EE 341 Spring 2013

Lab 4: Properties of Discrete-Time Fourier Transform (DTFT)

Objective

In this lab, we will learn properties of the discrete-time Fourier transform (DTFT), such as conjugate symmetry and discrete-time convolution via DTFT multiplication. At the end, you will experiment with examples of real-world data: stock prices, music, and animal sounds. Credit for the concepts, code, and data samples for this lab goes to Prof. Somsak Sukittanon at Univ. of Tennessee, who got his Ph.D. from UW Electrical Engineering, and won our EE dept's Best Teacher Award.

MATLAB commands for TAs to go over (Do not turn this part in)

```
[H,w] = freqz( b, a, N );
```

Brief description:

Example of how to use it:

```
[H,w] = freqz( b, a, N, 'whole' );
```

Brief description:

Example of how to use it:

```
[H,w] = freqz( b, a, N, fs );
```

Brief description:

Example of how to use it:

Part 1: The Frequency Domain

This lab makes use of the DTFT

$$X\left(e^{j\omega}\right) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n}$$

but will estimate it with relatively long yet finite sequences of data.

1. Case 1: You have a data sequence (not a difference equation), say $x[n]$. Examples are speech, music, and stock data. To estimate the Fourier transform, use `freqz` with b equal to the data and a = 1. I normally use N = 512 or 1024 to zero pad the data and make the DTFT estimate smooth. If $x[n]$ is longer than 1024, you need to use N equal or greater than the length of $x[n]$.

```
Usage:  [H, W] = freqz( B, A, N );
        [H, W] = freqz( x, 1, 512 );
```

   The Matlab program below shows how to plot and label the magnitude and phase response obtained from `freqz`.

2. Case 2: You have a difference equation such as

$$y[n]-a_2y[n-2]=b_0x[n]+b_1x[n-1]+b_2x[n-2]$$

   For this case, create a and b vectors in MATLAB to use with freqz. Be careful, because in the above example, $a_0 = 1$ and $a_1 = 0$ implicitly. Refer to the code below for a numeric example.

```
clear all;clc;close all;

B = [0.2929    0.5858    0.2929];
A = [1.0000    0        -0.1716];
N = 512;
[H,W] = freqz( B, A, N);

figure(1);
subplot(211); plot( W, abs(H) );                %% plot abs of complex
title('Magnitude response');
xlabel('Freq (rad)'); ylabel('Amplitude');

subplot(212); plot( W, angle(H) );              %% plot phase of complex
title('Phase response');
xlabel('Freq (rad)'); ylabel('Angle');
```

Assignment #1: Animal Sounds

1. Download the four animal sounds from the class web site. Create a new m-file with clear commands on the top.
2. Load each sound (using `wavread`) and play it out (using `soundsc`). Record the sampling rate and the length of the data in seconds and samples (use the `length` command for this).
3. Plot the time signal and the DTFT of each one. Please title your plots and label your axes. For each DTFT, choose N greater than the length of the signal in order to get a smooth plot. Comment on what you see in the DTFT's.
4. **For your report, include the following for each animal sound:**
   a. **Name of the audio file.**
   b. **Sampling rate (in Hertz).**
   c. **Length in seconds and in number of samples.**
   d. **Labeled time-domain and frequency-domain plots.**
   e. **Comments on what you see in the frequency domain plots.**

5. So far you have plotted with the frequency axis in radians, which may be hard to interpret. If you know the sampling rate of the data, however, you can "un-normalize" the frequency axis while keeping the same plot shape. To do this job, we can use `freqz` with different parameters:

```
[x, Fs, b] = wavread( 'cat.wav' );
[H, F] = freqz( x, 1, length(x), Fs );
```

6. Plot the animal sound DTFT's again in Hertz by following the example in part 5. Make sure to change the name of the x-axis label! **Include these plots in your report, and answer the question: What is the frequency of the first peak (fundamental frequency) of the cat sound?**

Part 2: The Time Domain

1. We learn how to do convolution using `filter.m`, which works similarly to `freqz`.

```
y = filter( b, a, x );
```

2. Case 1: If the filter impulse response is a finite-length sequence (not a difference equation), then *b* will be the impulse response samples and *a* = 1. `x` is the input sequence. The output will have length equal to input.
3. Case 2: If you have a difference equation, we just go ahead and use the *b* and *a* coefficients as in the following example:

$$y[n]-0.1716y[n-2]=0.2929x[n]+0.5858x[n-1]+0.2929x[n-2]$$

```
clear all;clc;close all;

B = [0.2929    0.5858    0.2929];
A = [1.0000    0         -0.1716];
x = randn( 1, 512 );
y = filter( B, A, x );

figure(1);
plot( x ); hold on;
plot( y, 'r' );
```

Assignment #2: Stock Price

1. Get a text file containing Microsoft stock price from the class web site.
2. Load the data using the `load` command.
3. If you plot the stock data, you will see fast fluctuations in the price. Long-term investors tend to look at the slower underlying trend using a low-pass filter. Here is the 30-point impulse response sequence for a 30-day moving average filter:

$$h[n] = \left[\frac{1}{30} \quad \frac{1}{30} \quad \frac{1}{30} \cdots \frac{1}{30} \right] \qquad 0 \le n \le 29$$

4. Create this filter in MATLAB (hint: use the `ones` command). Then apply this filter to the stock data. **Plot the original graph over time and plot the filtered sequence on the same axes in red color (use the `hold on` command to overlay two plots on the same axes). Include this graph in your report. Comment on the differences between the data and the filtered result.**
5. **Plot and turn in the DTFT magnitude and phase of h[n] using `freqz`. What kind of filter is it? Low-pass, high-pass, band-pass, or band-stop?**

Part 3: Simple Three-Band Equalizer

Use the file music.wav

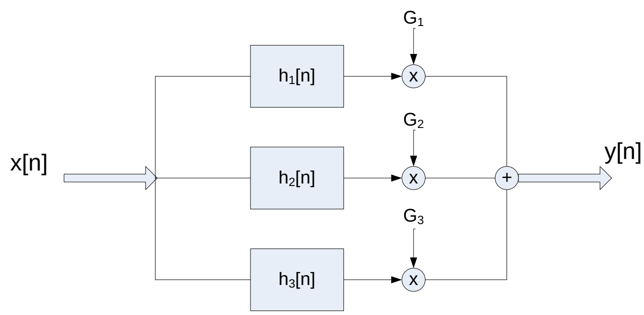
1. The music file is standard CD quality, sampled at 44.1 KHz with a 16-bit word size for each channel. **For your report, answer the following: If the file is 10 seconds long, then how many samples of data will there be? (Hint: the data samples are in stereo.) If we needed to transmit at this CD quality, with no compression, how many bits/second would be needed?**

Test the equalizer filters

2. Now, let's say we have three filters, each defined by the difference-equation coefficients in the table below. One is low-pass (LP), the other is band-pass (BP), and the final is high-pass (HP), so that all together they cover the full frequency range. You will learn how to design them later in a class such as EE 442. For this moment, just learn which one is LP, HP, BP. **Use `freqz` with `Fs = 44100` and plot the frequency response of each filter for your report.**

Filter coefficients	LP, HP, or BP?
B1 = 0.0495 0.1486 0.1486 0.0495 A1 = 1.0000 -1.1619 0.6959 -0.1378	
B2 = 0.1311 0 -0.2622 0 0.1311 A2 = 1.0000 -0.4824 0.8101 -0.2269 0.2722	
B3 = 0.0985 -0.2956 0.2956 -0.0985 A3 = 1.0000 0.5772 0.4218 0.0563	

3. The idea of an equalizer is to filter, weight, and sum. Audio mixing boards do this, though with many more than the three channels we have here. Use the MATLAB code below to finish your own equalizer script. Try different values for G1, G2, and G3 (shown in the block diagram) and listen to the resulting $y[n]$. **Turn in your script on E-Submit, with one setting for G1, G2, and G3, and attach a print-out with your lab report.**




```
clear all;clc;close all;

% prepare B,A for each filter here

% load music
[x,Fs,b] = wavread('music.wav');
x_left = x(:,1);
x_right = x(:,2);

% filter 1
y1_left = filter(B1,A1,x_left);
y1_right = filter(B1,A1,x_right);
y1 = [y1_left y1_right];


% continue here for filter 2 and 3

% at the end, we need to sum
G1 = 1.1;
G2 = 0.5;
G3 = 2;
y = G1*y1 + G2*y2 + G3*y3;
soundsc(y,Fs);
```

4. **For your report: Compare the filtered audio with the original signal. What does each filter do to the music? Why don't you hear y1, y2, y3 independently as distinct sources?**

Personal Check list for Lab 4 (do not turn in)

- ☐ Understand how to view the DTFT of real data such as sound, music, stock
- ☐ Understand how to filter real data such as stock, music
- ☐ Understand how to use MATLAB command `freqz` with sampling rate as input
- ☐ Understand how a simple 3-channel equalizer works and the role of dB magnitude to characterize gain.