Modeling Wireless Communications

Julian Rovee[1], Max Wang[2], Dr. Minnie Ho[3]

Henry M. Gunn High School[1,2], Intel[3]

INTRODUCTION

Overview

Wireless communication is one of the systems that allows our world to function the way it currently does. Every facet of our world is controlled by computers and the signals that they send and receive. Yet, very few people understand how these systems work.

In a basic communications system, binary data is translated into an analog signal and sent as an electromagnetic wave, and then that wave is received and translated back into binary data. The transmitter sends different signals to represent different pieces of data: for instance, one might send a sine wave to represent a 1 and a cosine wave to represent a 0. The receiver then captures these waves using an antenna, and decodes the sinusoids into 1s and 0s.

This concept of a wireless communications system seems very simple, however, in practice, there is much more to take into account. The main obstacle that wireless communications systems must overcome is noise. While the wave is traveling through the air, it is subject to being distorted by all kinds of noise. It can be affected by Gaussian noise, reflected, affected by the Doppler Effect, and all sorts of other things. By the time that the wave gets to the receiver, it could look completely different than what it originally looked like.

Goal

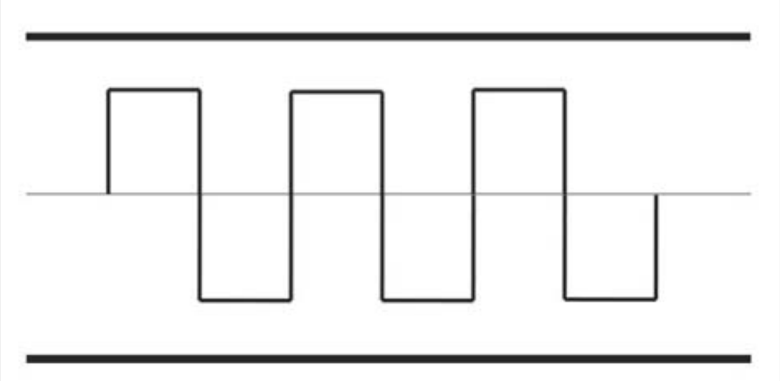
Our goal is to develop a mathematical model of a communications system in MATLAB which accurately models AWGN and Gaussian noise and recovers the data with BER matching the theoretical minimum BER at all signal to noise ratios.

BACKGROUND INFORMATION

BPSK

What is it?

BPSK stands for Binary Phase Shift-Keying, and is a basic model of a communications system. In a BPSK system, binary data, ones and zeroes, is converted into a square wave, which looks something like this.



The system then multiplies this square wave by a carrier wave. A carrier wave is a wave which is at a much higher frequency than the data so that the receiver can know what frequency to look for. This is the most simple transmitter, and is what we used for most of our modeling. Although it does have a lower data rate than other systems, it is easier to work with, which is what we need to be able to clearly see the effects of the algorithms on the signals.

BPSK Transmitter

Below is our code for the BPSK transmitter.

```
5       %% Transmitter
6       % Signal
7 -     N = 10^4;                           % number of bits or symbols
8 -     ip = rand(1,N)>0.5;                 % generating 0,1 with equal probability
9 -     s = 2*ip-1;                         % BPSK modulation 0 -> -1; 1 -> 1
10
11      % Carrier
12 -    nS = 100;                           % Sampling frequency
13 -    fc = 1e1;                           % Carrier Frequency
14 -    t = 0:(1/(nS)):(N-(1/nS));          % Time domain
15 -    car = cos(2*pi*fc*t);
16
17      % Modulation
18 -    m_y = rectpulse(s,nS)/(sqrt(nS));   % Turns impulse into square wave
19 -    m_y = car.*m_y;                     % Modulates signal
20
```

In lines 7-8, we establish a binary data stream of length "N." Then, in line 9, we BPSK modulate the data: any ones will remain ones (because 2(1) - 1 = 1), but zeroes become -1s (because 2(0) - 1 = -1). Lines 12-15 establish the carrier wave "car," and lines 18 and 19 turn the impulse data into a square wave and then modulate it by the carrier frequency.

BACKGROUND INFORMATION CONTINUED

AWGN

AWGN Channel

After transmitting the signal, it is passed through a channel to simulate real-world impairments. The first channel that we built was a simple AWGN (Additive White Gaussian Noise) channel. AWGN is complex noise which is normally distributed and equal in all directions.

The code for our simple channel is shown below:

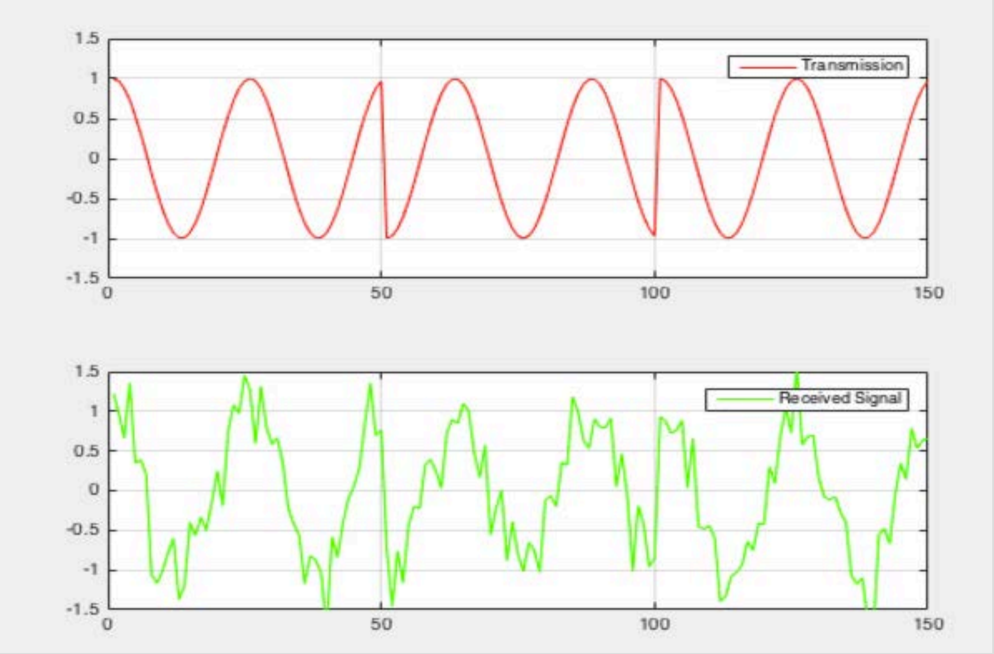
```
n = 1/sqrt(2)*[randn(1,N*nS) + sqrt(-1)*randn(1,N*nS)];

% Channel and noise addition
y1 = h.*m_y;
y2 = y1 + 10^(-Eb_N0_dB(ii)/20)*n;
```

The variable "n" is a list of complex numbers normally distributed around the origin. It is derived by adding a random real number to a random imaginary number in order to get a complex number, and then dividing by2 in order to return the noise to 0 db variance. The list is of length "N*nS" so that a noise value may be added at every sampling time of the signal.

In the AWGN code, the variable "h" is equal to 1, so y1 does not affect anything. However, y1 is important in the rayleigh code, as we will see later.

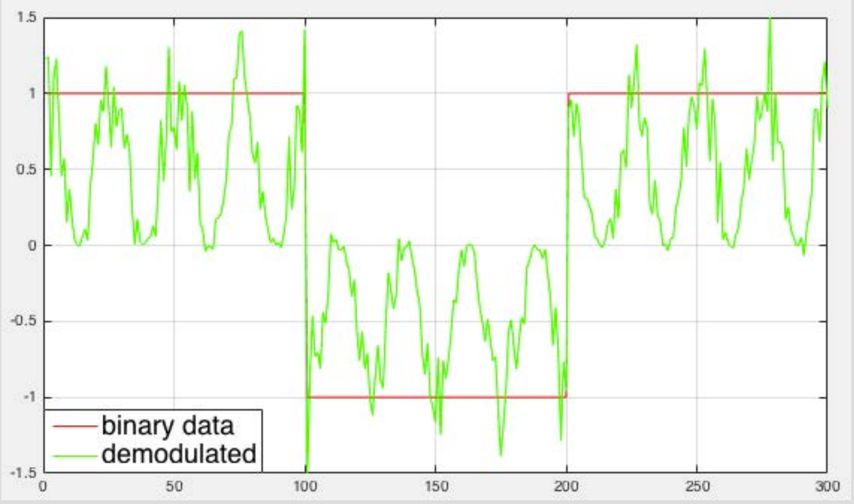
Finally, the variable "y2" adds the BPSK modulated signal to the Gaussian noise to generate a noisy signal that looks similar to the one shown below.



AWGN Receiver

Receivers are designed to take the noisy signal created by the channel and turn it back into the original binary stream with as much accuracy as possible. The ideal receiver will create a binary stream with a bit error rate matching the theoretical BER (look at the AWGN section under "background" for an explanation of theoretical BER). Our receiver is made up of two parts: a demodulator, which neutralizes the carrier in the received signal, and a decoder, which decides whether something is in positive phase or negative phase.

We demodulate the signal by multiplying the received signal by the carrier frequency. This allows the troughs of the carrier to cancel the troughs of the received signal (which should be roughly the same), and, in contrast with dividing by the carrier, it avoids division by zero. After the first step in demodulation, the signal looks like this, with the entirety of a symbol either above or below 0, depending on the original signal:



The second step in demodulation is taking the above signal and turning it into a string of length "N". Our decoder then checks whether the final sum was above zero. If the final sum is positive, it appends a "1" onto the output, if the final sum is negative, it appends a "0". For example, in the above picture our decoder would add samples 1-100 together, decide the sum was positive then add a one, repeat for samples 101-200 and add a zero, and repeat for samples 201-300 and add a one. It would return the list [1,0,1] which you can see clearly is the same as the original data. In fact, the only time our decoder makes an error is when the noise outweighs the signal in a majority of the samples in a symbol, and because this is measured in theoretical BER, our simulated BER curve should match our high enough to allow for it.

BACKGROUND INFORMATION CONTINUED

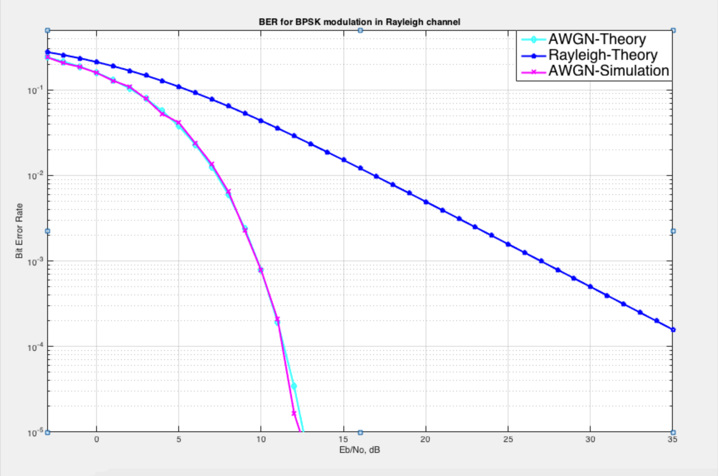
Rayleigh and BER

Rayleigh Channel and Receiver

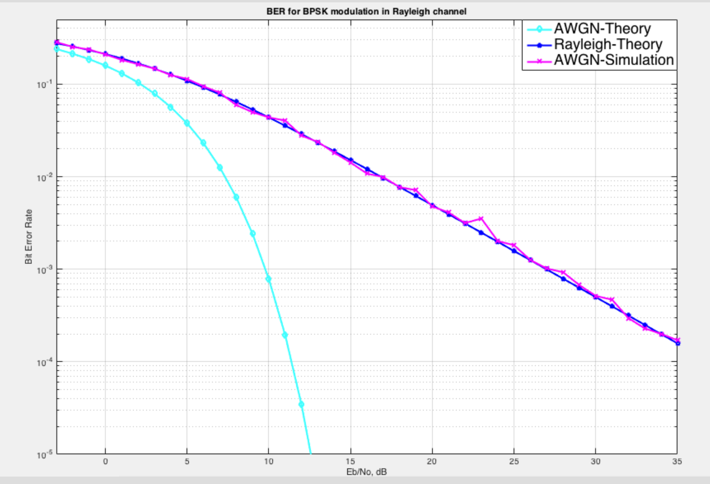
The rayleigh channel is the same as the AWGN channel except a random complex value is assigned to the variable "h". In rayleigh, one multiplies the received signal by a list of random complex numbers "h" before adding AWGN. Thus, the signal can drop out if the element of "h" is too close to zero. This makes rayleigh harder to correct, and it causes the theoretical BER of a rayleigh channel to be higher than that of a simple AWGN channel.

Our receiver was also very similar to the AWGN receiver, except that after demodulating but before decoding, we divided "h" out from the received signal. While it seems like it would be impossible, one can actually estimate the rayleigh channel because unlike AWGN it only changes when location changes. Dividing by "h" allows us to bring the signal back to around a constant amplitude in order decode it, but it also amplifies the awgn in the received signal. This is why simulation of rayleigh signal has a higher BER than the AWGN simulation.

BER Results



AWGN Simulation Results: Success



Rayleigh Simulation Results: Success

NEXT STEPS

In the future we could build on these findings and this system in order to explore many different areas.

First, in our system, we have been using BPSK in order to transmit our data. The problem with BPSK is that the number of bits per symbol (section of the wave) very low: for each symbol, one is only sending one bit. In future, research, we could expand on our work in order to implement new systems of sending data, such as QPSK (Quadrature Phase Shift Keying) or different levels of QAM (Quadrature Amplitude Modulation) like 16QAM, 32QAM, and 64QAM which all send data with a higher number of bits per symbol.

In addition, we could explore new impediments. In our current research, we have only explored two impediments: AWGN and Rayleigh fading. However, while these are the two most common impediments, there are many more which we have not explored. For instance, when one is using a phone in the car, the motion of the car subjects the signal to doppler fading. Or when a signal must pass through a wall, it is subject to dropoff, which is when the signal is uniformly lost. There are even other types of multipath fading besides Rayleigh fading, such as Rician fading.

Finally, while we have only been modelling the BER, in the future we could seek to improve BER by increasing diversity (having multiple transmission and reception antennae) or including redundancy in the code. These allow the signal to reach 10-6 errors per bit without having to blast the signal at very high intensities. One could explore and model the best methods or combination of methods for decreasing the BER.

ACKNOWLEDGEMENTS

Thank you Dr. Minnie Ho for your support.