# Module 7: Design of a Simple Communication System Part 1

© 2019 Christoph Studer (studer@cornell.edu); Version 0.1

This module will teach you the basics of digital communication. First, we show the basic components of a typical digital communication system. We then show how a speech signal can be modulated onto a certain frequency and how it can be demodulated to recover the original signal. Finally, we show how digital information can be modulated onto a certain frequency. In the next module, we will transmit our digital signal over a simple MATLAB channel and develop a demodulator that recovers our information. ***Remember: Please ask us in case you have questions!***

## 9 Design of a Simple Communication System

We briefly describe the basics of digital communication systems. We then use a MATLAB model for a simply noisy communication channel and design a simple yet robust communication system for this channel. As you will see, the transmitter of our digital communication system will be very similar to the synthesizer designed in the previous module.

### 9.1 Basics of Digital Communication Systems

We start by describing the basics of digital communication systems. As it turns out, most communication systems consist of the same main building blocks. Only the details within these blocks change from system to system. Figure 17 illustrates the key components of a digital communication system.

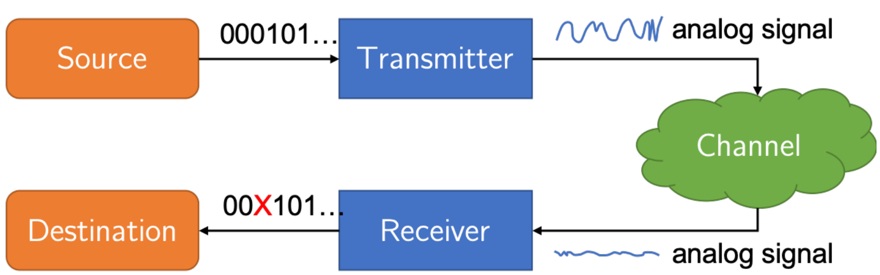



Figure 17: Illustration of a digital communication system. A source generates bits and modulates them onto an analog signal. This signal is transmitted over the channel. The receiver takes the distorted signal and generates estimates of the transmitted bits. A destination hopefully receives all bits without errors.

A *source* generates digital information in the form of bits, i.e., zeros and ones. The source could be a computer program (for example information from an email client or a web browser) or a cell-phone (for example information that contains digitized speech or chat messages). The *transmitter* takes a stream of bits (for example 1000 bits) and converts them into a continuous, analog waveform. This waveform is then transmitted over a *channel* either using an antenna or a loudspeaker. The channel can be air (as for acoustic and electromagnetic communication) or also a cable (which is the case for cable modems or fiber-optical communication systems). Unfortunately, channels cause unwanted distortions so that the output of the channel is no longer the same as that of the input. The most common source of such distortions are (i) *attenuation*, which is due to the fact that not all energy that has been transmitted arrives at the receiver, (ii) *noise*, which can originate from electronic components or interference caused by other communication systems that try to use the same frequency, and (iii) *multi-path propagation*, which is caused by the transmit signals being reflected at walls, trees, and other objects, resulting in multiple echoes that arrive at different times at the receiver. ***Try to remember Figure 16 from Module 5 where we simultaneously played and recorded a simple chirp signal!*** The *receiver* then takes the distorted signal and performs demodulation. The goal of demodulation is to generate estimates of the original digital information. In words, the receiver tries to make the best possible guess what the transmitted information was. The estimates of the transmitted bits are then passed to a *destination*. In general, the hope is that the destination receives exactly the information that the source generated. However, due to the various forms of distortions in the channel, errors may occur. Hence, in practice, one designs the transmitted signal in such a way that it is robust to distortions (caused by the transmitter, channel, and receiver), in a sense that a receiver is able to *reliably* extract the transmitted information.

In an acoustic communication system, which are, for example, used in practice for underwater communication[1], the transmitter would be a loudspeaker, the receiver a microphone, and the channel is the physical space between and around the loudspeaker and transmitter. The transmitter would take digital information, modulate it on one or many frequencies and transmit over the air (also known as the *channel*). The receiver would take the signal and estimate the transmitted bits. Note that the receiver has to know how the transmit signals are generated in order to undo the effect of the channel.

In a communication system that uses electromagnetic waves (such as Wi-Fi, LTE, or Bluetooth), the transmitter would generate voltages that are radiated over an antenna. The receiver would be another antenna (for example at a cell-phone, access point, base station, etc.) and generate estimates of the transmitted bits. As for the acoustic communication system, the channel is the physical space between and around the transmitter and receiver. It is important to realize that electromagnetic waves and acoustic waves have very similar propagation properties. The key difference is the speed of propagation: The speed of sound is about $343\,\mathrm{m/s}$ whereas the speed of electromagnetic waves in air are close to the speed of light, which is about $300,000,000\,\mathrm{m/s}$. Other properties, such as reflection, diffraction, bending, etc. are very similar but depend on the material and size of objects. This implies that by learning how acoustic communication works, one can apply that knowledge directly to electromagnetic waves. In contrast to electromagnetic waves, we are equipped with an "antenna" for acoustic waves (our ears!) and we can actually listen to the generated communication signals, which is great in building intuition.

### 9.2 Amplitude Modulation

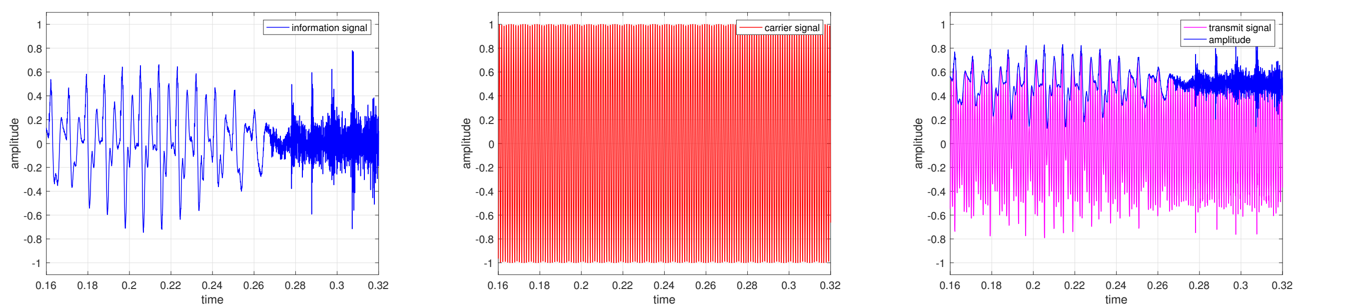
When building our first communication systems, we will use one of the simplest modulation schemes: *amplitude modulation* (AM, for short). The idea underlying AM is actually quite simple. Take a sine function at a given carrier (or center) frequency $f_c$ and modulate its amplitude over time using the *information signal* you wish to transmit over the air. Assume that the information signal you want to transmit is modeled as a continuous function $s(t)$ where $t$ represents time. Furthermore, assume that the amplitudes of the information signal are in the interval $[-1,+1]$; mathematically this can be written as $s(t) \in [-1,+1]$, $t \in \mathbb{R}$.

[1]Electromagnetic signals do not propagate well underwater as water is, in general, conducting. Hence, a number of communication systems that can be used to transmit information in water rely on acoustic data transmission.

Amplitude modulation would transmit the following modulated signal:

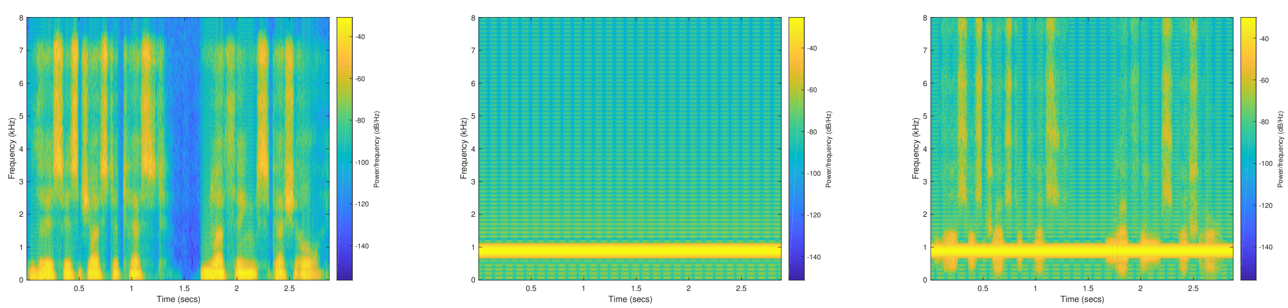
$$y(t) = \frac{1}{2}(s(t) + 1) * \sin(2\pi f_c t), \tag{6}$$

where $\frac{1}{2}(s(t)+1)$ is in the information signal now shifted to the interval $[0, 1]$ and modulates the amplitude of the sine function; if the information signal $s(t)$ is large (close to $+1$), then the amplitude of the sine wave will be large; if the information signal $s(t)$ is negative (close to $-1$), then the amplitude of the sine wave would be small.




(a) Information signal (speech). (b) 880 Hz carrier signal. (c) AM transmit signal.

Figure 18: Transmission of speech using amplitude modulation (AM) at a carrier frequency of 880 Hz.




(a) Information signal (speech). (b) 880 Hz carrier signal. (c) AM transmit signal.

Figure 19: Spectrogram of the speech signal, the carrier signal, and of the transmitted AM signal. We can see that the AM transmit signal has modulated the speech signal round the carrier frequency 880 Hz.

Figure 18 illustrates the principle of AM applied to the speech signal `speech-male.wav`. Figure 19 shows the corresponding spectrograms. We modulate the amplitude of the sine wave which results in the signal shown in the last figure in Figure 18. By inspecting the last figure of Figure 19, one can clearly observe that the transmitted AM signal $y(t)$ contains the speech signal around the carrier frequency of 880 Hz. This process is known as *modulation*, meaning that we have modulated the speech signal to the carrier frequency of $f_c = 880$ Hz. Remember the reason why we want to modulate an information signal onto a certain carrier frequency: Typically multiple communication systems are used at the same time and we want to transmit information at the same time. By modulating the information signal onto a certain location in the spectrum, we can have multiple wireless system to coexist! Different wireless systems occupy different frequencies and we can fill up the entire frequency spectrum with information signals.

What is quite surprising: AM radio uses *exactly* this strategy (maybe you have seen FM radio in your car before, but often you can also select AM radio which does not sound as good and is often used to only broadcast news...). Put simply, you have just learned how AM radios are transmitting music and news over the electromagnetic spectrum. Equation (6) is exactly describing the process AM radio transmitters are implementing! Note that AM radios are using carrier frequencies of around 540 kHz to 1,600 kHz, which are particularly good for transmitting signals over extremely long distances (almost around the entire world under good weather conditions).

**Activity 23: Listen to AM signals**

To get more intuition about how AM can be implemented in MATLAB, copy-paste the following code snipped into a MATLAB script and execute it:

```
% change this ID according to your settings
OutID = 3;
% define carrier frequency in Hz
FC = 880;
% load speech signal
[s,FS] = load_audio('examples/speech-male.wav');
s = s'; % convert into a row vector
% extract duration of speech signal
duration = length(s)/FS; % seconds
% generate sine wave at carrier frequency FC
t = linspace(0,length(s)/FS,length(s)); % sample indices
carrier = sin(2*pi*FC*t);
% AM modulation
y = carrier.*(s+1)*0.5;
% play AM modulated signal
play_audio(y,FS,OutID);
```

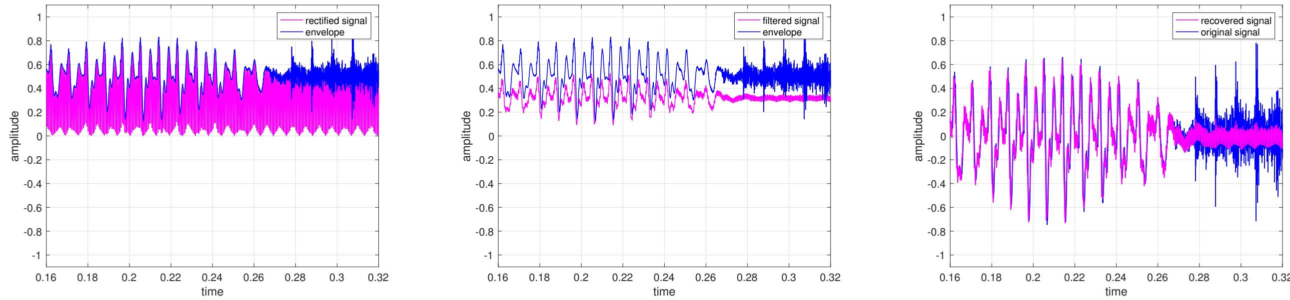
You should clearly hear the carrier frequency of 880 Hz and, a bit more quiet, the speech signal. Feel free to modulate other signals and try out other carrier frequencies.

### 9.3 Demodulating AM Signals

You probably wondered how does one get the original information signal back from an amplitude modulated signal. It turns out that the demodulation process is not too difficult. There exist complicated methods that yield excellent quality and simple methods that yield acceptable quality. We will explain one of the most simple demodulation methods, which is known as *envelope detector*. Consider the right-most graph in Figure 18. We are interested in getting the blue curve back—this is called the envelope of the AM signal. A simple trick of getting the signal back is to first "rectify" the received signal followed by filtering out unwanted frequencies. Let us look at these steps in detail.

Rectifying is the process of taking the absolute value of the signal $y(t)$ in equation (6). What this does, intuitively, is to push negative amplitudes to the positive side; the left-side of Figure 20 shows the outcome of this operation. The next step is to remove the carrier signal. By using rectification, we have doubled the carrier frequency to 1760 Hz.[2] However, as you can see on the left of Figure 20, the envelope that carries the speech signal remained unchanged. Hence, we now have to remove frequencies that are higher than 880 Hz, which can be achieved by using a low-pass filter (that only passes low frequencies). In the

[2] To see this, draw a sine wave and take its absolute value. You will see that the period of the resulting signal is half of that of the original sine wave. Hence, the frequency of the rectified signal has been doubled.




(a) Rectified AM signal. (b) Output of low-pass filter. (c) Recovered signal.

Figure 20: AM signal demodulation using an envelope detector. Left: the received signal is rectified; center: the rectified signal is low-pass filtered; right: the signal is amplified and the mean is removed.

following, we use a very simple low-pass filter that simply computes a running average of the neighboring 10 samples—in a real AM receiver, one would use much better filters. The center of Figure 20 shows the outcome of this operation. Finally, we amplify the filtered signal and remove the offset (the signal is not centered around zero). The result of this simple envelope detector is shown on the right side of Figure 20. You can see that lower frequencies (appearing at around $t = 0.18$ seconds) of the original speech signal (blue curve) are recovered almost perfectly; higher frequencies (which appear at around $t = 0.28$ seconds), however, are almost lost. This is a result of using a simple low-pass filter. Better filters would preserve most of the high frequencies but they are more difficult to understand!

**Activity 24: Listen to an AM demodulated signal**

To get more intuition about how AM demodulation can be implemented in MATLAB, copy-paste the following code and append it to the MATLAB script generated in Activity 1:

```
% rectify the received signal
yr = abs(y);
% apply a simple averaging low-pass filter (10 samples)
yf = conv(yr,ones(1,10)/10);
yf = yf(1:length(t));
% center the signal (remove the mean) and amplify it
yc = yf - mean(yf);
ya = yc*3.2; % manually tuned amplification parameter
% play the signal back
play_audio(ya,FS,OutID);
```

By executing this script, you should hear the original speech signal. Since we are not using a particularly good low-pass filter, the carrier signal is still audible (but not very loud); you should also hear that high frequencies are mostly gone. You can try to average more or fewer neighboring samples to hear the effect of the low-pass filter we are using. We emphasize that old AM radio receivers use exactly the same principle as implemented above, but are implemented using a few analog circuit components (they did not use MATLAB of course).

If you are interested in learning more why averaging neighboring samples removes high frequencies (i.e., implements a low-pass filter), feel free to talk to us—it is quite intuitive!

### 9.4 Digital Data Transmission using Amplitude Modulation

So far, we have explained how to amplitude-modulate an analog information signal $s(t)$ and how to demodulate the received signal $y(t)$. We now show how digital information can be transmitted using the same strategy. The principle is straightforward (after learning and understanding the principles of AM): We take a stream of bits and generate sine waves whose amplitudes are modulated depending on whether the bit was zero or one. For example, if a bit is zero, then we use a small amplitude; if a bit is one, then we use a large amplitude. In what follows, we will build a transmitter that modulates bits onto sine waves at a certain carrier frequency. We will later use a test channel to model effects such as attenuation and noise. Finally, we design a receiver that tries to recover the transmitted bits.

Let us start with the transmitter. First, we generate a row vector in MATLAB that contains the bits to be transmitted (these bits are information we would like to transmit to the receiver). You can use the following example:

```
bits = [1,0,0,1,0,1,1,1];
```

which means we want to transmit the eight bits starting from left to right. In practice, this bit vector would come from the application that wants to transmit bits over the air. Now, assume that for each bit, we will send a short sine wave at a certain carrier frequency. Let us define the carrier frequency `FC` of the sine wave as

```
FC = 880; % Hz
```

and define the sampling frequency `FS` to be 44, 100 Hz. For our simple transmitter, let us send sine waves of length 1000 samples (which is 1000/`FS` seconds) for every bit. This can be accomplished by the following two MATLAB commands:

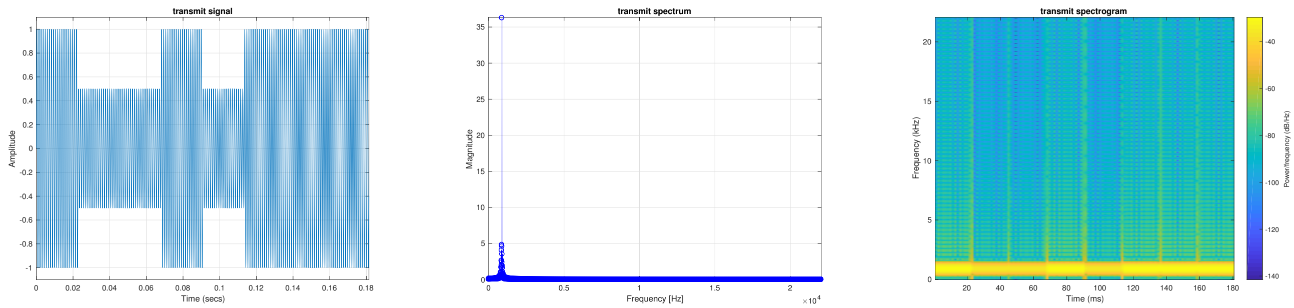
```
tx_len = 1000;
sine = sin(2*pi*FC*(0:tx_len-1)/FS);
```

The row vector `sine` consists of 1000 samples and contains a sine wave at the carrier frequency `FC` of 880 Hz. The statement `(0:tx_len-1)/FS` creates a row vector consisting of `tx_len` entries which contain sampling instants that are $T_s = 1/f_s$ seconds apart.

Now comes the important part. We have to define a convention of what amplitudes correspond to a bit of zero or a bit of one. In what follows, we assume that for a bit that is 1, we use an amplitude of 1; for a bit that is 0, we will use an amplitude of 0.5. Note that it is important for the receiver to know how the bits are mapped to amplitudes. We can now create a transmit signal that consists of eight (because we are sending eight bits) amplitude-modulated sine waves of length 1000 samples. If you remember how we designed our simple sound synthesizer, AM modulation uses exactly the same principle, but we only modulate the amplitude and not the frequency (the frequency is fixed at `FC`). We use the following set of MATLAB commands to accomplish this task:

```
tx = [];
for kk=1:length(bits)
    if bits(kk)==1
        tx = [ tx , sine ];
    else
        tx = [ tx , 0.5*sine ];
    end
end
y = tx;
```

As for the synthesizer, we first create an empty vector called `tx`. We then iterate through each bit of the vector `bits` (so we perform eight iterations). For each bit `bits(kk)`, we check whether it is 1. This is what this `if` statement is doing; the construct `bits(kk)==1` checks for equality. If the `kk`th bit is 1, then we append the sine function (which has amplitude 1). Otherwise, this is what the `else` statement is for, we append a sine wave with half the amplitude to our transmit vector `tx`.




(a) Time-domain digital AM signal.

(b) Spectrum of digital AM signal.

(c) Spectrogram of digital AM signal.

Figure 21: Digital data transmission using AM. Left: A bit 1 is modulated onto a sine wave with amplitude 1; a bit 0 is modulated onto a sine wave with amplitude 0.5. Middle: The spectrum is localized around the carrier frequency of 880 Hz. Right: The spectrogram shows only minor fluctuations over time.

Figure 21 shows the time-domain, spectrum, and spectrogram of the transmitted digital AM signal. On the left side, you can clearly see the eight bits modulated onto a sine wave. In the center, you can see that the signal's spectrum is localized around the carrier frequency of 880 Hz. On the right side, you see that the spectrogram shows only minor changes in the frequency domain over time. Note that the changes over time come from the changes in amplitude and the fact that we are concatenating[3] sine waves.

**Activity 25: Listen to the digital AM signal**

Implement the above digital amplitude modulator and listen to the waveform that it produces. It should sound almost like Morse code, especially if you increase the number of bits that you would like to transmit. You can change the carrier frequency and the duration of each sine wave (`tx_len`). Remember that if you reduce `tx_len`, you would send more bits in shorter time, which leads to higher throughput. For this case, the throughput can be calculated as

$$\textit{Throughput} = \frac{\textit{number of bits}}{\textit{second}} = \frac{\texttt{FS}}{\texttt{tx\_len}} \quad \text{bits/second.} \tag{7}$$

For the above example, the throughput was 44.1 bits/second. Further increasing the throughput will eventually result in transmission errors—we will discuss this later.

[3]Concatenation at this carrier frequency is not perfect so there are little "jumps" in the generated signals that cause these vertical lines in the spectrogram.