University of Washington
Department of Electrical Engineering
EE 235 Lab 6:
Applications of Fourier Transforms

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 encoded using Morse Code. This will be a two-week lab.

Important Concepts From Lecture You Will Use In Lab 6

- Classifying the filter type using the magnitude response of LTI systems
- Using the convolution property of the Fourier Transforms
- Using the modulation property of the Fourier Transforms

What Is Expected From You In Lab 6

- Completion of 4 pre-lab exercises (3 points)
- Completion of 2 in-lab check offs with TA (4 points)
- Completion of a lab report (3 points)

Note: all pre-lab exercises must be completed by each student individually before walking in to the lab. Each student will be checked off by the TA at the beginning of lab section.

PRE-LAB:

Read the Matlab Lab 6 Background, particularly the sections on the new built-in Matlab functions that you will need in this lab. This will be a 2-week lab. 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) Multiplication and division: Suppose you have the following Matlab variables

   $x1 = 2$ and $x2 = [1 \;\; 3]$

   Which of the following operations are valid in Matlab?
   i) x1 * x1, x1 .* x1, or both?
   ii) x2 * (x1 + x2), x2 .* (x1 + x2), or both?
   iii) x1 * (1 + x2), x1 .* (1 + x2), or both?
   iv) x1 / (2 * x1), x1 ./ (2 * x1), or both?
   v) x1 / (1 + x2), x1 ./ (1 + x2), or both?

2) ***Filtering***:
   a) Suppose a real-valued LTI filter has the frequency response: $\boldsymbol{H(jw) = \frac{3}{3+jw}}$
      i) Find the magnitude response **|H(jw)|**.
      ii) Draw a rough sketch of **|H(jw)| vs. w**.
      iii) Based on your sketch in (ii), classify the filter type.

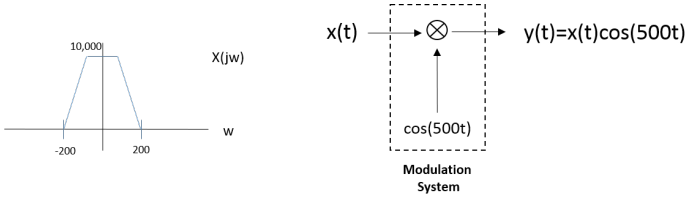
   b) Consider the frequency response used in (b), what should the filter coefficients **b** and **a** be for the function **lsim**?

   c) Consider the input signal **x(t) = cos(0.1t).**
      i) Do you expect this filter to pass (gain close to 1) or reject (gain << 1) this signal?
      ii) Find and sketch the Fourier transform **X(jw).**

d) Using **X(jw)** and **H(jw)** from parts (a) and (b), as well as the convolution property, find the output Fourier Transform **Y(jw)= X(jw)H(jw).** Hint: you will need to use the sampling property for unit impulses to get the final answer.

3) ***Modulation***, in general, is an important part of all communication systems. In short, modulation is a process that will transform a signal of low frequency to a signal of higher frequency. Transforming a signal to a higher frequency signal is important because low frequency signals cannot travel long distance over the air, but higher frequency signals can.

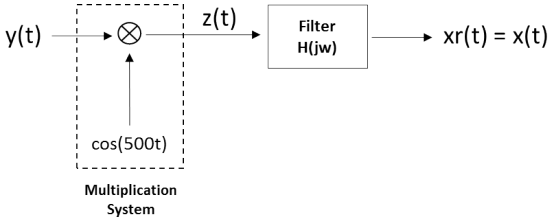
a) Consider the modulation system below, with input **X(jw)** provided on the left:



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) In Matlab, suppose you are given the row vector **x** for the input signal and the row vector **t** for the time samples. How would you implement the multiplication between **x(t)** and **cos(500t)** for **y(t)**? Would you use matrix multiplication * or elementwise array multiplication .* ?

4) ***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(500t)**, where **X(jw)** and **Y(jw)** are the same as in Pre-lab Exercise #3. Using your results from Exercise #3 and Fourier Transform properties, find an expression for **Z(jw)** in terms of **Y(jw)**. Using your sketch for **Y(jw)** in Exercise #3, sketch **Z(jw) vs. w**. You should notice that your plot for **Z(jw)** contains two parts:

- Original signal **X(jw)** centered around **w = 0** with amplitude scaled by 1/2.
- 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 **xr(t)**:

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

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

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?

5) Consider 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	· · -		

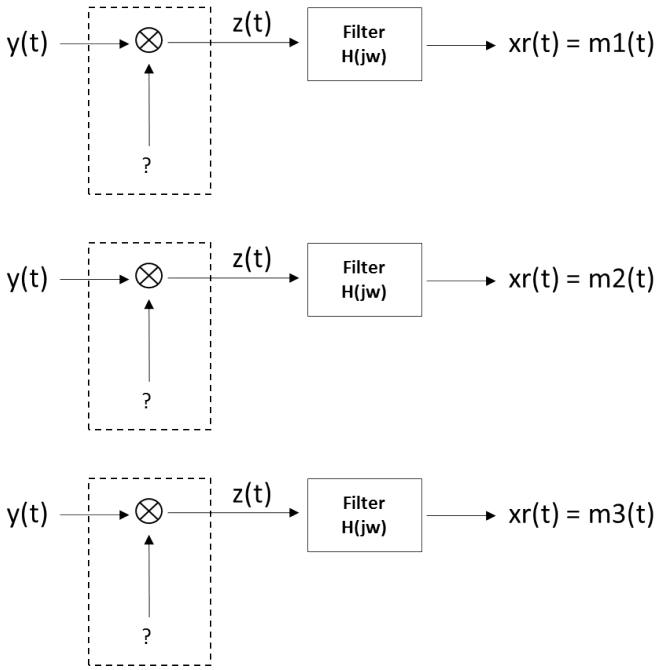
a) Suppose in Matlab you are given the prototype signals **dash** and **dot**. Using these variables, how would you construct the signal **x** for letter **X**? That is, fill in the blanks:

x = [_____ _____ _____ _____]

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

$$\boldsymbol{y(t) = m1(t)\cos(1000t) + m2(t)\cos(2000t) + m3(t)\cos(3000t)}$$

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. Note that this signal consists of a sum of three different modulated signals. This communication systems technique is called frequency division multiplexing. In the last exercise, you learned how to recover a single signal **x(t)** from **y(t) = x(t)cos(500t)**. That same process can actually be used to individually recover **m1(t)**, **m2(t)**, and **m3(t)** using three separate systems, as shown below:



What are the three cosine signals you need for each system above?

c) To pick the correct cosine signal in Matlab, we will make use of a decision statement inside of a loop with index variable **i**. Write a decision statement for the scenario below. Assume you are already given the time samples vector **t**:
   - If index **i** is 1, then assign your first cosine signal to the variable **c**
   - If index **i** is 2, then assign your second cosine signal to the variable **c**
   - If index **i** is 3, then assign your third cosine signal to the variable

d) Suppose the variable **i = 2**. Using this variable **i**, vector concatenation, and the function **num2str**, write the Matlab code to construct the variable **str** with value "Message m2(t)".

EXERCISE #1: Filtering

In this exercise, we will learn two ways to implement a filter in Matlab. We will use these techniques in the last two exercises.

Lab Exercise:

a) Create a new working directory in your U Drive called **Lab 6**. In this directory, open up a script and call it **Ex1.m**. Make sure to clear all variables and close all figures.

b) We will first generate the input signal in the time domain, and plot it with its Fourier Transform (magnitude only):

   i. Generate signal $\boldsymbol{x(t) = \cos(0.1t)}$ from $0 \le t < 500$ using a sampling rate **Fs** = 10. Store the time samples in **t** and the signal in a variable called **x**.

ii. Compute the the FFT of **x** using **N** = 8192 frequency samples. Using **fftshift,** store this result in variable **x_fft**. Using the **abs** function, compute the magnitude of the FFT and store it in **xw_abs**.
iii. Compute the corresponding vector of frequency samples **w** using **N** and **Fs**.
iv. Load figure window #1. Using a 2 x 2 subplot window, plot:
    - 1st subplot: **x vs. t** with x-axis limits [100, 400] and y-axis limits [-2, 2]
    - 2nd subplot: **xw_abs vs. w** with x_axis limits [-0.5, 0.5] and y-axis unchanged

    We will plot two more signals later in this exercise.
v. Run your script and verify your FFT plot is roughly consistent with your equation for **X(jw)** in pre-lab. Why won't it be identical to the pre-lab result?

b) Let us now compute the frequency response of the filter:

i. Using the same frequency sample vector **w** that was used for **x_abs**, generate the frequency response of the filter $\boldsymbol{H(jw) = \frac{3}{3+jw}}$. Make sure to use the correct division operator. Store the values in the variable **H_fft**. In Matlab, you can simply use "j" or "1i" for j, the imaginary number, but make sure you don't use "j" for other variable names.
ii. Using the **abs** function, compute the magnitude of the FFT and store it in **Hw_abs**.
iii. Load figure window #2, and plot **Hw_abs vs. w**. Adjust the x-axis to be between -25 and 25, and then leave the y-axis unchanged. Title and label your plot.
iv. Run your script and verify your plot of **Hw_abs** matches with your pre-lab.

c) Let us now compute the output signal **y(t)** using frequency domain methods first.

i. Recall the convolution property: Y(jw) = X(jw)H(jw). Using **x_fft** and **H_fft**, compute the output Fourier Transform **y_fft**. Make sure to use the correct multiplication operator.
ii. We can now convert the signal **y_fft** back to the time domain with an inverse Fourier Transform. Using the functions **ifft**, **fftshift**, and **real**, compute the time domain signal **y**.
iii. Compute the corresponding time samples vector **t_y** for the signal **y**.
iv. Go back and load (or access) figure window #1. Plot **y vs. t_y** as the 3rd subplot. Adjust the limits of the x-axis to be between 100 and 400, and the y-axis to be between -2 and 2. Title and label your plot.
v. Run your script and verify your plot for **y** is consistent with your answer to the question in pre-lab as to whether the filter passes or rejects the input signal **x(t)**.

d) Let us now compute the output signal **y(t)** using a special function in Matlab called **lsim**:

i. Using your pre-lab, define the filter coefficients **b** and **a** for the filter **H(jw)**.
ii. Using the **lsim** function, pass the input signal **x** (along with its time samples **t**) through the filter defined by coefficients **b** and **a**. Call the output of the filter **y**.
iii. Access figure #1 again and plot **y vs. t** as the 4th subplot. Use the same limits as the 3rd subplot. Title and label your plots.

e) Comment your code, and then run your script to view output plot **y**. Verify it matches the plot from (d).

Lab Check-Off #1 of 2: Show your plots to a Lab TA and demonstrate you know how to implement a filter and compute its output in Matlab.

Lab Report Question #1: A rushed student forgets to call the **figure** function with 1 argument, and instead types in the **figure** command with no function argument. Describe what would change if you ran **Ex1.m** again. Does Matlab produce the same number of figure windows? Do any of the figures change? If so, how do they change?

EXERCISE #2: Amplitude Modulation

In this exercise, we will shift the focus to another application of Fourier Transforms: communication systems and modulation. We will use this exercise to implement and analyze amplitude modulation in Matlab. The signals we will experiment with will be our representation of dash and dot signals used in International Morse Code.

Lab Exercise:

a) Open up a script file and call it **Ex2.m**. Clear all variables and close all figures.

b) From the class website, download and load the Matlab data file **MorseCode.mat**. This file will contain the following signals:
   - **Fs**: sampling rate of signals
   - **dash**: first prototype signal for International Morse Code
   - **dot**: second prototype signal for International Morse Code

c) Create the time samples vector **t_morse** using sampling rate **Fs** and the length of either **dash** or **dot**. Both signals are the same length, so using either prototype signals will work.

d) To view the two Morse code signals we will be using throughout Lab 6, plot the two prototype signals using a 2 x 1 subplot as follows:
   - **dash vs. t_morse**: x-axis unchanged, but y-axis range = [-40, 40]
   - **dot vs. t_morse**: x-axis unchanged, but y-axis range = [-40, 40]

e) Now let us use Matlab to construct one letter in the International Morse Code table:

   i) Using results from pre-lab, use vector concatenation of the signals **dash** and **dot** to construct the Morse Code signal for the letter X. Call this variable **x**.

   ii) Create the corresponding time samples vector **t** using **x** and **Fs**.

   iii) Using the functions **fft** and **fftshift**, compute the magnitude of the Fourier Transform of signal **x** using **N** = 8192 frequency samples. Call this variable **x_fft**. Using the function abs, compute the magnitude of **x_fft** and call this variable **x_abs**.

iv) Create the vector of frequency samples **w** using the number of frequency samples **N** and the sampling rate **Fs**.

v) Using a 2 x 1 subplot, plot the signal **x** in the time and frequency domain as follows:
- **x vs. t**: leave both x-axis and y-axis unchanged
- **x_abs vs. w**: x-axis range = [-1000, 1000] and y-axis unchanged

vi) Run your script and verify your plot for **x_abs**. The envelope should be similar to the envelope of the signal used in pre-lab.

f) Now we will implement amplitude modulation in Matlab:

i) Generate the signal **y(t) = x(t)cos(500t)** using the signal **x** and the same time samples vector **t** used by signal **x**. Call the resulting signal **y**. Note that both **x** and **cos(500t)** are row vectors, so make sure to use the correct type of multiplication. **Refer to Background Section (4), if needed**.

ii) Using the functions **fft**, **fftshift**, and **abs**, compute the magnitude of the Fourier Transform of signal **y** using **N** = 8192 frequency samples. Call this variable **y_abs**. Note: you do not need to compute a separate frequency samples vector since **N** and **Fs** are the same as the input signal.

iii) Using a 2 x 1 subplot, plot the signal **y** in the time and frequency domain as follows:
- **y vs. t**: no need to adjust axes
- **y_abs vs. w**: x-axis range = [-1000, 1000] and y-axis range = [0, 10000]

iv) Run your script and verify your plot for **y_abs** and the location of the carrier frequencies $\mathbf{w_c}$.

Lab Check-Off #2 of 2: Show your script **Ex2.m** to a Lab TA and demonstrate you understand how to modulate a signal.

Lab Report Question #2: A student accidentally uses the modulation signal **cos(50t)** in Matlab instead of **cos(500t)**. The student claims s/he sees the exact same graph for **Y(jw)** with a modulation frequency $\mathbf{w_M} = 50$ as his/her pre-lab. Explain why this student is incorrect. Also, explain why using any carrier frequencies where $\mathbf{w_M} < 200$ will not work.

Lab Report Question #3: When graphing the three plots in this exercise, the student uses the line of code **figure(1)** at all times instead of using the command **figure**. What changes do you expect when you run **Ex2.m** again?

EXERCISE #3: Amplitude Demodulation

In this exercise, we will learn how to undo the modulation operation in Exercise #2. This process in communication systems is called demodulation. Once a signal is transmitted to the intended receiver, demodulation helps to undo that transformation and recover the original signal for the receiver.

Lab Exercise:

a) Open up a script file and call it **Ex3.m**. Clear all variables and close all figures.

b) From the class website, download and then load the data file **Ex3.mat**. It will consist of:
   - **Fs**: sampling rate of signal y
   - **t**: vector of time samples for signal y
   - **y**: modulated signal from Exercise #2

c) We will begin by performing and analyzing the first step in the demodulation process:

   i) Generate the output of the multiplier, **z(t) = y(t)cos(500t)**, from the system diagram in pre-lab, and call the signal **z**.

   ii) Using the functions **fft**, **fftshift**, and **abs**, compute the magnitude of the Fourier Transform of signal **z** using **N** = 8192 frequency samples. Call this variable **z_abs**.

   iii) Create the vector of frequency samples **w** using the number of frequency samples **N** and the sampling rate **Fs**.

   iv) Using a 2 x 1 subplot, plot the signal **z** in the time and frequency domain as follows:
      - **z vs. t**: leave x-axis and y-axis unchanged
      - **z_abs vs. w**: x-axis range = [-1500, 1500] and y-axis range = [0, 10000]

   v) Run your script and verify your plot of **z_abs** matches **Z(jw)** from pre-lab.

d) We will now perform the second and last step in the demodulation process, and filter out the original signal.

   i) Define the filter coefficients **b** and **a** for filter **H(jw)** from pre-lab 4b (the 4th order filter).

   ii) Using the **lsim** function, pass the input signal **z** (along with its time samples **t**) through the first filter defined by coefficients **b** and **a**. Call the output of the filter **xr**.

   iii) Using the functions **fft**, **fftshift**, and **abs**, compute the magnitude of the Fourier Transform of signal **xr** using **N** = 8192 frequency samples. Call this variable **xr_abs**.

   iv) Using a 2 x 1 subplot, plot the following:
      - **xr vs. t**: no need to adjust axes
      - **xr_abs vs. w**: x-axis range = [-1000, 1000] and y-axis range = [0, 10000]

e) Comment your code, and then run your entire script and verify you were able to recover your original signal and that the signal matches the Morse Code representation for letter X to within a scaling factor.

Lab Report Question #4: Observe that the FFT of the recovered signal **xr(t)** has a max value around 5000, but the FFT of the original signal **x(t)** in Exercise #2 had a max value around

10,000. Clearly, there is something wrong with the low-pass filter. Explain the possible mistake with the filter and how it should be modified for next time.

Lab Report Question #5: In this exercise, we analyzed a system with input-output relationship **y(t) = x(t)cos(500t)**. A student in class claims this system is LTI. Explain why s/he is incorrect.

EXERCISE #4: Decoding a Morse Code Message

In this experiment, we will put together what we learned in the last three exercises to decode a Morse Code signal that consists of not just one letter, but rather three letters that make up an actual word.

Here is the scenario for this week: You are given a signal **y(t)** that contains a coded message from Agent 007 to you, as Agent 008, the code-breaking sleuth. The last words of the aging Agent 007 were "You can learn the key to the brain interface at u …" at which point Agent 007 saved the remaining message to a Matlab data. Your job now is to decipher the message encoded in **y(t)** and complete Agent 007's final words.

Here is what is known about the signal: **y(t)** is of the form

$$\boldsymbol{y(t) = m1(t)\cos(1000t) + m2(t)\cos(2000t) + m3(t)\cos(3000t)}$$

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. You will use your results from the pre-lab to implement a system for message recovery.

Lab Exercise:

a) Open a script file and call it **Ex4.m**. Clear all variables and close all figures.

b) From the class website, download and load the data file **Ex4.mat**. You should have:
   - **y**: message signal y(t)
   - **t**: time samples vector
   - **Fs**: sampling rate

c) We will perform the following tasks to prep for the upcoming loop:

   i. Load a figure window.

   ii. Define the filter coefficients **b** and **a** for the same low-pass filter you used in Exercise #3.

d) We will now implement the three systems using a for-loop. The outline for the for loop is as follows

```
% LOOP through exactly three times using the index variable i starting at i = 1

    % Implement your decision statement from pre-lab to decide which correct
    % cosine signal to use. Store the cosine signal in variable c.
```

% Compute the output of the multiplier using variables **y** and **c**. Store the output
% in the signal **z**.

% Pass the signal **z** (and its corresponding time sample **t**) through your low-pass
% filter using function **lsim**. Store the output in the signal **xr**.

% Using a 3 x 1 subplot, plot **xr vs. t** as the **ith** subplot.

% Label the axes.

% Using a string variable, create the title of the plot so that it displays the
% message index number in the title. So if i = 1, then the title will be **"Message
% m1(t)"**.

% END OF LOOP

e) Run your entire Matlab script and make sure you get a 3 x 1 plot window.

Lab Report Question #6:

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	· · -		

Using the International Morse Code table above, decode the letter from each signal, and complete Agent 007's final words: **"You can learn the key to the brain interface at u_____."**

Lab Report Question #7: In this lab, we decoded dash and dot signals by visual inspection of each graph. However, there are alternative ways to decode the message. A clever student in class realized you can decode a dash and dot signal using what we learned in Lab 4. Explain how you can use techniques from Lab 4, and in words, how you might implement that in Matlab.

ONE LAST MESSAGE

This quarter you have battled an evil EE235 student (twice!), decoded an insane amount of messages in Matlab, and learned how to perform convolution and convert signals from the time domain to the frequency domain, to name a few. 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 up and coming EE students to follow in your footsteps and take on the EE235 challenge. Good luck and keep in touch!

FOR LAST WEEK OF CLASS: LAB REPORT DUE

Turn in a PDF of your lab report (one per team) online via the link on the class website. A sample format for the lab report is on the class website. You may vary from this format but please include the basic sections. The report is due by 9pm on second day after your final lab section. Lab turn-in times will be checked, so no late lab reports will be accepted unless arranged in advance with the instructor.