# Homework for Lecture # 01

**Student ID: Q36114221**

**Name: SU PEI-JIN**

1. Solution # 01:

The signal constellation for $M$-ary PSK has $s_{i1} = A\cos\left[\frac{2\pi(i-1)}{M}\right]$ and $s_{i2} = A\sin\left[\frac{2\pi(i-1)}{M}\right]$ for $i = 1, \ldots, M$. The symbol energy is $E_s = A^2$. This constellation and the decision region $D_1$ are shown in Figure 1.

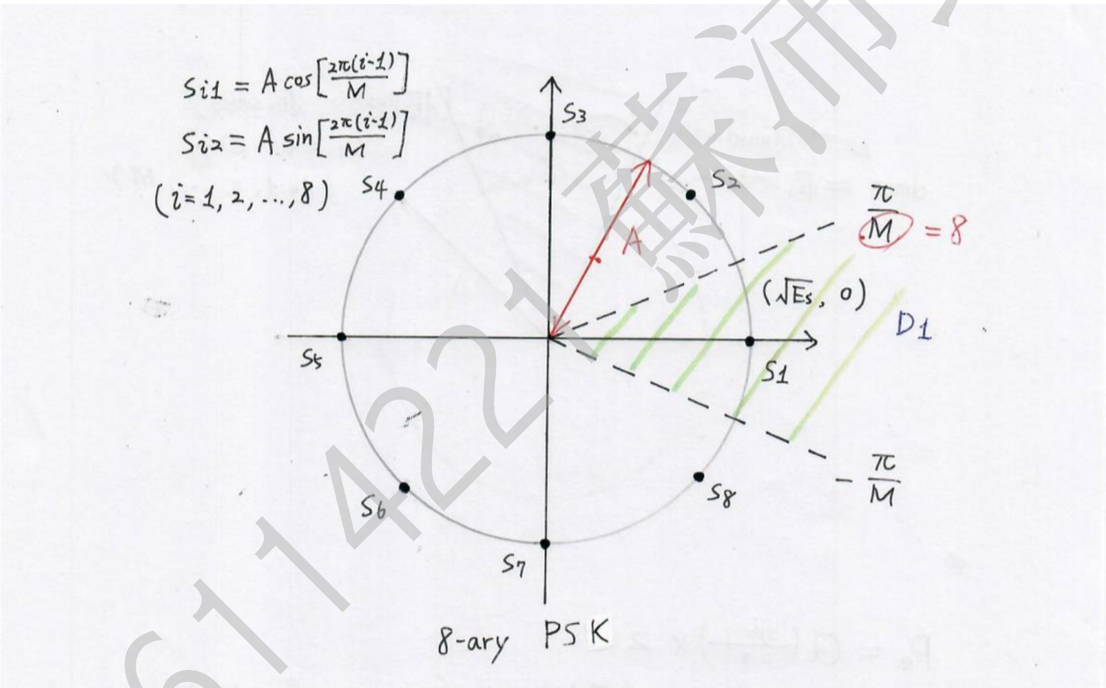



Figure 1. The signal constellation for 8-ary PSK

Note that since we assume the symbols are equiprobable, the decision regions are based on the minimum-distance detection rule. By symmetry of the constellation, the probability of symbol error for this constellation is equal to the error probability when $\boldsymbol{s}_1 = \left(\sqrt{E_s}, 0\right)$ is transmitted. The received vector $\boldsymbol{x}$ is given by

$$\boldsymbol{x} = (x_1, x_2) = \left(\sqrt{E_s} + n_1, n_2\right) \tag{1}$$

It is seen that $x_1$ and $x_2$ are independent Gaussian random variables with variance

$\sigma^2 = \frac{1}{2}N_0$ and means $\sqrt{E_s}$ and 0, respectively; hence

$$p(x_1, x_2) = \left(\left(\frac{1}{\sqrt{2\pi}\sigma}\right)e^{-\frac{\left(x_1-\sqrt{E_s}\right)^2}{2\sigma^2}}\right) \times \left(\left(\frac{1}{\sqrt{2\pi}\sigma}\right)e^{-\frac{x_2^2}{2\sigma^2}}\right)$$

$$= \frac{1}{\pi N_0}e^{-\frac{\left(x_1-\sqrt{E_s}\right)^2+x_2^2}{N_0}} \tag{2}$$

Since the received vector $x = re^{\mathrm{j}\theta}$ can be represented in polar, we introduce polar coordinates transformations of $(x_1, x_2)$ as

$$\begin{aligned} R &= \sqrt{x_1^2 + x_2^2} \\ \Theta &= \tan^{-1}\frac{x_2}{x_1} \end{aligned} \tag{3}$$

from which the joint pdf of $R$ and $\Theta$ can be derived as

$$p_{R,\Theta}(r,\theta) = \frac{r}{\pi N_0}e^{-\frac{r^2-2\sqrt{E_s}r\cos\theta+E_s}{N_0}} \tag{4}$$

By integrating over $r$, we derive the marginal pdf of $\Theta$ as

$$\begin{aligned} p_\Theta(\theta) &= \int_0^\infty p_{R,\Theta}(r,\theta)\,dr \\ &= \frac{1}{2\pi}e^{-\gamma_s\sin^2\theta}\int_0^\infty r\,e^{-\frac{\left(r-\sqrt{2\gamma_s}\cos\theta\right)^2}{2}}dr \end{aligned} \tag{5}$$

in which we have defined the *symbol SNR* or *SNR* per *symbol* as

$$\gamma_s = \frac{A^2}{N_0} = \frac{E_s}{N_0} \tag{6}$$

The decision region $D_1$ can be described as $D_1 = \{\theta : -\pi/M < \theta < \pi/M\}$; therefore, the symbol error probability is given by

$$P_e = 1 - \int_{-\pi/M}^{\pi/M} p_\Theta(\theta)d\theta \tag{7}$$

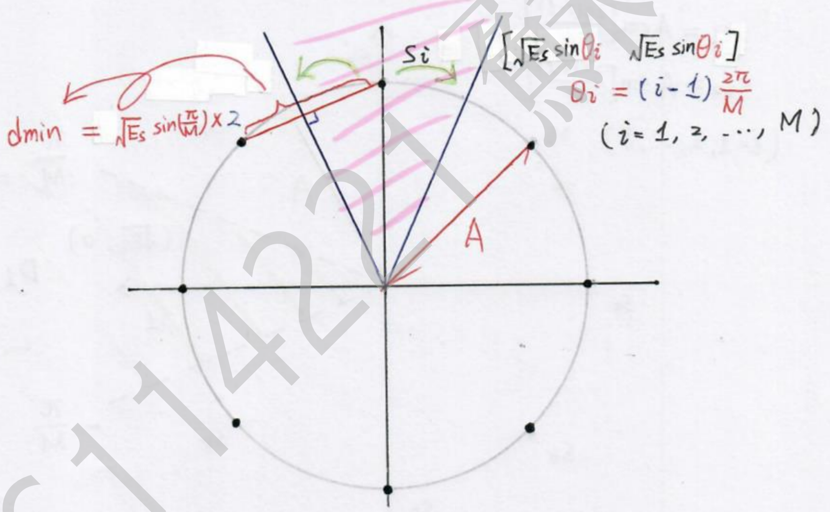
An approximation to the probability of symbol error for this constellation can be obtained by first approximating $p_\Theta(\theta)$. For $E_s/N_0 \gg 1$ and $|\theta| \leq \frac{1}{2}\pi$, $p_\Theta(\theta)$ is well approximated as

$$p_\Theta(\theta) \approx \sqrt{\frac{\gamma_s}{\pi}} \cos\theta \, e^{-\gamma_s \sin^2\theta} \tag{8}$$

By substituting for $p_\Theta(\theta)$ in Equation (7) and performing the change in variable from $\theta$ to $u = \sqrt{\gamma_s}\sin\theta$, we find that

$$\begin{aligned} P_e &\approx 1 - \int_{-\pi/M}^{\pi/M} \sqrt{\frac{\gamma_s}{\pi}} \cos\theta \, e^{-\gamma_s \sin^2\theta} d\theta \\ &\approx \frac{2}{\sqrt{\pi}} \int_{\sqrt{2\gamma_s}\sin\left(\frac{\pi}{M}\right)}^{\infty} e^{-u^2}\, du \\ &= 2Q\left(\sqrt{2\gamma_s}\sin\left(\frac{\pi}{M}\right)\right) \end{aligned} \tag{9}$$

Solution # 02:




$P_e = Q\left(\frac{d_{min}}{2\sigma}\right) \times 2$

$= 2Q\left(\frac{2\sqrt{E_s}\sin\left(\frac{\pi}{M}\right)}{2\sigma}\right)$ $\quad \left(d_{min} = 2\sqrt{E_s}\sin\left(\frac{\pi}{M}\right)\right)$

$= 2Q\left(\frac{\cancel{2}\sqrt{E_s}\sin\left(\frac{\pi}{M}\right)}{\cancel{2}\times\sqrt{\frac{N_0}{2}}}\right)$ $\quad \left(\sigma^2 = \frac{N_0}{2} \Rightarrow \sigma = \sqrt{\frac{N_0}{2}}\right)$

$= 2Q\left(\sqrt{\frac{2E_s}{N_0}}\sin\left(\frac{\pi}{M}\right)\right)$ $\quad \left(E_s = A^2 \Rightarrow \gamma_s = \frac{A^2}{N_0} = \frac{E_s}{N_0}\right)$

$= 2Q\left(\sqrt{2\gamma_s}\sin\left(\frac{\pi}{M}\right)\right)$

2. Solution:

$$y^2 \geq (y-x)^2 + x^2$$

$$\Longrightarrow \frac{y^2}{2} \geq \frac{(y-x)^2+x^2}{2}$$

$$\Longrightarrow -\frac{y^2}{2} \leq -\frac{(y-x)^2+x^2}{2}$$

$$\Longrightarrow e^{-\frac{y^2}{2}} \leq e^{-\frac{(y-x)^2+x^2}{2}}$$

$$\Longrightarrow \int_x^\infty e^{-\frac{y^2}{2}}dy \leq \int_x^\infty e^{-\frac{(y-x)^2+x^2}{2}}dy$$

$$\Longrightarrow \boxed{\frac{1}{\sqrt{2\pi}}\int_x^\infty e^{-\frac{y^2}{2}}dy} \leq \boxed{\frac{1}{\sqrt{2\pi}}\int_x^\infty e^{-\frac{(y-x)^2+x^2}{2}}dy}$$

$$\Downarrow$$

$$\boxed{Q(x) \triangleq \frac{1}{\sqrt{2\pi}}\int_x^\infty e^{-\frac{t^2}{2}}dt \quad (t=y)}$$

$$\boxed{\frac{1}{\sqrt{2\pi}}\int_x^\infty e^{-\frac{(y-x)^2+x^2}{2}}dy} = \frac{1}{\sqrt{2\pi}}\int_x^\infty e^{-\frac{(y-x)^2}{2}}\cdot e^{-\frac{x^2}{2}}\cdot dy$$

$$= \frac{1}{\sqrt{2\pi}}e^{-\frac{x}{2}}\int_x^\infty e^{-\boxed{\frac{(y-x)^2}{2}}}\cdot dy$$

Let $t=\frac{(y-x)^2}{2}$, $\frac{dt}{dy}=y-x \Longrightarrow dy=\frac{1}{y-x}dt$

$(y-x)^2=2t \Longrightarrow t=\sqrt{y-x}$

$$= \frac{1}{\sqrt{2t}}dt$$

$$= \frac{1}{\sqrt{2}}t^{-\frac{1}{2}}dt$$

$$\frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \int_x^\infty e^{-\frac{(y-x)^2}{2}} dy$$

$$= \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} \int_0^\infty e^{-t} \left(\frac{1}{\sqrt{2}} t^{-\frac{1}{2}}\right) dt$$

$$= \frac{1}{2\sqrt{\pi}} e^{-\frac{x^2}{2}} \boxed{\int_0^\infty t^{-\frac{1}{2}} e^{-t} dt}$$

$$\int_0^\infty t^{-\frac{1}{2}} e^{-t} dt = \int_0^\infty t^{(\frac{1}{2}-1)} e^{-t} dt$$

$$= \Gamma\left(\tfrac{1}{2}\right) = \sqrt{\pi}$$

Gamma function :

$$\Gamma(x) \triangleq \int_0^\infty t^{(x-1)} e^{-t} dt$$

$$\Gamma\left(\tfrac{1}{2}\right) = \sqrt{\pi}$$

$$= \frac{1}{2\cancel{\sqrt{\pi}}} \cdot e^{-\frac{x^2}{2}} \cdot \cancel{\sqrt{\pi}} = \frac{1}{2} e^{-\frac{x^2}{2}}$$

$$\because \frac{1}{2} e^{-\frac{x^2}{2}} \le e^{-\frac{x^2}{2}}$$

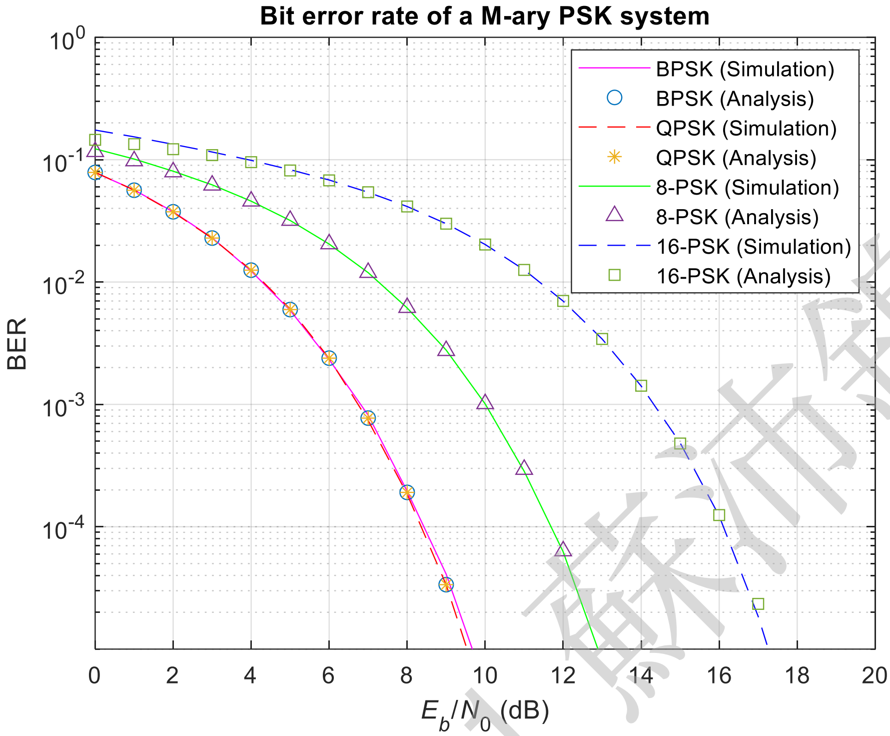
$$\therefore Q(x) \triangleq \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{y^2}{2}} dy \le \frac{1}{2} e^{-\frac{x^2}{2}} \le e^{-\frac{x^2}{2}}$$

$$\Longrightarrow Q(x) \le e^{-\frac{x^2}{2}} \quad \text{for } x > 0$$

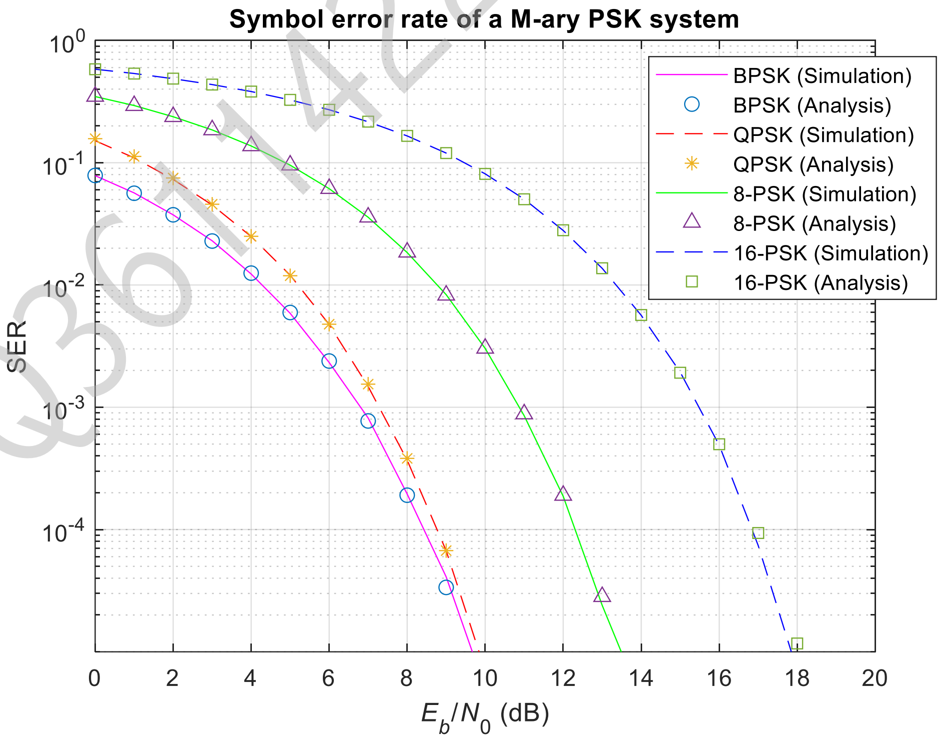
where $Q(x) \triangleq \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{t^2}{2}} dt$

3. Matlab Assignment ($M$-ary PSK)

(a) Bit error rate (BER) vs $E_b/N_0$ (dB)




(b) Symbol error rate (SER) vs $E_b/N_0$ (dB)



My Matlab codes:

```
% Problem # 03 (Matlab Assignment)
% Perform simulation and plot the bit error rate (BER) vs Eb/N0 (dB)
% and symbol error rate (SER) vs Eb/N0 (dB) of a M-ary PSK communication
% system (M = 2, 4, 8, 16) over AWGN channel.
% Written by P.-J. Su 2022/9/27
% Run: To compile and run this computer program using MATLAB.

% Clear Command Window
clc
% Remove all variables from the current workspace,
% releasing them from system memory.
clear
% Close all figures whose handles are visible.
close all
% Start stopwatch timer.
tic

SNR_in_dB = 0:1:20;     % SNR (Eb/N0) in dB

 % BPSK
[BER_sim_1, BER_theo_1, SER_sim_1, SER_theo_1] = PSK(2, SNR_in_dB);
 % QPSK
[BER_sim_2, BER_theo_2, SER_sim_2, SER_theo_2] = PSK(4, SNR_in_dB);
 % 8-PSK
[BER_sim_3, BER_theo_3, SER_sim_3, SER_theo_3] = PSK(8, SNR_in_dB);
 % 16-PSK
[BER_sim_4, BER_theo_4, SER_sim_4, SER_theo_4] = PSK(16, SNR_in_dB);

% Plot the bit error rate (BER) curves
clf
semilogy(SNR_in_dB, BER_sim_1, '-m');
hold on
semilogy(SNR_in_dB, BER_theo_1,'o','Color',[0 0.447 0.741]);
hold on
semilogy(SNR_in_dB, BER_sim_2, '--r');
hold on
semilogy(SNR_in_dB, BER_theo_2,'*','Color',[0.9290 0.6940 0.1250]);
```

```
hold on
semilogy(SNR_in_dB, BER_sim_3, '-g');
hold on
semilogy(SNR_in_dB, BER_theo_3,'^','Color',[0.4940 0.1840 0.5560]);
hold on
semilogy(SNR_in_dB, BER_sim_4, '--b');
hold on
semilogy(SNR_in_dB, BER_theo_4,'square','Color',[0.4660 0.6740 0.1880]);
axis([0 20 1e-5 1]);
title('Bit error rate of a M-ary PSK system');
legend('BPSK (Simulation)','BPSK (Analysis)', ...
       'QPSK (Simulation)','QPSK (Analysis)', ...
       '8-PSK (Simulation)','8-PSK (Analysis)', ...
       '16-PSK (Simulation)','16-PSK (Analysis)');
xlabel('\itE_{b}\rm/\itN\rm_{0} (dB)');
ylabel('BER');
grid on

% Plot the symbol error rate (SER) curves
figure
semilogy(SNR_in_dB, SER_sim_1, '-m');
hold on
semilogy(SNR_in_dB, SER_theo_1,'o','Color',[0 0.447 0.741]);
hold on
semilogy(SNR_in_dB, SER_sim_2, '--r');
hold on
semilogy(SNR_in_dB, SER_theo_2,'*','Color',[0.9290 0.6940 0.1250]);
hold on
semilogy(SNR_in_dB, SER_sim_3, '-g');
hold on
semilogy(SNR_in_dB, SER_theo_3,'^','Color',[0.4940 0.1840 0.5560]);
hold on
semilogy(SNR_in_dB, SER_sim_4, '--b');
hold on
semilogy(SNR_in_dB, SER_theo_4,'square','Color',[0.4660 0.6740 0.1880]);
axis([0 20 1e-5 1]);
title('Symbol error rate of a M-ary PSK system');
legend('BPSK (Simulation)','BPSK (Analysis)', ...
```

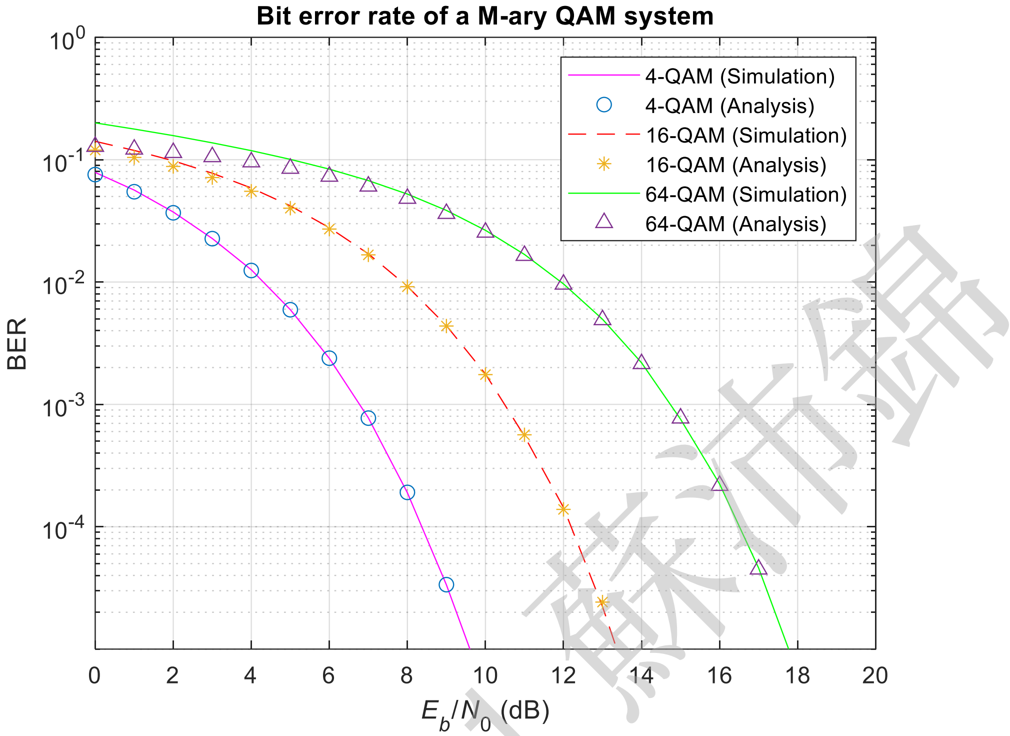
```
        'QPSK (Simulation)','QPSK (Analysis)', ...
        '8-PSK (Simulation)','8-PSK (Analysis)', ...
        '16-PSK (Simulation)','16-PSK (Analysis)');
xlabel('\itE_{b}\rm/\itN\rm_{0} (dB)');
ylabel('SER');
grid on

% Read elapsed time from stopwatch.
toc

function [BER_sim, BER_theo, SER_sim, SER_theo] = PSK(M, SNR_in_dB)
    N = 10^6;    % Number of data symbol
    k = log2(M);    % M = 2^k
    Es = 1;     % Energy per symbol
    SNR = 10.^(SNR_in_dB/10);     % SNR (Eb/N0)
    dmin = 2*sqrt(Es)*sin(pi/M);
    num_of_symbol_error = zeros(size(SNR_in_dB));    % Number of symbol error
    num_of_bit_error = zeros(size(SNR_in_dB));    % Number of bit error
    SER_theo = zeros(size(SNR_in_dB));
    BER_theo = zeros(size(SNR_in_dB));

    % Generation of transmitted bits based on the data source (Gray code)
    QPSK_Gray_code = [0 0; 0 1; 1 1; 1 0];
    PSK_8_Gray_code = [0 0 0; 0 0 1; 0 1 1; 0 1 0; ...
                      1 1 0; 1 1 1; 1 0 1; 1 0 0];
    PSK_16_Gray_code = [0 0 0 0; 0 0 0 1; 0 0 1 1; 0 0 1 0; ...
                       0 1 1 0; 0 1 1 1; 0 1 0 1; 0 1 0 0; ...
                       1 1 0 0; 1 1 0 1; 1 1 1 1; 1 1 1 0; ...
                       1 0 1 0; 1 0 1 1; 1 0 0 1; 1 0 0 0];

    % Mapping to the signal constellation
    PSK_mapping = zeros(M,2);
    for i = 1:M
        PSK_mapping(i,:) = [sqrt(Es)*cos((2*i-1)*pi/M) ...
                            sqrt(Es)*sin((2*i-1)*pi/M)];
    end

    for snr_Idx = 1:length(SNR_in_dB)
```

```
data_source = zeros(N,1);   % Data source
tx_signal = zeros(N,2);     % Transmitted symbols
rx_signal = zeros(N,2);     % Received symbols
tx_bit = zeros(N*k,1);      % Transmitted bits
rx_bit = zeros(N*k,1);      % Received bits

for i = 1:N
    % Generation of the data source
    data_source(i) = floor(M*rand) + 1;

    % The transmitted signal at the M-ary PSK mapper
    tx_signal(i,:) = PSK_mapping(data_source(i),:);

    % AWGN channel
    sigma = sqrt(Es/(2*k*SNR(snr_Idx)));
    noise = randn(1,2)*sigma;

    % The received signal at the detector
    rx_signal(i,:) = tx_signal(i,:) + noise;

    % Metric computation
    metrics = zeros(M,1);
    % Detection and symbol error calculation
    for j = 1:M
        metrics(j) = (rx_signal(i,1)-PSK_mapping(j,1))^2 + ...
                     (rx_signal(i,2)-PSK_mapping(j,2))^2;
    end
    [~, decision] = min(metrics);
    if (decision ~= data_source(i))
        num_of_symbol_error(snr_Idx) = num_of_symbol_error(snr_Idx)+1;
    end

     % Generation of transmitted bits and received bits
    for l = 1:k
        switch M
            case 2
                tx_bit(i) = data_source(i) - 1;
                rx_bit(i) = decision - 1;
```

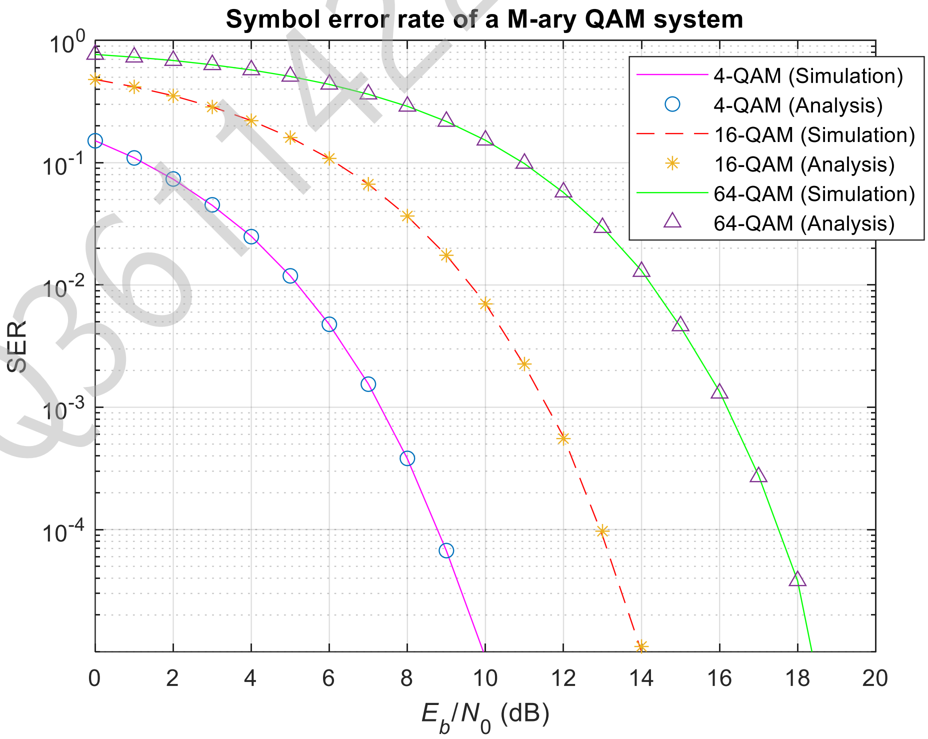
```
                case 4
                    tx_bit((i-1)*k + l) = QPSK_Gray_code(data_source(i),l);
                    rx_bit((i-1)*k + l) = QPSK_Gray_code(decision,l);
                case 8
                    tx_bit((i-1)*k + l) = PSK_8_Gray_code(data_source(i),l);
                    rx_bit((i-1)*k + l) = PSK_8_Gray_code(decision,l);
                case 16
                    tx_bit((i-1)*k + l) = PSK_16_Gray_code(data_source(i),l);
                    rx_bit((i-1)*k + l) = PSK_16_Gray_code(decision,l);
            end
        end
    end

    % Bit error calculation
    err_pat = xor(tx_bit, rx_bit);      % compare tx and rx bits
    num_of_bit_error(snr_Idx) = num_of_bit_error(snr_Idx) + sum(err_pat);

    % Error probability calculation (Analysis)
    switch M
        case 2
            SER_theo(snr_Idx) = qfunc(sqrt(2*SNR(snr_Idx)));    % Theoretical SER
            BER_theo(snr_Idx) = SER_theo(snr_Idx)/k;    % Theoretical BER
        case {4, 8, 16}
            SER_theo(snr_Idx) = 2*qfunc(dmin/(2*sigma));    % Theoretical SER
            BER_theo(snr_Idx) = SER_theo(snr_Idx)/k;    % Theoretical BER
    end
end

% Error probability calculation (Simulation)
SER_sim = num_of_symbol_error/(N);   % Simulated SER
BER_sim = num_of_bit_error/(N*k);    % Simulated BER
end
```

4. Matlab Assignment ($M$-ary QAM)

(a) Bit error rate (BER) vs $E_b/N_0$ (dB)




(b) Symbol error rate (SER) vs $E_b/N_0$ (dB)



My Matlab codes:

```
% Problem # 04 (Matlab Assignment)
% Perform simulation and plot the bit error rate (BER) vs Eb/N0 (dB)
% and symbol error rate (SER) vs Eb/N0 (dB) of a M-ary QAM communication
% system (M = 4, 16, 64) over AWGN channel.
% Written by P.-J. Su 2022/9/27
% Run: To compile and run this computer program using MATLAB.

% Clear Command Window
clc
% Remove all variables from the current workspace,
% releasing them from system memory.
clear
% Close all figures whose handles are visible.
close all
% Start stopwatch timer.
tic

SNR_in_dB = 0:1:20;     % SNR (Eb/N0) in dB

 % 4-QAM
[BER_sim_1, BER_theo_1, SER_sim_1, SER_theo_1] = QAM(4, SNR_in_dB);
 % 16-QAM
[BER_sim_2, BER_theo_2, SER_sim_2, SER_theo_2] = QAM(16, SNR_in_dB);
 % 64-QAM
[BER_sim_3, BER_theo_3, SER_sim_3, SER_theo_3] = QAM(64, SNR_in_dB);

% Plot the bit error rate (BER) curves
clf
semilogy(SNR_in_dB, BER_sim_1, '-m');
hold on
semilogy(SNR_in_dB, BER_theo_1,'o','Color',[0 0.447 0.741]);
hold on
semilogy(SNR_in_dB, BER_sim_2, '--r');
hold on
semilogy(SNR_in_dB, BER_theo_2,'*','Color',[0.9290 0.6940 0.1250]);
hold on
semilogy(SNR_in_dB, BER_sim_3, '-g');
```

```
hold on
semilogy(SNR_in_dB, BER_theo_3,'^','Color',[0.4940 0.1840 0.5560]);
axis([0 20 1e-5 1]);
title('Bit error rate of a M-ary QAM system');
legend('4-QAM (Simulation)','4-QAM (Analysis)', ...
       '16-QAM (Simulation)','16-QAM (Analysis)', ...
       '64-QAM (Simulation)','64-QAM (Analysis)');
xlabel('\itE_{b}\rm/\itN\rm_{0} (dB)');
ylabel('BER');
grid on

% Plot the symbol error rate (SER) curves
figure
semilogy(SNR_in_dB, SER_sim_1, '-m');
hold on
semilogy(SNR_in_dB, SER_theo_1,'o','Color',[0 0.447 0.741]);
hold on
semilogy(SNR_in_dB, SER_sim_2, '--r');
hold on
semilogy(SNR_in_dB, SER_theo_2,'*','Color',[0.9290 0.6940 0.1250]);
hold on
semilogy(SNR_in_dB, SER_sim_3, '-g');
hold on
semilogy(SNR_in_dB, SER_theo_3,'^','Color',[0.4940 0.1840 0.5560]);
axis([0 20 1e-5 1]);
title('Symbol error rate of a M-ary QAM system');
legend('4-QAM (Simulation)','4-QAM (Analysis)', ...
       '16-QAM (Simulation)','16-QAM (Analysis)', ...
       '64-QAM (Simulation)','64-QAM (Analysis)');
xlabel('\itE_{b}\rm/\itN\rm_{0} (dB)');
ylabel('SER');
grid on

% Read elapsed time from stopwatch.
toc

function [BER_sim, BER_theo, SER_sim, SER_theo] = QAM(M, SNR_in_dB)
    N = 10^6;    % Number of data symbol
```

```
k = log2(M);    % M = 2^k
SNR = 10.^(SNR_in_dB/10);     % SNR (Eb/N0)
dmin = 1;      % Minimum distance between symbol
Es = 2*(M-1)*(dmin^2)/3;     % Energy per symbol
num_of_symbol_error = zeros(size(SNR_in_dB));    % Number of symbol error
num_of_bit_error = zeros(size(SNR_in_dB));    % Number of bit error
SER_theo = zeros(size(SNR_in_dB));
BER_theo = zeros(size(SNR_in_dB));

% Generation of transmitted bits based on the data source (Gray code)
QAM_4_Gray_code = [0 1; 1 1; 0 0; 1 0];
QAM_16_Gray_code = [0 0 1 0; 0 1 1 0; 1 1 1 0; 1 0 1 0; ...
                  0 0 1 1; 0 1 1 1; 1 1 1 1; 1 0 1 1; ...
                  0 0 0 1; 0 1 0 1; 1 1 0 1; 1 0 0 1; ...
                  0 0 0 0; 0 1 0 0; 1 1 0 0; 1 0 0 0];
QAM_64_Gray_code = [0 0 0 1 0 0; 0 0 1 1 0 0; 0 1 1 1 0 0; 0 1 0 1 0 0; ...
                  1 1 0 1 0 0; 1 1 1 1 0 0; 1 0 1 1 0 0; 1 0 0 1 0 0; ...
                  0 0 0 1 0 1; 0 0 1 1 0 1; 0 1 1 1 0 1; 0 1 0 1 0 1; ...
                  1 1 0 1 0 1; 1 1 1 1 0 1; 1 0 1 1 0 1; 1 0 0 1 0 1; ...
                  0 0 0 1 1 1; 0 0 1 1 1 1; 0 1 1 1 1 1; 0 1 0 1 1 1; ...
                  1 1 0 1 1 1; 1 1 1 1 1 1; 1 0 1 1 1 1; 1 0 0 1 1 1; ...
                  0 0 0 1 1 0; 0 0 1 1 1 0; 0 1 1 1 1 0; 0 1 0 1 1 0; ...
                  1 1 0 1 1 0; 1 1 1 1 1 0; 1 0 1 1 1 0; 1 0 0 1 1 0; ...
                  0 0 0 0 1 0; 0 0 1 0 1 0; 0 1 1 0 1 0; 0 1 0 0 1 0; ...
                  1 1 0 0 1 0; 1 1 1 0 1 0; 1 0 1 0 1 0; 1 0 0 0 1 0; ...
                  0 0 0 0 1 1; 0 0 1 0 1 1; 0 1 1 0 1 1; 0 1 0 0 1 1; ...
                  1 1 0 0 1 1; 1 1 1 0 1 1; 1 0 1 0 1 1; 1 0 0 0 1 1; ...
                  0 0 0 0 0 1; 0 0 1 0 0 1; 0 1 1 0 0 1; 0 1 0 0 0 1; ...
                  1 1 0 0 0 1; 1 1 1 0 0 1; 1 0 1 0 0 1; 1 0 0 0 0 1; ...
                  0 0 0 0 0 0; 0 0 1 0 0 0; 0 1 1 0 0 0; 0 1 0 0 0 0; ...
                  1 1 0 0 0 0; 1 1 1 0 0 0; 1 0 1 0 0 0; 1 0 0 0 0 0];

% Mapping to the signal constellation
QAM_mapping = zeros(M,2);
QAM_mapping_Idx = 1;
for i = 1:sqrt(M)
    for j = 1:sqrt(M)
        QAM_mapping(QAM_mapping_Idx,1) = -(sqrt(M)-1) + 2*(j-1);
```

```
        QAM_mapping(QAM_mapping_Idx,2) = (sqrt(M)-1) - 2*(i-1);
        QAM_mapping_Idx = QAM_mapping_Idx + 1;
    end
end

for snr_Idx = 1:length(SNR_in_dB)
    data_source = zeros(N,1);   % Data source
    tx_signal = zeros(N,2);     % Transmitted symbols
    rx_signal = zeros(N,2);     % Received symbols
    tx_bit = zeros(N*k,1);      % Transmitted bits
    rx_bit = zeros(N*k,1);      % Received bits

    for i = 1:N
        % Generation of the data source
        data_source(i) = floor(M*rand) + 1;

        % The transmitted signal at the M-ary PSK mapper
        tx_signal(i,:) = QAM_mapping(data_source(i),:);

        % AWGN channel
        sigma = sqrt(Es/(2*k*SNR(snr_Idx)));
        noise = randn(1,2)*sigma;

        % The received signal at the detector
        rx_signal(i,:) = tx_signal(i,:) + noise;

        % Metric computation
        metrics = zeros(M,1);
        % Detection and symbol error calculation
        for j = 1:M
            metrics(j) = (rx_signal(i,1)-QAM_mapping(j,1))^2 + ...
                         (rx_signal(i,2)-QAM_mapping(j,2))^2;
        end
        [~, decision] = min(metrics);
        if (decision ~= data_source(i))
            num_of_symbol_error(snr_Idx) = num_of_symbol_error(snr_Idx)+1;
        end
```

```
            % Generation of transmitted bits and received bits
            for l = 1:k
                switch M
                    case 4
                        tx_bit((i-1)*k + l) = QAM_4_Gray_code(data_source(i),l);
                        rx_bit((i-1)*k + l) = QAM_4_Gray_code(decision,l);
                    case 16
                        tx_bit((i-1)*k + l) = QAM_16_Gray_code(data_source(i),l);
                        rx_bit((i-1)*k + l) = QAM_16_Gray_code(decision,l);
                    case 64
                        tx_bit((i-1)*k + l) = QAM_64_Gray_code(data_source(i),l);
                        rx_bit((i-1)*k + l) = QAM_64_Gray_code(decision,l);
                end
            end
        end

        % Bit error calculation (Simulation)
        err_pat = xor(tx_bit, rx_bit);      % compare tx and rx bits
        num_of_bit_error(snr_Idx) = num_of_bit_error(snr_Idx) + sum(err_pat);

        % Error probability calculation (Analysis)
        Q = qfunc(dmin/sigma);
        const = 1/sqrt(M) - 1;
        SER_theo(snr_Idx) = -4*const*Q*(1 + const*Q);    % Theoretical SER
        BER_theo(snr_Idx) = SER_theo(snr_Idx)/k;     % Theoretical BER
    end

    % Error probability calculation (Simulation)
    SER_sim = num_of_symbol_error/(N);   % Simulated SER
    BER_sim = num_of_bit_error/(N*k);    % Simulated BER
end
```

**References**

[1] J. G. Proakis and M. Salehi, *Digital Communications*, 5th ed. Singapore: McGraw-Hill, Inc., 2008.

[2] J. G. Proakis, M. Salehi, and G. Bauch, *Contemporary Communication Systems Using MATLAB*, 3rd ed. USA: Cengage Learning, 2013.