

# Lab 6

## Amplitude Modulation (AM) and Demodulation

October 22, 2015

This lab is divided into two parts. In Part I you will learn how to design an amplitude modulator (AM), and in Part II you will be able to demodulate an AM signal.

### 1 Introduction

Modulation is the modification of some aspect of a carrier signal. Given a signal of the form

$$x_c(t) = A(t) \cos[\omega_c t + \phi(t)]$$

where  $\omega_c$  is known as the carrier frequency. If  $A(t)$  varies linearly with the modulating signal (your message to be transmitted), then we have *linear modulation*. If  $\phi(t)$  varies with the modulating signal, then we have angle modulation. Figures 1, 2 and 3, show modulated signals using *amplitude modulation*, *phase modulation*, and *frequency modulation*, respectively. As seen in Figure 1, the message signal is modifying the amplitude of the carrier signal, hence the name amplitude modulation. In Figure 2, the oscillation increases or decreases based on the slope of the modulating signal. In Figure 3, the amplitude of the signal is modifying the rate of oscillation of the carrier signal.

In this lab you will design an AM modulator/demodulator. An AM signal is of the form

$$x_c(t) = [A + m(t)] \cos(\omega_c t)$$

It is important to choose  $A$  so that  $A + m(t) \geq 0$ . You will see why in the lab.

### 2 Prelab Part I: Amplitude Modulation

For this lab you will need to create an amplitude modulator. Use  $f_S = 48\text{kHz}$ . For your message, have the program generate a sinusoidal signal. You should be able to vary the message frequency dynamically as your code is running.

1. Write a MATLAB program to generate an AM signal. Your code should allow you to change the message frequency and  $A$  dynamically as the code is running.
2. Vary  $A$  and record its effect on the modulated signal.

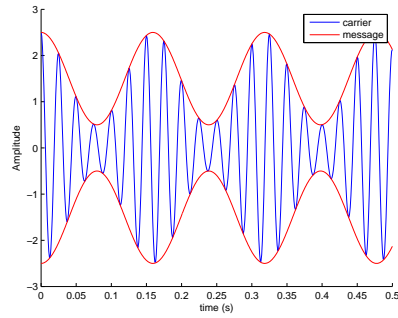


figure 1: Amplitude Modulation

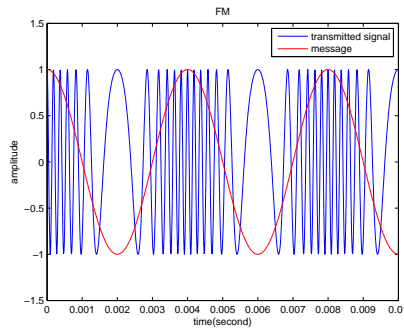


figure 2: Frequency Modulation

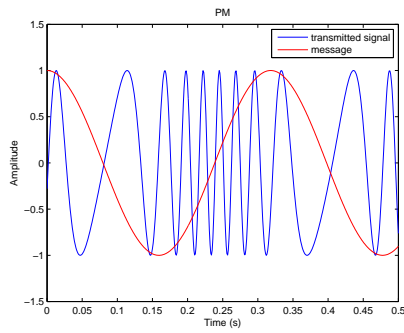


figure 3: Phase Modulation

### 3 Prelab Part II: AM Signal Demodulation

There are several ways to demodulate an AM signal. The one that will be used here is using a *square law detector*. The way this works is as follows: Take the AM signal, square it and then pass it through a lowpass filter.

1. Given

$$x_c(t) = [A + m(t)] \cos(\omega_c t) \quad (1)$$

show that you will be able to recover  $m(t)$  if you square  $x_c(t)$  and then pass it through a lowpass filter.

2. Use Matlab to show the signal provided in Equation 1.
3. Use the FFT function to show the magnitude of the signal provided in Equation 1.
4. Figure 4 shows the magnitude characteristics of a physically realizable filter. On the the figure identify: (a) where is the passband, (b) where is the stopband, and (c) where is the transition band. Also, using the figure explain what is meant by passband ripple and stopband ripple.
5. Use MATLAB to create an elliptic filter with the following specifications:  $R_p = 0.1$  dB and  $R_s = 50$  dB. Plot the magnitude and phase response of the filter. What is the order of the filter?
6. Use MATLAB to express the filter as a cascade of second-order systems. You can use the *tf2sos* function.

### 4 Lab Part I: Amplitude Modulation

- (a) Start a new project.
- (b) Write the code that will generate an AM signal.
- (c) Your code should allow you to change  $A$  and the message frequency dynamically as the code is running.
- (d) Vary  $A$  and record its effect on the modulated signal.

### 5 Lab part II: AM Signal Demodulation

- (a) Start a new project.
- (b) Write your code to demodulate an AM signal.
- (c) You will need another DSP board to generate an AM signal. Run your code from Part I on one board and connect it to the second board that is running your demodulation code.

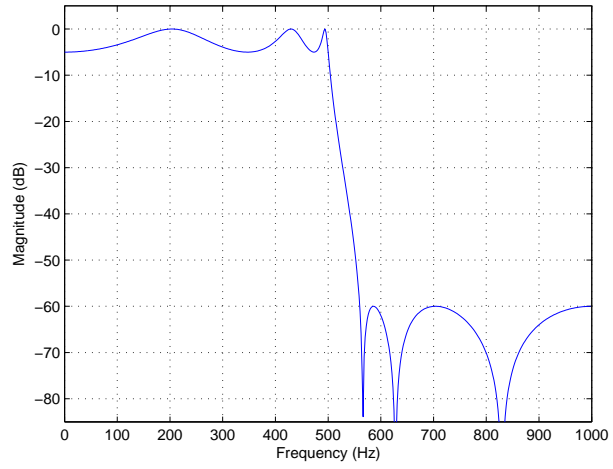


figure 4: Elliptic filter design specifications