Review of Discrete Fourier Transform

- $x[n]$ $-\infty < n < +\infty$
- Fourier Transform $X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n] e^{-j\omega n}$
- Z-transform $X(z) = \sum_{n=-\infty}^{\infty} x[n] z^{-n}$

- If $x[n]$ $0 \leqslant n \leqslant N-1$ (finite-duration sequence)
- → Discrete Fourier Transform (DFT)

4 Forms of Fourier Transform

Symbol:

$x_c(t)$ -------- aperiodic continuous signals

$\tilde{x}_c(t)$ -------- periodic continuous signals

Ω -------- analog frequency

ω -------- digital frequency "Sampled" frequency

$(\omega = \Omega T)$

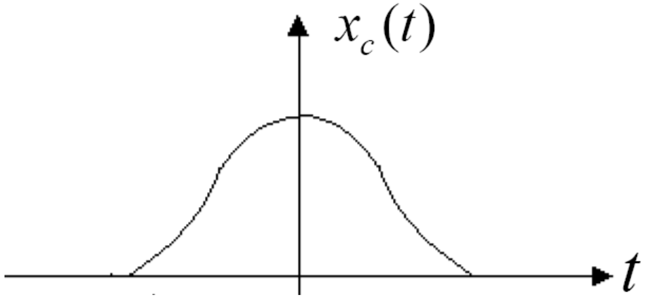
t_p -------- period of periodic signals such as $x_c(t)$

$X(j\Omega)$ --------Fourier transform of CT signals

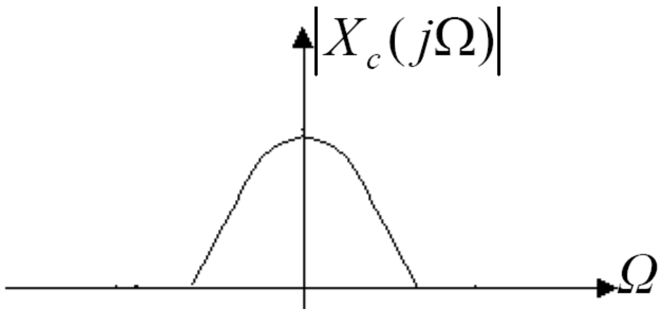
$X(e^{j\omega})$ --------Fourier transform of sequences

Continuous-Time and Continuous-Frequency

$$\begin{cases} X_c(j\Omega) = \int_{-\infty}^{\infty} x_c(t) e^{-j\Omega t} dt \\ x_c(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} X_c(j\Omega) e^{j\Omega t} d\Omega \end{cases}$$



Continuous
Aperiodic



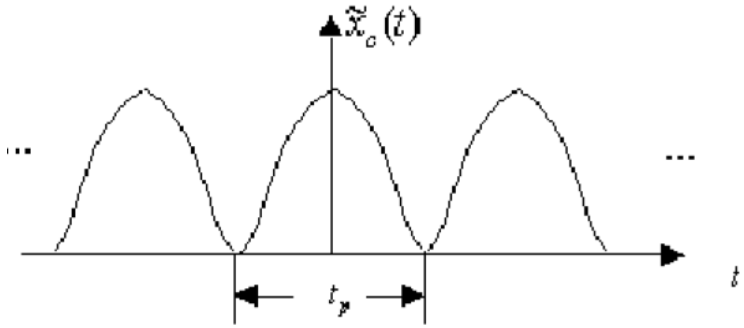
Continuous
Aperiodic

Continuous-Time and Discrete-Frequency

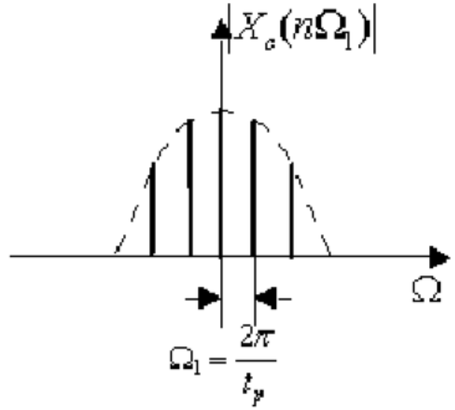
$$\begin{cases} X_c(n\Omega_1) = \dfrac{1}{t_p}\displaystyle\int_{-\frac{t_p}{2}}^{\frac{t_p}{2}} \tilde{x}_c(t)e^{-jn\Omega_1 t}dt \\ \tilde{x}_c(t) = \displaystyle\sum_{n=-\infty}^{\infty} X_c(n\Omega_1)e^{jn\Omega_1 t} \end{cases}$$

where: $\Omega_1 = \dfrac{2\pi}{t_p}$

Fourier series of periodic continuous signals



Periodic
Continuous

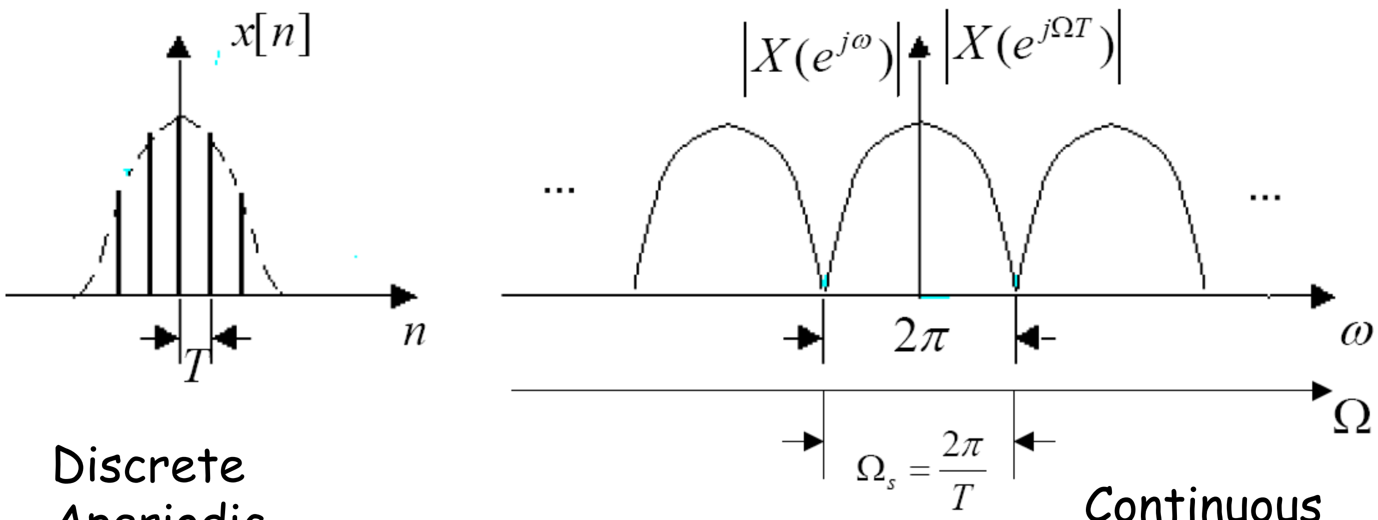


Discrete
Aperiodic

Discrete-Time and Continuous-Frequency

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

Fourier transform of aperiodic discrete signals



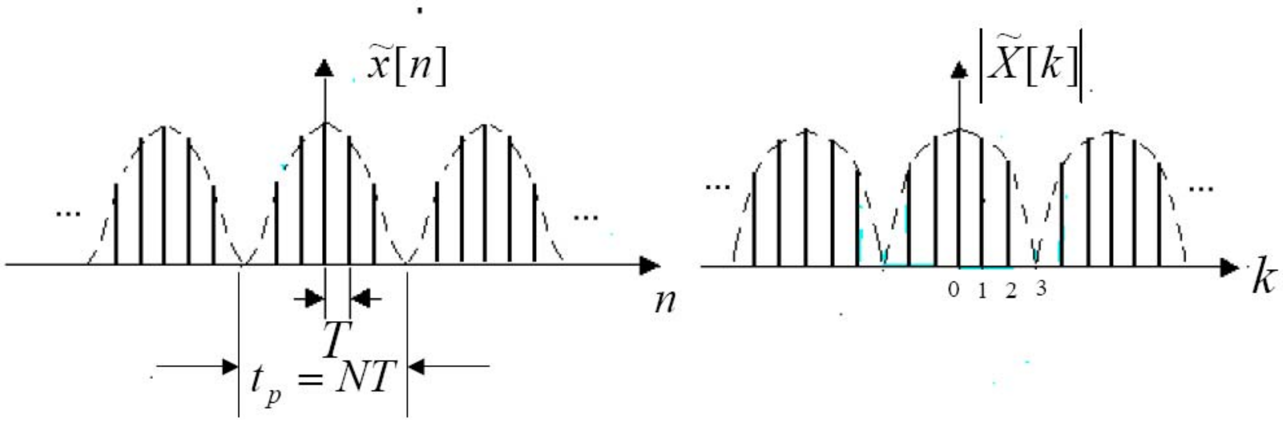
Discrete
Aperiodic

Continuous
Periodic

Discrete Fourier Transform

time-domain: periodic, discrete

frequency-domain: discrete, periodic



- DFT is identical to samples of Fourier transforms
- In DSP applications, we are able to store only a finite number of samples
- we are able to compute the spectrum only at specific discrete values of ω

Discrete Fourier Transform

- Discrete Fourier transform (DFT) pairs

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

N complex multiplications
N-1 complex additions

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

$$\text{where } W_N^{-kn} = e^{-j\frac{2\pi}{N}kn}$$

- DFT/IDFT can be implemented by using the same hardware
- It requires N^2 complex multiplications and N(N-1) complex additions

More About DFT

- Properties of Discrete Fourier Transform
- Linear Convolution and Discrete Fourier Transform
- Discrete Cosine Transform

Periodic Sequence

- Consider a periodic sequence $\tilde{x}[n]$ of period N
- The sequence can be represented by Fourier series

$$\tilde{x}[n] = \frac{1}{N}\sum_{k}\tilde{X}[k]e^{j(2\pi/N)kn}$$

- The Fourier series for any discrete-time signal with period N requires only N harmonically related complex exponentials.

$$\because \; e_{k+lN}[n] = e^{j(2\pi/N)(k+lN)n} = e^{j(2\pi/N)kn} = e_k[n]$$

$$\therefore \quad \tilde{x}[n] = \frac{1}{N}\sum_{k=0}^{N-1}\underline{\tilde{X}[k]}e^{j(2\pi/N)kn}$$

To obtain $\tilde{X}[k]$ Apply the Orthogonality property, we have

$$\sum_{n=0}^{N-1} \tilde{x}[n] e^{-j(2\pi/N)rn} = \sum_{n=0}^{N-1} \frac{1}{N} \sum_{k=0}^{N-1} \tilde{X}[k] e^{j(2\pi/N)(k-r)n}$$

Interchange the order of summation

$$\sum_{n=0}^{N-1} \tilde{x}[n] e^{-j(2\pi/N)rn} = \sum_{k=0}^{N-1} \tilde{X}[k] \left[\frac{1}{N} \sum_{n=0}^{N-1} e^{j(2\pi/N)(k-r)n} \right]$$

Because:

$$\frac{1}{N} \sum_{n=0}^{N-1} e^{j(2\pi/N)(k-r)n} = \begin{cases} 1, & k-r = mN, m \in Z \\ 0, & \text{otherwise} \end{cases}$$

$$\sum_{n=0}^{N-1} \tilde{x}[n] e^{-j(\frac{2\pi}{N})rn} = \tilde{X}[r]$$

The coefficients are also periodic with period N

DFS Representation of a Periodic Sequence

Define: $W_N = e^{-j(2\pi/N)}$

$$\tilde{x}[n] \xleftrightarrow{\mathcal{DFS}} \tilde{X}[k]$$

Synthesis equation

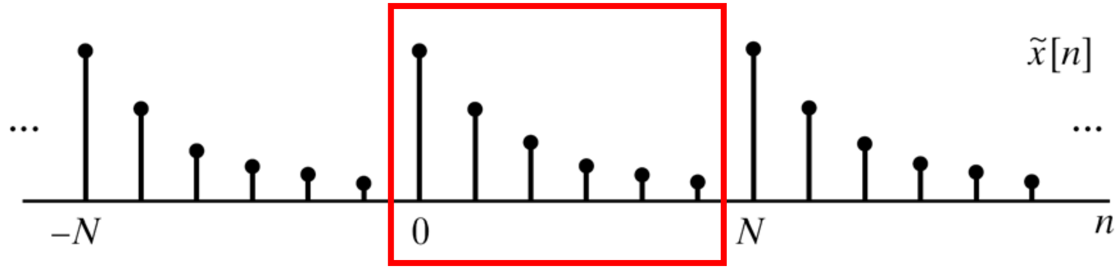
$$\tilde{x}[n] = \frac{1}{N}\sum_{k=0}^{N-1} \tilde{X}[k] W_N^{-kn}$$

Analysis equation

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

$\tilde{X}[k]$ and $\tilde{x}[n]$ are periodic sequence of period N

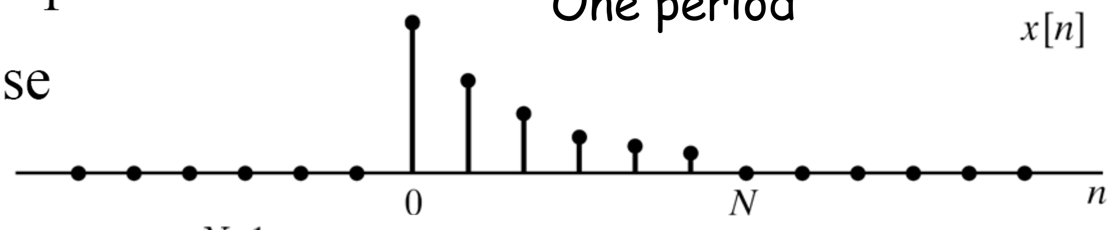
Physical Significance



Let

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

One period



Then

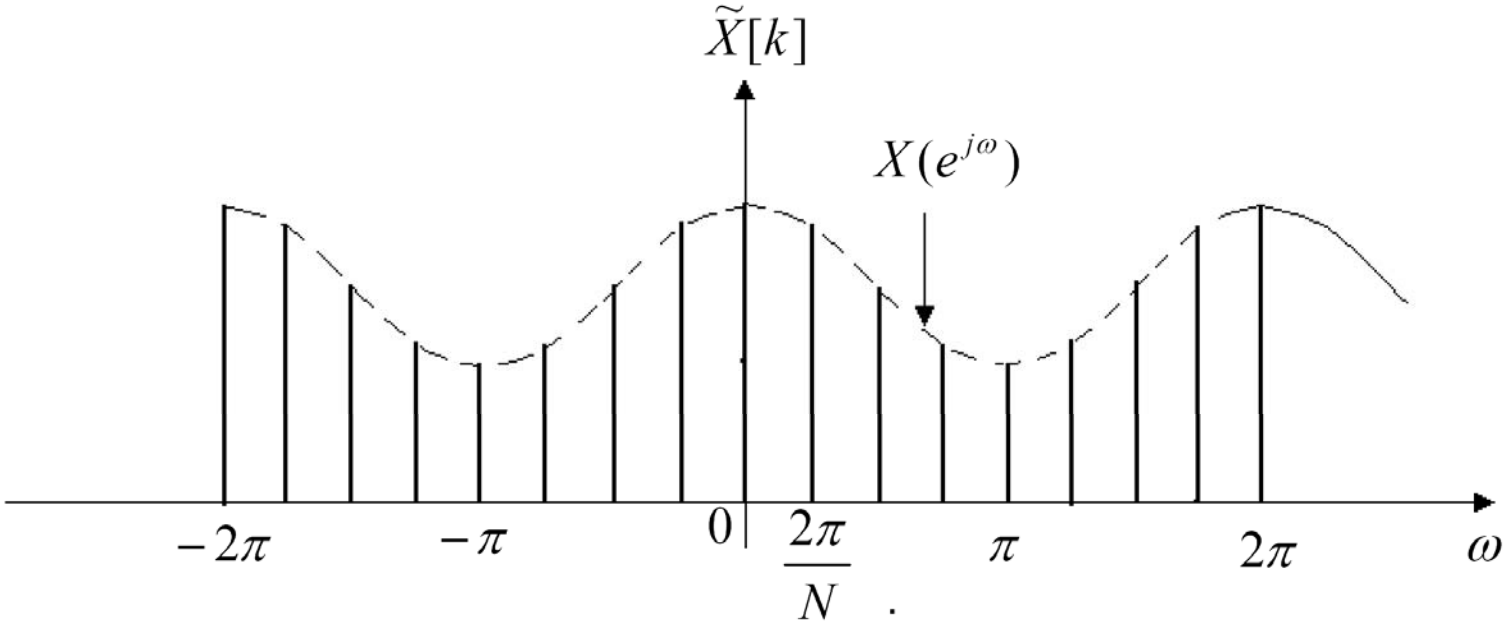
$$X(e^{j\omega}) = \sum_{n=0}^{N-1} x[n] e^{-j\omega n} = \sum_{n=0}^{N-1} \tilde{x}[n] e^{-j\omega n}$$

We have:

$$\tilde{X}[k] = \sum_{n=0}^{N-1} \tilde{x}[n] W_N^{kn} \quad \Rightarrow \quad \tilde{X}[k] = X(e^{j\omega})\big|_{\omega = 2\pi k / N}$$

$\tilde{X}[k]$ vs $X(e^{j\omega})$

Example



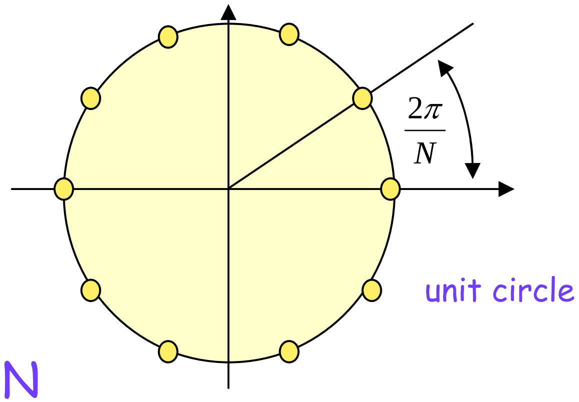
Sampling the Fourier Transform

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

Then

$$\tilde{X}[k] = X(e^{j\omega})\big|_{\omega=(2\pi/N)k} = X(e^{(2\pi/N)k})$$

or $$\tilde{X}[k] = X(z)\big|_{z=e^{j(2\pi/N)k}} = X(e^{(2\pi/N)k})$$

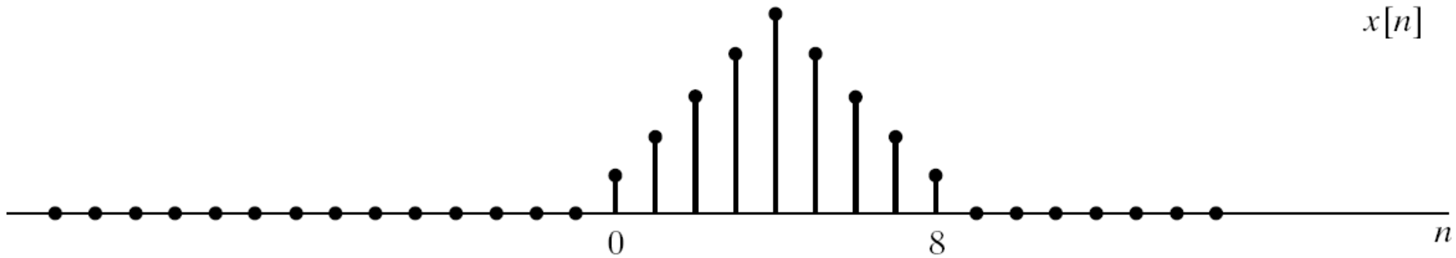


The sampling sequence is periodic with period N

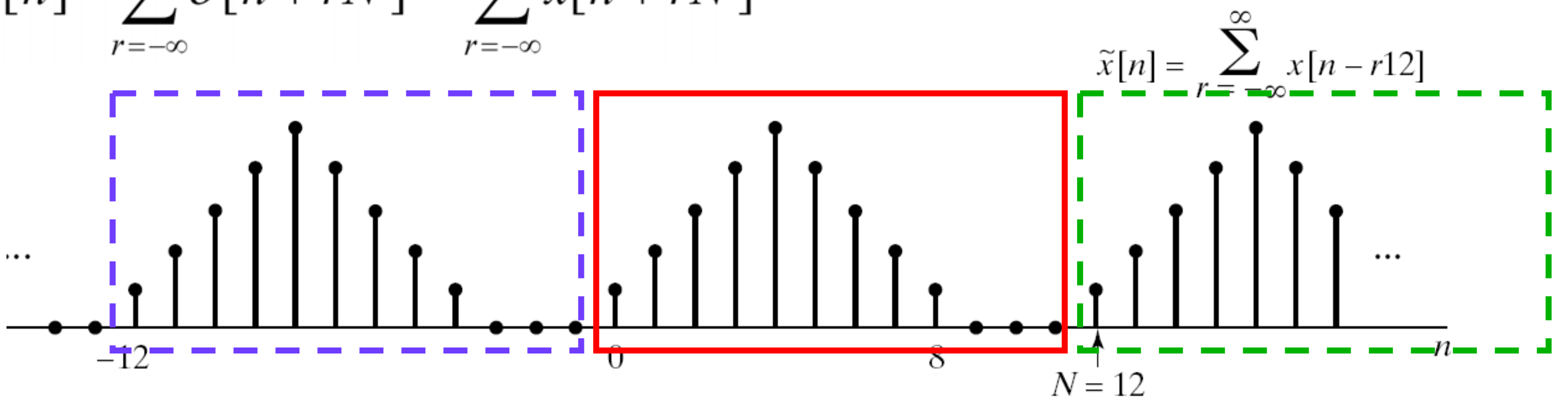
Since $$\tilde{x}[n] = \frac{1}{N}\sum_{k=0}^{N-1} \tilde{X}[k] W_N^{-kn} = \frac{1}{N}\sum_{k=0}^{N-1}\left[\sum_{m=-\infty}^{\infty} x[m] \underbrace{e^{-j(2\pi/N)km}}_{W_N^{km}}\right] W_N^{-kn}$$

$$= \sum_{m=-\infty}^{\infty} x[m]\left[\frac{1}{N}\sum_{k=0}^{N-1} W_N^{-k(n-m)}\right] = x[n] * \sum_{r=-\infty}^{\infty} \delta[n+rN]$$

$\tilde{x}[n]$ vs $x[n]$



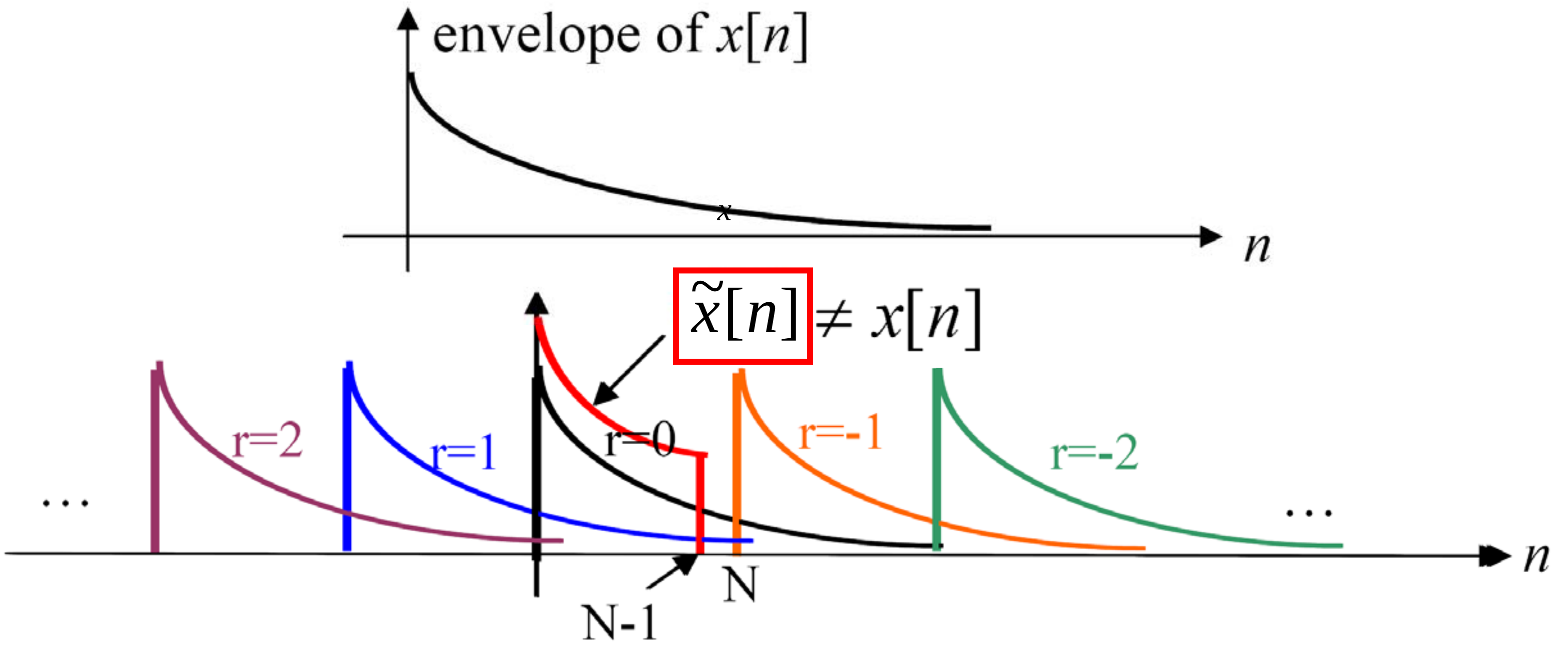
$$\tilde{x}[n] = x[n] * \sum_{r=-\infty}^{\infty} \delta[n + rN] = \sum_{r=-\infty}^{\infty} x[n + rN]$$



By adding together an infinite number of shifted replicas of x[n]

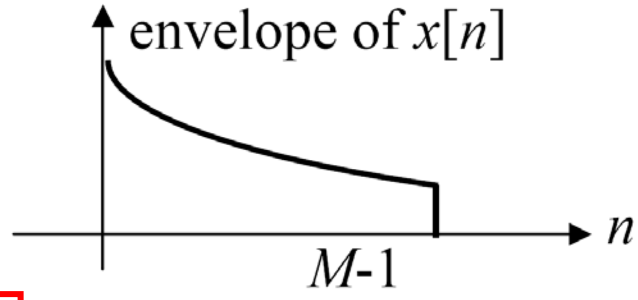
Aliasing Problem 1

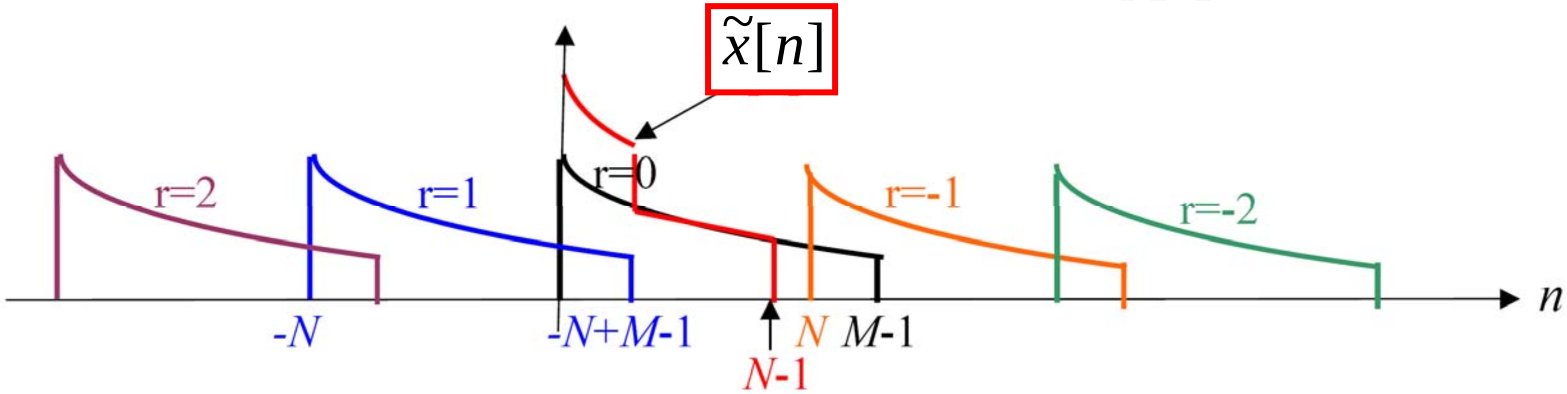
- x[n] is infinite-length sequence



Aliasing Problem 2

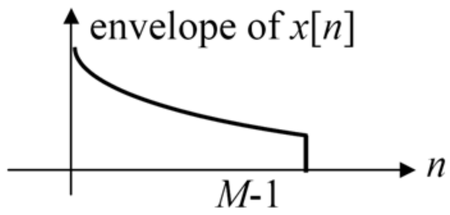
- If x[n] is finite-length sequence, 0≤n≤M-1
- Consider the case N<M



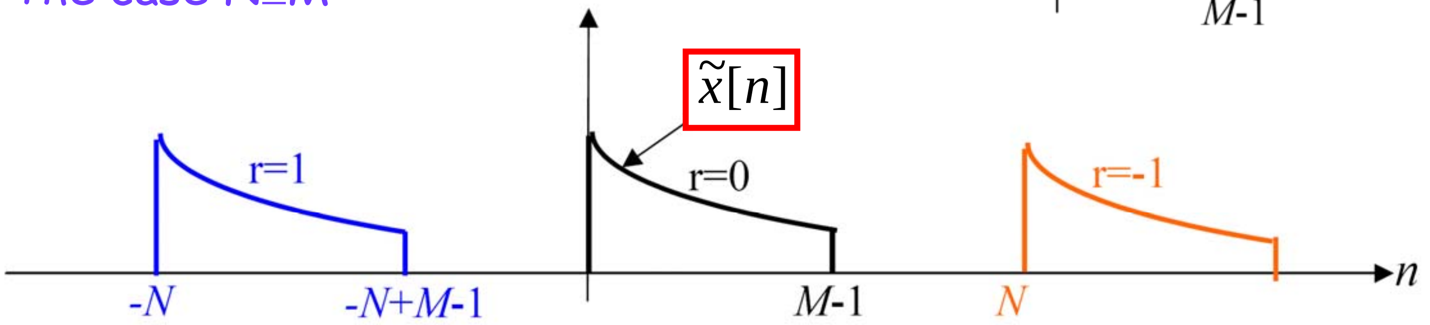


$$\tilde{x}[n] \neq x[n]$$

Concluding Remarks



The case N≥M



Conclusion: If the length of sequence $x[n]$ is M, then the sampling points N of its Fourier transform must be larger than or equal to M, otherwise ,we cannot recover $x[n]$ from $\tilde{x}[n]$, i.e.

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

or

$$\tilde{x}[n] = x[((n))_N]$$

(n modulo N)

Property of DFT

- Linearity

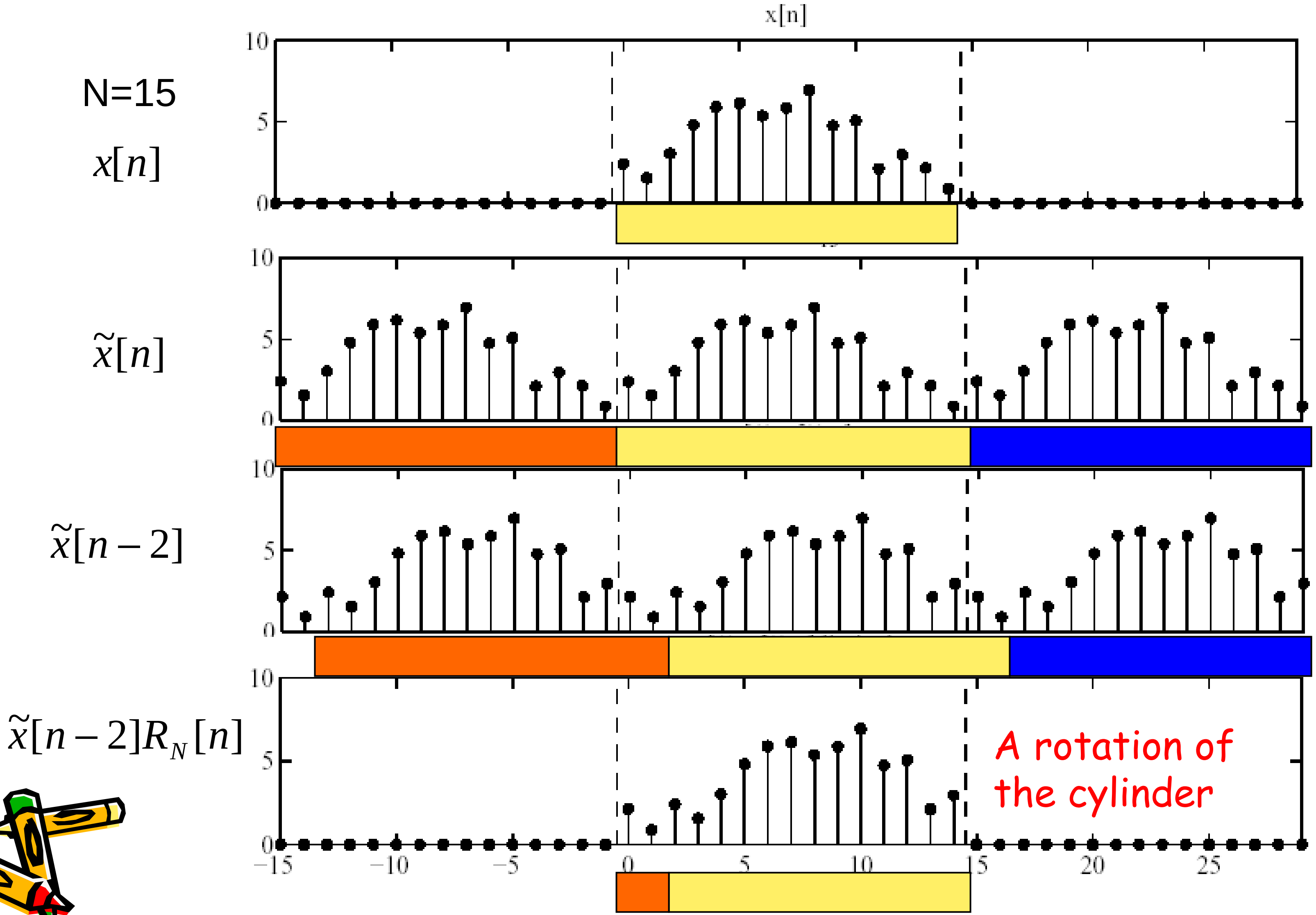
if $x_1[n] \overset{DFT}{\longleftrightarrow} X_1[k]$ of length N_1

$x_2[n] \overset{DFT}{\longleftrightarrow} X_2[k]$ of length N_2

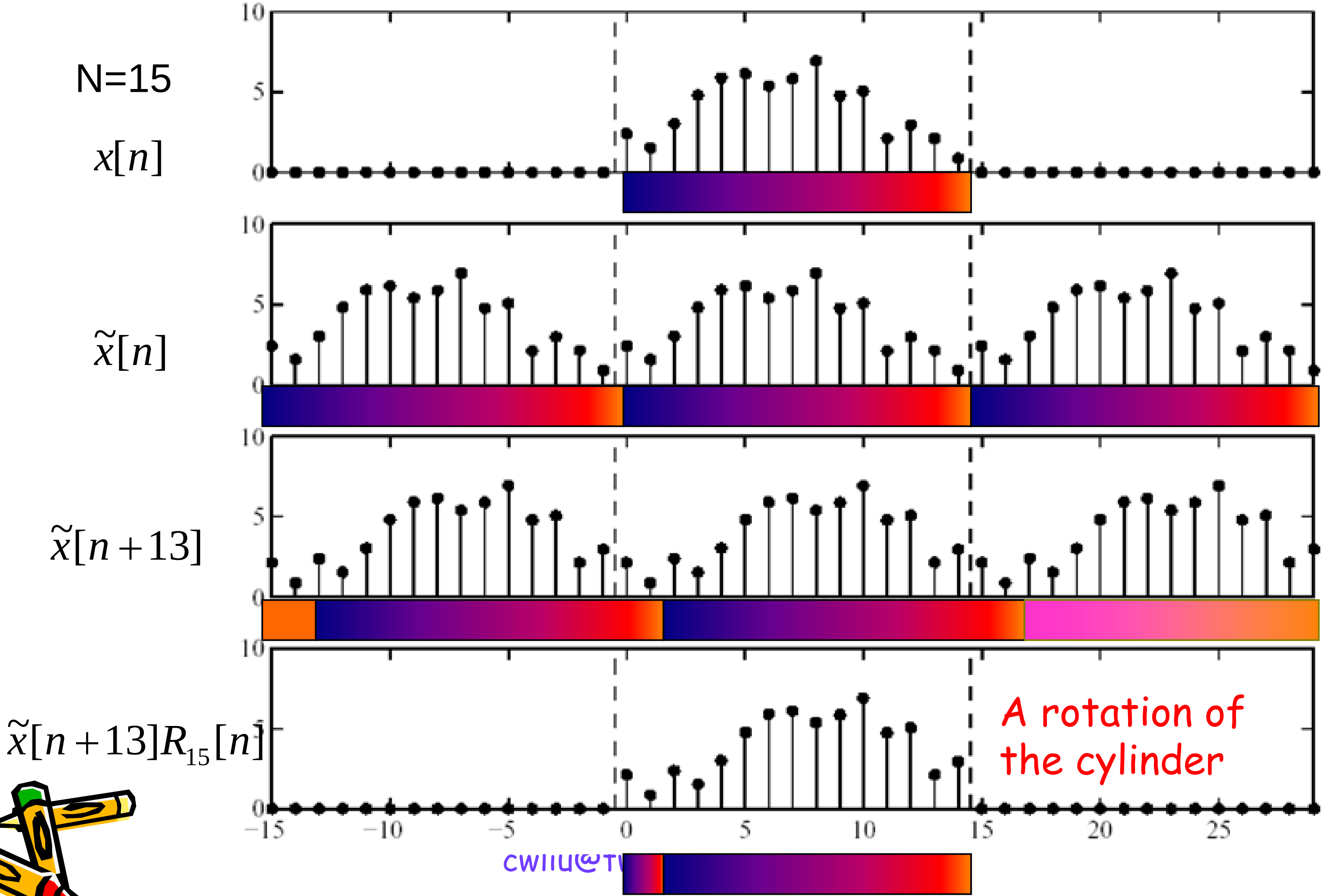
then $ax_1[n]+bx_2[n] \overset{DFT}{\longleftrightarrow} aX_1[k]+bX_2[k]$

of length $N_3=\max[N_1, N_2]$

Circular Shift of a Sequence



Circular Shift of a Sequence



Property of DFT

- Circular Shift

if $x[n] \overset{\mathcal{DFT}}{\longleftrightarrow} X[k]$ of length N

then $x[((n-m))_N] \overset{\mathcal{DFT}}{\longleftrightarrow} e^{-j(2\pi k/N)m} X[k]$

$0 \le m \le N-1$

A rotation of the sequence in the interval

that is $x[((n-m))_N] \overset{\mathcal{DFT}}{\longleftrightarrow} W_N^{mk} X[k]$

$0 \le n \le N-1$

On the other hand

$W_N^{-ln} x[n] \overset{\mathcal{DFT}}{\longleftrightarrow} X[((k-l))_N] \quad \mathbf{0 \le l \le N\text{-}1}$

Other Properties of DFT

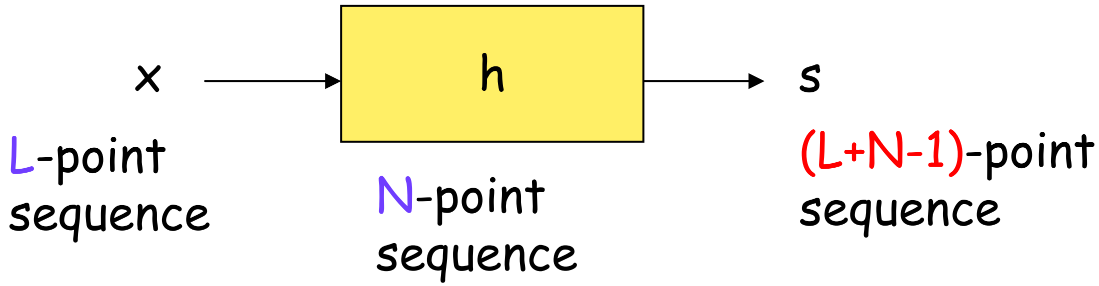
- Duality
  - 8.6.3

- Symmetry
  - 8.6.4

More About DFT

- Properties of Discrete Fourier Transform
- Linear Convolution and Discrete Fourier Transform
- Discrete Cosine Transform

Review of Convolution



- Given two sequences:
  - Data sequence x_i, $0 \le i \le N-1$, of length N
  - Filter sequence h_i, $0 \le i \le L-1$, of length L
- Linear convolution

NL multiplications

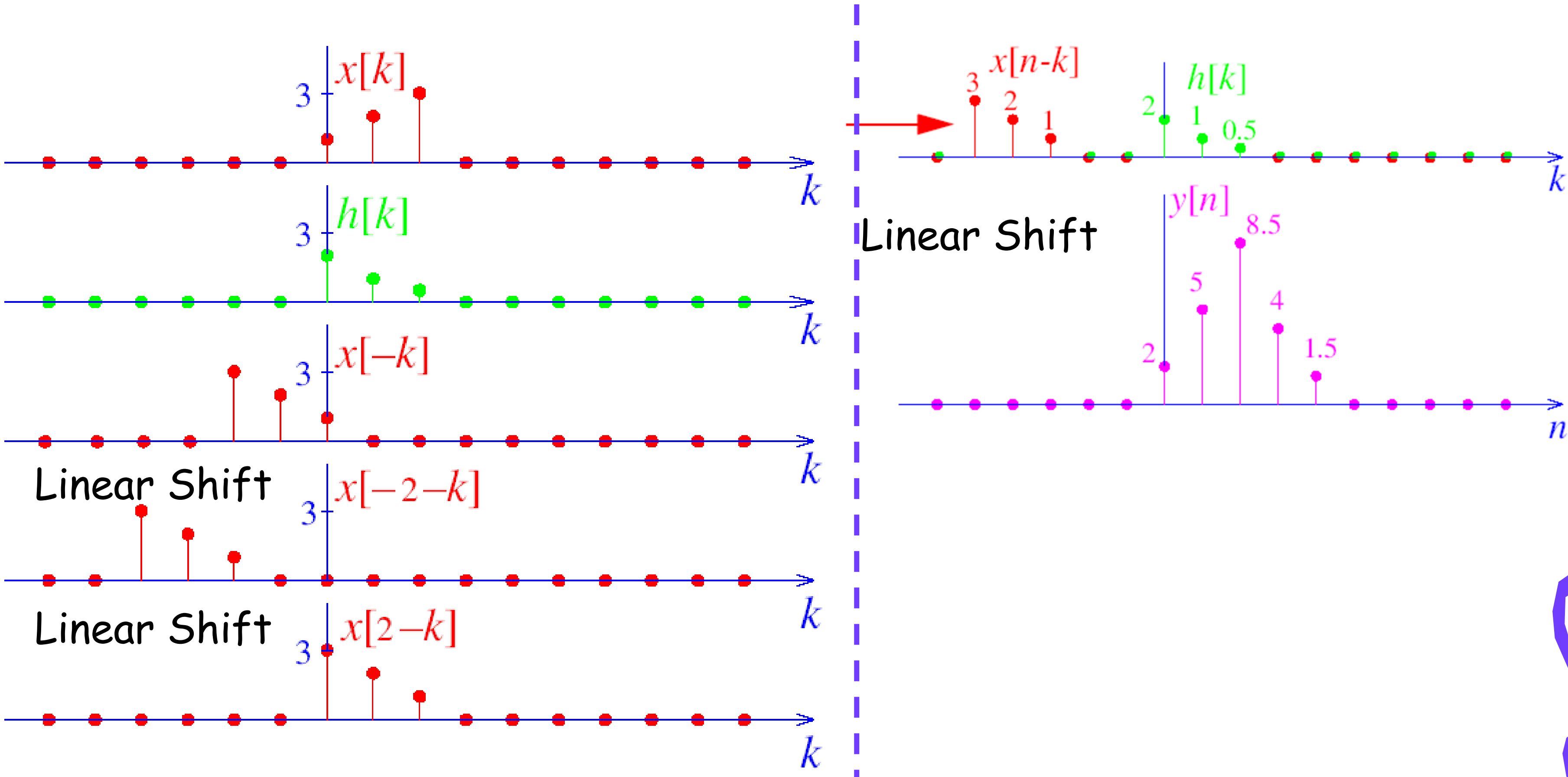
$$y_i = x_i * h_i = h_i * x_i, \quad i = 0,1,\ldots,L+N-2$$

- Direct computation, for example 2-by-2 convolution

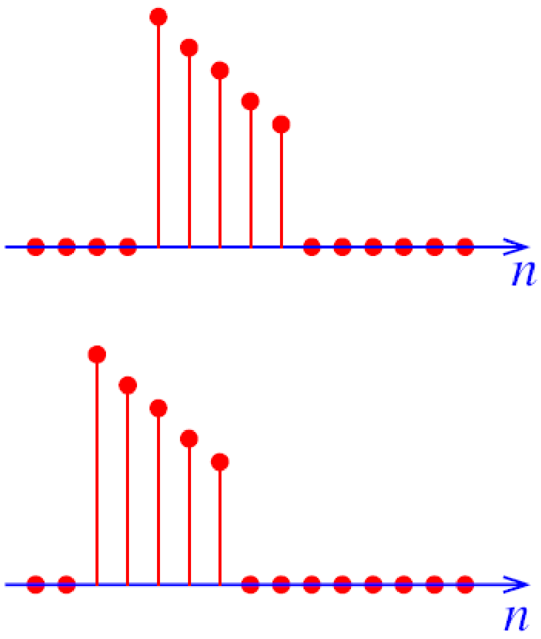
$$\begin{bmatrix} s_0 \\ s_1 \\ s_2 \end{bmatrix} = \begin{bmatrix} h_0 & 0 \\ h_1 & h_0 \\ 0 & h_1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \end{bmatrix}$$

require 4 multiplications and 1 addition

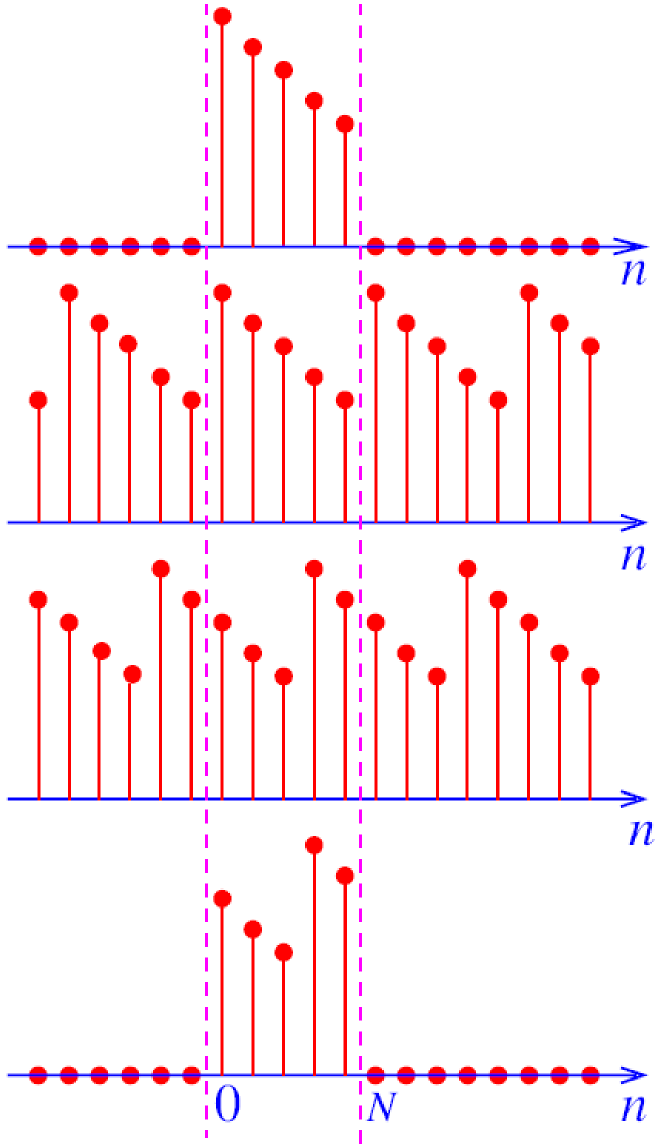
Linear Convolution



Linear Shift vs Circular Shift

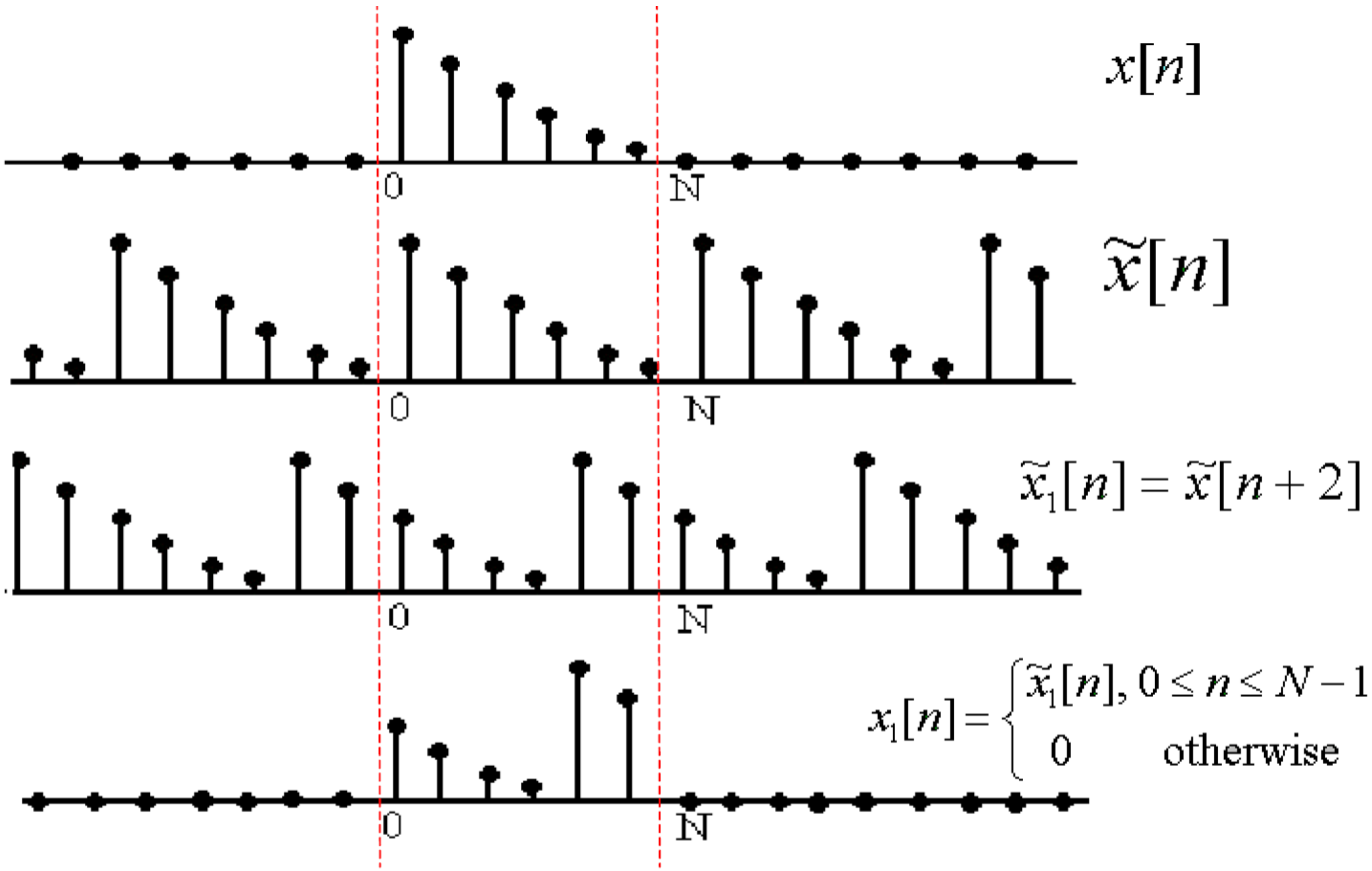


Conventional shift
(linear shift)

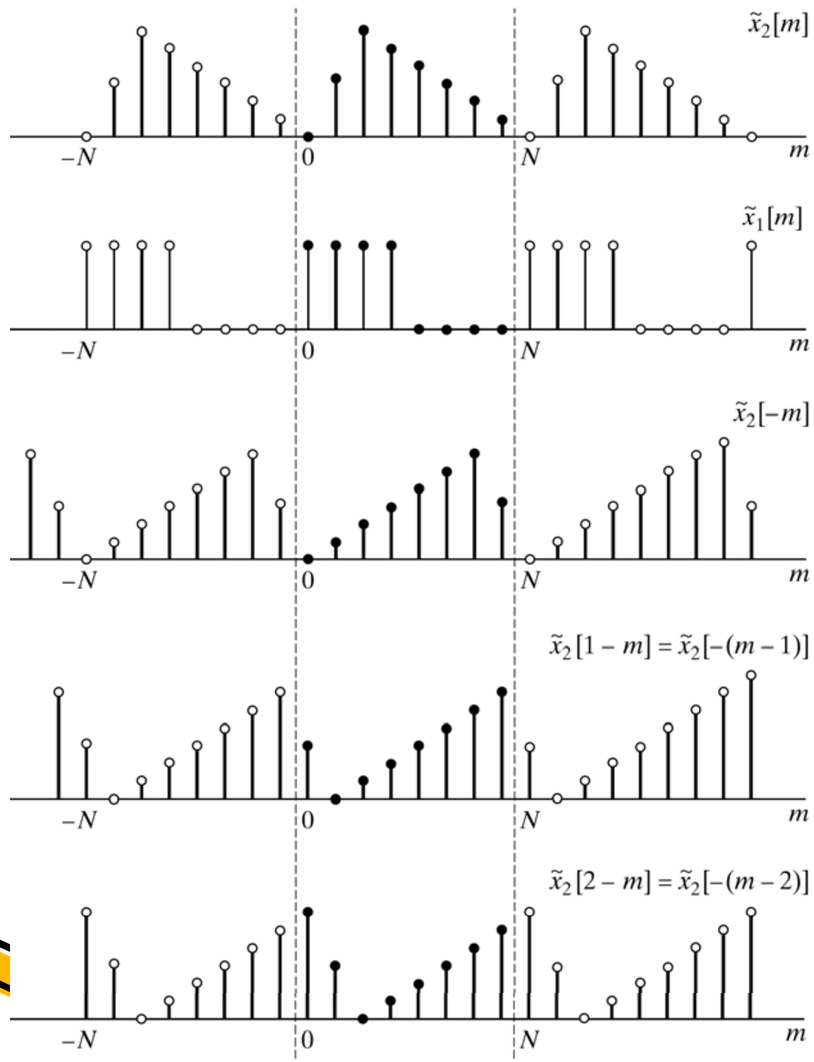


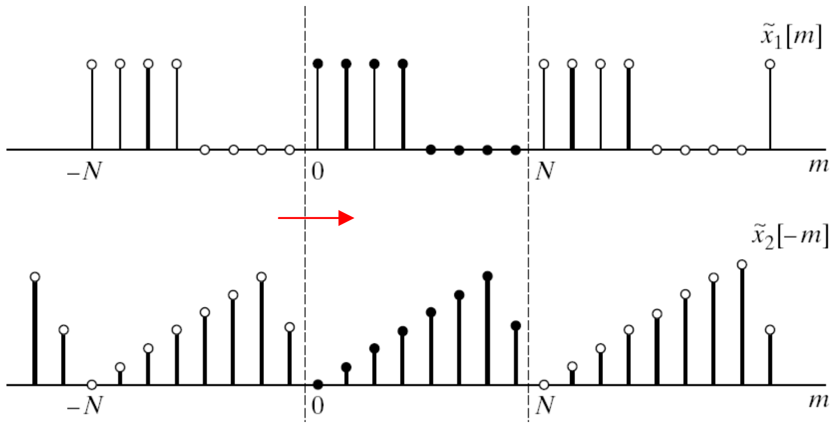
Circular Shift Example

$$x_1[n] \overset{\Delta}{=} x[((n-m))_N] \qquad (0 \le n \le N-1)$$



Periodic/Circular Convolution





Circular Shift

Circular Convolution Definition

- Suppose two finite-length duration sequences: $x_1[n]$ and $x_2[n]$ of length N

$$x_3[n] = \sum_{m=0}^{N-1} \tilde{x}_1[m]\,\tilde{x}_2[n-m] \quad 0 \le n \le N-1$$

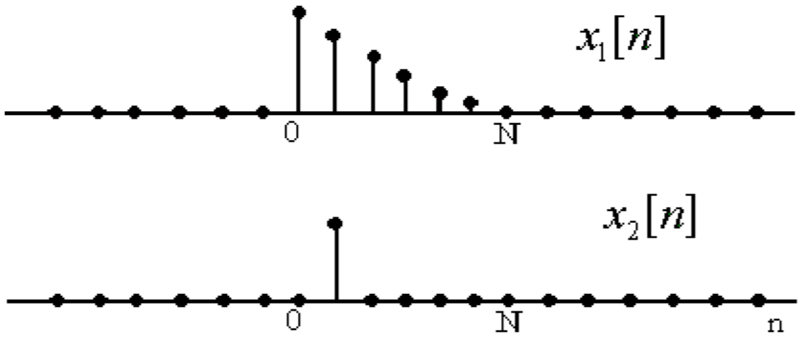
$$\text{or} \quad x_3[n] = \sum_{m=0}^{N-1} x_1[((m))_N]\,x_2[((n-m))_N] \quad 0 \le n \le N-1$$

$x_3[n]$ is also a finite-length duration sequences of length N

Computation for Circular Convolution

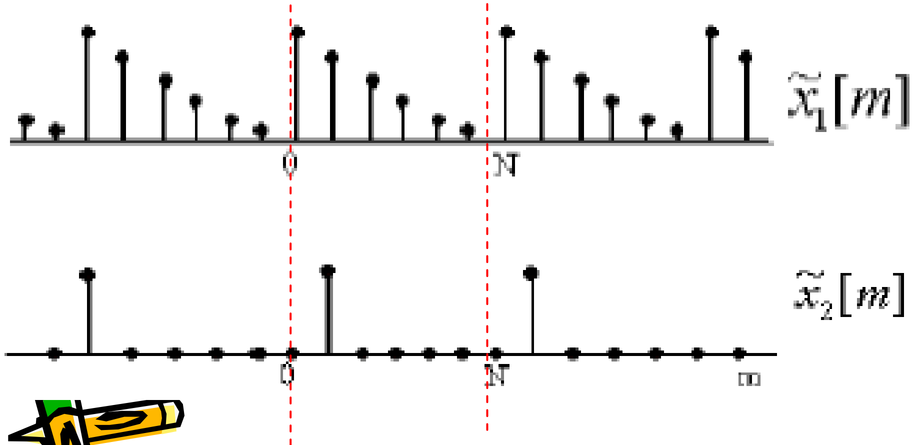
1. To period the two sequence with period N (large enough)
2. To compute the periodic convolution of the two periodic sequences
3. To get out the duration sequence between [0, N-1]

Example



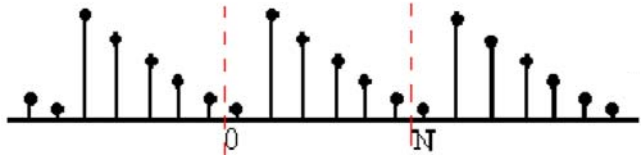
Step 1

Periodic the sequences:



Step 2

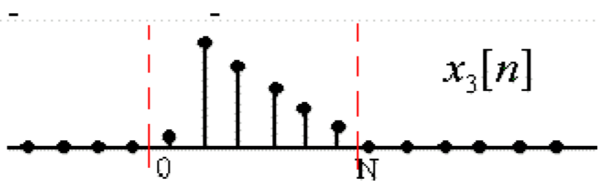
Periodic convolution



$$\tilde{x}_3[n] = \sum_{m=0}^{N-1} \tilde{x}_1[m]\tilde{x}_2[n-m]$$

Step 3

Get out a period



Circular Convolution Property

- Usually, we use the following notation to represent the circular convolution of length N

$$x_3[n] = x_1[n] \circledN x_2[n]$$

- Circular convolution property

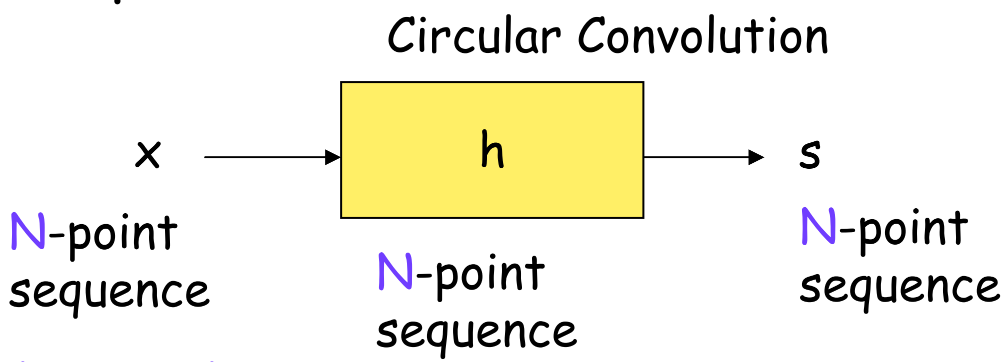
$$x_1[n] \circledN x_2[n] \overset{DFT}{\longleftrightarrow} X_1[k]X_2[k]$$

$$x_1[n]x_2[n] \overset{DFT}{\longleftrightarrow} \frac{1}{N} X_1[k] \circledN X_2[k]$$

where $X_1[k] \circledN X_2[k] = \sum_{l=0}^{N-1} X_1[l] X_2[((k-l))_N]$

Circular Convolution Implementation

- Direct Implementation



4×4 cyclic convolution

$$\begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \end{bmatrix} = \begin{bmatrix} h_0 & h_3 & h_2 & h_1 \\ h_1 & h_0 & h_3 & h_2 \\ h_2 & h_1 & h_0 & h_3 \\ h_3 & h_2 & h_1 & h_0 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

~ $O(N^2)$

16 multiplications
12 additions

Using Circular Convolution to Implement Linear Convolution

- Consider two sequences $x_1[n]$ of length L and $x_2[n]$ of length P, respectively
- The linear convolution $x_3=x_1[n] * x_2[n]$

$$x_3[n] = \sum_{m=-\infty}^{\infty} x_1[m] x_2[n-m]$$

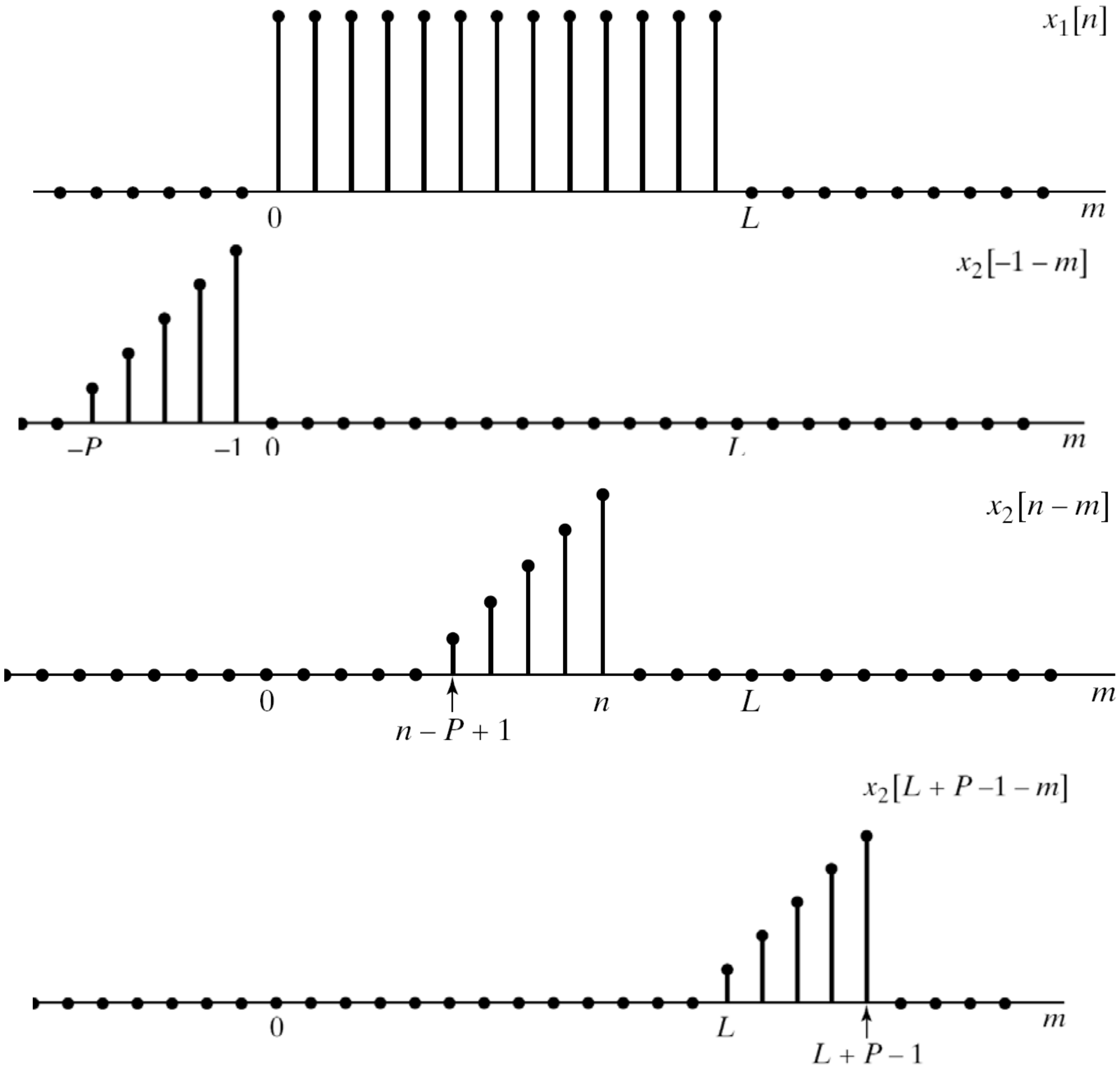
a sequence of length L+P-1

The same concept related to Winogrand Algorithm

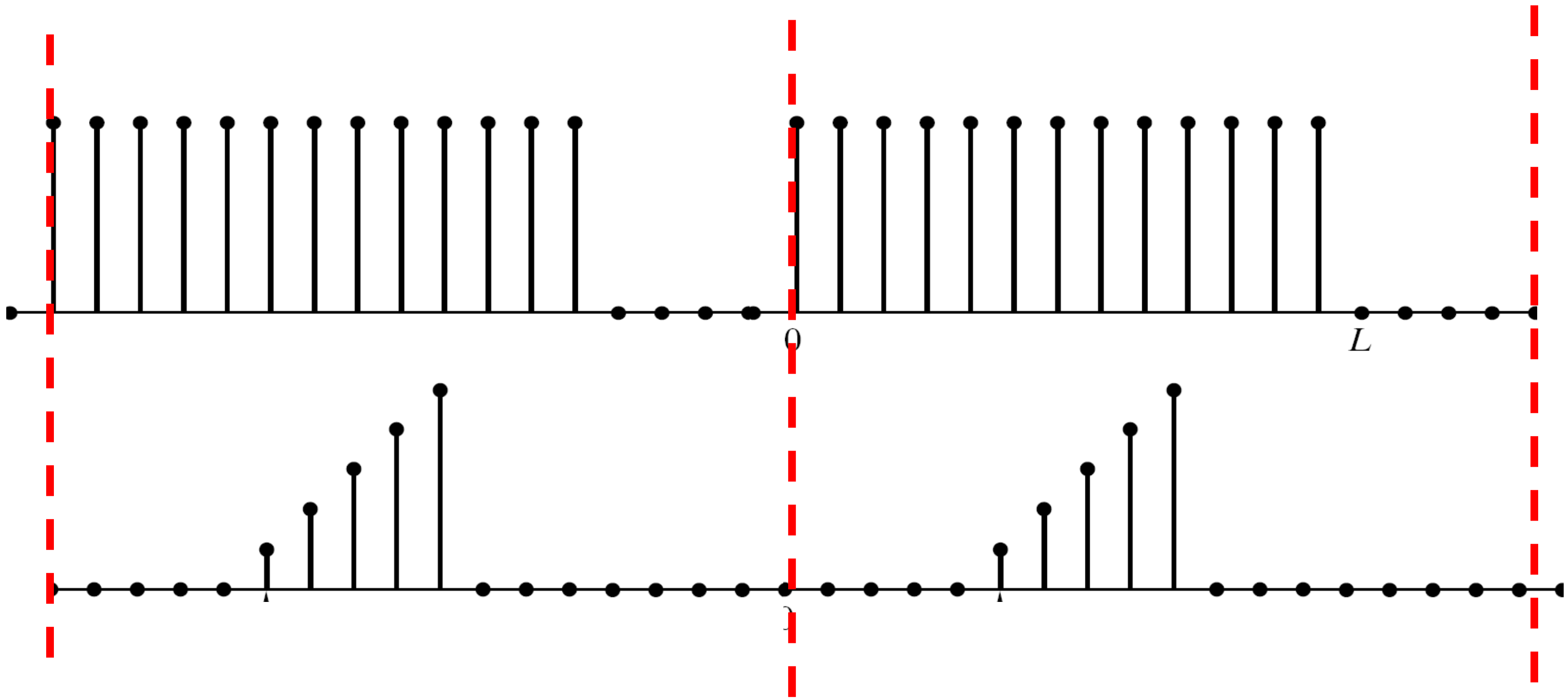
- Choose N, such that $N \geq L+P-1$, then

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

Linear Convolution



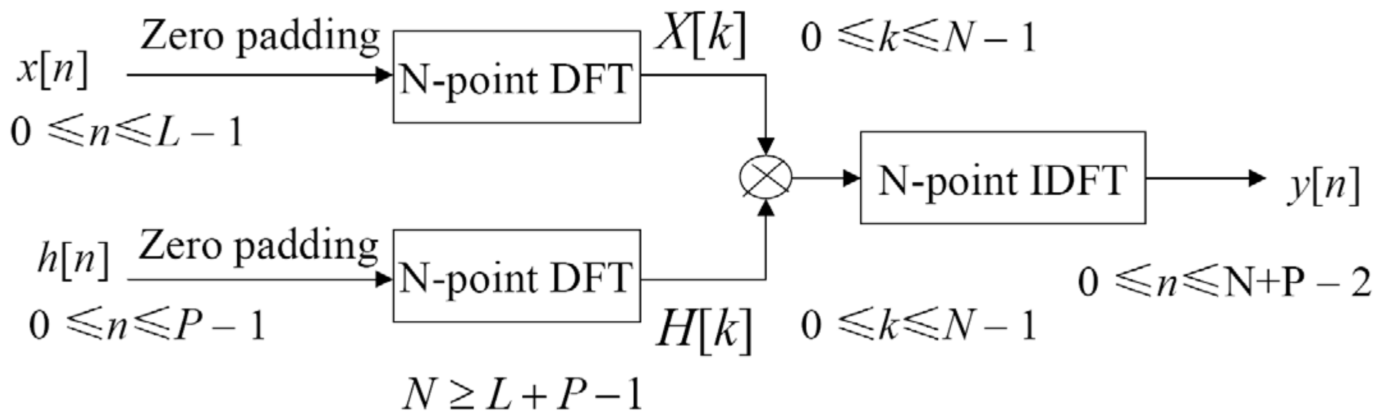
Circular Convolution with N=L+P-1



Time aliasing in the circular convolution of two finite-length sequence can be avoided if $N \geq L+P-1$

Concluding Remarks

- The convolution of two finite-length sequences can be interpreted by circular convolution with large enough length
- Circular convolution can be implemented by DFT/FFT



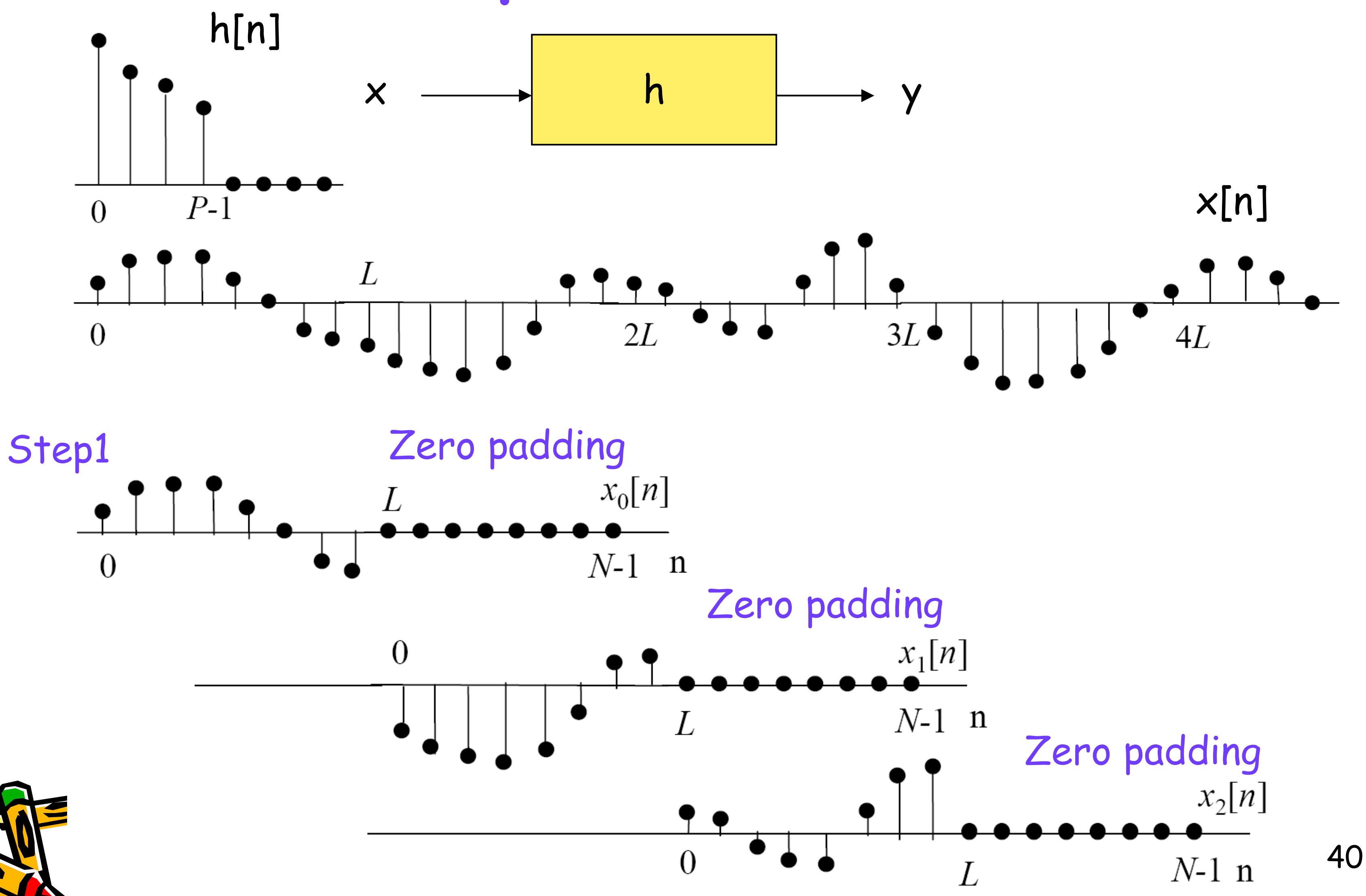
- However, in real applications....
  - For an FIR system, the input sequence is of indefinite duration
  - To store the entire input signal requires ?
    - A large delay in processing
    - An indefinite memory
  - Block convolution

Block Convolution

- Step1: To segment a sequence into sections of length L
- Step2: Each section is convolved with the finite-length impulse response of length P by using DFT/FFT of length N=L+P-1
- Step3: The filtered sections are fitted together in an appropriate way

- Overlap-add method
- Overlap-save method

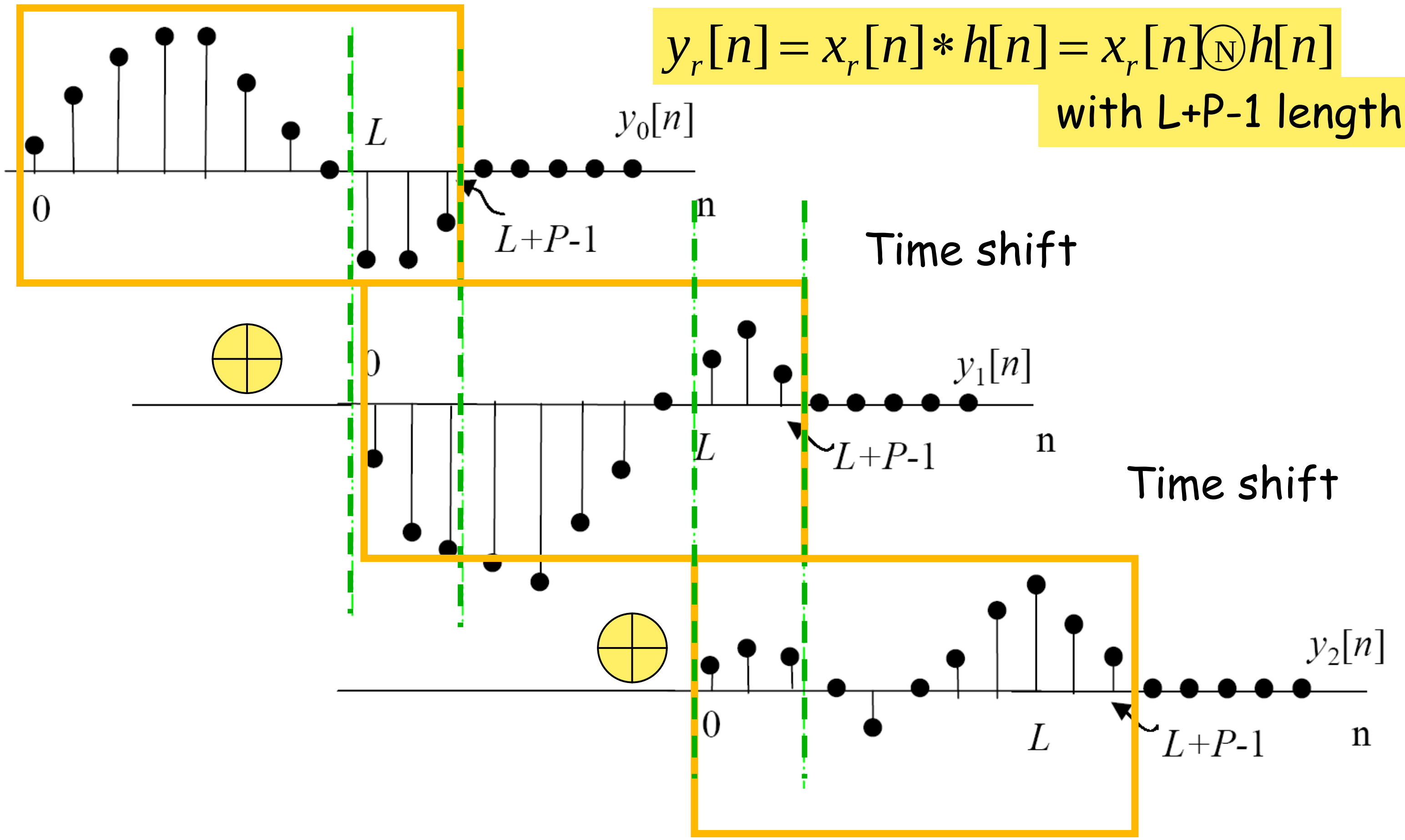
Overlap-Add Method



Step2 & Step3

$$y_r[n] = x_r[n] * h[n] = x_r[n] Ⓝ h[n]$$

with L+P-1 length

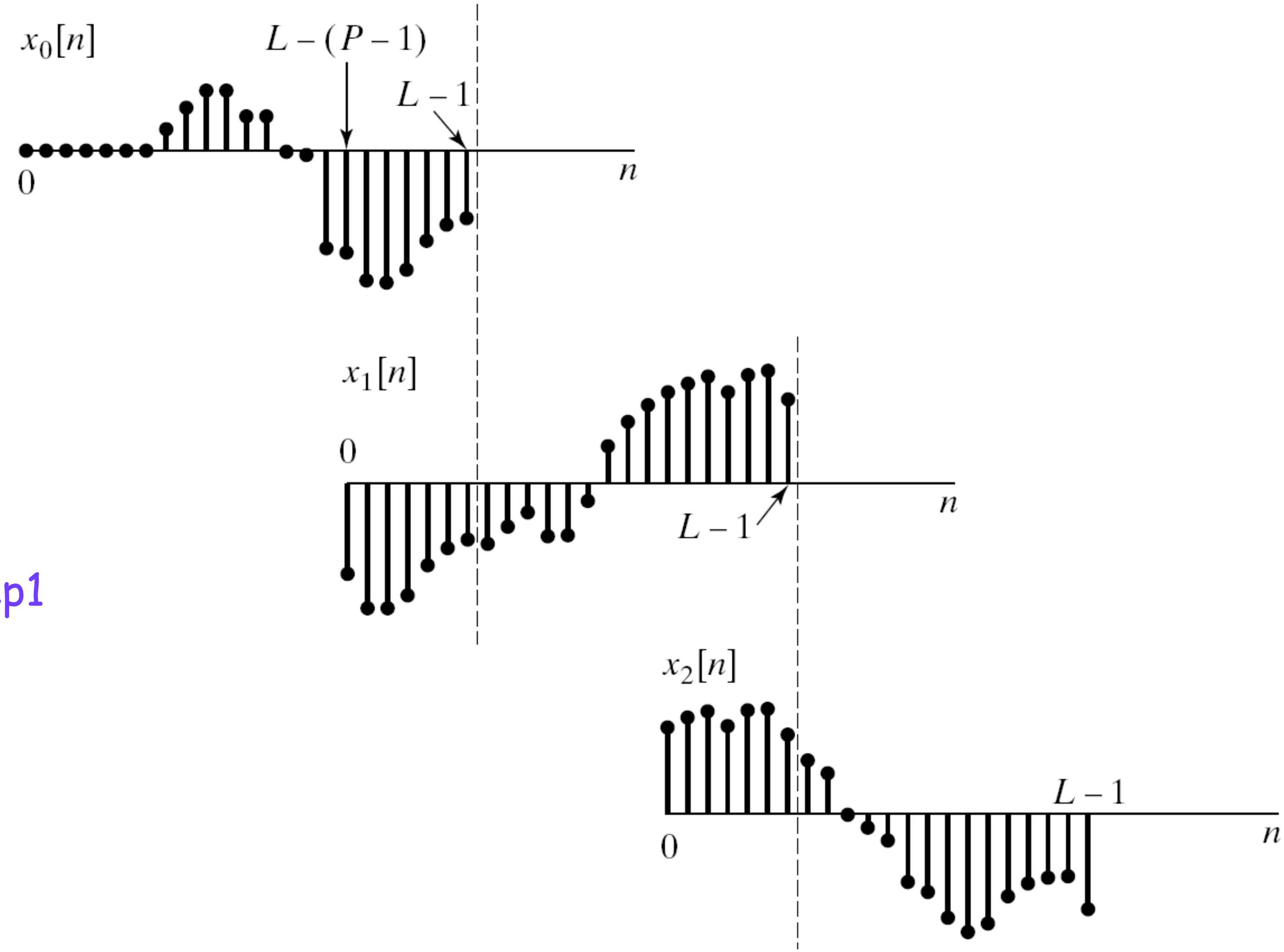


$$y[n] = x[n] * h[n] = \sum_{r=0}^{\infty} y_r[n - rL]$$

Overlap-Save Method

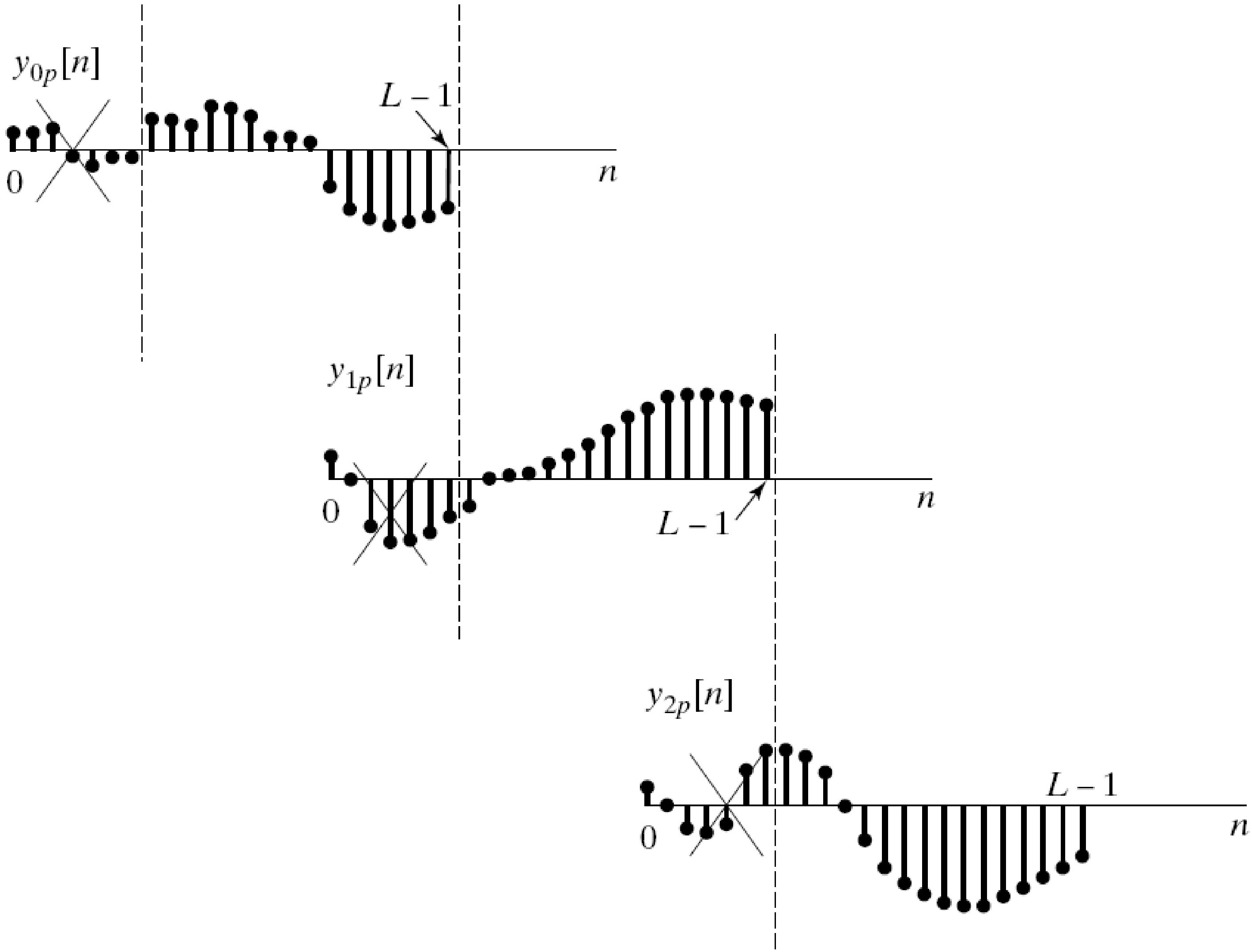
- Suppose $L > P$.
- Consider an L-point circular convolution of a P-point impulse response h[n] with an L-point input sequence $x_r[n]$
  - Due to aliasing problem, the first (P-1)-point of the result is incorrect
  - the remaining points [P, L-1] are identical to those that would be obtained by linear convolution

- Step1: To segment a sequence into sections of length L such that each section overlaps the preceding section by (P-1) points
- Step2: Each section is convolved with the finite-length impulse response of length P by using DFT/FFT of length L
- Step3: The first (P-1)-point of each filtered sequence must be discarded. The remaining samples from successive sections are then abutted to construct the final output.



Step1

Step2
&
Step3



Fast Convolution with the FFT

- Given two sequences x_1 and x_2 of length N_1 and N_2 respectively
  - Direct implementation requires N_1N_2 complex multiplications
- Consider using FFT to convolve two sequences:
  - Pick N, a power of 2, such that $N \geq N_1+N_2-1$
  - Zero-pad x_1 and x_2 to length N
  - Compute N-point FFTs of zero-padded x_1 and x_2, one obtains X_1 and X_2
  - Multiply X_1 and X_2
  - Apply the IFFT to obtain the convolution sum of x_1 and x_2
  - Computation complexity: $2(N/2)\log_2 N + N + (N/2)\log_2 N$

Example

- A sequence *x[n]* of length 1024
- FIR filter *h[n]* of length 34

- Direct computation: 34×1024=34816
- Using radix-2 FFT: 35840 (N=2048)
- Using overlap-add radix-2 FFT:
  - *x[n]* is segmented into a set of contiguous blocks of equal length 95
  - Apply radix-2 FFT of length 128
  - Each segment requires 1472 multiplications
  - This algorithm requires total 16192 multiplications