Xi'an Jiaotong-Liverpool University
西交利物浦大學
DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING

# Lab Assignment

Baseband Digital Signal Transceiver

**Sahand Sabour**

**EEE301**

DIGITAL AND ANALOGUE COMMUNICATIONS II

**Student ID**

1614650

# Introduction

Digital signal transmission is a fundamental task of modern communication systems. Baseband transmission is the main method of transmission that is used for this type of signal; where baseband is a term for the band of frequencies that represent the original signal at the destination. Hence, the use of devices that are utilized to transmit digital signals from a source, pass them through a channel that is generally exposed to noise, and receive and reconstruct the signals at the destination is necessary. This device is referred to as baseband transceiver (transmitter and receiver) and the schematic block diagram for such device is provided in the figure below (Figure 1). Baseband transceivers are widely used in short-range wireless systems such as local area networks (LAN), printers, etc.

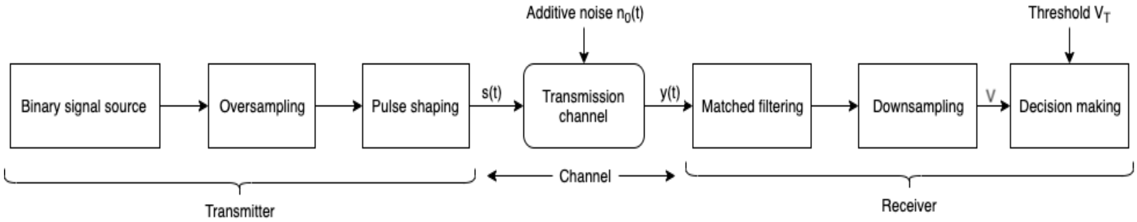


Figure 1: Basedband transceiver block diagram

In this assignment, a baseband transceiver for a random binary sequence as the source signal, whose channel is susceptible to additive white Gaussian noise (AWGN) and utilizes an optimum receiver for matched filtering is constructed. The rest of this report includes the parameter setting for such design as well as illustrate and thoroughly analyze the obtained results. The Matlab code is provided likewise.

### Math Modeling

$y(t) = s(t) + n_0(t)$ and the output decision based on threshold $V_T$ is as follows

$$\begin{cases} V > V_T & \text{decide that 1 is sent} \\ V < V_T & \text{decide that 0 is sent} \end{cases} \tag{1}$$

For a matched filter $h_{opt}(t) = s_1(T-t) - s_0(T-t)$

$$V_{Topt} = \frac{E_1 - E_0}{2} \tag{2}$$

is the corresponding optimal threshold, where $E_0$ and $E_1$ are the energy for symbols $s_0(t)$ and $s_1(t)$ respectively and are calculated as follows.

$$E_0 = \int_0^T s_0^2(t)dt \text{ and } E_1 = \int_0^T s_1^2(t)dt \tag{3}$$

The resulting minimum Bit Error Rate (BER) would be

$$P_{e,min} = Q\left[\sqrt{\frac{E_g}{2N_0}}\right] \tag{4}$$

where $N_0$ is the noise power and

$$E_g = \int_0^T (s_1(t) - s_0(t))^2 dt = E_0 + E_1 - \rho\sqrt{E_0 E_1} \tag{5}$$

$\rho$ is the correlation coefficient between symbols $s_0(t)$ and $s_1(t)$ and it is equal to

$$\rho = \frac{1}{\sqrt{E_0 E_1}} \int_0^T s_0(t)s_1(t)dt \tag{6}$$

Generally, signals would be transmitted within two types: Unipolar and Bipolar. The symbols $s_0(t)$ and $s_1(t)$ for these signaling methods can be written as follows

$$\text{Unipolar Signal} \begin{cases} s_0(t) = 0 \\ s_1(t) = A \end{cases} \quad \text{Bipolar Signal} \begin{cases} s_0(t) = -A \\ s_1(t) = A \end{cases} \tag{7}$$

Hence, the average bit energy $E_b = E_0 P(0) + E_1 P(1)$, where $P(0) = P(1) = 0.5$ due to the symbols being in binary, can be written as

$$\text{Unipolar Signal: } E_b = \frac{0 + A^2T}{2} = \frac{A^2T}{2} \tag{8}$$

$$\text{Bipolar Signal: } E_b = \frac{(-A)^2T + A^2T}{2} = \frac{2A^2T}{2} = A^2T \tag{9}$$

As calculated via (6), $\rho$ is 0 for unipolar signals and -1 for bipolar signals. Therefore, based on (2), it can be obtained as

$$\text{Unipolar Signal: } E_g = 0 + A^2T - 0 = A^2T \tag{10}$$

$$\text{Bipolar Signal: } E_g = A^2T + A^2T - (-2A^2T) = 4A^2T \tag{11}$$

based on (4) ,(10), and (11), the minimum BER can be rewritten as

$$\text{Unipolar Signal: } P_{e,min} = Q\left[\sqrt{\frac{E_b}{N_0}}\right] \tag{12}$$

$$\text{Bipolar Signal: } P_{e,min} = Q\left[\sqrt{\frac{2E_b}{N_0}}\right] \tag{13}$$

Therefore, it can be observed that the bipolar signaling requires only half the $E_B$ in comparison with the unipolar signaling. In theory, this would account for a difference of 3dB in BER.

For the pulse shaping, as required by the assignment, Rectangular and Root Raised Cosine (RRC) pulse shaping were to be used. In rectangular pulse shaping, the source signal would be oversampled by a given rate and therefore, is highly susceptible Inter-Symbol Interference (ISI). ISI is a form of distortion that occurs when a symbol interferes with the subsequent symbols. More specifically, this distortion would be produced as the result of spreading the pulse over its original interval, which is correspondent to the mentioned oversampling.

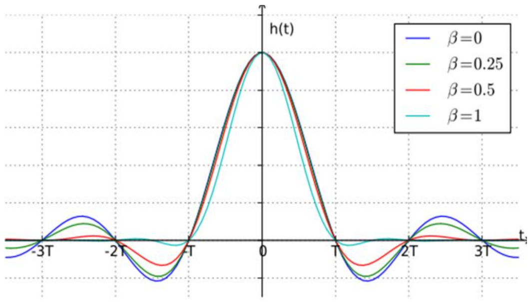


Figure 2: Impulse response of RRC

To minimize ISI, the total response, channel response, and the matched filter must follow the Nyquist ISI criterion. Hence, as RRC meets this criterion, it can be used for both pulse shaping and matched filtering. Figure 2 illustrates the impulse response of RRC. As shown by the figure, roll-off factor $\beta = 0$, has a fixed zero-crossing point, which is a good fit for removing ISI. However, in this figure, a single symbol is represented for an infinite time for different $\beta$,which is not practical. Hence, the signal needs to be truncated and due to this reason, RRC should be used for both oversampling and matched filtering to reduce ISI. Moreover, as displayed in Figure 3, with higher values of $\beta$, the effect of ISI on both sides of the signal would be reduced, which leads to less ISI. However, higher values of $\beta$ would require more bandwidth. Therefore, choosing the value of $\beta$ would be an optimization problem regarding both the bandwidth and ISI.

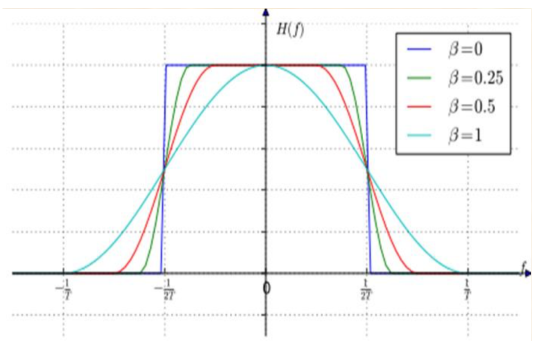


Figure 3: Frequency response of RRC

**Parameter Setting**

The parameters to be set for each method of pulse shaping (Rectangular and RRC) are provided in the tables below respectively.

| **Parameter** | **Variable** | **Value** | **Reason** |
|---|---|---|---|
| Bit rate | bitRate | 1 | Simplicity |
| Sample rate | sampleRate | 8 | $fs = 8Hz$ |
| Number of symbols | symbolNum | 4096 | need strong signal |
| Number of samples | sampleNum | 8 | need strong signal |
| Noise power | N0 | -10, -9, ..., 10 | Arbitrary |
| PSD window | PSDW | 4096 | required for FFT |

Table 1: Parameters for Rectangular pulse shaping

| **Parameter** | **Variable** | **Value** | **Reason** |
|---|---|---|---|
| Bit rate | bitRate | 1 | Simplicity |
| Sample rate | sampleRate | 8 | $fs = 8Hz$ |
| Number of symbols | symbolNum | 4096 | need strong signal |
| Number of samples | sampleNum | 8 | need strong signal |
| Noise power | N0 | -10, -9, ..., 10 | Arbitrary |
| PSD window | PSDW | 4096 | required for FFT |
| roll off factor $\beta$ | rollOff | 1 | better ISI removal |
| truncating symbols | span | 8 | better accuracy |

Table 2: Parameters for RRC pulse shaping

The corresponding Matlab code is provided in the appendices section of this report. In addition, the Matlab files were provided likewise. By using the Matlab IDE, the files can be opened and executed in order to obtain the results that are provided in this report.

# Results and Discussion

## Rectangular pulse shaping

Figures 4 and 5 respectively display the unipolar and bipolar random binary signals that are used as the input.

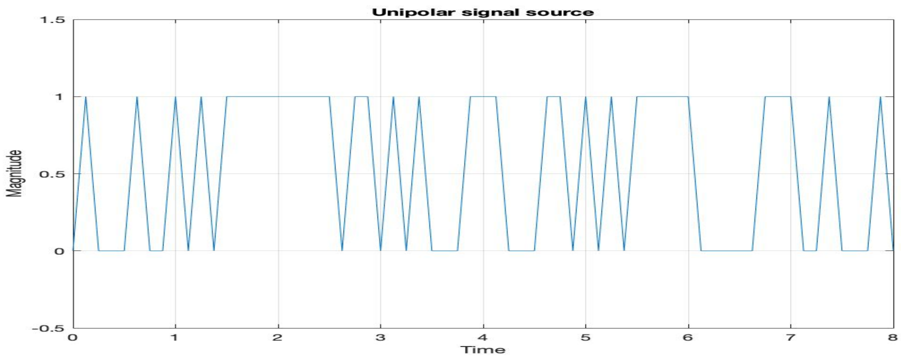


Figure 4: The unipolar source signal

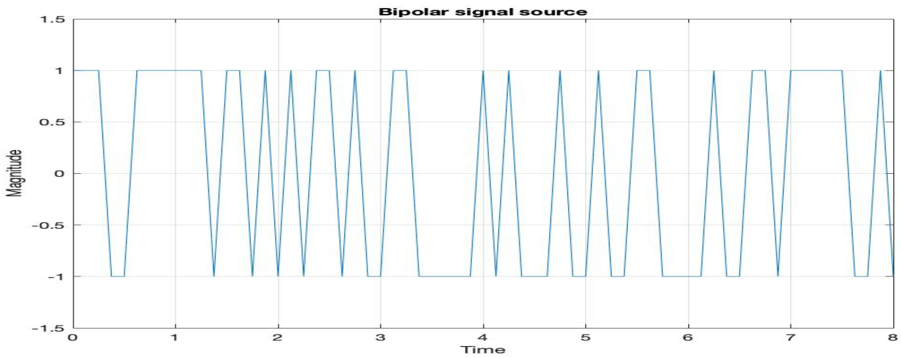


Figure 5: The bipolar source signal

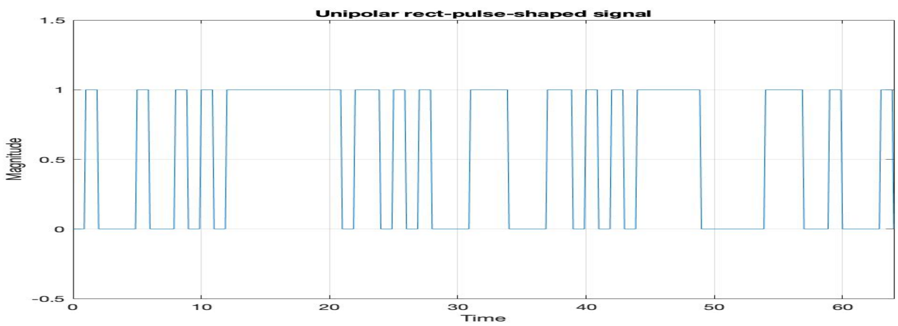


Figure 6: The unipolar rect-pulse-shaped signal

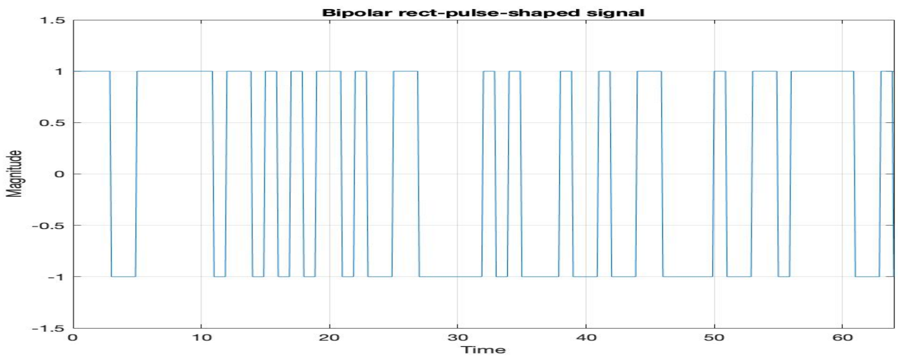


Figure 7: The bipolar rect-pulse-shaped signal

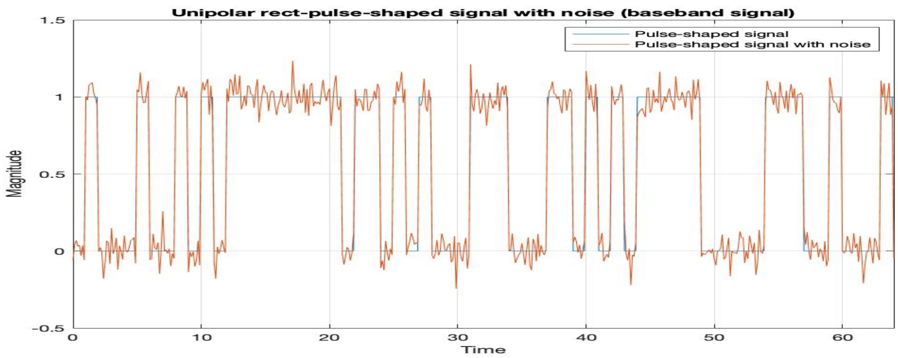


Figure 8: The unipolar rect-pulse-shaped signal with noise (baseband)

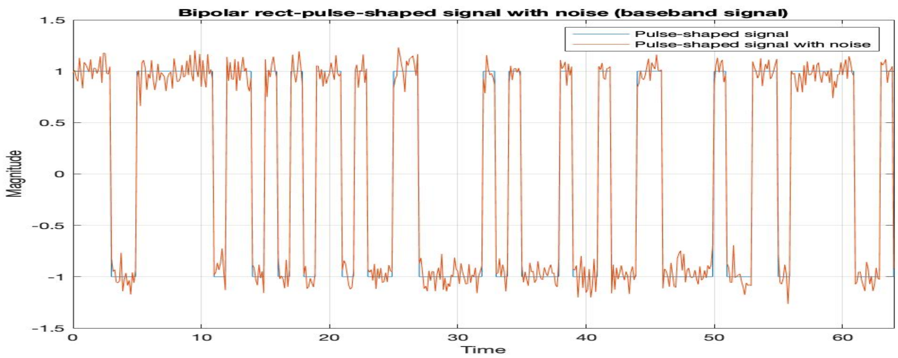


Figure 9: The bipolar rect-pulse-shaped signal with noise (baseband)

The above figures illustrate the oversampling and rectangular pulse shaping process for a given unipolar and bipolar random binary sequence. The resulting baseband signal was displayed likewise. The power spectrum density (PSD) of these signals are delineated in the below figures (Figures 10 and 11).

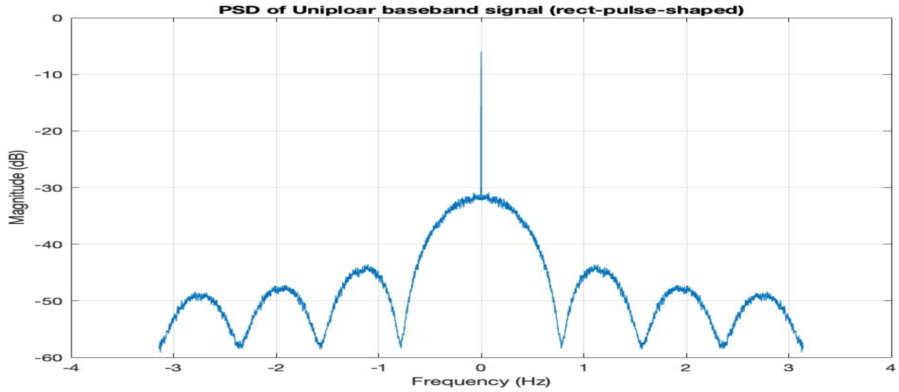


Figure 10: The power spectrum density of unipolar rect-shaped baseband signal

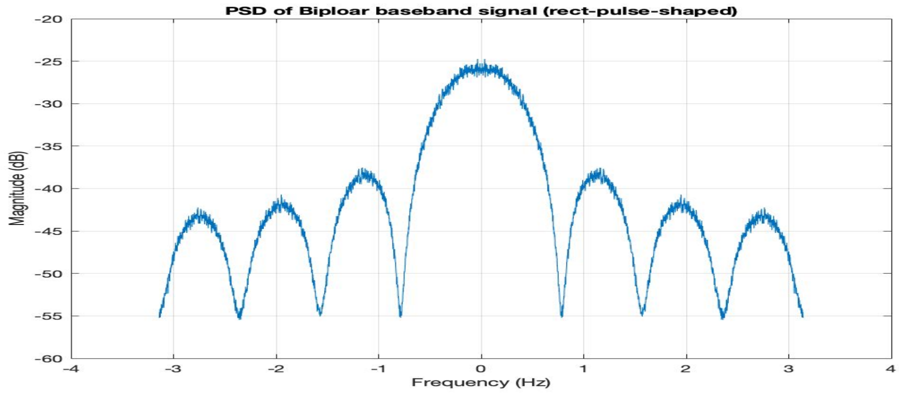


Figure 11: The power spectrum density of bipolar rect-shaped baseband signal

The above two figures demonstrate that given the same frequency (or bandwidth), the bipolar signaling has twice as much (6dB more) power as unipolar signaling.

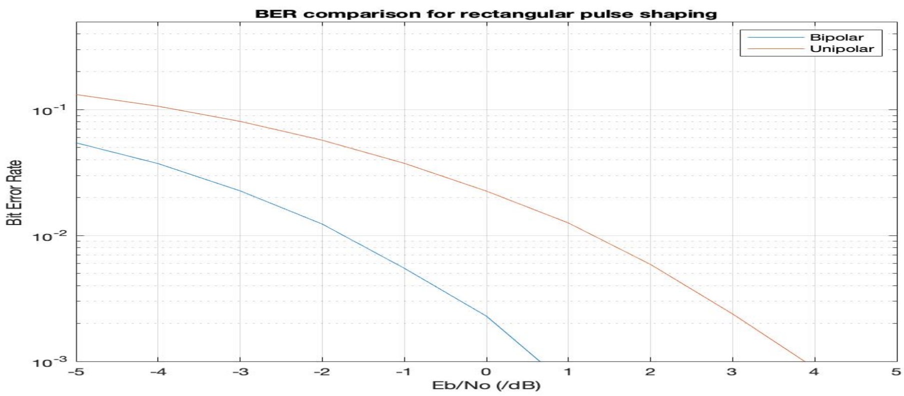


Figure 12: BER comparison for rectangular pulse-shaping

Moreover, based on Figure 12, it can be observed that for a given input, bipolar signaling can achieve a much lower BER in comparison to unipolar signaling. As mentioned in the math modeling section, the BER difference between the two would be 3dB, which is clearly displayed by the given figure. Conclusively, the below two figures (Figures 13 and 14) illustrate the comparison of the obtained analytical and the theoretical values of BER for both unipolar and bipolar signaling. As the analytical and the theoretical values are significantly similar, it is believed that the analytical performance of BER has been successfully verified.

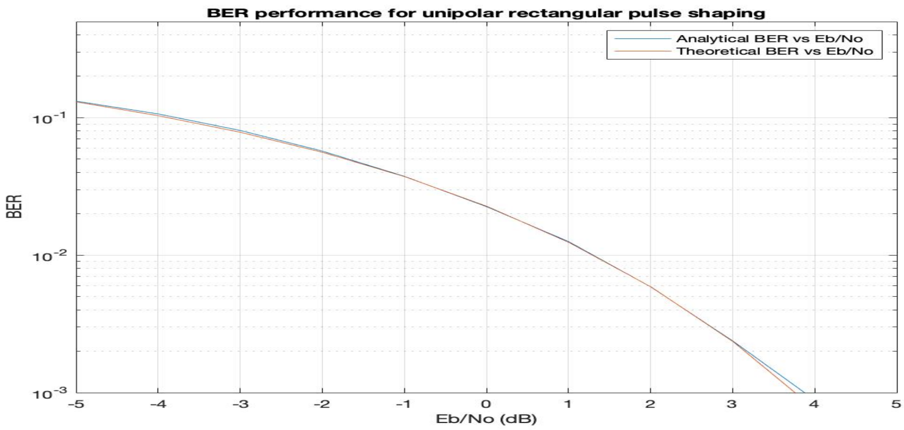


Figure 13: BER performance for bipolar rectangular pulse-shaping

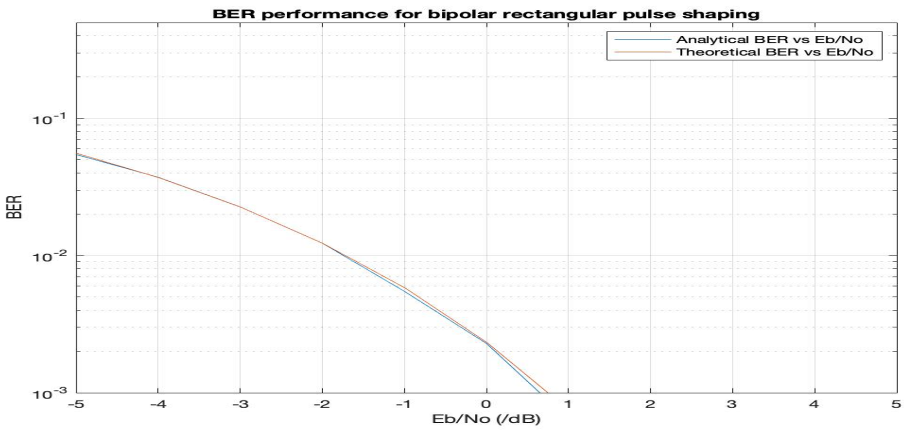


Figure 14: BER performance for bipolar rectangular pulse-shaping

## RRC pulse shaping

Figures 15 and 16 respectively display the unipolar and bipolar random binary signals that are used as the input for this section.

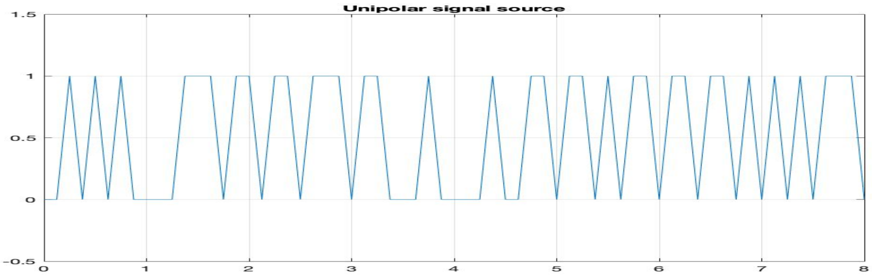


Figure 15: The unipolar source signal

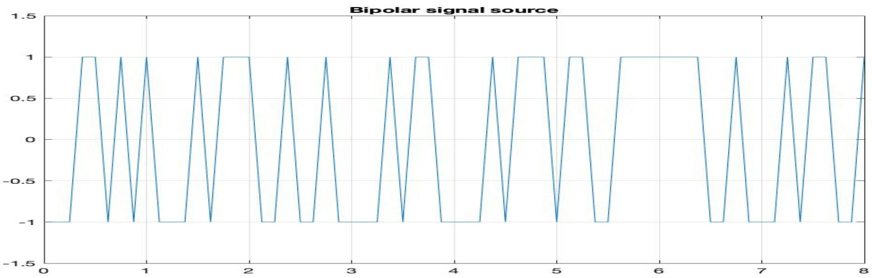


Figure 16: The bipolar source signal

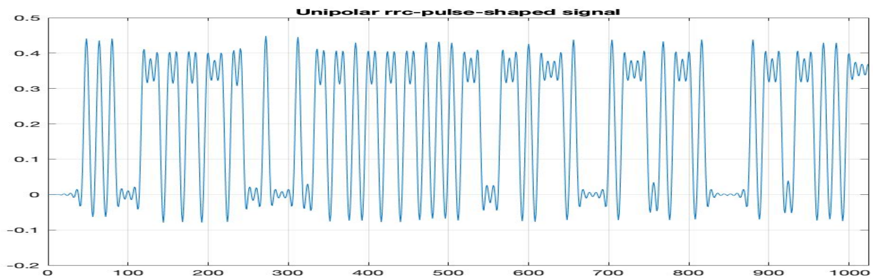


Figure 17: The unipolar RRC-pulse-shaped signal

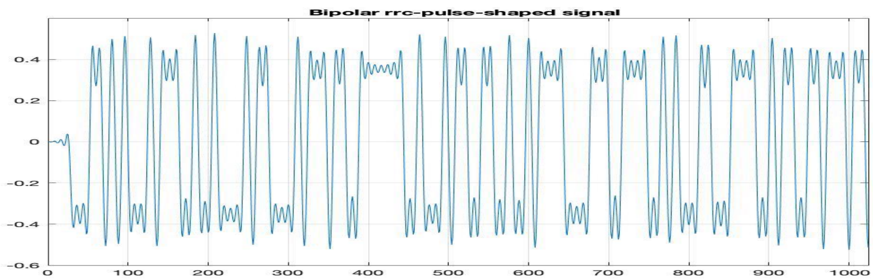


Figure 18: The bipolar RRC-pulse-shaped signal

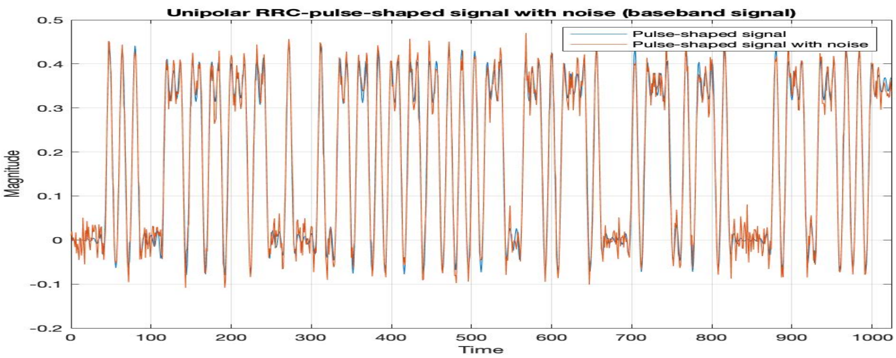


Figure 19: The unipolar RRC-pulse-shaped signal with noise (baseband)

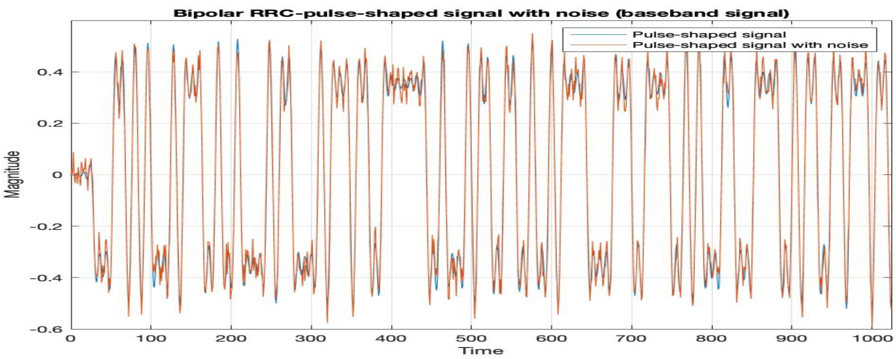


Figure 20: The bipolar RRC-pulse-shaped signal with noise (baseband)

The above figures illustrate the oversampling and RRC pulse shaping process for a given unipolar and bipolar random binary sequence. The corresponding baseband signal was plotted likewise. The power spectrum density (PSD) of these signals are displayed in the below figures (Figures 21 and 22) respectively.

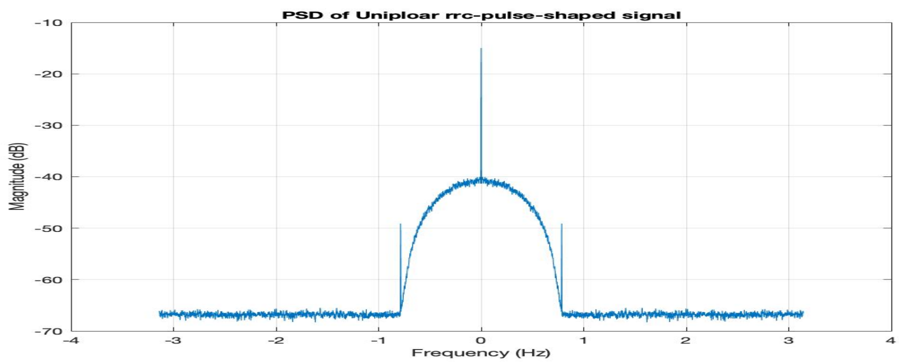


Figure 21: The power spectrum density of unipolar RRC-shaped baseband signal

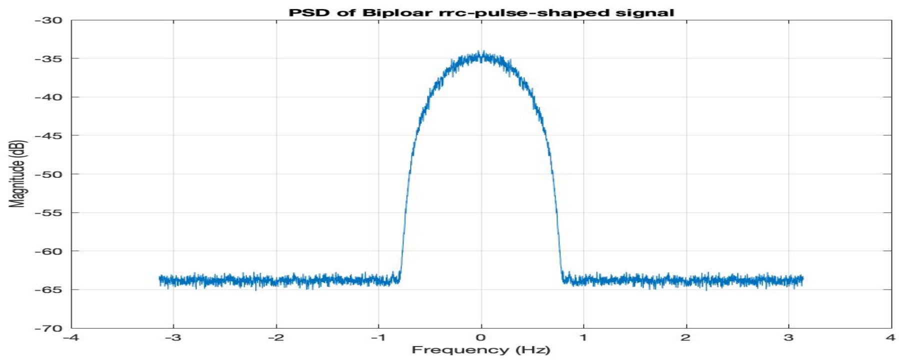


Figure 22: The power spectrum density of bipolar RRC-shaped baseband signal

The above two figures demonstrate that given the same frequency (or bandwidth), the bipolar signaling has around 6dB more power ($\times 2$ power) as unipolar signaling.

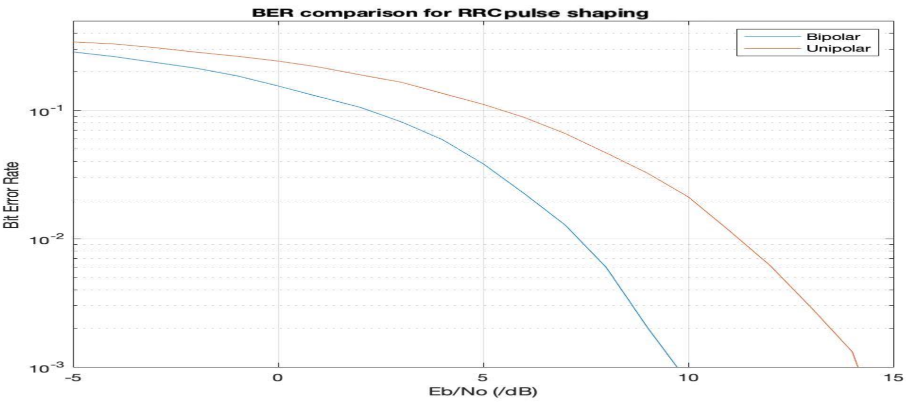


Figure 23: BER comparison for RRC pulse-shaping

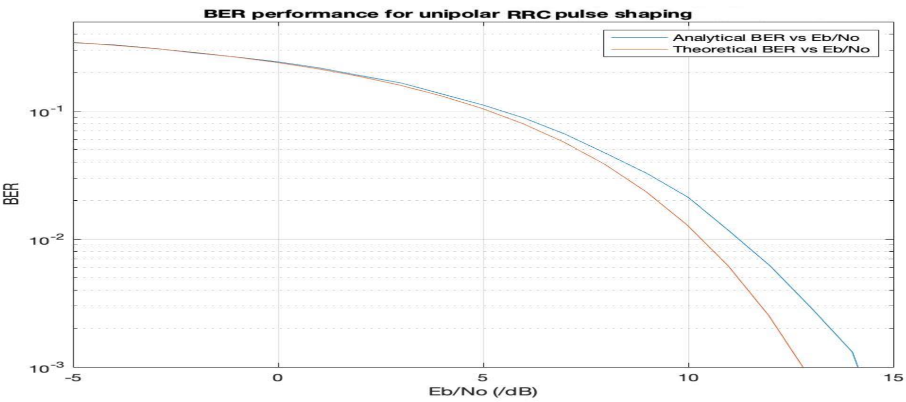


Figure 24: BER performance for bipolar RRC pulse-shaping

Moreover, based on Figure 23, it can be observed that for a given input, bipolar signaling can achieve a much lower BER in comparison to unipolar signaling. As mentioned in the math modeling section, the BER difference between the two would be 3dB, which is shown in the given figure. In addition, the below two figures (Figures 24 and 25) displays the comparison of the obtained analytical and the theoretical values of BER for both unipolar and bipolar signaling. As the analytical and the theoretical values are essentially similar, it can be claimed that the analytical performance of BER has been successfully verified.

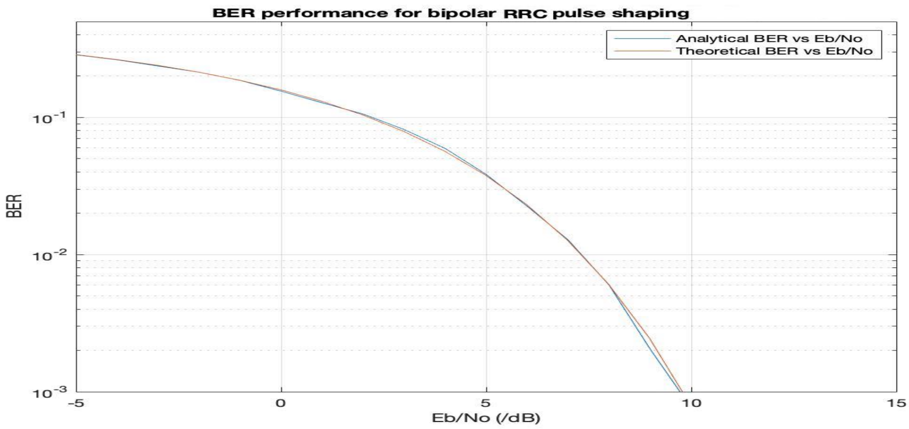


Figure 25: BER performance for bipolar RRC pulse-shaping

## Further Discussion

In addition, there are a number of questions to be explained for this assignment.

### a) Why is pulse shaping necessary?

Pulse shaping can be defined as the process of modifying a given waveform of transmitted pulses and it is used for setting a boundary for the effective bandwidth of the transmission channel. When transmitting a signal through a band-limited channel, if the signal's bandwidth becomes larger than the channel's bandwidth, mainly due to increase in the modulation rate, the channel would likely produce inter-symbol interference (ISI). By pulse shaping, the ISI can be controlled and reduced. Hence, it leads to more bandwidth efficiency, which is highly beneficial.

### b) What is the appropriate oversampling rate?

Oversampling is the process of sampling a given signal at a sampling frequency that is much higher than the Nyquist rate, where the Nyquist rate is twice the given bandwidth. When a signal is oversampled by $n$ symbols, its Signal to Noise (SNR) ratio would be $\sqrt{n}$. Therefore, the chosen oversampling must be high enough to considerably exceed the Nyquist rate but not significantly high as it would increase the SNR. There is no bandwidth for Rectangular pulse shaping. Hence, the oversampling rate could be set to an arbitrary value; it was set to 8 for this experiment due its simplicity for simulation. As for the RRC pulse-shaping, since the roll-off factor and bandwidth were set to 1, the Nyquist rate $f_s$ would be as $f_s > 2 \times bandwidth = 2$. Hence, comparatively higher values than 2, such as 8 can be appropriate.

**c) What is matched filtering and why it is necessary?**

Matched filtering is the process of convolution between an unknown signal with a conjugated known signal. With this process, the presence of the familiar signal in the unknown signal can be detected. More specifically, it is the process of detecting a known signal that is mixed with noise. Therefore, matched filtering is a linear filter that maximizes Signal to Noise Ratio (SNR) and subsequently, minimizes the BER, which could be both essential and beneficial in many applications.

# Conclusion

In this assignment, a baseband transceiver was constructed and it was analyzed for random unipolar and bipolar binary sequences as its input. The intermediate results of the transceiver were fully provided and discussed. In addition, the BER performance of the system on combinations of the waveform and pulse-shaping methods were illustrated and discussed.
Based on the comparison between the obtained results and the theory, it is believed that the process for this assignment was successful. Conclusively, this assignment was believed to be a valuable learning experience that allowed the students to think more critically and improve their skills regarding the Digital and Analougue Communications II module. Moreover, it also helped the students to better understand the material from the lectures and evidently, gain interest in this subject.

# Appendices

## Rectangular pulse shaping

```
clear, close all, clc;

% Setting the system paramters
sampleRate = 8; % the sample rate
bitRate = 1; % the bit rate
symbolNum = 4096; % number of symbols
sampleNum = 8; % number of samples per symbol
M = symbolNum*sampleNum; % number of sample columns
N = 1; % number of sample rows
signalRange = [0 1]; % Range for the random signal
sampleRateOD = 1; % used for over/down sampling

inc = bitRate/sampleRate; %The time increment
total = symbolNum*sampleNum*inc; % total data

T = 8; % signal period
t = 0:inc:total-inc; % data transmission time

% Creating the signal source
uniSrc = randi(signalRange, M, N); % Unipolar signal (0 or 1)
biSrc = 2*randi(signalRange,M, N)-1; % Bipolar signal (-1 or 1)
% Drawing the unipolar signal source
figure;
plot(t, uniSrc);
title('Unipolar signal source');
xlabel('Time');
ylabel('Magnitude');
grid on;
axis([0 8 -0.5 1.5]);
% Drawing the bipolar signal source
figure;
plot(t, biSrc);
title('Bipolar signal source');
xlabel('Time');
ylabel('Magnitude');
grid on;
axis([0 8 -1.5 1.5]);

% Rectangular Pulse Shaping
rectPulseUni = rectpulse(uniSrc, sampleNum); %unipolar
rectPulseBi = rectpulse(biSrc, sampleNum);%bipolar

% Drawing the unipolar signal source (pulse-shaped)
t1 = 0:inc:sampleNum*total-inc;
figure;
```

```
plot(t1, rectPulseUni);
title('Unipolar rect-pulse-shaped signal');
xlabel('Time');
ylabel('Magnitude');
grid on;
axis([0 64 -0.5 1.5]);

% Drawing the bipolar signal source (pulse-shaped)
figure;
plot(t1, rectPulseBi);
title('Bipolar rect-pulse-shaped signal');
xlabel('Time');
ylabel('Magnitude');
grid on;
axis([0 64 -1.5 1.5]);

% Plot the baseband with arbitary noise power
noiseRectUni = awgn(rectPulseUni, 20, 'measured'); %add noise
figure;
plot(t1, rectPulseUni);
hold on
plot(t1, noiseRectUni);
hold off
title('Unipolar rect-pulse-shaped signal with noise (baseband signal)');
xlabel('Time');
ylabel('Magnitude');
grid on;
legend('Pulse-shaped signal','Pulse-shaped signal with noise');
axis([0 64 -0.5 1.5]);
% plot the bipolar baseband
noiseRectBi = awgn(rectPulseBi, 20, 'measured'); %add noise
figure;
plot(t1, rectPulseBi);
hold on
plot(t1, noiseRectBi);
hold off
title('Bipolar rect-pulse-shaped signal with noise (baseband signal)');
xlabel('Time');
ylabel('Magnitude');
grid on;
legend('Pulse-shaped signal','Pulse-shaped signal with noise');
axis([0 64 -1.5 1.5]);

PSDW = 4096; % the PSD window size
%Plotting PSD for unipolar signaling
figure;
[pxx, freq] = pwelch(noiseRectUni, PSDW, PSDW/2, 'centered', 'power');
power = 10*log10(pxx);
plot(freq, power);
```

```
title('PSD of Uniploar baseband signal (rect-pulse-shaped)');
xlabel('Frequency (Hz)');
ylabel('Magnitude (dB)');
grid on;
%Plotting PSD for bipolar signaling
figure;
[pxx, freq] = pwelch(noiseRectBi, PSDW, PSDW/2, 'centered', 'power');
power = 10*log10(pxx);
plot(freq, power);
title('PSD of Biploar baseband signal (rect-pulse-shaped)');
xlabel('Frequency (Hz)');
ylabel('Magnitude (dB)');
grid on;

% Applying channel model
N0 = -10:1:10; %arbitrary channel power
BER_Uni= zeros(length(N0), 1); %list of BER for unipolar
BER_Bi= zeros(length(N0), 1); %list of BER for bipolar
index = 1;%indicates the BER index
for N0 = -10:1:10
noiseRectUni = awgn(rectPulseUni, N0, 'measured');%add noise
noiseRectBi = awgn(rectPulseBi, N0, 'measured');%add noise

% Matched Filter
HF=ones(sampleRate, 1);
% Matched Filtering for unipolar
rcvSignal = conv(noiseRectUni, HF).';
% circular shifting
rcvSignal= circshift(rcvSignal,[0, -7]);
%down sampling
rcvUni = downsample(rcvSignal, sampleNum);
% Matched Filtering for bipolar
rcvSignal = conv(noiseRectBi, HF).';
% circular shifting
rcvSignal= circshift(rcvSignal,[0, -7]);
%down sampling
rcvBi = downsample(rcvSignal, sampleNum);

% Pulse detection for unipolar
resultUni= [];% the results of pulse detection
for i=1:length(rcvUni)
if (rcvUni(i)>4) %compare to threshold
resultUni(i) = 1;
else
resultUni(i) = 0;
end
end
% Pulse detection for bipolar
resultBi= [];% the results of pulse detection
```

```
for i=1:length(rcvBi)
if (rcvBi(i)>0) %compare to threshold
resultBi(i) = 1;
else
resultBi(i) = -1;
end
end

errorNum = 0; %number of errors
for i=1:length(uniSrc)
if (uniSrc(i)~=resultUni(i))%if bit error
errorNum= errorNum+1;
end
end
%calculate BER
BER_Uni(index)= errorNum/length(uniSrc);

errorNum = 0;%number of errors
for i=1:length(biSrc)
if (biSrc(i)~=resultBi(i))%if bit error
errorNum= errorNum+1;
end
end
%calculate BER
BER_Bi(index)= errorNum/length(biSrc);
%increment index for next loop
index = index+1;
end

N0 = -10:1:10;%arbitrary noise power
EbN0 = 8./(10.^(N0./10)); %SNR for unipolar
tBER=qfunc(sqrt(EbN0)); %Theoretical BER for unipolar
B = 2/T; %bandwidth
figure;
%Draw the analytical BER vs Eb/No
semilogy(N0,(BER_Uni));
hold on
%Draw the theoretical BER vs Eb/No
semilogy(10*log10(B.*EbN0),(tBER));
hold off
title('BER performance for unipolar rectangular pulse shaping');
legend('Analytical BER vs Eb/No','Theoretical BER vs Eb/No');
xlabel('Eb/No (dB)');
ylabel('BER');
grid on;
axis([-5 5 0.001 0.5]);

figure;
N0 = -10:1:10;%arbitrary noise power
```

```
EbN0 = 8./(10.^(N0./10));%Theoretical BER for unipolar
B = 2/T;%bandwidth
tBER=qfunc(sqrt(2*EbN0)); %Theoretical BER for unipolar
%Draw the analytical BER vs Eb/No
semilogy(N0,(BER_Bi));
hold on
%Draw the theoretical BER vs Eb/No
semilogy(10*log10(B.*EbN0),(tBER));
hold off
title('BER performance for bipolar rectangular pulse shaping');
legend('Analytical BER vs Eb/No','Theoretical BER vs Eb/No');
xlabel('Eb/No (/dB)');
ylabel('BER');
grid on;
axis([-5 5 0.001 0.5]);

figure;
%Plot BER for bipolar
semilogy(N0,(BER_Bi));
hold on
%Plot BER for unipolar
semilogy(N0,(BER_Uni));
hold off
title('BER comparison for rectangular pulse shaping');
legend('Bipolar','Unipolar');
xlabel('Eb/No (/dB)');
ylabel('Bit Error Rate ');
grid on;
axis([-5 5 0.001 0.5]);
```

## RRC pulse shaping

```
clear, close all, clc;

% Setting the system paramters
sampleRate = 8; % the sample rate
bitRate = 1; % the bit rate
rollOff = 1; % roll-off factor
symbolNum = 4096; % number of symbols
sampleNum = 8; % number of samples per symbol
M = symbolNum*sampleNum; % number of sample columns
N = 1; % number of sample rows
signalRange = [0 1]; % Range for the random signal
sampleRateOD = 1; % used for over/down sampling
span=8; %used for the truncating window

inc = bitRate/sampleRate;%The time increment
total = symbolNum*sampleNum*inc;% total data
```

```
T = 8; % signal period
t = 0:inc:total-inc; % data transmission time

% Creating the signal source
uniSrc = randi(signalRange, M, N); % Unipolar signal (0 or 1)
biSrc = 2*randi(signalRange,M, N)-1; % Bipolar signal (-1 or 1)

% Drawing the unipolar signal source
figure;
plot(t, uniSrc);
title('Unipolar signal source');
grid on;
axis([0 8 -0.5 1.5]);
% Drawing the bipolar signal source
figure;
plot(t, biSrc);
title('Bipolar signal source');
grid on;
axis([0 8 -1.5 1.5]);

% --- RRC Pulse Shaping
% RRC coefficients
coeffs = rcosdesign(rollOff, span, sampleNum, 'sqrt');
% Oversampled and RCC shaped Unipolar signal
rrcPulseUni = upfirdn(uniSrc, coeffs, sampleNum);
% Oversampled and RCC shaped Bipolar signal
rrcPulseBi = upfirdn(biSrc, coeffs, sampleNum);

% Drawing the unipolar signal source (pulse-shaped)
t1 = 0:1:size(rrcPulseUni, 1)-1;
figure;
plot(t1, rrcPulseUni);
title('Unipolar rrc-pulse-shaped signal');
grid on;
axis([0 1024 -0.2 0.5]);

% Drawing the unipolar signal source (pulse-shaped)
figure;
plot(t1, rrcPulseBi);
title('Bipolar rrc-pulse-shaped signal');
grid on;
axis([0 1024 -0.6 0.6]);

% Plot the baseband with arbitary noise power
noiseRRCUni = awgn(rrcPulseUni, 20, 'measured'); % add noise to signal
figure;
plot(t1, rrcPulseUni);
hold on
```

```
plot(t1, noiseRRCUni);
hold off
title('Unipolar RRC-pulse-shaped signal with noise (baseband signal)');
xlabel('Time');
ylabel('Magnitude');
grid on;
legend('Pulse-shaped signal','Pulse-shaped signal with noise');
axis([0 1024 -0.2 0.5]);
% plot the bipolar baseband
noiseRRCBi = awgn(rrcPulseBi, 20, 'measured');
figure;
plot(t1, rrcPulseBi);
hold on
plot(t1, noiseRRCBi);
hold off
title('Bipolar RRC-pulse-shaped signal with noise (baseband signal)');
xlabel('Time');
ylabel('Magnitude');
grid on;
legend('Pulse-shaped signal','Pulse-shaped signal with noise');
axis([0 1024 -0.6 0.6]);

PSDW = 4096; % the PSD window size
%Plotting PSD for unipolar signaling
figure;
[pxx, freq] = pwelch(noiseRRCUni, PSDW, PSDW/2, 'centered', 'power');
power = 10*log10(pxx);
plot(freq, power);
title('PSD of Uniploar rrc-pulse-shaped signal');
xlabel('Frequency (Hz)');
ylabel('Magnitude (dB)');
grid on;
%Plotting PSD for bipolar signaling
figure;
[pxx, freq] = pwelch(noiseRRCBi, PSDW, PSDW/2, 'centered', 'power');
power = 10*log10(pxx);
plot(freq, power);
title('PSD of Biploar rrc-pulse-shaped signal');
xlabel('Frequency (Hz)');
ylabel('Magnitude (dB)');
grid on;

% Applying channel model
N0 = -40:1:40 %arbitrary channel power
BER_Uni= zeros(length(N0), 1); %list of BER for unipolar
BER_Bi= zeros(length(N0), 1); %list of BER for bipolar
index = 1;%indicates the BER index
for N0 = -40:1:40
SNR= N0- 10*log10(8); %calculate SNR
```

```
noiseRRCUni = awgn(rrcPulseUni, SNR, 'measured');%add noise
noiseRRCBi = awgn(rrcPulseBi, SNR, 'measured');%add noise

% Matched Filter and downsample for unipolar
rcvSignal = upfirdn(noiseRRCUni, coeffs, 1, sampleNum);
% Removing filter delay for unipolar
rcvUni = rcvSignal((sampleNum)+1:end-(sampleNum));
% Matched Filter and downsample for bipolar
rcvSignal = upfirdn(noiseRRCBi, coeffs, 1, sampleNum);
% Removing filter delay for bipolar
rcvBi = rcvSignal((sampleNum)+1:end-(sampleNum));

% Pulse detection for unipolar
resultUni= []; % the results of pulse detection
for i=1:length(rcvUni)
if (rcvUni(i)>0.4) %compare to threshold
resultUni(i) = 1;
else
resultUni(i) = 0;
end
end

% Pulse detection and down sampling for unipolar
resultBi= []; % the results of pulse detection
for i=1:length(rcvBi)
if (rcvBi(i)>0) %compare to threshold
resultBi(i) = 1;
else
resultBi(i) = -1;
end
end

errorNum = 0; %number of errors
for i=1:length(uniSrc)
if (uniSrc(i)~=resultUni(i)) %if bit error
errorNum= errorNum+1;
end
end
%calculate BER
BER_Uni(index)= errorNum/length(uniSrc);

errorNum = 0; %number of errors
for i=1:length(biSrc)
if (biSrc(i)~=resultBi(1, i)) %if bit error
errorNum= errorNum+1;
end
end
%calculate BER
BER_Bi(index)= errorNum/length(biSrc);
```

```
%increment index for next loop
index = index+1;
end
N0 = -40:1:40; %arbitrary noise power
EbN0 = 0.0625./(10.^(N0./10)); %SNR for unipolar
tBER=qfunc(sqrt(EbN0)); %Theoretical BER for unipolar
figure;
%Draw the analytical BER vs Eb/No
semilogy(N0,(BER_Uni));
hold on
%Draw the theoretical BER vs Eb/No
semilogy(10*log10(2*EbN0),(tBER));
hold off
title('BER performance for unipolar RRC pulse shaping');
legend('Analytical BER vs Eb/No','Theoretical BER vs Eb/No');
xlabel('Eb/No (/dB)');
ylabel('BER');
grid on;
axis([-5 15 0.001 0.5]);
EbN0 = 0.25./(10.^(N0./10));%SNR for bipolar
tBER=qfunc(sqrt(EbN0)); %Theoretical BER for bipolar
figure;
%Draw the analytical BER vs Eb/No
semilogy(N0,(BER_Bi));
hold on
%Draw the theoretical BER vs Eb/No
semilogy(10*log10(EbN0),(tBER));
hold off
title('BER performance for bipolar RRC pulse shaping');
legend('Analytical BER vs Eb/No','Theoretical BER vs Eb/No');
xlabel('Eb/No (/dB)');
ylabel('BER');
grid on;
axis([-5 15 0.001 0.5]);
N0 = -40:1:40; %arbitrary noise power
figure;
%Plot BER for bipolar
semilogy(N0,(BER_Bi));
hold on
%Plot BER for unipolar
semilogy(N0,(BER_Uni));
hold off
title('BER comparison for RRC pulse shaping');
legend('Bipolar','Unipolar')
xlabel('Eb/No (/dB)');
ylabel('Bit Error Rate ');
grid on;
axis([-5 15 0.001 0.5]);
```