Multi-Input Multi-output (MIMO) Communication

Date: 2020. 10. 13

The aim of this experiment is that through building MIMO system model, let us understand the basical concepts of MIMO system, such as channel capacity, fading channel, QAM malulation, space-tine coding, etc

1. MIMO system.

Multiple antennas are used at the signal transmitter and receiver: it has powerful performance-enhancing capabilities, especially in the wireless communications, where the channel is impaired predominantly by multi-path fading

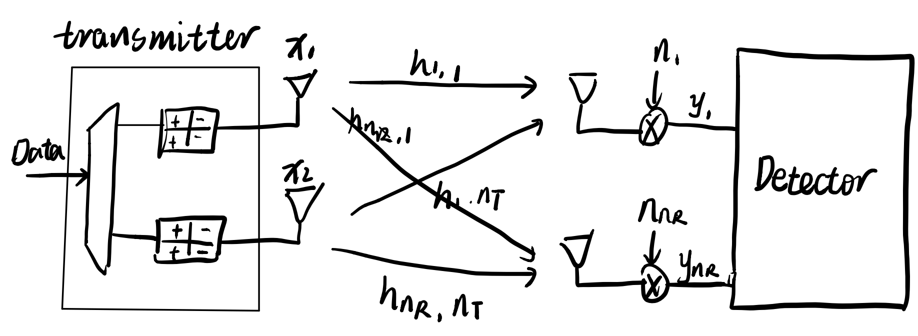
2. Benefits of MIMO technology.

using MIMO technology, we can achieve many gains:

- Array gain
- Spatial diversity gain
- Spatial multiplexing gain.

using MIMO technology can also achieve interference reduction and avoidance.

3. Structure



Hence. the n_R-dimensional received signal y can be written as:

$$y = \sqrt{\frac{\rho}{n_T}}\, Hx + n$$

x: is the n_T-dimensional complex transmitted vector

H: the n_R by n_T complex Gaussian channel matrix with i.i.d entries.

n: the n_R-dimensional i.i.d complex Gaussian noise vector.

And for H. n. have unit variance

x: is unit power constraint

Hence. ρ: average signal to noise ration (SNR) at each receive antenna.

what's more. in this case, signal power at each attenna is $\frac{\rho}{n_T}$, Amplitude is $\sqrt{\frac{\rho}{n_T}}$.

Data. 2020. 10. 15.

There is another important concept:

Capacity, it is a fundamental measure of the maximum amount of information which can be conveyed through a channel reliably.

when $n_R = n_T = n$,

$$C_H = \log_2 \left| I + \frac{\rho}{n_T} H \cdot H^H \right| \quad (bps/Hz)$$

☆. ρ is in dB: $\rho(dB) = 10 \log_{10}(\rho)$

I: Identity matrix

$|\cdot|$ means matrix determinat.

If the channel matrices H are statistically independent over the time

For the ergodic channels, Shannon capacity is equal to statistical average.

$$C_{ergodic} = E_H[C_H]$$

In the Exercise 1:

we have the figure, $n = 1, 2, 3, 4$.

and we try to get the figure when n is $5, 6, \cdots 10$.

H is complex Gaussian channel matrix

n_T is complex transmitted vector

randn( )

according to the question:

we need to use Monte Carlo simulation. and C_H.

a broad class of computational algorithms that rely on repeated random sampling to obtain numerical results.

for each n_R, n_T, SNR. we should generate a large number of H. then calculate C_H and take the average of C_H to get the MIMO capacity.

For the core code:

```
for id = 1:length(nTx)
    sum_rate_simu = zeros(1, length(SNR_dB));
    frame_num = 1e4;
    for n = 1:length(SNR_dB)
        sum_rate = 0;
        for m = 1:frame_num
            gamma_B_inst = Nakagami_m(m_B, SNR(n), nTx(id), nRx(id));
            gamma_B = abs(gamma_B_inst).^2;
            gamma_B_temp = [ ];
            id_ant_temp = [ ];
            gamma_B_ant = sum(gamma_B);
            SNR_b = sum(gamma_B_ant);
            capa = log2(1 + SNR_b);
            sum_rate = sum_rate + capa;
        end
        sum_rate_simu(n) = sum_rate / frame_num;
    end
    figure(1);plot(SNR_dB, sum_rate_simu); hold on;
    xlabel('SNR(dB)');
    ylabel('Capacity(bps/Hz)');
    legend('nr=nt=5','nr=nt=6','nr=nt=7','nr=nt=8','nr=nt=9','nr=nt=10');
end
```

universal Nakagami fading.

```
function h = Nakagami_m(m, gamma_avg, row, col)

    r = gamrnd(m, gamma_avg/m, row, col);  %Amplitude distribution

    o = 2*pi.*rand(row, col);              %Phase distribution

    h = sqrt(r).*exp(1i.*o);
end
```

in our experiment. we can set $m=1$, which means Rayleigh fading

in fact. we can also use another way to get H. as the questions said H is complex Gaussian channel matrix

Hence.

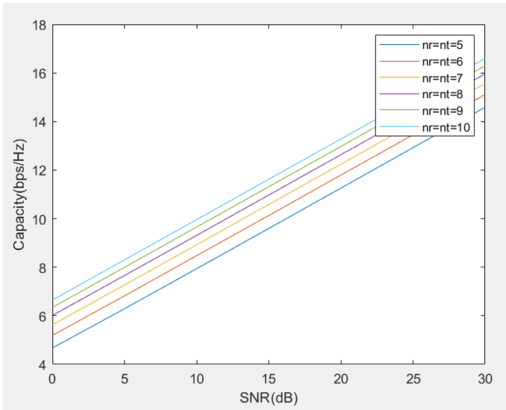
$H = sqrt(0.5) * (randn(n_R, n_T) + j * randn(n_R, n_T))$

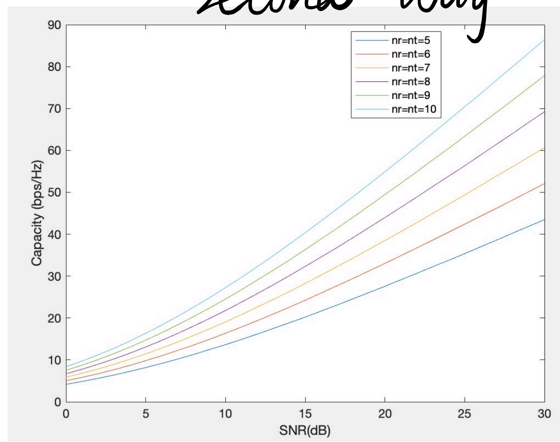
in this way the function gets easier.

Running the code. we can get the results:

after use the two ways to get H, we can get the final results.

First way　　second way





Analysis:

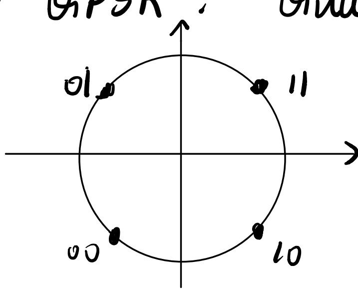
The two shapes have some different. but we can get the same trend. which is more attennas, higher capacity, which means MIMO system can increase the system capacity.

And higher SNR, the higher capacity which means there is higher power to transmit

Date 2020. 10. 16.

For exercise. 2. we need more basical concept of MIMO system.

1. QPSK: Quadrature phase-shift keying



this is a constellation diagram for QPSK. with Gray code. Each adjacent symbol only differs by one bit.

$00: -\frac{\sqrt{2}}{2} - \frac{\sqrt{2}}{2}j$ $\qquad$ $10: \frac{\sqrt{2}}{2} - \frac{\sqrt{2}}{2}j$

$01: -\frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2}j$ $\qquad$ $11: \frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2}j$

2. Receiver detection:

①. ML (maximum-likelihood) detector:

search all possible input x. the compute the distance between the output of possible x. and the practical output. The detector output the x corresponding to the minimum distance.

it is given by

$$\hat{x} = \arg\min_{x \in C} \left\| y - \sqrt{\frac{\rho}{n_T}} H x \right\|$$

The complexing of an ML receiver grows exponentially with Θ and n_T, which restricts its implementation to small signal signal constellation and small number of transmit attenna.

B. ZF (zero-forcing) detector

it is given by

$$\hat{x} = \Theta\{(\sqrt{\frac{\rho}{n_T}} H)^{-1} y\}$$

$$(\sqrt{\frac{\rho}{n_T}} \underline{H})^{-1} \underline{y} = (\sqrt{\frac{\rho}{n_T}} \underline{H})^{-1} (\sqrt{\frac{\rho}{n_T}} \underline{H} \underline{x} + \underline{n})$$

$$= \underline{x} + (\sqrt{\frac{\rho}{n_T}} \underline{H})^{-1} n$$

Hence the spatial interference is completely

```
function [detect] = mapping(xhat)

    %first symbol
    if (real(xhat(1,1))>0)&&(imag(xhat(1,1))>0)
        detect(1)=0;%45 degrees
        detect(2)=0;%45 degrees

    elseif (real(xhat(1,1))<=0)&&(imag(xhat(1,1))>0)
        detect(1)=0;%135 degrees
        detect(2)=1;%135 degrees

    elseif (real(xhat(1,1))<=0)&&(imag(xhat(1,1))<=0)
        detect(1)=1;%225 degrees
        detect(2)=1;%225 degrees

    elseif (real(xhat(1,1))>0)&&(imag(xhat(1,1))<=0)
        detect(1)=1;%315 degrees
        detect(2)=0;%315 degrees
    end
```

removed from the received signal. However ZF results in the noise enhancement that can be seen in the modification of noise covariance matrix.

C. MMSE (minimum mean-square error) detector.

it takes the noise variance into account.

it minimize the mean square error between the detection statistics and the transmitted signal vector

$$\hat{x} = Q\left\{\left(\frac{\rho}{n_T}\underline{H}^H\underline{H} + \underline{I}\right)^{-1}\sqrt{\frac{\rho}{n_T}}\,\underline{H}^H\underline{y}\right\}$$

compared with ZF, MMSE introduces a an $\underline{I}$ which reduce the input of noise.

Data: 2020.10.20.

1. according to the question. we simulate a 2-by-2 MIMO system. with QPSK with gray mapping and produce the performance curves of ML, ZF and MMSE.

2. core code:

for system simulation:

for a 2×2 MIMO system.

$\underline{x}$ has $2^{2\times 2} = 16$ values.

we can use the function to get gray code.

```
        if (real(xhat(2,1))>0)&&(imag(xhat(2,1))>0)
            detect(3)=0;%45 degrees
            detect(4)=0;%45 degrees

        elseif (real(xhat(2,1))<=0)&&(imag(xhat(2,1))>0)
            detect(3)=0;%135 degrees
            detect(4)=1;%135 degrees

        elseif (real(xhat(2,1))<=0)&&(imag(xhat(2,1))<=0)
            detect(3)=1;%225 degrees
            detect(4)=1;%225 degrees

        elseif (real(xhat(2,1))>0)&&(imag(xhat(2,1))<=0)
            detect(3)=1;%315 degrees
            detect(4)=0;%315 degrees
        end
    end
```

QPSK 部分:

```
%QPSK modulation
bit = (randn(1, N_Bit)<0.5);%random 0's and 1's
for n=1:length(bit)/2
    I=bit(2*n-1);
    Q=bit(2*n);

    if (I==0)&&(Q==0)
        qpsk(n)=exp(1i*pi/4);%45 degrees

    elseif (I==0)&&(Q==1)
        qpsk(n)=exp(1i*3*pi/4);%135 degrees

    elseif (I==1)&&(Q==1)
        qpsk(n)=exp(1i*5*pi/4);%225 degrees

    elseif (I==1)&&(Q==0)
        qpsk(n)=exp(1i*7*pi/4);%315 degrees
    end
end
```

For signal transmit and receive:

```
for h=1:number_H
    error_number1=zeros(1,36);
    error_number2=zeros(1,36);
    error_number3=zeros(1,36);

    H = (1/sqrt(2))*(randn(nR,nT)+1i*randn(nR,nT));
    Noise = (1/sqrt(2))*(randn(nR,1)+1i*randn(nR,1));
    for k=1:(N_Symbol/2)

        x=[qpsk(2*k-1);qpsk(2*k)];%transmitted vector
        y=(sqrt(SNR/nT))*H*x+Noise;%received signal
```

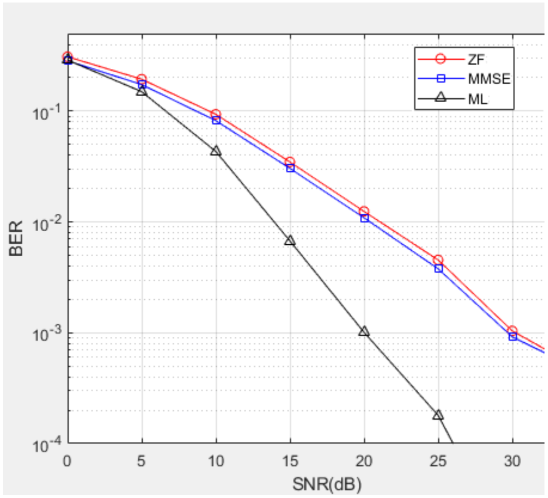
For the detector design. we just need to write the function carefully:

```
        detect1=MLmapping(map,d,e);
        bit_original=[bit(4*k-3),bit(4*k-2),bit(4*k-1),bit(4*k)];
        error_number1(s)=error_number1(s)+sum(bit_original~=detect1);%number of errors

        xhat1=inv(sqrt(SNR/nT)*H)*y;
        detect2=mapping(xhat1);
        error_number2(s)=error_number2(s)+sum(bit_original~=detect2);%number of errors

        xhat2=(inv(SNR/nT*(H')*H+I))*sqrt(SNR/nT)*(H')*y;
        detect3=mapping(xhat2);
        error_number3(s)=error_number3(s)+sum(bit_original~=detect3);%number of errors
    end
    BER1(s)=BER1(s)+error_number1(s)/N_Bit;
    BER2(s)=BER2(s)+error_number2(s)/N_Bit;
    BER3(s)=BER3(s)+error_number3(s)/N_Bit;
end
BER1(s)=BER1(s)/number_H;
BER2(s)=BER2(s)/number_H;
BER3(s)=BER3(s)/number_H;
```

then, we semilogy to plot each curve, we can get:



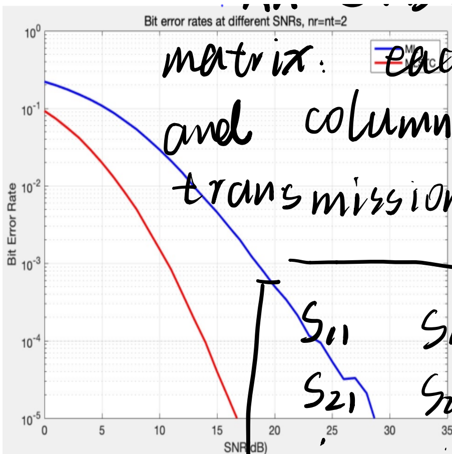
we can see that. under the same SNR. ML detector has the lowest Bit error rate, especially when SNR is large.

ZF has the highest BER. and ZF amplifes the noise. but MMSE reduce the impact of noise on detection. MMSE has a better performance.

Data. 2020. 10. 22.

according to the question, simulate the Alamouti code with QPSK and gray mapping

1. Space-time block coding is the technique used in wireless communications to transmit multiple copies of a data stream across a number of antennas and to exploit the various received versions of the data to improve the reliability of data transfer. An STBC is usually represented by a



matrix: each row represents a time slot and column represents one antenna's transmissions over time.

$$
\begin{array}{c}
\text{transmit antennas} \rightarrow \\
\begin{bmatrix} s_{11} & s_{12} & \cdots & s_{1n_T} \\ s_{21} & s_{22} & & s_{2n_T} \\ \vdots & \vdots & & \vdots \\ s_{T1} & s_{T2} & & s_{Tn_T} \end{bmatrix} \downarrow \text{time-slots}
\end{array}
$$

for Alamounti's code.

$$C_2 = \begin{bmatrix} c_1 & c_2 \\ -c_2^* & c_1^* \end{bmatrix}$$

the core code: with STC.

```
for n=1:length(SNRdB)
    Ctot_MLal=0;
    for m=1:xn/4
        H=sqrt(1/2)*(randn(nr,nt)+1i*randn(nr,nt));
        Xial=[QP_signal(2*m-1),-conj(QP_signal(2*m));QP_signal(2*m),conj(QP_signal(2*m-1))];
        Noise = sqrt(1/2)*(randn(nr,1)+1i*randn(nr,1));      %Use Noise power to generate noise signal
        Yial=sqrt(SNR(n)/nt)*H*Xial+Noise;

        %ML ALDetector
        cpal=zeros(1,16);
        for ii=1:16
            cpal(ii)=sum(abs(Yial(:,1)-sqrt(SNR(n)/nt)*H*code1(:,ii)).^2,1)+sum(abs(Yial(:,2)-sqrt(SNR(n)/nt)*H*code2(:,ii)).^2,1);
        end
        [~,D]=min(cpal);
        Xiout_MLal=code1(:,D);
        CBit_MLal=sum(real(Xiout_MLal)==real(Xial(:,1)))+sum(imag(Xiout_MLal)==imag(Xial(:,1)));
        Ctot_MLal=Ctot_MLal+CBit_MLal;

    end
    BE_MLal = xn-Ctot_MLal;                                  %Calcaulate the number of wrong bits
    BEP_MLal(n)= BE_MLal/xn;
end
```

Hence, we can get the result:

we cem see that ML with Alamouti code have a better preformance.