ESE 5310: Digital Signal Processing

Lecture 19: April 9, 2024

Discrete Fourier Transform, Pt 2

Today

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

Discrete Fourier Transform

- The DFT

$$\boxed{W_N \triangleq e^{-j2\pi/N}}$$

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

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

- It is understood that,

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

DTFT Vs. DFT

DTFT:

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

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

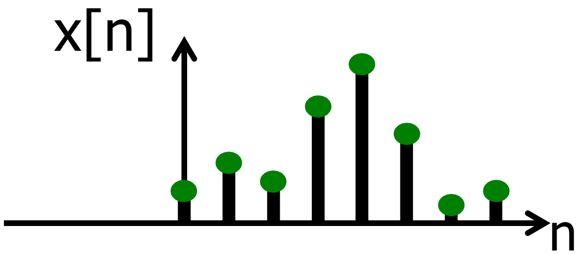
DFT:

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

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

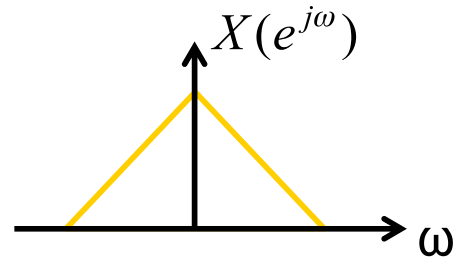
DFT Intuition

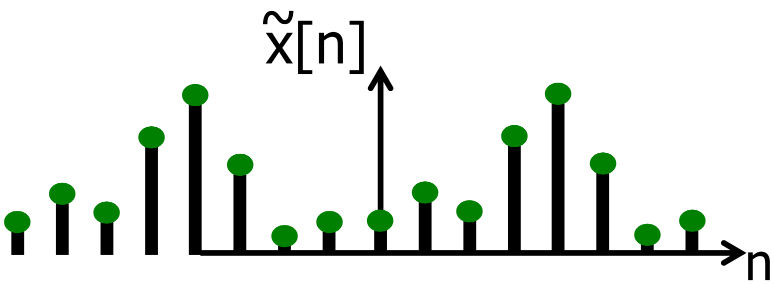
Time | **Transform** | **Frequency**



DTFT

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

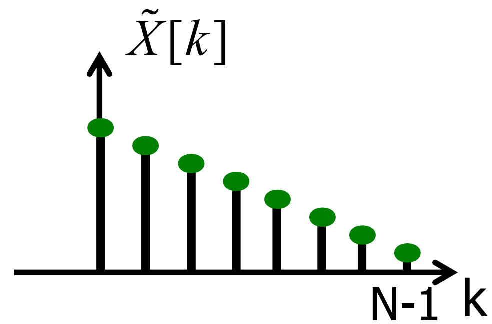


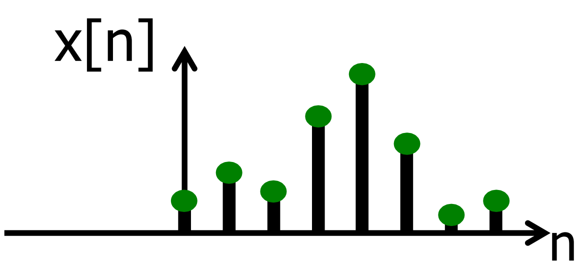


Periodic in N

DFS

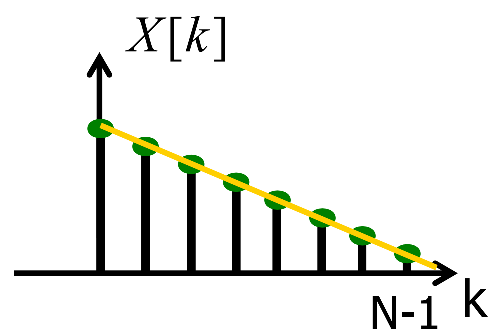
$$\tilde{x}[n] = \frac{1}{N}\sum_{k=0}^{N-1} \tilde{X}[k]W_N^{-kn}$$





DFT

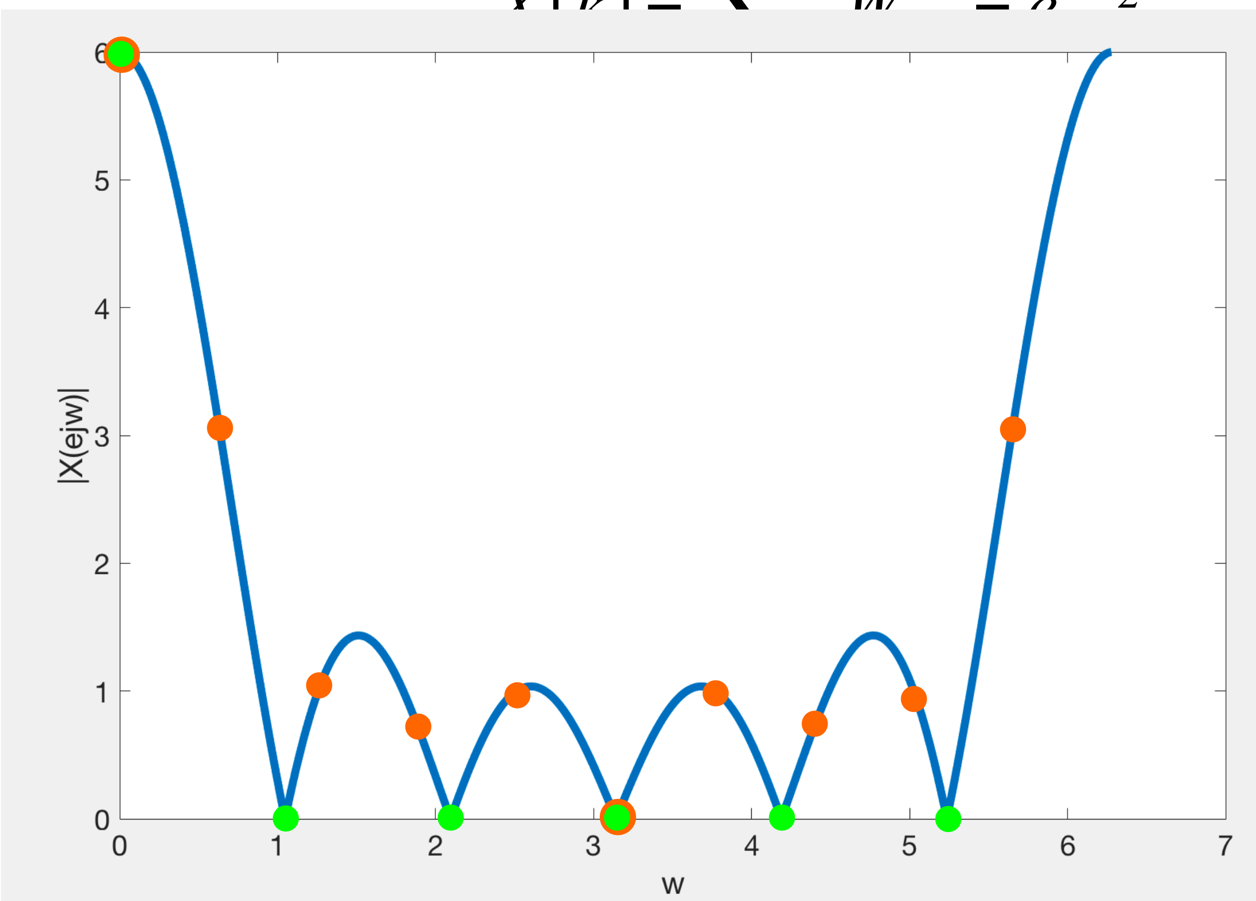
$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]W_N^{-kn}$$



DFT vs DTFT

- Back to example

$$X[k]=\sum^{5} W^{nk} = e^{-j\frac{\pi}{2}k}\frac{\sin\left(\frac{3\pi}{5}k\right)}{\sin\left(\frac{\pi}{10}k\right)}$$



“6-point” DFT

“10-point” DFT

Use `fftshift` to center around dc

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

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

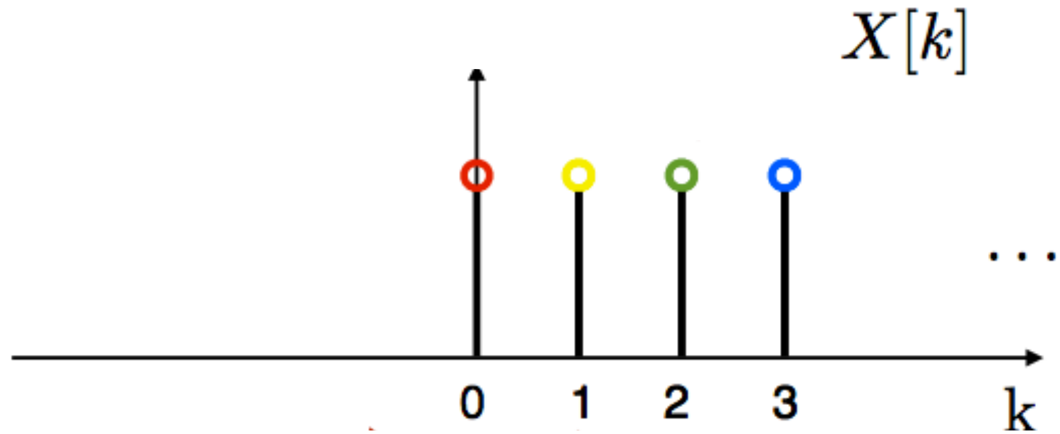
- Conjugate Symmetry for Real Signals

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

Example: Conjugate Symmetry

4-point DFT

–Symmetry

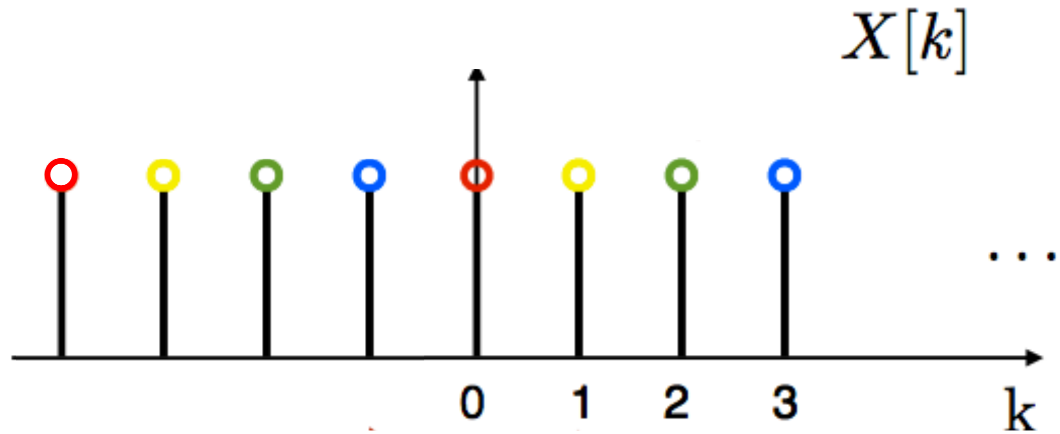


$$x[n] = x^*[n] \leftrightarrow X[k] = X^*[((-k))_N]$$

Example: Conjugate Symmetry

4-point DFT

–Symmetry

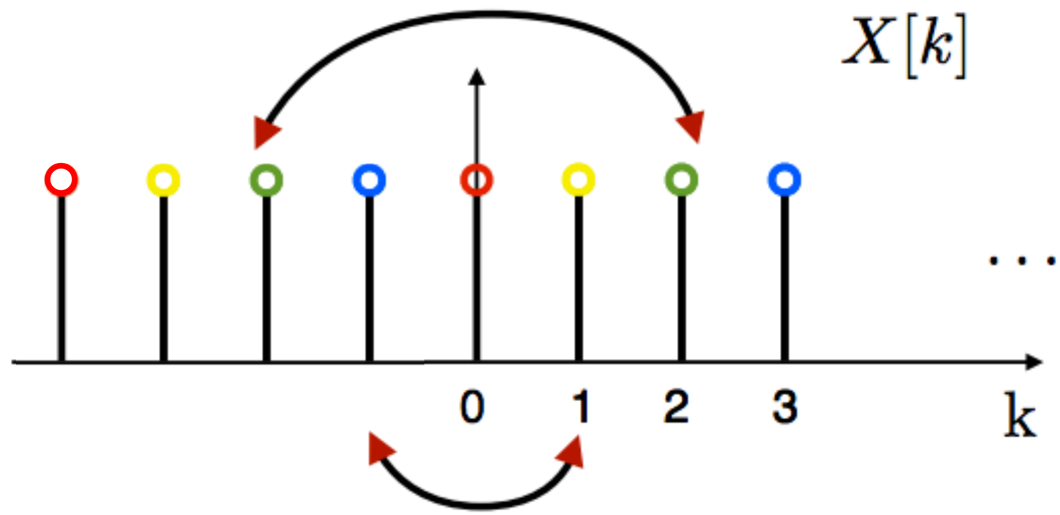


$$x[n] = x^*[n] \leftrightarrow X[k] = X^*[((-k))_N]$$

Example: Conjugate Symmetry

4-point DFT

–Symmetry

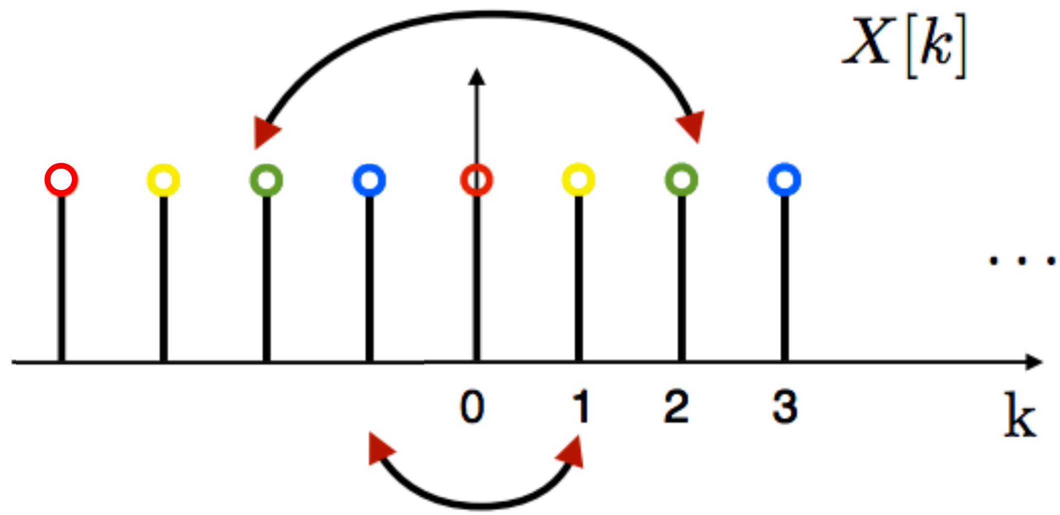


$$x[n] = x^*[n] \leftrightarrow X[k] = X^*[((-k))_N]$$

Example: Conjugate Symmetry

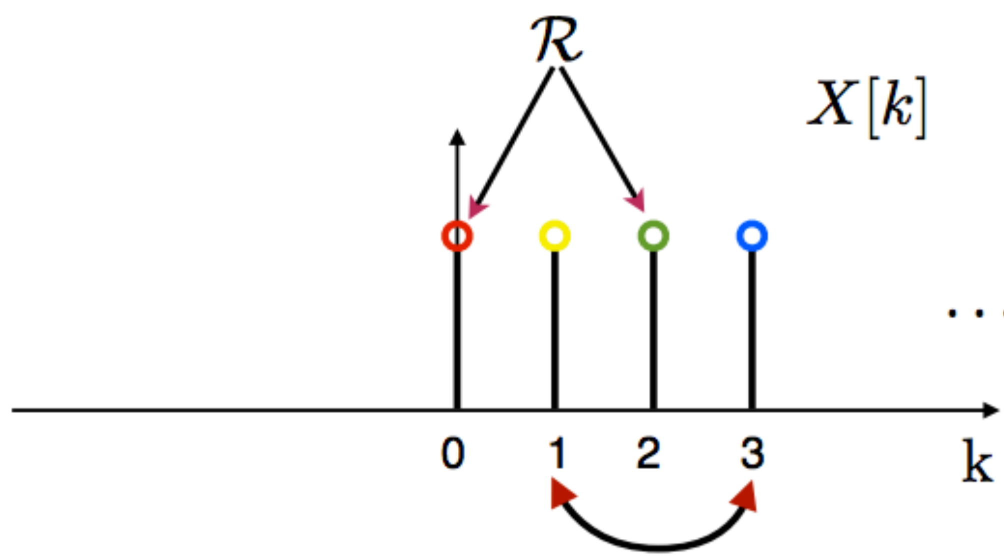
4-point DFT

–Symmetry



4-point DFT

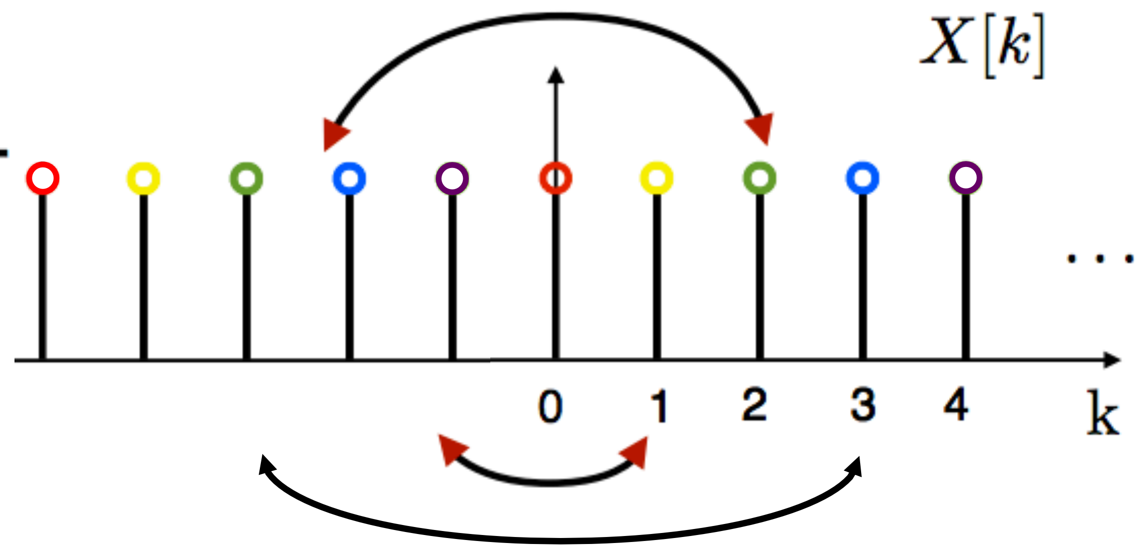
–Symmetry



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

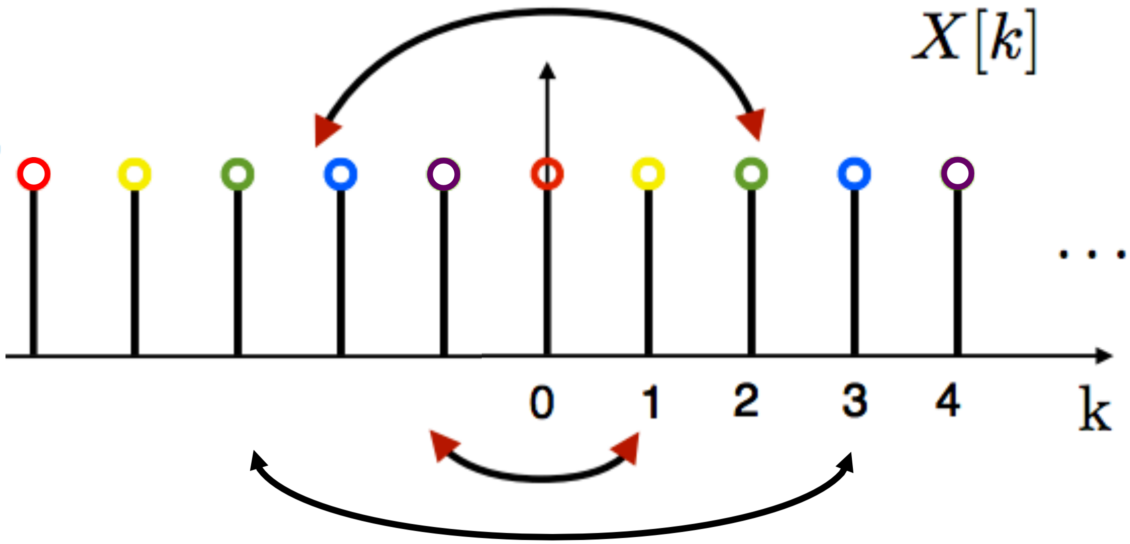
Example

5-point DFT
–Symmetry

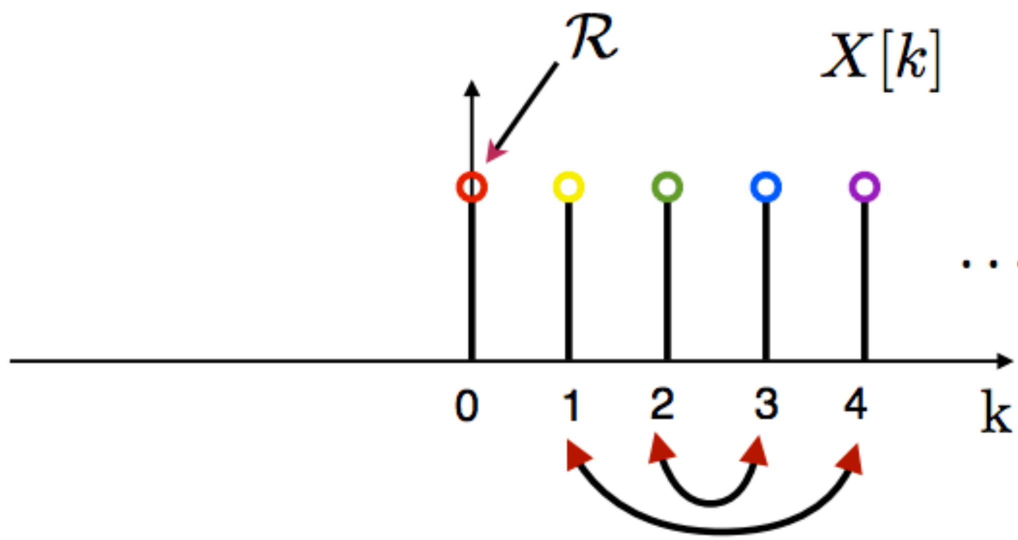


Example

5-point DFT
–Symmetry



5-point DFT
–Symmetry



Circular Convolution

- Circular Convolution:

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

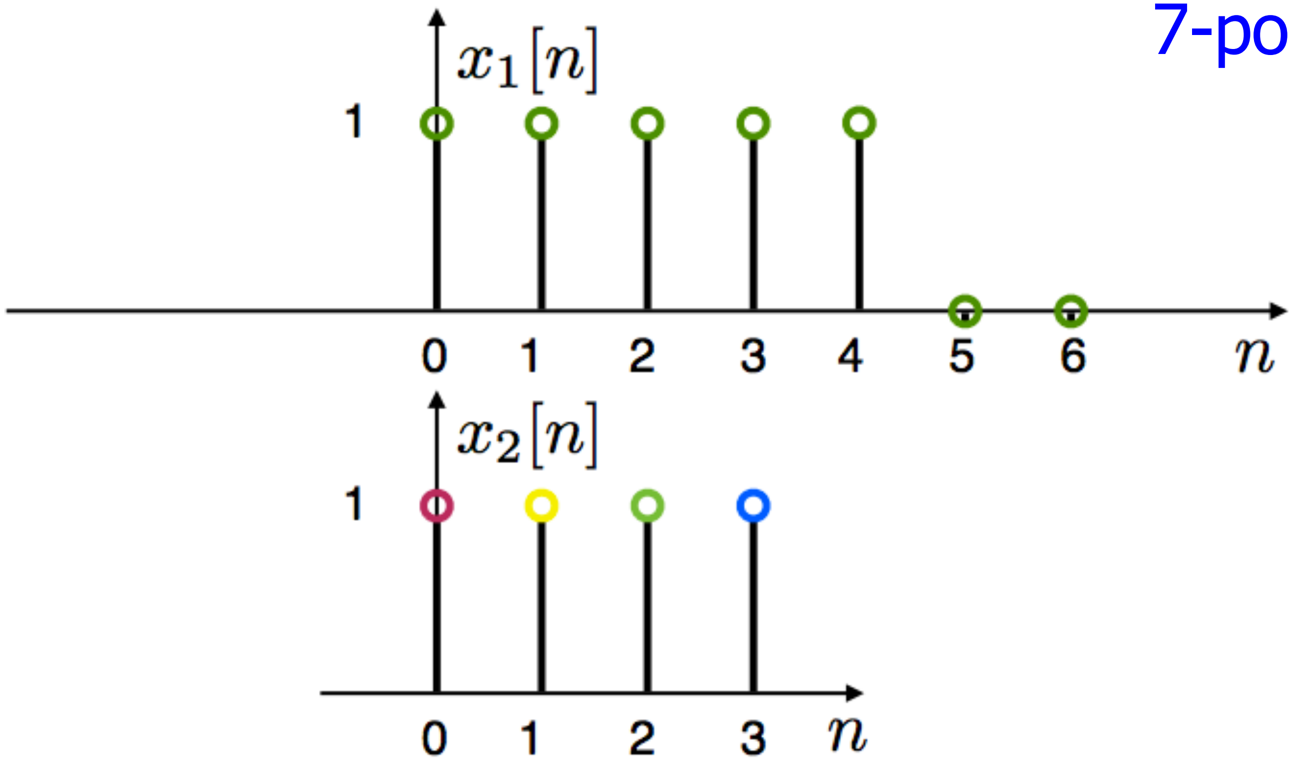
For two signals of length N

Note: Circular convolution is commutative

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

Compute Circular Convolution Sum

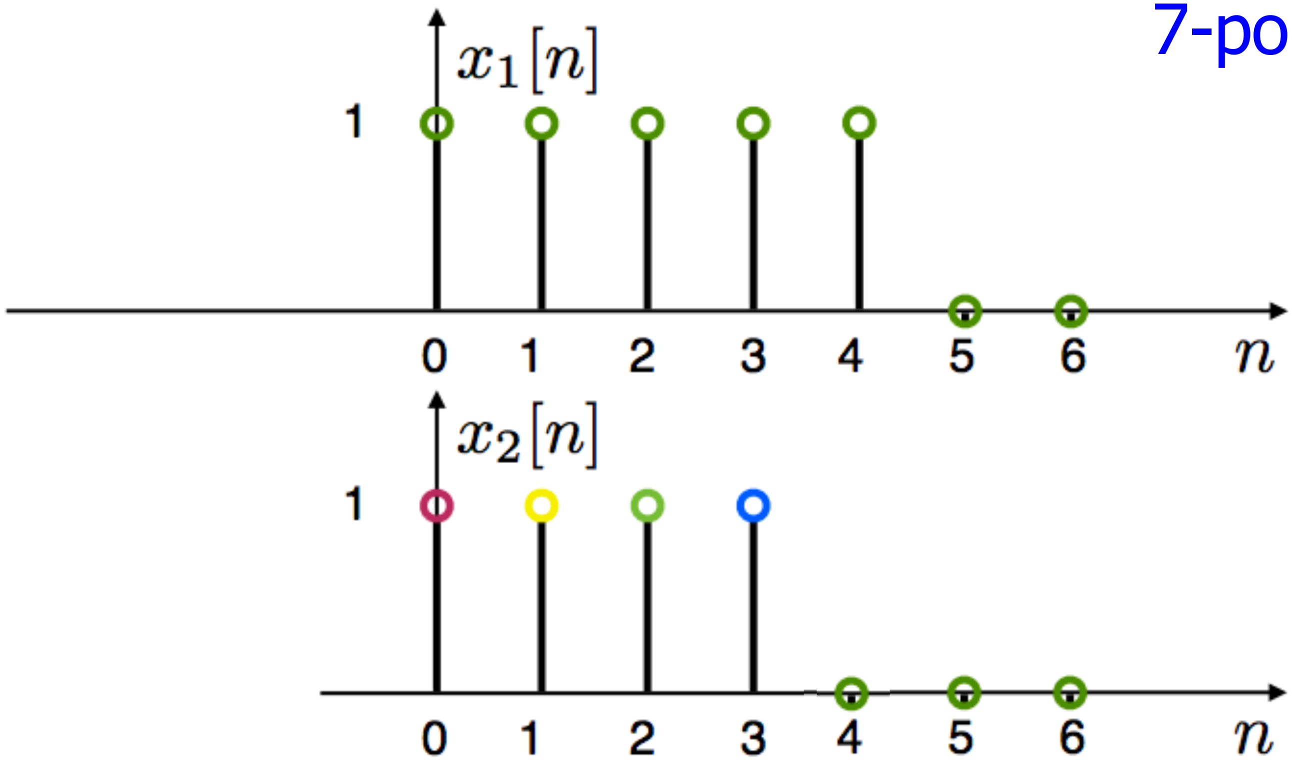
7-point convolution



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

Compute Circular Convolution Sum

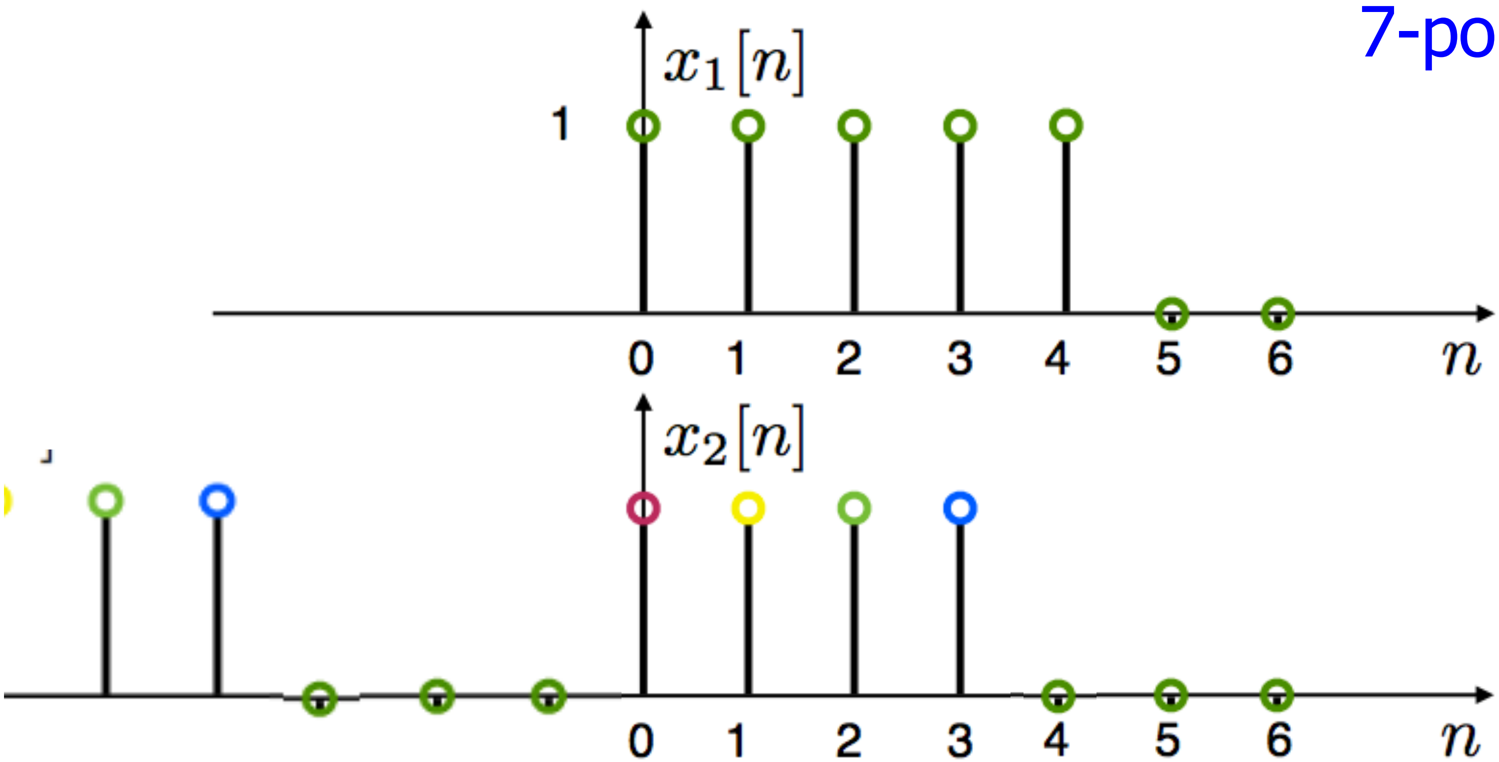
7-point convolution



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

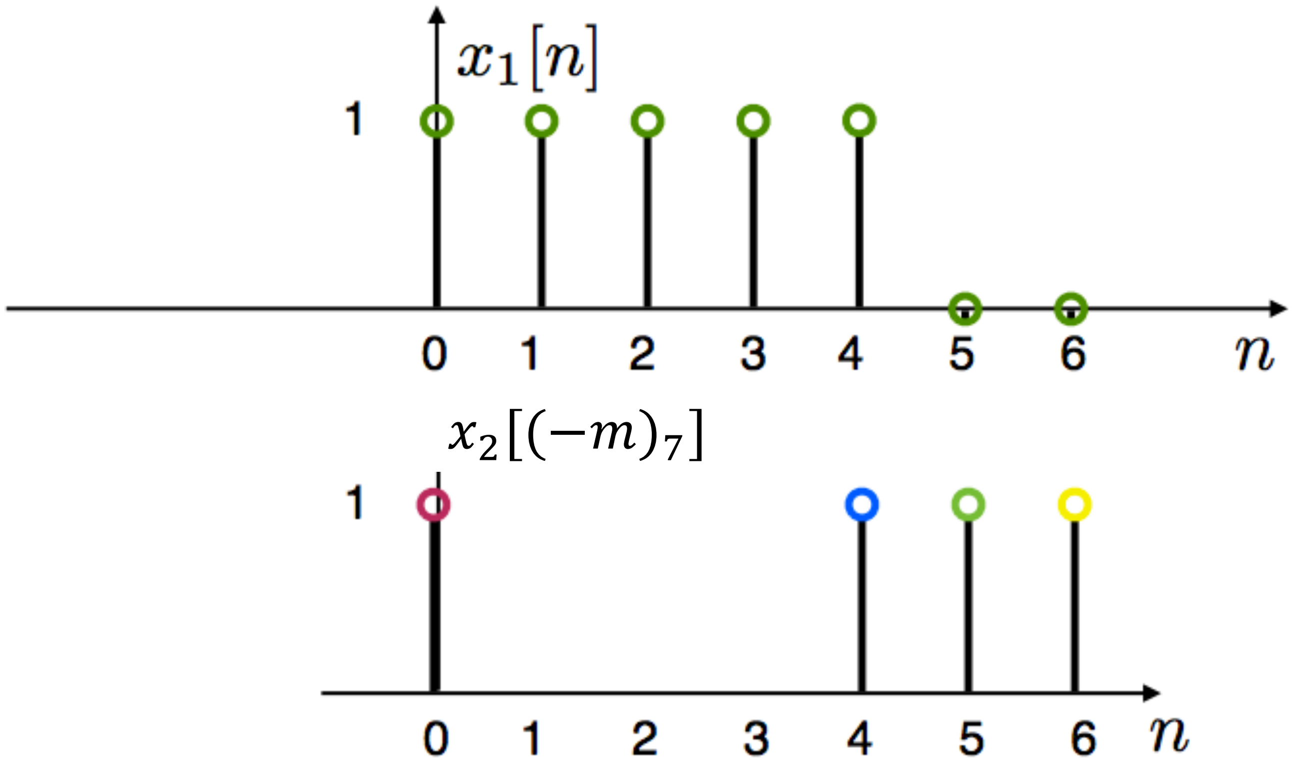
Compute Circular Convolution Sum

7-point convolution



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

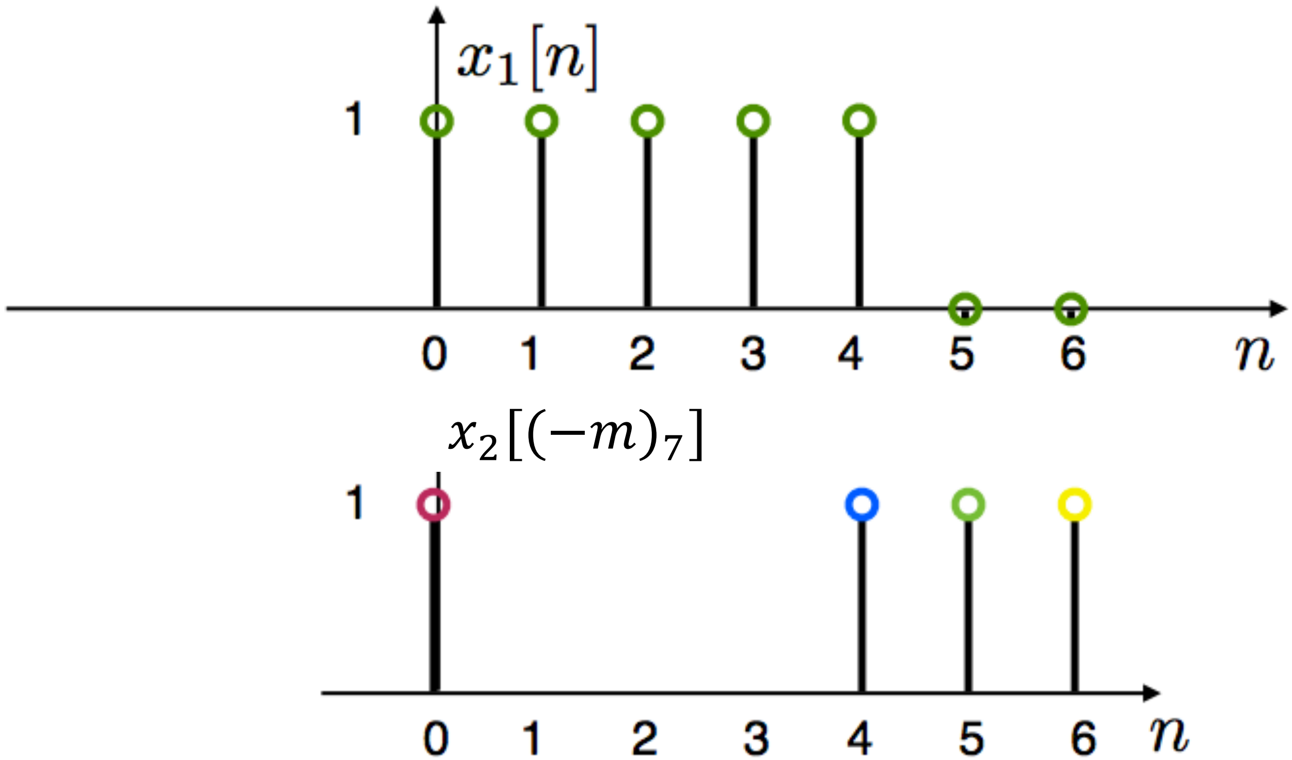
Compute Circular Convolution Sum



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

Compute Circular Convolution Sum

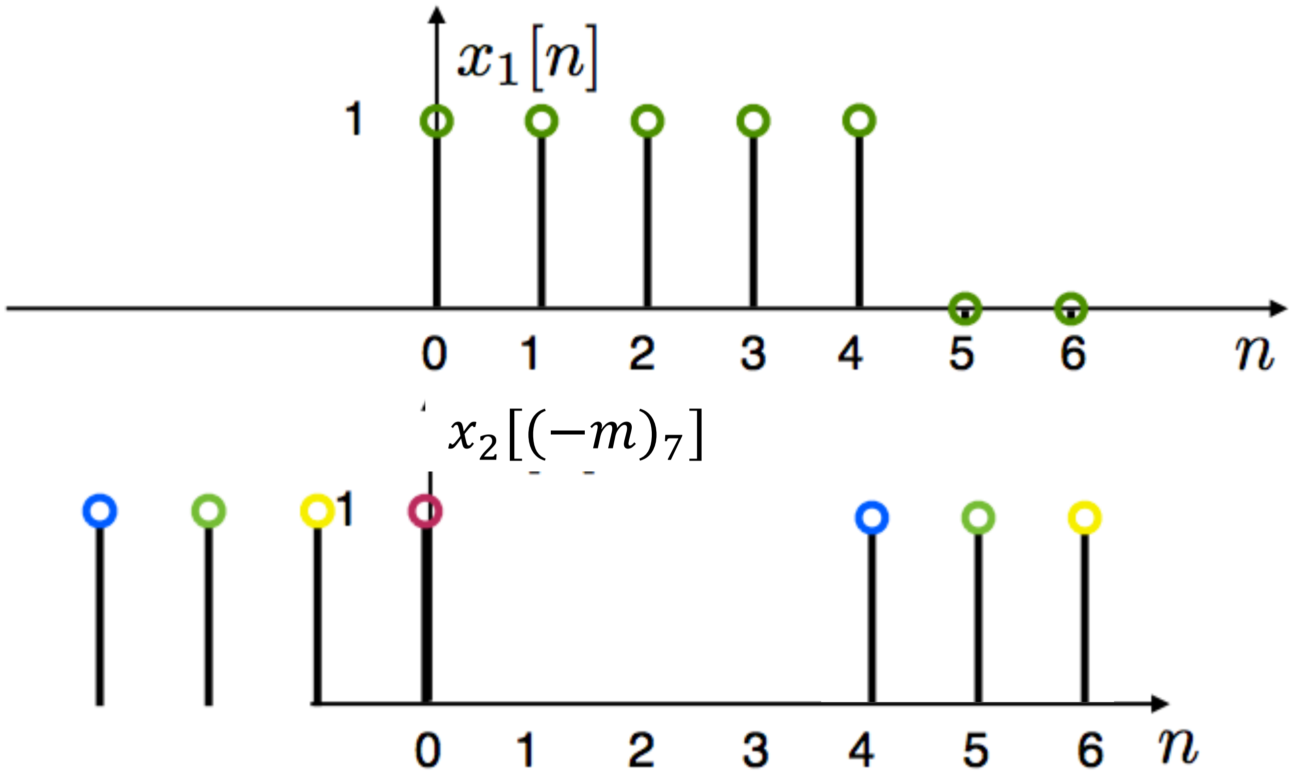
y[0]=2



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

Compute Circular Convolution Sum

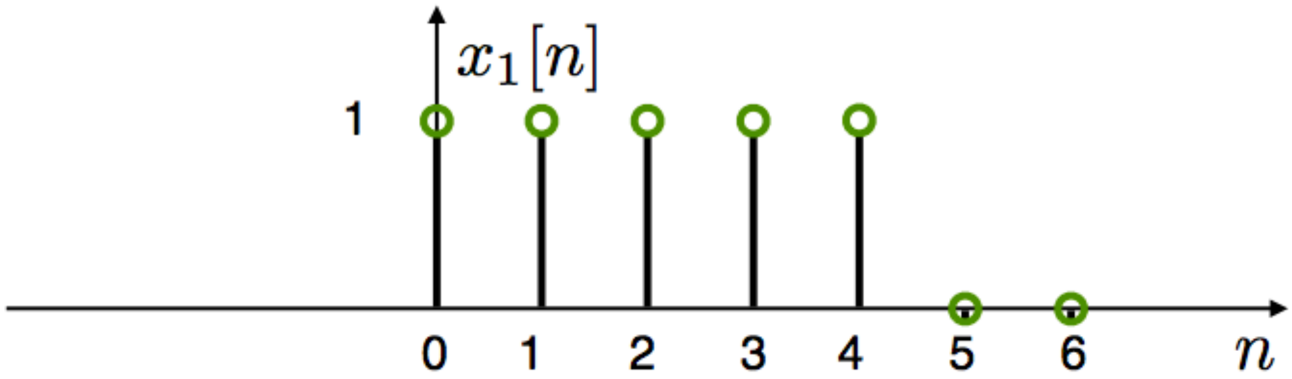
y[0]=2

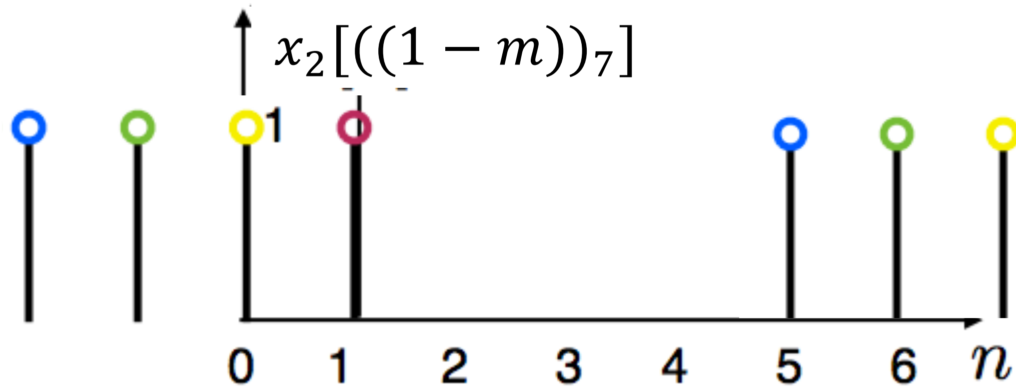


$$x_1[n] \; (N) \; x_2[n] \triangleq \sum_{m=0}^{N-1} x_1[m] x_2[((n-m))_N]$$

Compute Circular Convolution Sum

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

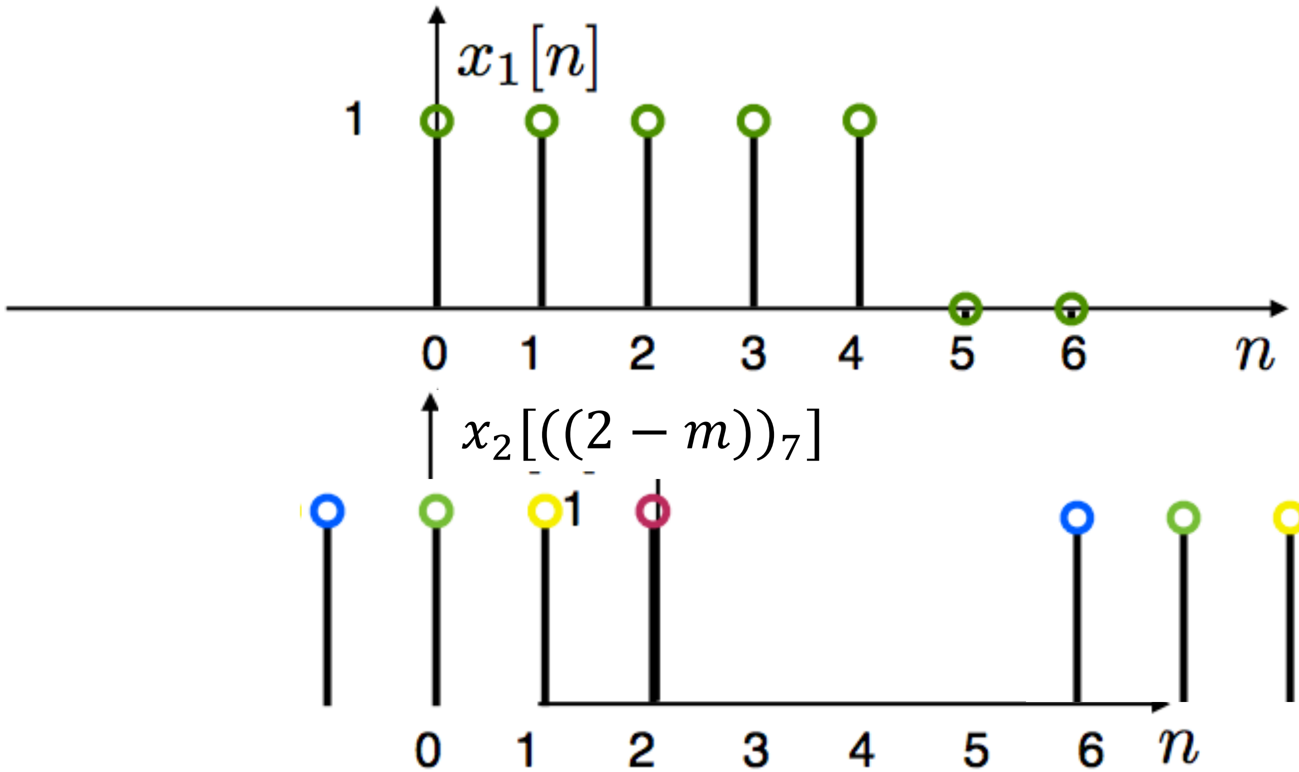




$$x_1[n] \circledN x_2[n] \triangleq \sum_{m=0}^{N-1} x_1[m] x_2[((n-m))_N]$$

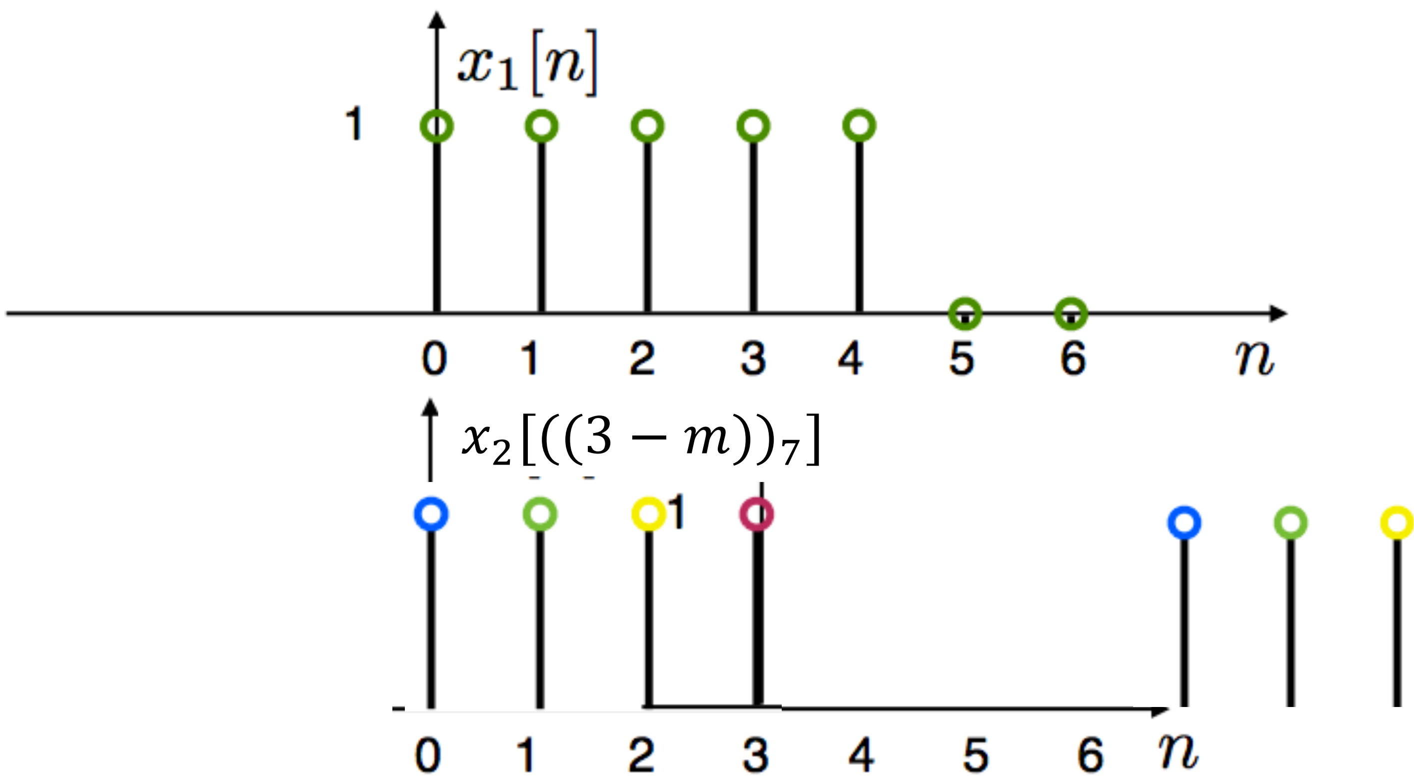
Compute Circular Convolution Sum

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



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

Compute Circular Convolution Sum

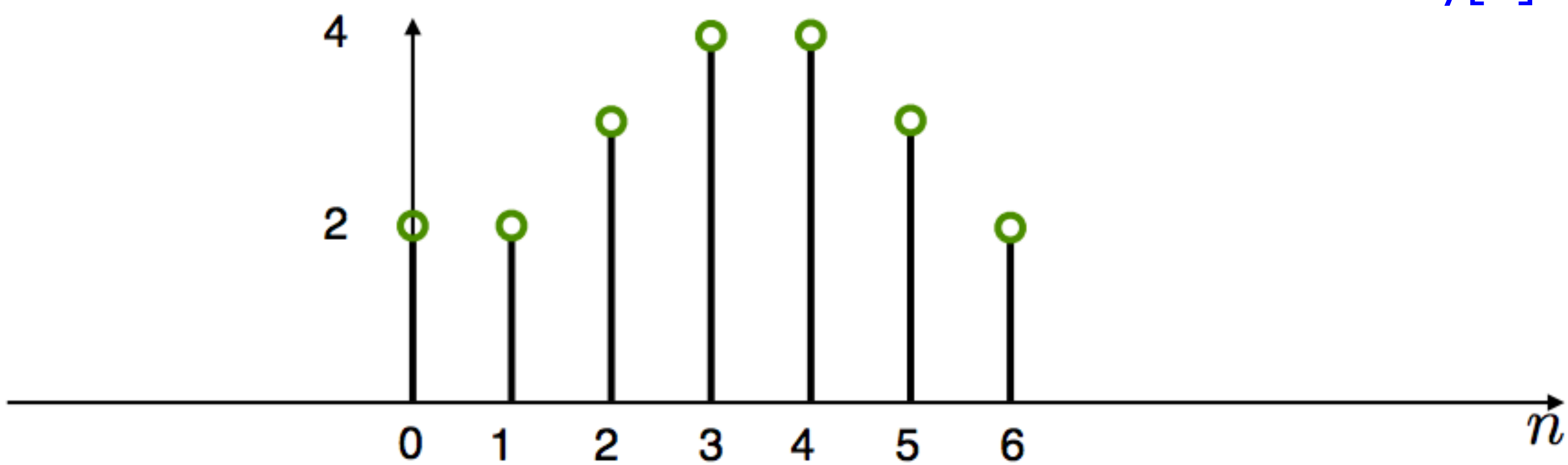


y[0]=2
y[1]=2
y[2]=3
y[3]=4

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

Result

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



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

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

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

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 circular convolution

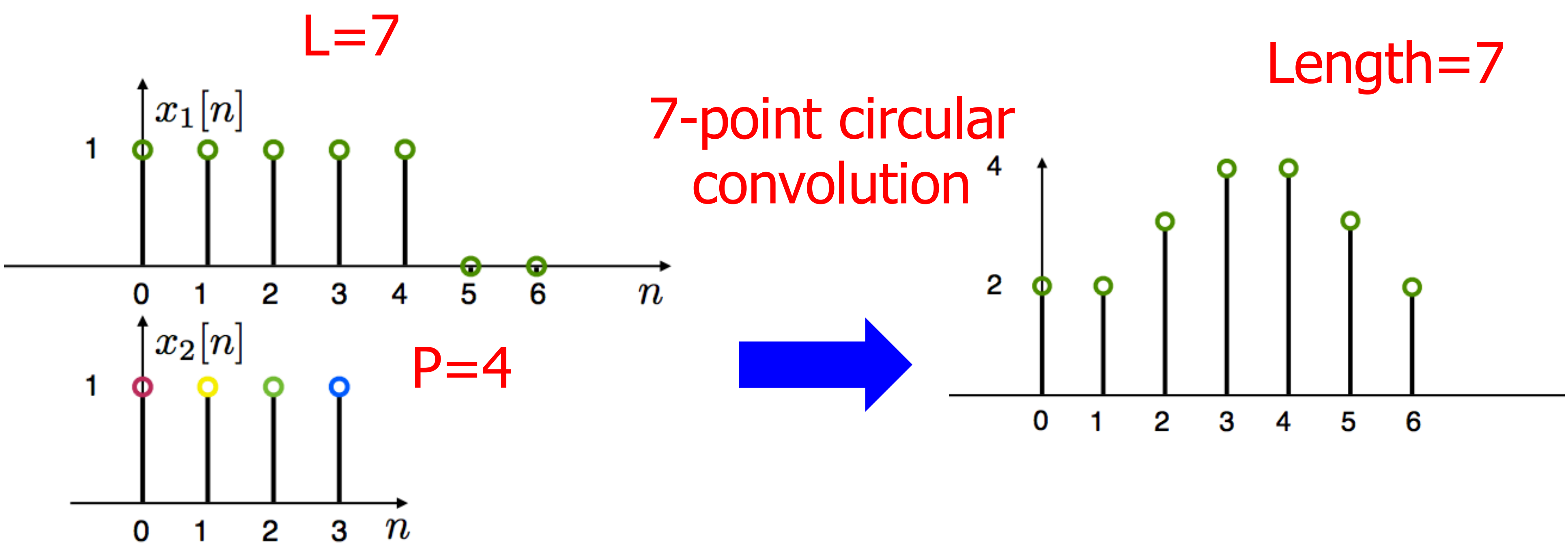
Linear Convolution

- We start with two non-periodic sequences:

$$x[n] \quad 0 \leq n \leq L-1$$
$$h[n] \quad 0 \leq n \leq P-1$$

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

Compute Circular Convolution Sum



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

Linear Convolution

- We start with two non-periodic sequences:

$$x[n] \quad 0 \leq n \leq L-1$$
$$h[n] \quad 0 \leq n \leq P-1$$

  - 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 $0 \leq n \leq L+P-2$ (ie. length M=L+P-1)

Requires L*P multiplications

Linear Convolution via Circular Convolution

- Zero-pad x[n] by P-1 zeros

$$x_{\mathrm{zp}}[n] = \begin{cases} x[n] & 0 \leq n \leq L-1 \\ 0 & L \leq n \leq L+P-2 \end{cases}$$

- Zero-pad h[n] by L-1 zeros

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

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

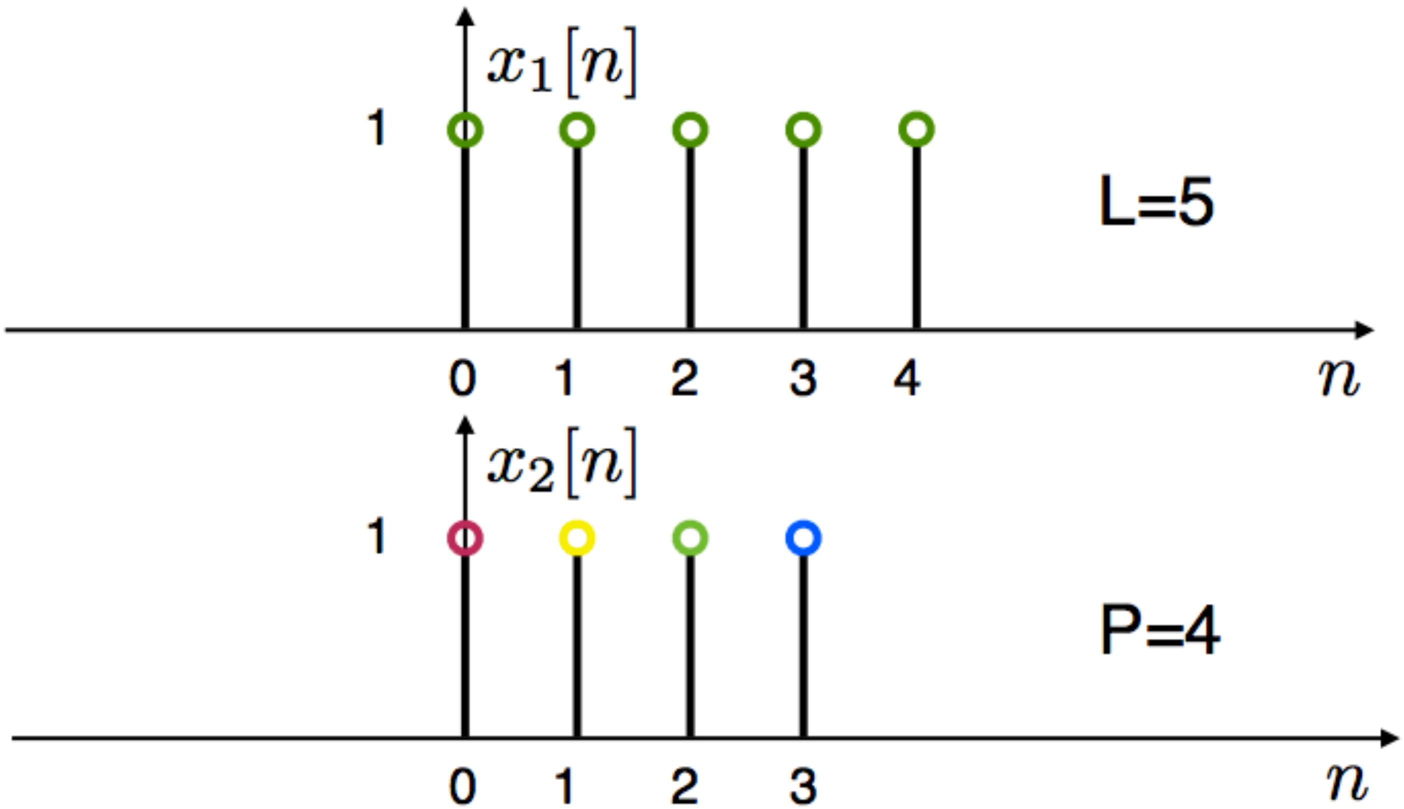
Linear Convolution via Circular Convolution

- 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

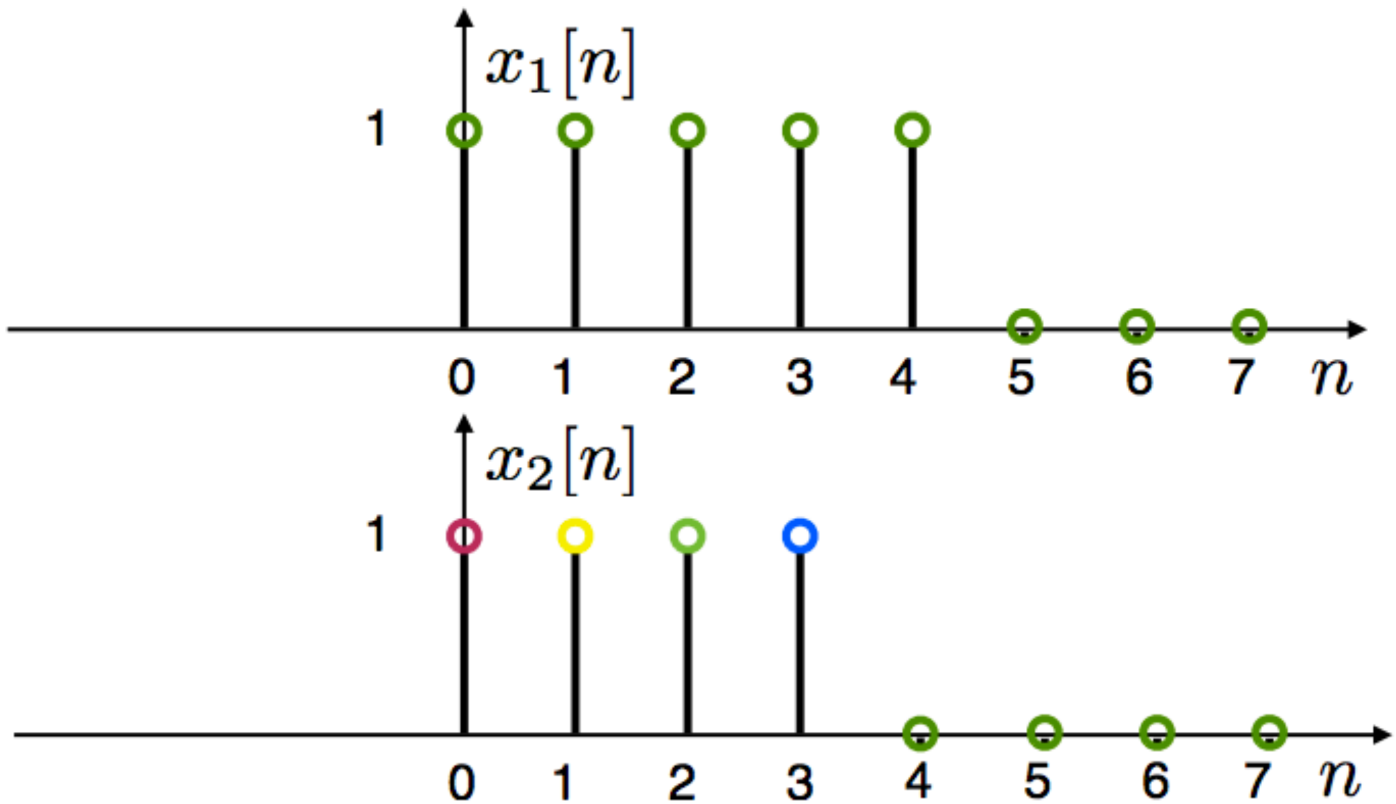
$$y[n] = x[n] * y[n] = \begin{cases} x_{zp}[n] \; \text{Ⓜ} \; h_{zp}[n] & 0 \le n \le M-1 \\ 0 & \text{otherwise} \end{cases}$$

Example



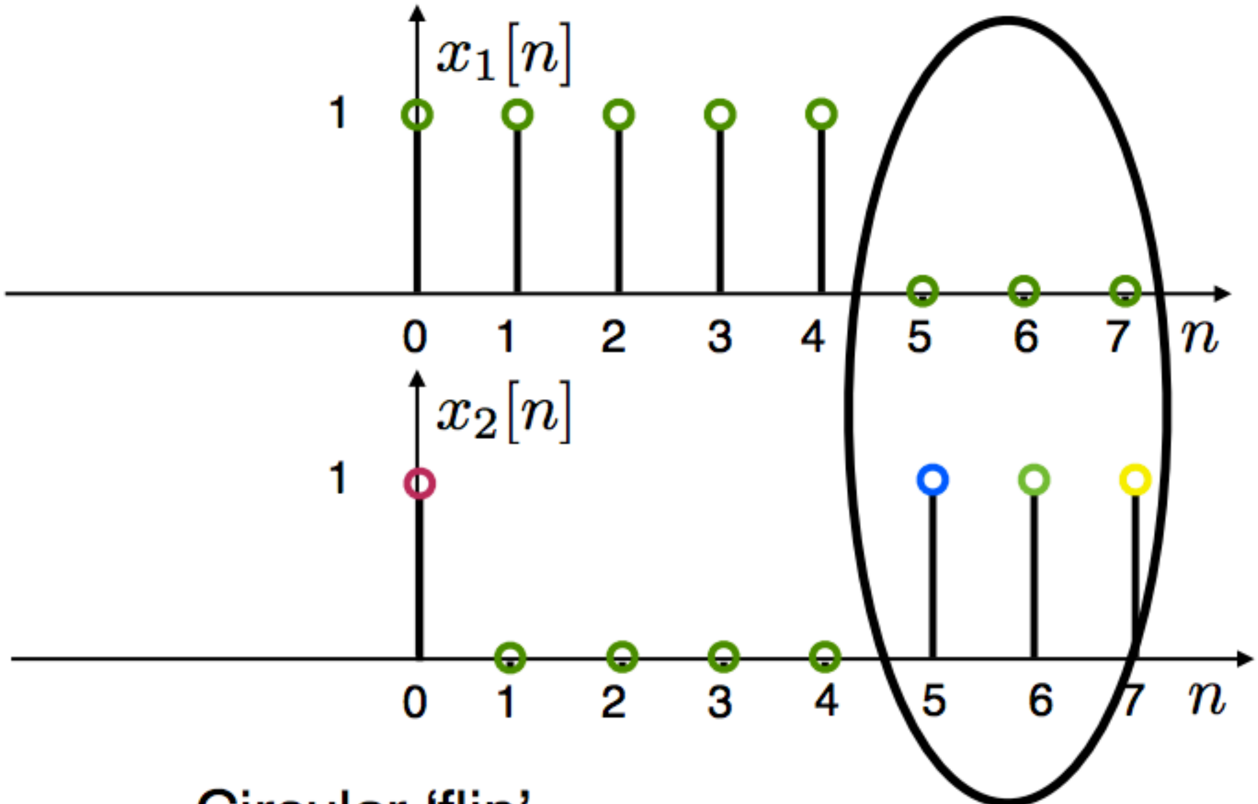
M = L + P - 1 = 8

Example



M = L + P - 1 = 8

Example



Circular 'flip'

M = L + P - 1 = 8

$$y[n] = x_1[n] \; ⑧ \; x_2[n] = x_1[n] * x_2[n]$$

Linear Convolution with DFT

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

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

for $0 \leq n \leq M-1$, $M=L+P-1$

Linear Convolution with DFT

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

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

for $0 \leq n \leq M-1$, $M=L+P-1$

- Advantage: DFT can be computed with $N\log_2 N$ complexity (FFT algorithm later!)

Linear Convolution with DFT

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

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

for $0 \le n \le M-1$, $M=L+P-1$

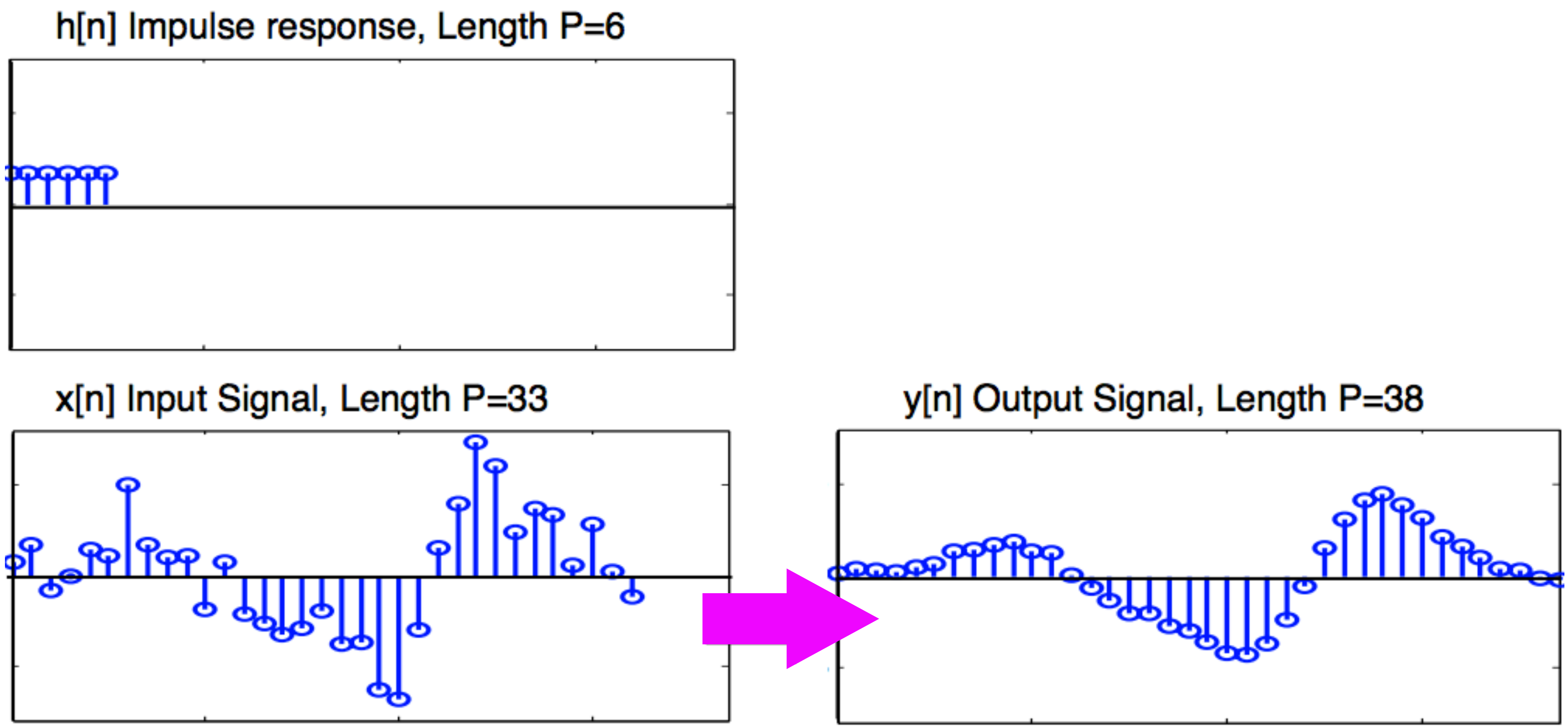
- Advantage: DFT can be computed with $N\log_2 N$ complexity (FFT algorithm later!)
- Drawback: Must wait for all the samples -- huge delay -- incompatible with real-time filtering

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 (via DFT)
  - Combine the results
    - Overlap-add
    - Overlap-save

Block Convolution

Example:



Overlap-Add Method

- Decompose into non-overlapping segments

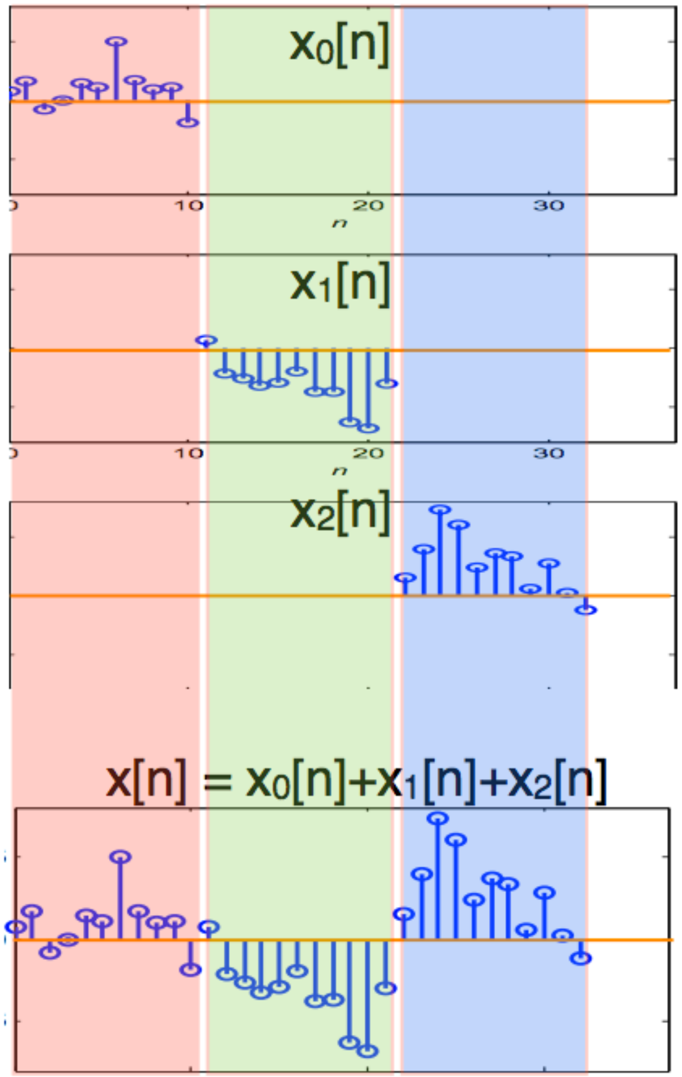
$$x_r[n] = \begin{cases} x[n] & rL \le n < (r+1)L \\ 0 & \text{otherwise} \end{cases}$$

- The input signal is the sum of segments

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

Example

L=11



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 M=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 circular convolution with the DFT
- Using the DFT:
  - Zero-pad $x_r[n]$ to length $M = L+P-1$
  - Zero-pad $h[n]$ to length M and compute $DFT_M\{h_{zp}[n]\}$
    - Only need to do once!

Overlap-Add Method

- 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:

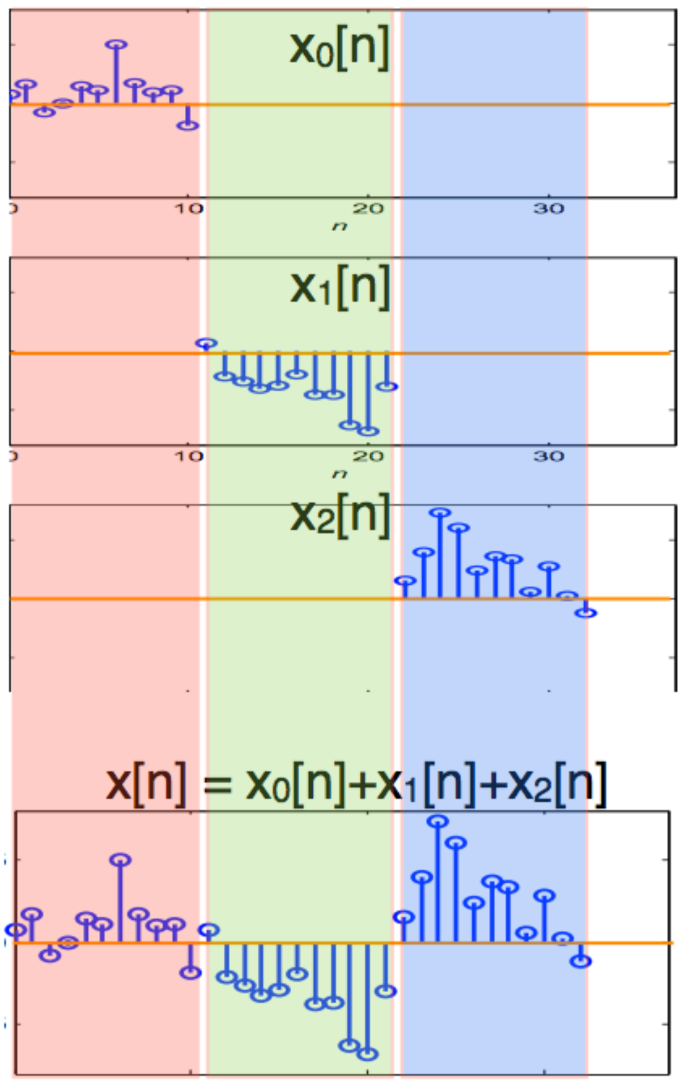
$$x_r[n] * h[n] = \mathrm{DFT}^{-1}\left\{\mathrm{DFT}\{x_{r,zp}[n]\} \cdot \mathrm{DFT}\{h_{zp}[n]\}\right\}$$

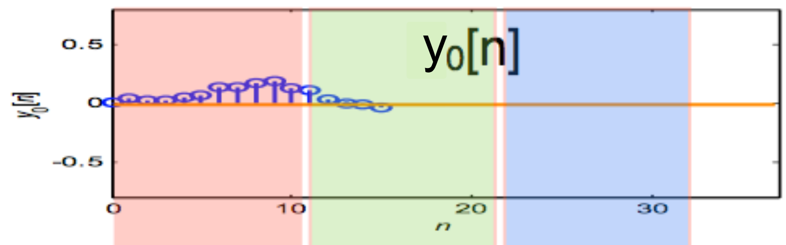
- Results are of length M=L+P-1
  - Neighboring results overlap by P-1
  - Add overlaps to get final sequence

Example of Overlap-Add

L+P-1=16

L=11





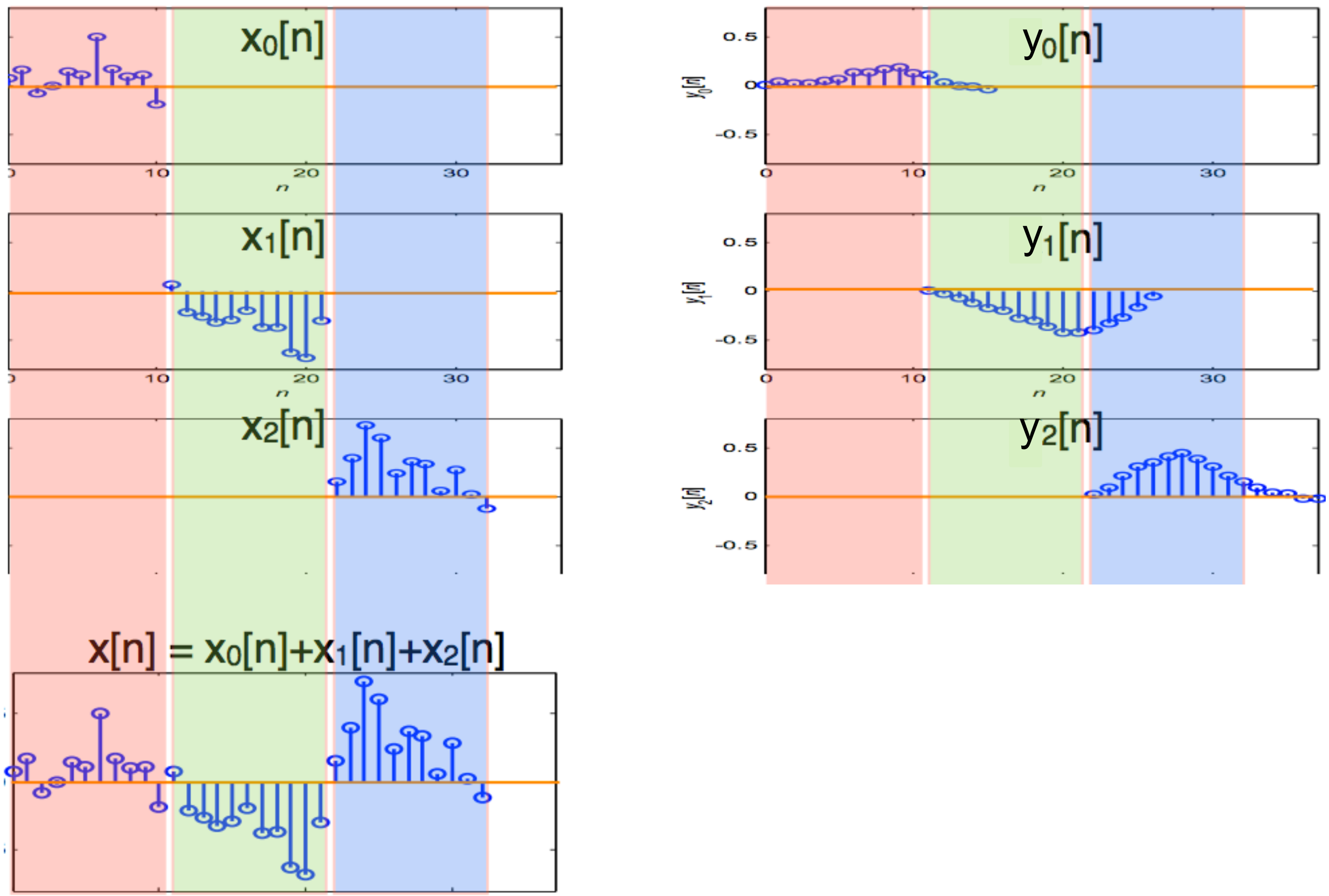
Example:

h[n] Impulse response, Length P=6

Example of Overlap-Add

L+P-1=16

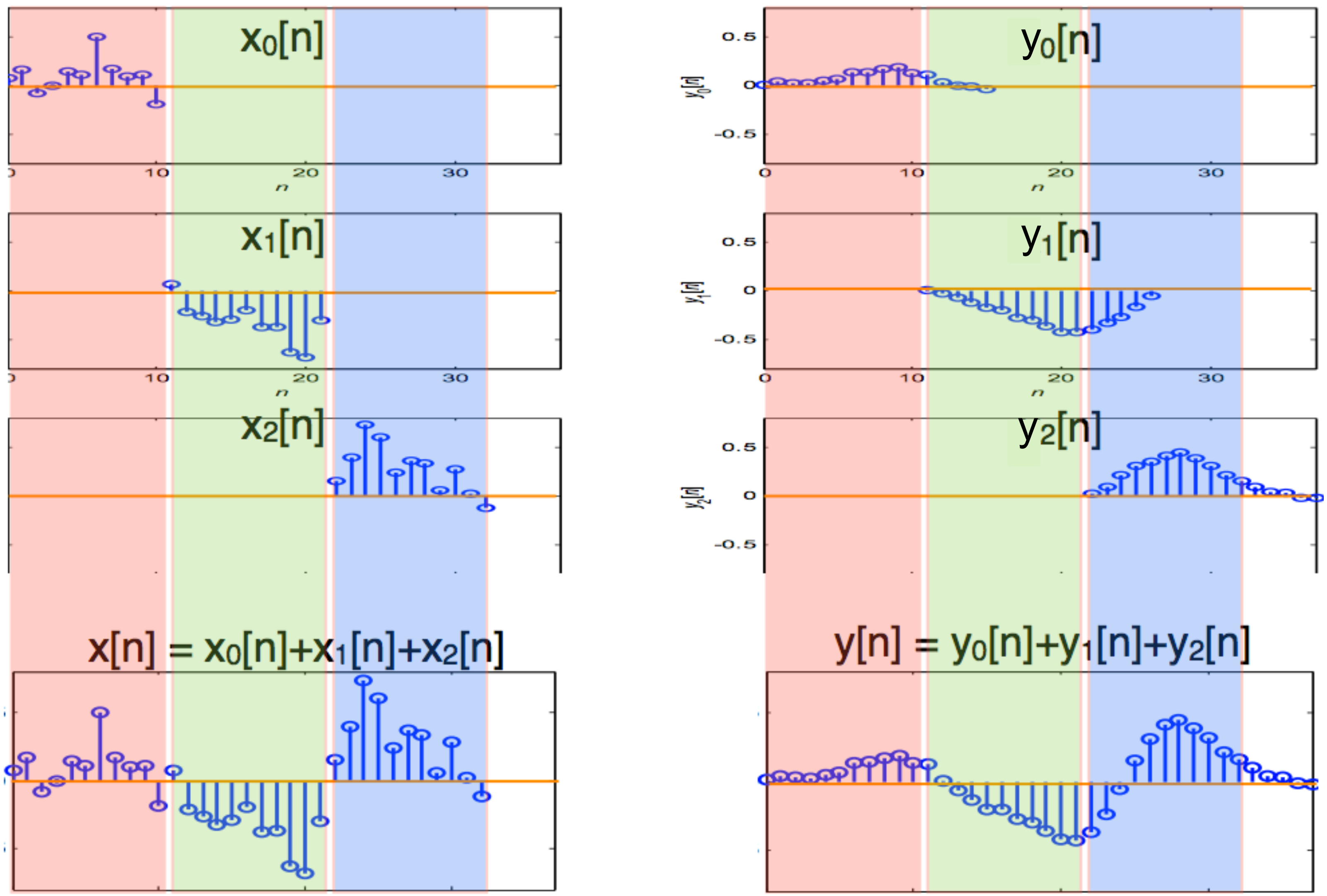
L=11



Example of Overlap-Add

L+P-1=16

L=11



Overlap-Save Method

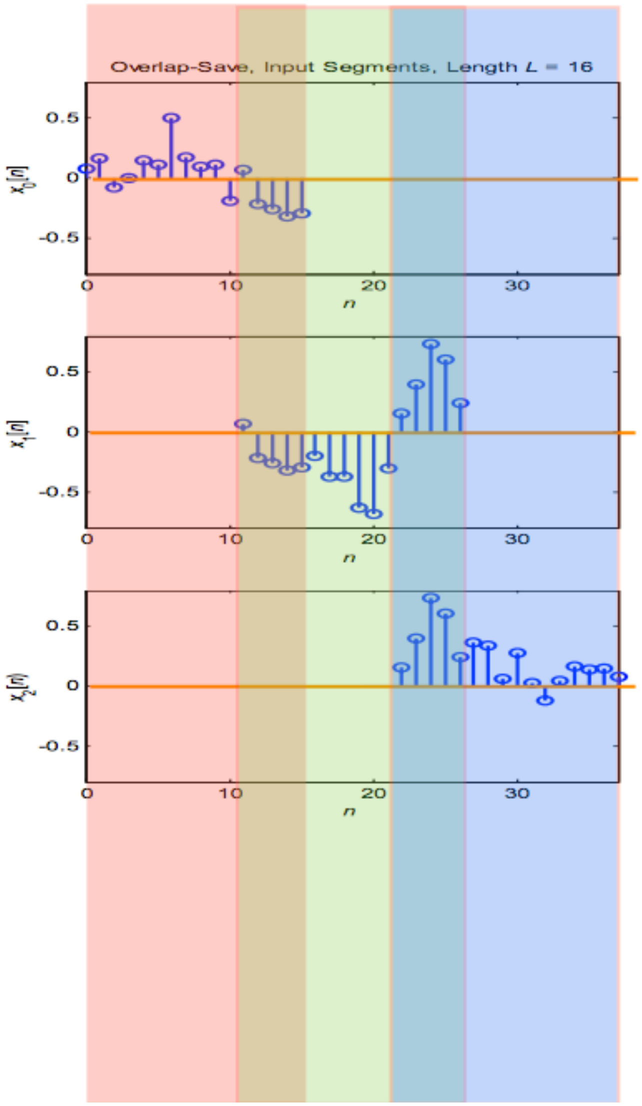
- Basic idea:
- Split input into overlapping segments with length L+P-1
  - P-1 sample overlap

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

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

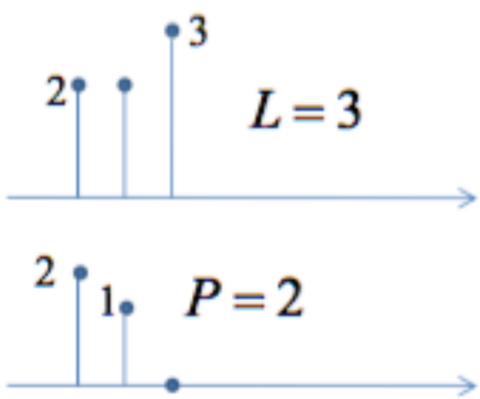
Example of Overlap-Save

16 sample input segments



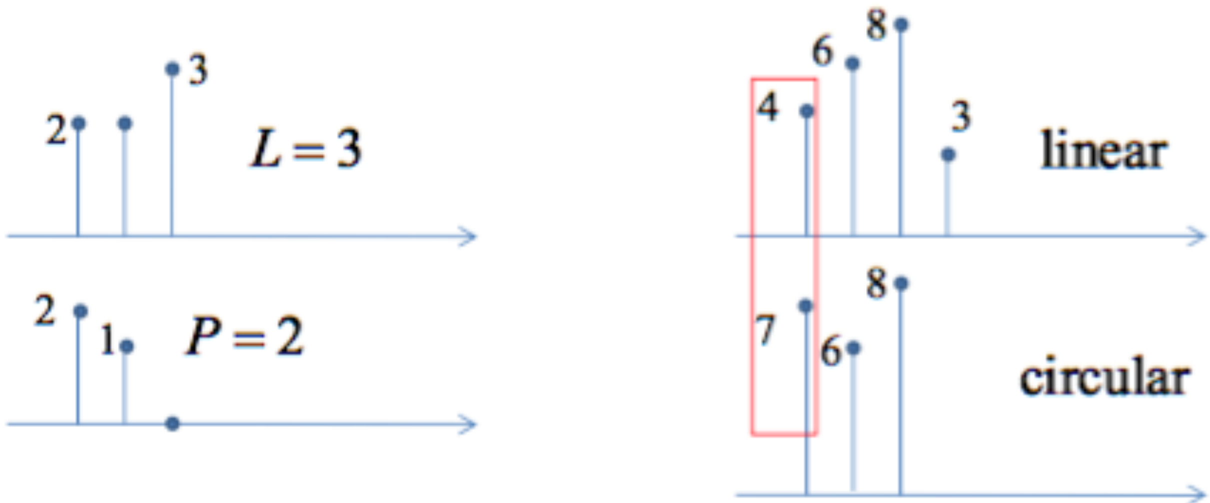
Circular to Linear Convolution

- 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



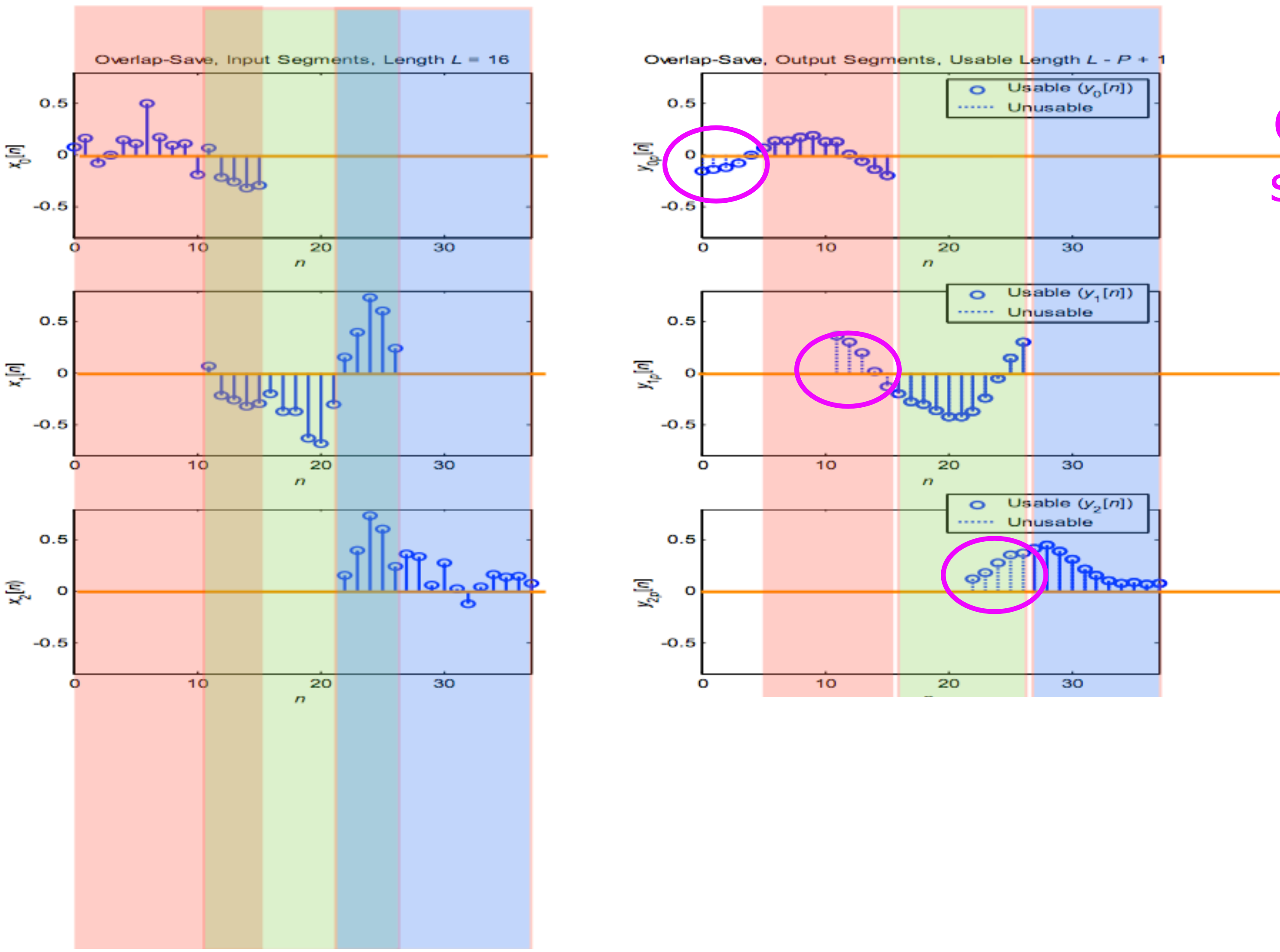
Circular to Linear Convolution

- 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



Example of Overlap-Save

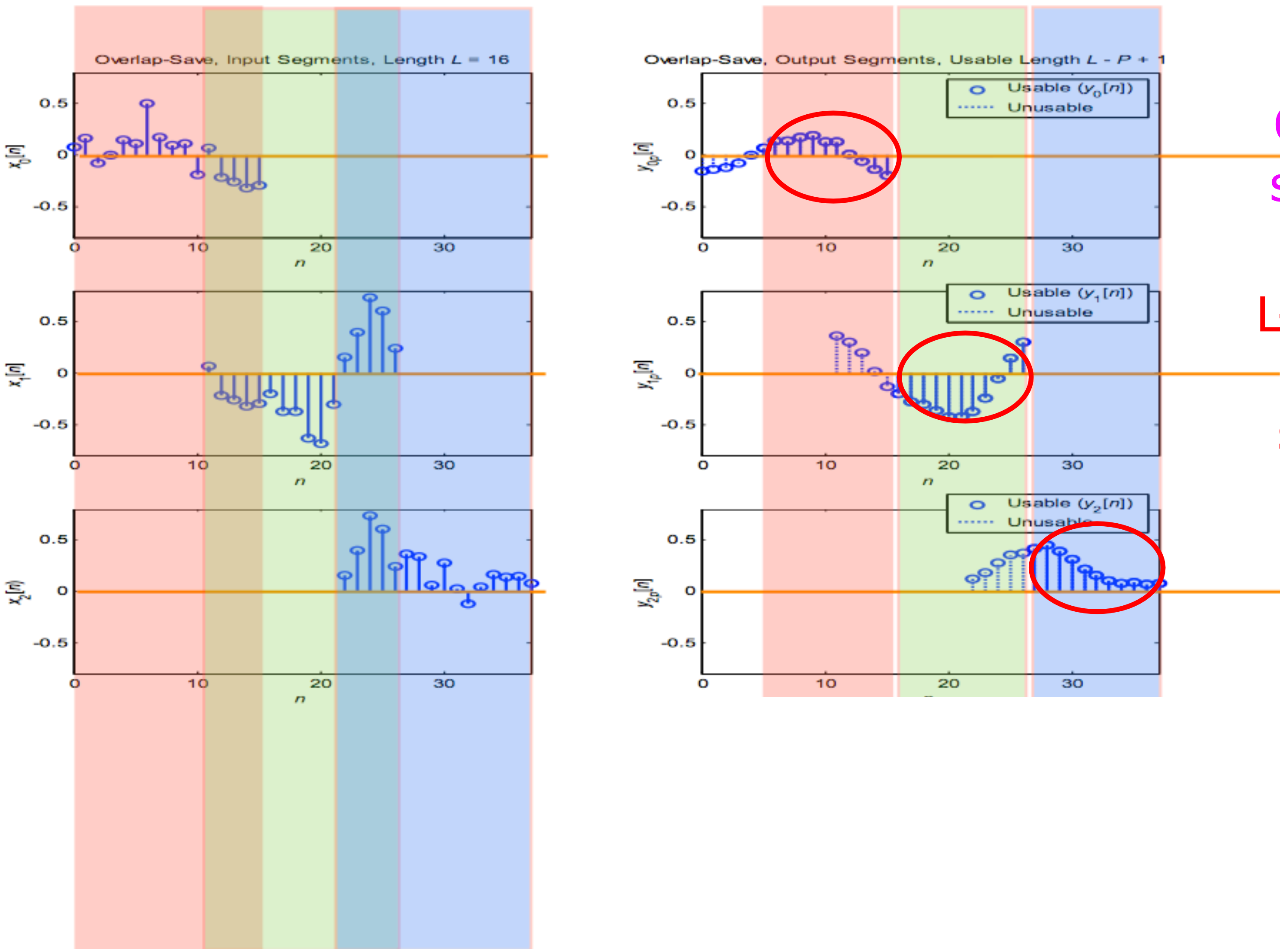
16 sample input segments



P-1=5 Overlap samples

Example of Overlap-Save

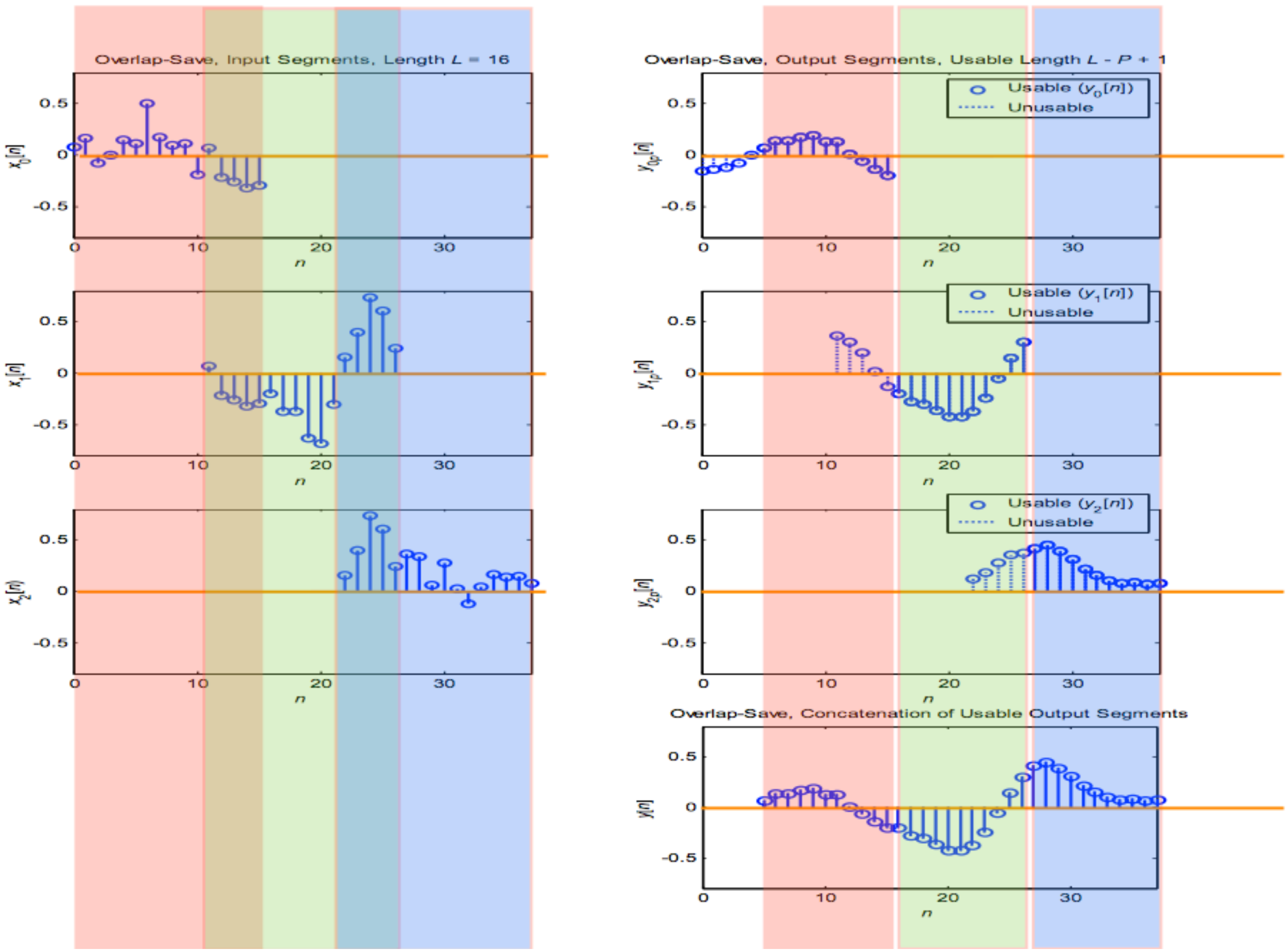
16 sample input segments



P-1=5 Overlap samples

L-P+1=11 keep samples

Example of Overlap-Save



Linear Convolution via Circular Convolution

- Zero-pad x[n] by P-1 zeros

$$x_{\text{zp}}[n] = \begin{cases} x[n] & 0 \leq n \leq L-1 \\ 0 & L \leq n \leq L+P-2 \end{cases}$$

- Zero-pad h[n] by L-1 zeros

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

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

Circular Conv. via Linear Conv. (w/ Aliasing)

- 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

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

Circular Conv. via Linear Conv. (w/ Aliasing)

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

- 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

Circular Conv. via Linear Conv. (w/ Aliasing)

- Given two N-point sequences ($x_1[n]$ and $x_2[n]$) and their N-point DFTs ($X_1[k]$ and $X_2[k]$)
- The N-point DFT of $x_3[n]=x_1[n]*x_2[n]$ is defined as

$$X_3[k] = X_3(e^{j(2\pi k/N)})$$

Circular Conv. via Linear Conv. (w/ Aliasing)

- Given two N-point sequences ($x_1[n]$ and $x_2[n]$) and their N-point DFTs ($X_1[k]$ and $X_2[k]$)
- 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 $X_3[k]=X_1[k]X_2[k]$, where the inverse DFT of $X_3[k]$ is

$$x_{3p}[n] = x_1[n] \, Ⓝ \, x_2[n]$$

Circular Conv. via Linear Conv. (w/ Aliasing)

- Given two N-point sequences ($x_1[n]$ and $x_2[n]$) and their N-point DFTs ($X_1[k]$ and $X_2[k]$)
- 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 $X_3[k]=X_1[k]X_2[k]$, where the inverse DFT of $X_3[k]$ is

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

Circular Conv. via Linear Conv. (w/ Aliasing)

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

Circular Conv. via Linear Conv. (w/ Aliasing)

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

- Thus

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

Circular Conv. via Linear Conv. (w/ Aliasing)

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

- Thus

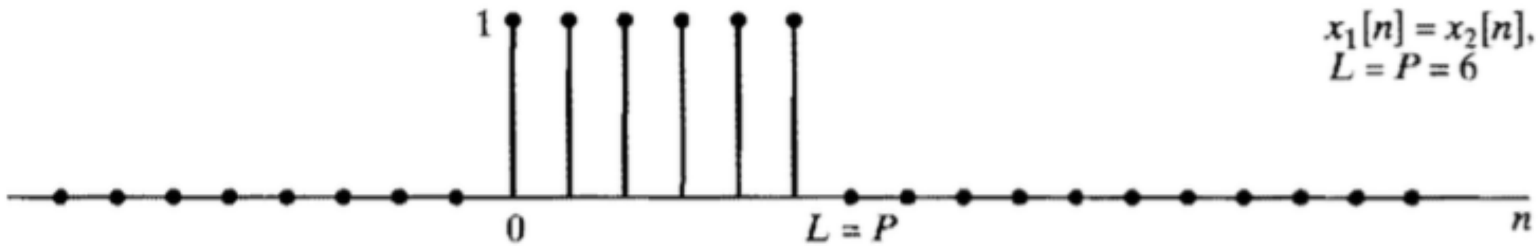
$$x_{3p}[n] = \begin{cases} \sum_{r=-\infty}^{\infty} x_1[n-rN] * x_2[n-rN] & 0 \le n \le N-1 \\ 0 & \text{else} \end{cases}$$

$$x_{3p}[n] = x_1[n] \, Ⓝ \, x_2[n]$$

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

Example 1:

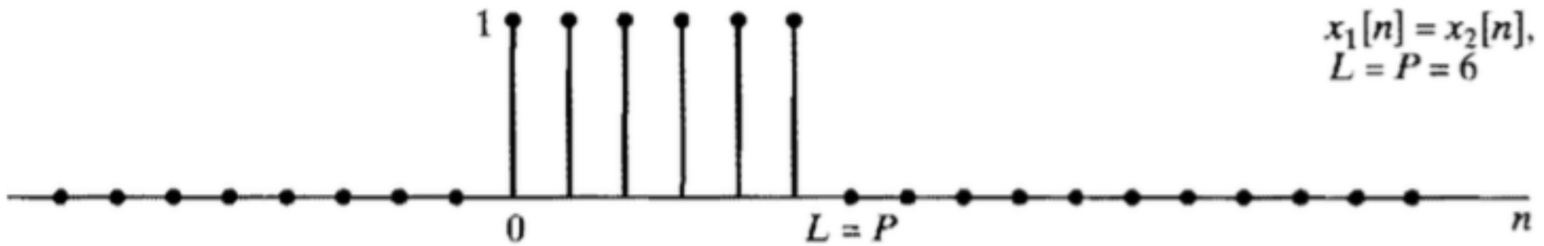
- Let



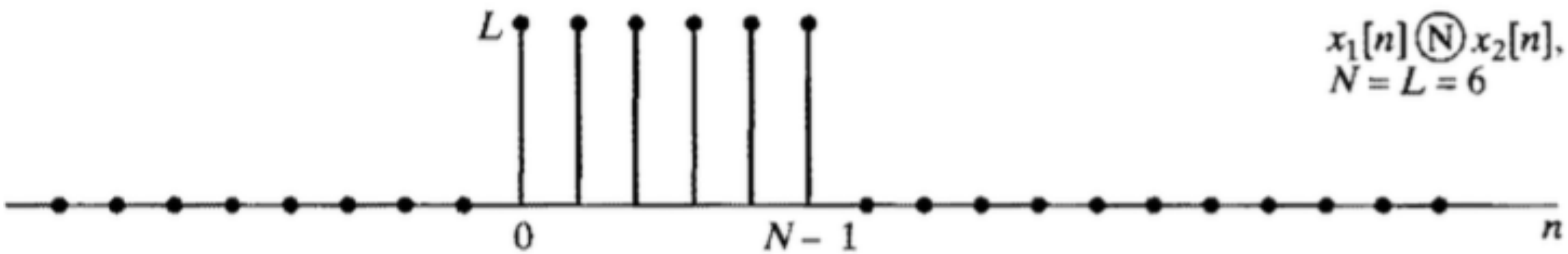
- The N=L=6-point circular convolution results in

Example 1:

- Let

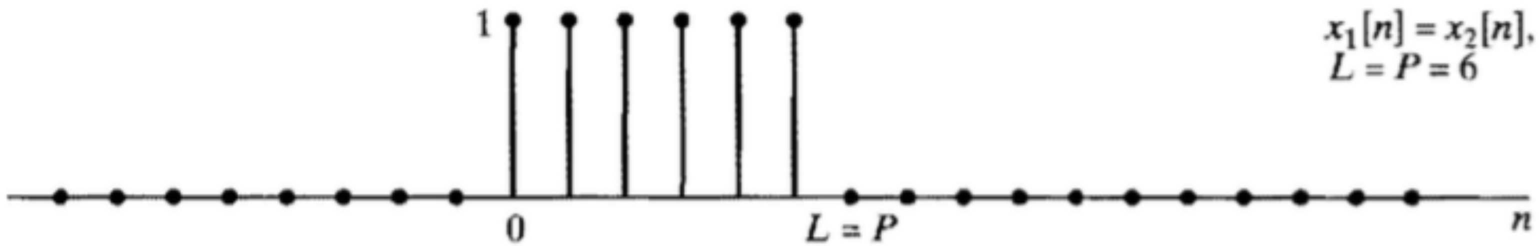


- The N=L=6-point circular convolution results in



Example 1:

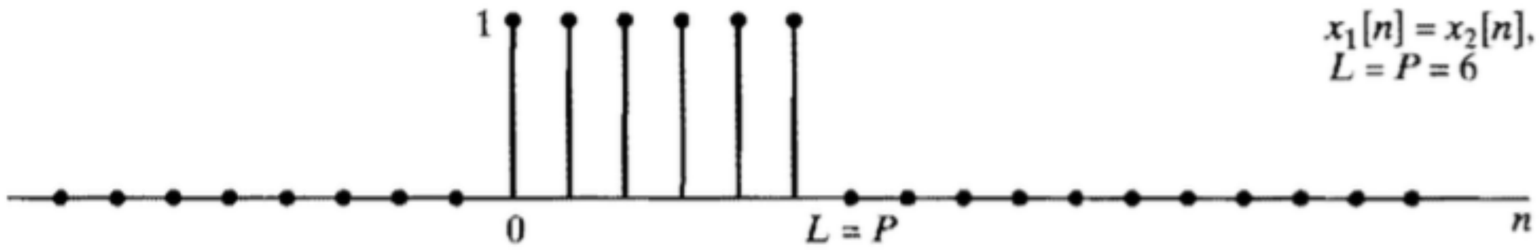
- Let



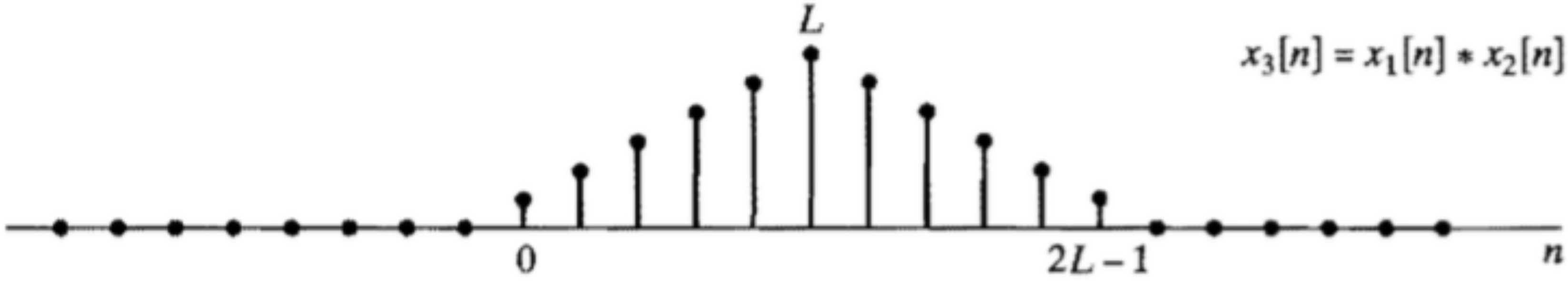
- The linear convolution results in

Example 1:

- Let

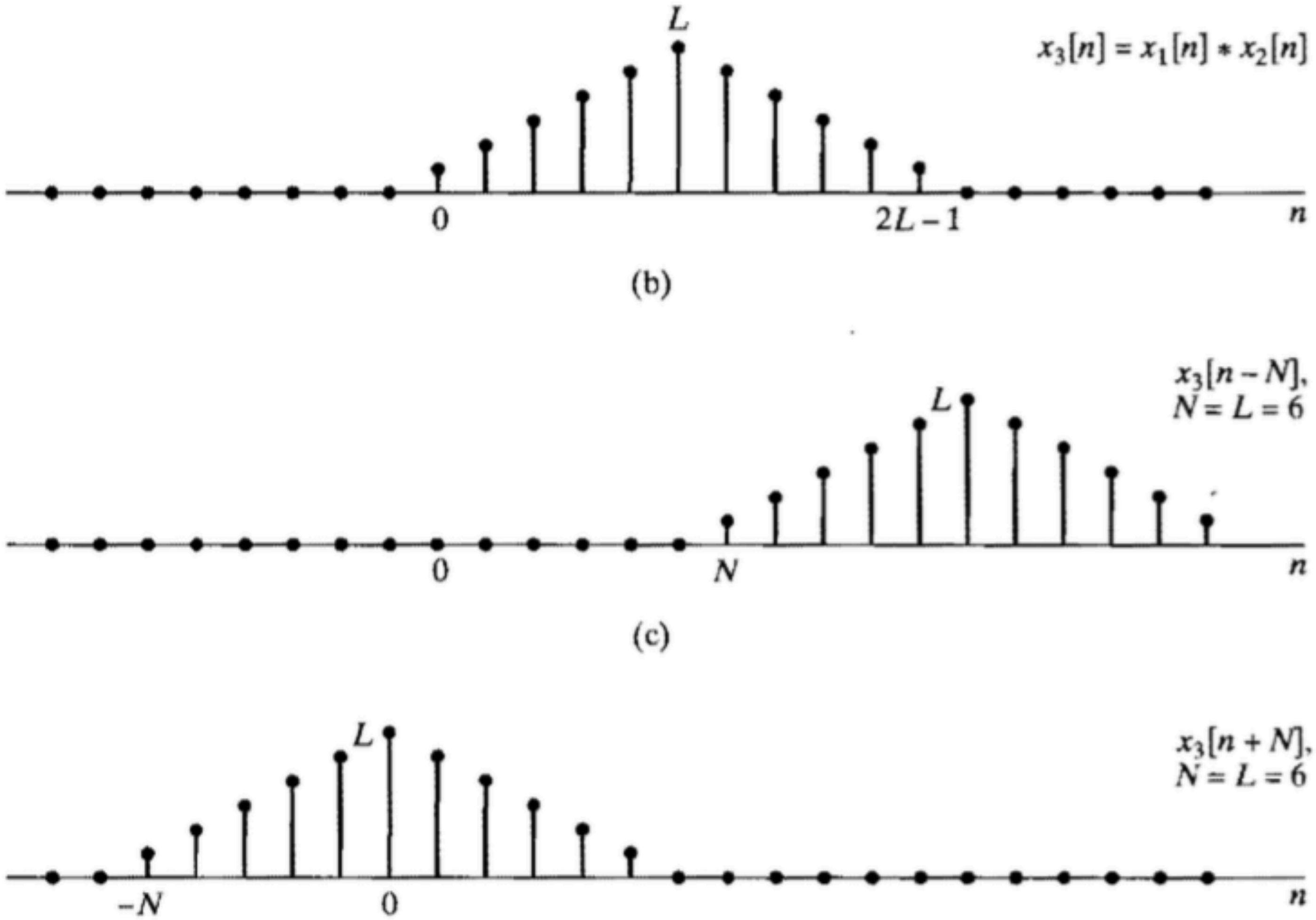


- The linear convolution results in



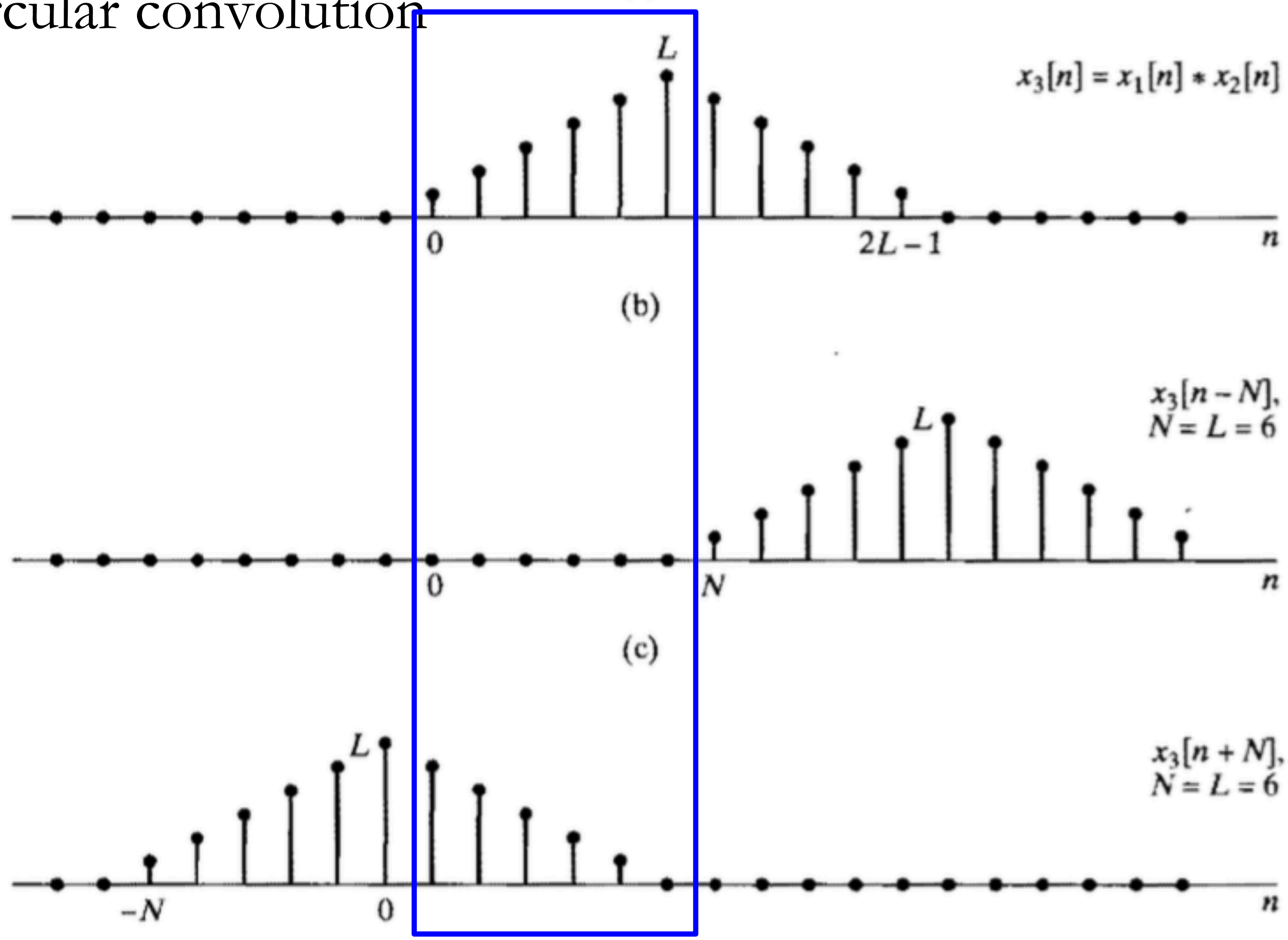
Example 1:

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



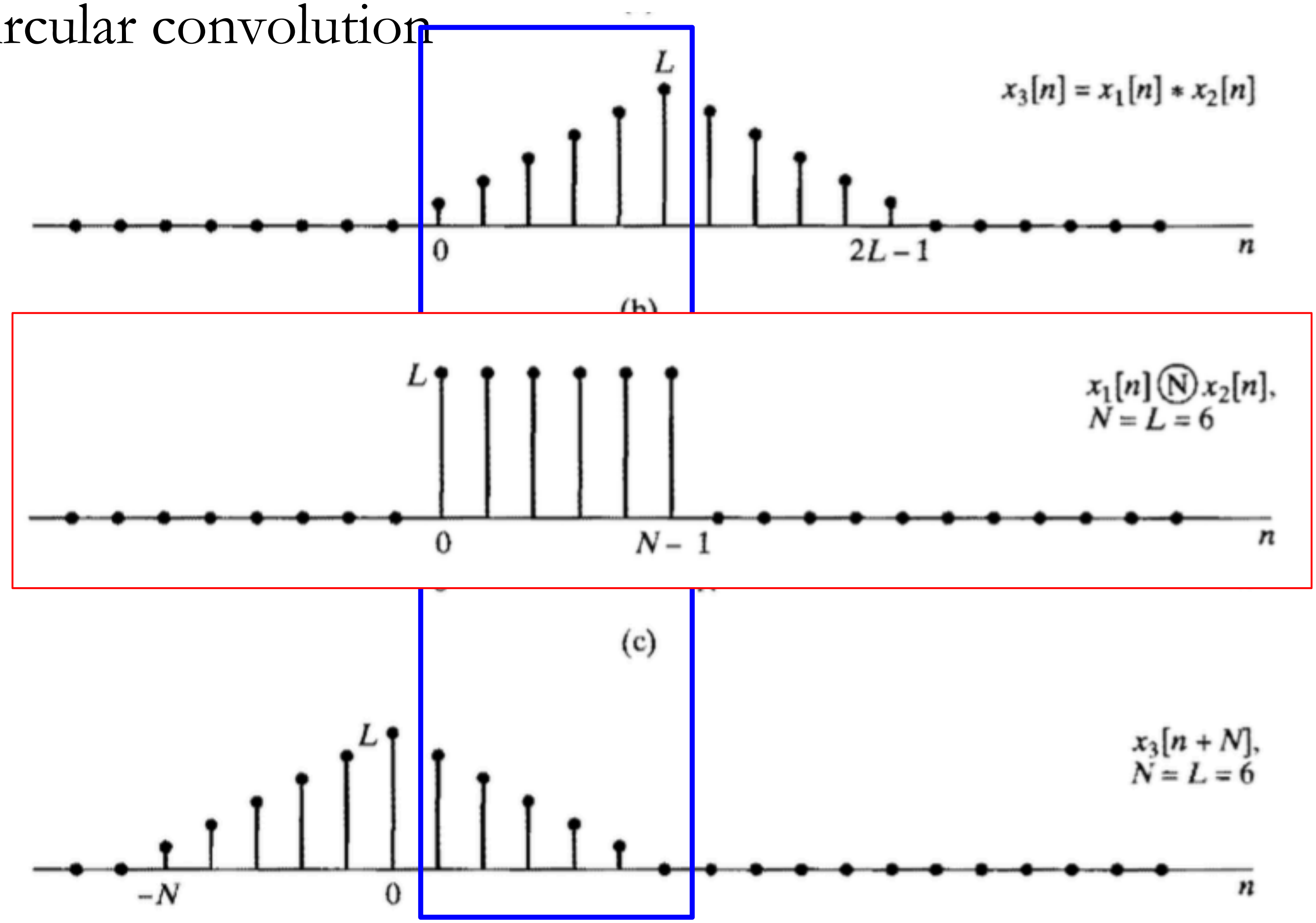
Example 1:

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



Example 1:

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

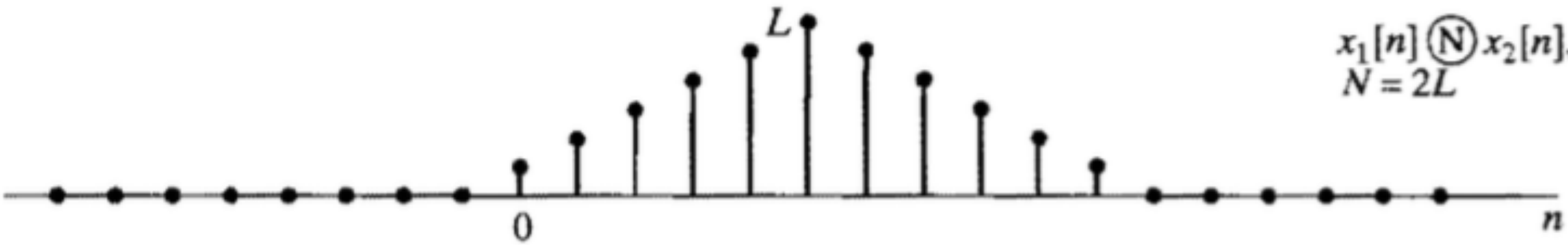


Example 1:

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

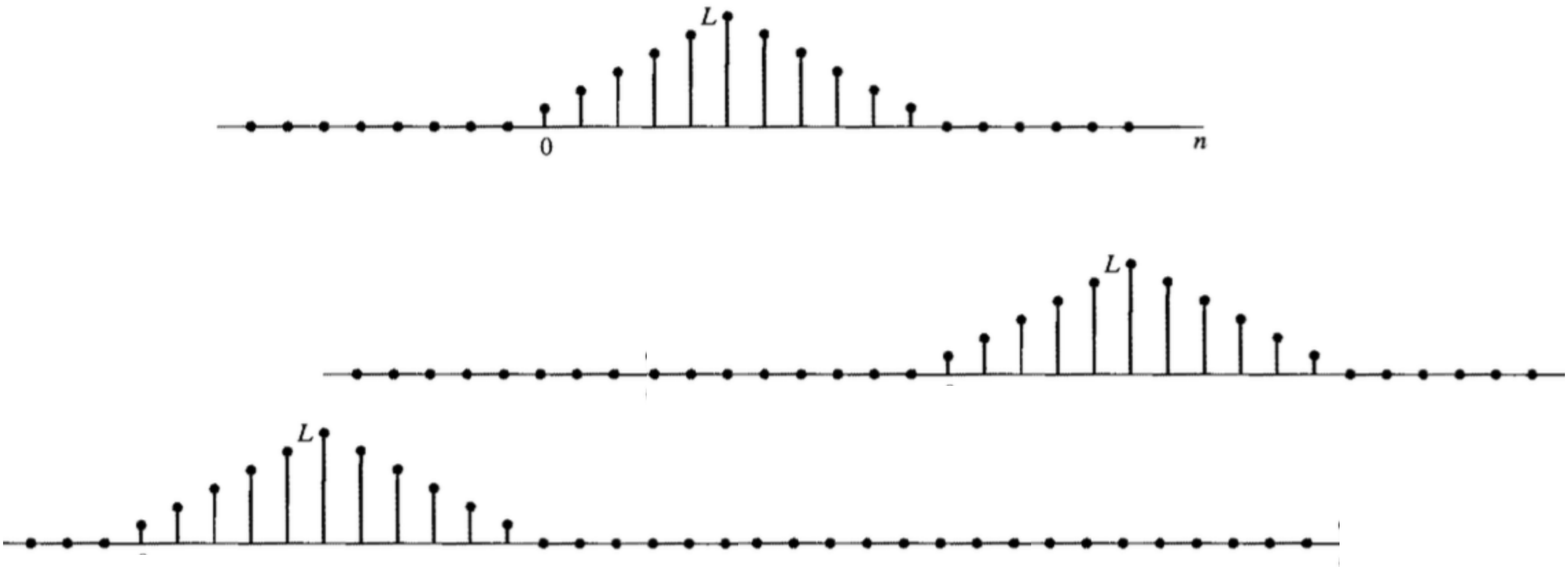
Example 1:

- If I want the circular convolution and linear convolution to be the same, what do I do?
  - Take the N=2L-point circular convolution



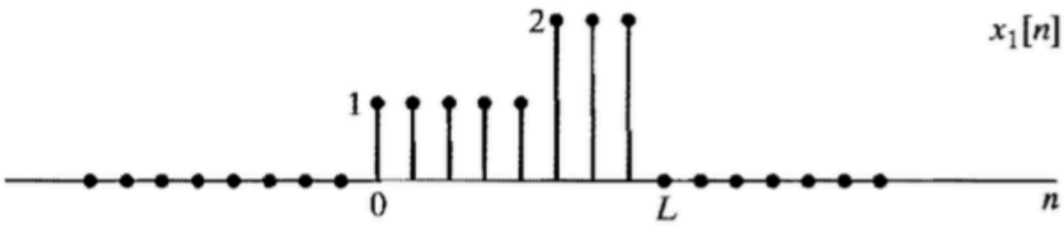
Example 1:

- If I want the circular convolution and linear convolution to be the same, what do I do?
  - Take the N=2L-point circular convolution

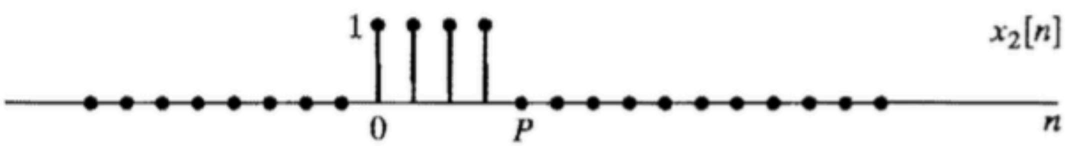


Example 2:

- Let



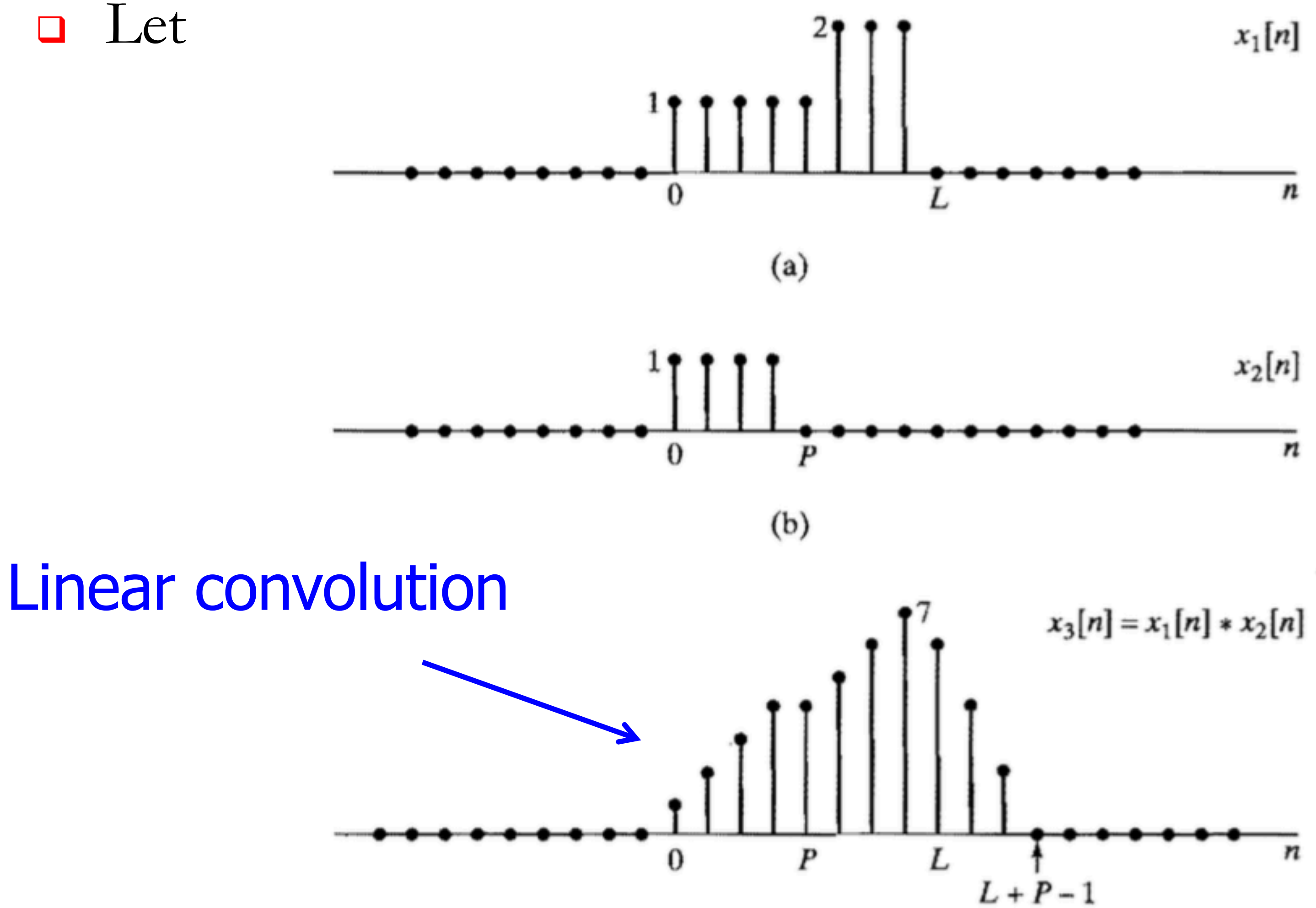
(a)



(b)

Example 2:

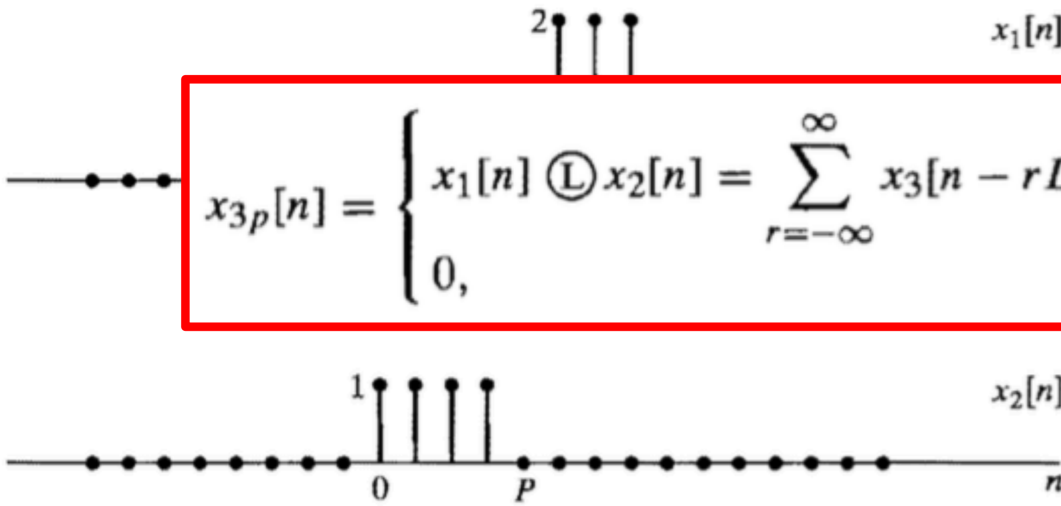
- Let



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

Example 2:

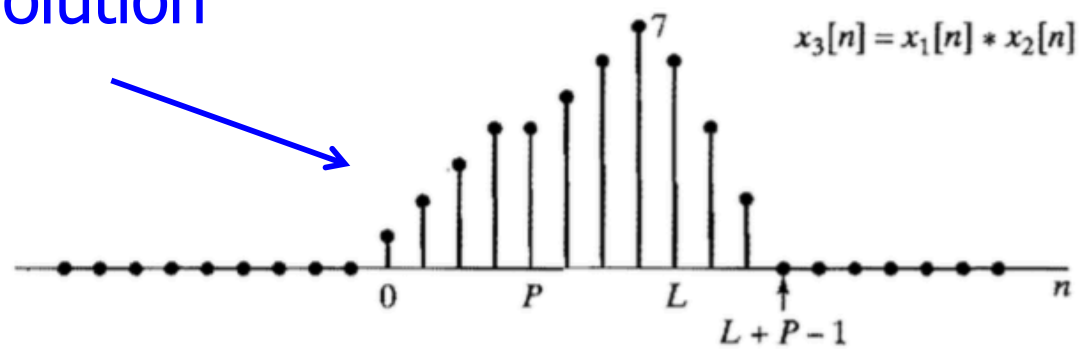
- Let



$$x_{3p}[n] = \begin{cases} x_1[n] \text{ Ⓛ } x_2[n] = \displaystyle\sum_{r=-\infty}^{\infty} x_3[n - rL], & 0 \le n \le L-1, \\ 0, & \text{otherwise.} \end{cases}$$

(b)

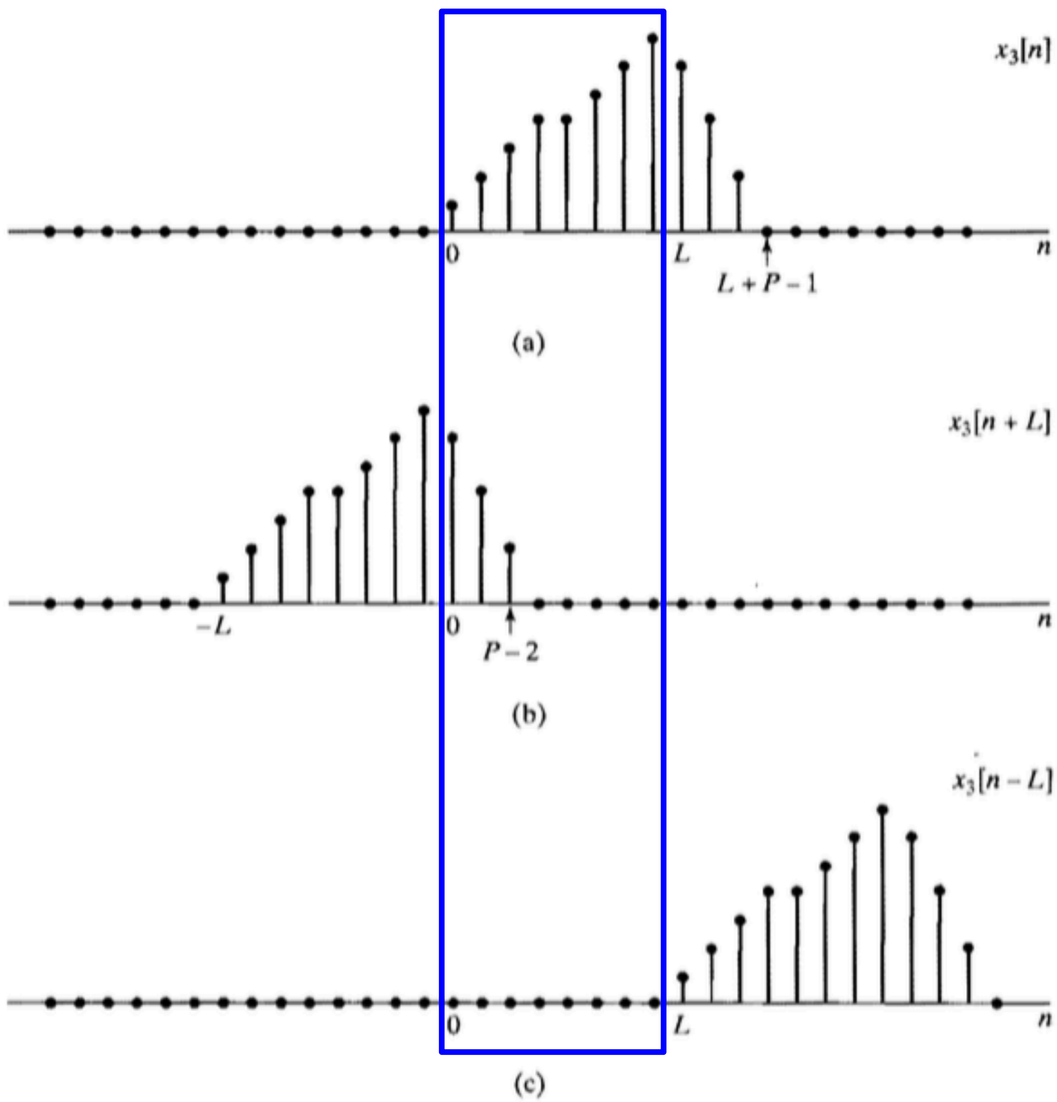
Linear convolution



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

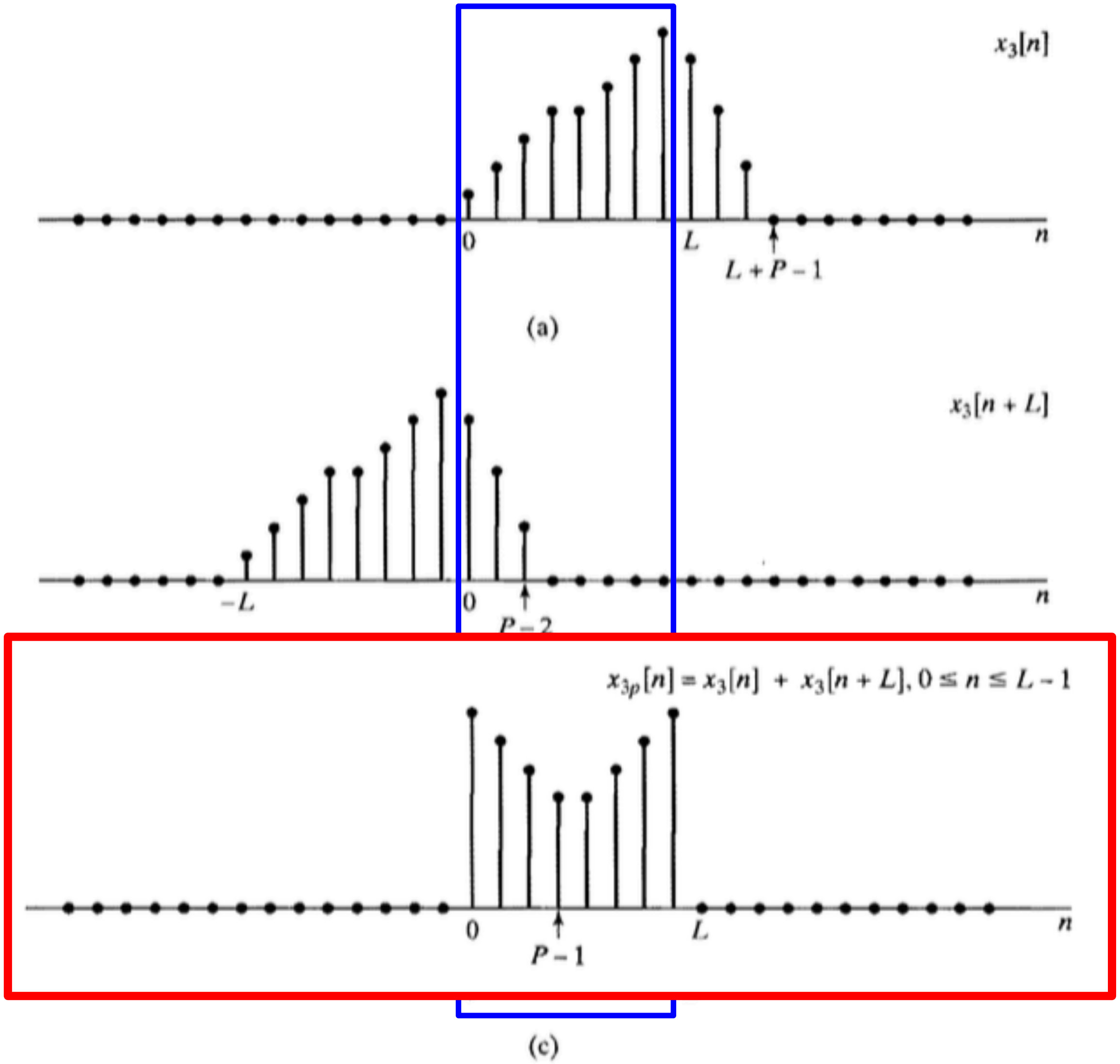
Example 2:

- The L-shifted linear convolutions



Example 2:

- The L-shifted linear convolutions



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
  - 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

Admin

- HW 8 out now
  - Due 4/16