ESE 531: Digital Signal Processing

Lec 20: April 11, 2017

Discrete Fourier Transform, Pt 2

Today

- Review: Discrete Fourier Transform (DFT)
- DFT Properties
  - Duality
  - Circular Shift
- Circular Convolution
- Fast Convolution Methods

Discrete Fourier Transform

- 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,

$$\begin{aligned} x[n] &= 0 \quad \text{outside } 0 \le n \le N-1 \\ X[k] &= 0 \quad \text{outside } 0 \le k \le N-1 \end{aligned}$$

DFT vs. DTFT

- For finite sequences of length N:
  - The N-point DFT of x[n] is:

$$X[k] = \sum_{n=0}^{N-1} x[n]W_N^{kn} = \sum_{n=0}^{N-1} x[n]e^{-j(2\pi/N)nk} \quad 0 \le k \le N-1$$

  - The DTFT of x[n] is:

$$X(e^{j\omega}) = \sum_{n=0}^{N-1} x[n]e^{-j\omega n} \quad -\infty < \omega < \infty$$

DFT vs. DTFT

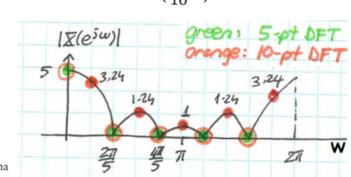
- The DFT are samples of the DTFT at N equally spaced frequencies

$$X[k] = X(e^{j\omega})|_{\omega = k\frac{2\pi}{N}} \quad 0 \le k \le N-1$$

DFT vs DTFT

- Back to example

$$\begin{aligned} X[k] &= \sum_{n=0}^{4} W_{10}^{nk} \\ &= e^{-j\frac{4\pi}{10}k}\frac{\sin(\frac{\pi}{2}k)}{\sin(\frac{\pi}{10}k)} \end{aligned}$$



Properties of the DFT

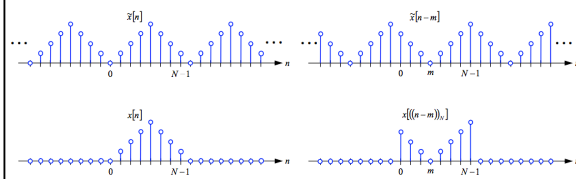
- Properties of DFT inherited from DFS
- Linearity

$$\alpha_1 x_1[n] + \alpha_2 x_2[n] \leftrightarrow \alpha_1 X_1[k] + \alpha_2 X_2[k]$$

- Circular Time Shift

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

Circular Shift



Properties of DFT

- Circular frequency shift

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

- Complex Conjugation

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

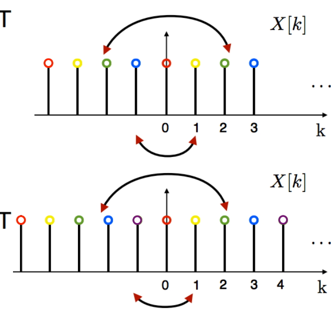
- Conjugate Symmetry for Real Signals

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

Example: Conjugate Symmetry

4-point DFT
–Symmetry

5-point DFT
–Symmetry

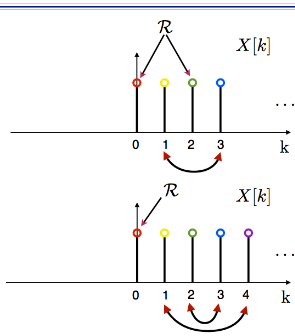


$$x[n] = x^*[n] \leftrightarrow X[k] = X^*[((-k))_N]$$

Example

4-point DFT
–Symmetry

5-point DFT
–Symmetry



Properties of the DFS/DFT

Discrete Fourier Series			Discrete Fourier Transform		
Property	***N*-periodic sequence**	***N*-periodic DFS**	**Property**	***N*-point sequence**	***N*-point DFT**
	$\tilde{x}[n]$ $\tilde{x}_1[n],\ \tilde{x}_2[n]$	$\tilde{X}[k]$ $\tilde{X}_1[k],\ \tilde{X}_2[k]$		$x[n]$ $x_1[n],\ x_2[n]$	$X[k]$ $X_1[k],\ X_2[k]$
Linearity	$a\tilde{x}_1[n]+b\tilde{x}_2[n]$	$a\tilde{X}_1[k]+b\tilde{X}_2[k]$	Linearity	$ax_1[n]+bx_2[n]$	$aX_1[k]+bX_2[k]$
Duality	$\tilde{X}[n]$	$N\,\tilde{x}[-k]$	Duality	$X[n]$	$N\,x[((-k))_N]$
Time Shift	$\tilde{x}[n-m]$	$W_N^{km}\tilde{X}[k]$	Circular Time Shift	$x[((n-m))_N]$	$W_N^{km}X[k]$
Frequency Shift	$W_N^{-ln}\tilde{x}[n]$	$\tilde{X}[k-l]$	Circular Frequency Shift	$W_N^{-ln}x[n]$	$X[((k-l))_N]$
Periodic Convolution	$\sum_{m=0}^{N-1}\tilde{x}_1[m]\tilde{x}_2[n-m]$	$\tilde{X}_1[k]\tilde{X}_2[k]$	Circular Convolution	$\sum_{m=0}^{N-1}x_1[m]x_2[((n-m))_N]$	$X_1[k]X_2[k]$
Multiplication	$\tilde{x}_1[n]\tilde{x}_2[n]$	$\frac{1}{N}\sum_{l=0}^{N-1}\tilde{X}_1[l]\tilde{X}_2[k-l]$	Multiplication	$x_1[n]x_2[n]$	$\frac{1}{N}\sum_{l=0}^{N-1}X_1[l]X_2[((k-l))_N]$
Complex Conjugation	$\tilde{x}^*[n]$	$\tilde{X}^*[-k]$	Complex Conjugation	$x^*[n]$	$X^*[((-k))_N]$

Properties (Continued)

Time-Reversal and Complex Conjugation	$\tilde{x}^*[-n]$	$\tilde{X}^*[k]$	Time-Reversal and Complex Conjugation	$x^*[((-n))_N]$	$X^*[k]$
Real Part	$\mathrm{Re}\{\tilde{x}[n]\}$	$\tilde{X}_{ep}[k] = \frac{1}{2}\left(\tilde{X}[k] + \tilde{X}^*[-k]\right)$	Real Part	$\mathrm{Re}\{x[n]\}$	$X_{ep}[k] = \frac{1}{2}\left(X[k] + X^*[((-k))_N]\right)$
Imaginary Part	$j\,\mathrm{Im}\{\tilde{x}[n]\}$	$\tilde{X}_{op}[k] = \frac{1}{2}\left(\tilde{X}[k] - \tilde{X}^*[-k]\right)$	Imaginary Part	$j\,\mathrm{Im}\{x[n]\}$	$X_{op}[k] = \frac{1}{2}\left(X[k] - X^*[((-k))_N]\right)$
Even Part	$\tilde{x}_{ep}[n] = \frac{1}{2}\left(\tilde{x}[n] + \tilde{x}^*[-n]\right)$	$\mathrm{Re}\{\tilde{X}[k]\}$	Even Part	$x_{ep}[n] = \frac{1}{2}\left(x[n] + x^*[((-n))_N]\right)$	$\mathrm{Re}\{X[k]\}$
Odd Part	$\tilde{x}_{op}[n] = \frac{1}{2}\left(\tilde{x}[n] - \tilde{x}^*[-n]\right)$	$j\,\mathrm{Im}\{\tilde{X}[k]\}$	Odd Part	$x_{op}[n] = \frac{1}{2}\left(x[n] - x^*[((-n))_N]\right)$	$j\,\mathrm{Im}\{X[k]\}$
Symmetry for Real Sequence	$\tilde{x}[n] = \tilde{x}^*[n]$	$\tilde{X}[k] = \tilde{X}^*[-k]$ $\begin{cases} \mathrm{Re}\{\tilde{X}[k]\} = \mathrm{Re}\{\tilde{X}[-k]\} \\ \mathrm{Im}\{\tilde{X}[k]\} = -\mathrm{Im}\{\tilde{X}[-k]\} \end{cases}$ $\begin{cases} \lvert\tilde{X}[k]\rvert = \lvert\tilde{X}[-k]\rvert \\ \angle\tilde{X}[k] = -\angle\tilde{X}[-k] \end{cases}$	Symmetry for Real Sequence	$x[n] = x^*[n]$	$X[k] = X^*[((-k))_N]$ $\begin{cases} \mathrm{Re}\{X[k]\} = \mathrm{Re}\{X[((-k))_N]\} \\ \mathrm{Im}\{X[k]\} = -\mathrm{Im}\{X[((-k))_N]\} \end{cases}$ $\begin{cases} \lvert X[k]\rvert = \lvert X[((-k))_N]\rvert \\ \angle X[k] = -\angle X[((-k))_N] \end{cases}$
Parseval's Identity	$\sum_{n=0}^{N-1}\tilde{x}_1[n]\tilde{x}_2^*[n] = \frac{1}{N}\sum_{k=0}^{N-1}\tilde{X}_1[k]\tilde{X}_2^*[k]$ $\sum_{n=0}^{N-1}\lvert\tilde{x}[n]\rvert^2 = \frac{1}{N}\sum_{k=0}^{N-1}\lvert\tilde{X}[k]\rvert^2$		Parseval's Identity	$\sum_{n=0}^{N-1}x_1[n]x_2^*[n] = \frac{1}{N}\sum_{k=0}^{N-1}X_1[k]X_2^*[k]$ $\sum_{n=0}^{N-1}\lvert x[n]\rvert^2 = \frac{1}{N}\sum_{k=0}^{N-1}\lvert X[k]\rvert^2$	

Duality

$$\text{If } x \xrightarrow{DFT} X, \text{ then } \{X[n]\}_{n=0}^{N-1} \xrightarrow{DFT} N\{x[((-k))_N]\}_{k=0}^{N-1}$$

Duality

$$\text{If } x \xrightarrow{DFT} X, \text{ then } \{X[n]\}_{n=0}^{N-1} \xrightarrow{DFT} N\{x[((-k))_N]\}_{k=0}^{N-1}$$

$$\tilde{x}[n] \overset{\mathcal{DFS}}{\longleftrightarrow} \tilde{X}[k],$$

$$\tilde{X}[n] \overset{\mathcal{DFS}}{\longleftrightarrow} N\tilde{x}[-k].$$

Proof of Duality

$$\text{DFT of } \{x[n]\}_{n=0}^{N-1} \text{ is } \quad X[k] = \sum_{p=0}^{N-1} x[p]e^{-j\frac{2\pi}{N}kp}; \qquad k \le 0 \le N-1$$

$$\text{DFT of } \{X[n]\}_{n=0}^{N-1} \text{ is } \quad \sum_{n=0}^{N-1}\underbrace{\sum_{p=0}^{N-1} x[p]e^{-j\frac{2\pi}{N}pn}}_{X[n]} e^{-j\frac{2\pi}{N}kn}, \qquad k \le 0 \le N-1$$

$$= \sum_{p=0}^{N-1} x[p] \underbrace{\sum_{n=0}^{N-1} e^{-j\frac{2\pi}{N}(p+k)n}}_{\substack{N \text{ for } ((p+k))_N = 0, \\ 0 \text{ otherwise}}}$$

$$((p+k))_N = 0 \quad \text{for } 0 \le p \;\&\; k \le N-1 \quad \Rightarrow p = ((-k))_N$$

$$p = -k + mN = ((-k))_N + rN + mN = ((-k))_N \quad \textit{because } 0 \le p \le N-1$$

$$\therefore \text{ DFT of } \{X[n]\}_{n=0}^{N-1} \text{ is } N\{x[((-k))_N]\}_{k=0}^{N-1}$$

Circular Convolution

- Circular Convolution:

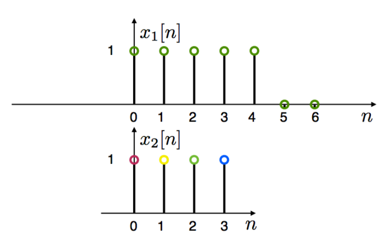
$$x_1[n] \circledN x_2[n] \triangleq \sum_{m=0}^{N-1} x_1[m]x_2[((n-m))_N]$$

For two signals of length N

Note: Circular convolution is commutative

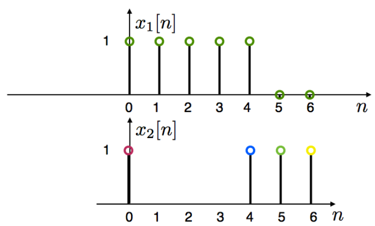
$$x_2[n] \circledN x_1[n] = x_1[n] \circledN x_2[n]$$

Compute Circular Convolution Sum



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

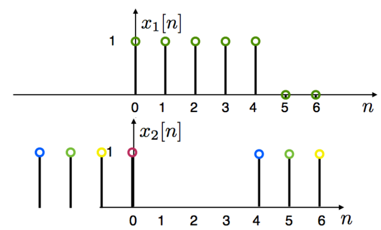
Compute Circular Convolution Sum



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

Compute Circular Convolution Sum

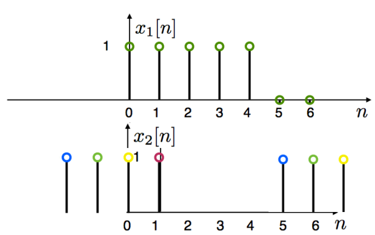
y[0]=2



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

Compute Circular Convolution Sum

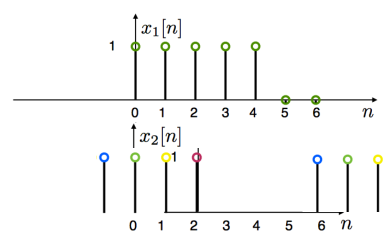
y[0]=2
y[1]=2



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

Compute Circular Convolution Sum

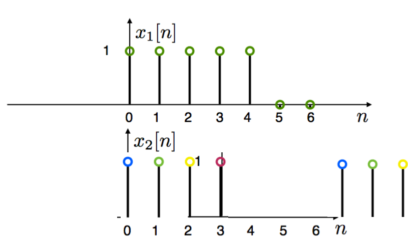
y[0]=2
y[1]=2
y[2]=3



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

Compute Circular Convolution Sum

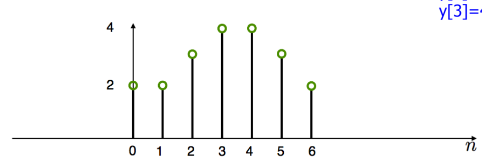
y[0]=2
y[1]=2
y[2]=3
y[3]=4



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

Result

y[0]=2
y[1]=2
y[2]=3
y[3]=4



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

Circular Convolution

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

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

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

Multiplication

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

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

Linear Convolution

- Next….
  - Using DFT, circular convolution is easy
    - Matrix multiplication… more later
  - 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)

Linear Convolution

- We start with two non-periodic sequences:

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

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

Linear Convolution

- We start with two non-periodic sequences:

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

  - 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 $0 \le n \le L+P-2$ with length M=L+P-1

Requires LP 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

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

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

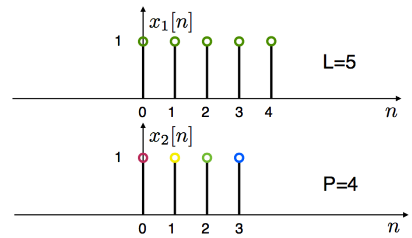
Linear Convolution via Circular Convolution

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

Linear convolution via circular

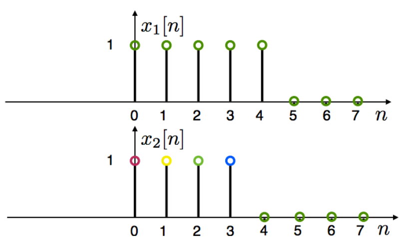
$$y[n] = x[n] * y[n] = \begin{cases} x_{zp}[n] \circledM h_{zp}[n] & 0 \le n \le M-1 \\ 0 & \text{otherwise} \end{cases}$$

Example



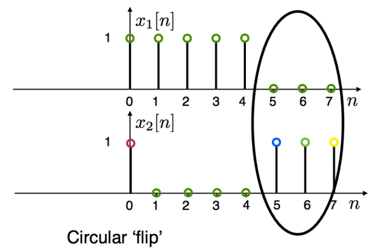
M = L + P - 1 = 8

Example



M = L + P - 1 = 8

Example



Circular 'flip'

M = L + P - 1 = 8

$$y[n] = x_1[n] \circledS x_2[n] = x_1[n] * x_2[n]$$

Linear Convolution with DFT

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

$$\begin{aligned} x[n] * h[n] &= x_{zp}[n] \circledM h_{zp}[n] \\ &= \mathcal{DFT}^{-1}\{\mathcal{DFT}\{x_{zp}[n]\} \cdot \mathcal{DFT}\{h_{zp}[n]\}\} \end{aligned}$$

for 0 ≤ n ≤ M-1, M=L+P-1

Linear Convolution with DFT

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

$$\begin{aligned} x[n] * h[n] &= x_{zp}[n] \circledM h_{zp}[n] \\ &= \mathcal{DFT}^{-1}\{\mathcal{DFT}\{x_{zp}[n]\} \cdot \mathcal{DFT}\{h_{zp}[n]\}\} \end{aligned}$$

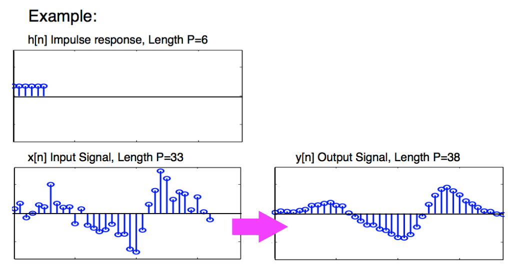
for 0 ≤ n ≤ M-1, M=L+P-1

- Advantage: DFT can be computed with $N\log_2 N$ 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
  - Combine the results

Block Convolution



Overlap-Add Method

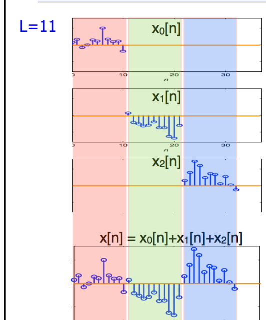
- Decompose into non-overlapping segments

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

- The input signal is the sum of segments

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

Example



Overlap-Add Method

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

- 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 N=L+P-1
    - h[n] has length P
    - $x_r[n]$ has length L

Overlap-Add Method

- We can compute $x_r[n]*h[n]$ using linear convolution
- Using the DFT:
  - Zero-pad $x_r[n]$ to length N
  - Zero-pad h[n] to length N and compute $DFT_N\{h_{zp}[n]\}$
    - Only need to do once: WHY?

Overlap-Add Method

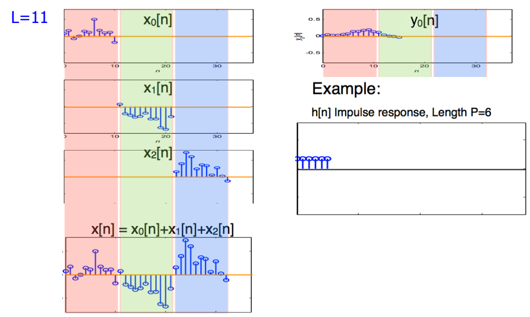
- We can compute $x_r[n]*h[n]$ using linear convolution
- Using the DFT:
  - Zero-pad $x_r[n]$ to length N
  - Zero-pad h[n] to length N and compute $DFT_N\{h_{zp}[n]\}$
    - Only need to do once: WHY?
  - Compute:

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

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

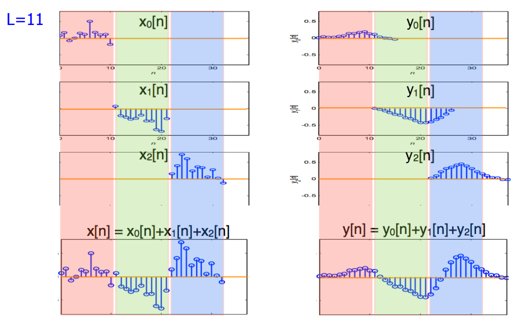
Example of Overlap-Add

L+P-1=16



Example of Overlap-Add

L+P-1=16



Overlap-Save Method

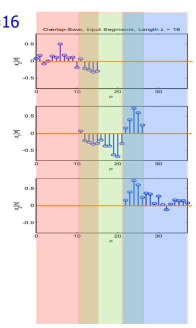
- Basic idea:
- Split input into (P-1) overlapping segments with length L+P-1

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

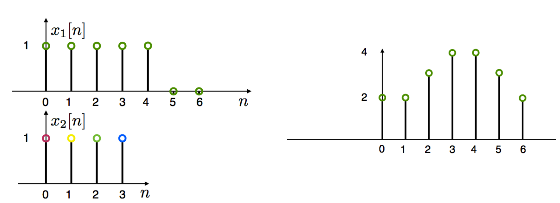
- Perform circular convolution in each segment, and keep the L sample portion which is a valid linear convolution

Example of Overlap-Save

L+P-1=16

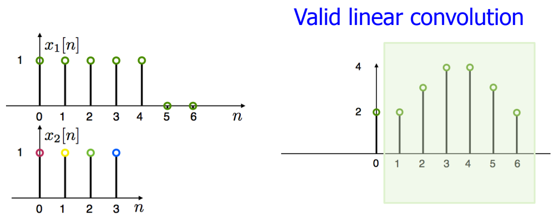


Recall Circular Convolution Sum



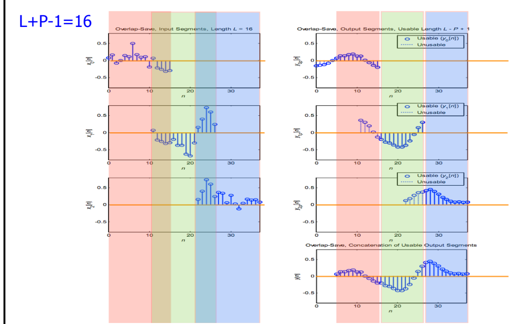
$$x_1[n] \circledN x_2[n] \triangleq \sum_{m=0}^{N-1} x_1[m] x_2[((n-m))_N]$$

Recall Circular Convolution Sum

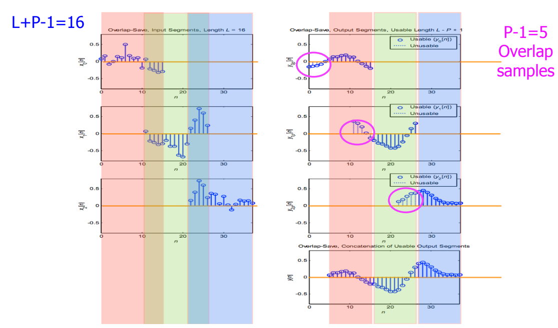


$$x_1[n] \circledN x_2[n] \triangleq \sum_{m=0}^{N-1} x_1[m] x_2[((n-m))_N]$$

Example of Overlap-Save

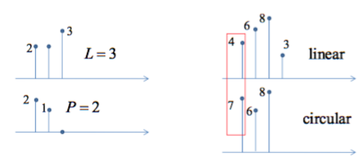


Example of Overlap-Save



Circular to Linear Convolution

- An L-point sequence circularly convolved with a P-point 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.



Big Ideas

- Discrete Fourier Transform (DFT)
  - For finite signals assumed to be zero outside of defined length
  - N-point DFT is sampled DTFT at N points
  - Useful properties allow easier linear convolution
- 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

Admin

- Project
  - Work in groups of up to 2
    - Can work alone if you want
    - Use Piazza to find partners
    - Report your groups to me by midnight
      - taniak@seas.upenn.edu