University of Washington
Department of Electrical Engineering
EE 235, Fall 2016

Report for Lab 6:
Application of Fourier Transforms

Report by:
Chumei Yang 1561830

Bowen Xiao 1561730

Luke Jiang 1560831

1 EXECUTIVE SUMMARY

In this section, you will write 1-2 paragraphs summarizing the following:

1) In this lab we will apply two Fourier Transform properties which are filtering and modulation. And we will apply these two properties in real world questions in terms of creating and decoding a message encoded using Morse Code
2) In the prelab, we learned the difference between " .* " and " * "; reviewed the concept of filters, as well as the concept about how to modulate and demodulate a signal.
3) We learned how to build a filter in MATLAB, and how to write a code to modulate or demodulate a signal. By applying both methods, we decoded a Morse Code.
4) Filtering. Firstly, we built a low pass filter, and in the next steps, we used this low pass filter to filter a signal in demodulation process.
5) Decoding. we decode three messages by using the same filters in previous exercise an function isim.

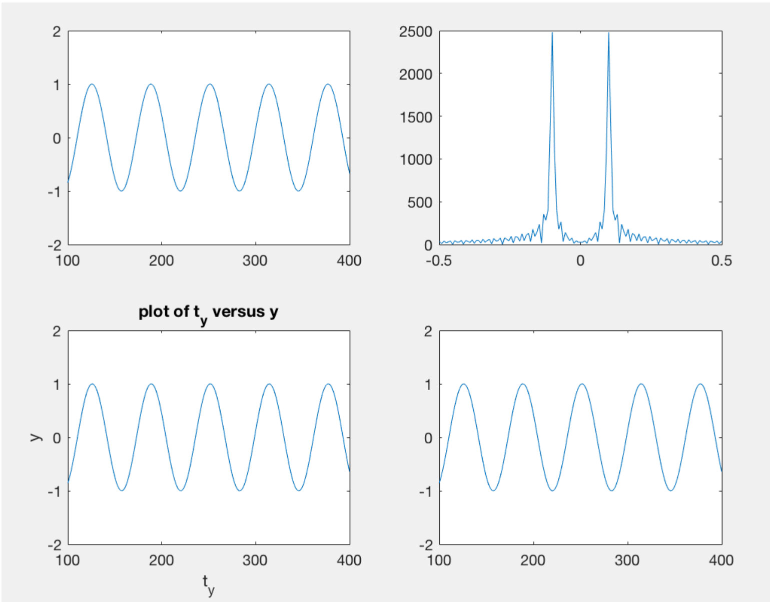
2 EXERCISE #1: FILTERING

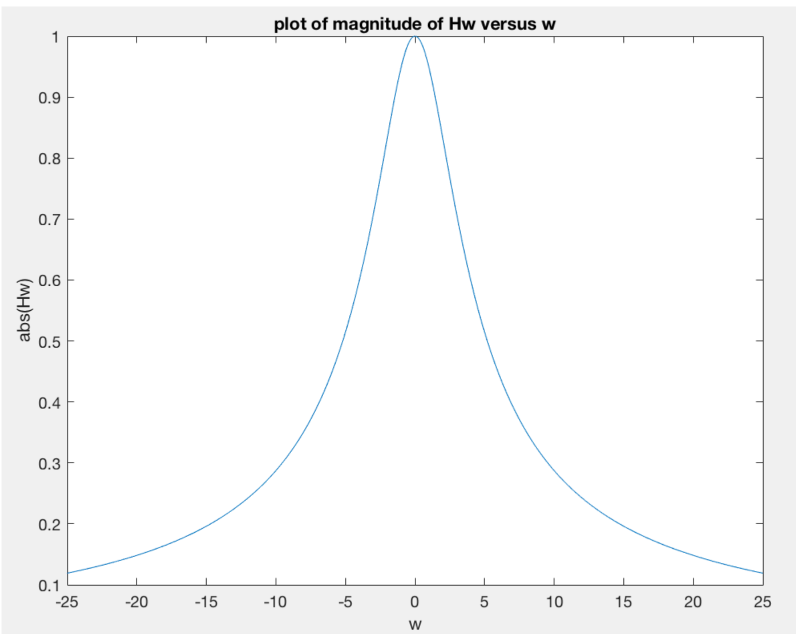
In this part, we first built a signal x(t)=cos(0.1t). When then created a low pass filter and make x(t) undergo the filter to produce y(t). We plotted the signal, the filter, and the output.

2.1 COMMAND WINDOW OUTPUT

>> Ex1

PLOTS





2.2 LAB REPORT QUESTIONS

In this section, for each question, restate the lab question and then provide your answer

Question(1/7): 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?

Answer: There would be a total of three figure windows. The first one with only two plots on top, the second one with the filter in frequency domain, and the third one with two plots in the bottom.

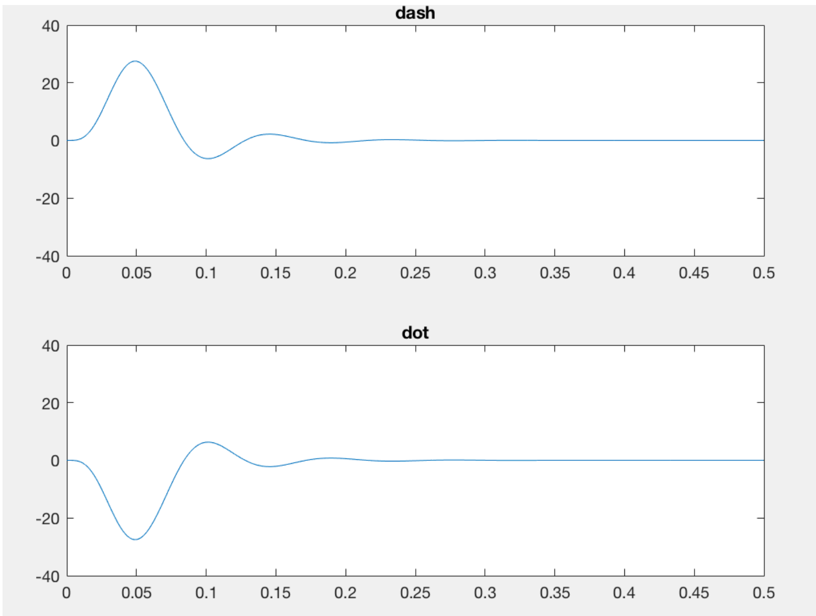
3 EXERCISE #2: AMPLITUDE MODULATION

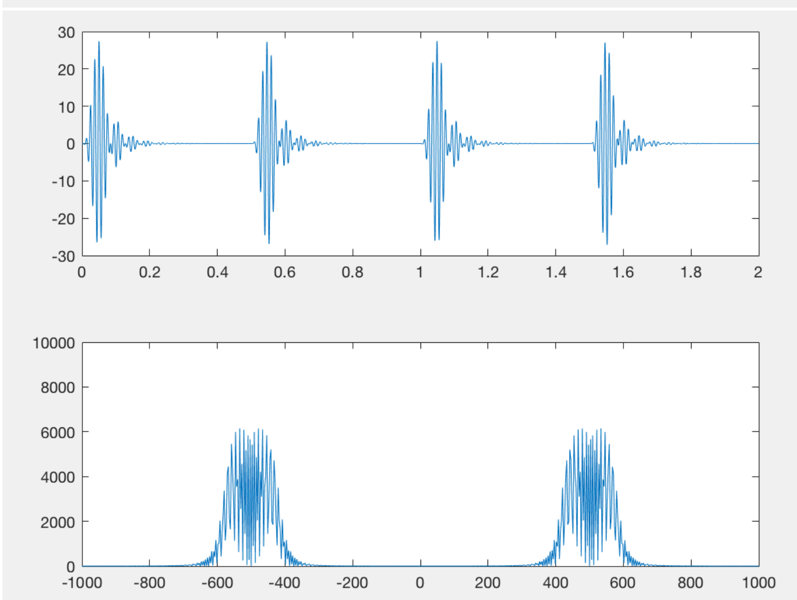
In this part, we created x(t) by concatenating dash and dot together, converted to frequency domain, modulated it with a cosine function, and plotted it in both time and frequency domains.

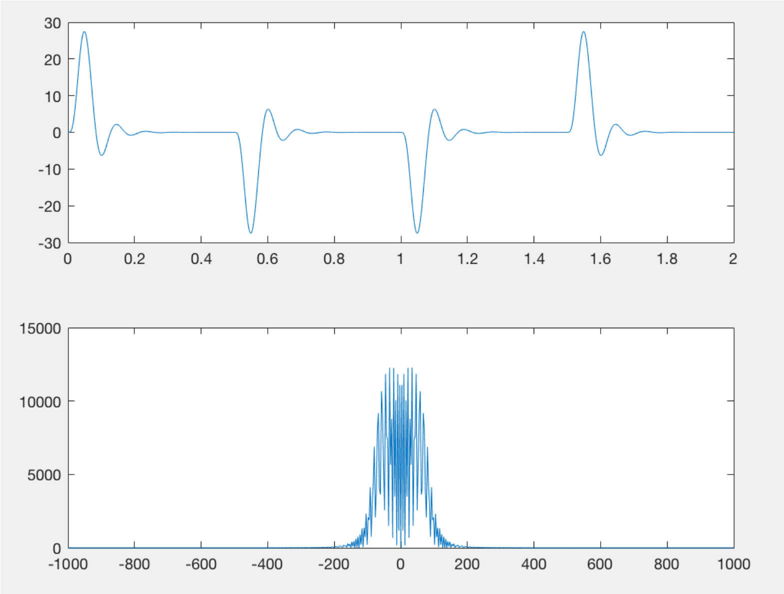
3.1 COMMAND WINDOW OUTPUT

>> Ex2

3.2 PLOTS







3.3 LAB REPORT QUESTIONS

Question(2/7): 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 $w_M = 50$ as his/her pre-lab. Explain why this student is incorrect. Also, explain why using any carrier frequencies where $w_M < 200$ will not work.

Answer: Because cos(50t) would shift make two copies of half-magnitude X(jw) centered at -50 and 50, and because the bandwidth of X(jw) is around 200, this would cause two copies to overlap, therefore making it unable to be recovered by demodulation. Because the bandwidth of X(jw) is 200, any wM less than that will cause the frequencies to overlap, which will cause aliasing and make it unable to be recovered.

Question(3/7): 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?

Answer: There will only be one figure window with the last plot.

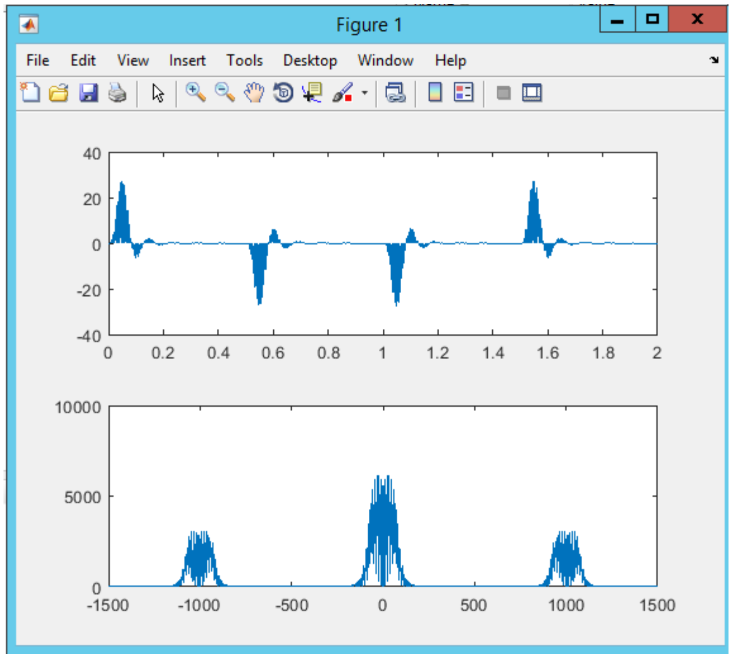
4 EXERCISE #3: AMPLITUDE DEMODULATION

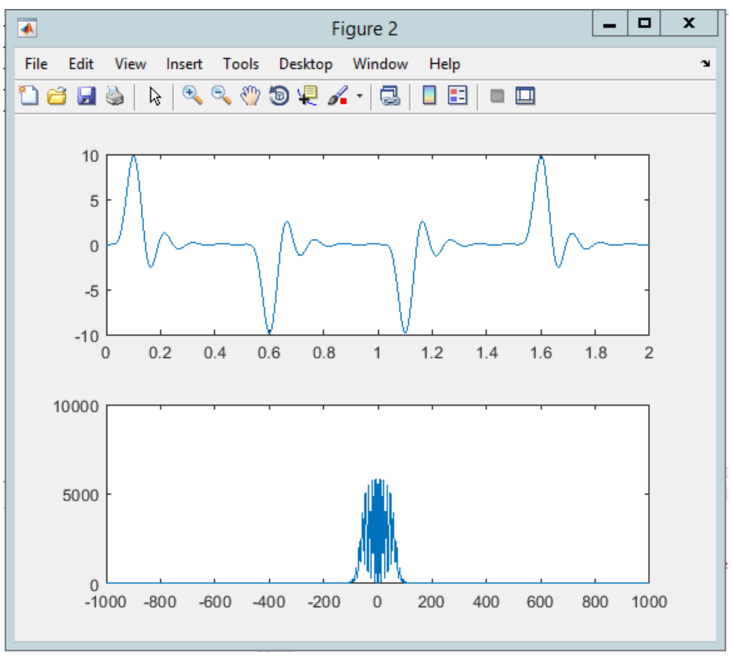
In this part, we learned how to undo the modulation and recover a signal to its original signal.

4.1 COMMAND WINDOW OUTPUT

```
>> Ex3
```

4.2 PLOTS (IF APPLICABLE)





4.3 LAB REPORT QUESTIONS (IF APPLICABLE)

Question (4/7): 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.

Answer:

The gain of the low pass filter is 0.5 so that the max value of xr(t) is 1/2 of the original signal in E#2.

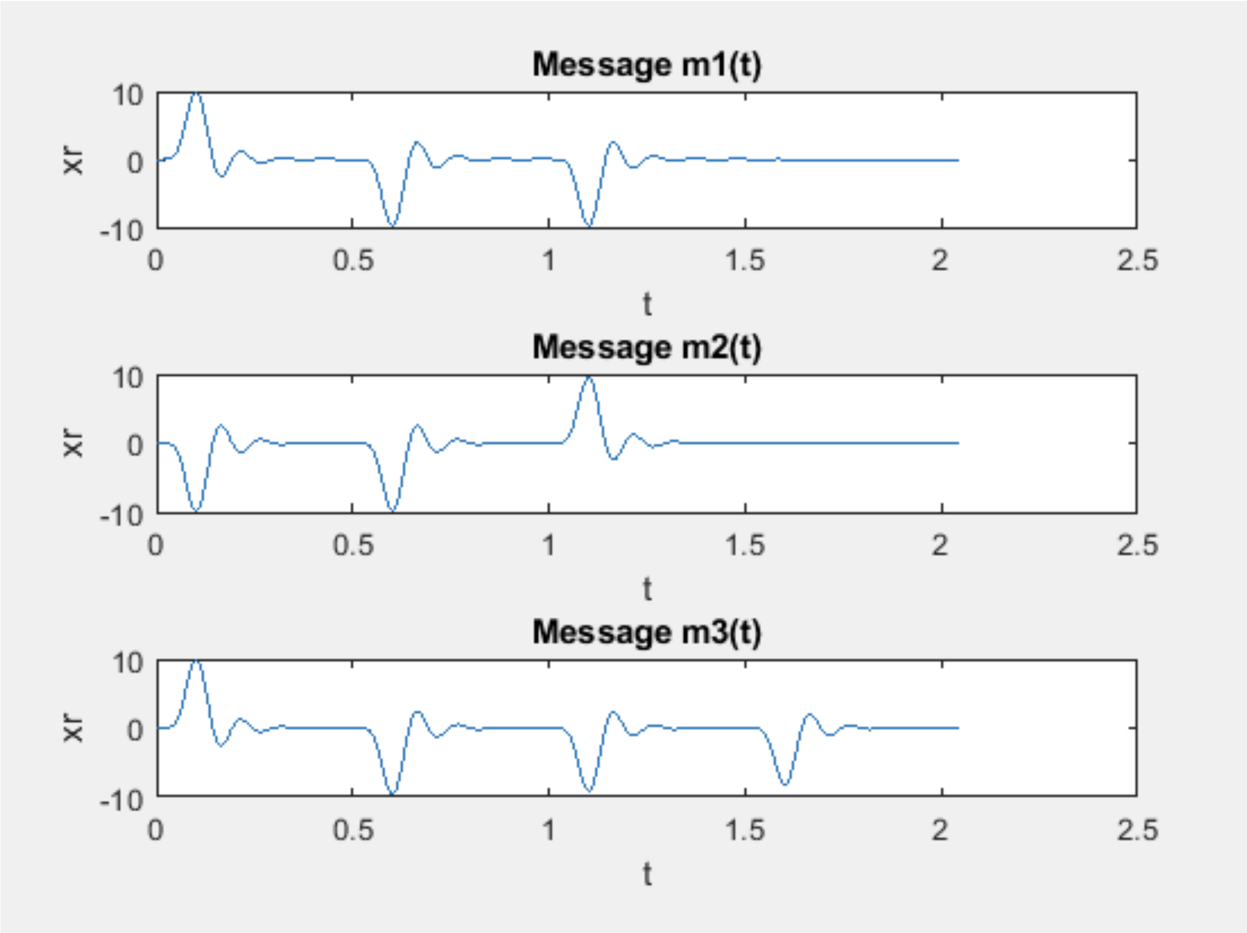
2 * H(jw) could solve this problem.

Question (5/7): 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.
Answer:
The system is not linear. Ay(t) = Ax(t)cos(500t); y(At) = x(At)cos(500At), Ay(t) != y(At) so it is not linear.

5 EXERCISE #4: DECODING A MORSE CODE MESSAGE

In this part, we decode three messages by using the same filters in previous exercise an function isim.

5.1 PLOTS



5.2 LAB REPORT QUESTIONS

Question (6/7): Using the InternationalMorse 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______."

Answer:
udub.

Question (7/7): I 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.

Answer:
From lab 4, we know that we can recover a signal by finding the maximum correlation of the unknown signal with a set of possible signals. Therefore, we first need to split the signal into four parts by multiplying the signal with four shifted pulse function so that each part represents a dash or dot or zero (no value). Then, we convolve each part with dash(t). If the result is positive, we write 1 to a result string, representing dash. If it's negative, we write zero to result string, representing dot. If it's very small and close to zero, we break the loop. Then, we construct an array of strings representing A to Z according to the Morse Code Table. And finally we compare the result string to the table to find the match.

Pseudo-code:

```
signal to be decoded x(t)
construct three pulse functions p1, p2, p3
array AZ = ["01" "1000" "1010" "100" "0" "0010" …]
string alphabet = "abc…z"
array signal_parts = x(t) * [p1 p2 p3]
result = ""
for i = 1:3
   int conv = conv(signal_parts[i], dash(t))
   if conv == 0
      break;
   elseif conv > 0
         result = [result, "1"];
      else
        result = [result, "0"];
end
for i = 1:26
   if (AZ[i] == result)
      decode = alphabet.charAt(i);
```