Ain Shams University

Faculty of Engineering

Fundamental of Communication Systems

Spring 2023

ECE252s

Fundamentals of communications Project

➢ Team Members

No.	Name	ID
1.	Abdullah Mohamed Ibrahim Hassan	2001826
2.	Rawan Khalid Mohamed Mohamed	2000248
3.	Nihal Osama Abou ElWafa El taftazani	2001314
4.	Areej Ahmed Medhat Mahmoud	2000136
5.	Habiba Mohamed Magdi Hassan	2000015
6.	Jana Maher Fouad Hossny	2000053
7.	Abdelrahman Ali Mostafa	2000312
8.	Ali Akmal Hamed Hamed Abdelrahman	2000895
9.	Ziad Yasser Mohamed Farag	2001377
10.	Menna Khaled Mohamed	1900476

➢ Notes:

- **We have a MATLAB license, so the final code is submitted on MATLAB.**
- **All figures done are zoomed from 10,000 bits to be clearly shown.**

➢ Part 1

Some important definitions:

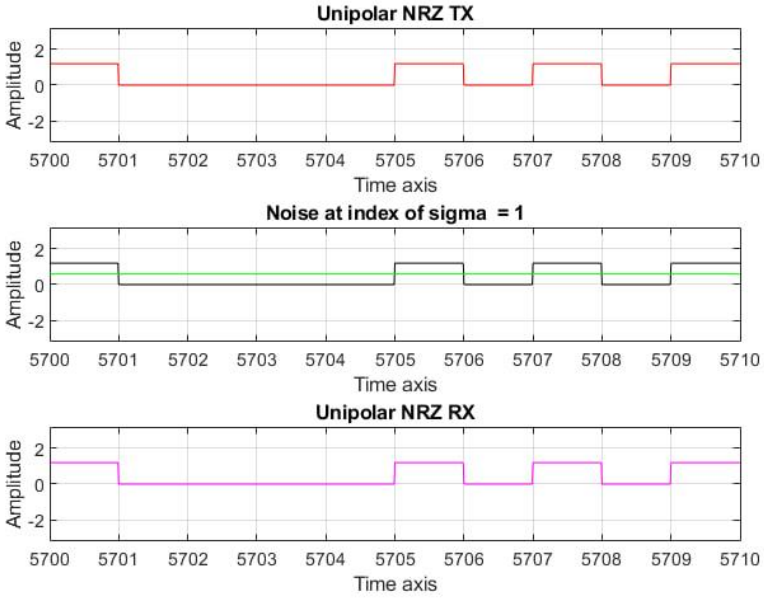
```
clear;                                          %to clear work space
num_of_bits=10000;                              %number of bits which we need
VCC=1.2;                                        %maximum supply voltage
GND=0;                                          %reference supply voltage
VDD=-1.2;                                       %minimum supply voltage
ts=0.01;                                        %sampling time "1 bit = 100 point"
T=10000;                                        %total time
N=ceil(T/ts);                                   %Normalization
df=1/T;                                         %frequency step
fs=1/ts;                                        %sampling frequency
LINE_CODE=randi([0 1],1,num_of_bits);           %random stream of 10k bits
t=0:ts:length(LINE_CODE)-ts;                    %time vector
threshold_UP=0.6*ones(1,length(t));             %decision level
threshold_P=0*ones(1,length(t));                %decision level
threshold_Manchester=0*ones(1,length(t));       %decision level
threshold1_BP=0.6*ones(1,length(t));            %decision level
threshold2_BP=-0.6*ones(1,length(t));           %decision level
if (rem(N,2)==0)                                %if normalization even number
   f=-(0.5*fs):df:(0.5*fs-df);
else                                            %if normalization even number
   f=-(0.5*fs-0.5*df):df:(0.5*fs-0.5*df);
end
sigma=linspace(0,1.2,10);                       %Random noise RMS (10 values)
```

convert line code to UP-NRZ

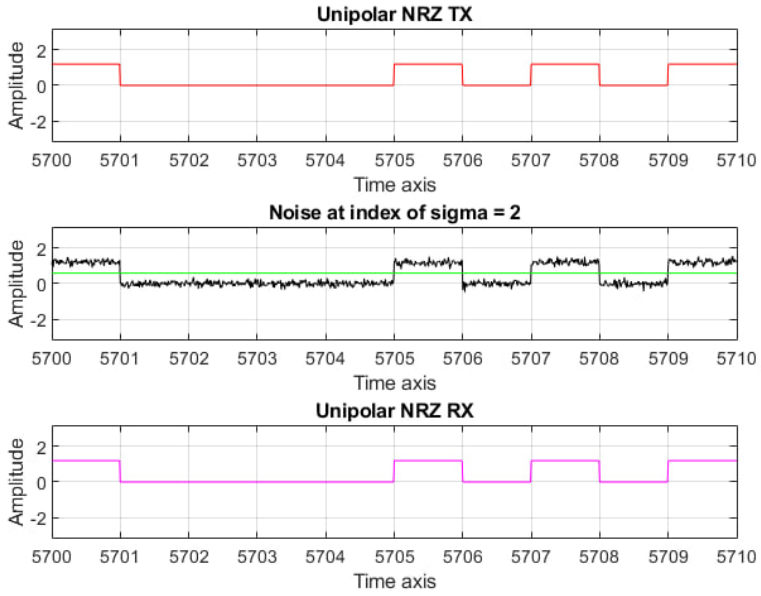
```
for unn=1:10
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Modify the amp. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i = 1:length(LINE_CODE)                     %loop to take all values in LINE_CODE array
    if LINE_CODE(i)==1                          %if current value ==1
        data_UP_NRZ(i)=VCC;                     %convert it to vcc and store it in data array
    else                                        %else "cuurent value ==0"
        data_UP_NRZ(i)=GND;                     %convert it to GND and store it in data array
    end                                         %end if condition
end                                             %end for loop
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% sampling data to be UP-NRZ %%%%%%%%%%%%%%%%%%%%%%%%%%%%
i=1;                                            %intialization i to be 1
for j=1:length(t)                               %for loop in time axis
    if t(j)<i                                   %if t(index)<i "counter"
        UP_NRZ(j)=data_UP_NRZ(i);               %store the same bit 100 times in UP_NRZ array
    else                                        %after 100 times, take the next bit
        i=i+1;                                  %next bit
        UP_NRZ(j)=data_UP_NRZ(i);               %make all bits repeated 100 times exactly
    end                                         %end if condition
end                                             %end for loop
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% TX-SIGNAL"UP-NRZ" %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(unn)                                     %UP_NRZ figure
subplot(3,1,1)                                  %to plot 3 figure
plot(t,UP_NRZ,'r');                             %plot TX signal "stream of 10k bits"
axis([0 length(LINE_CODE) -(VCC+2) (VCC+2)]);    %To improve the overall shape
```

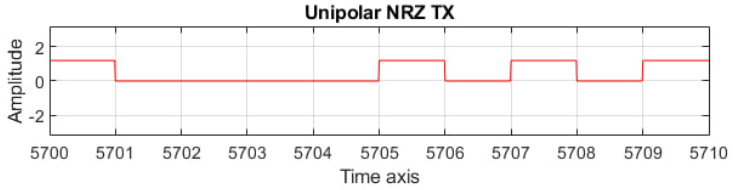
```
xlabel('Time axis');                                %x-axis
ylabel('Amplitude');                                %y-axis
title('Unipolar NRZ TX');                           %title
grid on;                                            %grid on
xlim([5700 5710])                                   %zoom in x-axis
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% ADD NOISE %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
NOISE_UP_NRZ = UP_NRZ +sigma(unn)*randn(size(UP_NRZ));%Add noise
figure(unn)                                                 %UP_NRZ figure
subplot(3,1,2)                                              %to plot 3 figure
plot(t,NOISE_UP_NRZ,'black',t,threshold_UP,'g')             %plot NOISE signal
axis([0 length(LINE_CODE) -(VCC+2) (VCC+2)]);               %To improve the overall shape
xlabel('Time axis');                                        %x-axis
ylabel('Amplitude');                                        %y-axis
title('Noise');
grid on;                                                    %grid on
xlim([5700 5710])                                           %zoom in x-axis
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% RX %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
c=0;                                                        %modify the lenght of RX array
for i=50:100:length(UP_NRZ)                                 %take the mid bits value
    if NOISE_UP_NRZ(i)>threshold_UP                         %"bit>thershold"
        for k=1:100                                         %loop to generate vcc one bit
            RX_UP_NRZ(k+c)=VCC;                             %put vcc value in rx array
        end                                                 %end for loop
    else                                                    %"bit<thershold"
        for k=1:100                                         %loop to generate GND one bit
            RX_UP_NRZ(k+c)=GND;                             %put GND value in rx array
        end                                                 %end for loop
    end                                                     %end for condition
    c=c+100;                                                %to shift the index
end                                                         %end for loop
figure(unn)                                                 %UP_NRZ figure
subplot(3,1,3)                                              %to plot 3 figure
plot(t,RX_UP_NRZ,'m')                                       %plot RX signal
axis([0 length(LINE_CODE) -(VCC+2) (VCC+2)]);               %To improve the overall shape
xlabel('Time axis');                                        %x-axis
ylabel('Amplitude');                                        %y-axis
title('Unipolar NRZ RX');                                   %title
grid on;                                                    %grid on
xlim([5700 5710])                                           %zoom in x-axis
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% BER %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
no_bit_error_UP_NRZ=0;                                      %number of bit error
for i=1:length(RX_UP_NRZ)                                   %to take all value
    if(UP_NRZ(i)~=RX_UP_NRZ(i))                             %to compare between two array
        no_bit_error_UP_NRZ=no_bit_error_UP_NRZ+1;          %count number bit error
    end                                                     %end if condition
end                                                         %end fot loop
no_bit_error_UP_NRZ=no_bit_error_UP_NRZ/fs;                 %normalize number bit error
a_n_bit_error_UP_NRZ(unn)=no_bit_error_UP_NRZ;              %store number bit error each value
of sigma
BER_UP_NRZ(unn)=no_bit_error_UP_NRZ/num_of_bits;            %calculate BER in each case
end                                                          %end loop of sigma
```

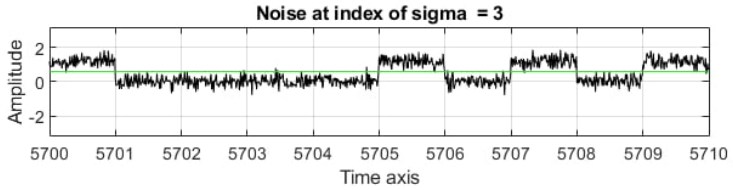
- **Note: this figure is at sigma index=1 (no noise)**

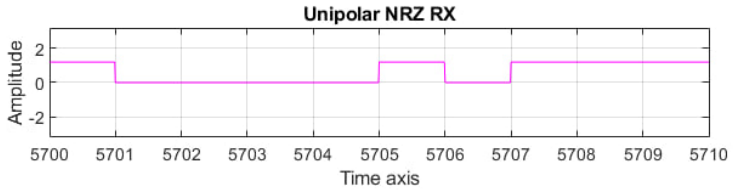


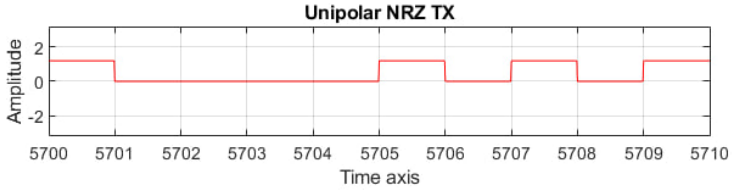
- **All upcoming figures contain noise (where, 9 values of sigma are swept to each figure)**

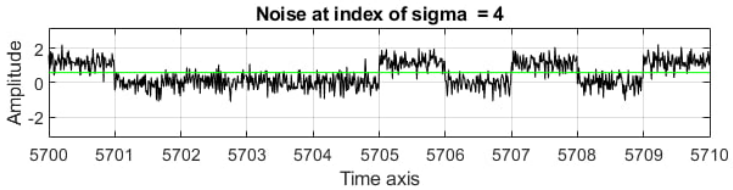


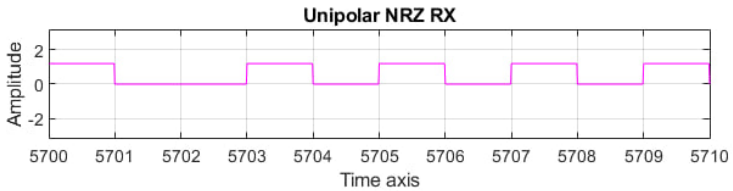

Unipolar NRZ TX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis

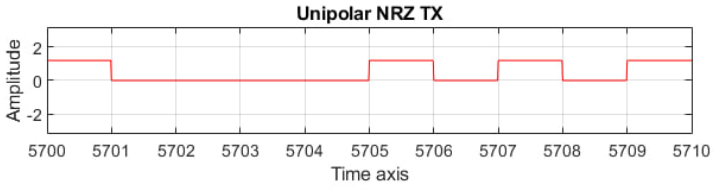

Noise at index of sigma = 3
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis

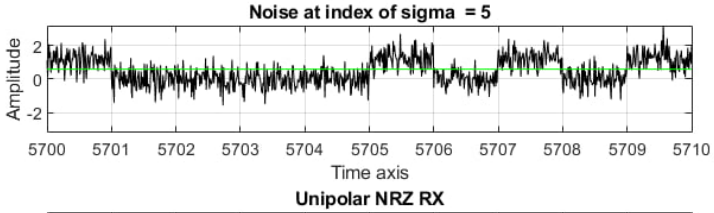

Unipolar NRZ RX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis

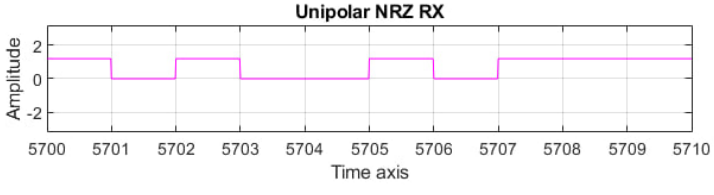

Unipolar NRZ TX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis

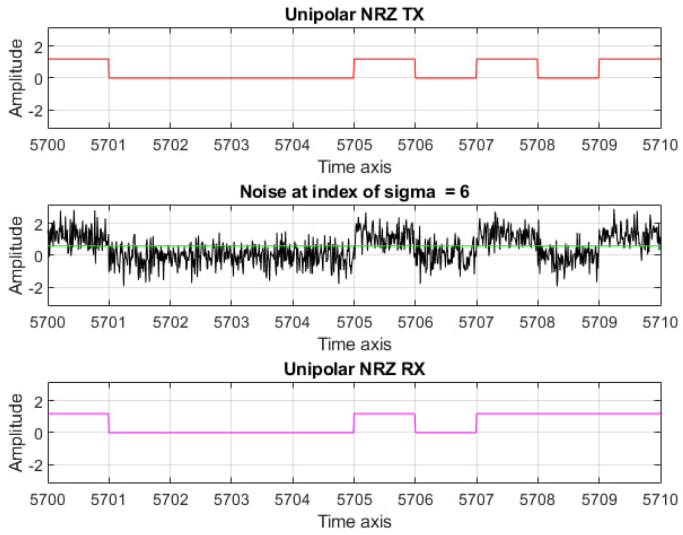

Noise at index of sigma = 4
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis

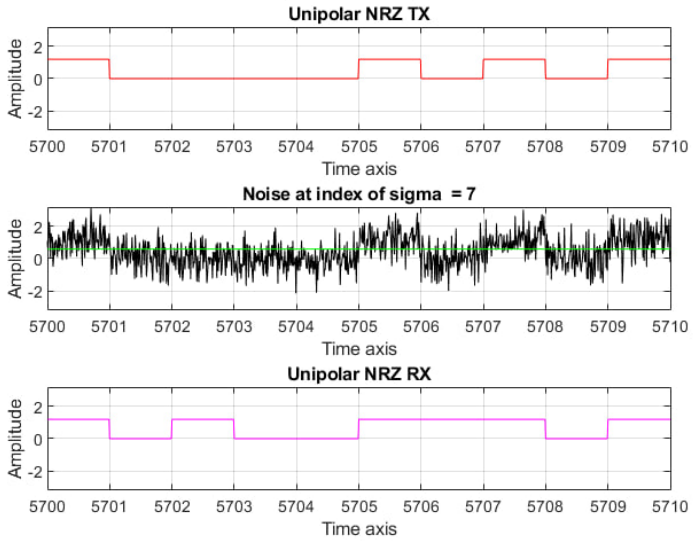

Unipolar NRZ RX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis

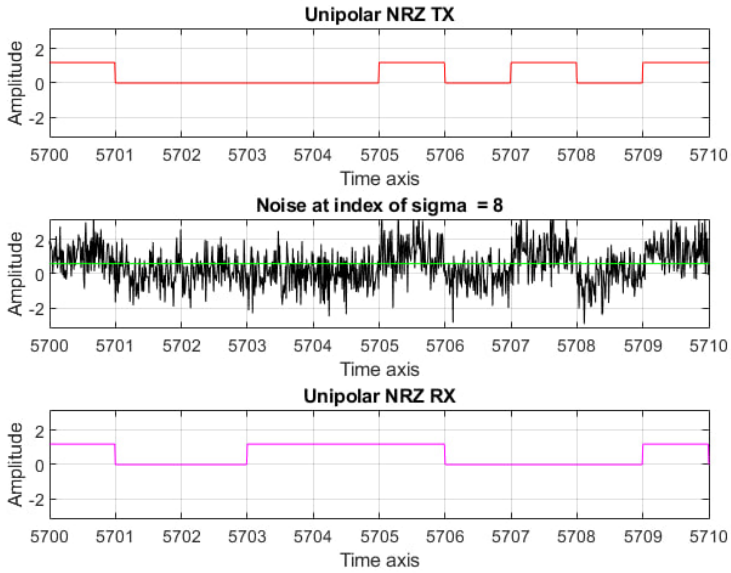

Unipolar NRZ TX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis

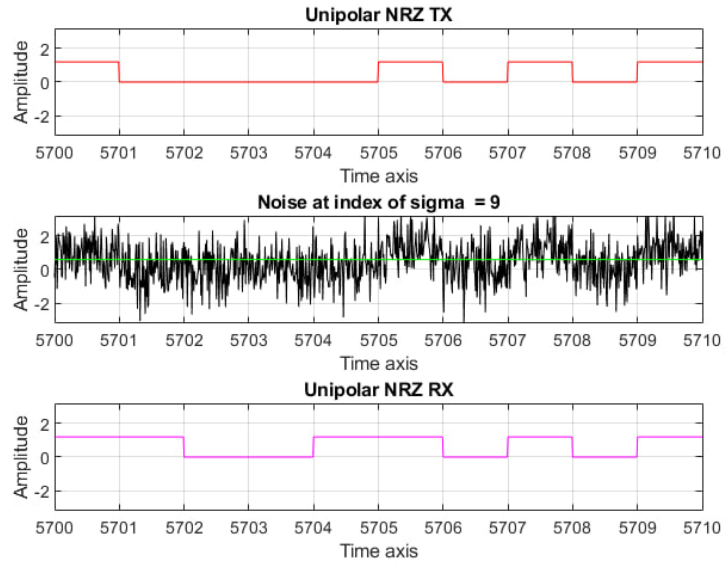

Noise at index of sigma = 5
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis

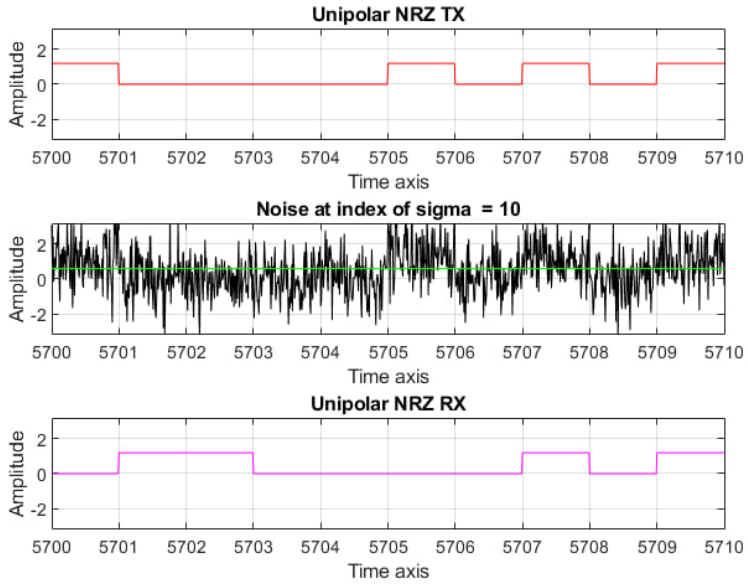

Unipolar NRZ RX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis


Unipolar NRZ TX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis
Noise at index of sigma = 6
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis
Unipolar NRZ RX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis


Unipolar NRZ TX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis
Noise at index of sigma = 7
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis
Unipolar NRZ RX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis


Unipolar NRZ TX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis
Noise at index of sigma = 8
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis
Unipolar NRZ RX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis





- **Number of bits containing errors in every case of sigma**

a_n_bit_error_UP_NRZ

1x10 double

	1	2	3	4	5	6	7	8	9	10
1	0	0	121	664	1348	1863	2274	2617	2934	3049

- **BER at every case of sigma**

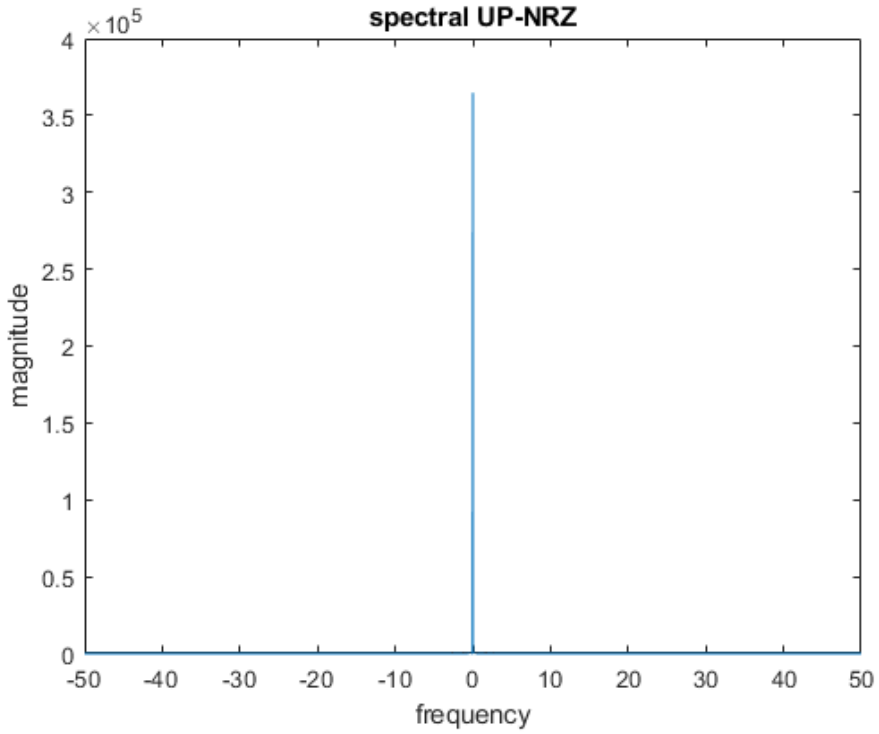
BER_UP_NRZ

1x10 double

	1	2	3	4	5	6	7	8	9	10
1	0	0	0.0121	0.0664	0.1348	0.1863	0.2274	0.2617	0.2934	0.3049

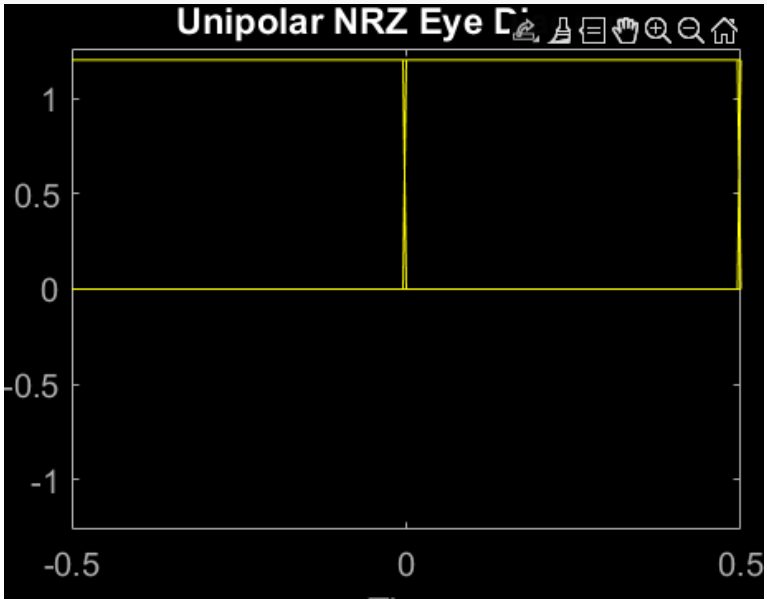
- **Spectral domain**

```
figure(52)
plot(f,(abs(fftshift(fft(UP_NRZ))).^2)/N)
xlabel('frequency');
ylabel('magnitude');
title('spectral UP-NRZ');
```



- **Eye diagram**

```
% Create eye diagram for unipolar nrz
figure(57);
eyediagram(UP_NRZ,200,1,0) ;
xlabel('Time');
ylabel('Amplitude');
title('Unipolar NRZ Eye Diagram');
set(gca, 'FontSize', 15);
```

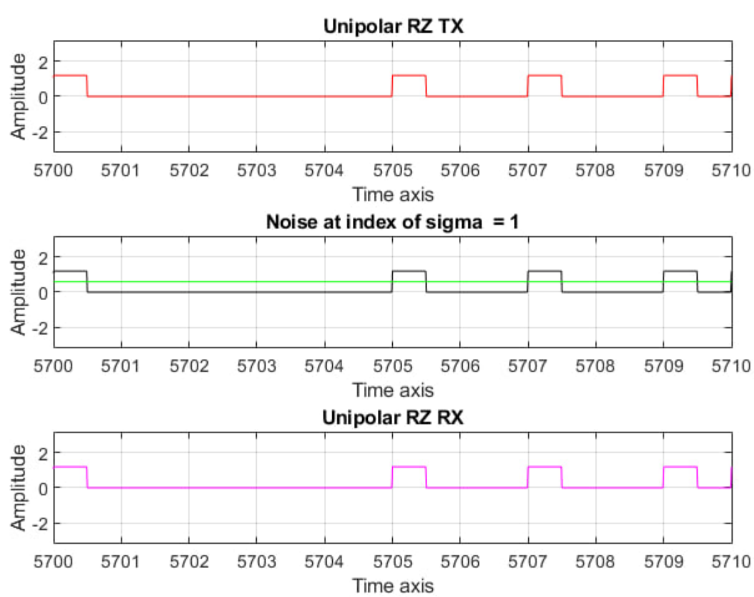


convert line code to UP-RZ:

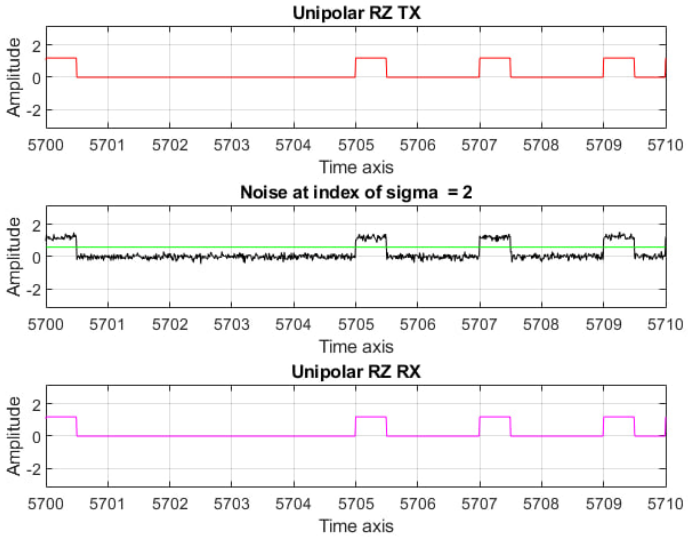
```
for unr=11:20
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Modify the amp. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i = 1:length(LINE_CODE)                %loop to take all values in LINE_CODE array
    if LINE_CODE(i)==1                     %if current value ==1
    data_UP_RZ(i)=VCC;                     %convert it to vcc
    else                                   %else "cuurent value ==0"
    data_UP_RZ(i)=GND;                     %convert it to GND
    end                                    %end if condition
end                                        %end for loop
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% sampling data to be UP-RZ %%%%%%%%%%%%%%%%%%%%%%%%%%
c=0;
for i=1:length(data_UP_RZ)
    if data_UP_RZ(i)==VCC
        for q=1:50
        UP_RZ(q+c)=VCC;
        UP_RZ(q+50+c)=GND;
        end
    else
        for q=1:100
        UP_RZ(q+c)=GND;
        end
    end
    c=c+100;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% TX-SIGNAL"UP-RZ" %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(unr)                                      %UP_NRZ figure
subplot(3,1,1)                                   %to plot 3 figure
plot(t,UP_RZ,'r');                               %plot TX signal "stream of 10k bits"
axis([0 length(LINE_CODE) -(VCC+2) (VCC+2)]);    %To improve the overall shape
xlabel('Time axis');                             %x-axis
ylabel('Amplitude');                             %y-axis
title('Unipolar RZ TX');                         %title
grid on;                                         %grid on
xlim([5700 5710])                                %zoom in x-axis
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% ADD NOISE %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
NOISE_UP_RZ = UP_RZ +sigma(unr-10)*randn(size(UP_RZ));   %Add noise
figure(unr)                                              %UP_RZ figure
subplot(3,1,2)                                           %to plot 3 figure
plot(t,NOISE_UP_RZ,'black',t,threshold_UP,'g')           %plot NOISE signal
axis([0 length(LINE_CODE) -(VCC+2) (VCC+2)]);            %To improve the overall shape
xlabel('Time axis');                                     %x-axis
ylabel('Amplitude');                                     %y-axis
title('Noise');
grid on;                                                 %grid on
xlim([5700 5710])                                        %zoom in x-axis
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% RX %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
c=0;
for i=25:100:length(NOISE_UP_RZ)
    if NOISE_UP_RZ(i)>threshold_UP
        for q=1:50
            RX_UP_RZ(q+c)=VCC;
            RX_UP_RZ(q+50+c)=GND;
        end
```

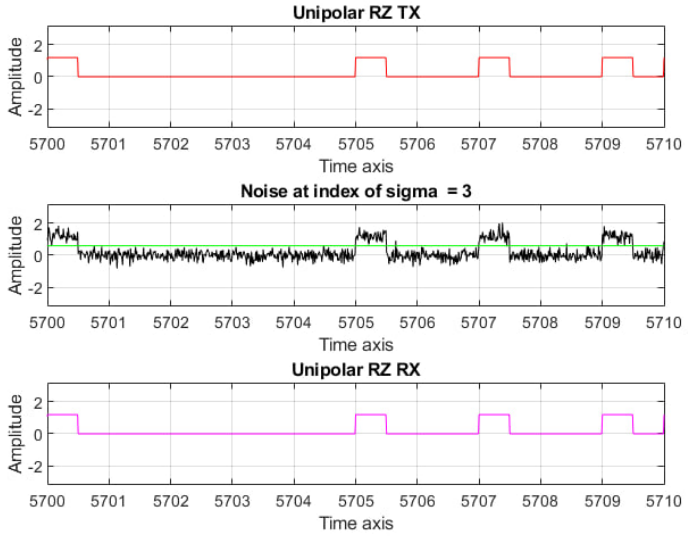
```
    else                                        % "bit<thershold"
        for q=1:100                             %loop to generate GND one bit
            RX_UP_RZ(q+c)=GND;                  %put GND value in rx array
        end                                     %end for loop
    end                                         %end for condition
c=c+100;                                        %to shift the index
end                                             %end for loop
figure(unr)                                     %UP_NRZ figure
subplot(3,1,3)                                  %to plot 3 figure
plot(t,RX_UP_RZ,'m')                            %plot RX signal
axis([0 length(LINE_CODE) -(VCC+2) (VCC+2)]); %To improve the overall shape
xlabel('Time axis');                            %x-axis
ylabel('Amplitude');                            %y-axis
title('Unipolar RZ RX');                        %title
grid on;                                        %grid on
xlim([5700 5710])                               %zoom in x-axis
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% BER %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
no_bit_error_UP_RZ=0;                                  %number of bit error
for i=1:length(RX_UP_RZ)                               %to take all value
    if(UP_RZ(i)~=RX_UP_RZ(i))                          %to compare between two array
        no_bit_error_UP_RZ=no_bit_error_UP_RZ+1;       %count number bit error
    end                                                %end if condition
end                                                    %end fot loop
no_bit_error_UP_RZ=no_bit_error_UP_RZ/fs*2;            %normalize number bit error
a_n_bit_error_UP_RZ(unr-10)=no_bit_error_UP_RZ;        %create array to store number bit
error each value of sigma
BER_UP_RZ(unr-10)=no_bit_error_UP_RZ/num_of_bits;      %calculate BER in each case
end                                                    %end loop of sigma
```

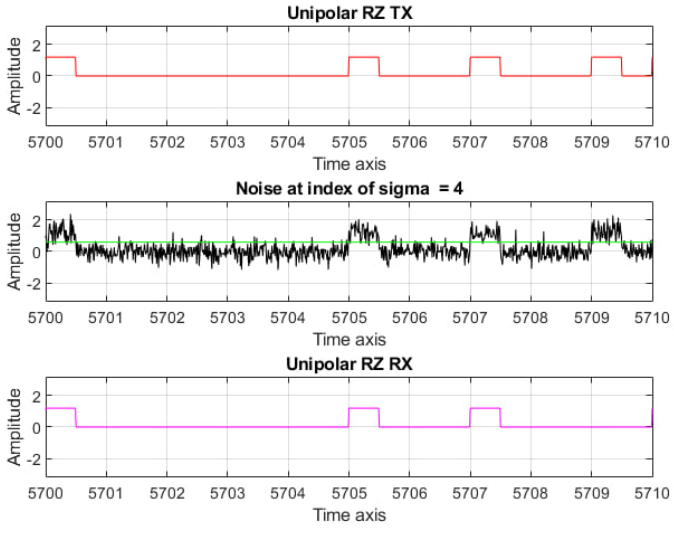
- **Note:** **this figure is at sigma index=1 (no noise)**

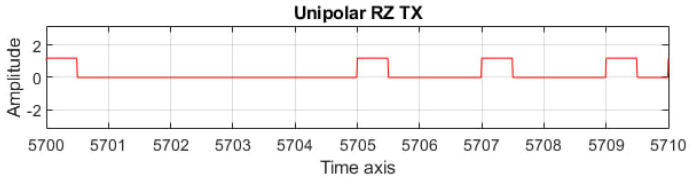


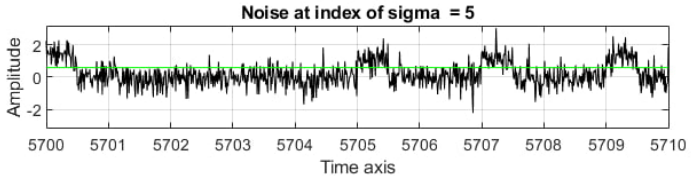
- **All upcoming figures contain noise (where, 9 values of sigma are swept to each figure)**

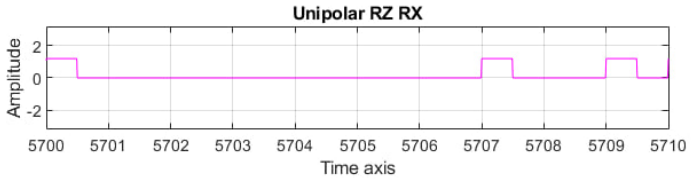

Unipolar RZ TX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis
Noise at index of sigma = 2
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis
Unipolar RZ RX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis

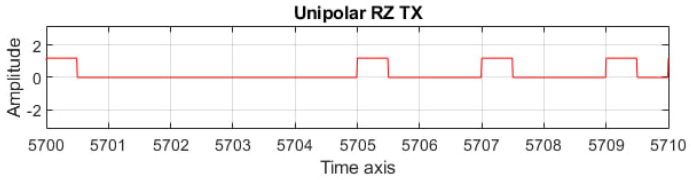

Unipolar RZ TX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis
Noise at index of sigma = 3
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis
Unipolar RZ RX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis

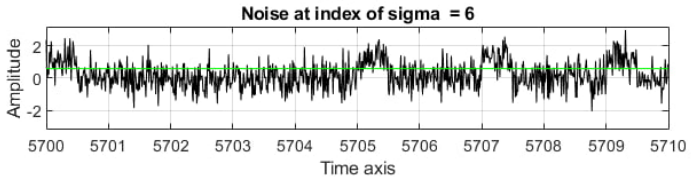

Unipolar RZ TX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis
Noise at index of sigma = 4
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis
Unipolar RZ RX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis

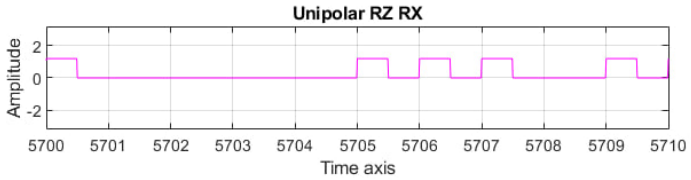

Unipolar RZ TX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis

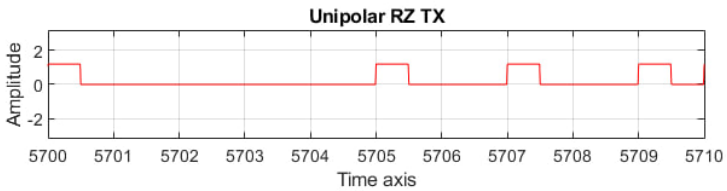

Noise at index of sigma = 5
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis

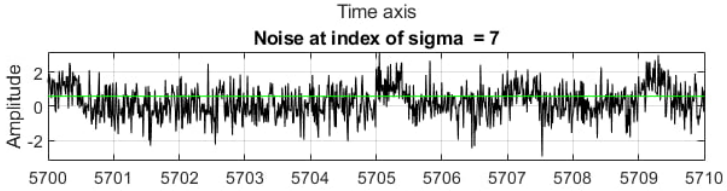

Unipolar RZ RX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis


Unipolar RZ TX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis

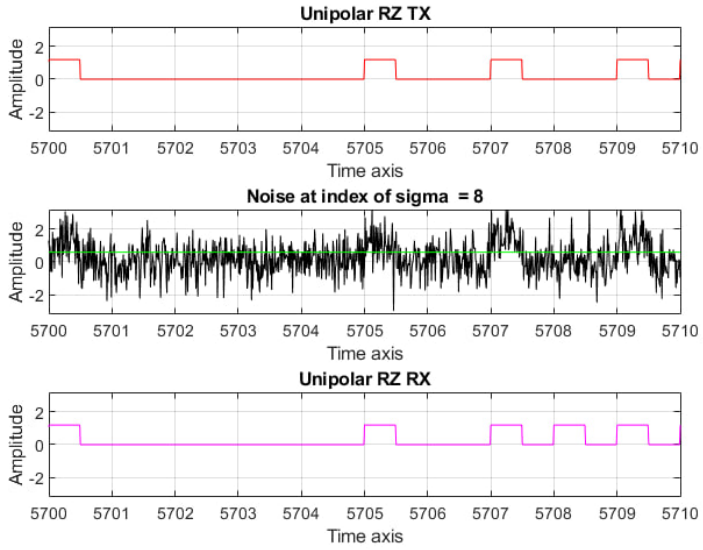

Noise at index of sigma = 6
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis

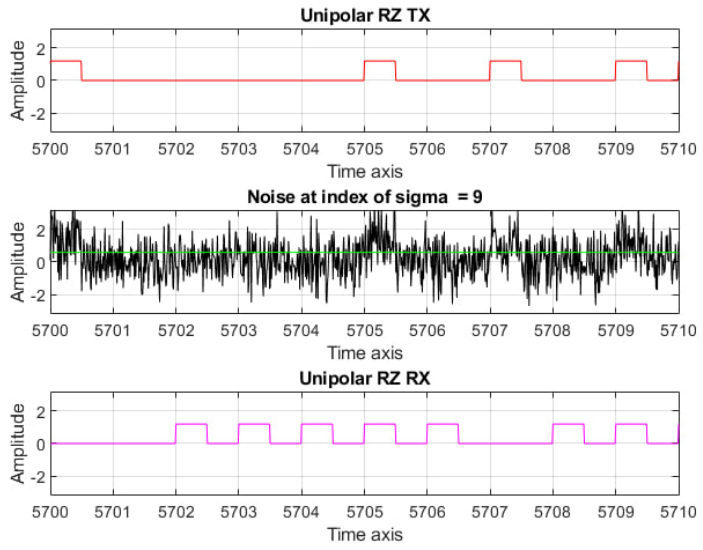

Unipolar RZ RX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis

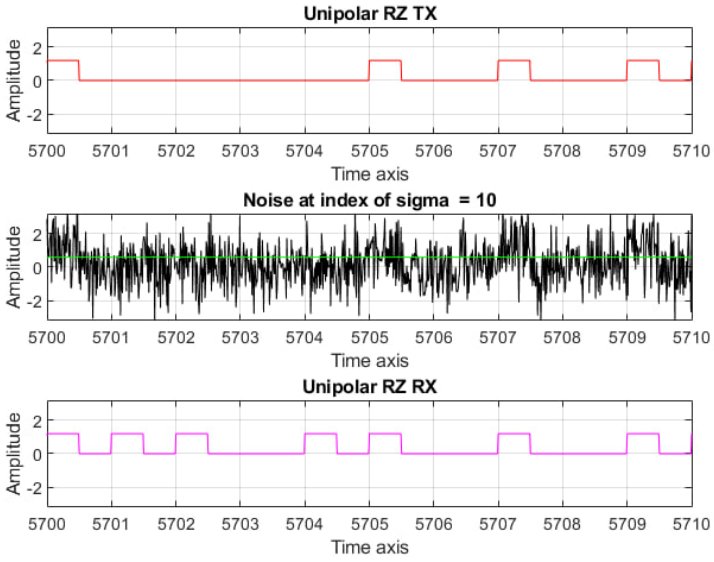

Unipolar RZ TX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis


Noise at index of sigma = 7
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis


Unipolar RZ RX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis


Unipolar RZ TX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis
Noise at index of sigma = 8
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis
Unipolar RZ RX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis


Unipolar RZ TX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis
Noise at index of sigma = 9
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis
Unipolar RZ RX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis


Unipolar RZ TX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis
Noise at index of sigma = 10
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis
Unipolar RZ RX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis

- **Number of bits containing errors in every case of sigma**

a_n_bit_error_UP_RZ

1x10 double

	1	2	3	4	5	6	7	8	9	10
1	0	0	119	685	1267	1884	2298	2583	2976	3099

- **BER at every case of sigma**

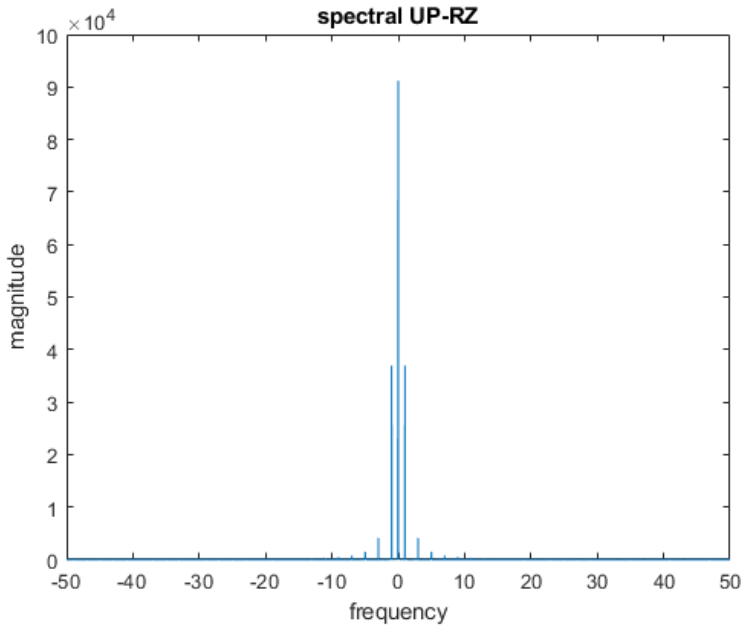
BER_UP_RZ

1x10 double

	1	2	3	4	5	6	7	8	9	10
1	0	0	0.0119	0.0685	0.1267	0.1884	0.2298	0.2583	0.2976	0.3099

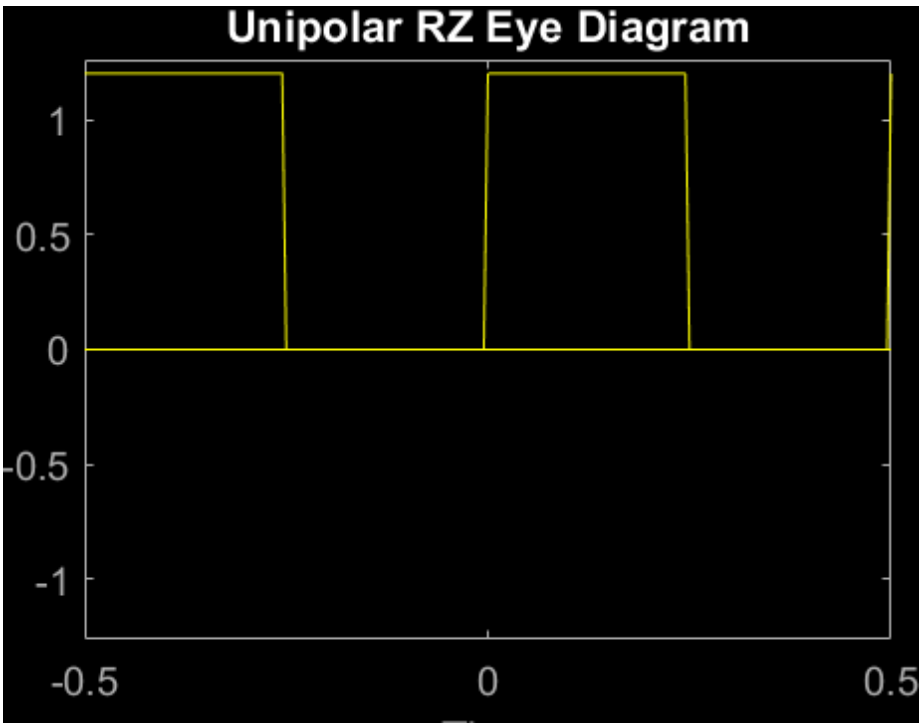
- **Spectral domain**

```
figure(53)
plot(f,(abs(fftshift(fft(UP_RZ))).^2)/N)
xlabel('frequency');
ylabel('magnitude');
title('spectral UP-RZ');
```



- **Eye diagram**

```
% Create eye diagram for Unipolar RZ
figure(59);
eyediagram(UP_RZ,200,1,0) ;
xlabel('Time');
ylabel('Amplitude');
title('Unipolar RZ Eye Diagram');
set(gca, 'FontSize', 15);
```

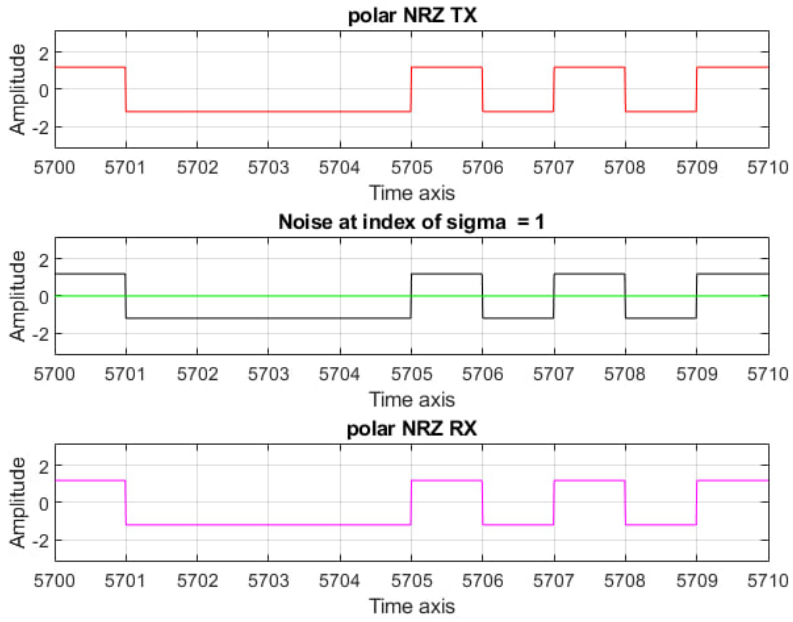


convert line code to P-NRZ:

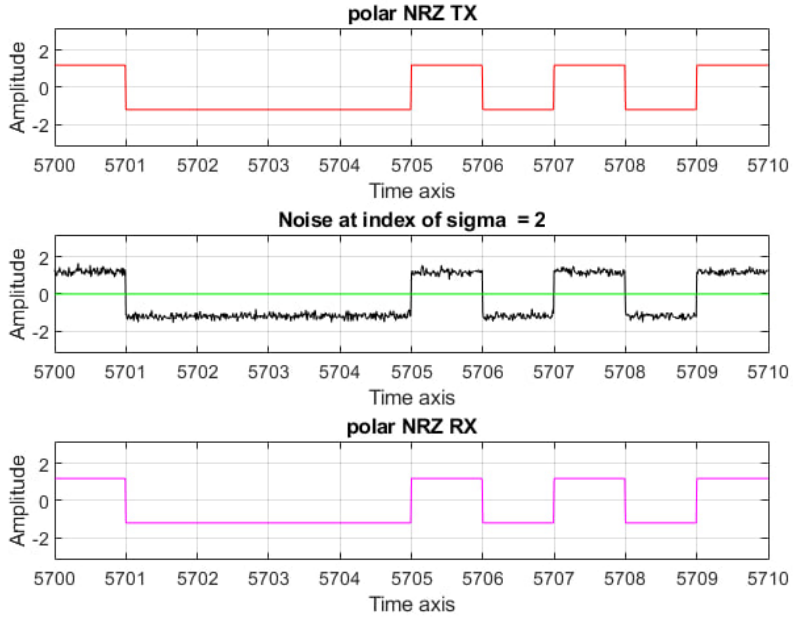
```
for pn=21:30
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Modify the amp.%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i = 1:length(LINE_CODE)      %loop to take all values in LINE_CODE array
    if LINE_CODE(i)==1           %if current value ==1
        data_P_NRZ(i)=VCC;       %convert it to vcc and store it in data array
    else                         %else "cuurent value ==0"
        data_P_NRZ(i)=VDD;       %convert it to VDD and store it in data array
    end                          %end if condition
end                              %end for loop
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%sampling data to be P-NRZ%%%%%%%%%%%%%%%%%%%%%%%%%%%%
i=1;                             %intialization i to be 1
for j=1:length(t)                %for loop in time axis
    if t(j)<i                    %if t(index)<i "counter"
        P_NRZ(j)=data_P_NRZ(i);  %store the same bit 100 times in P_NRZ array
    else                         %after 100 times, take the next bit
        i=i+1;                   %next bit
        P_NRZ(j)=data_P_NRZ(i);  %make all bits repeated 100 times exactly
    end                          %end if condition
end                              %end for loop

```

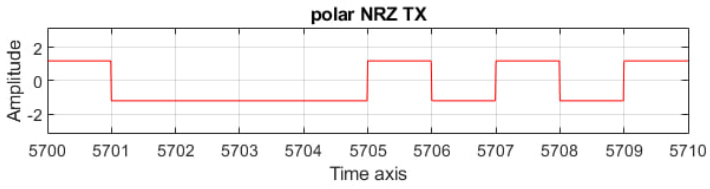
```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%TX-SIGNAL"P-NRZ"%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(pn)                                                  %P_NRZ figure
subplot(3,1,1)                                              %to plot 3 figure
plot(t,P_NRZ,'r');                                          %plot TX signal "stream of 10k bits"
axis([0 length(LINE_CODE) -(VCC+2) (VCC+2)]);               %To improve the overall shape
xlabel('Time axis');                                        %x-axis
ylabel('Amplitude');                                        %y-axis
title('polar NRZ TX');                                      %title
grid on;                                                    %grid on
xlim([5700 5710])                                           %zoom in x-axis
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%ADD NOISE%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
NOISE_P_NRZ = P_NRZ +sigma(pn-20)*randn(size(P_NRZ));%Add noise
figure(pn)                                                  %P_NRZ figure
subplot(3,1,2)                                              %to plot 3 figure
plot(t,NOISE_P_NRZ,'black',t,threshold_P,'g')               %plot NOISE signal
axis([0 length(LINE_CODE) -(VCC+2) (VCC+2)]);               %To improve the overall shape
xlabel('Time axis');                                        %x-axis
ylabel('Amplitude');                                        %y-axis
title('Noise');
grid on;                                                    %grid on
xlim([5700 5710])                                           %zoom in x-axis
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% RX %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
c=0;                                                    %modify the lenght of RX array
for i=50:100:length(P_NRZ)                              %take the mid bits value
    if NOISE_P_NRZ(i)>threshold_P                       %"bit>thershold"
        for k=1:100                                     %loop to generate vcc one bit
            RX_P_NRZ(k+c)=VCC;                          %put vcc value in rx array
        end                                             %end for loop
    else                                                %"bit<thershold"
        for k=1:100                                     %loop to generate VDD one bit
            RX_P_NRZ(k+c)=VDD;                          %put VDD value in rx array
        end                                             %end for loop
    end                                                 %end conditions
    c=c+100;                                            %to shift the index
end                                                     %end for loop
figure(pn)                                              %P_NRZ figure
subplot(3,1,3)                                          %to plot 3 figure
plot(t,RX_P_NRZ,'m')                                    %plot RX signal
axis([0 length(LINE_CODE) -(VCC+2) (VCC+2)]);           %To improve the overall shape
xlabel('Time axis');                                    %x-axis
ylabel('Amplitude');                                    %y-axis
title('polar NRZ RX');                                  %title
grid on;                                                %grid on
xlim([5700 5710])                                       %zoom in x-axis
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% BER %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
no_bit_error_P_NRZ=0;                                    %error number of bits "after transmitter"
for i=1:length(RX_P_NRZ)                                 %to take all value
    if(P_NRZ(i)~=RX_P_NRZ(i))                            %to compare between two array
        no_bit_error_P_NRZ=no_bit_error_P_NRZ+1;         %counter to count number bit error
    end                                                  %end if condition
end                                                      %end for loop
no_bit_error_P_NRZ=no_bit_error_P_NRZ/fs;                %normalize number bit error
a_n_bit_error_P_NRZ(pn-20)=no_bit_error_P_NRZ;
BER_P_NRZ(pn-20)=no_bit_error_P_NRZ/num_of_bits;         %calculate BER in each case
end                                                      %end loop of sigma
```

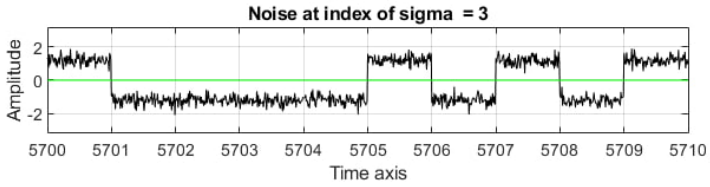
- **Note:** **this figure is at sigma index=1 (no noise)**

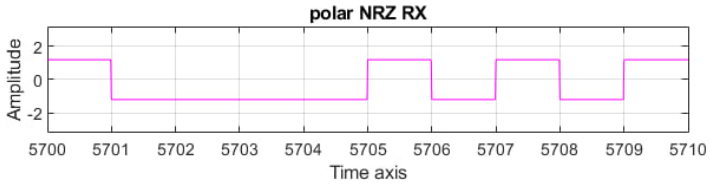


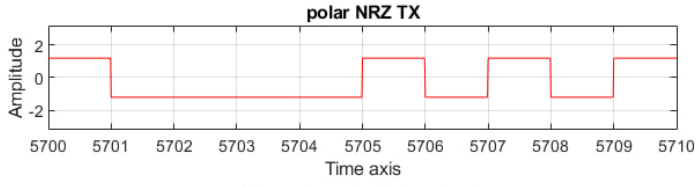
- **All upcoming figures contain noise (where, 9 values of sigma are swept to each figure)**

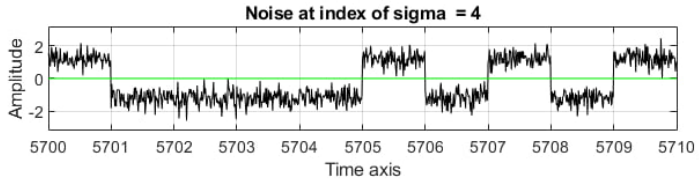


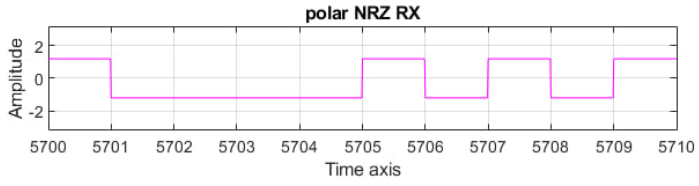

polar NRZ TX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis

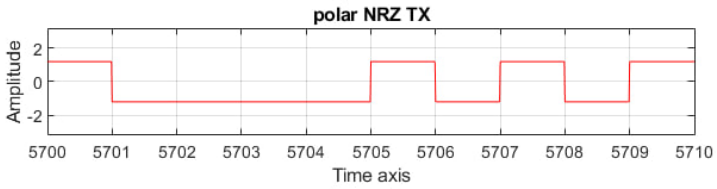

Noise at index of sigma = 3
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis

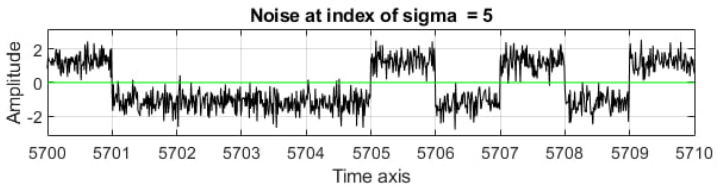

polar NRZ RX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis

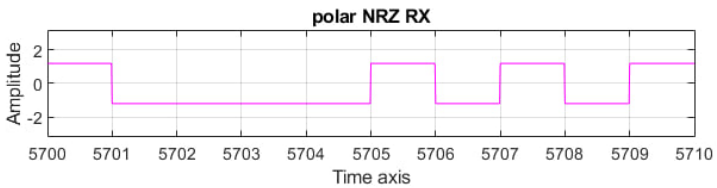

polar NRZ TX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis

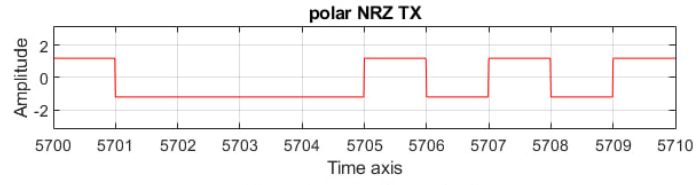

Noise at index of sigma = 4
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis

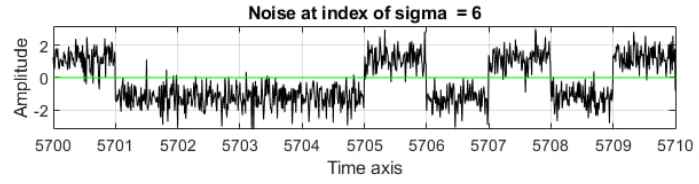

polar NRZ RX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis

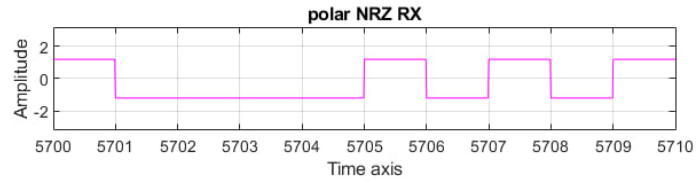

polar NRZ TX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis

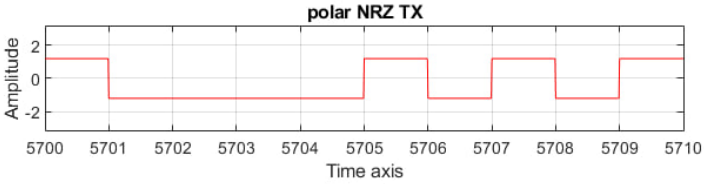

Noise at index of sigma = 5
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis

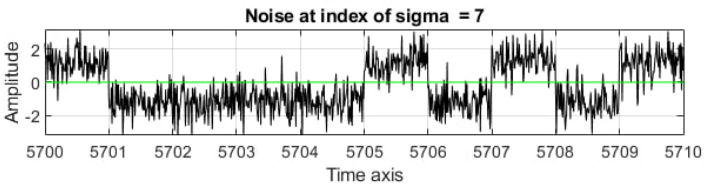

polar NRZ RX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis

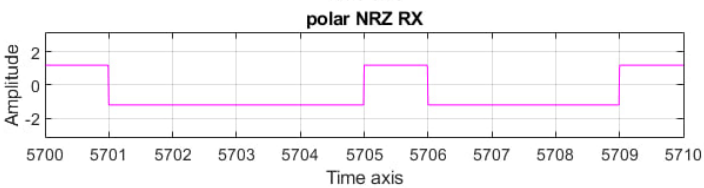

polar NRZ TX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis

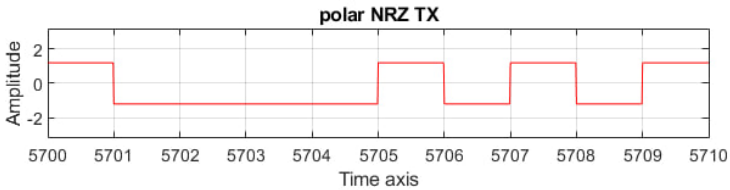

Noise at index of sigma = 6
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis

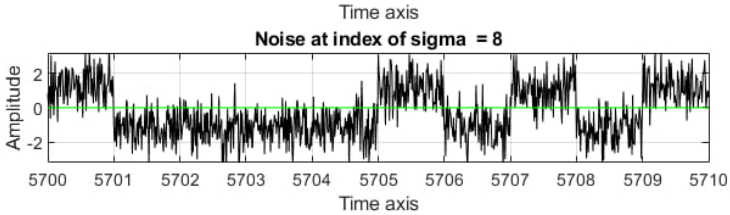

polar NRZ RX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis

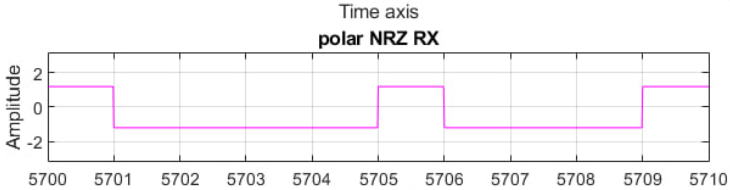

polar NRZ TX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis

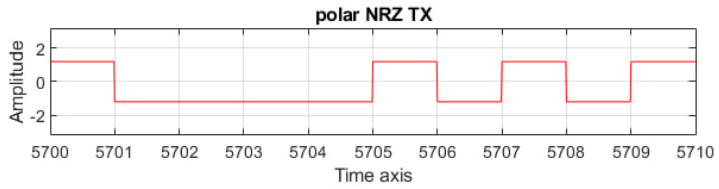

Noise at index of sigma = 7
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis

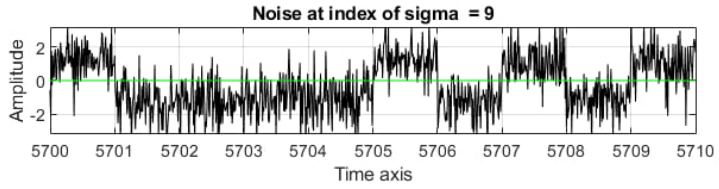

polar NRZ RX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis

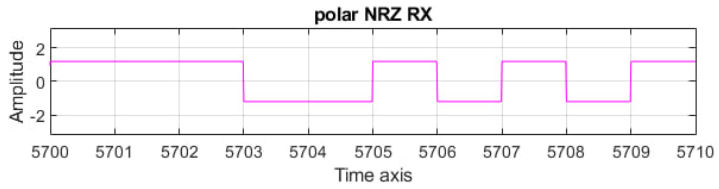

polar NRZ TX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis

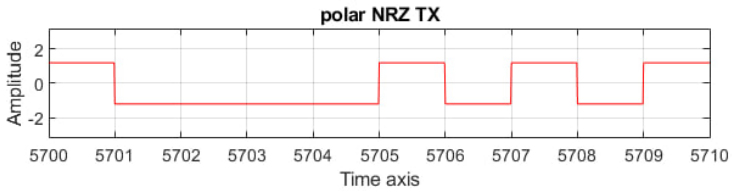

Noise at index of sigma = 8
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis

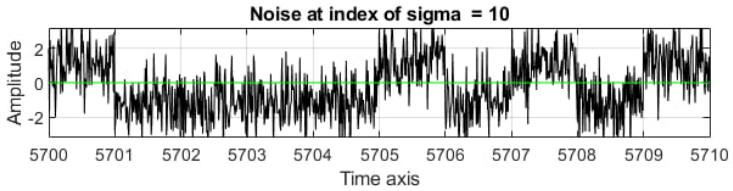

polar NRZ RX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis

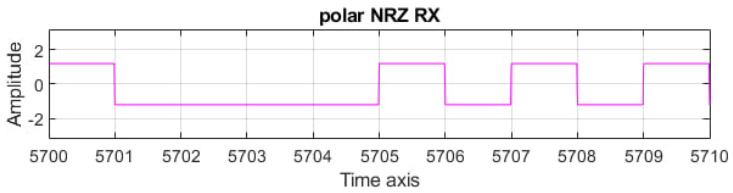












- **Number of bits containing errors in every case of sigma**

a_n_bit_error_P_NRZ

1x10 double

	1	2	3	4	5	6	7	8	9	10
1	0	0	0	19	118	346	651	968	1273	1627

- **BER at every case of sigma**

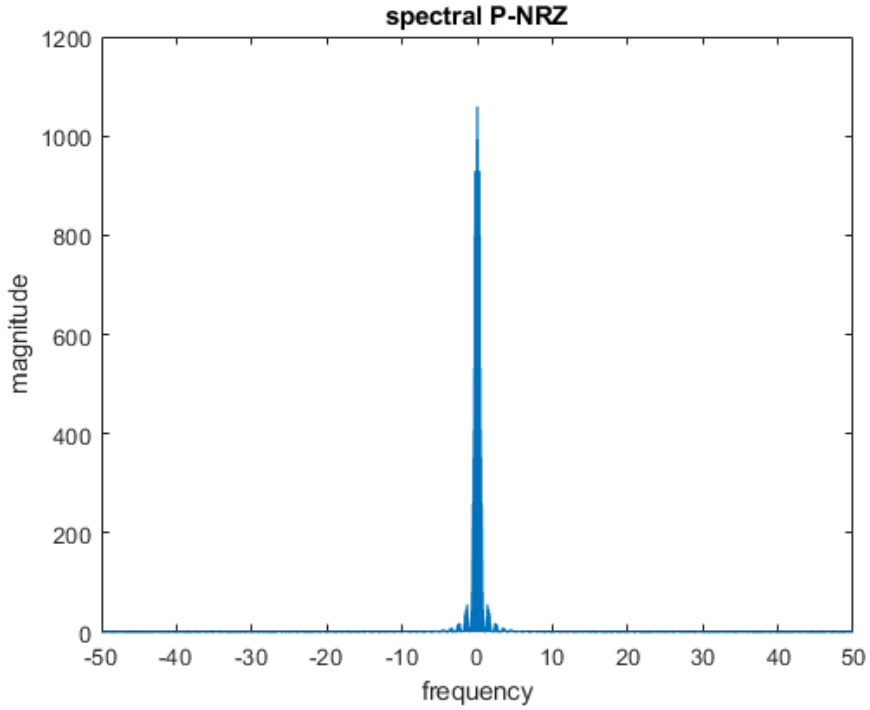
BER_P_NRZ

1x10 double

	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0.0019	0.0118	0.0346	0.0651	0.0968	0.1273	0.1627

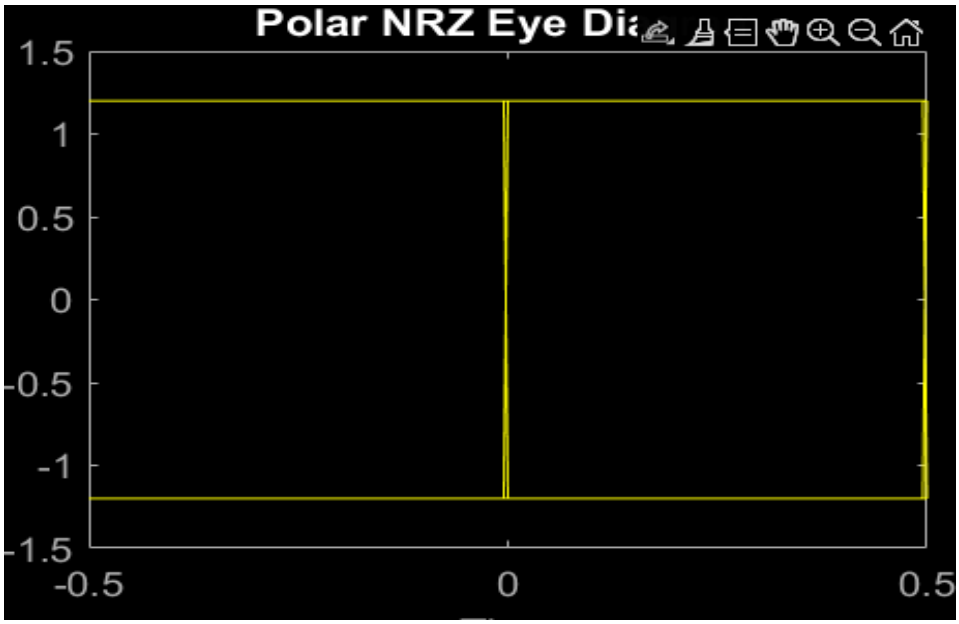
- **Spectral domain**

```
figure(54)
plot(f,(abs(fftshift(fft(P_NRZ))).^2)/N)
xlabel('frequency');
ylabel('magnitude');
title('spectral P-NRZ');
```



- **Eye diagram**

```
% Create eye diagram for Polar NRZ
figure(58);
eyediagram(P_NRZ,200,1,0) ;
xlabel('Time');
ylabel('Amplitude');
title('Polar NRZ Eye Diagram');
set(gca, 'FontSize', 15);
```

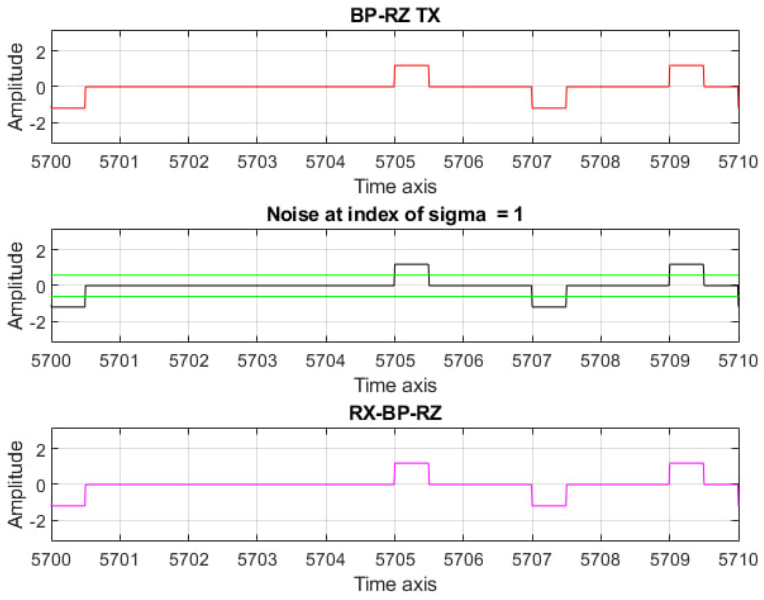


convert line code to BP-RZ:

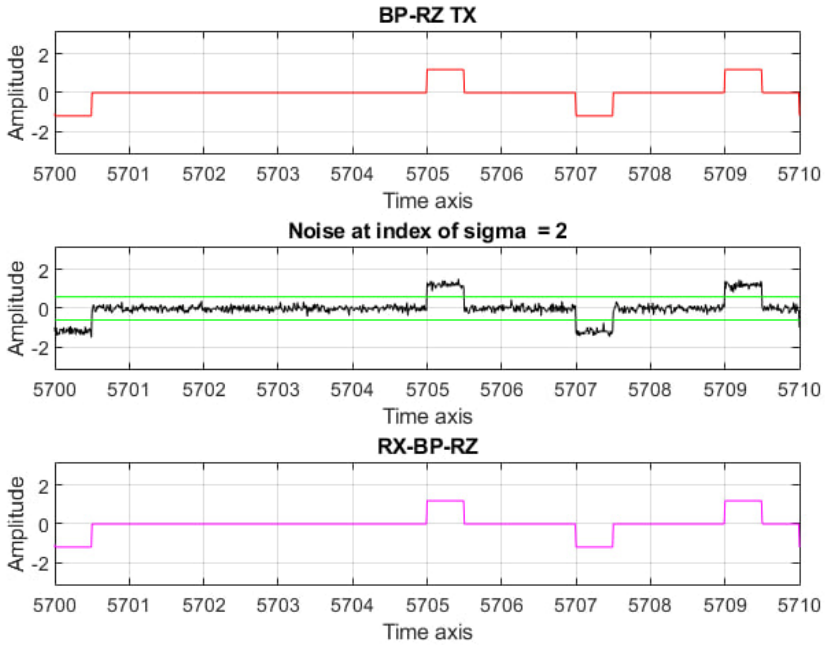
```
for bp=31:40
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Modify the amp. %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
LINE_CODE_BP=LINE_CODE;
for w=1:length(LINE_CODE_BP)
    if LINE_CODE_BP(w)==1
        LINE_CODE_BP(w)=1.2;
        break;
    end
end
for i = 1:length(LINE_CODE_BP)
    if LINE_CODE_BP(i)==0
        continue
    elseif LINE_CODE_BP(i)==1.2
        continue
    elseif LINE_CODE_BP(i)==1
            for e=(i-1):-1:1
                if LINE_CODE_BP(e)==VCC
                   LINE_CODE_BP(i)=VDD;
                   Break
            elseif LINE_CODE_BP(e)==VDD
                    LINE_CODE_BP(i)=VCC;
                   Break
            end
        end
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%sampling data to be P-NRZ%%%%%%%%%%%%%%%%%%%%%%%%%%%%
c=0;
for i=1:length(LINE_CODE_BP)
    if LINE_CODE_BP(i)==VCC
        for q=1:50
            BP_RZ(q+c)=VCC;
            BP_RZ(q+50+c)=GND;
        end
    elseif LINE_CODE_BP(i)==VDD
        for q=1:50
            BP_RZ(q+c)=VDD;
            BP_RZ(q+50+c)=GND;
        end
    else
        for q=1:100
            BP_RZ(q+c)=GND;
        end
    end
    c=c+100;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%TX-SIGNAL"BP_RZ"%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(bp)                                                  %BP_RZ figure
subplot(3,1,1)                                              %to plot 3 figure
plot(t,BP_RZ,'r');                                          %plot TX signal "stream of 10k bits"
axis([0 length(LINE_CODE_BP) -(VCC+2) (VCC+2)]);            %To improve the overall shape
xlabel('Time axis');                                        %x-axis
ylabel('Amplitude');                                        %y-axis
```

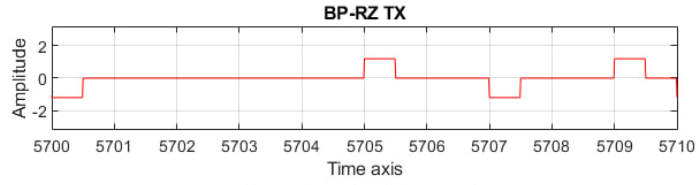
```
title('BP-RZ TX');                                          %title
grid on;                                                    %grid on
xlim([5700 5710])                                           %zoom in x-axis
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%ADD NOISE%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
NOISE_BP_RZ = BP_RZ +sigma(bp-30)*randn(size(BP_RZ)); %Add noise
figure(bp)                                                   %BP_RZ figure
subplot(3,1,2)                                               %to plot 3 figure
plot(t,NOISE_BP_RZ,'black',t,threshold1_BP,'g',t,threshold2_BP,'g')
axis([0 length(LINE_CODE_BP) -(VCC+2) (VCC+2)]);     %To improve the overall shape
xlabel('Time axis');                                        %x-axis
ylabel('Amplitude');                                        %y-axis
title('Noise');
grid on;                                                   %grid on
xlim([5700 5710])                                          %zoom in x-axis
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% RX %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
c=0;
for i=25:100:length(BP_RZ)
    if NOISE_BP_RZ(i)>threshold1_BP
        for q=1:50
            RX_BP_RZ(q+c)=VCC;
            RX_BP_RZ(q+50+c)=GND;
        end
    elseif NOISE_BP_RZ(i)<threshold2_BP
        for q=1:50
            RX_BP_RZ(q+c)=VDD;
            RX_BP_RZ(q+50+c)=GND;
        end
    else
        for q=1:100
            RX_BP_RZ(q+c)=GND;
        end
    end
    c=c+100;
end
figure(bp)
subplot(3,1,3)
plot(t,RX_BP_RZ,'m')
axis([0 length(LINE_CODE_BP) -(VCC+2) (VCC+2)]);
xlabel('Time axis');
ylabel('Amplitude');
title('RX-BP-RZ');
grid on;
xlim([5700 5710])
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
no_bit_error_BP_RZ=0;
for i=1:length(RX_BP_RZ)
    if(BP_RZ(i)~=RX_BP_RZ(i))
        no_bit_error_BP_RZ=no_bit_error_BP_RZ+1;
     end
end
no_bit_error_BP_RZ=no_bit_error_BP_RZ/fs*2;
a_n_bit_error_BP_RZ(bp-30)=no_bit_error_BP_RZ;
BER_BP_RZ(bp-30)=no_bit_error_BP_RZ/num_of_bits;
```

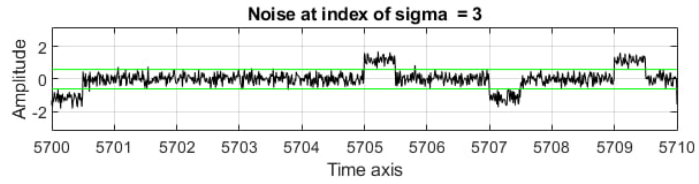
- **Note:** **this figure is at sigma index=1 (no noise)**

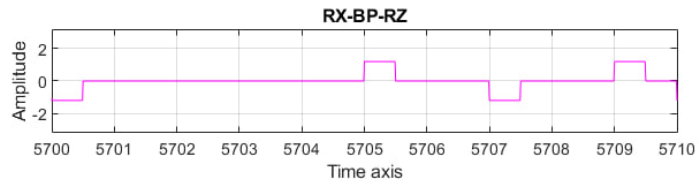


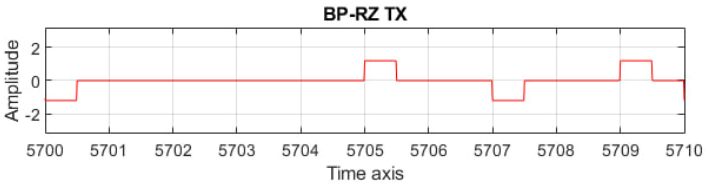
- **All upcoming figures contain noise (where, 9 values of sigma are swept to each figure)**

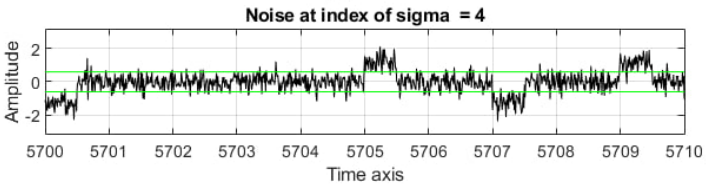


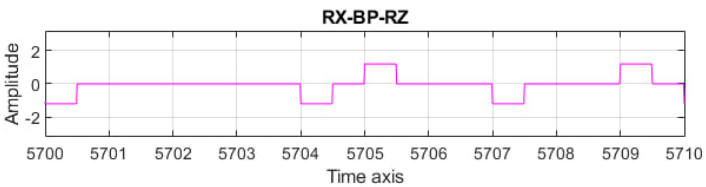

BP-RZ TX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis

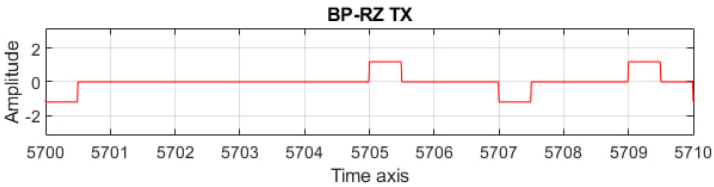

Noise at index of sigma = 3
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis

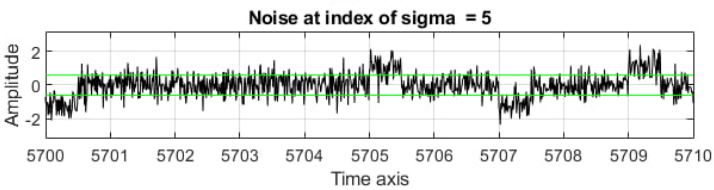

RX-BP-RZ
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis

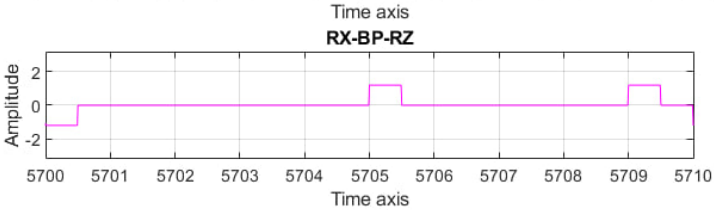

BP-RZ TX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis

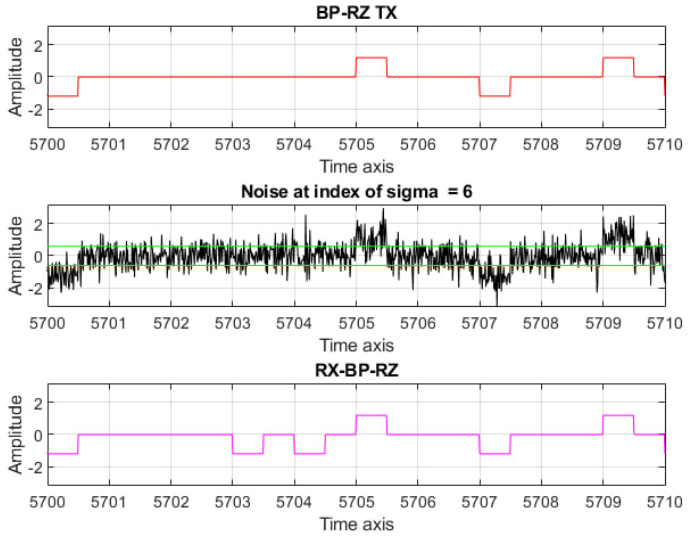

Noise at index of sigma = 4
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis

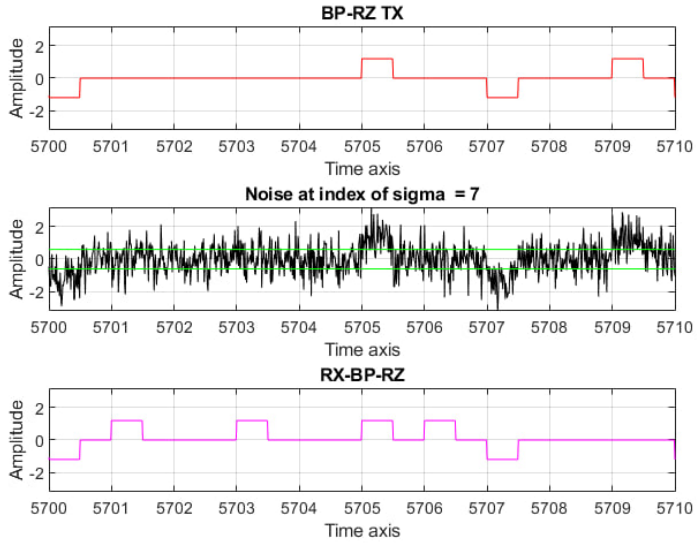

RX-BP-RZ
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis

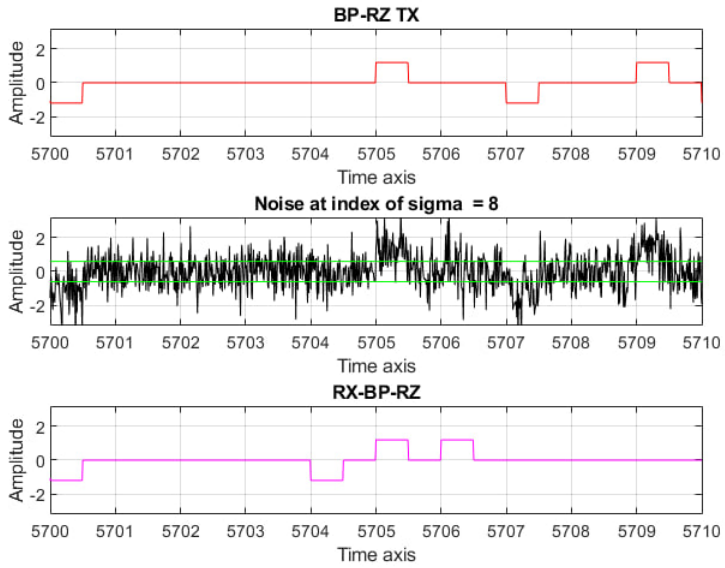

BP-RZ TX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis

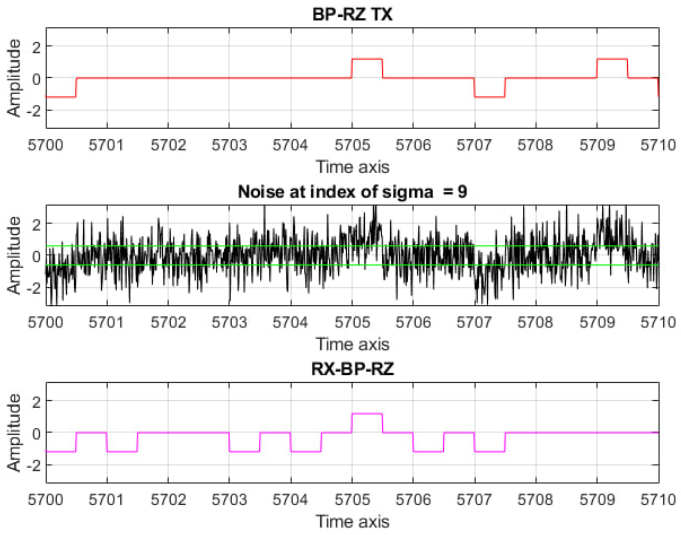

Noise at index of sigma = 5
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis

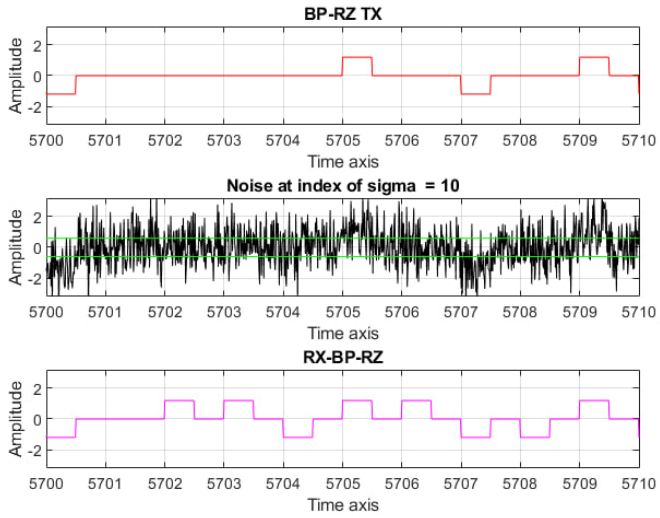

RX-BP-RZ
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis


BP-RZ TX
Amplitude
Time axis
Noise at index of sigma = 6
RX-BP-RZ


BP-RZ TX
Amplitude
Time axis
Noise at index of sigma = 7
RX-BP-RZ


BP-RZ TX
Amplitude
Time axis
Noise at index of sigma = 8
RX-BP-RZ





- **Number of bits containing errors in every case of sigma**

a_n_bit_error_BP_RZ

1x10 double

	1	2	3	4	5	6	7	8	9	10
1	0	0	196	1002	1968	2717	3331	3963	4317	4668

- **BER at every case of sigma**

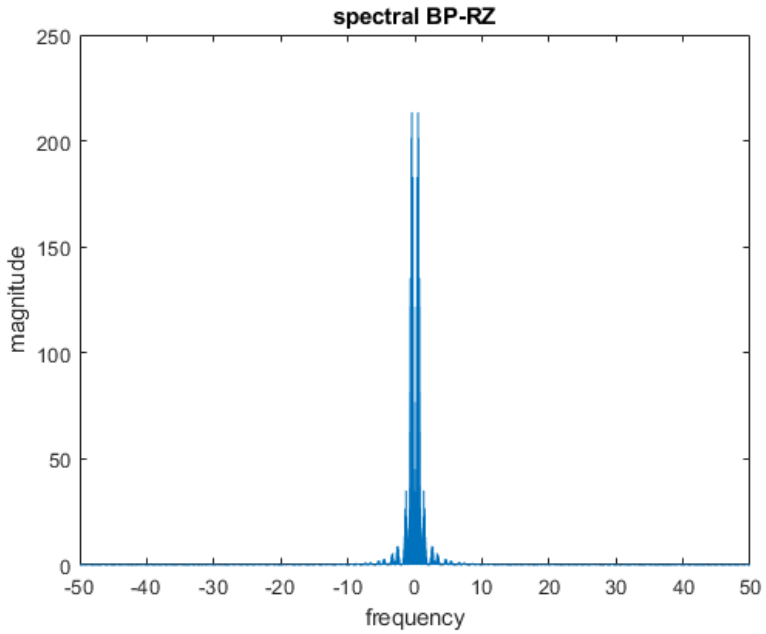
BER_BP_RZ

1x10 double

	1	2	3	4	5	6	7	8	9	10
1	0	0	0.0196	0.1002	0.1968	0.2717	0.3331	0.3963	0.4317	0.4668
2										

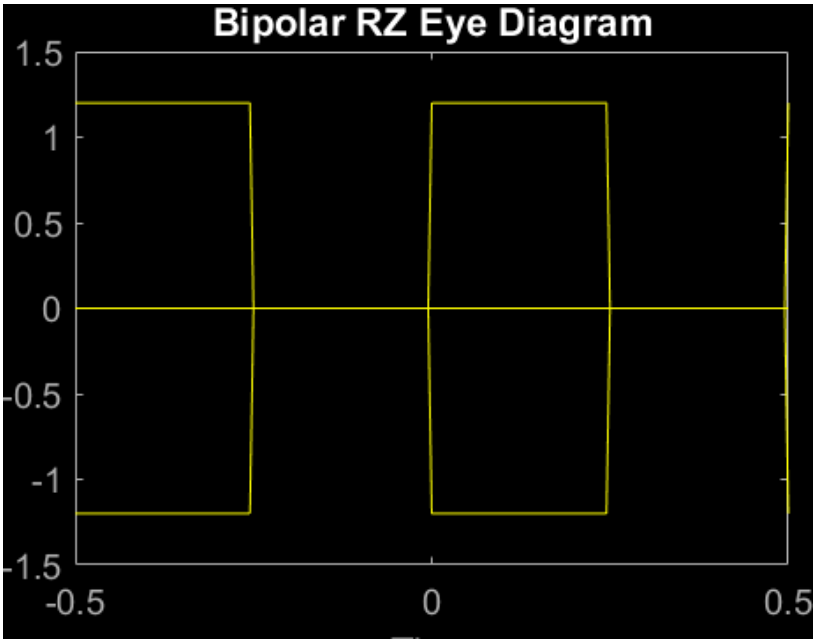
- **Spectral domain**

```
figure(55)
plot(f,(abs(fftshift(fft(BP_RZ))).^2)/N)
xlabel('frequency');
ylabel('magnitude');
title('spectral BP-RZ');
```



- **Eye diagram**

```
% Create eye diagram for Bipolar RZ
figure(60);
eyediagram(BP_RZ,200,1,0) ;
xlabel('Time');
ylabel('Amplitude');
title('Bipolar RZ Eye Diagram');
set(gca, 'FontSize', 15);
```



- **Bonus**

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% BONUS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i=1:length(RX_BP_RZ)
    if (RX_BP_RZ(i)== 1.2)
        break
    end
end

eee=0;
for BONUS=i:length(RX_BP_RZ)
    if RX_BP_RZ(BONUS)==GND
        continue
    elseif RX_BP_RZ(BONUS)==VCC
        for back=(BONUS-100):-100:1
            if RX_BP_RZ(back)==VCC
                eee=eee+1;
                break
            elseif RX_BP_RZ(back)==VDD
                break
            elseif RX_BP_RZ(back)==GND
                continue
            end
        end
    elseif RX_BP_RZ(BONUS)==VDD
        for back=(BONUS-100):-100:1
            if RX_BP_RZ(back)==VDD
                eee=eee+1;
                break
            elseif RX_BP_RZ(back)==VCC
                break
            elseif RX_BP_RZ(back)==GND
                continue
            end
        end
    end
end
error_detection(bp-30)= eee/50;
end
```

the algorithm

error detection circuit is designed by if 2 consecutive values of VCC or 2 consecutive values of VDD are found, 1 is added to the counter "eee", but the noise can change the original signal value from VCC or VDD to GND in this case we will not be able to identify the error.

Therefore, we find that the error detection circuit values are less than the actual values "in the case of comparing the receiving circuit with the transmitting circuit"

- **Error detection bits in every case of sigma**

error_detection

1x10 double

	1	2	3	4	5	6	7	8	9	10
1	0	0	186	836	1466	1902	2169	2427	2772	2918
2										

- **Actual error values**

a_n_bit_error_BP_RZ

1x10 double

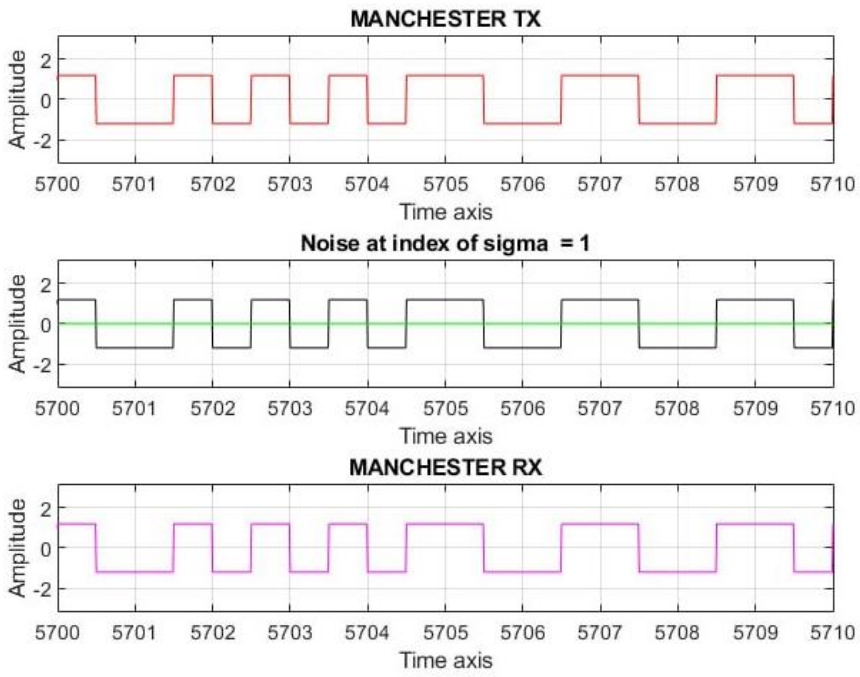
	1	2	3	4	5	6	7	8	9	10
1	0	0	196	1002	1968	2717	3331	3963	4317	4668

convert line code to MANCHESTER:

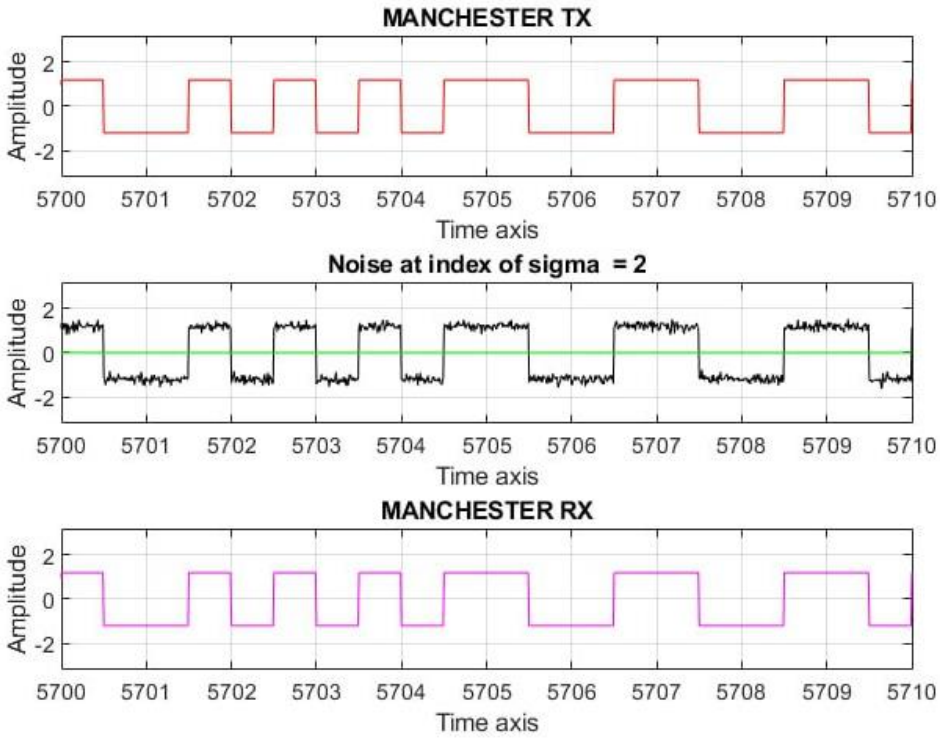
```
for m=41:50
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Modify the amp.%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for i = 1:length(LINE_CODE)      %loop to take all values in LINE_CODE array
if LINE_CODE(i)==1               %if current value ==1
  data_MANCHESTER(i)=VCC;        %convert it to vcc and store it in data array
else                             %else "cuurent value ==0"
  data_MANCHESTER(i)=VDD;        %convert it to VDD and store it in data array
end                              %end if condition
end                              %end for loop
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% sampling data to be P-NRZ %%%%%%%%%%%%%%%%%%%%%%%%%%
c=0;
for i=1:length(data_MANCHESTER)
if data_MANCHESTER(i)==VCC
 for q=1:50
 MANCHESTER(q+c)=VCC;
 MANCHESTER(q+50+c)=VDD;
 end
else
 for q=1:50
 MANCHESTER(q+c)=VDD;
 MANCHESTER(q+50+c)=VCC;
 end
end
c=c+100;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%TX-SIGNAL"P-NRZ"%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
figure(m)                                                    %MANCHESTER figure
subplot(3,1,1)                                               %to plot 3 figure
plot(t,MANCHESTER,'r');                                      %plot TX signal "stream of 10k bits"
axis([0 length(LINE_CODE) -(VCC+2) (VCC+2)]);                %To improve the overall shape
xlabel('Time axis');                                         %x-axis
ylabel('Amplitude');                                         %y-axis
title('MANCHESTER TX');                                      %title
grid on;                                                     %grid on
xlim([5700 5710])                                            %zoom in x-axis
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
NOISE_MANCHESTER = MANCHESTER +sigma(m-40)*randn(size(MANCHESTER));%Add noise
figure(m)                                                    %MANCHESTER figure
subplot(3,1,2)                                               %to plot 3 figure
plot(t,NOISE_MANCHESTER,'black',t,threshold_Manchester,'g')%plot NOISE signal
axis([0 length(LINE_CODE) -(VCC+2) (VCC+2)]);                %To improve the overall shape
xlabel('Time axis');                                         %x-axis
ylabel('Amplitude');                                         %y-axis
title('Noise');
grid on;                                                     %grid on
xlim([5700 5710])                                            %zoom in x-axis
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
c=0;                                                         %modify the length of RX array
for i=25:100:length(MANCHESTER)                              %take the mid bits value
if NOISE_MANCHESTER(i)>threshold_Manchester                   %"bit>threshold"
 for q=1:50
 RX_MANCHESTER(q+c)=VCC;
 RX_MANCHESTER(q+50+c)=VDD;
```

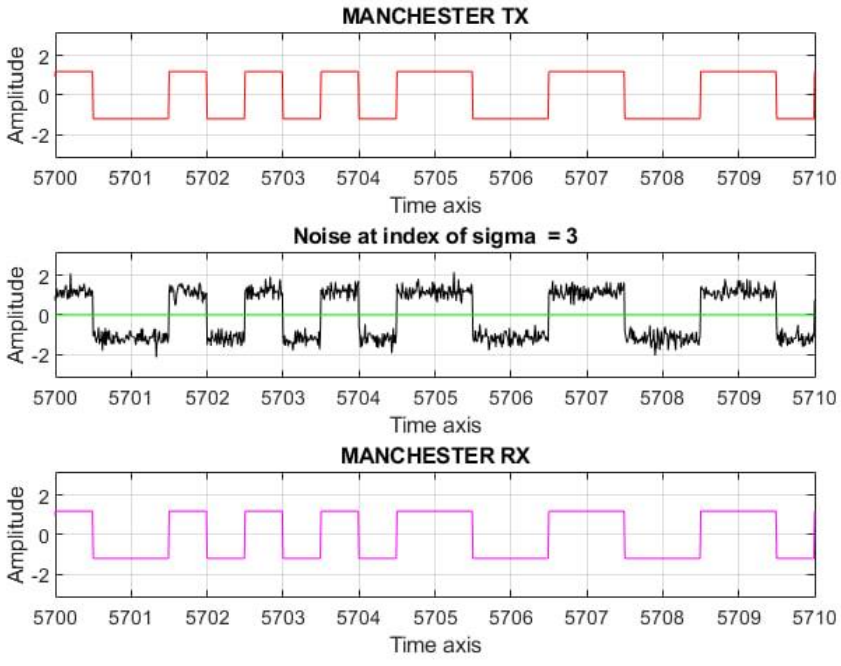
```
    end
else
    for q=1:50
    RX_MANCHESTER(q+c)=VDD;
    RX_MANCHESTER(q+50+c)=VCC;
    end
end
    c=c+100;
end
figure(m)
subplot(3,1,3)
plot(t,RX_MANCHESTER,'m')
axis([0 length(LINE_CODE) -(VCC+2) (VCC+2)]);
xlabel('Time axis');
ylabel('Amplitude');
title('MANCHESTER RX');
grid on;
xlim([5700 5710])
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% BER %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
no_bit_error_MANCHESTER=0;
for i=1:length(RX_MANCHESTER)                          %to take all value
if(MANCHESTER(i)~=RX_MANCHESTER(i))                    %to compare between two array
    no_bit_error_MANCHESTER=no_bit_error_MANCHESTER+1;  %counter to count number bit error
end                                                    %end if condition
end                                                    %end for loop
no_bit_error_MANCHESTER=no_bit_error_MANCHESTER/fs;    %normalize number bit error
a_n_bit_error_MANCHESTER(m-40)=no_bit_error_MANCHESTER;%create array to store number bit
error each value of sigma
BER_MANCHESTER(m-40)=no_bit_error_MANCHESTER/num_of_bits;  %calculate BER in each case
end                                                    %end loop of sigma
```

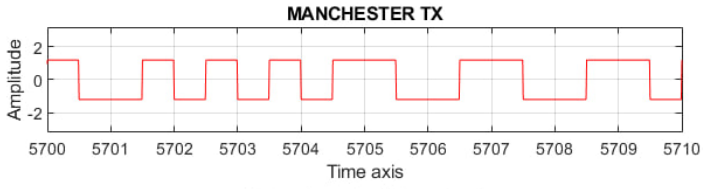
- **Note: this figure is at sigma index=1 (no noise)**

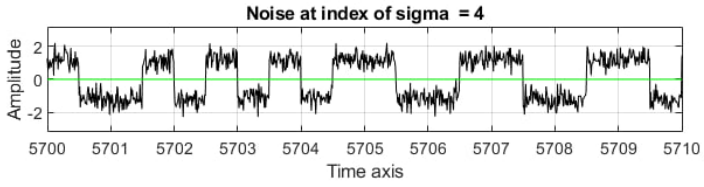


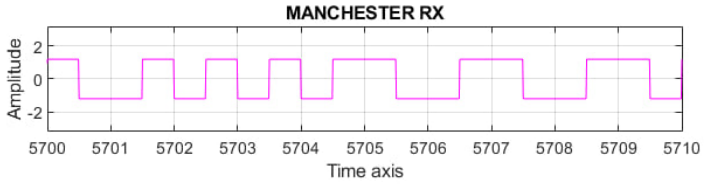
- **All upcoming figures contain noise (where, 9 values of sigma are swept to each figure)**

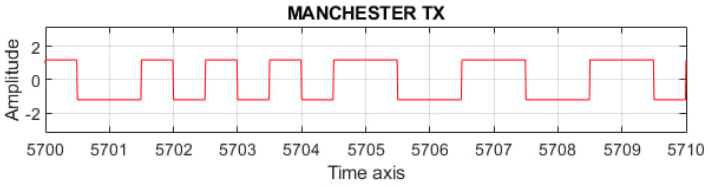


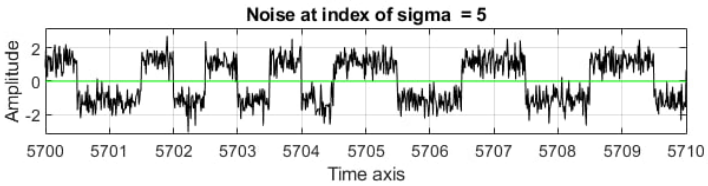


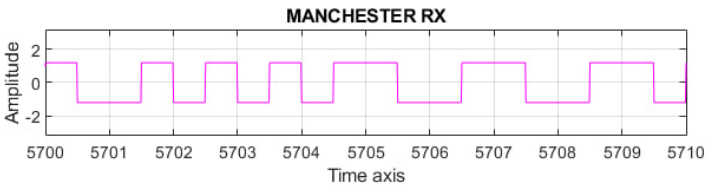

MANCHESTER TX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis

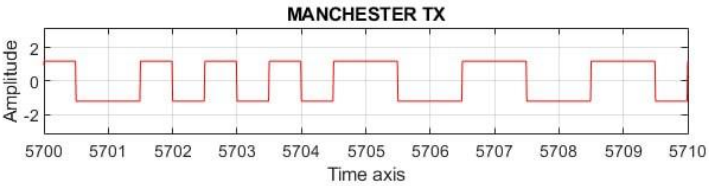

Noise at index of sigma = 4
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis

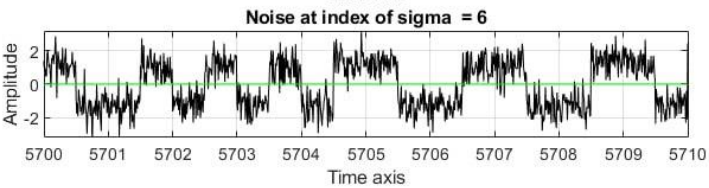

MANCHESTER RX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis

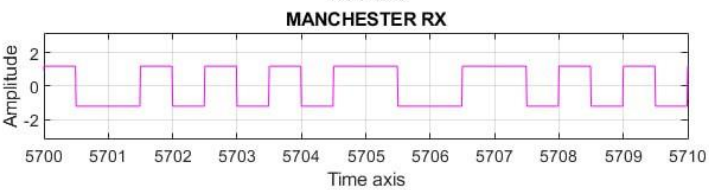

MANCHESTER TX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis

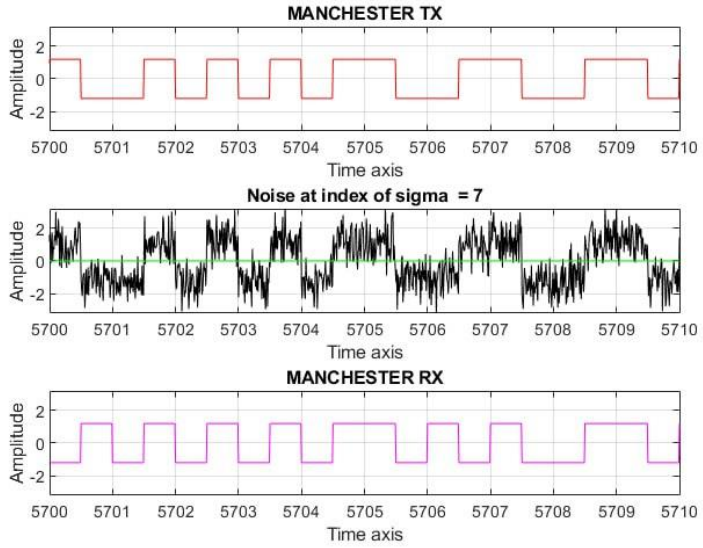

Noise at index of sigma = 5
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis

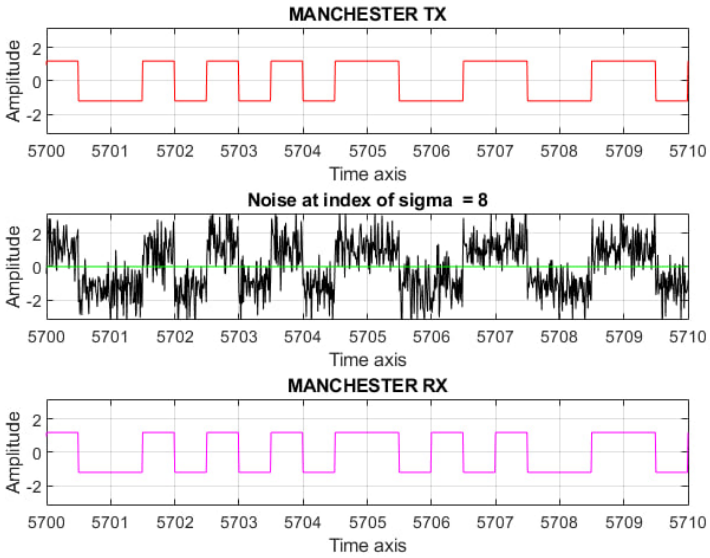

MANCHESTER RX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis

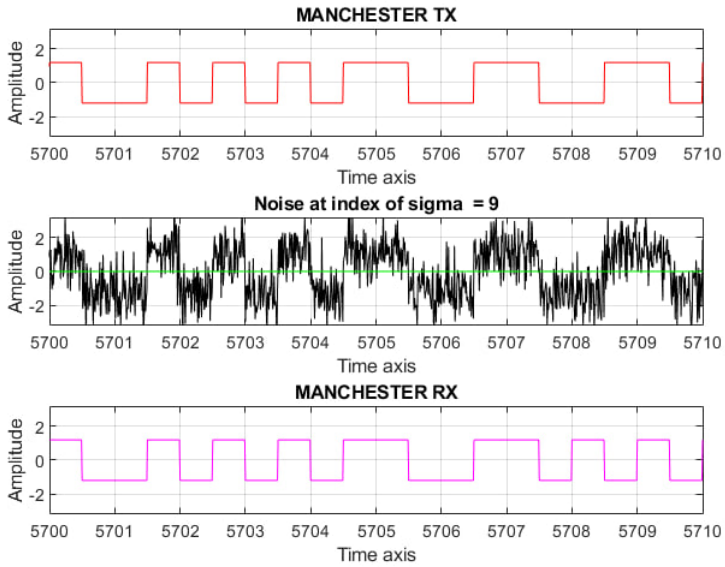

MANCHESTER TX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis

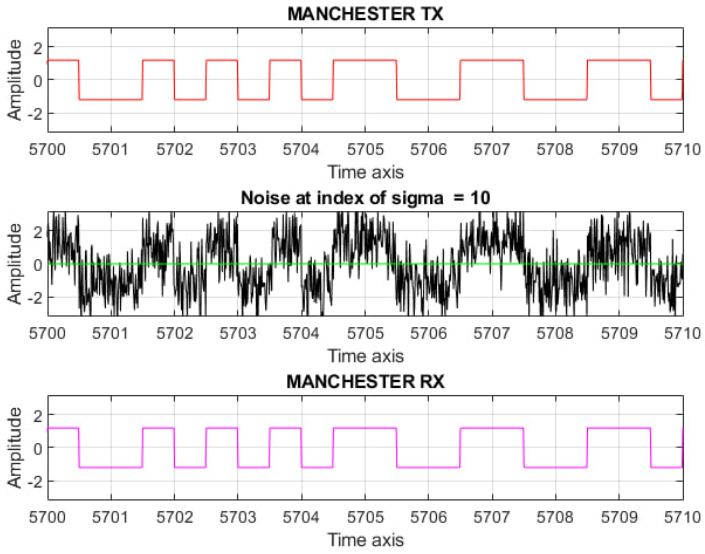

Noise at index of sigma = 6
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis


MANCHESTER RX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis


MANCHESTER TX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis
Noise at index of sigma = 7
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis
MANCHESTER RX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis


MANCHESTER TX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis
Noise at index of sigma = 8
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis
MANCHESTER RX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis


MANCHESTER TX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis
Noise at index of sigma = 9
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis
MANCHESTER RX
Amplitude
2
0
-2
5700 5701 5702 5703 5704 5705 5706 5707 5708 5709 5710
Time axis



- **Error detection bits in every case of sigma**

a_n_bit_error_MANCHESTER

1x10 double

	1	2	3	4	5	6	7	8	9	10
1	0	0	0	19	132	355	682	976	1259	1602

- **BER at every case of sigma**

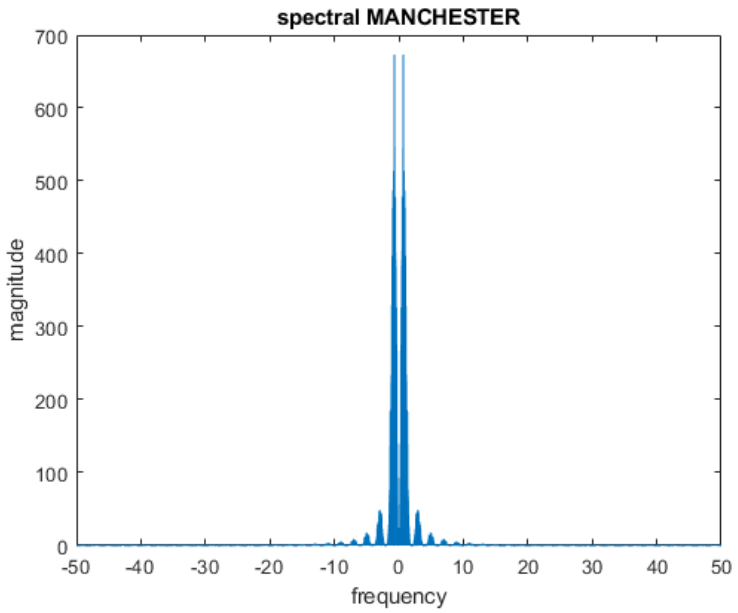
BER_MANCHESTER

1x10 double

	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0.0019	0.0132	0.0355	0.0682	0.0976	0.1259	0.1602
2										

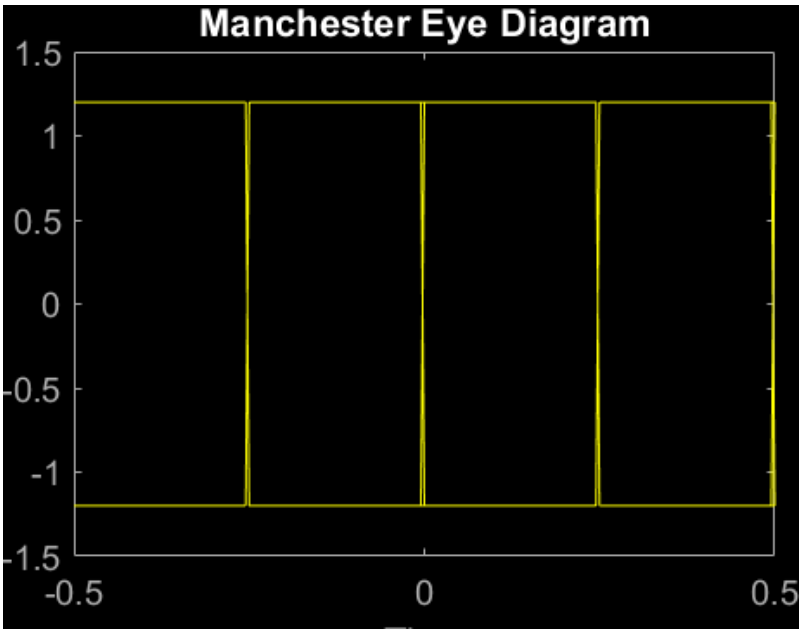
- **Spectral domain**

```
figure(56)
plot(f,(abs(fftshift(fft(MANCHESTER))).^2)/N)
xlabel('frequency');
ylabel('magnitude');
title('spectral MANCHESTER');
```



- **Eye diagram**

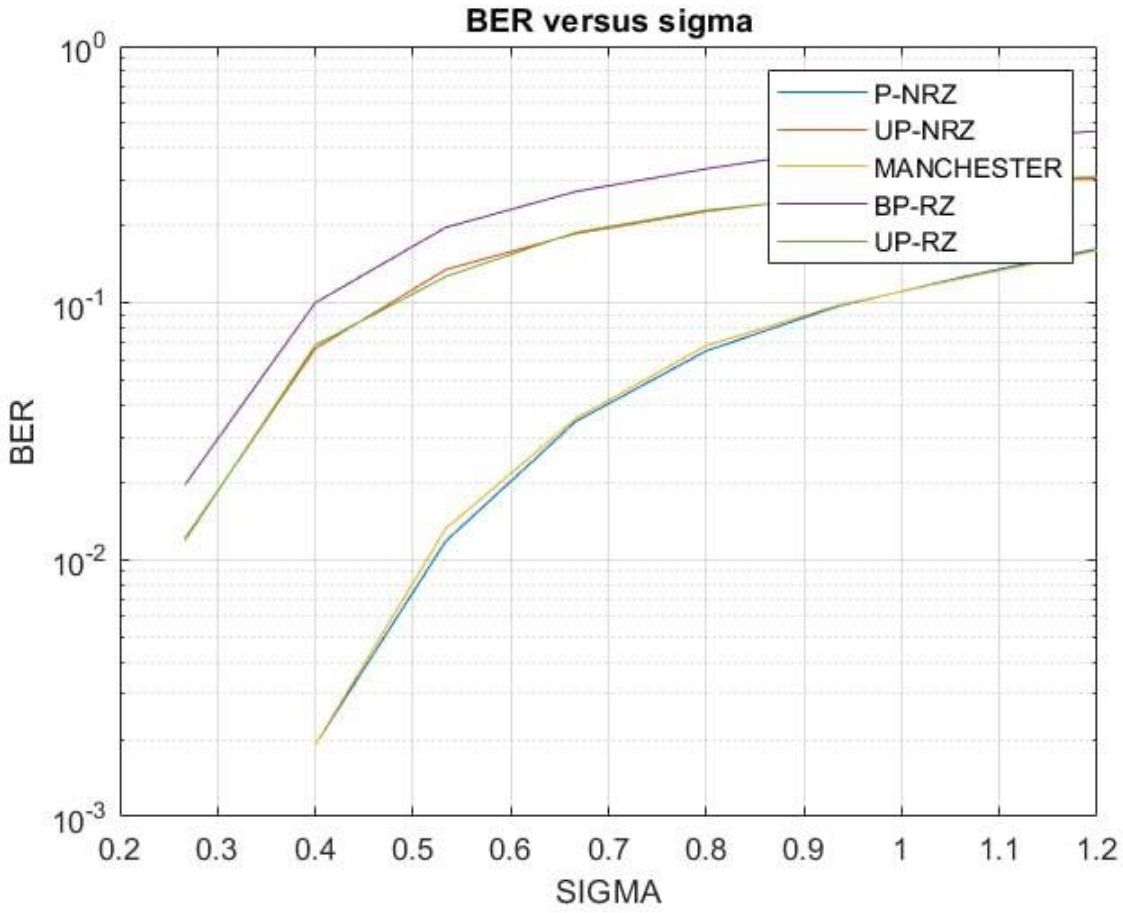
```
% Create eye diagram for Manchester
figure(61);
eyediagram(MANCHESTER,200,1,0) ;
xlabel('Time');
ylabel('Amplitude');
title('Manchester Eye Diagram');
set(gca, 'FontSize', 15);
```



- **BER curves**

Y-log scale

```
figure(51)
semilogy(sigma,BER_P_NRZ,sigma,BER_UP_NRZ,sigma,BER_MANCHESTER,sigma,BER_BP_RZ,sigma,BER_UP_RZ)
legend('P-NRZ','UP-NRZ','MANCHESTER','BP-RZ','UP-RZ')
xlabel('SIGMA');                                %x-axis
ylabel('BER');                                  %y-axis
title('BER versus sigma');                      %title
grid on;                                        %grid on
```

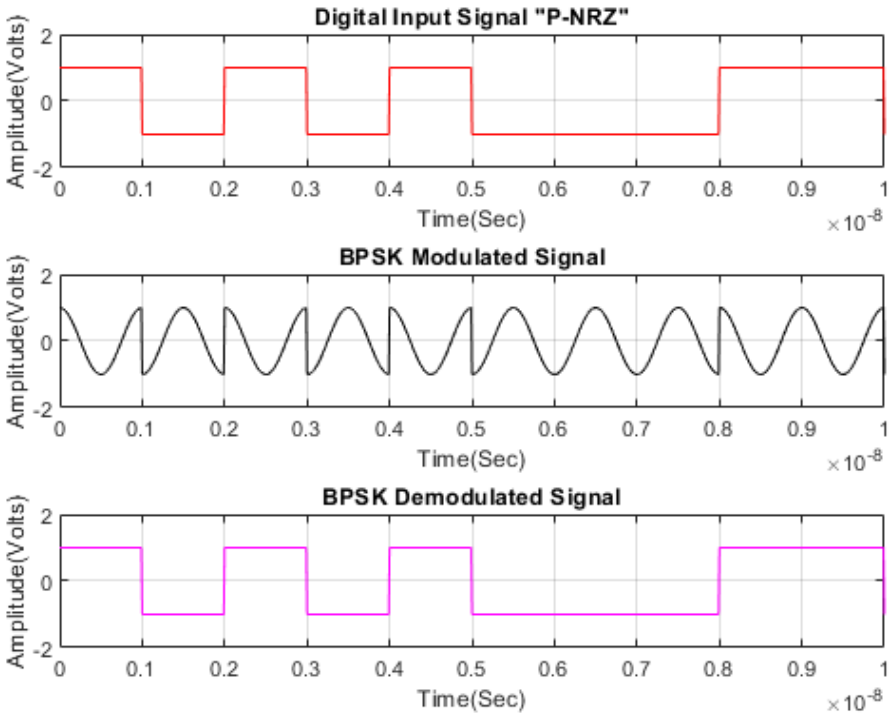


➢ Part 2

```
clear
fc=1e9;
ac=1;
Tb=1/fc;
no_of_bit=100;
nsc=100;
tsc=1e-11;
T=no_of_bit*Tb; %1e-7
tc=0:tsc:Tb-tsc;
c=cos(2*pi*fc*tc);
line_code=randi([0 1],1,no_of_bit);
data=line_code*2-1;
tm=linspace(0,T,10000);
N=ceil(T/tsc);                  %Normalization
df=1/T;                        %frequency step
fs=1/tsc;                       %sampling frequency
if (rem(N,2)==0)               %if normalization even number
   f=-(0.5*fs):df:(0.5*fs-df);
else                           %if normalization even number
   f=-(0.5*fs-0.5*df):df:(0.5*fs-0.5*df);
end

counter=0;
for i=1:100
    if data(i)==1
        for q=1:100
            P_NRZ(q+counter)=data(i);
        end
    else
        for q=1:100
            P_NRZ(q+counter)=data(i);
        end
    end
    counter=counter+100;
end
clear counter , q;
figure
subplot(3,1,1);
plot(tm,P_NRZ,'r');
grid on;
axis([0 1e-8 -2 2]);
xlabel('Time(Sec)');
ylabel('Amplitude(Volts)');
title('Digital Input Signal "P-NRZ"');
modulated = [];
for i=1:100
    if data(i)==1
        y = ac*cos(2*pi*fc*tc+0);   % Modulation signal with carrier signal 1
    else
        y = ac*cos(2*pi*fc*tc+pi);  % Modulation signal with carrier signal 0
    end
    modulated=[modulated y];
```

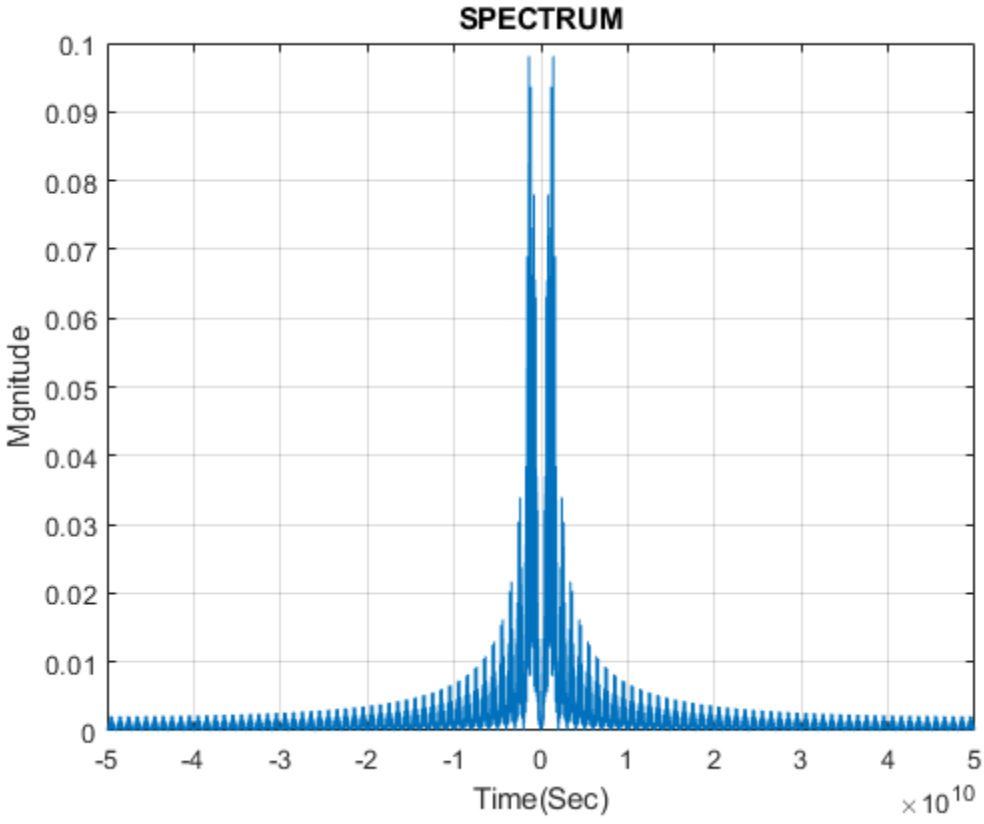
```
end
subplot(3,1,2);
plot(tm,modulated,'black');
xlabel('Time(Sec)');
ylabel('Amplitude(Volts)');
title('BPSK Modulated Signal');
grid on
axis([0 1e-8 -2 2]);
% ********************* Channel model h and w ****************************
h = 1;
w = 0;
% ********************* Received signal y ********************************
noise_signal = h.*modulated + w;
% ************************** BPSK Demodulation ***************************
demodulated = [];
for n = 100:100:length(modulated)
  c = cos(2*pi*fc*tc);
  mm = c.*noise_signal((n-(100-1)):n);
  z = trapz(tc,mm);
  if(z > 0)
    a = 1;
  else
    a = -1;
  end
  demodulated = [demodulated a];
end
 %%%%%%%%%%%%%%%%%% Represent demodulated information as digital signal %%%%%%%%%%%%%%%%%%
 counter=0;
 for i=1:100
     if demodulated(i)==1
         for q=1:100
             RX_P_NRZ(q+counter)=demodulated(i);
         end
     else
         for q=1:100
              RX_P_NRZ(q+counter)=demodulated(i);
        end
     end
     counter=counter+100;
 end
 subplot(3,1,3)
 plot(tm,RX_P_NRZ,'m');
 grid on
 xlabel('Time(Sec)');
 ylabel('Amplitude(Volts)');
 title('BPSK Demodulated Signal');
 axis([0 1e-8 -2 2]);
 %%%%%%%%%%%%%%%%%%%%%%%%%% BER %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 no_bit_error_P_NRZ=0;
 for i=1:length(RX_P_NRZ)
     if(P_NRZ(i)~=RX_P_NRZ(i))
         no_bit_error_P_NRZ=no_bit_error_P_NRZ+1;
     end
 end
 no_bit_error_P_NRZ=no_bit_error_P_NRZ/nsc;
```

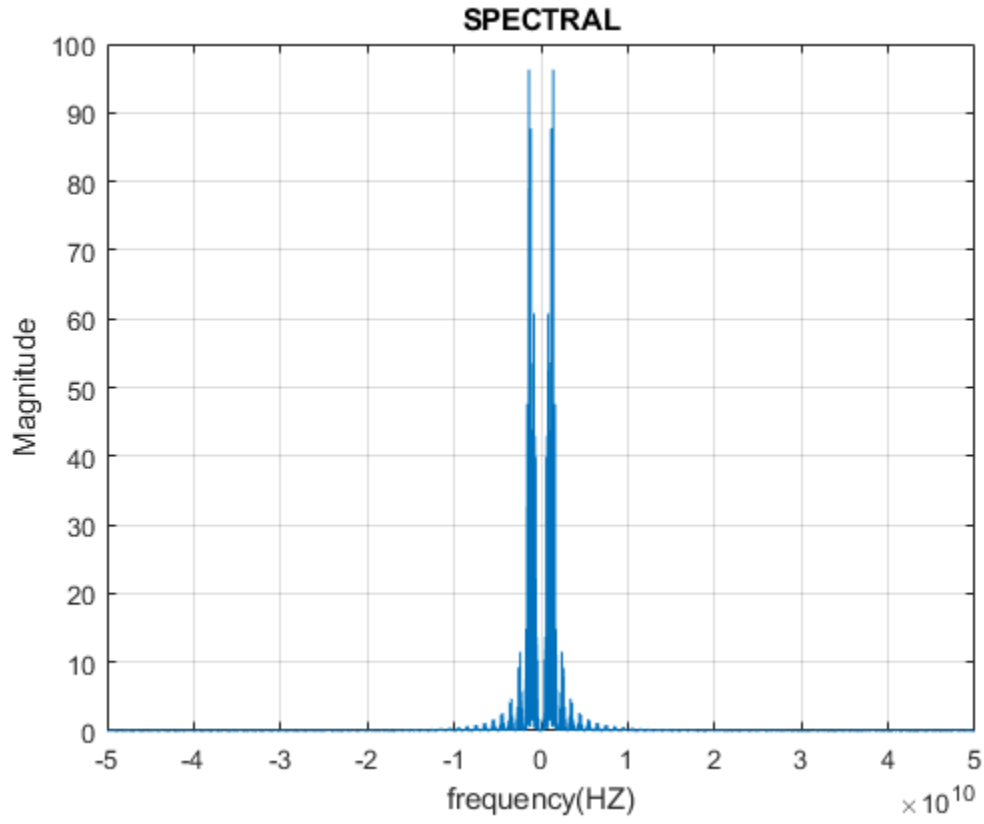
```
BER_P_NRZ=no_bit_error_P_NRZ/no_of_bit;
% ********************* SPECTRUM signal x ****************************
figure
plot(f,abs(fftshift(fft(modulated))/N))
xlabel('Time(Sec)');
ylabel('Mgnitude');
title('SPECTRUM');
grid on
% ********************* SPECTRAL signal x ****************************
figure
plot(f,(abs(fftshift((fft(modulated))))).^2/N))
xlabel('frequency(HZ)');
ylabel('Magnitude');
title('SPECTRAL');
grid on
```



The last figure are zoomed from 100 bits to be clearly shown

BER_P_NRZ 0


SPECTRUM
Mgnitude
0.1
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
-5
-4
-3
-2
-1
0
1
2
3
4
5
Time(Sec)
×10^10


SPECTRAL
Magnitude
100
90
80
70
60
50
40
30
20
10
0
-5
-4
-3
-2
-1
0
1
2
3
4
5
frequency(HZ)
×10^10