ECE3522 Practicum 2 : Bit Error Rate

Receive 5 additional points by submitting before 10/27/18 11:59 pm
Penalty will be imposed after 11/3/18 11:59 pm (1 point each hour; 5 points each day)

The sample Matlab code in the appendix computes the number of erroneously detected bits when transmitting binary data. In the code, a binary bit stream of "symbol 1" and "symbol 0" is transmitted. The total number of transmitted bits is *N*=10,000.

When "symbol 1" is transmitted, the actual signal level is 1. When "symbol 0" is transmitted, the actual signal level is -1.

At the receiver, the signal is added with noise, which follows a normal (Gaussian) distribution. The level of noise is determined by the input signal-to-noise ratio (SNR), which is typically given in dB. Given an SNR of a dB, the noise standard deviation is given by $10^{-a/20}$.

At the receiver, if the noisy signal (signal + noise) corresponding to a bit is higher than 0, the bit is decided as "symbol 1". Otherwise, the bit is decided as "symbol 0". The detected bit stream is compared with the transmitted data. For any bit that the detected data and the transmitted data are different, that bit is said to be erroneously detected.

1. When "symbol 1" is transmitted, a noisy signal is described as a random variable *X* following the normal distribution with expected value of 1 and standard deviation of $10^{-a/20}$. The probability to make an erroneous detection is equal to the probability that $P(X < 0)$ in this case. Compute this probability based on the cumulative distribution function (CDF) as:

   $$F_X(x) = \Phi\left(\frac{x-\mu}{\sigma}\right)$$

   where μ is the expected value of X, and σ is its standard deviation. The CDF $\Phi(z)$ for the standard normal distribution can be computed using Matlab function: `normcdf(z)`.

   You can report the above result as the theoretical BER. However, note that, precisely, the BER is given by $P_e = P[\text{Receive } 0|\text{transmit } 1]P[\text{transmit } 1] + P[\text{Receive } 1|\text{transmit } 0]P[\text{transmit } 0]$, which takes the same value when the decision threshold sits in the middle point of "symbol 1" and "symbol 0".

2. In the Matlab code, change the number of transmitted bits, *N*, to 100,000. Compute the simulated bit error rate (BER), which is the ratio between the number of erroneously detected bits and the total number of transmitted bits. Repeat this for 100 times and compute the average BER by averaging the results over the BER obtained from the 100 trials. Compare this averaged BER with the error probability $P(X < 0)$ computed in part 1. These two values should be close.

3. Change the input SNR to 6 dB. Repeat the above steps and report the theoretical error probability (as in part 1) and the simulated average BER (as in part 2). Comment how the error probability and average BER change as the input SNR increases.

Submit by e-mail to ece3522.temple@gmail.com

(a) A single Word or pdf file containing materials in the following order:

- Your name, TUid
- All the required values, figures, and observations
- Key references (website, book, paper, or student name) if any (particularly if you reused a substantial portion of the Matlab codes)
- Matlab codes

(b) All Matlab codes in .m format

Important: The Matlab code has to be included in the Word or pdf file report, AND as a separate attachment. Do not zip the files.

Requirement

A student can seek help, but the report must be his/her work and thus he/she has to understand every word and every line of Matlab code reported. In-class Quiz may be taken place after the submission of the practicum report to verify whether the students understand what they described in this report.

Appendix

```
%% ========== Matlab code sample for Practicum 2 ===========
% Simulation: Compute number of erroneously detected bits
%

clear

N = 10000;                    % number of bits transmitted

SNR = 5;                      % signal-to-noise ratio in dB

signal = randi([0 1], N, 1);     % bit stream with 0's & 1's

noise = randn(N,1);           % additive Gaussian noise

received = (signal*2-1) + noise * 10^(-SNR/20);
                              % received noisy signal

detect = (received > 0);      % detected result

num_error = sum(abs(detect - signal));
                              % number of erroneously detected bits

fprintf('%d bits transmitted; %d bits of errors detected\n', N, num_error)
```

Results:

```
10000 bits transmitted; 374 bits of errors detected
```