CMPE 576 Spring 2018: Assignment 2

Due date: 3 May 2018, in class
You may work in groups of two. In that case, please submit only one copy of the assignment with two names on it.
– Please submit a hardcopy and email me a softcopy of your assignment.
– Do not copy programs from each other.
– You might be asked to demonstrate the running of your programs.

Monte Carlo Simulation of Binary Transmission over a Gaussian Channel

Write a C/C++ or Matlab program to simulate binary transmission over a Gaussian channel in order to determine the capacity of the channel. A source generates bits where "0"s and "1"s are *equally likely*. The signal level for a "0" is $-\sqrt{P}$ and that for a "1" is $\sqrt{P}$. When signals are transmitted through the channel, they are corrupted by additive white Gaussian noise (AWGN) that is distributed as $\mathcal{N}(0, \sigma^2)$. More specifically, at discrete time i, the received signal Y_i is given by

$$Y_i = X_i + Z_i,$$

where X_i is random variable denoting the transmitted signal level by the source and $Z_i \sim \mathcal{N}(0, \sigma^2)$. The receiver decides that a "0" was sent if $Y_i \leq 0$, and that a "1" was sent if $Y_i > 0$.

In your program, the user must be able to specify the signal-to-noise ratio (SNR) value to be used in simulation. For this assignment, let us set the noise power $\sigma^2 = 0.25$ (i.e. you can make σ^2 a constant). Recall that the SNR is given by P/σ^2, or $10\log_{10}(P/\sigma^2)$ decibels (dB). You must implement your own Gaussian random number generator using either the Box-Muller or the Polar Method. Your program should perform the following steps to calculate the capacity of the channel:

1. Get the SNR value in dB as a parameter.
2. For 100,000 (or more) discrete time units, generate bits (0s and 1s being equally likely), transmit the signals corresponding to the bits through the noisy channel, and make a decision whether a "0" or a "1" was sent based on the value of Y_i.
3. Determine how many bits were detected incorrectly by the receiver. The fraction of bits that are transmitted unsuccessfully is an *estimate* of the cross-over probability $\hat{\epsilon}$.
4. Compare the estimated value $\hat{\epsilon}$ to the theoretical value, $\epsilon = 1 - \Phi(\sqrt{P}/\sigma)$.
5. Calculate the binary entropy function using the *estimate* of the cross-over probability: $H_b(\hat{\epsilon}) = -\hat{\epsilon}\log_2(\hat{\epsilon}) - (1-\hat{\epsilon})\log_2(1-\hat{\epsilon})$.
   Your program should correctly handle the case in which $\hat{\epsilon} = 0$!

6. Calculate the channel capacity using the *estimate* of the cross-over probability: $C(\hat{\epsilon}) = 1 - H_b(\hat{\epsilon})$.

Use an application such as Excel or Matlab to **plot** the value of the channel capacity as a function of the SNR where the SNR ranges between -20 dB and 20 dB with 1 dB intervals.

Appendix. Approximating $\Phi(z)$

$$\Phi(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z} e^{-x^2/2} dx.$$

You may use the following method for approximating $\Phi(z)$ for $z \geq 0$:

$c_1 = 0.127414796$

$c_2 = -0.142248368$

$c_3 = 0.710706871$

$c_4 = -0.726576013$

$c_5 = 0.530702714$

$p = 0.2316419$

$t = 1/(1 + pz)$

$r = t(c_1 + t(c_2 + t(c_3 + t(c_4 + tc_5))))$

$\Phi(z) = 1 - r \exp(-z^2/2)$

For $z < 0$, note that $\Phi(-z) = 1 - \Phi(z)$.