A Crash Course in Wireless Communications

Or How I Completed a Final Project in 4 Days

Jack Spencer "Danger" Langner
Dept. of Electrical Engineering
The Cooper Union
Manhattan, New York
langner@cooper.edu

***Abstract*—This document serves as a brief overview of my attempt to finish Project 0 for ECE-408: Wireless Communications, alternatively this project was also the final project for ECE-300: Communications Systems in 2015. For this project students had to recreate the theoretical Bit Error Rate (BER) curves for different orders of Quadrature Amplitude Modulated (QAM) signals. Once this was done, a channel with Inter Symbol Interference (ISI) was introduced, so the channel needed to be equalized in order to obtain a BER of 10^{-4} with QAM2, or equivalently BPSK. Finally, the students had to achieve a BER of 10^{-6} at 12 dB over a moderate ISI channel. It was found that Jack Langner either had enough skill or luck to complete all these tasks in one week.**

***Index Terms*—finagle, MATLAB, "I hate this."**

I. Introduction

Wireless communications is complicated, plain and simple. We want to send cute cat videos to each other but the world does not want us to enjoy these simple pleasures. To this end people have began sending data wireless because we do not want cabling everywhere and require these cat videos. When we begin sending information, we have to be aware that the signal will be interfered with and influence the received signal. This affect of sending information in a non ideal environment can be modeled through the Bit Error Rate (BER), which is measured on a logarithmic scale, and is calculated based on a variety of factors.

II. Theoretical Bit Error Rate

A. *Maintaining the Integrity of the Specifications*

MATLAB is very nice, with the proper toolboxes, specifically the Communications Toolbox, it is very easy to generate BER curves. These curves can be calculated either from the theoretical conditions of different modulations schemes and Error Correcting Codes (ECC). The first part of the project was to recreate the theoretical BER curves for different orders of Quadrature Amplitude Modulated (QAM) signals through simulation. Within the simulations, random data was generated and modulated with the given order of QAM. Then noise was added to reflect the signal to noise ratio (SNR), or energy per bit per power spectral density of the noise (E_b/N_0). Finally, the signal was demodulated and the recovered bits were compared to the original data that was sent and the differences were used to calculate the BER. This procedure was done for QAM of orders 2, 4, 8, and 16. Note, that for 2QAM (or binary phase shift keying), awgn did not properly scale the noise, which caused an issue with recreating the BER curve. In order to overcome this obstacle, I used my immense intellect to generate the noise myself. Since BPSK modulates individual bits and no error correcting code was used, the scaling from SNR to noise variance is striaght forwaard. Specifically, for Gaussian noise, the relationship is $\sigma^2 = 10^{-SNR/10}$ and if the noise is complex then σ^2 is divided by two.Additionally, with the built in MATLAB function awgn(), incorrect results were obtained for orders of QAM that were not perfect squares,e.g. 16, 64, etc., other than 8. This is why these constellations are ignored. Also, scaling noise is really hard and is probably worthy of a class on its own.

Thank you to Armaan Kohli for his help in understanding Professor Keene's requirements.

III. Moderate Intersymbol Interference

Once the theoretical BER curves were recreated, the task was to undo the action of a channel on the signal. In this situation, the channel was modeled as a finite impulse response filter. Furthermore, there were to channels to use, the moderate channel, which had the form of $1 + 0.2z^{-1} + 0.4z^{-2}$ and the severe channel which was $0.227 + 0.46z^{-1} + 0.688z^{-2} + 0.46z^{-3} + 0.227z^{-4}$. Only the moderate channel was handled for this project because that is all I had time for. With this model of the channel, the symbols mix with each other and bad things happen, if its bad enough the channel with no noise can make one symbol look another. The process of equalizing the channel is essentially finding the inverse filter, or inverse function, that represents the channel. To this end, since the channel has a finite impulse response, the inverse filter will have an infinite impulse response, which is to say that it will have infinite length. Therefore, the process of equalization can only approximate the true inverse. So, for a finite equalizer, a longer length is ideal but hard to make accurate, or feedback could be used which makes the filter appear infinite in duration. For this reason, a feedback equalizer was used so that less filter weights needed to be used and the algorithm used to determine the weights has less work overall to accomplish. So using a feedback equalizer based on the Least Mean Squares (LMS) adaptive filtering approach with a step size of 0.005, I was able to achieve good channel equalization for both QAM2 and QAM4. The question then becomes how “good is good?”, well the specification given to us was a BER of 10^{-4} at 12 dB. For QAM2 this was achieved at 9.5 db and 4QAM the target BER was acheived at 10 dB. Go Jack.

IV. ISI AND ECC

The final portion of the project was simple an objective, “Achieve BER of 10^{-6} at 12 dB SNR over moderate ISI channel using whatever means possible,” no big deal. I interpretted this to mean that we should build on top of what we already have, i.e. use an additional technique to reduce the BER while still using the modulation and channel equalization. Furthermore, there is a competitive portion of the project that relates to the bit rate / throughput of the channel because we are limited to transmitting 1,000 symbols. Since the number of symbols is restricted, it is beneficial to use higher order QAM constellations. However, during testing data that was modulated with higher order QAM constellations experienced an oddity where the BER would level off to about 10^{-2} BER, I refer to this phenomenon as “shelving”. Luckily, no shelving occurred with 2QAM (binary phase shift keying) so I got something that actually, unfortunately the bit rate is not great. This claim of my bit rate not being amazing is backed up by simple math that I have barely thunk about and may have possibly dreamt about. Fun fact, “dreamt” is the only word in the English language that ends with “mt”. So, with 2QAM every symbol represents exactly one bit, i.e. there is a one-to-one correspondence of a single bit to a symbol. Sinc the equalizer comes from an adaptive filter algorithm, there is a training set for every packet of 1,000 symbols. The symbols that make up the training sequence cannot be counted towards the BER and therefore the bit rate calculation. In this situation, a training sequence of 200 symbols was used, so this reduces the usable number of symbols, and therefore the usable number of bits to 800. For an Error Correcting Code (ECC), I opted to use a rate 2/3 convolutional code. This was done because for every 2 information bits, there is one parity bit so the SNR is not affected too much while providing good coding gain with moderate overhead. The overhead is in comparison to a rate 1/3 turbo code, which would essentially be composed of two interleaved rate 1/2 convolutional codes and requires a much more sophisticated decoding algorithm. All I can say is that for a BER of 10^{-4} an SNR of approximately 8 dB.

V. PICTURES

A. Recreate Theoretical BER Curves

The first four figures show that my simulation of the different QAM constellations agrees with the theoretical BER generated by the MATLAB function berawgn(). In the title, the constellation order is given along with how long it took the simulation to run. Blue curves represent the simulation and orange the theoretical curves.

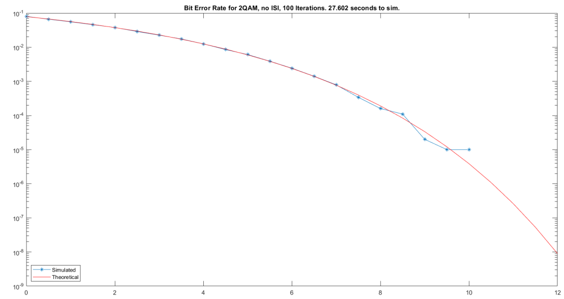

Fig. 1. QAM2

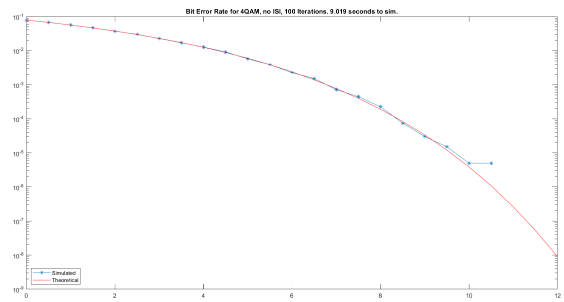

Fig. 2. QAM4

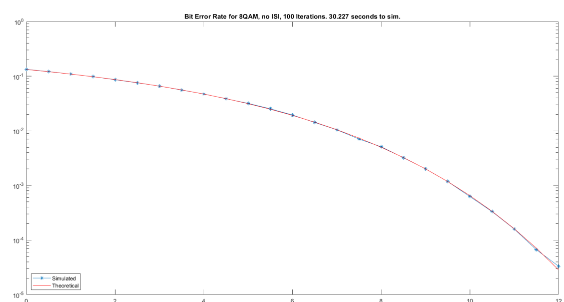

Fig. 3. QAM8

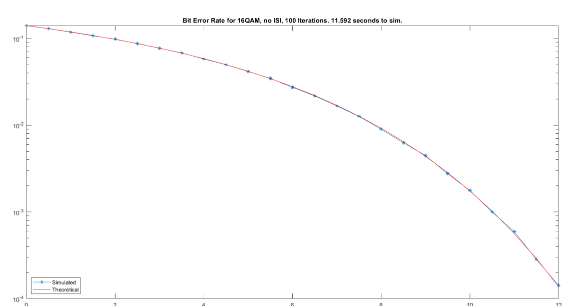

Fig. 4. QAM16

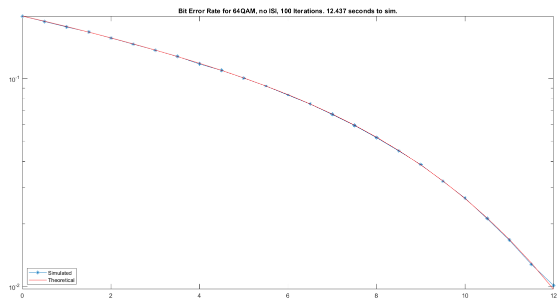

Fig. 5. QAM64

Note, that as the order of the QAM constellation increases, the BER gets worse, so to make the higher order constellations usable, ECC's seem to be necessary.

B. Moderate ISI

The next two figures show how my equalizer performed with the QAM2 and QAM4 constellations.

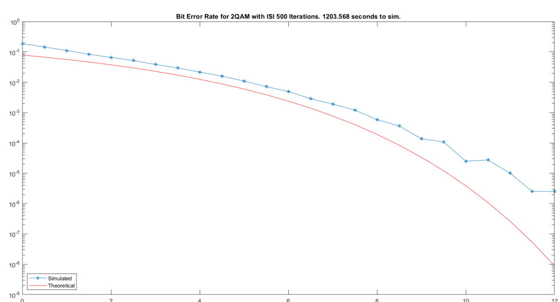

Fig. 6. QAM2 in moderate ISI channel

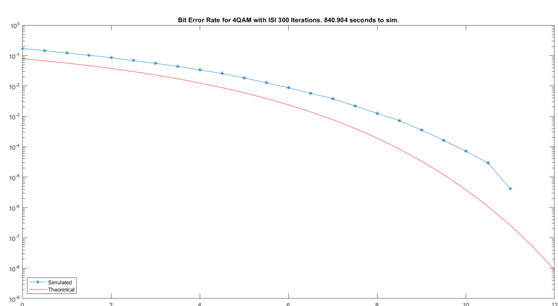

Fig. 7. QAM4 in moderate ISI channel

The equalizer did well, but we begin to see that increasing the order degrades performance, a common occurrence in this line of work. Note, the the blue curves are the simulated, equalized BER curves and the orange curves are the theoretical curves with no channel effect.

C. Full Channel Simulations

The next few figures are the simulation of part 2, where both an equalizer and ECC were used. In figures 9 and 10, a green trace is added. This BER curve is the output from the MATLAB bertool which can generate the theoretical BER curve of a modulation scheme with a given ECC.

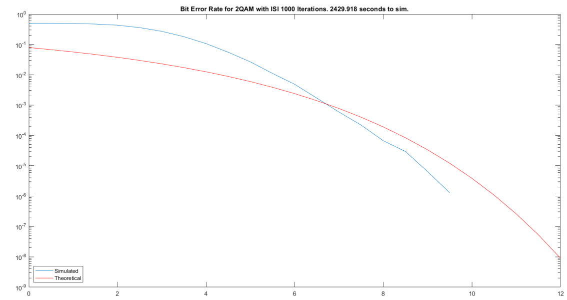

Fig. 8. QAM2 under full simulation

This figure represents the best graph that I obtained for the simulation.

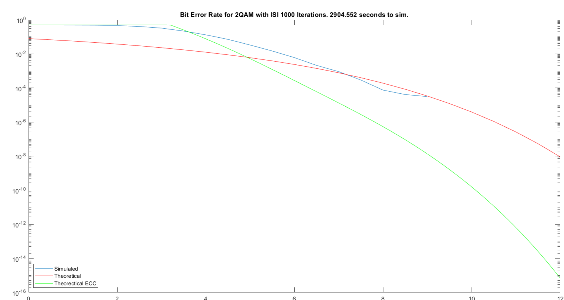

Fig. 9. QAM2 compared to theoretical ECC

While the performance of the simulated BER curve for QAM2 is degraded here, I included the output from the MATLAB bertool which can provide the theoretical BER curve of a modulation scheme with an ECC included. In this case, we see how the simulation compares to the theoretical ECC which did not have to deal with equalization.

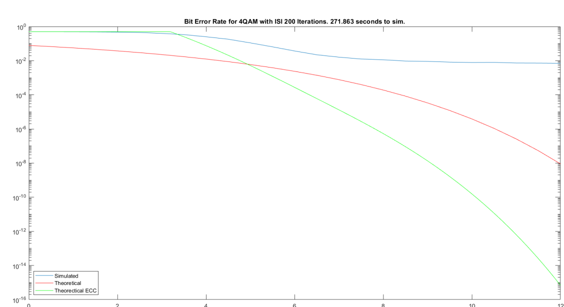

Fig. 10. QAM4 with shelving

Finally, I am including an example of the shelving that occurred for higher order modulation schemes. My guess as to why this happens is that the equalizer begins to introduce errors for the high SNR cases which procedes to mess up the ECC which is why it begins to correct and then levels out quickly as if it needs to run home and turn off the oven. I know I repeated myself a few times, but this is because I steadily realized there were better ways of writing this paper, but I did not feel the need to go back and fix it only address that I am aware of the redundancy.

ACKNOWLEDGMENT

Tim Hoerning for teaching the class on Wednesday. Armaan Kohli for putting up with me. Karol Wadolowski for being a friend, traveled down the road and back again, your heart is true, you're a pal and a confidant.

REFERENCES

Literally every MathWorks page relating to the built in functions I used.