clear

N = 10^6

rand('state',100);

randn('state',200);

% Transmitter

ip = rand(1,N)>0.5;

s = 2\*ip-1; % BPSK modulation 0 -> -1; 1 -> 1

n = 1/sqrt(2)\*[randn(1,N) + j\*randn(1,N)];

Eb\_N0\_dB = [-3:10];

for ii = 1:length(Eb\_N0\_dB)

% Noise addition

y = s + 10^(-Eb\_N0\_dB(ii)/20)\*n;

% receiver - hard decision decoding

ipHat = real(y)>0;

% counting the errors

nErr(ii) = size(find([ip- ipHat]),2);

end

simBer = nErr/N;

theoryBer = 0.5\*erfc(sqrt(10.^(Eb\_N0\_dB/10)));

% plot

close all

figure

semilogy(Eb\_N0\_dB,theoryBer,'b.-');

hold on

semilogy(Eb\_N0\_dB,simBer,'mx-');

axis([-3 10 10^-5 0.5])

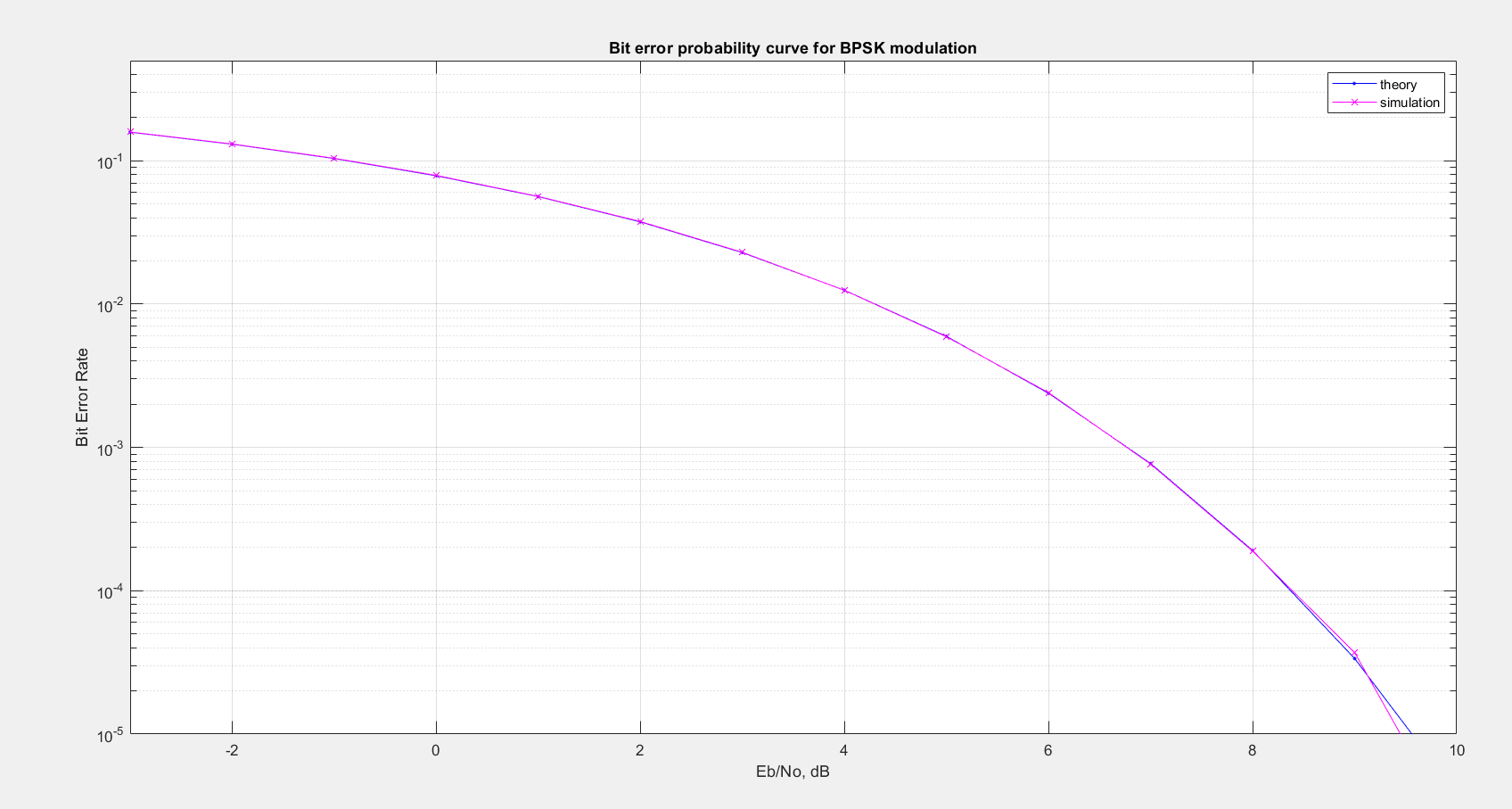
grid on

legend('theory', 'simulation');

xlabel('Eb/No, dB');

ylabel('Bit Error Rate');

title('Bit error probability curve for BPSK modulation');



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 -> 0

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)]; % Rayleigh channel

% Channel and noise Noise addition

y = h.\*s + 10^(-Eb\_N0\_dB(ii)/20)\*n;

% equalization

yHat = y./h;

% receiver - hard decision decoding

ipHat = real(yHat)>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

EbN0Lin = 10.^(Eb\_N0\_dB/10);

theoryBer = 0.5.\*(1-sqrt(EbN0Lin./(EbN0Lin+1)));

% plot

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');

