Experiment 01: Basic Notions

Bruno Korst - bkf@comm.utoronto.ca

Abstract
This experiment will provide you with the basic understanding of four commonly found topics in the study of wireless communication systems. You will explore each individually via simulations, in order to gain an intuition for what they represent and what their role is in wireless communications This experiment is meant to be a limited introduction to these topics.

Keywords
Noise — Interference — Channel — Equalization

Contents

Introduction

As you begin your studies in wireless systems, it is expected that you have already seen some of the general theory of communications. You know at least a few analog modulation schemes, you are familiar with time domain and frequency domanin (and the transforms that bring you from one to the other), you know what transmitters and receivers are and you know of some impairments that may occur in the process of sending and receiving a message, particularly in the analog domain.

In wireless systems, you will concentrate on communication systems using digital modulation, which accomplishes the transmission and reception of bits wirelessly. As you speak into the microphone of your mobile phone, the milivolts (analog) generated at the microphone are amplified, filtered, coded into bits, combined, modulated and shaped in the appropriate fashion and transmitted by radio to a cellular base station. From there, they might go through fibre, copper twisted pair (i.e. ”landline”), air again until they reach a destination. Those different media form the channel through which your modulated data will travel. At the destination, the process is reversed so that your voice can be made analog again to play through the speaker.

Since we referred to your phone as ”mobile”, we should bring in a variety of other factors that will add a few more twists in the path. Just to complete the picture, say that you are talking to your friend from the GO train at 120Km/h and your friend is driving through the alps in France, much to your chagrin, at 100 Km/h.

In the simulations below, you will explore some of these twists using `Simulink`. They may be familiar to you in general, but here you will see them as applied to wireless communications. They are:

- Noise, in particular Additive White Gaussian Noise (AWGN) and phase noise;

- Interference, caused by other transmitters and by multiple reflections; and
- Equalization, which is the attempt to compensate for dynamic changes on the channel.

These are only a few of the factors that can significantly impact the transmission (and correct reception) of a message.

There will be no **preparation** for this experiment, and completion is achieved by demonstrating the systems running and answering questions asked during the experiment.

1. Simulating Noise

Below you will experiment with two models to simulate two types of noise commonly affecting communication systems. From this point on, you will use all simulation models found in the folder `Documents/MATLAB/ECE464`.

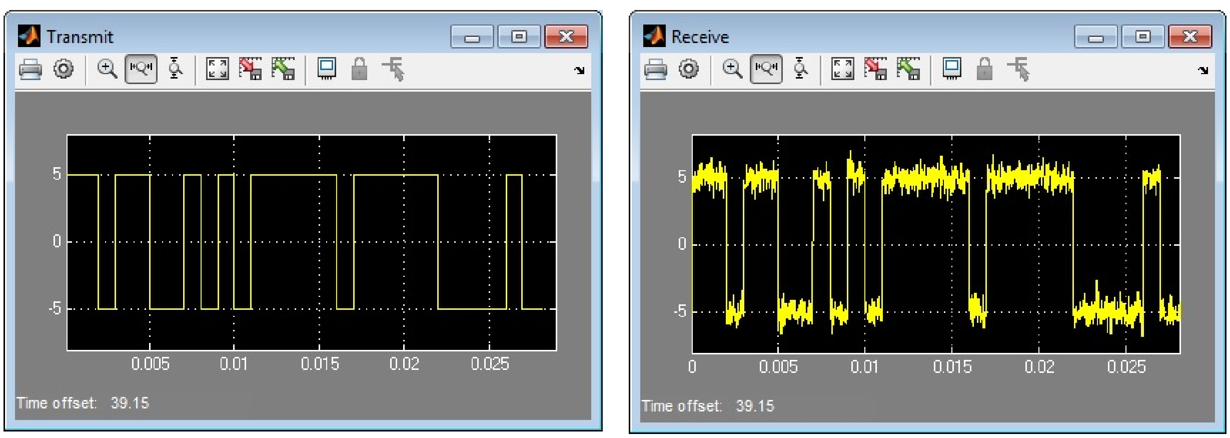
1.1 AWGN

Additive White Gaussian Noise (AWGN) is a noise model that best approaches naturally occurring noise. This is the noise "added" to a signal as it travels through a medium. The simulation presented here is a modified version of a model from [2].

Open the model `awgnoise.mdl` and briefly explore the blocks. The model is a bit generator which is conditioned prior to being put through a channel model with AWGN. There are two types of displays: time domain scopes and frequency domain scopes. The signal sent to the frequency domain scopes is windowed and averaged, in order to present a smooth display.

Run the model. If the scopes do not open automatically, double-click on them. Figure 1(a) and Figure 1(b) show the signal as it is transmitted and immediately after the channel model, with the noise added.

Try different values of signal to noise ratio (SNR) on the channel model, as well as different spectral average numbers on the spectrum scope subsystem. Your objective is to observe that for a very low signal to noise ratio (feel free to go negative), the signal becomes so buried in noise that it will not be detected.



(a) Transmitted Signal

(b) Received (after Channel) Signal

Figure 1. Simulated Additive White Gaussian Noise

1.2 Phase Noise

This will become particularly of interest when you experiment with digital modulation such as QPSK (the *P* standing for *phase*) and QAM. Phase noise relates to small variations in time (meaning: delay) a the signal arrives at the receiver. The model presented here is a variation of an example from [1].

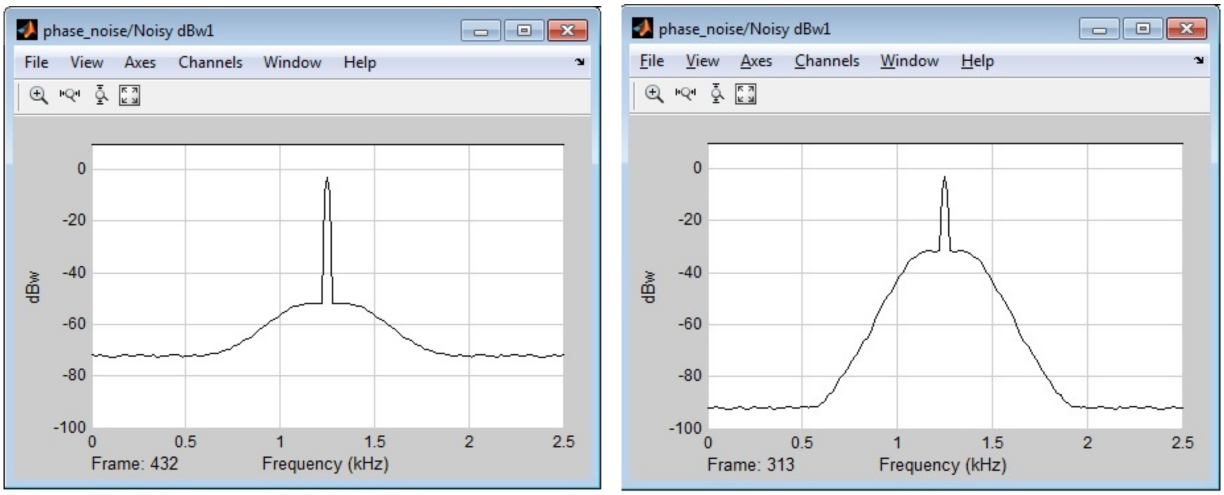
Open the model `phase_noise.slx` and explore its components. The model presents a sinusoid sampled at regular intervals, which generates complex values. There are four types of displays: a *scatter plot*, an *eye diagram*, two frequency domain displays (one averaged and one not averaged) and one value display that calculates the RMS phase noise in degrees.

The scatter plot superimposes the complex samples of the sinusoid. The frequency of the sinusoid is preset to generate four samples positioned symmetrically on the X-Y axes. This simulates a I/Q complex signal, typical of digital communications. Since we have phase noise, the samples will not always happen at the same location, which creates a "smudge" when you run the model. Run it now.

Intuitively, it is perhaps easier to think of phase noise in the time domain. The *eye diagram* display presents the noisy sinusoid displayed superimposed within a fixed window of time. Note that the position of the signal varies in time making the traces thicker.

Figures 2(a) and 2(b) show the effect of phase noise on the spectrum of the signal. Since the signal is a sinusoid, it should ideally be a delta on the frequency domain display. With phase noise you observe the shape presented below.

Your objective here is to vary the noise levels, the magnitude gain and the frequency of the signal



(a) Less Phase Noise

(b) More Phase Noise

Figure 2. Simulated Phase Noise

2. Simulating Interference

In this experiment, interference will be explored in two forms: that of a spurious signal source encroaching on a desired signal, and the interference caused by repeated and delayed versions of a signal being added to itself. The latter is particularly problematic in digital communications, as it is a source of inter-symbol interference, leading to errors.

2.1 Due to Multiple Sources

Open the `interference.slx` model. The model represents four signal sources (users), three of which are separated by a guard band. This separation by a guard band is typically regulated (that is, by Industry Canada), and one very clear example of this can be seen by inspecting the FM spectrum (88MHz to 108MHz). In the model, the fourth signal source is a "rogue" source which will represent a spurious signal interfering on the others.

Start your simulation with the slider gain at zero, by double-clicking on the slider gain and setting it to zero. Run it. The slider controls the level of the interfering source. As you increase the gain, it is hoped that you see clearly that at some point the three adjacent sources (two plus the interferer) become one. This is very likely to affect the reception of the two desired sources. Of course, if there is someone desiring to receive the signal from the interferer, the reception of that too will be affected.

Your task is to vary he gain of the interference and observe the effects, and then to position the interfering signal onto a clear portion of the spectrum. If you were to start your FM radio station, this is what you'd do. That is, this is what Industry Canada would do for you after you pay for the spectrum.

Figure 5 below shows the spectrum with the three users free of any interference.

2.2 Due to Reflections (ISI)

When a mobile phone user makes a call, or a police car receives details of an emergency call on their onboard computer, data is arriving to them along with replicas of itself. A direct signal is received from the transmitting antenna, but along with it reflections arrive shortly thereafter as a result of the signal bouncing from nearby buildings, buses and many other obstacles. This is called multipath propagation, and it is a source of a type of interference called inter-symbol interference (ISI). For now, you can think of each symbol as a defined set of the bits that represent your voice, for instance. This interference between (inter) symbols, if not minimized, leads to an increase in error rate as the receiver can no longer recover the original bits from the symbols.

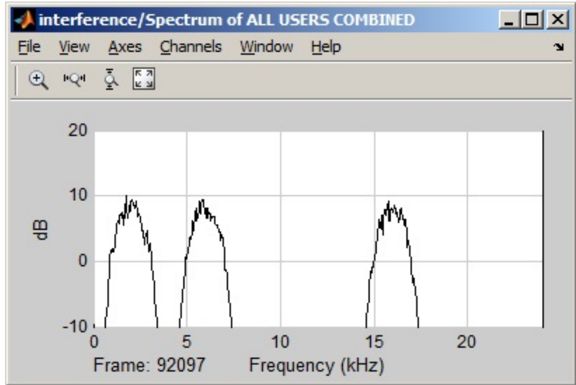


Figure 3. Three users and no interference

In the model `multipaths.slx`, you will find a QPSK modulator, which will generate the signal to be sent through a channel. The channel, in this case, is made of a direct path with some attenuation, added to three attenuated and delayed replicas of it. The delay is represented by a digital delay block, and the attenuation is controlled via a slider. The display is an *eye diagram*, which you have already used above. The model presented here is a variation of a model provided by [1].

Your objective is to change the value of the gain for each extra reflection and observe the effect of the extra paths. Note that the effect is visually similar to the one you observed when you ran the phase noise model. If you overlap many traces, you will see a thickening of the resulting line. ISI is typically identified through a wider crossing of traces resulting in a (horizontal) narrowing of the "eye" on the graph.

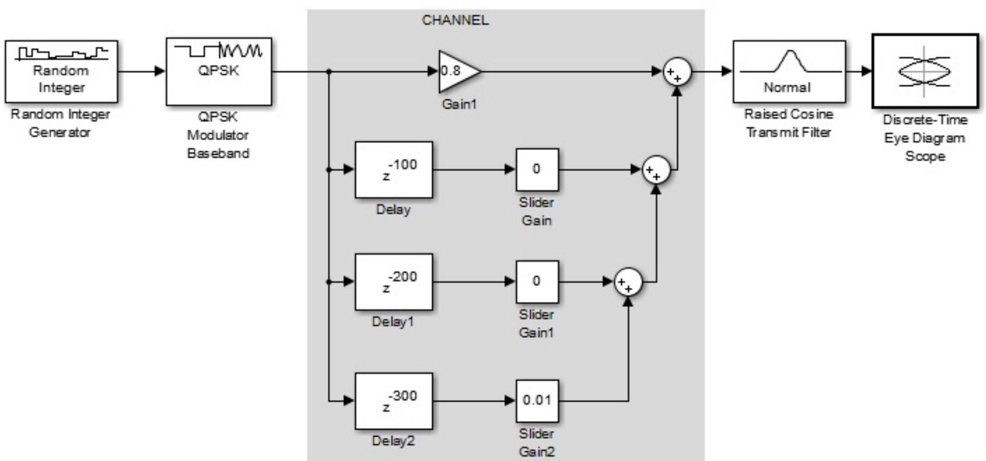


Figure 4. Multipath Model

3. Simulating an Equalizer

Now that you have simulated interference as well as the channel in which your transmitter and receiver are, you know that the transfer function between TX and RX varies dynamically, at times severely attenuating portions of the received signal spectrum. Ideally, the transfer fuction between TX and RX should be flat, with no change to the magnitude, and all frequency groups delayed by the same amount (i.e., it should have linear phase). This flat response, linear phase sounds great, but it is not what actually happens. One must, therefore, aim for a flat channel response by means of correcting it through an equalizer.

Equalization is, therefore, a correction applied to the channel transfer function in order to approximate it to a flat response.

Channel equalization in wireless systems is done dynamically, many times during transmission and reception in order to have an accurate correction of the channel response as transmitter or receiver are in movement. It is implemented in a variety of ways, but you will simulate one which uses the Least Mean Square (LMS) method. For an actual system, the process is done based on the need (i.e., how many times) and the resources you have (i.e., how fast it can be done).

You will explore the model `lmseq.slx` by changing some parameters that will help you to see how long it might take to compensate for changes in the channel. Open the model, and double-click on the equalizer block. The default parameters should be 6 taps, 1 sample per symbol, reference tap 2 and 0.01 for step size. When you run the model (run it now) two constellation displays will appear: one representing the signal before the equalizer and one after. The model presented here is a variation of a model provided by [1].

The signal used is a 16-QAM signal, which you might not have seen. For this experiment, it suffices that you understand that the dots (or "stars") of the constellation represent four bits each, and that it is imperative that the "clusters" of points be separated, or you will have errors in the decoding of the signal. The objective is, then, to keep the clusters separate and the dots within them as tight as possible to reduce the likelihood of error.

The *channel* is actually represented in the model by two blocks: the digital filter, which will impose a particular frequency response (transfer function) and the AWGN Channel, which will add Gaussian noise to the signal. The equalizer should handle (meaning: compensate for) the effects of both on the signal.

Note that the *before* constellation is no constellation at all, due to the adverse effects imposed by the channel.

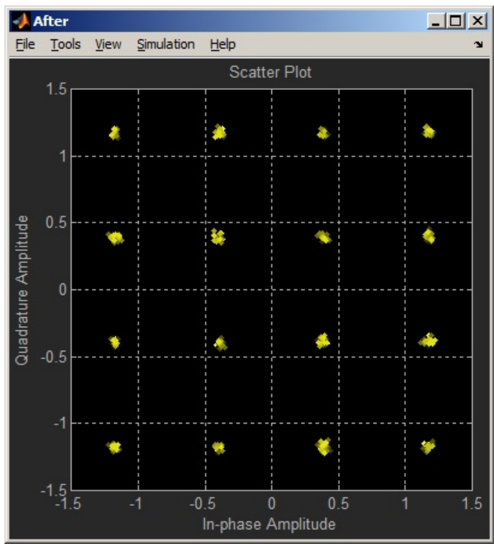


Figure 5. A very tight 16-QAM constellation, after 3 seconds of adaptation

Your task is to explore the limits of the equalizer, making it converge faster (i.e., making the constellation appear clearly within a few iterations), towards a very tight constellation, but with the smallest possible number of taps (that is, the length of the equalizer). It is all a trade-off. In addition, don't forget to vary the amount of noise from the AWGN channel. In other words, play with the number of taps, the step size and the signal to noise ratio to find a good compromise. Perhaps you should consider also the question: how fast is fast enough?

4. Accomplishments

In this experiment, you have hopefully gained an intuition for four of the most common (and problematic) topics in wireless communications. You have simulated and explored noise, interference and equalization to see the role played by them in real communication systems. In future experiments you will see more details, as the theory is covered.

Acknowledgments

Thanks for all the students who have provided input on the previous versions of this experiment.

References

[1] MATLAB and Simulink. *www.mathworks.com*. 2021.

[2] D. Silage. *Digital Communication Systems Using MATLAB and Simulink*. Bookstand Publishing, 2009.