Computer Simulation of BPSK Digital Modulation Scheme Using MATLAB

Kyurim Rhee

Dec 06-2010

Abstract: In this paper, a statistical computer simulation will be performed to simulate a digital communication over a White Gaussian noisy channel with BPSK(Binary Phase Shifting Key) digital modulation scheme. The simulated performance will be measured in BER(Bit Error Rate) vs. Eb/No(Energy Per Bit/Noise) and will be compared to a published theoretical performance. This paper will go over all aspects of the simulation including the random variable generator, applying the BPSK scheme in the simulation, the logic behind the method of simulation, and comparison of the result of the simulation with the theoretical values.

1. Introduction

In modern communication, digital modulation schemes have taken over analogue modulation schemes. There are three main types of keying techniques. They are ASK(Amplitude Shift Keying), FSK(Frequency Shift Keying), and PSK(Phase Shift Keying). These techniques differentiate the transmitted and received signal by varying Amplitude, Frequency and Phase of the signal for ASK, FSK, and PSK, respectively. Using these modulation schemes, the signal sent can be translated into an abstract idea called "symbols" which is used to determine the original bits transmitted and received. For the different keying techniques, there are multiple variations of modulation schemes which vary in the number of bits a symbol represents or the layout of the constellation. As the focus of this paper is on the simulation aspect of digital modulation scheme, this paper will focus on the most fundamental modulation scheme, BPSK. The block diagram below is a model of a end-to-end digital communication. This simulation will take place between the transmitting and receiving modulators highlighted in gray.

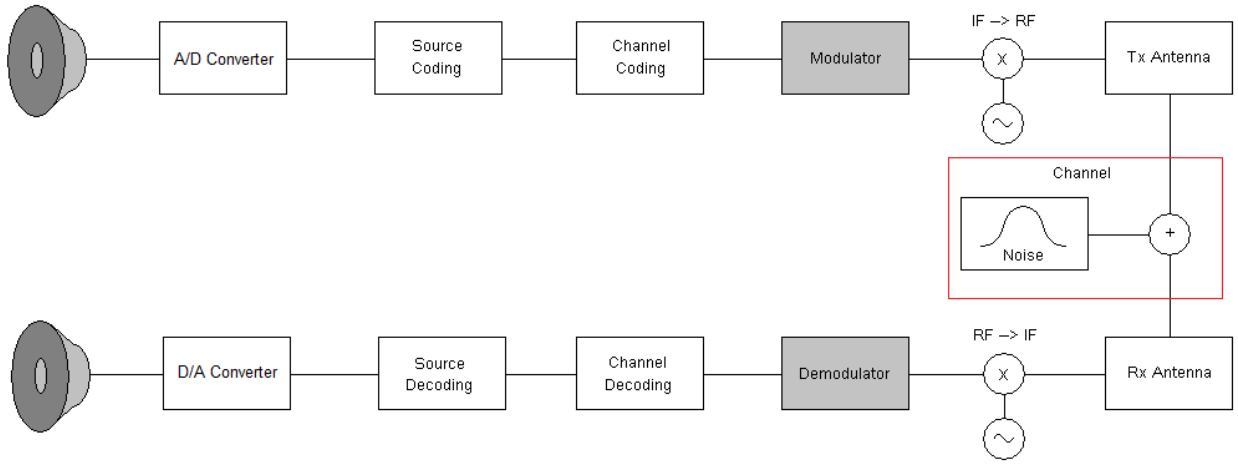

Figure 1: Digital Communication Block Diagram

2. Converting binaries into symbols

BPSK is a Binary Phase Shift Keying, meaning that for each symbol that is being transmitted, there is two possible locations where the symbols can reside. If a "0" is sent, then symbol with an energy of $-\sqrt{E_b}$ is sent. If "1" is sent, then symbol with an energy of $\sqrt{E_b}$ is sent, where Eb is the Energy per Bit. Below diagram shows these two points on the symbol constellation.

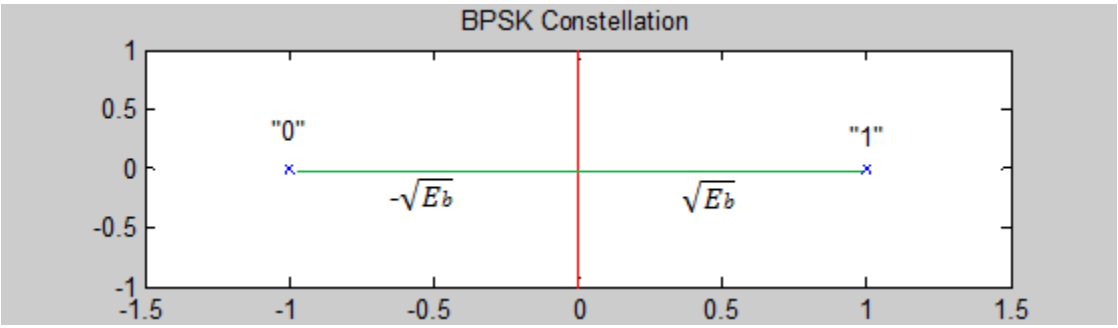

Figure 2: BPSK Constellation

This scenario can be simulated on MATLAB by first creating the binary random variable which represents the data that is being transmitted. With this data now available, we can translate this value into the BPSK symbols that will be transmitted over the link. The BPSK signals are defined in the following section.

3. BPSK Symbol Equations

Decision Region of BPSK is defined as follows:

Two BPSK signals, $S_m(t)$, m = 0, 1, are:

$$s_0(t) = \sqrt{E_b}\,\Phi(t) \qquad 0 \le t \le T_b$$

$$s_1(t) = \sqrt{E_b}\,\Phi(t) \qquad 0 \le t \le T_b$$

- E_b - Energy Per Bit
- T_b - Bit Duration
- $\Phi(t)$, the basis function is represented as:

$$\Phi(t) = \begin{cases} \sqrt{\frac{2}{T_b}}\cos(2\pi f_c t) & 0 \le t \le T_b \\ 0 & else \end{cases}$$

Therefore BPSK can then be represented as follows:

$$s_0(t) = \sqrt{\frac{2E_b}{T_b}}\cos(2\pi f_c t) \qquad 0 \le t \le T_b$$

$$s_1(t) = \sqrt{\frac{2E_b}{T_b}}\cos(2\pi f_c t) \qquad 0 \le t \le T_b$$

[1]

Applying this equation on MATLAB, we can now plot the constellation of symbols as shown on Figure 2.

4. Transmission (Adding White Gaussian Additive Noise)

The symbols are now ready to be transmitted. On the link, the signal will experience noise which will have a Gaussian distribution as shown in Figure 3.

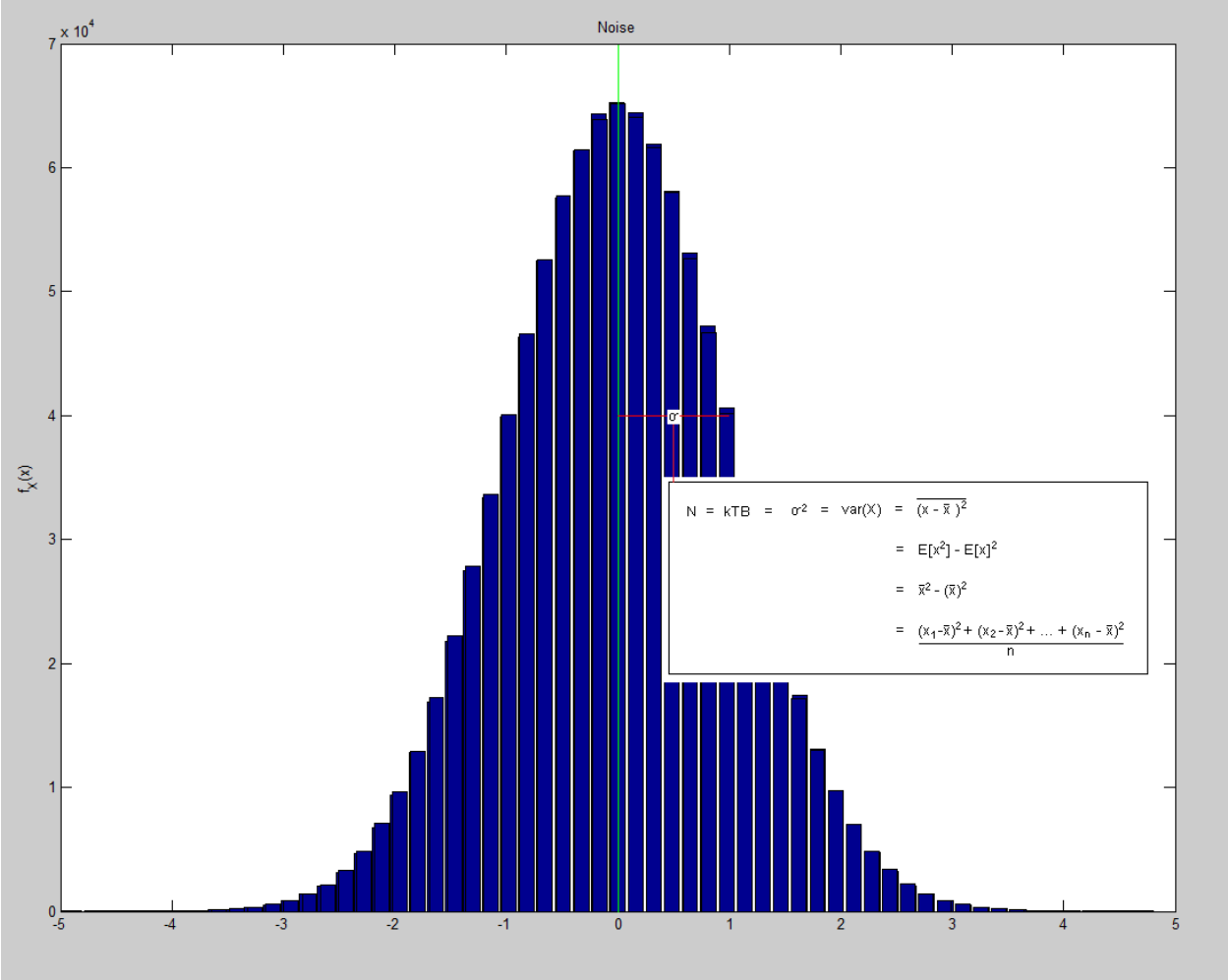

Figure 3: White Gaussian Noise

This noise will be added to the signal and will cause distortion to the signal. If the signal experiences noise that is very large, the noise may cause the signal to be read as a wrong value at the receiver. This effect can be shown on Figure 4. This diagram shows the "0" bits sent as red Xs. The Xs that have crossed over the green zero thresh hold to the "1" region are the bits that are transmitted with error. BER (Bit Error Rate) can now be calculated by simply counting all the bits that have crossed over and comparing it to the total number of bits sent.

As Pr(0 sent|1 received) = P(1 sent|0 received) [4], we can simplify the simulation by setting all the transmitted bits to either all "0", or all "1".

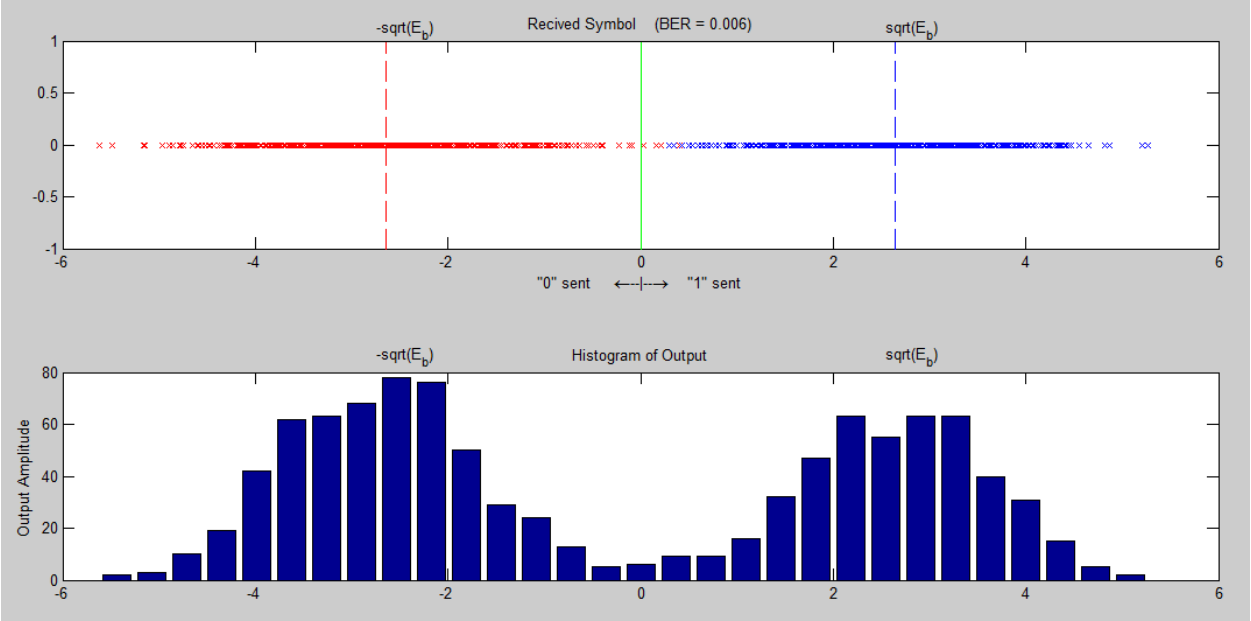

Figure 4: BPSK with Additive White Gaussian Noise

5. Simulation Results

The results of the simulation are shown in figures 5 and 6. As the Eb/No is increased, the accuracy of the symbols arriving in the intended region are increased. This results in less bits arriving in error. The same can be seen on figure 6 which shows the distribution of the symbols. The number of bits that are spilling across the threshold value of zero is reduced as Eb/No is increased.

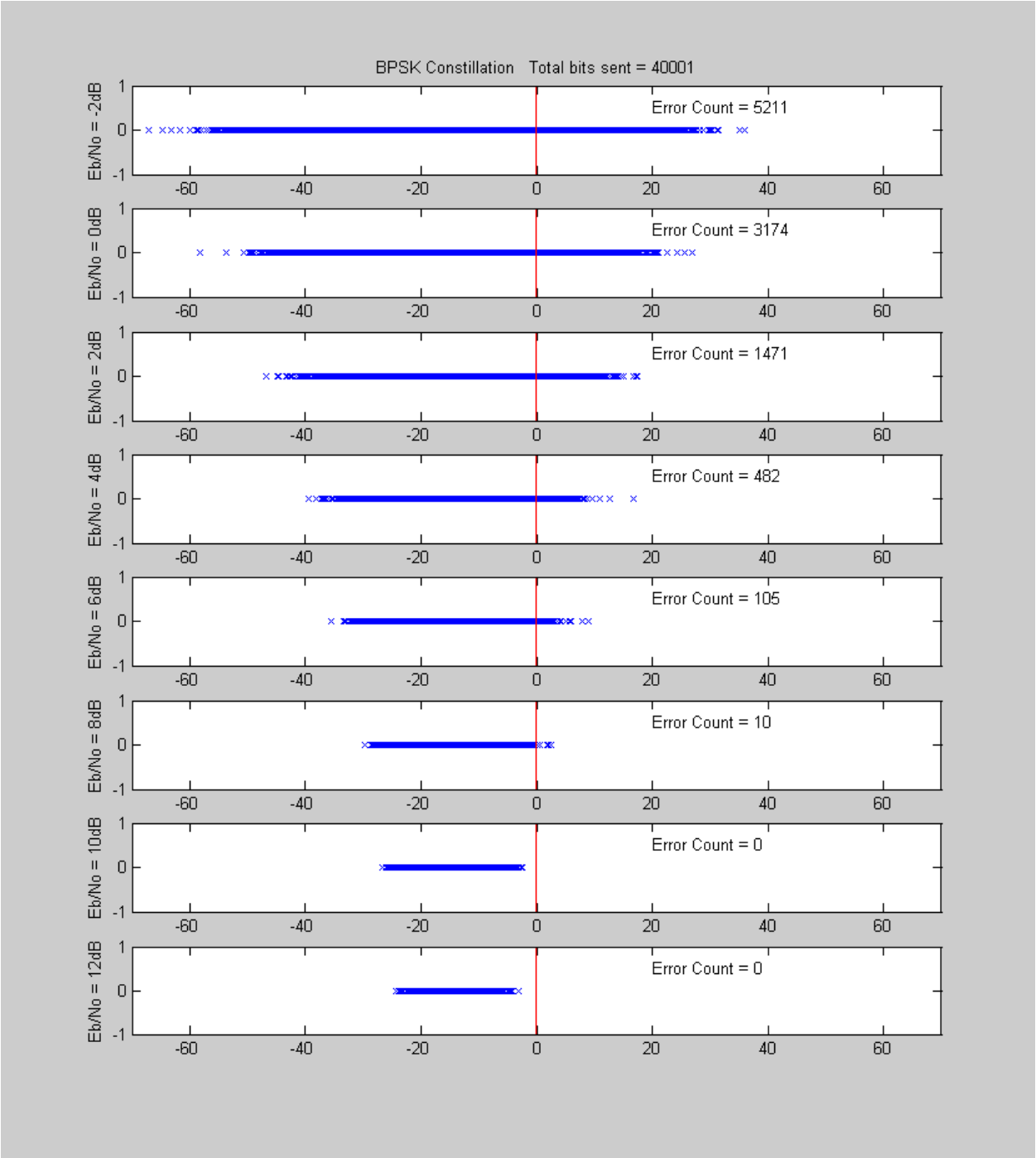

Figure 5: BPSK received symbols with varying Eb/No, all the bits set to “0”.

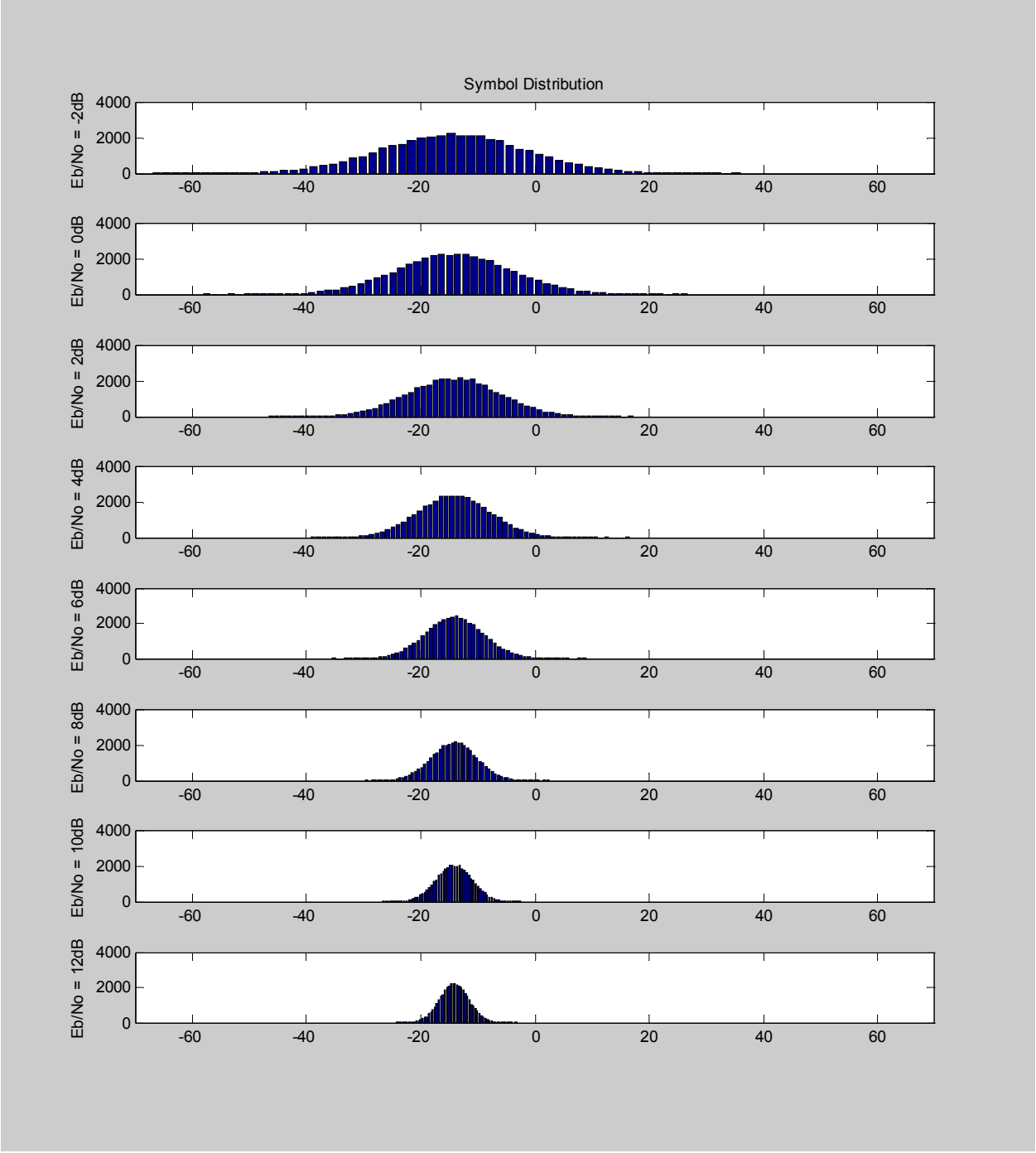

Figure 6: BPSK received symbol distribution.

6. Comparison of Simulation values to Theoretical Values

As noted by the Figure 5, the performance of the BPSK modulation, measured in BER, can be observed as a function of Eb/No. The simulated values can be gathered by collecting the BER for each of the Eb/No simulations that was run which are shown as red circles in Figure 7. The theoretical value of BPSK in AWGN is defined as:

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} e^{-t^2/2} dt = \frac{1}{2} erfc\left(\frac{x}{\sqrt{2}}\right), \quad x \geq 0$$

This value is shown as the blue line curve below.

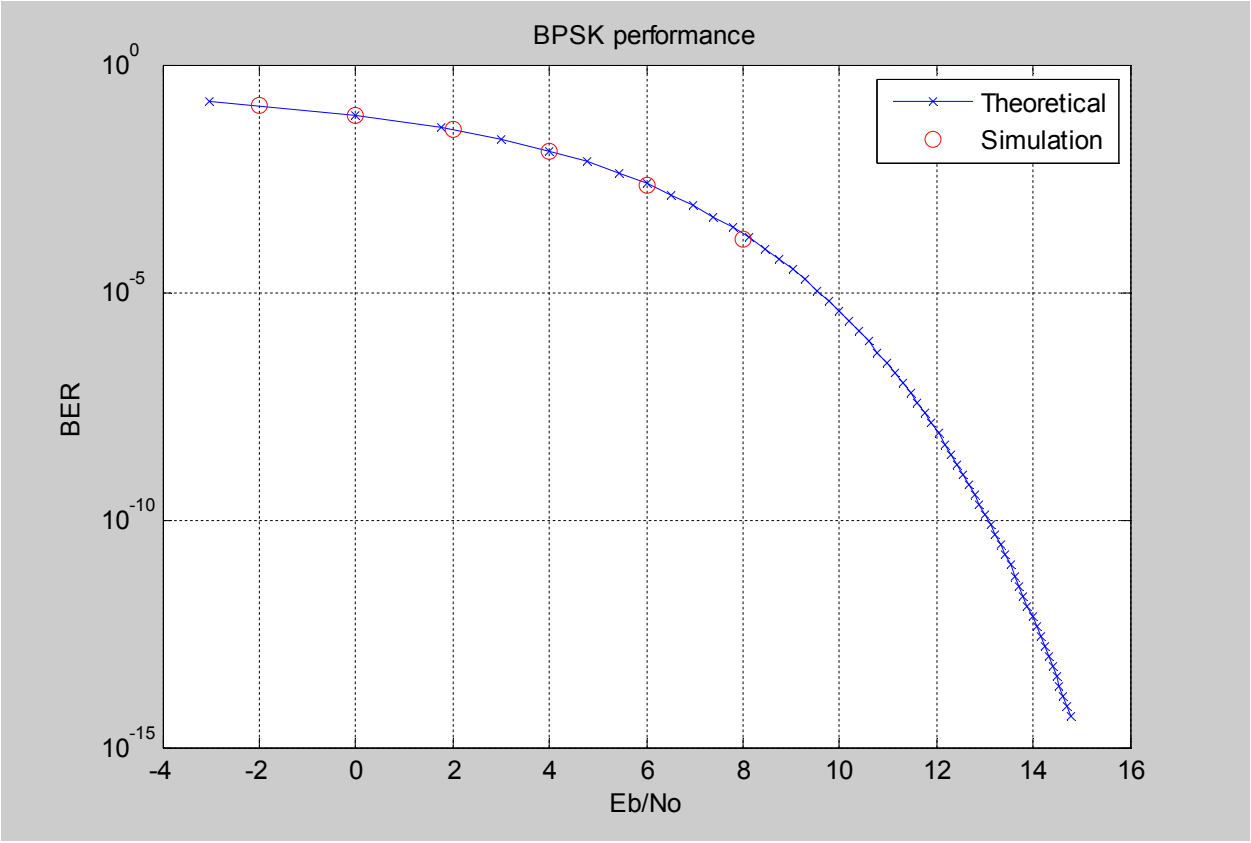

Figure 7: BPSK Performance (Eb/No vs. BER)

Note that the simulated values do not go further than Eb/No of 8dB. This is because as Eb/No increases, there are far less frequent number of errors occurring. To capture these instances of error, one would have to run this simulation with an extremely large number of samples, which was not feasible on a personal PC.

APPENDIX

MATLAB Code:

```
clear all; clc;

n1 = 1;
n0 = 0;
Eb = 100;
Tb = 1;
fc = 2.4*10^9;
EBNO = ([10^-.2, 10^0, 10^.2, 10^.4 10^.6, 10^.8, 10^1.0, 10^1.2]);   % base 10 values


for j = 1:length(EBNO)
%j = 1;
        t = linspace(0, Tb, 40001);%0:.001:Tb;
        k = length(t);
        % Create binary sequence
        % Bin = round(rand(1, k));
        Bin = zeros(1, k);

        % Define Noise
        No = Eb./EBNO;%.5;
        N = No(j)^.5 * randn(1,k);

        % Symbol Constillation
        Stx1 = (2*Eb/Tb)^.5*cos(2*pi*fc*t+pi*(1-n1));
        Stx0 = (2*Eb/Tb)^.5*cos(2*pi*fc*t+pi*(1-n0));

        % If Binary Code is "0"-send s0, if "1"-send s1
        for i = 1:k
            if Bin(i) == 0;
                Stx(i) = Stx0(i);
            else Bin(i) == 1;
                Stx(i) = Stx1(i);
            end
        end

        % Channel - Noise Gets added
        Srx = Stx + N;
        SRX(j,:) = Srx;

        % Test for values that were in Error

        Error = 0;
        ErrorNONE = 0;
        for i = 1:k
            if(Srx(i) >= 0)
                Error = Error +1;
            elseif (Srx(i) < 0)
                ErrorNONE = ErrorNONE +1;
            else
                PROB = 1
            end
        end
        e(j) = Error;
        e_N(j)= ErrorNONE;
        BER(j) = Error/(Error+ErrorNONE);
        Eb_No(j) = Eb/No(j);
        Eb_No_dB(j) = 10*log10(Eb_No(j));



end
e
e_N
No
BER
Eb_No
Eb_No_dB

figure(1); z = length(EBNO)
subplot(z,1,1); plot(SRX(1,:), zeros(1,k), 'x', [0 0],[-1 1],'r'); xlabel(''); ylabel('Eb/No = -2dB'); xlim([-70 70]); text(20,.5,['Error Count = ',num2str(e(1))]); title(['BPSK Constillation
Total bits sent = ',num2str(k)])
subplot(z,1,2); plot(SRX(2,:), zeros(1,k), 'x', [0 0],[-1 1],'r'); xlabel(''); ylabel('Eb/No = 0dB'); xlim([-70 70]); text(20,.5,['Error Count = ',num2str(e(2))]);
subplot(z,1,3); plot(SRX(3,:), zeros(1,k), 'x', [0 0],[-1 1],'r'); xlabel(''); ylabel('Eb/No = 2dB'); xlim([-70 70]); text(20,.5,['Error Count = ',num2str(e(3))]);
subplot(z,1,4); plot(SRX(4,:), zeros(1,k), 'x', [0 0],[-1 1],'r'); xlabel(''); ylabel('Eb/No = 4dB'); xlim([-70 70]); text(20,.5,['Error Count = ',num2str(e(4))]);
subplot(z,1,5); plot(SRX(5,:), zeros(1,k), 'x', [0 0],[-1 1],'r'); xlabel(''); ylabel('Eb/No = 6dB'); xlim([-70 70]); text(20,.5,['Error Count = ',num2str(e(5))]);
subplot(z,1,6); plot(SRX(6,:), zeros(1,k), 'x', [0 0],[-1 1],'r'); xlabel(''); ylabel('Eb/No = 8dB'); xlim([-70 70]); text(20,.5,['Error Count = ',num2str(e(6))]);
subplot(z,1,7); plot(SRX(7,:), zeros(1,k), 'x', [0 0],[-1 1],'r'); xlabel(''); ylabel('Eb/No = 10dB'); xlim([-70 70]); text(20,.5,['Error Count = ',num2str(e(7))]);
subplot(z,1,8); plot(SRX(8,:), zeros(1,k), 'x', [0 0],[-1 1],'r'); xlabel(''); ylabel('Eb/No = 12dB'); xlim([-70 70]); text(20,.5,['Error Count = ',num2str(e(8))]);

figure(2); z = length(EBNO)
subplot(z,1,1); [BAR,A] = hist(SRX(1,:), 60); bar(A, BAR); xlabel(''); ylabel('Eb/No = -2dB'); xlim([-70 70]); title('Symbol Distribution')
subplot(z,1,2); [BAR,A] = hist(SRX(2,:), 60); bar(A, BAR); xlabel(''); ylabel('Eb/No = 0dB'); xlim([-70 70])
subplot(z,1,3); [BAR,A] = hist(SRX(3,:), 60); bar(A, BAR); xlabel(''); ylabel('Eb/No = 2dB'); xlim([-70 70])
subplot(z,1,4); [BAR,A] = hist(SRX(4,:), 60); bar(A, BAR); xlabel(''); ylabel('Eb/No = 4dB'); xlim([-70 70])
subplot(z,1,5); [BAR,A] = hist(SRX(5,:), 60); bar(A, BAR); xlabel(''); ylabel('Eb/No = 6dB'); xlim([-70 70])
subplot(z,1,6); [BAR,A] = hist(SRX(6,:), 60); bar(A, BAR); xlabel(''); ylabel('Eb/No = 8dB'); xlim([-70 70])
subplot(z,1,7); [BAR,A] = hist(SRX(7,:), 60); bar(A, BAR); xlabel(''); ylabel('Eb/No = 10dB'); xlim([-70 70])
subplot(z,1,8); [BAR,A] = hist(SRX(8,:), 60); bar(A, BAR); xlabel(''); ylabel('Eb/No = 12dB'); xlim([-70 70])

figure(3)
hist(N), title('noise')

% Comparison with the Theoretical BPSK Value

Eb_No_THEO = 0:.5:30;
Eb_No_dB_THEO = 10*log10(Eb_No_THEO);
BER_THEO = .5*erfc((Eb_No_THEO).^.5);

figure(10)
semilogy(Eb_No_dB_THEO, BER_THEO, 'xb-',...
        Eb_No_dB, BER, 'or'); title('BPSK performance'); ylabel('BER'); xlabel('E_b/N_o (dB)'); grid on;
% subplot(212); stem(Eb_No_dB_THEO); title('BPSK performance'); ylabel('BER');
xlabel('Eb/No')



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Gaussian Dist Graph

N = randn(1, 1000000);
St = std(N);
[BAR,A] = hist(N, 60); bar(A, BAR);
hold on;
plot([0 0 ], [0 7*10^4], 'g-',...
        [0 St], [4*10^4 4*10^4], 'r-')
title('Noise'); ylabel('f_X(x)')
```

REFERENCES

[1] J. Mark & W, Zhuang. (2003) "Wireless Communication and Networking"
Pearson Education, Inc

[2] P. Nikitin, E. Normark, C. Wakayama, & R. Shi, (2010, Dec 01) "VHDL-AMS modeling and simulation of BPSK transceiver system"
University of Washington, Department of Electrical Engineering, Seattle, WA 98195-2500, USA
http://www.ee.washington.edu/faculty/nikitin_pavel/papers/ICCSC_2004.pdf

[3] Wikipedia, (2010, Dec 01) "Phase Shift Keying"
http://en.wikipedia.org/wiki/Phase-shift_keying#cite_note-4

[4] S. Sud, (2010 Nov 08) "Lecture 10: Digital Modulation"
TCOM551 Digital Communication, George Mason University