Communication Systems Design

Lab 4: MIMO Transmission System

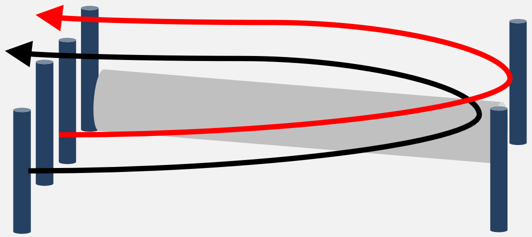
Dr. Wu Guang
wug@sustech.edu.cn
Electrical & Electronic Engineering
Southern University of Science and Technology

前沿通信系统设计
1 WiFi通信系统（6周）
实验目标：利用USRP实现802.11a/n图像传输
软件：MATLAB, 硬件：USRP
授课内容：MATLAB通信编程、USRP文本传输、MIMO系统、802.11a/n仿真、802.11a/n图像传输
2 5G/4G-LTE系统（5周）
实验目标：利用USRP实现LTE图像传输
软件：MATLAB，硬件：USRP
授课内容：小区搜索过程、MIB/SIB解码过程、LTE图像传输、LDPC编解码过程、srsLTE系统
3 无线网络传输系统（3周）
实验目标：利用Telos实现无线多跳网络传输数据
软件：TinyOS、NesC
授课内容：TinOS编程、MICAZ平台介绍、无线多跳网络数据收集、无线信道建模、无线定位、路由和数据收集
4 雷达感知系统（2周）
实验目标：利用KerberosSDR实现测向
软件：MATLAB，硬件：KerberosSDR、树莓派
授课内容：MUSIC算法、空间谱估计、KerberosSDR原理，无线开源项目，课程Presentation

Wireless Communication Base-station

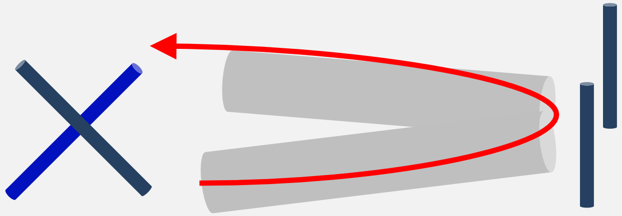
Multi Input Multi Output: MIMO

- Scenario A (Open area)



线性天线

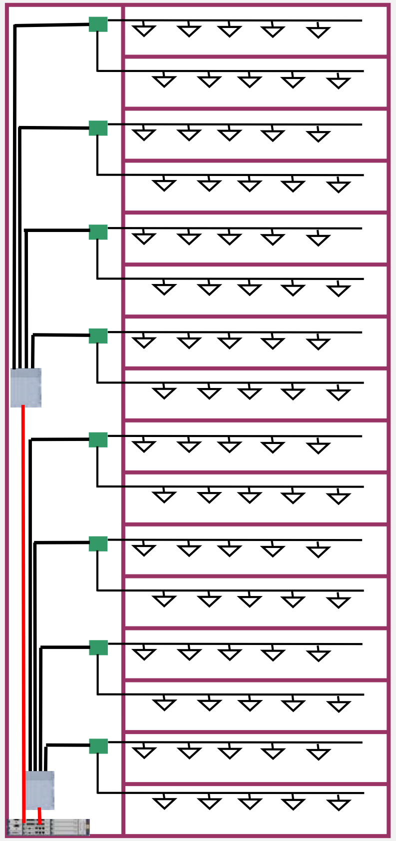
- Scenario B (City)



交叉极化

- Scenario C

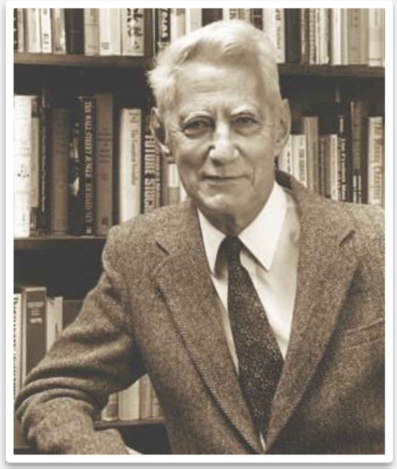
  (Indoor)



Why MIMO？

Beyond Shannon–Hartley theorem

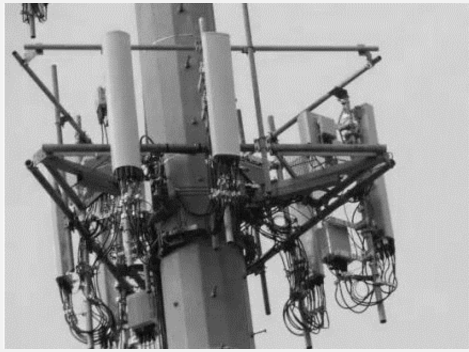
$$C = B\log_2(1 + \frac{S}{N})$$



C： the channel capacity in bits per second
B： the bandwidth of the channel in hertz
S/N： the signal-to-noise ratio (SNR)

How about MIMO systems ?

MIMO: Multi-Input and Multi-Output



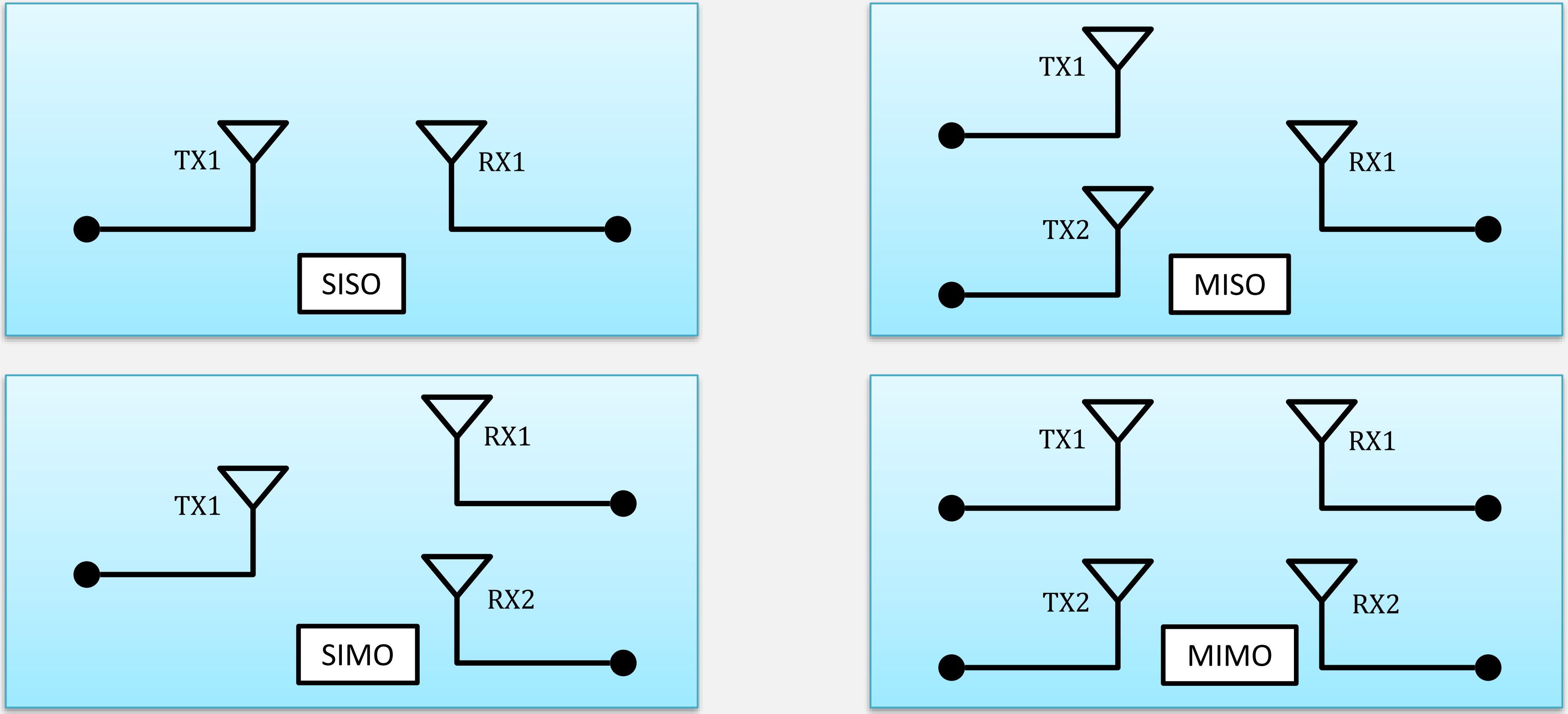
$$C = B\log_2(1 + \frac{S}{N})$$

$$\Downarrow$$

$$C = \mathbf{n}B\log\left(1 + \frac{h^2 S}{N}\right)$$

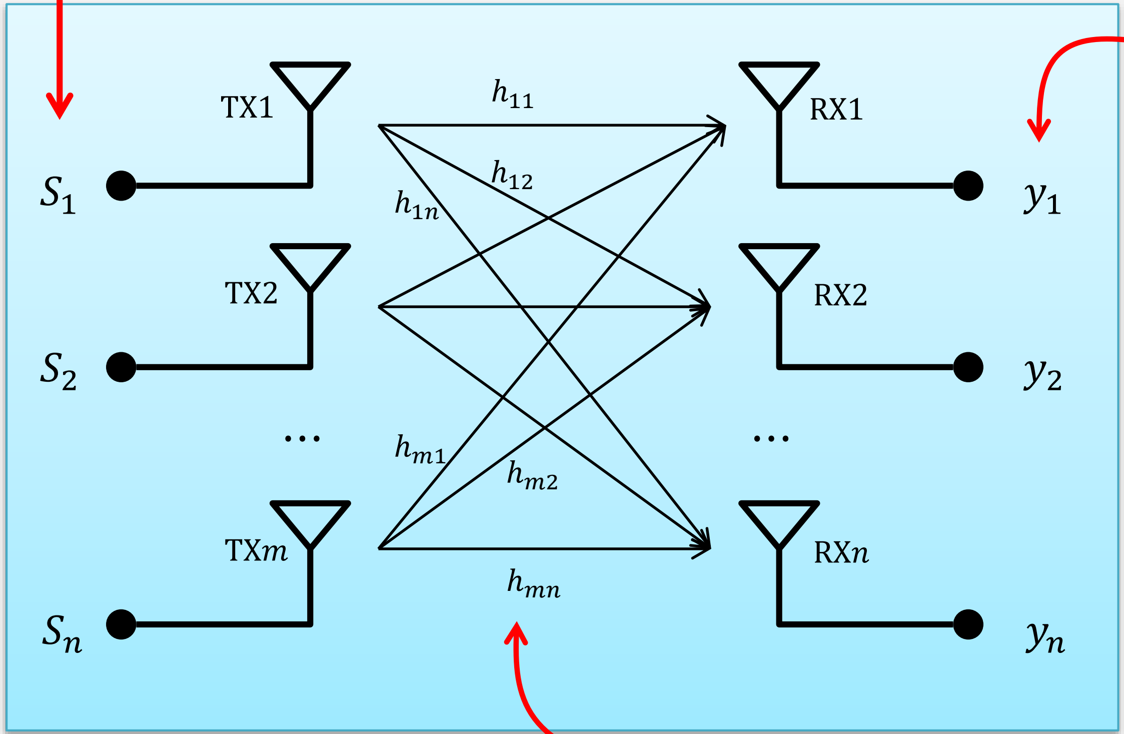
Both the transmitter and the receiver have n antennas. h is orthogonal channel matrix.

Diversity



TX1
RX1
SISO
TX1
RX1
TX2
MISO
RX1
TX1
RX2
SIMO
TX1
RX1
TX2
RX2
MIMO

MIMO transmission model



$$Y = Hs + n$$

Noise

$$\begin{bmatrix} h_{11} & \cdots & h_{m1} \\ \vdots & \ddots & \vdots \\ h_{1n} & \cdots & h_{mn} \end{bmatrix}$$

- Zero-forcing
- MMSE
- Pre-coding

MIMO Pre-coding

Example: 2 X 2 MIMO Precoding

Step 1: $\begin{bmatrix} r_1 \\ r_2^* \end{bmatrix} = \mathbf{H} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2^* \end{bmatrix}$

$\mathbf{H} = \mathbf{U \Sigma V^H}$, $\quad \boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\lambda_1} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\lambda_2} \end{bmatrix} \quad \mathrm{U^H U} = \mathbf{I_n}$, $\mathbf{V^H V} = \mathbf{I_m}$

Step 2: $\begin{bmatrix} r_1 \\ r_2^* \end{bmatrix} = \mathbf{U} \begin{bmatrix} \boldsymbol{\lambda_1} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\lambda_2} \end{bmatrix} \mathbf{V^H} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2^* \end{bmatrix}$

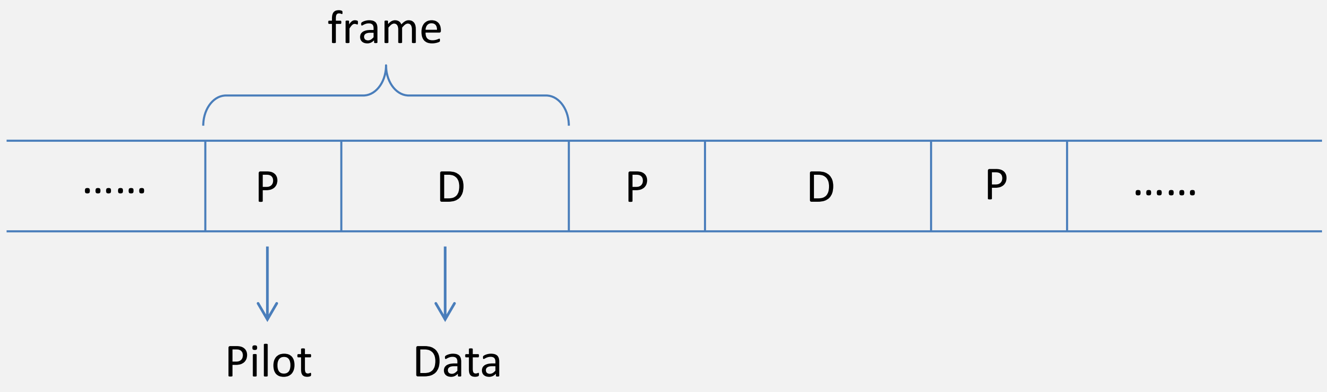
$\mathbf{s}' \rightarrow \mathbf{s},$

$\begin{bmatrix} \boldsymbol{s_1} \\ \boldsymbol{s_2} \end{bmatrix} = \mathbf{V} \begin{bmatrix} \boldsymbol{s_1'} \\ \boldsymbol{s_2'} \end{bmatrix}$

Step 3: $\begin{bmatrix} r_1 \\ r_2^* \end{bmatrix} = \mathbf{U} \begin{bmatrix} \boldsymbol{\lambda_1} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\lambda_2} \end{bmatrix} \mathbf{V^H V} \begin{bmatrix} \boldsymbol{s_1'} \\ \boldsymbol{s_2'} \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2^* \end{bmatrix} \Rightarrow \begin{bmatrix} r_1 \\ r_2^* \end{bmatrix} = \mathbf{U} \begin{bmatrix} \boldsymbol{\lambda_1} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\lambda_2} \end{bmatrix} \begin{bmatrix} \boldsymbol{s_1'} \\ \boldsymbol{s_2'} \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2^* \end{bmatrix}$

$\Rightarrow \mathrm{U^H} \begin{bmatrix} r_1 \\ r_2^* \end{bmatrix} = \begin{bmatrix} \boldsymbol{\lambda_1} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\lambda_2} \end{bmatrix} \begin{bmatrix} \boldsymbol{s_1'} \\ \boldsymbol{s_2'} \end{bmatrix} + \mathrm{U^H} \begin{bmatrix} n_1 \\ n_2^* \end{bmatrix}$

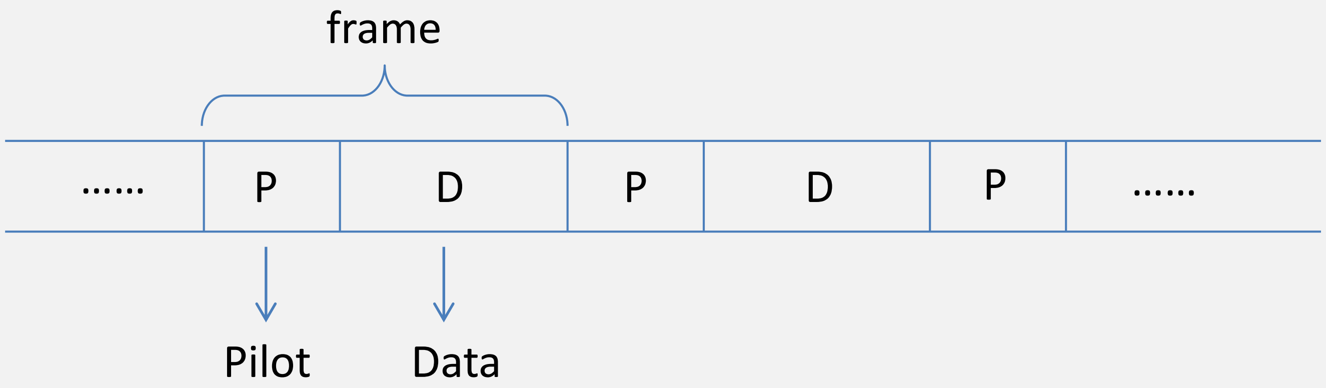
Zero-forcing Algorithm



Assume that channels are held constant for the whole frame.

$$\boxed{\hat{\mathbf{S}}_{ZF} = \left(\mathbf{H}^{\mathrm{H}}\mathbf{H}\right)^{-1}\mathbf{H}^{\mathrm{H}}\boldsymbol{r}_{\mathrm{D}}}$$

MMSE: Minimum Mean Squared Error

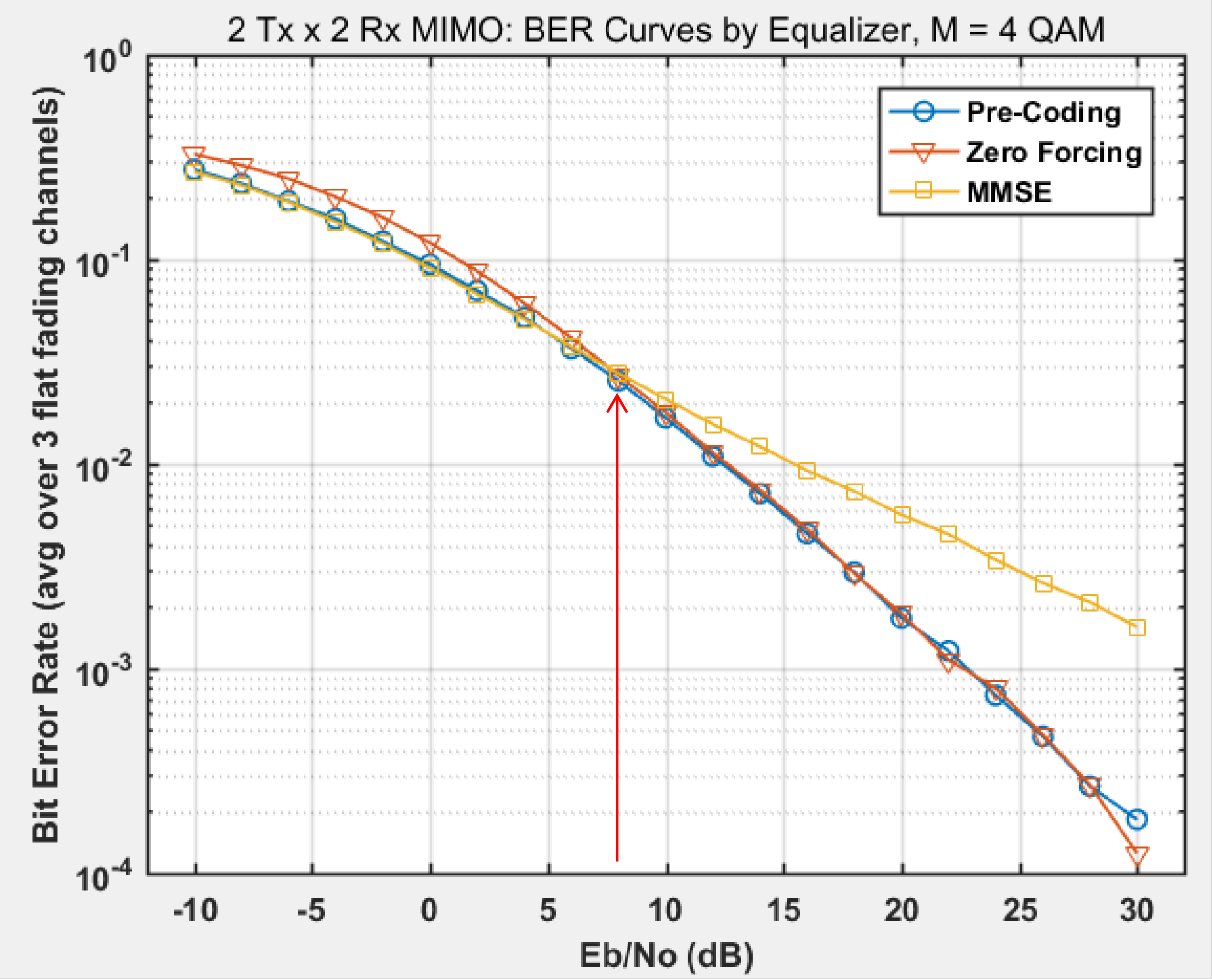


Assume that channels are held constant for the whole frame.

$$\boxed{\hat{\mathrm{S}}_{MMSE} = \left(\mathrm{H}^{\mathrm{H}}\mathrm{H} + \frac{\sigma_n^2}{\sigma_s^2} I\right)^{-1} \mathrm{H}^{\mathrm{H}} r_{\mathrm{D}}}$$

Pre-Coding/Zero-forcing/MMSE

```
%% MIMO
% model a 2x2 MIMO link with flat fading gains and 3 equalizer schemes:
% Pre-coding, Zero-forcing and MMSE
% ** Pre-coding has perfect CSIT, where Zero-forcing and MMSE has CSIR

clear; close; clc;
%% Parameter Setup
M = 4;              % modulation order--------------------->调制阶数
k = log2(M);        % coded bits per symbol---------------->单符号传递比特数
nSyms = 1e4;        % number of symbols to send------------>发送符号总数
nBits = nSyms * k;  % number of Bitsls to send------------->发送比特总数

nChan = 3;          % number of flat fading MIMO channels-->平坦衰落信道
EbNo = -10:2:30;    % Eb/No-------------------------------->信噪比
snrVector = EbNo + 10*log10(k); % Es/No before adding noise

% 2 x 2 MIMO channel
Mt = 2;             % ------------------------------------>发射机天线数
Mr = 2;             % ------------------------------------>接收机天线数

% initialize
berZeroForcing = zeros(nChan, length(snrVector));%-------->迫零算法
berMMSE = zeros(nChan, length(snrVector));%--------------->MMSE(最小均方)

```



```matlab
%% Transmit Precoding and Receiver Shaping Scheme
% Reference
% Goldsmith, $Wireless\;Communications$ [pp. 323-324]

% Transmit precoding: x = V*(x_hat)
% Receiver shaping: (y_hat) = (U_hermitian_transposed)*y

[berPreCoding]=PreCoding(M,nBits,nChan,snrVector,Mt,Mr);%-->需要编写的函数

```

Step1： Parameter Setup

```matlab
%% model a 2x2 MIMO link and a equalizer scheme: Pre-coding
function [berPreCoding]=PreCoding(M,nBits,nChan,snrVector,Mt,Mr)

%% 1: Parameter Setup
berPreCoding = zeros(nChan, length(snrVector));%---------->初始化误码率
% Transmit precoding: x = V*(x_hat)
% Receiver shaping: (y_hat) = (U_hermitian_transposed)*y

U = zeros(Mr, Mt, nBits);                        %----------->初始化U矩阵
S = zeros(Mr, Mt, nBits);                        %----------->初始化S矩阵
V = zeros(Mr, Mt, nBits);                        %----------->初始化V矩阵
prefiltered = zeros(Mt, 1, nBits);               %----------->初始化编码前矩阵
txData = zeros(Mt, 1, nBits);                    %----------->发送数据矩阵
rxData = zeros(Mr, 1, nBits);                    %----------->接收数据矩阵
```

Step2： MIMO Pre-Coding Programming

```
%% 2: MIMO precoding
disp('MIMO precoding');
for i = 1:nChan
    fprintf('Channel: %d\n', i);
    % unique MIMO channel for 'Mr' receive and 'Mt' transmit antennas
    H = ( randn(Mr, Mt, nBits) + 1j*randn(Mr, Mt, nBits) ) / sqrt(2);

    % generate a sequence of random message bits and QAM modulate
    data = randi([0 M-1], Mt, 1, nBits);
    dataMod = qammod(data, M);

    % precode
    for bit = 1:nBits
        % decompose channel matrix H by SVD--------------->奇异值分解
        [U(:,:,bit), S(:,:,bit), V(:,:,bit)] = svd(H(:,:,bit));
        % pre-code data for each bit: (x = V * x_hat)----->预编码
        prefiltered(:,:,bit) = V(:,:,bit) * dataMod(:,:,bit);
        % send over the fading channel             -------->信道上发送
        txData(:,:,bit) = H(:,:,bit) * prefiltered(:,:,bit);
    end
```

```
    fprintf('SNR:\t');
    for j = 1:length(snrVector)
        fprintf('%d\t', j);
        % add white Gaussian noise (x_noisy <-- x + noise)
        % for double-sided white noise, (y_hat = U^(H) * y)
        noise = randn(Mr, 1, nBits) + 1j*randn(Mr, 1, nBits) / sqrt(2);
        txNoisy = txData +  noise * 10^(-snrVector(j)/10/2);
        for bit = 1:nBits
            % post-code data for each bit: remove fading channel components
            % --------------------------------------------->数据恢复
            rxData(:,:,bit) = U(:,:,bit)' * txNoisy(:,:,bit);
        end

        % QAM demodulate and compute bit error rate
        rxData = qamdemod(rxData,M);                %-------->QAM解调
        [~,berPreCoding(i,j)] = biterr(data, rxData);
    end
    fprintf('\n');
end

%% 3: Take average of all 3 fading channels
berPreCoding = mean(berPreCoding);   %-------------------->计算平均误码率
```

Step2： Mean BER Calculation

```matlab
for j = 1:length(snrVector)
    fprintf('%d\t', j);
    % add white Gaussian noise (x_noisy <-- x + noise)
    % double-sided white noise, (y_hat = U^(H) * y)
    noise = randn(Mr, 1, nBits) + 1j*randn(Mr, 1, nBits) / sqrt(2);
    txNoisy = txData +  noise * 10^(-snrVector(j)/10/2);

    for bit = 1:nBits
        % (1) W_{zf} = H_{Pseudoinverse} = (H^{H} * H)^{-1} * H^{H}
        W(:, :, bit) = (H(:, :, bit)' * H(:, :, bit))^-1 * H(:, :, bit)';
        rxData(:, :, bit) = W(:, :, bit) * txNoisy(:, :, bit);
        % (2) or simply solve linear system H*x = y for x, if full rank
        % rxData(:, :, bit) = H(:, :, bit) \ txNoisy(:, :, bit);
    end

    % QAM demodulate and compute bit error rate
    rxData = qamdemod(rxData, M);
    [~, berZeroForcing(i, j)] = biterr(data, rxData);
end
```

Zero-forcing

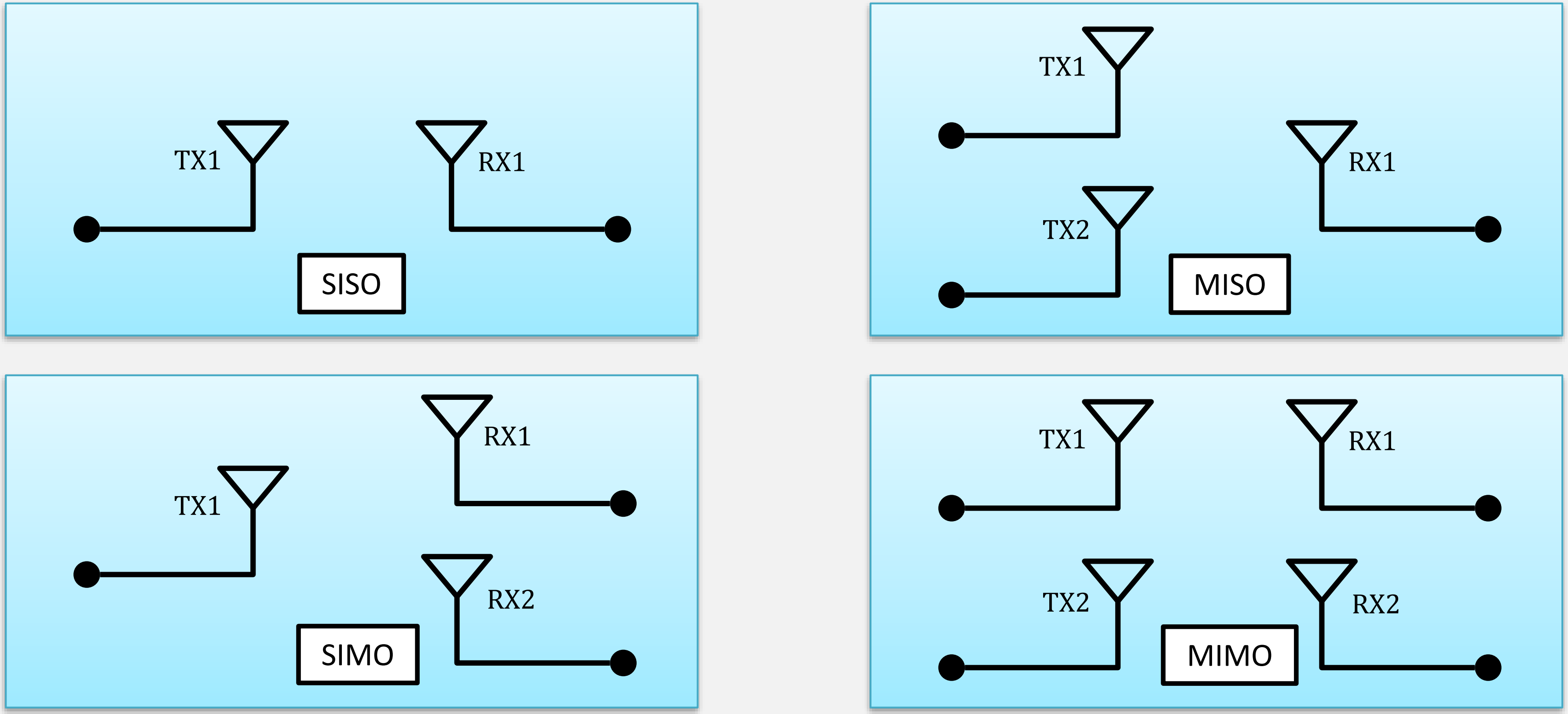
$$\boxed{\hat{\mathbf{S}}_{ZF} = \left(\mathbf{H}^{\mathbf{H}}\mathbf{H}\right)^{-1}\mathbf{H}^{\mathbf{H}}\boldsymbol{r}_{\mathbf{D}}}$$

```
for j = 1:length(snrVector)
    fprintf('%d\t', j);
    % add white Gaussian noise (x_noisy <-- x + noise)
    % for double-sided white noise,  (y_hat = U^(H) * y)
    noise = randn(Mr, 1, nBits) + 1j*randn(Mr, 1, nBits) / sqrt(2);
    txNoisy = txData +  noise * 10^(-snrVector(j)/10/2);

    for bit = 1:nBits
        % add noise variations
        W(:, :, bit) = (H(:, :, bit)' * H(:, :, bit) + ...
                   + eye(Mt)*10^(-snrVector(j)/10/2))^-1 * H(:, :, bit)';
        rxData(:, :, bit) = W(:, :, bit) * txNoisy(:, :, bit);
    end

    % QAM demodulate and compute bit error rate
    rxData = qamdemod(rxData, M);
    [~, berMMSE(i, j)] = biterr(data, rxData);
end
```

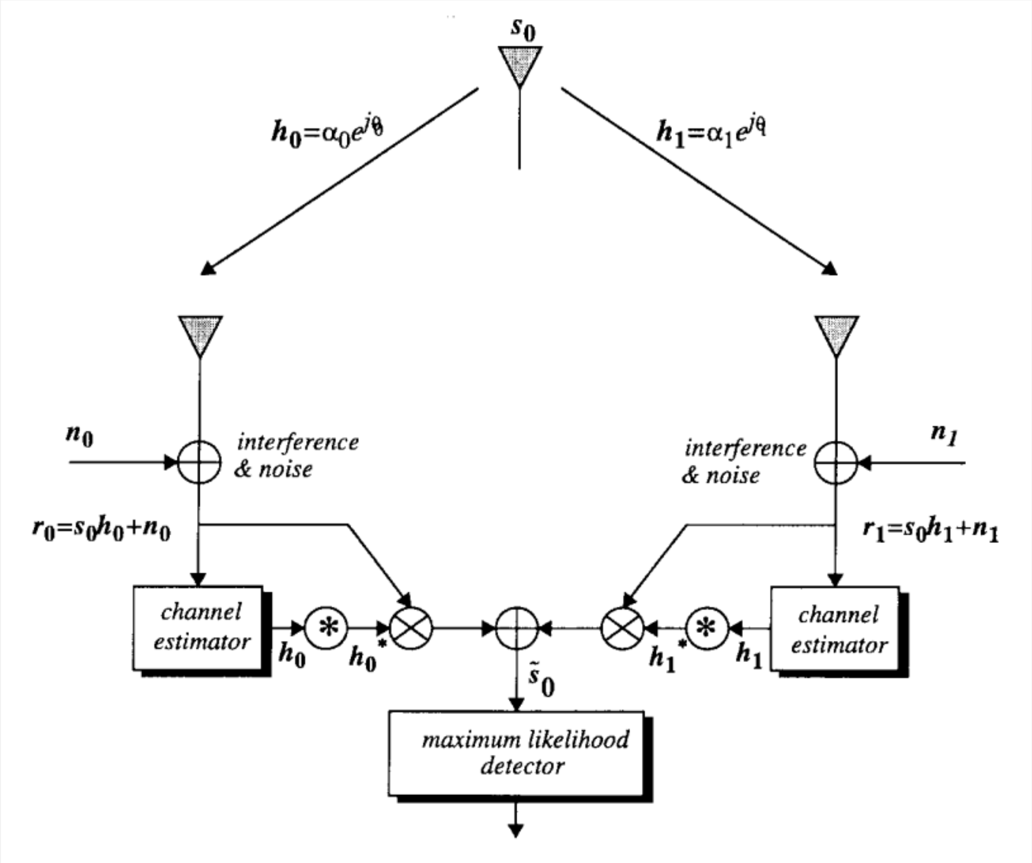
MMSE

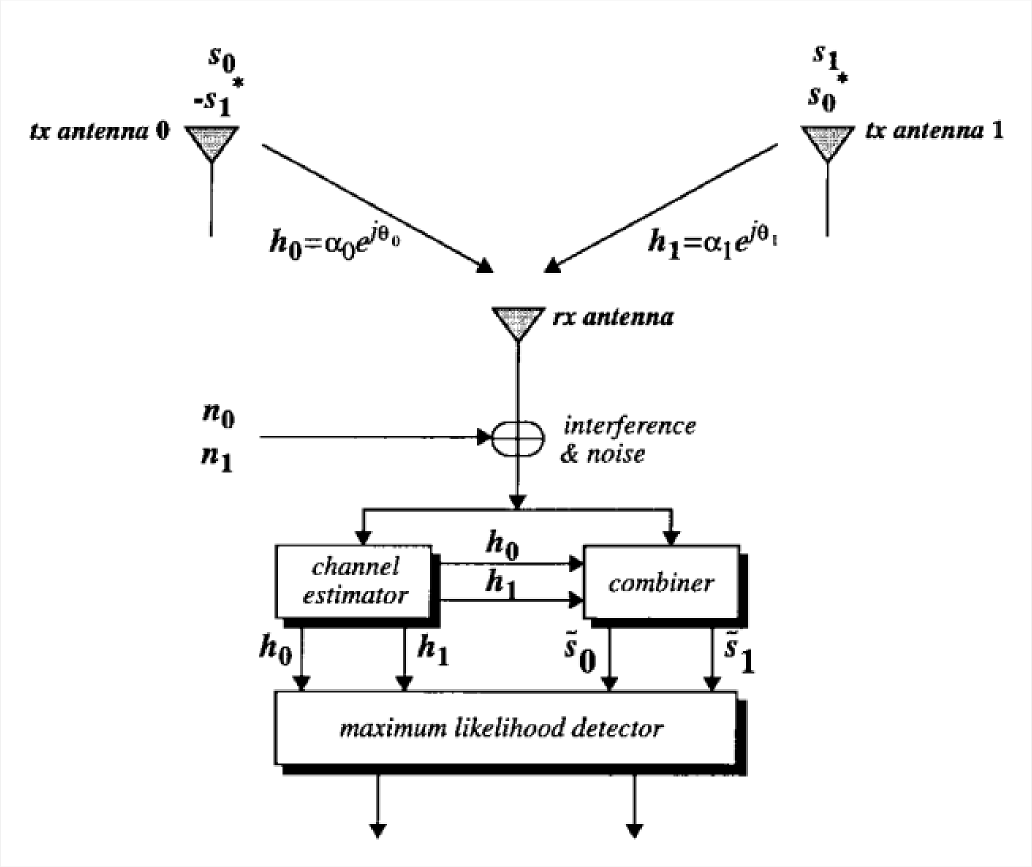
$$\hat{\mathbf{S}}_{MMSE} = \left(\mathbf{H}^{\mathrm{H}}\mathbf{H} + \frac{\sigma_n^2}{\sigma_s^2} I\right)^{-1} \mathbf{H}^{\mathrm{H}} r_{\mathrm{D}}$$

CSIT or CSIR ?


TX1
RX1
SISO
TX1
RX1
TX2
MISO
RX1
TX1
RX2
SIMO
TX1
RX1
TX2
RX2
MIMO

STBC: Space Time Block Code

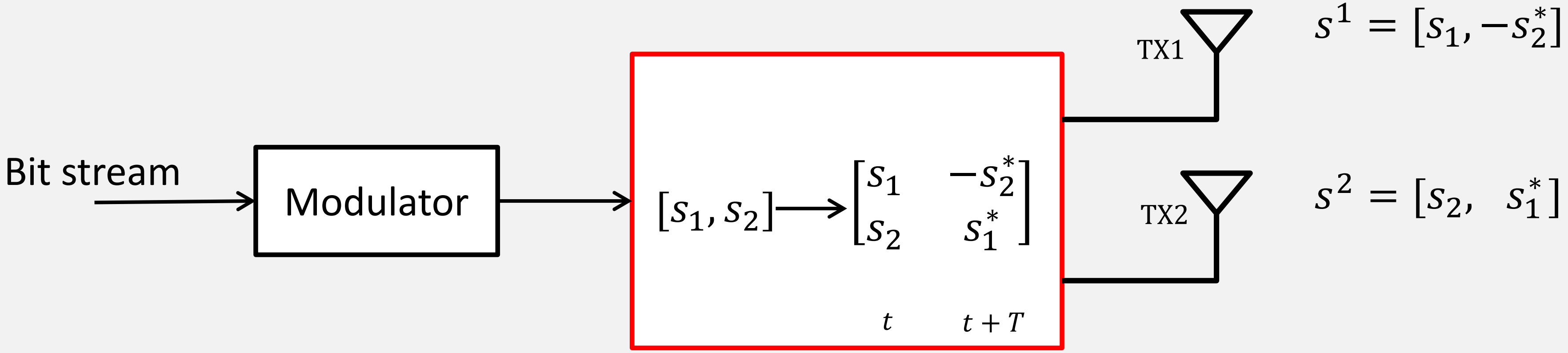




A Simple Transmit Diversity Technique for Wireless Communications, Siavash M. Alamouti, IEEE JOURNAL ON SELECT AREAS IN COMMUNICATIONS, VOL. 16, NO. 8, OCTOBER 1998.

Alamouti: Transmit Diversity

Alamouti



$$\mathbf{C} = \begin{bmatrix} s_1 & -s_2^* \\ s_2 & s_1^* \end{bmatrix} \qquad \mathbf{CC^H} = \begin{bmatrix} s_1 & -s_2^* \\ s_2 & s_1^* \end{bmatrix} \begin{bmatrix} s_1^* & s_2^* \\ -s_2 & s_1 \end{bmatrix} = (|s_1|^2 + |s_1|^2)\mathbf{I_2}$$

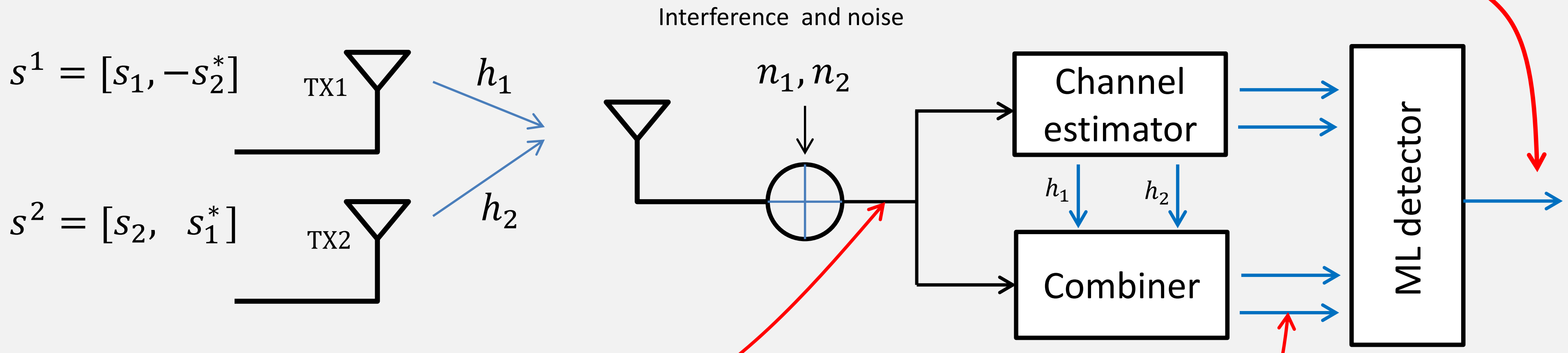
Identity matrix

A Simple Transmit Diversity Technique for Wireless Communications, Siavash M. Alamouti, IEEE JOURNAL ON SELECT AREAS IN COMMUNICATIONS, VOL. 16, NO. 8, OCTOBER 1998.

Alamouti 2X1

Alamouti

$$\min_{s_1 \in \{S_i\}} |s_1 - \tilde{s}_1| ; \min_{s_2 \in \{S_i\}} |s_2 - \tilde{s}_2|$$



$$\begin{cases} r_1 = h_1 s_1 + h_2 s_2 + n_1 \\ r_2 = -h_1 s_2^* + h_2 s_1^* + n_2 \end{cases}$$

$$\begin{cases} \tilde{s}_1 = (h_1^* r_1 + h_2 r_2^*) = (|h_1|^2 + |h_2|^2) s_1 + h_1^* n_1 + h_2 n_2^* \\ \tilde{s}_2 = (h_2^* r_1 - h_1 r_2^*) = (|h_1|^2 + |h_2|^2) s_2 - h_1 n_2^* + h_2^* n_1 \end{cases}$$

Matrix representation

$$\begin{bmatrix} r_1 \\ r_2^* \end{bmatrix} = \begin{bmatrix} h_1 & h_2 \\ h_2^* & -h_1^* \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2^* \end{bmatrix}$$

$$\mathbf{r} = \begin{bmatrix} r_1 \\ r_2^* \end{bmatrix} \quad \mathbf{H} = \begin{bmatrix} h_1 & h_2 \\ h_2^* & -h_1^* \end{bmatrix}$$

$$\mathbf{S} = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} \quad \mathbf{n} = \begin{bmatrix} n_1 \\ n_2^* \end{bmatrix}$$

$$\mathbf{r} = \mathbf{HS} + \mathbf{n}$$

$$\Downarrow$$

$$\boxed{\mathbf{H^H r} = \mathbf{S} + \mathbf{H^H n}}$$

$$\Updownarrow$$

$$\boxed{\begin{bmatrix} h_1^* & h_2 \\ h_2^* & -h_1 \end{bmatrix} \begin{bmatrix} r_1 \\ r_2^* \end{bmatrix} = (|h_1|^2 + |h_2|^2) \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} + \begin{bmatrix} h_1^* n_1 & h_2 n_2^* \\ h_2^* n_1 & -h_1 n_2^* \end{bmatrix}}$$

$$\boxed{\begin{cases} r_1 = h_1 s_1 + h_2 s_2 + n_1 \\ r_2 = -h_1 s_2^* + h_2 s_1^* + n_2 \end{cases}}$$

MIMO Pre-coding

Example: 2 X 2 MIMO Precoding

Step 1: $\begin{bmatrix} r_1 \\ r_2^* \end{bmatrix} = \mathbf{H} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2^* \end{bmatrix}$

$\mathbf{H} = \mathbf{U\Sigma V^H}$, $\quad \boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\lambda_1} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\lambda_2} \end{bmatrix} \quad \mathrm{U^H U} = \mathbf{I_n}$, $\mathbf{V^H V} = \mathbf{I_m}$

Step 2: $\begin{bmatrix} r_1 \\ r_2^* \end{bmatrix} = \mathbf{U} \begin{bmatrix} \boldsymbol{\lambda_1} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\lambda_2} \end{bmatrix} \mathbf{V^H} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2^* \end{bmatrix}$

$\mathbf{s}' \rightarrow \mathbf{s},$

$\begin{bmatrix} \boldsymbol{s_1} \\ \boldsymbol{s_2} \end{bmatrix} = \mathbf{V} \begin{bmatrix} \boldsymbol{s'_1} \\ \boldsymbol{s'_2} \end{bmatrix}$

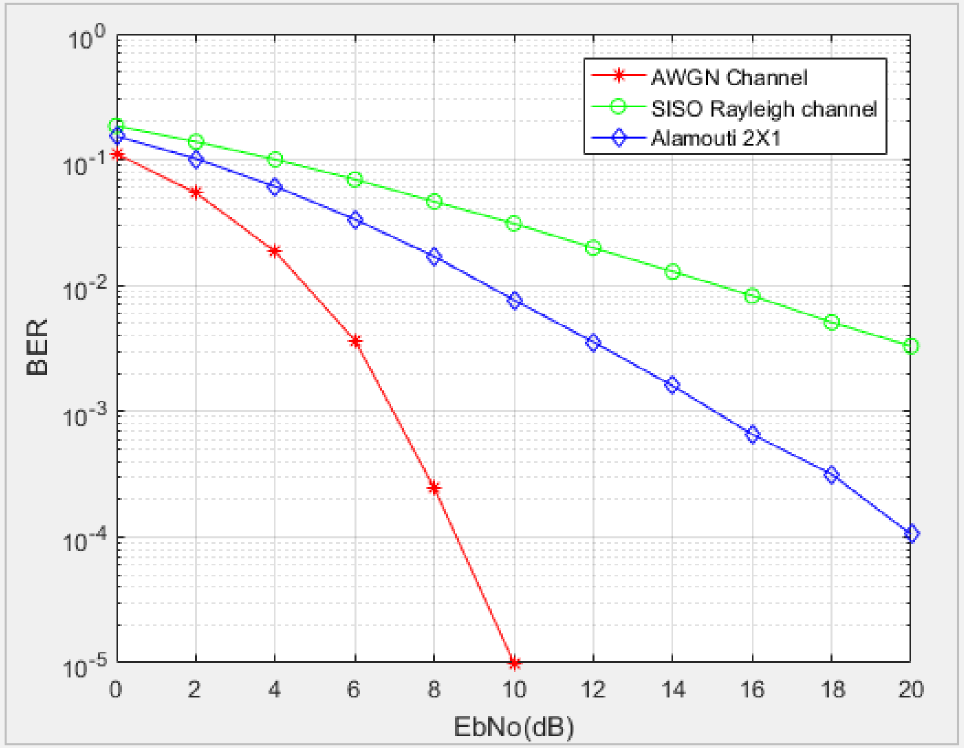
Step 3: $\begin{bmatrix} r_1 \\ r_2^* \end{bmatrix} = \mathbf{U} \begin{bmatrix} \boldsymbol{\lambda_1} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\lambda_2} \end{bmatrix} \mathbf{V^H V} \begin{bmatrix} \boldsymbol{s'_1} \\ \boldsymbol{s'_2} \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2^* \end{bmatrix} \Rightarrow \begin{bmatrix} r_1 \\ r_2^* \end{bmatrix} = \mathbf{U} \begin{bmatrix} \boldsymbol{\lambda_1} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\lambda_2} \end{bmatrix} \begin{bmatrix} \boldsymbol{s'_1} \\ \boldsymbol{s'_2} \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2^* \end{bmatrix}$

$\Rightarrow \mathrm{U^H} \begin{bmatrix} r_1 \\ r_2^* \end{bmatrix} = \begin{bmatrix} \boldsymbol{\lambda_1} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\lambda_2} \end{bmatrix} \begin{bmatrix} \boldsymbol{s'_1} \\ \boldsymbol{s'_2} \end{bmatrix} + \mathrm{U^H} \begin{bmatrix} n_1 \\ n_2^* \end{bmatrix}$

```
clear all
datasize=100000;
EbNo=0:2:20;
M=4;                                % QPSK modulation
x=randsrc(2,datasize/2,[0:3]);
x1=pskmod(x,M,pi/4);
h=randn(4,datasize/2)+j*randn(4,datasize/2);    %Rayleigh fading channel
h=h./sqrt(2);
```

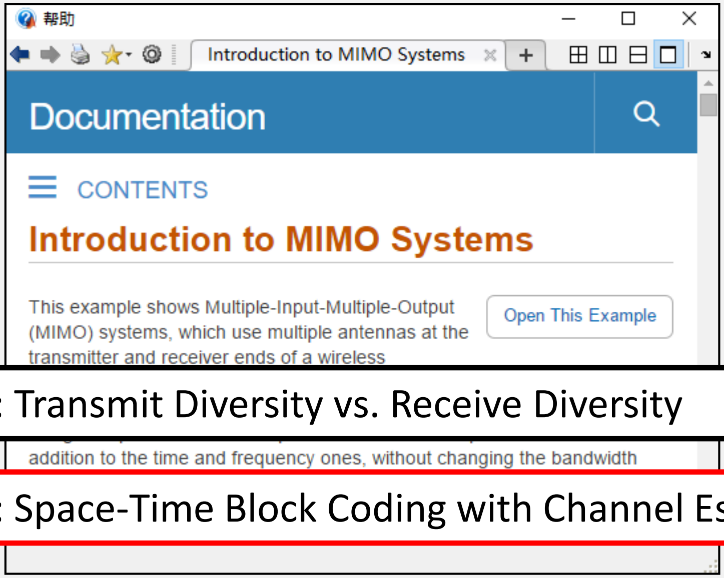
```
for indx=1:length(EbNo)
    sigma1=sqrt(1/(4*10.^(EbNo(indx)/10)));
    n=sigma1*(randn(2,datasize/2)+j*randn(2,datasize/2));
    y=x1+n;
    y1=x1+n./h(1:2,:);
    x2=pskdemod(y,M,pi/4);
    x3=pskdemod(y1,M,pi/4);
    sigma2=sqrt(1/(2*10.^(EbNo(indx)/10)));
    n=sigma2*(randn(4,datasize/2)+j*randn(4,datasize/2));
    n1(1,:)=(conj(h(1,:)).*n(1,:)+h(2,:).*conj(n(2,:)))./(sum(abs(h(1:2,:)).^2));
    n1(2,:)=(conj(h(2,:)).*n(1,:)-h(1,:).*conj(n(2,:)))./(sum(abs(h(1:2,:)).^2));
    y=x1+n1;
    x4=pskdemod(y,M,pi/4);
    n2(1,:)=(conj(h(1,:)).*n(1,:)+h(2,:).*conj(n(2,:))+...
            conj(h(3,:)).*n(3,:)+h(4,:).*conj(n(4,:)))./(sum(abs(h).^2));
    n2(2,:)=(conj(h(2,:)).*n(1,:)-h(1,:).*conj(n(2,:))+...
            conj(h(4,:)).*n(3,:)-h(3,:).*conj(n(4,:)))./(sum(abs(h).^2));
    y1=x1+n2;
    x5=pskdemod(y1,M,pi/4);
    [temp,ber1(indx)]=biterr(x,x2,log2(M));
    [temp,ber2(indx)]=biterr(x,x3,log2(M));
    [temp,ber3(indx)]=biterr(x,x4,log2(M));
    [temp,ber4(indx)]=biterr(x,x5,log2(M));

end
semilogy(EbNo,ber1,'-r*',EbNo,ber2,'-go',EbNo,ber3,'-bd',EbNo,ber4,'-k.')
```

$$\frac{\begin{bmatrix} h_1^* & h_2 \\ h_2^* & -h_1 \end{bmatrix}}{|h_1|^2+|h_2|^2}\begin{bmatrix} r_1 \\ r_2^* \end{bmatrix} = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} + \frac{\begin{bmatrix} h_1^* n_1 & h_2 n_2^* \\ h_2^* n_1 & -h_1 n_2^* \end{bmatrix}}{|h_1|^2+|h_2|^2}$$



Tutorial in MATLAB help documents

- hAlamoutiEnc = comm.OSTBCEncoder;
- hAlamoutiDec = comm.OSTBCCombiner;



PART 1: Transmit Diversity vs. Receive Diversity

PART 2: Space-Time Block Coding with Channel Estimation

```
K>> comm.OSTBCEncoder

ans =

  comm.OSTBCEncoder 具有属性:

    NumTransmitAntennas: 2

  显示 所有属性

K>> comm.OSTBCCombiner

ans =

  comm.OSTBCCombiner 具有属性:

    NumTransmitAntennas: 2
     NumReceiveAntennas: 1

  显示 所有属性
```

Simulation with comm. functions

- comm.OSTBCEncoder
- comm.OSTBCCombiner

CONTENTS

OSTBC Combining Algorithms

The OSTBC Combiner block supports five different OSTBC combining computation algorithms. Depending on the selection for **Rate** and **Number of transmit antennas**, you can select one of the algorithms shown in the following table.

Transmit Antenna	Rate	Computational Algorithm per Codeword Block Length
2	1	$\begin{pmatrix} \hat{s}_1 \\ \hat{s}_2 \end{pmatrix} = \frac{1}{\|H\|^2} \sum_{j=1}^{M} \begin{pmatrix} h^*_{1,j} r_{1,j} + h_{2,j} r^*_{2,j} \\ h^*_{2,j} r_{1,j} - h_{1,j} r^*_{2,j} \end{pmatrix}.$
3	1/2	$\begin{pmatrix} \hat{s}_1 \\ \hat{s}_2 \end{pmatrix} = \frac{1}{\|H\|^2} \sum_{j=1}^{M} \begin{pmatrix} h^*_{1,j} r_{1,j} + h_{2,j} r^*_{2,j} + h^*_{3,j} r_{3,j} \\ h^*_{2,j} r_{1,j} - h_{1,j} r^*_{2,j} - h_{3,j} r^*_{4,j} \end{pmatrix}.$

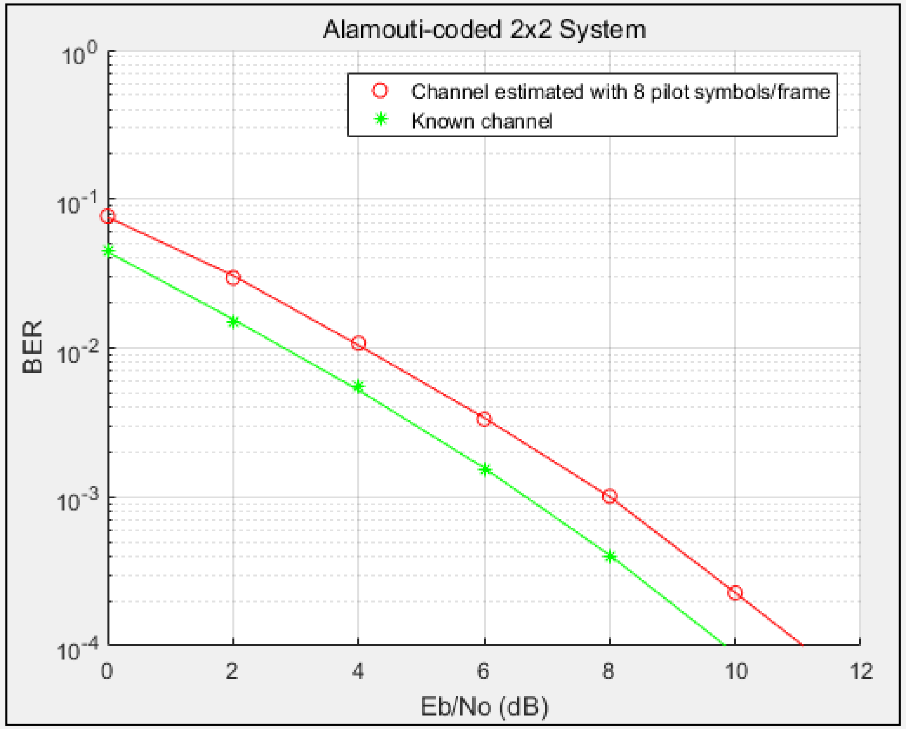
CONTENTS

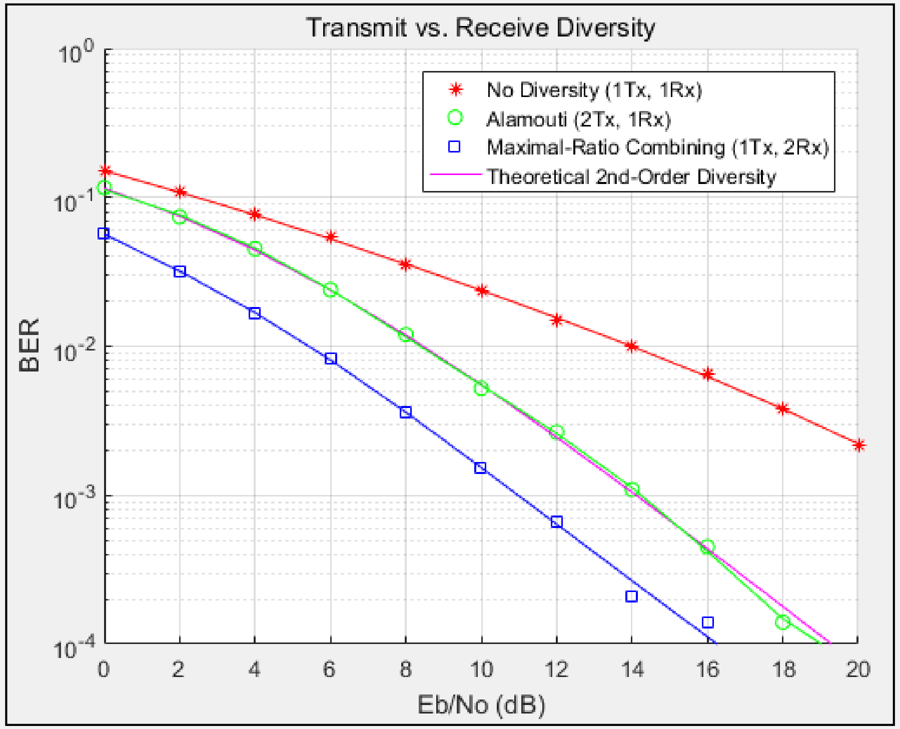
OSTBC Encoding Algorithms

The OSTBC Encoder block supports five different OSTBC encoding algorithms. Depending on the selection for **Rate** and **Number of transmit antennas**, the block implements one of the algorithms in the following table:

Transmit Antenna	Rate	OSTBC Codeword Matrix
2	1	$\begin{pmatrix} s_1 & s_2 \\ -s_2^* & s_1^* \end{pmatrix}$
3	1/2	$\begin{pmatrix} s_1 & s_2 & 0 \\ -s_2^* & s_1^* & 0 \\ 0 & 0 & s_1 \\ 0 & 0 & -s_2^* \end{pmatrix}$

Simulation results





Apply Alamouti 2X2 to the QPSK transceiver

```
H  llo world ,086
Hello world 1087
Hello world ,088
Hello y/rld 1089
Hello t?rb$ 1090
Hello y/rld 1091
Hello world ?p92
OEllo t?rkD 1093
H  llo workD 1094
Hello worVd 1095
Hello wor $ 1096
Hello world  097
Hello world 1098
Hello workD 1099
Error rate = 0.035893.
Number of detected errors = 402.
Total number of compared samples = 11200.
```

EbNo=10dB

```
Hello wn  ld 0X84
HeoHo world 2`85
Hello world 1D86
hf?<U wa2ld 1D87ello world 1088
Heb.W wmJld 0X89
Hello world 6  90
K5qlo wa2ld 0,91
Helo? wa2ld 6  92ello world 1093
 ello world 2`94
Hello wn  ld 1D95
Hello wbbld 10:f
Xello whRld 6  97
 elkO world 1098
Hello world 7x99
Error rate = 0.076696.
Number of detected errors = 859.
Total number of compared samples = 11200.
```

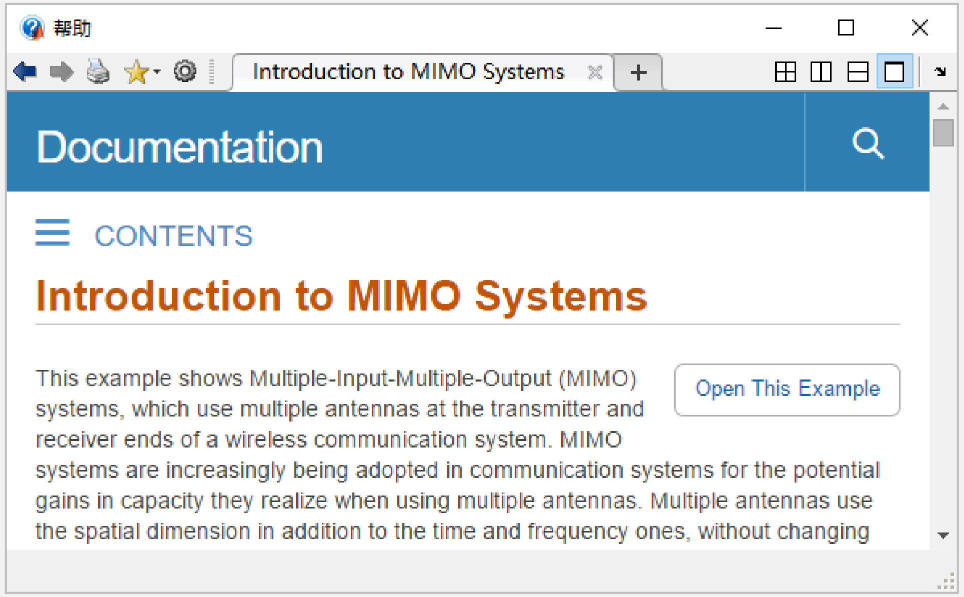
EbNo=5dB

```
HellH   ot-  TOX88
  ello y/uLj  1  87
F"Llo whH_    1088
Ue  "o#    f  d  1089
;K,l   MoL   P1090
F$>lo#'ow9:`2T~
|%b,b6?or  d  1    2
!q6la`Mop=d=1092
rd  lo ~  3`d 2z  3
!xb,  'Wi=<  D6  7
oello woqHy 1>~
Hello.7ojN  Z0  $7
H  lla`v=r*   1  w
|(}>r#'UrU3  1-M9
Error rate = 0.289464.
Number of detected errors = 3242.
Total number of compared samples = 11200.
```

EbNo=0dB

Assignments

- Read the example ‘Introduction to MIMO Systems’ in Communications System Toolbox.
- Read the paper: A Simple Transmit Diversity Technique for Wireless Communications.
- **Apply Alamouti 2X2 to the QPSK transceiver**
- Alamouti 2X2 implementation using USRP



- **Question ?**