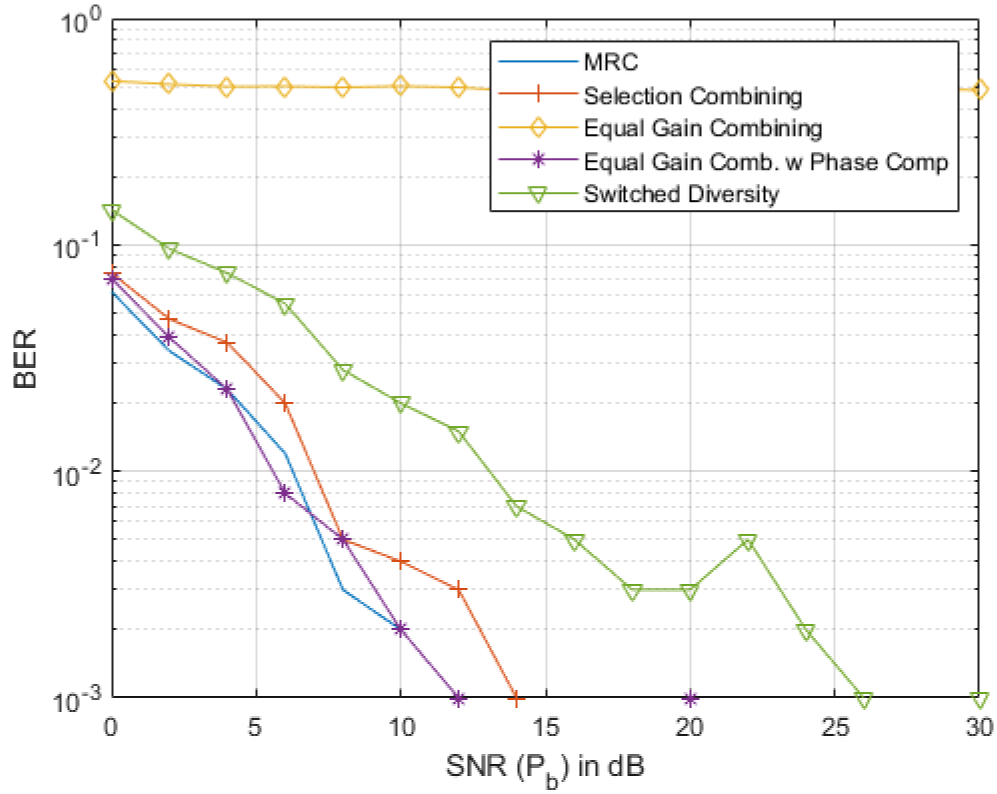
```
clear;
close all;
clc;

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



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

% Number of symbols per block
Ns = 1;

% 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 = 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
%
--------------------------------------------------------------


% decision statistics
one = [1;1];
Z_EQ = one'*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
%
--------------------------------------------------------------


% decision statistics
Z_EQPC = 0;
for i = 1:N
    Z_EQPC = Z_EQPC + exp(-1i*angle(H(i)))*Y(i);
end% 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 = 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')
```



Published with MATLAB® R2020a