Alexandria University
Faculty of Engineering
Electrical and Electronics Engineering
Department

Project details

Part 1: Inter-Symbol Interference due to band-limited channels

In this part, you will investigate another common channel in digital communication systems, which is the band-limited channel. As the name suggests, the channel only allows a limited range of frequency components to pass, while blocking frequency components outside this range. We investigate the simplest of such channels: a channel that has a flat response in the allowable range. Figure 1 shows the system that we will consider.

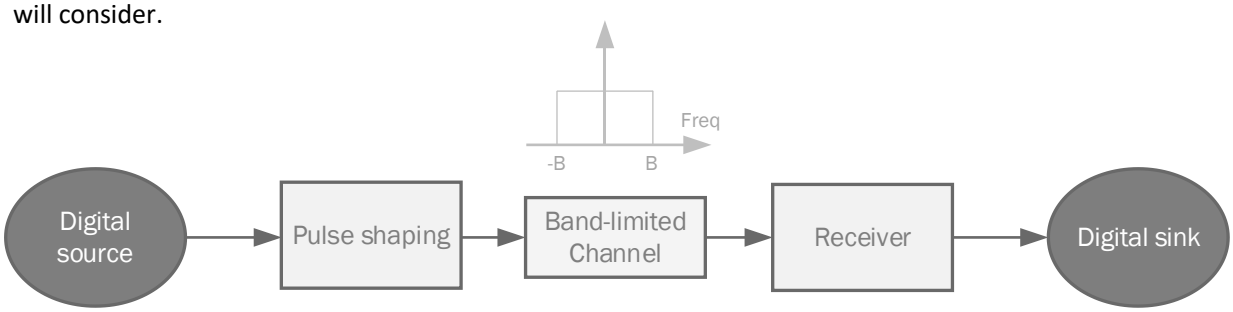

Figure 1 Communication system with a band-limited channel with bandwidth B.

The channel obviously limits the kind of signals that can pass unchanged through the channel, because if a signal has frequency components that are outside the allowable range of the channel, these components will not pass and therefore the output signal from the channel will be changed from the input signal. This issue will face the most common of signals that we use to represent bits: the square signal!
As you can see, the square signal is no longer a square signal coming out of the channel. In fact, the shape of the square signal out of the channel has "leaked" outside of the duration $T = 2/B$ that was intended at the beginning. If there are multiple square signals after each other (one square signal for each bit), these leaked parts will interfere with the signals of other bits. This phenomenon is called *Inter-Symbol Interference (ISI)*.

Guidelines: The second thing you need to show is the effect of two consecutive square signals as they pass through the channel. Consider the same channel and the same square pulse duration as before. The plots you need to show here are in time domain only. Namely,

1. ***Show two plots of the first square pulse: one before it passes through the channel, and one after.***
2. ***On top of the two previous plots, show similar plots for the second square pulse. Plot this pulse in the two plots using a different color, so that the shapes of the two pulses are distinguishable on the plots.***
3. ***Note that you will have to pass the two squares separately, i.e., you cannot create the two pulses together in the same vector and pass that into the channel. If you do it this way, you won't be able to clearly distinguish the two pulses.***

The procedure that you need to follow to generate the plots required above are shown in Figure 2.

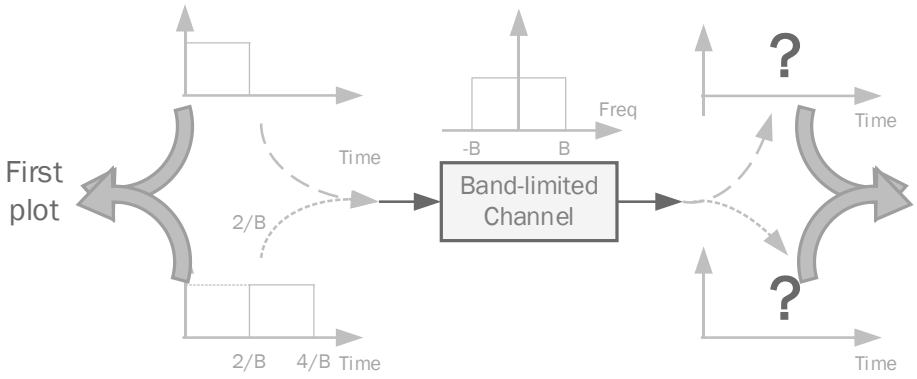

Figure 2 How to generate the two plots described above.

ISI can negatively affect the detection performance of multiple consecutive signals. To combat the effect of ISI in band-limited channels, one cannot use square pulses anymore. Instead, there are other pulse shapes that are better suited for such channels. You need to investigate such solutions.

Guidelines: the following aspects can be useful in investigate these solutions

1. ***Explain what is the mathematical criterion that ensures no ISI.***
2. ***Describe one or more pulse shapes that ensure that such condition is met.***
    a. ***For one or more of these pulse shapes, show (at least for one of those pulse shapes) plots of the pulse shape before and after the channel, both in time and frequency.***
3. ***(Optional) show the BER performance of using these pulse shapes in AWGN channels.***

Part 2: Inter-Symbol Interference due to multi-path channels

In this part, we consider another form of ISI that happens in channels that we face in wireless communication systems. The channel considered here is referred to as the *multipath channel*. To understand the effect of this channel, we refer to Figure 3.

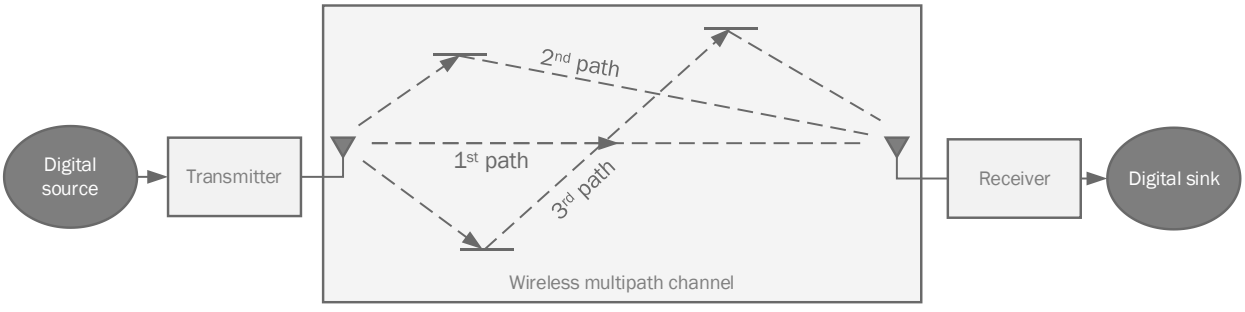

Figure 3 Signal propagation in a wireless multipath channel.

Alexandria University
Faculty of Engineering
Electrical and Electronics Engineering
Department

In wireless channels, signals are transmitted via electromagnetic waves which propagate through the air until it reaches the receiver. However, the nature of electromagnetic waves allow that multiple copies of the signal would travel around and reach the receiver at different times, such as is shown in Figure 3. This effect is what is known as the multipath effect. As is shown in the figure, a symbol transmitted by the transmitter would traverse multiple paths until it reaches the receiver. Therefore, the receiver is expected to receiver multiple copies of the same transmitted signal. Each of these copies would arrive at a different time (based on how long the path that the signal travelled through is) and with a different magnitude (based on how much attenuation that the signal suffered from during the transmission across the path).

This behavior of receiving multiple copies of the same signal is mathematically captured as follows. Let the first symbol transmitted by the transmitter be labeled as $x[0]$. Then, the receiver would receive a first copy of that symbol as

$$y[0] = h_0 x[0] + n[0]$$

where $y[0]$ is the received signal, h_0 is the channel effect of the first path on the transmitted signal $x[0]$, and $n[0]$ is the noise component.

Now let $x[1]$ be the second symbol transmitted by the transmitter. Ideally, the receiver would receive a signal $y[1]$ which corresponds to this signal (plus noise). However, the effect of multipath is that $y[1]$ will also include a copy of $x[0]$ which have passed through a longer path and therefore have arrived at a delayed time. Therefore, the second received signal can be written as

$$y[1] = h_0 x[1] + h_1 x[0] + n[1]$$

where h_1 is the channel effect of the second path on the transmitted signal. Note that we assume here that any symbol which travels through a particular path would have the same channel effect.

Assume we have a total of L paths in our channel. Then, the Lth received signal can be expressed as

$$y[L-1] = h_0 x[L-1] + h_1 x[L-2] + h_2 x[L-3] + \cdots + h_{L-2} x[1] + h_{L-1}\, x[0] + n[L-1]$$

where h_i corresponds to the channel effect of the $(i+1)$th path, and $x[i]$ is the $(i+1)$th transmitted symbol.

Let's write this set of equations on top of each other

$$\begin{aligned}
&y[0] = h_0 x[0] + n[0] \\
&y[1] = h_0 x[1] + h_1 x[0] + n[1] \\
&y[2] = h_0 x[2] + h_1 x[1] + h_2 x[0] + n[2] \\
&\vdots \\
&y[L-1] = h_0 x[L-1] + h_1 x[L-2] + h_2 x[L-3] + \cdots + h_{L-2} x[1] + h_{L-1} x[0] + n[L-1]
\end{aligned}$$

This can be written in a matrix form

$$\underbrace{\begin{bmatrix} y[0] \\ y[1] \\ y[2] \\ \vdots \\ y[L-1] \end{bmatrix}}_{Y} = \underbrace{\begin{bmatrix} h_0 & & & & & & \\ h_1 & h_0 & & & & & \\ h_2 & h_1 & h_0 & & & & \\ \vdots & & & \ddots & h_1 & h_0 & \\ h_{L-1} & h_{L-2} & h_{L-3} & \cdots & h_2 & h_1 & h_0 \end{bmatrix}}_{H} \underbrace{\begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ \vdots \\ x[L-3] \\ x[L-2] \\ x[L-1] \end{bmatrix}}_{X} + \underbrace{\begin{bmatrix} n[0] \\ n[1] \\ n[2] \\ \vdots \\ n[L-1] \end{bmatrix}}_{N}$$

So this set of equations results in the simple matrix equation $Y = HX + N$. From the receiver perspective, the terms of the equation are as follows:

1. The transmitted signal is X. This is what the receiver wants to know or estimate.
2. The received signal is Y. This is what the receiver measures from the channel, and is the main observation from which the receiver can figure out what X is.
3. The noise component N. This corresponds to the AWGN noise which corrupts Y.
4. The channel coefficients H. This term captures the effect of the channel on the transmitted symbols passing through different paths. There are mechanisms which allow the receiver to estimate the value of H. So we can safely assume that H is known at the receiver.

So, your goal is to answer the following question:

Knowing Y, H, and the statistics of the AWGN noise (i.e., mean and variance), what is the best way of estimating the transmitted symbols X?

Guidelines: The goal here is to give an answer to the previous question. Note that there are several techniques to solve this equation in the literature. You can investigate one or more of these solutions. Please perform simulations to show the BER vs E_b/N_o performance of the techniques you consider for estimating X. In doing these simulations, you can assume that the transmitted symbols are BPSK symbols with energy $E_b = 1$. You can also assume that the coefficients of the channel are Complex Gaussian with zero mean and variance 1.

Part 3: Comparisons of coding techniques

In this part, you will investigate different coding techniques and their effectiveness in combatting channel degradation. The general idea of channel coding is shown in Figure 4.

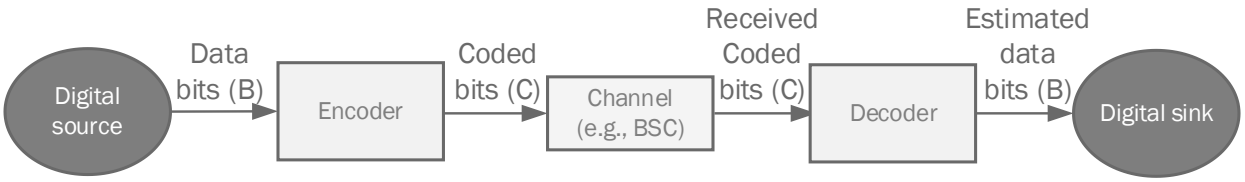

Figure 4 The process of channel coding.

In any channel coding technique, a set of B information bits are converted into a set of C bits that are transmitted through the channel. The set of B bits are called uncoded bits and the set of C bits are called coded bits. The value of B is sometimes referred to as *block length*, and the value of C is sometimes referred to as the *code length*. The typical situation is that $C > B$, which means that more bits are transmitted than the actual needed amount of information bits. This extra amount of bits (i.e., $C - B$ bits) is the cost that is paid to get better protection against detrimental channel effects. This extra amount is referred to as *code redundancy*. The amount of redundancy in a code is captured by the quantity $r = B/C$ which is called the *code rate*.

In this project, we investigate the error correcting performance of different codes across a Binary Symmetric Channel (BSC) with a transition probability p where $0 \leq p \leq 0.5$. We want to assess the performance of these codes for different values of p.

For a particular set of B uncoded bits and a given coding rate r, different codes or coding techniques would generate the C coded bits differently. We consider the following coding techniques in this project:

Repetition code: in this code, each information bit is repeated L times. If the coding rate of the repetition code is r, this means that each information bit is repeated $L = \lceil \frac{1}{r} \rceil$ times. The decoding algorithm suitable for BSC is a majority rule vote.

Convolutional code: this code generates a sequence of coded bits corresponding to an input stream of information bits. The encoding of convolutional codes is based on a given Generator polynomial, and the decoding of convolutional codes is using Viterbi algorithm. Feel free to choose any Generator polynomial you may think is useful. However, choosing a realistic convolutional code (e.g., one that is used in the LTE standard for example?) would show good effort from your side!

Polar codes: this is a form of linear block codes which has shown very good performance in different communication systems. The encoding of polar codes is using the polarization transformation, and the decoding can be done using Successive Cancellation (SC) decoding. The details of polar codes will be explained in an additional video that will be published soon.

Note: you need to implement your own encoder and decoder functions, and not use built-in or ready-made ones. However, feel free to use these available functions to verify your own implementations.