ECE 345: Linear Systems and Signals Fall 2020
Professors Sarwate and El Rouayheb Due: November, 16, 2020

Lab 3

This lab is made up of two mini-labs. See the Canvas page for the lab for details and for file downloads. To turn in this lab, you need to submit a `.zip` file containing the following three files:

- a completed lab report **in PDF format**.
- your code for part 1 of the minilab
- your code for part 2 of the minilab (in a separate file/set of files)

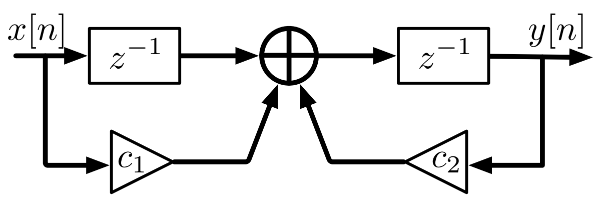
Make sure your file names are labeled so that we can easily find your material. For example, `lab3problem1d.m` for part d of the first half of the minilab.

Points will be taken off for non-PDF submissions. Why are we being sticklers about formatting? In the real world, your work will also have to fit formatting guidelines. If you bid on a government contract, for example, and don't follow the guidelines, your bid/proposal could be rejected without review. Get in the habit of following the formatting rules!

Along those lines, do not forget to label your plots – it's part of the rubric!

1 Mini-lab 1: Z-transforms

Consider the following block diagram for a discrete time system. Recall that z^{-1} is a unit delay and that triangles represent constant gains of c_1 and c_2.



(a) Determine the system transfer function $H(z)$.

(b) Find the poles and zeros.

(c) Find values of c_1 and c_2 that make the system stable.

(d) Plot the pole-zero diagram in MATLAB using the `roots` function. Use `axis equal` to make the axes square and plot the unit circle `ucirc = exp(j * linspace(0, 2*pi, 200))` using `plot(real(ucirc), imag(ucirc), 'k:')`. Use the values of c_1 and c_2 that you found.

(e) Plot the frequency response $|H(e^{j\omega})|$ for $\omega \in [-\pi, \pi]$ using the values of c_1 and c_2 that you found. This is the z-transform evaluated on the unit circle in the complex plane. This is referred to as the Discrete time Fourier transform (DTFT). Hint: You may want to use `linspace` to generate evenly spaced points spaced between $-\pi$ and π, and the `H = @(z) H(z)` way of defining a function/signal.

(f) Find values of c_1 and c_2 such that the system acts approximately like a lowpass filter. That is, $|H(e^{j\omega})|$ is small for large ω close to $\pm\pi$, and $|H(e^{j\omega})|$ is large (or close to 1) for small ω close to 0. Plot the frequency response $|H(e^{j\omega})|$ for $\omega \in [-\pi, \pi]$ using the new values of c_1 and c_2 that you found.

(g) If $x[n] = 0$ does $y[n] = 0$?

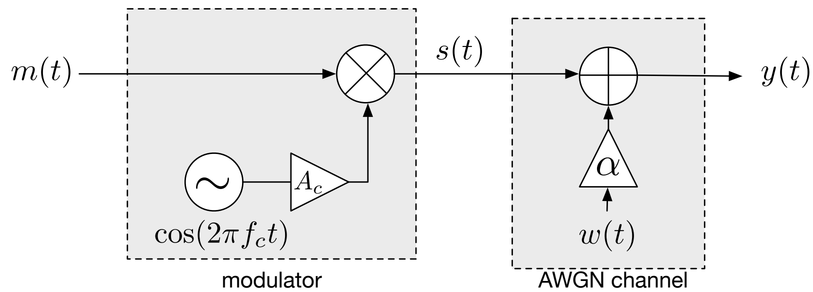
(h) What kind of filter do you get by setting $c_1 = -1$ and $c_2 = -0.8$: highpass, lowpass, or bandpass?

2 Mini-lab 2: AM Radio

Amplitude modulation (AM) radios use a high-frequency carrier wave to transmit an analog (CT) message signal. If the message signal is $m(t)$, the modulated signal is

$$s(t) = A_c m(t) \cos(2\pi f_c t), \tag{1}$$

where $c(t) = A_c \cos(2\pi f_c t)$ is called the *carrier signal* and f_c is the *carrier frequency*. The modulated signal is called a *passband* signal because the frequency content of the signal is centered at the carrier frequency. For example, WADB in Asbury Park is 1310 AM, which corresponds to $f_c = 1310$ kHz. Here is a a block diagram of the modulator.



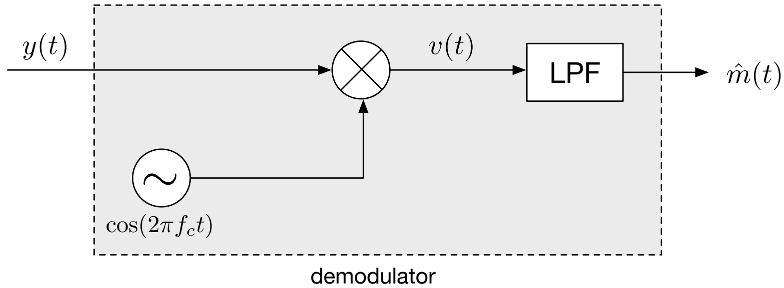
Communication channels are *noisy* and the goal of communication engineering is to enable reliable communication even though there is noise. One of the simplest models for a communication channel is as a system that adds random Gaussian-distributed noise to the input: this is called the additive white Gaussian noise (AWGN) channel. In this problem we will simulate an AM transmitter, AWGN channel, and receiver.

(a) Download the file `oddity.wav` from the assignment web page. Use the `audioread` function to read in the audio file (see the documentation). Verify that the sampling rate is indeed $f_s = 44.1$ kHz. MATLAB will read in both stereo tracks (left and right); extract the first column (which should be the left track). We will call this signal $m(t)$. Plot $m(t)$ and then use `soundsc` to listen to the clip. For a CT signal represented in a computer using a sampling rate of f_s, each element in the array represents the signal value in an interval of length $\frac{1}{f_s}$. So the total energy contributed by one element of the array is the amplitude squared times $\frac{1}{f_s}$. Compute the total energy in the track you extracted. Hint: You may want to use the MATLAB `sum` function.

(b) Suppose we want to simulate transmission for the AM 1310 radio station. Note that the carrier frequency $f_c = 1310$ kHz is much higher than the sampling rate of the audio signals you've generated so far. That means that you will have to set the sampling rate for the simulation to be even higher. In this case, try a sampling rate of $f_0 = 10^7$ (10 MHz) for the simulation. However, the $m(t)$ from the audio file is sampled at 44.1 kHz. Use the `resample` function to change the sampling rate of $m(t)$ from f_s to $f_0 = 10$ MHz. Generate a carrier wave $c(t) = \cos(2\pi f_c t)$, where $f_c = 1310$ kHz and the modulated signal $s(t) = m(t)c(t)$ with $A_c = 10$ using MATLAB's element-wise `.*` multiplication. Generate a plot of $s(t)$ (remember that you have to recompute the time axis). Can you see some similarities between it an $m(t)$? Compute the energy of $s(t)$ empirically in MATLAB (remember that the sampling rate for this signal is now f_0).

(c) Now that you've generated your AM radio signal, it's time to simulate the channel. The function `randn(1,n)` generates n independent random Gaussian noise with mean 0 and variance 1. Use this function to generate a noise vector that is the same length as your modulated signal $s(t)$: we'll call this the noise process $w(t)$. Multiplying the vector by a constant α makes the noise variance α^2. For $\alpha = 0.1$, generate noise $\alpha w(t)$ of the same length as $s(t)$ and compute $y(t) = w(t) + s(t)$. This models the signal at the receiver. Generate a plot of $s(t)$ and a plot of $y(t)$. Can you tell the difference? Try it again for $\alpha = 0.5$ and see if it looks different.

(d) The last part of our radio system is the *demodulator*, which is shown below.



The type of AM modulation you simulated is called Double Sideband Suppressed Carrier (DSB-SC[1]). Write a formula for the signal $v(t)$ in terms of the input signal $m(t)$, the noise $w(t)$, and the carrier wave $\cos(2\pi f_c t)$. Use the double-angle formula for the cosine to rewrite the formula in the form

$$v(t) = C_1 m(t) + C_2 m(t) \cos(4\pi f_c t) + C_3 w(t) \cos(2\pi f_c t), \tag{2}$$

where the constants C_1, C_2, and C_3 depend on the carrier gain A_c and noise scaling α. This shows the signal $v(t)$ is the message we want plus "other stuff" that we don't want. Note that this operation is assuming *perfect synchronization* between the transmitter and receiver. Perfect synchronization is impossible: in practical systems you have to also design a synchronization method.

(e) In MATLAB, multiply $y(t)$ by the output of the oscillator to produce $v(t) = y(t) \cos(2\pi f_c t)$. This operation moves the signal from what is called the *passband* (around the carrier frequency) to the *baseband*. Plot the output using $\alpha = 0.1$ (low noise) and $\alpha = 0.5$ (high noise) and compare it to $m(t)$. Compared to your $m(t)$ can you see evidence of the non-message parts of signal?

(f) The last part of the demodulator tries to clean up the signal by removing some of the interference using a *lowpass* filter. You can think of it as a system that "smoothes" the input signal by removing high-frequency artifacts. You can implement a lowpass filter in MATLAB using `lowpass(z,fpass,f0)`. Use the lowpass filter on $v(t)$ with $f_{\text{pass}} = f_s/2$ to produce a smoothed signal. Plot the output of the lowpass filter for the low noise and high noise settings

(g) Use the `resample` function again to shift the sampling rate from f_0 to $f_s = 44.1$ kHz to produce your final signal $\hat{m}(t)$. Plot $\hat{m}(t)$ for the low noise and high noise settings (remember that you have to recompute the time axis). Can you tell the difference? Use `soundsc` to listen both versions of $\hat{m}(t)$. Can you hear the additional noise?

(h) The *signal-to-noise ratio* (SNR) is the ratio of the signal power to the noise power. What parameters of the system have the most effect on the SNR? If you wanted to improve the quality of reception, what parameters can you control as the system designer?

You can go back and try different noise levels (values of α and different f_{pass} to see how these parameters affect the intelligibility of the audio signal or try it with your own audio clips.

Even though you might not listen to much AM radio, amplitude modulation is still used many modern communication systems in the guise of quadrature amplitude modulation (QAM[2]). The Apollo 11 mission didn't use AM but instead used *phase modulation* (PM) in a system called the Unified S-Band system[3]. As you may be able to guess, in phase modulation the signal $m(t)$ is used to modify the *phase* of the carrier wave: $s(t) = A_c \cos(2\pi f_c t + k_p m(t))$. This has the advantage that the amplitude of the output signal is constant. Phase modulation and frequency modulation (FM) are referred to as "angle modulation" schemes and require more complicated demodulators.

[1] If you're a fan of acronyms then you might like communications engineering. It's *full* of acronyms.
[2] More acronyms!
[3] `https://en.wikipedia.org/wiki/Unified_S-band`