CL 1

```
%% Initialization
clc;
clear;
close all;

%% Set basic parameters
nr = [5 6 7 8 9 10]; % number of receive antennas
nt = nr; % number of transmit antennas

p = 31; % take 31 samples
SNR_dB = (0:30/(p-1):30); % SNR_dB = 0, 0.3, ... , 29.7, 30
SNR = 10.^(SNR_dB/10); % calculate SNR, SNR_dB = 10log10(SNR),
SNR_dB = 1, ... , 1000

C_H = zeros(10000,p); % initialize capacity C_H under certain
nr with 10000 rows, each row is calculated from a random H,
with length p
C_ergodic = zeros(length(nr),p); % initialize mean capacity
C_ergodic, each row is the mean of C_H under certain nr, with
length p

% Avoid the pseudorandomness
rand('twister',mod(floor(now*8640000),2^31-1))

%% Calculation
for j = 1:length(nr) % take different nr = [5 6 7 8 9 10]

    I = eye(nr(j)); % build nr*nr identical matrix
    H = zeros(nr(j)); % initialize channel H (nr*nr)

    for L = 1:10000 % use i.i.d. random H to calculate C_H
        for m = 1:nr(j)
            for n = 1:nr(j)
                % find elements of H, H is a complex Gaussian
random matrix with i.i.d. entries
                H(m,n) = (randn+1i*randn)/sqrt(2); % divided
by sqrt(2) to keep variance = 1
            end
        end
        H_h = H'; % H_h is the conjugate transpose of H

        for k = 1:p % calculate C_H with H, H_h, and p values
of SNR from 1-1000 (0-30dB)
            C_H(L,k) = log2(det(I+SNR(k)/nt(j).*(H*H_h)));
        end
    end
```

```
    C_ergodic(j,:) = mean(C_H); % C_ergodic is the average of
10000 rows of C_H
end

C_ergodic_abs = abs(C_ergodic); % take absolute value to plot
the figure (the imaginary part is about 1.0e-16 degree,
ignorable)

%% Plot the figure
set(gcf,'Position',[10,10,700,500]); % Set the figure size
set(gca,'fontname','Times New Roman'); % Set the font type

a = plot(SNR_dB,C_ergodic_abs(1,:),'LineWidth',1.5); % Plot
the figure
hold on
b = plot(SNR_dB,C_ergodic_abs(2,:),'LineWidth',1.5);
c = plot(SNR_dB,C_ergodic_abs(3,:),'LineWidth',1.5);
d = plot(SNR_dB,C_ergodic_abs(4,:),'LineWidth',1.5);
e = plot(SNR_dB,C_ergodic_abs(5,:),'LineWidth',1.5);
f = plot(SNR_dB,C_ergodic_abs(6,:),'LineWidth',1.5);

a.Color = [1 0 0]; % Set the colour of lines
b.Color = [0 0.9 1];
c.Color = [1 0 1];
d.Color = [1 0.7 0];
e.Color = [0 0.8 0];
f.Color = [0 0 0.8];

grid on
legend('nr = 5','nr = 6','nr = 7','nr = 8','nr = 9','nr =
10','FontSize',11); % Set the legend
xlabel('SNR (dB)','FontSize',13); % Set the axes
ylabel('Capacity (bps/Hz)','FontSize',13);
title('MIMO Capacity','FontSize',15); % Set the title
text(22, 15, sprintf('Yuxiang
Zheng\n17/10/2021'),'FontSize',12);
```

CL 2&3

```
%% Initialization
clc;
clear;
```

```
close all;

%% Set basic parameters
n = 2; % number of receive antennas
p = 8; % take 8 samples
L = 0;
Loop = 1000;
b = 10^6; % transmit 10^6 bits

I = eye(n,n);

SNR_dB = (0:35/(p-1):35); % SNR_dB = 0, 0.35, ... , 34.65, 35
SNR = 10.^(SNR_dB/10); % calculate SNR, SNR_dB = 10log10(SNR),
SNR_dB = 1, ... , 3162.3

N_L = zeros(n,b/2);
ML = zeros(16,0);
ML_coded = zeros(16,0);

BER_ML = zeros(Loop,p);
BER_ZF = zeros(Loop,p);
BER_MMSE = zeros(Loop,p);
BER_ML_coded = zeros(Loop,p);

mean_BER_ML = zeros(1,p);
mean_BER_ZF = zeros(1,p);
mean_BER_MMSE = zeros(1,p);
mean_BER_ML_coded = zeros(1,p);

%% Gray Mapping

x_00 = sqrt(0.5)*(-1-1i); % four types of symbols
x_01 = sqrt(0.5)*(-1+1i);
x_10 = sqrt(0.5)*(+1-1i);
x_11 = sqrt(0.5)*(+1+1i);

x_gray = [[x_00 x_00 x_00 x_00 x_01 x_01 x_01 x_01 x_10 x_10
x_10 x_10 x_11 x_11 x_11 x_11]; % 16 kinds of symbol
combinations
          [x_00 x_01 x_10 x_11 x_00 x_01 x_10 x_11 x_00 x_01
x_10 x_11 x_00 x_01 x_10 x_11]];
x_gray_coded = zeros(n,n,16);
for q = 1:16
    x_gray_coded(:,1,q) = x_gray(:,q);
    x_gray_coded(:,2,q) = [-
conj(x_gray(2,q));conj(x_gray(1,q))]; % 16 kinds of coded
symbol combinations
end
```

```
c_all =  sqrt(0.5)*(randsrc(n,b/4)+1i*randsrc(n,b/4)); %
generate b/2 symbols


%% Plot the figure
set(gcf,'Position',[10,10,700,500]); % Set the figure size
set(gca,'fontname','Times New Roman'); % Set the font type

f_1 = semilogy(SNR_dB,mean_BER_ML,'LineWidth',1.5); % Plot the
figure
axis([min(SNR_dB) max(SNR_dB) 1/b 1])
hold on
f_2 = semilogy(SNR_dB,mean_BER_ZF,'LineWidth',1.5);
f_3 = semilogy(SNR_dB,mean_BER_MMSE,'LineWidth',1.5);
f_4 = semilogy(SNR_dB,mean_BER_ML_coded,'LineWidth',1.5); %
Plot the figure

f_1.Color = [1 0 0]; % Set the colour of lines
f_2.Color = [0 0.9 1];
f_3.Color = [1 0 1];
f_4.Color = [0 0.8 0];

grid on
legend('BER ML','BER ZF','BER MMSE','BER ML
coded','FontSize',11); % Set the legend
xlabel('SNR (dB)','FontSize',13); % Set the axes
ylabel('BER','FontSize',13);
title({'BER vs SNR',['Loops : ',num2str(L)]},'FontSize',15); %
Set the title
text(2, 0.5*10^-5, sprintf('Yuxiang
Zheng\n21/10/2021'),'FontSize',12);
drawnow

f_1.YDataSource = 'mean_BER_ML';
f_2.YDataSource = 'mean_BER_ZF';
f_3.YDataSource = 'mean_BER_MMSE';
f_4.YDataSource = 'mean_BER_ML_coded';

%% Main Loop
for L = 1:Loop

    H = sqrt(0.5)*(randn(n,n)+1i*randn(n,n)); % for each loop,
generate a H
    N_L = sqrt(0.5)*(randn(n,b/2)+1i*randn(n,b/2)); % 16 for
each loop, generate a set of noise

    for k = 1:p
        E_ML = 0;
        E_ZF = 0;
```

```
        E_MMSE = 0;
        E_ML_coded = 0;

        %% Transmission & detection
        for o = 1:b/4
            c = c_all(:,o); % take one column of the symbol
set
            c_coded(:,1) = c;
            c_coded(:,2) = [-conj(c(2));conj(c(1))]; %
generate the coded c

            N = N_L(:,(2*o-1)); % take one column of the noise
set
            N_coded = N_L(:,(2*o-1):2*o); % take two columns
of the noise set for coded c

            A = sqrt(SNR(k)/n);
            y = A*H*c + N; % calculate received y
            y_coded = A*H*c_coded + N_coded; % calculate
received & coded y

            for q = 1:16
                ML(q) = norm(y-A*H*x_gray(:,q));
            end
            [~,Col]=min(ML); % ML detection
            R_ML = x_gray(:,Col);

            R_ZF = (A*H)\y; % Zero forcing
            R_MMSE = (A^2*(H'*H)+I)\(A*H'*y); %  minimum mean-
square error detection

            for q = 1:16
                ML_coded(q) = (norm(y_coded-
A*H*x_gray_coded(:,:,q), 'fro'))^2;
            end
            [~,Col]=min(ML_coded); % ML detection for coded
signal
            R_ML_coded = x_gray(:,Col);

            %% Check the correctness
            if (sign(real(R_ML(1,1))) ~= sign(real(c(1,1))))
|| (sign(imag(R_ML(1,1))) ~= sign(imag(c(1,1))))
                E_ML = E_ML + 1;
            end
            if (sign(real(R_ML(2,1))) ~= sign(real(c(2,1))))
|| (sign(imag(R_ML(2,1))) ~= sign(imag(c(2,1))))
                E_ML = E_ML + 1;
            end
```

```
            if (sign(real(R_ZF(1,1))) ~= sign(real(c(1,1))))
|| (sign(imag(R_ZF(1,1))) ~= sign(imag(c(1,1))))
                E_ZF = E_ZF + 1;
            end
            if (sign(real(R_ZF(2,1))) ~= sign(real(c(2,1))))
|| (sign(imag(R_ZF(2,1))) ~= sign(imag(c(2,1))))
                E_ZF = E_ZF + 1;
            end

            if (sign(real(R_MMSE(1,1))) ~= sign(real(c(1,1))))
|| (sign(imag(R_MMSE(1,1))) ~= sign(imag(c(1,1))))
                E_MMSE = E_MMSE + 1;
            end
            if (sign(real(R_MMSE(2,1))) ~= sign(real(c(2,1))))
|| (sign(imag(R_MMSE(2,1))) ~= sign(imag(c(2,1))))
                E_MMSE = E_MMSE + 1;
            end

            if (sign(real(R_ML_coded(1,1))) ~=
sign(real(c(1,1)))) || (sign(imag(R_ML_coded(1,1))) ~=
sign(imag(c(1,1))))
                E_ML_coded = E_ML_coded + 1;
            end
            if (sign(real(R_ML_coded(2,1))) ~=
sign(real(c(2,1)))) || (sign(imag(R_ML_coded(2,1))) ~=
sign(imag(c(2,1))))
                E_ML_coded = E_ML_coded + 1;
            end
        end

        % Calculate BER
        BER_ML(L,k) = E_ML/b;
        BER_ZF(L,k) = E_ZF/b;
        BER_MMSE(L,k) = E_MMSE/b;
        BER_ML_coded(L,k) = E_ML_coded/b;
    end

    %% Calculate average BER
    mean_BER_ML = mean(BER_ML(1:L,:),1);
    mean_BER_ZF = mean(BER_ZF(1:L,:),1);
    mean_BER_MMSE = mean(BER_MMSE(1:L,:),1);
    mean_BER_ML_coded = mean(BER_ML_coded(1:L,:),1);

    title({'BER vs SNR',['Loops :
',num2str(L)]},'FontSize',15); % Set the title

    refreshdata
    drawnow
end
```