N);
y = m + n;
y = y';
%plot(y)
%% Creating a Maximum Likelihood Estimator
est_m(mc) = (1/(N))*sum(y);
end
%% expectation of the Maximum Likelihood Estimator
exp_est_m = sum(est_m)/mc;
%%calculating the variance of
var_est_m(i) = (sum((est_m - exp_est_m).^2))/mc;
CRLB(i) = v/N;
i = i+1;

end
plot((1:1:10).^2,var_est_m);
title('Comparing the CRLB to the variance')
ylabel('var(m^ML)')
xlabel('sigma^2')
hold on
plot((1:1:10).^2,CRLB,'r-');
```

```
legend('var(m^ML)','CRLB');
```

Apendix 1.4 Code of Non Linear Estimation

```
clc
clear all
close all
%% generating and ploting y(t)
%% y (t) = a1 sin(2?f0t) + a2 sin(2?f1t) + n(t), t= 1,....,N
a1 = 1;
a2 = 1;
f0 = 0.25;
f1 = 0.4;
i = 1; %counter
for std_dev = [0.1, 0.5, 1]; %standard devviation
m = 5; %%choose m = 5
N = 1000;
t = 1:1:N;
n = std_dev .* randn(1,N);
y = (a1.*sin(2*pi*f0.*t)) + (a2.*sin(2*pi*f1.*t)) + n;
y = y';
hold on
subplot(3,1,i)

stem(y(1:1:100));
hold on
plot(y(1:1:100), 'r-');
hold off
title(['plot of y for variance = ' num2str(var(n))])
ylabel('Amplitude of Y')
xlabel('time t')
axis([1,100,-4,4])
i = i+1;
end
```

```
clc
close all
%% y (t) = a1 sin(2?f0t) + a2 sin(2?f1t) + n(t), t= 1,....,N
%% plotting the Least square criterion as a function of fo with fixed
variance

i = 1;
J1 = zeros(length(0:0.01:0.5),1);
for f0 = 0:0.01:0.5
J = (y - (a1.*sin(2*pi*f0.*t')) - (a2.*sin(2*pi*f1.*t'))).^2;
J = sum(J);
J1(i) = J;
i = i+1;
end
plot(0:0.01:0.5,J1);
title('Plot of J as a function of fo for a1 = 1 and a2 = 0')
ylabel('magnitude of J')
```

```
xlabel('values of fo')
[f0_value,J_value] = ginput(1); %picking the minimum coordinates
disp('minimum coordinate of J as picked by manually by ginput() is')
disp(['value for f0 = ' num2str(f0_value)])
disp(['at J = ' num2str(J_value)])



clc
close all
%% y (t) = a1 sin(2?f0t) + a2 sin(2?f1t) + n(t), t= 1,....,N
%% plotting J vs variance with fixed fo
f0 = 0.25;
a1 = 1;
a2 = 0;
N = 10000;
t = 1:1:N;
i=1;
var_ = zeros(length(0.1:0.1:1),1);
J1 = zeros(length(0.1:0.1:1),1);
for s_d = 0.1:0.1:1
    n1 = s_d .* n;
    y = (a1.*sin(2*pi*f0.*t)) + (a2.*sin(2*pi*f1.*t)) + n1;
    var_(i) = var(y);
    y = y';
    J = (y - (a1.*sin(2*pi*f0.*t')) - (a2.*sin(2*pi*f1.*t'))).^2;
    J1(i) = sum(J);
    i = i + 1;
end
plot(var_,J1)
title('Plot of J vs variance with fo constant')
ylabel('Amplitude of J')
xlabel('Value of variance')



%% a1 and a2 must be set to one in Least_square_criterion.m
%% a1=a2=1 and variance constant, this code plot J(fo,f1) vs fo, f1 in a 3D
plot
clc
close all
%% y (t) = a1 sin(2?f0t) + a2 sin(2?f1t) + n(t), t= 1,....,N
i = 1;
k = 1;
J1 = zeros(length(0:0.01:0.5),length(0:0.01:0.5));
for f1 = 0:0.01:0.5
    for f0 = 0:0.01:0.5
        J = (y - (a1.*sin(2*pi*f0.*t')) - (a2.*sin(2*pi*f1.*t'))).^2;
        J1(i,k) = sum(J);
        k = k+1;
    end
    i = i+1;
    k=1;
end
surf(0:0.01:0.5,0:0.01:0.5,J1);
title('Plot of J(fo,f1) vs fo, fi')
xlabel('values of fo')
ylabel('values of f1')
```

```
zlabel('Amplitude of J(fo,f1)')
%[f0_value,f1_value,J_value] = ginput(1) %picking the minimum coordinates
%plot3(0:0.01:0.5,0:0.01:0.5,J1);
```

Appendix 2

Appendix 2.1 Code on the part; Parameters estimation of an AR process

```
%% plotting a and selecting the useful part into variable A
close all
subplot(2,1,1)
plot(a)
title('plot of the data of "a"')
ylabel('Amplitude')
xlabel('time')
A = a(1:3930); %%the useful part of a
subplot(2,1,2)
plot(A)
title('plot of A (useful part of "a") ')
ylabel('Amplitude')
xlabel('time')
```

```
clc
%% calculating the roots of the polynomial
rt = roots(Den);
plot(real(rt), imag(rt), '*')
title('Plot of roots in complex plane');
ylabel('Imaginary');
xlabel('Real');

%% calculating the modulus and phases of rt
modulus = abs(rt);
ang = angle(rt);
```

```
clc
%% calculating the frequency Fi
Fi = (1/(2*pi))* Fe .* ang;
disp('Frequencies = ')
disp(Fi)
```

```
%% plotting the frequency and impulse response of H(z)
close all
clc
freqz(rt)
title('Frequency response of H(z)');
figure
impz(rt)
```

Appendix 2.2 Code on the part; Analysis of the Sound

```
clc
%% calculating the fourier trasnform of the signal A and plotting it
L = 3930;   %length of the signal
NFFT = 2^nextpow2(L); % Next power of 2 from length of y
Y = fft(A,NFFT)/L;
f = Fe/2*linspace(0,1,NFFT/2+1);

% Plot single-sided amplitude spectrum.
plot(f,2*abs(Y(1:NFFT/2+1)))
title('Single-Sided Amplitude Spectrum of A')
xlabel('Frequency (Hz)')
ylabel('Magnitude of the FFT of A')
```

Appendix 2.3 Code on the part; Synthesis of the Sound

```
N_TF=8192;
Te = 1/Fe;
T = 5e-3;
s = A;
Num = 1;
e=zeros(1,length(s)+500);
e(1:round(T/Te):length(s))=1;  %%input signal
t=(1:length(e))*Te;
s=[s zeros(1,length(t)-length(s))];
s_synthe=filter(Num,Den,e); s_synthe=s_synthe-mean(s_synthe);
TF_s_synthe=fft(s_synthe,N_TF);
figure(1);
subplot(211); plot(t,s,'b'); title('recorded A')
subplot(212); plot(t,e,'g'); title('input')
xlabel('t (s)'); figure(2);
subplot(211); plot(t,e,'g'); title('input')
subplot(212); plot(t,s_synthe,'r'); title('synthensized A');
xlabel('t (s)'); figure(3);
subplot(211); plot(t,s,'b'); title('recorded A')
subplot(212); plot(t,s_synthe,'r'); title('synthensized A');
xlabel('t (s)'); figure(4);
z=round(length(s)/4):round(length(s)/2);
subplot(211); plot(t(z)*1e3,s(z),'b'); title('recorded A')
subplot(212); plot(t(z)*1e3,s_synthe(z),'r'); title('synthesized A');
xlabel('t (*s)'); figure(5);
n=round(N/3);
TF_s=fft(s,N_TF); TF_s=TF_s(1:N_TF/2);
subplot(211); plot(f(1:n)*1e-3,abs(TF_s(1:n))/max(abs(TF_s)),'b');
title('Tranfer function of recorded A ');
subplot(212); plot(f(1:n)*1e-
3,abs(TF_s_synthe(1:n))/max(abs(TF_s_synthe)),'r');
title('Transfer Function of synthesized A');
xlabel('f (kHz)');
```