Department of Electronic and Telecommunication Engineering

Faculty of Engineering

University of Moratuwa, Sri Lanka

EN2040 Random Signals and Processes

SIMULATION ASSIGNMENT REPORT

P.M.N.S. Bandara - 180066F

This is submitted as a partial fulfillment for the module

EN2040: Random Signals and Processes

September 12, 2021

List of Figures

List of Tables

Contents

Chapter 1

Introduction

The simulation task is synoptically described as following:

- Rectangular pulses of $\pm A$ 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:

$$S = \begin{cases} +A & \text{if} \quad D = 1 \\ -A & \text{if} \quad D = 0 \end{cases} \tag{1.1}$$

- 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:

$$R = S + N \tag{1.2}$$

  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:

$$Y = \begin{cases} +A & \text{if} \quad \mathrm{R} > \tau \\ -A & \text{if} \quad \mathrm{R} \leq \tau \end{cases} \tag{1.3}$$

In the subsequent implementation stage, several activities have been implemented in order to evaluate the impacts of AWGN, addition of interference and the scaling to the signal transmission.

Chapter 2

Implementation and Results

Problem 1 Generate a binary sequence of length $L = 1000$, considering $D \in \{0, 1\}$ and 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.

The MATLAB implementation for this problem is given in the Figure () in Appendix.

Problem 2 Generate an AWGN sequence, also of length $L = 1000$, considering $\sigma^2 = 1$.

The MATLAB implementation for this problem is given in the Figure 2.31 in Appendix. A possible alternative approach is, in addition, given in the Figure 2.32 in Appendix.

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

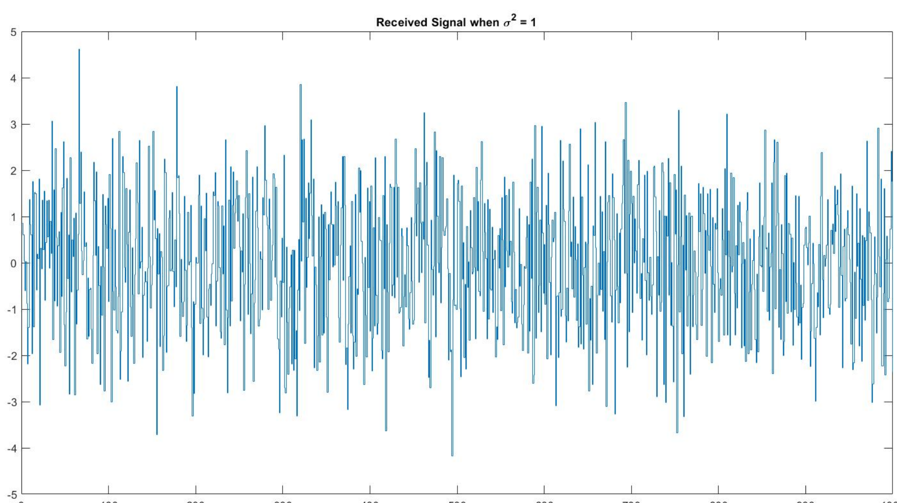
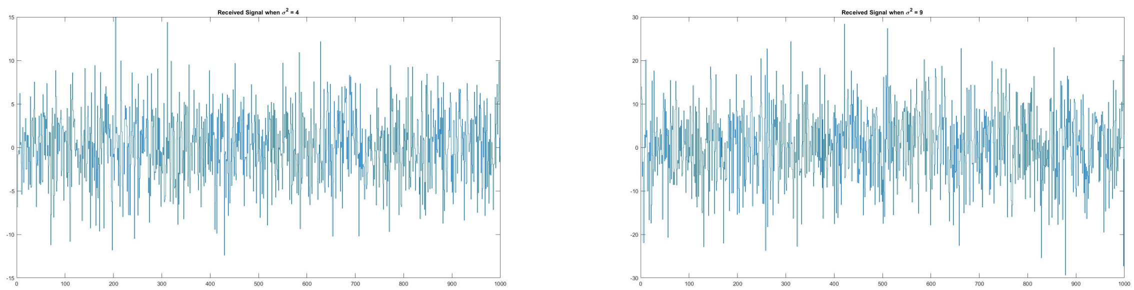


Figure 2.1: Received Signal (R) when $\sigma^2 = 1$



(a) R when $\sigma^2 = 4$ (b) R when $\sigma^2 = 9$

Figure 2.2: The impact of the variance of noise on R by varying σ^2

From Figure 2.1 and 2.2, it is observed that when the variance of the noise (σ^2) increases, the amplitudes of the received signal (R) also increase. The MATLAB implementation for plotting R could be found in the Figure 2.31 in Appendix.

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

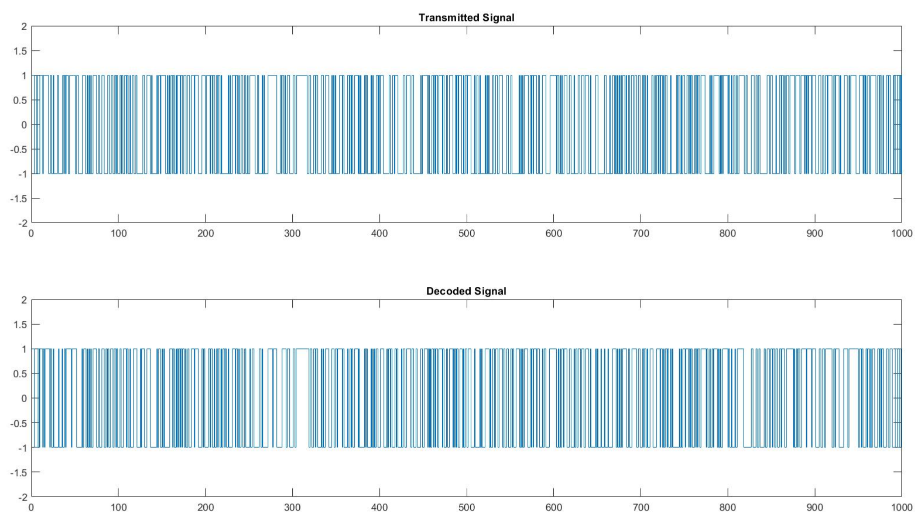


Figure 2.3: Transmitted signal (S) and the Decoded signal (Y)

From Figure 2.3 and 2.4, it is shown that the most of the bits were correctly identified (i.e. decoded - same as the transmitted bit), but there are some observable errors in decoding some of the bits (i.e. some bits were identified to be 1 when the corresponding transmitted bit is 0 and vice versa).

The MATLAB implementation for this problem could be found in the Figure 2.33 in Appemndix.

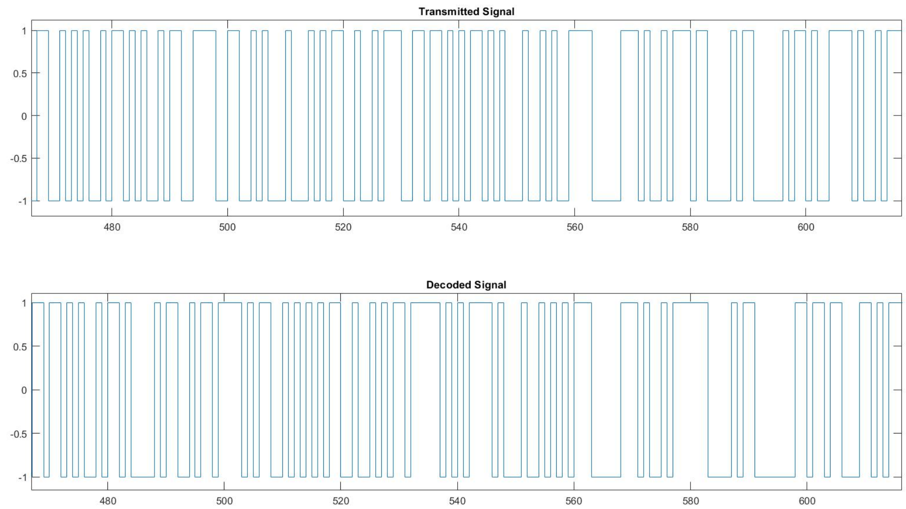


Figure 2.4: Zoomed-In transmitted signal (S) and the decoded signal (Y) for better comparison

Problem 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.

The MATLAB implementation for obtaining the plots in Figure 2.5 and 2.6 is shown in the Figure 2.31 in Appendix by assigning 100000 to L.

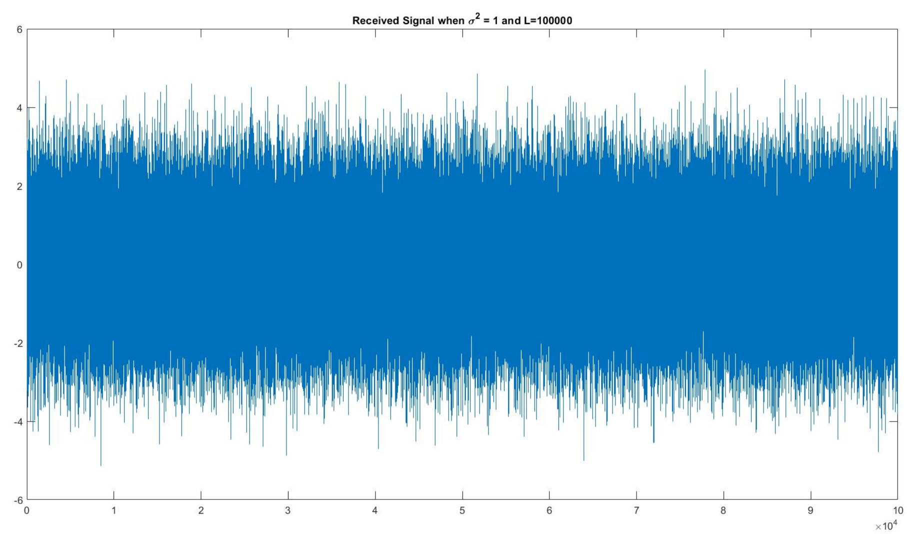


Figure 2.5: Received Signal (R) when $\sigma^2 = 1$ and $L = 100000$

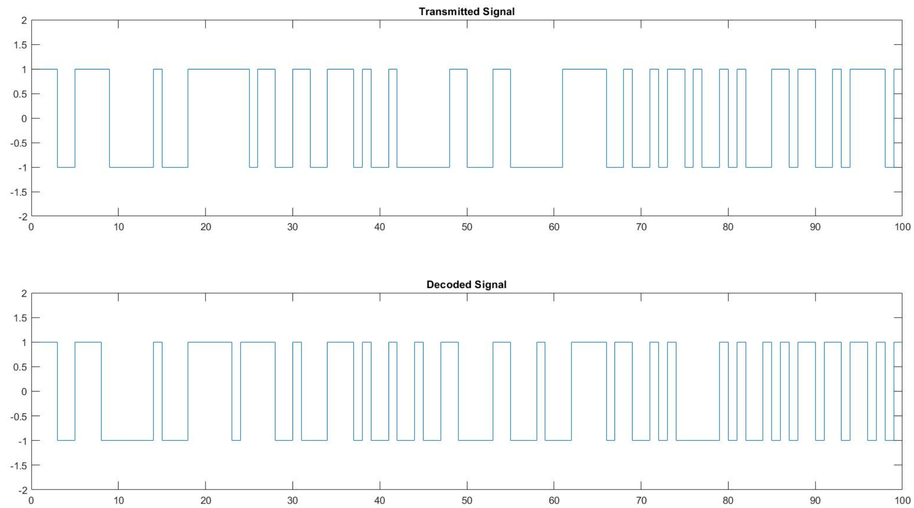
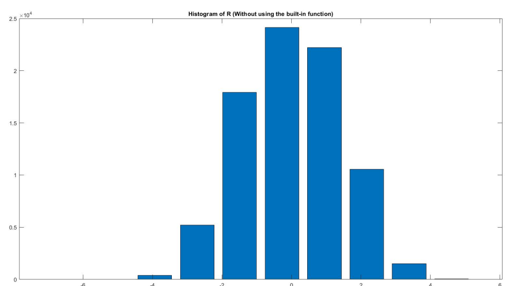
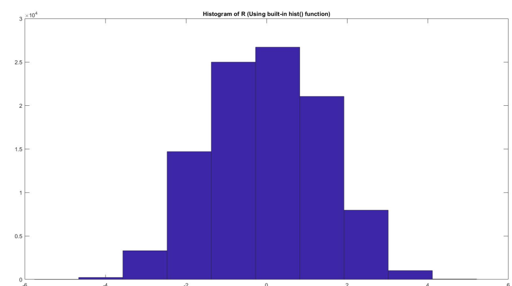


Figure 2.6: Zoomed-In transmitted signal (S) and the decoded signal (Y) for better comparison when $\sigma^2 = 1$ and $L = 100000$



(a) Without the built-in function



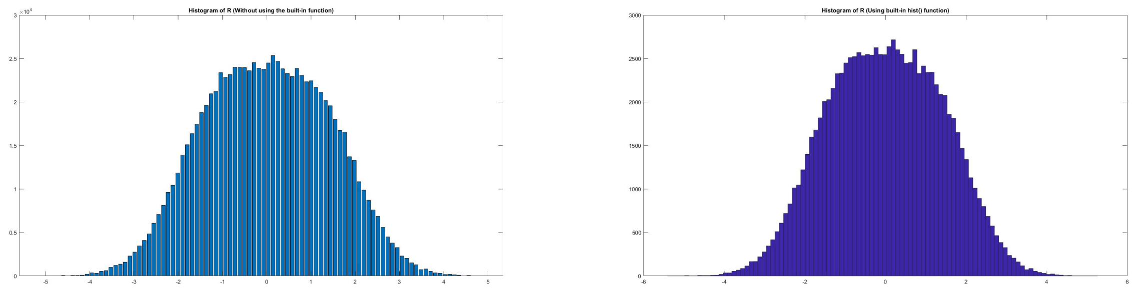
(b) With built-in hist() function

Figure 2.7: Histograms of the received sequences when no of bins = 10

- From Figure 2.6, similar to $L = 1000$, most of the transmitted bits are correctly decoded while there exist some bit errors in decoding.
- From Figure 2.7, it is observed that the both histograms are much similar with slight difference in the corresponding frequency values in each bin (This may be due to the difference in the range of each bin).

The MATLAB implementation for histogram generation could be found in the Figure 2.34 in Appendix.

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



(a) Without the built-in function

(b) With built-in hist() function

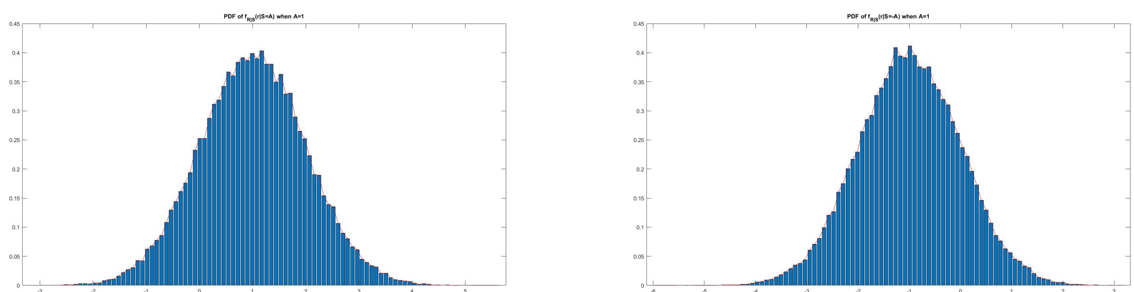
Figure 2.8: Histograms of the received sequences when no of bins = 100

When the number of bins increased from 10 to 100, it is observable that the shapes of the histograms in Figure 2.8 look more similar in shape while having a significant difference in the range of y-axis. This is because the built-in hist() function is presented with normalized frequency values for each bin.

The MATLAB implementation for these histogram generations could be found in Figure 2.34 in Appendix.

(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.

By taking the number of bins as 100:



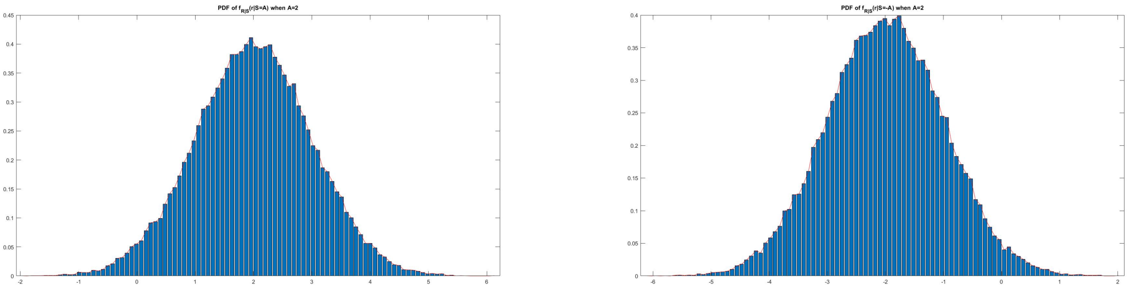
(a) $f_{R|S}(r|S=A)$

(b) $f_{R|S}(r|S=-A)$

Figure 2.9: Conditional probability density functions when $A=1$

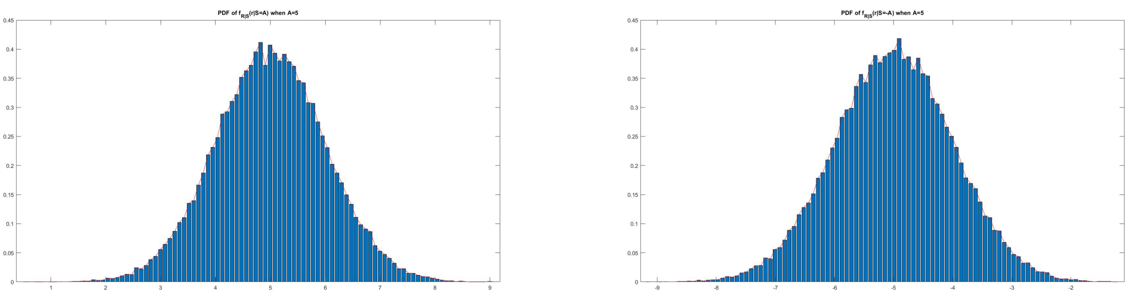
From the Figures 2.9, 2.10 and 2.11, it is observed that when A is increased (from 1 to 5), the mode and the mean of the corresponding conditional PDFs have been shifted away from the zero (i.e. such that when A=5, the mean for $f_{R|S}(r|S=A)$ is around 5 in the x-axis).

The MATLAB implementations for obtaining the Figures 2.9, 2.10 and 2.11 could be found in the Figure 2.35 and 2.36 in Appendix.



(a) $f_{R|S}(r|S=A)$ (b) $f_{R|S}(r|S=-A)$

Figure 2.10: Conditional probability density functions when $A=2$



(a) $f_{R|S}(r|S=A)$ (b) $f_{R|S}(r|S=-A)$

Figure 2.11: Conditional probability density functions when $A=5$

(c) Find $E[R|S=A]$, $E[R|S=-A]$ and $E[R]$.

Expected value could be obtained by:

$$E[X] = \int_0^\infty x f_X(x)\,dx \tag{2.1}$$

Since the case presents a discrete set of points, the expected value could be calculated as:

$$E[X] = \sum_{i=1}^{N} x_i f_{X_i}(x_i)\Delta x_i \tag{2.2}$$

Using a MATLAB calculation (shown in the Figure 2.37 in Appendix), the following expected values are obtained.

Table 2.1: Obtained expected values

A	$E[R\|S=A]$	$E[R\|S=-A]$	$E[R]$
1	0.9979	-0.9960	7.7416×10^{-4}
2	2.0056	-2.0028	0.0017
5	4.9993	-5.0004	-8.5333×10^{-4}

(d) Similarly, sketch the PDF $f_R(r)$.

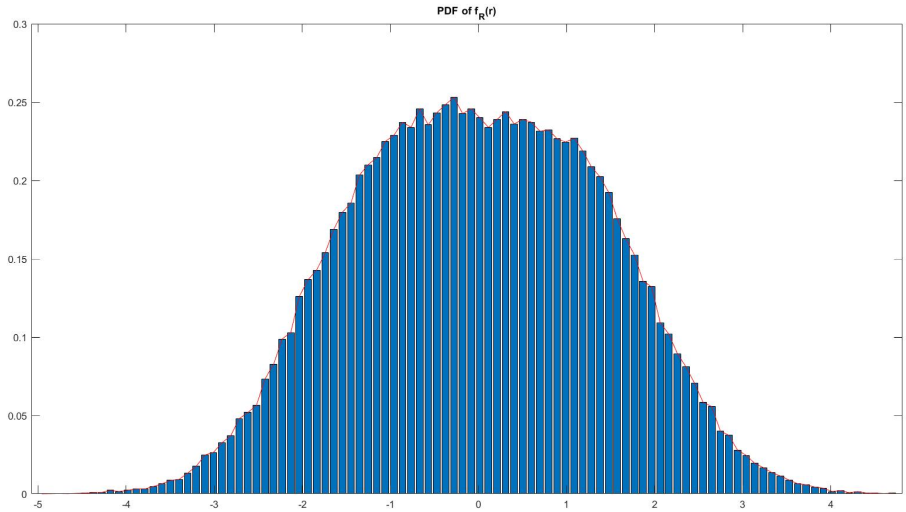


Figure 2.12: PDF $f_R(r)$ when A=1
(MATLAB implementation is attached in the Figure 2.37 in Appendix)

Problem 6 Now, consider that there is interference I from other transmitters in addition to noise. Thus, the received sample can then be written as,

$$R = S + N + I \tag{2.3}$$

Assuming that I is also Gaussian distributed with zero mean and variance $\sigma^2 = 1$, repeat step 5 - (b) to (d) and discuss the impact of the addition of interference.

MATLAB implementation for adding the interference to the received sequence is shown in the Figure 2.38 in Appendix, following the same steps for calculating and plotting the respective PDFs and expected values as shown in the Figure 2.35, 2.36 and 2.37 in Appendix.

When $A = 1$:

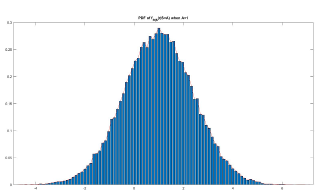


Figure 2.13: $f_{R|S}(r|S = A)$

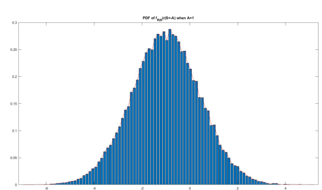


Figure 2.14: $f_{R|S}(r|S = -A)$

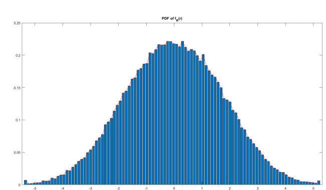


Figure 2.15: PDF $f_R(r)$

When $A = 2$:

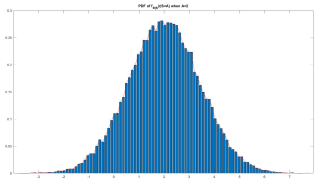


Figure 2.16: $f_{R|S}(r|S=A)$

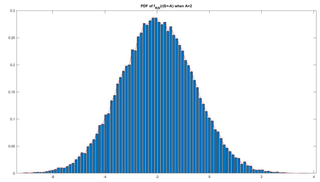


Figure 2.17: $f_{R|S}(r|S=-A)$

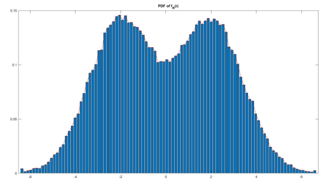


Figure 2.18: PDF $f_R(r)$

When $A=5$:

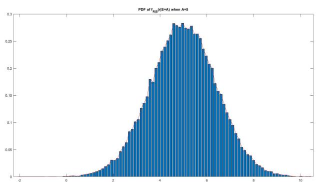


Figure 2.19: $f_{R|S}(r|S=A)$

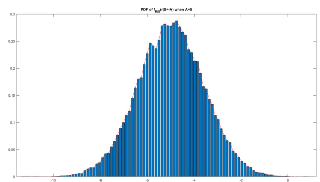


Figure 2.20: $f_{R|S}(r|S=-A)$

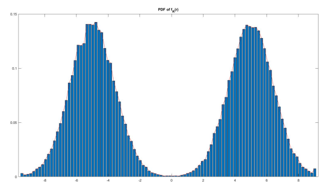


Figure 2.21: PDF $f_R(r)$

Table 2.2: Obtained expected values

A	$E[R\|S=A]$	$E[R\|S=-A]$	$E[R]$
1	0.9945	-1.0084	-0.0071
2	2.0097	-1.9914	0.0091
5	4.9979	-5.0017	-0.0021

By observing the above figures and the table, it is conveyed that the range of x-axis of PDFs in each step (when A is increased) has been increased (due to the increment of the amplitudes of the received signals) when the interference is added. In addition, the shift of mean and mode of the distributions are also present.

It is also observed that when A is increased, a significant change in the PDF of $f_R(r)$.

Problem 7 Finally, consider that the received signal is amplified by a factor of α such that,

$$R = \alpha S + N \tag{2.4}$$

Repeat step 5 - (b) to (d) and discuss the impact of scaling.

When the scaling factor (α) is chosen to be 3 and $A=1$:

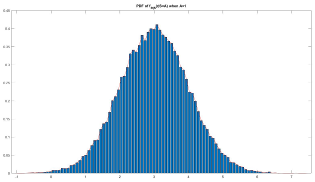


Figure 2.22: $f_{R|S}(r|S=A)$

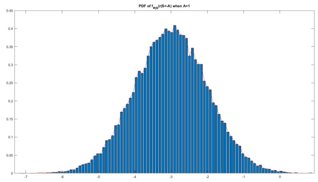


Figure 2.23: $f_{R|S}(r|S=-A)$

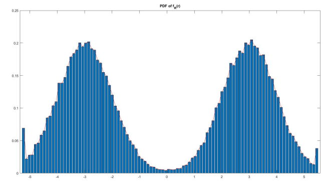


Figure 2.24: PDF $f_R(r)$

When the scaling factor (α) is chosen to be 3 and $A=5$:

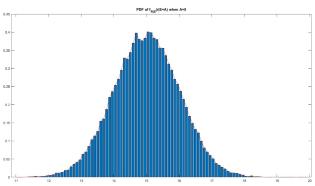


Figure 2.25: $f_{R|S}(r|S=A)$

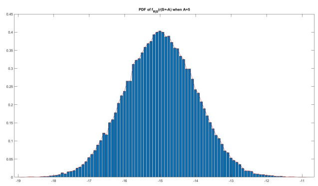


Figure 2.26: $f_{R|S}(r|S=-A)$

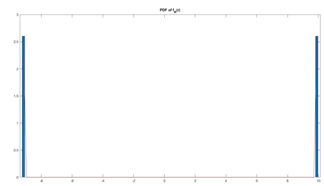


Figure 2.27: PDF $f_R(r)$

When the scaling factor (α) is chosen to be 6 and $A=1$:

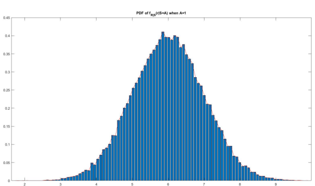


Figure 2.28: $f_{R|S}(r|S=A)$

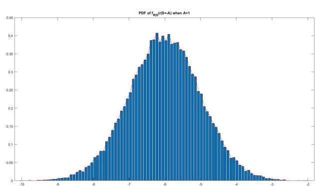


Figure 2.29: $f_{R|S}(r|S=-A)$

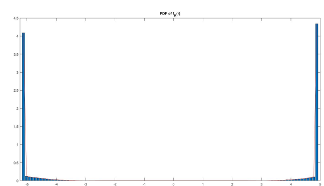


Figure 2.30: PDF $f_R(r)$

Table 2.3: Obtained expected values

α	A	$E[R\|S=A]$	$E[R\|S=-A]$	$E[R]$
3	1	3.0053	-3.0017	0.0029
3	2	15.0085	-14.9933	0.0415
6	1	5.9958	-6.0038	-0.1006

By observing the above figures and the table, it is conveyed that the shift of mean and mode of the distributions are critical as an effect of scaling.

It is also observed that when A and α is increased, a significant change/separation in the shape of the PDF of $f_R(r)$.

MATLAB implementation for scaling in the transmitted sequence is shown in the Figure 2.39 in Appendix, following the same steps for calculating and plotting the respective PDFs and expected values as shown in the Figure 2.35, 2.36 and 2.37 in Appendix.

Appendix

EN2040 Simulation Assignment - MATLAB scripts

```
clc;
clear all;

L = 1000; %L = 100000 for question 5

%generate a equiprobable binary sequence
D = zeros(1,L); %generate a sequence of L zeros
p = randperm(L,L/2); %choose L/2 numbers randomly between 1 and 1000 without replacement
D(p) = ones(1,L/2); %replace the zeros in D with ones in the randomly chosen places
A = 1;

%generate a sequence of pulses
S = zeros(1,L); %generate a sequence of L zeros
for i = 1:L
    if D(i) == 0 %assign -A if D = 0
        S(i) = -1*A;
    else
        S(i) = A; %assign A if D = 1
    end
end

%generate AWGN with mean = 0 and variance = 1
m = 0;
sigma = 1;
N = m + sigma*randn(1,L);

%generate the received signal and plot it
R = S + N;
figure;
stairs([1:L],R);
title("Received Signal when \sigma^2 = 1 and L=1000");
```

Figure 2.31: MATLAB implementation of the generation of the binary sequence and AWGN sequence with L=1000 and $\sigma^2 = 1$

```
%generating AWGN with mean = 0 and variance = 1
m = 0;
sigma = 1;
N = m + sigma*randn(1,L);
%generating the received signal and plotting it
R = S + N;
figure;
stairs([1:L],R);
title("Received Signal");

SNR=snr(R,N);
x2=awgn(S,SNR,'measured');
figure;
stairs([1:L],x2);
title("Received Signal - AWGN");
```

Figure 2.32: MATLAB implementation of a possible alternative to generate the AWGN sequence with L=1000 and $\sigma^2 = 1$

```
%generate Y sequence
tau = 0;
Y = zeros(1,L);
for j = 1:L
    if R(j) > tau
        Y(j) = A;
    else
        Y(j) = -1*A;
    end
end

%plot transmitted signal and Y sequence, and compare
figure;
subplot(2,1,1);
stairs([1:L],S);
title("Transmitted Signal");
xlim([0 L]); %xlim([0 L/1000]) is taken when L = 100000
ylim([-1*A-1 A+1]);
subplot(2,1,2);
stairs([1:L],Y);
title("Decoded Signal");
xlim([0 L]); %xlim([0 L/1000]) is taken when L = 100000
ylim([-1*A-1 A+1]);
```

Figure 2.33: MATLAB implementation of the decoding of the received signal and plotting S and Y sequences

```
%generate the bins sequence
bin_n = 10; %bin_n=100 for question 5(a)
R_max = max(R);
R_min = min(R);
width = (R_max-R_min)/(bin_n-1);
bins = [R_min-width/2:width:R_max];

%count y values for each bin
y_values = zeros(1,bin_n);
for k = 1:L
    for a = 1:bin_n
        if (R(k) >= bins(a)-width/2) && (R(k) < bins(a)+width/2)
            y_values(a) = y_values(a) + 1;
        end
    end
end
new = y_values/width;

%plot the histogram
figure;
bar(bins,new);
title("Histogram of R (Without using the built-in function)");

%use the buit-in function hist()
figure;
hist(R,bin_n);
title("Histogram of R (Using built-in hist() function)");
```

Figure 2.34: MATLAB implementation of generating the histograms of received signals: With and without using built-in hist() function

```matlab
%plot the pdf of f_R|S(r|S=A)
list_R1 = []; %create a list containing R values when S = A
ind = 1;
for b = 1:L
    if S(b) == A
        list_R1(ind) = R(b);
        ind = ind + 1;
    end
end

bin_1 = 100;
R_max1 = max(list_R1);
R_min1 = min(list_R1);
width_1 = (R_max1-R_min1)/(bin_1-1); %set bin width
bins_1 = [R_min1-width_1/2:width_1:R_max1]; %create the bins list
[y_val1,x_val1] = hist(list_R1,bins_1); %plot the histogram
y_val1 = y_val1/((ind-1)*width_1);
figure;
bar(x_val1,y_val1);
hold on;
plot(x_val1,y_val1,'r'); %plot the pdf
title("PDF of f_{R|S}(r|S=A) when A=1");
```

Figure 2.35: MATLAB implementation of plotting $f_{R|S}(r|S=A)$

```matlab
%plot the pdf of f_R|S(r|S=-A)
list_R0 = []; %create a list containing R values when S = -A
ind1 = 1;
for c = 1:L
    if S(c) == -1*A
        list_R0(ind1) = R(c);
        ind1 = ind1 + 1;
    end
end
bin_2 = 100;
R_max2 = max(list_R0);
R_min2 = min(list_R0);
width_2 = (R_max2-R_min2)/(bin_2-1); %set bin width
bins_2 = [R_min2-width_2/2:width_2:R_max2]; %create the bins list
[y_val2,x_val2] = hist(list_R0,bins_2); %plot the histogram
y_val2 = y_val2/((ind1-1)*width_2);
figure;
bar(x_val2,y_val2);
hold on;
plot(x_val2,y_val2,'r'); %plot the pdf
title("PDF of f_{R|S}(r|S=-A) when A=1");
```

Figure 2.36: MATLAB implementation of plotting $f_{R|S}(r|S=-A)$

```
%calculate E[R|S=A]
ER_SA = 0;
for i1 = 1:bin_1
    ER_SA = ER_SA + (x_val1(i1)*y_val1(i1)*width_1);
end
ER_SA

%calculate E[R|S=-A]
ER_SMA = 0;
for i2 = 1:bin_2
    ER_SMA = ER_SMA + (x_val2(i2)*y_val2(i2)*width_2);
end
ER_SMA
%calculate E[R]
[y_val,x_val] = hist(R,bins);
y_val = y_val/(L*width);
E_R = 0;
for i3 = 1:bin_n
    E_R = E_R + (x_val(i3)*y_val(i3)*width);
end
E_R
%plot the pdf of f_R(r)
figure;
bar(x_val,y_val);
hold on;
plot(x_val,y_val,'r');
title("PDF of f_R(r)");
```

Figure 2.37: MATLAB implementation of calculating the required expected values and plotting $f_R(r)$

```
%generate interference
m_i = 0;
sigma_i = 1;
I = m_i + sigma_i*randn(1,L);
%generate the recieved signal
R = S + N + I;
```

Figure 2.38: MATLAB implementation of adding interference signal

```
%scaling factor
alpha = 3;
%generate the recieved signal and plot it
R = alpha*S + N;
```

Figure 2.39: MATLAB implementation of adding the scaling impact to the transmitted signal