

Contents

- [Transmitter](#)
- [Plotting](#)

```
clc;
clear;
close all;
clear
N = 10^6; % number of bits or symbols
```

Transmitter

```
ip = rand(1,N) > 0.5; % generating 0,1 with equal probability
s = 2*ip - 1; % BPSK modulation: 0 -> -1, 1 -> 1
Eb_N0_dB = [-3:35]; % multiple Eb/N0 values

for ii = 1:length(Eb_N0_dB)
    n = 1/sqrt(2) * [randn(1,N) + j*randn(1,N)]; %white Gaussian noise, 0dB variance
    %h = 1/sqrt(2) * [randn(1,N) + j*randn(1,N)];
    x_1 = [randn(1,N)];
    x_2 = [randn(1,N)];
    h = sqrt(x_1.^2 + x_2.^2); % Rayleigh channel
    % Channel and noise Noise addition
    y = h.*s + 10.^(-Eb_N0_dB(ii)/10) * n;
    % equalization
    yHat = y ./ h;
    % receiver - hard decision decoding
    %ipHat = real(yHat) > 0;
    ipHat = real(y) > 0;
    % counting the errors
    nErr(ii) = size(find(ip - ipHat), 2);
end

simBer = nErr/N; % simulated BER
theoryBerAWGN = 0.5 * erfc(sqrt(10.^(Eb_N0_dB/10))); % theoretical BER for AWGN
EbN0Lin = 10.^(Eb_N0_dB/10);
theoryBer = 0.5 * (1 - sqrt(EbN0Lin ./ (EbN0Lin + 1))); % theoretical BER for Rayleigh
```

Plotting

```
close all
figure
semilogy(Eb_N0_dB, theoryBerAWGN, 'cd-', 'LineWidth', 2);
hold on
semilogy(Eb_N0_dB, theoryBer, 'bp-', 'LineWidth', 2);
semilogy(Eb_N0_dB, simBer, 'mx-', 'LineWidth', 2);
axis([-3 35 10^-5 0.5])
grid on
legend('AWGN-Theory', 'Rayleigh-Theory', 'Rayleigh-Simulation');
xlabel('Eb/No, dB');
ylabel('Bit Error Rate');
title('BER for BPSK modulation in Rayleigh channel');
```

