EEE 552 – Matlab Exercise–Symbol Error Probability Comparison of two 8-QAM Constellations

This Matlab exercise requires you to calculate the symbol error probability P_e for M-QAM where M=8. The main goal is to plot P_e versus E_b/N_0 and compare with corresponding bounds.

I would like you to simulate two different 8-QAM constellations, on pp. 105 of our Proakis (5th Edition). These are in Figure 3.2-4 (left), and the (fat) rectangular $M = 8$ constellation in Figure 3.2-5. These simulations will be based on Monte Carlo simulations where you generate symbols, corrupt them with noise, make decision on the received corrupted value, and compare with the original symbol to count errors. You would do this at each E_b/N_0 until you get about 100 errors. Then you estimate the symbol error probability by taking the ratio of the number of errors to the total number of symbols "transmitted", at that E_b/N_0. After doing this over an appropriate range of E_b/N_0, you can get a P_e versus E_b/N_0 curve. Do this for both 8-QAM constellations.

More specifically,

1. Select the desired value of E_b/N_0 (without loss of generality you can assume that $N_0 = 1$)
2. Select the transmitted signal index m from the set $\{1, 2, \ldots, M\}$ with equal probability
3. Calculate $E_s = E_b \log_2 M$ and generate the two components of the noise as n_1 and n_2 are generated as independent Gaussian random variables with mean 0 and variance $N_0/2 = 1/2$. Since the m^{th} signal is transmitted, we make an error when (r_1, r_2) falls outside the m^{th} decision region. (Before the simulation starts, for each constellation, you need to identify the decision region for every point in the constellation, so that in the simulations you can identify which decision region the received 2-D vector (r_1, r_2) has fallen into).
4. After we generate (r_1, r_2), we test if it falls in the correct decision region. If so, we count no error. If not, an error has occurred.
5. Go back to step 3 until 100 errors occur. This can be accomplished by a while loop.
6. For the current value of E_b/N_0, estimate P_e by taking the ratio 100 divided by the number of iterations required to get those 100 errors.

Compare the two Monte Carlo simulations and explain which 8-QAM constellation is better. For both constellations, also plot the lower bound and the upper bound we derived in class. So each 8-QAM constellation should have a Monte Carlo simulation (this will be called "simulation" in your report), *and* it will have analytical upper and lower bounds for each constellation.

How to Write the Report:

Present these results in a series of plots. The plots and the style of the report-writeup should follow the following guidelines, or address the following issues:

- How tight are the upper and lower bounds? In answering this question, please be quantitative, and use the conventional approach of fixing an appropriate BER and comparing the difference in E_b/N_0 in dBs, when comparing the curves.
- How do the two constellations compare?

- Type your report.
- The report should be self-contained.
- All notation should be defined.
- Someone knowledgable about digital communication (like your future self in 10 years) should be able to follow your report.
- The plots should be in the format provided in the textbook (SNR should be in dBs etc).
- The P_e versus E_b/N_0 plots for the exact expressions, Monte Carlo simulations, and bounds should all be in the same range (i.e., you should not plot the exact expression up to $E_b/N_0 = 30dB$ but only plot the Monte Carlo up to $E_b/N_0 = 10dB$, for example). Typically you want to plot the symbol error probability down to 10^{-5} or 10^{-6}.
- If you choose not to use a while loop, but use a for loop instead, then make sure you run the simulation enough times especially at high SNRs to collect enough errors to give you a good estimate of the symbol error rate. If you do not do this, your curves will not be smooth, and I will take off points. Note that when E_b/N_0 is quite high, the errors are extremely rare. So to get 100 errors, you will have to "transmit" many millions of symbols. This will take some time, even with today's fast computers. This is not the kind of simulation that will run in a microsecond and stop. It might take minutes, or hours. So after all your code is ready, you might have to run the simulation for a couple of hours while you get lunch, or run the code overnight. All this is to say, don't leave this work to the last minute!
- Keep the report short by incorporating multiple plots on the same figure (which also helps the comparisons), and also by putting multiple plots on the same page. Make sure it is clear which curve is which by using legends or some other way.
- Keep the report short.