University of Illinois at Urbana-Champaign
Department of Electrical and Computer Engineering

ECE 311: Digital Signal Processing Lab
Chandra Radhakrishnan
Peter Kairouz

LAB 2: **DTFT, DFT, and DFT Spectral Analysis**

Summer 2011

1 Overview

In this lab we will perform frequency domain analysis of Discrete-Time signals using MATLAB. The objective of this lab is to teach you how interpret and analyze the frequency spectra of digital signals with fixed frequency contents and others with time varying frequency contents such as speech signals.

2 DTFT

The DTFT is used to represent discrete-time signals in terms of complex exponential signals $e^{j\omega n}$. The DTFT of a discrete-time signal $x[n]$ is given by,

$$X_d(w) = \sum_{-\infty}^{\infty} x[n]e^{-j\omega n},$$

which is called the DTFT analysis equation. The synthesis equation is given by,

$$x[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} X_d(\omega)e^{j\omega n}d\omega$$

Note the DTFT is periodic with period 2π. The two major limitations for any numerical implementation of DTFT are that $x[n]$ must have finite length and $X(e^{j\omega})$ can only be computed at a finite number of samples of the continous frequency variable ω.

3 DFT and IDFT

In order to compute the DTFT of a signal $x[n]$ using MATLAB the signal $x[n]$ must be first truncated to form a finite length signal. This is not necessarily a limitation since in real life applications, we only have finite recordings of sampled and quantized (i.e. digitized) continuous time signals. Also, $X(e^{jw})$ can be computed only at a discrete set of frequency samples, w_k. This is also not necessarily a limitation: if enough samples are chosen, the discrete time signal could be perfectly reconstructed using the inverse DFT (IDFT) synthesis equation. In addition, in the case when enough samples are chosen, the plot of these frequency samples will be a good representation of the actual DTFT. For computational efficiency, the best set of frequency samples is the set of equally spaced points in the interval $0 \leq \omega \leq 2\pi$ given by $\omega_k = \frac{2\pi k}{N}$ for $k = 0, 1, \ldots, N-1$. For a signal $x[n]$ which is nonzero only for $0 \leq n \leq M-1$ the DFT is defined by,

$$X[k] = \sum_{n=0}^{M-1} x[n]e^{-j2\pi kn/N}, \quad k = 0, 1, \ldots, N-1$$

The Inverse DFT (IDFT) is given by,

$$x[n] = \sum_{k=0}^{N-1} x[n] e^{j2\pi kn/N}, \quad n = 0, 1, \ldots, M-1$$

The function `fft` implements the DFT operation in a computationally efficient manner. FFT stands for Fast Fourier Transform and will be studied in greater details towards the end of the course. If `x` is a vector containing $x[n]$ for $0 \leq n \leq M-1$ and $N \geq M$, then `X=fft(x,N)` computes N evenly spaced samples of the DTFT of `x` and stores these samples in the vector `X`. If $N < M$, then the MATLAB function `fft` truncates `x` to its first N samples before computing the DTFT, thus yielding incorrect values for the samples of the DTFT. The function `ifft` can be used to compute the IDFT efficiently. Also note that the `fft` function computes the DFT in the interval $[0, 2\pi]$. To reorder the samples in the interval $[-\pi, \pi]$ you can use the function `fftshift`.

4 Example

Consider the following signal,

1.
$$x[n] = u[n] - u[n-11]$$

   - What is the DTFT of the Signal?
   - Compute the DFT using the `fft` function. Plot the magnitude of the spectrum.
   - One can improve the resolution of the DFT by *zero padding* the input and then computing its DFT. This can be done using the `fft(x,N)` command by choosing $N > M$, where M is the sequence length. Choose $N = 100, 500, 1000$. Does the DFT look similar to the DTFT?

2. A MATLAB code to plot the DFT

```
N = 8;
n = 0:(N-1);
x = (0.7).^n;
k = 0:(N-1);
Xdft_8 = fft(x,N);
mag_Xdft_8 = abs(Xdft_8);
phase_Xdft_8 = angle(Xdft_8);
figure, subplot(2,1,1);
stem(k, mag_Xdft_8), grid on;
title(‘‘Magnitude of 8-point DFT of x[n]’’), xlabel(‘‘k’’), ylabel(‘‘| X[k]|’’);
xlim([-0.2 8.2]);
subplot(2,1,2);
stem(k, phase_Xdft_8), grid on;
title(‘‘Phase of 8-point DFT of x[n]’’), xlabel(‘‘k’’), ylabel(‘‘< X[k] (radians)’’);
xlim([-0.2 8.2]);
```

5 Homework - Due 06/28/2011 at 5:00 PM

1. Write a function called `func_MyDFT(x,N)` to compute the DFT of the sequence `x`. Note N is the DFT length which may be different from the sequence length M. If $N > M$ the function should *zero pad* the input before computing

the DFT. If $N < M$ then the input must be truncated. Test your function by comparing the the output of your function with that obtained by `fft`. You can use `randn` function to generate the test vectors.

2. Let $x[n]$ be a discrete time sequence:

$$x[n] = \begin{cases} (0.7)^n, & 0 \le n \le 7 \\ 0, & else \end{cases}$$

   (a) Determine the analytical expression for the DTFT of $x[n]$ and plot the magnitude and phase of the DTFT.

   (b) Compute in MATLAB the 8-point DFT of $x[n]$, $0 \le n \le 7$ using the DFT function you wrote in Problem 1. Plot the magnitude and phase. You can use the `stem`, `abs`, and `angle` commands.

   (c) Compute, in MATLAB, the 16-point DFT of $x[n]$, $0 \le n \le 15$ and stem plot its magnitude and phase. This can be accomplished by modifying the commands provided in part (b). Comment on the effect of zero-padding the signal on its DFT.

   (d) Compute, in MATLAB, the 128-point DFT of $x[n]$, $0 \le n \le 127$ and plot its magnitude and phase. Note: the `plot` command is used instead of stem when many dense points exist to avoid appearance of a black blob.

   (e) Compare the results from part (d) to the plots of part (a). How does this relate to the relationship between digital frequency ω and DFT index k?

3 . Consider the following two sequences:

$$x[n] = [1\ 1\ 1\ 1]$$
$$y[n] = [1\ 1\ 1\ 1]$$

   (a) Perform the linear convolution of the two sequences using the `conv` command. Plot the result using `stem`.

   (b) Analytically perform the cyclic convolution of the two sequences.

   (c) Compute the DFT of $x[n]$ and $y[n]$. Store the result in vectors `Xk` and `Yk`. Now perform a *point-by-point* product of `Xk` and `Yk`. i.e. compute `Zk = Xk.* Yk`. Take the inverse DFT of `Zk`. In case you get complex-valued outputs, extract the real part using the `real` command. Verify that your result agrees with the result obtained in Part (b). Plot your result using `stem`. How many samples of the result agree with the result in Part (a).

   (d) Increase the length of $x[n]$ and $y[n]$ by zero-padding the sequences. First add one zero to both the sequences. Repeat the steps in Part (b). How many samples agree with the result in Part (a). How many zeros will be needed to obtain the result in Part (a).

   (e) Consider the following two sequences:

$$x[n] = [-3\ 5\ 8\ 6\ 2\ 2]$$
$$y[n] = [1\ 1\ 4\ 2]$$

   Repeat parts (a) through (d) for this new case. Can you generalize the results for any vector `x` of length `L` and any vector `y` of length `M` (i.e. how many zeros should you add to each of `x` and `y` so that the output of circular convolution is equal to that of linear convolution)?

4. Download the file “tones.mat” from the course web page. The file contains the a signal which has multiple tones in it. Load the signal using the following commands,

```
s = load(tones.mat);
x = s.y1;
```

The variable `x` should now contain the signal.

(a) Compute the DFT of `x` using a transform length $N = 25$. Plot the magnitude of the DFT using the `plot` command. How many distinct frequencies do you see?

(b) Experiment with the sequence length (by adding a different number of zeros). Compute the DFT for these different sequence lengths obtained by zero-padding. Can you find more frequencies? How many tones can you distinguish and what are their values?

5. Consider the following signal

$$x(t) = \begin{cases} cos(2\pi * 100t) + cos(2\pi * 102t), & 0 \leq t \leq 0.3 \text{ sec} \\ cos(2\pi * 103t) + cos(2\pi * 105t), & 0.3 \leq t \leq 0.9 \text{ sec} \end{cases}$$

Note that the signal not only has different frequencies in it but also that the frequency content of the signal changes with time. Assume that we are sampling the signal at 1 KHz, that is $F_s = 1000$ Hz to obtain a discrete time signal $x[n]$ given below:

$$x[n] = \begin{cases} cos(0.2\pi n) + cos(0.204\pi n), & 0 \leq n \leq 300 \\ cos(0.206\pi n) + cos(0.210\pi n), & 300 < n \leq 900 \end{cases}$$

In this problem, we would like to find out the frequency content of this signal as accurately as possible. We will do this by applying a windowing operation to the signal.

(a) First let us look at the effects of DFT sampling on two window functions $w_1[n]$ and $w_2[n]$ given below:

$$w_1[n] = \begin{cases} 1, & n = 0, 1, \ldots, 300 \\ 0, & else \end{cases}$$

$$w_1[n] = \begin{cases} 1, & n = 0, 1, \ldots, 600 \\ 0, & else \end{cases}$$

i. Compute the $M = 900$ point DFT of the window functions $w_1[n]$, $w_2[n]$. Plot the magnitude of the DFT's. How do these compare (comment and explain analytically why you got the following plots)? You may plot only the first 100 samples of the spectrum to give a clearer picture.

ii. Compute the $M = 1800$ point DFT of the window functions $w_1[n]$, $w_2[n]$ and plot the magnitude of the computed DFT's. Do you see any differences compared to Part i ? Again, you may plot only the first 100 samples of the spectrum.

(b) We will now try to perform spectral estimation of the signal $x[n]$. We will do this by windowing the signal. Let us first consider the signal in the interval $0 \leq n \leq 300$. In this period the signal does not change its frequency content.

i. Multiply $x[n]$ by the window function $w_1[n]$ to obtain $x_1[n] = x[n]w_1[n]$ (Note: this is equivalent to extracting the first 300 samples of $x[n]$). Compute the $N = 300$ point DFT of $x_1[n]$. Plot the magnitude. Comment on the frequency content.

ii. Next multiply $x[n]$ by the shifted window function to obtain $x_2[n] = x[n]w_1[n - 300]$. Find the $N = 300$ point DFT of $x_2[n]$. Comment on its frequency content.

Are the results of this exercise satisfactory?

(c) Let us now use the window $w_2[n]$ to perform the spectral estimation. Multiply $x[n]$ by $w_2[n]$ to obtain $x_3[n] = x[n]w_2[n]$. Compute the DFT of $x_3[n]$ and plot its magnitude. Did the frequency resolution improve?

6. In this exercise we will consider a simple example for using the `spectrogram` function in MATLAB. Download the file `mtlb.mat` file from the course web page. This file contains a speech signal of duration 4001 samples sampled at 7418 Hz. We will compute the Short Time Fourier Transform (STFT) of this signal using the `spectrogram` function.

   (a) First load the file using load as shown below:

   ```
   load mtlb;
   ```

   Now plot the speech signal using the plot command.

   (b) We will use the following form of the `spectrogram` function:

   ```
   spectrogram(x,len,overlap,N,Fs,axis)
   ```

   here

   ```
   x   = vector that hold input signal
   len = an integer value here denotes the window length
   overlap = number of samples by which sections overlap
   N = The DFT length
   Fs = Sampling frequency
   axis = The axis to be used for the frequency. You can specify axis as 'y-axis'
   ```

   Generate the spectrogram of the speech signal for window length of 256, time lapse between blocks of 35, and transform length of 512 using the spectrogram command. Comment on your results.

Deliverables

- Email your code, figures, calculation and answers as a *.pdf* or *.doc* file to **ece311lab.uiuc@gmail.com**. Be sure to name your document in the form- **ECE311Lab2_firstname_lastname.doc/pdf**.
- Late reports will reduce the grade by 20% per day.
- Make sure to present a clear and concise report having figures labeled and centered.
- **Reminder: Homework is due on 06/28/2011**