Department of Electronic & Telecommunication Engineering

University of Moratuwa

EN 2040 – Random Signals and Processes

Simulation Assignment

Name	**Index number**
Premathilaka H. D. M.	180497C

This report is submitted in partial fulfillment of the requirements for the module EN 2040 – Random Signals and Processes.

Date of submission: 16/09/2021

Question 01

- Taking $\Pr(D = 0) = \Pr(D = 1) = 0.5$

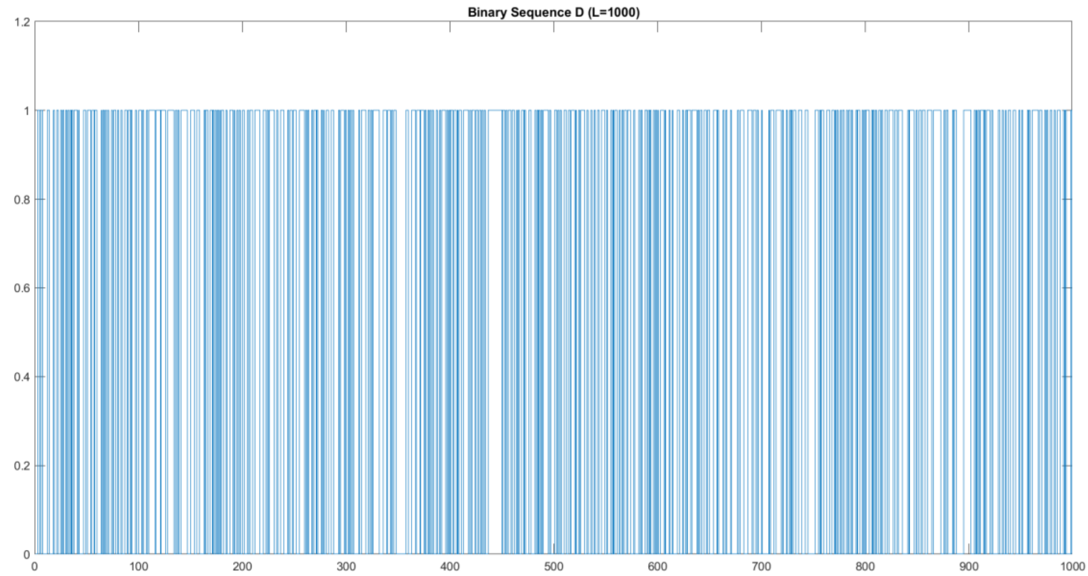

Figure 1 - Binary Sequence (L = 1000)

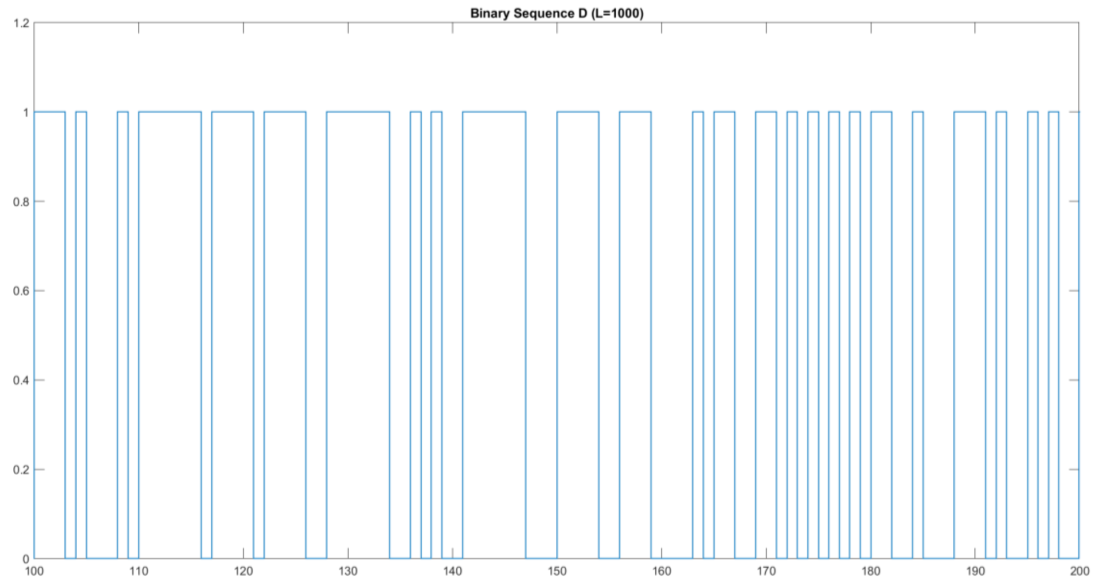

Figure 2 - Binary Sequence (from 100th to 200th divisions)

- The transmitted signal S is defined as;

$$S = \begin{cases} +A & if\ D = 1 \\ -A & if\ D = 0 \end{cases}$$

- Taking A = 1

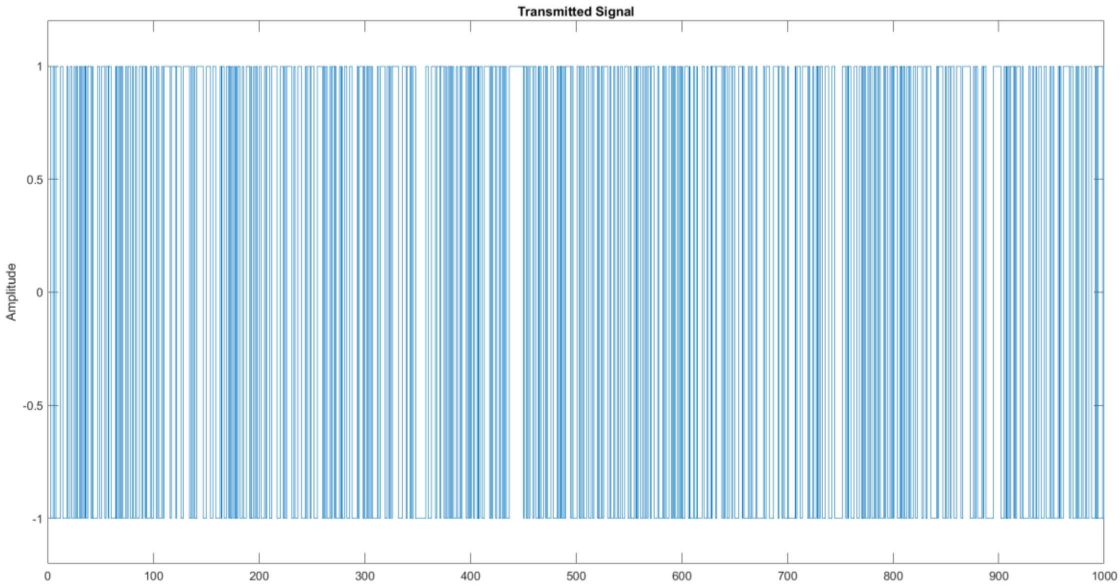

Figure 3 - Transmitted Signal (L =1000)

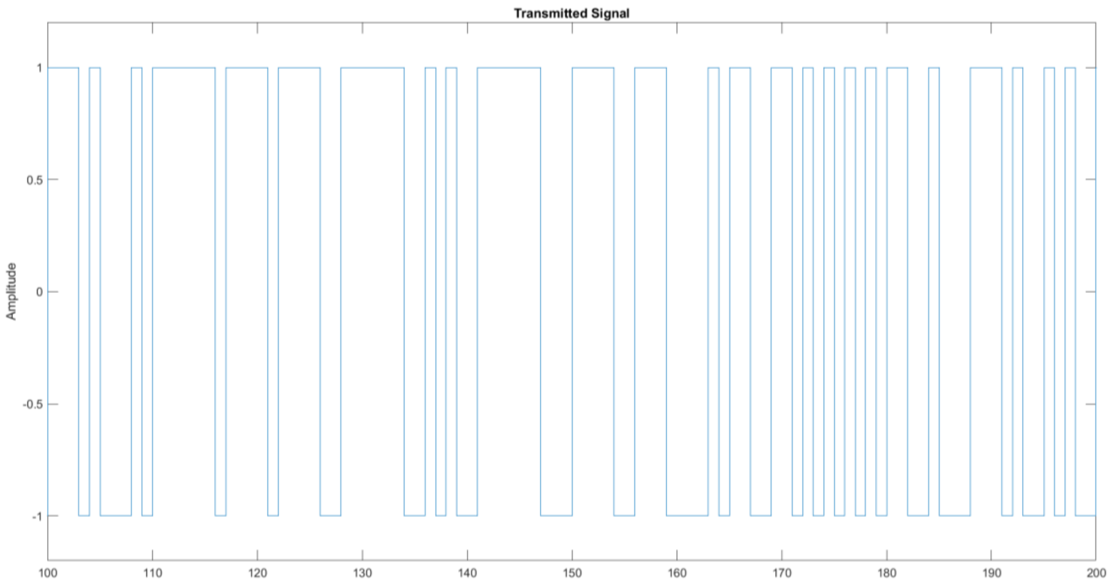

Figure 4 - Transmitted Signal (from 100th to 200th division)

Question 02

- Generating AWGN of length L = 1000 and $\sigma^2 = 1$

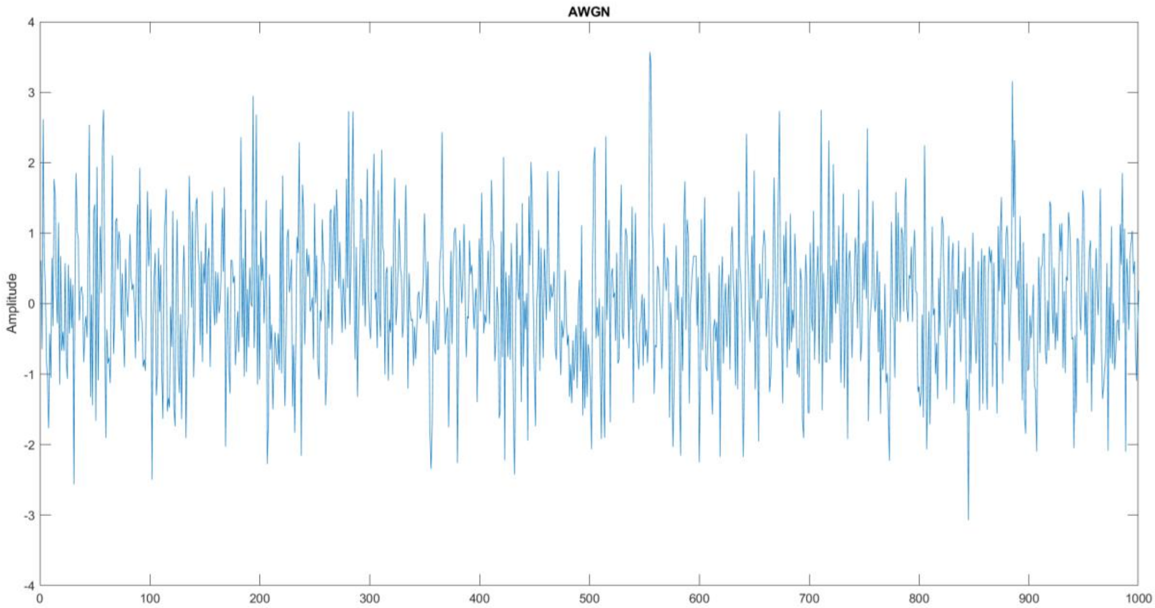

Figure 5 - AWGN (L = 1000 and variance =1)

Question 03

- The received signal R is given by;

$$R = S + N$$

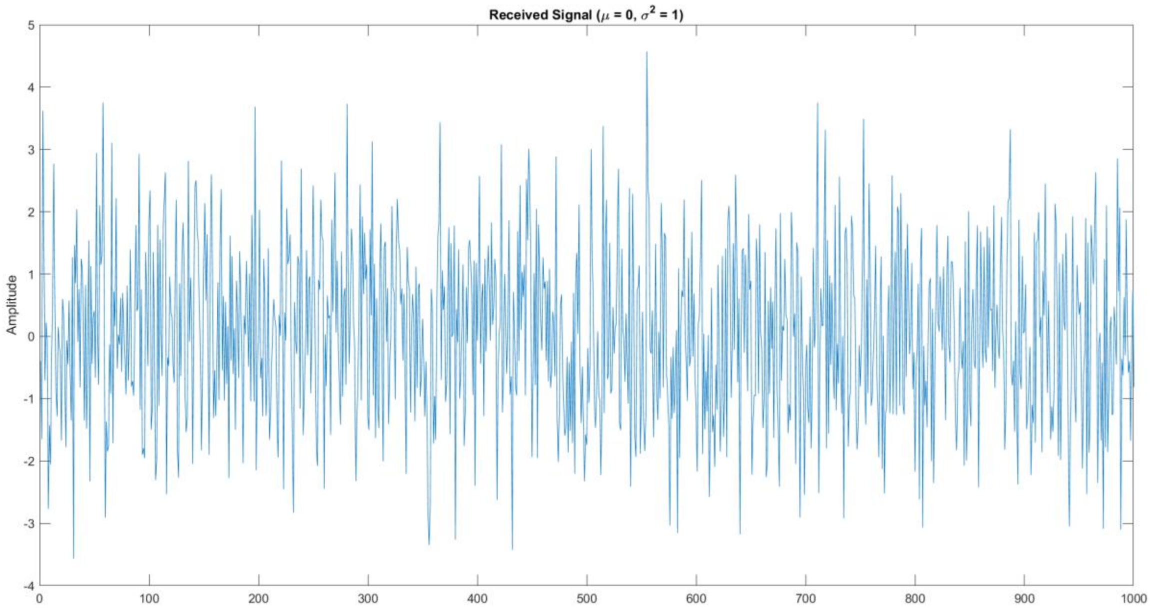

Figure 6 - Received Signal - (Variance of noise = 1)

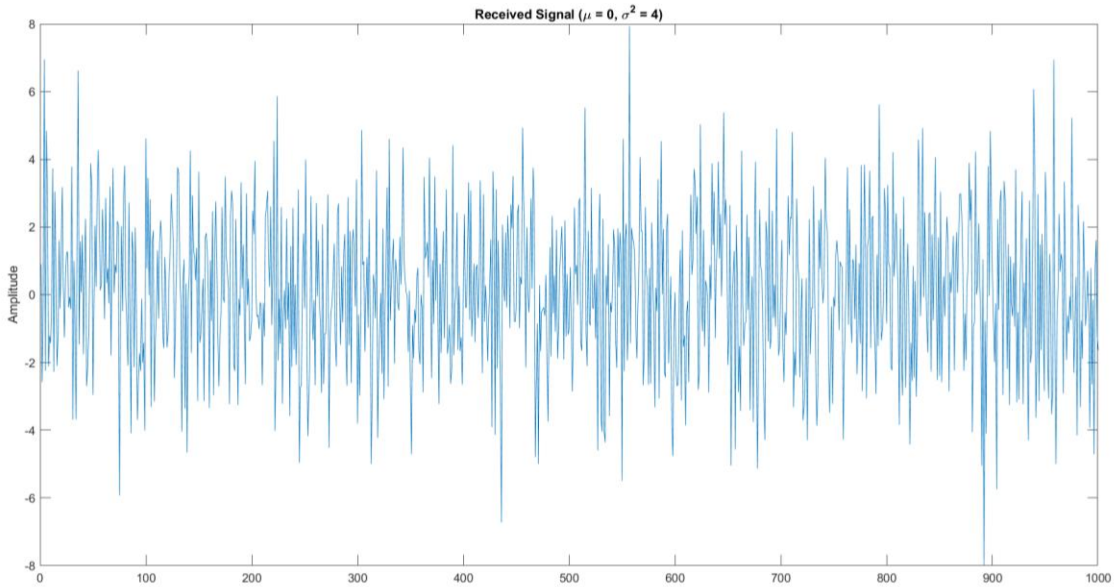

Figure 7 - Received Signal (Variance of noise = 4)

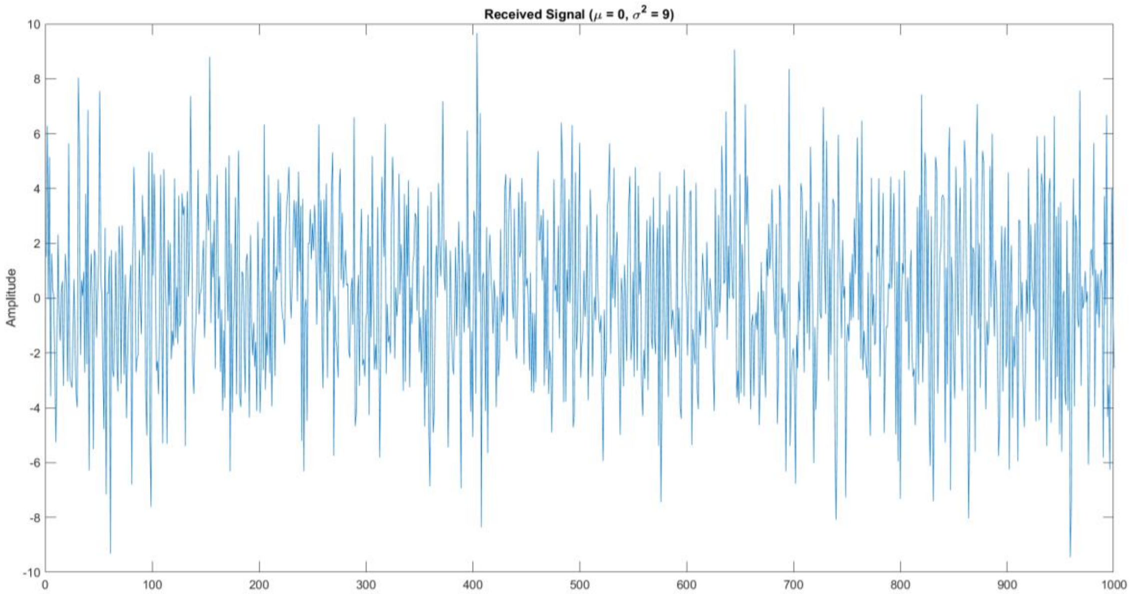

Figure 8 - Received signal (Variance of noise = 9)

- As the variance of noise increases, the range ($\approx\ \pm 3\sigma$) of values that the noise can take, increases.
- As a result, the amplitudes of the received signal R increase when the variance of noise is increased.

Question 04

- The received signal R is decoded to produce the output signal Y. (Thresholding R)

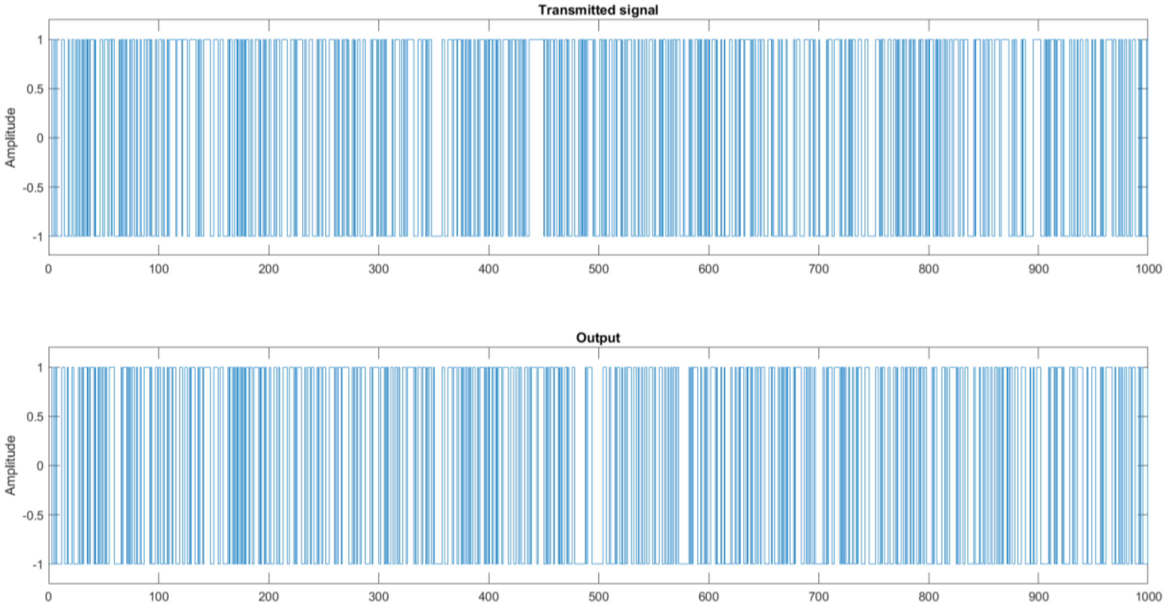

Figure 9 - Comparing the transmitted and output signal

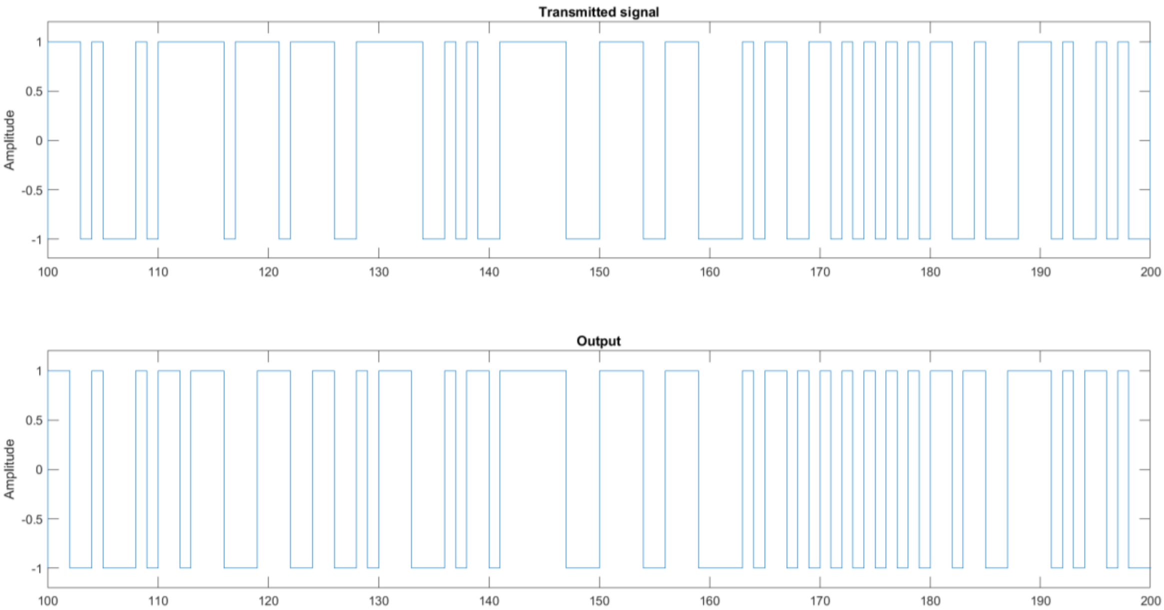

Figure 10 - Comparison (100th to 200th divisions)

$$Error\ percentage = \frac{No.of\ error\ bits}{Total\ length\ of\ the\ sequence} \times 100\%$$

$$Error\ percentage = 16.30\%$$

Question 05

- Now, the length of the sequence is increased to 100000.
- Steps in Q1-Q4 were repeated and the received signal is as follows.

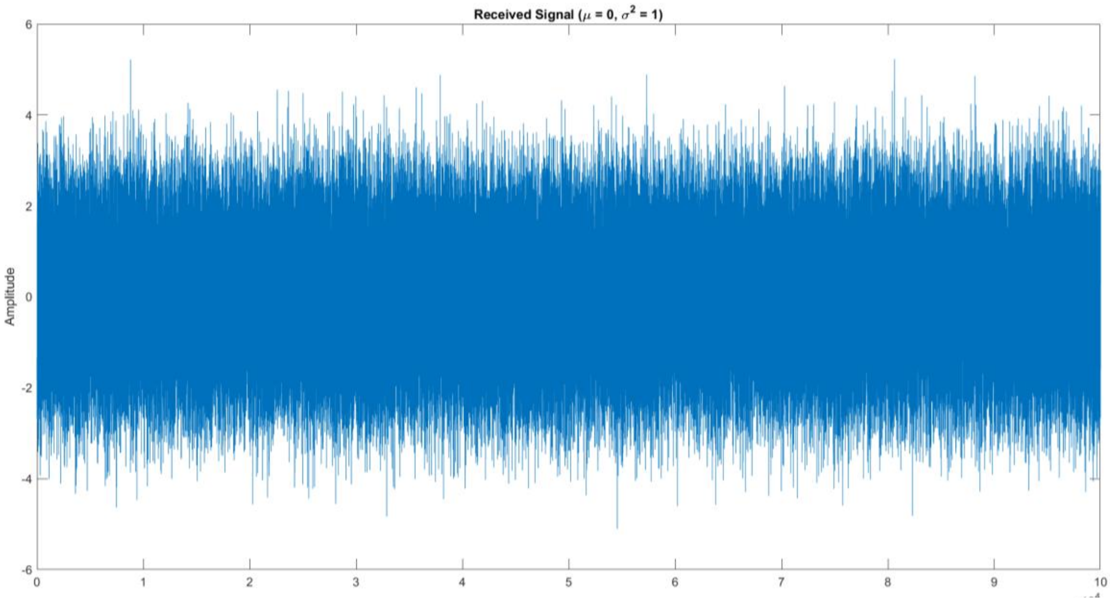

Figure 11 - Received Signal (L = 100000)

- To notice the differences between the transmitted signal and the output signal, we will consider the segment from the 10000th division to the 10100th division.

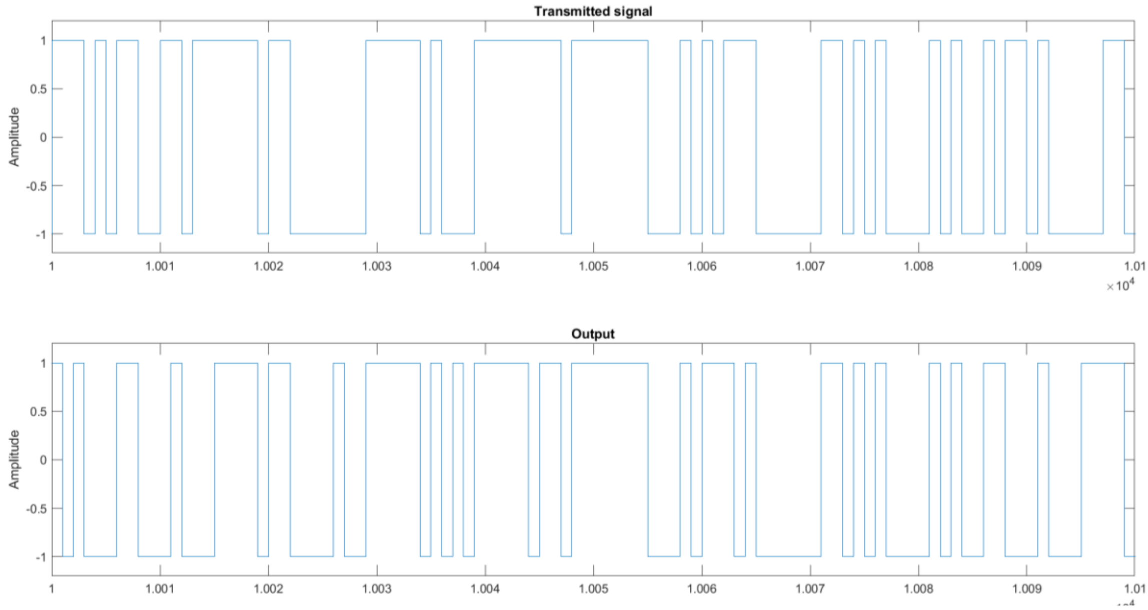

Figure 12 - Comparison between the transmitter signal and the output signal

$$Error\ percentage = \frac{No.of\ error\ bits}{Total\ length\ of\ the\ sequence} \times 100\%$$

$$Error\ percentage = 16.21\%$$

- The histogram generated (without using the built-in function **hist** in MATLAB) for the received signal is shown below. The no. of bins = 10.

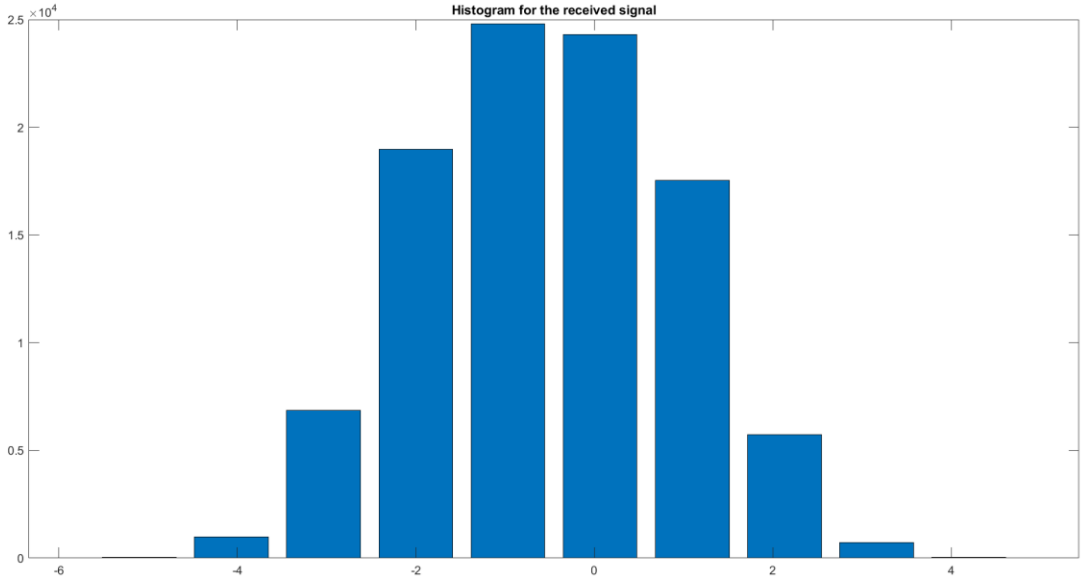

Figure 13 - Histogram for the received signal R

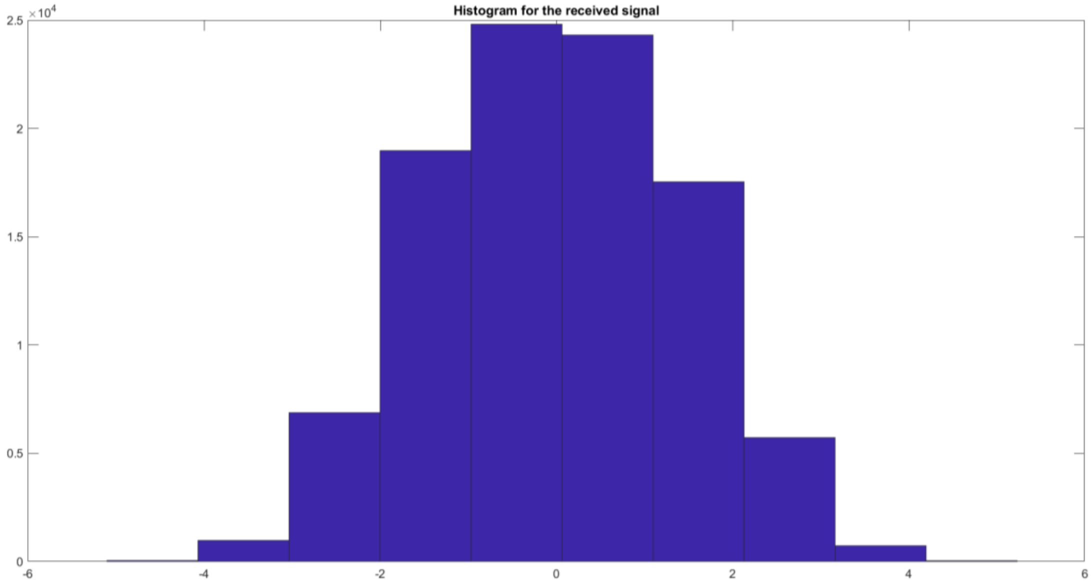

Figure 14 - Histogram of the received signal (Using the built-in 'hist' function)

- We can clearly observe that the two representations are nearly identical. In other words, the bars and the corresponding columns have nearly equal heights.

Question 05 (a)

- Let's plot the histograms for the received signal after increasing the no. of bins to 100.

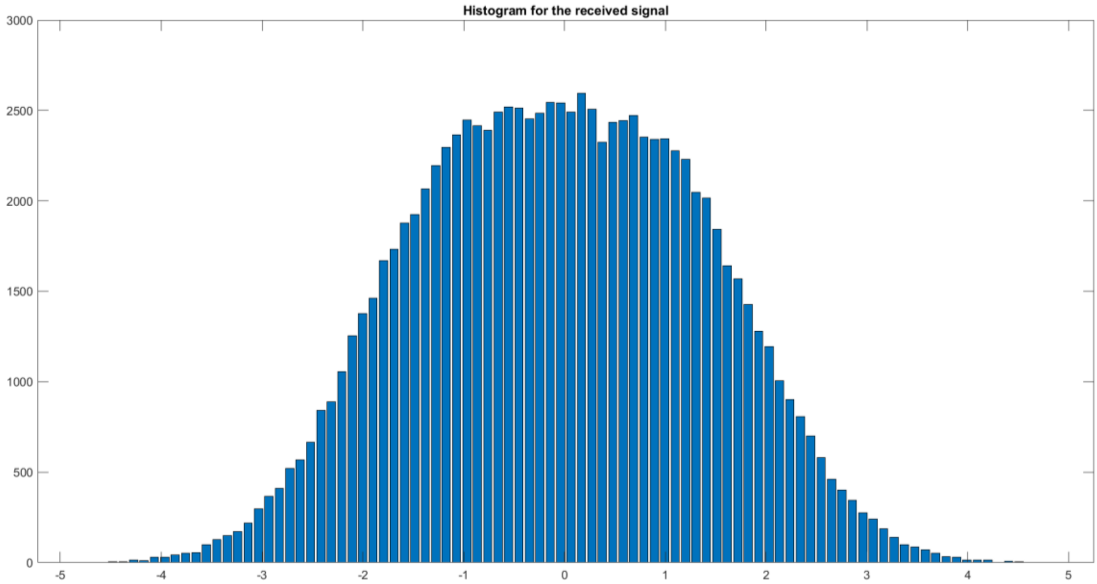

Figure 15 - Histogram for the received signal (bins = 100)

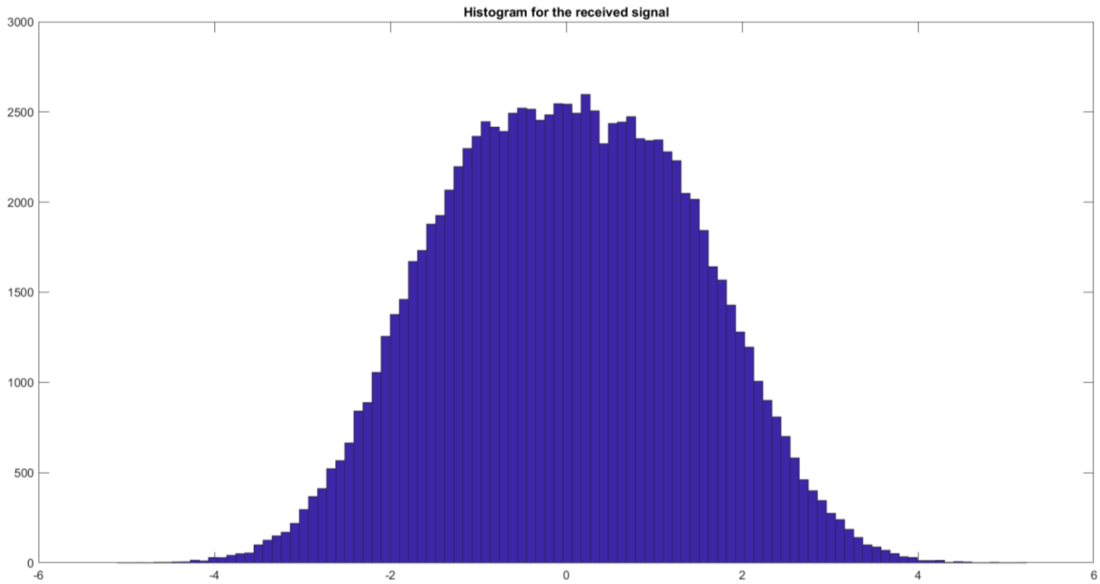

Figure 16 - Histogram for the received signal (bins = 100) using 'hist' function in MATLAB

- Similar to the previous case, both representations look identical. Moreover, as we increase the no. of bins, the outline of the columns/bars trace a curve similar to the normal distribution curve. The outline curve is given as a supplementary plot.

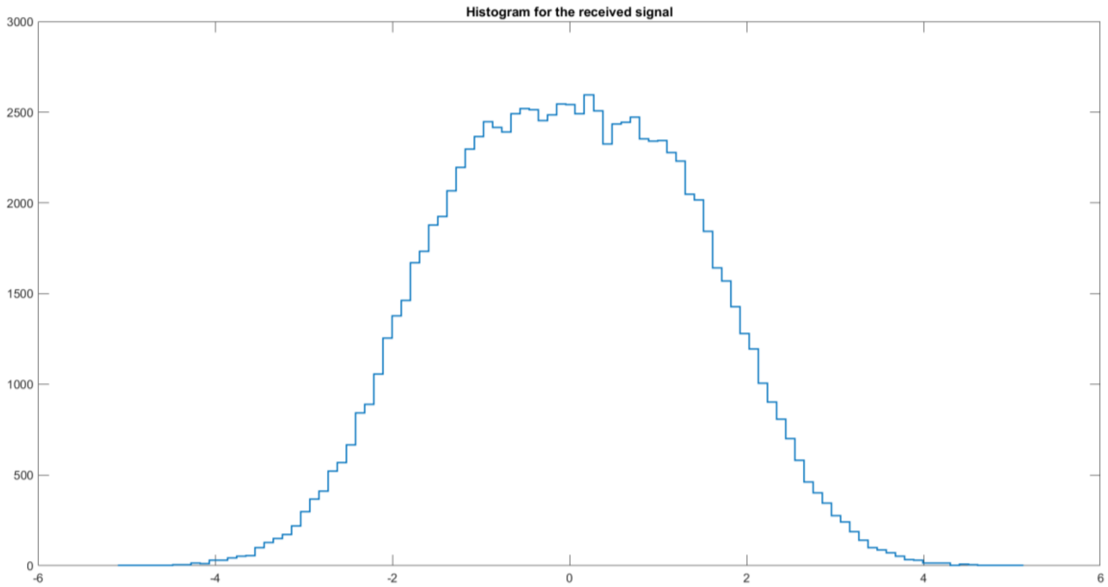

Figure 17 - Outline curve

Question 05 (b)

- We will use 100 bins in our histogram so that the histogram emphasizes the Gaussian nature of the signals.
- $Case\ I - f_{R|S}(r|S = A)\ and\ A = 1$

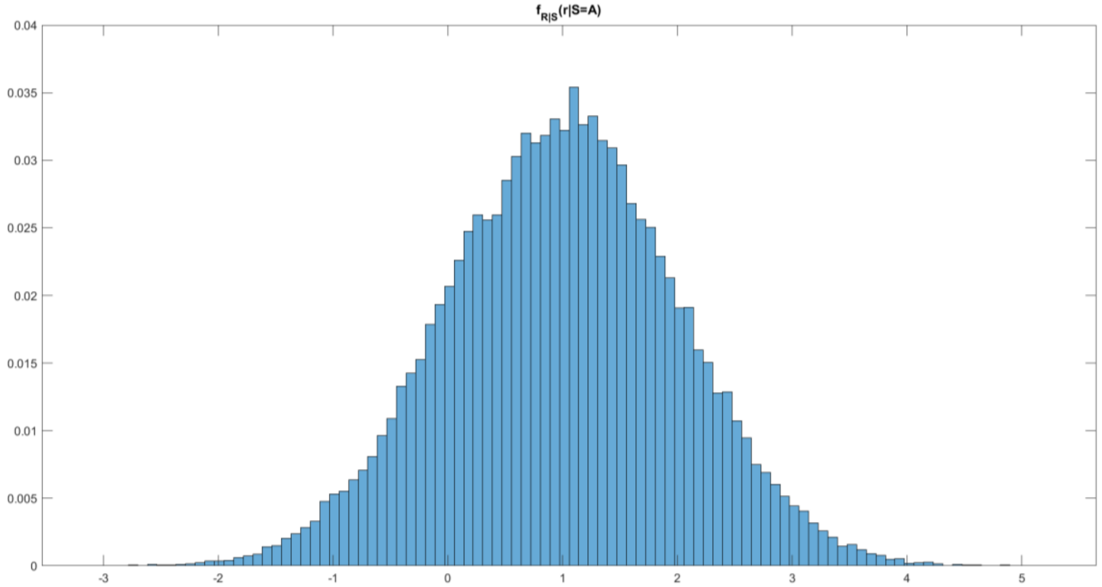

Figure 18 - f_(R|S) (r|S=A) and A=1

- $Case\ II - f_{R|S}(r|S = -A)\ and\ A = 1$

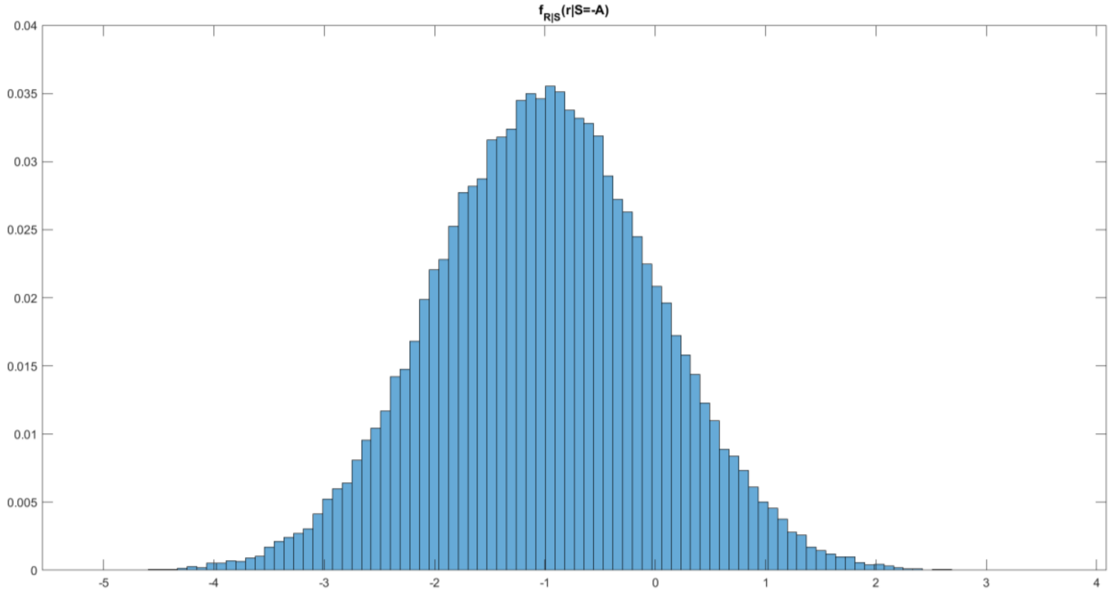

Figure 19 - f_(R|S) (r|S=-A) and A=1

- $Case\ III - f_{R|S}(r|S = A)\ and\ A = 2$

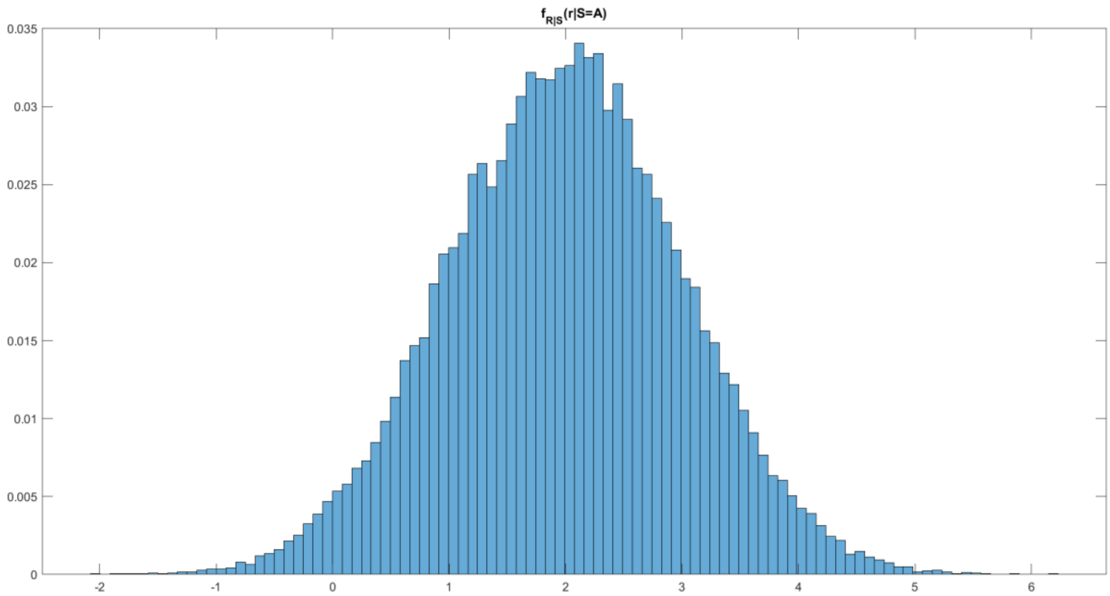

Figure 20 - f_(R|S) (r|S=A) and A=2

- $Case\ IV - f_{R|S}(r|S=-A)\ and\ A=2$

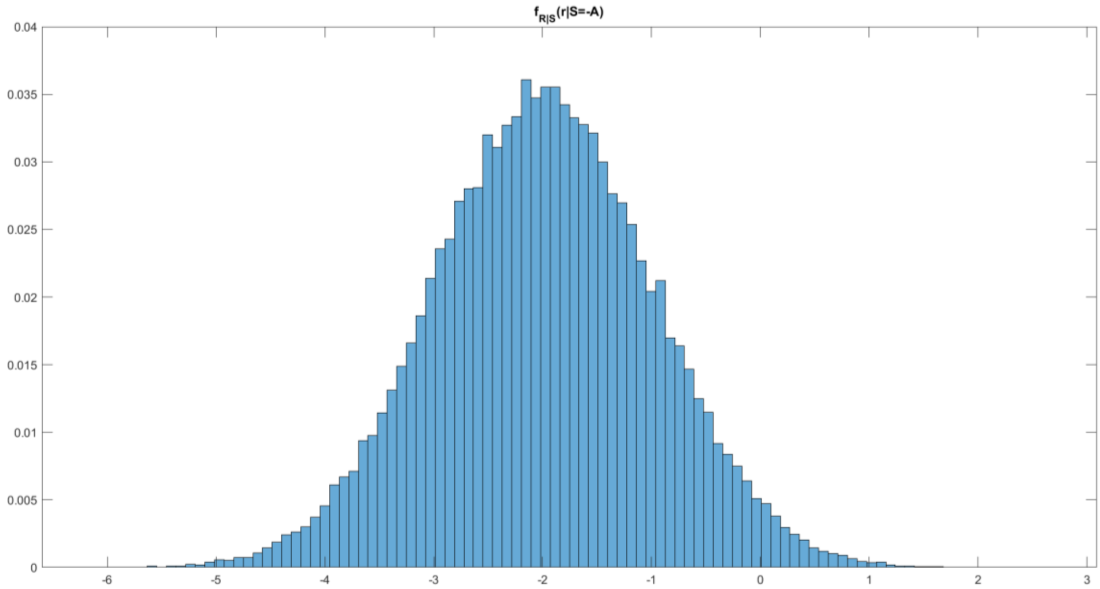

Figure 21 - f_(R|S) (r|S=-A) and A=2

- $Case\ V - f_{R|S}(r|S=A)\ and\ A=3$

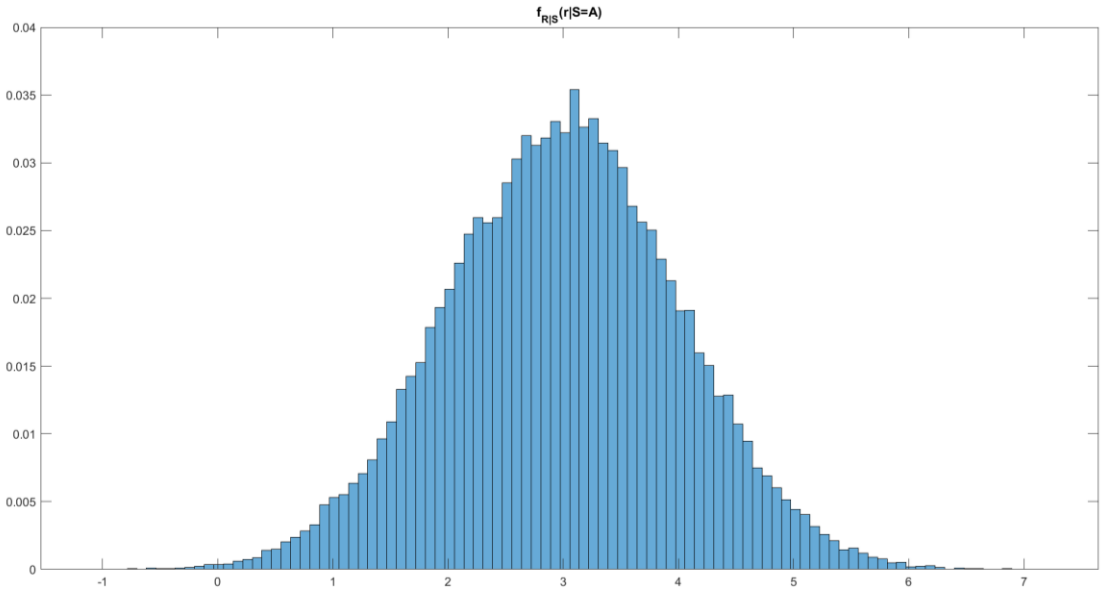

Figure 22 - f_(R|S) (r|S=A) and A=3

- $Case\ VI - f_{R|S}(r|S = -A)\ and\ A = 3$

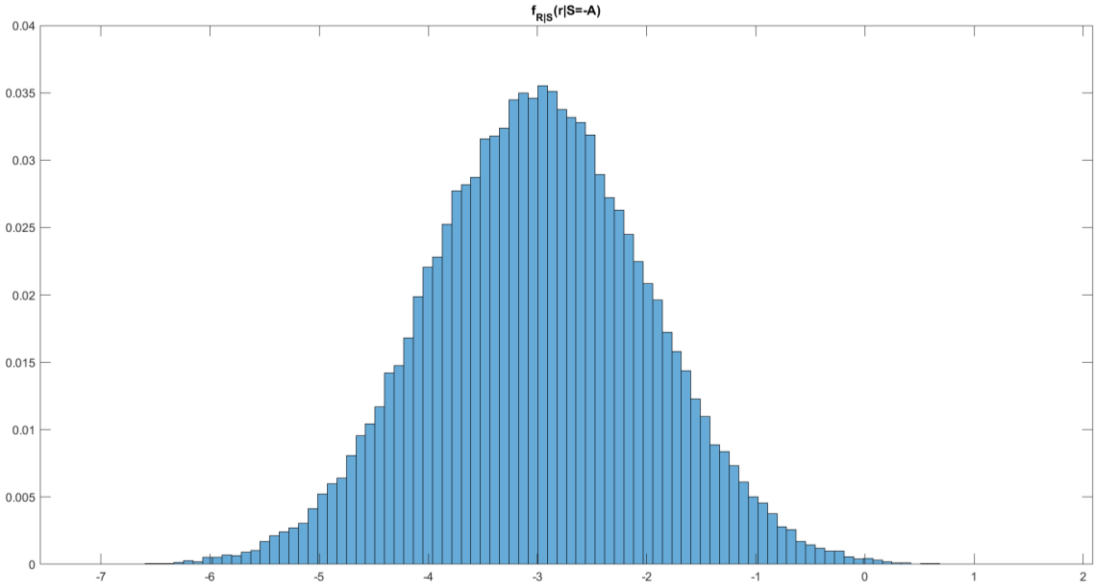

Figure 23 - f_(R|S) (r|S=-A) and A=3

- The span of the bell-shaped distribution increases when we increase the value of A from 1 to 3.

Question 05 (c)

- The expected value of a continuous random variable is given by;

$$E[X] = \int_{-\infty}^{\infty} x f_X(x) dx$$

- The expected value of a discrete random variable is given by;

$$E[X] = \sum_{-\infty}^{\infty} x P(X = x)$$

A	E[R|S=A]	E[R|S=-A]	E[R]
1	0.9945 (≈ 1)	-0.9947 (≈ -1)	0.0000052 (≈ 0)
2	1.9944 (≈ 2)	-1.9946 (≈ -2)	0.0001875 (≈ 0)
3	2.9945 (≈ 3)	-2.9947 (≈ -3)	-0.00003225 (≈ 0)

Question 05 (d)

- $Case\ I - f_R(r)\ and\ A = 1$

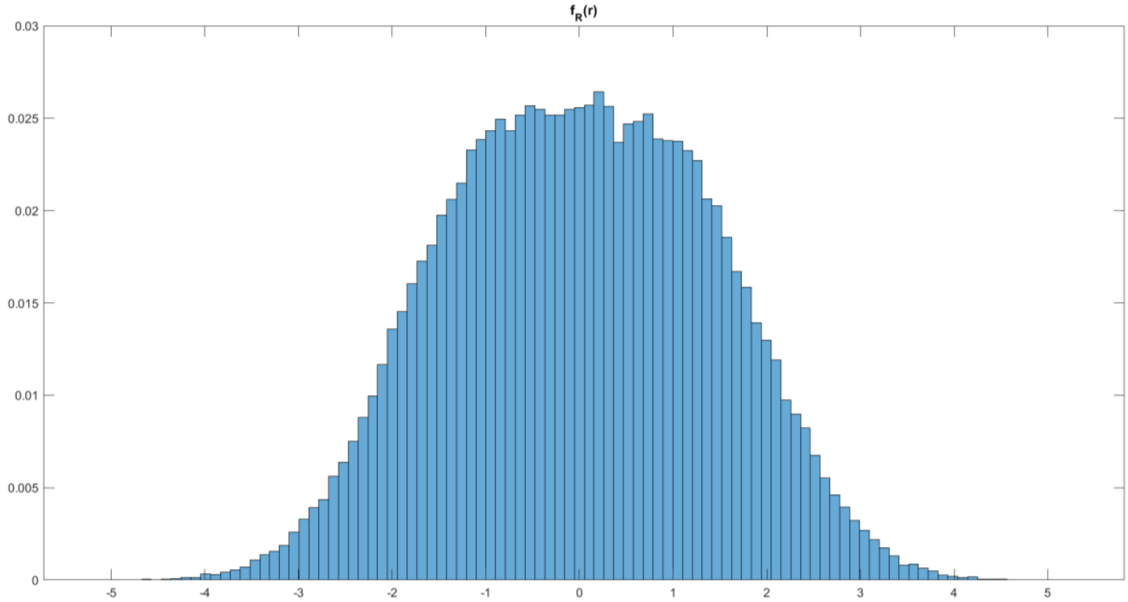

Figure 24 - f_R (r) and A=1

- $Case\ II - f_R(r)\ and\ A = 2$

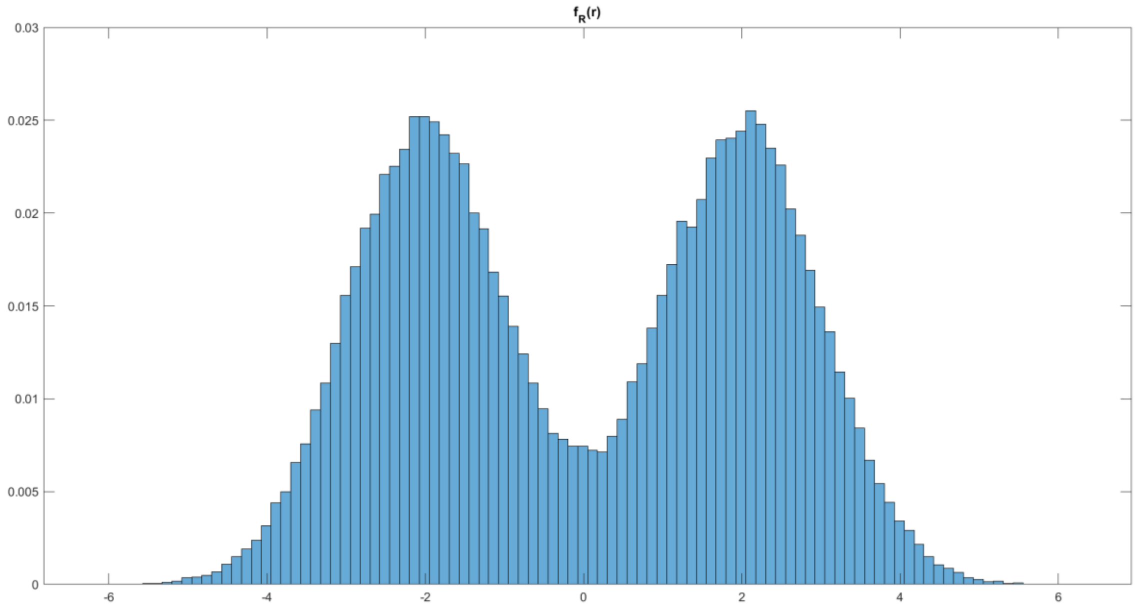

Figure 25 - f_R (r) and A=2

- $Case\ III - f_R(r)\ and\ A = 3$

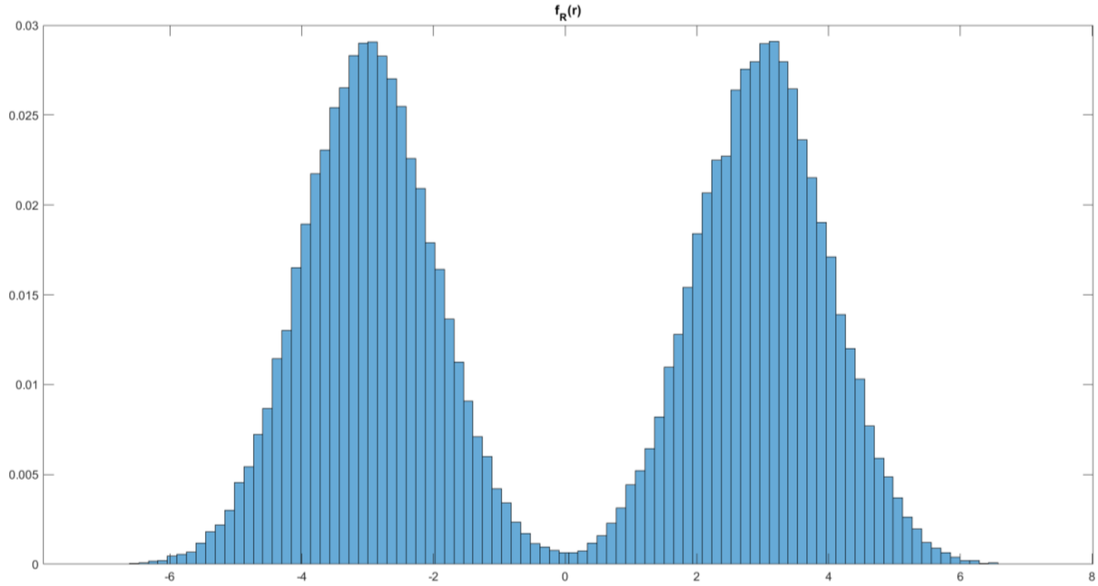

Figure 26 - f_R (r) and A=3

Question 06

- Taking the interference from other transmitters into consideration, we can write an expression for the received signal R.

$$R = S + N + I$$

- It is given that 'I' obeys a Gaussian distribution with zero mean and unit variance.

Discussion:

- Let's compare the error percentages for the two cases; with/ without I. When A = 1, we found out earlier that the error percentage is 16.21% (without interference).
- Now, when we calculate the error percentage of the received signal (with interference), we get 24.05%. (Calculation performed in the code).
- Thus, it is clear that the interference, just like noise, corrupts the transmitted signal.
- Thus, necessary measures must be taken to mitigate the adverse effects caused by interference.

- $Case\ I - f_{R|S}(r|S = A)\ and\ A = 1$

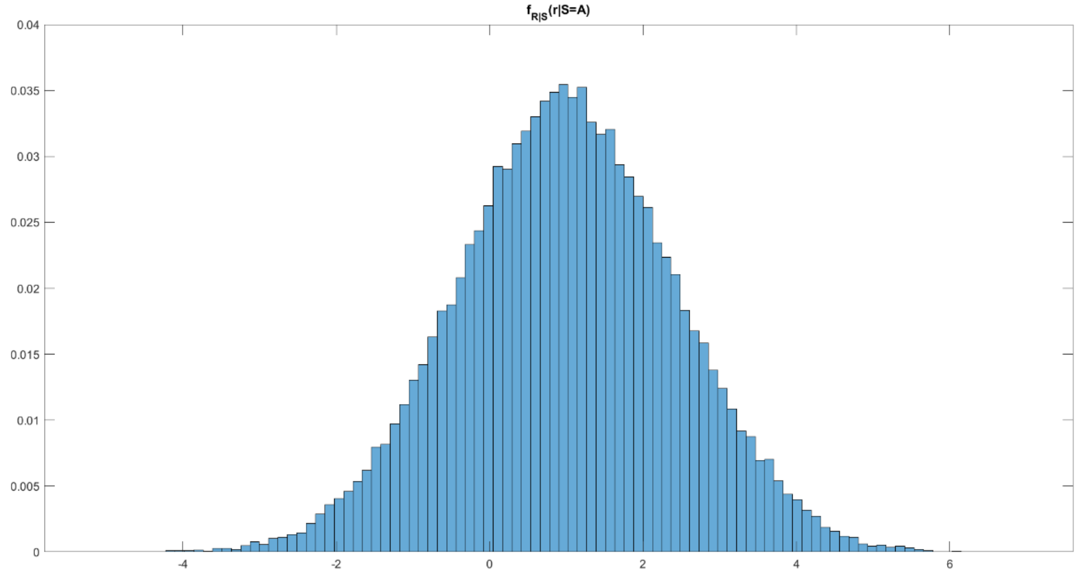

Figure 27 - f_(R|S) (r|S=A) and A=1

- $Case\ II - f_{R|S}(r|S = -A)\ and\ A = 1$

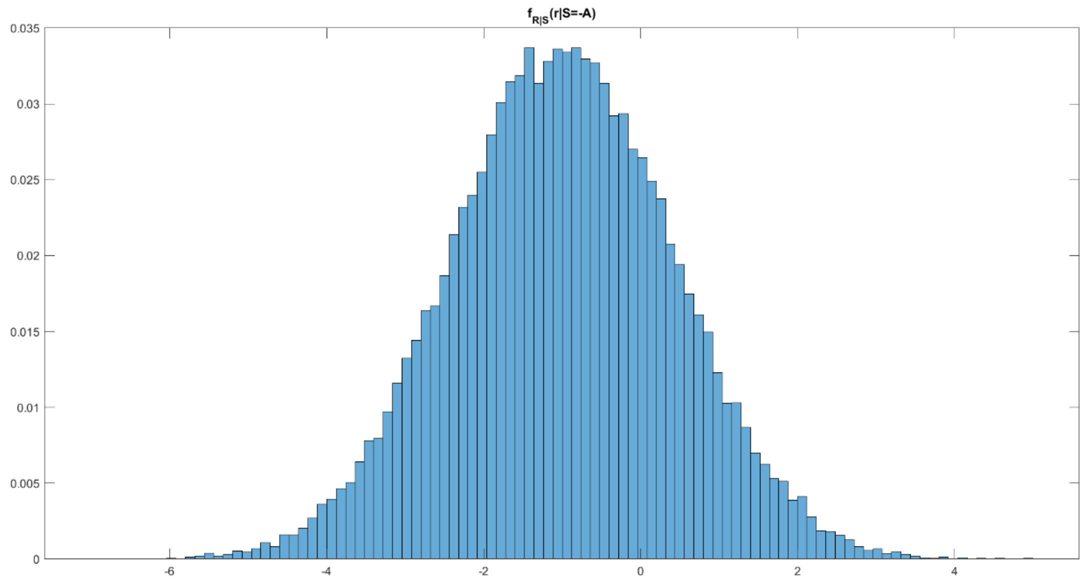

Figure 28 - f_(R|S) (r|S=-A) and A=1

- $Case\ III - f_{R|S}(r|S = A)\ and\ A = 2$

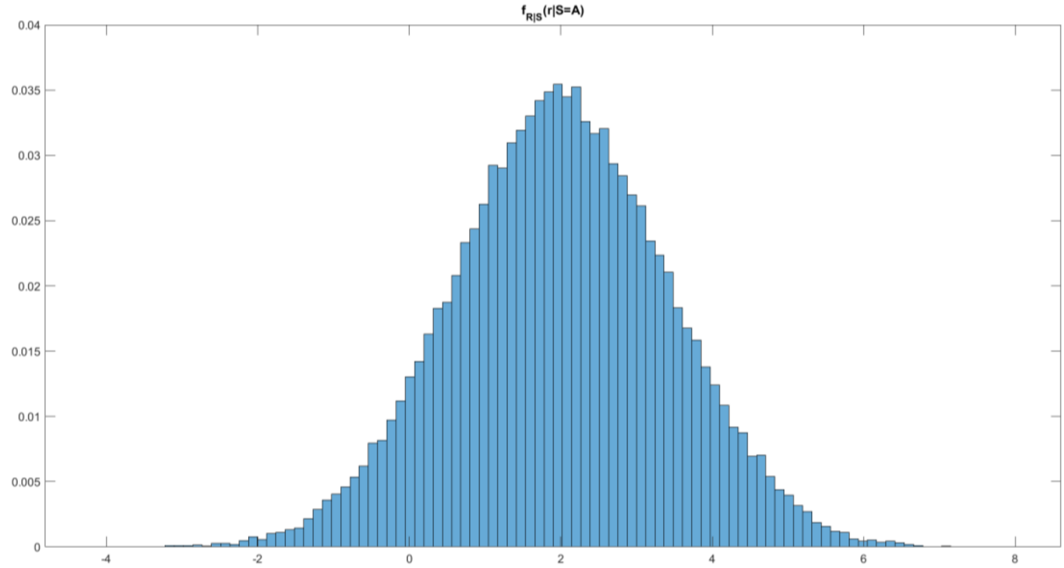

Figure 29 - f_(R|S) (r|S=A) and A=2

- $Case\ IV - f_{R|S}(r|S = -A)\ and\ A = 2$

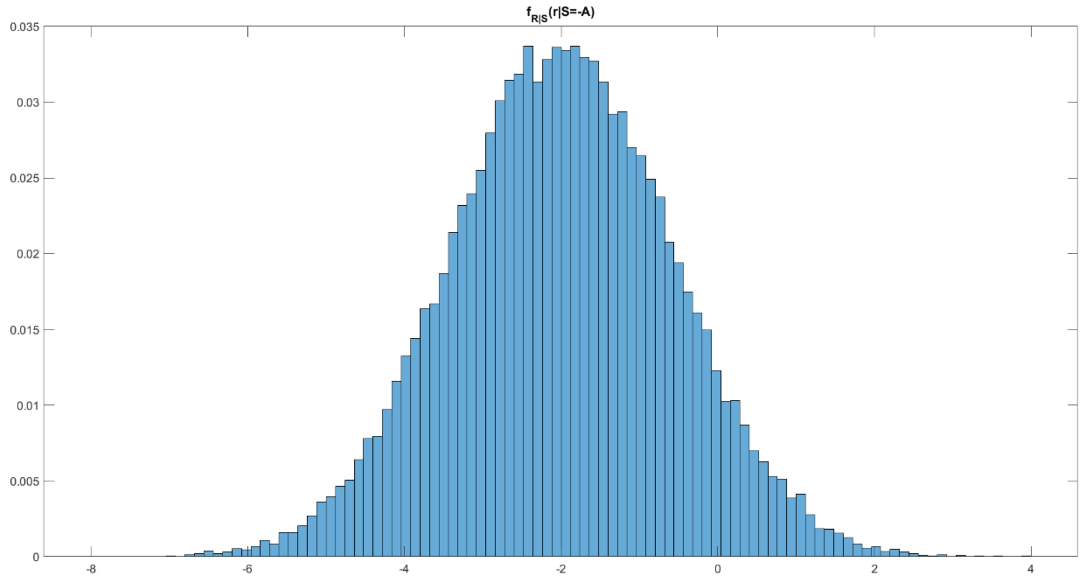

Figure 30 - f_(R|S) (r|S=-A) and A=2

- $Case\ V - f_{R|S}(r|S = A)\ and\ A = 3$

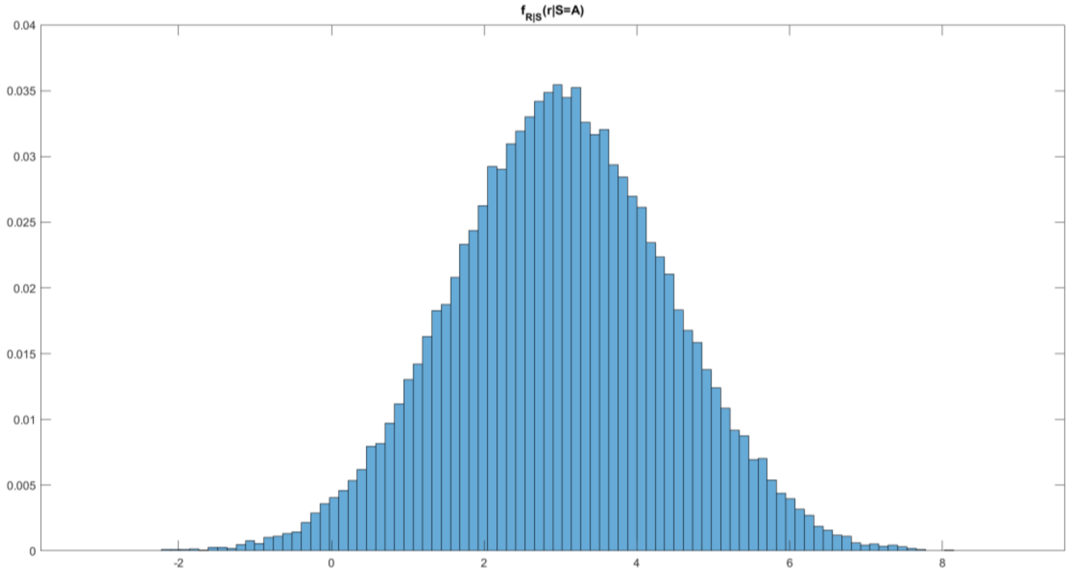

Figure 31 - f_(R|S) (r|S=A) and A=3

- $Case\ VI - f_{R|S}(r|S = -A)\ and\ A = 3$

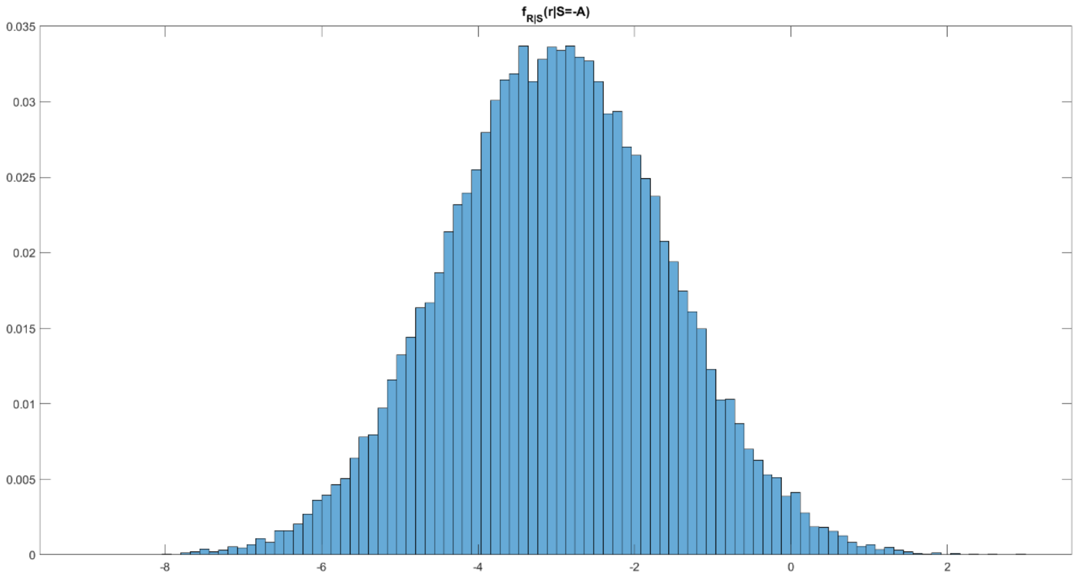

Figure 32 - f_(R|S) (r|S=-A) and A=3

A	E[R|S=A]	E[R|S=-A]	E[R]
1	1.0003 (≈ 1)	-0.9972 (≈ -1)	0.0013 (≈ 0)
2	2.0003 (≈ 2)	-1.9972 (≈ -2)	0.0015 (≈ 0)
3	3.0003 (≈ 3)	-2.9972 (≈ -3)	0.0015 (≈ 0)

- $Case\ I - f_R(r)\ and\ A = 1$

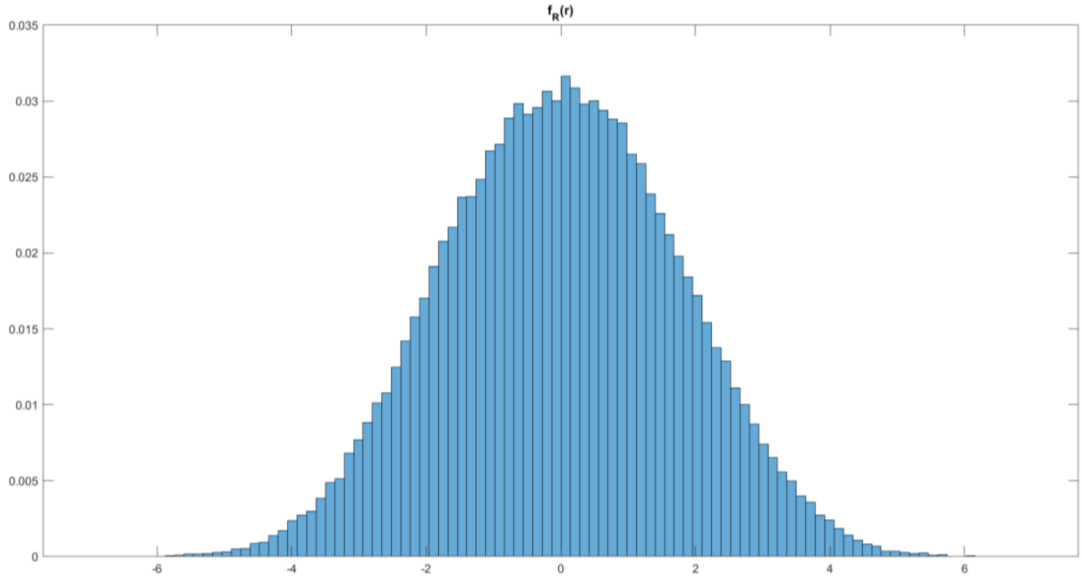

Figure 33 - f_R (r) and A=1

- $Case\ II - f_R(r)\ and\ A = 2$

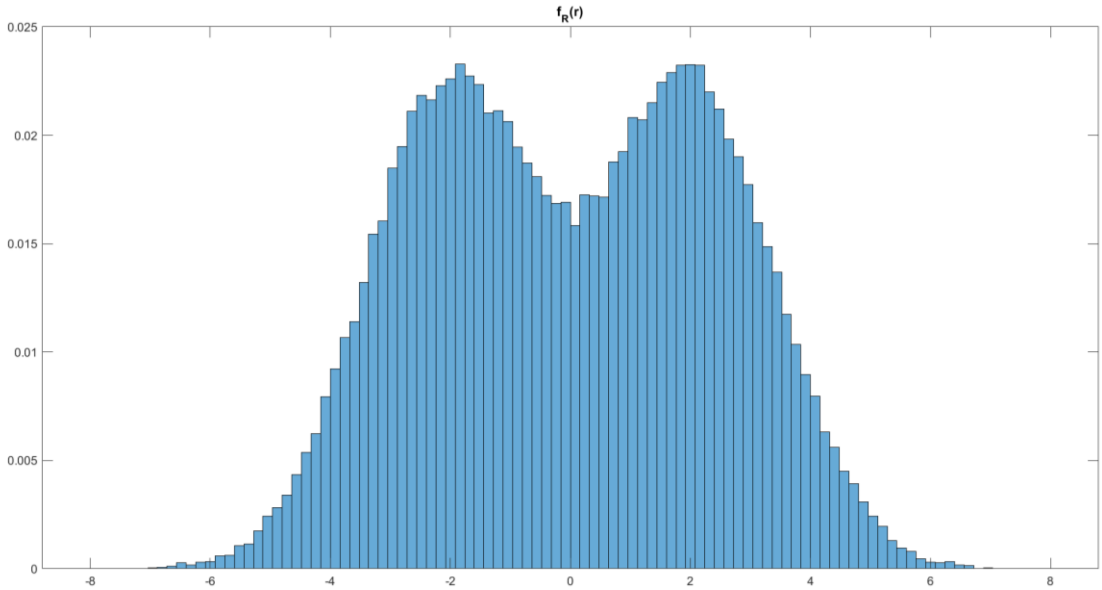

Figure 34 - f_R (r) and A=2

- $Case\ III - f_R(r)\ and\ A = 3$

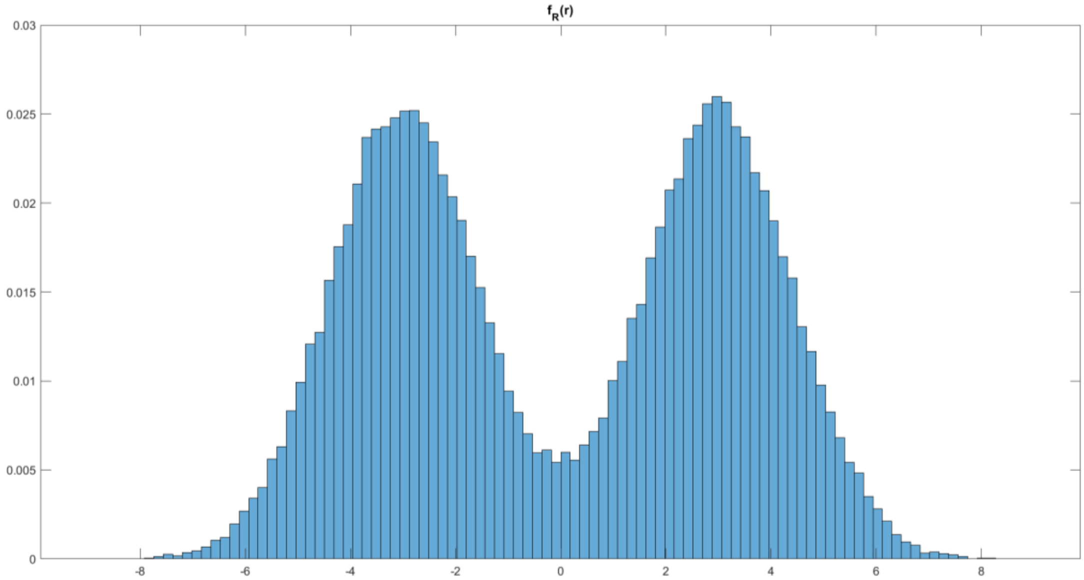

Figure 35 - f_R (r) and A=3

Question 07

- Let's take the scaling factor to be α. Then,

$$R = \alpha S + N$$

Discussion:

- Let's compare the error percentages for two cases; α =1 and α = 2. When α = 1, we found out earlier that the error percentage is 16.21%.
- Now, when we calculate the error percentage of the received signal (when α =2), we get 2.39%. (Calculation performed in the code).
- Thus, it is clear that by scaling the transmitter signal S, the error percentage could be reduced significantly.

- $Case\ I - f_{R|S}(r|S = A)\ and\ \alpha = 2$

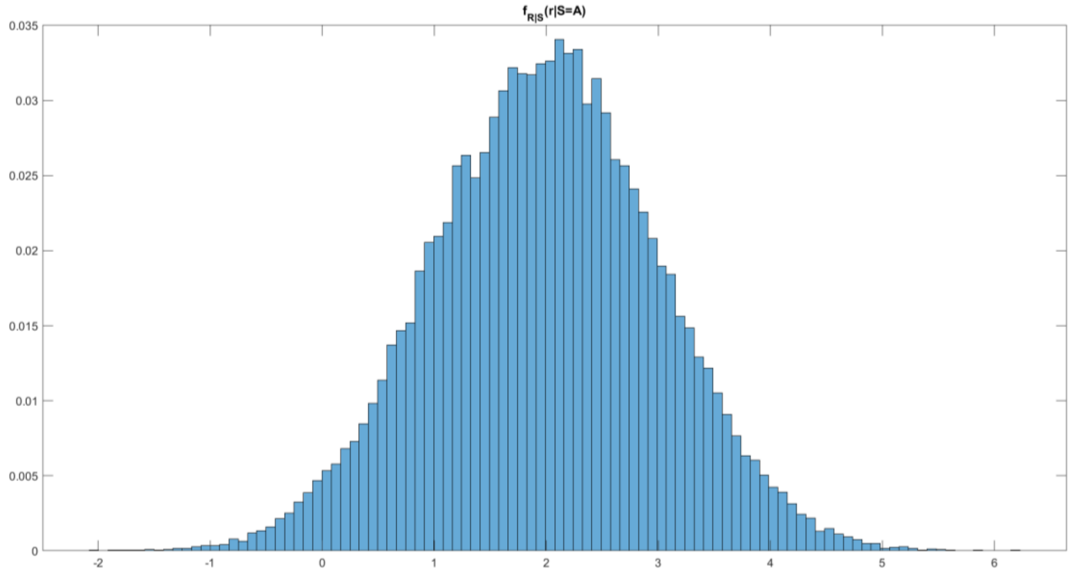

Figure 36 - f_(R|S) (r|S=A) and alpha=2

- $Case\ II - f_{R|S}(r|S = -A)\ and\ \alpha = 2$

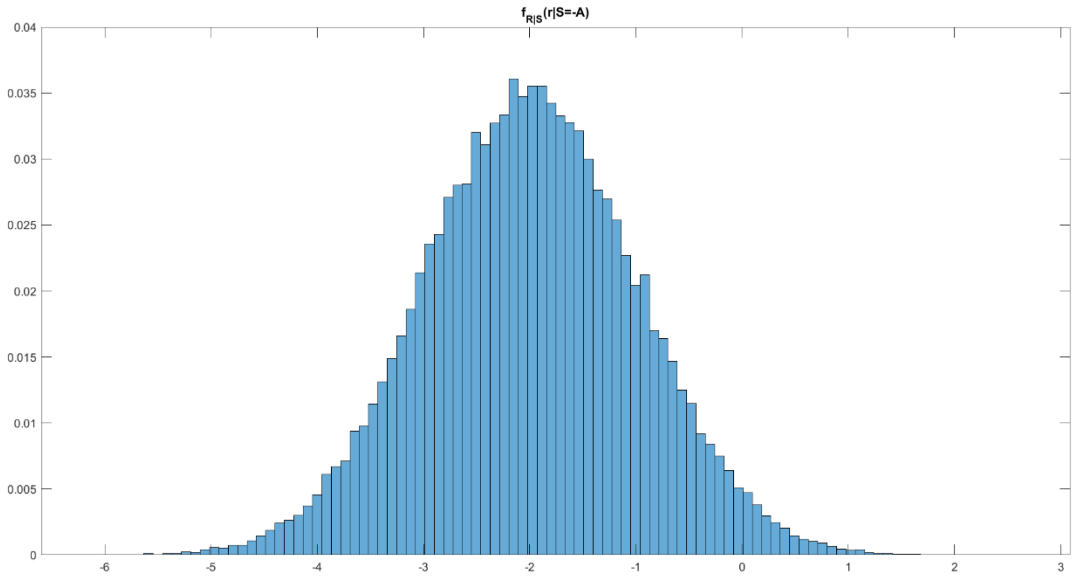

Figure 37 - f_(R|S) (r|S=-A) and alpha=2

- $Case\ III - f_{R|S}(r|S = A)\ and\ \alpha = 3$

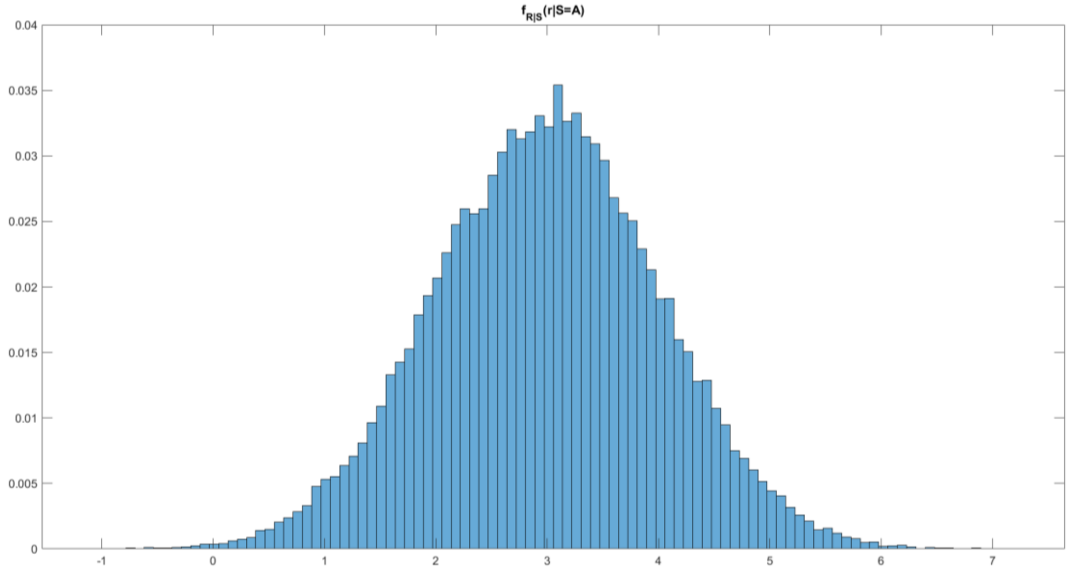

Figure 38 - f_(R|S) (r|S=A) and alpha=3

- $Case\ IV - f_{R|S}(r|S = -A)\ and\ \alpha = 3$

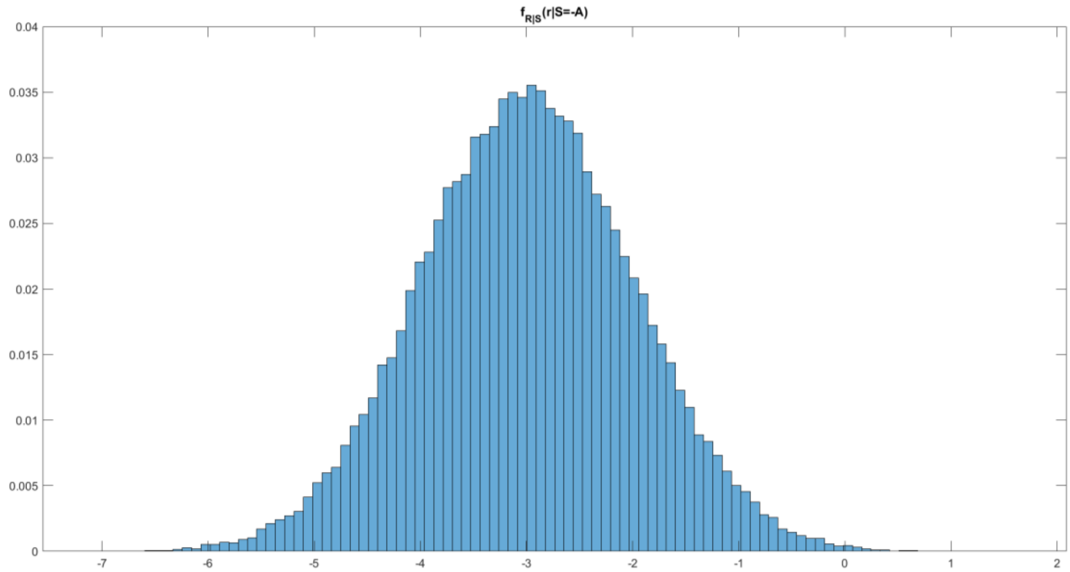

Figure 39 - f_(R|S) (r|S=-A) and alpha=3

α	E[R|S=A]	E[R|S=-A]	E[R]
1	0.9945 (≈ 1)	-0.9947 (≈ -1)	0.0000052 (≈ 0)
2	1.9944 (≈ 2)	-1.9946 (≈ -2)	0.0001875 (≈ 0)
3	2.9945 (≈ 3)	-2.9947 (≈ -3)	-0.00003225 (≈ 0)

- $Case\ I - f_R(r)\ and\ \alpha = 2$

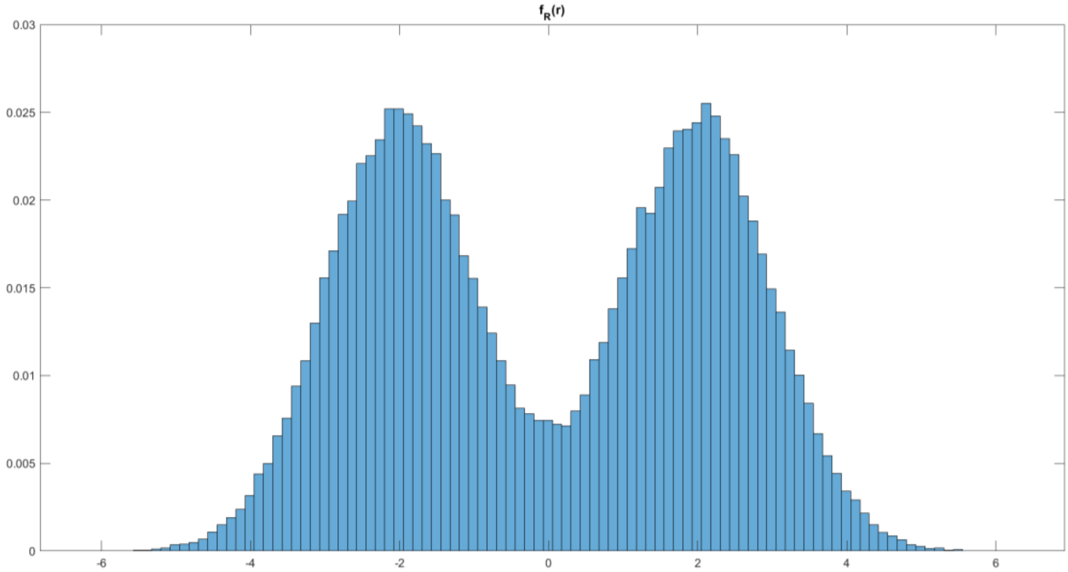

Figure 40 - f_R (r) and alpha=2

- $Case\ II - f_R(r)\ and\ \alpha = 3$

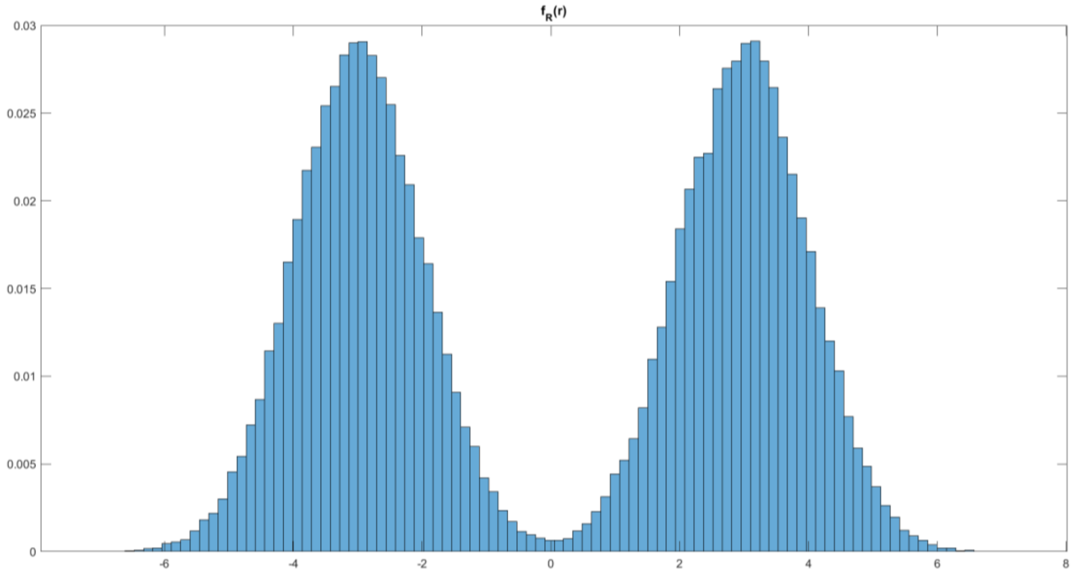

Figure 41 - f_R (r) and alpha=3

Appendix I (Q1 – Q4)

Q1 - Generating a binary sequence D. (D ==> {0,1}, Pr(D=0) = Pr(D=1) = 0.5)

```
L = 1000; %Length of the sequence
D = zeros(1,L);

rng('default') %random generator is fixed
i = randperm(L,L/2); %random indices
D(i) = 1; % D--> equiprobable data array
stairs(D)
title('Binary Sequence D (L=1000)')
ylim([0,1.2])
```

Q1 - Using the binary sequence to generate a stream of rectangular pulses. (A = 1)

```
A = 1; %Amplitude of the rectangular pulses
S = zeros(1,L); %Stream of rectangular pulses
S(D==1) = A;
S(D==0) = -A;

stairs(S)
title('Transmitted Signal')
ylabel('Amplitude')
ylim([-1.2,1.2])
```

Q2 - Generating an AWGN sequence ($\mu = 0, \sigma^2 = 1$)

```
mu = 0; %Mean
sigma = 1; %Standard deviation
N = normrnd(mu, sigma, [1,L]); %AWGN

plot(N)
title('AWGN')
ylabel('Amplitude')
```

Q3 - Received Signal (R = S + N)

```
R = S + N; %Received signal

plot(R)
title('Received Signal (\mu = 0, \sigma^{2} = 1)')
ylabel('Amplitude')

%Now let's plot the received signal when sigma = 2 and sigma = 3
R_sigma_2 = S + normrnd(mu, 2, [1,L]); %sigma = 2
plot(R_sigma_2)
title('Received Signal (\mu = 0, \sigma^{2} = 4)')
ylabel('Amplitude')

R_sigma_3 = S + normrnd(mu, 3, [1,L]); %sigma = 3
plot(R_sigma_3)
title('Received Signal (\mu = 0, \sigma^{2} = 9)')
ylabel('Amplitude')
```

Q4 - Decoding the received signal (Threshold $\tau = 0$)

```
tau = 0; %threshold
Y = zeros(1,L); %Final output
Y(R>tau) = A;
Y(R<=tau) = -A;

stairs(Y)
title('Signal after decoding')
ylabel('Amplitude')
ylim([-1.2,1.2])

%Comparing the transmitted signal S and the output Y
figure;
subplot(2,1,1);
stairs(S)
title('Transmitted signal')
ylim([-1.2,1.2])
ylabel('Amplitude')

subplot(2,1,2);
stairs(Y)
title('Output')
ylim([-1.2,1.2])
ylabel('Amplitude')

%Calculating the error percentage
error = (sum(Y ~= S))*100/L'
sprintf('The error percentage is %.2f%%', error)
```

Appendix II (Q5)

Q5 - L = 100000 (Repeating the steps 1-4)

```
L = 100000; %Length of the sequence
D = zeros(1,L);

rng('default')
j = randperm(L,L/2); %random indices
D(j) = 1; % D--> equiprobable data array

A = 3; %Amplitude of the rectangular pulses
S = zeros(1,L); %Stream of rectangular pulses
S(D==1) = A;
S(D==0) = -A;

mu = 0; %Mean
sigma = 1; %Standard deviation
N = normrnd(mu, sigma, [1,L]); %AWGN

R = S + N; %Received signal
figure;
plot(R)
title('Received Signal (\mu = 0, \sigma^{2} = 1)')
```

```
ylabel('Amplitude')

tau = 0; %threshold
Y = zeros(1,L); %Final output
Y(R>tau) = A;
Y(R<=tau) = -A;

%Comparing the transmitted signal S and the output Y
figure;
subplot(2,1,1);
stairs(S)
title('Transmitted signal')
ylim([-1.2,1.2])
xlim([10000,10100]) %limiting the x-range to view the differences
ylabel('Amplitude')

subplot(2,1,2);
stairs(Y)
title('Output')
ylim([-1.2,1.2])
xlim([10000,10100])
ylabel('Amplitude')

%Calculating the error percentage
error = (sum(Y ~= S))*100/L'
sprintf('The error percentage is %.2f%%', error)
```

Q5 – Generating Histograms

```
bins = 10; %no.of bins
range_R =  range(R);
max_R = max(R);
min_R = min(R);
class_width = range_R/bins;
bin_lims = min_R:class_width:max_R;

%initiating an array for the heights of the bars
heights = zeros(1,bins);

for i = 1:bins
    if i == bins
        heights(i) = sum(R>=bin_lims(i) & R<=bin_lims(i+1));
    else
        heights(i) = sum(R>=bin_lims(i) & R<bin_lims(i+1));
    end
end

%Creating the histogram
figure;
stairs(bin_lims(1:bins),heights,'LineWidth',1.2)
title('Histogram for the received signal')

figure;
bar(bin_lims(1:bins),heights)
title('Histogram for the received signal')
```

```
%Using the built-in function; hist
figure;
hist(R, bins);
title('Histogram for the received signal')
```

Q5a - Changing the no. of bins to 100

```
bins = 100; %no.of bins
range_R =  range(R);
max_R = max(R);
min_R = min(R);
class_width = range_R/bins;
bin_lims = min_R:class_width:max_R;

%initiating an array for the heights of the bars
heights = zeros(1,bins);

for i = 1:bins
    if i == bins
        heights(i) = sum(R>=bin_lims(i) & R<=bin_lims(i+1));
    else
        heights(i) = sum(R>=bin_lims(i) & R<bin_lims(i+1));
    end
end

%Creating the histogram
figure;
stairs(bin_lims(1:bins),heights,'LineWidth',1.2)
title('Histogram for the received signal')

figure;
bar(bin_lims(1:bins),heights)
title('Histogram for the received signal')

%Using the built-in function; hist
figure;
hist(R, bins);
title('Histogram for the received signal')
```

Q5b - Plotting the pdf of $f_{R|S}(r|S = A)$

```
R_S_pos_A = R(S==A); %Extracting R vals when S=A

%we will use 100 bins
bins = 100;
h1 = histogram(R_S_pos_A,'NumBins',bins,'Normalization','probability') %Normalizing
title('f_{R|S}(r|S=A)')

x1 = h1.Values;
y1 = h1.BinEdges;
```

Q5b - Plotting the pdf of $f_{R|S}(r|S = -A)$

```
R_S_neg_A = R(S==-A); %Extracting R vals when S=-A
```

```
%we will use 100 bins
bins = 100;
h2 = histogram(R_S_neg_A,'NumBins',bins,'Normalization','probability') %Normalizing
title('f_{R|S}(r|S=-A)')

x2 = h2.Values;
y2 = h2.BinEdges;
```

Q5c – Finding $E[R|S=A],\ E[R|S=-A]\ \ and\ \ E[R]$

```
%Finding E[R|S=A]
E_R_S_pos_A = 0;
for i = 1:bins
    E_R_S_pos_A = E_R_S_pos_A + (x1(i)*(0.5*(y1(i)+y1(i+1))));
end

%Finding E[R|S=-A]
E_R_S_neg_A = 0;
for i = 1:bins
    E_R_S_neg_A = E_R_S_neg_A + (x2(i)*(0.5*(y2(i)+y2(i+1))));
end

h3 = histogram(R,'NumBins',bins,'Normalization','probability') %Normalizing
title('f_{R}(r)')

x3 = h3.Values;
y3 = h3.BinEdges;

E_R = 0;
for i = 1:bins
    E_R = E_R + (x3(i)*(0.5*(y3(i)+y3(i+1))));
end
```

Q5d - Sketching the PDF $f_R(r)$

```
h = histogram(R,'NumBins',bins,'Normalization','probability') %Normalizing
title('f_{R}(r)')
```

Appendix III (Q6)

Q6 - Including interference (R = S + N + I)

```
L = 100000; %Length of the sequence
D = zeros(1,L);

rng('default')
j = randperm(L,L/2); %random indices
D(j) = 1; % D--> equiprobable data array

A = 1; %Amplitude of the rectangular pulses
S = zeros(1,L); %Stream of rectangular pulses
S(D==1) = A;
```

```
S(D==0) = -A;

mu = 0; %Mean of the noise
sigma = 1; %Standard deviation
N = normrnd(mu, sigma, [1,L]); %AWGN

mu_i = 0; %Mean of the interference
sigma_i = 1;%Standard deviation of the interference
I = normrnd(mu_i, sigma_i, [1,L]);

R = S + N + I; %Received signal
figure;
plot(R)
title('Received Signal (\mu = 0, \sigma^{2} = 1)')
ylabel('Amplitude')

tau = 0; %threshold
Y = zeros(1,L); %Final output
Y(R>tau) = A;
Y(R<=tau) = -A;

%Comparing the transmitted signal S and the output Y
figure;
subplot(2,1,1);
stairs(S)
title('Transmitted signal')
ylim([-1.2,1.2])
xlim([10000,10100]) %limiting the x-range to view the differences
ylabel('Amplitude')

subplot(2,1,2);
stairs(Y)
title('Output')
ylim([-1.2,1.2])
xlim([10000,10100])
ylabel('Amplitude')

%Calculating the error percentage
error = (sum(Y ~= S))*100/L'
sprintf('The error percentage is %.2f%%', error)
```

Q6 - Plotting the pdf of $f_{R|S}(r|S = A)$

```
R_S_pos_A = R(S==A); %Extracting R vals when S=A

%we will use 100 bins
bins = 100;
figure;
h1 = histogram(R_S_pos_A,'NumBins',bins,'Normalization','probability') %Normalizing
title('f_{R|S}(r|S=A)')

x1 = h1.Values;
y1 = h1.BinEdges;
```

Q6 - Plotting the pdf of $f_{R|S}(r|S=-A)$

```
R_S_neg_A = R(S==-A); %Extracting R vals when S=-A

%we will use 100 bins
bins = 100;
h2 = histogram(R_S_neg_A,'NumBins',bins,'Normalization','probability') %Normalizing
title('f_{R|S}(r|S=-A)')

x2 = h2.Values;
y2 = h2.BinEdges;
```

Q6 – Finding $E[R|S=A],\ E[R|S=-A]\ \ and\ \ E[R]$

```
%Finding E[R|S=A]
E_R_S_pos_A = 0;
for i = 1:bins
    E_R_S_pos_A = E_R_S_pos_A + (x1(i)*(0.5*(y1(i)+y1(i+1))));
end

%Finding E[R|S=-A]
E_R_S_neg_A = 0;
for i = 1:bins
    E_R_S_neg_A = E_R_S_neg_A + (x2(i)*(0.5*(y2(i)+y2(i+1))));
end

h3 = histogram(R,'NumBins',bins,'Normalization','probability') %Normalizing
title('f_{R}(r)')

x3 = h3.Values;
y3 = h3.BinEdges;

E_R = 0;
for i = 1:bins
    E_R = E_R + (x3(i)*(0.5*(y3(i)+y3(i+1))));
end
```

Q6 - Sketching the PDF $f_R(r)$

```
h = histogram(R,'NumBins',bins,'Normalization','probability') %Normalizing
title('f_{R}(r)')
```

Appendix IV (Q7)

Q7 - Impact of Scaling - $R = \alpha S + N$

```
L = 100000; %Length of the sequence
D = zeros(1,L);
alpha = 2;

rng('default')
```

```
j = randperm(L,L/2); %random indices
D(j) = 1; % D--> equiprobable data array

A = 1; %Amplitude of the rectangular pulses
S = zeros(1,L); %Stream of rectangular pulses
S(D==1) = A;
S(D==0) = -A;

mu = 0; %Mean
sigma = 1; %Standard deviation
N = normrnd(mu, sigma, [1,L]); %AWGN

R = alpha*S + N; %Received signal
figure;
plot(R)
title('Received Signal (\mu = 0, \sigma^{2} = 1)')
ylabel('Amplitude')

tau = 0; %threshold
Y = zeros(1,L); %Final output
Y(R>tau) = A;
Y(R<=tau) = -A;

%Comparing the transmitted signal S and the output Y
figure;
subplot(2,1,1);
stairs(S)
title('Transmitted signal')
ylim([-1.2,1.2])
xlim([10000,10100]) %limiting the x-range to view the differences
ylabel('Amplitude')

subplot(2,1,2);
stairs(Y)
title('Output')
ylim([-1.2,1.2])
xlim([10000,10100])
ylabel('Amplitude')

%Calculating the error percentage
error = (sum(Y ~= S))*100/L'
sprintf('The error percentage is %.2f%%', error)
```

Q7 - Plotting the pdf of $f_{R|S}(r|S = A)$

```
R_S_pos_A = R(S==A); %Extracting R vals when S=A

%we will use 100 bins
bins = 100;
figure;
h1 = histogram(R_S_pos_A,'NumBins',bins,'Normalization','probability') %Normalizing
title('f_{R|S}(r|S=A)')

x1 = h1.Values;
y1 = h1.BinEdges;
```

Q7 - Plotting the pdf of $f_{R|S}(r|S=-A)$

```
R_S_neg_A = R(S==-A); %Extracting R vals when S=-A

%we will use 100 bins
bins = 100;
h2 = histogram(R_S_neg_A,'NumBins',bins,'Normalization','probability') %Normalizing
title('f_{R|S}(r|S=-A)')

x2 = h2.Values;
y2 = h2.BinEdges;
```

Q7 – Finding $E[R|S=A],\ E[R|S=-A]\ \ and\ \ E[R]$

```
%Finding E[R|S=A]
E_R_S_pos_A = 0;
for i = 1:bins
    E_R_S_pos_A = E_R_S_pos_A + (x1(i)*(0.5*(y1(i)+y1(i+1))));
end

%Finding E[R|S=-A]
E_R_S_neg_A = 0;
for i = 1:bins
    E_R_S_neg_A = E_R_S_neg_A + (x2(i)*(0.5*(y2(i)+y2(i+1))));
end

h3 = histogram(R,'NumBins',bins,'Normalization','probability') %Normalizing
title('f_{R}(r)')

x3 = h3.Values;
y3 = h3.BinEdges;

E_R = 0;
for i = 1:bins
    E_R = E_R + (x3(i)*(0.5*(y3(i)+y3(i+1))));
end
```

Q7 - Sketching the PDF $f_R(r)$

```
h = histogram(R,'NumBins',bins,'Normalization','probability') %Normalizing
title('f_{R}(r)')
```