Simulation of the Detector Performance for Binary Antipodal Signals

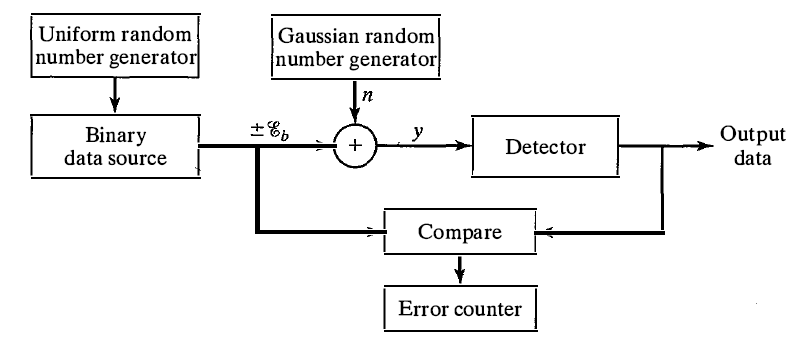
**Aim:**

To estimate and plot the probability of error when binary antipodal signals are transmitted over an additive white Gaussian noise channel.

The purpose of this problem is to estimate and plot the probability of error when binary antipodal signals are transmitted over an additive white Gaussian noise channel. The model of the binary communication system employing antipodal signals is shown in Figure. As shown, we simulate the generation of the random variable y, which is the input to the detector. A uniform random number generator is used to generate the binary information sequence of zeros and ones from the data source. The sequence of zeros and ones is mapped into a sequence of ±1. A Gaussian noise generator is used to generate a sequence of zero-mean Gaussian numbers with variance σ2. For equally probable zeros and ones, the detector compares the random variable y with the threshold zero. (Cross correlation can also be applied) If y > 0, the decision is made that the transmitted bit is a zero. If y < 0, the decision is made that the transmitted bit is a 1. The output of the detector is compared with the transmitted sequence of information bits, and the bit errors are counted.

Perform the simulation for the transmission of 10,000 bits at several different values of SNR, which covers the range of SNR 0 to 10. Plot the error · probability as a function of the SNR.

Compare the estimated error probability with the theoretical error probability given by the formula



**Tools used : Python 2.7**

**Steps for Simulation**

A- Simulation of the Detector Performance for Binary Antipodal Signals In this part you will simulate - in python - the performance of a BPAM communication system transmitting 10,000 binary antipodal signals over an additive white Gaussian noise (AWGN) channel.

**Generation of signals at the transmitter side**

• Generate the random binary information sequence b of 0's and 1’s from the data source. Assume that the bits are equally probable. You can use a uniform random number generator function to generate numbers in the range [0, 1]. If a number generated is in the range [0, 0.5], then you can consider that the binary source output is a '0'. Otherwise, it is a 1'.

• Map the previously generated sequence of bits into a sequence of symbols s. The bits '0' and '1' will be mapped into symbols 1 and -1 respectively, where to represents the signal energy per bit. For convenience, you may normalize the signal energy unity.

**Channel Model**

• Use a Gaussian noise generator is used to generate a sequence n of zero-mean Gaussian numbers with variance =N0/2. The values of may be defined by the values of the SNRs in dB.

SNRdB= 10 log10(S/N)

SNRdB/10= log10(S/N)

10^SNRdB/10 =S/N

S/N = 10^SNRdB/10

N/S = 10^(-SNRdB/10)

N = S\*10^(-SNRdB/10)

Variance = S\*10^(-SNRdB/10)

Variance = S\*10^(-SNRdB/10)

Sigma\*Sigma = S\*10^(-SNRdB/10)

Sigma = S\*10^(-SNRdB/20)

n = np.random.normal(mean,sigma,10000)

**Detection of bits at the receiver side**

• Assume that the detector will use the optimal threshold a\*, compare the received signal r= s + n with a\* to decide whether the originally transmitted bit is '0' or '1'.

**Count the number of bit errors.**

**Performance of the system**

• Vary the values of (sigma)2, and perform the simulation for the transmission of the 10,000 bits at several different values of SNR, which covers the range of SNRs 0< 10 log10(Eb/N) = 10 dB.

Deliverables

a) In this part you will assume that the transmitted symbol is always +1. Plot the distributions of the received signals r when the SNR is 2 dB, and 10 dB.

Use the histogram

import numpy as np

import math as ma

import matplotlib.pyplot as plt

s = np.random.normal(0,1,10000)

count, bins, ignored = plt.hist(s, 100, density=True)

plt.plot(bins, np.ones\_like(bins), linewidth=2, color='r')

plt.show()

b) In this part you will assume that the transmitted symbol could be +1 or -1. In a new figure, plot the error probability P as a function of the SNR.

c) In the same figure of part b), plot the theoretical error probability given by the formula

use the math.erfc(x)

\displaystyle{ Q(z) = \frac{1}{2} erfc \left( \frac{z}{\sqrt{2}}\right)}  \quad\quad  (9) 

Sample Results



Histogram of the noise added (mean zero, variance 1)





**Result:**

Plotted the probability of error for binary antipodal signals over an additive white Gaussian noise channel and found that SNR increase Probability of error decreases.