Homework project 3
EQ2411 - Advanced Digital Communication

Hareekeshav Sethumadhavan Srinivasan

I. Problems

The typed document contains answers to Problem 1 and plots of problem 2. the rest is in the written document along with this.

Problem 1

(a) The noise W is generated using the randn function in MATLAB, which generates samples from a Gaussian distribution with zero mean and unit variance. Therefore, each element of the noise vector W follows a complex Gaussian distribution with zero mean and variance var_w. so p(w) is CN(0, var_W).

The channel vector H is generated similarly using the randn function in MATLAB, where each element is an independent sample from a Gaussian distribution with zero mean and unit variance. Therefore, each component of H follows a complex Gaussian distribution with zero mean and variance 1. p(H) is CN(0,1)

1. **PDF of the Noise (W)**:

$$p(W) = \frac{1}{\pi \text{var}_W} e^{-\frac{|W|^2}{\text{var}_W}}$$

2. **PDF of the Channel Vector (H)**:

$$p(H) = \frac{1}{\pi} e^{-|H|^2}$$

(b) The updated MATLAB code for this problem will be available in the appendix

(c) For the following values

Figure 1 represents Number of symbols per block Ns = 10000, Number of diversity branches N = 2, Number of simulations Nsim = 1000.

Figure 2 represents Number of symbols per block Ns = 10000, Number of diversity branches N = 3, Number of simulations Nsim = 1000.

From the graphs, we can notice that selection combining, MRC and Equal gain combining w phase compensation achieve full diversity gain N. The switched diversity somewhat does it but the performance is not good enough. At high SNR values, the BER curves of combining methods achieving full diversity gain N will exhibit a linear decrease on a logarithmic scale.

The reason why the BER curve for Equal gain combining remains a straight line and does not curve downwards for increasing SNR is because it does not fully exploit the diversity available in the channel and does not use the channel information. Unlike techniques such as Maximum Ratio Combining (MRC), which adaptively combines signals based on their channel gains, Equal-gain combining simply combines all signals with equal weight.

When the full diversity gain N is not reached, it implies that the system's performance is not fully utilizing the available diversity provided by the diversity branches.

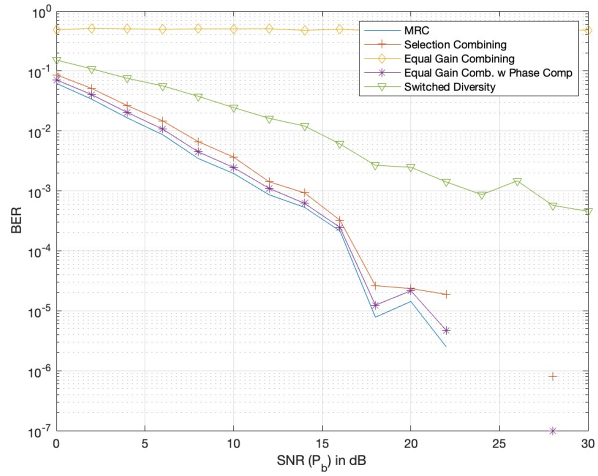

Fig. 1: N=2

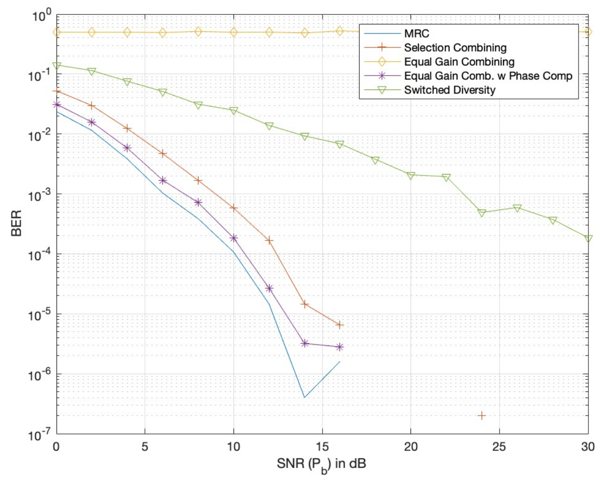

Fig. 2: N=3

Also, High levels of interference or noise can obscure the desired signal and degrade the performance of diversity combining techniques. In scenarios where interference or noise levels are high, the system may struggle to achieve the full diversity gain.

1. Maximum Ratio Combining (MRC):

- **Advantages**:
  - Achieves full diversity gain when channel knowledge is perfect.
  - Provides optimal performance in Rayleigh fading channels.
- **Disadvantages**:
  - Requires accurate channel state information at the receiver.
  - Higher complexity.

2. Selection Combining :

- **Advantages**:
  - Simplicity in implementation compared to more complex techniques.

  - Achieves diversity gain by selecting the best branch at each instant.
- **Disadvantages**:
  - Suboptimal performance compared to MRC.
  - Requires multiple receive antennas, increasing hardware complexity.

3. Equal Gain Combining :

- **Advantages**:
  - Simplicity in implementation, as it assigns equal weights to all branches.
  - No need for channel estimation or knowledge.
- **Disadvantages**:
  - Does not achieve full diversity gain in all scenarios.

4. Equal Gain Combining with Phase Compensation:

- **Advantages**:
  - Addresses phase differences between branches,improving performance.
  - Simplicity in implementation, performs better than selection combining.
- **Disadvantages**:
  - Requires additional phase compensation, increasing complexity compared to standard EGC.

5. Switched Diversity :

- **Advantages**:
  - Offers diversity gain by randomly selecting one branch, potentially mitigating fading.
  - Simplicity in implementation.
- **Disadvantages**:
  - Suboptimal performance compared to techniques like MRC and Selection Combining, especially in scenarios with rapidly changing channels.
  - May not achieve full diversity gain due to the random selection process.

Problem 2

Figure 3 and Figure 4 represent the plots of 2a annd 2b respectively

Appendix

MATLAB code

```
clear all;
close all;

% ---------------------------------------------------------------
% In this part, we will study BPSK transmission over a channel with N
% parallel diversity branches.
% ---------------------------------------------------------------



% ---------------------------------------------------------------
% Define basic parameters
```

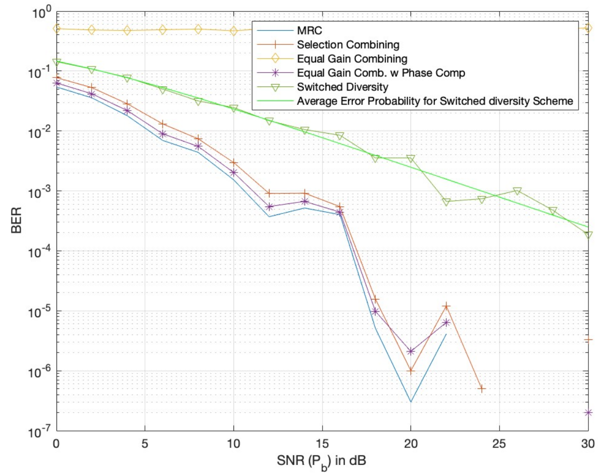

Fig. 3: average error probability of switched diversity compared with problem 1

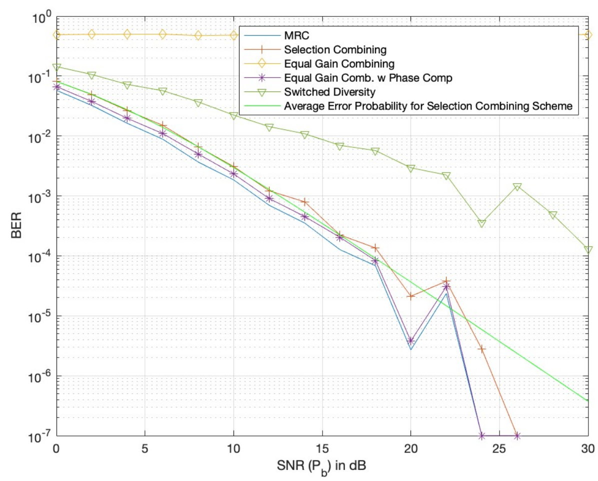

Fig. 4: average error probability of selection combining compared with problem 1

```
% ----------------------------------------------------------------------

% Number of symbols per block
Ns = 10000;

% Number of diversity branches
N = 2;

% Set the SNR range for the average SNR without fading
SNR_dB = [0:2:30];

% Number of simulations
N_sim = 1000;
```

```
% ----------------------------------------------------------------------
% Simulation
% ----------------------------------------------------------------------

% Initialize variables for the results

BER_MRC   = zeros(1,length(SNR_dB));
BER_SC    = zeros(1,length(SNR_dB));
BER_EQ    = zeros(1,length(SNR_dB));
BER_EQPC  = zeros(1,length(SNR_dB));
BER_SD  = zeros(1,length(SNR_dB));

for ii_sim = 1: N_sim

    for ii_snr = 1:length(SNR_dB)

        % ------------------------------------------------------------------
        % Generate a block of BPSK symbols

        X = 1-2*(rand(1,Ns)>0.5);


        % ------------------------------------------------------------------
        % Generate and simulate the channel
        % ------------------------------------------------------------------

        % Generate a fading vector
        H = 1/sqrt(2)*(randn(N,1) + j*randn(N,1));

        % Generate channel noise
        var_W = 10^(-SNR_dB(ii_snr)/10);

        W = (randn(N,Ns) + j*randn(N,Ns))*sqrt(var_W/2);


        % Simulate the channel
        Y = H*X + W;


        % ------------------------------------------------------------------
        % Diversity combining
        % ------------------------------------------------------------------

        % ------------------------------------------------------------------
        % Maximum ratio combining
        % ------------------------------------------------------------------

        % decision statistics
        Z_MRC = H'* Y;  % Add your code here
```

```
% HD symbol estimates
x_MRC = 1-2*(Z_MRC<0);

% calculate the bit-error rate for this block
ber_MRC = sum( ne(x_MRC,X) )/Ns;

% Update the average error probability
BER_MRC(ii_snr) =  BER_MRC(ii_snr) + ber_MRC/N_sim;


% ------------------------------------------------------------------
% Selection combining (i.e., select the path with the highest energy)
% ------------------------------------------------------------------

% received energy
Er = sum(abs(H).^2,2)/Ns;
[E_max,i_max] = max(Er);

% decision statistics
Z_SC = conj(H(i_max)) * Y(i_max, :);  % Add your code here

% HD symbol estimates
x_SC = 1-2*(Z_SC<0);

% calculate the bit-error rate for this block
ber_SC = sum( ne(x_SC,X) )/Ns;

% Update the average error probability
BER_SC(ii_snr) =  BER_SC(ii_snr) + ber_SC/N_sim;


% ------------------------------------------------------------------
% Equal gain combining
% ------------------------------------------------------------------

ones_vector = ones(N, 1);

% decision statistics
Z_EQ = ones_vector' * Y; % Add your code here

% HD symbol estimates
x_EQ = 1-2*(Z_EQ<0);

% calculate the bit-error rate for this block
ber_EQ = sum( ne(x_EQ,X) )/Ns;

% Update the average error probability
BER_EQ(ii_snr) =  BER_EQ(ii_snr) + ber_EQ/N_sim;

% ------------------------------------------------------------------
```

```
        % Equal gain combining with phase compensation
        % ----------------------------------------------------------------

        % Compute phase offsets
phi = angle(H);

% Initialize the combined signal
Z_EQPC = zeros(1, Ns);

% Perform phase compensation and combine the signals
for i = 1:N
    % Phase compensation: Multiply each received signal by the complex conj
    Y_compensated = exp(-1j * phi(i)) * Y(i, :);
    % Combine compensated signals
    Z_EQPC = Z_EQPC + Y_compensated;
end

        % decision statistics
%         Z_EQPC = % Add your code here

        % HD symbol estimates
        x_EQPC = 1-2*(Z_EQPC<0);

        % calculate the bit-error rate for this block
        ber_EQPC = sum( ne(x_EQPC,X) )/Ns;

        % Update the average error probability
        BER_EQPC(ii_snr) =  BER_EQPC(ii_snr) + ber_EQPC/N_sim;


        % ----------------------------------------------------------------
        % Switched diversity combining (pick one path at random)
        % ----------------------------------------------------------------

        % select a branch at random
        ii_branch = ceil(N*rand(1));

        % decision statistics
        Z_SD = conj(H(ii_branch)) * Y(ii_branch, :);% Add your code here

        % HD symbol estimates
        x_SD = 1-2*(Z_SD<0);

        % calculate the bit-error rate for this block
        ber_SD = sum( ne(x_SD,X) )/Ns;

        % Update the average error probability
        BER_SD(ii_snr) =  BER_SD(ii_snr) + ber_SD/N_sim;

        end
```

```
end

% ----------------------------------------------------------------
% Plot the results
% ----------------------------------------------------------------

semilogy(SNR_dB, BER_MRC,'-'), hold on, grid on
semilogy(SNR_dB, BER_SC,'-+')
semilogy(SNR_dB, BER_EQ,'-d')
semilogy(SNR_dB, BER_EQPC,'-*')
semilogy(SNR_dB, BER_SD,'-v')
xlabel('SNR (P_b) in dB')
ylabel('BER')
legend('MRC',...
    'Selection Combining',...
    'Equal Gain Combining',...
    'Equal Gain Comb. w Phase Comp',...
    'Switched Diversity')
```