Aristotle University of Thessaloniki
Department of Electrical and Computer Engineering
Telecommunications Division

Ανατολή Μανωλίδου 10874
Σταύρος Σπυριδόπουλος 10845

Telecommunication Systems II

Research Project

Contents

1 Problem 1

1.1 Theoretical Analysis

Euclidean Distance

Examples of reference [1],[2].
Minimum Euclidean Distance, commonly used as d_{min}, is the minimum distance between all pairs of symbols in a constellation. The bigger the euclidean distance, the better are the chances of the receiver guessing correctly the signal that the transmitter sent.

1. ***M-PAM***
   In a Pulse-Amplitude Modulation (PAM) constellation, the Euclidean distance between symbols is:
   $$d_{s_i,s_j} = \sqrt{||s_i - s_j||^2} = 2 \cdot ||i - j|| \cdot \sqrt{E_g} \tag{1}$$
   while the minimum Euclidean distance between points of a PAM constellation is given as :
   $$d_{min} = \sqrt{||s_i - s_{i\pm1}||^2} = 2 \cdot \sqrt{E_g} \qquad for\ \ 1 \leq i \leq M \tag{2}$$
   where E_g is the average energy per pulse.
   Because :
   $$E_g = \frac{3 \cdot E_s}{M^2 - 1} \tag{3}$$
   the above equation can be written as :
   $$(2) \Rightarrow d_{min} = 2 \cdot \sqrt{E_s} \cdot \sqrt{\frac{3}{M^2 - 1}} \tag{4}$$
   where E_s is the average symbol energy of the constellation.

2. ***M-PSK***
   In a Phase Shift Keying Modulation (PSK) constellation, the Euclidean distance between symbols is :
   $$d_{s_i,s_j} = \sqrt{||s_i - s_j||^2} = \sqrt{2 \cdot E_s \cdot (1 - \cos\frac{2 \cdot \pi}{M} \cdot (i - j))} \tag{5}$$
   while the minimum Euclidean distance between points of a PSK constellation is given as :
   $$d_{min} = \sqrt{2 \cdot E_s \cdot (1 - \cos\frac{2 \cdot \pi}{M})} = 2 \cdot \sqrt{E_s} \cdot \sin\frac{\pi}{M} \tag{6}$$
   where Es is the average symbol energy of the constellation.

3. ***M-QAM***
   In a Quadrature Amplitude Modulation (QAM) constellation, the minimum Euclidean distance between points of a M-QAM constellation is given as :
   $$d_{min} = 2 \cdot \sqrt{E_g} \tag{7}$$
   where E_g is the average energy per pulse.
   Because (3) is valid in this kind of constellation too, the minimum Euclidean distance is given as :
   $$(7) \Rightarrow d_{min} = \sqrt{\frac{6 \cdot E_s}{M - 1}} \tag{8}$$

4. ***M-CQAM***
   In a Circular Quadrature Amplitude Modulation (QAM) constellation, the minimum Euclidean distance between points of a M-CQAM constellation is given as :

$$d_{min} = \sqrt{R_2^2 + R_1^2 - 2R_2R_1\cos(\theta_2 - \theta_1)} \tag{9}$$

   Because in our case we utilize a constellation with equal distances and 45° shift between each circle layer we have the following equation :

$$(9) \Rightarrow d_{min} = \sqrt{5R^2 - 4R^2 \cdot \cos 45^\circ} \Rightarrow d_{min} = R \cdot \sqrt{5 - 2\sqrt{2}} \tag{10}$$

   For outer layers the factor $f = \sqrt{5 - 2\sqrt{2}}$ become larger so the distances between symbols follow the same behaviour.
   As mentioned above, due to the parameters for the CQAM design :

$$E_s = \frac{4}{M}\sum_{i=1}^{M/4} R_i^2 = \frac{4}{M}\sum_{i=1}^{M/4}(i \cdot R)^2 = \frac{4}{M} \cdot R^2 \cdot \sum_{i=1}^{M/4} i^2 \Rightarrow R = \sqrt{\frac{E_s \cdot M}{4 \cdot \sum_{i=1}^{M/4} i^2}} \tag{11}$$

$$(10) \Rightarrow d_{min} = \sqrt{\frac{E_s \cdot M}{4 \cdot \sum_{i=1}^{M/4} i^2} \cdot (5 - 2\sqrt{2})} \tag{12}$$

When calculating the above formulas on constellations of $M = 16$ and $E_s = 1$ we get :

- $d_{min,16-PAM} = 2 \cdot \sqrt{1} \cdot \sqrt{\frac{3}{16^2-1}} = 0.2169$
- $d_{min,16-PSK} = 2 \cdot \sqrt{1} \cdot \sin\frac{\pi}{16} = 0.3902$
- $d_{min,16-QAM} = \sqrt{\frac{6 \cdot E_s}{M-1}} = 0.6324$
- $d_{min,16-QAM} = \sqrt{\frac{E_s \cdot 16}{4 \cdot \sum_{i=1}^{M/4} i^2} \cdot (5 - 2\sqrt{2})} = 0.5381$

It shows that a 16-QAM constellation has the largest minimum Euclidean Distance (d_{min}) among the 3 constellations we analyzed. This condition indicates that the symbols of a 16-QAM constellation are spaced further apart than symbols in the other 2 modulation schemes. This provides more resistance to noise and interference errors when utilized.

Peak-to-Average Power Ratio (PAPR)

Peak-to-Average Power Ratio (PAPR) measures the ratio of the peak power to the average power of a signal. High PAPR indicates large power spikes, which can cause inefficiencies and distortions.

$$PAPR = \frac{Peak\ Power}{Average\ Power} \tag{13}$$

In our use-case (13) can be expressed as :

$$PAPR = \frac{Peak\ Constellation\ Energy}{Average\ Constellation\ Energy} = \frac{E_{max}}{E_s} \tag{14}$$

1. ***M-PAM***
   In a M-PAM constellation, E_{max} can be written as :

$$E_{max} = \frac{3 \cdot E_s \cdot (M-1)^2}{M^2 - 1} \tag{15}$$

Resulting in a equation like this for calculating the PAPR :

$$(14),(15) \Rightarrow PAPR_{PAM} = \frac{3\cdot(M-1)^2}{M^2-1} \tag{16}$$

2. ***M-PSK***
In a M-PAM constellation, every signal has the same distance from point (0,0) hence $E_{max} = E_s$.
This results in :

$$PAPR_{PSK} = \frac{E_{max,PSK}}{E_{s,PSK}} = 1 \tag{17}$$

no matter what is the average symbol or pulse function energy of the constellation.

3. ***M-QAM***
In a M-QAM constellation, E_{max} can be written as :

$$E_{\max} = 2\left(\sqrt{M}-1\right)^2 \tag{18}$$

Resulting in a equation like this for calculating the PAPR :

$$(14),(18) \Rightarrow PAPR_{QAM} = \frac{E_{max,QAM}}{E_{s,QAM}} = \frac{2\left(\sqrt{M}-1\right)^2}{\frac{2(M-1)}{3}} = \frac{3\cdot\left(\sqrt{M}-1\right)^2}{M-1} \tag{19}$$

4. ***M-CQAM***
In a M-CQAM constellation, E_{max} can be written as :

$$E_{\max} = R_{M/4}^2 \tag{20}$$

where M/4 represents the number of circle layers of the constellation and $R_{M/4}$ is the radius of the outer circle layer. Resulting in a equation like this for calculating the PAPR :

$$(11),(20) \Rightarrow PAPR_{CQAM} = \frac{E_{max,CQAM}}{E_{s,CQAM}} = \frac{R_{M/4}^2}{\frac{4}{M}\sum_{i=1}^{M/4} R_i^2} = \frac{(\frac{M}{4}R)^2}{\frac{4}{M}\cdot R^2\cdot\sum_{i=1}^{M/4} i^2} \tag{21}$$

$$\Rightarrow PAPR_{CQAM} = \frac{M^3}{4^3\cdot\sum_{i=1}^{M/4} i^2} \tag{22}$$

When calculating the above formulas on constellations of $M = 16$ we get :

- $PAPR_{16-PAM} = \frac{3\cdot(16-1)^2}{16^2-1} = 2.6471$
- $PAPR_{16-PSK} = 1$
- $PAPR_{16-QAM} = \frac{3\cdot\left(\sqrt{16}-1\right)^2}{16-1} = 1.8$
- $PAPR_{16-CQAM} = \frac{16^3}{4^3\cdot\sum_{i=1}^{4} i^2} = \frac{32}{15} = 2.1333$

It is clear that a 16-PSK constellation is more effective that the other 2 regarding PAPR. This results to increases of transmission range and capacity, improving overall system performance and spectrum efficiency.

1.2 Simulation Results

In this subsection the simulation results can be found. For checking the MATLAB code used to create the graphs below, the reader can take a detour on Section (3.1)

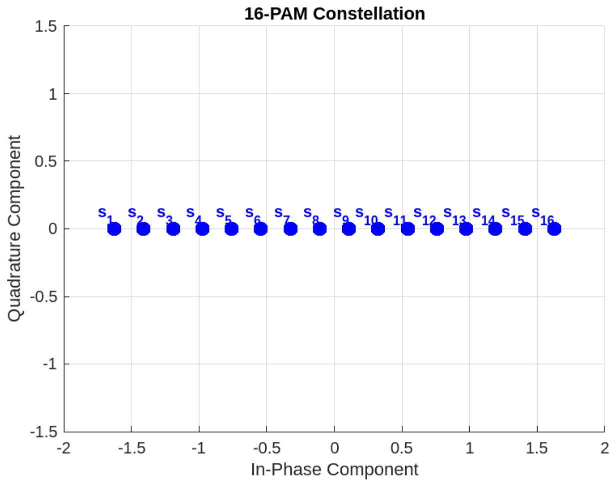

Figure 1: 16-PAM constellation

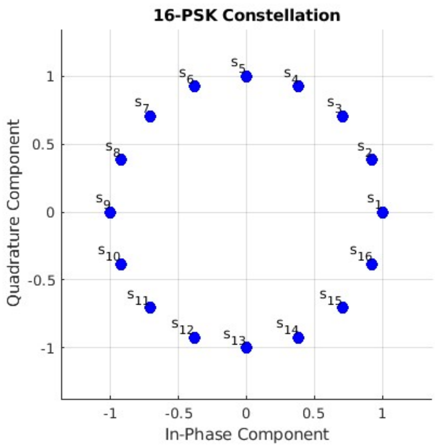

Figure 2: 16-PSK constellation

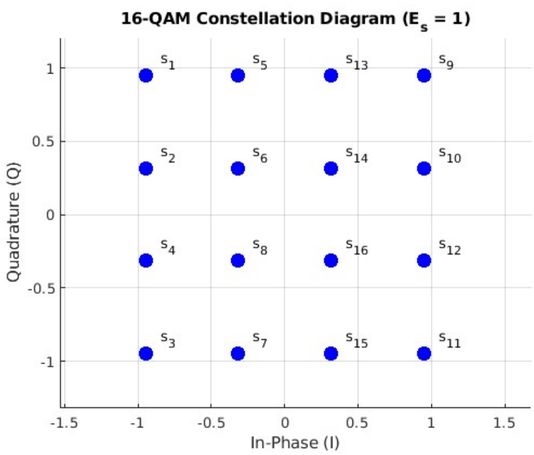

Figure 3: 16-QAM constellation

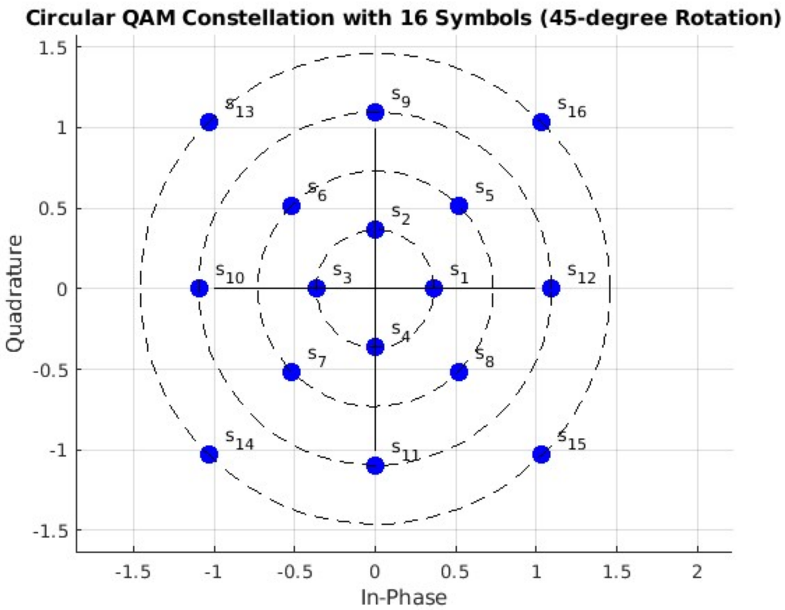

Figure 4: 16-QAM constellation

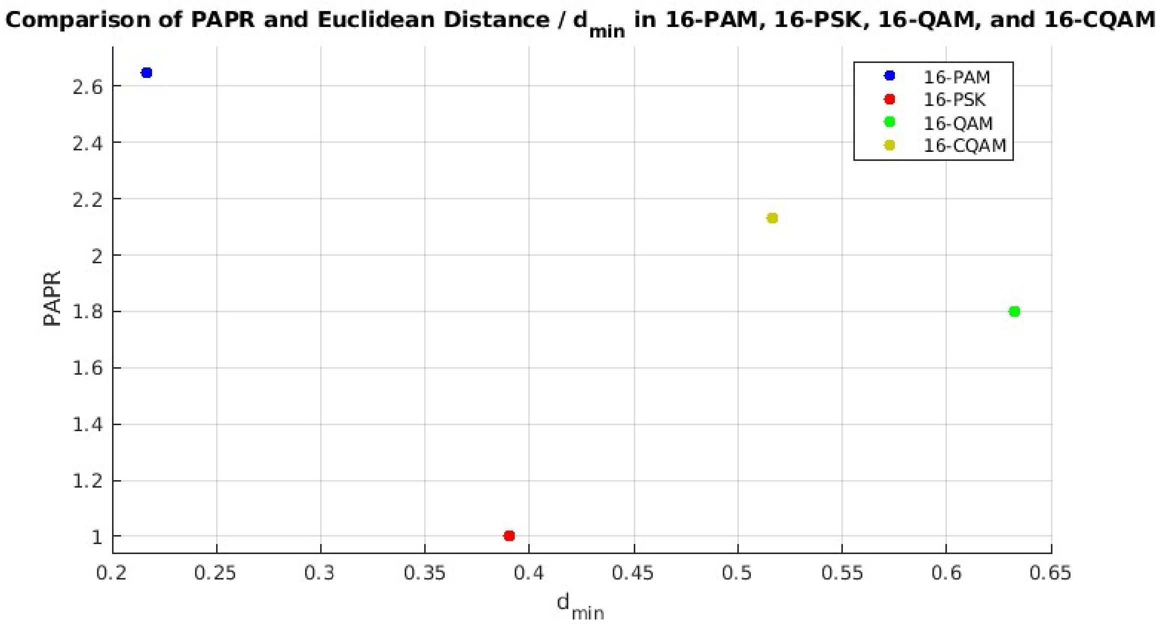

Figure 5: PAPR vs d_{min}

2 Problem 2

2.1 Theoretical Analysis

Symbol Error Probability (SEP)

Examples of reference [1],[3].
Symbol Error Probability is an important measure in constellation telecommunications systems. It measures the likelihood of signals being misinterpreted during transmission due to noise and interference.

1. ***M-PAM***
   The formula used to calculate the symbol error probability of a M-PAM constellation for equiprobable symbols is :

$$\begin{aligned} P_{s(M-PAM)} &= \frac{2(M-1)}{M} \cdot Q(\sqrt{\frac{2E_g}{No}}) \\ (3) \Rightarrow P_{s(M-PAM)} &= \frac{2(M-1)}{M} \cdot Q(\sqrt{\frac{6E_s}{(M^2-1) \cdot No}} \end{aligned} \tag{23}$$

2. ***M-PSK***
   The formula used to calculate the SEP of a M-PSK constellation for equiprobable symbols is :

$$P_{s(M-PSK)} \approx 2Q(\sqrt{\frac{2E_s}{No}} \cdot (\sin\frac{\pi}{M})) \tag{24}$$

   Since all of the symbols lie in the same circle and have equal energy.

3. ***M-QAM***
   The formula used to calculate the SEP of a M-PAM constellation for equiprobable symbols is :

$$P_{s,\sqrt{M}} = 2(1 - \frac{1}{\sqrt{M}})Q(\sqrt{\frac{3}{M-1}\frac{E_s}{No}}) \tag{25}$$

   Where $P_{\sqrt{M}}$ is the SEP of a band-pass $\sqrt{M}$-PAM signal. Therefore, this is the formula utilized on the calculations :

$$P_{s(M-QAM)} = 1 - (1 - P_{\sqrt{M}})^2 \tag{26}$$

4. ***M-CQAM***
   Ways to calculate the SEP of a CQAM constellation can be found in [3] and get utilized using the formula below :

$$d_n = \sqrt{(5 + 2n \cdot (n+1)) - (2 + n \cdot (n-1)) \cdot \sqrt{2}} \tag{27}$$

   where d_i is the distance between neighboring symbols of circle layers i and $i+1$.

Signal to Noise ratio (SNR)

Examples of reference [1],[4].
The Signal to Noise Ratio is widely accepted as the easiest to understand evaluation criteria of a telecommunication system. It is defined as the ratio of the received signal's power to the noise power introduced by the receiver.

$$SNR = \frac{P_r}{P_N} \tag{28}$$

As mentioned in [4], we can represent power spectral density of the noise (N_0) as :

$$N_0 = P_N / B \tag{29}$$

This leads to this SNR expression :

$$SNR = \frac{E_s}{N_0} \tag{30}$$

1. ***M-PAM***
   For M-PAM constellations :

$$E_s = \frac{1}{M} \sum_{i=1}^{M} E_{s_i} \tag{31}$$

   So SNR is :

$$(30), (31) \Rightarrow SNR_{M-PAM} = \frac{1}{M \cdot N_0} \sum_{i=1}^{M} E_{s_i} \tag{32}$$

2. ***M-PSK***
   For M-PSK constellations, the Average Symbol Energy is :

$$E_s = \frac{1}{M} \sum_{i=1}^{M} R^2 = R^2 \tag{33}$$

   Because all symbols lie on the same circle of radius R. So SNR is eqaul to :

$$(30), (33) \Rightarrow SNR_{M-PSK} = \frac{R^2}{N_0} \tag{34}$$

3. ***M-QAM***
   Same as in M-PAM constellations (31), the average energy of the symbols (E_s) is equal to:

$$E_s = \frac{1}{M} \sum_{i=1}^{M} E_{s_i}$$

   Therefore,

$$(30), (31) \Rightarrow SNR_{M-QAM} = \frac{1}{M \cdot N_0} \sum_{i=1}^{M} E_{s_i}$$

4. ***M-CQAM***
   As mentioned in (11), the average symbol energy of a circular M-QAM constellation is equal to :

$$E_s = \frac{4}{M} \sum_{i=1}^{M/4} R_i^2 = \frac{4}{M} \cdot R^2 \cdot \sum_{i=1}^{M/4} i^2 \tag{35}$$

   For that reason :

$$SNR_{M-CQAM} = \frac{4}{M \cdot N_0} \cdot R^2 \cdot \sum_{i=1}^{M/4} i^2 \tag{36}$$

So, when $M = 16$, $E_s = 1$, $N_0 = 0,1$ and $SNR = \frac{1}{0.1} = 10$ is assumed :

- $P_{s(16-PAM)} = \dfrac{2(16-1)}{16} \cdot Q(\sqrt{\dfrac{6 \cdot 1}{(16^2-1) \cdot 0.1}}) = \dfrac{15}{8} \cdot Q(\dfrac{0.6}{255})$

  $\Rightarrow P_{s(M-PAM)} \approx 0,91$

- $P_{s(16-PSK)} \approx 2Q(\sqrt{\dfrac{2}{0.1}}) = 2Q(4.47) = 3.9 \cdot 10^{-6}$

- $P_{s,\sqrt{16}} = 2(1 - \dfrac{1}{\sqrt{16}})Q(\sqrt{\dfrac{3}{16-1}\dfrac{E_s}{0.1}}) = \dfrac{3}{2}Q(\sqrt{\dfrac{1}{0.3}})$

  $\Rightarrow P_{s(M-QAM)} = 1 - (1 - \dfrac{3}{2} \cdot Q(\sqrt{\dfrac{1}{3 \cdot 0.1}}))^2 \approx 1 - (1 - \dfrac{3}{2} \cdot 0.034)^2 = 0.099$

- [2] $\Rightarrow P_{s(16-CQAM)} = \dfrac{1}{M} \sum_{i=1}^{N} v(i) \cdot Q[\sqrt{SNR \cdot \sin \pi/n \cdot \sqrt{R_i^2 - \epsilon(R_i)}}]$

2.2 Simulation Results

In this subsection the simulation results can be found. For checking the MATLAB code used to create the graph below, the reader can take a detour on Section (3.2)

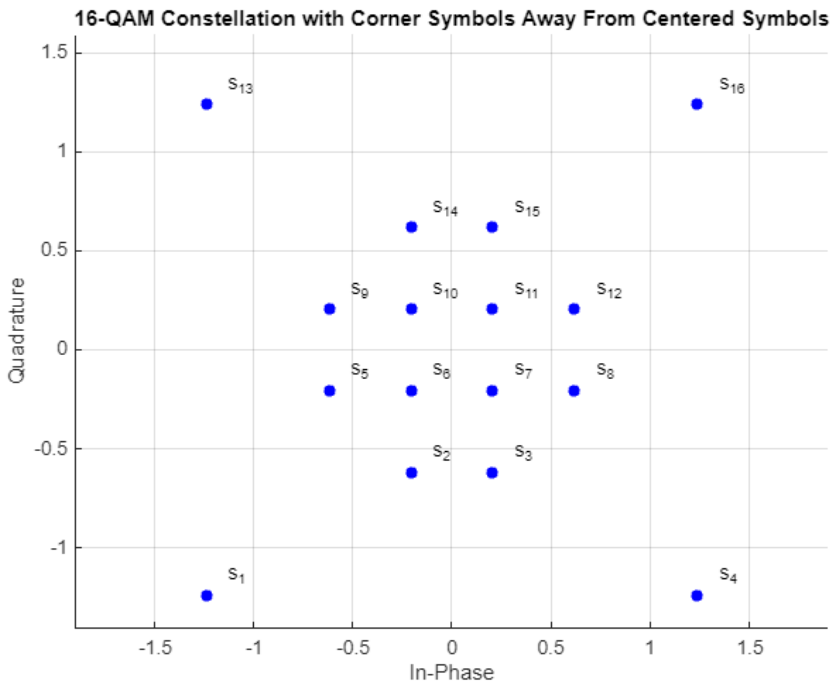

Figure 6: State-of-the-Art Constellation Graph from [5]

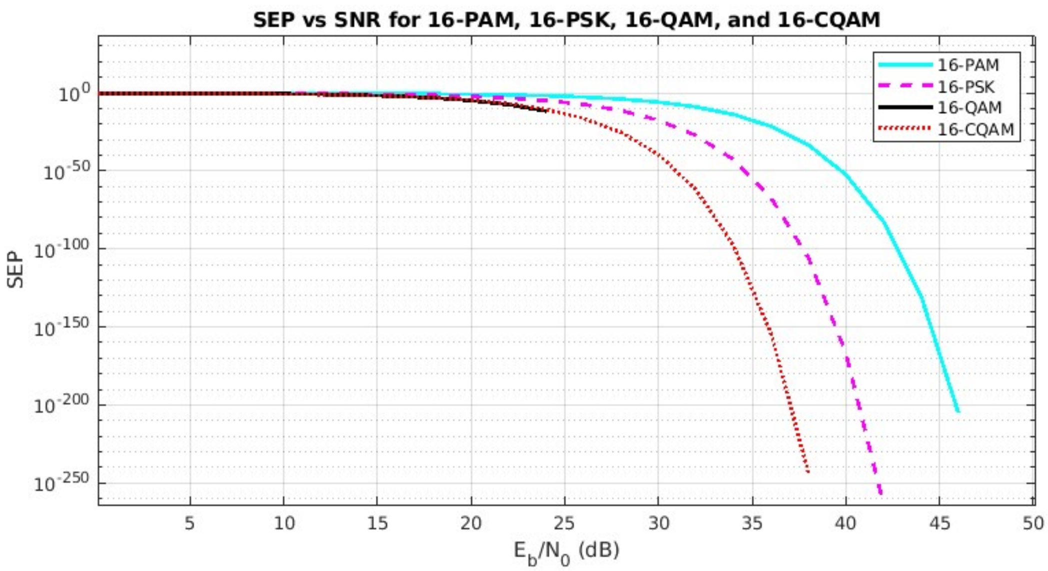

Figure 7: SEP vs SNR

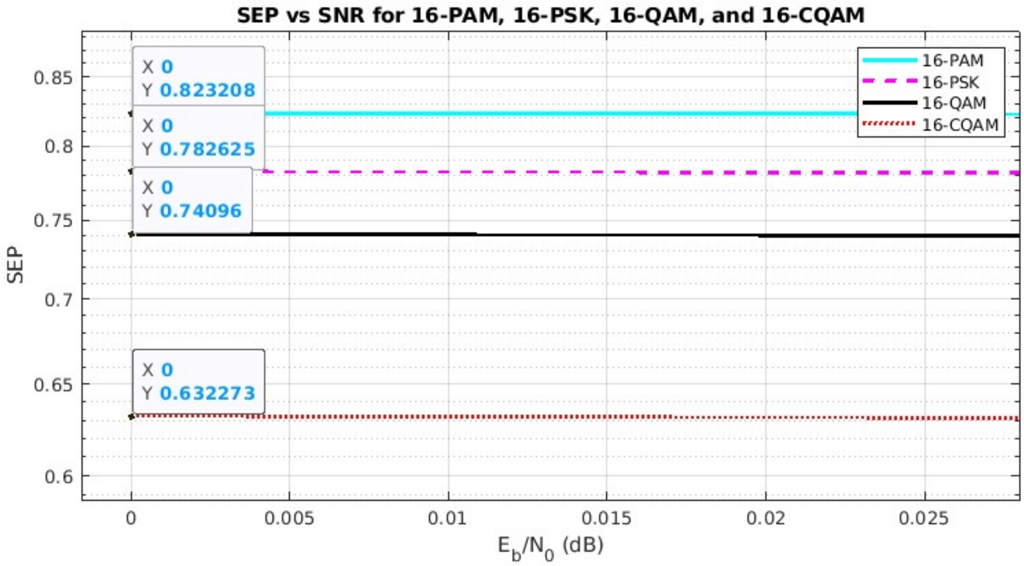

Figure 8: SEP vs SNR (start of graph)

3 Simulation Code

3.1 Problem 1 Code Sample

16-PAM

```
% Number of levels in PAM (M-PAM)
M = 16; % Example for 4-PAM

% Symbol Energy Es
Es = 1; % Set the desired symbol energy

% Generate M-PAM constellation points
% The PAM constellation points for M-PAM are centered around zero and spaced by 2.
constellation_points = -(M-1):2:(M-1);

% Calculate the average energy of the generated constellation
average_energy = mean(constellation_points.^2);

% Normalize the constellation to have the desired symbol energy Es
normalized_constellation = constellation_points * sqrt(Es / average_energy);

% Display the constellation points
disp('Normalized PAM Constellation Points:');
disp(normalized_constellation);

% Plot the constellation
figure;
stem(normalized_constellation, zeros(size(normalized_constellation)), 'filled');
title(sprintf('%d-PAM Constellation', M));
xlabel('In-Phase Component');
ylabel('Quadrature Component');
grid on;
axis([-M M -1 1]);

% Annotate each point with its symbol label
for i = 1:M
    % Label with symbol's index
    label = sprintf('s_{%d}', i);
    text(normalized_constellation(i), 0, label, 'VerticalAlignment', 'bottom',
        'HorizontalAlignment', 'right', 'FontSize', 10, 'FontName', 'Helvetica');
end
```

16-PSK

```
% Number of levels in PSK (M-PSK)
M = 16; % Example for 4-PSK

% Symbol Energy Es
Es = 1; % Set the desired symbol energy

% Generate M-PSK constellation points
% The PSK constellation points are uniformly distributed on the unit circle.
theta = (0:M-1) * (2 * pi / M); % Angles for PSK points
constellation_points = exp(1i * theta); % Complex exponentials for PSK points

% Normalize the constellation to have the desired symbol energy Es
```

```
normalized_constellation = constellation_points * sqrt(Es);

% Display the constellation points
disp('Normalized PSK Constellation Points:');
disp(normalized_constellation);

% Plot the constellation
figure;
hold on; % Hold the plot for multiple plotting commands
title(sprintf('%d-PSK Constellation', M));
xlabel('In-Phase Component');
ylabel('Quadrature Component');
grid on;
axis equal;
axis([-1.5 1.5 -1.5 1.5]);

% Annotate each point with its symbol label
for i = 1:M
    % Plot each point as a dot
    plot(real(normalized_constellation(i)), imag(normalized_constellation(i)), 'o',
        'MarkerSize', 8, 'MarkerEdgeColor', 'b', 'MarkerFaceColor', 'b');
    % Label with only the symbol index
    label = sprintf('s_{%d}', i);
    text(real(normalized_constellation(i)), imag(normalized_constellation(i)), label,
        'VerticalAlignment', 'bottom', 'HorizontalAlignment', 'right', 'FontSize', 10,
        'FontName', 'Helvetica');
end

hold off; % Release the plot hold
```

16-QAM

```
% MATLAB script to plot the 16-QAM constellation diagram
% Define the 16-QAM modulation order
M = 16; % Modulation order for 16-QAM

% Generate the symbol indices
symbolIndices = 0:M-1;

% Generate 16-QAM constellation points using MATLAB's qammod function
constellation_points = qammod(symbolIndices, M, 'UnitAveragePower', true);

% Plot the constellation diagram
figure;
%plot(real(constellation_points), imag(constellation_points), 'bo', 'MarkerSize', 10,
    'LineWidth', 2);
scatter(real(constellation_points), imag(constellation_points), 100, 'b', 'filled');
title('16-QAM Constellation Diagram (E_s = 1)');
xlabel('In-Phase (I)');
ylabel('Quadrature (Q)');
grid on;
axis equal;

% Label the constellation points
for i = 1:length(constellation_points)
    label = sprintf('s_{%d}', i);
```

```
    text(real(constellation_points(i)) + 0.1, imag(constellation_points(i)) + 0.1,
        label);
end

% Display the plot
hold off;
```

16-CQAM

```
% Parameters
num_symbols_per_circle = 4;
num_circles = 4;

% Initial radii
initial_radii = [1, 2, 3, 4];

% Calculate the scaling factor
scaling_factor = sqrt(4 / sum(initial_radii .^ 2)); %0.3651

% Scale the radii
scaled_radii = initial_radii * scaling_factor;

% Generate constellation points
constellation_points = [];
for i = 1:num_circles
    radius = scaled_radii(i);
    angles = linspace(0, 2 * pi, num_symbols_per_circle + 1);
    angles(end) = []; % remove the last angle to avoid overlapping the first point
    rotated_angles = angles + (i - 1) * pi / 4; % Rotate each circle by 45 degrees
    points = radius * exp(1j * rotated_angles);
    constellation_points = [constellation_points; points.'];
end

% Plot the constellation
figure;
hold on;
scatter(real(constellation_points), imag(constellation_points), 100, 'b', 'filled');

% Plot dashed circles
theta = linspace(0, 2*pi, 100);
for i = 1:num_circles
    radius = scaled_radii(i);
    x = radius * cos(theta);
    y = radius * sin(theta);
    plot(x, y, '--k');
end

line([-1 1], [0 0], 'Color', 'k'); % x-axis
line([0 0], [-1 1], 'Color', 'k'); % y-axis
xlabel('In-Phase');
ylabel('Quadrature');
title('Circular QAM Constellation with 16 Symbols (45-degree Rotation)');
grid on;
axis equal;

for i = 1:length(constellation_points)
    label = sprintf('s_{%d}', i);
```

```
    text(real(constellation_points(i)) + 0.1, imag(constellation_points(i)) + 0.1,
        label);
end

hold off;

% Calculate dmin
distances = [];
for i = 1:length(constellation_points)
    for j = i+1:length(constellation_points)
        distances(end+1) = abs(constellation_points(i) - constellation_points(j));
    end
end
dmin = min(distances);

% Display dmin
disp(scaled_radii)
disp(['dmin = ', num2str(dmin)]);

sum_abs_values = sum(abs(constellation_points).^2);
Es = sum_abs_values / 16;
disp(['Sum of absolute values of each symbol = ', num2str(sum_abs_values), ', thus Es
    = ', num2str(Es)]);
```

3.2 Problem 2 Code Sample

State-of-the-Art Constellation Design

```
% Define 16-QAM constellation points
M = 16; % Number of symbols
k = log2(M); % Bits per symbol
d = 2; % Distance between adjacent symbols

% Define the standard 16-QAM constellation
real_part = repmat([-3 -1 1 3], 1, 4);
imag_part = repelem([-3 -1 1 3], 4);

% Combine real and imaginary parts
constellation = real_part + 1i * imag_part;

% Move the four corner symbols much further
constellation(1) = -6 - 6i; % Bottom left
constellation(4) = 6 - 6i; % Bottom right
constellation(13) = -6 + 6i; % Top left
constellation(16) = 6 + 6i; % Top right

% Calculate energy scaling factor to make average symbol energy Es = 1
avg_symbol_energy = mean(abs(constellation).^2);
scaling_factor = sqrt(1 / avg_symbol_energy);

% Scale the constellation points
constellation = constellation * scaling_factor;

% Plot the constellation
figure;
scatter(real(constellation), imag(constellation), 'filled', 'MarkerFaceColor', 'b');
title('16-QAM Constellation with Corner Symbols Away From Centered Symbols');
xlabel('In-Phase');
ylabel('Quadrature');
grid on;
axis equal;
ylim([-1.5 1.5]); % Set the y-axis limits from -1.5 to 1.5

% Annotate points
for i = 1:length(constellation)
    label = sprintf('s_{%d}', i);
    text(real(constellation(i)) + 0.1, imag(constellation(i)) + 0.1, label);
end

% Display average symbol energy
disp(['Average Symbol Energy (Es): ', num2str(mean(abs(constellation).^2))]);
```

Graph for SEP vs SNR

```
% Parameters
M = 16;
N = 4; % Number of circles for CQAM
SNR_dB = 0:2:60; % Extended SNR range to reach higher values
num_symbols = 1e5; % Total number of symbols for simulation

% Define the constellation points for 16-PAM, 16-PSK, and 16-QAM
constellation_pam = pammod(0:(M-1), M, 0, 'gray'); % 16-PAM
constellation_psk = pskmod(0:(M-1), M, 0, 'gray'); % 16-PSK
constellation_qam = qammod(0:(M-1), M, 'gray'); % 16-QAM
constellation_cqam = generate_16CQAM(M, N);

% Function to generate 16-CQAM constellation
function constellation = generate_16CQAM(M, N)
    n = M / N;
    R1 = 1; % Smallest radius
    constellation = [];
    for i = 1:N
        Ri = R1 * i; % Increasing radius for each circle
        base_angle = pi/4 * (i-1); % Rotate each circle by 45 degrees more than the
            previous one
        angles = (0:n-1) * (2 * pi / n) + base_angle;
        constellation = [constellation, Ri * exp(1i * angles)]; % Append points
    end
end

% Function to calculate theoretical SEP for 16-PAM
function sep = sep_pam(SNR)
    M = 16;
    sep = 2 * (M - 1) / M * qfunc(sqrt(6 * SNR / (M^2 - 1)));
end

% Function to calculate theoretical SEP for 16-PSK
function sep = sep_psk(SNR)
    M = 16;
    sep = 2 * qfunc(sqrt(2 * SNR) * sin(pi / M));
end

% Function to calculate theoretical SEP for 16-QAM
function sep = sep_qam(SNR)
    M = 16;
    Ps_M = 2 * (1 - 1/sqrt(M)) * qfunc(sqrt(3 / (M - 1) * SNR));
    sep = 1 - (1 - Ps_M)^2;
end

% Function to calculate theoretical SEP for 16-CQAM
function sep = sep_cqam(SNR)
    M = 16;
    R = 1; % Smallest radius
    N = 4;
    sum_sq = sum((1:N).^2);
    Es_avg = (4 / M) * R^2 * sum_sq;
    SNR_eff = SNR * Es_avg;
    sep = 2 * (M - 1) / M * qfunc(sqrt(6 * SNR_eff / (M^2 - 1)));
end

% Calculate SEP vs SNR for each modulation scheme
```

```matlab
SEP_vs_SNR = zeros(length(SNR_dB), 4);

for i = 1:length(SNR_dB)
    SNR_linear = 10^(SNR_dB(i)/10);
    SEP_vs_SNR(i, 1) = sep_pam(SNR_linear);
    SEP_vs_SNR(i, 2) = sep_psk(SNR_linear);
    SEP_vs_SNR(i, 3) = sep_qam(SNR_linear);
    SEP_vs_SNR(i, 4) = sep_cqam(SNR_linear);
end

% Plot SEP vs SNR
figure;
semilogy(SNR_dB, SEP_vs_SNR(:, 1), 'c-', 'LineWidth', 2); % Cyan solid line
hold on;
semilogy(SNR_dB, SEP_vs_SNR(:, 2), 'm--', 'LineWidth', 2); % Magenta dashed line
semilogy(SNR_dB, SEP_vs_SNR(:, 3), 'k-', 'LineWidth', 2); % Black dash-dot line
semilogy(SNR_dB, SEP_vs_SNR(:, 4), 'r:', 'LineWidth', 2); % Yellow dotted line
xlabel('E_b/N_0 (dB)');
ylabel('SEP');
legend('16-PAM', '16-PSK', '16-QAM', '16-CQAM');
title('SEP vs SNR for 16-PAM, 16-PSK, 16-QAM, and 16-CQAM');
grid on;
```

References

[1] Γεώργιος Ν. Καραγιαννίδης, *Τηλεπικοινωνιακά Συστήματα.* Εκδόσεις Τζιόλα, 2018.

[2] C. P. Ghassan M. Kraidy and I. Krikidis, "Fundamentals of circular qam for wireless information and power transfer," *SPAWC*, 2021.

[3] M. I. Hisham Alasady and Q. Rahmanz, "Symbol error rate calculation and data predistortion for 16-qam transmission over nonlinear memoryless satellite channels," *Wiley InterScience*, 2006.

[4] Q. Chaudhari, *Wireless Communications from the Ground Up: An SDR Perspective.* uknown, 2018.

[5] K. C.-H. Manuel Jose Lopez Morales and A. G. Armada.