Department of Electronic and Telecommunication Engineering

University of Moratuwa, Sri Lanka

EN 2040 Random Signals and Processes

Simulation Assignment

Project Report

Submitted by

Thalagala B.P.   180631J

Submitted on

September 17, 2021

Contents

List of Figures

PDF is clickable

Note:
All the materials and executable `MATLAB R2018a` *Live Script related to the project can also be found at* *https://github.com/bimalka98/Digital-Signal-Processing*

1 Question 1 and Question 2

Refer the Appendix for the Code. For the generation of the AWGN, MATLAB's `normrnd(`μ`,`σ`,[1, samples])` function which returns random samples from a normal distribution with mean μ and standard deviation σ was used.

this page was intentionally left blank

2 Question 3

Plot of the sequence of Received signal and observations of the impact of the variance of noise on it by varying $\sigma^2 = 1$.

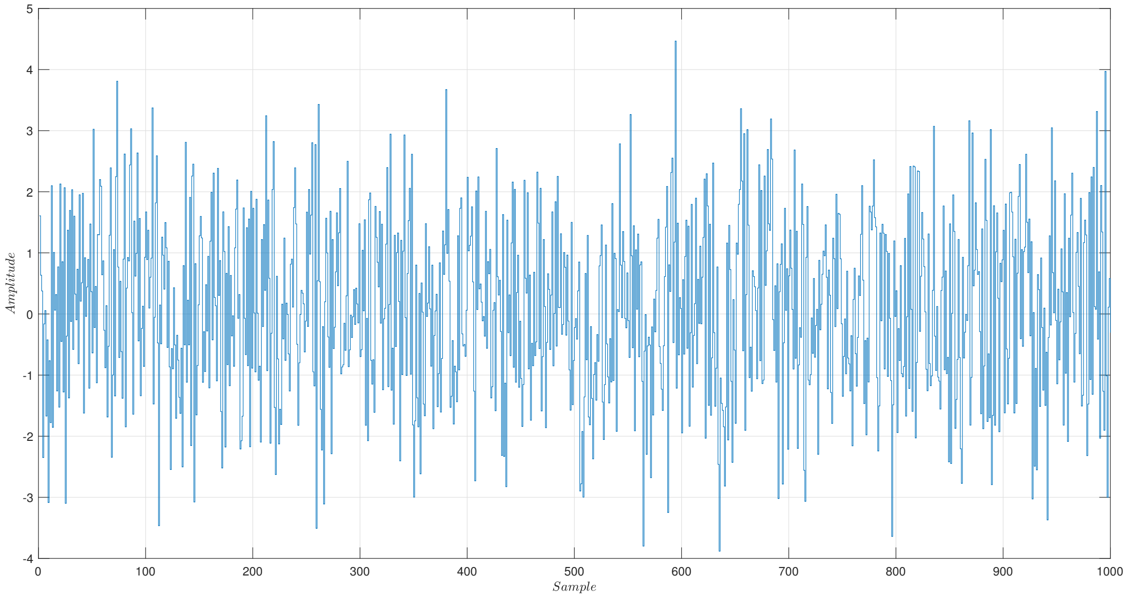

Figure 1: Sequence of the Received signal when $\sigma^2 = 1$

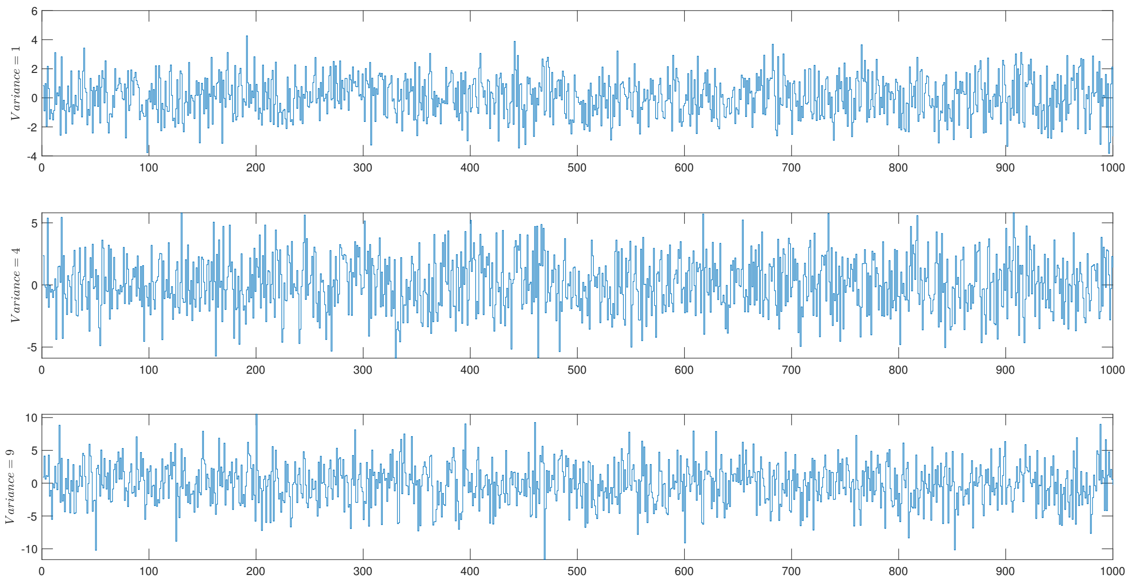

Figure 2: Impact of the variance of noise on the Received signal

As illustrated in the above figure when the variance of the noise increases the range of the values the received sample can be in increases (observe the change in the range of Y axis). This essentially increases the bit errors at the receiving end.

3 Question 4

Sketching and comparing the sequence of Y(signal recovered through threshold) with the transmitted signal.

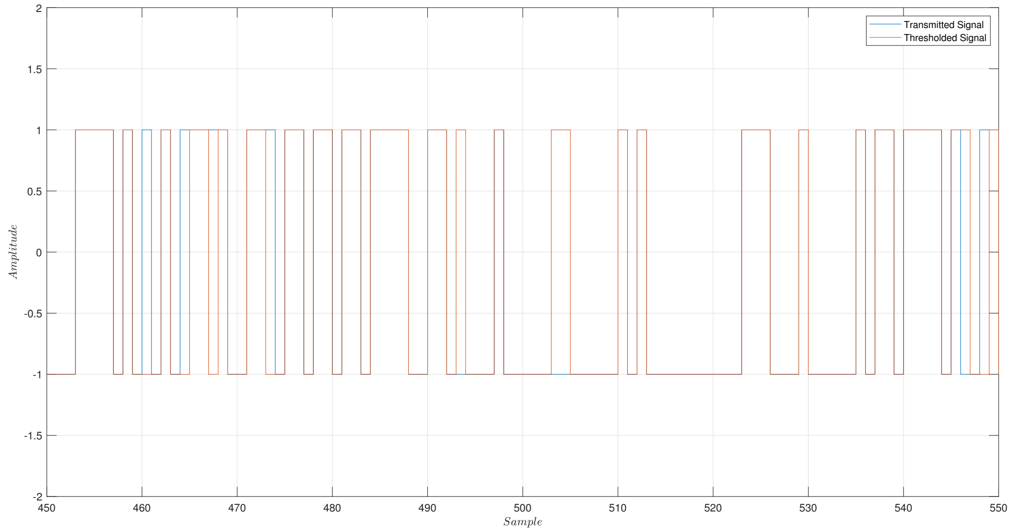

Figure 3: Sequence of Y when the number of samples is 1000

Most of the transmitted bits have been identified correctly by the receiver after the thresholding(perfectly aligned curves). But some of the bits have been identified incorrectly after the thresholding. Change of signal levels due to the unwanted noise added during the process of transmission is the reason.

4 Question 5

Repeating the above steps for a sequence of length L = 100,000.

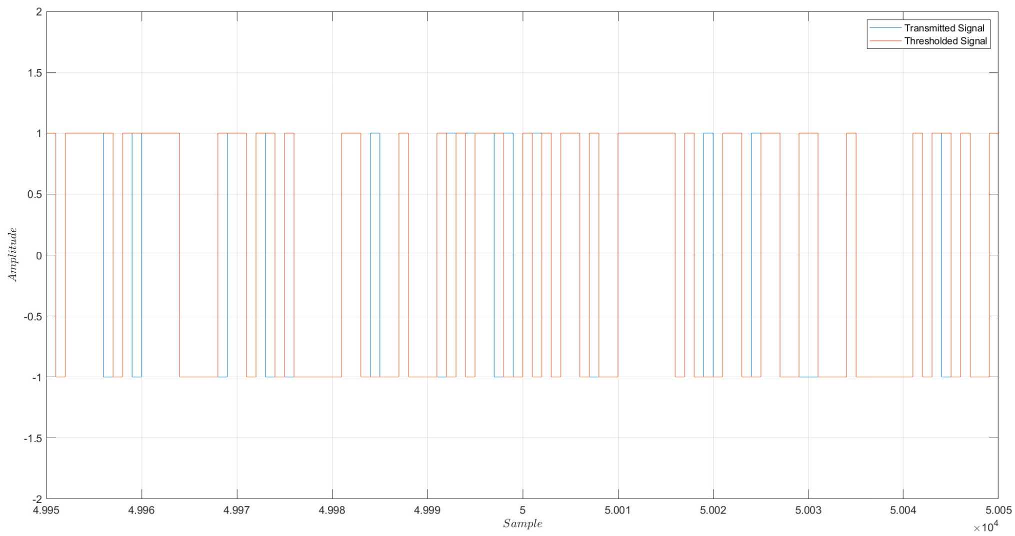

Figure 4: Sequence of Y when the number of samples is 100,000

The same observations made regarding the previous figure is applicable to this scenario as well. But probability of error has a significant reduction when transmitting a huge number of bits.

4.1 Histogram of the received sequence taking the no of bins as 10

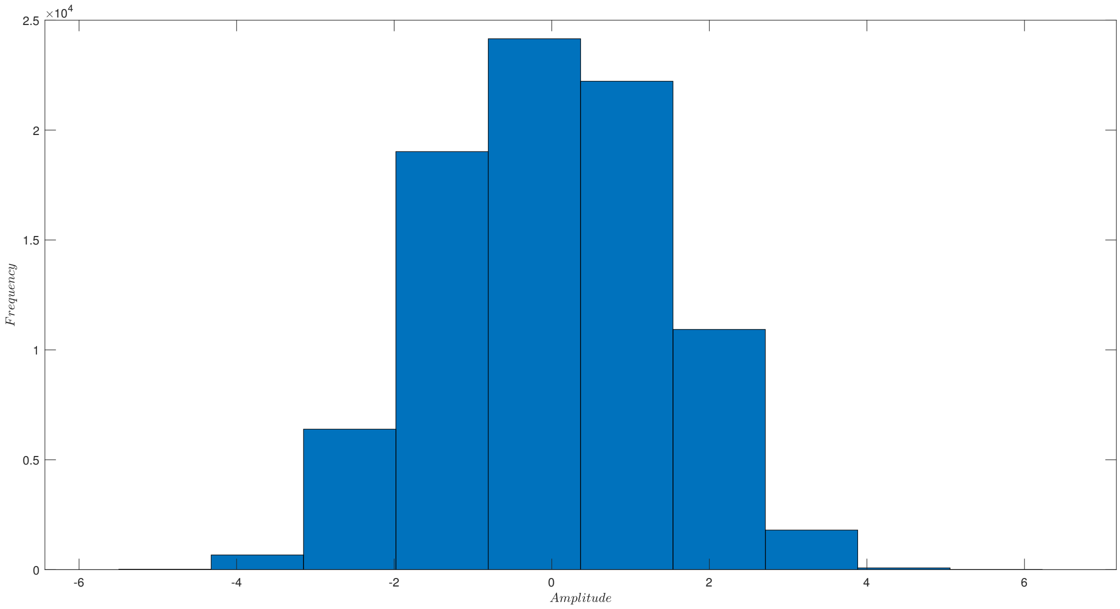

Figure 5: Histogram of the received sequence using the user defined function: bins = 10

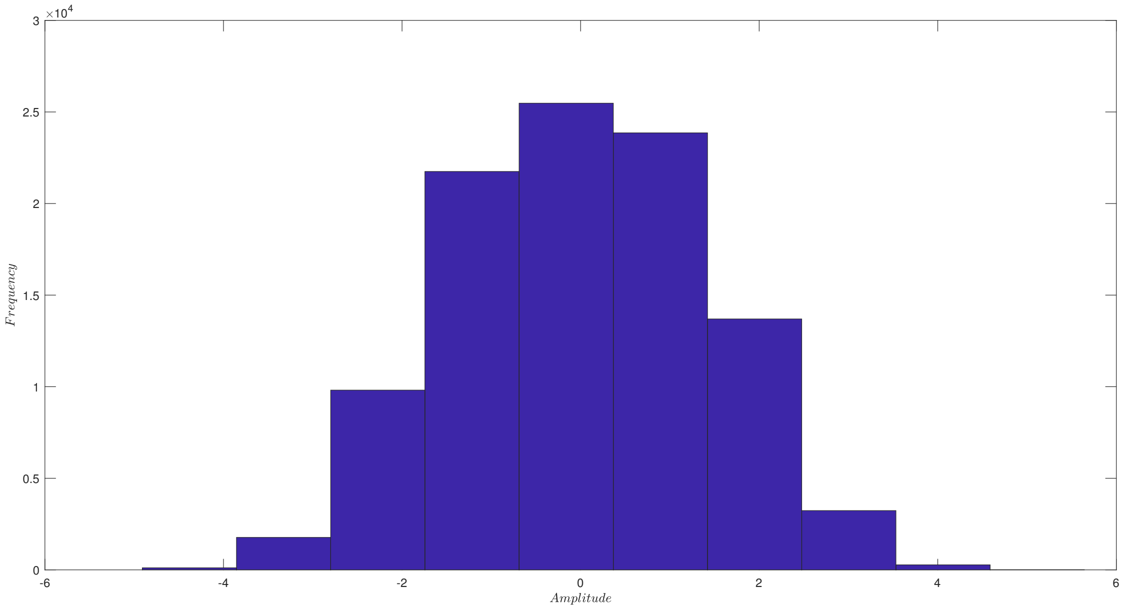

Figure 6: Histogram of the received sequence using the MATLAB's built-in `hist()` fucntion: bins = 10

Comparison : Histograms generated using the custom user defined function and the built-in function are almost the same. The reason for the difference in frequencies can be identified as the range differences of bins considered in the two algorithms. Code of the user defined fucntion can be found at the end of the Appendix.

4.2 Impact when the number of bins is changed from 10 to 100

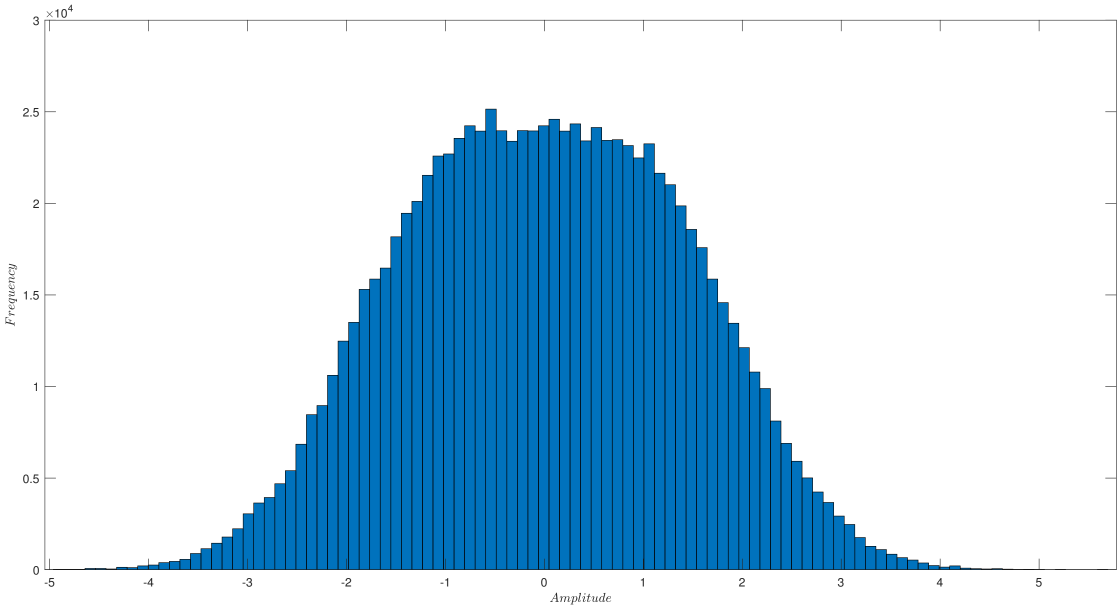

Figure 7: Histogram of the received sequence using the user defined function: bins =100

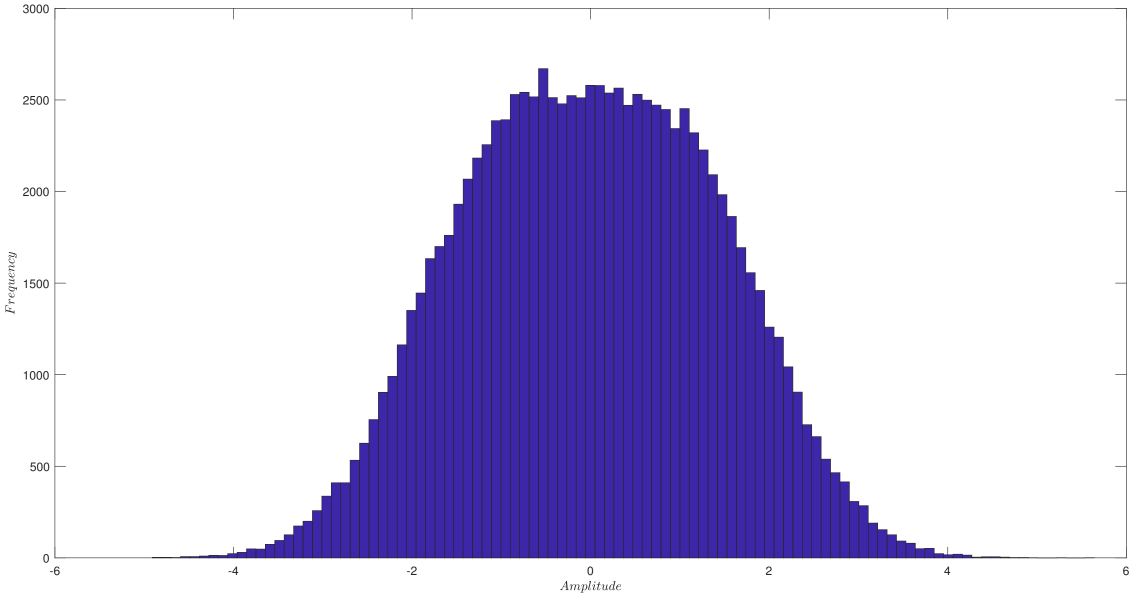

Figure 8: Histogram of the received sequence using the MATLAB's built-in `hist()` fucntion: bins = 100

Impact: When the number of bins is changed from 10 to 100, shape of the normal distribution becomes clearly visible and differences between two histograms become negligible as the ranges of the bins become more similar.

4.3 Conditional PDFs when number of bins = 100 and A = 1

Dependent axis represents the normalized frequencies calculated as follows. And therefore following figures illustrates the approximation for the required Probability Density Functions.

$$Normalized\ Freq = \frac{Frequency}{Total\ Number\ of\ Samples \times Width\ of\ a\ Bin}$$

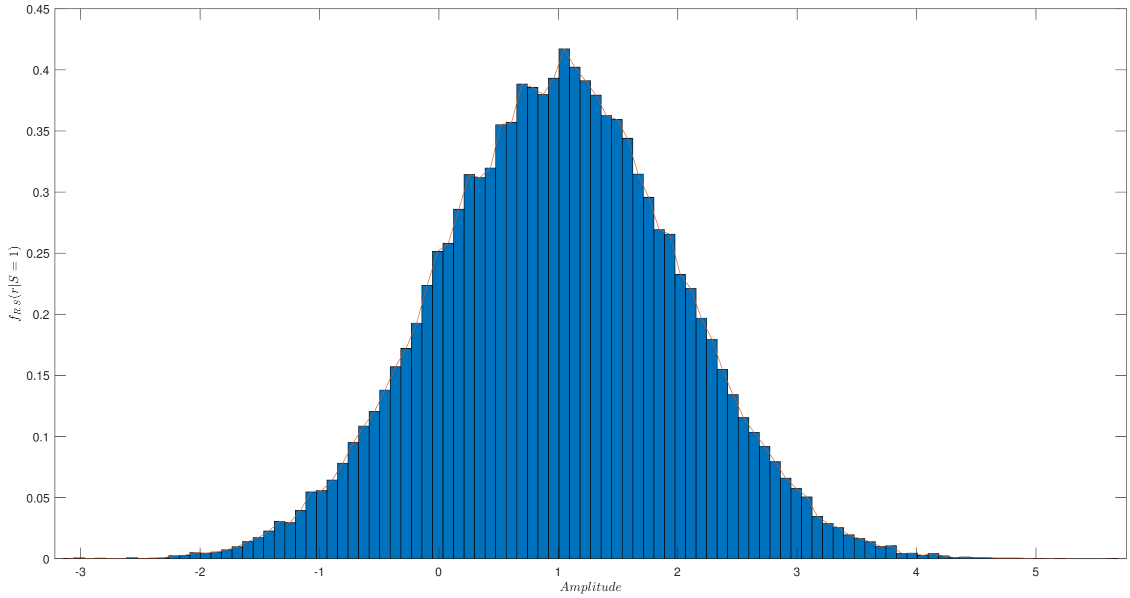

Figure 9: $f_{R|S}(r|S = A)$ when A = 1: Mean ≈ 1

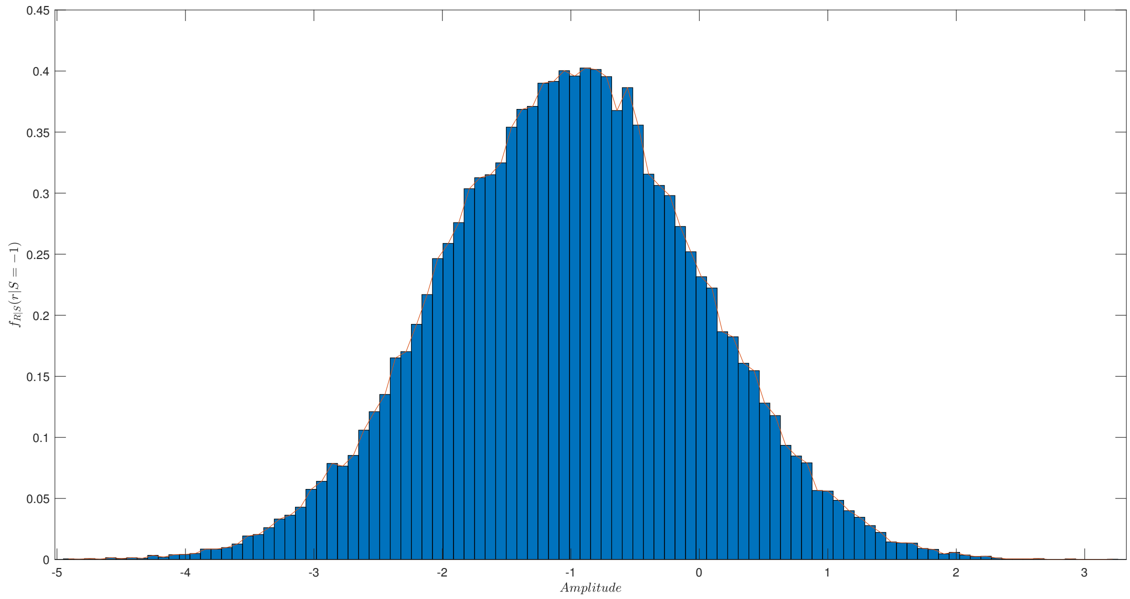

Figure 10: $f_{R|S}(r|S = -A)$ when A =1: Mean ≈ -1

Observe that when the distribution is conditioned on S, the mean of the distribution shifts along the amplitude axis towards the S.

4.4 Impact of A on the Conditional PDFs

As illustrated in the following figures, when the distribution is conditioned on S =A , the mean of the distribution shifts along the amplitude axis towards the S = A. But the variance of the distribution remains the same as it is only affected by the AWGN.

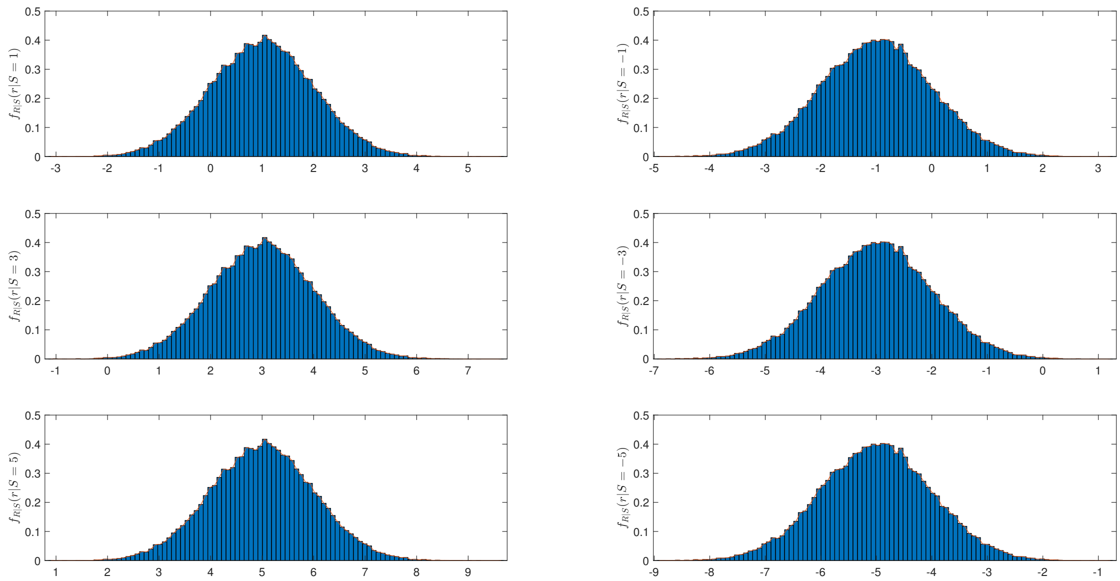

Figure 11: Impact of A on the Conditional PDFs

4.5 Expected Values of the distributions when A = 1

The expected value of a continuous Random variable X which has a PDF of $f_X(x)$ is given as,

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

The same formula can be redefined for discrete case as follows where Δx represents the width of a bin which is equal in our cases.

$$E[X] = \sum_{i=1}^{N} x_i.P(x_i).\Delta x$$

Therefore the requires expected values can be easily found using the `Expected(x, px, bin_width)` function defined in the appendix. The required parameters of that function can be found using the `myHistogram(valueArray,numberOfBins, Normalized)` function.

Expected value $E[R]$	=	-0.00073493
Expected value $E[R\|S=A]$	=	1.0004
Expected value $E[R\|S=-A]$	=	-1.0011

4.6 Marginal PDF of R $f_R(r)$

Dependent axis represents the normalized frequencies calculated as follows as mentioned previously. Therefore, the following figure illustrates the approximation for the required Marginal Probability Density Functions of the continuous Random Variable R.

$$Normalized\ Freq = \frac{Frequency}{Total\ Number\ of\ Samples \times Width\ of\ a\ Bin}$$

Expected value of the Random Variable ($E[R]$ = -0.00073493) reaches zero as the probabilities Pr(D = 0) = Pr(D = 1) = 1/2 are equal.

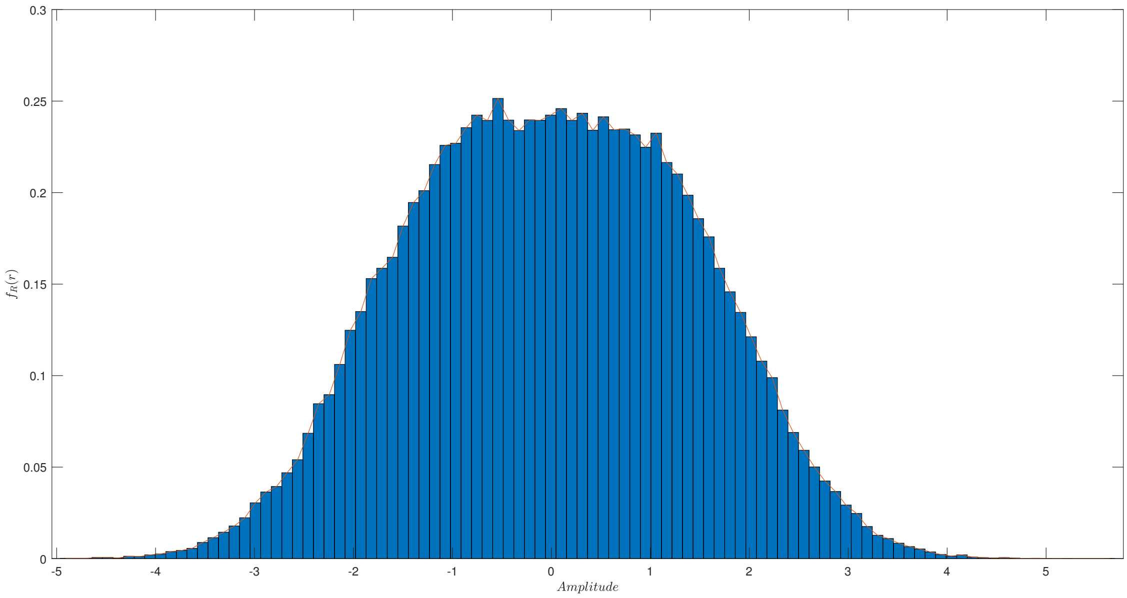

Figure 12: Marginal PDF of R $f_R(r)$

5 Question 6: Effect of Interference

Interference does nothing other than increasing the variance of the distribution. When comparing the following figures with the figures of the previous section it is clearly visible. As the variance increases, to keep the area under the graph a constant, the peak of the graph lowers. Therefore following two observations can be made when additional interference is added to the transmitted signal.

- Variance of the distribution increases. Which eventually increases the probability of error.
- Peak value of the distribution decreases to keep the area under the curve constant.

These observations are common to all the figures and subsections of this section.

5.1 Conditional PDFs when number of bins = 100, A = 1 and under additional interference

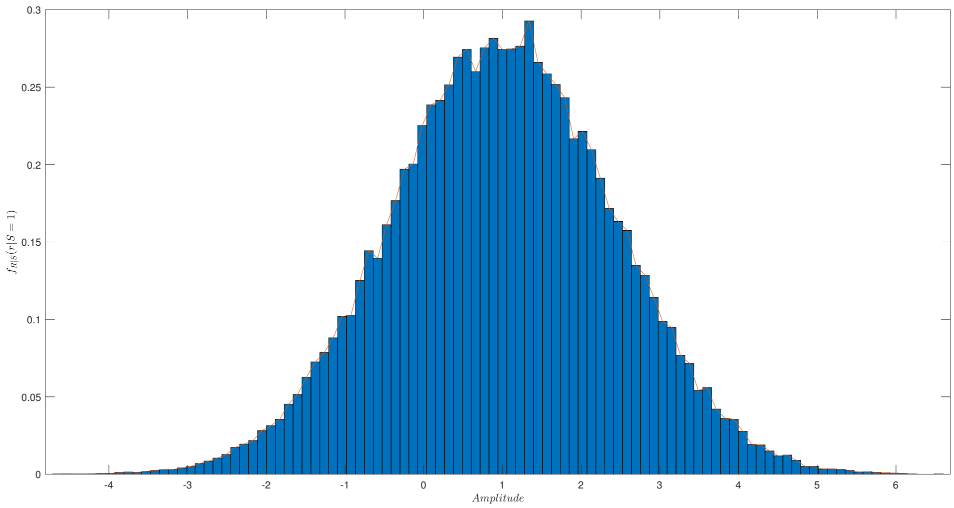

Figure 13: $f_{R|S}(r|S = A)$ when interference is added

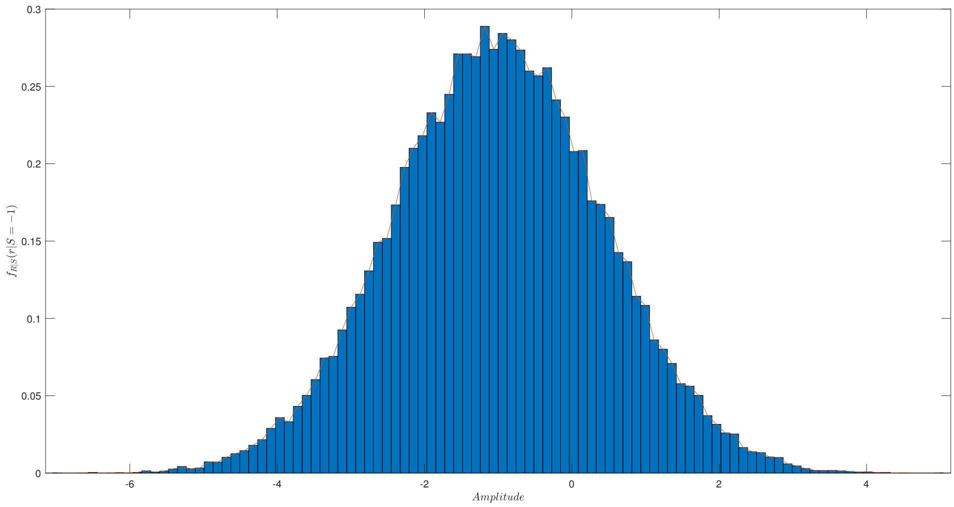

Figure 14: $f_{R|S}(r|S = -A)$ when interference is added

5.2 Impact of A on the Conditional PDFs under additional interference

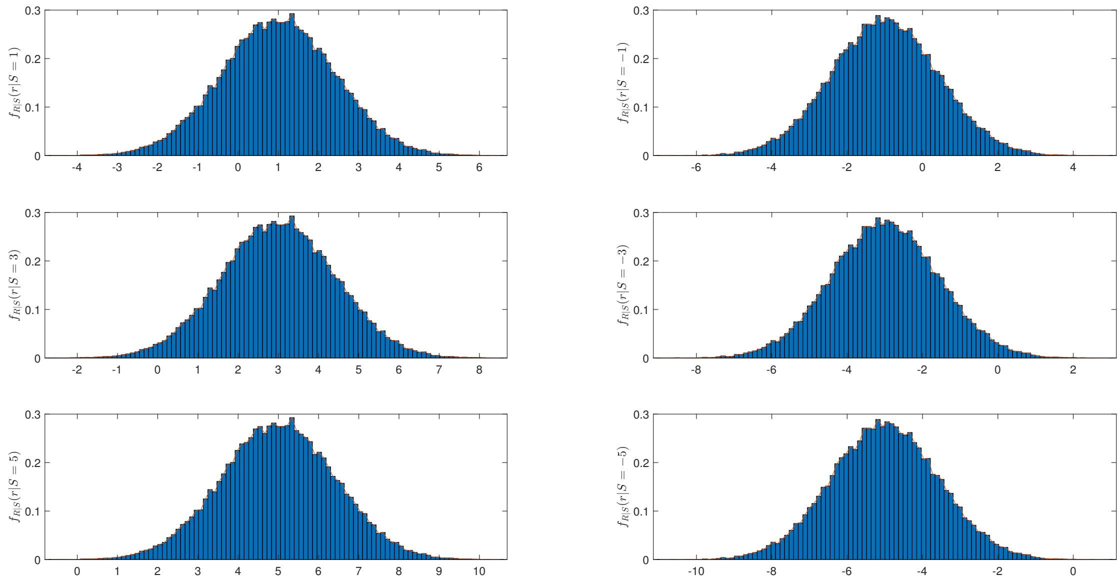

Figure 15: Impact of A on the Conditional PDFs under additional interference

5.3 Expected Values of the distributions and under additional interference

As previously described interference affects only to the variance of the distribution. Therefore the Expected Values remains almost the same.

$$\begin{array}{lcr} \text{Expected value } E[R] & = & \text{-0.0012806} \\ \text{Expected value } E[R|S=A] & = & 0.99798 \\ \text{Expected value } E[R|S=-A] & = & \text{-0.99949} \end{array}$$

5.4 Marginal PDF of R $f_R(r)$ and under additional interference

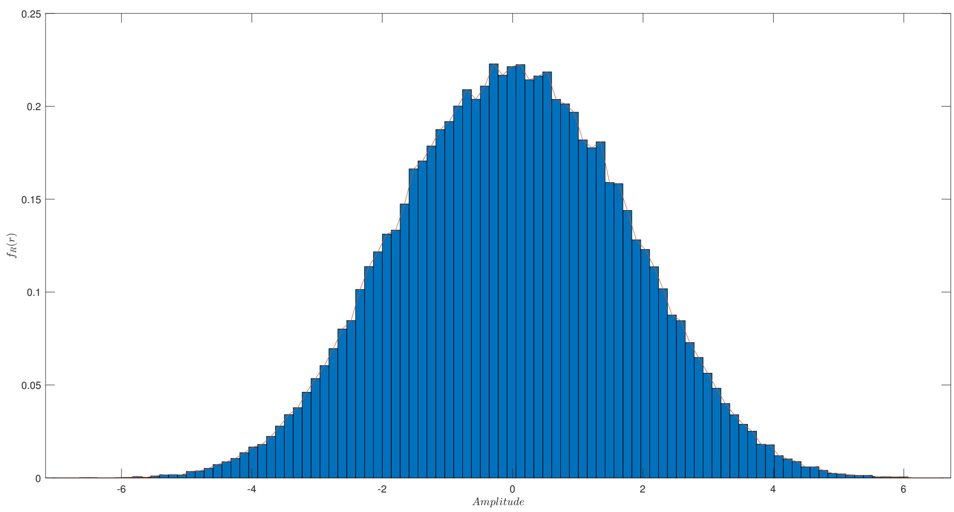

Figure 16: Marginal PDF of R $f_R(r)$ under additional interference

6 Question 7: Effect of Scaling when the number of bins = 100 and A = 1 and $\sigma^2 = 1$

Effect of scaling of the pulse sequence(S), on the conditional probability distributions has the same effect as changing the signal level 'A'. Therefore the figures and the subsections will be the same as that of Question 5.

Therefore only the behavior of Marginal distribution of R(received signal) is illustrated here. As it was not considered in earlier sections.

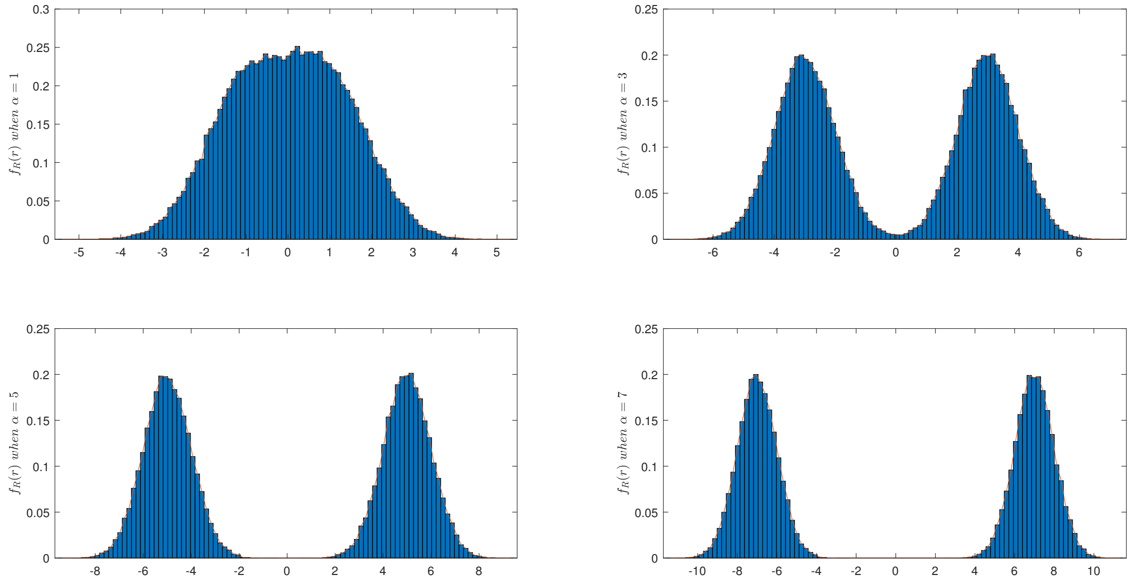

Figure 17: Effect of Scaling on the Marginal PDF of R $f_R(r)$

The first sub-figure of the above figure shows the instance without any scaling. when the scale is increased, the distribution tends to divide in to two separate peaks centered around $+\alpha$ and $-\alpha$ as expected due to the lower variance than the scaling factor.

Appendices

A Matlab Code for the Simulation

EN2040 Random Signals and Processes

Simulation Assignment 180631J

Rectangular pulses of ±A carry binary equiprobable data over a communication channel. Binary data D $\in$ {0, 1} is mapped to the amplitude of the rectangular pulses S as follows:

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

The channel is corrupted by additive white Gaussian noise (AWGN) of zero mean and variance σ^2 . At the receiver,the received signal is sampled and compared with a threshold $\tau = 0$. A received signal sample is given by R = S + N, where N represents the random effect of noise. Decoding is done by considering the amplitude of R. To this end, the decision is taken as follows:

$$Y = \begin{cases} +A & if \;\; R > \tau \\ -A & if \;\; R \leq \tau \end{cases}$$

1. Generate a binary sequence of length L = 1000, considering D $\in$ {0, 1} and Pr (D = 0) = Pr (D = 1) = 1/2. Use the binary sequence to generate a stream of rectangular pulses of amplitude S, where S $\in$ {−A, +A} with A = 1.

```
L = 1000;   % length of the binary sequence
A = 1;      % amplitude of the pulse
bin_seq = randi([0 1], 1, L);
pulse_seq = zeros(1,L); % transmitted signal
for index = 1:L
    if bin_seq(index) == 1
        pulse_seq(index) = +A;
    else
        pulse_seq(index) = -A;
    end
end
```

2. Generate an AWGN sequence, also of length L = 1000, considering σ^2= 1.

```
mu = 0;                     % mean
variance = 1;               % variance
sigma = sqrt(variance);     % standard deviation
awgn_seq = normrnd(mu,sigma,[1, L]);
```

3. Plot the sequence of R and observe the impact of the variance of noise on R by varying σ^2 .

```
R = pulse_seq + awgn_seq;   % received signal
stairs(R);
ylabel("$ Amplitude$",'Interpreter','latex');
xlabel("$ Sample$",'Interpreter','latex');
grid on;
figure;
```

```
for sigma2 = 1:3
    subplot(3,1,sigma2);
    awgn_seq2 = normrnd(mu,sigma2,[1, L]);
    R2 = pulse_seq + awgn_seq2;
    stairs(R2);
    ylabel("$Variance = " +num2str(sigma2^2)+"$",'Interpreter','latex');
end
```

4. Sketch and compare the sequence of Y with the transmitted signal.

```
xspan = 50;             % to visualize a limited number of samples
figure;
% sketching the transmitted signal
stairs(pulse_seq, 'DisplayName', 'Transmitted Signal','MarkerSize',10);
hold on;
threshold = 0;          % thresholding the received signal
Y = zeros(1,L);
for index = 1:L
    if R(index) > threshold
        Y(index) = +A;
    else
        Y(index) = -A;
    end
end
% sketching the thresholded signal

stairs(Y,'DisplayName','Thresholded Signal','MarkerSize',10);
ylabel("$ Amplitude$",'Interpreter','latex');
xlabel("$ Sample$",'Interpreter','latex');
grid on; legend('show');
ylim([-2 2]); xlim([(L/2 -xspan) (L/2 +xspan)]);
hold off;
```

Comparison: Some of the transmitted pulses have been identified by the receiver as they are. while some of them have been identified incorrectly due to the AWGN added in the channel.

5. Repeat the above steps for a sequence of length L = 100,000. Write a code to generate and plot the histogram of the received sequence taking the no of bins as 10. Compare your result with the one generated from the built-in function hist() of MATLAB.

(a) Change the no of bins from 10 to 100 and observe the impact.

(b) By selecting a suitable value for the no of bins, sketch the conditional PDFs $f_{R|S}(r|S = A)$ and $f_{R|S}(r|S = -A)$ with the use of the normalized histograms. Change the value of A and observe the impact.

(c) Find E [R|S = A] ;E [R|S = �A] and E [R].

(d) Similarly, sketch the PDF fR (r).

Repeating the previous steps for a sequence of length L = 100,000

```
L = 100000;   % length of the binary sequence
A = 1;      % amplitude of the pulse
bin_seq = randi([0 1], 1, L);
pulse_seq = zeros(1,L); % transmitted signal
for index = 1:L
    if bin_seq(index) == 1
        pulse_seq(index) = +A;
    else
        pulse_seq(index) = -A;
    end
end

% generating AWGN noise
mu = 0;                      % mean
variance = 1;                % variance
sigma = sqrt(variance);      % standard deviation
awgn_seq = normrnd(mu,sigma,[1, L]);

R = pulse_seq + awgn_seq;    % received signal

xspan = 50;                  % to visualize a limited number of samples
figure;
% sketching the transmitted signal
stairs(pulse_seq, 'DisplayName', 'Transmitted Signal','MarkerSize',10 );
hold on;
threshold = 0;               % thresholding the received signal
Y = zeros(1,L);
for index = 1:L
    if R(index) > threshold
        Y(index) = +A;
    else
        Y(index) = -A;
    end
end
% sketching the thresholded signal
stairs(Y,'DisplayName','Thresholded Signal','MarkerSize',10);
ylabel("$ Amplitude$",'Interpreter','latex');
xlabel("$ Sample$",'Interpreter','latex');
grid on; legend('show');
ylim([-2 2]); xlim([(L/2 -xspan) (L/2 +xspan)]);
hold off;
```

Code to generate and plot the histogram of the received sequence

```
% taking the no of bins as 10. Functions is at the end of the script.
figure; myHistogram(R,10,0);
ylabel("$ Frequency$",'Interpreter','latex');
xlabel("$ Amplitude$",'Interpreter','latex');
% histogram generated from the built-in function hist() of MATLAB.
figure; hist(R,10);
ylabel("$ Frequency$",'Interpreter','latex');
xlabel("$ Amplitude$",'Interpreter','latex');
```

Comparison: Histograms generated using the custom code and the built-in function are almost the same. The reason for the difference in frequencies can be identified as the range difference of bins considered in the two algorithms.

Impact when the number of bins is changed from 10 to 100

```
% histogram generated from the custom function.
figure; myHistogram(R, 100,0);
ylabel("$ Frequency$",'Interpreter','latex');
xlabel("$ Amplitude$",'Interpreter','latex');
% histogram generated from the built-in function hist() of MATLAB.
figure; hist(R,100);
ylabel("$ Frequency$",'Interpreter','latex');
xlabel("$ Amplitude$",'Interpreter','latex');
% By selecting a suitable value for the no of bins,
% sketch the conditional PDFs  with the use of the normalized histograms
% Change the value of A and observe the impact.
bins = 100;
Normalized = 1; % to get the probability as the dependant axis
```

Marginal PDF f_{R}(r)

```
[f_R, r, bin_width] = myHistogram(R, bins,Normalized);
ylabel("$f_{R}(r)$",'Interpreter','latex');
xlabel("$ Amplitude$",'Interpreter','latex');
hold on; plot(r, f_R);hold off;
% area of the graph should be 1 to be a PDF
disp("Area of the bar graph = " + num2str(sum(f_R.*bin_width)))
disp("Expected value E[R] = " + num2str(Expected(r, f_R, bin_width)))
```

Conditional PDF f_{R|S}(r|S = A); A = 1

```
valArray1 = getConditionedVlaues(R, pulse_seq, A);
[f_RS1, rs1, bin_width] = myHistogram(valArray1, bins,Normalized);
ylabel("$f_{R|S}(r|S=" +num2str(A)+")$",'Interpreter','latex');
xlabel("$ Amplitude$",'Interpreter','latex');
hold on; plot(rs1, f_RS1);hold off;
disp("Area of the bar graph = " + num2str(sum(f_RS1.*bin_width)))
disp("Expected value E[R|S=A] = " + num2str(Expected(rs1, f_RS1, bin_width)))
```

Conditional PDF f_{R|S}(r|S = -A); A = 1

```
valArray2 = getConditionedVlaues(R, pulse_seq, -A);
[f_RS2, rs2, bin_width] = myHistogram(valArray2, bins,Normalized);
ylabel("$f_{R|S}(r|S=" +num2str(-A)+")$",'Interpreter','latex');
xlabel("$ Amplitude$",'Interpreter','latex');
hold on; plot(rs2, f_RS2);hold off;
disp("Area of the bar graph = " + num2str(sum(f_RS2.*bin_width)))
disp("Expected value E[R|S=-A] = " + num2str(Expected(rs2, f_RS2, bin_width)))
```

Change the value of A to observe the impact

```
position = 1; % subplots position
for A = [1,3,5]
    for i = [1, -1]
        subplot(3,2,position);
        ylabel("$f_{R|S}(r|S=" +num2str(i*A)+")$",'Interpreter','latex')
        R = pulse_seq*A + awgn_seq;   % received signal
        valArray1 = getConditionedVlaues(R, pulse_seq*A, i*A);
        [counts, samples, bin_width] = myHistogram(valArray1, bins,Normalized);
        hold on;
        ylabel("$f_{R|S}(r|S=" +num2str(i*A)+")$",'Interpreter','latex')
        plot(samples, counts);hold off;
        position = position +1;
    end
end
```

6. Now, consider that there is interference I from other transmitters in addition to noise. Thus, the received sample can then be written as R = S + N + I: Assuming that I is also Gaussian distributed with zero mean and variance = 1, repeat step 5 � (b) to (d) and discuss the impact of the addition of interference.

```
% generating AWGN noise and interference
mu = 0;                     % mean
variance = 1;               % variance
sigma = sqrt(variance);     % standard deviation
awgn_seq = normrnd(mu,sigma,[1, L]);
interference = normrnd(mu,sigma,[1, L]);

R = pulse_seq + awgn_seq + interference;   % received signal
```

Marginal PDF f_{R}(r) (under Interference)

```
figure;
[f_R, r, bin_width] = myHistogram(R, bins,Normalized);
ylabel("$f_{R}(r)$",'Interpreter','latex');
xlabel("$ Amplitude$",'Interpreter','latex');
hold on; plot(r, f_R);hold off;
% area of the graph should be 1 to be a PDF
disp("Area of the bar graph = " + num2str(sum(f_R.*bin_width)))
disp("Expected value E[R] = " + num2str(Expected(r, f_R, bin_width)))
```

Conditional PDF f_{R|S}(r|S = A); A = 1 (under Interference)

```
A = 1; figure;
valArray1 = getConditionedVlaues(R, pulse_seq, A);
[f_RS1, rs1, bin_width] = myHistogram(valArray1, bins,Normalized);
ylabel("$f_{R|S}(r|S=" +num2str(A)+")$",'Interpreter','latex');
xlabel("$ Amplitude$",'Interpreter','latex');
hold on; plot(rs1, f_RS1);hold off;
disp("Area of the bar graph = " + num2str(sum(f_RS1.*bin_width)))
```

```
disp("Expected value E[R|S=A] = " + num2str(Expected(rs1, f_RS1, bin_width)))
```

Conditional PDF f_{R|S}(r|S = -A); A = 1 (under Interference)

```
figure;
valArray2 = getConditionedVlaues(R, pulse_seq, -A);
[f_RS2, rs2, bin_width] = myHistogram(valArray2, bins,Normalized);
ylabel("$f_{R|S}(r|S=" +num2str(-A)+")$",'Interpreter','latex');
xlabel("$ Amplitude$",'Interpreter','latex');
hold on; plot(rs2, f_RS2);hold off;
disp("Area of the bar graph = " + num2str(sum(f_RS2.*bin_width)))
disp("Expected value E[R|S=-A] = " + num2str(Expected(rs2, f_RS2, bin_width)))
```

Change the value of A to observe the impact (under Interference)

```
figure;
position = 1; % subplots position
for A = [1,3,5]
    for i = [1, -1]
        subplot(3,2,position);
        ylabel("$f_{R|S}(r|S=" +num2str(i*A)+")$",'Interpreter','latex')
        R = pulse_seq*A + awgn_seq + interference;   % received signal
        valArray1 = getConditionedVlaues(R, pulse_seq*A, i*A);
        [counts, samples, bin_width] = myHistogram(valArray1, bins,Normalized);
        hold on;
        ylabel("$f_{R|S}(r|S=" + num2str(i*A) + ")$",'Interpreter','latex')
        plot(samples, counts);hold off;
        position = position +1;
    end
end
```

7. Finally, consider that the received signal is amplified by a factor of α such that $R = \alpha.S + N$ Repeat step 5 � (b) to (d) and discuss the impact of scaling.

```
A = 1;
bin_seq = randi([0 1], 1, L);
pulse_seq = zeros(1,L); % transmitted signal
for index = 1:L
    if bin_seq(index) == 1
        pulse_seq(index) = +A;
    else
        pulse_seq(index) = -A;
    end
end

figure;
position = 1; % subplots position
for alpha = [1,3,5,7]
    subplot(2,2,position);
    R = alpha*pulse_seq + awgn_seq;
    [f_R, r, bin_width] = myHistogram(R, bins,Normalized);
    hold on;
    ylabel("$f_{R}(r) ~ when ~\alpha = "+num2str(alpha)+"$",'Interpreter','latex');
```

```
    plot(r, f_R);hold off;
    position = position +1;
end
```

User defined functions

1. **Function to generate the Histogram**

```
function [counts, samples, bin_width] = myHistogram(valueArray,numberOfBins, Normalized)
    bins = numberOfBins;
    minR = min(valueArray);
    maxR = max(valueArray);
    bin_width = (maxR - minR)/(bins -1);        % width of a bin
    bin_lower_bound = minR - bin_width/2;       % lower limit of a bin
    bin_upper_bound = minR + bin_width/2;       % upper limit of a bin

    % array to store the number of values in a given bin
    counts = zeros(1,bins);
    samples = minR: bin_width: maxR; % middle of every bar

    for bin = 1:bins
        counts(bin) = 0;
        for sample = valueArray
            if bin_lower_bound <= sample && sample < bin_upper_bound
                counts(bin) = counts(bin) +1;
            end
        end
        bin_lower_bound = bin_upper_bound;
        bin_upper_bound = bin_upper_bound + bin_width;
    end

    counts = counts/bin_width; % to get the height of the bar to plot

    % normalization
    if Normalized
        counts = counts/length(valueArray);
    end
    bar(samples, counts, 'BarWidth', 1) % plotting the histogram
end
```

2. Function to get the PDFs

```
function [valueArray] = getConditionedVlaues(receivedSignal, transmittedSignal, condition)
    valueArray = zeros(1, sum(transmittedSignal == condition));
    index = 1;
    for i = 1:length(transmittedSignal)
        if transmittedSignal(i) == condition
            valueArray(index) = receivedSignal(i);
            index = index +1;
        end
    end
end
```

3. Function to calculate the expected values

```
function [value] = Expected(x, px, bin_width)
    value = sum((x.*px)*bin_width)
end
```