

ECE 146B/UCSB: Digital Communication

Lab 2: Simulation of linear modulation in AWGN

Reading: Chapter 4 (linear modulation) and Chapter 6 (Section 6.1 on Gaussian random variables and Q function). We also state and use probability of error expressions from Chapter 6, but reading about how these are derived is not required at this point.

Lab Objectives: To implement in complex baseband a linearly modulated system for a variety of signal constellations. To estimate the performance of these schemes for an ideal channel via simulation, and compare with analytical expressions.

Laboratory Assignment

0) For this lab, you can either build on your own code from Lab 1, or get a template for the solutions to Lab 1 from the course home page. **We will use a trivial channel filter throughout this lab as well.**

1) Write a matlab function `randbit` that generates random bits taking values in $\{0,1\}$ (not ± 1) with equal probability.

2) Write the following functions mapping bits to symbols for different signal constellations. **Write the functions to allow for vector inputs and outputs.** The mapping is said to be a Gray code, or Gray labeling, if the bit map for nearest neighbors in the constellation differ by exactly one bit. In all of the following, **choose the bit map to be a Gray code.**

(a) `bpskmap`: input a 0/1 bit, output a ± 1 bit.

(b) `qpskmap`: input 2 0/1 bits, output a symbol taking one of four values in $\pm 1 \pm j$.

(c) `fourpammap`: input 2 0/1 bits, output a symbol taking one of four values in $\{\pm 1, \pm 3\}$.

(d) `sixteenqammap`: input 4 0/1 bits, output a symbol taking one of 16 values in $\{b_c + jb_s : b_c, b_s \in \{\pm 1, \pm 3\}\}$.

(e) `eightpskmap`: input 3 0/1 bits, output a symbol taking one of 8 values in $e^{j2\pi i/8}, i = 0, 1, \dots, 7$.

3) *BPSK symbol generation*: Use part 1 to generate 12000 0/1 bits. Map these to BPSK (± 1) bits using `bpskmap`. Pass these through the transmit and receive filter in lab 1 to get noiseless received samples at rate $4/T$, as before.

4) *Adding noise*: We consider discrete time additive white Gaussian noise (AWGN). At the input to the receive filter, add independent and identically distributed (iid) complex Gaussian noise, such that the real and imaginary part of each sample are iid $N(0, \sigma^2)$ (you will choose $\sigma^2 = \frac{N_0}{2}$ corresponding to a specified value of $\frac{E_b}{N_0}$, as described in part 5). Pass these (rate $4/T$) noise samples through the receive filter, and add the result to the output of part 3.

Remark: If the n th transmitted symbol is $b[n]$, the average received energy per symbol is $E_s = E[|b[n]|^2] \|g_T * g_C\|^2$. Divide that by the number of bits per symbol to get E_b . The noise variance per dimension is $\sigma^2 = \frac{N_0}{2}$. This enables you to compute E_b/N_0 for your simulation model. The signal-to-noise ratio E_b/N_0 is usually expressed in decibels (dB): $E_b/N_0(\text{dB}) = 10 \log_{10} E_b/N_0(\text{raw})$. Thus, if you fix the transmit and channel filter coefficients, then you can simulate any given value of E_b/N_0 in dB by varying the value of the noise variance σ^2 .

5) Plot the ideal bit error probability for BPSK, which is given by $Q(\sqrt{2E_b/N_0})$, on a log scale as a function of E_b/N_0 in dB over the range 0-10 dB. Find the value of E_b/N_0 that corresponds to an error probability of 10^{-2} .

6) For the value of E_b/N_0 found in part 5, choose the corresponding value of σ^2 in part 1. Find the decision statistics corresponding to the transmitted symbols at the input and output of the receive filter, as in lab 1 (parts 5 and 6). Plot the imaginary versus the real parts of the decision statistics; you should see a noisy version of the constellation.

7) Using an appropriate decision rule, make decisions on the 12000 transmitted bits based on the 12000 decision statistics, and measure the error probability obtained at the input and the output. Compare the results with the ideal error probability from part 5. You should find that the error probability based on the receiver input samples is significantly worse than that based on the receiver output, and that the latter is a little worse than the ideal performance because of the ISI in the decision statistics.

8) Now, map 12000 0/1 bits into 6000 4PAM symbols using function `fourpammap` (use as input 2 parallel vectors of 6000 bits). It is shown in Chapter 6 that a good approximation (called the nearest neighbors approximation) to the ideal *bit* error probability for Gray coded 4PAM is given by $Q\left(\sqrt{\frac{4E_b}{5N_0}}\right)$. As in part 5), plot this on a log scale as a function of E_b/N_0 in dB over the range 0-10 dB. What is the value of E_b/N_0 (dB) corresponding to a bit error probability of 10^{-2} ?

9) Choose the value of the noise variance σ^2 corresponding to the E_b/N_0 found in part 7. Now, find decision statistics for the 6000 transmitted symbols *based on the receive filter output only*.

(a) Plot the imaginary versus the real parts of the decision statistics, as before.

(b) Determine an appropriate decision rule for estimating the two parallel bit streams of 6000 bits from the 6000 complex decision statistics.

(c) Measure the *bit* error probability, and compare it with the ideal bit error probability.

10) Repeat parts 8 and 9 for QPSK, the ideal bit error probability for which, as a function of E_b/N_0 , is the same as for BPSK.

11) Repeat parts 8 and 9 for 16QAM (4 bit streams of length 3000 each), the ideal bit error probability for which, as a function of E_b/N_0 , is the same as for 4PAM.

12) Repeat parts 8 and 9 for 8PSK (3 bit streams of length 4000 each). The ideal bit error probability for Gray coded 8PSK is approximated by (using the nearest neighbors approximation) $Q\left(\sqrt{\frac{(6-3\sqrt{2})E_b}{2N_0}}\right)$.

13) Since all your answers above will be off from the ideal answers because of some ISI, run a simulation with 12000 bits sent using Gray-coded 16-QAM with no ISI. To do this, generate the decision statistics by adding noise directly to the transmitted symbols, setting the noise variance appropriately to operate at the required E_b/N_0 . Do this for two different values of E_b/N_0 , the one in part 11 and a value 3 dB higher. In each case, compare the nearest neighbors approximation to the measured bit error probability, and plot the imaginary versus real part of the decision statistics.

Lab Report: Your lab report should document the results of the preceding steps in order. Describe the reasoning you used and the difficulties you encountered.

Tips: Vectorize as many of the functions as possible, including both the bit-to-symbol maps and the decision rules. Do BPSK and 4-PAM first, where you will only use the real part of the complex decision statistics. Leverage this for QPSK and 16-QAM, by replicating what you did for the imaginary part of the decision statistics as well. To avoid confusion, keep different matlab files for simulations regarding different signal constellations, and keep the analytical computations and plots separate from the simulations.