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

ECE 311: Digital Signal Processing Lab

LAB 3: **Sampling Effects and Windowing Effects**
Fall 2015

---

1 Objective

The objective is to learn the sampling effects and the windowing effects in DFT spectral analysis.
Key concepts:

- Sampling effects
- Windows and windowing effects

2 Background

2.1 A continuous time signal and its CTFT

A continuous time-domain signal $x_a(t)$ and its CTFT are related by the Fourier transform

$$X_a(\Omega) = \int_{-\infty}^{\infty} x_a(t)e^{-j\Omega t}dt$$

2.2 Sampling Effects

Mathematically, the discrete-time signal $x[n]$ is related to the continuous time-domain signal $x_a(t)$ through $x[n] = x_a(nT)$, where T is the sampling period.

$$\begin{array}{c} x_a(t) \rightarrow A/D \rightarrow x[n] \\ \uparrow T \end{array}$$

The relationship between $X_a(\Omega)$, the CTFT of $x_a(t)$, and $X_d(\omega)$, the DTFT of $x[n]$ is given by

$$X_d(\omega) = \frac{1}{T}\sum_{k=-\infty}^{\infty} X_a\left(\frac{\omega+2\pi k}{T}\right)$$

Suppose $x_a(t)$ is band-limited, *i.e.*, $X_a(\Omega) = 0$, $|\Omega| > B$, then if $T < \frac{\pi}{B}$ (Nyquist rate),

$$X_d(\omega) = \frac{1}{T}X_a\left(\frac{\omega}{T}\right), |\omega| < \pi$$
$$x_a(t) = \sum_{n=-\infty}^{\infty} x[n]\text{sinc}\left(\frac{\pi}{T}(t - nT)\right)$$

2.3 Windowing

Since DFT is defined for finite-length signals. A hypothetical discrete-time infinitely long signal $x[n]$ must be truncated before the DFT is applied. The truncation operation, or more generally, windowing operation, is defined as

$$\hat{x}[n] = x[n]w[n],$$

where $w[n]$ is a pre-selected window function, and $\hat{x}[n]$ is the windowed time signal.
Some common windows are:

- Rectangular window (truncation):

$$w[n] = \begin{cases} 1, & n = 0, 1, \cdots, N-1, \\ 0, & \text{otherwise.} \end{cases}$$

- Triangular window (Bartlett window):

$$w[n] = \begin{cases} \frac{2n}{N}, & n = 0, 1, \cdots, \frac{N}{2}, \\ w[N-n], & n = \frac{N}{2}+1, \cdots, N-1, \\ 0, & \text{otherwise.} \end{cases}$$

- Hamming window

$$w[n] = \begin{cases} 0.54 - 0.46\cos(\frac{2\pi n}{N}), & n = 0, 1, \cdots, N-1, \\ 0, & \text{otherwise.} \end{cases}$$

- Hanning window

$$w[n] = \begin{cases} 0.5 - 0.5\cos(\frac{2\pi n}{N}), & n = 0, 1, \cdots, N-1, \\ 0, & \text{otherwise.} \end{cases}$$

- Kaiser window

$$w[n] = \begin{cases} I_0\Big[\beta\big(1 - ((n-\frac{N}{2})/\frac{N}{2})^2\big)^{1/2}\Big], & n = 0, 1, \cdots, N-1, \\ 0, & \text{otherwise.} \end{cases}$$

  where I_0 is the zeroth-order modified Bessel function of the first kind:

$$I_0(x) = \frac{1}{\pi}\int_0^{\pi} \cosh(x\cos\theta)d\theta.$$

  The choice of β affects the tradeoff between the main lobe width and side lobe heights.

2.4 Windowing Effects

Suppose the DTFT of a window is given by

$$W_d(\omega) = \sum_{n=0}^{N-1} w[n]e^{-j\omega n}.$$

Then $\hat{X}_d(\omega)$ (the DTFT of $\hat{x}[n]$) is related to $X_d(\omega)$ (the DTFT of $x[n]$) by the convolution (denoted $\otimes$)

$$\begin{aligned} \hat{X}_d(\omega) &= X_d(\omega) \otimes W_d(\omega) \\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi} X_d(\upsilon)W_d(\omega-\upsilon)d\upsilon. \end{aligned}$$

The DFT of $\hat{x}[n]$ is given by $\hat{X}[k] = \hat{X}_d(\omega)\big|_{\omega=\frac{2\pi k}{N}}$.

2.5 Example: Rectangular window (time truncation) of a sinusoid

Given a signal $x_a(t) = \cos(\Omega_0 t)$, its CTFT is given by

$$X_a(\Omega) = \pi\delta(\Omega - \Omega_0) + \pi\delta(\Omega + \Omega_0).$$

After sampling the signal with sampling rate T, the discrete signal is $x[n] = \cos(\Omega_0 T n)$ and its DTFT is given by

$$\begin{aligned} X_d(\omega) &= \sum_{n=-\infty}^{\infty} \cos(\Omega_0 n T) e^{-j\omega n} \\ &= \pi\delta(\omega - \Omega_0 T) + \pi\delta(\omega + \Omega_0 T),\ \ |\omega| < \pi. \end{aligned}$$

Consider a rectangular window $w[n]$ of length N, its DTFT is given by

$$W_d(\omega) = \textstyle\sum_{n=0}^{N-1} e^{-j\omega n} = e^{-j\frac{(N-1)\omega}{2}} \frac{\sin(N\omega/2)}{\sin(\omega/2)},\ \ |\omega| < \pi$$

Then the DTFT of $\hat{x}[n] = x[n]w[n]$ is given by

$$\begin{aligned} \hat{X}_d(\omega) &= X_d(\omega) \otimes W_d(\omega) \\ &= \frac{1}{2} e^{-j(\omega - \Omega_0 T)\frac{(N-1)}{2}} \frac{\sin((\omega - \Omega_0 T)\frac{N}{2})}{\sin((\omega - \Omega_0 T)\frac{1}{2})} + \frac{1}{2} e^{-j(\omega + \Omega_0 T)\frac{(N-1)}{2}} \frac{\sin((\omega + \Omega_0 T)\frac{N}{2})}{\sin((\omega + \Omega_0 T)\frac{1}{2})} \end{aligned}$$

The DFT of $\hat{x}[n]$ is given by

$$\hat{X}[k] = \hat{X}_d(\omega)\big|_{\omega = \frac{2\pi k}{N}}, k = 0, 1, \ldots, N-1.$$

2.6 Example: Resolving two sinusoids

Considering $\hat{X}_d(\omega)$ from the previous example, there are nulls ($\hat{X}_d(\omega) = 0$) whenever $\omega = \Omega_0 T \pm 2\pi m/N$, where $m = 1, 2, 3 \ldots N-1$. The main lobe of the sinc function is centered about $m = 0$ (or any m that is a multiple of N). Thus, the main lobe has a width of $\Delta\omega = 4\pi/N$. To resolve two sinusoidal components using this knowledge, we can impose criteria for the overlap of the main lobes.

1. <u>No overlap of main lobes</u>

$$\begin{aligned} \Omega_1 T - \frac{2\pi}{N} &> \Omega_0 T + \frac{2\pi}{N} \\ NT &> \frac{4\pi}{\Omega_1 - \Omega_0} \end{aligned}$$

2. <u>1/2 overlap of main lobes</u>

$$\begin{aligned} \Omega_1 T - \frac{2\pi}{N} &> \Omega_0 T \\ NT &> \frac{2\pi}{\Omega_1 - \Omega_0} \end{aligned}$$

3 Lab Demo

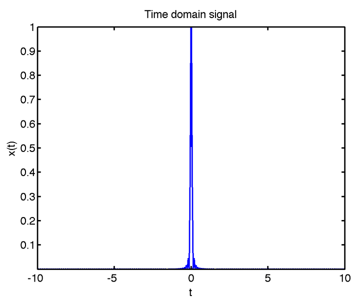
3.1 Sampling effects: A 1D example

Consider a signal $x_a(t) = \text{sinc}^2(f_0 t)$. Its Fourier transform is a triangular function with width $2f_0$. Figure 1a shows $x_a(t)$ when $f_0 = 8$ Hz. Figure 1b to Fig. 1c show the DFT spectra of the discrete signals sampled at Nyquist rate, $2\times$ Nyquist rate, and $0.7\times$ Nyquist rate, respectively. As expected, no aliasing is found in the DFT spectra shown in Fig. 1b and Fig. 1c, and aliasing is indeed obvious in Fig. 1d.

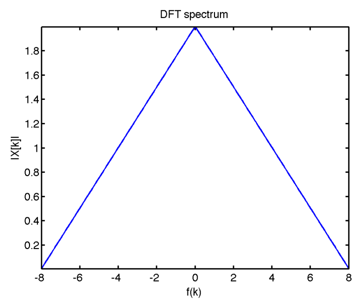
The signal $x_a(t)$ is then modulated by a complex exponential function,

$$x_1(t) = e^{i2\pi f_1 t}\text{sinc}^2(f_0 t),$$

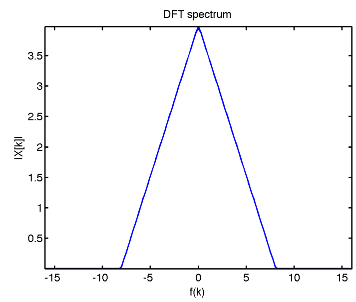
where $f_1 = 4$ Hz. The Fourier transform of $x_1(t)$ is also a triangular function with width $2f_0$ as that of $x_a(t)$ but is shifted to the right by f_1. The modulated signal is then sampled at the Nyquist rate (What is the Nyquist rate for $x_1(t)$?). The corresponding DFT spectrum is shown in Fig. 1e. If the modulated signal is still sampled at a rate of $2f_0$, the corresponding DFT spectrum has aliasing as shown in Fig. 1f. However, this aliasing is not catastrophic. If the bandwidth of the signal and the modulation frequency is known in advance, the DFT spectrum can be recovered correctly.



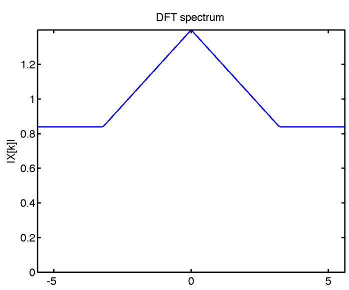
(a)



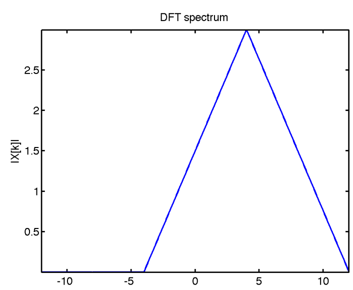
(b)



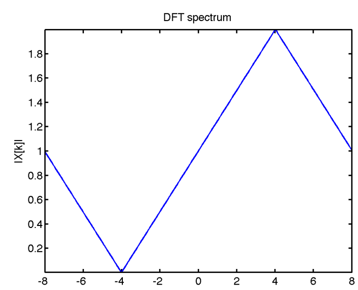
(c)



(d)



(e)



(f)

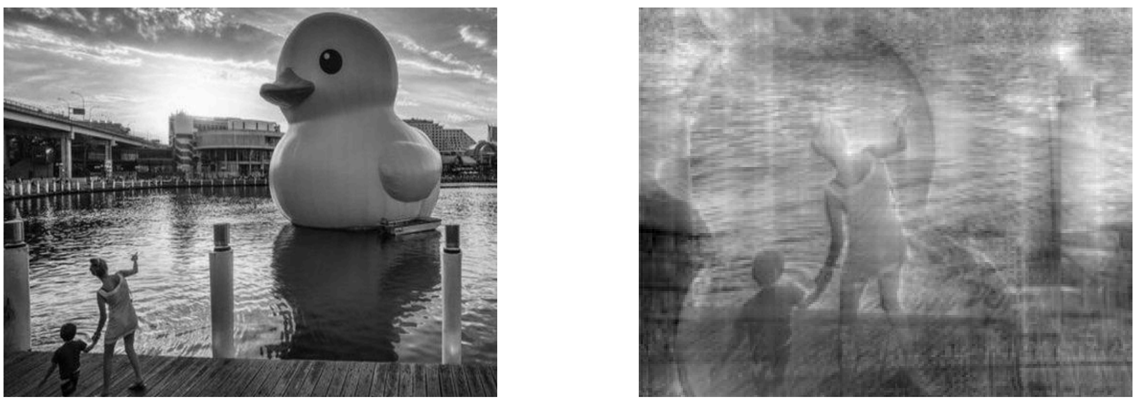
Figure 1: (a) Time-domain signal, (b) DFT spectrum, Nyquist rate, (c) DFT spectrum, $2\times$ Nyquist rate, (d) DFT spectrum, $0.7\times$ Nyquist rate, (e) DFT spectrum of the modulated signal, Nyquist rate, (f) DFT spectrum of the modulated signal, below Nyquist rate.

3.2 Sampling effects: A 2D example

Shown below are the original image (Fig. 2a) and the reconstructed image (Fig. 2b) after undersampling by a factor of 2 in both the horizontal and vertical directions. The undersampling in this example was conducted by first transforming the image into the frequency-domain and then only taking every other frequency-domain point while discarding the rest. This image was then reconstructed using the 2D inverse DFT This undersampling results in spatial aliasing in both directions of the picture. Notice how parts of the image have been shifted both left-right and up-down.

MATLAB code:

```
L = imread('rubberducky.jpg');
Lk = fft2(L);
Lkxy = Lk(1:2:end, 1:2:end);
kxy = ifft2(Lkxy);figure,
imagesc(abs(kxy));
title('');colormap('gray');
axis image off;
```



(a) (b)

Figure 2: (a) Original image, (b) aliased image.

3.3 Windows

Rectangular windows of different length and the corresponding DFT spectra are shown in Fig. 3. Note that while the main lobe width of the DFT spectrum decreases as the window length increases, the side lobe height remains unchanged.

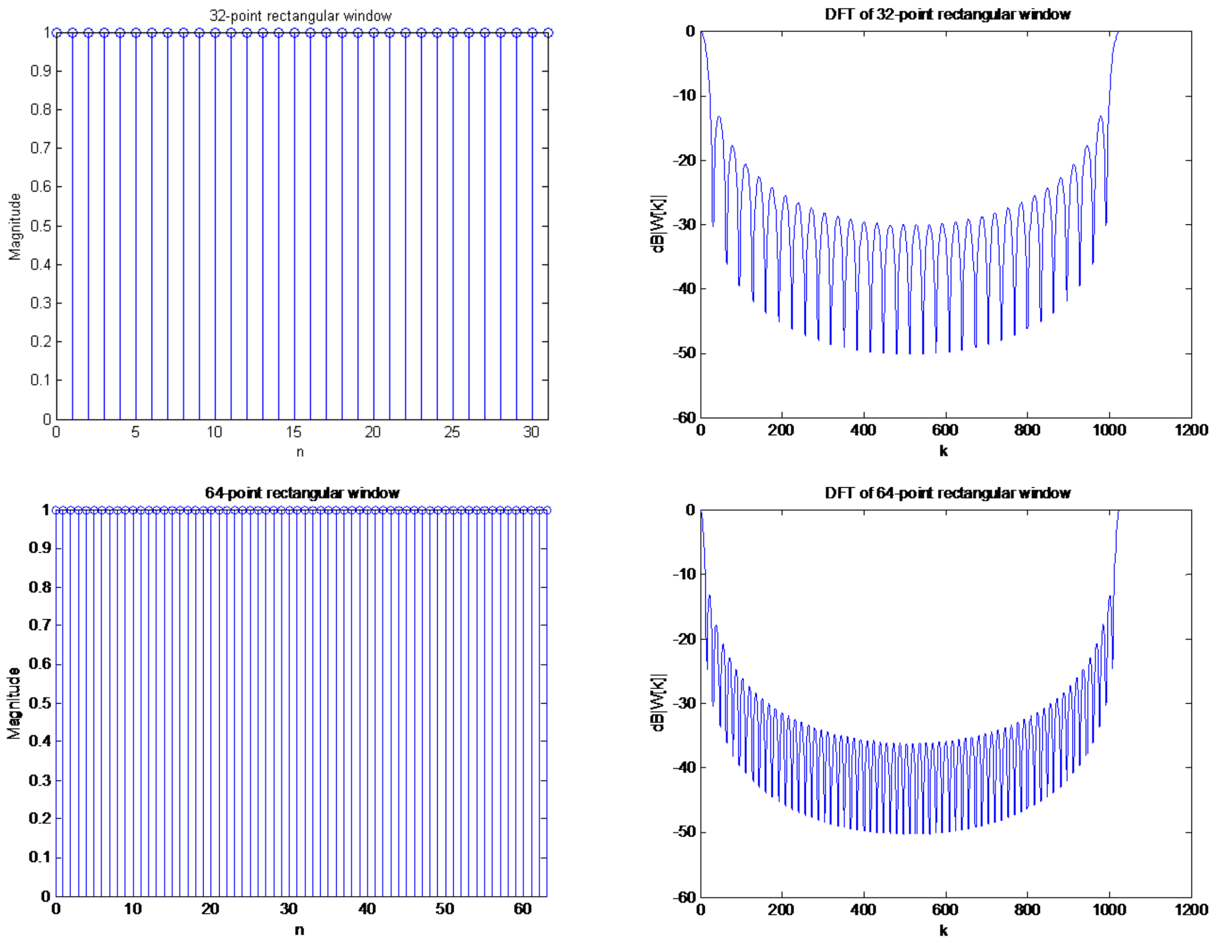


Figure 3: Rectangular window and its DFT spectrum.

Hamming windows of different length and the corresponding DFT spectra are shown in Fig. 4.

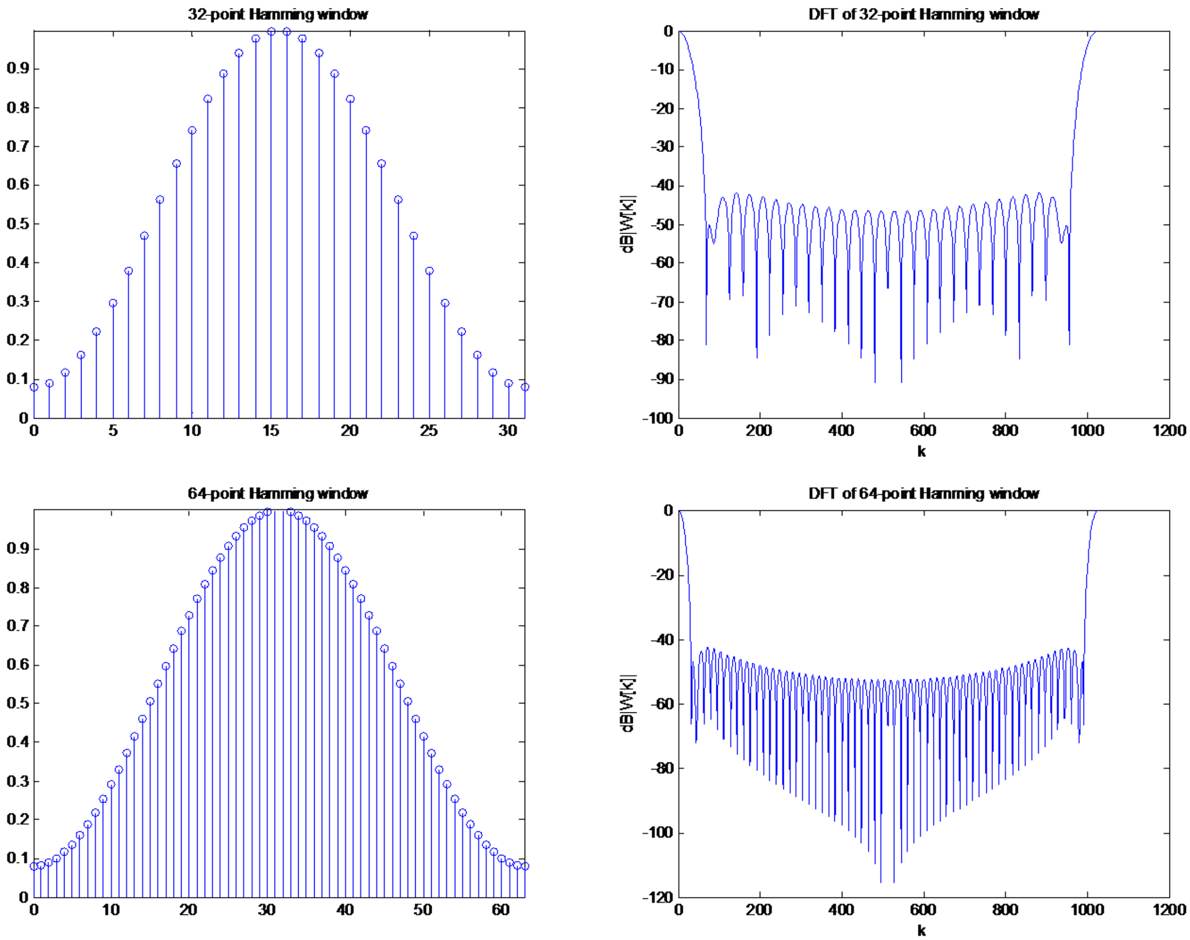


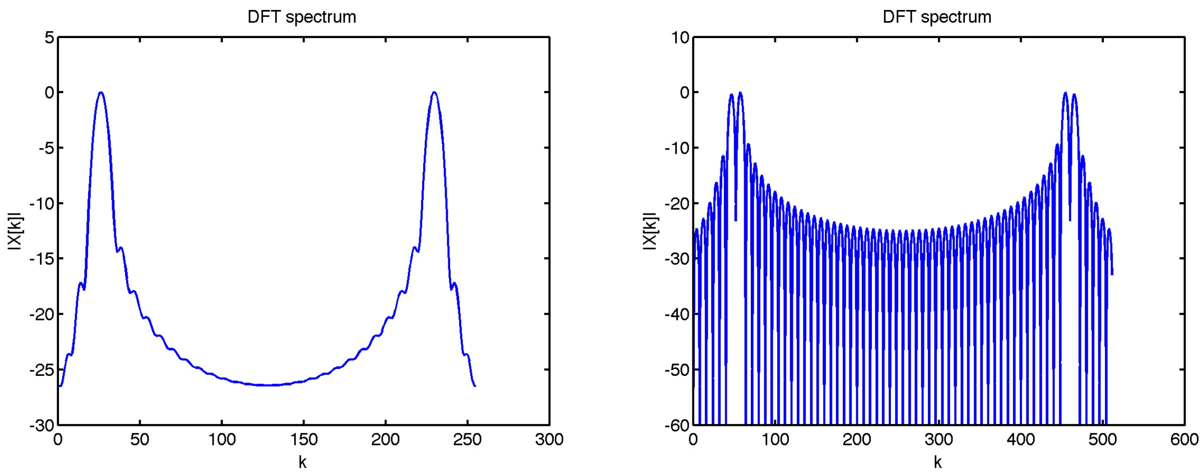
Figure 4: Hamming window and its DFT spectrum.

3.4 Windowing effects: 1D examples

1. Consider a signal

$$x_a(t) = \cos(\tfrac{75}{8}\pi t) + \cos(\tfrac{175}{16}\pi t).$$

The signal is sampled at a sampling rate of 50 samples/s, i.e., $x[n] = x_a(nT), T = 0.02s$. Rectangular windows of different lengths are applied. The corresponding DFT spectra are shown in Fig. 5a and Fig. 5b. The frequency components can be well detected when a 64-point rectangular window is used, but not when a 32-point rectangular window is used. This example illustrates that one can improve spectral resolution by increasing the length of a window.



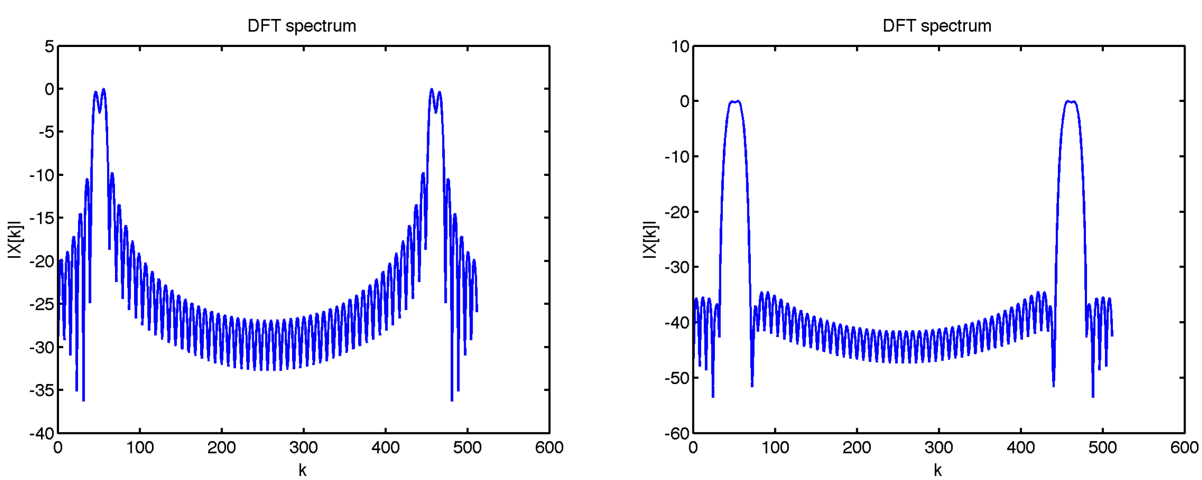
(a) (b)

Figure 5: (a) DFT spectrum with a 32-point rectangular window, (b) DFT spectrum with a 64-point rectangular window.

2. Consider a signal

$$x_a(t) = \cos(\tfrac{75}{8}\pi t) + \cos(\tfrac{675}{64}\pi t).$$

The signal is sampled at a sampling rate of 50 samples/s, i.e., $x[n] = x_a(nT)$, $T = 0.02s$. A rectangular window and a Hamming window, both of length 64, are applied. The corresponding DFT spectra are shown in Fig. 6a and Fig. 6b. The frequency components can be well detected using the rectangular window, but not using the Hamming window. This example illustrates that with the same length, the rectangular window can achieve better spectral resolution than the Hamming window, due to its narrower main lobe width.



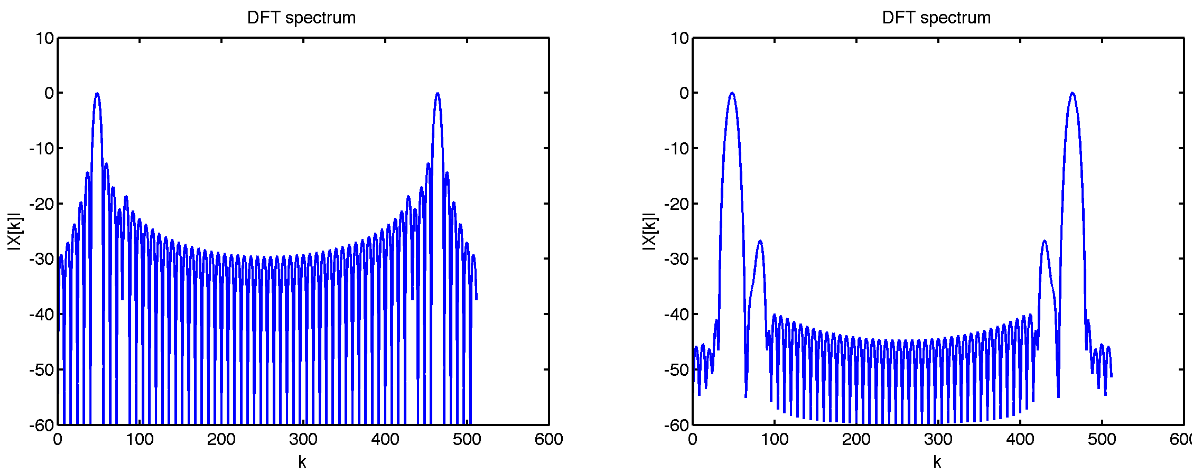
(a) (b)

Figure 6: (a) DFT spectrum with a 64-point rectangular window, (b) DFT spectrum with a 64-point Hamming window.

3. Consider a signal

$$x_a(t) = \cos(\tfrac{75}{8}\pi t) + 10^{-28/20}\cos(\tfrac{125}{8}\pi t).$$

The signal is sampled at a sampling rate of 50 samples/s, i.e., $x[n] = x_a(nT)$, $T = 0.02s$. A rectangular window and a Hamming window, both of length 64, are applied. The corresponding DFT spectra are shown in Fig. 7a and Fig. 7b. The frequency components can be well detected by the Hamming window, but not by the Rectangular window. This example illustrates that, in the presence of a strong signal, the Hamming window can better detect a weak signal than the rectangular window, due to its smaller side lobe height.



(a) (b)

Figure 7: (a) DFT spectrum with a 64-point rectangular window, (b) DFT spectrum with a 64-point Hamming window.

3.5 Windowing effects: A 2D example

Figure 8a shows the famous "Rubbery Ducky" image. Figure 8b show the reconstructed image after truncation in the frequency domain. The truncation is obtained by first performing 2D DFT on the original image to get its spectrum, then taking only the central or lower frequency parts of the spectrum (128×128 out of 512×512), and zero-padding the rest. The image is reconstructed by performing 2D inverse DFT. Note that the truncation is equivalent to performing windowing on the spectrum with a rectangular window. Significant ring artifacts are found in the reconstructed image (Fig. 8b). The ring artifacts can be effectively by applying a Hamming window after truncation in the frequency domain. The reconstructed image is shown in Fig. 8c, where the ring artifacts are reduced at cost of resolution.

(a) (b) (c)

Figure 8: (a) Original image, (b) reconstructed image after truncation in the frequency domain, (c) reconstructed image after truncation and windowing by a hamming window in the frequency domain.

MATLAB code:

```
%load the image
L = imread('rubberducky.jpg');
[rows,cols] = size(L);
rowhalf = ceil(rows/2);
colhalf = ceil(cols/2);
figure;
imagesc(abs(L));
colormap('gray');
axis image off;
% keep the center of the spectrum
Lk = fft2(L);
Lk = fftshift(Lk);
LkZP = zeros(size(L));
LkZP(rowhalf-64:rowhalf+63, colhalf-64:colhalf+63) = Lk(rowhalf-64:rowhalf+63, colhalf-64:colhalf+63);
% reconstruct image
L2 = ifft2(ifftshift(LkZP));
figure, imagesc(abs(L2));
colormap('gray');
axis image off
ham = zeros(1,cols);
ham(colhalf-64:colhalf+63) = hamming(128);
% apply Hamming window along rows
Lkhamx = zeros(rows,cols);
for kk=rowhalf-64:rowhalf+63
    Lkhamx(kk, :) = LkZP(kk, :).*ham;
end
% apply Hamming window along columns
ham = zeros(rows,1);
ham(rowhalf-64:rowhalf+63) = hamming(128);
Lkhamxy = Lkhamx;
for kk = colhalf-64:colhalf+63
    Lkhamxy(:, kk)=LkZP(:,kk).*ham;
end
Lhamxy = ifft2(ifftshift(Lkhamxy));
figure, imagesc(abs(Lhamxy));
colormap('gray');
axis image off;
```

4 Lab Report - Due 10/12/2015 at 5:00 PM (Submit Online)

1. Tradeoffs between T and N

   HINT: This problem is similar to problem 3 of HW 3 from ECE 310

   Let $x_a(t) = \cos(2\pi f_1 t) + 1.25\cos(2\pi f_2 t)$ where $f_1 = 4Hz$ and $f_2 = 7Hz$. In this exercise, we investigate how our choices of the signal length N and sampling period T affect the analysis of the spectrum. We will use a 256-point DFT to get a fine sampling of $X_d(\omega)$ for each choice of N and T.

   (a) Compute and plot the magnitude of the 256-point DFT $X[k]_{k=0}^{M-1}$ for each of the following cases: (i) $N = 64$, $T = 1/30$, (ii) $N = 64$, $T = 1/6$, (iii) $N = 32$, $T = 1/30$. In one case, aliasing is present - which is it?

   (b) For each of the cases in part (a), determine the analog frequencies corresponding to $k = 61$ and $k = 161$.

   HINT: Remember that the DFT is computed over the $[0, 2\pi]$ interval of the DTFT!

   (c) Given that only $N = 128$ samples of $x_a(t)$ are to be acquired, how would you choose T to best resolve the sinusoidal components? Find the value of T which causes the main lobes to overlap by 1/2. Plot the result.

   (d) Given that $T = 1/30$, find the N which causes the main lobes to overlap by 1/2. Also find the minimum $N_{min} < N$ signal length at which you can resolve these two sinusoids, via plotting the DFT in `Matlab`. Plot both results.

2. Study sampling effects

   (a) Consider a signal $x_1(t) = \text{sinc}^2(f_0 t), f_0 = 32Hz$. What is the Nyquist sampling rate for $x_1(t)$? Sample the signal $x_1(t)$ at i) the Nyquist sampling rate, ii) 3 times the Nyquist sampling rate, and iii) 0.8 times the Nyquist sampling rate, for $|t| < 10$. Plot the sampled signals. Plot and compare the resulting DFT spectra.

   HINT: Use the 'sinc' function as defined by Matlab, $sinc(x) = \sin(\pi x)/\pi x$ for $x \neq 0$, and $sinc(x) = 1$ for $x = 0$. The time domain signal $sinc(f_0 t) = sin(\pi f_0 t)/(\pi f_0 t)$ will give you a frequency domain 'rect' function of total width f_0 (cutoff frequency $f_0/2$). Recall that the fourier transform of the $sinc^2$ function is a convolution of two rectangles.

   (b) Modulate the signal by a complex exponential function: $x_2(t) = e^{-i2\pi f_1 t} sinc^2(f_0 t), f_1 = 8Hz$. What is the Nyquist sampling rate for $x_2(t)$? Sample the signal $x_2(t)$ at the Nyquist sampling rate of $x_2(t)$and at the Nyquist sampling rate of $x_1(t)$, for $|t| < 10$. Plot the sampled signals. Plot and compare the resulting DFT spectra.

   (c) Load the "`rubberducky.jpg`" image. Perform a 2D DFT to get the spectrum of the image. Undersample the spectrum by a factor of 2 along the i) rows, ii) columns, and iii) both rows and columns. Perform an inverse 2D DFT on the undersampled spectra to reconstruct images. Display the reconstructed images. What do you observe and why?

3. Window functions

   (a) In the time domain, plot the following windows of 32-point length: (i) Rectangular window, (ii) Triangular window, (iii) Hamming window, (iv) Hanning (Hann) window, (v) Kaiser window.

   (b) Plot the 256-point DFT of the above 32-point windows. More specifically, plot the magnitude of the DFT spectra of the windows in decibels (dB).

   *HINT: First normalize the DFT magnitude spectrum to 1, then display the spectrum in dB using 20*log10(X). Also note that 20*log10(0) = -Inf. Typically `Matlab` will ignore infinite* *values when plotting. You can add a very small non-zero value to the magnitude spectrum to avoid this, but make sure it does not significantly affect the side lobe height (try* `eps` *which matlab recognizes as the number* 2^{-52}*).*

(c) Use the plot trace tool to determine the main lobe width and the magnitude of the highest side lobe for each of the above windows.

4. Study windowing effects: 1D case

(a) Download the `signal.mat` file from the course website, where x is an unknown signal of length 32. It is known that the sampling period is 0.02 seconds, which satisfies the Nyquist criterion, and the signal contains 5 frequency components, two of which have much weaker magnitude compared to the other three components. Plot the magnitude DFT spectrum using each of the 5 windows from the previous problem to compare windowing effects. Find the five frequencies present in the signal.

(b) Thus far in ECE 311, we have typically considered signals with frequency components that do not change over time (even in the telephone problem, each dialed number had two *constant* tones). Consider a signal with two time-dependent, logarithmic chirp frequency components, given below.

```
t = 0:0.001:6; Fs = 1000; % 6 seconds @ 1kHz sample rate
% make two logarithmic chirps offset by 20 Hz
fo = 45; f1 = 450; offset = 20;
ya = chirp(t,fo,6,f1,'logarithmic'); yb = chirp(t,fo+offset,6,f1+offset,'logarithmic');
y = ya + yb;
L = 256; nfft = 1024; %Hamming (default) window length is 256, DFT length is 1024
spectrogram(y,L,128,nfft,Fs,'yaxis');% view chirps
soundsc(y,Fs) % listen to chirps
```

The spectrogram windows the time signal with a length L=256 hamming window starting every 128 samples. For each windowed signal, a 1024-point DFT is taken, and the magnitude spectrum (one-sided) is shown as a heat map; each DFT is given as a vertical slice of the spectrogram.

Show the spectrogram result in your lab report. Then, consider *one* short segment of the time-domain signal beginning at t=4 seconds. Plot the magnitude 1024-point DFT spectrum for multiple window lengths, including L=256, L=512 and L=1024. What happens to the magnitude spectrum as you increase the window length? Why? This is why the 'short-time' fourier transform is often critical in signal processing!

5. Study windowing effects: 2D case

(a) Generate a 256×256 image using the command `I = phantom(256)`. Display the image in gray scale.

(b) Perform 2D DFT to get the spectrum of the image using the command `s = fftshift(fft2(I))`. Truncate the spectrum `s` by taking only the central or lower frequency parts of the spectrum (64 ×64 out of 256×256), and zero-padding the rest. Reconstruct and display the resulting image from the truncated spectrum. What is your observation and why?

HINT: Use `ifft2(ifftshift(st))`*, where* `st` *is the truncated spectrum, to reconstruct the image.*

(c) Apply a Hamming window of length 64 along i) row, ii) column, and iii) both row and column of the truncated spectrum in b. Reconstruct and display the resulting images from the spectra. How does this differ from using the rectangular window in part (b), and why?