**ECE 300 Communications Systems, Project 1 and 2**

**Assigned: 11/13/2018**

**Part 1 Due: 12/52018, before class. Code review during class.**

**Due: 12/19/2019, 5pm. No Extensions. Go Enjoy Winter break**

This goal of this project is to give you hands on experience in designing and simulating a communication link in MATLAB. The goal is two-fold:

* Meet a specific BER(bit error rate) requirement
* Maximize the total number bit rate (i.e. the number of bits you can squeeze through the channel).

These two goals are in general conflicting.

You will be provided a skeleton script that simulates a basic Binary modulation through an AWGN channel. In this script there are 3 channels. The first channel has moderate ISI, and the second has severe ISI, and the third is a time-varying multipath channel with moderate ISI. You should focus your efforts on dealing with the moderate ISI channel.

At your disposal is all of the functionality available in the Communications Toolbox(type help comm for a list). However, you are restricted to using only linear modulation schemes such as M-ary PSK, PAM and QAM. No FSK or MSK, as these effectively increase the bandwidth you are occupying. Also, do not do any pulse shaping, as this will just cause you pain and offer no benefit.

You may work in groups of 2 or 3.

**Part 1:**

There are two things to do in part 1. First, modify the skeletonScript for higher orders of QAM. i.e., get at least 4 and 16 QAM working. Start with 4-ary PAM. Working means you match theoretical performance curves that you generate using BERAWGN. Whenever making a communications simulation, you should ALWAYS check your baseline BER/SER against BERAWGN to ensure you have properly scaled the noise.

After you have matched the theoretical curves, turn on the moderate ISI channel. This will make your BER significantly worse, and you will need to add an equalizer to this simulation to mitigate the ISI. Unequalized, BPSK over this channel will give you a BER of around 10e-3. You should be able to equalize the moderate ISI channel and bring the BER down to 10e-4. Without proper equalization, it is highly unlikely you will be able to meet the requirements of part 2. The Communications Toolbox provides a suite of equalization structures and algorithms. I recommend investigating the LMS, RLS algorithms using a linear or decision feedback equalizer.

It is quite likely you will have difficulty with this section, I urge you to start early, and contact me with any questions.

It is very important to properly scale your noise otherwise your results will be inaccurate. See the following link for how to do this:

https://www.mathworks.com/help/releases/R2018b/comm/ug/awgn-channel.html

**Part 2:**

Using whatever combination of modulation and coding you like, get the BER of the link over the moderate ISI channel down to 10e-6 at an SNR of 12 dB. In order to reduce the BER, you will need to use an error correcting code. You may use either a symbol level code such as Reed-Solomon, or a bit-level code, such as BCH or Convolutional encoding

Once you’ve gotten the BER down to 10e^-6, then you can experiment with increasing the bit rate. You can generally do this by increasing the modulation order, and decreasing the amount of training symbols you use for equalization.

**Summary of requirements:**

Part 1: Match AWGN performance for 4 and 16 QAM with no ISI. Achieve BER of 10^-4 at 12 dB SNR using BPSK + Adaptive Equalizer over moderate ISI channel.

Part 2: Achieve BER of 10^-6 at 12 dB SNR over moderate ISI channel using whatever means possible.

**Constraints:**

You MUST use an equalizer; you may not simply invert the channel.

You have a maximum of ~1000 symbols to transmit per packet. Squeeze as many bits into these symbols as you can.

**Calculating bit rate:**

In general, the bit rate is usable information transmitted through the channel. Training symbols don’t count as usable information, neither do parity bits. Your symbol rate is then the number of usable symbols transmitted/1000. From there you can figure out a bit rate, i.e. the # of usable BITS per symbols sent. As you begin to calculate your bit rate, I highly recommend you contact me to check that your calculations are correct.

**Calculating the correct SNR:**

This can also be tricky when you start using error control codes. Make sure you set it correctly. See this URL for more information:

https://www.mathworks.com/help/releases/R2018b/comm/ug/configure-ebno-for-awgn-channels-with-coding.html

**Report:**

For both portions, put your code in a MATLAB publisher file, it should be well-commented code that describes what you are doing and why, and what the results are. Divide the code into 2-3 sections for example, with some comments explaining each section. Comment the important lines of code throughout. Make figures, and make sure that all your plots have axes. Do not give me 100 figures; give me a few figures, with legends on them that illustrate what you did. Publish the file as a PDF.

**Grading:**

Report: 20%.

Code: 10% Be sure to also e-mail me ALL of the code necessary to run your assignment.

Technical portion 70%. The benchmark for a B is to achieve the specified BER in for each part. To get an A, you have to put in enough effort to show that you attempted to maximize the bit-rate for each scenario. You can illustrate this for example, by having multiple sections of your code that showed what worked, what didn’t, etc., and what you finally chose as your system.

Final note on complexity: Your code must run in a ‘reasonable’ amount of time on my own computer. If your code takes forever to run even a few iterations, you have likely designed a receiver that is too complex for realistic implementation.