G. S. Sanyal School of Telecommunications
Indian Institute of Technology Kharagpur

MOOC: Spread Spectrum Communications & Jamming

Assignment 7: Solution to MATLAB code for BER generation of Spread Spectrum QPSK system over AWGN Channel.

Due date: Max. marks: 20

Write a MATLAB code to generate Bit Error Rate (BER) vs Bit-Energy-to-Noise-Power-Spectral-Density ratio (Eb/No) and Signal-Power-to-Noise-Power ratio (SNR) plot for Spread Spectrum Quadrature Phase Shift Keying (QPSK) system over Additive White Guassian Noise (AWGN) channel. Assume system employs a Hadamard sequence of length 4 for spreading data. Fig. 1 depicts a QPSK Spread Spectrum modulator and demodulator system. Referring to the same, and following the table of notations (Table 1), answer the following questions:

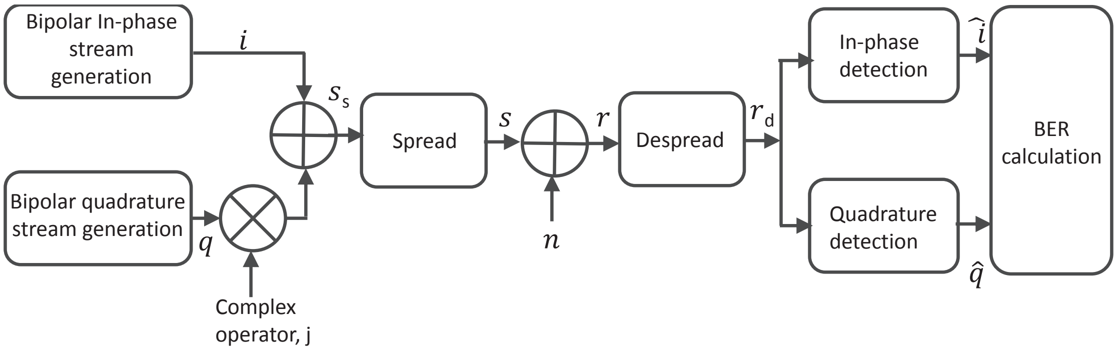


Figure 1: QPSK Spread Spectrum modulator modulator and demodulator system

Parameter	**Mathematical notation**	**MATLAB variable**
In-phase bipolar data	i	I_data
Quadrature bipolar data	q	Q_data
Baseband complex data	s_s	base_sig
Spreading code	c	spread_code
Transmit spread signal	s	tx_data
AWGN channel noise	n	n_AWGN
Received signal	r	rec_data
Despread signal	r_d	despread_data
Decoded in-phase data	$\hat{i}$	decod_sig_I
Decoded quadrature data	$\hat{q}$	decod_sig_Q

Table 1: Table of notations.

1. The spreading code can be generated using the sequence of MATLAB commands:
   Solution: The spreading code can be generated using commands in line numbers 14-15 of the appended MATLAB code.
   (correct option ii.)

2. The 3^{rd} row of the Hadamard matrix, code_mat, in question 1 is the code sequence:
   Solution: The Hadamard matrix, code_mat is generated using command in the line number 14-15 of the appended MATLAB code. The 3^{rd} row of the Hadmard matrix corresponds to '1 1 -1 -1' of the specified length (len_code =4)
   (correct option ii.)

3. The MATLAB command to generate the spread signal:
   Solution: The spread signal is generated using command in the line number 22 of the appended MATLAB code.
   (correct option iv.)

4. The power of the spread QPSK data stream can be calculated using the following MATLAB command:
   Solution: The power of the spread QPSK data stream is calculated using the command in the line number 23 of the appended MATLAB code.
   (correct option i.)

5. In comparison to the BER vs Eb/No plot of a conventional QPSK modulation scheme (without spreading), the BER vs Eb/No plot for the Spread Spectrum QPSK scheme is offset towards the left by:
   Solution: Since bit energy before and after the spreading process are same, BER vs

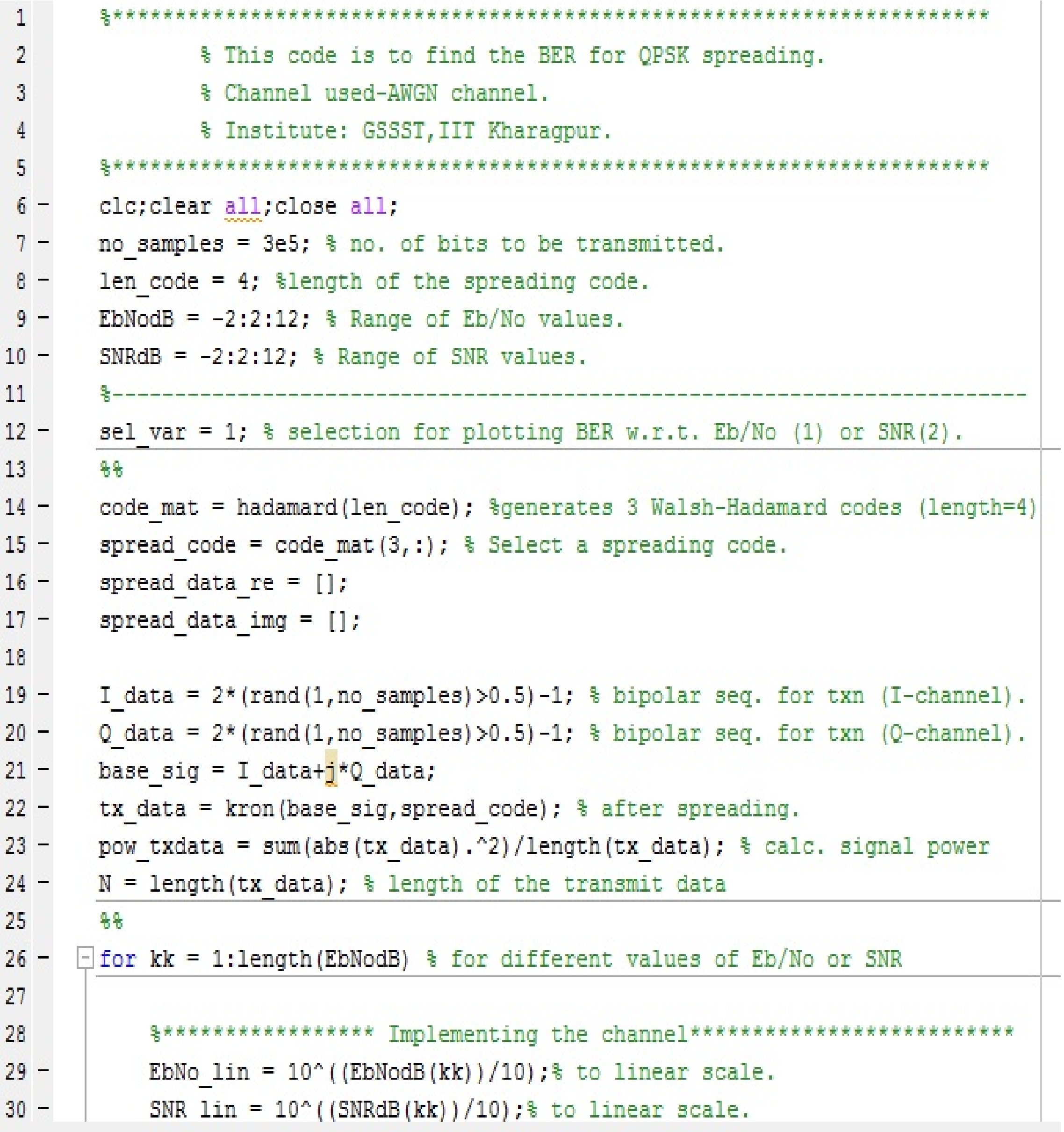
Eb/No for spread QPSK is same as unspread QPSK scheme (Refer Assignment 6).
(correct option i.)

6. In comparison to the BER vs SNR plot of a conventional QPSK modulation scheme (without spreading), the BER vs SNR plot for the Spread Spectrum QPSK scheme is offset towards the left by:
   Solution: Since despreading results the signal power in L (length of the spreading code) bits adding up, BER vs SNR for spread QPSK shows an offset towards left of $10log_{10}(L)$=6 dB.
   (correct option iii.)

7. The Bit Error Rate plot versus Eb/No for the spread spectrum QPSK system is:
   Solution: The BER vs Eb/No is plotted using commands in line numbers 62-64 with sel_var=1 of the appended MATLAB code.
   (correct option i.)

8. The Bit Error Rate plot versus SNR for the spread spectrum QPSK system is:
   Solution: The BER vs SNR is plotted using commands in line numbers 66-68 with sel_var=2 of the appended MATLAB code.
   (correct option ii.)

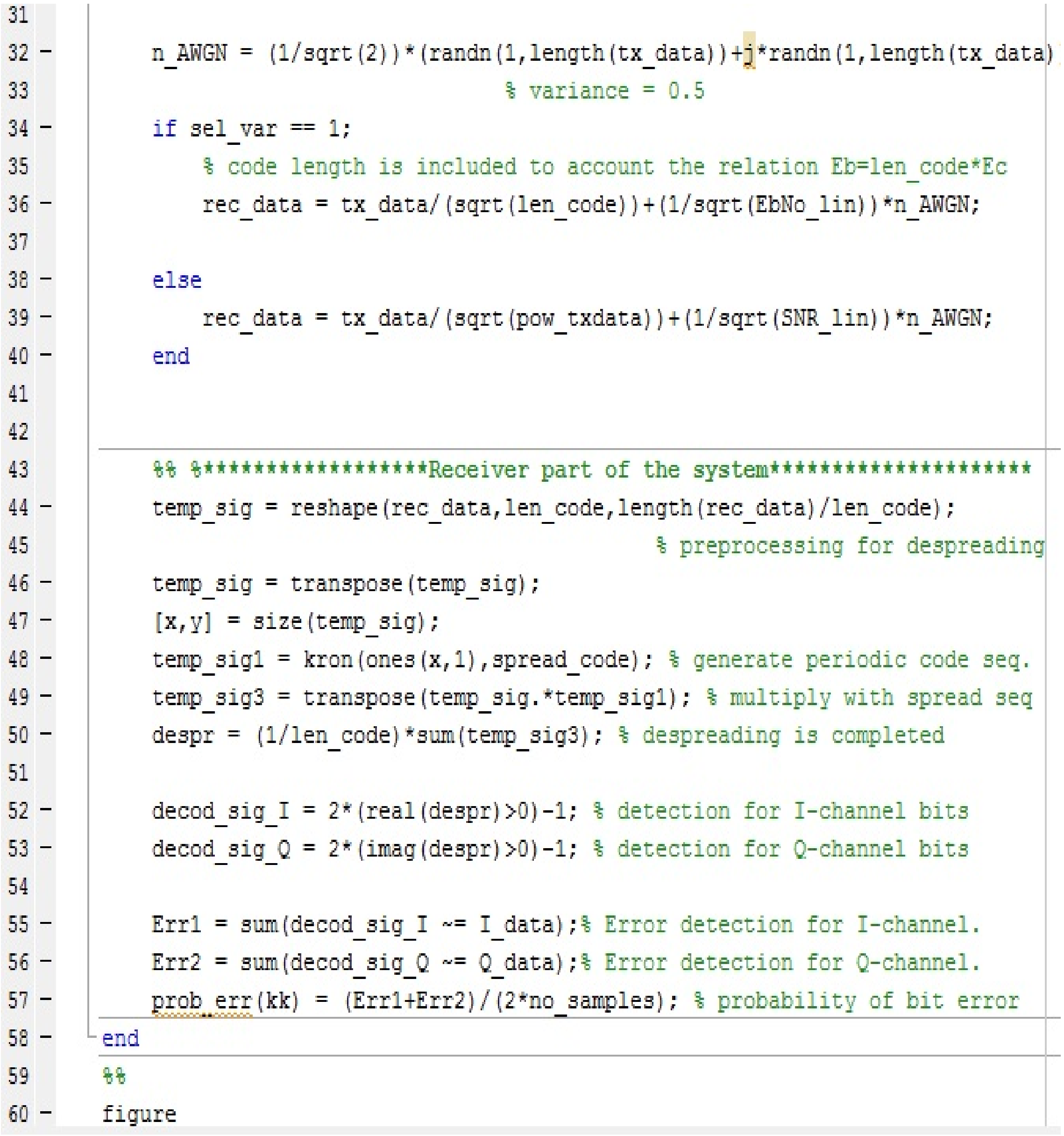
Appendix



```
%**********************************************************************
        % This code is to find the BER for QPSK spreading.
        % Channel used-AWGN channel.
        % Institute: GSSST,IIT Kharagpur.
%**********************************************************************
clc;clear all;close all;
no_samples = 3e5; % no. of bits to be transmitted.
len_code = 4; %length of the spreading code.
EbNodB = -2:2:12; % Range of Eb/No values.
SNRdB = -2:2:12; % Range of SNR values.
%--------------------------------------------------------------------------
sel_var = 1; % selection for plotting BER w.r.t. Eb/No (1) or SNR(2).
%%
code_mat = hadamard(len_code); %generates 3 Walsh-Hadamard codes (length=4)
spread_code = code_mat(3,:); % Select a spreading code.
spread_data_re = [];
spread_data_img = [];

I_data = 2*(rand(1,no_samples)>0.5)-1; % bipolar seq. for txn (I-channel).
Q_data = 2*(rand(1,no_samples)>0.5)-1; % bipolar seq. for txn (Q-channel).
base_sig = I_data+j*Q_data;
tx_data = kron(base_sig,spread_code); % after spreading.
pow_txdata = sum(abs(tx_data).^2)/length(tx_data); % calc. signal power
N = length(tx_data); % length of the transmit data
%%
for kk = 1:length(EbNodB) % for different values of Eb/No or SNR

    %***************** Implementing the channel*************************
    EbNo_lin = 10^((EbNodB(kk))/10);% to linear scale.
    SNR_lin = 10^((SNRdB(kk))/10);% to linear scale.
```



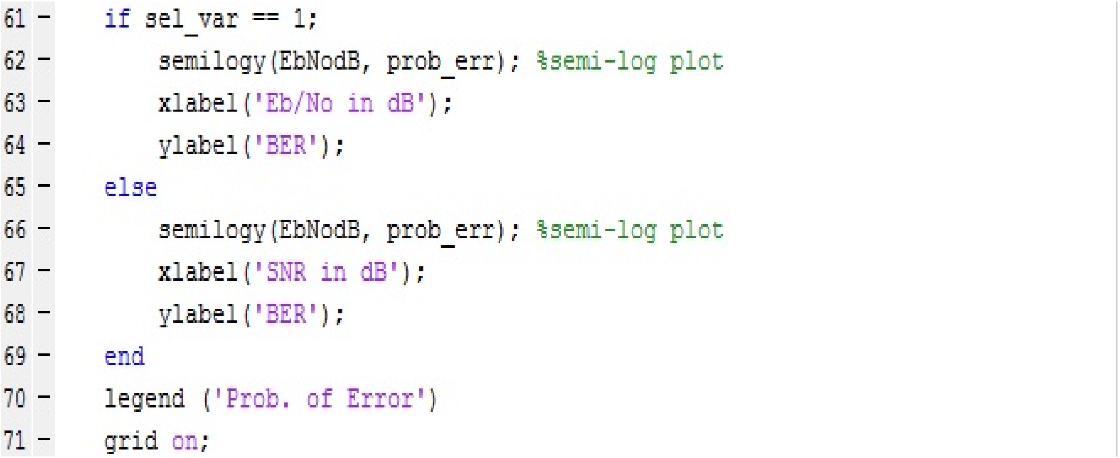
```
    n_AWGN = (1/sqrt(2))*(randn(1,length(tx_data))+j*randn(1,length(tx_data)
                                 % variance = 0.5
    if sel_var == 1;
        % code length is included to account the relation Eb=len_code*Ec
        rec_data = tx_data/(sqrt(len_code))+(1/sqrt(EbNo_lin))*n_AWGN;

    else
        rec_data = tx_data/(sqrt(pow_txdata))+(1/sqrt(SNR_lin))*n_AWGN;
    end


    %% %******************Receiver part of the system*********************
    temp_sig = reshape(rec_data,len_code,length(rec_data)/len_code);
                                             % preprocessing for despreading
    temp_sig = transpose(temp_sig);
    [x,y] = size(temp_sig);
    temp_sig1 = kron(ones(x,1),spread_code); % generate periodic code seq.
    temp_sig3 = transpose(temp_sig.*temp_sig1); % multiply with spread seq
    despr = (1/len_code)*sum(temp_sig3); % despreading is completed

    decod_sig_I = 2*(real(despr)>0)-1; % detection for I-channel bits
    decod_sig_Q = 2*(imag(despr)>0)-1; % detection for Q-channel bits

    Err1 = sum(decod_sig_I ~= I_data);% Error detection for I-channel.
    Err2 = sum(decod_sig_Q ~= Q_data);% Error detection for Q-channel.
    prob_err(kk) = (Err1+Err2)/(2*no_samples); % probability of bit error
end
%%
figure
```



```
if sel_var == 1;
    semilogy(EbNodB, prob_err); %semi-log plot
    xlabel('Eb/No in dB');
    ylabel('BER');
else
    semilogy(EbNodB, prob_err); %semi-log plot
    xlabel('SNR in dB');
    ylabel('BER');
end
legend ('Prob. of Error')
grid on;
```