Mobile and Personal Communication Lab

EEE G592

K Srinivasan - 2020H1240101P
Gayathri K S - 2020H1240103P
Amrith Aathmaram - 2020H1240104P

Lab Group Project Report

Objective

Consider the system model having

a) Two Transmit antennas and one receive antenna (2 x 1) and analyse transmit diversity (MISO)
b) One Transmit antenna and two receive antennas (1 x 2) and analyse receive diversity (SIMO)
c) 2 x 2, 4 x 4 MIMO setup

Compare the BER of SISO, MISO, SIMO and MIMO using coherent BPSK modulation over flat-fading Rayleigh channels corrupted by AWGN.

Theory

Fading effects can cause the received signal to strength to random vary, making decoding of such received signals quite erroneous. Effects of fading can be minimised using equalizers. An equalizer makes using of the channel estimate acquired during the pilot transmission phase to do additional processing on the received signal before decoding. This can be employed for a signal transmit and receive antenna case, but, the performance of such systems still does not compare with systems that are only affected by receiver additive noise. Improvisation in the performance can be achieved by making use of diversity techniques. Space diversity is a popular approach that can be used. Space diversity makes use of the fact that different antennas in an antenna array would undergo uncorrelated fading, if spaced appropriately. Hence, the chances that the communication links between all the transmit and receive antennas pairs in a communication system are bad simultaneously is minimised and optimal combining of the signals received can be used to improve the detection.

Receive Diversity

Maximum-ratio combining (MRC) also known as ratio-squared combining and pre-detection combining is a method of diversity combining utilized here to implement receive diversity for a SIMO system. This is a technique wherein weighted signal inputs from all receive antennas are coherently combined. The weighting factor is chosen so as to cancel the phase of the complex channel gain along with providing a gain which is proportional to the received rms signal value so as to have a co-phased addition at the receiver and thereby boosting the received SNR. The chosen weight is inversely proportional to the noise level N_0.

Hence, $\alpha_i = a_i e^{-j\theta_i}$ is the weighting factor where, θ_i is the phase of the signal arriving on the i^{th} branch.

The envelope at the combiner output is given by,

$$r = \sum_{i=1}^{M} (a_i r_i)$$

Consider the received signal $y = hs + n$ where $n \sim CN(0, I_{NXN})$

The least-square solution is given by

$$\hat{s} = (h^* h)^{-1} h^* y$$

The least square solution for an N antenna system which is the MRC solution can be written as

$$\hat{s} = \frac{h^*{}_0 y_0 + h^*{}_1 y_1 + \cdots \dots \dots .. + h^*{}_{N-1} y_{N-1}}{|h_0|^2 + \dots \dots \dots + |h_{N-1}|^2}$$

For 1x 2 SIMO system,

$$\hat{s} = \frac{h^*{}_0 y_0 + h^*{}_1 y_1}{|h_0|^2 + |h_1|^2}$$

<u>Transmit Diversity</u>

Transmit diversity makes use of multiple transmit antennas and a single receive antenna. Transmit diversity can be applied in two primary ways. In the first method, the transmitting system estimates the channel and uses it to do additional processing on the transmitted data that make it less susceptible to fading effects. The second method assumes that the transmitter has no information regarding the channel. In this method, the data symbols are encoded using Space Time Block Codes before transmission. Space Time Block Codes encode the data to be transmitted across two dimensions, space and time. Encoding along the spatial dimensions deals with which symbols and in what form are to be transmitted from each antenna of the transmitter, while encoding along the time dimension deals with which symbols in what form should be transmitted in a particular time slot of data transmission. Space Time Block Codes are essentially a combination of the two. They encode each input symbol to convert it into a form known to the receiver and transmit the symbol from one of the transmitting antennas in one of the time instants over which the code exists. A popular implementation of Space Time Block Codes is the Alamouti code. The basic Alamouti code is uses transmit diversity with the help of two transmit antennas and one receive antenna. The encoding structure takes two input symbols at a time and encodes them to generate four encoded symbols which are transmitted 2 symbols at a time (from both transmitting antennas), over two time instances. The 2 x 1 Alamouti Code is an example of a full rate code, i.e., one where effectively one symbol is decoded in every time instant at the receiver. Based on the knowledge of the encoding, the receiver does appropriate manipulation on the received symbols to get an estimate of the data sent over the channel. The encoding for the (2 x 1) Alamouti Code is as follows:

If x_1 and x_2 are two input symbols, assuming that the channel flat-fading coefficients remain constant for two time instants, the encoded input symbols are given as

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

where each time instant is represented by a column of the matrix and row is the antenna from which it should be transmitted.

The receiver receives symbols y_1 and y_2 at the two time instances

$$\begin{bmatrix} y(1) \\ y^*(2) \end{bmatrix} = y$$

Channel fading matrix

$$[h_1 \quad h_2] = h$$

The received symbols can be stacked to get

$$\underbrace{\begin{bmatrix} y(1) \\ y^*(2) \end{bmatrix}}_{\mathbf{y}} = \underbrace{\begin{bmatrix} h_1 & h_2 \\ h_2^* & -h_1^* \end{bmatrix}}_{\mathbf{H}} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \underbrace{\begin{bmatrix} n(1) \\ n^*(2) \end{bmatrix}}_{\mathbf{n}}$$

The columns of **H** matrix are orthogonal, hence, **y** is multiplied by the hermitian of each column divided by its norm value, the estimate of transmitted symbol with the same index as the column selected, can be obtained. This operation does not affect the additive noise characteristics because, the multiplication is by a unit magnitude complex number which only introduces rotation. Since, the noise random variable is assumed to be circularly symmetry, its statistics like noise variance will not change. The operations can be represented as follows

$$\mathbf{c}_1 = \begin{bmatrix} h_1 \\ h_2^* \end{bmatrix}, \mathbf{c}_2 = \begin{bmatrix} h_2 \\ -h_1^* \end{bmatrix}$$

$$\mathbf{w}_1 = \frac{1}{\|\mathbf{c}_1\|}\mathbf{c}_1$$

$$\widehat{\mathbf{x_1}} = \mathbf{w}_1^H \mathbf{y}$$

$$\mathbf{w}_2 = \frac{\mathbf{c}_2}{\|\mathbf{c}_2\|}$$

$$\widehat{\boldsymbol{x_2}} = \mathbf{w}_2^H \mathbf{y}$$

MIMO Diversity

(2 x 2 Setup)

An extended form of the Alamouti code is used to implement the (2 x 2) MIMO setup. The transmit symbols are encoded into a (2 x 2) matrix where each column represents the instant at which the symbols are sent, and each row represents the antenna used for transmission. The encoded symbols $[s_1 \; s_2 \;]$ are represented in matrix form as

$$\begin{bmatrix} s_1 & -s_2^* \\ s_2 & s_1^* \end{bmatrix} = \mathrm{S}$$

The channel matrix for the (2 x 2) setup is given by

$$\underbrace{\begin{bmatrix} h_{11} & h_{21} \\ h_{12} & h_{22} \end{bmatrix}}_{\mathbf{H}}$$

The received symbols at the receiver is given by

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{bmatrix} = \mathbf{HS} + \mathbf{N}$$

To express the received signal in terms of s_1 and s_2 alone, we modify the channel matrix as

$$\underbrace{\begin{bmatrix} h_{11} & h_{21} \\ -h_{21}^* & h_{11}^* \end{bmatrix}}_{\mathbf{H_1}} \qquad \underbrace{\begin{bmatrix} h_{12} & h_{22} \\ -h_{22}^* & h_{12}^* \end{bmatrix}}_{\mathbf{H_2}}$$

$$\underbrace{\begin{bmatrix} \mathbf{H_1} \\ \mathbf{H_2} \end{bmatrix}}_{\mathbf{H_T}}$$

A composite vector that is a function of all the symbols received at each antenna at each time instant can be expressed as

$$\boldsymbol{Y} = \begin{bmatrix} r_{11} \\ r_{12}^* \\ r_{21} \\ r_{22}^* \end{bmatrix} = \mathbf{H_T} \begin{bmatrix} S_1 \\ s_2 \end{bmatrix}$$

To get back the original signal at the receiver, we need to eliminate the H matrix as in 2x1 case. Since the modified H_T matrix is not square, we have to take the Hermitian transposition of the composite channel matrix, given as

$$\mathrm{H_T}^{\mathrm{H}} = \begin{bmatrix} h_{11}^* & h_{21} & h_{12}^* & h_{22} \\ h_{21}^* & -h_{11} & h_{22}^* & -h_{12} \end{bmatrix}$$

On multiplying $\mathrm{H_T}^{\mathrm{H}}$ and H_T, we get

$$\begin{bmatrix} |h_{11}|^2 + |h_{12}|^2 + |h_{21}|^2 + |h_{22}|^2 & 0 \\ 0 & |h_{11}|^2 + |h_{12}|^2 + |h_{21}|^2 + |h_{22}|^2 \end{bmatrix}$$

Suppose d = $|h_{11}|^2 + |h_{12}|^2 + |h_{21}|^2 + |h_{22}|^2$.

To recover the original signal back, we do the following

$$\frac{1}{\mathrm{d}}\mathrm{H_T}^{\mathrm{H}}\boldsymbol{Y} = \frac{1}{\mathrm{d}}\mathrm{H_T}^{\mathrm{H}}\mathrm{H}_{\mathbf{T}}\mathbf{S} + \frac{1}{\mathrm{d}}\mathrm{H_T}^{\mathrm{H}}\mathbf{N}$$

$$= \mathbf{S} + \frac{1}{\mathrm{d}}\mathrm{H_T}^{\mathrm{H}}\mathbf{N} = \hat{\mathbf{S}}$$

<u>(4 x 4 Setup)</u>

Transmission of symbols in the (4 x 4) MIMO setup can be done after encoding using STBCs. Again, an extended form of the Alamouti code can be used to achieve this, wherein, every four symbols from the input are taken and encoded into a (4 x 4) matrix where each column represents the time instant in which the transmission should occur and each column, the antenna used. This is again a full rate code. The encoded symbols (input $[s_1\ s_2\ s_3\ s_4]$) are represented in matrix form as

$$\begin{bmatrix} s_1 & s_2^* & s_3^* & s_4 \\ s_2 & -s_1^* & s_4^* & -s_3 \\ s_3 & s_4^* & -s_1^* & -s_2 \\ s_4 & -s_3^* & -s_2^* & s_1 \end{bmatrix} = \mathrm{S}$$

The channel matrix for the 4x4 setup if given by:

$$\underbrace{\begin{bmatrix} h_{11} & h_{21} & h_{31} & h_{41} \\ h_{12} & h_{22} & h_{32} & h_{42} \\ h_{13} & h_{23} & h_{33} & h_{43} \\ h_{14} & h_{24} & h_{34} & h_{44} \end{bmatrix}}_{\mathbf{H}}$$

The received symbols at the receiver can be represented as

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{14} \\ r_{21} & r_{22} & r_{23} & r_{24} \\ r_{31} & r_{32} & r_{33} & r_{34} \\ r_{41} & r_{42} & r_{43} & r_{44} \end{bmatrix} = \mathbf{HS} + \mathbf{N}$$

Representing column i of $\mathbf{R}$ as $\mathbf{R_i}$. A composite vector that is a function of all the symbols received at each antenna at each time instant can be expressed as

$$\boldsymbol{Y} = \begin{bmatrix} R_1 \\ R_2^* \\ R_3^* \\ R_4 \end{bmatrix}$$

Although the STBC used is an extension of the Alamouti Code, the orthogonality property is not retained, hence, the output can be estimated using the standard zero forcing method. The procedure for estimation of the transmitted symbols is given below.

$$\underbrace{\begin{bmatrix} h_{11} & h_{21} & h_{31} & h_{41} \\ h_{12} & h_{22} & h_{32} & h_{42} \\ h_{13} & h_{23} & h_{33} & h_{43} \\ h_{14} & h_{24} & h_{34} & h_{44} \end{bmatrix}}_{\mathbf{H_1}} \quad \underbrace{\begin{bmatrix} -h_{21}^* & h_{11}^* & -h_{41}^* & h_{31}^* \\ -h_{22}^* & h_{12}^* & -h_{42}^* & h_{32}^* \\ -h_{23}^* & h_{13}^* & -h_{43}^* & h_{33}^* \\ -h_{24}^* & h_{14}^* & -h_{44}^* & h_{34}^* \end{bmatrix}}_{\mathbf{H_2}} \quad \underbrace{\begin{bmatrix} -h_{31}^* & -h_{41}^* & h_{11}^* & h_{21}^* \\ -h_{32}^* & -h_{42}^* & h_{12}^* & h_{22}^* \\ -h_{33}^* & -h_{43}^* & h_{13}^* & h_{23}^* \\ -h_{34}^* & -h_{44}^* & h_{14}^* & h_{24}^* \end{bmatrix}}_{\mathbf{H_3}}$$

$$\underbrace{\begin{bmatrix} h_{41} & -h_{31} & -h_{21} & h_{11} \\ h_{42} & -h_{32} & -h_{22} & h_{12} \\ h_{43} & -h_{33} & -h_{23} & h_{13} \\ h_{44} & -h_{34} & -h_{24} & h_{14} \end{bmatrix}}_{\mathrm{H}_4}$$

$$\underbrace{\begin{bmatrix} H_1 \\ H_2 \\ H_3 \\ H_4 \end{bmatrix}}_{\mathbf{H_T}}$$

Using these matrices, the expression for the composite vector **Y** can be written as

$$\boldsymbol{Y} = \boldsymbol{H_T} \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \end{bmatrix}$$

Hence, by using zero-forcing decoding, the estimates of the transmitted can be obtained as

$$\begin{bmatrix} \hat{s}_1 \\ \hat{s}_2 \\ \hat{s}_3 \\ \hat{s}_4 \end{bmatrix} = \boldsymbol{H_T^\dagger} Y$$

Code

BPSK SISO

```
clc; clear; close all;

M = 2; %Constellation Size
N = 10 ^ 6; %No. of bits
data_points = 0 : M - 1; %Symbols for the modulation scheme
Eb = 1; %Bit Energy
constellation = -sqrt(Eb) * exp(-1i*2*pi*data_points/M); %Constellation Points
bitstream_decoded_rayleigh = [];
bitstream_decoded_awgn = [];

EbNo_dB = [-5 : 0.5 : 15]; %Array of SNR values used for simulation (in dB)
ber_array_rayleigh = [];
ber_array_awgn = [];
EbNo = 10 .^ (EbNo_dB/10); %SNR values in linear scale

bitstream = randi([0 1],N,1); %Random bitstream generation
bitstream_constellation_rep = -sqrt(Eb) * exp(-1i*2*pi*(bitstream)/M); %Converting
%bitstream representation to constellation points

for j = 1 : length(EbNo)

    N0_vector = sqrt((Eb/EbNo(j))/2) * randn(N,1) + 1i * (sqrt((Eb/EbNo(j))/2) * randn(N,1));
%Generating noise samples from a
    %Circularly Symmetric Gaussian Distribution of variance N0/2 along each dimension
    h = (1/sqrt(2)) * (randn(1,N) + 1i * randn(1,N)); %Rayleigh Flat Fading factor (single
tap)
    rcvd_constellation_awgn = bitstream_constellation_rep + N0_vector; %Adding AWGN
    rcvd_constellation_rayleigh = bitstream_constellation_rep .* h(:) + N0_vector;
    rcvd_constellation_rayleigh_equalised =  rcvd_constellation_rayleigh./(h(:)); %Zero
forcing equaliser

    %Minimum Euclidean Distance decoding (for the fading + AWGN corrupted received bits)
    for k = 1 : length(rcvd_constellation_rayleigh_equalised)
        EucD = abs(constellation - rcvd_constellation_rayleigh_equalised(k) *
ones(size(constellation)));
        %Computing the Euclidean distance of the received symbol from each contellation point
        %for the given Modulation Scheme
        [~,pos] = min(EucD); %Minimum Euclidean distance computation
        bitstream_decoded_rayleigh(k) = data_points(pos); %Decision based on minimum
Euclidean
        %distance
    end

    [~,ber] = biterr(bitstream,bitstream_decoded_rayleigh'); %BER computation
    ber_array_rayleigh(j) = ber;

    %Minimum Euclidean Distance decoding (for the AWGN corrupted received bits)
    for k = 1 : length(rcvd_constellation_awgn)
        EucD = abs(constellation - rcvd_constellation_awgn(k) * ones(size(constellation)));
        %Computing the Euclidean distance of the received symbol from each contellation point
        %for the given Modulation Scheme
        [~,pos] = min(EucD); %Minimum Euclidean distance computation
        bitstream_decoded_awgn(k) = data_points(pos); %Decision based on minimum Euclidean
        %distance
```

```
    end

    [~,ber] = biterr(bitstream,bitstream_decoded_awgn'); %BER computation
    ber_array_awgn(j) = ber;

end

semilogy(EbNo_dB,ber_array_awgn,'-bo','LineWidth',2);hold on;
semilogy(EbNo_dB,ber_array_rayleigh,'-kp','LineWidth',2);
legend('AWGN','Rayleigh fading + AWGN');
xlabel('$\frac{Eb}{N0} (dB)$','Interpreter','latex');
ylabel('BER');
title('Comparison of BER vs. SNR for AWGN and Rayleigh fading channel');
grid on;

save('SISO_BPSK.mat','EbNo_dB','ber_array_awgn','ber_array_rayleigh');
```

<u>BPSK SIMO (Receive Diversity)</u>

```
function [SER,legendtext] = MRC(L)

if nargin < 1
    L = 2;
end

% Find the SER for SIMO case with L channel outputs(antennaes),
% decoding done by Maximal-Ratio Combining(MRC),combining signals from all the receive
antennae
%
% Input:
% L = Row vector with each value tells number of receive antennaes
% eg. L = [1 3] or [1:4]
%
% Output:
% SER = A LxNumOfSnrPoints matrix with each row corresponds to particular L outputs
%
% Plot:
% A plot corresponding to each value in vector L

EbNo_dB = [-5 : 0.5 : 15]; %Array of SNR values used for the simuation (dB)
EbNo = 10.^(EbNo_dB/10);
sd = sqrt(1./EbNo);  % Standard Deviation
var = sd.^2;        % Variance

% Number of Receive antenae to simulate
%L = 2;
% Number of Monte carlo iterationT = 10 ^ 6;

x_bpsk = sign(randn(1, T));

SER = zeros(length(L),length(sd));

for p = 1 : length(L)
    noise = zeros(L(p),T);
    h_channel = zeros(L(p),T);
    y = zeros(L(p),T);
    y_new = zeros(1,T);
```

```
    x_hat = zeros(1,T);

    for k = 1 : length(sd)

noise = (sqrt(L)) * ((1/sqrt(2)) * sd(1,k) * randn(L(p),T) + 1i * (1/sqrt(2)) * sd(1,k) *
randn(L(p),T));

      % Generating COMPLEX RANDOM CHANNEL(LxT) WITH MEAN 0 & VAR = 1
      h_channel = (1/sqrt(2)) * randn(L(p),T) + 1i * (1/sqrt(2)) * randn(L(p),T);

      %%STEP 3 OF M-C SIMULATION

      y = h_channel .* repmat(x_bpsk,L(p),1) + noise;

      y_new = dot(h_channel./repmat(sqrt(sum(abs(h_channel).^2,1)),L(p),1),y,1);
      %y_new = conj(h_channel./abs(h_channel)).*y;

      x_hat = sign(real(y_new));    %% since BPSK, so taking real sufficient statistic

      SER(p,k) = mean(x_hat~=x_bpsk);

    end
end

%figure(1)
%plot(abs(y(1,:)))
%plot(real(y(30,:)),imag(y(30,:)),'o')
%%PLOTTING
figure(1);
semilogy(EbNo_dB,SER,'b-s','linewidth',2);
grid on
axis([min(EbNo_dB) max(EbNo_dB) 1e-6 1])
title('BPSK - SIMO(MRC): SER Simulation with varying Receive antennae')
xlabel('signal-to-noise ratio (SNR) [dB]')
ylabel('symbol error rate (SER)')
legendtext ='';
for p = 1:length(L)
    legendtext = [legendtext; sprintf('MRC L = %d ',L(p))];
end
legend(legendtext);

ber_array_MRC_1x2 = SER;
save('SIMO_1x2_BPSK.mat','EbNo_dB','ber_array_MRC_1x2');
```

BPSK MISO (2 x 1 Transmit Diversity)

```
clc; clear; close all;

M = 2; %Constellation Size
N = 10 ^ 6; %No. of bits
data_points = 0 : M - 1; %Symbols for the modulation scheme
Eb = 1; %Bit Energy
constellation = -sqrt(Eb/2) * exp(-1i*2*pi*data_points/M); %Constellation Points
bitstream_decoded_OSTBC_2 = [];

EbNo_dB = [-5 : 0.5 : 15]; %Array of SNR values used for simulation (in dB)
ber_array_OSTBC_2 = [];
```

```
EbNo = 10 .^ (EbNo_dB/10); %SNR values in linear scale

bitstream = randi([0 1],N,1); %Random bitstream generation
bitstream_constellation_rep = -sqrt(Eb/2) * exp(-1i*2*pi*(bitstream)/M); %Converting
%bitstream representation to constellation points (sqrt(1/2)) factor is added to ensure Eb
%energy is transmitted in every time instant

h = (1/sqrt(2)) * (randn(1,N) + 1i * randn(1,N)); %Rayleigh Flat Fading factor (single tap).
%N/2 coefficients are used for characterising the fading coefficients for Antenna 1 &
remaining N/2 for antenna 2
%It is assumed that the channel matrix is constant user two time instances
h_mod = kron(reshape(h,2,N/2),ones(1,2)); %Fading coefficients for each antenna are stored in
each row
STBC_coded_data = zeros(2,N);
STBC_coded_data(:,1 : 2 : end) = reshape(bitstream_constellation_rep,2,N/2);
%[x1(1:2:N);x2(1:2:N)]
STBC_coded_data(:,2 : 2 : end) = repmat([-1;1],1,N/2) .*
flipud(reshape(conj(bitstream_constellation_rep),2,N/2));
%[-x2*(2:2:N);x1*(2:2:N)]

for j = 1 : length(EbNo)

    N0_vector = sqrt((Eb/EbNo(j))/2) * randn(1,N) + 1i * (sqrt((Eb/EbNo(j))/2) * randn(1,N));
%Generating noise samples from a
    %Circularly Symmetric Gaussian Distribution of variance N0/2 along each dimension

    rcvd_data = sum(STBC_coded_data .* h_mod,1) + N0_vector;
    y = reshape(rcvd_data,2,N/2);
    y(2,:) = conj(y(2,:)); % [y1(1 : N/2); y2*(1 : N/2)]

    C = zeros(2,N);
    C(:,[1 : 2 : end]) = reshape(h,2,N/2);
    C(:,[2 : 2 : end]) = repmat([1;-1],1,N/2) .*  flipud(reshape(h,2,N/2));
    C(2,:) = conj(C(2,:));

    C1 = C(:,[1 : 2 : end]);
    C1_norm = sqrt(sum(abs(C1) .^ 2,1));
    C2 = C(:,[2 : 2 : end]);
    C2_norm = sqrt(sum(abs(C2) .^ 2,1));

    rcvd_data_STBC_decoded = zeros(1,N);
    rcvd_data_STBC_decoded(1 : 2 : end) = sum((conj(C1) .* y),1) ./ C1_norm;
    rcvd_data_STBC_decoded(2 : 2 : end) = sum((conj(C2) .* y),1) ./ C2_norm;

    %Minimum Euclidean Distance decoding (for the fading + AWGN corrupted received bits)
    for k = 1 : length(rcvd_data_STBC_decoded)
        EucD = abs(constellation - rcvd_data_STBC_decoded(k) * ones(size(constellation)));
        %Computing the Euclidean distance of the received symbol from each contellation point
        %for the given Modulation Scheme
        [~,pos] = min(EucD); %Minimum Euclidean distance computation
        bitstream_decoded_OSTBC_2(k) = data_points(pos); %Decision based on minimum Euclidean
        %distance
    end

    [~,ber] = biterr(bitstream,bitstream_decoded_OSTBC_2'); %BER computation
    ber_array_OSTBC_2(j) = ber;

end
```

```
semilogy(EbNo_dB,ber_array_OSTBC_2,'-rp','LineWidth',2);
legend('2 x 1 OSTBC with Rayleigh fading + AWGN');
xlabel('$\frac{Eb}{N0} (dB)$','Interpreter','latex');
ylabel('BER');
title('Comparison of BER vs. SNR for (2x1) Transmit diversity for BPSK');
grid on;

save('MISO_BPSK.mat','EbNo_dB','ber_array_OSTBC_2');
```

BPSK MIMO (2 x 2)

```
clc; clear; close all;

M = 2; %Constellation Size
N = 10 ^ 6; %No. of Bits
data_points = 0 : M - 1; %Symbols of the constellation
Eb = 1;
constellation = -sqrt(Eb) * exp(-1i*2*pi*data_points/M); %Constellation Points

EbNo_dB = [-5 : 0.5 : 15]; %Array of SNR values used for the simuation (dB)
ber_array_alamouti_2x2 = zeros(size(EbNo_dB));
EbNo = 10 .^ (EbNo_dB/10); %SNR values in the Linear Scale

bitstream = randi([0 1],N,1); %Random bitstream generation
bitstream_constellation_rep = -sqrt(Eb/2) * exp(-1i*2*pi*(bitstream)/M); %Converting the 
bitstream representation to
%constellation points; sqrt(1/2) factor is added to ensure that Eb energy is transmitted in 
every time instant

symbolsTxD = reshape(bitstream_constellation_rep,2,N/2); %Every two symbols are encoded using 
2x2 Alamouti codes and
%sent from 2 transmitting antennas in 2 time instances

%Alamouti Encoding & Decoding {Every 2 symbols are encoded using 2x2 Alamouti code and 
corrupted using AWGN and Rayleigh
%fading before decoding}

for ii = 1 : length(EbNo)

        numErrs = 0;

        for iii = 1 : size(symbolsTxD,2)

                s = symbolsTxD(:,iii); %Extract 2 symbols from the bitstream in each iteration
                s_1 = s; %Encoded symbols in the first time instance
                s_2 = alamoutiEncoder2x2MIMO(s.',2);

                S = [s_1 s_2]; %Encoded Symbol Matrix

                h = sqrt(0.5) .* (randn(2,2) + 1i * randn(2,2)); %Channel Fading Matrix for 
2x2 links between Tx and Rx, it is expected
                %that the Channel Fading Matrix remains constant over 2 time instances
                N0_matrix = sqrt((Eb/EbNo(ii))) * randn(2,2) + 1i * (sqrt((Eb/EbNo(ii))) * 
randn(2,2)); %Generating Noise Samples
                %from a Circularly Symmetric Gaussian Distribution of Variance N0/2 along each
```

```
dimension

                R = h*S + N0_matrix; %Received symbols at each antenna element over 2 time
instances

                %Decoding Operations
                Y = [R(:,1); conj(R(:,2))];

                H_comp_1 = h;
                H_comp_2 = alamoutiEncoder2x2MIMO(h,2);

                H_comp = [H_comp_1; H_comp_2]; %Composite Channel Matrix
                C1 = H_comp(:,1);
                C1_norm = sqrt(sum(abs(C1) .^ 2,1));
                C2 = H_comp(:,2);
                C2_norm = sqrt(sum(abs(C2) .^ 2,1));

                s_hat_1 = sum((conj(C1) .* Y),1) ./ C1_norm;
                s_hat_2 = sum((conj(C2) .* Y),1) ./ C2_norm;
                s_hat = [s_hat_1; s_hat_2];

                s_decoded = zeros(size(s_hat));
                %Minimum Euclidean Distance decoding
                for k = 1 : length(s_hat)
                        EucD = abs(constellation - s_hat(k) * ones(size(constellation)));
                 %Computing the Euclidean distance of the received symbol from each
contellation point
                 %for the given Modulation Scheme
                 [~,pos] = min(EucD); %Minimum Euclidean distance computation
                 s_decoded(k) = constellation(pos); %Decision based on Minimum Euclidean
Distance
        end

        s = s > 0;
        s_decoded = s_decoded > 0;
        [Errors,~] = biterr(s,s_decoded);
        numErrs = numErrs + Errors;

       end

       ber_array_alamouti_2x2(ii) = numErrs/N;

end

semilogy(EbNo_dB,ber_array_alamouti_2x2,'-rp','LineWidth',2);
legend('2 x 2 MIMO using Alamouti Codes in Rayleigh Fading + AWGN');
xlabel('$\frac{Eb}{N0} (dB)$','Interpreter','latex');
ylabel('BER');
title('Comparison of BER vs. SNR for (2x2) Transmit diversity for BPSK');
grid on;

save('MIMO_2x2_BPSK.mat','EbNo_dB','ber_array_alamouti_2x2');
```

```
function encodedMatrix = alamoutiEncoder2x2MIMO(inp,timeSlot)

%Input
%inp is expected either be a 1x2 vector(for encoding data symbols) or 2x2
%matrix (for encoding the channel matrix)
%TimeSlot defines the time slot in which the data symbols are being sent

%Output
%encodedMatrix is a 2x1 vector in case of data symbol encoding and a 2x2
%matrix in case of encoding of the channel matrix

    temp = zeros(size(inp));

    if(size(inp,1) == 1) %Encoding data

        %Encoding the data symbols from each antenna for different time
        %instances

        if(timeSlot == 2)
            temp(:,1) = -conj(inp(:,2));
            temp(:,2) = conj(inp(:,1));

        end

        encodedMatrix = temp(:);

    else
        %Encoding of the channel matrix in order to compute the composite
        %channel matrix
        if(timeSlot == 2)

            temp(:,1) = conj(inp(:,2));
            temp(:,2) = -conj(inp(:,1));

        end

        encodedMatrix = temp;

    end

end
```

<u>BPSK MIMO (4 x 4)</u>

```
clc; clear; close all;

M = 2; %Constellation Size
N = 10 ^ 6; %No. of Bits
data_points = 0 : M - 1; %Symbols of the constellation
Eb = 1;
constellation = -sqrt(Eb) * exp(-1i*2*pi*data_points/M); %Constellation Points

EbNo_dB = [-5 : 0.5 : 15]; %Array of SNR values used for the simuation (dB)
ber_array_alamouti_4x4 = zeros(size(EbNo_dB));
EbNo = 10 .^ (EbNo_dB/10); %SNR values in the Linear Scale
```

```
bitstream = randi([0 1],N,1); %Random bitstream generation
bitstream_constellation_rep = -sqrt(Eb/4) * exp(-1i*2*pi*(bitstream)/M); %Converting the
bitstream representation to
%constellation points; sqrt(1/4) factor is added to ensure that Eb energy is transmitted in
every time instant

symbolsTxD = reshape(bitstream_constellation_rep,4,N/4); %Every four symbols are encoded
using 4x4 Alamouti codes and
%sent from 4 transmitting antennas in 4 time instances

%Alamouti Encoding & Decoding {Every 4 symbols are encoded using 4x4 Alamouti code and
corrupted using AWGN and Rayleigh
%fading before decoding}

for ii = 1 : length(EbNo)

        numErrs = 0;

        for iii = 1 : size(symbolsTxD,2)

                s = symbolsTxD(:,iii); %Extract 4 symbols from the bitstream in each iteration
                s_1 = s; %Encoded symbols in the first time instance
                s_2 = alamoutiEncoder4x4MIMO(s.',2);
                s_3 = alamoutiEncoder4x4MIMO(s.',3);
                s_4 = alamoutiEncoder4x4MIMO(s.',4);

                S = [s_1 s_2 s_3 s_4]; %Encoded Symbol Matrix

                h = sqrt(0.5) .* (randn(4,4) + 1i * randn(4,4)); %Channel Fading Matrix for
4x4 links between Tx and Rx, it is expected
                %that the Channel Fading Matrix remains constant over 4 time instances
                N0_matrix = sqrt(1/0.5) * sqrt(((Eb)/EbNo(ii))) * randn(4,4) + 1i *
sqrt(1/0.5) *(sqrt(((Eb)/EbNo(ii))) * randn(4,4)); %Generating Noise Samples
                %from a Circularly Symmetric Gaussian Distribution of Variance N0/2 along each
dimension

                R = h*S + N0_matrix; %Received symbols at each antenna element over 4 time
instances

                %Decoding Operations
                Y = [R(:,1); conj(R(:,2)); conj(R(:,3)); R(:,4)];

                H_comp_1 = h;
                H_comp_2 = alamoutiEncoder4x4MIMO(h,2);
                H_comp_3 = alamoutiEncoder4x4MIMO(h,3);
                H_comp_4 = alamoutiEncoder4x4MIMO(h,4);
                H_comp = [H_comp_1; H_comp_2; H_comp_3; H_comp_4]; %Composite Channel Matrix

                s_hat = pinv(H_comp) * Y; %The estimates of the 4 transmitted symbols is
obtained by zero-forming receive combining
        s_decoded = zeros(size(s_hat));

                %Minimum Euclidean Distance decoding
        for k = 1 : length(s_hat)
                        EucD = abs(constellation - s_hat(k) * ones(size(constellation)));
                %Computing the Euclidean distance of the received symbol from each
contellation point
```

```
                %for the given Modulation Scheme
                [~,pos] = min(EucD); %Minimum Euclidean distance computation
                s_decoded(k) = constellation(pos); %Decision based on Minimum Euclidean
Distance
        end

        s = s > 0;
        s_decoded = s_decoded > 0;
            [Errors,~] = biterr(s,s_decoded);
            numErrs = numErrs + Errors;

       end

       ber_array_alamouti_4x4(ii) = numErrs/N;

end

semilogy(EbNo_dB,ber_array_alamouti_4x4,'-rp','LineWidth',2);
legend('4 x 4 MIMO using Alamouti Codes in Rayleigh Fading + AWGN');
xlabel('$\frac{Eb}{N0} (dB)$','Interpreter','latex');
ylabel('BER');
title('Comparison of BER vs. SNR for (4x4) Transmit diversity for BPSK');
grid on;

save('MIMO_4x4_BPSK.mat','EbNo_dB','ber_array_alamouti_4x4');
```

```
function encodedMatrix = alamoutiEncoder4x4MIMO(inp,timeSlot)

%Input
%inp is expected either be a 1x4 vector(for encoding data symbols) or 4x4
%matrix (for encoding the channel matrix)
%TimeSlot defines the time slot in which the data symbols are being sent

%Output
%encodedMatrix is a 4x1 vector in case of data symbol encoding and a 4x4
%matrix in case of encoding of the channel matrix

    temp = zeros(size(inp));

    if(size(inp,1) == 1) %Encoding data

        %Encoding the data symbols from each antenna for different time
        %instances

        if(timeSlot == 2)
            temp(:,1) = conj(inp(:,2));
            temp(:,2) = -conj(inp(:,1));
            temp(:,3) = conj(inp(:,4));
            temp(:,4) = -conj(inp(:,3));
        elseif(timeSlot == 3)
            temp(:,1) = conj(inp(:,3));
            temp(:,2) = conj(inp(:,4));
            temp(:,3) = -conj(inp(:,1));
            temp(:,4) = -conj(inp(:,2));
        elseif(timeSlot == 4)
```

```
            temp(:,1) = inp(:,4);
            temp(:,2) = -inp(:,3);
            temp(:,3) = -inp(:,2);
            temp(:,4) = inp(:,1);
        end

        encodedMatrix = temp(:);

    else
        %Encoding of the channel matrix in order to compute the composite
        %channel matrix

        if(timeSlot == 2)
            temp(:,2) = conj(inp(:,1));
            temp(:,1) = -conj(inp(:,2));
            temp(:,4) = conj(inp(:,3));
            temp(:,3) = -conj(inp(:,4));
        elseif(timeSlot == 3)
            temp(:,3) = conj(inp(:,1));
            temp(:,4) = conj(inp(:,2));
            temp(:,1) = -conj(inp(:,3));
            temp(:,2) = -conj(inp(:,4));
        elseif(timeSlot == 4)
            temp(:,4) = inp(:,1);
            temp(:,3) = -inp(:,2);
            temp(:,2) = -inp(:,3);
            temp(:,1) = inp(:,4);
        end

        encodedMatrix = temp;

    end

end
```

Simulation Results

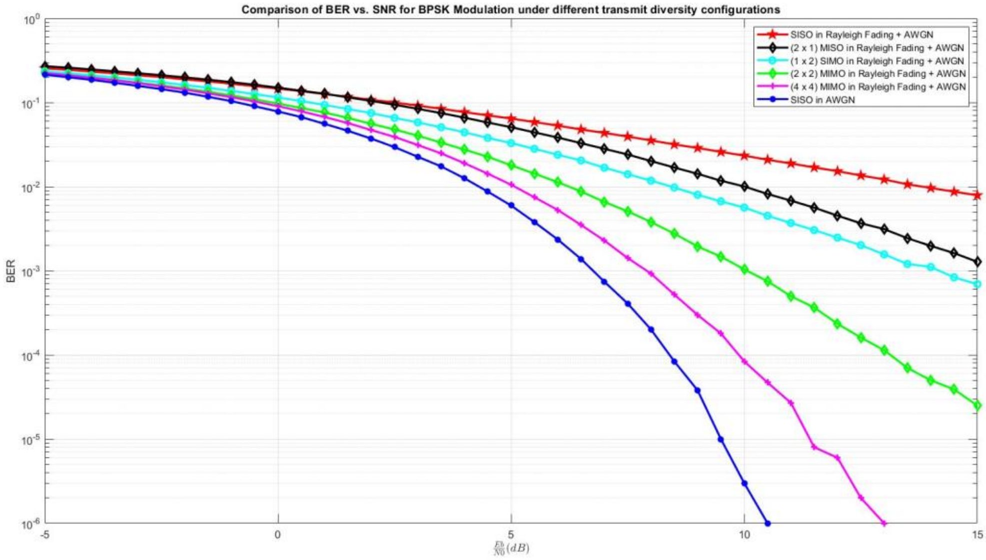

Figure 1: Comparison between the BER vs SNR graphs of different diversity configurations and SISO BPSK in Rayleigh fading channel

Observations and Conclusions

Multi antenna communication systems were designed and simulated using MATLAB. Different diversity techniques and diversity orders were simulated and their results were compared to each other and to the baseline SISO communication system affected by Rayleigh fading and AWGN, in order to study the advantage gained by using diversity to combat the effects of fading.

From *Figure 1*, it can be observed that:

1) There is a noticeable drop off in the performance of the BPSK modulated communication system when fading is introduced, as compared to when only AWGN is present.
2) Adding one or more antennas to the SISO system helps in improving the performance in the presence of fading.
3) Although, the total number of antennas in the communication system is the same for both the 2 x 1 setup (transmit diversity) that uses Alamouti Codes and the 1 x 2 setup (receive diversity) that uses MRC, the BER vs SNR curves obtained for the 1 x 2 setup is better. This is because the received SNR in case of the (1 x 2) setup is about 3 dB more than the 2 x 1 MISO system (this is caused due to lack of knowledge of the channel at the transmitter).
4) Both the MIMO configurations perform better than MISO and SIMO under Rayleigh fading.
5) The Alamouti codes used for the 2 x 1 MISO and 2 x 2 MIMO configurations are orthogonal over all time instants, hence, the received symbol can be computed by

exploiting the orthogonality of the composite channel matrix, without changing the additive noise characteristics.

6) The extended Alamouti Code used in the 4 x 4 MIMO configuration is not orthogonal over all the time instants. Hence, the estimates of the transmitted signal are obtained by using the zero-forcing approach, which could lead to noise amplification. But it can be observed that the 4 x 4 MIMO configuration still gives a BER performance that is closest to one obtained when only AWGN is present.