Discrete Fourier Transform (DFT)

- DFT:
  - Fourier Transform of short duration signals
- DFT: Sampling of the DTFT
  - What happens when we sample in the frequency domain?
- Convolution with DTF
- DFT of long signals
  - The effect of windowing
- The DFT as a Linear Transform X=Wx
  - DFT as a vector-matrix operation
- FFT

M. Amer
Concordia University, Department of ECE

This lecture is based on:

- Chapter 8, A.V. Oppenheim and R.W. Schafer, *Discrete-Time Signal Processing*, Prentice-Hall, 3rd ed, 2010.
- Slides from http://faculty.nps.edu/rcristi/EC3400online/weekly_schedule/week3.htm

Discrete-time signal transforms

Transform Name	Forward Transform	Inverse Transform	Notes
z-Transform	$X(z) = \sum_{n=-\infty}^{\infty} x[n]z^{-n}$ $z = re^{j\omega}$	Partial fractions, Power series, Inspection.	has ROC
DTFT * continuous in freq.	$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n}$	$x[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(e^{j\omega})e^{j\omega n} d\omega$	Periodic (2π)
DFS * periodic signal * discrete in freq.	$\tilde{X}[k] = \sum_{n=0}^{N-1} \tilde{x}[n]W_N^{nk}$	$\tilde{x}[n] = \frac{1}{N} \sum_{k=0}^{N-1} \tilde{X}[k]W_N^{-nk}$	$W_N = e^{-j\frac{2\pi}{N}} = e^{-j\omega_o}$ $\tilde{x}[n] = \tilde{x}[n+N]$ N is the period
DFT * discrete in freq. * samples from DTFT.	$X[k] = \sum_{n=0}^{N-1} x[n]W_N^{nk}$	$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k]W_N^{-nk}$	$W_N = e^{-j\frac{2\pi}{N}} = e^{-j\omega_o}$

- DTFT: Discrete-time Fourier tarnsform.
- DFS : Discrete Fourier series.
- DFT : Discrete Fourier transform

Fourier Analysis of Discrete Time Signals

For a discrete-time sequence *x[n]*, we define two classes of Fourier Transforms:

1. the DTFT (Discrete Time FT) for sequences having **infinite** duration

$$X(\omega) = DTFT\left\{x(n)\right\} = \sum_{n=-\infty}^{+\infty} x(n)e^{-j\omega n}$$

$$x(n) = IDTFT\left\{X(\omega)\right\} = \frac{1}{2\pi}\int_{-\pi}^{+\pi} X(\omega)e^{j\omega n}d\omega$$

2. the DFT (Discrete FT) for sequences having **finite** duration

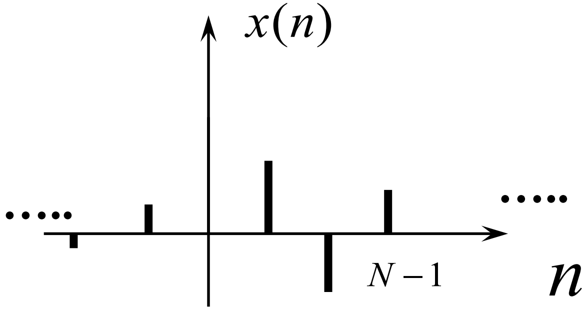
$$X(k) = \sum_{n=0}^{N-1} x(n){w_N}^{kn}, \quad w_N = e^{-j2\pi/N}$$

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k){w_N}^{-kn}, \quad w_N = e^{-j2\pi/N}$$

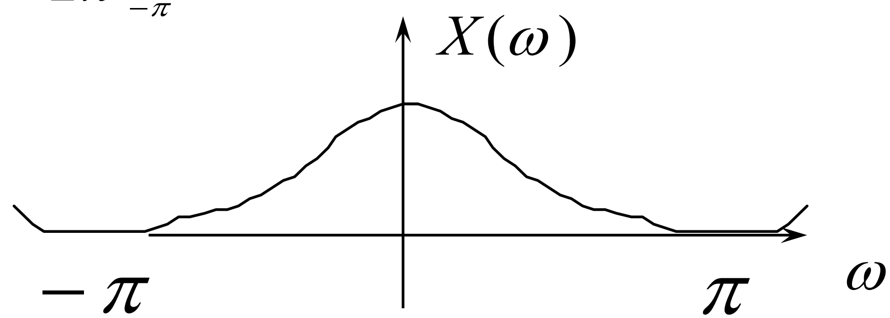
Discrete Time Fourier Transform (DTFT)

$$X(\omega) = DTFT\{x(n)\} = \sum_{n=-\infty}^{+\infty} x(n) e^{-j\omega n}$$

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



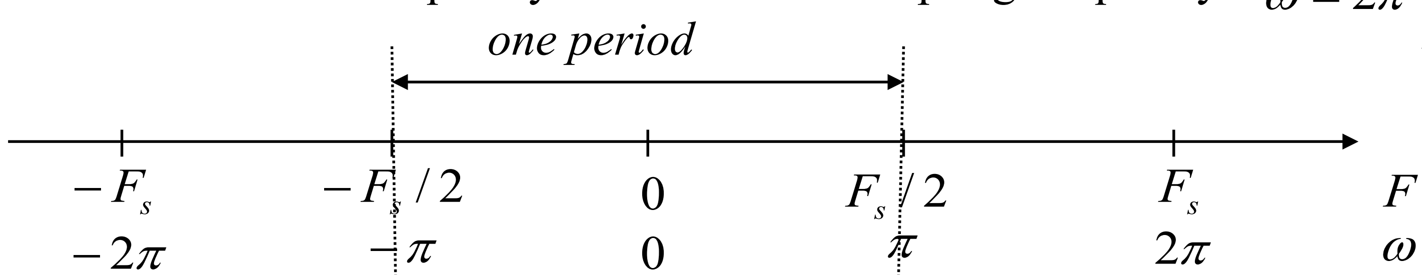
discrete time



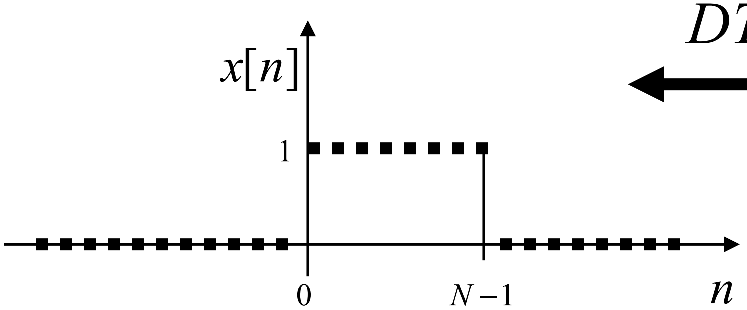
continuous frequency

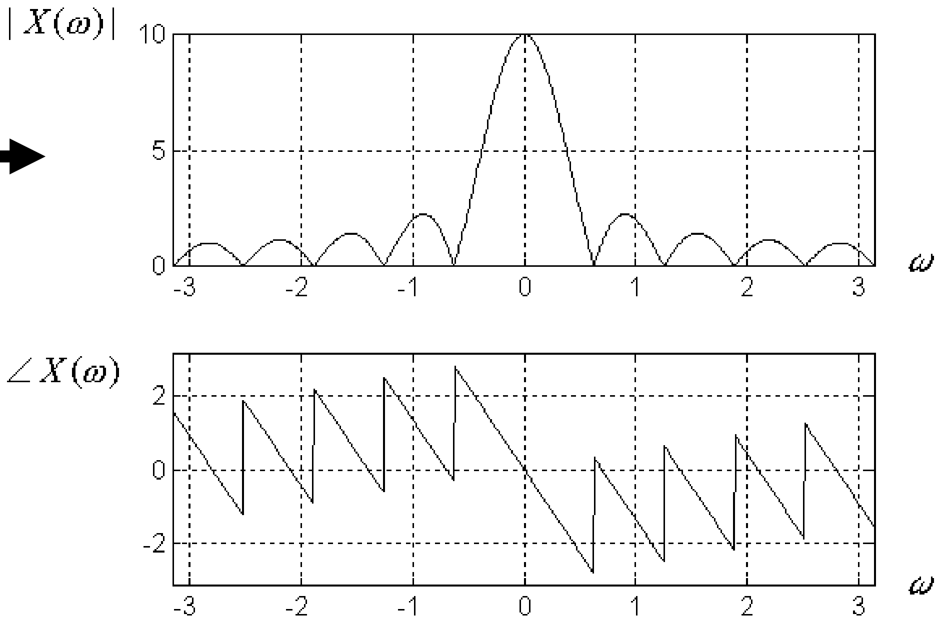
Observations:

- The DTFT is periodic with period 2π
- The frequency ω
  - is the digital frequency and is therefore limited to the interval $-\pi < \omega < +\pi$
  - is a normalized frequency relative to the sampling frequency $\omega = 2\pi \frac{F}{F_s}$



Example 1:



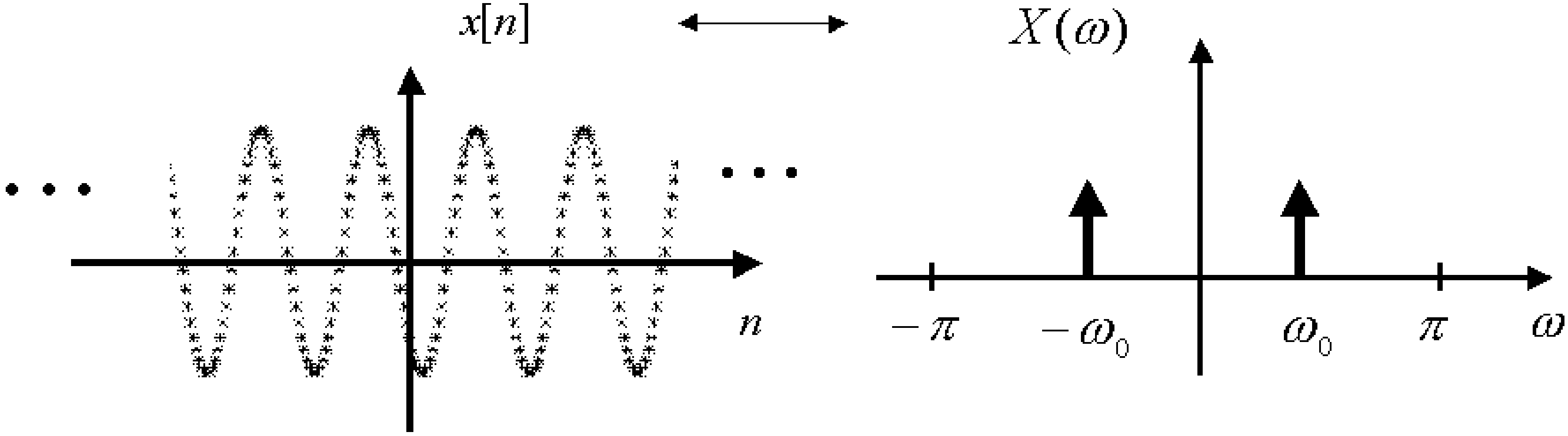


since

$$X(\omega) = \sum_{n=0}^{N-1} e^{-j\omega n} = \frac{1-e^{-j\omega N}}{1-e^{-j\omega}}$$

$$= e^{-j\omega(N-1)/2} \frac{\sin(\omega N/2)}{\sin(\omega/2)}$$

Example 2:



$$x[n] = A\cos(\omega_0 n + \alpha)$$

$$X(\omega) = A\pi e^{j\alpha}\delta(\omega - \omega_0) + \\ + A\pi e^{-j\alpha}\delta(\omega + \omega_0)$$

Discrete Fourier Transform (DFT)

- Given a discrete-time finite sequence $x = [x(0), x(1), \ldots, x(N-1)]$

its DFT is a discrete-frequency finite sequence

$$X = DFT(x) = [X(0), X(1), \ldots, X(N-1)]$$

$$X(k) = \sum_{n=0}^{N-1} x(n) w_N^{\;kn}, \quad w_N = e^{-j2\pi/N}$$

- Given a discrete-frequency finite sequence $X = [X(0), X(1), \ldots, X(N-1)]$

its inverse DFT (IDFT) is a discrete-time finite sequence

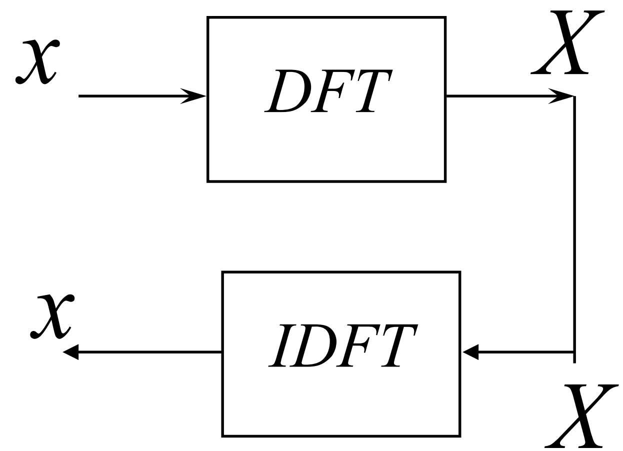
$$x = IDFT(X) = [x(0), x(1), \ldots, x(N-1)]$$

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) w_N^{\;-kn}, \quad w_N = e^{-j2\pi/N}$$

Observations:

- The DFT and the IDFT form a transform pair

back to the same signal !

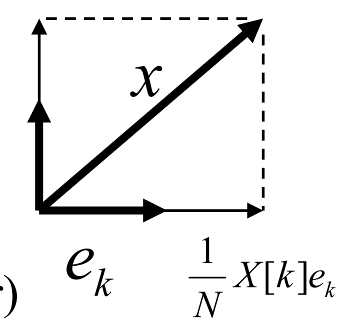


- *The DFT is a numerical algorithm ➔ it can be computed by a digital computer*

- DT exponentials are orthogonal, i.e., the dot-product of complex exponentials of the same frequency is *N,* meaning they are mutually independent from one another

$$\sum_{k=0}^{N-1} W_N^{k(l-n)} = \begin{cases} N, & l = n \\ 0, & l \neq n \end{cases}$$

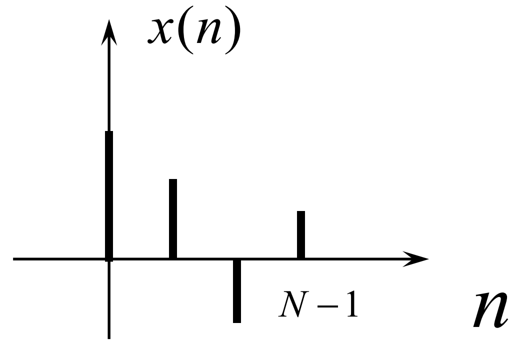
$= N\delta[(k)_N]$



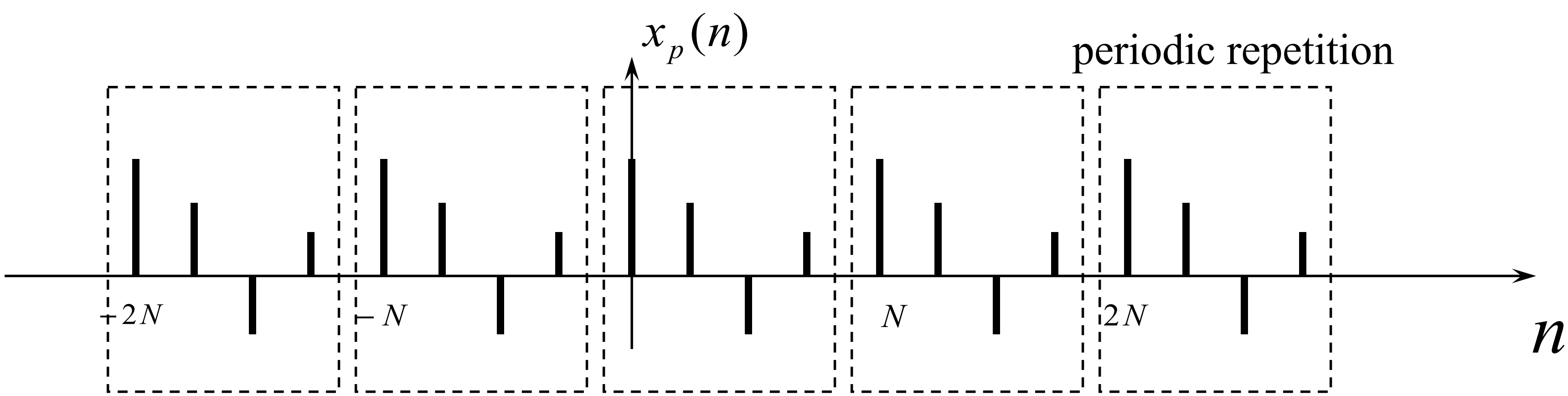
➔we can view DFT as orthogonal linear transform X=**W**x (see later)

- **Periodicity:** From the IDFT expression, we notice that the sequence *x(n)* can be interpreted as one period of a periodic sequence $x_p(n)$:

$$x_p(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k)w_N^{-kn} = \frac{1}{N}\sum_{k=0}^{N-1} X(k)w_N^{-kn}w_N^{-kN} = \frac{1}{N}\sum_{k=0}^{N-1} X(k)w_N^{-k(n+N)} = x_p(n+N)$$



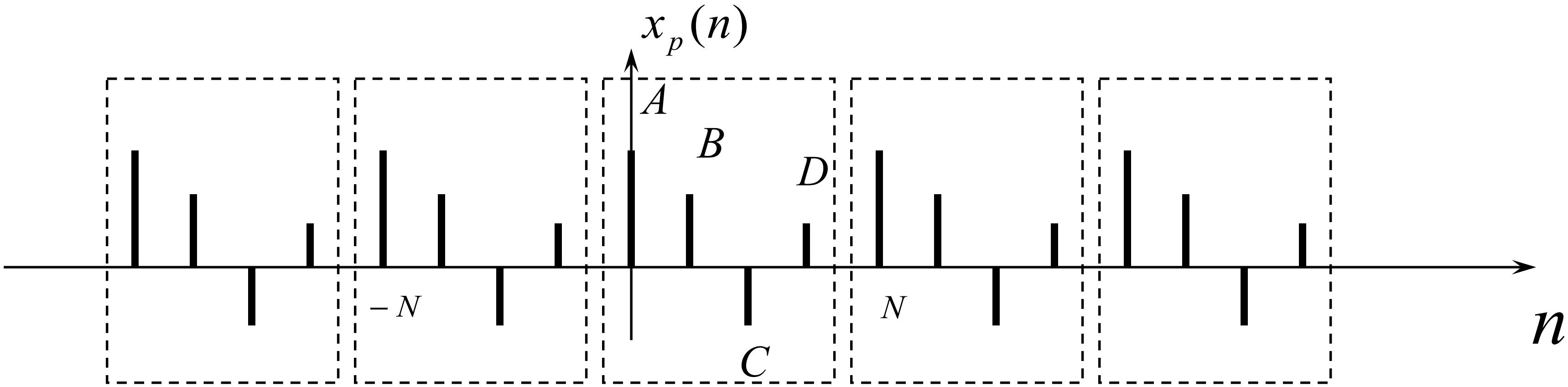
original sequence

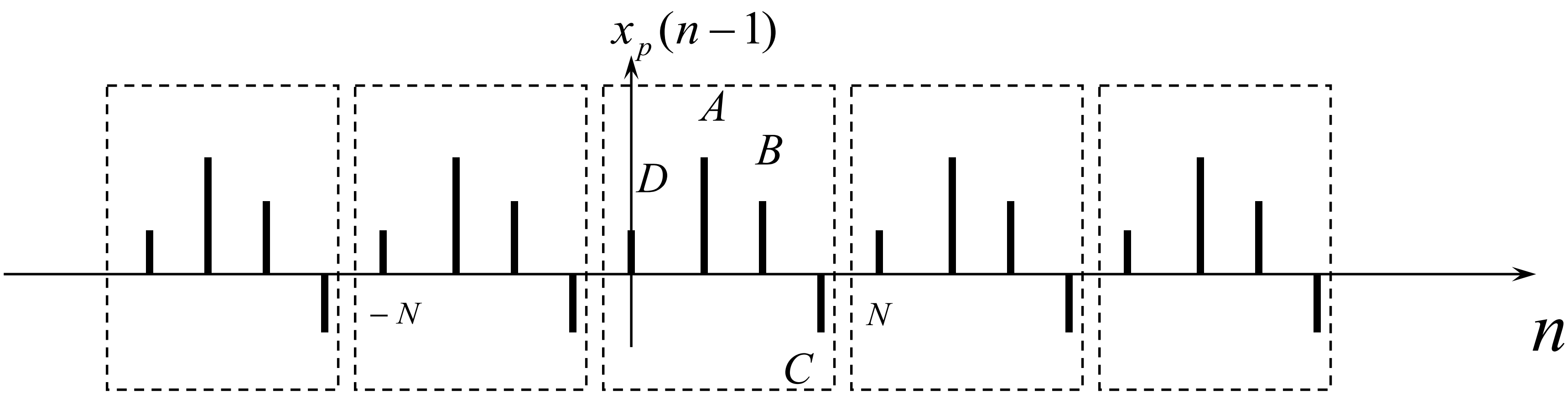


periodic repetition

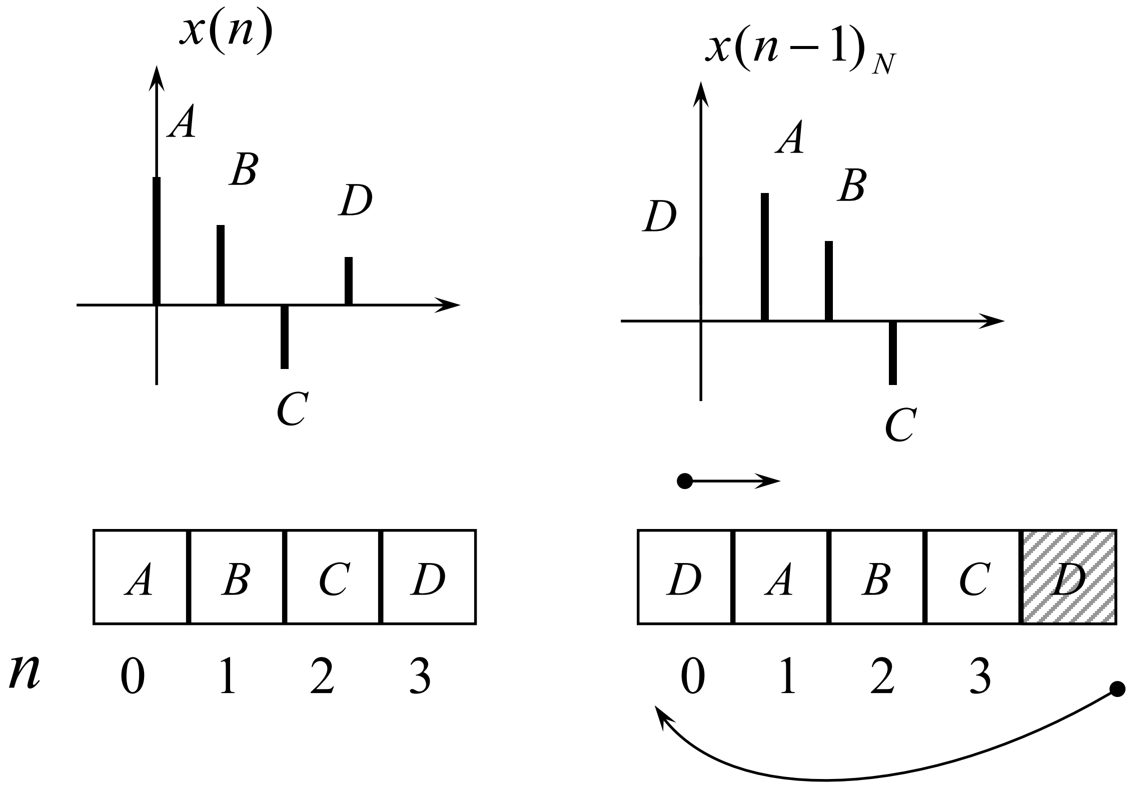
- This has a consequence when we define a time shift of the sequence

- For example, see what we mean with $x(n-1)$: start with the periodic extension $x_p(n)$





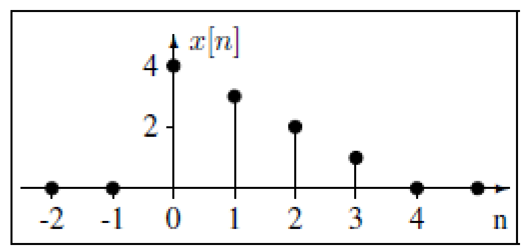
If we look at just one period we can define the *circular shift*



Modulo function

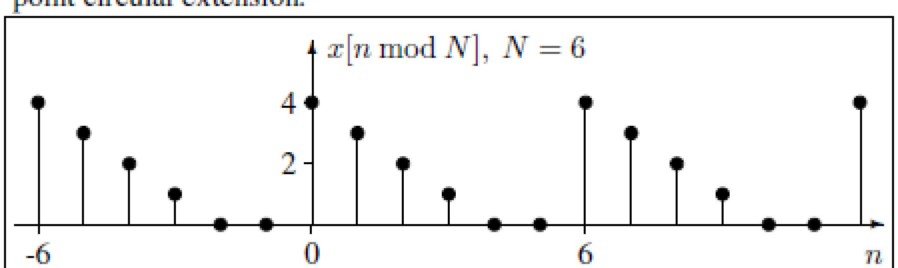
- The notation $(k)_N$ or $k \bmod N$ denotes the remainder when k is divided by N
  - For negative k, the remainder is between 0 and N-1
  - $(3)4=3$; $(6)4=2$; $(-3)4 =((-1)(4)+1))4=1$; $(-6)4 = ((-2)(4) + 2) = 2$
  - $(k)_N$ is a periodic function of k with period N : $(k + N)_N = (k)_N$

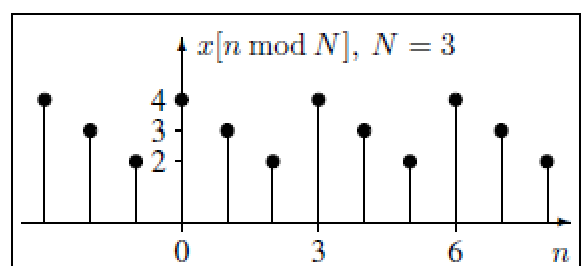
- For **time-limited** signal $x[n]$, the N **-point circular extension** of $x[n]$ is $x((n))_N = x[n \bmod N]$
  - $x[n \bmod N]$ is N -periodic signal
  - $x[n \bmod N]$ consists of shifted replicates of $x[n]$



Example. $x[n] = \{\underline{4}, 3, 2, 1\} = \begin{cases} 4-n, & 0 \le n \le 3 \\ 0, & \text{otherwise}, \end{cases}$

N-point circular extension:

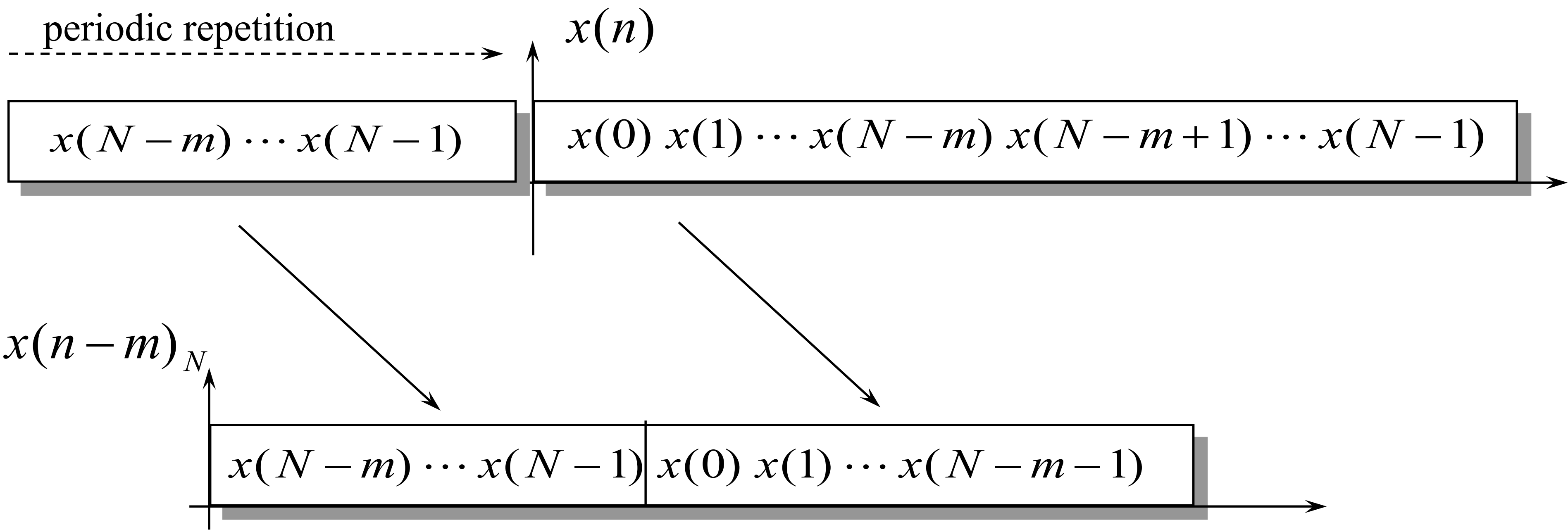




k	$\langle k\rangle_4$
$\vdots$	$\vdots$
-8	0
-7	1
-6	2
-5	3
-4	0
-3	1
-2	2
-1	3
0	0
1	1
2	2
3	3
4	0
5	1
6	2
7	3
$\vdots$	$\vdots$

Properties of the DFT

- One to one $\quad x(n) \leftrightarrow X(k) \quad$ with no ambiguity
- Time shift $\quad DFT\left[x(n-m)_N\right] = w_N^{\;km} X(k)$ where $x(n-m)_N$ is a circular shift



- Circular time reversal: $x[(-n)_N]$

$$
x[(-n)_N] = \begin{cases} x[0]; & n = 0 \\ x[N-n]; & 1 \le n \le N-1 \\ periodic; & otherwise \end{cases}
$$

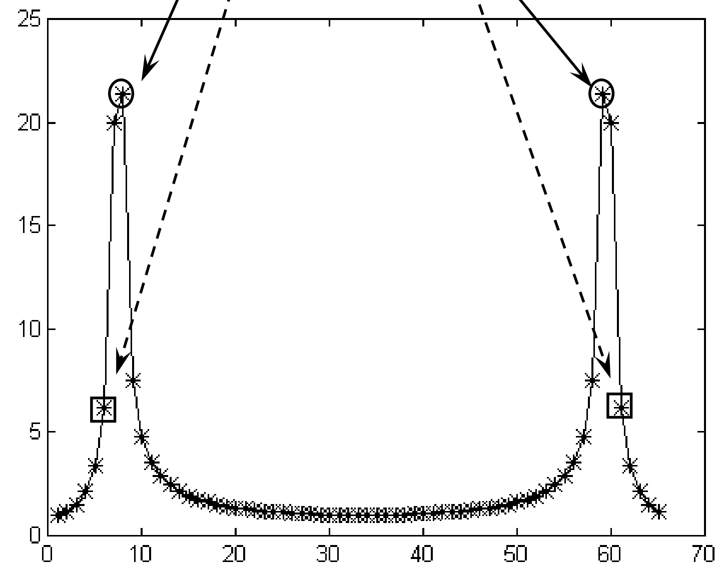
Example 1: $x[n] = (1; 3; 5; 2)$; $x[(-n)_4] = (1; 2; 5; 3)$;

Example 2: $x[n] = (10; 11; 12; 13; 14)$; $x[(-n)_6] = (10; 0; 14; 13; 12; 11)$;

Properties of the DFT

- Real sequences $X(k) = X^*(N-k)$

$$|X(k)| = |X(N-k)|$$



- Circular convolution

$$y(n) = x_1(n) \otimes x_2(n)$$
$$= \sum_{k=0}^{N-1} x_1(k) \underbrace{x_2(n-k)_N}_{circular\ shift}$$

where both sequences x_1, x_2 must have the same length N.

Then:

$$DFT\left[x_1(n) \otimes x_2(n)\right] = X_1(k) X_2(k), \quad k = 0, \ldots, N-1$$

Properties of the DFT

- **Periodicity** X[k + N] = X[k]
  - $x_p(n)$ = IDFT(X); frequency-domain sampling leads to periodic replication in the time domain
- A signal $x[n]$ is called ***N*-point circularly even** iff its *N*-point circular extension, $x[n \bmod N]$, is even, i.e., $x[n \bmod N] = x[-n \bmod N]$
  - Example 1: $x[n] = \{\underline{4}, 3, 2, 1\}$ is not 4-point circularly even since $x[n \bmod N]$ is not even
  - Example 2: $x[n] = \{\underline{4}, 3, 2, 1, 2, 3\}$ is 6-point circularly even since $x[n \bmod 6]$ is even but not 8-point circularly even since $x[n \bmod 8]$ is not even

- If $x[n]$ is real, then its DFT has **circular symmetry** $X[k] = X^*[-k \bmod N]$

- If $x[n]$ is real and circularly even, then its $X[k]$ is also real and circularly even

TABLE 8.2 SUMMARY OF PROPERTIES OF THE DFT

All operations modulo N,
i.e., output in the range $0 \rightarrow N-1$

Finite-Length Sequence (Length N)	N-point DFT (Length N)
1. $x[n]$	$X[k]$
2. $x_1[n], x_2[n]$	$X_1[k], X_2[k]$
3. $ax_1[n]+bx_2[n]$	$aX_1[k]+bX_2[k]$
4. $X[n]$	$Nx[((-k))_N]$
5. $x[((n-m))_N]$	$W_N^{km}X[k]$
6. $W_N^{-\ell n}x[n]$	$X[((k-\ell))_N]$
7. $\sum_{m=0}^{N-1} x_1[m]x_2[((n-m))_N]$	$X_1[k]X_2[k]$
8. $x_1[n]x_2[n]$	$\frac{1}{N}\sum_{\ell=0}^{N-1} X_1[\ell]X_2[((k-\ell))_N]$
9. $x^*[n]$	$X^*[((-k))_N]$
10. $x^*[((-n))_N]$	$X^*[k]$
11. $\mathcal{R}e\{x[n]\}$	$X_{ep}[k]=\frac{1}{2}\{X[((k))_N]+X^*[((-k))_N]\}$
12. $j\mathcal{I}m\{x[n]\}$	$X_{op}[k]=\frac{1}{2}\{X[((k))_N]-X^*[((-k))_N]\}$
13. $x_{ep}[n]=\frac{1}{2}\{x[n]+x^*[((-n))_N]\}$	$\mathcal{R}e\{X[k]\}$
14. $x_{op}[n]=\frac{1}{2}\{x[n]-x^*[((-n))_N]\}$	$j\mathcal{I}m\{X[k]\}$
Properties 15–17 apply only when $x[n]$ is real.	
15. Symmetry properties	$\begin{cases} X[k] = X^*[((-k))_N] \\ \mathcal{R}e\{X[k]\} = \mathcal{R}e\{X[((-k))_N]\} \\ \mathcal{I}m\{X[k]\} = -\mathcal{I}m\{X[((-k))_N]\} \\ \lvert X[k]\rvert = \lvert X[((-k))_N]\rvert \\ \angle\{X[k]\} = -\angle\{X[((-k))_N]\} \end{cases}$
16. $x_{ep}[n]=\frac{1}{2}\{x[n]+x[((-n))_N]\}$	$\mathcal{R}e\{X[k]\}$
17. $x_{op}[n]=\frac{1}{2}\{x[n]-x[((-n))_N]\}$	$j\mathcal{I}m\{X[k]\}$

Circularly even & circularly odd

Discrete Fourier Transform (DFT)

1. DFT:
   - Fourier Transform of short duration signals
2. DFT: Sampling of the DTFT
3. Convolution with DTF
4. DFT of long signals
   - The effect of windowing
5. The DFT as a Linear Transform
   - DFT as a vector-matrix operation
6. FFT

Based on:
- Chapter 8, A.V. Oppenheim and R.W. Schafer, *Discrete-Time Signal Processing*, Prentice-Hall, 3rd ed, 2010.
- Slides from http://faculty.nps.edu/rcristi/

The DFT is sampling the DTFT

- In the DTFT $$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n}$$

  ➔ The summation over *n* is infinite

  ➔ The independent variable ω is continuous

  Thus, DTFT is not numerically computable transform

- To numerically represent the continuous frequency DTFT, we must take samples of it ➔ DFT

The DFT is sampling the DTFT

- Consider an aperiodic *x[n]* with a DTFT $x[n] \xleftrightarrow{DTFT} X(e^{j\omega})$
- Assume a sequence is obtained by sampling the DTFT

$$\tilde{X}[k] = X(e^{j\omega})\Big|_{\omega=(2\pi/N)k} = X\left(e^{j(2\pi/N)k}\right);\ 0 \le k \le L-1$$

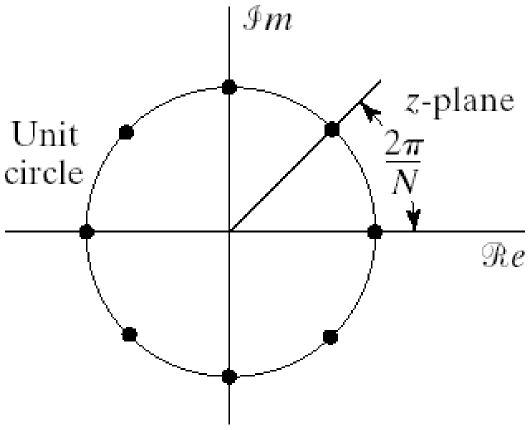
- Since the DTFT is periodic, the resulting sequence is also periodic
- $\tilde{X}[k]$ could be the DFS of a sequence & the corresponding sequence is

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

- We can also write it in terms of the z-transform

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

- The equidistant sampling points are shown in figure
  - N : the length of *x[n]*
  - L: the length of *X[k]*
  - L>=N



The DFT is sampling the DTFT

- The only assumption made on *x[n]*: its DTFT exist

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

- Combine the equations gives

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

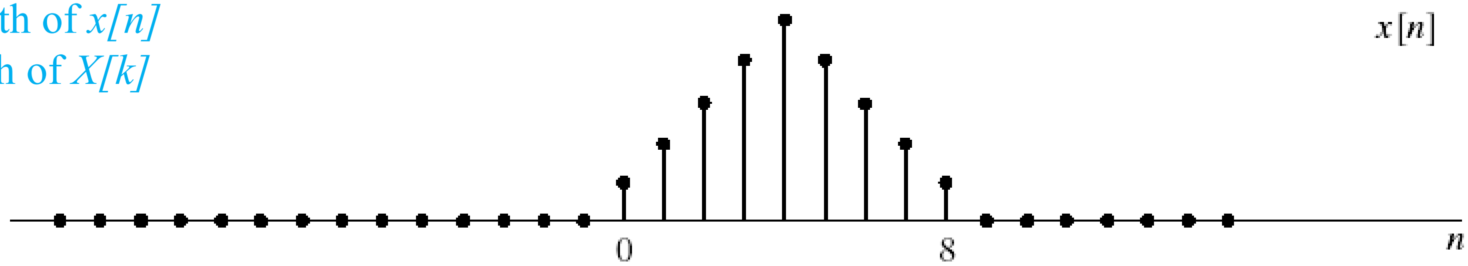
$$=\sum_{m=-\infty}^{\infty}x[m]\left[\frac{1}{N}\sum_{k=0}^{N-1}e^{j(2\pi/N)k(n-m)}\right]=\sum_{m=-\infty}^{\infty}x[m]\tilde{p}[n-m]$$

$$\tilde{p}[n-m]=\frac{1}{N}\sum_{k=0}^{N-1}e^{j(2\pi/N)k(n-m)}=\sum_{r=-\infty}^{\infty}\delta[n-m-rN]$$

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

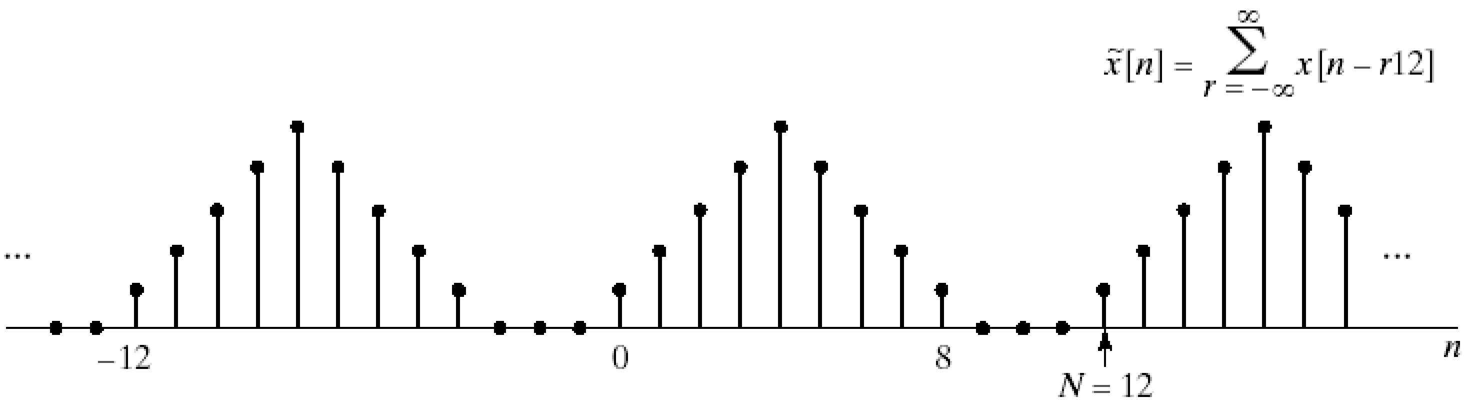
Sampling the DTFT

N : the length of *x[n]*
L: the length of *X[k]*



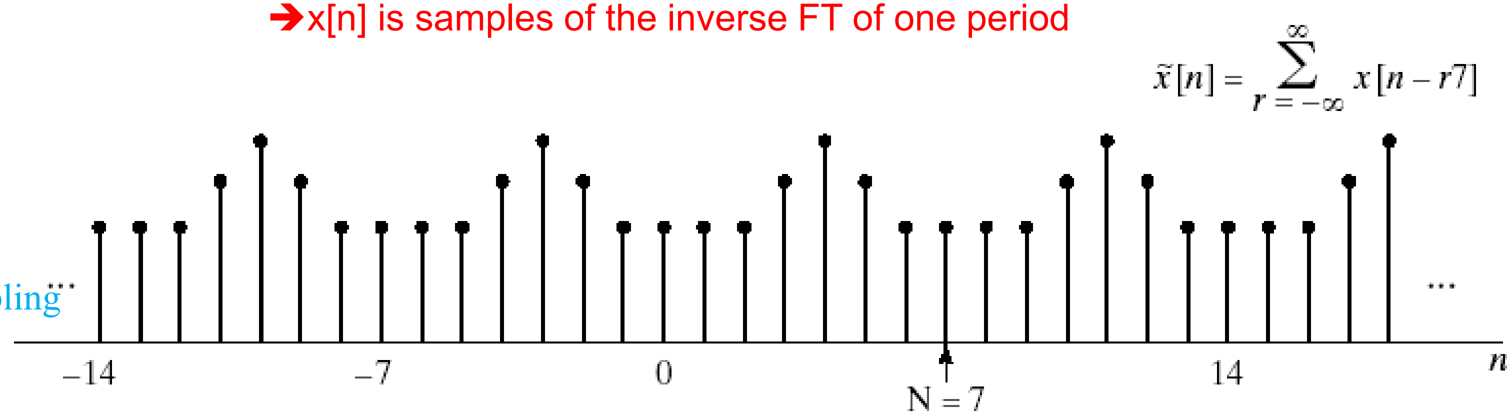
(a)

L>=N



➔x[n] is samples of the inverse FT of one period

L<N
➔aliasing
Under-sampling



➔ x[n] are still samples of the FT; But, one period is no longer identical to x[n]

Sampling the DTFT

- If $x[n]$ is of finite length **&** we take <u>sufficient number</u> of samples $X[k]$ from the DTFT, then
  - the DTFT is recoverable from these samples $X[k]$

  equivalently
  - $x[n]$ is recoverable from $\tilde{x}[n]$

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

➔No need to know the DTFT at all frequencies, to recover $x[n]$

- If not, time-domain aliasing occurs in $\tilde{x}[n]$
- Time-domain aliasing can be avoided only if
  1. $x[n]$ has finite length N (i.e., time-limited)
     - just as frequency-domain aliasing can be avoided only for signals being band-limited
  2. we take a <u>sufficient</u> number $L>=N$ of equally spaced samples $X[k]$ of the DTFT of $x[n]$
     - We must have at least as many frequency samples as the signal's length

Sampling the DTFT: summary

$$X[k] \xleftrightarrow{DFT} x[n]$$

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

$0 \le k \le L-1,\ L >= N$

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

$0 \le k \le L-1,\ where\ \ L >= N$

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk} \quad \text{analysis equation}$$

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{-kn} \quad \text{synthesis equation}$$

In order to avoid time-domain aliasing, The sampling duration of $X(e^{j\omega})$ must be $\leq \frac{2\pi}{N}$. N is the length of $x(n)$.

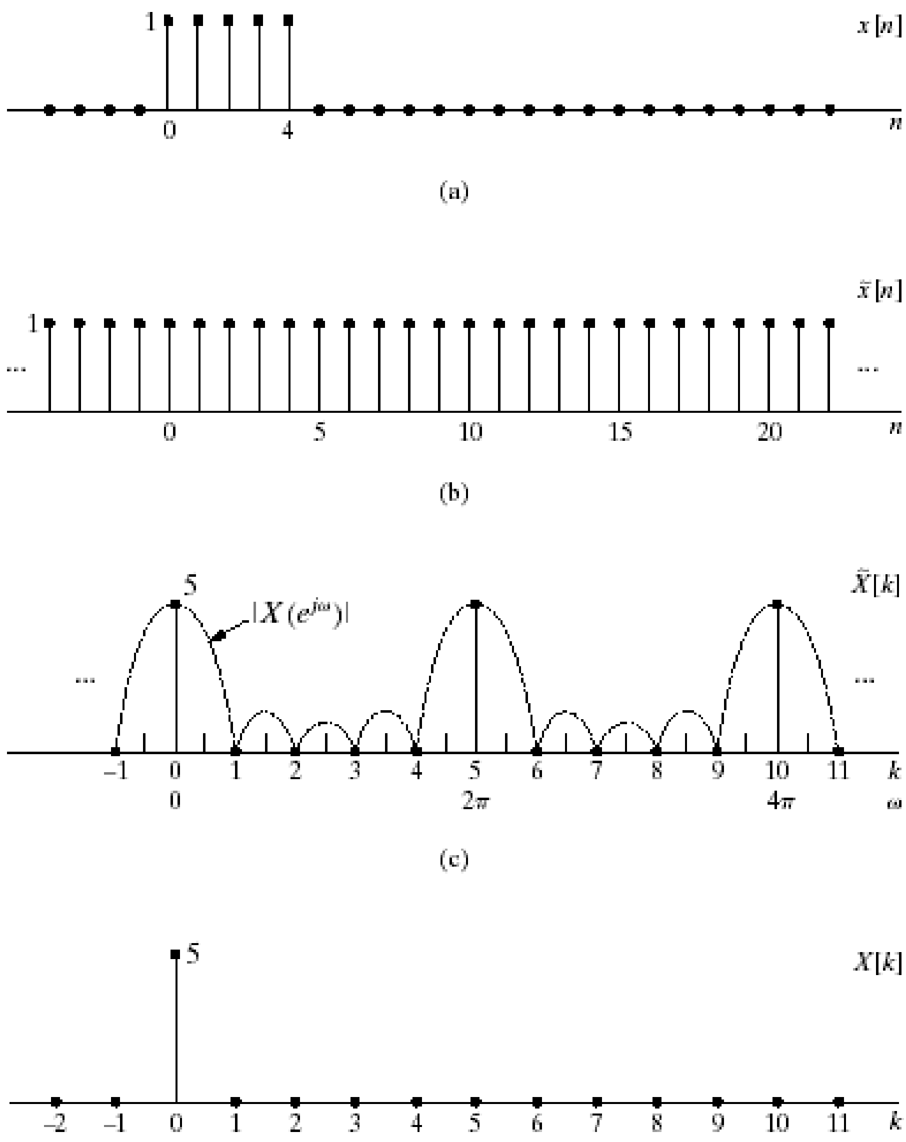
DFT: Example 1

- DFT of a rect. pulse *x[n]*, *N*=5
- Consider *x[n]* of any length L>=N
- Let L=N=5
- Calculate and sample DTFT

$$\tilde{X}[k] = \sum_{n=0}^{4} e^{-j(2\pi k/5)n}$$

$$= \frac{1 - e^{-j2\pi k}}{1 - e^{-j(2\pi k/5)}}$$

$$= \begin{cases} 5 & k = 0, \pm 5, \pm 10, \ldots \\ 0 & \text{else} \end{cases}$$

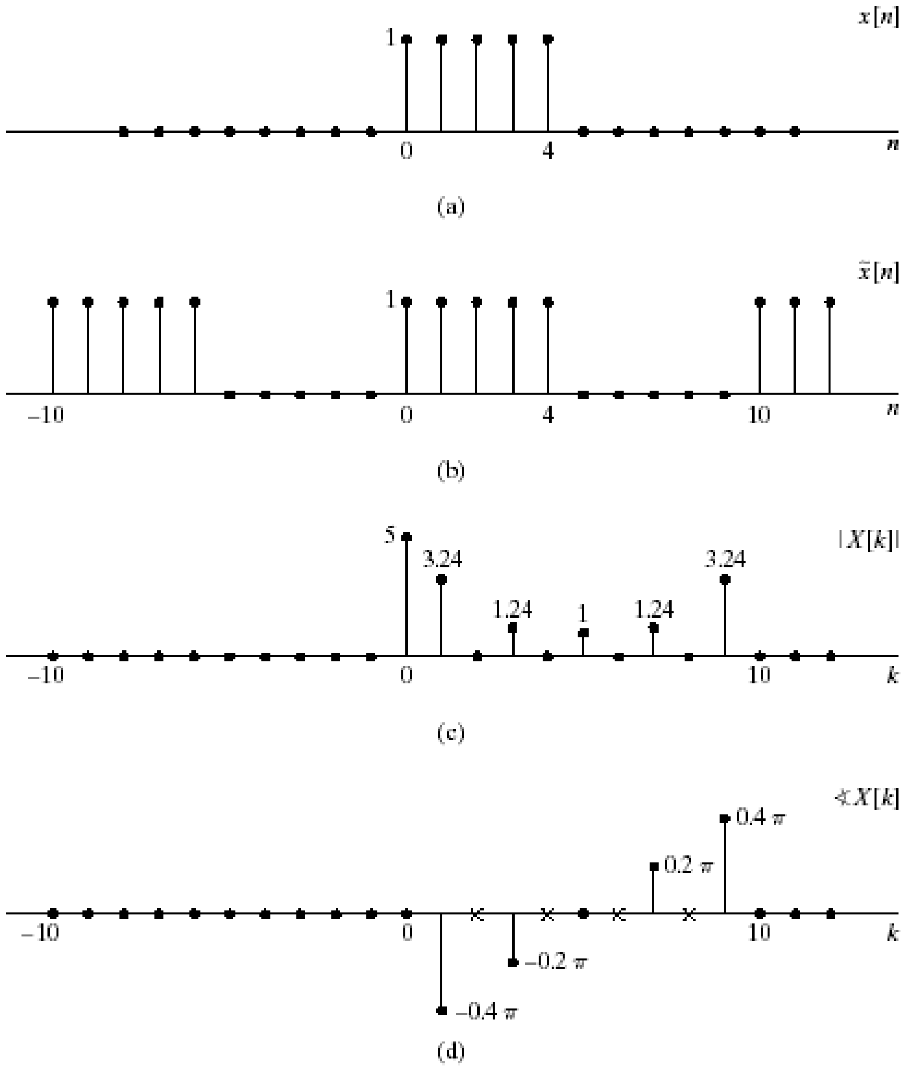


DFT: Example 1

- Let length of *x[n]* be *L=2N=10*
- We get a different set of DFT coefficients
- Still samples of the DTFT but in different places

➔ *x[n]* = Inverse *X[k]*

x[n] not unique but depends on relation *L* & *N*



DFT: Example 1; summary

: DFT of a rectangular pulse $x(n) = \begin{cases} 1 & 0 \le n \le 4 \\ 0 & \text{otherwise} \end{cases}$

$$X(e^{j\omega}) = \sum_{n=0}^{4} e^{-jn\omega} = \frac{1 - e^{-j5\omega}}{1 - e^{-j\omega}}$$

5-pt DFT of $x(n)$: $X(k) = X(e^{j\omega})\Big|_{\omega = \frac{2\pi}{5}k} = 5\,\delta(k) \quad 0 \le k \le 4$

10-pt DFT of $x(n)$:

$$X(k) = X(e^{j\omega})\Big|_{\omega = \frac{2\pi}{10}k} = \sum_{n=0}^{4} e^{-j\frac{2\pi}{10}nk} = \frac{1 - e^{-j\pi k}}{1 - e^{-j\frac{2\pi}{10}k}}$$

➔ The larger the DFT size L, the more details in the inverse DFT, i.e., *x[n]* can be seen

DFT: Example 2

Assume $x(n)=\left(\frac{1}{2}\right)^n u(n)$. $X(e^{jw})=\mathcal{F}[x(n)]$

Let $y(n)$ denote a 10-point sequence. i.e. $y(n)=0$ $n<0$ or $n \geq N$

Assume $Y(k)=X(e^{jw})\big|_{w=\frac{2\pi}{10}k}$, determine $y(n)$

$$y(n)=\sum_{r=-\infty}^{\infty} x(n+rN)=\sum_{r=0}^{\infty} x(n+10r)=\sum_{r=0}^{\infty}\left(\frac{1}{2}\right)^{n+10r}=\frac{1024}{1023}\left(\frac{1}{2}\right)^n \quad (0\leq n\leq 9)$$

$$0 \leq k \leq L-1, \; where \;\; L >= N$$

DFT: Example 3

Let $x(n)=0 \quad n<0$ or $n \geq N$

$$\tilde{X}(k) = X(e^{j\omega})\Big|_{\omega=\frac{2\pi}{64}k}, \quad \tilde{X}(k) = \begin{cases} 1 & k=32 \\ 0 & \text{otherwise} \end{cases}$$

(a) If $N=64$, find $x(n)$. Is $x(n)$ unique?

(b) If $N=3\times 64=192$, find $x(n)$. Is $x(n)$ unique?

Solution (a)

$$x(n) \overset{64\text{ pt DFT}}{\longleftrightarrow} X(k) = \tilde{X}(k)$$

$$\therefore x(n) = \frac{1}{64}\sum_{k=0}^{63} X(k)\, e^{j\frac{2\pi}{64}kn} = \frac{1}{64}e^{j\pi n} = \begin{cases} \frac{1}{64} & n \text{ even} \\ -\frac{1}{64} & n \text{ odd} \end{cases}$$

$$0 \le k \le L-1, \ \ where \ \ L >= N$$

DFT: Example 3

(b). Let $x_1(n)$ be of length 64, which corresponds to $\tilde{X}(k)$

Then $x_1(n) = \text{IDFT}[X(k)] = \begin{cases} \frac{1}{64} & n \text{ even} \\ -\frac{1}{64} & n \text{ odd} \end{cases}$

Note that $x_1(n) = \sum_{r=0}^{2} x(n+64r)$

Therefore $x(n)$ is not unique in order to get $\tilde{X}(k)$

For example,

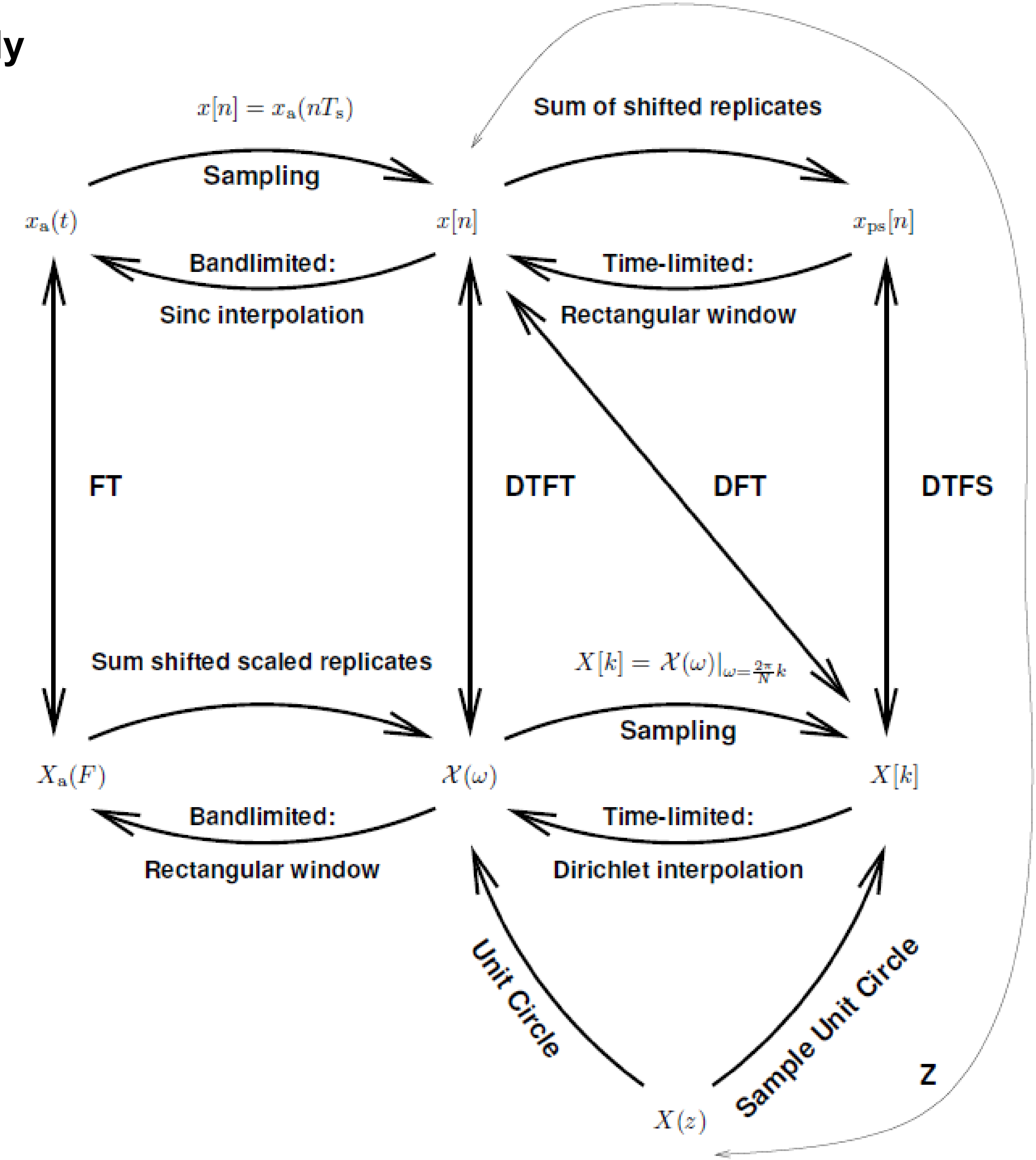
1st choice, $x(n) = \begin{cases} x_1(n) & 0 \leq n \leq 63 \\ 0 & 64 \leq n \leq 191 \end{cases}$

2nd choice,

$x(n) = \frac{1}{3}[x_1(n) + x(n+64) + x(n+128)]$

$0 \leq n \leq 63$

The FT Family



Discrete Fourier Transform (DFT)

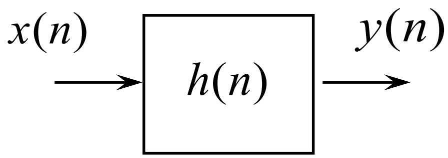
1. DFT:
    - Fourier Transform of short duration signals
2. DFT: Sampling of the DTFT
3. Convolution with DTF
4. DFT of long signals
    - The effect of windowing
5. The DFT as a Linear Transform
    - DFT as a vector-matrix operation
6. FFT

Based on:
- Chapter 8, A.V. Oppenheim and R.W. Schafer, *Discrete-Time Signal Processing*, Prentice-Hall, 3rd ed, 2010.
- Slides from http://faculty.nps.edu/rcristi/

Convolution of Finite Sequences

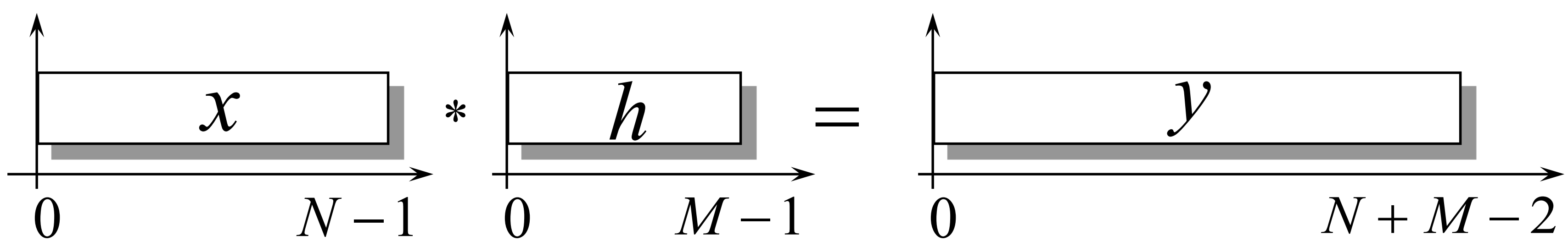
- <u>Linear convolution</u> *y[n]=x[n]*h[n]*

➔When *x[n]* and *h[n]* are finite sequences, the duration of *y[n] is N+M-2*

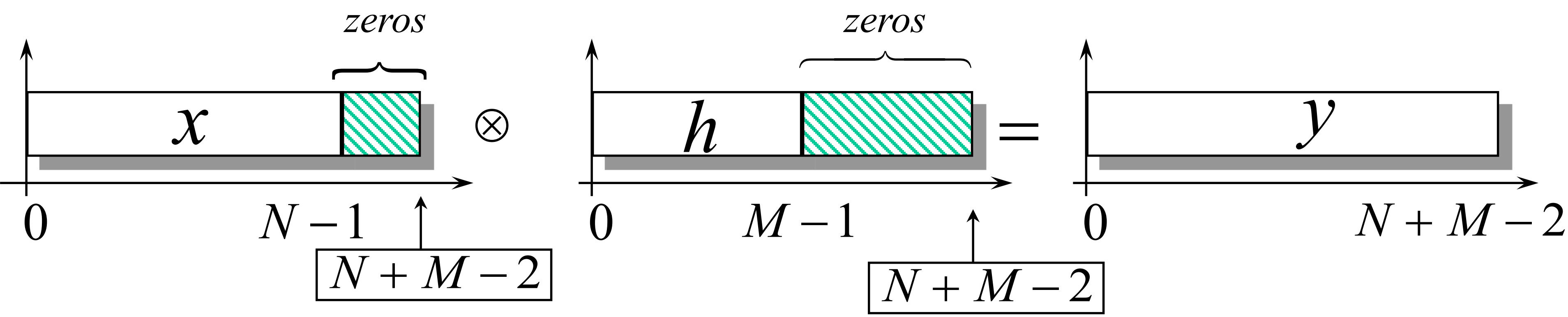


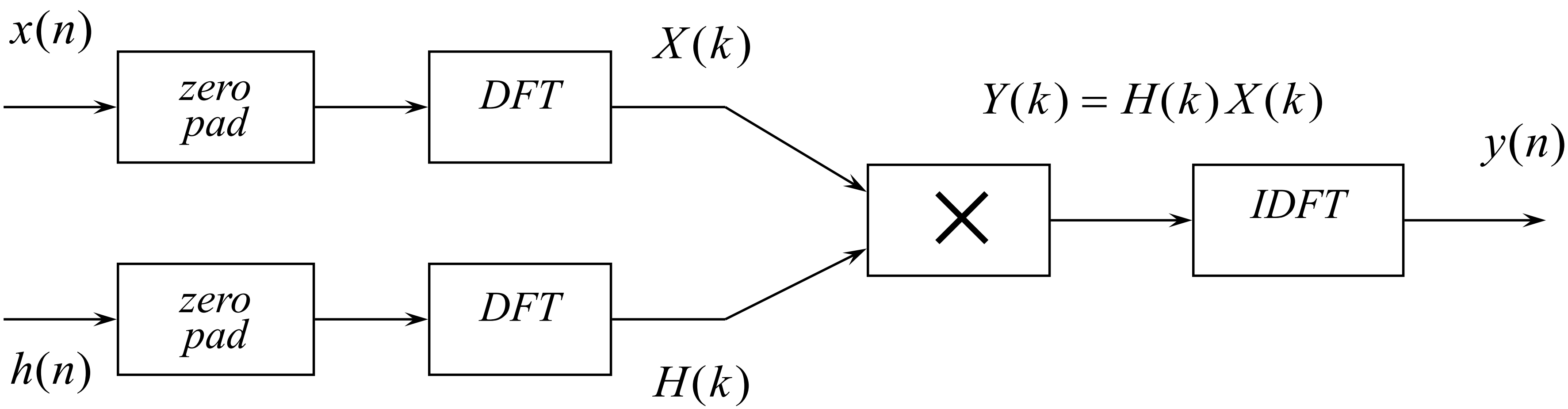
$$y(n) = \sum_{k=0}^{N-1} x(k)\underbrace{h(n-k)}$$

$$\boxed{\begin{array}{c} \textit{zero when} \\ n-k > M-1 \textit{ for all } k, \\ i.e. \quad n > \underbrace{N-1}_{\max k} + M-1 \end{array}}$$



- To have all sequences of the same length, we pad them with zeros➔use circular convolution





Convolution of Finite Sequences

- Linear Convolution: $$y(n) = \sum_{k=0}^{N-1} x(k)\underbrace{h(n-k)}$$
  - $x[n]$ is multiplied by a time–reversed and linearly-shifted $h[n]$

  ➔ Length of $y[n]$ is N+M-2
  - We can use DFT to compute linear convolution
  - If you do not use a sufficient number of points in the DFT, you will get overlap (aliasing)
- N-point circulation convolution: $$s[n] = x[n] \; Ⓝ \; h[n] = \sum_{m=0}^{N-1} x[m]\, h[n - m \bmod N]$$
  - $x[n]$ is multiplied by circularly time-reversed and circularly-shifted $h[n]$
  - Since DFT is a limited length sequence, convolution is done modulo N, i.e., when we flip and shift the sequence, we do it mod N ➔Length of $y[n]$ is N
- Circular convolution $s[n]$ equals linear convolution $y[n]$ plus time aliasing
  - If $H[k]$ and $X[k]$ are sampled adequately from their respective DTFT, then $S[k]$ are samples from $Y(w)$, the DTFT of $y[n]$, and hence $s[n]$ will be the N-point periodic superposition of $y[n]$, the inverse DTFT of $Y(w)$

  ➔$s[n]$ is a time-domain aliased version of $y[n]$; a sum of N-point shifted replicates of y[n] $$s[n] = \sum_{l=-\infty}^{\infty} y[n - lN] = \sum_{l=-\infty}^{\infty} (x * h)[n - lN]$$

Convolution of Finite Sequences

- Two ways to calculate N -point circular convolution
  1. Using DFT: $y[n]$=IDFT { $Y[k]=X[k]H[k]$ }
     - We need only values for $n = 0, \ldots, N-1$
  2. Using $$x[n] \; Ⓝ \; h[n] = \sum_{m=0}^{N-1} x[m]\, h[n - m \bmod N]$$

a) Take *one* of the two sequences, e.g., $h[n]$, and form its N -point circular extension $h[n \bmod N]$

b) Perform ordinary convolution of that extended signal $h[n \bmod N]$ with the time-limited signal $x[n]$

c) We need only bother to compute the results for $n = 0, \ldots, N-1$

Convolution of Finite Sequences

Example

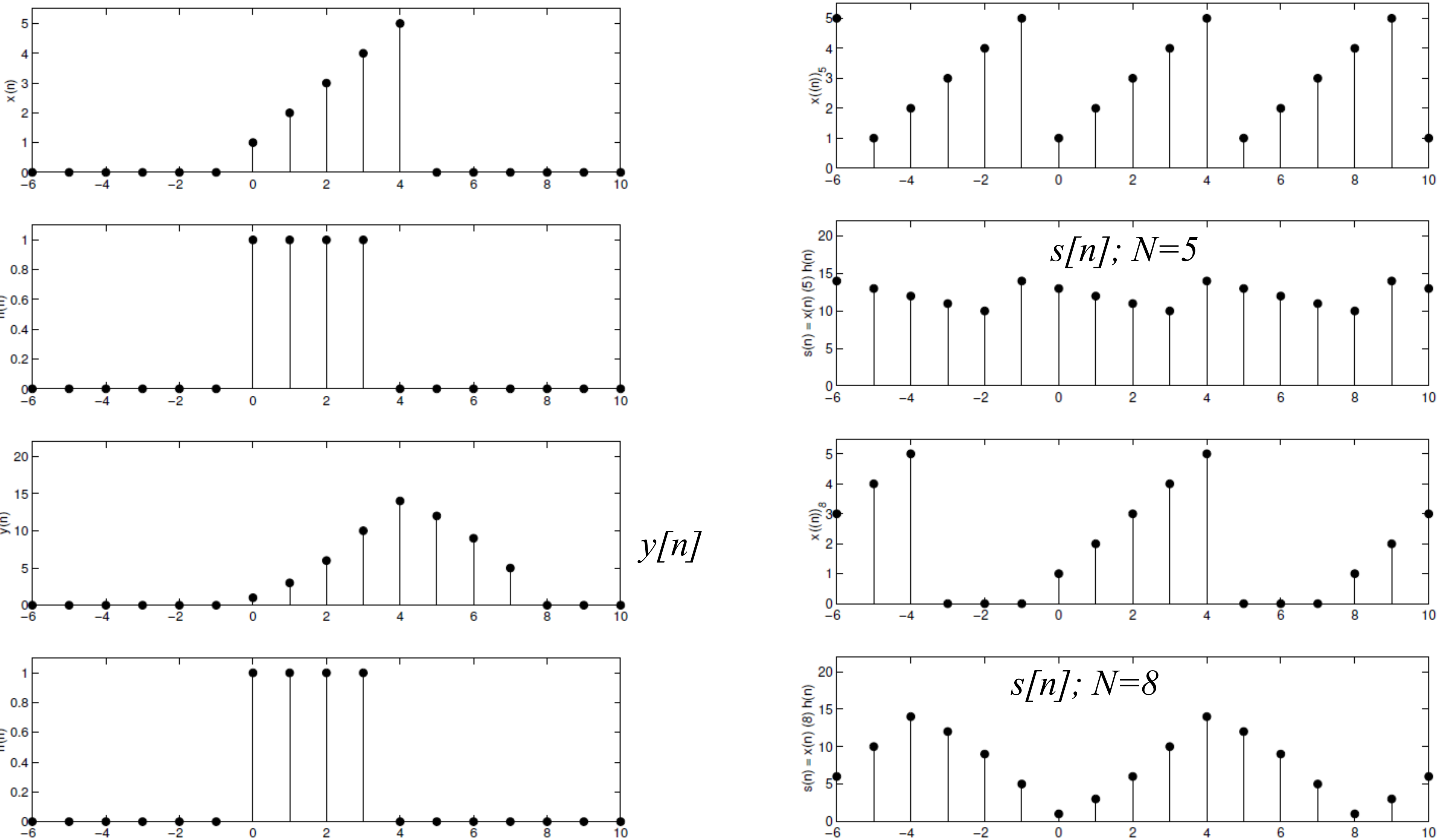
- Given $x[n] = \{\underline{2}, 0, 3, -1\}$, $h[n] = \{\underline{10}, 20, 30, 40\}$

a) Find their 4-point circular convolution
   - s = ifft(fft([2 0 3 -1]) .* fft([10 20 30 40]))
   - $s[n] = \{\underline{90}, 130, 50, 130\}$
   - Since $s[n]$ is the IDFT of $S[k] = H[k]\, X[k]$, it is periodic with period $N = 4$

b) Find their linear convolution
   - Compute using: y = conv([2 0 3 -1], [10 20 30 40])
   - $y[n] = h[n] * x[n] = \{20; 40; 90; 130; 70; 90; -40\}$
   - The result has N+M-1=4+4-1=7 nonzero values
   - All other values are zero

→ *s[n]* not equal *y[n]*, i.e., 4-point circular convolution <u>did not</u> result in the linear convolution

– Solution: make $x[n]$ and $h[n]$ of length N+M-2 by zero-padding


x(n)
y(n)
y[n]
x((n))5
s[n]; N=5
s(n) = x(n) (5) h(n)
x((n))8
s[n]; N=8
s(n) = x(n) (8) h(n)

Circular Convolution: Example 1

$$x_1[n] = \delta[n - n_0]$$

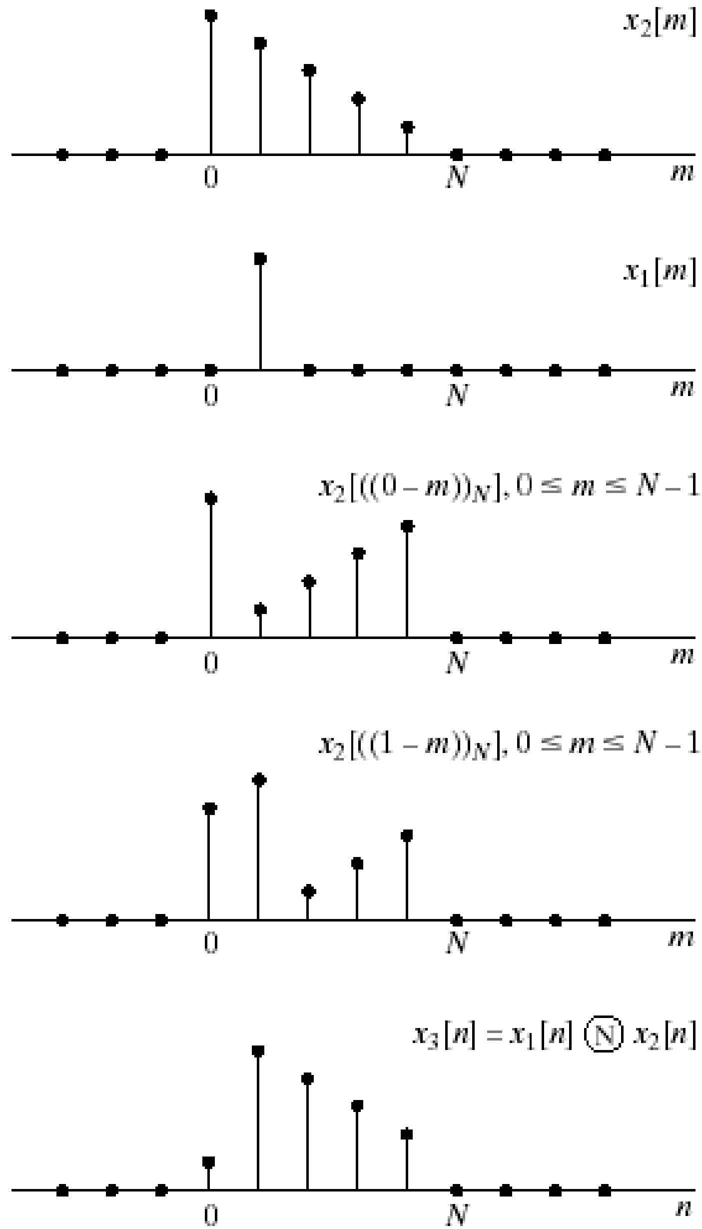
$$X_1[k] = W_N^{kn_0}$$

$$X_3[k] = W_N^{kn_0} X_2[k]$$

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

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

➔ Since x1 is just a shifted impulse, the circular convolution coincides with a circular shift of x2 by on point



Circular Convolution: Example 2a

- Two rect. x[n]: N=6

$$x_1[n] = x_2[n] = \begin{cases} 1 & 0 \le n \le K-1 \\ 0 & else \end{cases}$$

- DFT of each sequence L=N=6

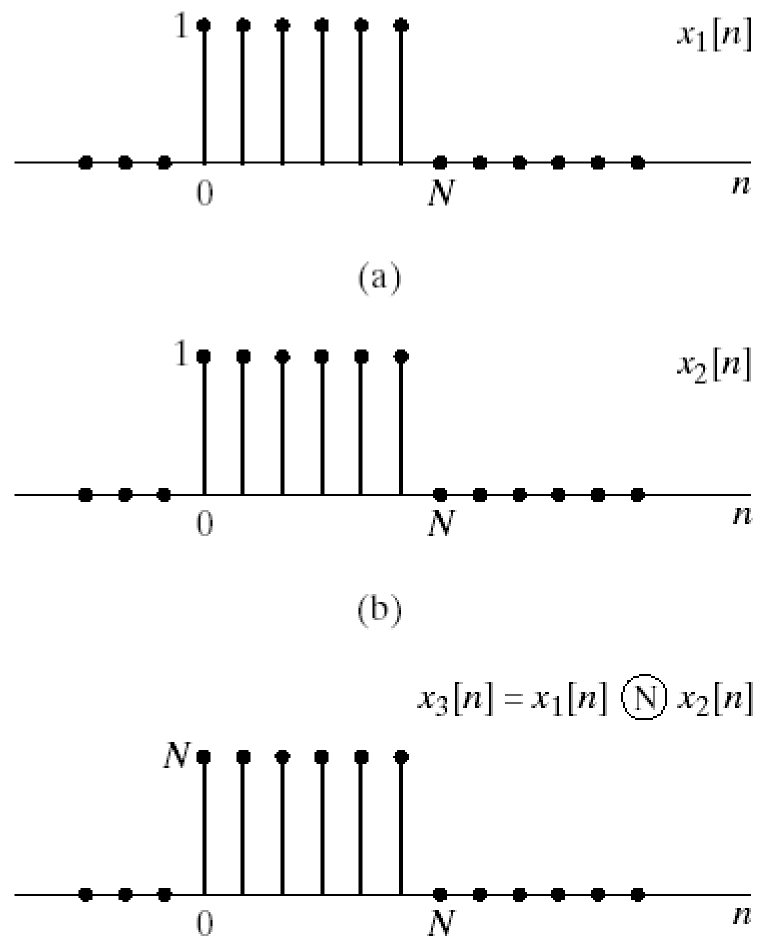
$$X_1[k] = X_2[k] = \sum_{n=0}^{N-1} e^{-j\frac{2\pi}{N}kn} = \begin{cases} N & k=0 \\ 0 & else \end{cases}$$

- Multiplication of DFTs

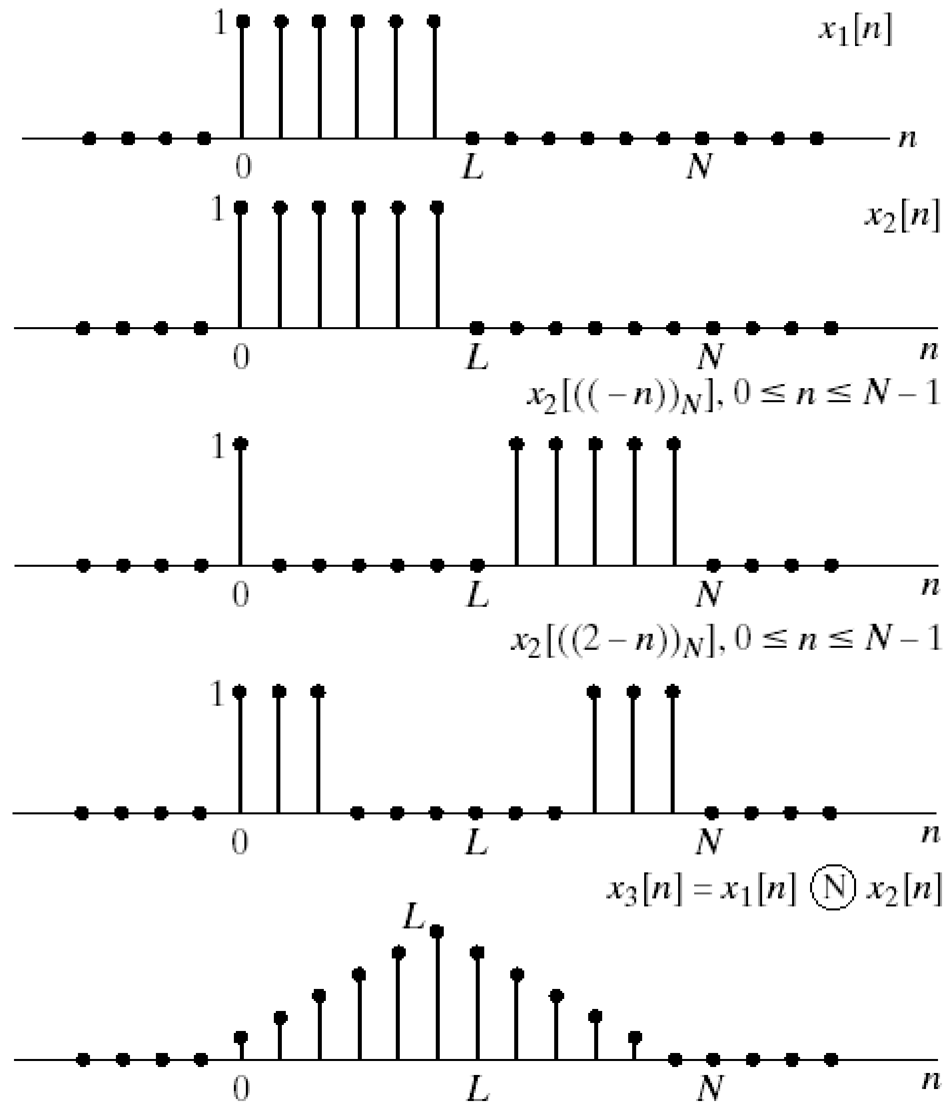
$$X_3[k] = X_1[k] X_2[k] = \begin{cases} N^2 & k=0 \\ 0 & else \end{cases}$$

- Inverse DFT

$$x_3[n] = \begin{cases} N & 0 \le n \le N-1 \\ 0 & else \end{cases}$$



Circular Convolution: Example 2b



- Augment zeros to each sequence ➔N=2L
- The DFT of each sequence L<N

(solve using finite series formula)

$$X_1[k] = X_2[k] = \frac{1 - e^{-j\frac{2\pi Lk}{N}}}{1 - e^{-j\frac{2\pi k}{N}}}$$

- Multiplication of DFTs

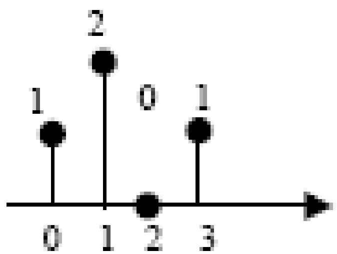
$$X_3[k] = \left(\frac{1 - e^{-j\frac{2\pi Lk}{N}}}{1 - e^{-j\frac{2\pi k}{N}}}\right)^2$$

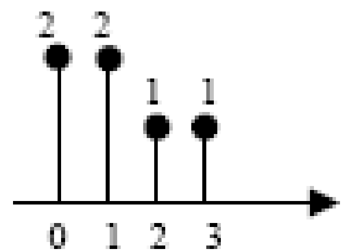
$$x_3[n] = \sum_{m=0}^{N-1} x_1[m] x_2[((n-m))_N]$$

➔ **Inverse DFT is not unique; depends on relation L & N**

Circular convolution: Example 3

$$g[n] = \{1,2,0,1\} \qquad and \qquad h[n] = \{2,2,1,1\}$$





From the definition of the circular convolution:

$$y_c[n] = g[n] \textcircled{4} h[n] = \sum_{m=0}^{3} g[m]h\left[\langle n-m \rangle_N\right] \qquad 0 \le n \le 3$$

Therefore:

$$y_c[0] = \sum_{m=0}^{3} g[m]h\left[\langle -m \rangle_N\right] \qquad 0 \le n \le 3$$

The circular time-reversed sequence $h[\langle -m\rangle_4]$ is as shown below:

$n = 0$	$n = 1$	$n = 2$	$n = 3$
$h[\langle -m\rangle_4]$	$h[\langle 1-m\rangle_4]$	$h[\langle 2-m\rangle_4]$	$h[\langle 3-m$

By performing the product of g[m] with $h[\langle -m\rangle_4]$ for each value of m in the range $0 \le m \le$
And summing the products we get:

$$y_C[0] = g[0]\cdot h[0] + g[1]\cdot h[3] + g[2]\cdot h[2] + g[3]\cdot h[1]$$
$$= (1\times 2) + (2\times 1) + (0\times 1) + (1\times 2) = 6$$

$$y_C[1] = \sum g[m] h[\langle 1-m\rangle_4]$$

$$y_C[1] = g[0]h[1] + g[1]\cdot h[0] + g[2]\cdot h3 + g[3]\cdot h[2]$$
$$= (1\times 2) + (2\times 2) + (0\times 1) + (1\times 1) = 7$$

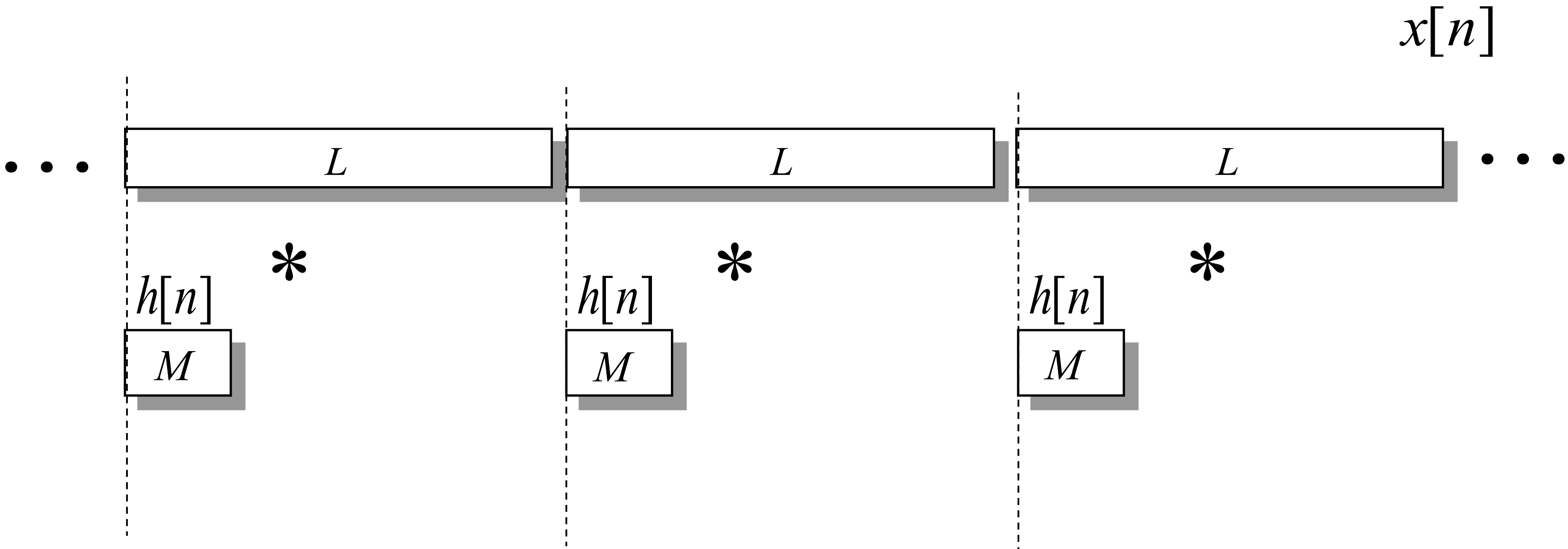
$$y_C[2] = g[0]h[2] + g[1]\cdot h[1] + g[2]\cdot h[0] + g[3]\cdot h[3]$$
$$= (1\times 1) + (2\times 2) + (0\times 2) + (1\times 1) = 6$$

$$y_C[3] = g[0]h[3] + g[1]\cdot h[2] + g[2]\cdot h[1] + g[3]\cdot h[0]$$
$$= (1\times 1) + (2\times 1) + (0\times 2) + (1\times 2) = 5$$

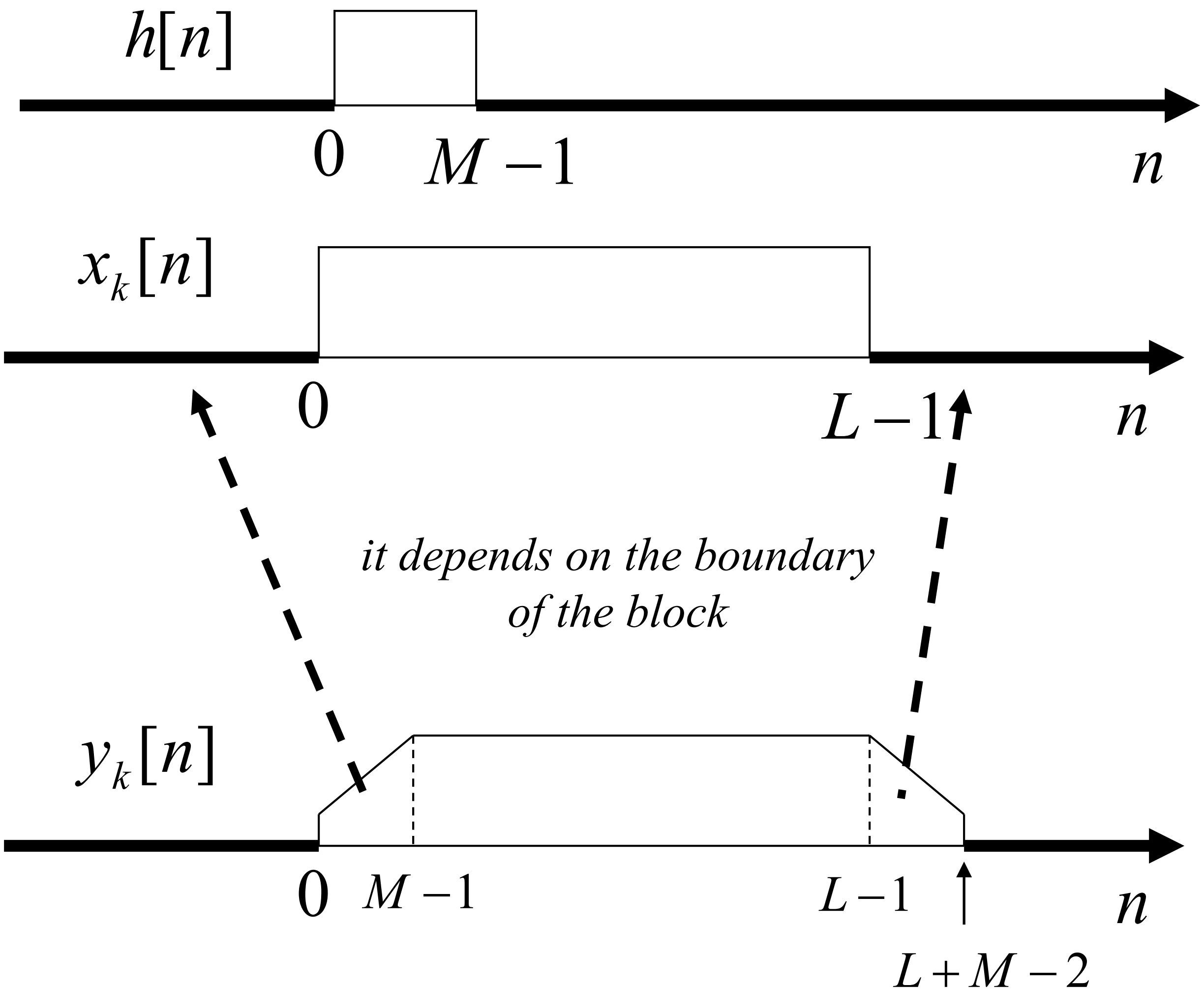
<u>Ans</u>: $y_C[N] = \{6,7,6,5\}$

Convolution of Long Data Streams

- <u>Problem</u>: in general the input signal *x[n]* is much longer than the <u>impulse response</u>
- Example: music CD: 4001x 0^6 samples!
- Since Convolution is LTI, we can do <u>block convolution</u> by subdividing *x[n]* into smaller sections L
    - o Then we can use circular convolution (instead of linear convolution) to compute each section since FFT is highly efficient to compute circular convolutions



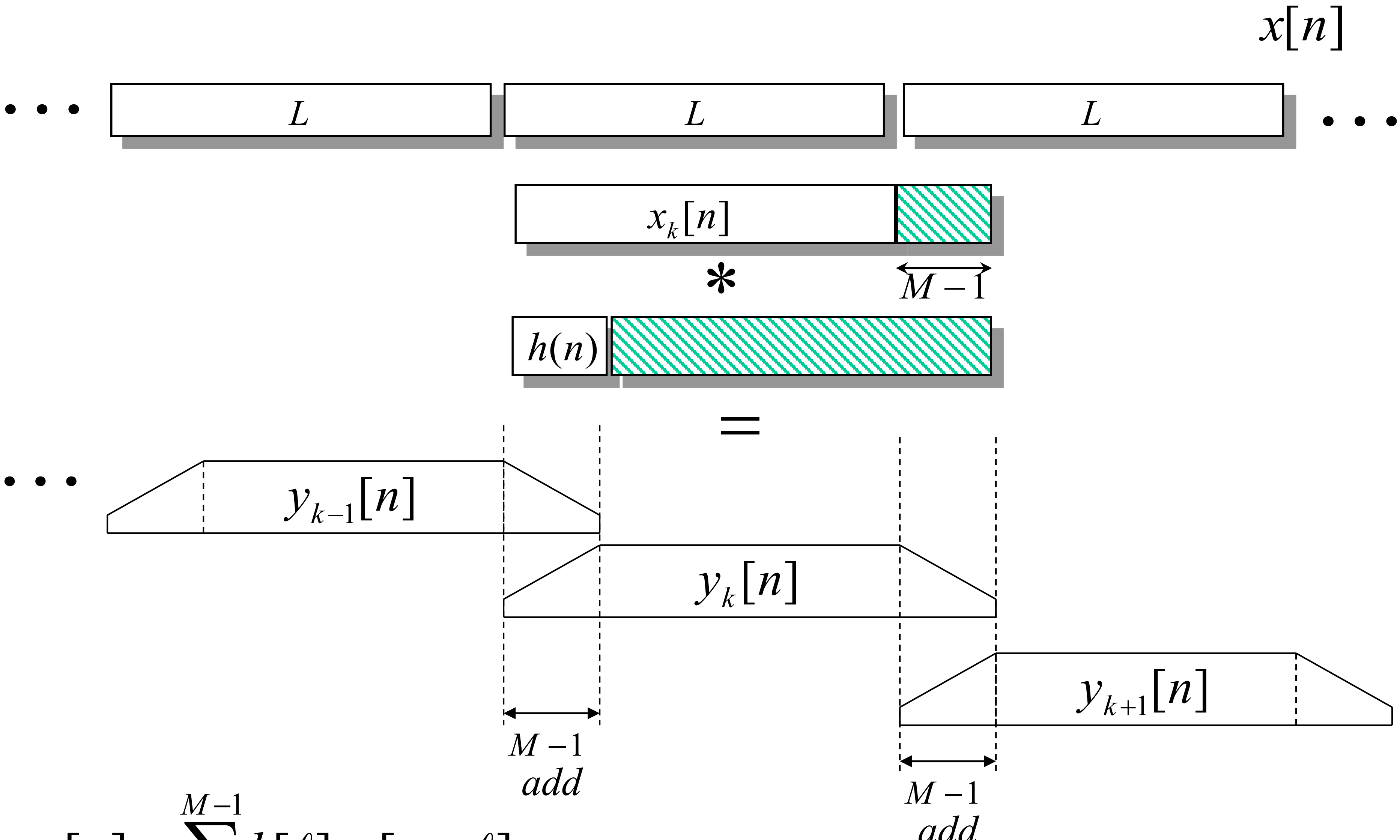
- The convolution of every block by itself



$$y_k[n] = \sum_{\ell=0}^{M-1} h[\ell] x_k[n-\ell]$$

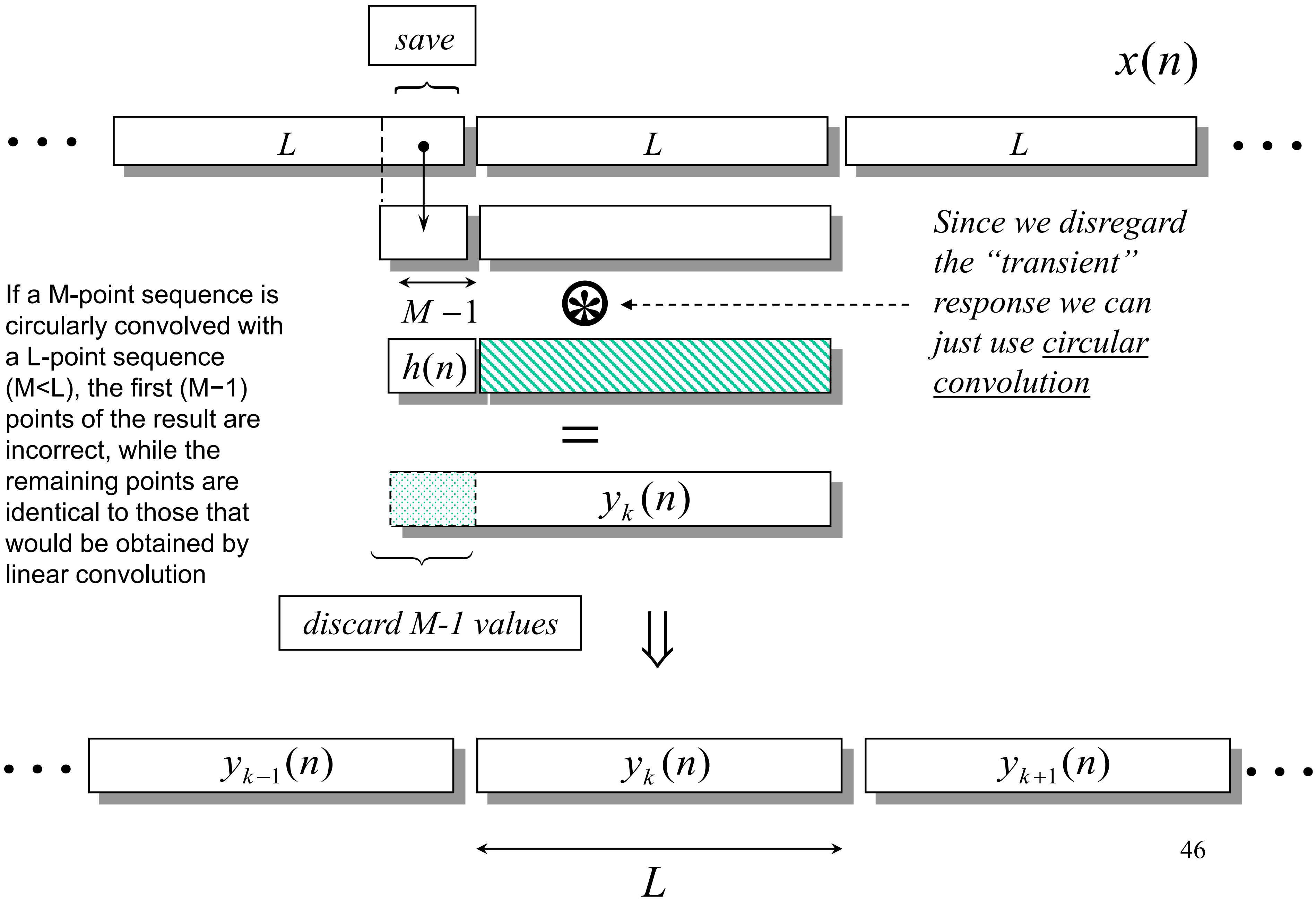
- There are 2 methods to perform block convolution: Overlap & Add and Overlap & Save

1. Overlap and Add: Convolve each section and add the "tail" to the next section



$$y_k[n] = \sum_{\ell=0}^{M-1} h[\ell] x_k[n-\ell]$$

2. Overlap and Save: Separate *x[n]* into overlapping sections of length L, so that each section overlaps the preceding section by (M−1) points

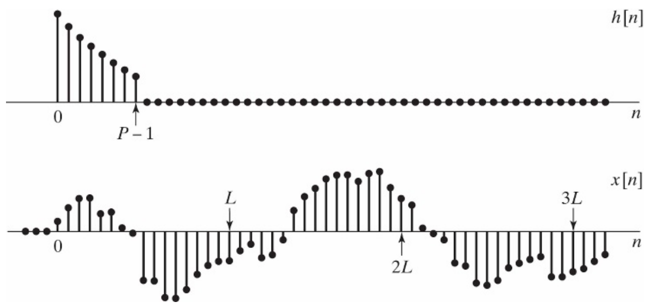


If a M-point sequence is circularly convolved with a L-point sequence (M<L), the first (M−1) points of the result are incorrect, while the remaining points are identical to those that would be obtained by linear convolution

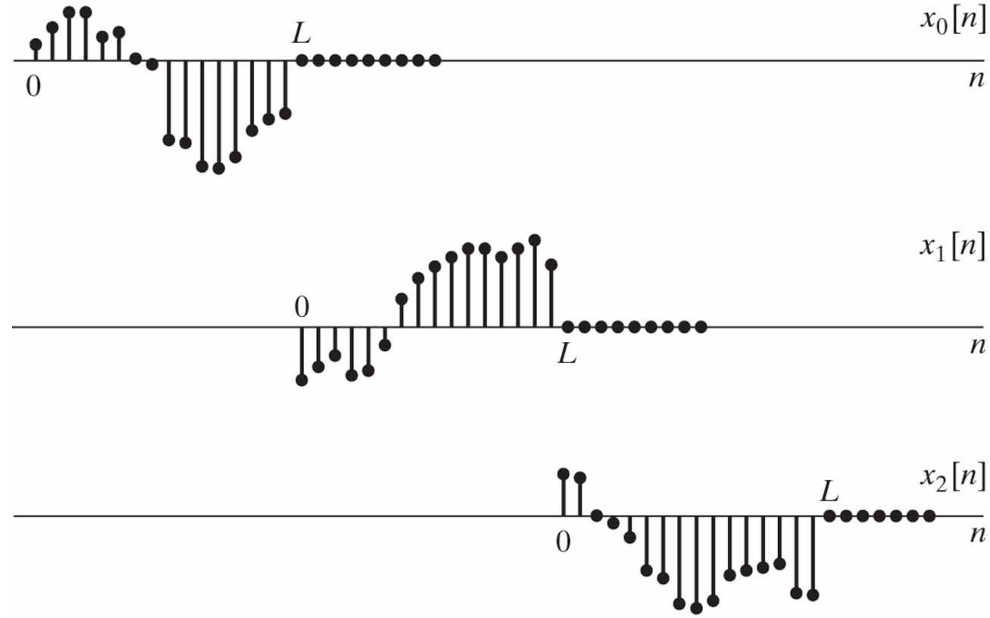
Since we disregard the "transient" response we can just use circular convolution

Convolution of Long Data Streams

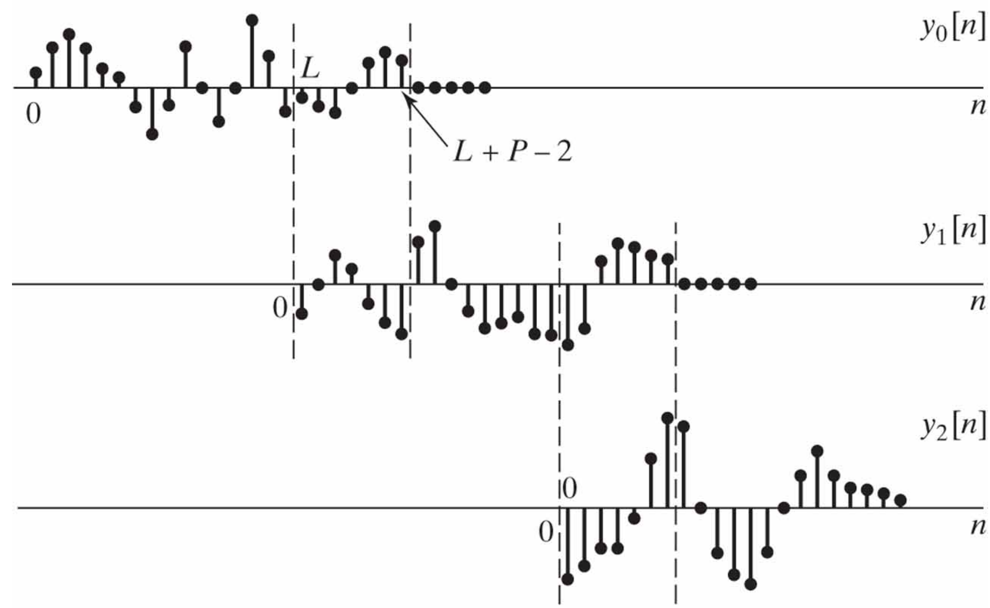
Example: Overlap and Add



Finite-length impulse response $h[n]$ and indefinite-length signal $x[n]$



(a)



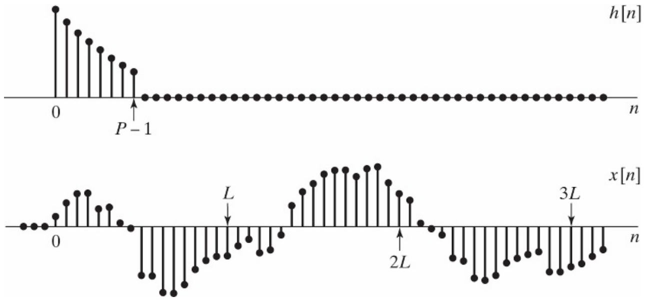
(b)

(a) Decomposition of $x[n]$ into non-overlapping sections of length L

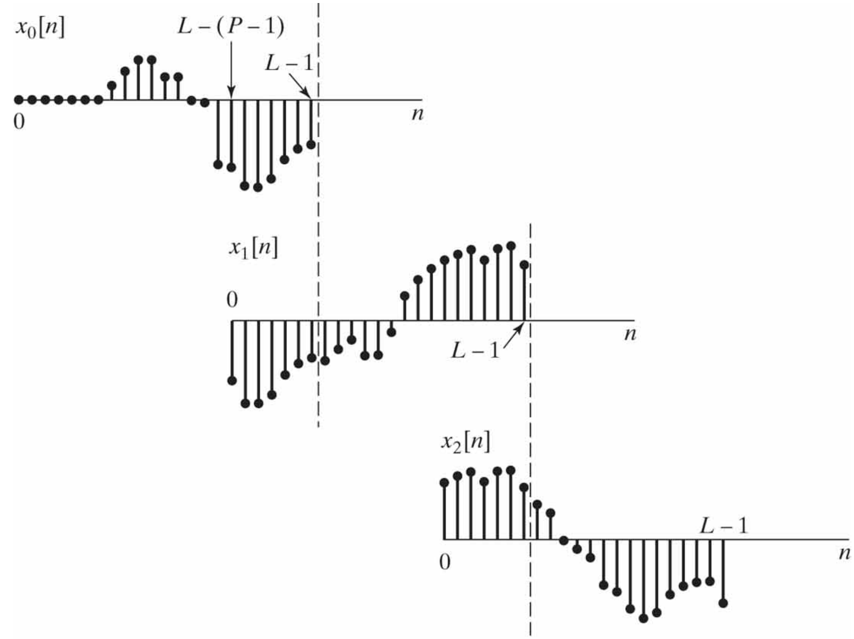
(b) Result of convolving each section with $h[n]$

Convolution of Long Data Streams

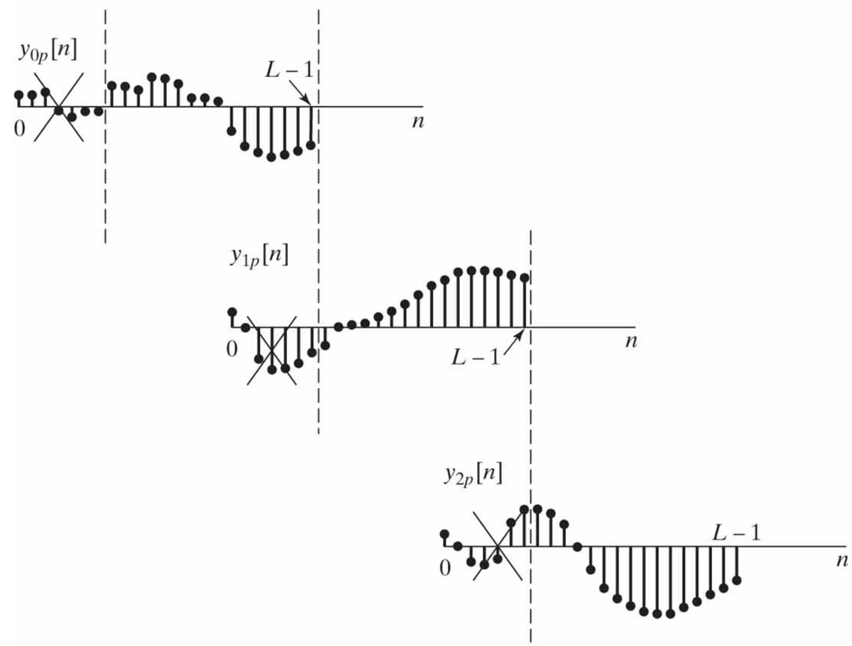
Example: Overlap and Save



Finite-length impulse response $h[n]$ and indefinite-length signal $x[n]$



(a)



(b)

(a) Decomposition of $x[n]$ into overlapping sections of length L
(b) Result of convolving each section with $h[n]$

X indicates the portion of each filtered section to be discarded in forming the linear convolution

Discrete Fourier Transform (DFT)

1. DFT:
    - Fourier Transform of short duration signals
2. DFT: Sampling of the DTFT
3. Convolution with DTF
4. DFT of long signals
    - The effect of windowing
5. The DFT as a Linear Transform
    - DFT as a vector-matrix operation
6. FFT

Based on:

- Chapter 8, A.V. Oppenheim and R.W. Schafer, *Discrete-Time Signal Processing*, Prentice-Hall, 3rd ed, 2010.
- Slides from http://faculty.nps.edu/rcristi/

DFT of long signals

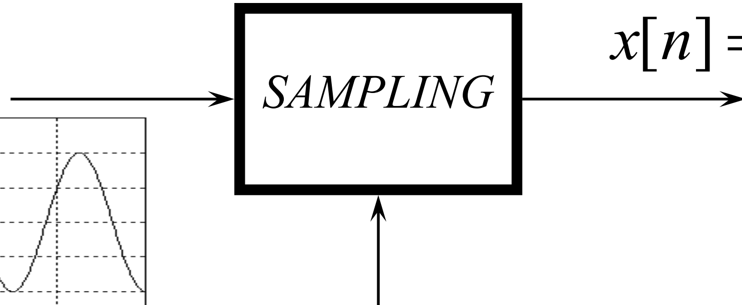
The effect of <u>windowing</u> on the DTFT

1. The original signal $x(t)$ is digitized to $x[n]$
2. Real signals are not short in duration (not time-limited; not finite duration)
3. But the DFT is applied to a finite duration $x[n]$
4. The DFT yields N samples of the DTFT at equally spaced intervals $2\pi/N$
5. For a signal that is very long, e.g., a speech signal or a music piece, the DFT is calculated over successive overlapping short intervals

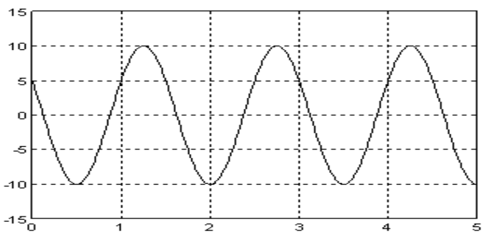
The effect of <u>windowing</u> on the DTFT

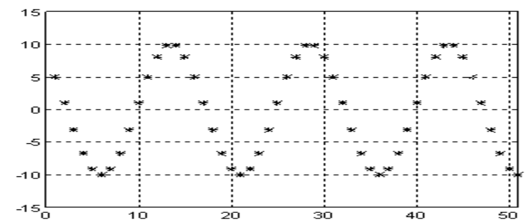
CTFT → DTFT

$x(t)$



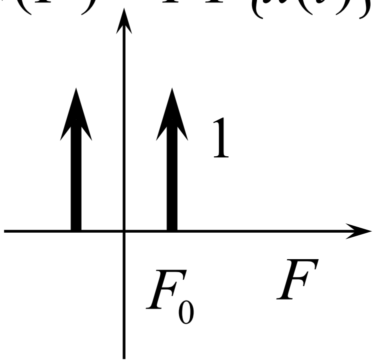
$x[n] = x(nT_s)$



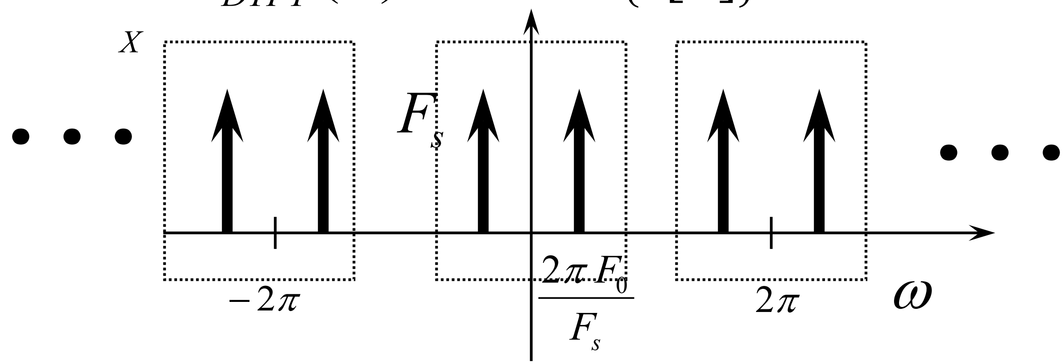


$F_s = 1/T_s$

$X_{FT}(F) = FT\{x(t)\}$



$X_{DTFT}(\omega) = DTFT\{x[n]\}$

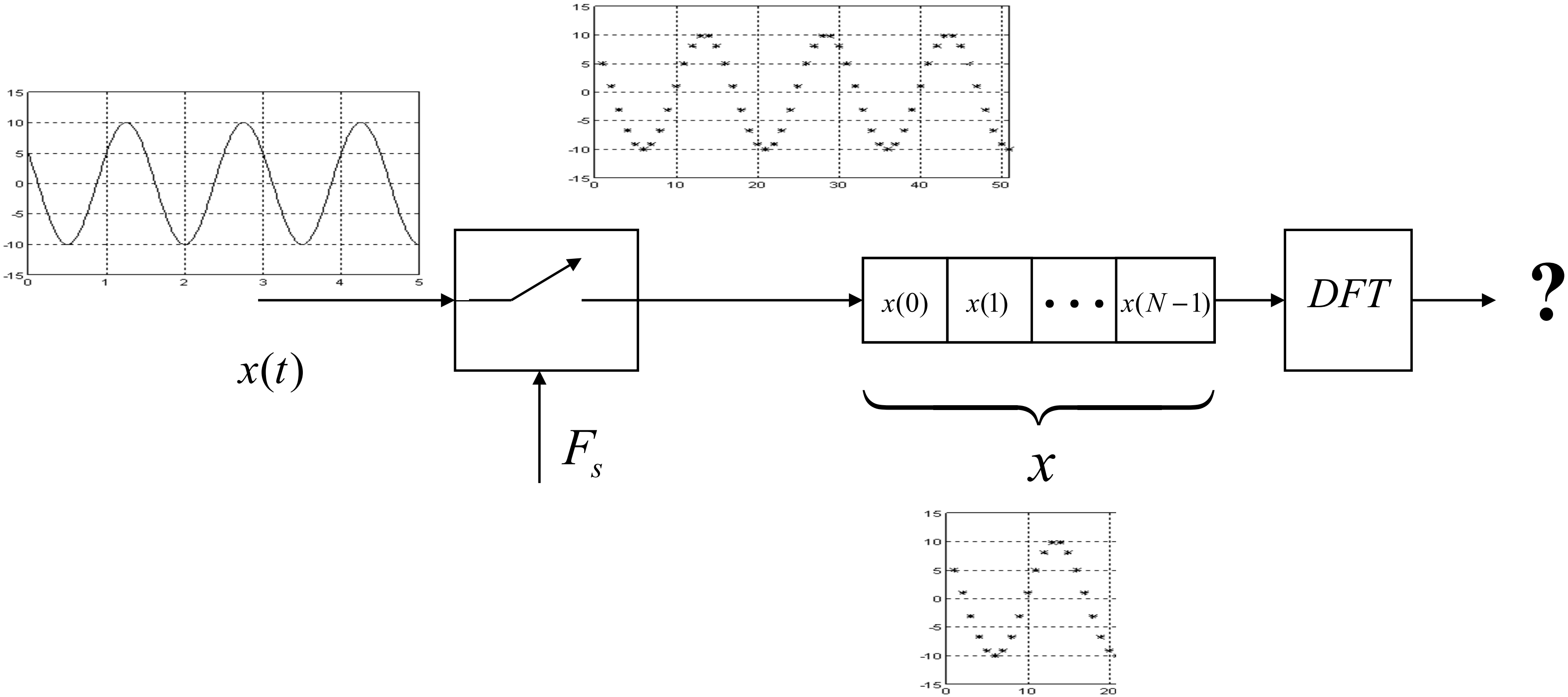


It can be shown that $X_{FT}(F)$ and $X_{DTFT}(\omega)$ are related as

$$X_{DTFT}(\omega)\Big|_{\omega = 2\pi F/F_s} = F_s \sum_{k=-\infty}^{+\infty} X_{FT}(F - kF_s)$$

The effect of <u>windowing</u> on the DTFT

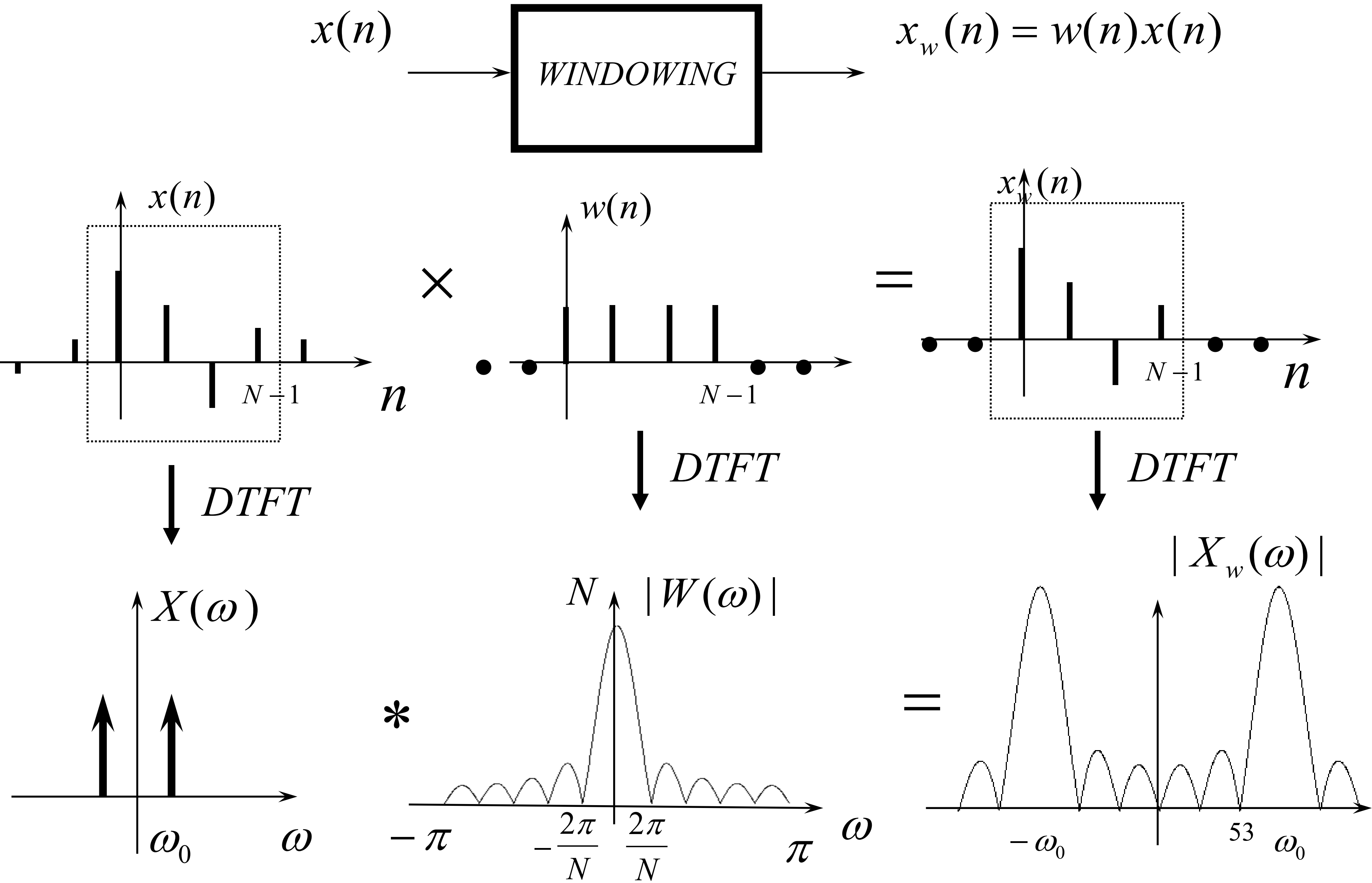
Consider a Sinusoid:



we sample **...** and take a finite set of samples (**<u>window</u>**) ...

<u>Problem</u>: how the DFT is going to look like?

The effect of windowing on the DTFT



Then we can relate DFT and DTFT as follows:

$$X(k)=\sum_{n=0}^{N-1} x(n)e^{-j\omega n}\Bigg|_{\omega=2\pi k/N} = \sum_{n=-\infty}^{+\infty} w(n)x(n)e^{-j\omega n}\Bigg|_{\omega=2\pi k/N}$$

$$DFT\{x(n)\} \quad = \quad samples\ of \quad DTFT\{w(n)x(n)\}$$

In formulas:

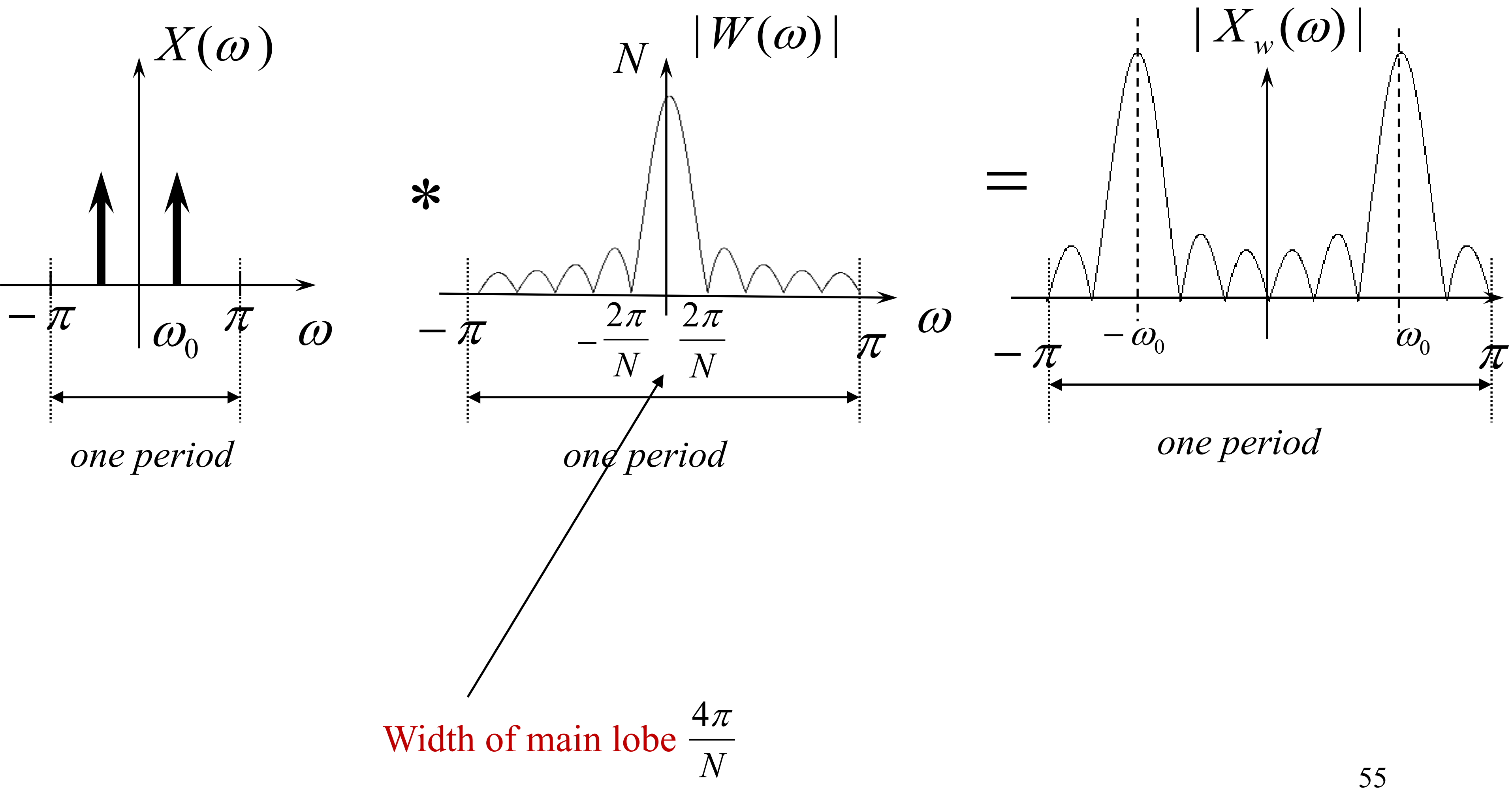
$$\boxed{X_{DFT}(k)=\frac{1}{2\pi}X_{DTFT}(\omega)*W(\omega)\Big|_{\omega=2\pi k/N}}$$

with: $W(\omega)=DTFT\{w(n)\}$

$$=\sum_{n=-\infty}^{+\infty} w(n)e^{-j\omega n}=\sum_{n=0}^{N-1} e^{-j\omega n}=e^{-j\omega(N-1)/2}\frac{\sin(\omega N/2)}{\sin(\omega/2)}$$

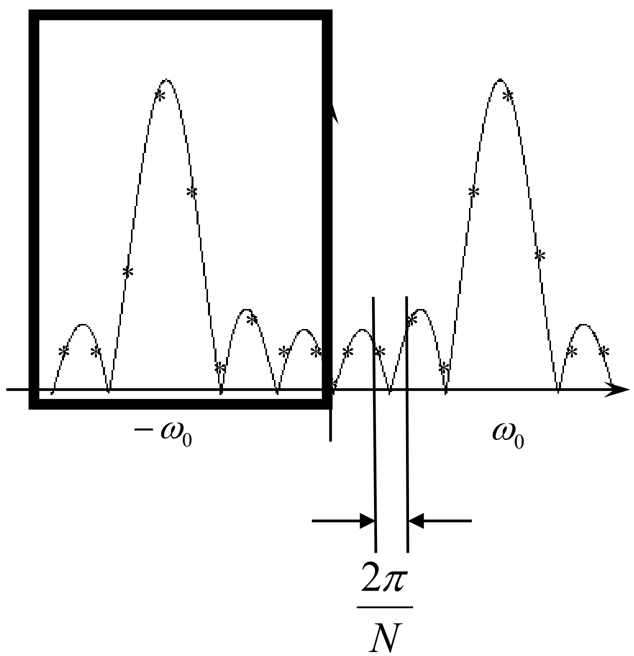
- So if we have a sampled sinusoid at frequency ω_0, the DFT of a finite length of sequence looks like this:

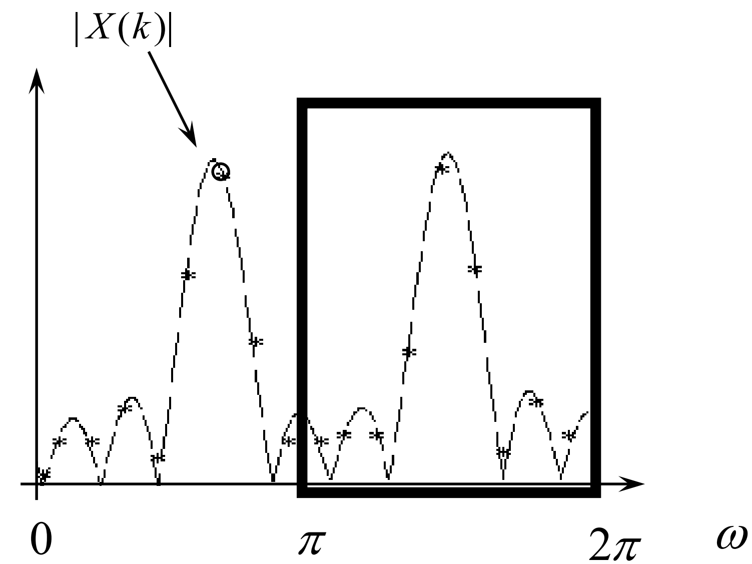
➢ First we look at the DTFT of the windowed sequence:



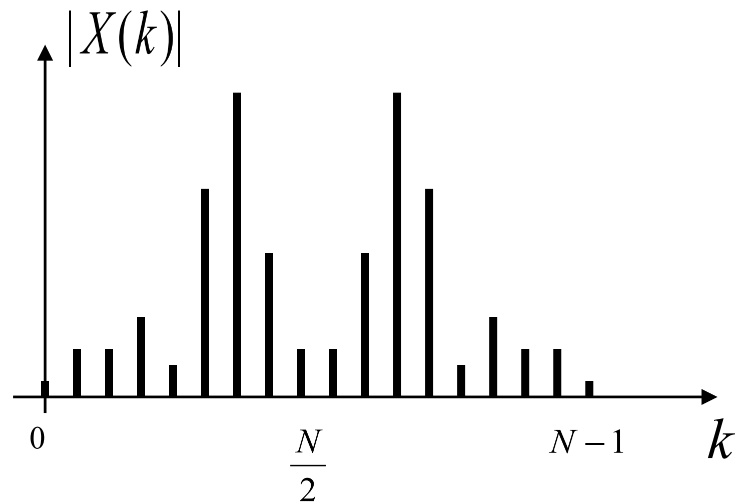
- Then we take samples of the DTFT as

$$X_{DFT}(k) = X_{windowed}(\omega)\Big|_{\omega = k2\pi/N} \qquad \textit{For } k=0,\ldots,N\text{-}1$$





$$X(k) = DFT\{[x(0),\ldots,x(N-1)]\}$$

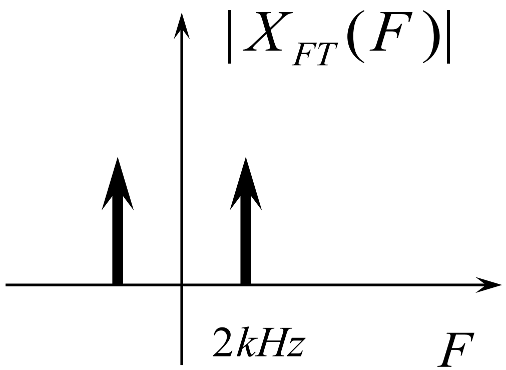


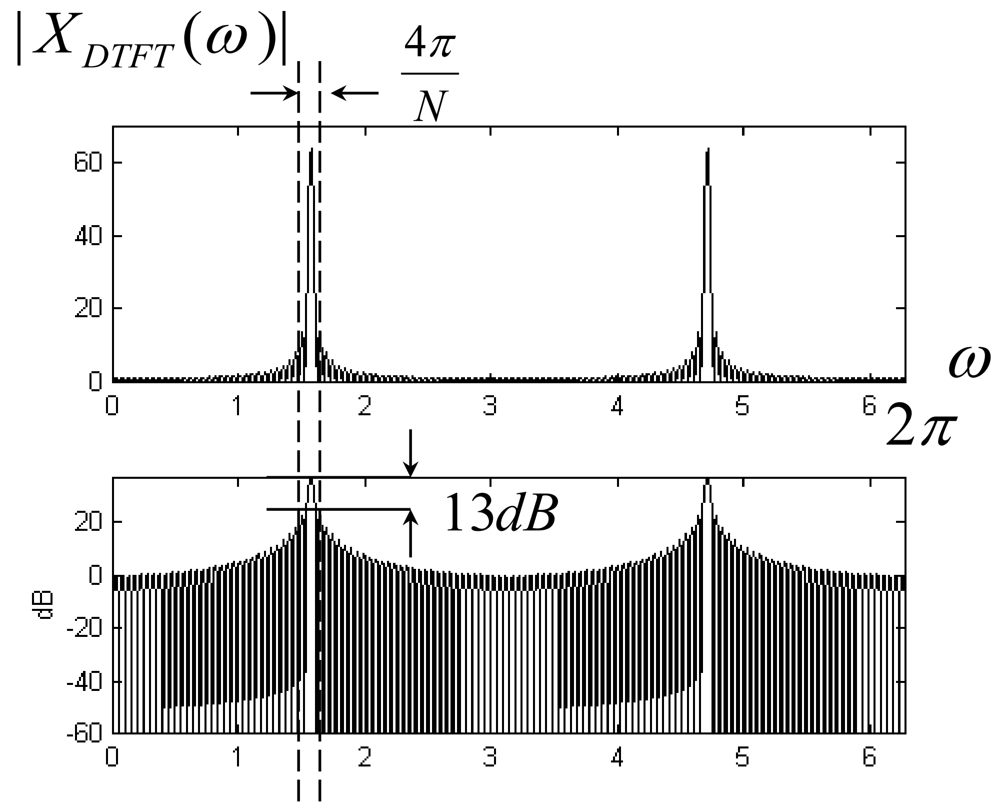
Consequence: when we take an N-point DFT of a sinusoid, we have two effects:

- *<u>Loss of Resolution</u>*: the sinusoid is not exactly localized in frequency;
- *<u>Sidelobes</u>*: other frequencies (artifacts) appear.

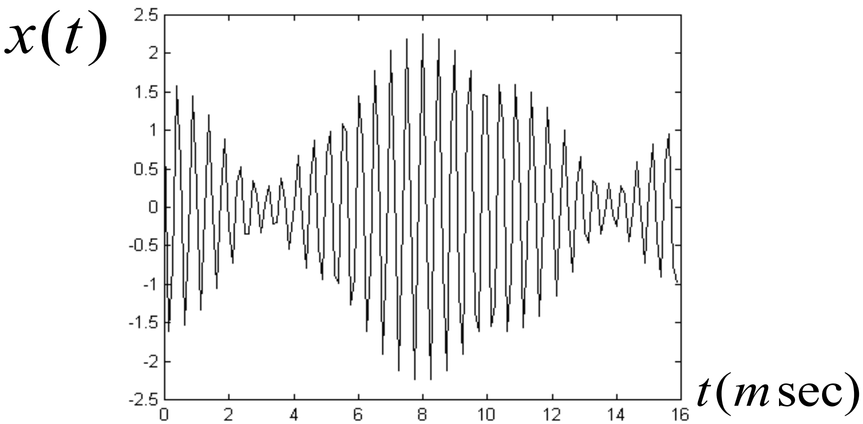
Example: consider a sinusoid of frequency 2kHz, sampled at 8kHz.

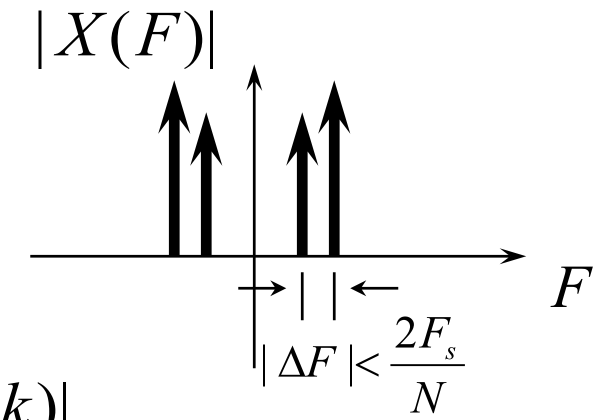
Take N=128 points.



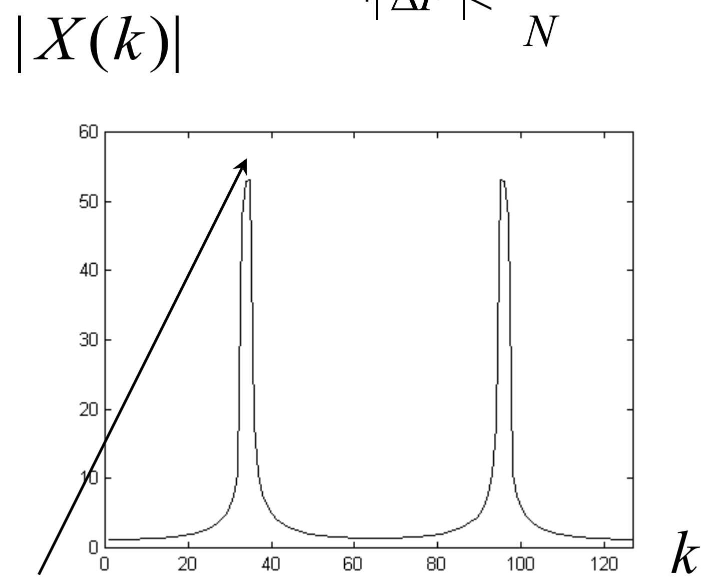


- We have to be careful when we apply the DFT, since we can miss some frequency components
- Example 1: 2 frequency components F_0 and $F_0 + \Delta F$ too close to each other, i.e., $|\Delta F| < \frac{2F_s}{N}$





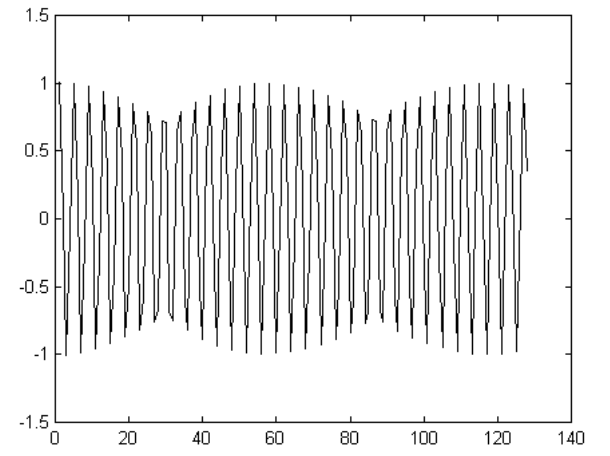
$$N = 128, \quad F_s = 8kHz$$

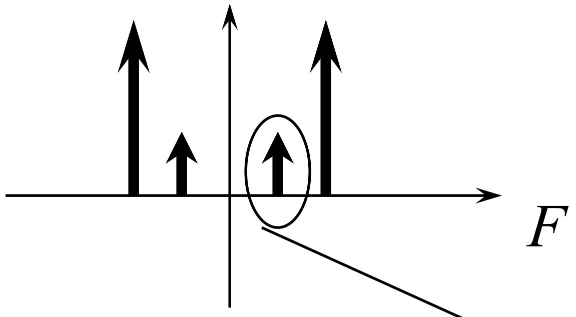


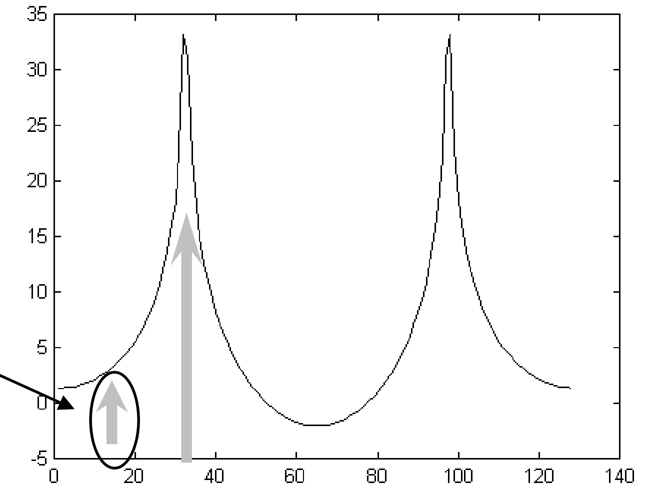
➔Distance between two distinguishable frequency components in *x(t)* must be larger than 2Fs/N

We can see only one peak since the two main lobes due to the two frequencies merge together

• Example 2: One *weak* component next to a *strong* component







The weak component is "buried" under the sidelobes of the strong component

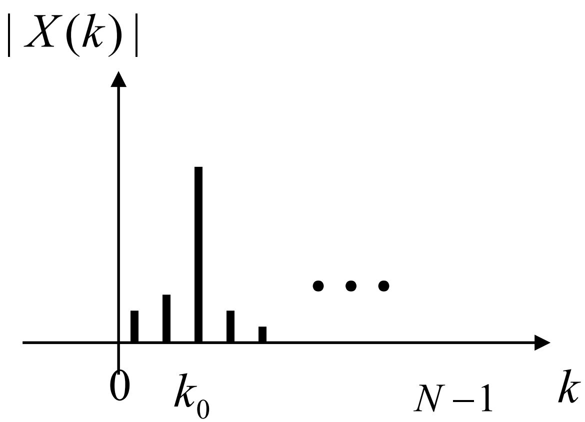
Solutions to <u>windowing</u> on the DTFT

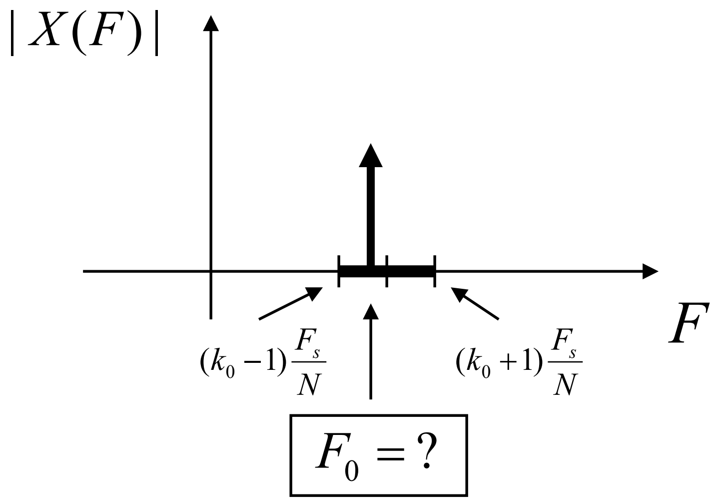
There are two main problems when using the DFT to estimate the frequency spectrum

1. <u>Loss of resolution</u>: a peak of the DFT at an index k_0 signifies a sinusoid at a frequency which can be anywhere in the interval

$$(k_0 - 1)\frac{2\pi}{N} < \omega < (k_0 + 1)\frac{2\pi}{N}$$ *in digital frequency (<u>radians</u>), or*

$$(k_0 - 1)\frac{F_s}{N} < F < (k_0 + 1)\frac{F_s}{N}$$ *in analog frequency (<u>Hz</u>)*





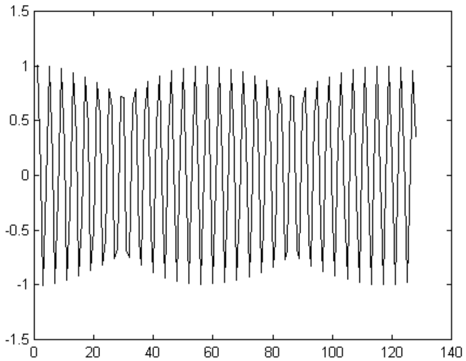
To <u>improve the resolution</u>: use more data points! (i.e., larger N)

2. Artifacts which can hide other frequency components.

To reduce artifacts use a different window than rectangular

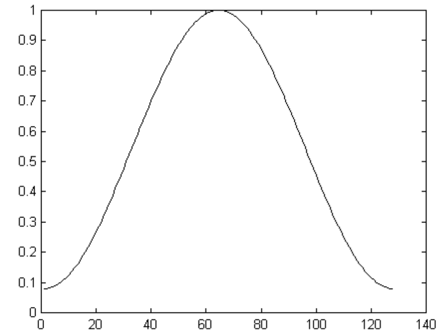
DFT with Window (Hamming Window):

$x(n)$



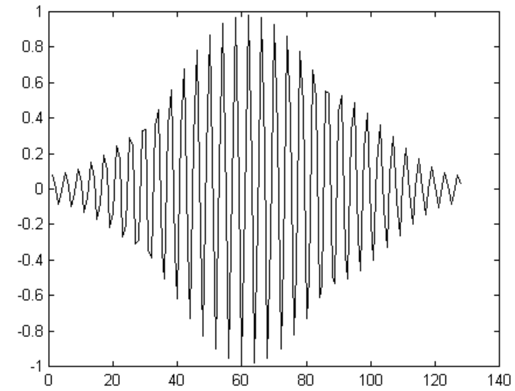
$\times$

$w(n)$

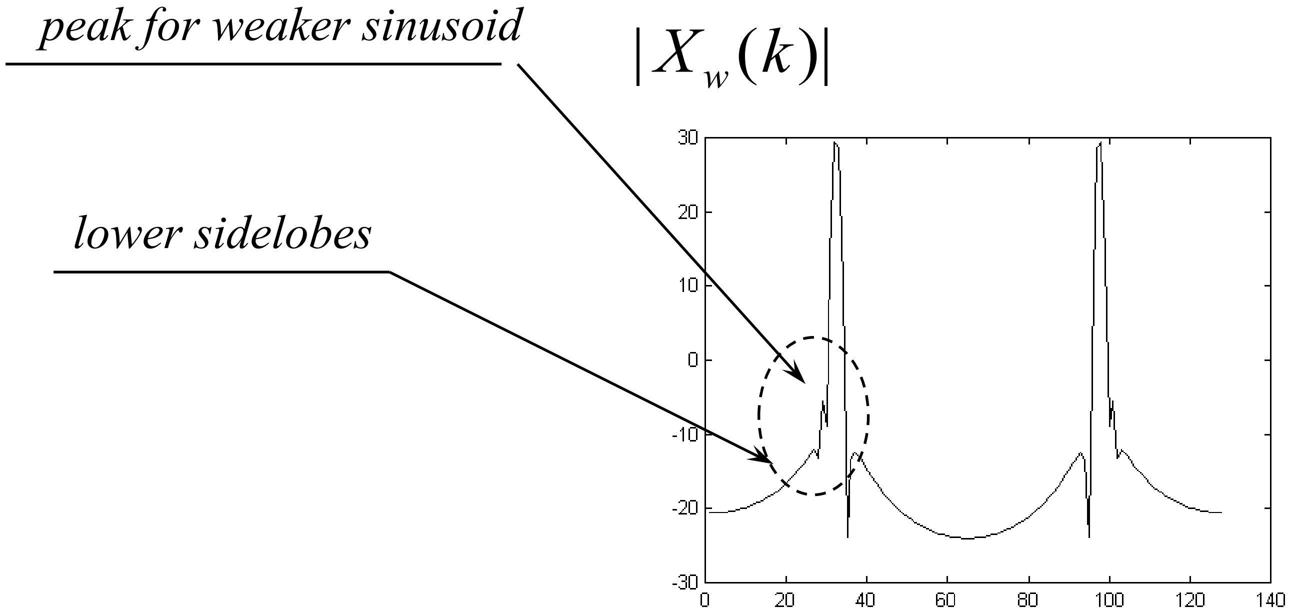


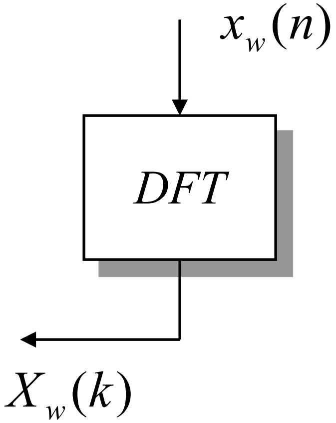
$=$

$x_w(n) = x(n)w(n)$



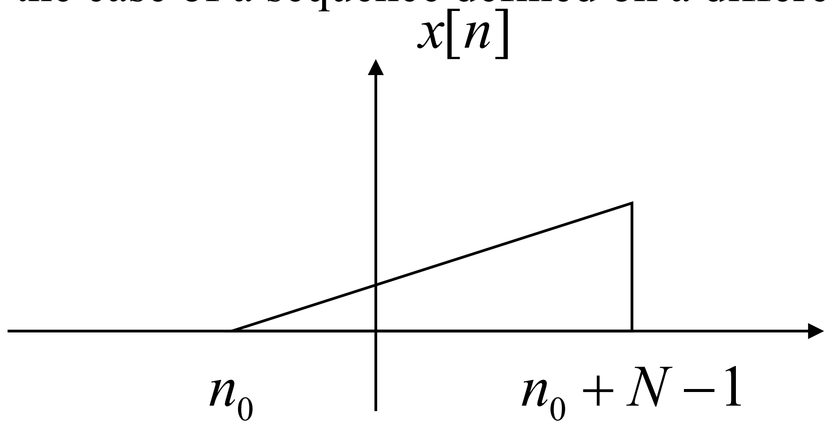
$|X_w(k)|$



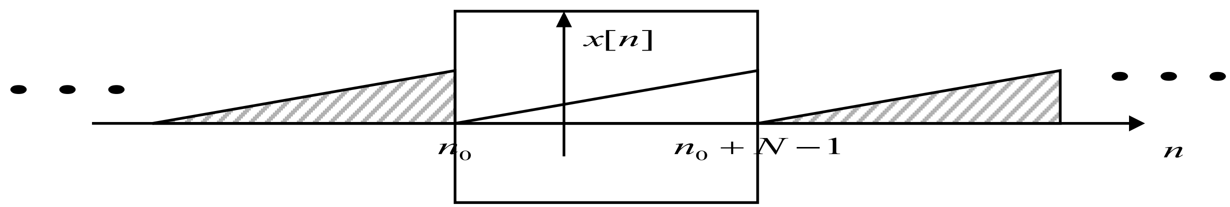


Extension to General Intervals of DFT Definition

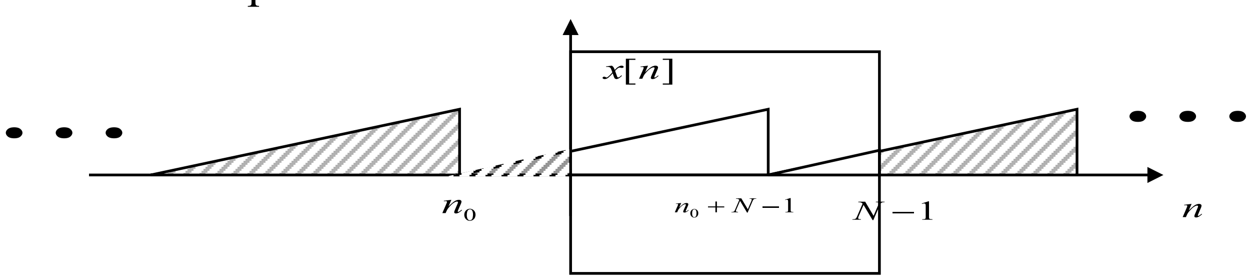
- DFT defined for x[n] 0<=n<N
- Take the case of a sequence defined on a different interval:



- How do we compute the DFT, without reinventing a new formula?
- First see the periodic extension, which looks like this:

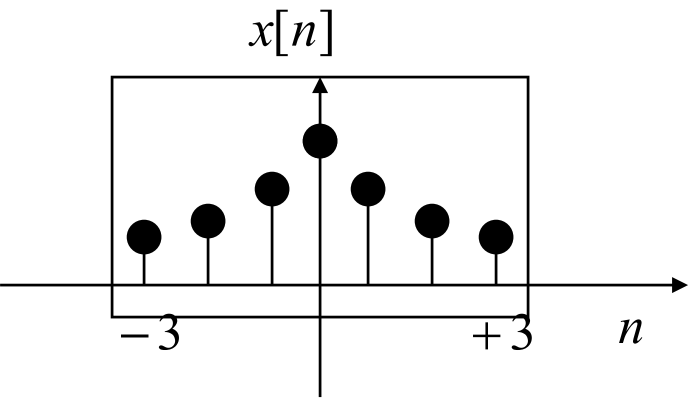


- Then look at the period $0 \le n \le N-1$



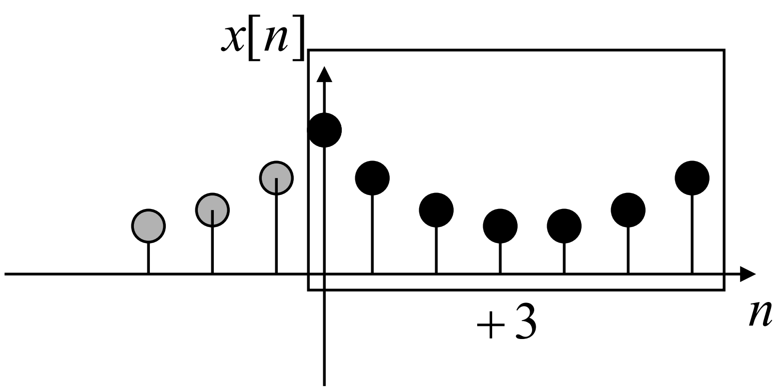
- Example: determine the DFT of the finite sequence

$$x[n] = 0.8^{|n|} \quad \text{if} \;\; -3 \le n \le +3$$



Then take the DFT of the vector

$$x = \left[x[0], x[1], \ldots, x[3], x[-3], \ldots, x[-1]\right]$$



Discrete Fourier Transform (DFT)

1. DFT:
   - Fourier Transform of short duration signals
2. DFT: Sampling of the DTFT
3. Convolution with DTF
4. DFT of long signals
   - The effect of windowing
5. The DFT as a Linear Transform
   - DFT as a vector-matrix operation
6. FFT

Based on:

- Chapter 8, A.V. Oppenheim and R.W. Schafer, *Discrete-Time Signal Processing*, Prentice-Hall, 3rd ed, 2010.
- Slides from http://faculty.nps.edu/rcristi/

DFT as a Vector-Matrix Operation

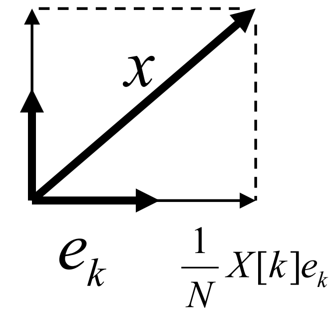
- Let

$$x = \begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[N-1] \end{bmatrix}, \qquad X = DFT\{x\} = \begin{bmatrix} X[0] \\ X[1] \\ \vdots \\ X[N-1] \end{bmatrix} \qquad e_k = \begin{bmatrix} 1 \\ w_N^{-k} \\ \vdots \\ w_N^{-k(N-1)} \end{bmatrix},$$

- Then:

$$X[k] = e_k^{*T} x$$

$$x = \frac{1}{N}\left(X[0]e_0 + X[1]e_1 + \ldots + X[N-1]e_{N-1}\right)$$



$$X = DFT\{x\} = \begin{bmatrix} X[0] \\ X[1] \\ \vdots \\ X[N-1] \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & w_N & \cdots & w_N^{N-1} \\ \vdots & & \ddots & \\ 1 & w_N^{N-1} & & w_N^{(N-1)(N-1)} \end{bmatrix}}_{W_N} \begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[N-1] \end{bmatrix}$$

- Linear Transform

$$\Rightarrow \boxed{\begin{aligned} X &= W_N x \\ x &= \underbrace{\frac{1}{N} W_N^{*T}}_{W_N^{-1}} X \end{aligned}}$$

FFT: Fast Fourier transform

- FFT is a direct computation of the DFT
- FFT is a set of algorithms for the efficient and digital computation of the N-point DFT, rather than a new transform
- Use the number of arithmetic multiplications and additions as a measure of computational complexity

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

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

- The DFT pair was given as
- Baseline for computational complexity:
  - Each DFT coefficient requires
    - N complex multiplications & N-1 complex additions
  - All N DFT coefficients require
    - N^2 complex multiplications & N(N-1) complex additions
- Complexity in terms of real operations
    - **$4N^2$ real multiplications**
    - **2N(N-1) real additions**

FFT

- Most fast methods are based on symmetry properties of DFT
  - Conjugate symmetry

$$e^{-j(2\pi/N)k(N-n)} = e^{-j(2\pi/N)kN} e^{-j(2\pi/N)k(-n)} = e^{j(2\pi/N)kn}$$

  - Periodicity in n and k

$$e^{-j(2\pi/N)kn} = e^{-j(2\pi/N)k(n+N)} = e^{j(2\pi/N)(k+N)n}$$

- The Second Order Goertzel Filter
  - Approximately N^2 real multiplications and $2N^2$ real additions
  - Do not need to evaluate all N DFT coefficients
- Decimation-In-Time FFT Algorithms
  - **$(N/2)\log_2 N$ complex multiplications and additions**

Symmetry and periodicity of complex exponential

- Complex conjugate symmetry

$$W_N^{k[N-n]} = W_N^{-kn} = (W_N^{kn})^* = \mathrm{Re}\{W_N^{kn}\} - j\,\mathrm{Im}\{W_N^{kn}\}$$

- Periodicity in n and k

$$W_N^{kn} = W_N^{k(n+N)} = W_N^{(k+N)n}$$

- For example

$$\mathrm{Re}\{x[n]\}\,\mathrm{Re}\{W_N^{kn}\} + \mathrm{Re}\{x[N-n]\}\,\mathrm{Re}\{W_N^{k[N-n]}\}$$
$$= (\mathrm{Re}\{x[n]\} + \mathrm{Re}\{x[N-n]\})\,\mathrm{Re}\{W_N^{kn}\}$$

➔The number of multiplications is reduced by a factor of 2