ECE 348: Digital Signal Processing Lab
Lab 1 (Spring 2020) — 100 points

Waheed U. Bajwa

January 25, 2020

General Instructions

Please submit a written report within 2 weeks of *your* lab session, describing the purposes of the experiments and the methods used, and include all graphs and Matlab code. Examples of all the graphs that you need to generate in this lab and include in your report are contained in this handout. The reports must be uploaded to Sakai under Assignments within the allotted time frame. The written report will count *only if* you had attended the *full* double-period lab session in which you are registered.

1 Convolution

The convolution of an order-M causal filter, h_n, $n = 0, 1, \dots, M$, and a length-L causal signal, x_n, $n = 0, 1, \dots, L-1$, results in a length-$(L+M)$ causal output signal given by:

$$y_n = \sum_{m=\max(0,n-L+1)}^{\min(n,M)} h_m\, x_{n-m}\,, \quad \text{for} \quad n = 0, 1, \dots, L+M-1. \tag{1}$$

Equation (1) can be evaluated numerically using the built-in MATLAB function **conv**, with usage:

```
y = conv(h,x);
```

where `h,x,y` are the corresponding arrays.

Problem 1.1 (10 points)**.** Consider an integrator-like filter defined by the I/O equation:

$$y(n) = \frac{1}{15}[x(n) + x(n-1) + x(n-2) + \cdots + x(n-14)].$$

This filter accumulates (integrates) the present and past 14 samples of the input signal. The factor $1/15$ represents only a convenient scale factor for plotting purposes. Such filters are also used in finance, e.g., referred to as 15-day moving averages (if n represents days). It follows that the impulse response of this filter will be:

$$h_n = \begin{cases} 1/15, & \text{for } 0 \le n \le 14, \\ 0, & \text{otherwise.} \end{cases}$$

To observe the steady-state response of this filter, as well as the input-on and input-off transients, consider a *square wave* input signal x_n of length $L = 200$ and period of $K = 50$ samples. Such a signal may be generated by the simple vectorized MATLAB code:

```
n = 0:L-1;
x = double(rem(n,K) < K/2);       % REM is the remainder function
```

Using the function **conv**, compute the output signal y_n and plot it versus n on the same graph with x_n. As the square wave periodically goes on and off, you can observe the on-transient, steady-state, and off-transient behavior of the filter (see Fig. 1(a)).

Problem 1.2 (5 points)**.** Repeat Problem 1.1 for the filter:

$$h_n = \begin{cases} 0.25(0.75)^n, & \text{for } 0 \le n \le 14, \\ 0, & \text{otherwise.} \end{cases}$$

This filter acts more like an RC-type integrator than an accumulator (see Fig. 1(b)).

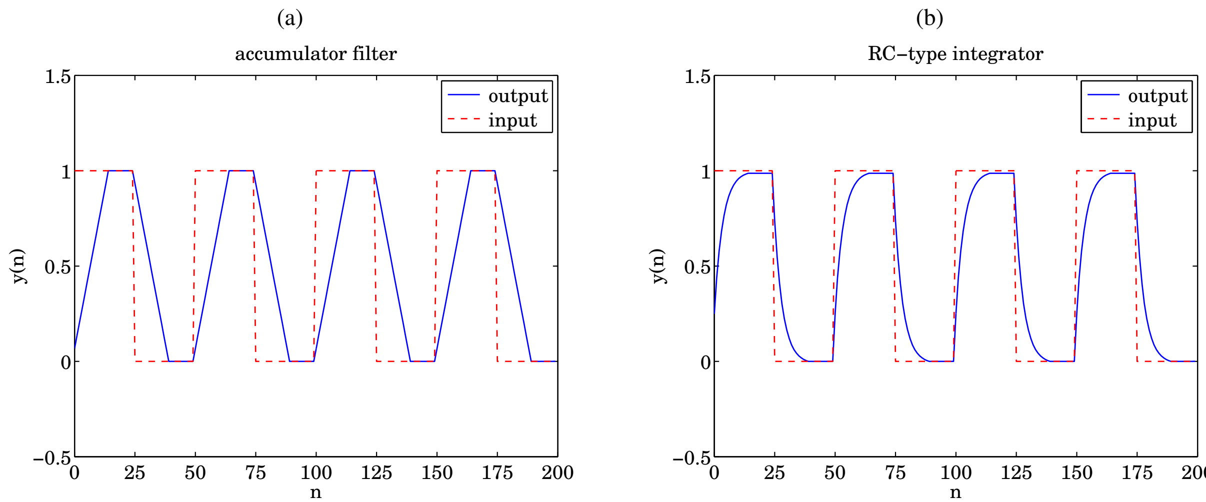

Fig. 1: Plots for (a) Problem 1.1 and (b) Problem 1.2.

Problem 1.3 (10 points)**.** To demonstrate the concepts of impulse response, linearity, and time-invariance, consider a filter with finite impulse response $h_n = (0.95)^n$, for $0 \le n \le 24$. The input signal,

$$x(n) = \delta(n) + 2\delta(n-40) + 2\delta(n-70) + \delta(n-80), \qquad n = 0, 1, \dots, 120,$$

consists of four impulses of the indicated strengths occurring at the indicated time instances. Note that the first two impulses are separated by more than the duration of the filter, whereas the last two are separated by less. The input vector x can be constructed with the help of the following anonymous function implementing the discrete-time delta function $\delta(n)$:

```
d = @(n) double(n==0);          % n can be a vector of indices
```

Using the function **conv**, compute the filter output y_n for $0 \le n \le 120$ and plot it on the same graph with x_n (5 points). Comment (5 points) on the resulting output with regard to linearity and time invariance (see Fig. 2).

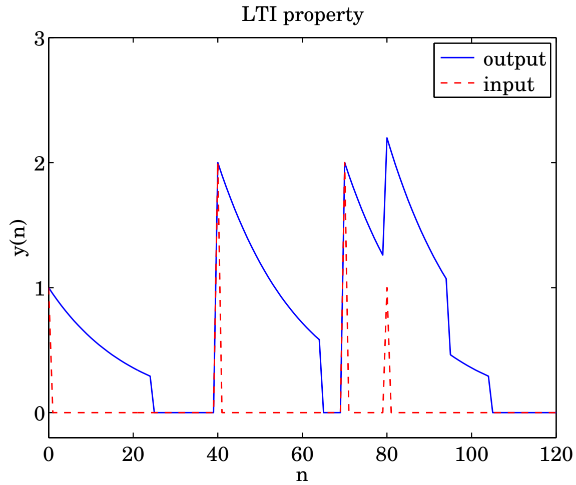

Fig. 2: Plot for Problem 1.3.

2 System Identification

For this experiment, please download the p-coded MATLAB function **syst.p** from Sakai Resources for this lab. It implements the input/output relationship of an unknown LTI system that has a finite impulse response. The objective is for you to determine this impulse response. The function can be used like any other Matlab function, except that you cannot edit it. It has usage:

```
y = syst(x,id);
```

where `x` is a vector of input samples and `y` is the resulting vector of output samples. The function is customized to each student, and you must enter *your* own Rutgers ID number (as an integer) as the second function input denoted by `id`.

Problem 2.1 (5 points)**.** Send an impulse $\delta(n)$ into this system to determine the impulse response $h(n)$.

Problem 2.2 (10 points)**.** Similarly, determine the system output by sending in the delayed impulse $\delta(n-2)$ (2.5 points). Repeat with the input $x(n) = 3\delta(n) + 2\delta(n-2)$ (2.5 points). Are the results consistent with linearity and time-invariance (5 points)?

Problem 2.3 (10 points)**.** Consider the two input vectors:

```
x = [1,  1,  1,  1,  1,  1,  1,  1,  1,  1];
x = [1, -1,  1, -1,  1, -1,  1, -1,  1, -1];
```

For each case, determine the corresponding output in two ways: (i) by using the above function (5 points), and (ii) by convolving x with the impulse response h that you found in Problem 2.1 (5 points).

3 DTFT Computation

The DTFT of a length-L sequence, $x(n)$, $n = 0, 1, \ldots, L-1$, is defined by:

$$X(\omega) = \sum_{n=0}^{L-1} x(n)\, e^{-j\omega n}, \tag{2}$$

where ω is the digital frequency in units of radians per sample. In the textbook, this quantity is denoted by $X\left(e^{j\omega}\right)$. However, here we will use the simpler notation $X(\omega)$ for programming purposes. Equation (2) can be evaluated numerically using, for example, the built-in function **freqz**, with usage:

```
X = freqz(x,1,w);
```

where x is the length-L time sequence, w is a length-N vector of frequencies at which to evaluate (2), say, $w = [w_1, w_2, \ldots, w_N]$, and X is the corresponding vector of DTFT values at those frequencies, that is, $X = \big[X(w_1), X(w_2), \ldots, X(w_N)\big]$.

Problem 3.1 (10 points)**.** Consider a length-L pulse signal consisting of L ones:

$$p(n) = u(n) - u(n-L) = \begin{cases} 1\,, & \text{if } \ 0 \le n \le L-1, \\ 0\,, & \text{otherwise,} \end{cases} \tag{3}$$

where $u(n)$ is the unit-step function. This can be generated for any vector of n's by the MATLAB function (assuming L was already defined):

```
p = @(n) double(n>=0 & n<=L-1);
```

Now, choose $L = 40$, and construct $p(n)$ and plot it versus n over the time interval $-5 \le n \le 45$ using a **stem** plot (see Fig. 3(a)) (1 point).

Next, show that the DTFT of $p(n)$ is given by the analytical expression (2.5 points):

$$P(\omega) = \sum_{n=0}^{L-1} p(n)\, e^{-j\omega n} = \frac{1-e^{-j\omega L}}{1-e^{-j\omega}} = e^{-j\omega(L-1)/2}\, \frac{\sin(\omega L/2)}{\sin(\omega/2)}. \tag{4}$$

For the purpose of programming (4) in MATLAB, it proves convenient to re-write it in terms of MATLAB's **sinc** function, which is vectorized and avoids a computational issue at $\omega = 0$:

$$P(\omega) = L\, e^{-j\omega(L-1)/2}\, \frac{\text{sinc}\left(\frac{\omega L}{2\pi}\right)}{\text{sinc}\left(\frac{\omega}{2\pi}\right)}, \tag{5}$$

where

$$\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x}.$$

Given a value for L, write an anonymous MATLAB function $P(\omega)$ implementing (5). It should be vectorized with respect to the variable ω (3 points).

Next, using your anonymous function P, calculate the corresponding DTFT of $p(n)$ for $L = 40$ at $N = 1001$ equally-spaced frequency points over the interval $-\pi \le \omega \le \pi$ and plot the following normalized quantity versus ω,

$$F(\omega) = \left| \frac{P(\omega)}{P(\omega_0)} \right|,$$

where $\omega_0 = 0$ (see Fig. 3(b)) (2 points). Moreover, verify that the DTFT computed using the analytical expression (5) agrees with the numerical computation using the **freqz** function over the same set of N frequencies (1.5 points).

Problem 3.2 (10 points)**.** Consider a length-L finite portion of a sinusoid of frequency ω_0, defined as the sinusoidally modulated pulse signal $p(n)$ of Problem 3.1:

$$s(n) = \sin(\omega_0 n)\, p(n) = \begin{cases} \sin(\omega_0 n)\,, & \text{if } \ 0 \le n \le L-1, \\ 0\,, & \text{otherwise.} \end{cases} \tag{6}$$

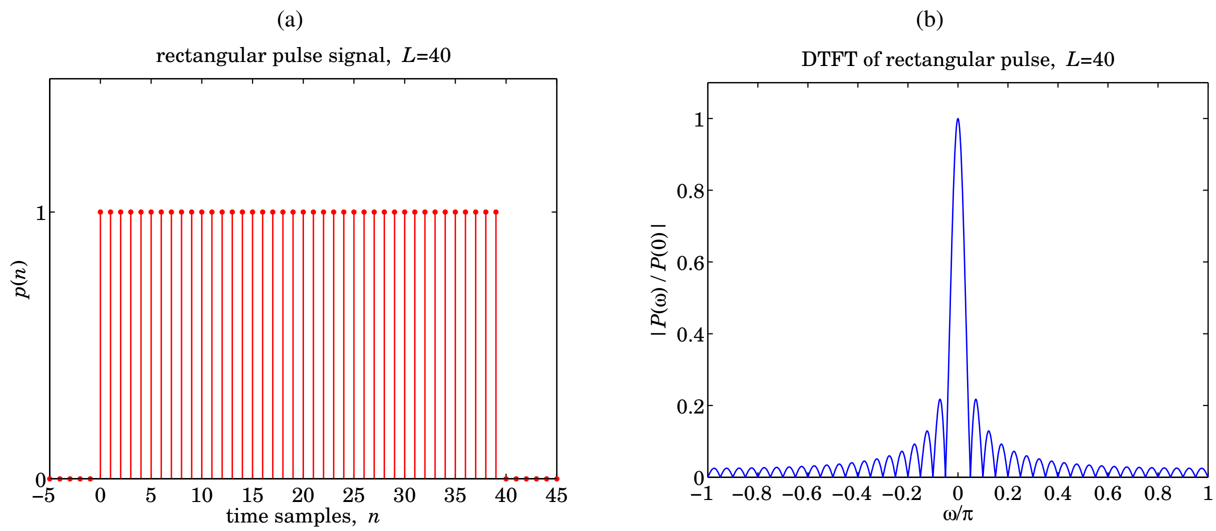

Fig. 3: Plots for Problem 3.1.

Using the modulation property of the DTFT, show that the DTFT of $s(n)$ is given by the following analytical expression in terms of the DTFT $P(\omega)$ of (5) (2.5 points):

$$S(\omega) = \frac{1}{2j}\big[P(\omega-\omega_0) - P(\omega+\omega_0)\big]. \tag{7}$$

Choose $L = 40$ and $\omega_0 = 0.2\pi$ radians/sample. Using your anonymous MATLAB function $P(\omega)$ of Problem 3.1, evaluate (7) over the same set of N frequencies as in Problem 3.1, and plot the following normalized quantity versus ω (2.5 points):

$$F(\omega) = \left|\frac{S(\omega)}{S(\omega_0)}\right|$$

for the present value of ω_0. Plot also the sinusoidal signal $s(n)$ versus n using a **stem** plot over the time range $-5 \le n \le 45$ (2.5 points). See example graphs in Fig. 4.

Moreover, re-calculate the DTFT $S(\omega)$ using **freqz** and verify that you get the same results as (7) to within the double-precision floating-point accuracy of MATLAB (2.5 points).

Problem 3.3 (10 points)**.** Next, consider a length-L signal consisting of the sum of two sinusoids of frequencies $\omega_1 = 0.2\pi$ and $\omega_2 = 0.4\pi$, with relative amplitudes of 1 and 0.8, as defined below:

$$s(n) = \big[\sin(\omega_1 n) + 0.8\sin(\omega_2 n)\big]p(n) = \begin{cases} \sin(\omega_1 n) + 0.8\sin(\omega_2 n)\,, & \text{if } \ 0 \le n \le L-1, \\ 0\,, & \text{otherwise.} \end{cases} \tag{8}$$

Show that the DTFT of $s(n)$ is given analytically as the linear combination (2.5 points):

$$S(\omega) = S_1(\omega) + 0.8\,S_2(\omega) \tag{9}$$

where $S_1(\omega)$ and $S_2(\omega)$ are given by (7), with ω_0 replaced by ω_1 and ω_2, respectively. For convenience, you may wish to define an anonymous Matlab function for (9).

For $L = 40$ and using (9), plot the DTFT $S(\omega)$ versus ω, normalized with respect to its value at $\omega = \omega_1$, that is, evaluate and plot the quantity (2.5 points):

$$F(\omega) = \left|\frac{S(\omega)}{S(\omega_1)}\right|.$$

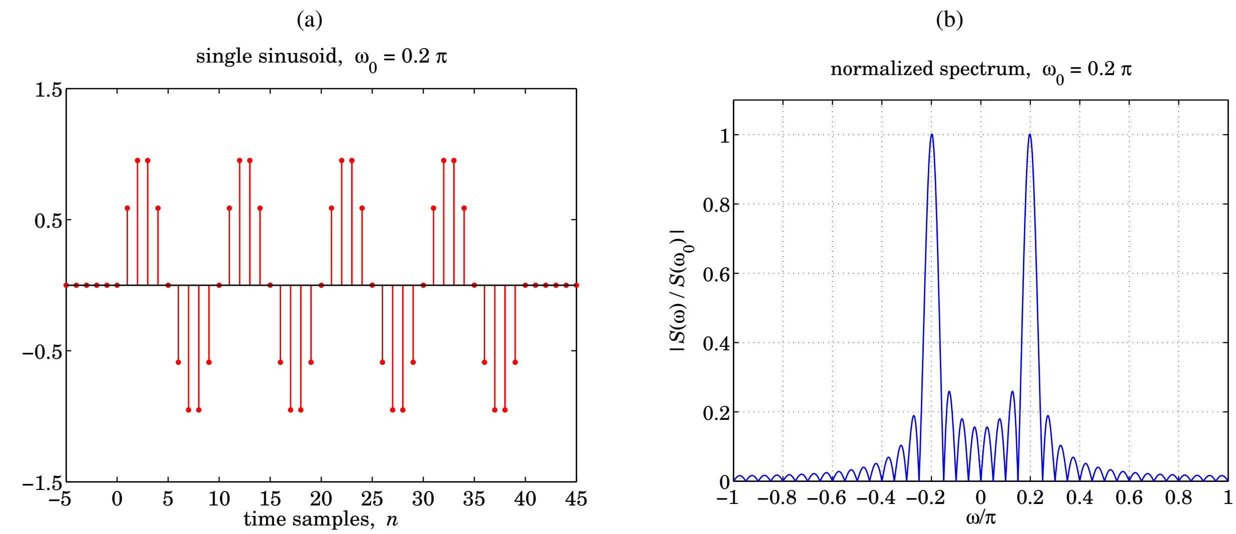

Fig. 4: Plots for Problem 3.2.

Also, plot $s(n)$ of (8) versus n using a **stem** plot over the range $-5 \leq n \leq 45$. See Fig. 5 for reference purposes (2.5 points).

And again, re-calculate the DTFT using **freqz** and verify that you get essentially the same results as with the analytical formula (2.5 points).

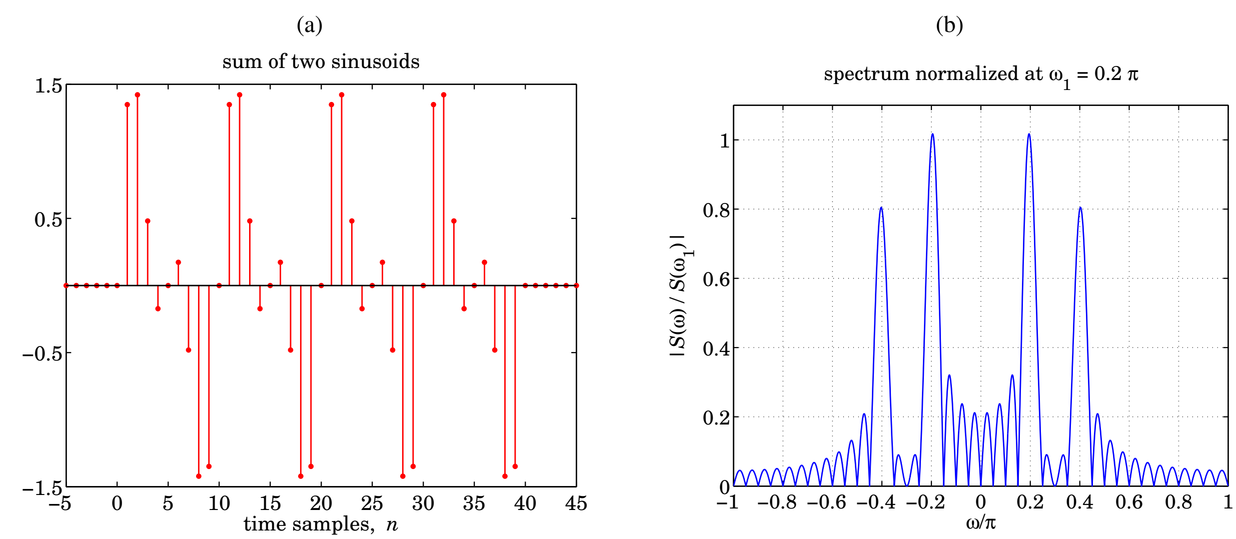

Fig. 5: Plots for Problem 3.3.

Problem 3.4 (10 points)**.** If you zoom into the spectral peaks at ω_1 and ω_2, you will notice that the peaks do not occur exactly at ω_1 and ω_2 as they are in the case of infinitely-long sinusoids. This phenomenon is due to the finite-duration of the sinusoids, which causes the spectral peaks to broaden and interact with each other, shifting them slightly relative to one another.

For the particular choices of L, ω_1, ω_2 of Problem 3.3, determine the actual location of the spectral peaks (5 points) using the built-in MATLAB function, **fminbnd**. To do so, define a function handle for the negative magnitude of (9),

$$f(\omega) = -\big|S(\omega)\big|$$

and pass it into **fminbnd**, searching first near the true frequency ω_1, and then near ω_2. You should obtain the following peak frequencies, to be compared with $\omega_1 = 0.2\pi$, $\omega_2 = 0.4\pi$,

$$\omega_{1,\,\text{peak}} = 0.1950\pi\,, \quad \omega_{2,\,\text{peak}} = 0.4030\pi\,.$$

The same biasing effect occurs also in the single-sinusoid case and gets more pronounced when ω_0 is near 0 or near π. This is so because the peaks at $\pm\omega_0$ are now closer and interact more with each other. Using the parameters from Problem 3.2 and **fminbnd**, verify (5 points) that the actual peak maximum of the finite-length signal is at,

$$\omega_{0,\,\text{peak}} = 0.1983\pi$$

as compared to $\omega_0 = 0.2\pi$.

Problem 3.5 (10 points)**.** Graphically verify that the above effect tends to disappear as the signal length L increases, resulting in narrower peaks that interact less with each other. Listed below are the peak frequencies for the double and single sinusoidal cases for lengths $L = 80$ (5 points) and $L = 160$ (5 points), and shown also are the corresponding spectra in Fig. 6:

$$L = 80\,, \omega_{1,\,\text{peak}} = 0.1987\pi\,, \quad \omega_{2,\,\text{peak}} = 0.4008\pi\,, \quad \omega_{0,\,\text{peak}} = 0.1996\pi$$
$$L = 160\,, \omega_{1,\,\text{peak}} = 0.1997\pi\,, \quad \omega_{2,\,\text{peak}} = 0.4002\pi\,, \quad \omega_{0,\,\text{peak}} = 0.1999\pi$$

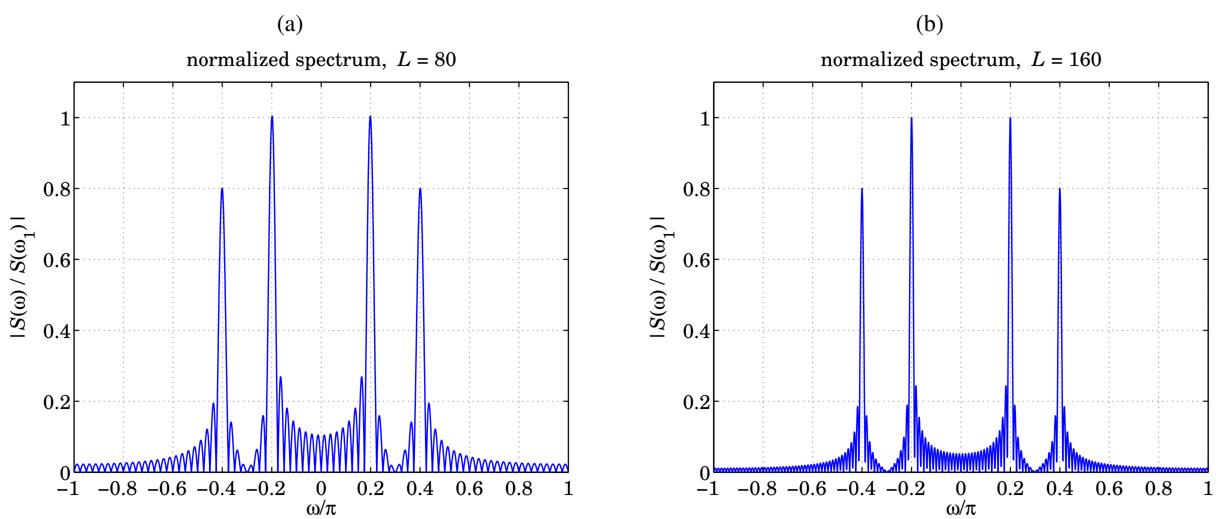

Fig. 6: Plots for Problem 3.5.

Some other observations are that the peak widths get narrower with increasing L, but the sidelobes immediately to the left and right of each mainlobe appear to not diminish with increasing L. These and other spectral analysis issues, such as frequency leakage and resolution, and the effect of using non-rectangular windows, will be explored in a future lab as well in future classes.