CALIFORNIA INSTITUTE OF TECHNOLOGY
DEPARTMENT OF ELECTRICAL ENGINEERING

EE 163
Communication Theory I

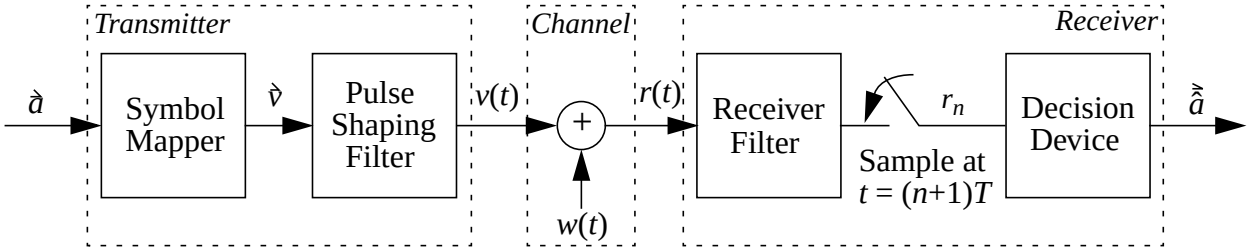
Winter 2003

http://ee163.caltech.edu

Communication System Analysis Project
Part 1: Binary Phase Shift Keying

We are going to investigate the performance of BPSK, in terms of probability of bit error, P_ε, as a function of signal-to-noise ratio (SNR).

Consider the communication system shown below, and go through the step by step instructions on simulating it:



Input Data Sequence

- We will first generate a block of $N_a = 1,000$ random bits, $\vec{a} = [a_0, a_1, a_2, \ldots, a_{N_a-1}]$, with $a_n \in \{0,1\}$.
  Hint: In C/C++ `lrand48()&1` generates one random bit.

Symbol Mapper

- Next, we will generate a block of symbols for transmission. For $n = 0,1,2,\ldots,N_a-1$, let

$$v_n = (1-2a_n)\sqrt{E_b} = \begin{cases} +\sqrt{E_b}, & \text{if } a_n = 0 \\ -\sqrt{E_b}, & \text{if } a_n = 1 \end{cases},$$

  where $\sqrt{E_b}$ is the transmitted bit energy.
  Hint: Use a look-up table to implement this function.

AWGN channel

- The combined effect of pulse shaping at the transmitter, the AWGN channel, the receiver filter, and the signal sampler can be modeled by an equivalent discrete-time channel that produces:

$$r_n = v_n + w_n,$$

  where $\vec{w} = [w_0, w_1, w_2, \ldots, w_{N_a-1}]$ are independent zero mean Gaussian random variables, with variance $N_0/2$.

  Hints:

  - If X_1 and X_2 are independent random variables uniformly distributed over $[0,1)$, then

$$Y_1 = \sqrt{-2\ln X_1}\cos(2\pi X_2)$$
$$Y_2 = \sqrt{-2\ln X_1}\sin(2\pi X_2)$$

    are independent Gaussian random variables, each with zero mean and unit variance.

- In C/C++ `drand48()` generates a random value in $[0,1)$.
- To investigate system performance at different SNRs (E_b/N_0), you may want to keep E_b fixed (at for example, $E_b = 1$), and adjust N_0 to achieve the desired signal-to-noise ratio.

Decision Device

- Next, we will generate the received data bits. For $n = 0, 1, 2, \ldots, N_a - 1$, select

$$\hat{a}_n = \begin{cases} 0, & \text{if } r_n \geq 0 \\ 1, & \text{if } r_n < 0 \end{cases}$$

Data Sink

- We now compare each received bit, $\hat{a}_n$, with the corresponding transmitted bit, a_n, and count the number of errors, N_ε.

  For a single transmitted block, the estimated probability of bit error is then given by

$$\hat{P}_\varepsilon = \frac{N_\varepsilon}{N_a}$$

Notes

- To obtain an accurate measurement of the probability of bit error when simulating communication systems, one often needs to transmit several million bits. For this assignment, it is recommended to break the number of bits you need to transmit into blocks, with a block size of $N_a = 1,000$ bits, and then transmit several thousand blocks.

  If only a single bit at a time ($N_a = 1$) is transmitted, your program will either run slowly, or require a lot of revisions in the future if you use the code for more elaborate communication systems. If all the bits in a single block (e.g., $N_a =$ several million) are transmitted, your simulation will use a lot of memory, and may run *really* slowly, if at all.

- If N_f is the number of transmitted blocks, and $N_{\varepsilon,i}$ is the number of bit errors in the i^{th} block, then the estimated probability of bit error is

$$\hat{P}_\varepsilon = \frac{1}{N_f N_a} \sum_{i=1}^{N_f} N_{\varepsilon,i}.$$

- As confirmation of the validity of your code, you should generate a plot of the probability of a bit error vs. SNR, for SNR $\in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$ dB. This plot should show three curves, corresponding to

  1. The theoretical probability of a bit error
  2. The estimated probability of a bit error for the case of 100 simulated blocks of data (transmitted), and
  3. The estimated probability of a bit error when sufficient simulated blocks of data are transmitted, so that 50 block errors are counted for each SNR point.

- You may do this programming assignment by yourself, or in groups of 2. You can use either Matlab, or C/C++.

- Each individual or group should hand in a 1-2 page report that presents and discusses the results found. Please hand in (email) your source code as well, in one compressed file (zip or tar).