Wireless Communication Systems HW4

通訊所一年級 陳劭珩 *110064533*

December 21, 2021

1. Problem description

Assume that there are L (L = 1, 2, 3, 4) diversity branches of uncorrelated Rayleigh fading signals. Each branch has the same average symbol energy–to–noise power ratio E_s/N_0, *for* E_s/N_0 = *1, 3, 5, 7 and 9 dB.*

Simulate the QPSK bit error probability (at least to $P_b = 10^{-4}$*) for the following techniques:*

(a) Selective Combining

(b) Maximal Ratio Combining

(c) Equal Gain Combining

(d) Direct Combining (which combines all paths directly and then compensates the overall phase shift before demodulation.)

(1. You may generate the fading gains via combining a Rayleigh random number and a uniform random phase, or via combining two Gaussian random variables (Complex Gaussian)

2. For coherence detection, you must equalize the phase before demodulation).

1.1 Answer

In wireless communication, "diversity" is a common technique for combatting fading and co–channel interference, and for further improving the reliability of a message signal by using two or more communication channels with different properties, and it is based on individual channels experience different levels of fading and interference.

On the other hand, "diversity combining" is the technique applied to combine the multiple received signals from a diversity reception into a single improved signal.

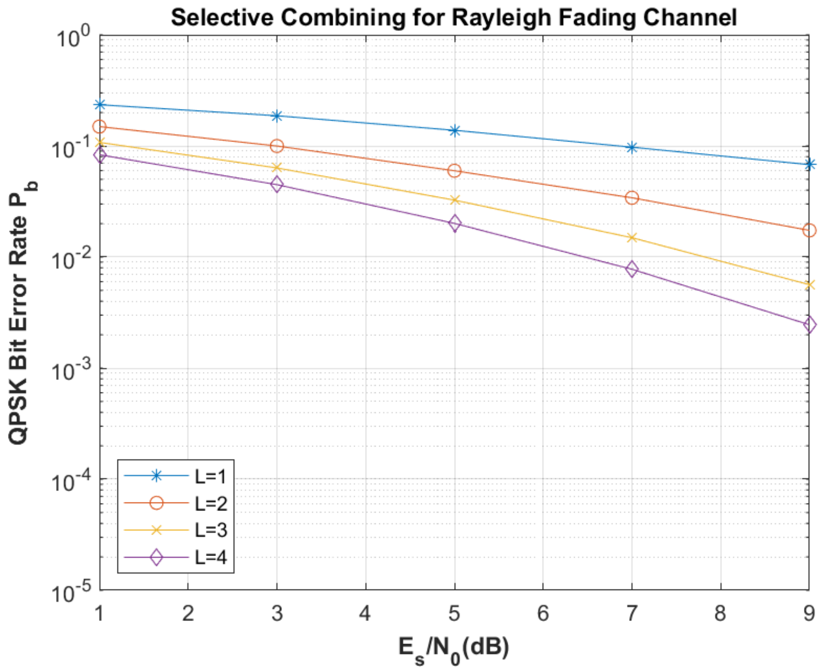
There are two basic types of combining, one will combine the multiple received signals before detection, which is called "Predetection combining." The other, by contrast, will combine signals after detection, which is called "Postdetection combining."

(a) Selective Combining for Rayleigh fading channel

在多個接收到的 diversity branches 中，選擇一個瞬間 SNR 最大的訊號去做 coherent detection。 The effective instantaneous bit energy-to-noise ratio is

$$\gamma_s^S = \max\{\gamma_1, \gamma_2, \cdots, \gamma_L\}$$

– L is the number of diversity branches



(b) Maximal Ratio Combining for Rayleigh fading channel

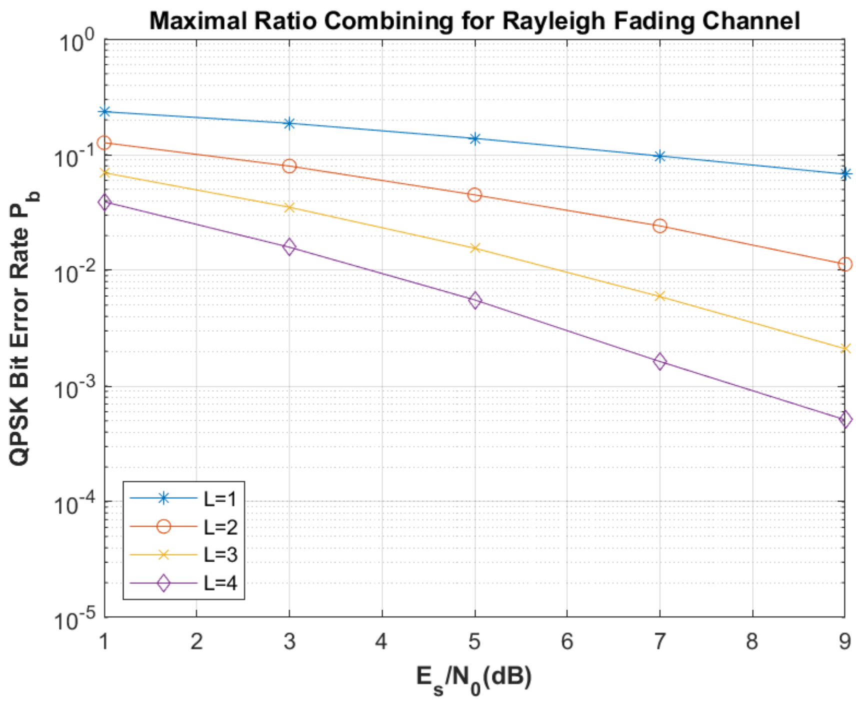
把所有接收到的 diversity branch 都乘上各自 channel gain 的 conjugate 作為 weighting 取一個權重，再 combine 所有 branch。

Weighting factor

$$\tilde{r}(t) = \sum_{k=1}^{L} g_k^* \tilde{r}_k(t) = \sum_{k=1}^{L} g_k^* g_k \tilde{s}(t) + \sum_{k=1}^{L} g_k^* \tilde{n}_k(t)$$

$$\boxed{g_k = \alpha_k e^{j\phi_k}; \quad g_k^* g_k = \alpha_k^2}$$

Channel gain

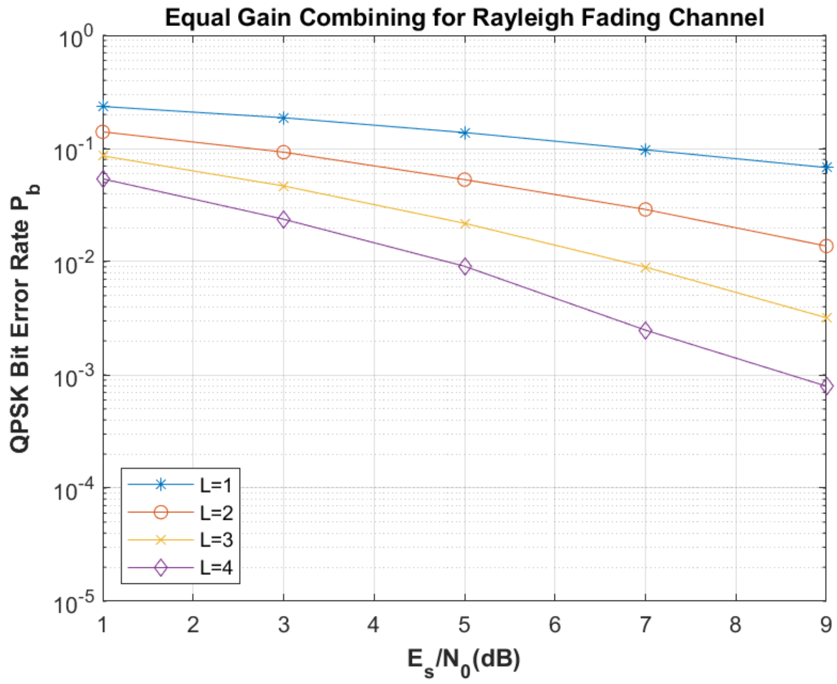


(c) Equal Gain Combining for Rayleigh fading channel

對所有接收到的 diversity branch 都對相位直接補償，也就是去除相位後再 combine 做 coherent detection。

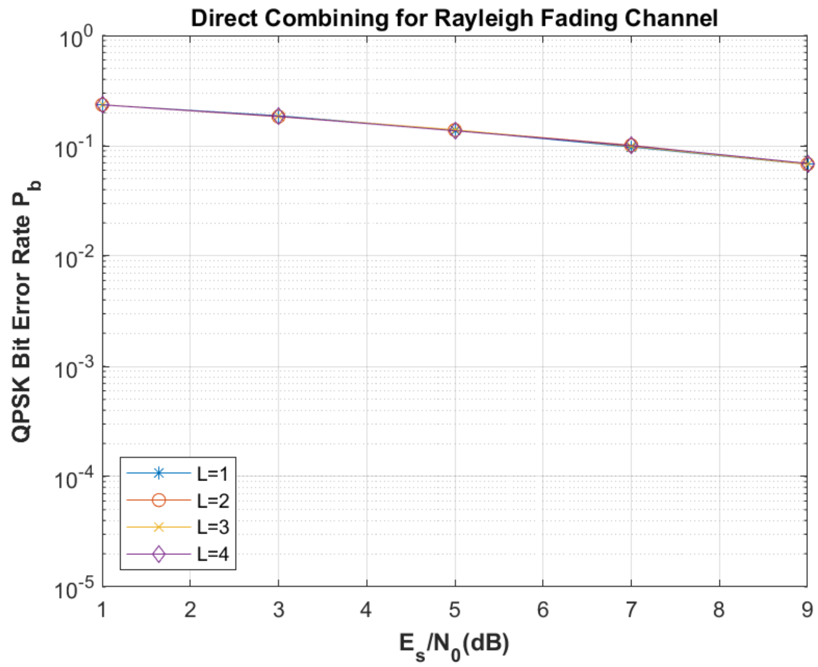
For pre-detection combining, the signal after the diversity combiner is

$$\tilde{\mathbf{r}}(t) = \sum_{k=1}^{L} e^{-j\phi_k} \tilde{r}_k(t) = \sum_{k=1}^{L} e^{-j\phi_k} g_k \tilde{s}(t) + \sum_{k=1}^{L} e^{-j\phi_k} \tilde{n}_k(t)$$



(d) Direct Combining for Rayleigh fading channel

將所有接收到的 diversity branch 都相加後再做相位補償，因為沒有配合其他任何機制，因此不管接收到的 branch 為何，最終結果應該都一樣。



1.2 Code

```
%
% Wireless Communication Systems HW4, 通訊所一年級 110064533 陳劭珩
%
% 1.Assume that there are L(L=1,2,3,4) diversity branches of uncorrelated
% Rayleigh fading signals. Each branch has the same average symbol energy-
% to-noise power ratio E_s/N_0, for E_s/N_0 = 1,3,5,7,9 dB.
%
% Simulate the QPSK bit error probability (at least to P_b=10^(-4)) for
% the following 4 Combining techniques:
%   (a) Selective Combining
%   (b) Maximal Ratio Combining
%   (c) Equal Gain Combining
%   (d) Direct Combining (which combines all paths directly and then
%       compensates the overall phase shift before demodulation.
%
% (1.You may generate the fading gains via combining a Rayleigh random
% number and a uniform random phase, or via combining two Gaussian random
% variables (Complex Gaussian.)
% 2.For coherence detection, must equalize the phase before demodulation.)
%
% clear;
% clc;
%
N = 1e5;                % number of simulations, say 100000 symbols
L = 4;                  % diversity branch L = 1~4
SNR = [1 3 5 7 9];      % SNR = Es/N0 = 1,3,5,7,9 dB, in our case
Es = 1;                 % set symbol energy = 1 Joule
N0 = Es*10.^(-SNR/10); % compute N0 from given SNR, Es. derived as below
% 10log(Es/N0) = SNR(dB) --> log(Es/N0) = SNR/10 --> 10^(SNR/10) = Es/N0
% --> N0 = Es*10^-(SNR/10)
%
% signal mapping
s00 = [ 1  1];
s01 = [ 1 -1];
s11 = [-1  1];
s10 = [-1 -1];
%
```

```
% source input, store the real parts and imaginary parts of the complex
% input signal as source_real, and source_imag, separately.
source_real = zeros(1, N);
source_imag = zeros(1, N);
r = zeros(1, L);        % received signal
BER = zeros(L, length(SNR), L);  % Bit Error Rate
%
% L = 1~4 diversity branches
%
for L_idx = 1 : L
    for k = 1 : length(SNR)
        num_bit_error = zeros(L, 1);
        for i = 1 : N
            % Initialize the random source signal for different SNR cases
            temp = rand; % uniformly distributed in (0, 1)
            if temp < 0.25
                % if rand < 0.25, then set the source input as '00'
                source_real(i) = 0;
                source_imag(i) = 0;
            elseif 0.25 < temp && temp < 0.5
                % if 0.25 < rand < 0.5, then set the source input as '01'
                source_real(i) = 0;
                source_imag(i) = 1;
            elseif 0.5 < temp && temp < 0.75
                % if 0.5 < rand < 0.75, then set the source input as '11'
                source_real(i) = 1;
                source_imag(i) = 0;
            elseif 0.75 < temp
                % if 0.5 < rand < 0.75, then set the source input as '10'
                source_real(i) = 1;
                source_imag(i) = 1;
            end
            %
            %
            diversity=zeros(L,1);
            gain = 0;
            %
            % Generate the received signal
            for j = 1 : L_idx
                %
                % Rayleigh = the amplitube of two complex Gaussian combined
```

```
        rayleigh = sqrt(1/2) * (randn + 1i*randn);
        % AWGN noise
        noise = sqrt(N0(k)/2) * (randn + 1i*randn);
        %
        % received signal = sorce signal * channel fading + noise
        if ((source_real(i)==0) && (source_imag(i)==0))
            % r = s00 * Rayleigh fading + noise
            r(j) = rayleigh * sqrt(Es/2) * s00 * [1; 1i] + noise;
        elseif ((source_real(i)==0) && (source_imag(i)==1))
            % r = s01 * Rayleigh fading + noise
            r(j) = rayleigh * sqrt(Es/2) * s01 * [1; 1i] + noise;
        elseif (source_real(i) == 1) && (source_imag(i) == 0)
            % r = s10 * Rayleigh fading + noise
            r(j) = rayleigh * sqrt(Es/2) * s10 * [1; 1i] + noise;
        elseif (source_real(i) == 1) && (source_imag(i) == 1)
            % r = s11 * Rayleigh fading + noise
            r(j) = rayleigh * sqrt(Es/2) * s11 * [1; 1i] + noise;
        end
        %
        % technique (1) Selective Combining
        if abs(conj(rayleigh) * r(j)) > abs(diversity(1))
            diversity(1) = conj(rayleigh) * r(j);
        end
        %
        % technique (2) Maximal Ratio Combining
        diversity(2) = diversity(2) + conj(rayleigh) * r(j);
        %
        % technique (3) Equal Gain Combining
        temp_equal_gain = conj(rayleigh) * r(j) / abs(rayleigh);
        diversity(3) = diversity(3) + temp_equal_gain;
        %
        % technique (4) Direct Combining
        diversity(4) = diversity(4) + r(j);
        gain = gain + rayleigh;
    end
    diversity(4) = conj(gain) * diversity(4) / abs(gain);
    %
```

```
    %
    for j = 1 : L
        %
        % compute the correlation matrixs
        %
        c00 = real(diversity(j))*s00(1)*sqrt(Es/2) ...
            + imag(diversity(j))*s00(2)*sqrt(Es/2);
        c01 = real(diversity(j))*s01(1)*sqrt(Es/2) ...
            + imag(diversity(j))*s01(2)*sqrt(Es/2);
        c10 = real(diversity(j))*s10(1)*sqrt(Es/2) ...
            + imag(diversity(j))*s10(2)*sqrt(Es/2);
        c11 = real(diversity(j))*s11(1)*sqrt(Es/2) ...
            + imag(diversity(j))*s11(2)*sqrt(Es/2);
        %
        % decide the received signal based on the correlations
        %
        % constellation point (received_real, received_imag)
        max_corr = max([c00 c01 c10 c11]);
        if (c00 == max_corr)      % s00
            received_real=0;
            received_imag=0;
        elseif (c01 == max_corr) % s01
            received_real=0;
            received_imag=1;
        elseif (c10 == max_corr) % s10
            received_real=1;
            received_imag=0;
        elseif (c11 == max_corr) % s11
            received_real=1;
            received_imag=1;
        end
        %

        if (received_real ~= source_real(i))
            num_bit_error(j) = num_bit_error(j) + 1;
        end
        if (received_imag ~= source_imag(i))
            num_bit_error(j) = num_bit_error(j) + 1;
        end
    end
end
BER(:, k, L_idx) = num_bit_error / (2*N);
```

```
    end
end
%
% plot and save, 4 diversity techniques
%
for L_idx = 1 : L
    figure; % plot different L separately
    semilogy(SNR, BER(L_idx,:,1), '-*', ...
             SNR, BER(L_idx,:,2), '-o', ...
             SNR, BER(L_idx,:,3), '-x', ...
             SNR, BER(L_idx,:,4), '-d');
    hold on;
    legend('L=1', 'L=2', 'L=3', 'L=4', 'location', 'southwest');
    xlabel('E_s/N_0(dB)', 'FontWeight', 'bold');
    ylabel('QPSK Bit Error Rate P_b', 'FontWeight', 'bold');
    xlim([1 9]);
    ylim([10^-5 1]);
    grid on;
    if L_idx == 1
        title('Selective Combining for Rayleigh Fading Channel');
        %
        % save as Selective Combining for Rayleigh Fading Channel.png
        fname = 'Selective Combining for Rayleigh Fading Channel.png';
        print(fname, '-dpng');
    elseif L_idx == 2
        title('Maximal Ratio Combining for Rayleigh Fading Channel');
        %
        % save as Maximal Ratio Combining for Rayleigh Fading Channel.png
        fname = 'Maximal Ratio Combining for Rayleigh Fading Channel.png';
        print(fname, '-dpng');
    elseif L_idx == 3
        title('Equal Gain Combining for Rayleigh Fading Channel');
        %
        % save as Equal Gain Combining for Rayleigh Fading Channel.png
        fname = 'Equal Gain Combining for Rayleigh Fading Channel.png';
        print(fname, '-dpng');

    elseif L_idx == 4
        title('Direct Combining for Rayleigh Fading Channel');
        %
        % save as Direct Combining for Rayleigh Fading Channel.png
        fname = 'Direct Combining for Rayleigh Fading Channel.png';
        print(fname, '-dpng');
```

2. Problem description

Repeat the problem for uncorrelated Ricean fading with K = 1.

2.1 Answer

先求得當 K = 1 時，組成 Ricean fading 的兩個 Gaussian r.v.的 mean 跟 variance，計算過程如下，因此推知可以藉由兩個統計特性為 N(0.5, 0.25)的 Gaussian 來模擬。並可透過將 Matlab 內建統計特性為 N(0, 1)的 randn 平移得到統計特性為 N(0.5, 0.25)的 Gaussian。

$$p_\alpha(x) = \frac{x}{\sigma^2} e^{-\frac{(x^2+s^2)}{2\sigma^2}} \cdot I_0\left(\frac{x \cdot s}{\sigma^2}\right), \quad x \geq 0$$

$g_I(t), g_Q(t) \sim$ independent Gaussian r.v. with non-zero-mean $m_I(t)$ and $m_Q(t)$

$\boxed{s^2 = m_I^2 + m_Q^2}$ $\quad \boxed{K = \frac{s^2}{2\sigma^2}}$ $\quad \boxed{\Omega_p = s^2 + 2\sigma^2}$

$K = 1 \hookrightarrow s^2 = 2\sigma^2$, assume that average power

$E[\alpha^2] = \Omega_p = 1 \hookrightarrow 1 = s^2 + 2\sigma^2 = 2\sigma^2 + 2\sigma^2$

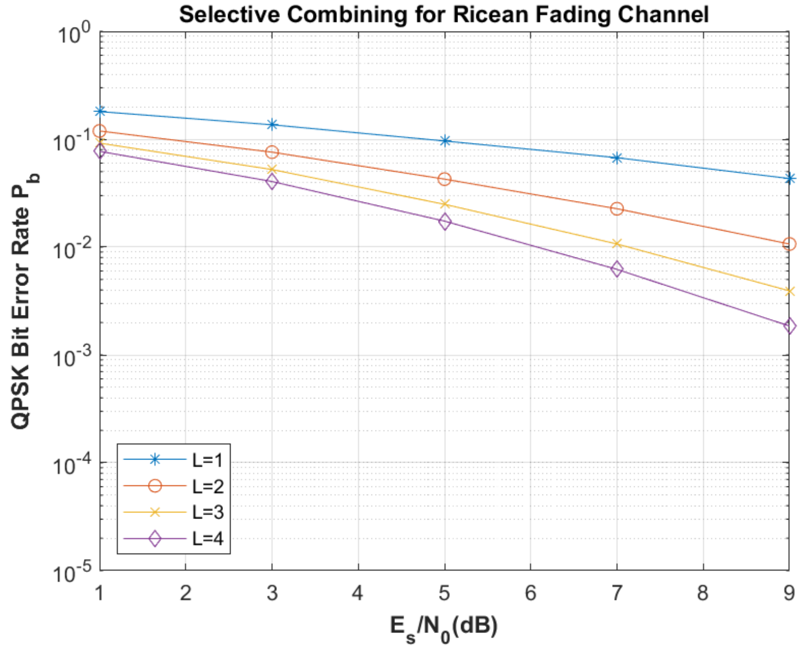
$\Rightarrow$ variance $\sigma^2 = \frac{1}{4}$, then $s^2 = 2\sigma^2 = \frac{1}{2}$

assume that $m_I = m_Q = m$ Ricean variance

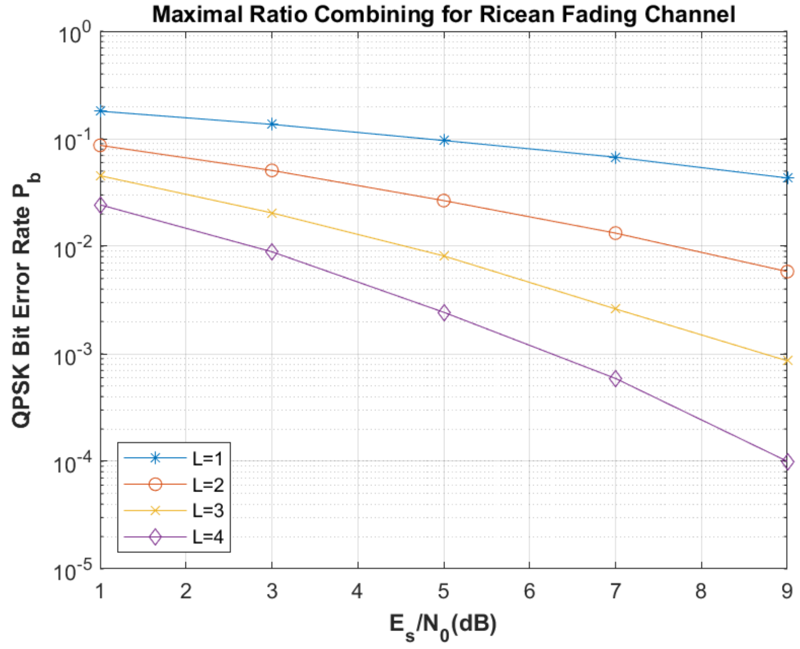
$\hookrightarrow s^2 = m^2 + m^2 = \frac{1}{2}$ $\quad m = \frac{1}{2}$ Ricean mean

$\therefore$ we can generate a Ricean fading with $K=1$ via combining two Gaussian r.v. with characteristic $N(m, \sigma^2) = N(\frac{1}{2}, \frac{1}{4})$

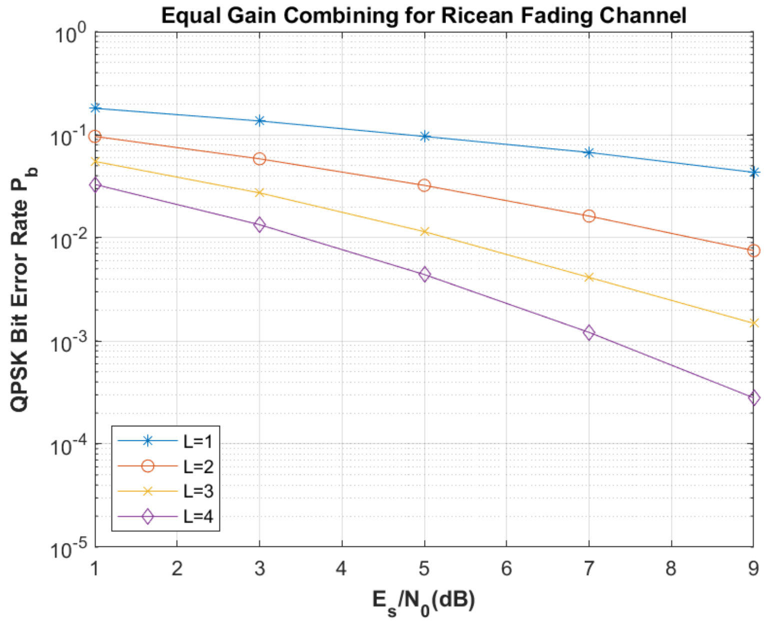
(a) Selective Combining for Ricean fading channel



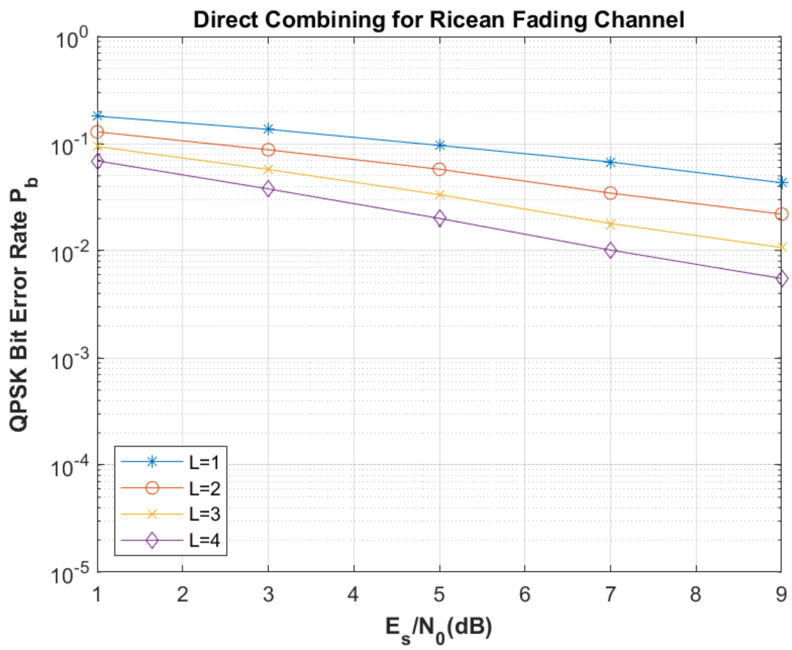
(b) Maximal Ratio Combining for Ricean fading channel



(c) Equal Gain Combining for Ricean fading channel



(d) Direct Combining for Ricean fading channel



2.2 Code

```
%
% Wireless Communication Systems HW4, 通訊所一年級 110064533 陳劭珩
%
% 2.(Repeat the problem for uncorrelated Ricean fading with K = 1.) Assume
% that there are L(L=1,2,3,4) diversity branches of uncorrelated Ricean
% fading signals with K=1. Each branch has the same average symbol energy-
% to-noise power ratio E_s/N_0, for E_s/N_0 = 1,3,5,7,9 dB.
%
% Simulate the QPSK bit error probability (at least to P_b=10^(-4)) for
% the following 4 Combining techniques:
%   (a) Selective Combining
%   (b) Maximal Ratio Combining
%   (c) Equal Gain Combining
%   (d) Direct Combining (which combines all paths directly and then
%       compensates the overall phase shift before demodulation.
%
% (1.You may generate the fading gains via combining a Rayleigh random
% number and a uniform random phase, or via combining two Gaussian random
% variables (Complex Gaussian.)
% 2.For coherence detection, must equalize the phase before demodulation.)
%
% clear;
% clc;
%
N = 1e5;                  % number of simulations, say 100000 symbols
L = 4;                    % diversity branch L = 1~4
SNR = [1 3 5 7 9];        % SNR = Es/N0 = 1,3,5,7,9 dB, in our case
Es = 1;                   % set symbol energy = 1 Joule
N0 = Es*10.^(-SNR/10); % compute N0 from given SNR, Es. derived as below
% 10log(Es/N0) = SNR(dB) --> log(Es/N0) = SNR/10 --> 10^(SNR/10) = Es/N0
% --> N0 = Es*10^-(SNR/10)
%
K = 1;                    % Ricean factor K = 1
Ricean_mean = 1/2;        % Ricean mean m = 0.5
Ricean_sigma = 1/2;       % Ricean variance σ^2 = 0.25
%
%
SelectiveComb_data = zeros(1, N);
SelectiveComb_error = zeros(L, length(SNR));
SelectiveComb_BER = zeros(L, length(SNR));
```

```
%
MaxRatioComb_data = zeros(L, N);
MaxRatioComb_error = zeros(L, length(SNR));
MaxRatioComb_BER = zeros(L, length(SNR));
%
EqualGainComb_data = zeros(L, N);
EqualGainComb_error = zeros(L, length(SNR));
EqualGainComb_BER = zeros(L, length(SNR));
%
DirectComb_error = zeros(L, length(SNR));
DirectComb_BER = zeros(L, length(SNR));
%
% Ricean fading channel case
%
for N0_idx = 1 : length(N0)
    %
    % QPSK modulation
    %
    x = 2*round(rand(1, N)) - 1;
    y = 2*round(rand(1, N)) - 1;
    modulated_data = sqrt(1/2) * (x + 1i*y);
    %
    % Channel
    %
    channel_gain = Ricean_sigma .* (randn(L,N) + randn(L,N)*1i) ...
                 + Ricean_mean * (1 + 1i);
    awgn_noise = sqrt(N0(N0_idx)/2) * randn(L,N) ...
               + sqrt(N0(N0_idx)/2) * 1i*randn(L,N);
    branch_signal = channel_gain .* modulated_data;
    rx_data = branch_signal + awgn_noise;
    %
    % L = 1~4 diversity branches
    %
    for L_idx = 1 : L
        %
        % Selctive Combining
        %

        if L_idx == 1
            % equalize the phase
            SelectiveComb_data = rx_data(L_idx,:) ...
                                 .* exp(-1i*angle(channel_gain(L_idx,:)));
```

```
else
    [~, max_idx] = max(abs(channel_gain(1:L_idx, :)));
    for N_idx = 1 : N
        SelectiveComb_data(N_idx) = rx_data(max_idx(N_idx), N_idx)
            .* exp(-1i*angle(channel_gain(max_idx(N_idx), N_idx)));
    end
end
detect_real = 2*(real(SelectiveComb_data) > 0) - 1;
detect_imag = 2*(imag(SelectiveComb_data) > 0) - 1;
SelectiveComb_error(L_idx, N0_idx) = length(find(detect_real - x))
                                   + length(find(detect_imag - y));
SelectiveComb_BER = SelectiveComb_error / (2*N);
%
% Maxmimal Ratio Combining
%
MaxRatioComb_data(L_idx,:) = conj(channel_gain(L_idx,:)) ...
                             .* rx_data(L_idx,:);
if L_idx == 1
    detect_real = 2*(real(MaxRatioComb_data(1:L_idx,:)) > 0) - 1;
    detect_imag = 2*(imag(MaxRatioComb_data(1:L_idx,:)) > 0) - 1;
else
    detect_real = 2*(real(sum(MaxRatioComb_data(1:L_idx,:)))>0) -1;
    detect_imag = 2*(imag(sum(MaxRatioComb_data(1:L_idx,:)))>0) -1;
end
MaxRatioComb_error(L_idx, N0_idx) = length(find(detect_real - x))
                                  + length(find(detect_imag - y));
MaxRatioComb_BER = MaxRatioComb_error / (2*N);
%
% Equal Gain Combining
%
EqualGainComb_data(L_idx,:) = rx_data(L_idx,:) ...
                            .* exp(-1i*angle(channel_gain(L_idx,:)));
if L_idx==1
    detect_real=2*((real(EqualGainComb_data(1:L_idx,:)) > 0)) - 1;
    detect_imag=2*((imag(EqualGainComb_data(1:L_idx,:)) > 0)) - 1;
else
    detect_real=2*((real(sum(EqualGainComb_data(1:L_idx,:)))>0))-1;

    detect_imag=2*((imag(sum(EqualGainComb_data(1:L_idx,:)))>0))-1;
end
```

```
        EqualGainComb_error(L_idx, N0_idx) = length( find(detect_real - x))
                                           + length(find(detect_imag - y));
        EqualGainComb_BER = EqualGainComb_error / (2*N);
        %
        % Direct Combining
        %
        if L_idx == 1
            phase_equalize = exp(-1i*angle(channel_gain(L_idx,:)));
            detect_real = (2*(real(rx_data(1:L_idx,:) ...
                        .*phase_equalize) > 0)) - 1;
            detect_imag = (2*(imag(rx_data(1:L_idx,:) ...
                        .*phase_equalize) > 0)) - 1;
        else
            phase_equalize = exp(-1i*angle(sum(channel_gain(1:L_idx,:))));
            detect_real = (2*(real(sum(rx_data(1:L_idx,:) ...
                        .*phase_equalize)) > 0)) - 1;
            detect_imag = (2*(imag(sum(rx_data(1:L_idx,:) ...
                        .*phase_equalize)) > 0)) - 1;
        end
        DirectComb_error(L_idx,N0_idx) = length(find(detect_real - x)) ...
                                       + length(find(detect_imag - y));
        DirectComb_BER = DirectComb_error / (2*N);
    end
end
%
% plot and save, 4 diversity techniques
%
figure(1);
% semilogy(SNR, SelectiveComb_BER,'-*'); % plot different L separately
semilogy(SNR, SelectiveComb_BER(1, :), '-*', ...
         SNR, SelectiveComb_BER(2, :), '-o', ...
         SNR, SelectiveComb_BER(3, :), '-x', ...
         SNR, SelectiveComb_BER(4, :), '-d');
title('Selective Combining for Ricean Fading Channel');
xlim([1 9]);
ylim([10^-5 1]);
xlabel('E_s/N_0(dB)', 'FontWeight', 'bold');

ylabel('QPSK Bit Error Rate P_b', 'FontWeight', 'bold');
legend('L=1', 'L=2', 'L=3', 'L=4', 'location', 'southwest');
grid on;
```

```
%
% save as Selective Combining for Ricean Fading Channel.png
fname = 'Selective Combining for Ricean Fading Channel.png';
print(fname, '-dpng');
%
figure(2);
% semilogy(SNR, MaxRatioComb_BER,'-o'); % plot different L separately
semilogy(SNR, MaxRatioComb_BER(1, :), '-*', ...
         SNR, MaxRatioComb_BER(2, :), '-o', ...
         SNR, MaxRatioComb_BER(3, :), '-x', ...
         SNR, MaxRatioComb_BER(4, :), '-d');
title('Maximal Ratio Combining for Ricean Fading Channel');
xlim([1 9]);
ylim([10^-5 1]);
xlabel('E_s/N_0(dB)', 'FontWeight', 'bold');
ylabel('QPSK Bit Error Rate P_b', 'FontWeight', 'bold');
legend('L=1', 'L=2', 'L=3', 'L=4', 'location', 'southwest');
grid on;
%
% save as Maximal Ratio Combining for Ricean Fading Channel.png
fname = 'Maximal Ratio Combining for Ricean Fading Channel.png';
print(fname, '-dpng');
%
figure(3);
% semilogy(SNR, EqualGainComb_BER,'-x'); % plot different L separately
semilogy(SNR, EqualGainComb_BER(1, :), '-*', ...
         SNR, EqualGainComb_BER(2, :), '-o', ...
         SNR, EqualGainComb_BER(3, :), '-x', ...
         SNR, EqualGainComb_BER(4, :), '-d');
title('Equal Gain Combining for Ricean Fading Channel');
xlim([1 9]);
ylim([10^-5 1]);
xlabel('E_s/N_0(dB)', 'FontWeight', 'bold');
ylabel('QPSK Bit Error Rate P_b', 'FontWeight', 'bold');
legend('L=1', 'L=2', 'L=3', 'L=4', 'location', 'southwest');
grid on;
%
% save as Equal Gain Combining for Ricean Fading Channel.png
fname = 'Equal Gain Combining for Ricean Fading Channel.png';
print(fname, '-dpng');
```

```
%
figure(4);
% semilogy(SNR, DirectComb_BER,'-d'); % plot different L separately
semilogy(SNR, DirectComb_BER(1, :), '-*', ...
         SNR, DirectComb_BER(2, :), '-o', ...
         SNR, DirectComb_BER(3, :), '-x', ...
         SNR, DirectComb_BER(4, :), '-d');
title('Direct Combining for Ricean Fading Channel');
xlim([1 9]);
ylim([10^-5 1]);
xlabel('E_s/N_0(dB)', 'FontWeight', 'bold');
ylabel('QPSK Bit Error Rate P_b', 'FontWeight', 'bold');
legend('L=1', 'L=2', 'L=3', 'L=4', 'location', 'southwest');
grid on;
%
% save as Direct Combining for Ricean Fading Channel.png
fname = 'Direct Combining for Ricean Fading Channel.png';
print(fname, '-dpng');
%
%
```

3. Problem description

Compare and discuss the results of different cases.

3.1 Answer

(1)當在相同通道情況下，4 種 Combining technique 比較，MRC > EGC > SC > DC。

Maximal Ratio Combining 效果最好，因為乘上最適當的 weighting 去取權重、

Equal Gain Combining 效果次之，Equal Gain 去除相位後有利用全部 branch 的資訊，雖然部分資訊可能受干擾和雜訊影響很大，多少還是比 Selective 好、

Selective Combining 效果再次之，因為 Selective 只取 SNR 最好的，因此捨棄掉其他可能 SNR 也不錯的 branch，效果有限。

最後是沒有做什麼處理就直接相加總的 Direct Combining。

(2)在不同通道的情況下，Rayleigh 跟 Ricean 比較，Ricean> Rayleigh。

因為 Ricean 有包含 LOS 的成分，因此當 diversity branch 越多，越有機會包含到

LOS 成分的信號，而又因為有 LOS 成分的訊號較為穩定，因此在相同 SNR 及採用相同 Combining technique 的情況下，去 error rate 比較一定是 Ricean 優於 Rayleigh。

4. Feedbacks

難得稍微有時間喘口氣，所以寫個心得湊頁數+反思一下，可以當我在自言自語 XD 開始寫這次作業、跑模擬時發現很多先前學得觀念都沒有理解得很透徹，但其實寫完作業也感覺還是沒有搞懂，比如說 QPSK 概念上知道，但在模擬的時候還是一知半解，不知道怎麼把用弦波模擬和用星座圖模擬的觀念連結起來，對我來講目前這兩個概念是分開理解的，但醬感覺蠻怪的。所以先用 numpy 跟 matplotlib 簡單模擬 QPSK signaling，把想法用 code 實作看看(但 signaling 跟 modulation 是一樣的意思嗎?)輸出波形看起來應該是對的...? 但要從弦波去生星座圖座標點好像又卡住了。(關於為什麼要用 python 寫，是因為不擅長...也不喜歡寫 Matlab，很想脫離 Matlab 的掌控 XD) ((但還是得練習@@ 所以這次作業 2 題用稍微不同的概念跟程式邏輯去模擬

```
# QPSK signaling simulation
import numpy as np
import matplotlib.pyplot as plt

# source input
t1 = np.arange(0, 8.5, 0.5)

plt.subplot(4, 1, 1)
y1 = [0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1]
plt.plot(t1, y1, drawstyle='steps-post')
plt.xlim(0, 8)
plt.ylim(-0.5, 1.5)
plt.title('Input Signal')

# common variable of I and Q channel
a = 1 / np.sqrt(2) # normalized amplitude
tI = np.arange(0, 9, 1)

# Real part, I Signal
plt.subplot(4, 1, 2)
yI = [-a, a, -a, a, -a, a, -a, a, a]
plt.plot(tI, yI, drawstyle='steps-post')
plt.xlim(0, 8)
plt.ylim(-2, 2)
plt.title('I Signal')
```

```
# Imaginary part, Q Signal
plt.subplot(4, 1, 3)
yQ = [a, -a, -a, a, -a, a, -a, a, a]
plt.plot(tI, yQ, drawstyle='steps-post')
plt.xlim(0, 8)
plt.ylim(-2, 2)
plt.title('Q Signal')

# QPSK Signal
plt.subplot(4, 1, 4)
t = np.arange(0, 9, 0.01)

def output_waveform(I, Q, t):
    rect_wave = []
    for i in range(0, len(I)):
        t_temp = t[((i)*100) : ((i+1)*100)]
        yI_temp = yI[i]*np.ones(100)
        yQ_temp = yQ[i]*np.ones(100)
        wave_temp = yI_temp*np.cos(2*np.pi*5*t_temp) - yQ_temp*np.sin(2*np.pi*5*t_temp)
        rect_wave.append(wave_temp)
    return rect_wave

rect_wave = output_waveform(yI, yQ, t)
plt.plot(t, np.array(rect_wave).flatten(), 'r')
plt.xlim(0, 8)
plt.ylim(-2, 2)
plt.title('QPSK Signal')

plt.tight_layout()
plt.show()
```