Lecture I213E – Class 10

Discrete Signal Processing

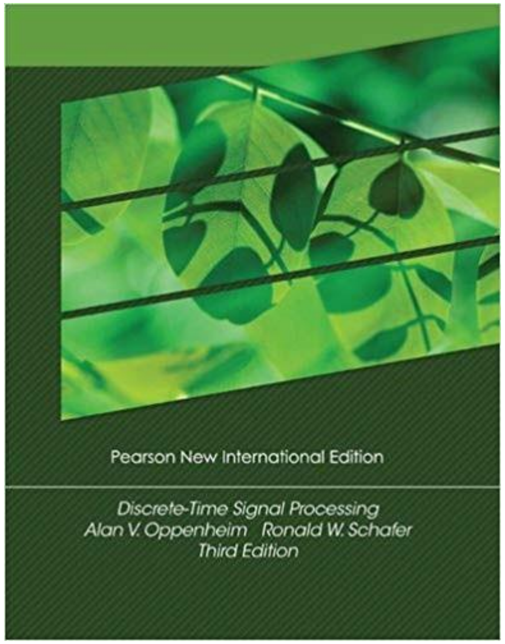
Sakriani Sakti

JAPAN ADVANCED INSTITUTE OF SCIENCE AND TECHNOLOGY
1990

Course Materials

- **Materials**

  → Lecture notes will be uploaded before each lecture
  https://jstorage-2018.jaist.ac.jp/s/PGXRrC7iFmN2FWo
  Pass: dsp-i213e-2022
  (Slide Courtesy of Prof. Nak Young Chong)



- **References**

  → Chi-Tsong Chen:
  Linear System Theory and Design, 4th Ed., Oxford University Press, 2013.

  → Alan V. Oppenheim and Ronald W. Schafer:
  Discrete-Time Signal Processing, 3rd Ed., Pearson New International Ed., 2013.

Related Courses & Prerequisite

- **Related Courses**
  - → I212 Analysis for Information Science
  - → I114 Fundamental Mathematics for Information Science

- **Prerequisite**
  - → None

Evaluation

▪ Viewpoint of evaluation

→ Students are able to understand:

- Basic principles in modeling and analysis of linear time-invariant systems
- Applications of mathematical methods and tools to different signal processing problems.

▪ Evaluation method

→ Homework, term project, midterm exam, and final exam

▪ Evaluation criteria

→ Homework/labs (30%), term project (30%)
midterm exam (15%), and final exam (25%)

Contact

- **Lecturer**

  → Sakriani Sakti

- **TA**

  Tutorial hours & Term project

  → WANG Lijun (s2010026)

  → TANG Bowen (s2110411)

  Homework

  → PUTRI Fanda Yuliana (s2110425)

- **Contact Email**

  → dsp-i213e-2022@ml.jaist.ac.jp

Schedule

- **December 8th, 2022 – February 9th, 2023**

- **Lecture Course Term 2-2**

  → Tuesday 9:00 — 10:40

  → Thursday 10:50 — 12:30

- **Tutorial Hours**

  → Tuesday 13:30-15:10

Schedule

Dec

Sun	Mon	Tue	Wed	Thu	Fri	Sat
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Jan

Sun	Mon	Tue	Wed	Thu	Fri	Sat
1	2	3	4	~~5~~	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	~~24~~	25	26	27	28
29	30	31				

Feb

Sun	Mon	Tue	Wed	Thu	Fri	Sat
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28				

Lecture:
Tuesday 9:00 — 10:40
Thursday 10:50 — 12:30

Tutorial:
Tuesday 13:30 — 15:10

Course review &
term project evaluation
(on tutorial hours)

Midterm & final exam
Thursday 10:50 — 12:30

Syllabus

Class	Date	Lecture Course Tue 9:00 — 10:40 / Thr 10:50 — 12:30	Tutorial Hours Tue 13:30 — 15:10
1	12/08	Introduction to Linear Systems with Applications to Signal Processing	
2	12/13	State Space Description	○
3	12/15	Linear Algebra	
4	12/20	Quantitative Analysis (State Space Solutions) and Qualitative Analysis (Stability)	○
5	12/22	Discrete-time Signals and Systems	
X	01/05		
6	01/10	Discrete-time Fourier Analysis	
7	01/10*	Review of Discrete-time Linear Time-Invariant Signals and Systems (on Tutorial Hours)	
	01/12	Midterm Exam	
8	01/17	Sampling and Reconstruction of Analog Signals	○
9	01/19	z-Transform	
X	01/24		○
10	01/26	Discrete Fourier Transform	
11	01/31	FFT Algorithms	○
12	01/02	Implementation of Digital Filters	
13	02/07	Digital Signal Processors and Design of Digital Filters	
14	02/07*	Review of the Course and Term Project Evaluation (on Tutorial Hours)	
	02/09	Final exam	

Class 10
Discrete Fourier Transform

Function Transformation

	Complex Freq.	Real Freq.	
		Periodic	Not Periodic
Continuous Time	LT	**CTFS**	**CTFT**
Discrete Time	**ZT**	**DFT**	**DTFT**

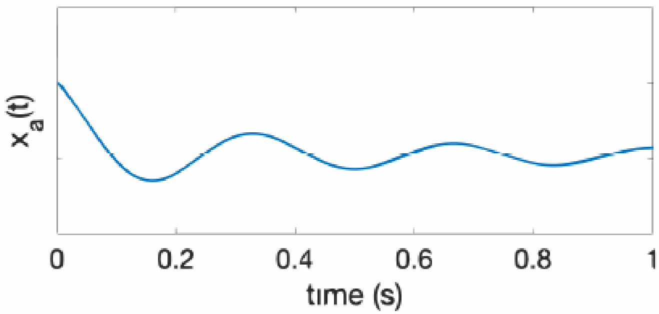
- **Discrete Fourier Transform**

→ Computer-based frequency domain analysis

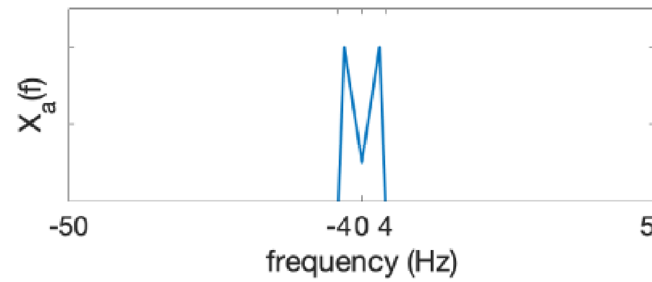
  - Spectral analysis (e.g. Finding periodicities)
  - Denoising
  - Compression (e.g. JPEG)
  - Filtering
  - “Fast” Convolution

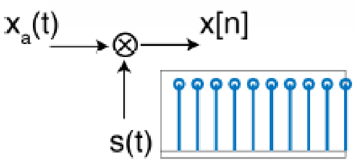
Discrete Transform: DTFT, Z-Transform & DFT

Review DTFT



Fourier Transfrom →





A/D ↓ ↑ D/A

reconstruction:

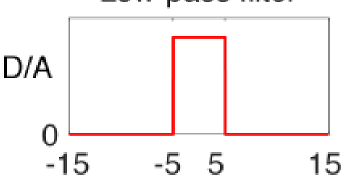
sampling: $x[n] = x_a(nT)$

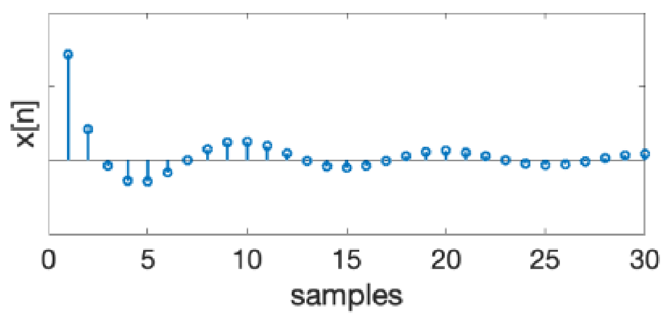
$$x_a(t) = \sum_{n=-\infty}^{\infty} x[n] \frac{\sin(2\pi / T_s (t - nT_s))}{\pi / T_s (t - nT_s)}$$

$$X(f) = F_s \sum_{k=-\infty}^{\infty} X_a(f - kF_s)$$

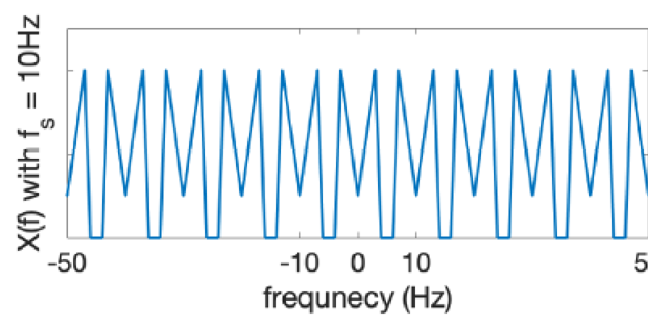
A/D ↓ ↑ D/A

Low-pass filter





Discrete Time Fourier Transform →



Time domain

Frequency domain

DTFT & Z-Transform

- **Discrete-time Fourier Transform (DTFT)**

  → Frequency-domain representation for absolutely summable sequences

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

- **Z-Transform**

  → A generalized frequency-domain representation for arbitrary sequences

$$X(z) = \sum_{n=-\infty}^{\infty} x(n) z^{-n}$$

- **Two features in common**
  1. Defined for infinite-length sequences (From $-\infty$ to ∞)
  2. Functions of continuous variables: ω or $z = re^{j\omega}$

  → Easiest to appreciate when they are defined as mathematical expressions.

But, practically (in a computer program),
how to evaluate infinite sums at uncountably infinite frequencies?

DTFT & Z-Transform

- **The discrete-time Fourier transform and the z-transform are not numerically computable transforms**

We turn our attention to a numerically computable transform

Computer analysis: discrete time and finite duration (N)

Solutions:

- only consider a finite number of samples in time, and
- only consider a finite number of frequencies.

Discrete Fourier Series & Transform

- **What are we supposed to do?**

→ **Sample the discrete-time Fourier transform in the frequency domain** (or z-transform on the unit circle).

→ **Construct a periodic sequence:**
A periodic sequence can always be represented by a linear combination of harmonically related complex exponentials (which is a form of sampling).

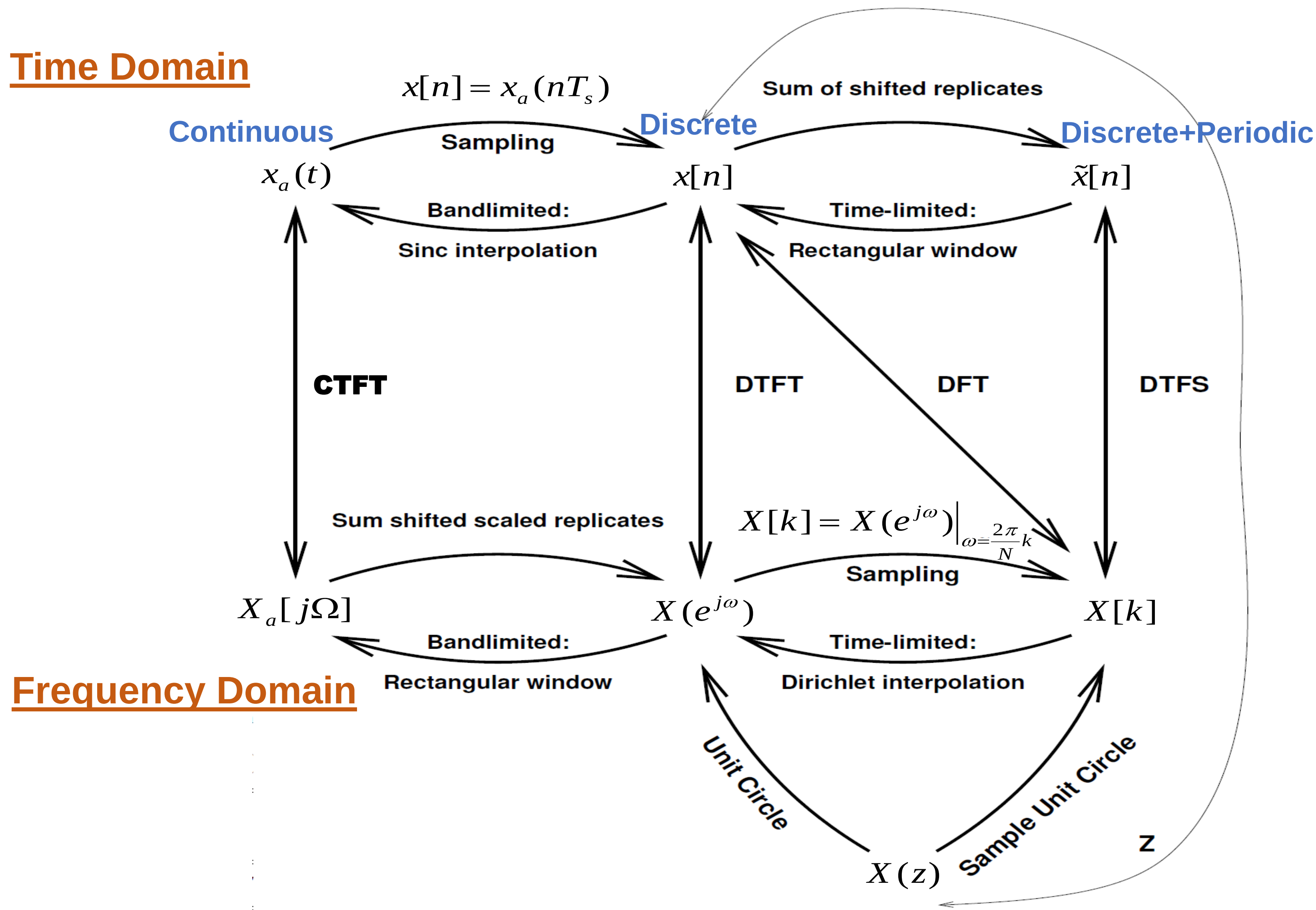
Discrete Fourier Series

→ **Study the effects of sampling in the time domain** and **the issue of reconstruction in the z-domain**.

→ **Extend the Discrete Fourier Series to finite-duration sequences**.

Discrete Fourier Transform

Fourier Transform Family Relationship



Discrete-time Fourier Series (DTFS)/ Discrete Fourier Series (DFS)

Discrete Fourier Series

- **Periodic Sequence**

$$\boxed{\tilde{x}(n) = \tilde{x}(n + kN), \quad \forall n, k}$$

the fundamental period of the sequence

The periodic functions can be synthesized as a linear combination of complex exponentials whose frequencies are multiples (or harmonics) of the fundamental frequency $(2\pi / N)$.

There are a finite number of harmonics; the frequencies are

$$\left\{ \frac{2\pi}{N} k, \quad k = 0,1,\ldots,N-1 \right\}$$

Discrete Fourier Series

- **Analysis: a DFS Equation**

$$\tilde{X}(k) = \sum_{n=0}^{N-1} \tilde{x}(n) e^{-j\frac{2\pi}{N}nk}, \quad k = 0, \pm 1, \ldots, \qquad W_N \equiv e^{-j\frac{2\pi}{N}}$$

$$\tilde{X}(k) \equiv DFS\left[\tilde{x}(n)\right] = \sum_{n=0}^{N-1} \tilde{x}(n) W_N^{nk}$$

- **Synthesis: an inverse DFS Equation**

$$\tilde{x}(n) = \frac{1}{N} \sum_{k=0}^{N-1} \tilde{X}(k) e^{j\frac{2\pi}{N}kn}, \quad n = 0, \pm 1, \ldots, \qquad W_N \equiv e^{-j\frac{2\pi}{N}}$$

$$\tilde{x}(n) \equiv IDFS\left[\tilde{X}(k)\right] = \frac{1}{N} \sum_{k=0}^{N-1} \tilde{X}(k) W_N^{-nk}$$

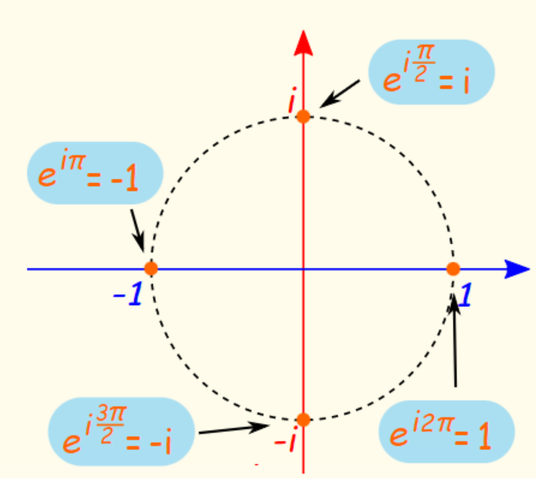
Example

Find the DFS representation:

$$\tilde{x}(n) = \{\ldots,0,1,2,3,\underset{\uparrow}{0},1,2,3,0,1,2,3,\ldots\}$$

$$\tilde{X}(k) = \sum_{n=0}^{3} \tilde{x}(n) W_4^{nk}, \quad k = 0, \pm 1, \pm 2, \ldots$$

$$\boxed{N = 4, \quad W_4 = e^{-j\frac{2\pi}{4}} = -j}$$



$$\tilde{X}(0) = \sum_{0}^{3} \tilde{x}(n) W_4^{0n} = \tilde{x}(0) + \tilde{x}(1) + \tilde{x}(2) + \tilde{x}(3) = 6$$

$$\tilde{X}(1) = \sum_{0}^{3} \tilde{x}(n) W_4^{1n} = \sum_{0}^{3} \tilde{x}(n)(-j)^{n} = (-2 + 2j)$$

$$\tilde{X}(2) = \sum_{0}^{3} \tilde{x}(n) W_4^{2n} = \sum_{0}^{3} \tilde{x}(n)(-j)^{2n} = -2$$

$$\tilde{X}(3) = \sum_{0}^{3} \tilde{x}(n) W_4^{3n} = \sum_{0}^{3} \tilde{x}(n)(-j)^{3n} = (-2 - 2j)$$

Example: MATLAB

DFS matrix

$$W_N \equiv \left[W_N^{kn}\ _{0\le k,n\le N-1}\right] = \begin{array}{c c} & n \rightarrow \\ \begin{array}{c} k \\ \downarrow \end{array} & \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & W_N^1 & \cdots & W_N^{(N-1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & W_N^{(N-1)} & \cdots & W_N^{(N-1)^2} \end{bmatrix} \end{array}$$

```
function [Xk] = dfs(xn,N)
% Compute Discrete Fourier Series Coefficients
% ---------------------------------------------------------
% [Xk] = dfs(xn,N)
% Xk = DFS coeff. array over 0 <= k <= N-1
% xn = One period of periodic signal over 0 <= n <= N-1
% N = Fundamental period of xn
%
n = [0:1:N-1];              % row vector for n
k = [0:1:N-1];              % row vector for k
WN = exp(-j*2*pi/N)         % Wn factor
nk = n'*k;                  % creates a N by N matrix of nk value
WNnk = WN .^ nk;            % DFS matrix
Xk = xn * WNnk;             % row vector for DFS coefficients
```

Example: MATLAB

```
function [xn] = idfs(Xk,N)
% Compute Inverse Discrete Fourier Series
% -----------------------------------------------
% [xn] = idfs(Xk,N)
% xn = One period of periodic signal over 0 <= n <= N-1
% Xk = DFS coeff. array over 0 <= k <= N-1
% N = Fundamental period of Xk
%
n = [0:1:N-1];            % row vector for n
k = [0:1:N-1];            % row vector for k
WN = exp(-j*2*pi/N)       % Wn factor
nk = n'*k;                % creates a N by N matrix of nk value
WNnk = WN .^ (-nk);       % IDFS matrix
xn = (Xk * WNnk)/N;       % row vector for IDFS values
```

The DFS in the previous example can be computed using MATLAB as

```
>> xn = [0,1,2,3]; N = 4; Xk = dfs(xn,N)
Xk =
    6.0000   -2.0000 + 2.0000i   -2.0000 – 0.0000i   -2.0000 – 2.0000i
```

Relation to Z-Transform

- **Z-Transform of Finite-duration Sequence**

  $x(n)$ a finite-duration sequence of duration N

$$x(n) = \begin{cases} Nonzero, & 0 \le n \le N-1 \\ 0, & elsewhere \end{cases}$$

$$X(z) = \sum_{n=0}^{N-1} x(n) z^{-n}$$

- **Z-Transform of Periodic Sequence**

  Construct $\tilde{x}(n)$ a periodic sequence by periodically repeating $x(n)$ with period N

$$x(n) = \begin{cases} \tilde{x}(n), & 0 \le n \le N-1 \\ 0, & elsewhere \end{cases}$$

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

$$\tilde{X}(k) = X(z)\Big|_{z=e^{j\frac{2\pi}{N}k}}$$

N evenly spaced samples of the z-transform $X(z)$ around the unit circle

Relation to DTFT

- **DTFT of Finite-duration Sequence**

  $x(n)$ a finite-duration sequence of duration N (absolutely summable)

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

- **Sampling interval in the frequency domain**

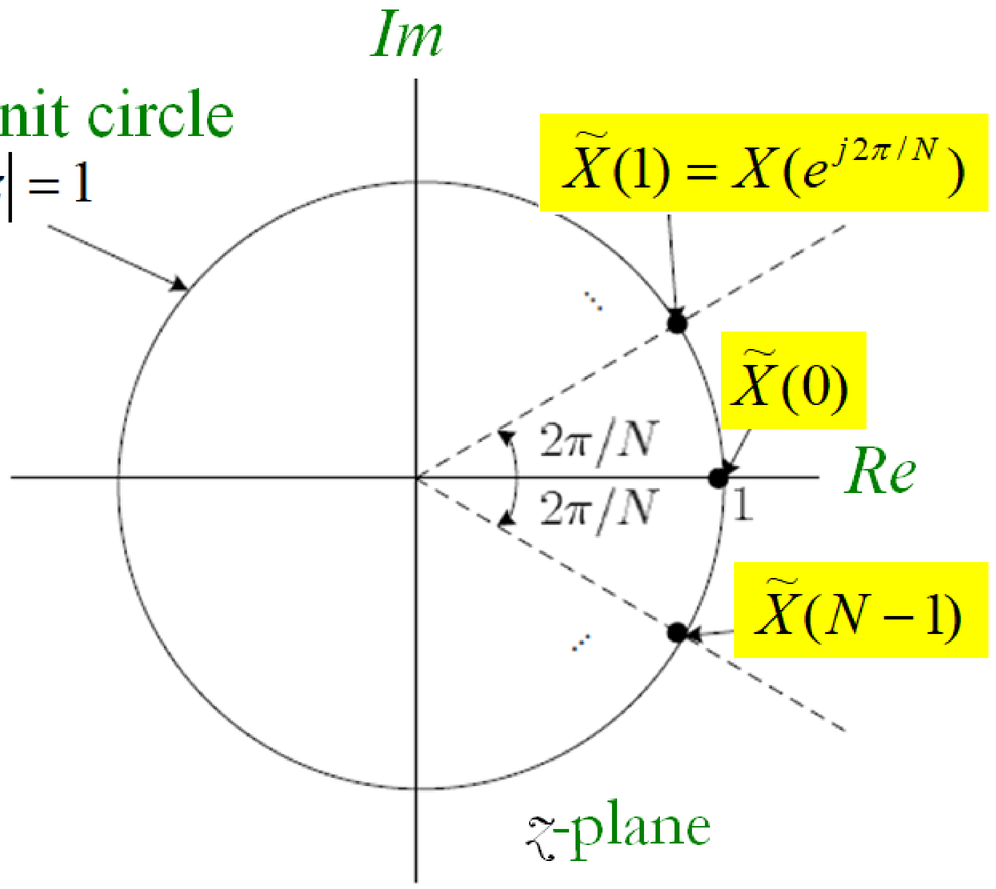
$$DFS \quad \tilde{X}(k) = X(e^{j\omega_k}) = X(e^{jk\omega_0})$$

$$\tilde{X}(k) = X(e^{j\omega})\Big|_{\omega=\frac{2\pi}{N}k}$$

$$\omega_0 \equiv \frac{2\pi}{N}, \quad \omega_k \equiv \frac{2\pi}{N}k = k\omega_0$$

The DFS *is obtained by evenly sampling the* DTFT *at* $\omega_0 = \dfrac{2\pi}{N}$ *intervals.*

Relation to Z-Transform & DTFT



Relationship between $\tilde{X}(k)$, $X(e^{j\omega})$, *and* $X(z)$

Homework #10.1 (1 pt.): Due Feb 7

Let $x(n) = \{\underset{\uparrow}{0}, \quad 1, \quad 2, \quad \text{A}\}$.

1. Compute its discrete-time Fourier transform $X(e^{j\omega})$.

2. Sample $X(e^{j\omega})$ at $k\omega_0 = \dfrac{2\pi}{4}k, \quad k = 0, 1, 2, 3.$

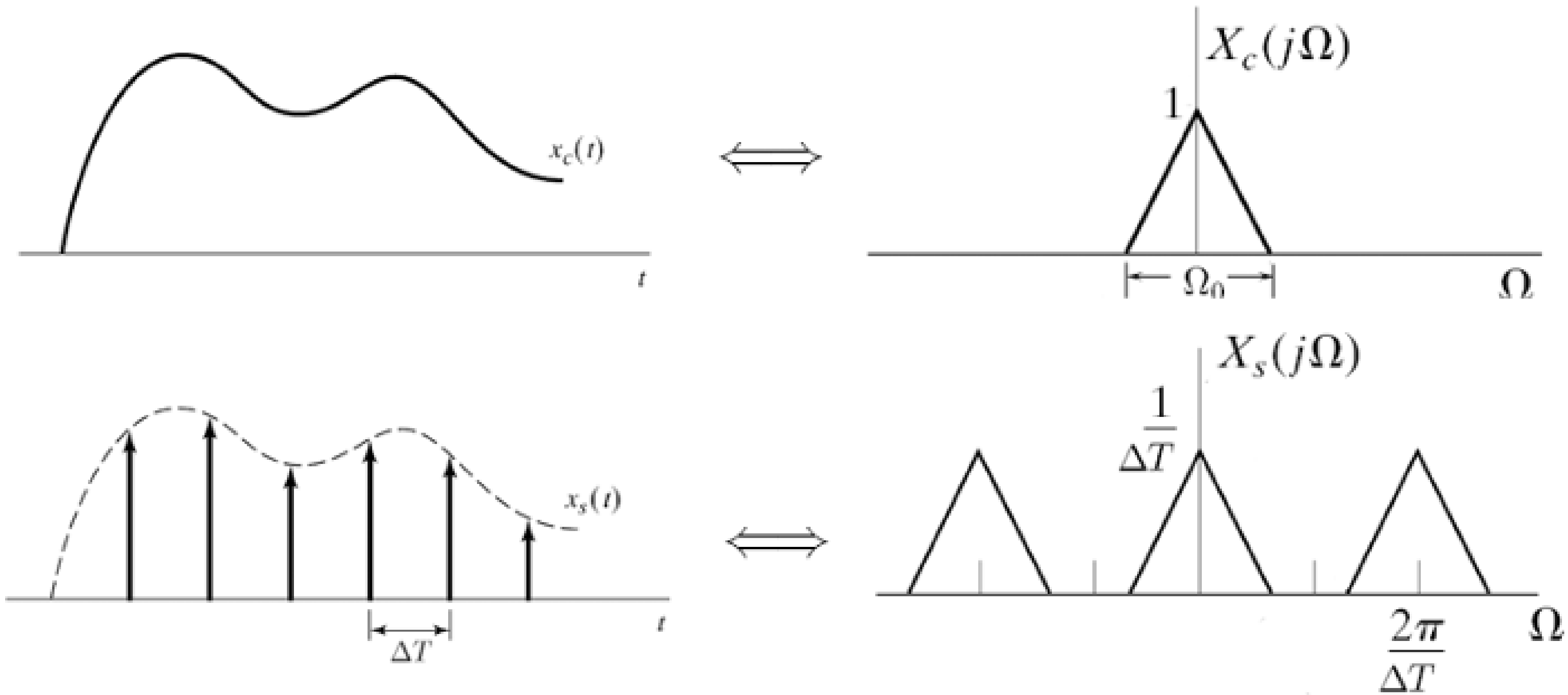
   Show that it is equal to $\tilde{X}(k)$.

Use Your ID: sGFEDCBA

Sampling & Reconstruction

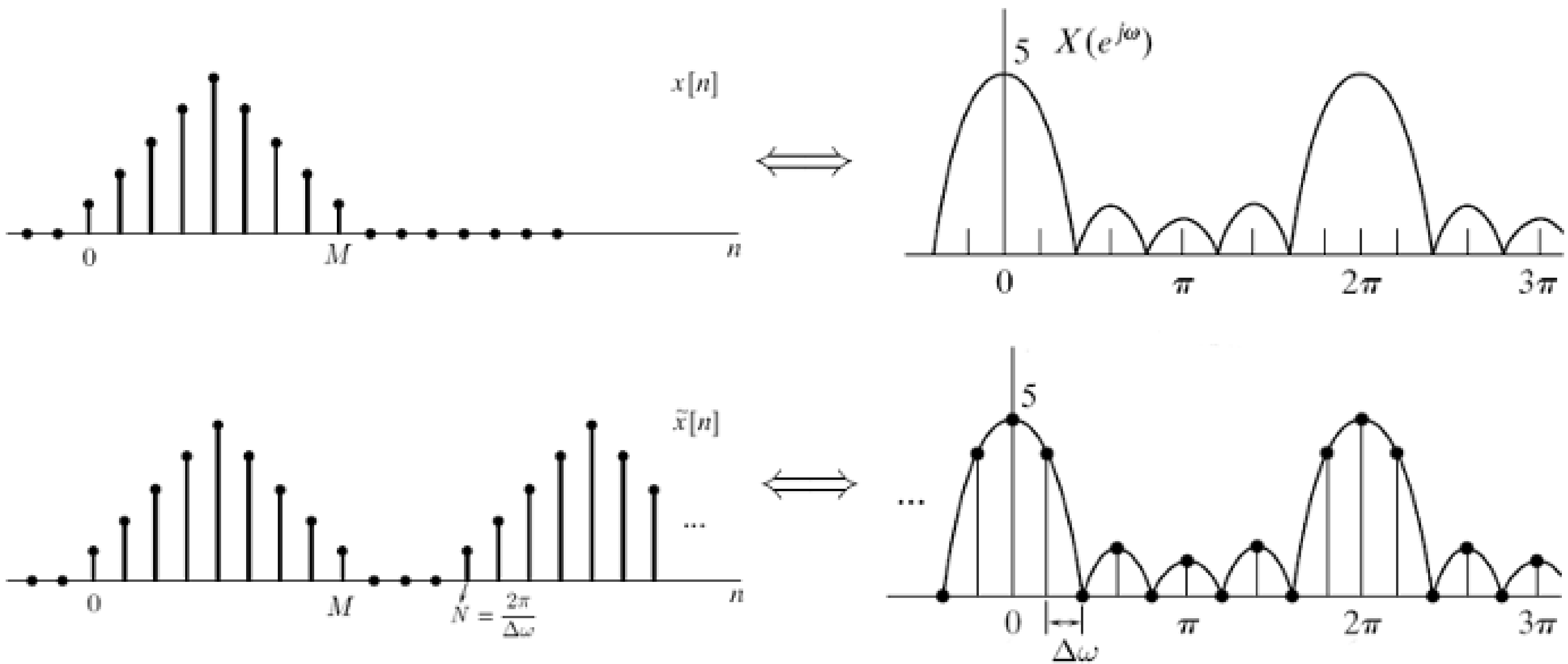
Sampling in Time Domain

- Sampling in time corresponds to replication in the frequency domain



Sampling in Frequency Domain

- Sampling in frequency corresponds to replication in the time domain



Sampling in Z-Domain & Reconstruction

Sampling in Z-Domain

$x(m)$ a finite-duration sequence of duration N

$$X(z) = \sum_{m=-\infty}^{\infty} x(m) z^{-m}$$

We sample X(z) on the unit circle at equispaced points separated in angle by $\omega_0 = 2\pi / N. \longrightarrow$ a DFS sequence

$$\tilde{X}(k) \equiv X(z)\Big|_{z=e^{j\frac{2\pi}{N}k}}, \quad k = 0, \pm 1, \pm 2, \ldots$$

$$= \sum_{m=-\infty}^{\infty} x(m) e^{-j\frac{2\pi}{N}km} = \sum_{m=-\infty}^{\infty} x(m) W_N^{km}$$

$$\tilde{x}(n) = IDFS\left[\tilde{X}(k)\right]$$

Sampling in Z-Domain & Reconstruction

- **Relation between the arbitrary $x(n)$ and the periodic $\tilde{x}(n)$.**

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

$$= \frac{1}{N}\sum_{k=0}^{N-1}\left\{\sum_{m=-\infty}^{\infty}x(m)W_N^{km}\right\}W_N^{-kn}$$

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

$$\tilde{x}(n) = \sum_{m=-\infty}^{\infty}x(m)\underbrace{\frac{1}{N}\sum_{0}^{N-1}W_N^{-k(n-m)}}_{=\begin{cases}1, & n-m=rN\\ 0, & elsewhere\end{cases}} = \sum_{m=-\infty}^{\infty}x(m)\sum_{r=-\infty}^{\infty}\delta(n-m-rN)$$

$$= \sum_{r=-\infty}^{\infty}\sum_{m=-\infty}^{\infty}x(m)\delta(n-m-rN)$$

$$\tilde{x}(n) = \sum_{r=-\infty}^{\infty}x(n-rN) = \cdots + x(n+N) + x(n) + x(n-N) + \cdots$$

Sampling in Z-Domain & Reconstruction

▪ Reconstruction

Sampling X(z) on the unit circle, we obtain a periodic sequence in the time domain. A linear combination of the original sequence x(n) and its infinite replicas, each shifted by multiples of $\pm N$.

▪ Aliasing in Time Domain

If $x(n) = 0$ *for* $n < 0$ *and* $n \geq N$, *then there will be no overlap or aliasing in the time domain.*

Able to recognize and recover $x(n)$ *from* $\tilde{x}(n)$,

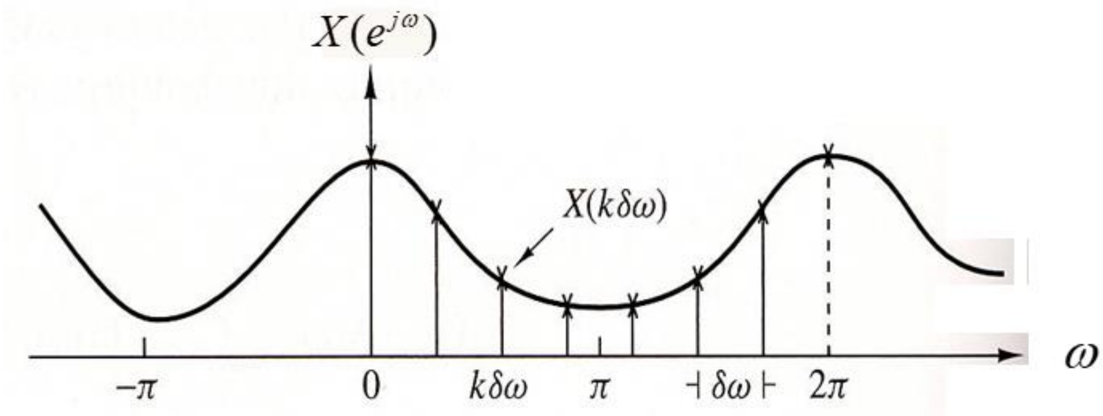
$$x(n) = \tilde{x}(n) \qquad for \qquad 0 \leq n \leq (N-1)$$

$$x(n) = \tilde{x}(n)\boxed{R_N(n)} = \begin{cases} \tilde{x}(n), & 0 \leq n \leq N-1 \\ 0, & else \end{cases}$$

A rectangular window of length N

Sampling in Frequency & Reconstruction

- ## Sampling DTFT



Frequency-domain sampling of the Fourier transform

N equidistance samples in the interval $0 \le \omega < 2\pi$ *with spacing* $\delta\omega = 2\pi / N$.

DTFT at $\omega = 2\pi k / N$,

$$X\left(e^{j\frac{2\pi}{N}k}\right) = \sum_{n=-\infty}^{\infty} x(n) e^{-j2\pi kn/N}, \quad k = 0, 1, \ldots, N-1$$

Sampling in Frequency & Reconstruction

- **Relation between the arbitrary $x(n)$ and the periodic $\tilde{x}(n)$.**

$$X\left(e^{j\frac{2\pi}{N}k}\right) = \cdots + \sum_{n=-N}^{-1} x(n)e^{-j2\pi kn/N} + \sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N}$$

$$+ \sum_{n=N}^{2N-1} x(n)e^{-j2\pi kn/N} + \cdots$$

$$= \sum_{r=-\infty}^{\infty} \sum_{n=rN}^{rN+N-1} x(n)e^{-j2\pi kn/N}$$

Changing the index from n to $n-rN$, and exchanging the order of the summation,

$$X\left(e^{j\frac{2\pi}{N}k}\right) = \sum_{n=0}^{N-1}\left[\boxed{\sum_{r=-\infty}^{\infty} x(n-rN)}\right]e^{-j2\pi kn/N}, \quad k = 0,1,2,\ldots,N-1.$$

$\tilde{x}(n)$

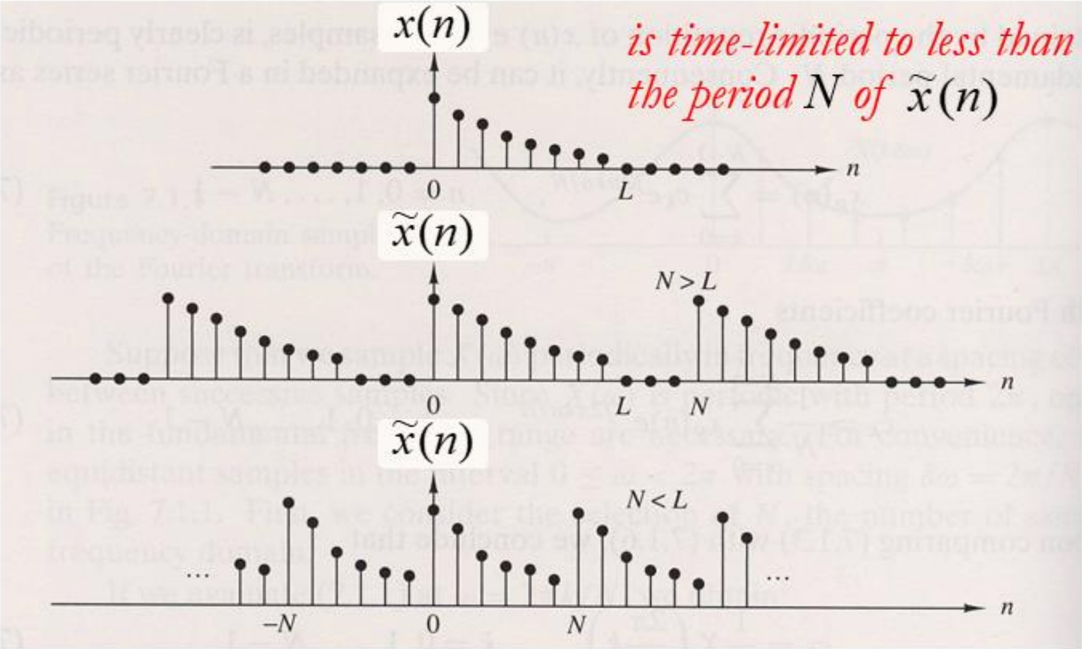
- *Periodic with fundamental period N*
- *Can be expanded in a Fourier series*

Sampling in Frequency & Reconstruction

- **Reconstruction**

$$\tilde{x}(n) = \frac{1}{N}\sum_{k=0}^{N-1} X\left(e^{j\frac{2\pi}{N}k}\right)e^{j2\pi kn/N}, \quad n = 0,1,\ldots,N-1$$

$$\boxed{\tilde{x}(n) \rightarrow x(n)}$$



Aperiodic sequence $x(n)$ of length L and its periodic extension $\tilde{x}(n)$ for $N \geq L$ and $N < L$.

no aliasing *aliasing in the time domain*

Sampling in Frequency & Reconstruction

- **Reconstruction**

The spectrum of an aperiodic discrete-time signal with finite duration L can be exactly recovered from its samples at frequencies $\omega_k = 2\pi k / N$, *if* $N \geq L$.

$$x(n) = \begin{cases} \tilde{x}(n), & 0 \leq n \leq N-1 \\ 0, & elsewhere \end{cases}$$

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

Example (1)

- **Aliasing in Time Domain**

$$x_1(n) = \{\underset{\uparrow}{6},5,4,3,2,1\}, \qquad \omega_k = \frac{2\pi k}{4}, \quad k = 0,\pm1,\pm2,\pm3,\ldots$$

$X_1(e^{j\omega})$ DTFT

$\widetilde{X}_2(k)$ DFS sequence

$\widetilde{x}_2(n)$ Inverse DFS

$$\widetilde{x}_2(n) = \sum_{r=-\infty}^{\infty} x_1(n-4r)$$

$$\widetilde{x}_2(n) = \{\ldots,8,6,4,3,\underset{\uparrow}{8},6,4,3,8,6,4,3,\ldots\}$$

$x(4)$ *is aliased into* $x(0)$, *and* $x(5)$ *is aliased into* $x(1)$.

Example (2)

Aliasing in Time Domain

$$x(n) = (0.7)^n u(n)$$

Sample its z-transform on the unit circle with N = 5, 10, 20, 50.

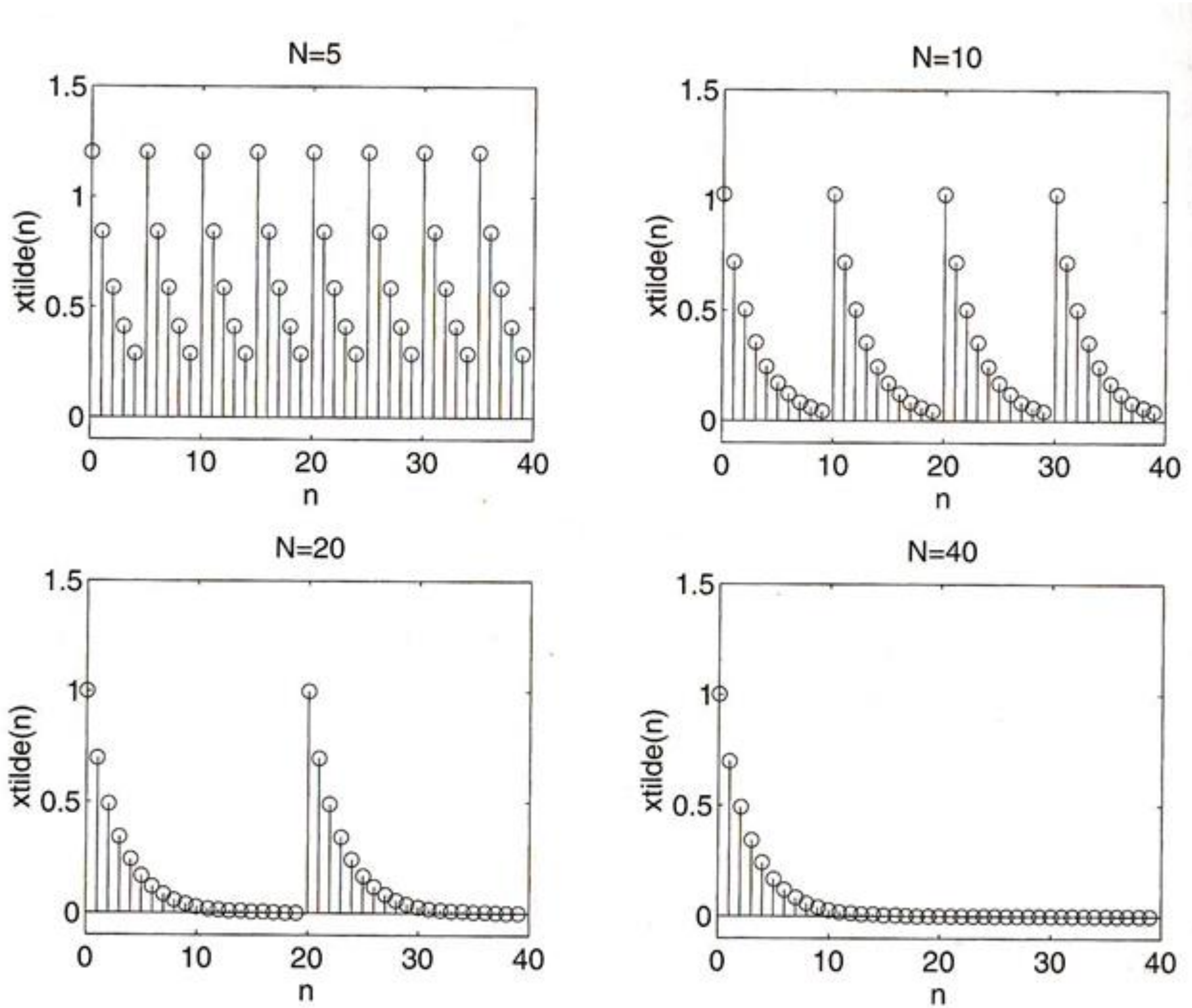
$$X(z) = \frac{1}{1-0.7z^{-1}} = \frac{z}{z-0.7}, \quad |z| > 0.7$$

$$\tilde{X}(k) = X(z)\Big|_{z=e^{j2\pi k/N}}, \quad k = 0, \pm 1, \pm 2, \ldots$$

```
>> N = 5; k = 0:1:N-1;                                % sample index
>> wk = 2*pi*k/N; zk = exp(j*wk);                     % samples of z
>> Xk = (zk)./(zk-0.7);                               % DFS as samples of X(z)
>> xn = real(idfs(Xk,N));                             % IDFS
>> xtilde = xn'* ones(1,8); xtilde = (xtilde(:))';    % Periodic sequence
>> subplot(2,2,1); stem(0:39,xtilde);axis([0,40,-0.1,1.5])
>> xlabel('n'); ylabel('xtilde(n)'); title('N=5')
```

Example (2)

- **Aliasing in Time Domain**



Aliasing for N=5 and N=10

Discrete Fourier Transform

Key Concept of Discrete Fourier Transform

1) The discrete Fourier transform (DFT) is used for computer-based frequency domain analysis of signals.

   a) The DFT requires signals be of finite duration, N.

   b) The DFT produces N frequency-domain coefficients.

2) A length-N DFT computes N samples of the discrete-time Fourier transform at frequency intervals of $\frac{2\pi}{N}$.

3) Sampling in frequency causes replication of the signal in the time domain at integer multiples of the DFT length.

4) Use of a length-N DFT implies the time signal is N-periodic.

DFS & DFT

- **DFT: From DFS**

  → the primary period of the DFS

  → the ultimate numerically computable Fourier transform for arbitrary finite-duration sequences

We define a finite-duration sequence $x(n)$ *that has* N *samples over* $0 \le n \le N-1$ *as an* N*-point sequence.*

$$\tilde{x}(n) = \sum_{r=-\infty}^{\infty} x(n-rN) \Rightarrow \tilde{x}(n) = x(n \bmod N)$$

$$x((n))_N \equiv x(n \bmod N)$$

$$\tilde{x}(n) = x((n))_N \qquad \textit{(Periodic extension)}$$

$$x(n) = \tilde{x}(n) R_N(n) \qquad \textit{(Window operation)}$$

```
function m = mod(n,N)
% Computes m = (n mod N) index
% ------------------------------------
% m = mod(n,N)
m = rem(n,N); m = m+N; m = rem(m,N)
```

Example

$$x(n) = \left\{ \underset{\uparrow}{1}, \quad 2, \quad 3, \quad 4 \right\}$$

$$((n))_m$$

$$qm + R$$

$$y(n) = x((n-1))_4$$

$$y(0) = x((-1))_4 = x(3) = 4$$

$$y(1) = x((0))_4 = x(0) = 1$$

$$y(2) = x((1))_4 = x(1) = 2$$

$$y(3) = x((2))_4 = x(2) = 3$$

$$y(n) = x((n+2))_4$$

$$y(0) = x((2))_4 = x(2) = 3$$

$$y(1) = x((3))_4 = x(3) = 4$$

$$y(2) = x((4))_4 = x(0) = 1$$

$$y(3) = x((5))_4 = x(1) = 2$$

DFS & DFT

DFT: From DFS

DFS is practically equivalent to the DFT when $0 \le n \le N-1$.

$DFS \quad \tilde{x}(n)$

The Discrete Fourier Transform of an N-point sequence

$$X(k) \equiv DFT[x(n)] = \begin{cases} \tilde{X}(k), & 0 \le k \le N-1 \\ 0, & elsewhere \end{cases} = \tilde{X}(k) R_N(k)$$

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk}, \qquad 0 \le k \le N-1$$

The inverse Discrete Fourier Transform of an N-point DFT X(k)

$$x(n) \equiv IDFT[X(k)] = \tilde{x}(n) R_N(n)$$

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{-kn}, \qquad 0 \le n \le N-1$$

(*Not defined outside* $0 \le n \le N-1$)

DFS & DFT

- ## DFT: From DFS

DFS is practically equivalent to the DFT when $0 \le n \le N-1$.

```
Function [Xk] = dft(xn,N)
% Compute Discrete Fourier Transform
% -----------------------------------------------
% [Xk] = dft(xn,N)
% Xk = DFT coeff. array over 0 <= k <= N-1
%  xn = N-point finite-duration sequence
%   N = Length of DFT
%
n = [0:1:N-1];              % row vector for n
k = [0:1:N-1];              % row vector for k
WN = exp(-j*2*pi/N);        % Wn factor
nk = n'*k;                  % creates an N by N matrix of nk values
WNnk = WN .^ nk;            % DFT matrix
Xk = xn * WNnk;             % row vector for DFT coefficients

function [xn] = idft(Xk,N)
% Compute Inverse Discrete Fourier Transform
% ---------------------------------------------------
% [xn] = idft(Xk,N)
%  xn = N-point sequence over 0 <= n <= N-1
% Xk =  DFT coeff. array over 0 <= n <= N-1
%   N =  Length of DFT
%
n = [0:1:N-1];              % row vector for n
k = [0:1:N-1];              % row vector for k
WN = exp(-j*2*pi/N);        % Wn factor
nk = n'*k;                  % creates an N by N matrix of nk values
WNnk = WN .^ (-nk);         % IDFT matrix
xn = (Xk * WNnk)/N;         % row vector for IDFT values
```

DTFT & DFT

- **DTFT:**

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

- **Sampling ω at $\omega_k = \frac{2\pi}{N}k, k = 0,1,2,\ldots,N-1$**

Define DFT

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

Inverse DFT

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

DTFT & DFT

- **DFT:**

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

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

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

$$\begin{pmatrix} X(0) \\ X(1) \\ X(2) \\ \vdots \\ X(N-1) \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & W & W^2 & W^3 & \cdots & W^{N-1} \\ 1 & W^2 & W^4 & W^6 & \cdots & W^{N-2} \\ 1 & W^3 & W^6 & W^9 & \cdots & W^{N-3} \\ \vdots & & & & & \\ 1 & W^{N-1} & W^{N-2} & W^{N-3} & \cdots & W \end{pmatrix} \begin{pmatrix} x(0) \\ x(1) \\ x(2) \\ \vdots \\ x(N-1) \end{pmatrix}$$

Example (1):

- **DTFT**

$$x(n) = \begin{cases} 1, & 0 \le n \le 3 \\ 0, & otherwise \end{cases}$$

$$X(e^{j\omega}) = \sum_{0}^{3} x(n) e^{-j\omega n}$$

$$\sum_{0}^{3} x(n) e^{-j\omega n} = e^{-j0\omega} + e^{-j1\omega} + e^{-j2\omega} + e^{-j3\omega}$$

$$= \left(e^{j\frac{3}{2}\omega} + e^{j\frac{1}{2}\omega} + e^{-j\frac{1}{2}\omega} + e^{-j\frac{3}{2}\omega} \right) e^{-j\frac{3}{2}\omega}$$

$$= 2\left(\cos\frac{3}{2}\omega + \cos\frac{1}{2}\omega \right) e^{-j\frac{3}{2}\omega}$$

$$= \frac{\sin(2\omega)}{\sin(\omega/2)} e^{-j\frac{3}{2}\omega}$$

$$2\left(\cos\frac{3}{2}\omega + \cos\frac{1}{2}\omega \right)$$

$$= 2\left(\cos\left(\omega + \frac{\omega}{2}\right) + \cos\left(\omega - \frac{\omega}{2}\right) \right)$$

$$= 2\left(2\cos\omega \cos\frac{\omega}{2} \right)$$

$$= 2\left(2\sin\omega \cos\omega \frac{\cos(\omega/2)}{\sin\omega} \right)$$

$$= 2\left(\sin 2\omega \frac{1}{2\sin(\omega/2)} \right)$$

$$= \frac{\sin 2\omega}{\sin(\omega/2)}$$

Example (1):

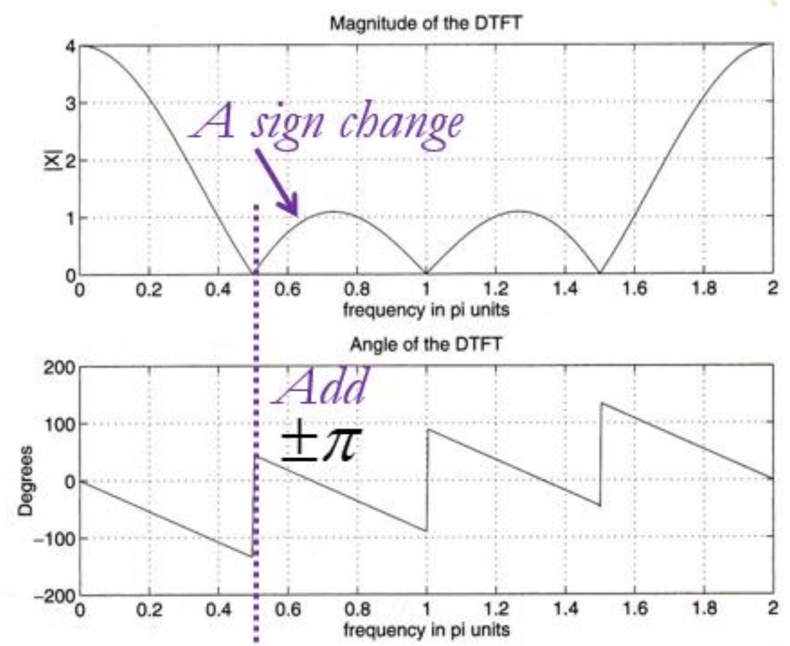
- **DTFT**

$$x(n)=\begin{cases}1, & 0\le n\le 3\\ 0, & otherwise\end{cases}$$

$$X(e^{j\omega})=\sum_{0}^{3}x(n)e^{-j\omega n}=\frac{\sin(2\omega)}{\sin(\omega/2)}e^{-j3\omega/2}$$

$$\left|X(e^{j\omega})\right|=\left|\frac{\sin(2\omega)}{\sin(\omega/2)}\right|$$

$$\angle X(e^{j\omega})=\begin{cases}-\dfrac{3\omega}{2}, & when\quad \dfrac{\sin(2\omega)}{\sin(\omega/2)}>0\\ -\dfrac{3\omega}{2}\pm\pi, & when\quad \dfrac{\sin(2\omega)}{\sin(\omega/2)}<0\end{cases}$$



Frequency response: magnitude and phase

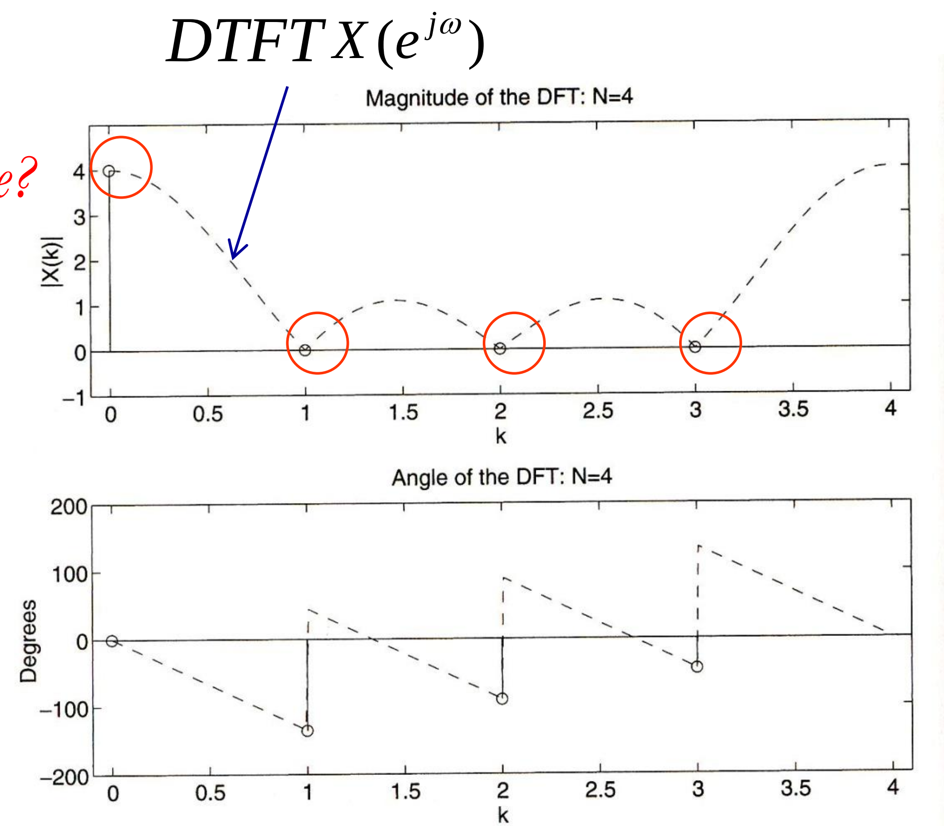
Example (1):

- **DFT: Sampling DTFT**

$$X_4(k) = \sum_{n=0}^{3} x(n) W_4^{nk}; \quad k = 0,1,2,3; \quad W_4 = e^{-j2\pi/4} = -j$$

$$X_4(k) = \left\{ \underset{\uparrow}{4}, \quad 0, \quad 0, \quad 0 \right\}$$

Why only one non-zero sample?




```
>> x = [1,1,1,1]; N = 4; X = dft(x,N);
>> magX = abs(X), phaX = angle(X)*180/pi
magX =
      4.0000      0.0000      0.0000      0.0000
phaX =
           0   -134.9810   -90.0000   -44.9979
```

Example (1):

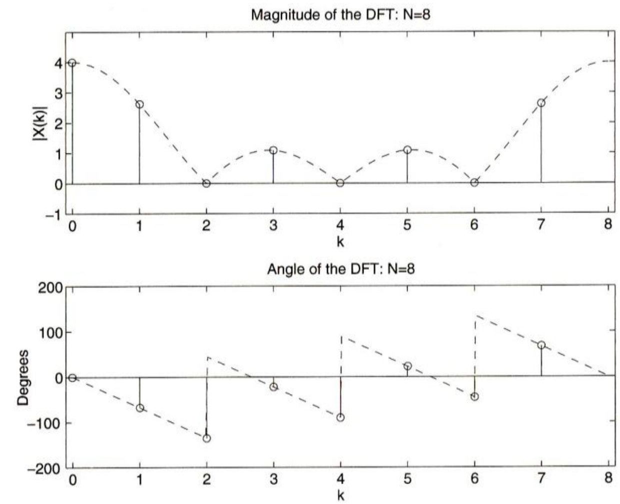
- **DFT: Sampling DTFT**

How can we obtain other samples of the DTFT?

– at dense (finer) frequencies

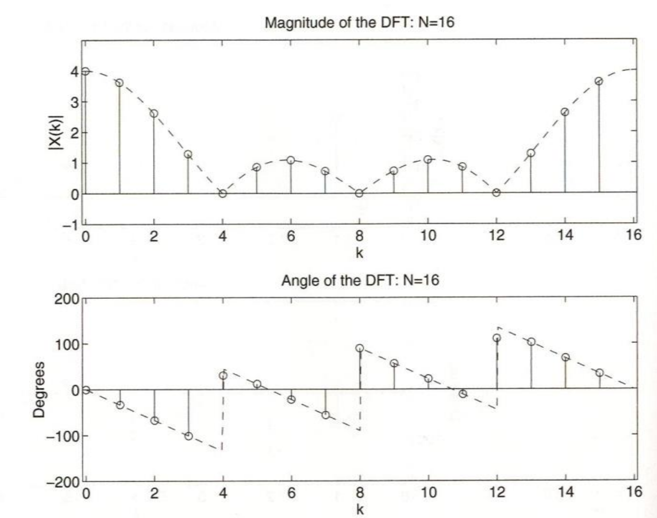
$$x(n)=\{\underset{\uparrow}{1},\ 1,\ 1,\ 1,\ \boxed{0,\ 0,\ 0,\ 0}\}$$

$$W_8=e^{-j\pi/4},\quad \omega_0=2\pi/8=\pi/4$$



$$x(n)=\{\underset{\uparrow}{1},\ 1,\ 1,\ 1,\ \boxed{0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0}\}$$

$$W_{16}=e^{-j\pi/8},\quad \omega_0=2\pi/16=\pi/8$$



zero-padding operation: more zeros are appended to the original sequence

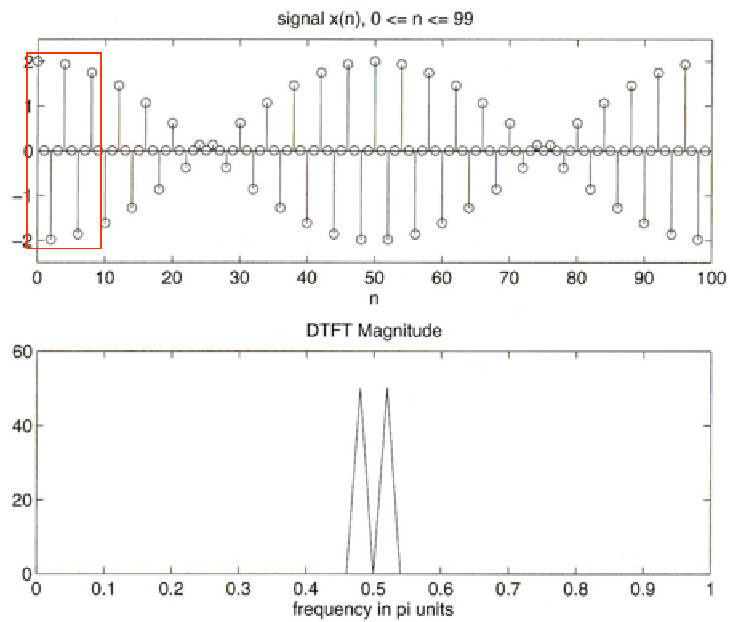
Example (2):

$$x(n) = \cos(0.48\pi n) + \cos(0.52\pi n)$$

Determine and plot the discrete-time Fourier transform of $x(n), \quad 0 \le n \le 100.$

```
>> subplot(2,1,1); stem(n,x);
>> title('signal x(n), 0 <= n <= 99'); xlabel('n')
>> X = dft(x,100); magX = abs(X(1:1:51));
>> k = 0:1:50; w = 2*pi/100*k;
>> subplot(2,1,2); plot(w/pi,magX); title('DTFT Magnitude');
>> xlabel('frequency in pi units')
```

One has to obtain more data from the experiment or observations



High-resolution spectrum: clearly shows two frequencies, which are very close to each other

Properties of DFT

Properties of the DFT

1. Linearity *The DFT is a linear transform*

$$DFT[ax_1(n)+bx_2(n)]=aDFT[x_1(n)]+bDFT[x_2(n)]$$

2. Circular folding

$$x((-n))_N=\begin{cases} x(0), & n=0 \\ x(N-n), & 1\le n\le N-1 \end{cases}$$

$$DFT[x((-n))_N]=X((-k))_N=\begin{cases} X(0), & k=0 \\ X(N-k), & 1\le k\le N-1 \end{cases}$$

Proof:

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

$$X(k)=\sum_{m=N}^{1}x(N-m)W_N^{(N-m)k}$$

$$=\sum_{m=1}^{N}x(N-m)W_N^{(N-m)k}$$

$$=\sum_{m=0}^{N-1}x(N-m)W_N^{(N-m)k}$$

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

$$=\sum_{n=0}^{N-1}x(N-n)W_N^{Nk}W_N^{-nk}$$

$$=\sum_{n=0}^{N-1}x(N-n)W_N^{-nk}$$

$$X(N-k)=\sum_{n=0}^{N-1}x(N-n)W_N^{-n(N-k)}$$

$$=\sum_{n=0}^{N-1}x(N-n)W_N^{-nN}W_N^{nk}$$

$$=\sum_{n=0}^{N-1}x(N-n)W_N^{nk}$$

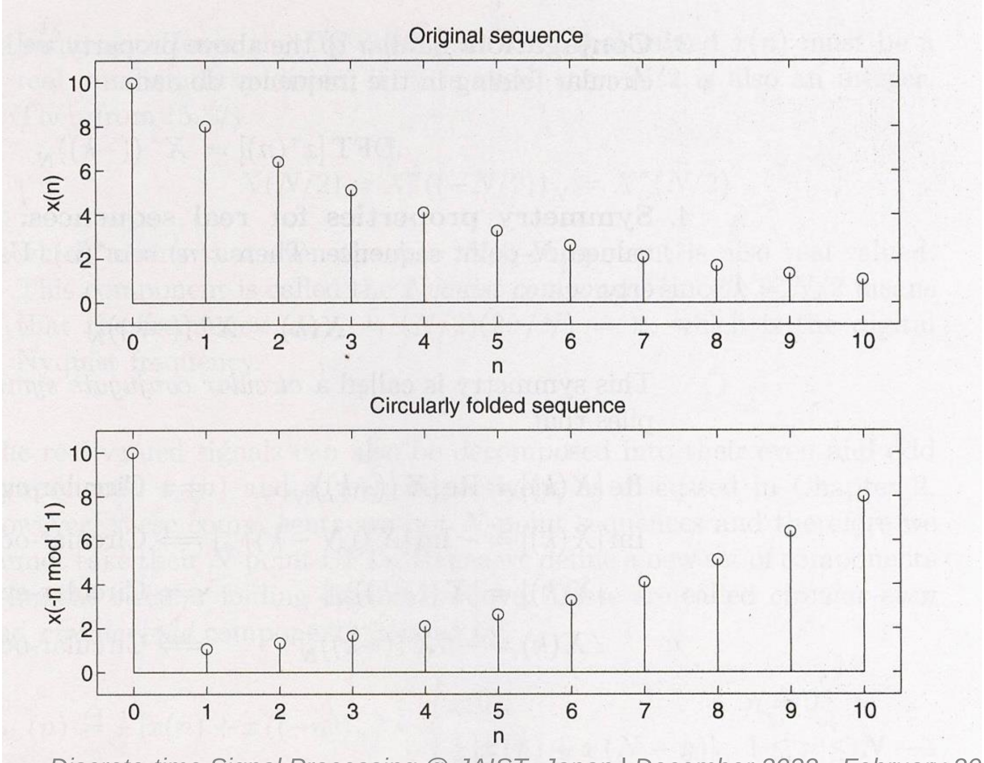
$$\boxed{\begin{aligned}&DFT\left[x(N-n)\right]=X(N-k)\\&DFT\left[x((-n))_N\right]=X((-k))_N\end{aligned}}$$

Circular folding (time reversal)

Example: $x(n) = 10(0.8)^n, \quad 0 \le n \le 10.$

Determine and plot $x((-n))_{11}$.

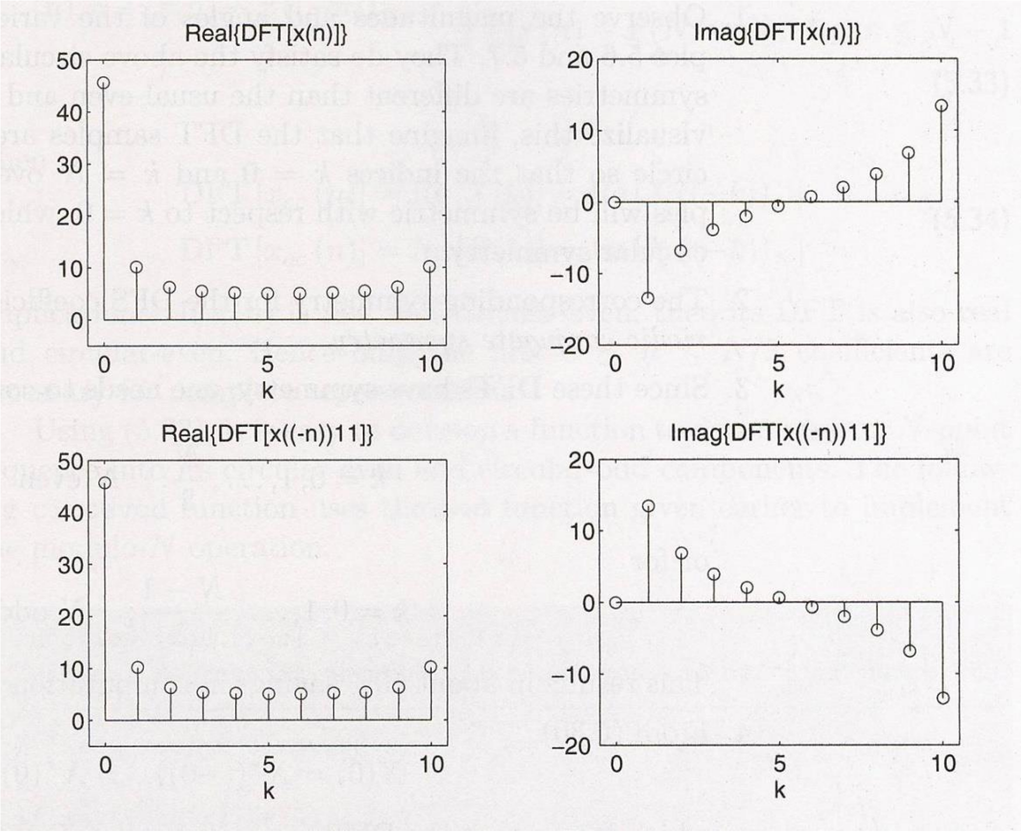
```
>> n = 0:10; x = 10*(0.8) .^ n;  y = x(mod(-n,11)+1);
>> subplot(2,1,1); stem(n,x); title('Original sequence')
>> xlabel('n'); ylabel('x(n)');
>> subplot(2,1,2); stem(n,y); title('Circularly folded sequence')
>> xlabel('n'); ylabel('x(-n mod 11)');
```



Example:

Verify the circular folding property.

```
>> X = dft(x,11); Y = dft(y,11);
>> subplot(2,2,1); stem(n,real(X));
>> title('Real{DFT[x(n)]}'); xlabel('k');
>> subplot(2,2,2); stem(n,imag(X));
>> title('Imag{DFT[x(n)]}'); xlabel('k');
>> subplot(2,2,3); stem(n,real(Y));
>> title('Real{DFT[x((-n))11]}'); xlabel('k');
>> subplot(2,2,4); stem(n,imag(Y));
>> title('Imag{DFT[x((-n))11]}'); xlabel('k');
```



3. Conjugation *the circular folding in the frequency domain*

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

4. Symmetry properties for real sequence

$$X(k) = X^*((-k))_N$$

circular conjugate symmetry

$$\mathrm{Re}[X(k)] = \mathrm{Re}[X((-k))_N]$$ *circular-even sequence*

$$\mathrm{Im}[X(k)] = -\mathrm{Im}[X((N-k))_N]$$ *circular-odd sequence*

$$|X(k)| = |X((-k))_N|$$ *circular-even sequence*

$$\angle X(k) = -\angle X((-k))_N$$ *circular-odd sequence*

The DFT *samples are arranged around a circle so that the indices* $k=0$ *and* $k=N$ *overlap. Symmetric w.r.t.* $k=0$.

Proof: Conjugation

$$DFT\left[x^*(n)\right] = \sum_{n=0}^{N-1} x^*(n) e^{-j\frac{2\pi}{N}nk}$$

$$= \left[\sum_{n=0}^{N-1} x(n) e^{j\frac{2\pi}{N}nk}\right]^*$$

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

$$= \left[X((-k))_N\right]^*$$

$$= X^*((-k))_N$$

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

$$IDFT\left[X^*(k)\right] = \frac{1}{N}\sum_{k=0}^{N-1} X^*(k) e^{j\frac{2\pi}{N}nk}$$

$$= \left[\frac{1}{N}\sum_{k=0}^{N-1} X(k) e^{-j\frac{2\pi}{N}nk}\right]^*$$

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

$$= \left[x((-n))_N\right]^*$$

$$= x^*((-n))_N$$

$$= x^*(N-n)$$

Proof: Periodic & Symmetry

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk}, \qquad W_N = e^{-j\frac{2\pi}{N}}$$

$$= \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi nk}{N}}$$

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

$$= \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi nk}{N}} e^{-j\frac{2\pi nN}{N}}$$

$$= \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi nk}{N}} e^{-j2\pi n}$$

$$= \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi nk}{N}}$$

$$= X(k)$$

Periodic

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

$$X^*(k) = \sum_{n=0}^{N-1} x^*(n) W_N^{-nk}$$

$$= \sum_{n=0}^{N-1} x(n) W_N^{-nk}$$

$$= \sum_{n=0}^{N-1} x(n) W_N^{-nk} W_N^{nN}$$

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

$$= X(N-k)$$

Symmetry

Proof: Circular-even and odd

circular-even

$$x_{ec}(n) \equiv \frac{1}{2}\left[x(n) + x((-n))_N\right] = \begin{cases} x(0), & n = 0 \\ \frac{1}{2}\left[x(n) + x(N-n)\right], & 1 \le n \le N-1 \end{cases}$$

circular-odd

$$x_{oc}(n) \equiv \frac{1}{2}\left[x(n) - x((-n))_N\right] = \begin{cases} 0, & n = 0 \\ \frac{1}{2}\left[x(n) - x(N-n)\right], & 1 \le n \le N-1 \end{cases}$$

$$DFT[x_{ec}(n)] = \mathrm{Re}[X(k)] = \mathrm{Re}[X((-k))_N]$$

$$DFT[x_{oc}(n)] = \mathrm{Im}[X(k)] = \mathrm{Im}[X((-k))_N]$$

```
function [xec, xoc] = circevod(x)
% signal decomposition into circular-even and circular-odd parts
% -----------------------------------------------------------------------
% [xec, xoc] = circevod(x)
%
if any(img(x)    = 0)
        error('x~is not a real sequence')
end
N = length(x); n = 0:(N-1)
xec = 0.5*(x + x(mod(-n,N)+1)); xoc = 0.5 *(x – x(mod(-n,N)+1));
```

Proof: Circular-even and odd

$$X(k) = A + jB, \quad X^*(k) = A - jB$$

$$DFT\left[x_{ec}(n)\right] = \frac{1}{2}\left\{DFT\left[x(n)\right] + DFT\left[x((-n))_N\right]\right\}$$

$$= \frac{1}{2}\left\{X(k) + X((-k))_N\right\}$$

$$= \frac{1}{2}\left\{X(k) + X^*(k)\right\}$$

$$= A$$

$$DFT\left[x_{oc}(n)\right] = \frac{1}{2}\left\{DFT\left[x(n)\right] - DFT\left[x((-n))_N\right]\right\}$$

$$= \frac{1}{2}\left\{X(k) - X((-k))_N\right\}$$

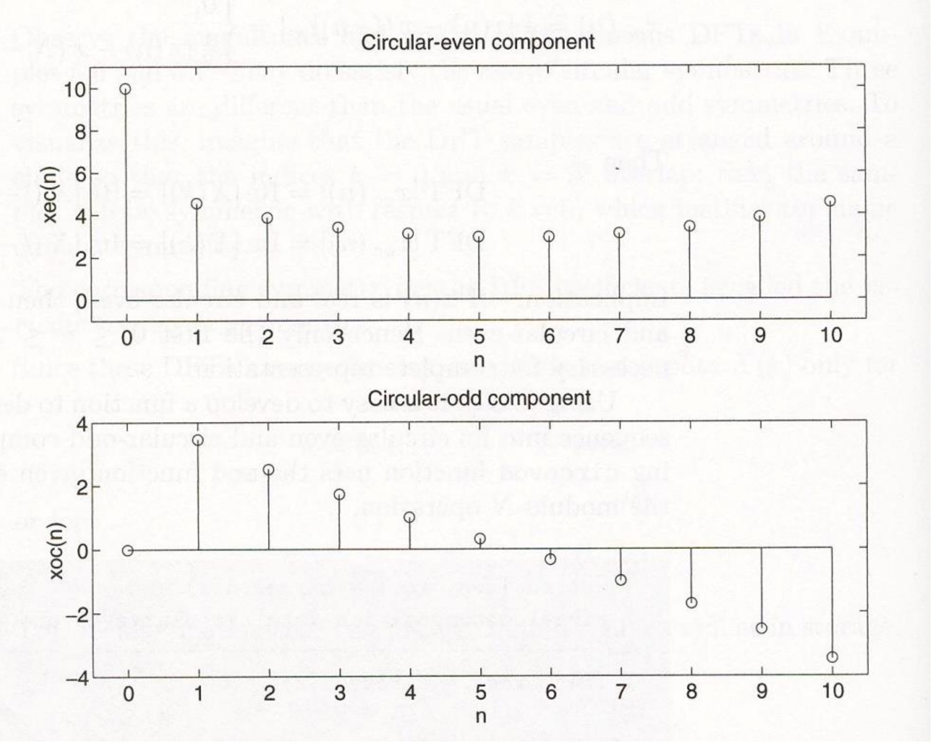
$$= \frac{1}{2}\left\{X(k) - X^*(k)\right\}$$

$$= jB$$

Example: $x(n) = 10(0.8)^n, \quad 0 \le n \le 10.$

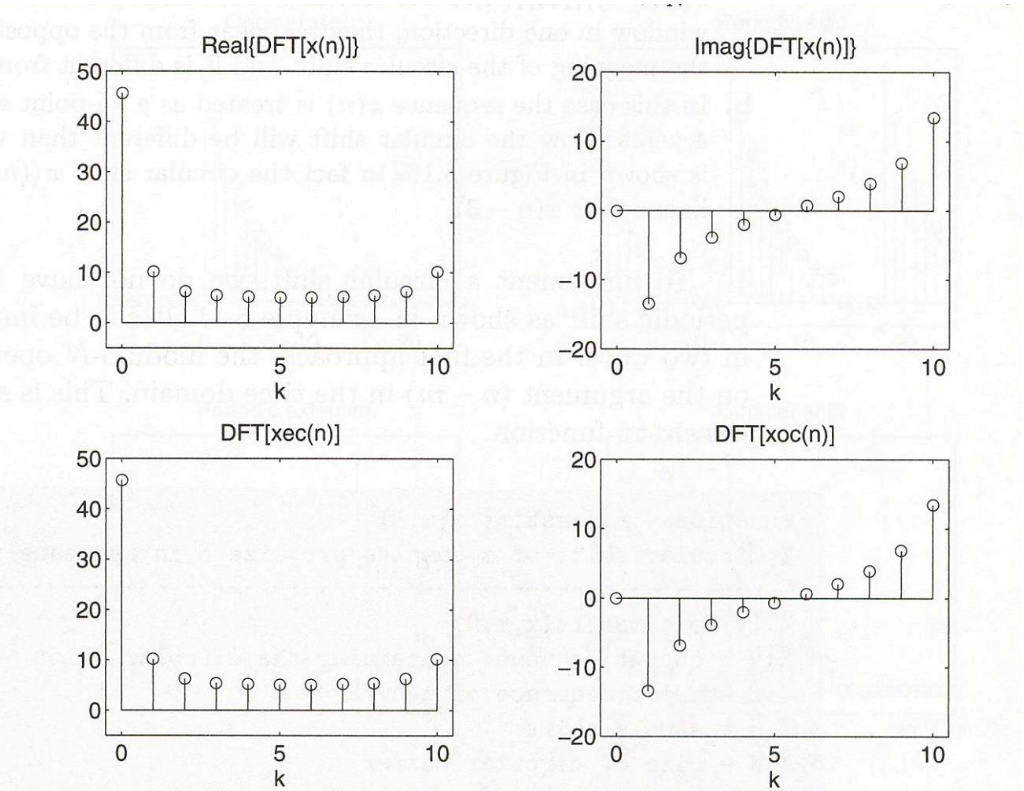
Decompose and plot the $x_{ec}(n)$ *and* $x_{oc}(n)$ *components.*

```
>> n = 0:10; x = 10*(0.8) .^ n;  y = x(mod(-n,11)+1);
>> [xec,xoc] =circevod(x);
>> subplot(2,1,1); stem(n,xec); title('Circular-even component')
>> xlabel('n'); ylabel('xec(n)'); axis([-0.5,10.5,-1,11])
>> subplot(2,1,2); stem(n,xoc); title('Circular-odd component')
>> xlabel('n'); ylabel('xoc(n)'); axis([-0.5,10.5,-4,4])
```



Example:

```
>> X = dft(x,11); Xec = dft(xec,11); Xoc = dft(xoc,11);
>> subplot(2,2,1); stem(n,real(X)); axis([-0.5,10.5,-5,50])
>> title('Real{DFT[x(n)]}'); xlabel('k');
>> subplot(2,2,2); stem(n,imag(X));  axis([-0.5,10.5,-20,20])
>> title('Imag{DFT[x(n)]}'); xlabel('k');
>> subplot(2,2,3); stem(n,real(Xec)); axis([-0.5,10.5,-5,50])
>> title('DFT[xec(n)]'); xlabel('k');
>> subplot(2,2,4); stem(n,imag(Xoc));  axis([-0.5,10.5,-20,20])
>> title('DFT[xoc(n)]'); xlabel('k');
```



5. Circular shift of a sequence

$$\tilde{x}(n-m) = x((n-m))_N \quad \textit{a periodic shift of } \tilde{x}(n)$$

$$\tilde{x}(n-m)R_N(n) = x((n-m))_N R_N(n) \quad \textit{circular shift of } x(n)$$

$$DFT\left[x((n-m))_N R_N(n)\right] = W_N^{km} X(k)$$

The sequence $x(n)$ *is wrapped around a circle.*

Rotate the circle by k *samples.*

Unwrap the sequence from $0 \le n \le N-1$.

6. Circular shift in the frequency domain

$$DFT\left[W_N^{-\ln} x(n)\right] = X((k-l))_N R_N(k)$$

Recall Properties of DTFT (2): Time Shifting

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

$$x[k] = x[k-m]$$

$$\sum_{k=-\infty}^{\infty} x[k-m]e^{-j\omega k}, \quad k-m=n$$

$$= \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega(n+m)}$$

$$= \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n}e^{-j\omega m}$$

$$= X(e^{j\omega})e^{-j\omega m}$$

Properties of DTFT (3): Frequency Shifting

$$F[x(k)e^{j\omega_0 k}] = X(e^{j(\omega-\omega_0)})$$

$$x[k] = x[k]e^{j\omega_0 k}$$

$$\sum_{k=-\infty}^{\infty} x[k]e^{j\omega_0 k}e^{-j\omega k}$$

$$= \sum_{k=-\infty}^{\infty} x[k]e^{-j(\omega-\omega_0)k}$$

$$= X(e^{j(\omega-\omega_0)})$$

Proof: Circular time & frequency shift

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

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

$$x(n-m) = \frac{1}{N}\sum_{k=0}^{N-1} X(k) W_N^{-k(n-m)}$$

$$X(k-l) = \sum_{k=0}^{N-1} x(n) W_N^{(k-l)n}$$

$$x((n-m))_N = \frac{1}{N}\left\{\sum_{k=0}^{N-1} X(k) W_N^{km}\right\} W_N^{-kn}$$

$$= \left\{\sum_{k=0}^{N-1} x(n) W_N^{-ln}\right\} W_N^{kn}$$

$$= IDFT\left[X(k) W_N^{km}\right]$$

$$X((k-l))_N = DFT\left[W_N^{-\ln} x(n)\right]$$

$$\boxed{DFT\left[x((n-m))_N\right] = W_N^{km} X(k)}$$

$$\boxed{DFT\left[W_N^{-\ln} x(n)\right] = X((k-l))_N}$$

Circular time shift

Circular frequency shift

Example:

$$x(n) = 10(0.8)^n, \quad 0 \le n \le 10.$$

$x((n+4))_{11} R_{11}(n)$ *a circular shift by* 4 *samples toward the left*

$x((n-3))_{15} R_{15}(n)$ *a circular shift by* 3 *samples toward the right*

```
function y = cirshftt(x,m,N)
% Circular shift of m samples wrt size N in sequence x: (time domain)
% -----------------------------------------------------------------------
% [y] = cirshftt(x,m,N)
% y = output sequence containing the circular shift
% x = input sequence of length <= N
% m = sample shift
% N = size of circular buffer
%   Method: y(n) = x((n-m) mod N)
% Check for the length of x
if length(x) > N
        error('N must be >= the length of x')
end
x = [x zeros(1,N-length(x))];
n = [0:1:N-1]; n = mod(n-m,N); y = x(n+1);
```

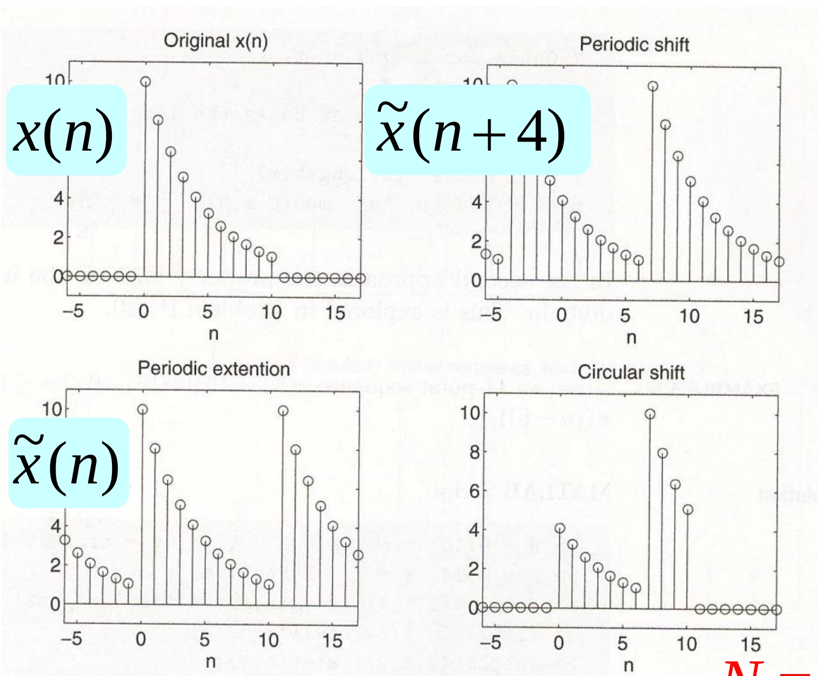
Example:

$$x(n) = 10(0.8)^n, \quad 0 \le n \le 10.$$

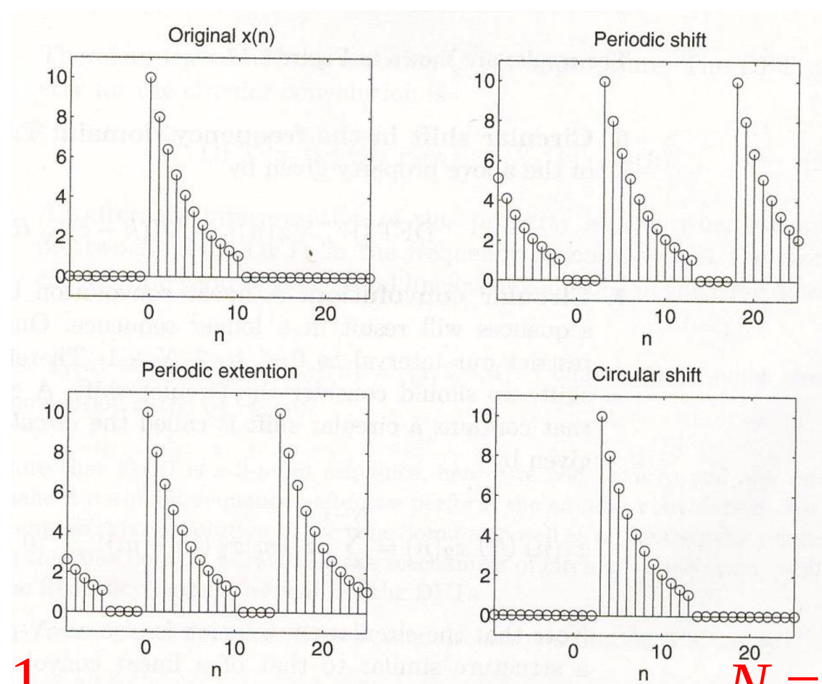
$x((n+4))_{11} R_{11}(n)$ *a circular shift by* 4 *samples toward the left*

$x((n-3))_{15} R_{15}(n)$ *a circular shift by* 3 *samples toward the right*

$x\big((n+4)\big)_{11} R_{11}(n)$ versus $x\big((n+4)\big)_{15} R_{15}(n)$

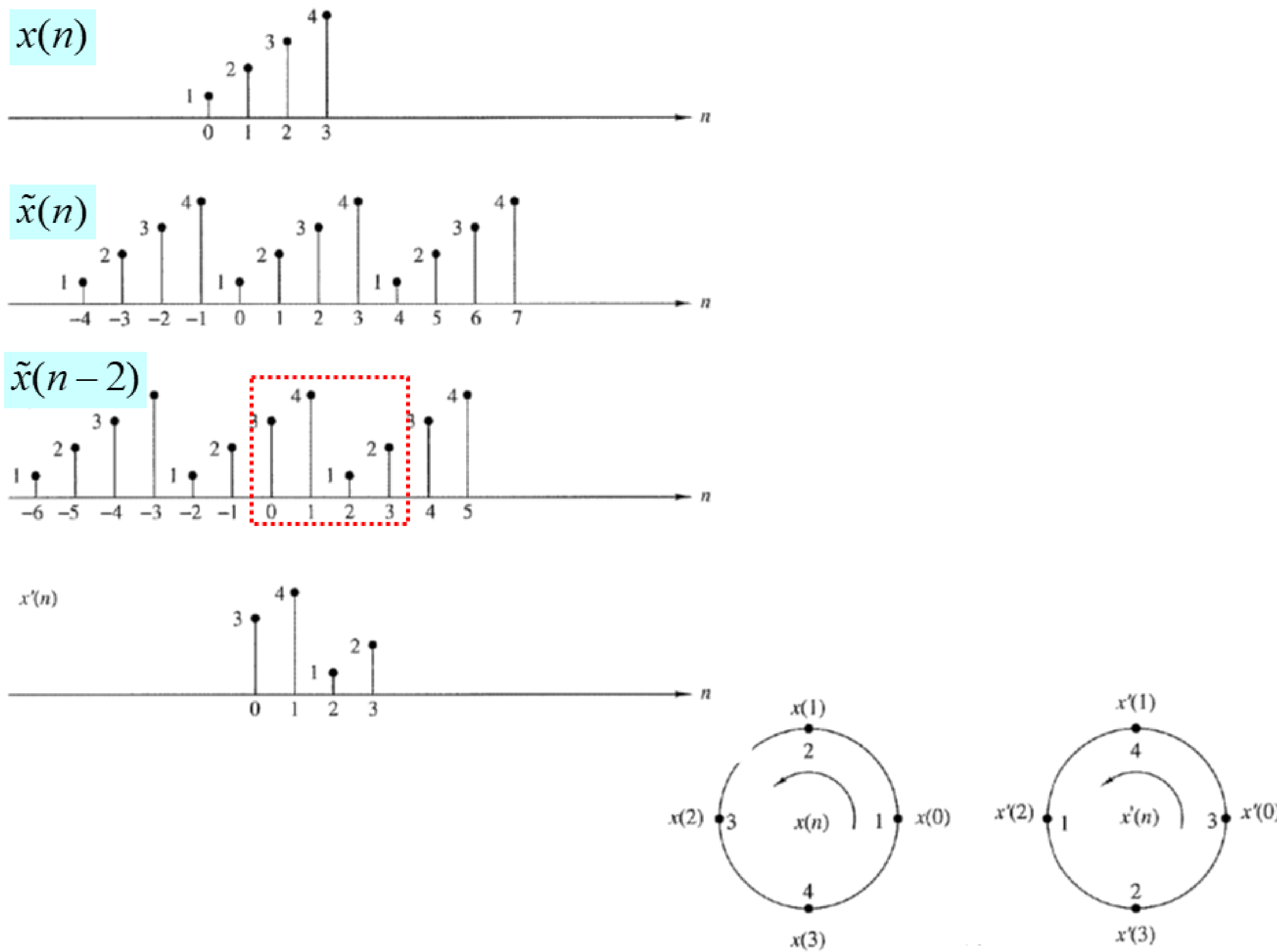


$N = 11$



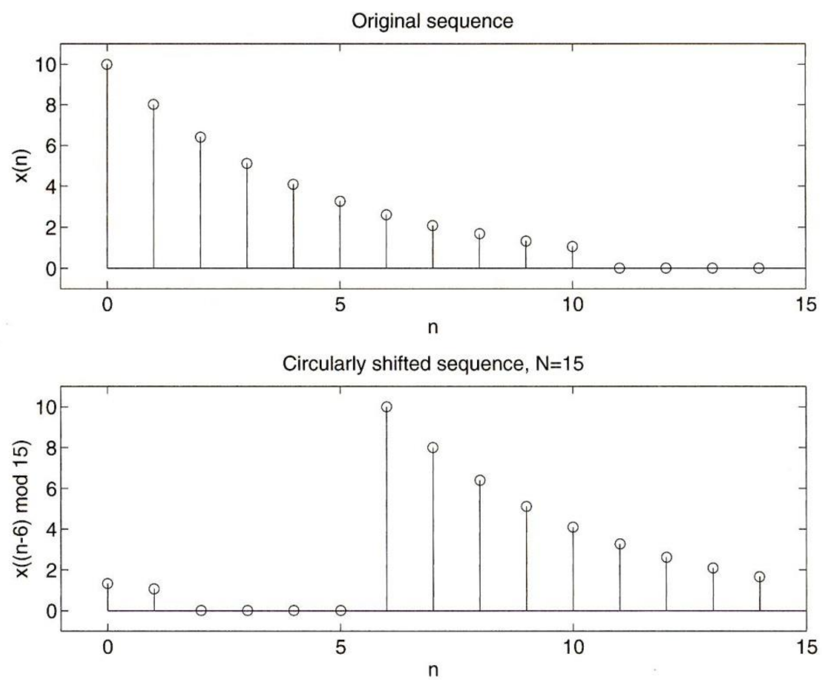
$N = 15$

Example:



Lab #10.2 (1 pt.): Due Feb 7

Given an 11*-point sequence* $x(n) = 10(0.8)^n, \quad 0 \le n \le 10,$
determine and plot $x((n - \text{A}))_{15}$.



Use Your ID: sGFEDCBA

7. Circular convolution

→Convolution of periodic sequence

a convolution operation that contains a circular shift

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

A linear convolution between two N-point sequences will result in a longer sequence.

$$DFT\left[x_1(n) \; \text{Ⓝ} \; x_2(n)\right] = X_1(k) \cdot X_2(k)$$

When we multiply two N-point DFTs in the frequency domain, we get the circular convolution (and not the usual linear convolution) in the time domain.

Proof: Circular convolution

$$X_1[k] = \sum_{n=0}^{N-1} x_1(n) e^{-j2\pi nk/N}, \quad k = 0,1,\ldots,N-1$$

$$X_2[k] = \sum_{n=0}^{N-1} x_2(n) e^{-j2\pi nk/N}, \quad k = 0,1,\ldots,N-1$$

$$X_3[k] = X_1[k] X_2[k], \quad k = 0,1,\ldots,N-1$$

$$x_3(n) = \frac{1}{N} \sum_{k=0}^{N-1} X_3[k] e^{j2\pi kn/N}$$

$$= \frac{1}{N} \sum_{k=0}^{N-1} X_1[k] X_2[k] e^{j2\pi kn/N}$$

Proof: Circular convolution

$$x_3(n) = \frac{1}{N}\sum_{k=0}^{N-1}\left[\sum_{m=0}^{N-1} x_1(m)e^{-j2\pi km/N}\right]\left[\sum_{l=0}^{N-1} x_2(l)e^{-j2\pi kl/N}\right]e^{j2\pi kn/N}$$

$$= \frac{1}{N}\sum_{m=0}^{N-1} x_1(m)\sum_{l=0}^{N-1} x_2(l)\boxed{\left[\sum_{k=0}^{N-1} e^{j2\pi k(n-m-l)/N}\right]}$$

$$\boxed{\sum_{k=0}^{N-1} a^k} = \begin{cases} N, & a=1 \\ \dfrac{1-a^N}{1-a}, & a \neq 1 \end{cases}, \qquad a = e^{j2\pi(n-m-l)/N}$$

$$= \begin{cases} N, & l = n-m+pN = ((n-m))_N, \quad p \text{ an integer} \\ 0, & \text{otherwise} \end{cases}$$

$$x_3(n) = \sum_{m=0}^{N-1} x_1(m)x_2((n-m))_N, \quad n = 0,1,\ldots,N-1$$

Example: $x_1(n) = \{1, \quad 2, \quad 2\}, \quad x_2(n) = \{1, \quad 2, \quad 3, \quad 4\}$

Compute the 4*-point circular convolution* $x_1(n)$ ④ $x_2(n)$.

Time-domain approach $x_1(n)$ ④ $x_2(n) = \sum_{m=0}^{3} x_1(m) x_2((n-m))_4$

$$x_1(m) = \{1, \quad 2, \quad 2, \quad 0\}, \quad x_2(m) = \{1, \quad 2, \quad 3, \quad 4\}$$

for $n = 0$ $\sum_{m=0}^{3} x_1(m) \cdot x_2((0-m))_4 = \sum_{m=0}^{3} \left[\{1,2,2,0\} \cdot \{1,4,3,2\}\right] = \sum_{m=0}^{3} \{1,8,6,0\} = 15$

for $n = 1$ $\sum_{m=0}^{3} x_1(m) \cdot x_2((1-m))_4 = \sum_{m=0}^{3} \left[\{1,2,2,0\} \cdot \{2,1,4,3\}\right] = \sum_{m=0}^{3} \{2,2,8,0\} = 12$

for $n = 2$ $\sum_{m=0}^{3} x_1(m) \cdot x_2((2-m))_4 = \sum_{m=0}^{3} \left[\{1,2,2,0\} \cdot \{3,2,1,4\}\right] = \sum_{m=0}^{3} \{3,4,2,0\} = 9$

for $n = 3$ $\sum_{m=0}^{3} x_1(m) \cdot x_2((3-m))_4 = \sum_{m=0}^{3} \left[\{1,2,2,0\} \cdot \{4,3,2,1\}\right] = \sum_{m=0}^{3} \{4,6,4,0\} = 14$

$$\boxed{x_1(n) \text{ ④ } x_2(n) = \{15, \quad 12, \quad 9, \quad 14\}}$$

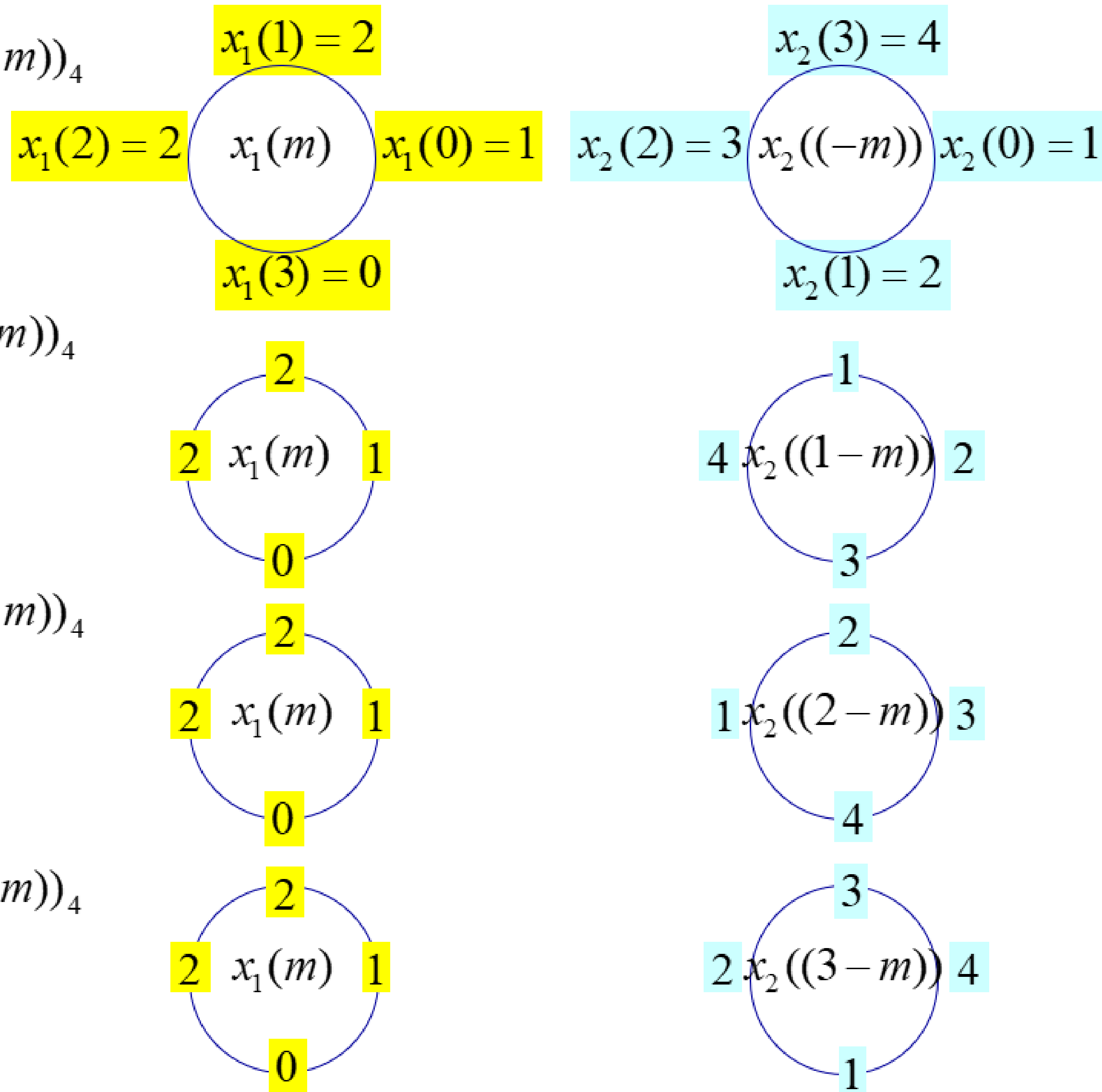
Example: $x_1(n)=\{1, \quad 2, \quad 2\}, \quad x_2(n)=\{1, \quad 2, \quad 3, \quad 4\}$

$$\sum_{m=0}^{3} x_1(m)\cdot x_2((0-m))_4$$

$$\sum_{m=0}^{3} x_1(m)\cdot x_2((1-m))_4$$

$$\sum_{m=0}^{3} x_1(m)\cdot x_2((2-m))_4$$

$$\sum_{m=0}^{3} x_1(m)\cdot x_2((3-m))_4$$



Example:

Frequency-domain approach

DFT of $x_1(n)$ $\quad x_1(n) = \{1,2,2,0\} \Rightarrow X_1(k) = \{5, -1-j2, 1, -1+j2\}$

DFT of $x_2(n)$ $\quad x_2(n) = \{1,2,3,4\} \Rightarrow X_2(k) = \{10, -2+j2, -2, -2-j2\}$

$$X_1(k) \cdot X_2(k) = \{50, 6+j2, -2, 6-j2\}$$

IDFT $\quad \boxed{x_1(n) \; ④ \; x_2(n) = \{15, \quad 12, \quad 9, \quad 14\}}$

the same as before

Example:

```
function y = circonvt(x1,x2,N)
% N-point circular convolution between x1 and x2: (time domain)
% ---------------------------------------------------------------------------
% [y] = circonvt(x1,x2,N)
%  y = output sequence containing the circular convolution
% x1 = input sequence of length N1 <= N
% x2 = input sequence of length N2 <= N
%  N = size of circular buffer
%  Method: y(n) = sum (x1(m)*x2((n-m) mod N))
% Check for the length of x1
if length(x1) > N
        error('N must be >= the length of x1')
end
% Check for the length of x2
if length(x2) > N
        error('N must be >= the length of x2')
end
x1 = [x1 zeros(1,N-length(x1))];
x2 = [x2 zeros(1,N-length(x2))];
m = [0:1:N-1]; x2 = x2(mod(-m,N)+1); H = zeros(N,N);
for n = 1:1:N
H(n,:) = cirshftt(x2,n-1,N);
end
y = x1*conj(H');
```

Lab #10.3 (2 pt.): Due Feb 7

$$x_1(n) = \{1, \quad 2, \quad 2\}, \quad x_2(n) = \{1, \quad 2, \quad 3, \quad 4\}$$

```
>> x1 = [1,2,2]; x2 = [1,2,3,4];  y = circonvt(x1, x2, 4)
y =
     15      12       9      14
```

$$\boxed{x_1(n) \;④\; x_2(n) = \{15, \quad 12, \quad 9, \quad 14\}}$$

$$Compute\ x_1(n)\ ⑤\ x_2(n).$$

$$Compute\ x_1(n)\ ⑥\ x_2(n).$$

$$Compute\ x_1(n)\ Ⓐ\ x_2(n).$$

Use Your ID: sGFEDCBA

8. Multiplication

$$DFT\left[x_1(n)\cdot x_2(n)\right]=\frac{1}{N}X_1(k)\ \text{N}\ X_2(k)$$

9. Parseval's relation *computes the energy in the frequency domain*

$$\varepsilon_x=\sum_{n=0}^{N-1}\left|x(n)\right|^2=\frac{1}{N}\sum_{k=0}^{N-1}\left|X(k)\right|^2$$

$$\frac{\left|X(k)\right|^2}{N}$$ *the energy spectrum of finite-duration sequences*

$$\frac{\left|\tilde{X}(k)\right|^2}{N}$$ *the power spectrum of periodic sequences*

Proof: Multiplication

$$DFT\left[x_1(n)\cdot x_2(n)\right]=\sum_{n=0}^{N-1}x_1(n)\cdot x_2(n)e^{-j\frac{2\pi}{N}nk}$$

$$=\sum_{n=0}^{N-1}\left\{\frac{1}{N}\sum_{l=0}^{N-1}X_1(l)e^{j\frac{2\pi}{N}nl}\right\}x_2(n)e^{-j\frac{2\pi}{N}nk}$$

$$=\frac{1}{N}\sum_{l=0}^{N-1}X_1(l)\left\{\sum_{n=0}^{N-1}x_2(n)e^{-j\frac{2\pi n(k-l)}{N}}\right\}$$

$$=\frac{1}{N}\sum_{l=0}^{N-1}X_1(l)X_2((k-l))_N$$

$$=\frac{1}{N}X_1(k)\ N\ X_2(k)$$

Proof: Parseval's relation

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

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

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

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

$$= \frac{1}{N}\sum_{k=0}^{N-1} X^*(k) X(k)$$

$$\boxed{\sum_{n=0}^{N-1} |x(n)|^2 = \frac{1}{N}\sum_{k=0}^{N-1} |X(k)|^2}$$

Linear Convolution using DFT

Linear Convolution Using the DFT

The DFT operations result in a circular convolution.
How to use the DFT to perform a linear convolution?
How to make a circular convolution identical to the linear convolution?

N_1*-point* N_2 *-point sequence*

$$x_3(n) = x_1(n) * x_2(n)$$

$$= \sum_{k=-\infty}^{\infty} x_1(k) x_2(n-k) = \sum_{0}^{N_1-1} x_1(k) x_2(n-k)$$

$(N_1 + N_2 - 1)$ *-point sequence*

Circular Convolution: Aliasing

$$N = N_1 + N_2 - 1, \quad x_1(n), x_2(n) \quad N\text{-}\textit{point sequence}$$

$$\begin{aligned}
x_4(n) &= x_1(n) \; \textcircled{N} \; x_2(n) \\
&= \left[\sum_{m=0}^{N-1} x_1(m) x_2((n-m))_N \right] R_N(n) \\
&= \left[\sum_{m=0}^{N-1} x_1(m) \sum_{r=-\infty}^{\infty} x_2(n-m-rN) \right] R_N(n) \\
&= \left[\sum_{r=-\infty}^{\infty} \underbrace{\sum_{m=0}^{N_1-1} x_1(m) x_2(n-m-rN)}_{x_3(n-rN)} \right] R_N(n) \\
&= \left[\sum_{r=-\infty}^{\infty} x_3(n-rN) \right] R_N(n)
\end{aligned}$$

$$\boxed{x_4(n) = x_3(n); \quad 0 \le n \le N-1}$$

The circular convolution is an aliased version of the linear convolution.

Circular Convolution: Aliasing

Example $x_1(n) = \{1, \quad 2, \quad 2, \quad 1\}, \;\; x_2(n) = \{1, \quad -1, \quad -1, \quad 1\}$

Determine their linear convolution.

```
>> x1 = [1,2,2,1]; x2 = [1,-1,-1,1];  x3 = conv(x1, x2)
x3 =      1     1    -1    -2    -1     1     1
```

Hence the linear convolution is a 7*-point sequence.*

```
>> x4 = circonvt(x1,x2,7)
x4 =      1     1    -1    -2    -1     1     1
```

In order to use the DFT for linear convolution, we must choose N properly.

Circular Convolution: Aliasing

Example $x_1(n) = \{1, \quad 2, \quad 2, \quad 1\}, \quad x_2(n) = \{1, \quad -1, \quad -1, \quad 1\}$

```
        1  2  2 1
        1 -1 -1 1
        ---------
        1  2  2 1
     -1 -2 -2 -1
  -1 -2 -2 -1
 1  2  2 1
-----------------
1 1 -1 -2 -1 1 1
```

N=5

2 2 -1 -2 -1

If we make both $x_1(n)$ *and* $x_2(n)$ $N = N_1 + N_2 - 1$ *point sequences by padding an appropriate number of zeros, then the circular convolution is identical to the linear convolution.*

Example $x_3(n) = \{1,1,-1,-2,-1,1,1\}$ *Error Analysis*

When $N = \max(N_1, N_2)$ *is chosen for circular convolution, then the first* **(M-1)** *samples are in error, where* $M = \min(N_1, N_2)$.

$$x_4(n) = x_1(n) \; 6 \; x_2(n) = \{2,1,-1,-2,-1,1\}$$
$$e(n) = \{2,1,-1,-2,-1,1\} - \{1,1,-1,-2,-1,1\}, \quad 0 \le n \le 5$$
$$= \{1,0,0,0,0,0\}$$
$$= x_3(n+6)$$

$$x_4(n) = x_1(n) \; 5 \; x_2(n) = \{2,2,-1,-2,-1\}$$
$$e(n) = \{2,1,-1,-2,-1\} - \{1,1,-1,-2,-1\}, \quad 0 \le n \le 4$$
$$= \{1,1,0,0,0\}$$
$$= x_3(n+5)$$

$$x_4(n) = x_1(n) \; 4 \; x_2(n) = \{0,2,0,-2\}$$
$$e(n) = \{0,2,0,-2\} - \{1,1,-1,-2\}, \quad 0 \le n \le 3$$
$$= \{-1,1,1,0\}$$
$$= x_3(n+4)$$

Block Convolution

We want to filter an input sequence that is being received continuously, such as a speech signal from a microphone.

Some practical problems:

1. *will have to compute a large* DFT
2. *output samples are not available until all input samples are processed*
3. *an unacceptably large amount of delay*

Therefore

1. *segment the infinite-length input sequence into smaller sections (or blocks)*
2. *process each section using the* DFT
3. *assemble the output sequence from the outputs of each section*

Example $x(n)=(n+1),\quad 0\le n\le 9,\quad g(n)=\{\underset{\uparrow}{1},0,-1\}$

the overlap-save method using N=6 *to compute* $y(n)=x(n)*g(n)$.

$$x_1(n)=\{0,0,\boxed{1,2,3,4}\}$$
$$x_2(n)=\{3,4,\boxed{5,6,7,8}\}$$
$$x_3(n)=\{7,8,\boxed{9,10,0,0}\}$$
$$y_1=x_1(n)\ ⑥\ g(n)=\{\boxed{-3,-4},1,2,2,2\}$$
$$y_2=x_2(n)\ ⑥\ g(n)=\{\boxed{-4,-4},2,2,2,2\}$$
$$y_3=x_3(n)\ ⑥\ g(n)=\{\boxed{7,8},2,2,-9,-10\}$$

$$\boxed{\begin{array}{l} N=6, M=3 \\ N=L+M-1 \\ L=4 \end{array}}$$

$$y(n)=\{\underset{\uparrow}{1},2,2,2,2,2,2,2,2,2,-9,-10\}$$

Agrees with the linear convolution

$$x(n)*g(n)=\{\underset{\uparrow}{1},2,2,2,2,2,2,2,2,2,-9,-10\}$$

```
function [y] = ovrlpsav(x,g,N)
% Overlap-Save method of block convolution
% ------------------------------------------------
% [y] = ovrlpsav(x,g,N)
%  y = output sequence
%  x = input sequence
%  g = impulse response
%  N = block length
%
Lenx = length(x); M = length(g); M1 = M-1; L = N-M1;
g = [g zeros(1,N-M)];
%
x = [zeros(1,M1), x, zeros(1,N-1)]; % preappend (M-1) zeros
K = floor((Lenx+M1-1)/(L));         % # of blocks
Y = zeros(K+1,N);
% convolution with successive blocks
for k = 0:K
  xk = x(k*L+1:k*L+N);
  Y(k+1,:) = circonvt(xk,g,N);
end
Y = Y(:,M:N)';                      % discard the first (M-1) samples
y = (Y(:))';                        % assemble output
```

```
>> n = 0:9; x = n+1; g = [1,0,-1]; N = 6; y = ovrlpsav(x,g,N)
y =
  1    2    2    2    2    2    2    2    2    2   -9   -10
```

Thank you

JAIST
JAPAN
ADVANCED INSTITUTE OF
SCIENCE AND TECHNOLOGY
1990