EXPERIMENT CL Demonstrator: Yan Su

Multi-input multi-output communications

EX.1

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

$$C_H=\log_2\left|I+\frac{\rho}{n_T}HH^H\right|$$

$$C_H=\log_2\left|I+\frac{HR_{xx}H^H}{\sigma_N^2}\right|$$

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

$P_n=1$ for noise with unit variance

$$\rho=\frac{n_T\cdot P_x}{P_n}=n_T\cdot\left(\sqrt{\frac{\rho}{n_T}}\right)^2=\rho$$

Let $y=Hx+n$

$x, n\rightarrow$ i.i.d Gaussian

For Gaussian $P(x)=\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left[-\frac{(x-m)^2}{2\sigma^2}\right]$

For SISO

$$H(x)=-\int_{-\infty}^{+\infty}p(x)\log p(x)\,dx$$

$$=E[-\log p(x)]$$

$$=E\left[-\log\left(\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left[-\frac{(x-m)^2}{2\sigma^2}\right]\right)\right]$$

$$=\frac{1}{2}\log 2\pi\sigma^2+E\left[\frac{(x-m)^2\log e}{2\sigma^2}\right]$$

$$\frac{1}{2\sigma^2}E[(x-m)^2]\log e=\frac{1}{2\sigma^2}\cdot\sigma^2\cdot\log e$$

$$=\frac{1}{2}\log e$$

$$H(x)=\frac{1}{2}\log 2\pi\sigma^2+\frac{1}{2}\log e$$

$$=\frac{1}{2}\log 2\pi e\sigma^2$$

For MIMO

For n-dimensionality Gaussian $X=(x_1,x_2,\cdots,x_N)^T$

$$P_X(x_1,x_2,\cdots,x_N)=\frac{1}{\sqrt{(2\pi)^N|\Sigma|}}\exp\left[-\frac{1}{2}(X-\mu)^T\Sigma^{-1}(X-\mu)\right]$$

$$\Sigma=\sigma_x^2 I_N,\quad |\Sigma|=\prod_{i=1}^{N}\sigma_i^2,\quad \mu=(\mu_1,\cdots,\mu_N)^T$$

$$P_X(x_1,x_2,\cdots,x_N)=\prod_{i=1}^{N}f(x_i)$$

$$H(X)=-\int_{-\infty}^{+\infty}P_X(x_1,x_2,\cdots,x_N)\cdot\log P_X(x_1,x_2,\cdots,x_N)\,dx_1\,dx_2\cdots dx_N$$

$$=-E\left[\log\left(\frac{1}{\sqrt{(2\pi)^N|\Sigma|}}e^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)}\right)\right]$$

$$=\frac{1}{2}\log(2\pi)^N|\Sigma|+E\left[\frac{1}{2}(X-\mu)^T\Sigma^{-1}(X-\mu)\cdot\log e\right]$$

$$=\frac{1}{2}\log(2\pi)^N|\Sigma|+\frac{1}{2}\log e\sum_{i=1}^{N}E\left[\frac{(x_i-\mu_i)^2}{\sigma_i^2}\right]$$

$$=\frac{1}{2}\log(2\pi)^N|\Sigma|+\frac{1}{2}\log e\times N$$

$$=\frac{1}{2}\log(2\pi e)^N|\Sigma|$$

```matlab
clc;
clear all;
close all;
%% basic parameters
N = 5:10;
N_ave = 10000;
SNR_dB = 0:30;
SNR = power(10, SNR_dB/10);
C_buff= zeros(1,N_ave);
C_ergodic = zeros(size(N,2), size(SNR,2));
%% calculate the capacity
for n = 1:size(N,2)
    for k = 1:size(SNR,2)
        for i=1:N_ave
            H = sqrt(1/2) * (randn(N(n)) + 1i*randn(N(n)));
            C_H = det(eye(N(n)) + SNR(k) / N(n) * H * H');
            C_H = log2(C_H);
            C_buff(i) = C_H;
        end
        % get average value
        C_ergodic(n,k) = real(mean(C_H));
    end
end

%% plot
figure(1);
ylabel('Capacitty(bps/Hz)');
xlabel('SNR(dB)');
labels = strings(1, size(N,2));
for n = 1:size(N,2)
    plot(SNR_dB, C_ergodic(n,:));
    hold on;
    labels(n) = ['n_t = n_r = ' num2str(N(n))];
end
legend(labels);
```

Complex Gaussian

$$\Sigma_X = R_{xx} = E[X X^H]$$

$$\Sigma_N = R_{nn} = E[N N^H] = \sigma_N^2 I_N$$

$$\Sigma_Y = E[Y Y^H] = H R_{xx} H^H + \sigma_N^2 I_N$$

$$\boxed{C(H) = \max I(x;y)}$$

$$= H(Y) - H(Y|X)$$

$$= H(Y) - H(N)$$

$$= \frac{1}{2}\log(2\pi e)^N|\Sigma_Y| - \frac{1}{2}\log(2\pi e)^N|\Sigma_N|$$

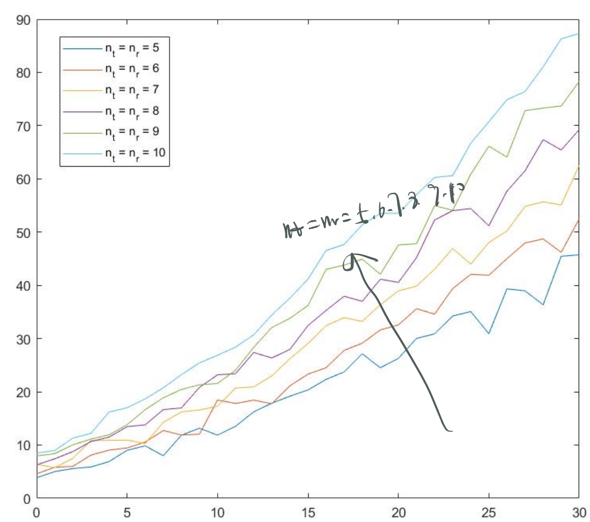
$$= \frac{1}{2}\log\frac{|\Sigma_Y|}{|\Sigma_N|}$$

$$= \frac{1}{2}\log\frac{|H R_{xx} H^H + \sigma_N^2 I_N|}{|\sigma_N^2 I_N|}$$

$$= \frac{1}{2}\log\left|I_N + \frac{H R_{xx} H^H}{\sigma_N^2}\right|$$

$$C = \frac{1}{T}C(H) = 2B \cdot C(H)$$

$$= B\log\left|I_N + \frac{H R_{xx} H^H}{\sigma_N^2}\right|$$

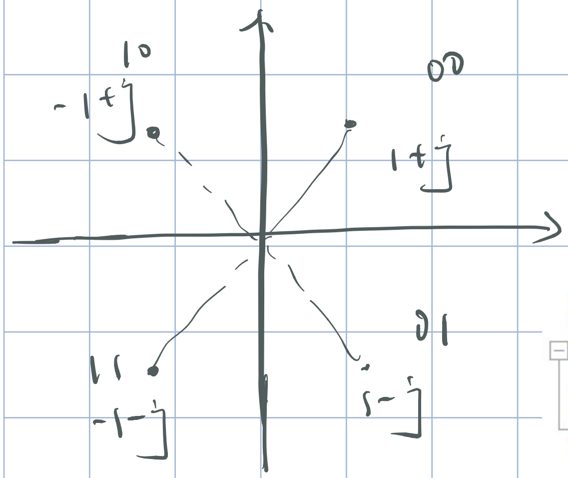


As the Monte Carlo simulation is used to to calculate the statistical average, the smooth line cannot be obtained.

However, the tendency is still clear. With the increasement of antenna number N_T and N_r, the capacity also increases in the conditions of different SNRs

EX.2

QPSK with Gray Coding



$s \to b_1 b_2$

Gray coding:

$$\boxed{(-2b_1+1) + j(-2b_2+1)}$$

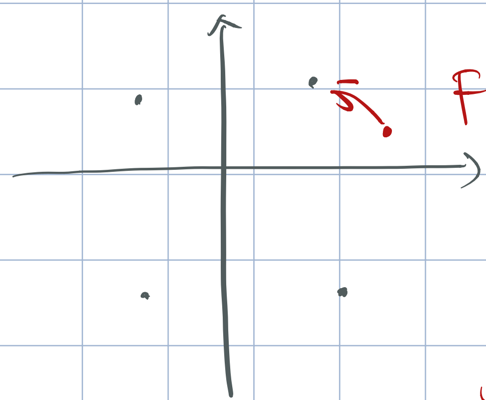
```
%% QPSK function
function result = QPSK_map(bit_1, bit_2)
result = sqrt(1/2) * ((-2*bit_1+1) + (-2*bit_2+1)*1i); % Gray coding
end
```

Maximum likelihood (ML)

$$\hat{x} = \arg\max_{x \in C} p(y|x)$$

$$= \arg\min_{x \in C} \left\| y - \sqrt{\frac{\rho}{n_T}} H x \right\|^2$$

Try all the possible x and find the most suitable one.



Find the closest point

$Q\{\cdot\}$ Quantization rule?

the nearest one

$$y = \sqrt{\frac{\rho}{n_T}} H x + n$$

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

$$n \approx 0 \Rightarrow x = \left(\sqrt{\frac{\rho}{n_T}} H\right)^{-1} y$$

Zero-forcing (ZF)

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

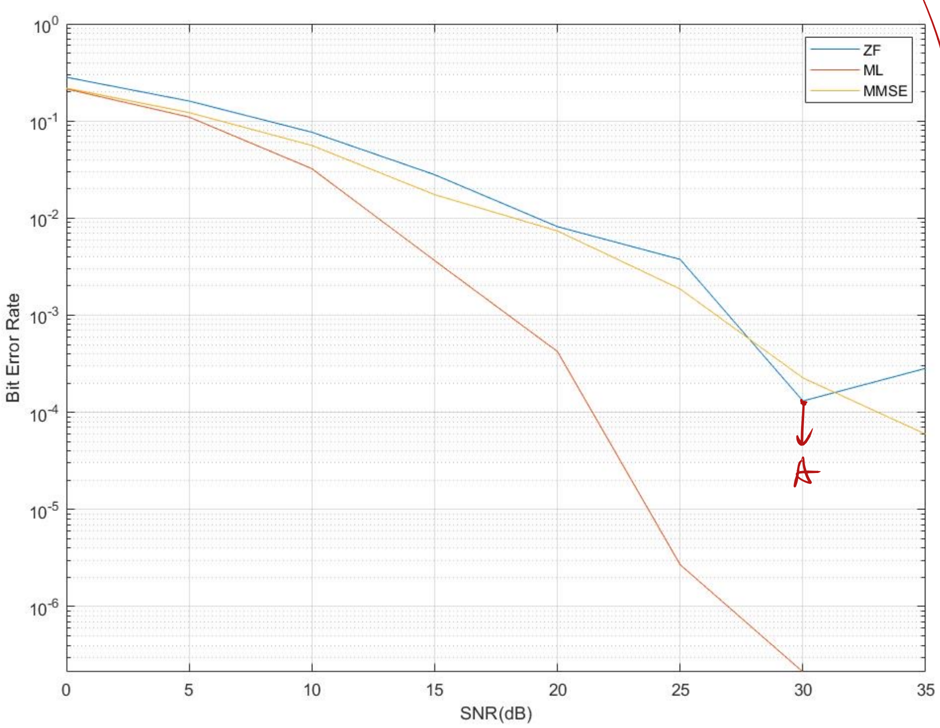
Minimum Mean Square Error (MMSE)

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

take noise into account

~~300 H matrix~~
~~1,000,000 bits~~

Take too much time on my laptop. around 50 h by estimation



Due to the limited performance, I only generate 200 H matrices, each carrying 100,000 bits, which are much lower than the recommended parameters. So, there is an anomaly point A in the figure.

But, in the figure, we can still obvious that ML has the best performance in terms of BER. MMSE is the second one and ZF is the last one.

While in term of computation complexity, ML is the worst. In this experimence, ML takes a very long time to do the detection

EX.3

Space-time coding

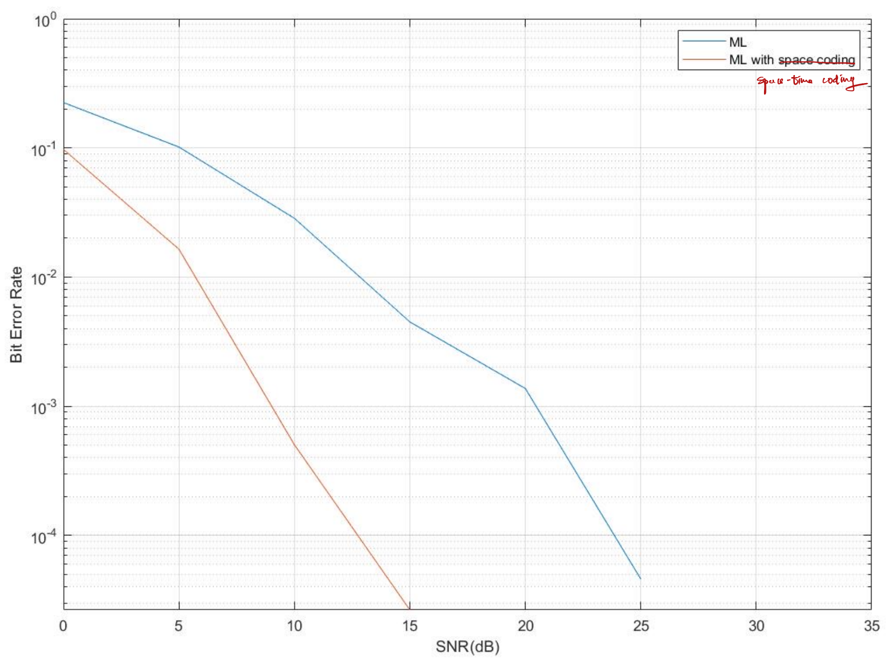
$$X = \begin{bmatrix} x_1 & -x_2^* \\ x_2 & x_1^* \end{bmatrix}$$

$$Y = \sqrt{\frac{P}{n_T}} H X + \underset{\sim}{N} \longrightarrow$$ note that N is also 2×2 matrix

Also use the ML detector

$$\hat{x} = \underset{(x_1, x_2)\in C}{\arg\min} \underbrace{\left\| Y - \sqrt{\frac{P}{n_T}} H X \right\|_F^2}_{\text{Frobenius norm}}$$

In MATLAB, Frobenius norm can be achieved by using function $\boxed{\texttt{norm(N,'fro');}}$



Similar to EX 2, I only use 200 H matrix and 10,000 bits for this experiment due to the performance limitation of my laptop.

In this figure, it can be noticed that by using the space-time coding, the performance of ML detector is highly improved