CHAPTER 5:

DISCRETE FOURIER TRANSFORM (DFT)

Lecture 9: DFT and Inverse DFT

Lecture 10: Fast Fourier Transform (FFT)

Duration: 4 hrs

Lecture 9

DFT and Inverse DFT

- **Duration:** **2** hrs
- **Outline:**

1. Review of DTFT of DT periodic signals

2. DFT and Inverse DFT

3. Frequency resolution

4. DFT properties

Procedure to calculate DTFT of periodic signals

Step 1:

Start with $\mathbf{x_0(n)}$ – one period of $\mathbf{x(n)}$**,** with zero everywhere else

Step 2:

Find the DTFT $\mathbf{X_0(\Omega)}$ of the signal $x_0[n]$ above

Step 3:

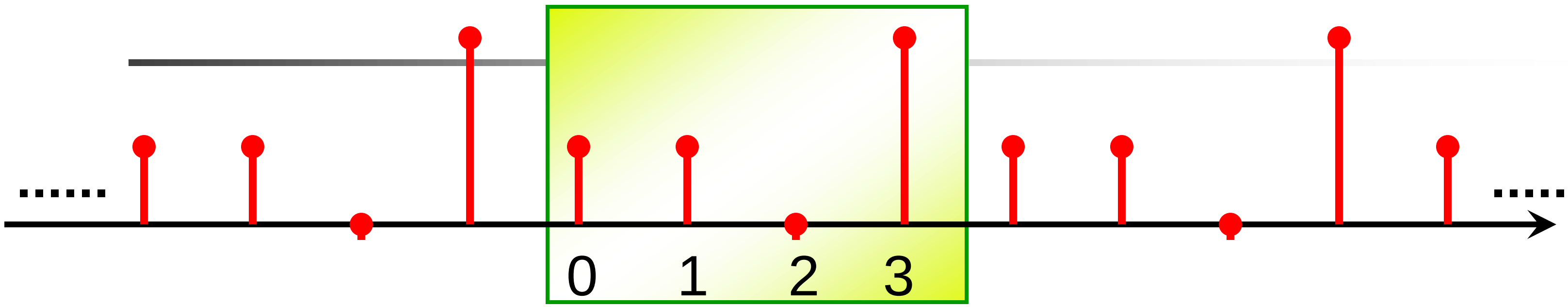
Find $X_0(\Omega)$ at **N** equally spacing frequency points $\mathbf{X_0(k2\pi/N)}$

Step 4:

Obtain the DTFT of x(n):

$$\boxed{X(\Omega) = \frac{2\pi}{N} \sum_k X_0(k\frac{2\pi}{N})\delta(\Omega - k\frac{2\pi}{N})}$$

Example of calculating DTFT of periodic signals



$$X_0(\Omega) = \sum_{n=0}^{3} x_0(n) e^{-j\Omega n} = 1 + e^{-j\Omega} + 2e^{-j3\Omega}$$

$$X_o\left(\frac{2\pi k}{4}\right) = 1 + e^{-j\frac{2\pi k}{4}} + 2e^{-3j\frac{2\pi k}{4}} \qquad k = 0, 1, 2, 3$$

k = 0 → $X_0(0) = 4$; k = 1 → $X_0(1) = 1+j$

k = 2 → $X_0(2) = -2$; k = 3 → $X_0(3) = 1-j$

Example (cont)

$$X(\Omega) = \frac{2\pi}{N} \sum_{k=-\infty}^{\infty} X_o(\frac{2\pi k}{N})\delta(\Omega - \frac{2\pi k}{N})$$

$$= \frac{2\pi}{4} \sum_{k=-\infty}^{\infty} X_o(\frac{2\pi k}{4})\delta(\Omega - \frac{2\pi k}{4})$$

For one period $0 \le \Omega < 2\pi$

$$\frac{\pi}{2}\left\{4\delta(\Omega) + (1+j)\delta(\Omega - \frac{\pi}{2}) - 2\delta(\Omega - \pi) + (1-j)\delta(\Omega - \frac{3\pi}{2})\right\}$$

Lecture 9

DFT and Inverse DFT

- **Duration:** **2** hrs
- **Outline:**

1. Review of DTFT of DT periodic signals

2. DFT and Inverse DFT

3. Frequency resolution

4. Applications

DFT to the rescue!

Could we calculate the **frequency spectrum** of a signal using a **digital computer** with **CTFT/DTFT**?

- ➢ Both CTFT and DTFT produce continuous function of frequency → can't calculate an infinite continuum of frequencies using a computer
- ➢ Most real-world data is not in the simple form such as $a^n u(n)$

DFT can be used as a FT approximation that can calculate a **finite set of discrete-frequency spectrum values** from a finite set of discrete-time samples of an analog signal

Building the DFT formula

Continuous time signal **x(t)**

sample

Discrete time signal **x(n)**

window

Discrete time signal **x_0(n)**

Finite length

"Window" x(n) is like multiplying the signal by the finite length rectangular window

$$w_R = \begin{cases} 1 & n = 0, 1, \cdots N-1 \\ 0, & otherwise \end{cases}$$

$$\boxed{x_o[n] = x[n]\, w_R[n]}$$

Building the DFT formula (cont)

Continuous time signal **x(t)**

sample

Discrete time signal **x(n)**

window

Discrete time signal $\mathbf{x_0(n)}$

Finite length

DTFT

$$X_0(\Omega) = \sum_{n=-\infty}^{\infty} x_0[n] e^{-j\Omega n} = \sum_{n=0}^{N-1} x_0[n] e^{-j\Omega n}$$

Discrete Time Fourier Transform (DTFT), $X_o(\Omega)$

(periodic over $[0, 2\pi)$)

Building the DFT formula (cont)

Continuous time signal **$x(t)$**

sample

Discrete time signal **$x(n)$**

window

Discrete time signal **$x_0(n)$**

Finite length

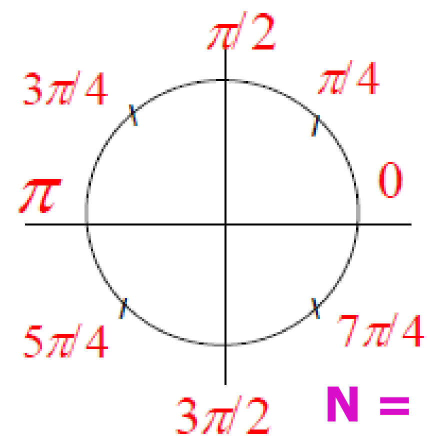
DFT

DTFT

Discrete Fourier Transform DFT **$X(k)$**

Discrete + periodic with period **N**

Sample at N values around the unit circle



N = 8

$X_0(\Omega)$

Continuous + periodic with period 2π

DFT and inverse DFT formulas

Notation: $W_N = e^{-j\frac{2\pi}{N}}$

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

k = 0,1,…,N-1

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

n = 0,1,…,N-1

Note that the DFT is a sequence of N numbers (in the frequency domain), just like $x_o[n]$ is a sequence of N numbers in the time domain

You only have to store N points

Examples of calculation DFT and IDFT

Ex.1. Find the DFT of x(n) = 1, n = 0, 1, 2, ..., (N-1)

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

k = 0,1,...,N-1

k = 0 → X(k) = X(0) = N

k ≠ 0 → X(k) = 0

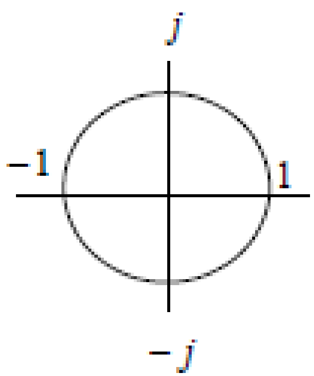
$$\Rightarrow \boxed{X[k] = N\delta[k]}$$

Examples of calculation DFT and IDFT

Ex.2. Given y(n) = δ(n-2) and N = 8, find Y(k)

$$Y[k] = \sum_{n=0}^{N-1} y[n] W_N^{kn} = \sum_{n=0}^{7} \delta[n-2] e^{-j2\pi kn/N}$$

Using N=8



$$= e^{-j4\pi k/8} = e^{-j\pi k/2} = (-j)^k$$

$$= [1, -j, -1, j, 1, -j, -1, j]$$

Examples of calculation DFT and IDFT

Ex.3. Find the IDFT of X(k) = 1, k = 0, 1, ..., 7.

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

$$\boxed{x[n] = \frac{1}{8}\sum_{k=0}^{7} W_8^{-kn} = \tfrac{1}{8} N\delta[n] = \delta[n]}$$

Examples of calculation DFT and IDFT

Ex.4. Given x(n) = δ(n) + 2δ(n-1) +3δ(n-2) + δ(n-3) and N = 4. Find X(k).

$$X[k] = 1 + 2e^{-j\pi k/2} + 3e^{-j\pi k} + e^{-j3\pi k/2}$$

$$= 1 + 2(-j)^k + 3(-1)^k + (j)^k$$

$$X[0] = 7$$

$$X[1] = 1 - 2j - 3 + j = -2 - j$$

$$X[2] = 1 - 2 + 3 - 1 = 1$$

$$X[3] = 1 + 2j - 3 - j = -2 + j$$

- x = [1 2 3 1];
- >> X = fft(x)

- X = 7.0000 -2.0000 - 1.0000i 1.0000 -2.0000 + 1.0000i
- 7.0000 1.7071 - 5.1213i -2.0000 - 1.0000i 0.2929 + 0.8787i

- Columns 5 through 8

- 1.0000 0.2929 - 0.8787i -2.0000 + 1.0000i 1.7071 + 5.1213i

Lecture 9

DFT and Inverse DFT

- **Duration:** **2** hrs
- **Outline:**

1. Review of DTFT of DT periodic signals

2. DFT and Inverse DFT

3. Frequency resolution

4. DFT properties

Frequency resolution of the DFT

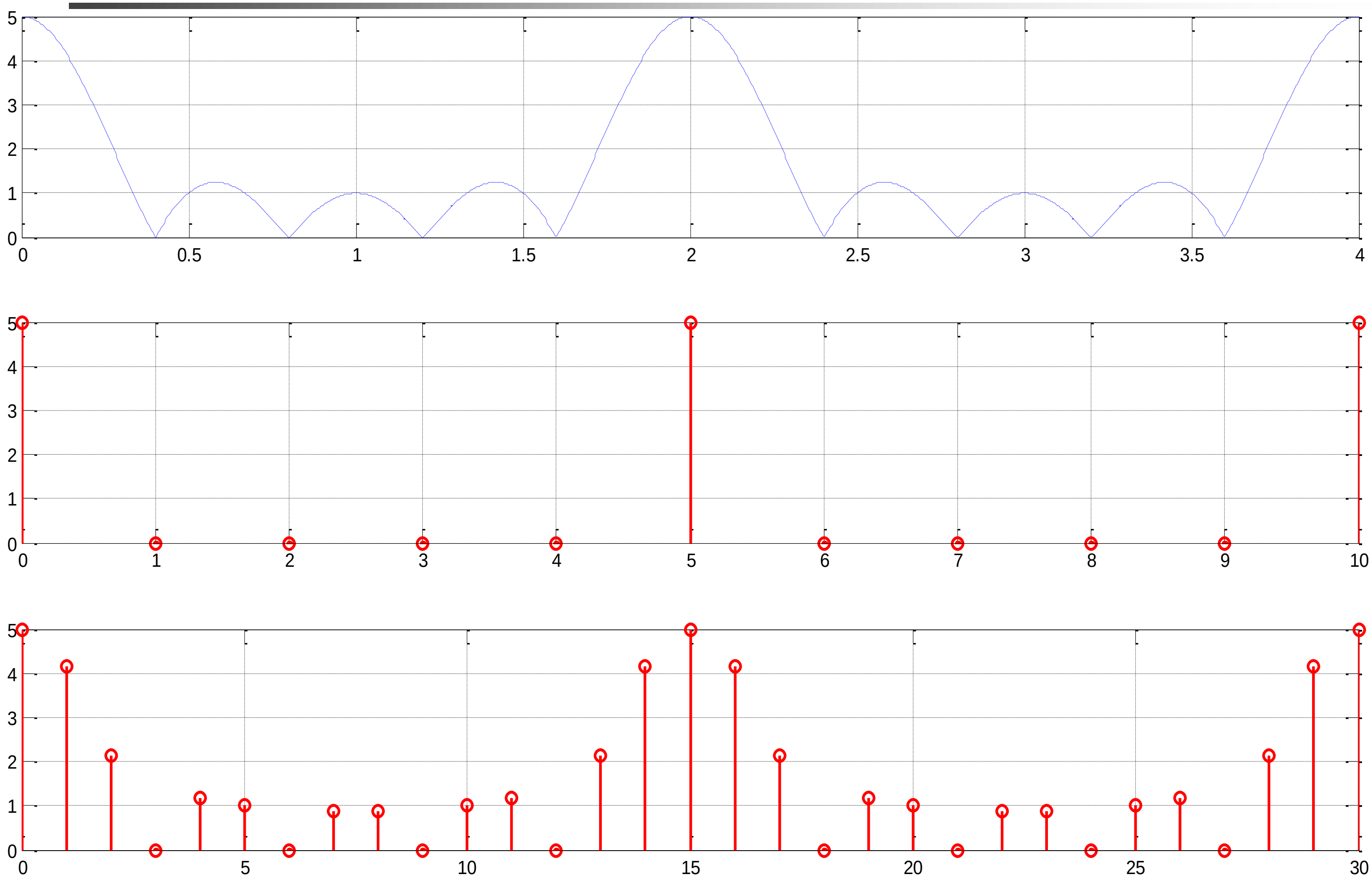
Discrete frequency spectrum computed from DFT has the spacing between frequency samples of:

$$\boxed{\Delta f = \frac{f_s}{N}} \qquad \boxed{\Delta\Omega = \frac{2\pi}{N}}$$

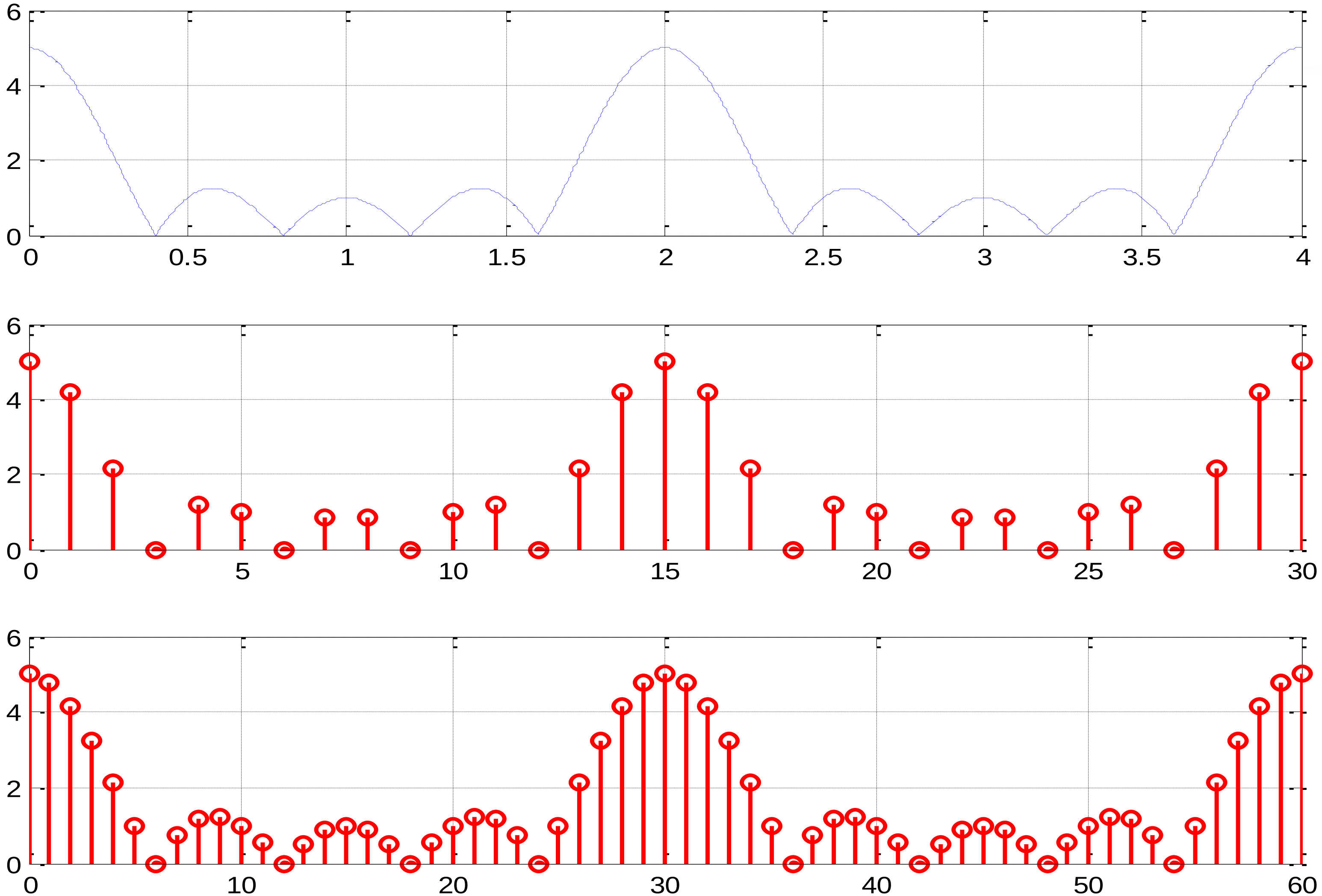
→ The choice of N determines the resolution of the frequency spectrum, or vice-versa

→ To obtain the adequate resolution, some zeros can be appended to the signal (*zero padding*)

Examples of N = 5 and N = 15



Examples of N = 15 and N = 30



Lecture 9

DFT and Inverse DFT

- **Duration:** **2** hrs
- **Outline:**

1. Review of DTFT of DT periodic signals

2. DFT and Inverse DFT

3. Frequency resolution

4. DFT properties

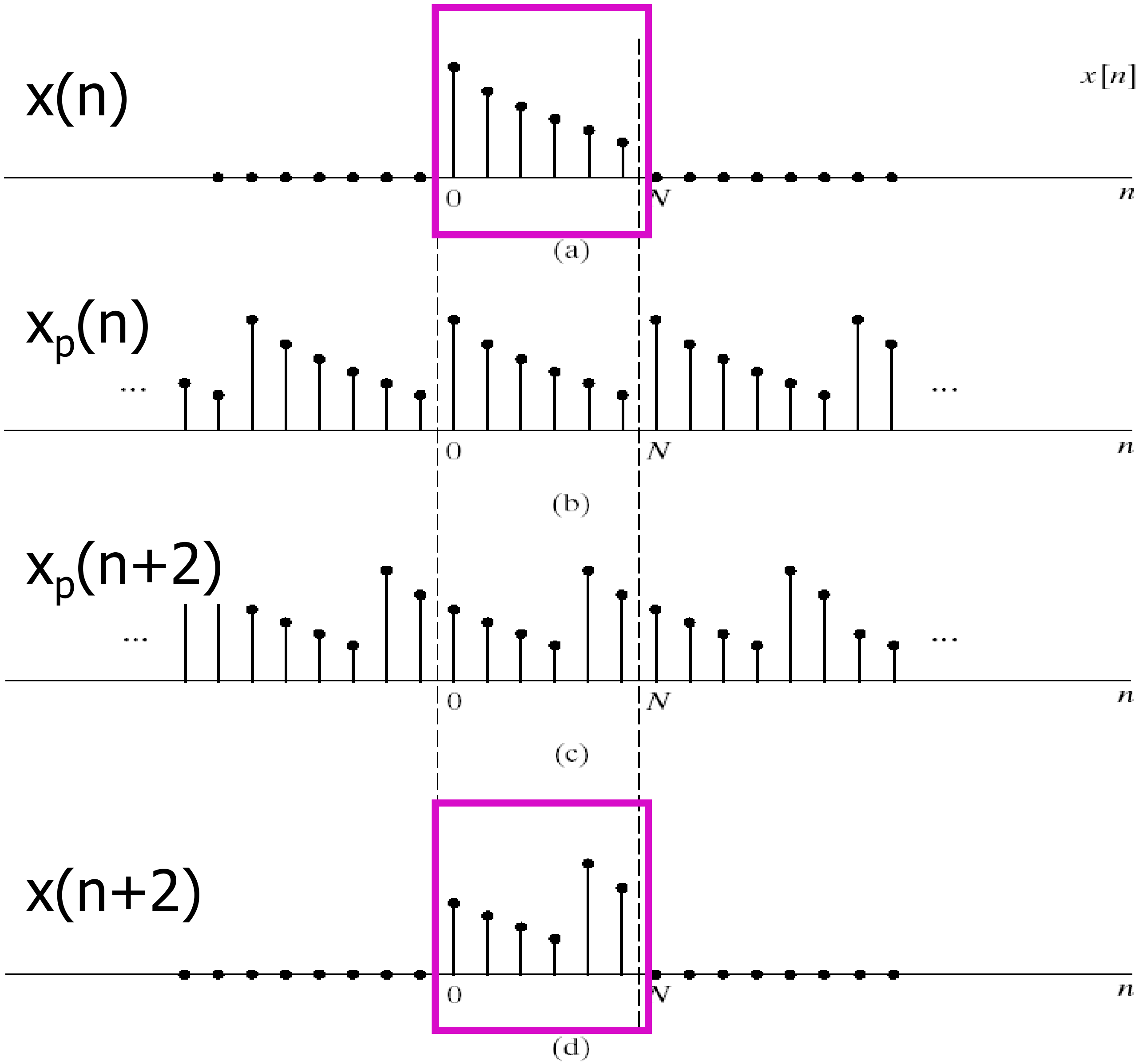
Circular shift property of the DFT

$x(n)$	N periodic extension $\longrightarrow$	$x_p(n)$
Finite length	$\longleftarrow$ truncate	Infinite length, periodic

$x(n)$ is one period of signal $x_p(n)$

$$\boxed{x[n-m] \overset{\text{DFT}}{\longleftrightarrow} W^{km} X[k]}$$

Circular shift property of the DFT



Circular shift by ***m*** is the same as a shift by ***m modulo N***

Recall linear convolution

$$y[n] = x_1[n] * x_2[n] = \sum_{p=-\infty}^{\infty} x_1[p] x_2[n-p]$$

- N_1: the non-zero length of $x_1(n)$; N_2: the non-zero length of $x_2(n)$; $N_y = N_1 + N_2 - 1$
- The shift operation is the regular shift
- The flip operation is the regular flip

Circular convolution of the DFT

$$y[n] = x_1[n] \otimes x_2[n] = \sum_{p=0}^{N-1} x_1[p] x_2[n-p]_{\mathrm{mod}\, N}$$

- The non-zero length of $x_1(n)$, $x_2(n)$ and y(n) can be no longer than N
- The shift operation is circular shift
- The flip operation is circular flip

$$x_1[n] \otimes x_2[n] \overset{DFT}{\leftrightarrow} X_1[k] X_2[k]$$

Direct method to calculate circular convolution

1. Draw a circle with *N* values of *x(n)* with *N* equally spaced angles in a counterclockwise direction.

2. Draw a smaller radius circle with *N* values of *h(n)* with equally spaced angles in a clockwise direction. Superimpose the centers of 2 circles, and have *h(0)* in front of *x(0).*

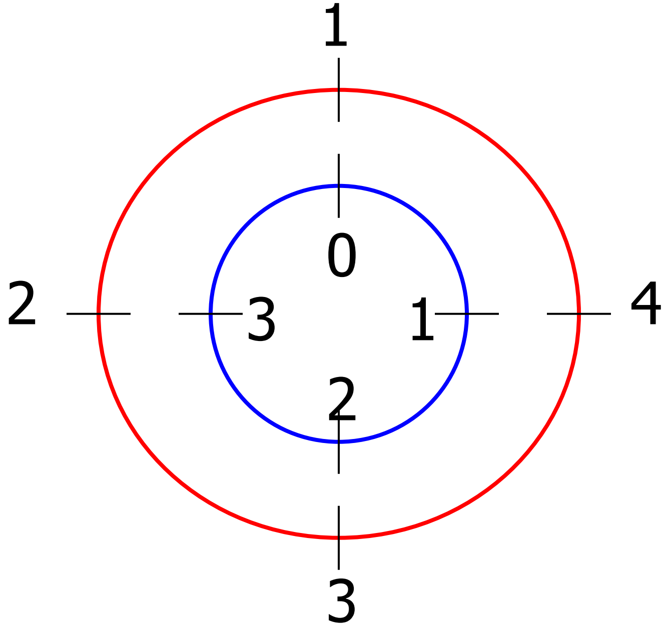
3. Calculate *y(0)* by multiplying the corresponding values on each radial line, and then adding the products.

4. Find succeeding values of *y(n)* in the same way after rotating the inner disk counterclockwise through the angle $2\pi k/N$

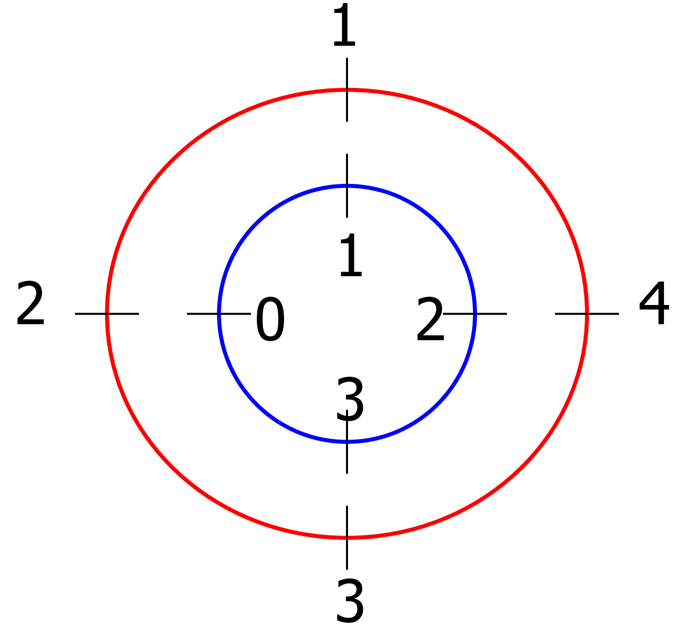
Example to calculate circular convolution

Evaluate the circular convolution, *y(n)* of 2 signals:

$x_1(n) = [\ 1\ 2\ 3\ 4\ ];\ x_2(n) = [\ 0\ 1\ 2\ 3\ ]$

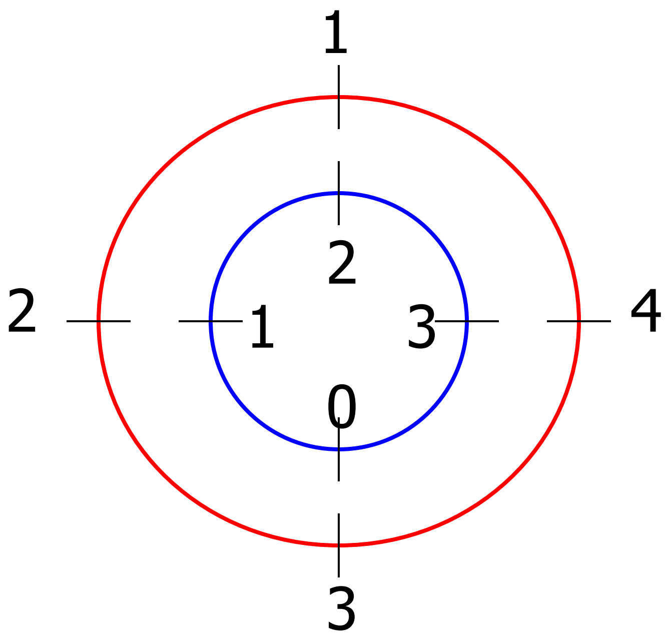


y(0) = 16

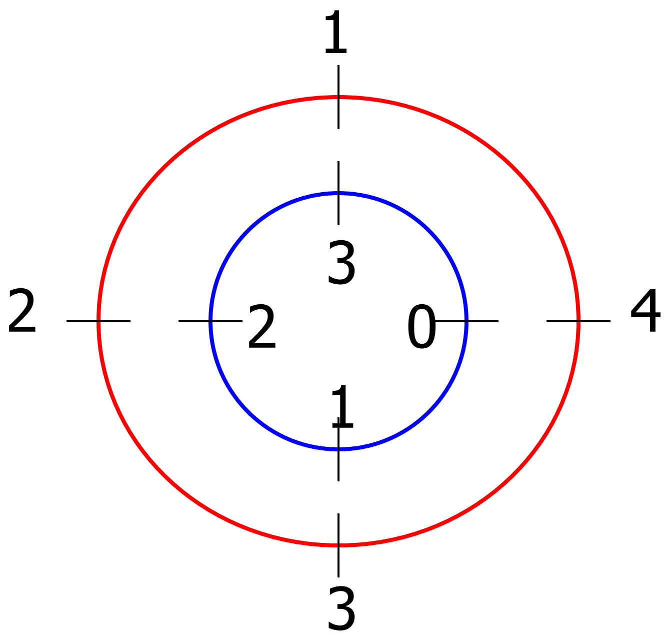


y(1) = 18

Example (cont)

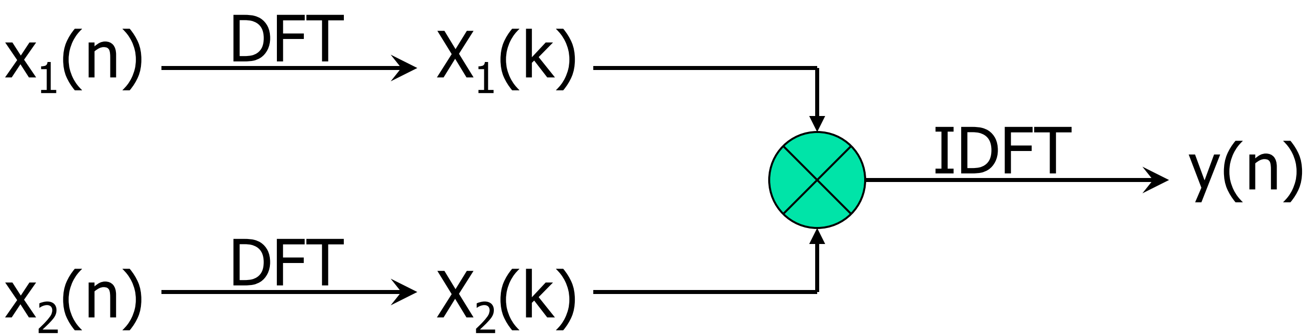


y(2) = 16



y(3) = 10

Another method to calculate circular convolution



Ex. $x_1(n) = [\ 1\ 2\ 3\ 4\ ]$; $x_2(n) = [\ 0\ 1\ 2\ 3\ ]$

$X_1(k) = [\ 10, -2+j2, -2, -2-j2\ ]$;

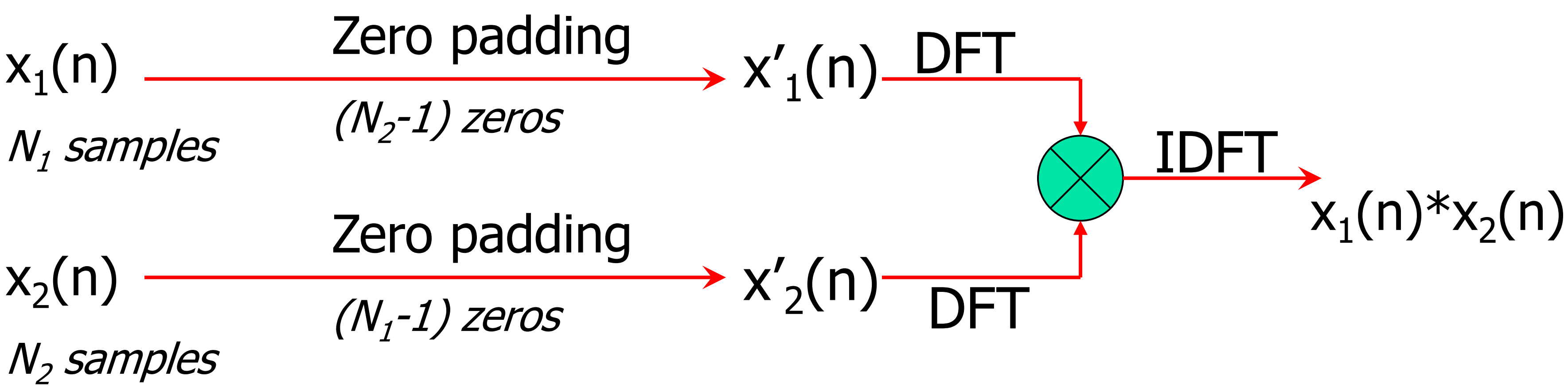
$X_2(k) = [\ 6, -2+j2, -2, -2-j2\ ]$;

$Y(k) = X_1(k).X_2(k) = [\ 60, -j8, 4, j8\ ]$

$y(n) = [\ 16, 18, 16, 10\ ]$

Calculation of the linear convolution

The circular convolution of 2 sequences of length N_1 and N_2 can be made **equal to** the linear convolution of 2 sequences by zero padding both sequences so that they both consists of N_1+N_2-1 samples.



Example of calculation the linear convolution

1 2 3 4]; $x_2(n)$ = [0 1 2 3]

$x'_1(n)$ = [1 2 3 4 0 0 0]; $x'_2(n)$ = [0 1 2 3 0 0 0]

$X'_1(k)$ = [10, -2.0245-j6.2240, 0.3460+j2.4791, 0.1784-j2.4220, 0.1784+j2.4220, 0.3460-j2.4791, -2.0245-j6.2240];

$X'_2(k)$ = [6, -2.5245-j4.0333, -0.1540+j2.2383, -0.3216-j1.7950, -0.3216+j1.7950, -0.1540-j2.2383, -2.5245+j4.0333];

Y'(k) = [60, -19.9928+j23.8775, -5.6024+j0.3927, -5.8342-j0.8644, -4.4049+j0.4585, -5.6024-j0.3927, -19.9928 +j23.8775]

IDFT{Y'(k)} = y'(n) = [0 1 4 10 16 17 12]

HW

Prob.1

Compute the DFT with N time samples:

$$\text{(a)} \quad x[n] = \delta[n]$$

$$\text{(b)} \quad x[n] = a^{n}\{u[n] - u[n-N]\}$$

$$\text{(c)} \quad x[n] = \begin{cases} 1 & n \text{ even} \\ 0 & n \text{ odd} \quad 0 \le n \le N-1 \end{cases}$$

HW

Prob.2

Given the two four-point sequences:

$$x(n) = [\ 1 \ \ 0.75 \ \ 0.5 \ \ 0.25\ ]$$

$$y(n) = [\ 0.75 \ \ 0.5 \ \ 0.25 \ \ 1]$$

Express the DFT $Y(k)$ in terms of the DFT $X(k)$

HW

Prob.3

Given signals below and their DFT-5

(a) $x_1[n] = \delta[n-1] + 2\delta[n-2] + 3\delta[n-3] + 4\delta[n-4]$

(b) $x_2[n] = \delta[n-1]$

(c) $s[n] = \delta[n]$

1. Find y[n] so that $Y[k] = X_1[k].X_2[k]$

2. Does $x_3[n]$ exist, if $S[k] = X_1[k].X_3[k]$?

HW

Prob.4 Given x(n) and its 8-point DFT, X(k)

$$x[n]=\begin{cases}1, & 0\le n\le 3\\ 0, & 4\le n\le 7\end{cases}$$

Express the DFTs of the signals below in terms of X(k).

$$\text{(a)}\quad x_1[n]=\begin{cases}1, & n=0\\ 0, & 1\le n\le 4\\ 1, & 5\le n\le 7\end{cases}$$

$$\text{(b)}\quad x_2[n]=\begin{cases}0, & 0\le n\le 1\\ 1, & 2\le n\le 5\\ 0, & 6\le n\le 7\end{cases}$$

HW

Prob.5 The 8-point DFTs of x(n) and h(n) are:

$$X(k) = [0, -j0.707, -j, -j0.707, 0, j0.707, j, j0.707]$$

and

$$H(k) = [3, 2.414, 1, -0.414, -1, -0.414, 1, 2.414]$$

Find the value of y(2), where y(n) is the circular convolution of x(n) and h(n).

Lecture 10

Fast Fourier Transform (FFT)

- **Duration: 2** hrs
- **Outline:**

1. What is FFT?

2. The decomposition-in-time Fast Fourier Transform algorithm

Recall DFT and IDFT definition

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

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

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

DFT plays an important role in the analysis, design and implementation of the DT signal processing algorithms and systems.

Major reason: existence of efficient algorithms for computing DFT called **FFT**

Direct computation of the DFT

$$\boxed{X[k]=\sum_{n=0}^{N-1}x[n]W_N^{nk} \quad 0\le k\le N-1}$$

The direct computation of the DFT requires:

1. ***N*** complex multiplications for each of *k*
2. ***N²*** complex multiplications for all *N* points of *X(k)*
3. ***(N-1)*** complex summations for each of *k*
4. ***N(N-1)*** complex summations for all *N* points of *X(k)*

→ **FFT** optimize computational processes (1) & (2) in different algorithms

Decomposition in time FFT (DIT-FFT)

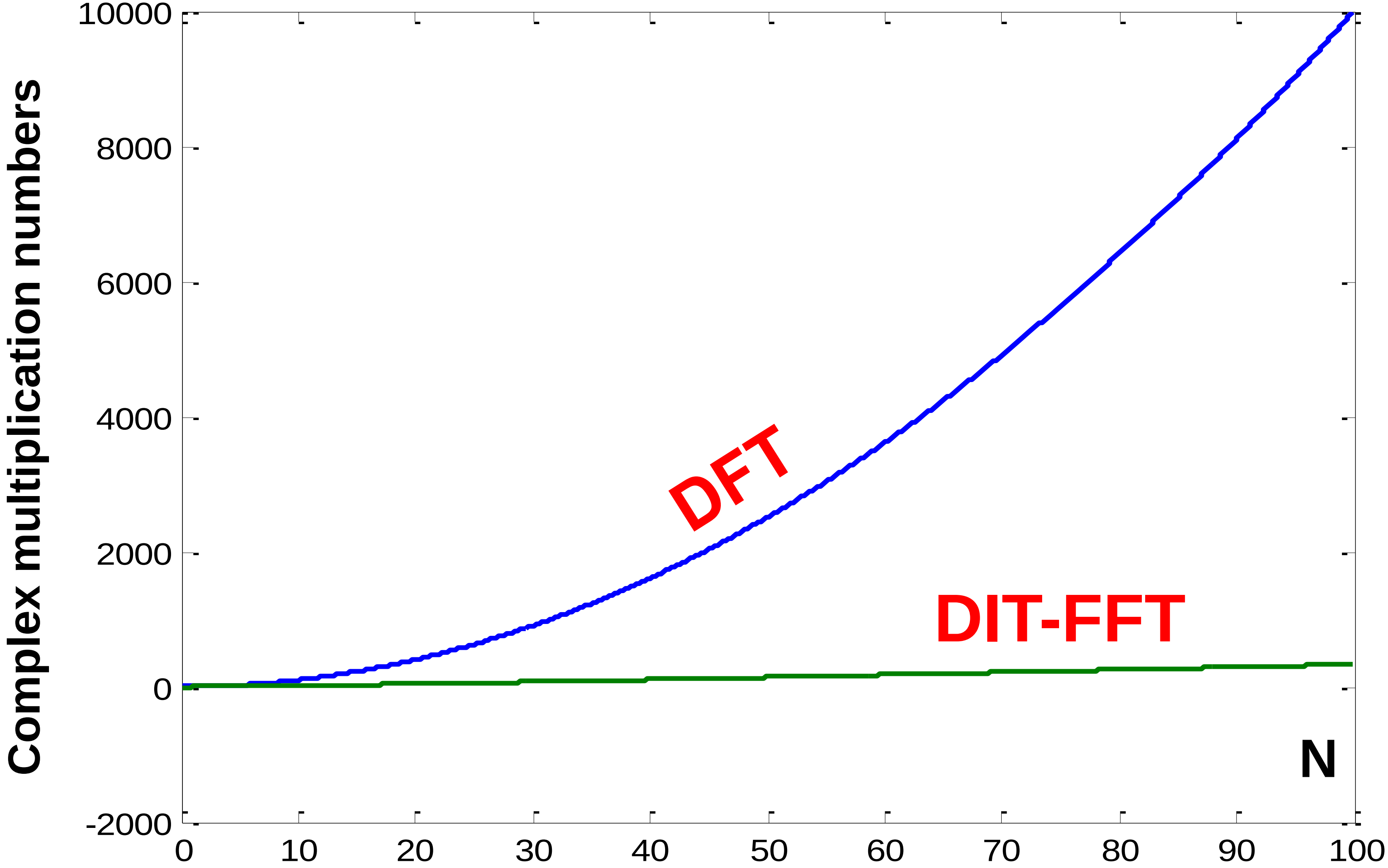
Breaking an N-point DFT into smaller DFTs

→ Fewer calculations with the same output

For example: N is radix-2 number

$$\boxed{N^2 \Rightarrow \frac{N}{2}\log_2 N \text{ complex multiplications}}$$

Comparing DFT and FFT efficiency



Computation of DFT

- Some properties of $\left\{W_N^{nk}\right\}$ can be exploited

$$W_N^{k(N-n)} = W_N^{-kn} = \left(W_N^{kn}\right)^*, \text{complex conjugate symmetry}$$

$$W_N^{kn} = W_N^{k(n+N)} = W_N^{(k+N)n}, \text{periodicity in n and k}$$

- Other useful properties:

$$W_N^{(k+\frac{N}{2})n} = W_N^{kn} W_N^{\frac{nN}{2}} = W_N^{kn} e^{-jn\pi} = \begin{cases} W_N^{kn}, & \text{if n even} \\ -W_N^{kn}, & \text{if n odd} \end{cases}$$

$$W_N^{2kn} = W_{\frac{N}{2}}^{kn}$$

Lecture 10

Fast Fourier Transform (FFT)

- **Duration: 2** hrs
- **Outline:**

1. What is FFT?

2. The decomposition-in-time Fast Fourier Transform algorithm

DIT-FFT with N as a 2-radix number

- *G(k)* is N/2 points DFT of the even numbered data: x(0), x(2), x(4), …., x(N-2).
- *H(k)* is the N/2 points DFT of the odd numbered data: x(1), x(3), …, x(N-1).

$$X[k] = \sum_{n\,\mathrm{even}} x[n]W^{kn} + \sum_{n\,\mathrm{odd}} x[n]W^{kn}$$

DIT-FFT (cont)

$$X[k] = \sum_{m=0}^{\frac{N}{2}-1} x[2m]W^{2mk} + \sum_{m=0}^{\frac{N}{2}-1} x[2m+1]W^{k(2m+1)}$$

$$= \sum_{m=0}^{\frac{N}{2}-1} x[2m](W^2)^{mk} + W^k \sum_{m=0}^{\frac{N}{2}-1} x[2m+1](W^2)^{mk} =$$

$$\mathrm{W}_{\mathrm{N}}^{2} = \left(\mathrm{e}^{-\mathrm{j}2\pi/\mathrm{N}}\right)^{2} = \mathrm{e}^{-\mathrm{j}2\pi/(\mathrm{N}/2)} = \mathrm{W}_{\mathrm{N}/2}$$

$$\boxed{\mathrm{X(k)} = \mathrm{G(k)} + \mathrm{W}_{\mathrm{N}}^{\mathrm{k}}\mathrm{H(k)}, \quad \mathrm{k} = 0,1,\ldots,\mathrm{N}-1}$$

G(k) and *H(k)* are of length N/2; *X(k)* is of length *N*

G(k)=G(k+N/2) and *H(k)=H(k+N/2)*

8-point FFT

$$X[k]_8 = G[k]_4 + W_8^k H[k]_4$$

$$X[0] = G[0] + W_8^0 H[0]$$

$$X[1] = G[1] + W_8^1 H[1]$$

$$X[2] = G[2] + W_8^2 H[2]$$

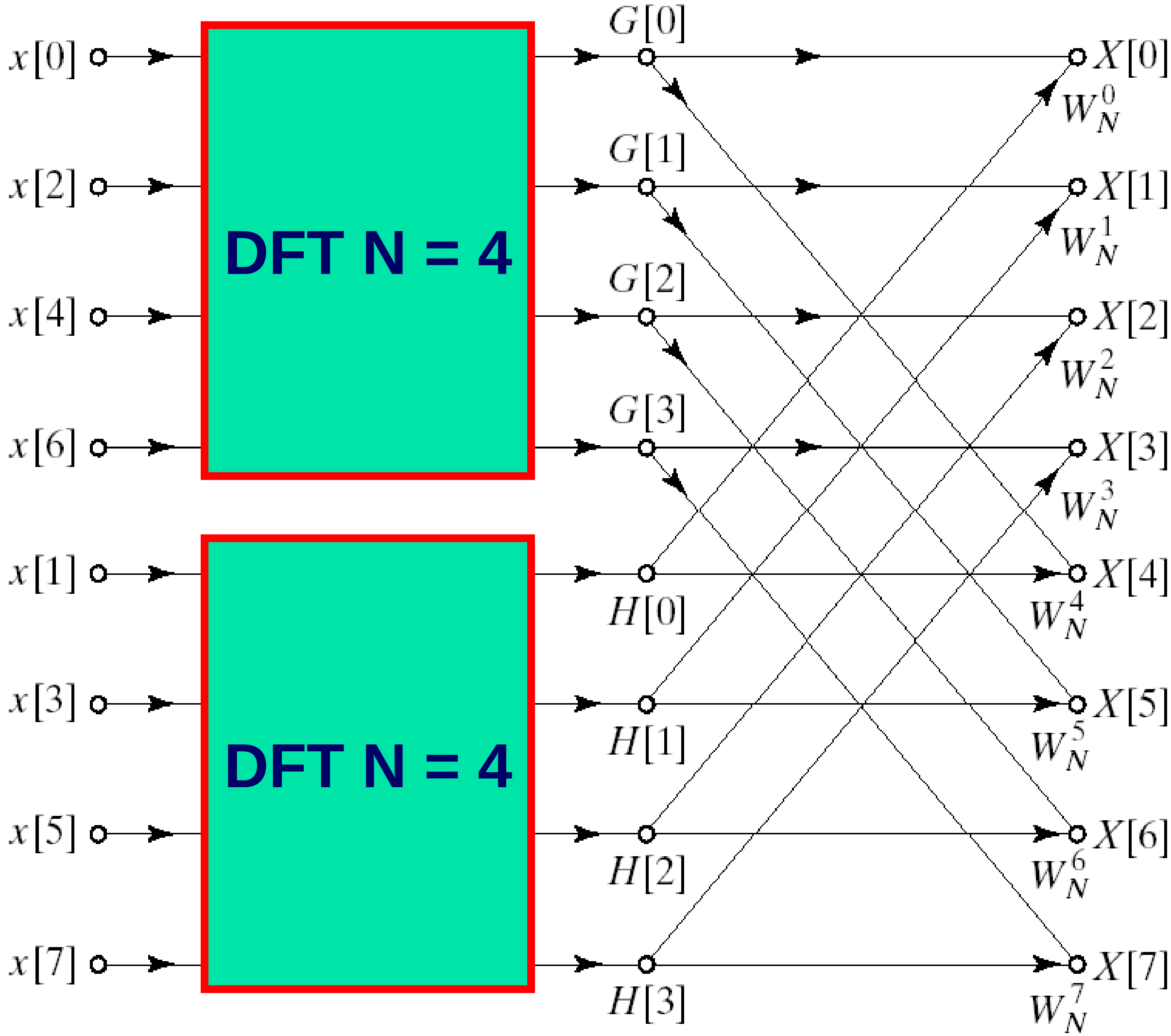
$$X[3] = G[3] + W_8^3 H[3]$$

$$X[4] = G[0] + W_8^4 H[0]$$

$$X[5] = G[1] + W_8^5 H[1]$$

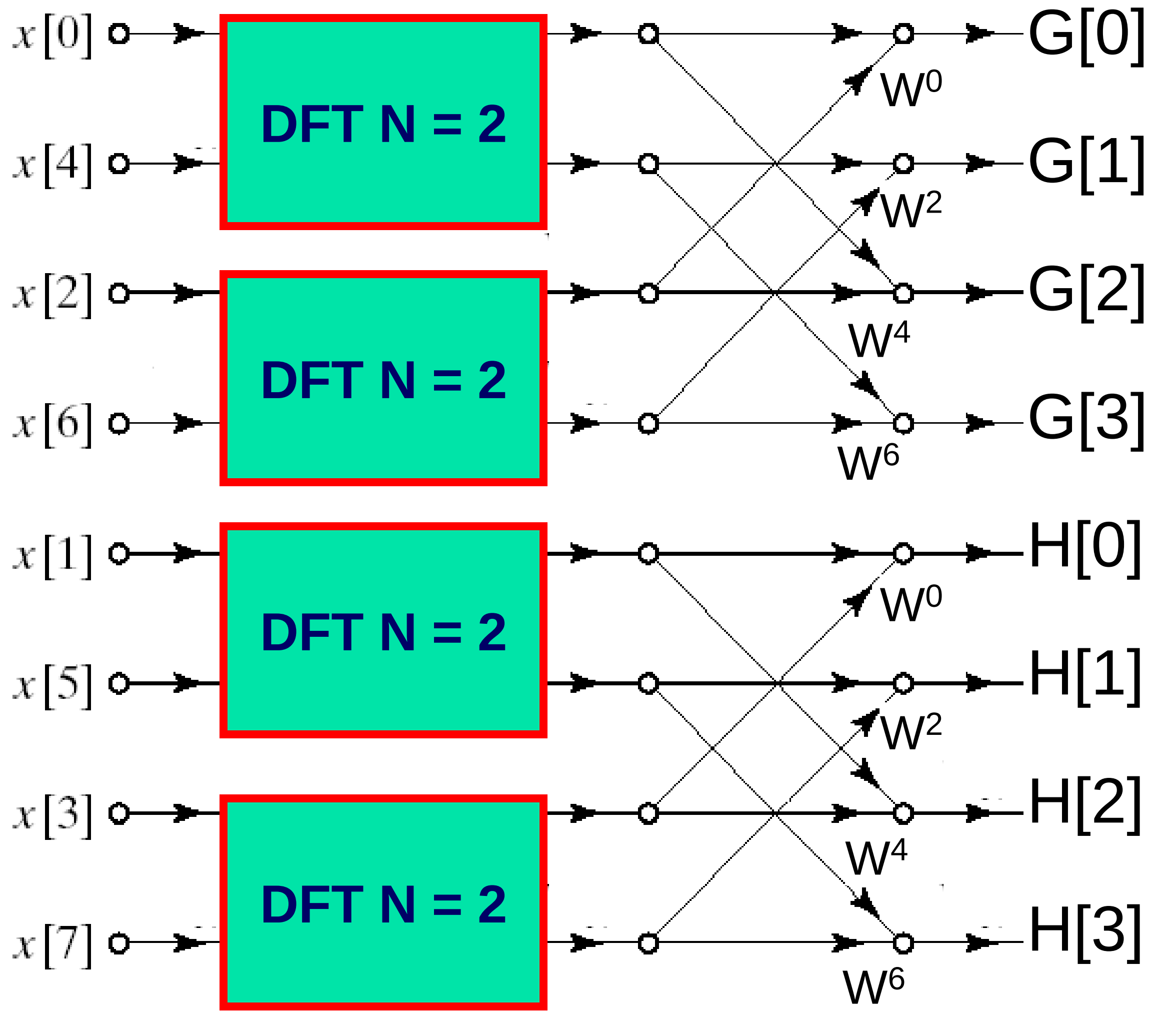
$$X[6] = G[2] + W_8^6 H[2]$$

$$X[7] = G[3] + W_8^7 H[3]$$

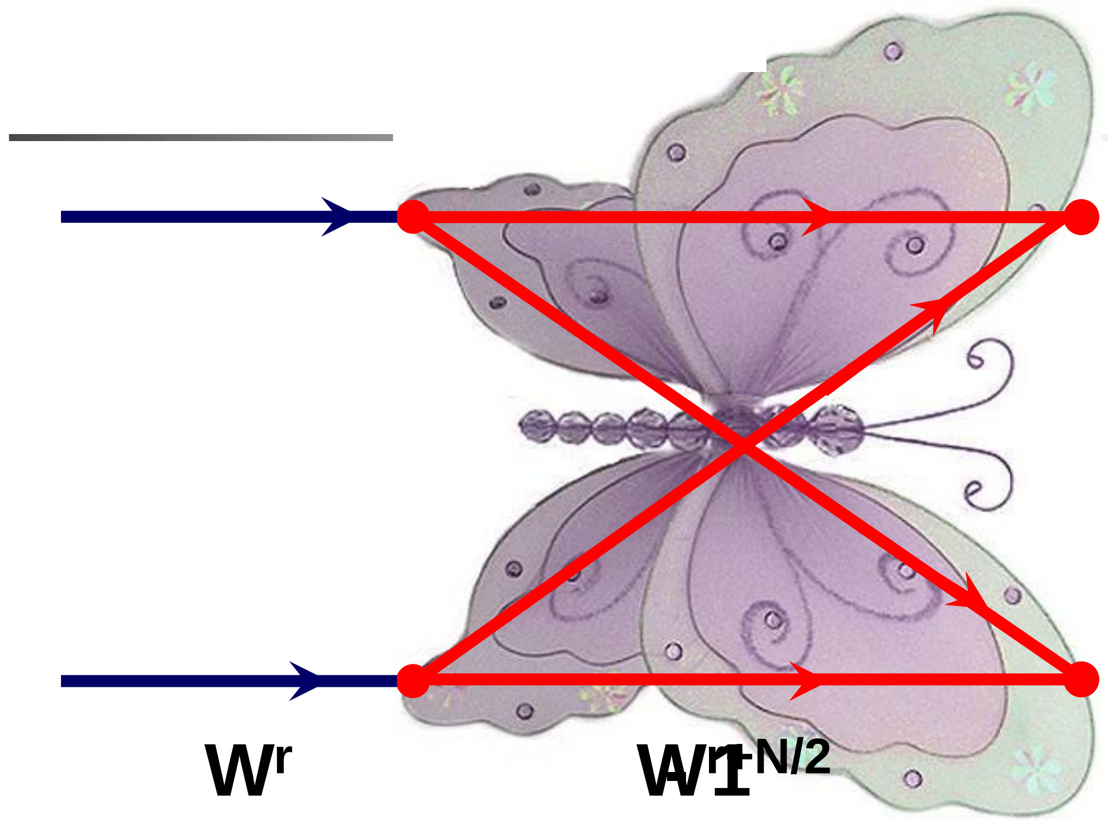


The new computation counts are reduced

8-point FFT (cont)

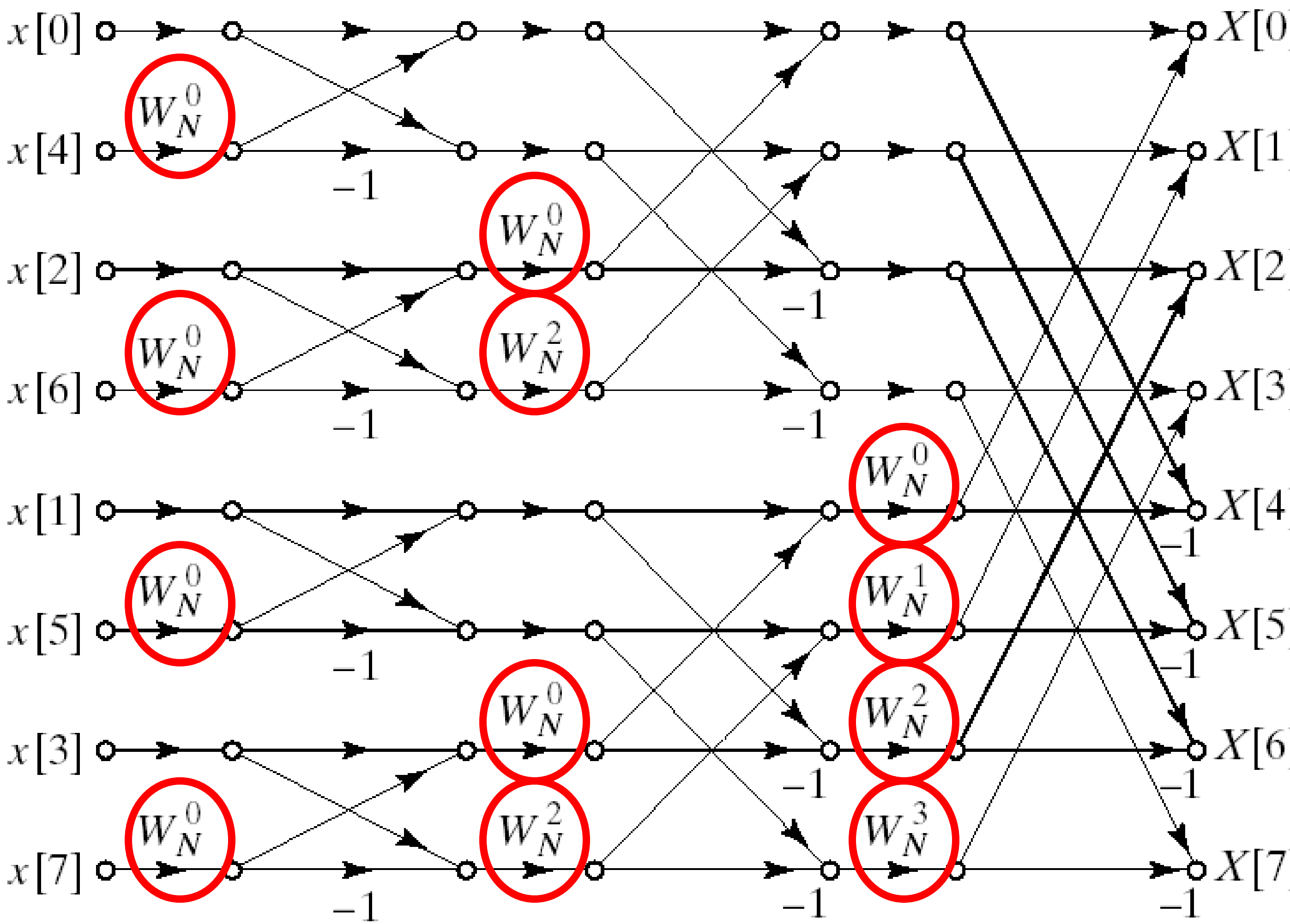


2-point FFT



$$\boxed{W^{r+N/2} = -W^r}$$

8-point FFT



Now the overall computation is reduced to:

$$N^2 \Rightarrow \frac{N}{2}\log_2 N \text{ complex multiplications}$$

HW

Prob.6

(a) Draw an eight-point DIT FFT signal-flow diagram, and use it to solve for the DFT of the sequence x(n)

$$\mathbf{x(n) = (0.5)^{n}\,[u(n) - u(n-8)]}$$

(b) Use Matlab (*fft*) to confirm the result of part (a)