Overview:

- **Motivation**
  Many applications require analyzing the frequency content of signals
  - Speech processing – study resonances of vocal tract
  - Radar – Doppler processing
  - Sonar – study channel effects on long-range transmissions
- **Basic spectral analyzer setup**
  - Windowing
  - Frequency-sampling
  - Relating DFT output to CT spectrum
- **Examples**
Basic spectral analyzer

- Anti-alias filtering and A/D conversion ← for tonight assume negligible errors

- Windowing operation
  - Necessary because DFT only works on finite-length sequences.
  - Typically use same windows as for filter design

- Frequency-sampling
  - Use FFT to compute samples of the DTFT

Understanding windowing & frequency-sampling crucial to interpreting results ⇒
Windowing and frequency-sampling

What does the windowing operation do?

\[ v[n] = w[n] \cdot x[n] \xrightarrow{\mathcal{F}} V(e^{j\omega}) = \frac{1}{2\pi} \int_{-\pi}^{+\pi} X(e^{j\theta})W(e^{j(\omega-\theta)})d\theta \]

- Convolution with window function smears the spectrum of \( x[n] \)
- Smoothes sharp peaks and discontinuities

What is the output of the \( N \)-point DFT operation?

\[
V[k] = \sum_{n=0}^{N-1} v[n]e^{-j\frac{2\pi kn}{N}} \quad k = 0, \ldots, N-1
\]

\[
= V(e^{j\omega})\bigg|_{\omega = \frac{2\pi k}{N}}
\]

- DFT is samples of the DTFT evenly spaced around the unit circle
Relating DFT output to the CT spectrum

Suppose $x_c(t)$ is sampled at a rate of $f_s = \frac{1}{T}$.

What CT frequencies correspond to the $N$ DFT samples?

Recall relationship between DT and CT frequencies: \[ \omega = \Omega T \]

Thus the CT frequency associated with the $k$th bin is:

\[ \Omega_k = \frac{\omega_k}{T} = \frac{2\pi k}{N T} = \frac{2\pi k f_s}{N} \]

Using this relationship, we create a vector of analog frequencies to plot the spectrum with

Consider analyzing a signal consisting of 2 sinusoids ⇒
**Two sinusoid example**

Consider \( s_c(t) \) equal to the sum of 2 sinusoids:  

\[
s_c(t) = A_0 \cos(\Omega_0 t) + A_1 \cos(\Omega_1 t)
\]

The sampled signal \( x[n] \) becomes:

\[
x[n] = A_0 \cos(\omega_0 n) + A_1 \cos(\omega_1 n) \quad \omega_0 = \Omega_0 T \quad \omega_1 = \Omega_1 T
\]

The windowed signal \( v[n] \) is:

\[
v[n] = A_0 w[n] \cos(\omega_0 n) + A_1 w[n] \cos(\omega_1 n)
\]

\[
= \frac{A_0}{2} w[n] \left(e^{j\omega_0 n} + e^{-j\omega_0 n}\right) + \frac{A_1}{2} w[n] \left(e^{j\omega_1 n} + e^{-j\omega_1 n}\right)
\]

Taking the DTFT:

\[
V(e^{j\omega}) = \frac{A_0}{2} \left\{ W(e^{j(\omega - \omega_0)}) + W(e^{j(\omega + \omega_0)}) \right\} + \frac{A_1}{2} \left\{ W(e^{j(\omega - \omega_1)}) + W(e^{j(\omega + \omega_1)}) \right\}
\]

Copies of \( W(e^{j\omega}) \) at \( \pm \omega_0, \pm \omega_1 \)
Two effects of the window

Resolution

- controlled by mainlobe width
- high resolution requires narrow ML

What controls mainlobe width?
- length of window ← primary factor
- type of window ← secondary factor
  e.g., Hanning has wider ML than boxcar

Leakage

- controlled by sidelobe height
- energy at $\omega_0$ “leaks out” and can mask lower power signal at $\omega_1$

What controls sidelobe height?
- type of window
  see table in Ch. 7

Consider example ⇒
Example (compare to 10.3 in text)

- 64-point rectangular window
- Mainlobe width $\approx \frac{4\pi}{64}$

$v[n] = A_0 \cos(\omega_0 n) + A_1 \cos(\omega_1 n)$

$A_0 = 1$, $A_1 = .75$

$\omega_0 = \frac{2\pi}{6}$, $\omega_1 = \frac{2\pi}{3}$

$\Rightarrow$ copies of $W(e^{j\omega})$ at $\pm \frac{2\pi}{6}$, $\pm \frac{2\pi}{3}$
Example 10.3 continued

- $A_0 = 1, A_1 = .75$
- $\omega_0 = \frac{2\pi}{14}, \omega_1 = \frac{4\pi}{15}$

$\Rightarrow$ As sinusoids move closer together, leakage causes change in amplitudes
Example 10.3 continued

- $A_0 = 1, A_1 = 0.75$
- $\omega_0 = \frac{2\pi}{14}, \omega_1 = \frac{4\pi}{25}$

- Sinusoids not resolved when they move close enough
- Mainlobe width $\approx \frac{4\pi}{64} = 0.0625\pi$
- $\omega_1 - \omega_0 = 0.0171\pi$

Consider Matlab demo 1 ⇒
Matlab demo: `windowsine_demos1`

Parameters:

- Sampling frequency: $f_s = 100$ Hz
- Window length: 100 points $\rightarrow \approx 1$ second of data
- DFT length: 8192 points (lots of samples!)
- $x_c(t)$ consists of the sum of two sinewaves
  - a 5 Hz sinewave
  - a sinewave of varying frequency: 10 Hz (1st case) to 5 Hz (last case)

See what happens to spectral plot in demo 1 ⇒
Spectral sampling

Consider a 64-point section of $v[n] = A_0 \cos(\omega_0 n) + A_1 \cos(\omega_1 n)$ where

- $A_0 = 1$, $A_1 = .75$
- $\omega_0 = \frac{2\pi}{14}$, $\omega_1 = \frac{4\pi}{25}$

\[|V[k]|\]

\[|V(e^{j\omega})|\]

- Results of 64-point DFT
- DTFT (shown between 0 and $2\pi$)

$\Rightarrow$ DFT is samples of the DTFT
Spectral sampling can be misleading!

Consider a 64-point section of \( v[n] = A_0 \cos(\omega_0 n) + A_1 \cos(\omega_1 n) \) where

- \( A_0 = 1 \), \( A_1 = .75 \), \( \omega_0 = \frac{2\pi}{16} \), \( \omega_1 = \frac{2\pi}{8} \)

We can zero-pad \( v[n] \) and take a larger FFT to get more samples \( \Rightarrow \)
Zero-padding

Results of padding $v[n]$ with 64 zeros and taking 128-pt DFT

- Samples no longer at zero-crossings of DTFT
- “Clean” look of 64-point DFT was a bit of an illusion because samples occurred at zero-crossings of spectrum

Can zero-padding improve resolution?

Consider Matlab demo 2 ⇒
Zero-padding Matlab demo: windowsine_demo2

Parameters:

- $x_c(t)$ consists of the sum of two sinewaves
  - a 5 Hz sinewave
  - a 5.5 Hz sinewave
- Sampling frequency: $f_s = 100$ Hz
- Window length: 100 points $\Rightarrow \approx 1$ second of data
- DFT length varies from 128 to 32768

See what happens to spectral plot in demo 2 $\Rightarrow$
Why aren’t the sinusoids resolved in `windowsine_demo2`?

Window: 100 point rectangular window

Mainlobe width: $\Delta \omega_{ML} = \frac{4\pi}{100} \quad \rightarrow \quad \Delta \Omega_{ML} = \frac{4\pi}{100} f_s = \frac{4\pi}{100} \cdot 100 = 4\pi$

Thus the mainlobe width corresponds to approximately 2 Hz

Since the sinewaves are 0.5 Hz apart, they won’t be resolved

⇒ Better resolution requires longer window (more data!)

Consider third Matlab demo ⇒
Matlab demo: windowsine_demo3

This demo illustrates how increasing the window length improves resolution.

Parameters:

- \( x_c(t) \) consists of the sum of two sinewaves
  - a 5 Hz sinewave
  - a 5.5 Hz sinewave

- Sampling frequency: \( f_s = 100 \) Hz

- DFT length: 8192 points (lots of samples!)

- Window length varies from 100 to 1000 in increments of 25

See what happens to spectral plot in demo 3 ⇒