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

ECE 311: Digital Signal Processing Lab

LAB 2: **Frequency Representation and Spectral Analysis**
Fall 2015

1 Objective

In this lab you will perform spectral analysis of discrete-time signals using MATLAB. The objective is to help you learn how to interpret and analyze digital signals in the frequency domain.
Key concepts:

- Spectral shape
- Spectral resolution
- Zero-padding effects on DFT spectral analysis
- Magnitude spectrum
- Phase spectrum

2 Background

2.1 The Discrete Time Fourier Transform (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 gven by,

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

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 that the DTFT $X_d(\omega)$ is periodic with period 2π. However, the DTFT is not practical for numerical implementation because:

1. $x[n]$ has an infinite number of samples
2. $X_d(\omega)$ is a continuous function of ω.

2.2 The Discrete Fourier transform (DFT) and the Inverse DFT (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. Also, $X_d(\omega)$ can be computed only at a discrete set of frequency samples, w_k. For computational efficiency, the 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 $x[n]$ that is nonzero only for $0 \leq n \leq N-1$, the DFT is defined by

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-j\frac{2\pi kn}{N}}, k = 0, 1, \ldots, N-1.$$

The Inverse DFT (IDFT) is given by

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]e^{j\frac{2\pi kn}{N}}, n = 0, 1, \ldots, N-1.$$

2.3 The Fast Fourier Transform (FFT) and Inverse FFT (IFFT)

The Fast Fourier Transform and implements the DFT in a computationally efficient manner. It is implemented in MATLAB by the function `fft.m`. The function `fft.m` is primarily used in two ways:

- `X = fft(x);` returns the DFT of `x` that has the same length as the input vector `x`.
- `X = fft(x,n);` returns the n-point DFT of `x`.

NOTE: The FFT computes the DFT on the interval$[0, 2\pi]$. The samples can be reordered to the interval $[-\pi, \pi]$ by the function `fftshift.m`.

Similarly, the inverse Fourier transform is implemented in a computationally efficient manner by the function `ifft.m`. It is also primarily used in two ways:

- `X = ifft(x);` returns the IDFT of `x` that has the same length as the input vector `x`.
- `X = ifft(x,n);` returns the n-point IDFT of `x`.

3 Lab Demo

3.1 DTFT and DFT of a cosine signal

a. Consider a discrete cosine signal of infinite length

$$x[n] = \cos(\Omega_0 Tn),$$

where T is the sampling period. The DTFT of $x[n]$ on the interval $[-\pi, \pi]$ is given by

$$X_d(\omega) = \pi(\delta(\omega - \Omega_0 T) + \delta(\omega + \Omega_0 T)).$$

Outside this range the DTFT repeats periodically as discussed in section 2.1. The above expression may be proved by considering the inverse DTFT:

$$\begin{aligned} x[n] &= \frac{1}{2\pi}\int_{-\pi}^{\pi} X_d(\omega)e^{j\omega n}d\omega \\ &= \frac{1}{2}\left(e^{j\Omega_0 Tn} + e^{-j\Omega_0 Tn}\right) \\ &= \cos(\Omega_0 Tn). \end{aligned}$$

b. Consider a discrete cosine signal of finite length

$$x[n] = cos(\Omega_0 nT), n = 0, 1, \ldots, N-1.$$

The DFT is given by

$$
\begin{aligned}
X(\omega) &= \sum_{n=0}^{N-1} \cos(\Omega_0 nT) e^{-j\omega n} \\
&= \frac{1}{2} \sum_{n=0}^{N-1} \left(e^{j\Omega_0 Tn} + e^{-j\Omega_0 Tn} \right) e^{-j\omega n} \\
&= \frac{1}{2} \sum_{n=0}^{N-1} \left(e^{-j(\omega-\Omega_0 T)n} + e^{-j(\omega+\Omega_0 T)n} \right) \\
&= \frac{1}{2} e^{-j(\omega-\Omega_0 T)\frac{N-1}{2}} \frac{\sin((\omega-\Omega_0 T)\frac{N}{2})}{\sin((\omega-\Omega_0 T)\frac{1}{2})} + \frac{1}{2} e^{-j(\omega+\Omega_0 T)\frac{N-1}{2}} \frac{\sin((\omega+\Omega_0 T)\frac{N}{2})}{\sin((\omega+\Omega_0 T)\frac{1}{2})},
\end{aligned}
$$

where $\omega = \frac{2\pi k}{N}, k = 0, 1, \ldots, N-1$.

c. Compute the 16-point DFT of the cosine signal using the `fft.m` function of MATLAB.

$$x[n] = \cos(\Omega_0 Tn), n = 0, 1, \ldots, N-1,$$

where $\Omega_0 = \frac{75\pi}{4}, T = \frac{1}{50}$, and $N = 16$.

d. Compare the output of the `fft.m` function with the analytical solution in (b). **Hint:** Consider the `diric` function.

3.2 Dual-tone multi-frequency (DTMF) signal detection

DTFM signaling is the basis of modern voice communications. It is used not only to dial numbers and configure switchboards, but it is also utilized in systems such as voicemail, telephone banking, and automated customer service.

Figure 1 shows a chart of the frequencies used in a modern telephone pad. Each button has a signal containing two frequency components, the combination of row frequency and the column frequency on the chart.

	1209 Hz	1336 Hz	1477 Hz
697 Hz	1	ABC 2	DEF 3
770 Hz	GHI 4	JKL 5	MNO 6
852 Hz	PRS 7	TUV 8	WXY 9
941 Hz	*	0	#

Figure 1: Frequencies of a telephone pad

Figure 2 shows the time-domain signal waveforms of each of the buttons on the keypad, while Fig. 3 shows the DFT spectra. While the waveforms of the time-domain signals look similar to each other, the frequency representations are easily distinguishable, making it simple to detect DTMF signals in the frequency domain.

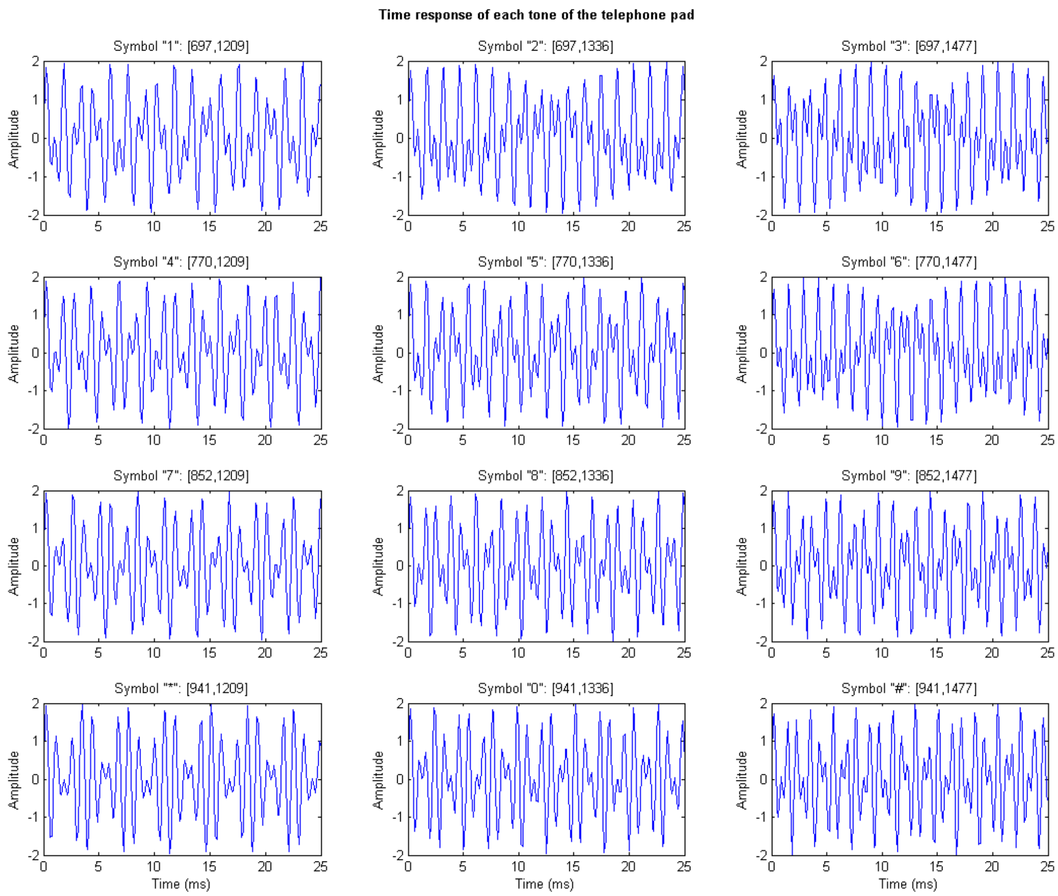


Figure 2: Time-domain signal waveforms corresponding to telephone buttons.

3.3 Nuclear Magnetic Resonance (NMR) spectrum analysis

An NMR spectrum can reveal important chemical, structural, biological information about a sample in an *in vivo* experiment. Figure 4a shows the magnitude of a 128-point time-domain NMR signal. Figure 4b shows the 128-point DFT spectrum of the signal, where a water peak around 0 Hz, and a N-actyleaspartate (NAA) peak around 330 Hz are visible. Zero-padding can improve the visual quality of a DFT spectrum. As shown in Fig. 4c, the 1024-point DFT spectrum of the same signal reveals more peaks, notably a creatine peak around 209 Hz and choline peak around 185 Hz.
However, **zero-padding cannot improve the spectral resolution of the DFT spectrum**, which is limited by the signal length. As shown in Fig. 5a to Fig. 5c, zero-padding can not resolve the creatine and choline peaks, due to poor spectral resolution of the 32-point signal.

3.4 Magnitude and phase spectrum

Just like a 1D signal, multidimensional signals (e.g., an image) can also be represented and analyzed in frequency domain using a multidimensional DFT. Figures 6b and 6c show the magnitude (on a logarithmic scale) and phase spectrum of the image shown in Fig. 6a. The magnitude spectrum in Fig. 6b shows

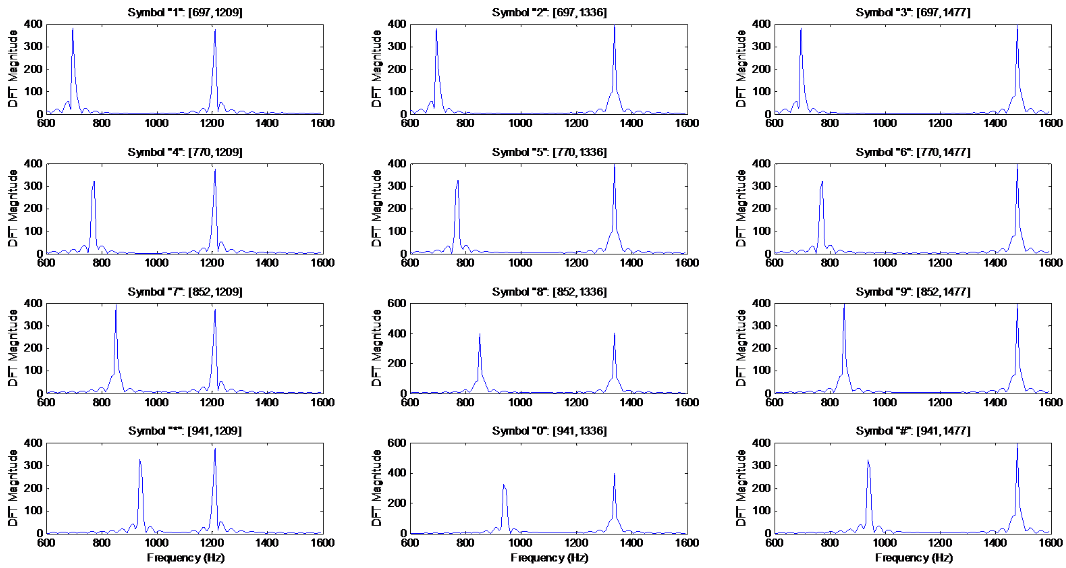
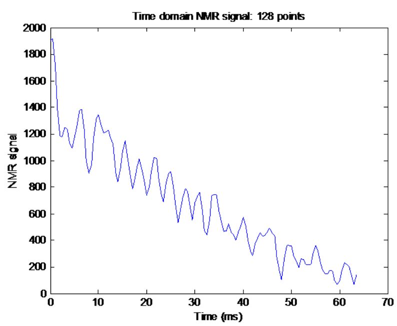
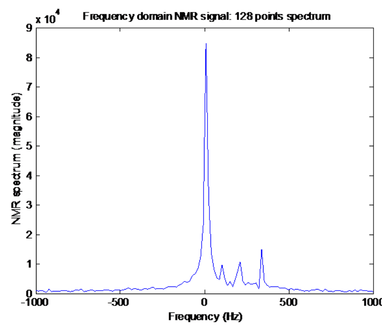


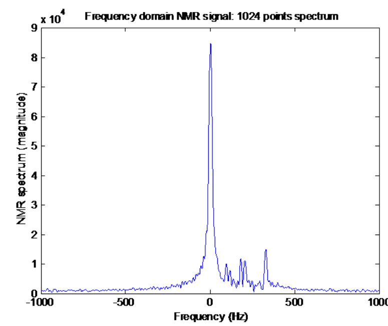
Figure 3: DFT spectra corresponding to telephone buttons.



(a)

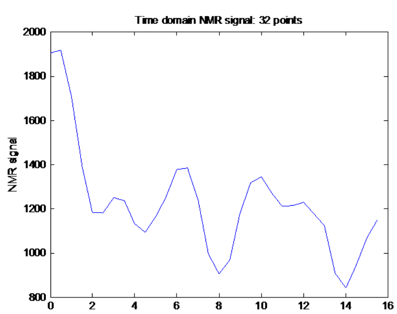


(b)

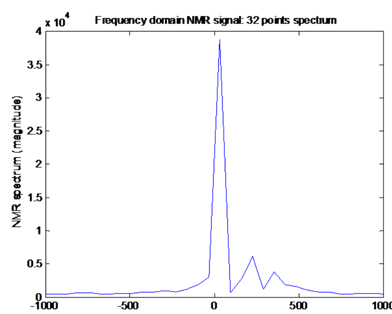


(c)

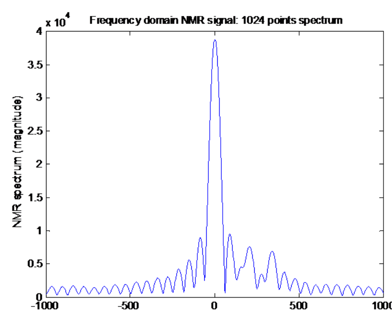
Figure 4: (a) Magnitude plot of a 128-point time domain NMR signal, (b) 128-point DFT spectrum of the signal in (a), (c) 1024-point DFT (zero-padded) spectrum of the signal in (a).



(a)



(b)

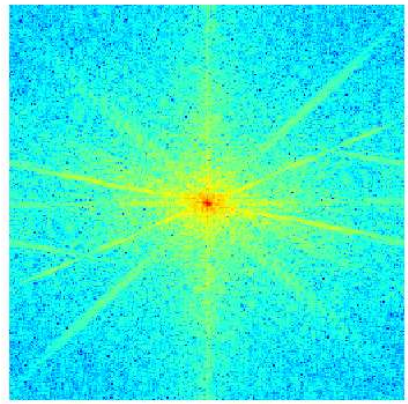


(c)

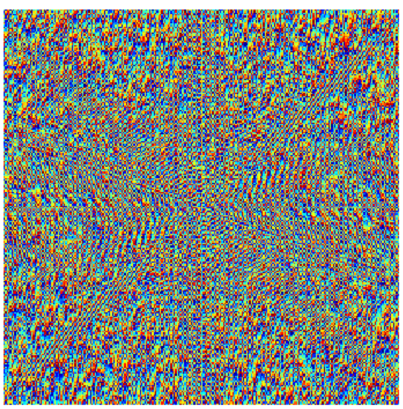
Figure 5: (a) Magnitude plot of a 32-point time domain NMR signal, (b) 128-point DFT spectrum of the signal in (a), (c) 1024-point DFT (zero-padded) spectrum of the signal in (a).

that most of the energy of the image is located in the low frequency region. The phase spectrum is given in Fig. 6c. Figure 6d shows a reconstructed image when the phase spectrum is corrupted by random noise.

(a)



(b)



(c)

(d)

Figure 6: (a) Original image, (b) magnitude spectrum, (c) phase spectrum, (d) image with corrupted phase.

4 Lab Report - Due 10/6/2015 at 5:00 PM (Submit Online)

1. Write a function called `func_MyDFT(x,M)` to compute the DFT of the sequence `x`. Note `M` is the DFT length which may be different from the sequence length N. If $M > N$ the function should *zero pad* the input before computing the DFT. If $M < N$ then the input must be truncated. Test your function by comparing the output of your function with that obtained by `fft.m`. You can use the `randn.m` function to generate the test vectors.

2. Consider the following signal

$$x[n] = [-3, 8, 8, 12, -3, 12, 8, 8]$$

   (a) Compute the 8-point DFT of x[n]. Plot the magnitude and the phase using the `stem` function (also for the other questions of this problem). Is the DFT real? Why?

   For parts (b)-(e), plot the magnitude and the phase of the DFT described, using the `stem` function. Compared with part (a), what do you observe? Explain your observation in terms of the DFT and its properties.

   (b) Shift (circular shift) the signal by 3, compute the 8-point DFT of the shifted signal.

   (c) Shift (circular shift) the signal by -3, compute the 8-point DFT of the shifted signal.

   (d) Compute the 16-point DFT of the signal using zero padding.

   (e) Compute the 32-point DFT of the zero interpolated signal

$$x[n] = [-3, 0, 0, 8, 0, 0, 8, 0, 0, 12, 0, 0, -3, 0, 0, 12, 0, 0, 8, 0, 0, 8, 0, 0].$$

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

$$x[n] = \begin{cases} (0.8)^n, & \text{if } 0 \le n \le 8 \\ 0, & \text{else} \end{cases}$$

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

   (b) In MATLAB, compute the 9-point DFT of $x[n], 0 \le n \le 8$ using the DFT function you wrote in Problem 1. Plot the magnitude and phase using the `stem` function.

   (c) Compute the 16-point DFT of $x[n], 0 \le n \le 15$ using the DFT function you wrote in Problem 1. Plot the magnitude and phase using the `stem` function. Comment on the effect of zero-padding the signal on its DFT.

   (d) Compute the 128-point DFT of $x[n], 0 \le n \le 127$ using the DFT function you wrote in Problem 1. Plot the magnitude and phase using the `plot` function instead of `stem` when many dense points exist, to avoid the appearance of a solid blue area.

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

4. Download the file `tones.mat` from the course web page (this should be included in the zipped assignment file for lab 2). The file contains a signal that has multiple tones in it. Load the signal using the command `load tones;`. The variable `x` should now contain the signal.

(a) Truncate the length of `x` to be 16. Compute the 16-point DFT of `x`. Plot the magnitude of the DFT using the plot command. How many distinct tone frequencies do you see?

(b) Try to improve the resolution of DFT spectrum by using zero padding on the length-16 signal. Can you find more tone frequencies? How many tones can you distinguish and what are their values?

(c) Compute the DFT of `x` without truncation of `x` ($M = N = 36$). Try to improve the resolution of the DFT spectrum by using zero padding. Can you find more tone frequencies? How many tones can you distinguish and what are their values?

(d) When we truncate the signal `x`, we are using a rectangular, or 'boxcar' window. Try at least two other windowing methods, such as the Hamming and Hanning (Hann) windows. These windows may be applied using the `hamming.m` and `hann.m` functions as follows

```
L = length(x);
x_windowed = hamming(L).*x;
```

Compute the DFT of `x_windowed` with and without zero padding. How many tones can you distinguish and what are their values? Comment on the differences between results obtained using various windowing methods, and explain them in terms of the DTFT of the windows. What are the tradeoffs in using different windows?

5. Finish the following tasks related to the lab demo:

(a) Load the file `DialNumber.mat`. The variable `dialTone` records a piece of 10-digit dialing signal, which is sampled at $F_s = 1/T = 8000$ samples/s. Calculate the DFT spectrum of every 100-ms segment of the signal. What is the dialed telephone number? Calculate and display the short-time Fourier transform of the signal using the following commands:

```
figure;
windowLength = 128; noverlap = 100;
spectrogram(dialTone,windowLength,noverlap,512,8000);
```

(b) Load the file `NMRSpec.mat`. The variable `st` contains an NMR signal. The data-sampling rate is $F_s = 1/T = 2000$ samples/s. Calculate the DFT spectrum of the signal. Plot the real and the imaginary parts of the spectrum. Plot the magnitude and the phase of the spectrum. Calculate the 32-point DFT spectrum of the first 32 samples of the signal. Can you distinguish the peaks corresponding to creatine (around 209 Hz) and choline (around 185 Hz)? Will zero-padding help you out? Why?

(c) Load the image `cameraman.tif` by calling `Im = imread('cameraman.tif');`. You may view the image using `imagesc(Im)`. Calculate the magnitude spectrum and the phase spectrum of the image.

(i) Add a linear phase along each row to the original phase spectrum (add a phase term changing linearly with the column index). Reconstruct the image using the magnitude and the new phase spectrum. What has changed about the image? Can you explain this change in terms of the DFT?

(ii) Add a linear phase along each column to the original phase spectrum (add a phase term changing linearly with the row index). Reconstruct the image using the magnitude and the new phase spectrum. What has changed about the image? Can you explain this change in terms of the DFT?

You can find useful code in the MATLAB file `demo_MagnPhSpec.m`. Add enough linear phase to the original phase spectrum that you can see a distinct effect.