**Error rate of the AWGN channel with antipodal signaling**

Deriving a closed for expression for the bit error rate (BER) of a channel is not always possible. In such cases it is necessary to use computer simulations to find the BER for some specific Signal to Noise Ratio (SNR) or some other parameter.

This assignment deals with simulation of error probability in an AWGN channel (Additive White Gaussian Noise) using Matlab/Python. Alternatively, you can use C/C++ and plot your results with some other program. However, MATLAB/Python are far better than C/C++ when it comes to vector/matrix operations. You need to use some of the Matlab built in functions such as qfunc(), which computes the Q-function). Python does not have a built in Q-function (as far as I remember) but you can use the error function erf() to calculate the Q-function. You can read more about Q-function [here](http://www.wikiwand.com/en/Q-function) or [here](https://en.wikipedia.org/wiki/Q-function).

1. Generate a binary (0 and 1) sequence of length ten million with bit probabilities . Map each zero to −1 and each 1 to +1 to generate an antipodal signaling scheme (binary ASK or binary PSK are both examples of antipodal signaling) with (the constellation is , which in this case is simply ±1). Verify the number of zeros and ones to make sure they are about five million each (it is ok for the numbers to off a little bit).
2. For six values of SNR:

compute the variance of the noise sequence (basically, since is fixed, you need to adjust the variance of the noise sequence so that SNR becomes some specific value), generate the corresponding noise sequence. So you want to generate six sequences of zero-mean, IID Gaussian (AKA normal) random variables, each with length ten million and some variance. Add each noise sequence to the data sequence you have already generated from part one to obtain the received sequence for that particular and then use the optimal detection rule to detect the received sequences. The optimal decision rule (in this case) would map received signal to 1 and 0 based on the sign of the received signal (i.e, if the received signal is positive you map it to one else you map it to 0).

1. Since you are adding noise to the original sequence, not all bits are received correctly. By comparing detected and transmitted sequences determine for each value of . Plot as a function of in dB, using logarithmic scale for . You should get a plot similar to the one we had in the slides (for example see Figure 6.8 and 6.9 in Stallings’ book).
2. On the same plot, draw a plot of the equation that gives you the value of as a function of in binary antipodal signaling, i.e.,

In your plot, again use logarithmic scale for and dB for and compare the two plots. Where do you see better agreement between the two plots, at lower or higher SNR values? Justify your answer.

1. Repeat parts 1—3, but with . In this case use the same values for as in part 2. Compare the results with part 1—3. Can you explain the difference in the resulting error rate plots? Which one performs worse? Can you suggest a way to improve the error rate in that case?