Assignment 1 - Simulation of a 16-QAM System

Digital Communications, 2013/2014

You are required to write, in MATLAB, a time domain simulation of a communication system which uses M-ary quadrature amplitude modulation (QAM) with $M = 16$ symbols. The system model is shown in Figure 1 below. The channel model uses symbol rate sampling and the only channel impairment is additive white Gaussian noise (AWGN). The constellation should be Gray coded.

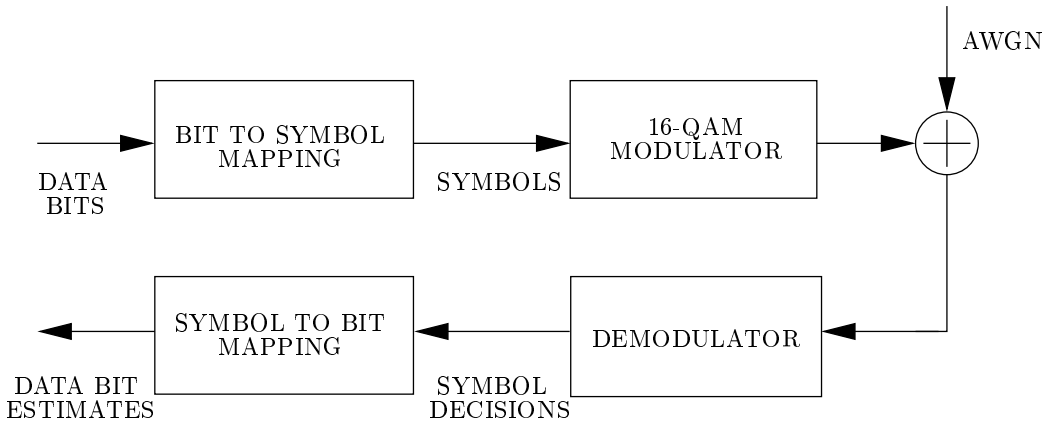

Figure 1: Block Diagram of the 16-QAM system to be simulated.

Note that **it is sufficient to simulate the equivalent vector channel** (you do not need to simulate the waveforms).

The following are the requirements:

- Use your simulation to plot the symbol error rate (SER) versus E_s/N_0 curve for the system. Plot SER on a log scale and E_s/N_0 in dB.
- On the same graph, plot the theoretical SER curve for 16-QAM (i.e., substitute $M = 16$ into the approximate formula for P_e we derived in class).
- From the curves, estimate the value of E_s/N_0 above which the system SER lies below 10^{-4} (or, if this takes too long a time, 10^{-3}).

- On a different graph, plot the bit error rate (BER) versus E_b/N_0 curve for the system. Plot BER on a log scale and E_b/N_0 in dB.
- Your program should consist of a single m-file script, and should be appropriately annotated with comments. You should not use any procedures from the MATLAB communications toolbox.
- General comments about the methods you used and the results obtained should be included in comments at the end of your program. Any other relevant points should also be clearly indicated in these comments.
- The answers to the questions asked above should also be stated clearly as part of these comments.
- The working m-file, together with the relevant graphs of results, should be uploaded via Blackboard. Please use .jpg, .eps or .pdf format for figures (do **not** submit figures in .fig format).
- The deadline is **5pm on Monday 24th March 2014**.
- **And most importantly:** The program you submit should be **your own work**. Programs will be scrutinized for evidence of copying. Programs in which copying is found will NOT be awarded a pass grade.

You may find the following MATLAB tips useful:

- The function `rand` generates a random number which is uniformly distributed between 0 and 1. Thus for example, `b = rand < 0.5` generates a random bit `b`.
- The function `randn` generates a Gaussian-distributed random number with mean 0 and variance 1. Thus, multiplying this number by σ produces a Gaussian-distributed random number with mean 0 and variance σ^2.
- The function `semilogy(x,y)` plots x against y, rather like `plot(x,y)`, except that it uses a log scale for the y-axis.