ESE 5310: Digital Signal Processing

Lecture 20: March 30, 2023 Discrete Fourier Transform, Pt 2





- Review:
 - Discrete Fourier Transform (DFT)
- Circular Convolution
- Fast Convolution Methods
- □ Time-aliasing w/ Convolution



• The DFT

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] W_N^{-kn} \quad \text{Inverse DFT, synthesis}$$

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{kn} \quad \text{DFT, analysis}$$

□ It is understood that,

$$x[n] = 0$$
 outside $0 \le n \le N-1$
 $X[k] = 0$ outside $0 \le k \le N-1$



DTFT:

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

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

DFT:

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

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

DFT Intuition





Adapted from M. Lustig, EECS Berkeley



Circular frequency shift

$$x[n]e^{j(2\pi/N)nl} = x[n]W_N^{-nl} \leftrightarrow X[((k-l))_N]$$

Complex Conjugation

$$x^*[n] \leftrightarrow X^*[((-k))_N]$$

Conjugate Symmetry for Real Signals

$$x[n] = x^*[n] \leftrightarrow X[k] = X^*[((-k))_N]$$





$$x[n] = x^*[n] \leftrightarrow X[k] = X^*[((-k))_N]$$

8





$$x[n] = x^*[n] \leftrightarrow X[k] = X^*[((-k))_N]$$

9





$$x[n] = x^*[n] \leftrightarrow X[k] = X^*[((-k))_N]$$
 10









 $x[n] = x^*[n] \leftrightarrow X[k] = X^*[((-k))_N]$







Circular Convolution:

$$x_1[n] \otimes x_2[n] \triangleq \sum_{m=0}^{N-1} x_1[m] x_2[((n-m))_N]$$

For two signals of length N

Note: Circular convolution is commutative

 $x_2[n] \otimes x_1[n] = x_1[n] \otimes x_2[n]$

























m=0



• For $x_1[n]$ and $x_2[n]$ with length N

$x_1[n] \otimes x_2[n] \leftrightarrow X_1[k] \cdot X_2[k]$

• Very useful!! (for linear convolutions with DFT)



• For $x_1[n]$ and $x_2[n]$ with length N

$$x_1[n] \cdot x_2[n] \leftrightarrow \frac{1}{N} X_1[k] \otimes X_2[k]$$

Linear Convolution

□ Next....

- Using DFT, circular convolution is easy
 - Matrix multiplication
- But, linear convolution is useful, not circular
- So, show how to perform linear convolution with circular convolution
- Use DFT to do linear convolution (via circular convolution)



□ We start with two non-periodic sequences:

$$\begin{array}{ll} x[n] & 0 \leq n \leq L-1 \\ h[n] & 0 \leq n \leq P-1 \end{array} \end{array}$$

• E.g. x[n] is a signal and h[n] a filter's impulse response





$$x_1[n] \otimes x_2[n] \stackrel{\Delta}{=} \sum_{m=0}^{N-1} x_1[m] x_2[((n-m))_N]$$



□ We start with two non-periodic sequences:

$$\begin{aligned} x[n] & 0 \leq n \leq L-1 \\ h[n] & 0 \leq n \leq P-1 \end{aligned}$$

E.g. x[n] is a signal and h[n] a filter's impulse response
We want to compute the linear convolution:

$$y[n] = x[n] * h[n] = \sum_{m=0}^{L-1} x[m]h[n-m]$$

• y[n] is nonzero for $0 \le n \le L+P-2$ (ie. length M=L+P-1)

Requires L*P multiplications

Linear Convolution via Circular Convolution

Zero-pad x[n] by P-1 zeros $x_{zp}[n] = \begin{cases} x[n] & 0 \le n \le L-1 \\ 0 & L \le n \le L+P-2 \end{cases}$ Zero-pad h[n] by L-1 zeros

$$h_{\rm zp}[n] = \begin{cases} h[n] & 0 \le n \le P-1\\ 0 & P \le n \le L+P-2 \end{cases}$$

□ Now, both sequences are length M=L+P-1

Linear Convolution via Circular Convolution

- □ Now, both sequences are length M=L+P-1
- We can now compute the linear convolution using a circular one with length M=L+P-1

Linear convolution via circular

$$y[n] = x[n] * y[n] = \begin{cases} x_{zp}[n] \textcircled{0} h_{zp}[n] & 0 \le n \le M - 1 \\ 0 & \text{otherwise} \end{cases}$$





M = L + P - 1 = 8





M = L + P - 1 = 8





Linear Convolution with DFT

 In practice we can implement a circular convolution using the DFT property:

$$\begin{aligned} x[n] * h[n] &= x_{zp}[n] \bigotimes h_{zp}[n] \\ &= \mathcal{DFT}^{-1} \left\{ \mathcal{DFT} \left\{ x_{zp}[n] \right\} \cdot \mathcal{DFT} \left\{ h_{zp}[n] \right\} \right\} \\ &\text{for } 0 \le n \le M-1, M=L+P-1 \end{aligned}$$

Linear Convolution with DFT

 In practice we can implement a circular convolution using the DFT property:

$$\begin{split} x[n] * h[n] &= x_{zp}[n] \textcircled{0} h_{zp}[n] \\ &= \mathcal{DFT}^{-1} \left\{ \mathcal{DFT} \left\{ x_{zp}[n] \right\} \cdot \mathcal{DFT} \left\{ h_{zp}[n] \right\} \right\} \\ & \text{for } 0 \le n \le M-1, M=L+P-1 \end{split}$$

Advantage: DFT can be computed with Nlog₂N complexity (FFT algorithm later!)

Linear Convolution with DFT

 In practice we can implement a circular convolution using the DFT property:

$$\begin{split} x[n] * h[n] &= x_{zp}[n] \textcircled{M} h_{zp}[n] \\ &= \mathcal{DFT}^{-1} \left\{ \mathcal{DFT} \left\{ x_{zp}[n] \right\} \cdot \mathcal{DFT} \left\{ h_{zp}[n] \right\} \right\} \\ & \text{for } 0 \le n \le M-1, M=L+P-1 \end{split}$$

- Advantage: DFT can be computed with Nlog₂N complexity (FFT algorithm later!)
- Drawback: Must wait for all the samples -- huge delay -- incompatible with real-time filtering

Block Convolution

- **Problem:**
 - An input signal x[n], has very long length (could be considered infinite)
 - An impulse response h[n] has length P
 - We want to take advantage of DFT/FFT and compute convolutions in blocks that are shorter than the signal
- Approach:
 - Break the signal into small blocks
 - Compute convolutions (via DFT)
 - Combine the results
 - Overlap-add
 - Overlap-save



Example:




Decompose into non-overlapping segments

$$x_r[n] = \begin{cases} x[n] & rL \le n < (r+1)L \\ 0 & \text{otherwise} \end{cases}$$

• The input signal is the sum of segments

$$x[n] = \sum_{r=0}^{\infty} x_r[n]$$





Penn ESE 5310 Spring 2023 – Khanna Adapted from M. Lustig, EECS Berkeley



$$x[n] = \sum_{r=0}^{\infty} x_r[n]$$

□ The output is:

$$y[n] = x[n] * h[n] = \sum_{r=0}^{\infty} x_r[n] * h[n]$$

• Each output segment $x_r[n] * h[n]$ is length M=L+P-1

- h[n] has length P
- $x_r[n]$ has length L



- We can compute $x_r[n]*h[n]$ using circular convolution with the DFT
- Using the DFT:
 - Zero-pad $x_r[n]$ to length M
 - Zero-pad h[n] to length M and compute $DFT_M{h_{zp}[n]}$
 - Only need to do once!



- We can compute $x_r[n]*h[n]$ using circular convolution with the DFT
- Using the DFT:
 - Zero-pad x_r[n] to length M
 - Zero-pad h[n] to length M and compute $DFT_N{h_{zp}[n]}$
 - Only need to do once!
 - Compute:

$x_r[n] * h[n] = \mathrm{DFT}^{-1} \left\{ \mathrm{DFT} \{ x_{r,zp}[n] \} \cdot \mathrm{DFT} \{ h_{zp}[n] \} \right\}$

- □ Results are of length M=L+P-1
 - Neighboring results overlap by P-1
 - Add overlaps to get final sequence



L+P-1=16





Example:

h[n] Impulse response, Length P=6

mm



L+P-1=16



Penn ESE 5310 Spring 2023 – Khanna Adapted from M. Lustig, EECS Berkeley



L+P-1=16



Penn ESE 5310 Spring 2023 – Khanna Adapted from M. Lustig, EECS Berkeley



- Basic idea:
- Split input into overlapping segments with length L+P-1
 - P-1 sample overlap

$$x_r[n] = \begin{cases} x[n] & rL \le n < (r+1)L + P\\ 0 & \text{otherwise} \end{cases}$$

 Perform circular convolution in each segment, and keep the L sample portion which is a valid linear convolution





Circular to Linear Convolution

- An L-point sequence circularly convolved with a Ppoint sequence
 - with L P zeros padded, P < L
- gives an *L*-point result with
 - the first *P* 1 values *incorrect* and
 - the next L P + 1 the *correct* linear convolution result



Circular to Linear Convolution

- An L-point sequence circularly convolved with a Ppoint sequence
 - with L P zeros padded, P < L
- gives an *L*-point result with
 - the first *P* 1 values *incorrect* and
 - the next L P + 1 the *correct* linear convolution result











Linear Convolution via Circular Convolution

Zero-pad x[n] by P-1 zeros $x_{zp}[n] = \begin{cases} x[n] & 0 \le n \le L-1 \\ 0 & L \le n \le L+P-2 \end{cases}$ Zero-pad h[n] by L-1 zeros

$$h_{\rm zp}[n] = \begin{cases} h[n] & 0 \le n \le P-1\\ 0 & P \le n \le L+P-2 \end{cases}$$

□ Now, both sequences are length M=L+P-1

□ If the DTFT X($e^{j\omega}$) of a sequence x[n] is sampled at N frequencies $\omega_k = 2\pi k/N$, then the resulting sequence X[k] corresponds to the periodic sequence

$$\tilde{x}[n] = \sum_{r=-\infty}^{\infty} x[n-rN].$$

□ And
$$X[k] = \begin{cases} X(e^{j(2\pi k/N)}), & 0 \le k \le N-1, \\ 0, & \text{otherwise,} \end{cases}$$
 is the DFT of one period given as

$$x_p[n] = \begin{cases} \tilde{x}[n], & 0 \le n \le N-1, \\ 0, & \text{otherwise.} \end{cases}$$

$$x_p[n] = \begin{cases} \tilde{x}[n], & 0 \le n \le N-1, \\ 0, & \text{otherwise.} \end{cases}$$

- If x[n] has length less than or equal to N, then
 x_p[n]=x[n]
- However if the length of x[n] is greater than N, this might not be true and we get aliasing in time
 - N-point convolution results in N-point sequence

- □ Given two N-point sequences (x₁[n] and x₂[n]) and their N-point DFTs (X₁[k] and X₂[k])
- □ The N-point DFT of $x_3[n] = x_1[n] * x_2[n]$ is defined as

$$X_{3}[k] = X_{3}(e^{j(2\pi k/N)})$$

- □ Given two N-point sequences (x₁[n] and x₂[n]) and their N-point DFTs (X₁[k] and X₂[k])
- □ The N-point DFT of $x_3[n] = x_1[n] * x_2[n]$ is defined as

$$X_{3}[k] = X_{3}(e^{j(2\pi k/N)})$$

□ And X₃[k]=X₁[k]X₂[k], where the inverse DFT of X₃[k] is

$$x_{3p}[n] = x_1[n] \bigotimes x_2[n]$$

- □ Given two N-point sequences (x₁[n] and x₂[n]) and their N-point DFTs (X₁[k] and X₂[k])
- □ The N-point DFT of $x_3[n] = x_1[n] * x_2[n]$ is defined as

$$X_{3}[k] = X_{3}(e^{j(2\pi k/N)})$$

□ And X₃[k]=X₁[k]X₂[k], where the inverse DFT of X₃[k] is

$$x_{3p}[n] = \begin{cases} \sum_{r=-\infty}^{\infty} x_3[n-rN], & 0 \le n \le N-1, \\ 0, & \text{otherwise,} \end{cases}$$

$$x_{3p}[n] = \begin{cases} \sum_{r=-\infty}^{\infty} x_3[n-rN], & 0 \le n \le N-1, \\ 0, & \text{otherwise,} \end{cases}$$

$$x_{3p}[n] = \begin{cases} \sum_{r=-\infty}^{\infty} x_3[n-rN], & 0 \le n \le N-1, \\ 0, & \text{otherwise,} \end{cases}$$

Thus

$$x_{3p}[n] = \begin{cases} \sum_{r=-\infty}^{\infty} x_1[n-rN] * x_2[n-rN] & 0 \le n \le N-1 \\ 0 & \text{else} \end{cases}$$

$$x_{3p}[n] = x_1[n] \bigotimes x_2[n]$$

The N-point circular convolution is the sum of linear convolutions shifted in time by N











The sum of N-shifted linear convolutions equals the N-point circular convolution





The sum of N-shifted linear convolutions equals the N-point circular convolution





The sum of N-shifted linear convolutions equals the N-point circular convolution





If I want the circular convolution and linear convolution to be the same, what do I do?



□ If I want the circular convolution and linear convolution to be the same, what do I do?

• Take the N=2L-point circular convolution





□ If I want the circular convolution and linear convolution to be the same, what do I do?

• Take the N=2L-point circular convolution











□ What does the L-point circular convolution look like?





□ What does the L-point circular convolution look like?



□ The L-shifted linear convolutions



Penn ESE 5310 Spring 2023 - Khanna


□ The L-shifted linear convolutions



Penn ESE 5310 Spring 2023 - Khanna



- Discrete Fourier Transform (DFT)
 - For finite signals assumed to be zero outside of defined length
 - N-point DFT is sampled DTFT at N points
 - DFT properties inherited from DFS, but circular operations!
- Fast Convolution Methods
 - Use circular convolution (i.e DFT) to perform fast linear convolution
 - Overlap-Add, Overlap-Save
- Circular convolution is linear convolution w/ aliasing
 - Must do a DFT long enough to get no aliasing

Penn ESE 5310 Spring 2023 – Khanna Adapted from M. Lustig, EECS Berkeley



□ HW 7 out now

Penn ESE 5310 Spring 2023 – Khanna Adapted from M. Lustig, EECS Berkeley