1 Project Objective & Learning Outcomes

The objective of this project is to introduce the concept of pulse shaping for reliable data transmission and the introduction of receiver implementations for recovering analog waveforms.

From this project, it is expected that the following learning outcomes are achieved:

- Obtain an understanding of how sampled information can be transformed into various pulse shapes, transmitted across a channel, and successfully recovered at the receiver.

- Learn about Nyquist type pulse shapes, such as sinc pulses and raised cosine pulses.

- Experience the effect of intersymbol interference and realize how equalization can counteract these adverse effects.

- Gain proficiency in implementing amplitude modulation and phase modulation systems that take baseband signals and transform them into passband signals.

- Devise approaches that can convert passband signals, such as amplitude modulation and phase modulation, back down to bandpass signals.

2 Preparations

For this project, we will be depending on several functions available from the CommPy toolkit. Since this toolkit was not initially installed on your ECE3311 Ubuntu image for VirtualBox, you will need to install it via terminal. To do this, please run the following command from terminal:

```
pip3 install scikit-commpy
```

Once installed, we should be good to go with the rest of this project.

3 Pulse Shaping Fundamentals

It is important that one enters this project with the mindset that we will take information, transform into another form or representation suitable for transmission across some medium, and then attempt to reconstruct this received signal into the original information. An illustration highlighting this viewpoint is shown in Figure 1, where we have an information source and an information sink, and we wish to have an information exchange between these two locations. We assume that there exists some sort of medium separating these two locations, and that sending the desired information is either impossible or significantly inefficient in its current form. Consequently, some sort of "processing"
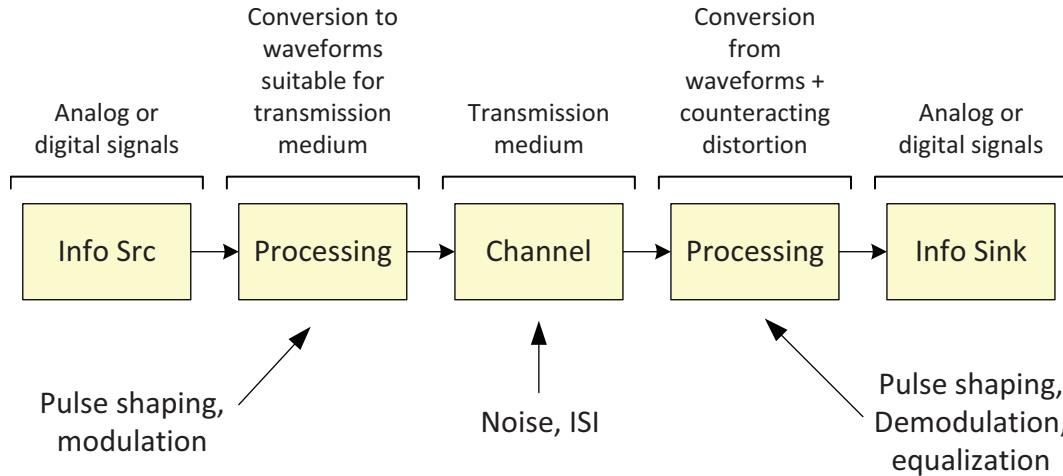
Figure 1: Illustration of a generic communication system, with an information source and sink, a channel separating the two locations, and some information processing designed to enable efficient and reliable information between the two points.

(either analog processing, digital processing, or a combination of the two) is required to convert this information into a format that is more amenable to the transmission medium.

As an example of processing information before and after it is communicated across a transmission channel, let us look what happens when we take an impulse train of samples with amplitude values of ± 1 and we wish to communicate it across a channel. Since the channel can potentially introduce distortion to the transmission, we often apply a pulse shaping filter to the outgoing transmission in order to mitigate impairments such as intersymbol interference (ISI). One pulse shape that is sometime used in practice is the *sinc*, which is equivalent to $\sin(x)/x$. Sinc pulse shapes satisfy the Nyquist criterion for zero-ISI, which means when we transmit a string of information possessing sinc pulse shapes, they do not incur any ISI at the receiver when we recover the desired samples. This transmission and reception process of data involving sinc pulse shapes is presented in the Project 3 Jupyter Notebook available from Canvas.

We observe in Figure 2 how these original data samples and the resulting pulse shaped transmitted signal all appear when properly time aligned. In particular, we can see how the amplitude values of the original samples and the reconstructed ones are exactly the same, and that they have the exact same values as those corresponding points on the transmitted pulse shaped signal at the exact same time instants. Even though the pulse shaped signal does not immediately appear to contain the original samples, with the proper timing we can extract this information perfectly due to the Nyquist criterion for zero ISI.

4 Raised Cosine Pulse Shapes

Similar to the sinc pulse shaping example from the previous section, another pulse shape that satisfies the Nyquist criterion for zero ISI is the *raised cosine* pulse shape. To generate the raised cosine pulse shape, please refer to the Project 3 Jupyter Notebook available from Canvas. In this case, we can control the roll-off factor of the raised cosine pulse shape, *i.e.*, the amount that the pulse shape tappers off, the sampling rate of the filter, and several other parameters.
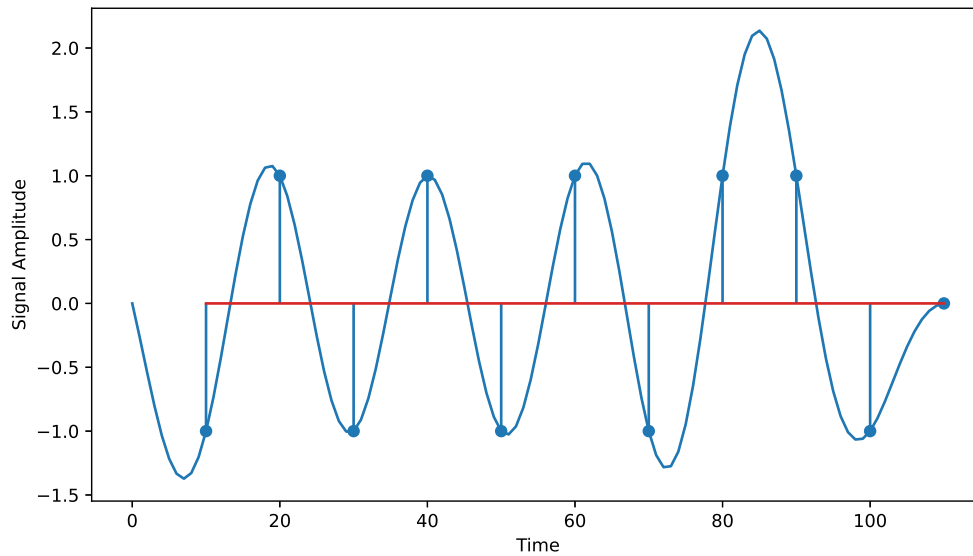
Figure 2: Example of how sinc pulse shaping can be applied to an impulse train and then the original samples can be recovered with no ISI present.

Question 1 (2 points): Based on the sinc pulse shape example in Section 3, you are tasked with implementing a similar pulse shaping transmission system using the raised cosine pulse shape. Furthermore, you will need to demonstrate that it is possible to extract the original samples from the recovered pulse shaped signal, similar to the plot shown in Figure 3.

5   Square-Root Raised Cosine Pulse Shapes

When a raised cosine filter is applied to a communication system in order to pulse shape a transmission, it is almost always applied at the transmitter while the receiver is only tasked with recovering the desired samples. However, there is an alternative approach to implementing the raised cosine pulse shaping between the transmitter and the receiver. Instead of employing the raised cosine pulse shaping filter only at the transmitter, we can instead split the raised cosine filter between the two ends of the channel such that we they are both applied to the sampled data, the aggregate response is a raised cosine filter. This split pulse shaping filter at the transmitter and the receiver is called a square-root raised cosine filter (implementation details located in Project 3 Jupyter Notebook available from Canvas).

Although splitting up the raised cosine filter between the transmitter and the receiver in order to perform pulse shaping at the transmitter and filtering at the receiver achieves the overall effect of raised cosine pulse shaping, only performing one of these operations instead of both of them in unison will yield pulse shapes that do **not** satisfy the Nyquist criterion for zero ISI.
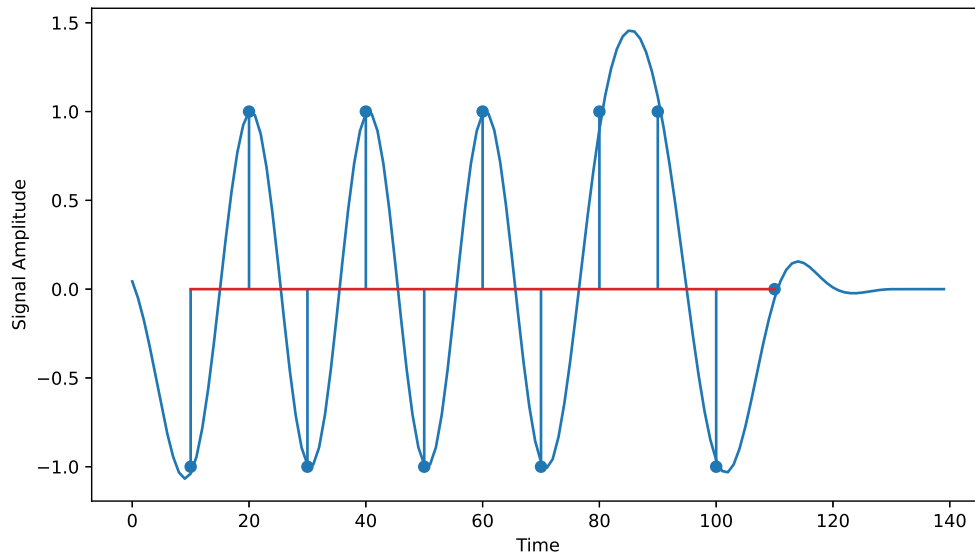
Figure 3: Example of how raised cosine pulse shaping can be applied to an impulse train and then the original samples can be recovered with no ISI present.

Question 2 (2 points): Please demonstrate that only using the square-root raised cosine pulse shaping filter at the transmitter will not eliminate ISI when the received signal is sampled in order to recover the original data, *i.e.*, produce the plot shown in Figure 4.

Based on Figure 4, we see that we have a problem if we only use a square-root raised cosine pulse shaping filter at the transmitter since it does not satisfy the Nyquist criterion for zero ISI. However, when another square-root raised cosine filter is employed at the receiver, the recovered samples do not suffer the impact of ISI, *i.e.*, the combine effect of the two square-root raised cosine filters satisfies Nyquist criterion for zero ISI (see Figure 5).

Question 3 (2 points): Please demonstrate how two square-root raised cosine filters in series can yield recovered samples that do not suffer the impact of ISI.

6   Intersymbol Interference & Equalization

So far, we have talked about ISI but we have not really explored how to generate it within a computer simulation. One approach for generating the effects of ISI when a transmission is sent through a channel is to apply an FIR filter to the samples of that signal. What the FIR filter does is it takes samples of the signal and adds weighted versions of these samples to subsequent samples of the transmission, effectively "smearing" the samples across time. As oppose to noise, which is completely random and added to a transmission, a channel that introduces ISI is deterministic (*i.e.*, has a known and defined behavior) and takes the sample information of the signal and distributes it to other time instants within the signal.

Suppose we take one of our pulse shaped signals from the previous sections and pass that signal
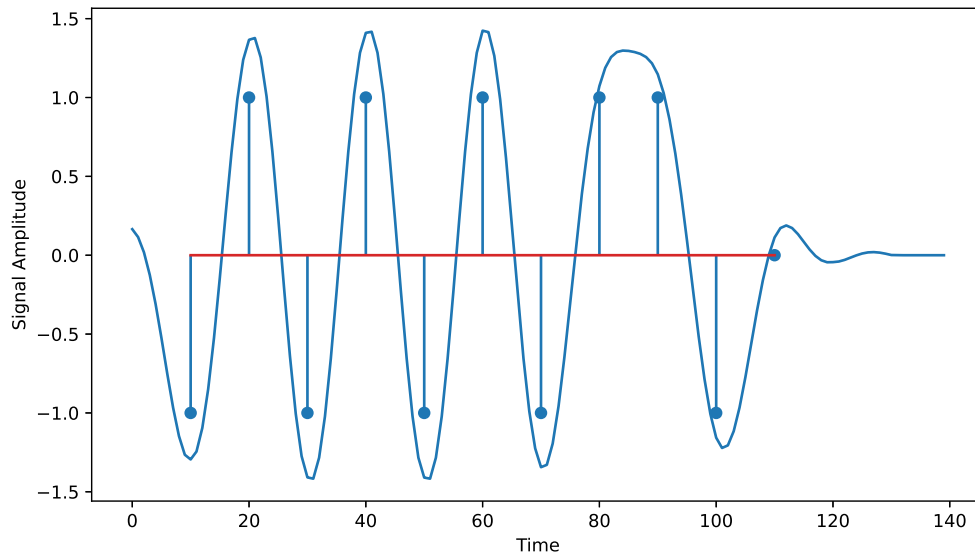
Figure 4: Example showing how the application of a single square-root raised cosine pulse shaping filter at the transmitter does not effectively compensate for the ISI introduced in the communication system.

through the FIR filter `h`, we should end up experiencing a situation where the recovered samples do not match the original sample data, as shown in Figure 6. Despite the use of our pulse shaping filters, the introduction of a channel with a profile that is not ideal can be problematic since the pulse shaping filters are only designed to handle ISI with an ideal channel.

> Question 4 (2 points): Please demonstrate this same situation by replicating the impact of a non-ideal channel as shown in Figure 6.

To counteract the effects of a non-ideal channel that introduces ISI, many communication systems use a type of filter called an *equalizer*, which theoretically is nothing more than the inverse filter of the channel filter. Consequently, a communication system would possess pulse shaping filters in combination with an equalizer to counteract the effects of the channel. For example, if we apply the inverse filter to `h` to the transmission shown in Figure 6, we obtain a signal whose recovered samples are identical to the original data (see Figure 7).

> Question 5 (2 points): For this section, please obtain a filter that can mitigate the impact of h, and demonstrate its effectiveness using a plot similar to Figure 7. It is recommended to use a function such as `scipy.signal.lfilter`.

As a side note, equalizer design is more trickier than it appears since the inverse of an FIR filter is often an IIR filter, which is difficult to model and implement. Furthermore, if we implement the FIR equivalent of an IIR filter, we end up with an approximation of the target equalizer and this could lead to imperfect results when recovering the desired samples.
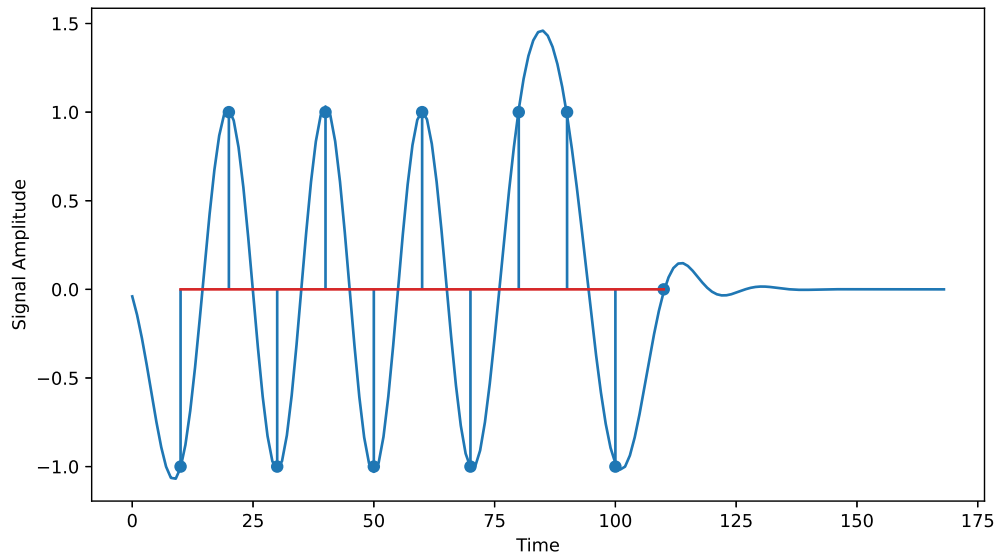
Figure 5: Example of when two square-root raised cosine filters are employed in a communication system, with one applied at the transmitter while the other is applied at the receiver. Note how the ISI effects are mitigated relative to when just one square-root raised cosine filter is employed.

7 AM, PM, & FM

Up to this point in this project, we have employed baseband impulse trains with varying amplitudes as sources of information. We will now switch over to examining analog waveforms that are moduluated from baseband frequencies, *i.e.*, frequencies centered around 0 Hz, to passband frequencies. Specifically, we will explore three forms of passband modulation: *Amplitude Modulation* (AM), *Phase Modulation* (PM), and *Frequency Modulation* (FM). Although the process of generating these passband waveforms is relatively straightforward, it is the recovery of the baseband analog waveform from these modulation schemes that will be more challenging.

AM waveforms are relatively straightforward to generate, as well as to understand in both the time and frequency domains. In Figure 8, we can readily observe how the envelope of the original analog waveform is present in the AM signal. Notice how the sinusoidal signal effectively "fills in" the envelope of the original analog waveform. Although the analog waveform is present, it is the higher frequency shaped sinusoidal signal that is transmitted across the channel.

Similarly, PM waveforms manipulate the phase information of sinusoidal signals, which carry this information across a channel to a target receiver. An example of a PM waveform is shown in Figure 9. Notice how the amplitude level remains the same for the waveform. However, the phase of the sinusoidal waveform is changing over time, although it is difficult to observe.

> Question 6 (3 points): Please implement and demonstrate a PM waveform similar to the one shown in Figure 9.
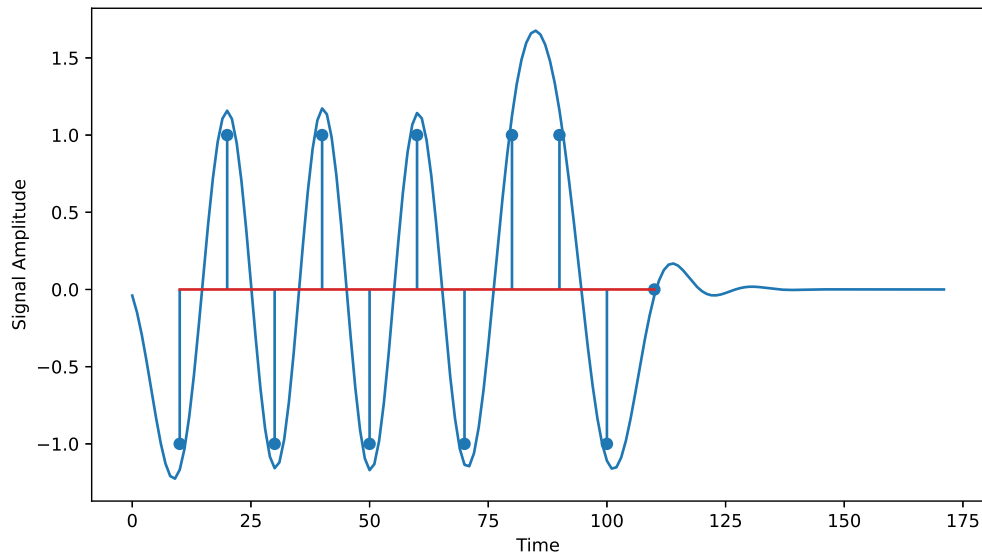
Figure 6: Illustration of how a non-ideal channel filter can affect the ISI performance of a communication system.

8 AM & PM Detectors

As mentioned before, the challenge with using AM and PM modulation is not so much with the transmitter as it is with the receiver. For instance, with an AM waveform, there exists several approaches of building detectors that can help recover the original baseband analog waveform, including Hilbert transforms and square law devices. Suppose we focus on the second approach, where we apply a square law device to an AM waveform, apply a lowpass filter to the result, and then take the square root of the output. Although this series of operations appear somewhat unclear in the time domain, looking at the process in the frequency domain will make much more sense. Here is the recommended approach to implementing an AM detector:

1. Take the product of the AM waveform with itself, which contains a $\cos(\omega_c t)$. This operation will result in a DC term and a double frequency term since multiplying a $\cos(\omega_c t)$ with itself will yield $\cos^2(\omega_c t)$, which can then be rewritten via trignometric identity to be equal to $0.5 + 0.5 \cos(2\omega_c t)$.

2. Apply a lowpass filter to remove the double frequency term, leaving the DC term which is the original analog waveform.

3. Apply a square root function to the filter output to obtain the baseband signal.

Question 7 (5 points): Please implement your own AM waveform envelope detector using a square law device.

Question 8 (5 points): Please implement a product detector for recovering analog signals from PM waveforms.
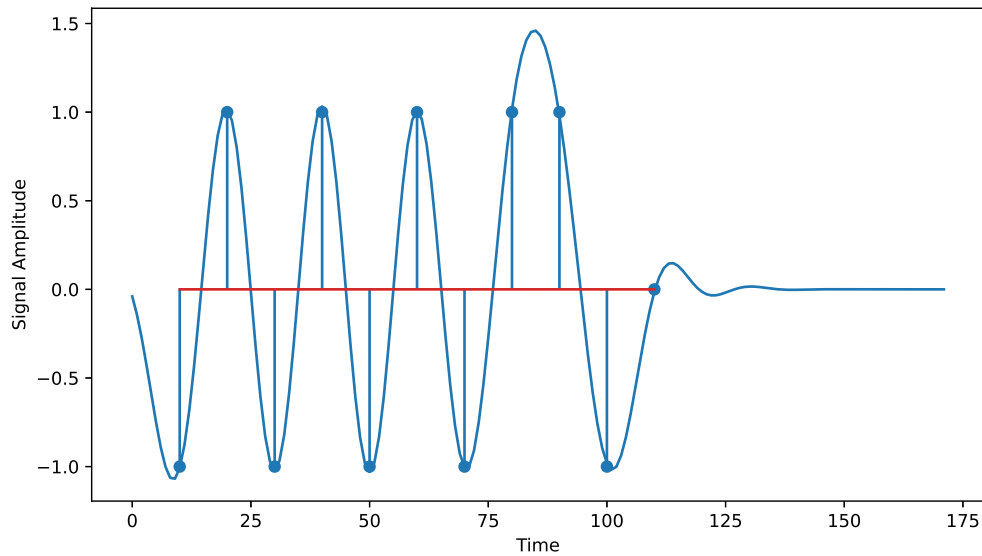
Figure 7: Demonstration of an equalizer applied to the signal shown in Figure 6. Note how the effects of ISI have been mitigated.

9   Make Your Own FM Transmitter & Receiver

Worcester has about a half-dozen local radio stations. In this section, you will be given four seconds of a mystery FM station and asked to take it through several filtering and decimating steps so that you can play it out of your speakers and listen to it. If we interpret the signal as an 8-bit integer, the first thing you would want to do is to center it at 0 Hz and turn the sequential in-phase/quadrature values into single array of complex elements.

Wireless transmissions can be shifted or spread in the frequency domain by a variety of causes, including Local Oscillator (LO) drift and Doppler effects. In the next project you will implement a Phase-Locked Loop (PLL) which will seek out the received carrier and demodulate it to baseband.

> Question 9 (5 points): Manually shift the first 5,000 samples plotted by 1.663 MHz such that it falls on the ideal 108 MHz carrier it was originally transmitted on. Use the function `plot_fft_IQ.py` available in Canvas to plot the result.

Worcester's radio traffic is not very busy. Nevertheless, if you observe several other stations located within your frequency window of interest, this will result in co-channel interference as well as increase the noise floor, thus potentially corrupting the final audio signal. As a result, filters are need to extract the target signal.

> Question 10 (5 points): To remove the side noise and begin lowering the sampling rate to the audio range, decimate the shifted signal by 8 and filter with a Hamming window. Plot your result.

Now we are ready to demodulate to baseband. The human ear can only hear frequencies up to about 22 kHz, so we will not be able to hear anything if we play the file as it is now.
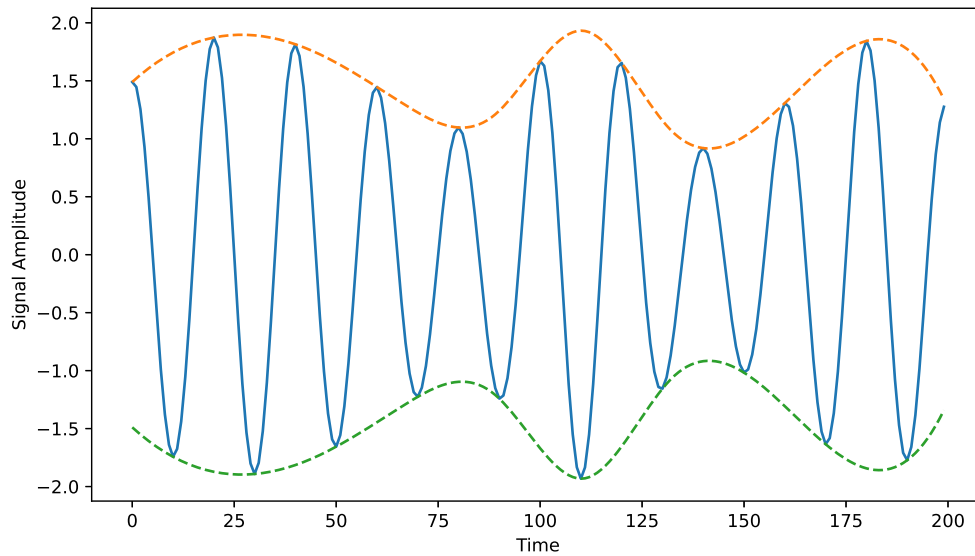
8

Figure 8: Example of an AM waveform. Notice that the envelope contains the original baseband analog waveform.

Question 11 (5 points): Using the supplied `FM_IQ_Demod.py` function, demodulate the signal to baseband. Plot the resulting signal.

Question 12 (5 points): Using a Hamming window and a decimation factor 10, you should be able to listen what you have demodulated. Describe what you hear, and plot the final signal.

10 Report Submission

For each project in this course, the report submission to be uploaded to the ECE3311 Canvas website will consist of a single comprehensive Jupyter Notebook (.ipynb) file and nothing else unless it is requested. For this project, only the Jupyter Notebook is to be submitted electronically by the due date. Failure to submit the Jupyter Notebook by the specified due date and time will result in a grade of "0%" for the project.

Several important items to keep in mind when preparing your submission:

- Include the course number, project team number, names of team members, and submission date at the top of the Jupyter Notebook.

- Make sure your source code is thoroughly documented such that it is made clear what exact the program is doing line-by-line. Adequate commenting is worth **5 points** of the total score for this project.

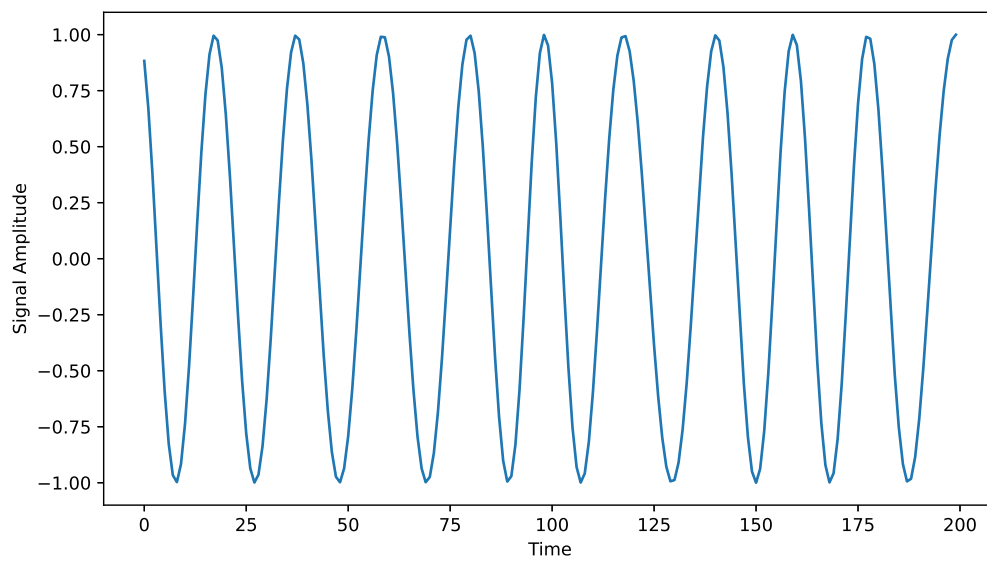- Responses to all questions indicated in the project handout. Please make sure that the responses are of sufficient detail.

Figure 9: Example of a PM waveform.