EE 4440 – Comm Theory – AM Lab

Purpose:
The purpose of this lab is to investigate Amplitude Modulation (AM) and the reconstruction of AM signals using an envelope detector.
Note: For this lab, no formal lab report is due. Be sure to answer all questions in the lab (questions are in **BOLD** typeface). Please turn in the answers to the questions with your sketches. Typed reports are required for this lab. Be sure to label all sketches and keep the sketches relatively neat and clean.

Equipment:
Only two pieces of equipment are necessary to complete this lab. The first is the Agilent 54622D mixed-signal oscilloscope. A more detailed description of the oscilloscope's general features can be found in lab 1. The second piece of equipment used will be the computer with the LabVIEW software. Using the Data Acquisition Boards (DAQ Boards) of the LabVIEW environment allows the student observe signals generated inside the LabVIEW software and perform further analysis on data acquired from real world signals.

Procedure and Questions:

Notes:
- The sampling rates being used in this lab cause the oscilloscope to behave rather erratically. One method to trigger the oscilloscope successfully is to use the trigger holdoff feature. To activate this, first put the oscilloscope in default mode (see Lab 1 if you don't remember how). Then press the Mode/Coupling button. Use the Entry Knob to adjust the holdoff value until it read 955 μs. This will help keep the display stable throughout the lab.
- Adjust the trigger level as necessary to further stabilize the display
- For the entire lab, the FFT size/span are 100 kHz/50 kHz or 50 kHz/25 kHz. Feel free to use other settings, but be sure you are seeing the "interesting" details and your sketches reflect them. Learn to use good judgment on oscilloscope settings.
- Read the directions thoroughly. Small notes are scattered throughout the lab to help you.
- All LabVIEW modules should be stopped using the STOP button or the ESC key.

Part 1: AM Tone Modulation

i) Open the LabVIEW module labeled amtransmissionwavefile under the Lab 2 folder on the desktop.

ii) Connect the oscilloscope to the DAC0 channel on the DAQ board (Yellow wire). Setup the oscilloscope to display both the time domain signal of Channel 1 and also the FFT of this signal.

iii) Run the LabVIEW module and adjust the parameters if necessary to generate a 5 kHz sine wave with 100% modulation on a 3-volt amplitude 20 kHz sine carrier wave.

iv) Gradually increase the carrier frequency and observe the changes in both the frequency and time domains on the oscilloscope.

v) Gradually increase the message frequency and observe the changes in the time and frequency domains.

vi) Gradually vary the percent modulation between 0 and 120% (0 to 1.20) and observe the changes in the time and frequency domains. You may want to increase the carrier frequency to higher values (like around 60 kHz) for more visible results.

vii) Return the carrier to 3-volts and a frequency to 20 kHz, and adjust the message frequency to 2 kHz, and set the waveform to 50% modulation.

  - **Sketch the signal in both the time and frequency domains. Be sure to label your plots, especially in the frequency domain.**

viii) **Compute the percent modulation from the sketch of the time domain signal.** Note: This means do this accurately from your sketch. Show your calculations!

ix) **From the sketch of the frequency domain signal, compute the percent of the power in the sidebands.**

Part 2: AM modulation with a triangular message signal

Set the LabVIEW module to generate a 2 kHz triangle message signal carried on a 3-volt 20 kHz sine wave carrier.

- **Sketch the time domain and frequency domain plots.**

Part 3: AM modulation with a square carrier

One way to generate an AM signal is to pass the signal [A+m(t)] through an electronic analog switch that is switching at the carrier frequency, between [A+m(t)] and ground. This is equivalent to multiplying [A+m(t)] by a square wave carrier. Remember that a square wave can be decomposed into its fundamental sinusoid and its harmonics. Thus, modulating in this way is equivalent to simultaneously modulating with sinusoids at frequencies of 20 kHz (fundamental), 60 kHz (3rd harmonic), 100 kHz (5th harmonic), and etc. In practice the unwanted frequency terms would be filtered out before transmission.

- Change the message signal back to a sinusoid and to 5 kHz.
- Change the carrier to a square wave of 20 kHz. The displayed spectrum may contain significant noise between the carrier fundamental and the odd harmonics. This may include even harmonics (if the square wave is not perfect) or aliased harmonics.

 Sketch the spectrum (leave out the noise).**

Part 4: Demodulation by Envelope Detection

Design a basic envelope detector using a diode, resistor, and capacitor. Let the capacitor value be 0.01 μF. Choose a resistor value given a carrier frequency of 20 kHz and a tone message frequency of 1 kHz. (Hint: Look at your prelab). **Sketch your envelope detector circuit.**

The second envelope detector used for the lab is a software envelope detector in LabVIEW. The envelope detector is constructed via an absolute value and low pass filter. The reconstructed signal shows up on the DAC1 channel (Orange wire) of the DAQ card.

i) Apply an AM signal (f_c=20 kHz, f_m=1 kHz, μ=0.5) to the envelope detector. Be sure both the carrier and message signals are sinusoidal. Observe the input on channel 1 of the oscilloscope and the output on channel 2. To save room on the display, turn off the FFT function of the oscilloscope for now. If necessary, change your resistor to reduce the amount of ripple if it is excessive (possibly due to non-ideal component values or errors in the design).

ii) **Sketch the output and label it carefully**

iii) **What is the peak amplitude of the AM signal and the demodulated signal? Use the cursors to find these values. Explain the difference in these values.** Note: There may still be some DC offset in your amplitudes. Ignore this offset, calculate the amplitude based on the peak to peak value of the envelope detector output.

iv) Increase the carrier frequency in increments of 10 kHz. **Does this increase or decrease the amplitude of the ripple in the output? Why?**

v) Reset the carrier to 20 kHz and change the message signal to a triangle. **Describe any distortion you see in the demodulated triangle wave? Is it significant?**

vi) Stop the LabVIEW module (***be sure to hit the Stop button in the module, not the stop sign next to the run button***).

vii) In LabVIEW there should be a menu bar along the top of the window. Go to the Window menu above and click Show Diagram.
 In the lower middle section of the diagram, there should be a large label stating 'Put your MATLAB code in the box below'. Below it is a block labeled "MATLAB Script". Replace the code in this block (waveout=wavein;) with the low pass filter you designed as part of the prelab. **Be sure to include your MATLAB code in the write up.**

viii) Click back onto the panel window (original window). Ensure the Sampling Frequency is set to 500000.
 Change the Sampling Size to be 100000.
 Run the LabVIEW module again.

ix) Connect the oscilloscope so the output of the hardware envelope detector is shown on channel 1 and the software envelope detector is shown on channel 2.
 Reset the signal to the 20 kHz sinusoid with 1 kHz sinusoidal message

signal and a μ-value of 0.50.

 Note: DAC0 should still be connected to the input of the hardware envelope detector. DAC1 should **<u>ONLY</u>** be connected to the oscilloscope, not the output.

x) Vary both the carrier frequency and message frequency while observing the results of both the hardware and software envelope detectors.

xi) Return to a carrier of 20 kHz and a message frequency of 1 kHz. Change the message signal to a triangle wave once more.

 Which envelope detector does a better job at accurately reproducing the triangle wave better? Explain.

xii) Change the message signal to "Wave File" and observe the modulated .wav file. If you like, change the wave file to a different one and observe the changes. ONLY USE .WAV FILES IN THE C:\WINDOWS\MEDIA folder. Other .wav files may have different sampling rates which will cause problems in the software envelope detector. Note: Since wave files are relatively long lasting, you may need to adjust the oscilloscope horizontal time base to 10 ms/div in order to see useful results.

xiii) When you have finished experimenting with the different waveforms and their settings, stop the LabVIEW module once more (remember, press the STOP button or press ESC, **<u>DO NOT PUTSH THE STOP SIGN</u>**). When the module has stopped (it may take a few seconds), click onto the MATLAB Command Window open.

xiv) Type in 'soundsc(waveform,22050)' and press return. The reconstructed signal of the software envelope detector will then be played. Next, type 'soundsc(original,22050)' and press enter to play the original sound. Compare the two sound files. **What do you notice about the reconstructed waveform and how may it be restrictive in AM's use?**

Part 5: Amplitude modulation of a pulse train

 Change the Sample Length back to 50000.

 Return to the previous set up with channel 1 of the oscilloscope on the hardware envelope detector input and channel 2 on the output. The software envelope detector output is no longer needed, so do not worry about it.

 Restart the LabVIEW module. Change the module to generate a 20 kHz sinusoidal carrier wave modulating a 1 kHz square wave with a duty cycle of 25%.

i) **Sketch <u>one</u> of the demodulated pulses (from the hardware envelope detector output) and explain the long delay at the end of the pulse.**

ii) **Sketch the frequency spectrum of the AM signal (from the hardware envelope detector input).**

iii) **Explain the observed spectrum. For example, if harmonics are missing, point this out and explain. What is expected compared to what you see?**

Part 6: Single Side Band modulation of a signal

Single Side Band-With Transmitted Carrier (SSB-TC or SSB+C) is a method of transmitting signals without using as much bandwidth. Two methods to generate an SSB signal are the phasing method or the filtering method. The LabVIEW module we will use next generates SSB using the phasing method and the Hilbert Transform (see page 198-203 of the book for more information).

i) Stop and close down the open LabVIEW module (click No to any save changes dialogs). Open the second module in the Lab 2 folder, ssbhilbert.

ii) Run the LabVIEW module. Be sure the oscilloscope is connected to DAC0 on the DAQ card.

iii) Observe the frequency spectra of the SSB signal. Try toggling the Sideband switch. Observe the changes in the frequency domain.

- **Sketch the frequency spectrum of the SSB USB (upper side-band) signal.**

iv) Now determine if an envelope detector will successfully reconstruct a SSB-WC signal.

- Connect the DAC0 channel to the envelope detector if it is still not connected.
- Connect the channel 2 probe to the output of the envelope detector.
- **Briefly describe the result and explain whether the envelope detector worked or not.**

EE 4440 – Comm Theory – Prelab for Lab AM

The following two questions should be answered before coming to lab. Both will be required for completing the lab.

1) Design the simple envelope detector you will use in the lab using a diode, resistor, and capacitor. The capacitor value is specified at 0.01 μF. Read part 4 of the lab for further design specifications

2) Write MATLAB code for a simple low pass filter (a simple averaging filter will work. See hint below.) using the input variable of wavein and the output variable of waveout. The averaging filter should average about 25 terms while if a true bandpass is designed, be sure to keep the cut-off frequency high enough to keep any signal content of interest, but filter out the carrier frequency. Write the MATLAB code to use the 'filter' command in MATLAB. If you are unsure on the syntax, see the MATLAB help files or consult with the TAs or instructor. A small side note on the filters, avoid complicated filter designs to reduce real-time requirements since both MATLAB and LabVIEW are running.
Hint: Recall from EE3220 that a simple 3-th order "moving-average" filter is given by y[n] = {x[n] + x[n-1] + x[n-2] + x[n-3]}/4. This is a low-pass filter with a sinc-like frequency response. This could be implemented in matlab as: b=[1, 1, 1, 1]/4; y=filter(b,1,x);