Digital Signal Processing
Lecture 10 - Discrete Fourier Transform

Electrical Engineering and Computer Science
University of Tennessee, Knoxville

November 12, 2015
Overview

1. Recap
2. DTFT
3. DFS
4. DFT
Review - 1

- Introduction
- Discrete-time signals and systems - LTI systems
  - Unit sample response \( h[n] \): uniquely characterizes an LTI system
  - Linear constant-coefficient difference equation
  - Frequency response: \( H(e^{j\omega}) \)
  - Complex exponentials being eigenvalues of an LTI system: \( y[n] = H(e^{j\omega})x[n] \)
  - Fourier transform
- \( z \) transform
  - The \( z \)-transform, \( X(z) = \sum_{n=\infty}^{n=-\infty} x[n]z^{-n} \)
  - Region of convergence - the \( z \)-plane
  - System function, \( H(z) \)
  - Properties of the \( z \)-transform
  - The significance of zeros
  - The inverse \( z \)-transform
- Relationships between the \( n \), \( \omega \), and \( z \) domains: Knowing the correspondence between \( h[n] \), \( H(e^{j\omega}) \), and the pole-zero plot
Review - 2

- Design structures
  - Block diagram vs. Signal flow graph: Knowing how to determine system function, unit sample response, or difference equation from the graphs
  - Different design structures: Knowing pros and cons of each form [Direct form I (zeros first), Direct form II (poles first) - Canonic structure, Transposed direct form II (zeros first), Cascade form, Parallel form, Coupled form]
  - Specific to IIR or FIR: Feedback in IIR (computable vs. noncomputable), Linear phase in FIR
  - Metrics: computational resource and precision
  - Sources of errors: Knowing the concept of pole sensitivity of 2nd-order structures leading to coupled form design, and coefficient quantization examples between direct form vs. cascade form
Filter design

- IIR: CT → DT (impulse invariance vs. bilinear transformation)
- FIR

- Knowing the characteristics of the four types of causal linear phase FIR filters
- Window method - Kaiser window: must use minimum specs; the approximation error is scaled by the size of the jump that produces them
- Optimal method (Alternation theorem <knowing how to determine the number of alternations>, PM algorithm)
Discrete-Time Fourier Transform (DTFT)

- Fourier transform representation of $x[n]$

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n} \quad (1)$$

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

- Existence of Fourier transform

  - Absolutely summable (a sufficient condition),
    $$\sum_{n=-\infty}^{\infty} |x[n]| < \infty$$
    leading to uniform convergence,
    $$\lim_{M \to \infty} \int_{-\pi}^{\pi} |X(e^{j\omega}) - X_M(e^{j\omega})| d\omega = 0$$

  - Square summable, $\sum_{n=-\infty}^{\infty} |x[n]|^2 < \infty$, leading to mean-square convergence,
    $$\lim_{M \to \infty} \int_{-\pi}^{\pi} |X(e^{j\omega}) - X_M(e^{j\omega})|^2 d\omega = 0$$
Discrete Fourier Series (DFS)

- Periodic sequence does not satisfy either absolutely summable or square summable, therefore, it does not have a Fourier representation.

- However, sequences expressed as a sum of complex exponentials can be considered to have an FT representation, i.e., as a train of impulses.

\[ X(e^{j\omega}) = \sum_{r=-\infty}^{\infty} \sum_{k} 2\pi a_k \delta(\omega - \omega_k + 2\pi r) \]

- Interpret DTFT of a periodic signal to be an impulse train in the frequency domain with the impulse values proportional to the DFS coefficients.
DFS - cont’d

\[ \tilde{X}[k] = \sum_{n=0}^{N-1} \tilde{x}[n] W_N^{kn} \]

\[ \tilde{x}[n] = \frac{1}{N} \sum_{k=0}^{N-1} \tilde{X}[k] W_N^{-kn} \]

where \( W_N = e^{-j(2\pi/N)} \).

Since any periodic sequence can be represented as a sum of complex exponentials

\[ X(e^{j\omega}) = \sum_{k=-\infty}^{\infty} \frac{2\pi}{N} \tilde{X}[k] \delta(\omega - \frac{2\pi k}{N}) \]
Finite-length signal vs. periodic signal

\[
\tilde{x}[n] = x[n] \ast \tilde{p}[n] = x[n] \ast \sum_{r=-\infty}^{\infty} \delta[n - rN] = \sum_{r=-\infty}^{\infty} x[n - rN]
\]

\[
\tilde{x}[n] = x[(n)_{N}]
\]

\[
\tilde{X}[k] = X[(k)_{N}]
\]
Discrete Fourier Transform (DFT)

- \( X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn}, \ 0 \leq k \leq N - 1 \)
- \( x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] W_N^{-kn}, \ 0 \leq n \leq N - 1 \)
- DFT coefficients are samples of the z-transform at equal-sapce points on the unit circle

\[
X(z) = \sum_{n=0}^{N-1} x[n] z^{-n}, \ X(k) = X(z)|_{z=W_N^{-k}}, \ k = 0, \cdots, N-1
\]
Relationship between DTFT, DFS, and DFT
Properties of DFT

- Shifting properties
- Duality
- Convolution property - circular convolution. Think about circular convolution as wrapping the sequence on the surface of two cylinders, one inside another; then convolution is done by rotate the inner cylinder with respect to the outer one.

Circular convolution = linear convolution plus aliasing
Shifting property

- **Diagram (a)**: Sequence $x[n]$.
- **Diagram (b)**: Sequence $\bar{x}[n]$ shifted by $2$.
- **Diagram (c)**: Sequence $\bar{x}_1[n] = \bar{x}[n + 2]$.
- **Diagram (d)**: Sequence $x_1[n] = \begin{cases} \bar{x}_1[n], & 0 \leq n \leq N - 1 \\ 0, & \text{otherwise} \end{cases}$.
Duality property

Recap
DTFT
DFS
DFT
Convolution property

\[ x_2[m] \]

\[ x_1[m] \]

\[ x_2[((0 - m)_N), 0 \leq m \leq N - 1] \]

\[ x_2[((1 - m)_N), 0 \leq m \leq N - 1] \]

\[ x_3[n] = x_1[n] \otimes x_2[n] \]
Use circular convolution to implement linear convolution
Use circular convolution to implement linear convolution - aliasing

\[ x_{3}[n] = x_{1}[n] \circledast x_{2}[n], \quad N = L \]

\[ x_{3}[n] = x_{1}[n] \circledast x_{2}[n], \quad N = L + P - 1 \]
Use circular convolution to implement linear convolution - zero padding

- Frequency-domain calculation
  - (a) \(x_1[n] \leftrightarrow X_1[k], x_2[n] \leftrightarrow X_2[k]\)
  - (b) \(X_3[k] = X_1[k] \cdot X_2[k]\)
  - (c) \(x_3[n] = x_1[n] \circledast x_2[n]\) = inverse DFT of \(X_3[k]\) (note that I use \(\circledast\) to represent circular convolution.)

- Length of circular convolution
  - \(x_1[n]\) of length \(L\)
  - \(x_2[n]\) of length \(P\)
  - \(x_3[n] = x_1[n] \cdot x_2[n]\) of length \(L + P - 1\)
  - circular convolution has to be longer than the linear convolution. Time aliasing can be avoided if \(N \geq L + P - 1\)
Example - 2D Convolution

**FIGURE 4.38**
Illustration of the need for function padding.
(a) Result of performing 2-D convolution without padding.
(b) Proper function padding.
(c) Correct convolution result.
Example - 2D Convolution

**Figure 4.39** Padded lowpass filter in the spatial domain (only the real part is shown).

**Figure 4.40** Result of filtering with padding. The image is usually cropped to its original size since there is little valuable information past the image boundaries.
Example - 2D Convolution

Difference image from convolution in the spatial domain

Convolution in the frequency domain

No padding

With padding

Conv. spatially
Discrete Cosine Transform

- **Forward transform**
  \[ T(u, v) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y)g(x, y, u, v) \]

  \[ g(x, y, u, v) = \alpha(u)\alpha(v) \cos \left( \frac{(2x + 1)u\pi}{2N} \right) \cos \left( \frac{(2y + 1)v\pi}{2N} \right) \]

- **Inverse transform**
  \[ f(x, y) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} T(u, v)h(x, y, u, v) \]

  \[ g(x, y, u, v) = h(x, y, u, v) \]
Discrete Cosine Transform

http://www.it.cityu.edu.hk/~itaku/lecture/Chap4.2.html
Discrete Cosine Transform

**FIGURE 8.32** The periodicity implicit in the 1-D (a) DFT and (b) DCT.