University of Washington ECE Department
EE 235 Lab 6 – Modulation & Filtering

In this lab, we will investigate two important Fourier Transform applications: filtering and modulation. We will apply both of these to a real-world problem of creating and decoding a message using Morse Code. This will be a two-week lab.

Lab 6 Turn-in Checklist

- 5 pre-lab exercises
- 4 lab assignments with 4 TA check-offs
- Lab report, submitted as a Jupyter notebook following the format provided in earlier labs

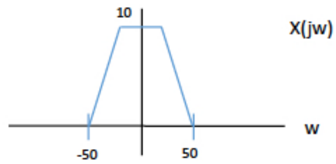
Note: All assignments except the prelab should be completed in groups of 2-3 people.
The pre-lab exercises are to be completed individually and submitted via Canvas before lab sections.

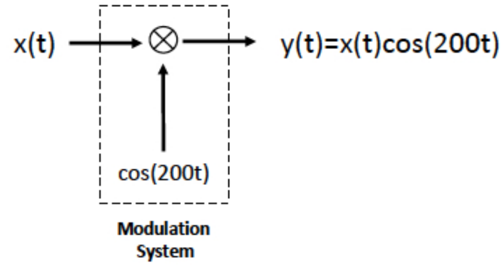
In the first week of the lab, you should do Pre-lab parts 1-3, and assignments 1-3.

Pre-lab (Part I)

Read the Lab 6 Background document, then complete the following exercises. Only the first three questions of the pre-lab are due in week 1, but you will get further with the lab if you do more than that.

1. ***Filtering***.
   a. Suppose a real-valued LTI filter has the frequency response: $H(jw) = \frac{\alpha}{\alpha + jw}$
      i. Find the magnitude response **|H(jw)|**. Draw a rough sketch of **|H(jw)| vs. w** and label the axes using $\alpha = 200$.
      ii. Based on your sketch in (ii), classify the filter type.
      iii. What should the filter coefficients **b** and **a** be for the function ***signal.lsim***?
   b. Consider the input signal **x1(t) = cos(100t).** Is this input in the pass band of the filter? Find the response **y1(t)** of the filter to this input.
   c. If you were to implement x1(t) on the computer with fs=4000 Hz for 500ms, how long would the time vector be?
2. ***Modulation***, in general, is an important part of all communication systems. In short, modulation is a process that transforms a low frequency signal into a signal of higher frequency. This is useful because low frequency signals cannot travel long distance over the air, but higher frequency signals can.
   a. Consider the modulation system below, with input **X(jw)** provided on the left:

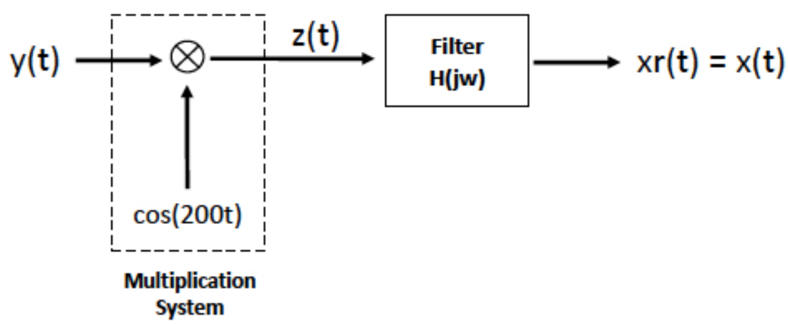




Using Fourier Transform properties, find the expression for **Y(jw)** in terms of **X(jw)**. Then, using the plot for **X(jw)** above, sketch **Y(jw) vs. w**. What is the carrier frequency for this system?

b. For the signal X(jw) above, what is the smallest carrier frequency that you could use without causing the shifted signals to overlap in frequency?

3. ***Demodulation*** is used to undo the modulation operation. It is performed using a two-stage process, as shown below.



The first stage requires another multiplication operation using the same sinusoidal signal that was used for modulation. The second stage involves a low pass filter to recover the signal **x(t)**. The purpose of these two stages is to recover the original **X(jw)**, which is centered on **w = 0**.

a. Suppose **y(t) = x(t)cos(200t)**, where **X(jw)** and **Y(jw)** are the same as in part 2. Using Fourier Transform properties, find an expression for **Z(jw)** in terms of **Y(jw)** and sketch **Z(jw) vs. w**. You should notice that your plot for **Z(jw)** contains two parts: the original signal **X(jw)** centered around **w = 0** with amplitude scaled by ½, and other copies of **X(jw)** centered at higher frequencies

b. The low-pass filter we will use to recover **X(jw)** has the following LCCDE relating input **z(t)** to output $\mathbf{x_r(t)}$:

$$(240)\frac{d^4x_r(t)}{dt^4} + (3\times10^4)\frac{d^3x_r(t)}{dt^3} + (2.2\times10^6)\frac{d^2x_r(t)}{dt^2} + (10^8)\frac{dx_r(t)}{dt} + (2\times10^9)x_r(t) = (2\times10^9)z(t)$$

i. Find the filter's frequency response$H(jw) = \frac{Xr(jw)}{Z(jw)}$.

ii. What should the filter coefficients **b** and **a** be for the function **lsim**?

iii. What is the DC gain |H(j0)| of this filter?

4. The **Morse Code** was an early means of communicating information digitally. We use it for communicating messages in this lab.

a. The Morse Code, which uses two symbols, a dot and a dash, so we will need two signals to represent them. As we did in lab 4, we will choose one signal to be the reverse of the other.

$$x_{dash}(t) = 50te^{-15t}u(t) \quad x_{dot}(t) = -x_{dash}(t)$$

What should *A* be if you want the area of the signal to be 1? Using a definition of bandwidth as the frequency where the magnitude is reduced by a factor of $\sqrt{2}$, find the bandwidth of these signals. Will this bandwidth work with a carrier frequency of w_c=200? If not, how would you change the exponent?

b. Given the International Morse Code table below:

A	· -	H	· · · ·	O	- - -	V	· · · -
B	- · · ·	I	· ·	P	· - - ·	W	· - -
C	- · - ·	J	· - - -	Q	- - · -	X	- · · -
D	- · ·	K	- · -	R	· - ·	Y	- · - -
E	·	L	· - · ·	S	· · ·	Z	- - · ·
F	· · - ·	M	- -	T	-		
G	- - ·	N	- ·	U	· · -		

Letting **dah** be the array for $x_{dash}(t)$ and **dit** be the array for $x_{dot}(t)$, give the Python code that you would use to create a time signal **xm** that communicates the letter 'X' as a sequence of dots and dashes.

c. To decode a message with a sequence of dots and dashes, we need to compare the received signal in each time slot to the two possible signals and find the closest match. One way to find the closest match is to measure the sum of squared errors at each time sample. Write code that computes this quantity for a given signal using the numpy.sum function.

5. **Frequency-division multiplexing** is a method for sending multiple messages together at the same time. Assuming that each message is limited to a low frequency bandwidth, the different messages are modulated to different, non-overlapping positions in frequency.

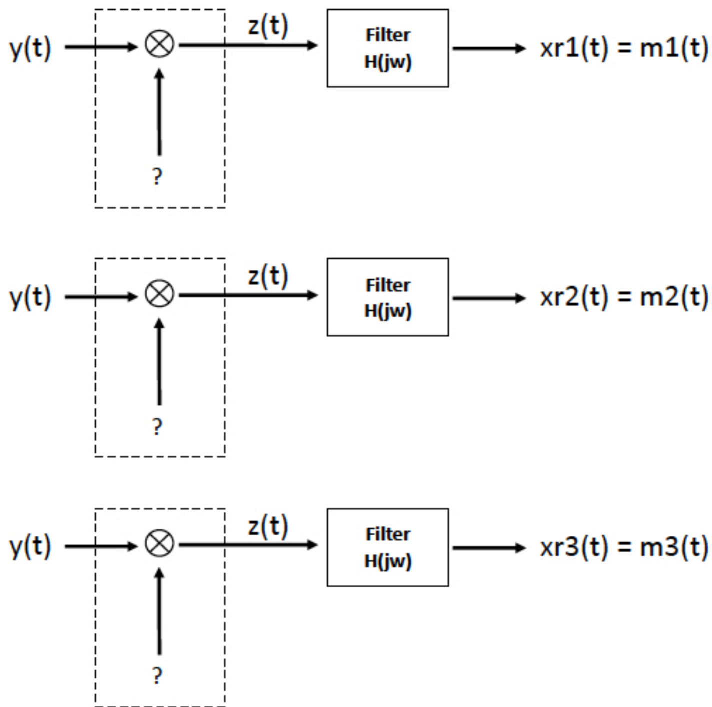
The individual messages can be recovered by demodulating using a carrier matched to the appropriate position in frequency.

Consider a signal **y(t)** of the form

$$y(t) = m1(t)cos(100t) + m2(t)cos(200t) + m3(t)cos(400t)$$

where **m1(t)**, **m2(t)**, and **m3(t)** each correspond to a single letter of the alphabet which has been encoded using International Morse Code. This signal consists of a sum of three different modulated signals.

In the last exercise, you learned how to recover a single signal **x(t)** from **y(t) = x(t)cos(200t)**. That same process can actually be used to individually recover **m1(t), m2(t)**, and **m3(t)** using three separate systems, as shown below:



a. What three carrier signals are needed for each system shown? In other words, what frequency cosine will give **m1(t)**? With **m2(t)**? With **m3(t)**?

b. The signals that you transmit for any given letter can have different lengths (some letters have one symbol but others have up to four symbols), but you can't determine the length until after you demodulate. If a letter has less than four symbols, then the ending slots of the original signal will be zero valued, but after recovery there may some non-zero (but low energy) values due to non-ideal filtering. Specify a test for determining when the message ends, i.e. when a slot has zero value.

Lab Assignments

This lab has 4 assignments. Each should be given a separate code cell in your Notebook, and each should be associated with a markdown cell with subtitle and discussion. As in previous labs, your notebook should start with a markdown title and overview cell, which should be followed by an import cell that has the import statements for all assignments. In this lab, you will need to use most of the import commands that you have used in earlier labs, plus import signal from scipy. As always, you should add comments to your code for clarity.

You will again be working with concepts from previous labs, so you may want to refer back to the background files for those labs or the py_ref document.

Assignment 1: Filtering

In this Assignment, we'll implement and analyze a continuous-time filter using the Python scipy signal package. Write your code in a new cell for Assignment 1, inside your Lab 6 notebook.

A. Using fs=4000 Hz, create a time vector **t** for the range [0,500ms). Use **t** to create the signal **x1** corresponding to $x_1(t)$ from part 1 of the prelab.

B. Define a filter for a first-order system using the parameters you found in the pre-lab, part 1. Using the ***signal.freqresp*** function, find the frequency response of the system and plot the magnitude |H(jw)| and phase in side-by-side (1x2) plots. For the magnitude, create the plot with overlayed version of standard magnitude and magnitude with a dB scale $(20\log_{10}|H(j\omega)|$, specifically with the left axis corresponding to the linear amplitude and the right to dB. Label the frequency axis using frequency in rad/s.

C. Using the ***signal.lsim*** function, find the output **y1** that is the response of the system to **x1**. Create a new plot, and plot **x1** and **y1** together on the same plot. Confirm that the resulting signal matches what you expect from the pre-lab in terms of changes in amplitude and phase (associated with using a non-ideal filter).

Assignment Check-Off #1 of 4: Demonstrate this Assignment to the lab TA .

Report discussion:
If you put two identical ideal low-pass filters in series, then the overall response will be unchanged. Comment on what will happen if you put two identical versions of the above filter in series. Do you expect the amplitude of the response to x1 to be greater or smaller than for a single filter? (Note that you can test this by letting the output y1 be an input to the same filter.)

Assignment 2: Amplitude Modulation

Modulation, an important part of all communication systems, is a process which transforms a signal of low frequency to a signal of higher frequency. This is useful because low frequency signals cannot travel longer distances over the air, but high frequency signals can. In this exercise, we implement and analyze amplitude modulation in software.

Use a new cell for Assignment 2. Comment your code and title/label your plots.

A. First create two signals to communicate.

   a. Using sampling frequency of **fs**=4000 Hz, create time signals **x1** and **x0** corresponding to

   $$x_1(t) = 50te^{-15t}u(t) \qquad x_0(t) = -x_1(t)$$

   using a time vector defined over [0,0.5).

   b. Using 2x1 subplots, plot the time signal for **x1** and the magnitude of its Fourier transform. For the Fourier transform, use **Nfft = 8192** samples, and in the plot use rad/s and limit the frequency axis to the range [-100,100).

B. Next create and modulate the message signal.

   a. Construct a message signal **xm** for communicating the bit sequence [1 0 1 0] by concatenating **x1** and **x0**.

   b. Create the carrier signal: **c(t)=cos(200t)**.

   c. Finally, implement the modulated signal: **y(t) = xm(t)c(t)**

C. Using 2x1 subplots, plot the time signals **xm** and **y**.

D. Using 3x1 subplots, plot the magnitude of the Fourier transforms for **xm, c** and **y**. Again use **Nfft = 8192** samples, and a frequency axis of rad/s in the plot. For **xm**, limit the frequency axis to the range [-150,150), but for the other two plots you should use the range [-500,500). Verify that the Fourier transforms correctly put the carrier frequency at w_C = 200 rad/s.

Assignment Check-Off #2 of 4: Demonstrate this assignment to the lab TA

Report discussion:
In this Assignment, we analyzed a system with input-output relationship **y(t) = x(t)cos(200t)**. A student in class claims this system is LTI. Explain why they are incorrect.

Assignment 3: Amplitude Demodulation

Now we'll learn to use (ideal) demodulation, to undo the modulation operation performed in **2.0**. In a communication system, once a signal is transmitted to the intended receiver, the receiver will use demodulation to that transformation and recover the original signal, as in part 3 of the pre-lab. The first stage is a multiplication operation using the same carrier that was used for modulation. The second stage is a filter used to fully recover the signal **x(t)**. The purpose of these two stages is to recover the original **X(jw)**, which is centered on **w = 0**.

Use a new cell for Assignment 3. Comment your code and title/label your plots.

A. Using the carrier from assignment 2, create the signal **z(t)=y(t)c(t).** Using 2x1 subplots, plot the time signal **z** and the magnitude of its Fourier transform. For the Fourier transform, use **Nfft = 8192** samples, and in the plot use rad/s and limit the frequency axis to the range [-1000,1000).

B. Using the scipy signal package, create a continuous-time filter using the coefficients that you found in part 3 of the pre-lab. Find the frequency response of the filter using the ***signal.freqresp*** function. Using a single plot with parallel axes, plot the magnitude (dB scale) and phase of the system frequency response, with magnitude on the left axis and phase on the right axis. Use frequency range [-300,300) in rad/s.

C. Use the ***signal.lsim*** function to filter **z(t)** with the filter you designed, giving the recovered signal **xr(t)**. Using 2x1 subplots, plot the time signal **xr** and the magnitude of its Fourier transform. For the Fourier transform, use **Nfft = 8192** samples, and in the plot use rad/s and limit the frequency axis to the range [-150,150) rad/s. Because the filter is not ideal, you should notice some differences in the recovered and original signals.

Assignment Check-Off #3 of 4: Demonstrate this assignment to the lab TA

Report discussion:
A student decides to use a shorter signal of the form $x_1(t) = Ate^{-150t}u(t)$, thinking that the message can be communicated in a tenth of the time. They claim to see the exact same graph for xr(t) as they saw in their pre-lab. Explain why the student is incorrect, and why this solution will not work.

Assignment 4: Decoding a Morse Code Message

We will decode a Morse Code signal of a three letter word, using what we learned in the previous three exercises. This Assignment is a modification from an exercise in *Computer Explorations in Signals and Systems using Matlab* by John R. Buck, Michael M. Daniel, and Andrew C. Singer.

You are given a signal **y(t)** which contains a simple message from Agent 007. When loading the file, you magically transform into Agent 008, the code-breaking sleuth. The last words of the aging Agent 007 were "The future of technology lies in..." at which point Agent 007 saved the remaining message to a csv file and decided to return to school and give up the next four years to pursue the American dream of getting a PhD. Your job now is to decipher the message encoded in **y(t)** and decode Agent 007's final words. The signal was obtained by adding (not concatenating) three 2 second message signals each modulated with carrier frequencies of 100, 200 and 400 rad/sample, and digitized with a sampling frequency of 4kHz.

A. Using a new, separate cell, write a function to decode the sequence of dots and dashes in a received message of fixed length 2 s with 0.5 s time slots for each symbol. Be mindful of the fact that not all letters have the same number of Morse code characters. If there are fewer than 4 symbols, the remaining length is padded with zeros, so you will have to have a check to see if the time-slice in question is empty or a Morse code signal. You should use your work from pre-lab parts 4 and 5. Test your function by creating the signal for ‘X’, and decoding the original and a modulated-demodulated recovered version.

B. Start another new cell. Load the data file **‘message_modulated.csv’** which is available from the course web site. The file contains the signal y(t) with three messages combined using frequency division multiplexing.

C. Write a loop that:

   a. Assigns the demodulating frequencies to extract the respective message signal from the composite signal, using your work from the pre-lab part 5 and the knowledge that the carrier signals are y1(t)=cos(100t), y2(t)=cos(200t), y3(t)=cos(400t);

   b. Demodulates the signal and then filters it as was done in the previous parts and uses the function from part (A) to recover the $\mathbf{i^{th}}$ recovered message and output **str**; and

   c. Prints out the “Message m**i**(t) is: **str**” with the values filled in for **i** and **str**

Assignment Check-Off #4 of 4: Demonstrate this assignment to the lab TA

Report discussion:
In this lab, we matched the received and candidate signals using a minimum squared error criterion. Explain why the matched filter from lab 4 might be more effective.
Use the Morse Code table provided in pre-lab part (4) to decode the letter from each signal plot.
Complete Agent 007’s final words: **“The future of technology lies in ________.”**

Team Report

When you've tested and cleaned up all your code (remember, you should only submit code for the Assignments, each in their own cell), go to 'File' then 'Download as', then select '.ipynb'. The file you download is a Notebook that your TA will be able to open and grade for you, once you submit it on Canvas. Remember, only one notebook per team! Make sure that your notebook is titled Lab6-XYZ.ipynb, where XYZ are the initials of the lab partners. You may want to also download the file as pdf to have a nicer documentation of your records.

Submit the .ipynb file via Canvas.

Closing Note

This quarter you have decoded innumerable messages, and learned how to perform convolution, convert signals from the time domain to the frequency domain, and filter signals. What an accomplishment! Unfortunately, all good things must come to an end. It was a pleasure working with you, Agent 008, but it is now time for a new group of EE students to follow in your footsteps and take on the EE235 challenge. Good luck and keep in touch!