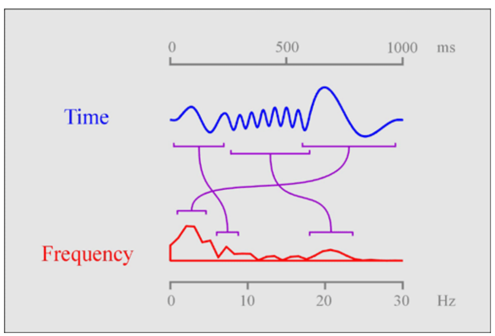
Ch3: Frequency Analysis for DT Signals

Information source and input transducer → Source Coding → Channel Coding → Modulator → Channel → Demodulator (Matched Filter) → Channel Decoding → Source Decoding → Information sink and output transducer

Ch3.2: DFT

- **Questions to be answered:**
  - **Discrete Fourier Transform:** Sampling the frequency domain
  - **Circular Convolution:** From linear convolution to circular convolution with the help of DFT.



Reading: Mitra (Ch5.1 – 5.4 & 5.10)

Signal processing

- also want frequency have discrete !

DTFT :

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

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

DFT

Want to get discrete in freq. domain

- Do sampling
- If Do correctly → can recover the original

Ch3.2: Discrete Fourier Transform

Analysis :

sythnesis :

- **Discrete Fourier Transform**
- Circular Convolution

DTFT

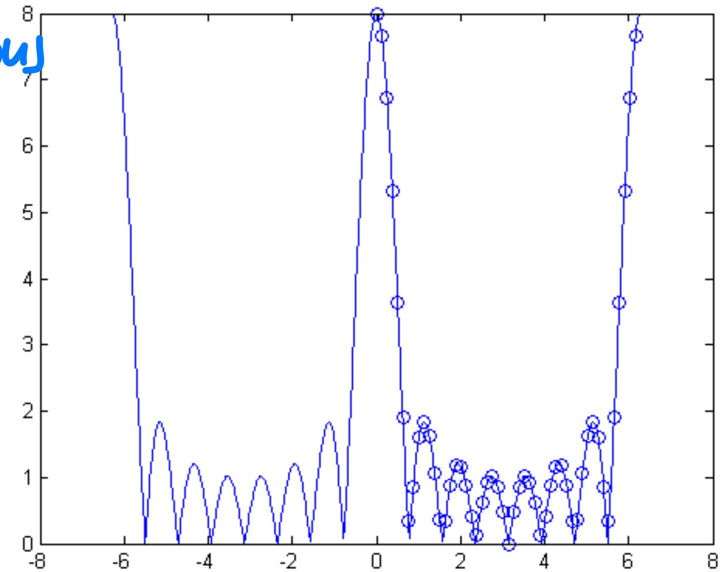
$x[n] \longrightarrow X(e^{jw})$ ← continuous

$\Downarrow$ (sample)

DFT

$x[n] \longrightarrow X[k]$ ← discrete

$0 \le n \le K-1$ $\quad$ $0 \le k \le N-1$



4G 5G

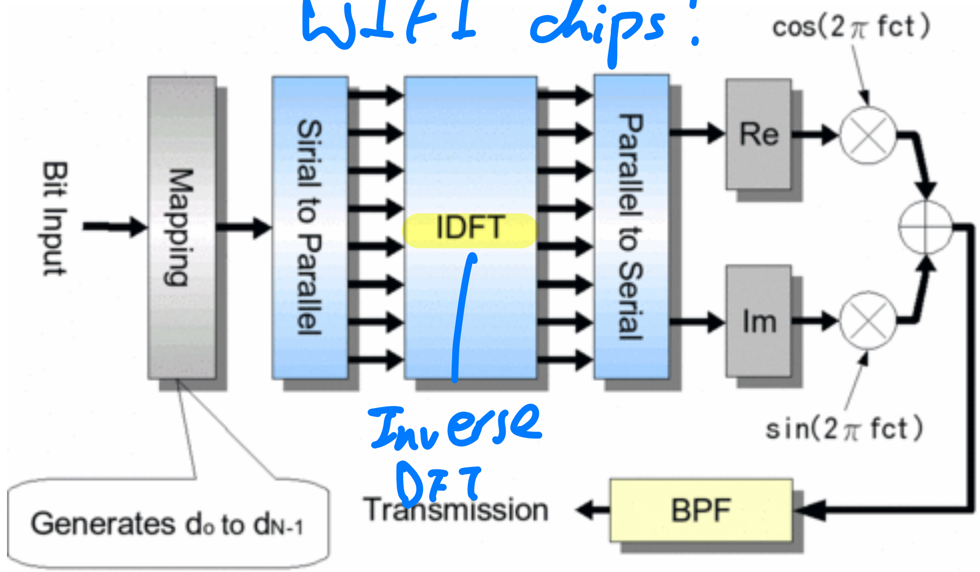
WiFi and OFDM

DFT in communication devices

手机, iPad etc.

WIFI chips!

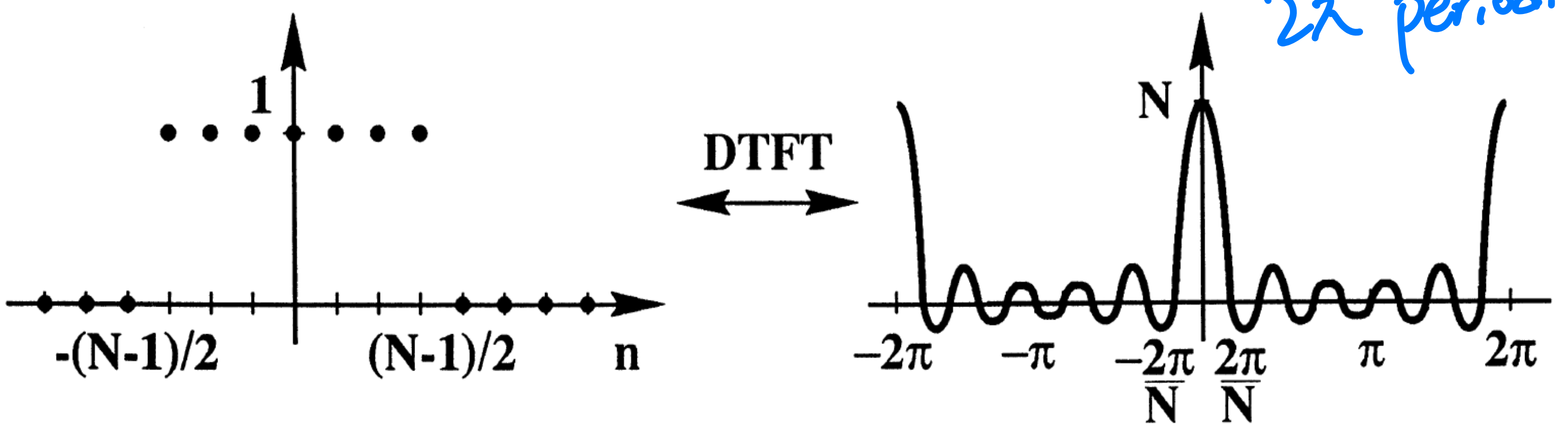
⇒



Inverse DFT

Discrete Time Fourier Transform (DTFT)

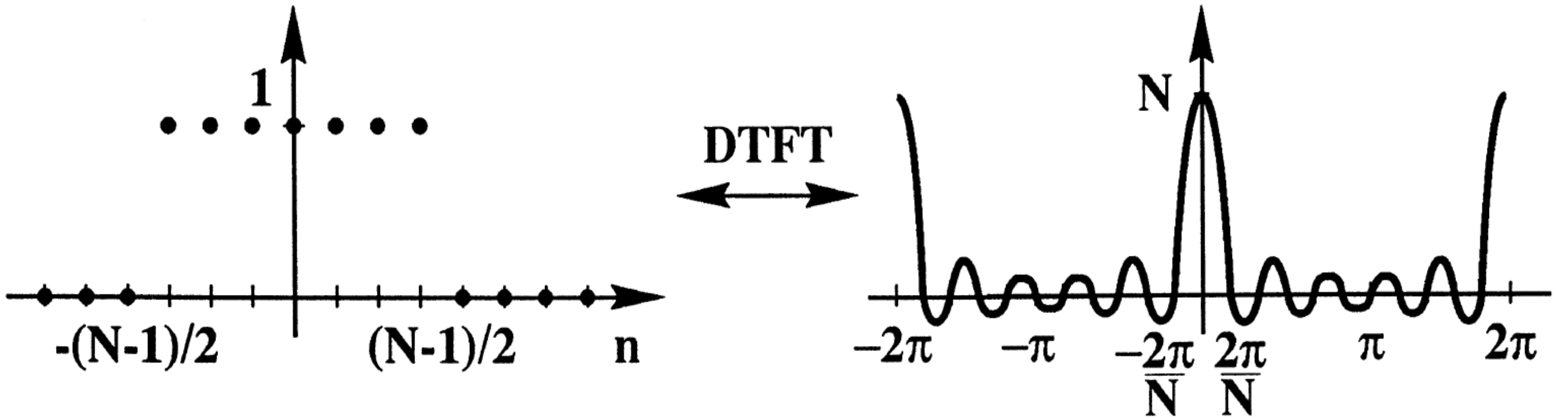
2π periodic



- With DTFT, we have limited number of samples in time domain, but continuous function in the frequency domain.
- As a result, DTFT is not directly applicable to the digital analysis. (**Why?**) ? Cannot analysis the waveform
- **Questions:**
  - **Can we reconstruct the time-domain samples by part of the frequency domain signals? (Periodic?)** Some redundency!
  - **Can we utilize limited number of samples from the frequency-domain to reconstruct the time-domain samples? (Sampling Frequency-Domain?)**

sampling theory

Discrete Fourier Transform (DFT)



- The simplest relation between a length-N sequence $x[n]$ and its DTFT $X(e^{j\omega})$ is obtained by uniformly sampling $X(e^{j\omega})$ on the ω-axis at $\boxed{\omega_k} = \frac{2\pi k}{N}, 0 \le k \le N-1$.

$\sum x[n] e^{-j\omega n}$

- From the definition of DTFT, we have

DFT

$$X[k] = X(e^{j\omega})|_{\omega_k = \frac{2\pi k}{N}} = \sum_{n=-\infty}^{\infty} x[n] e^{-j\boxed{\frac{2\pi k n}{N}}}$$

Sampled value

Sampling!

- **Note:** $X[k]$ is also a length-N sequence in the frequency domain.

Inverse Discrete Fourier Transform (IDFT)

- The sequence $X[k]$ is called the **discrete Fourier transform (DFT)** of sequence $x[n]$.

N is length of original sequence

- Utilizing the notation $\boxed{W_N} = e^{-j\frac{2\pi}{N}}$, DFT is usually expressed as

constant for given N

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

very similar

n is fixed! N-length sequence!

- The inverse discrete Fourier transform (IDFT) is given by

Just sum them together

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

$= x[n]$

↝ Need to prove for correctness!

- **Question:** Can we uniquely reconstruct $x[n]$ from $X[k]$? In other words, is $\tilde{x}[n] = x[n]$?

inner product of two products!

不用理会是否能construct! 能給 original signal 就行!

Inverse Discrete Fourier Transform (IDFT)

- From the definition of the inverse discrete Fourier transform (IDFT), $\tilde{x}[n] = \frac{1}{N}\sum_{k=0}^{N-1} X[k]\, W_N^{-kn}$.

  DFT

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

$$\tilde{x}[n] = \frac{1}{N}\sum_{k=0}^{N-1}\left(\sum_{l=0}^{N-1} x[l]\, W_N^{kl}\right) W_N^{-kn}$$

Group together!

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

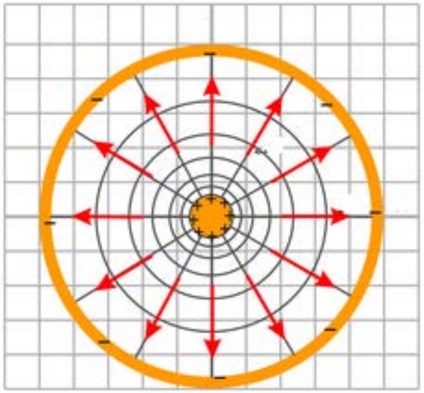
$$= \sum_{l=0}^{N-1} x[l] \left\{ \frac{1}{N}\sum_{k=0}^{N-1} W_N^{\boxed{-(n-l)}k} \right\} = \begin{cases} 1 & l = n \\ 0 & l \neq n \end{cases}$$

- Given $\sum_{k=0}^{N-1} W_N^{-(n-l)k} = \begin{cases} N, & \text{for } n - l = rN, r \text{ an integer} \\ 0, & \text{otherwise} \end{cases}$,

  we have $\tilde{x}[n] = x[n]$.

  $= \sum x[l]\, g[l] = x[n]$

- Thus, we can reconstruct $x[n]$ from $X[k]$.



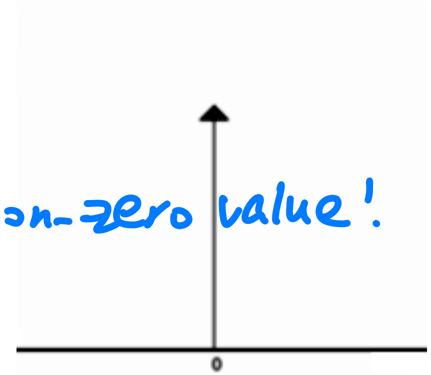
l is only from 0 to $N-1$!!!

Discrete Fourier Transform

- **Example:** Consider the length-N sequence

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

one non-zero value!



- Its N-point DFT is given by

one terms!

$$X[k] \overset{\text{def}}{=} \sum_{n=0}^{N-1} \underbrace{x[n]W_N^{kn}}_{\text{only one}} = \underbrace{x[0]}_{=1}\underbrace{W_N^0}_{=1}$$

X[k]

$$= \underline{1}, 0 \le k \le N-1$$

0 k-1

Shifted x[n]

- **Example:** Consider the length-N sequence

y[n] 1 m n

$$y[n] = \begin{cases} 1, & n = m, 0 \le m \le N-1 \\ 0, & \text{otherwise} \end{cases}$$

- Its N-point DFT is given by

shifted in freq. domain

$$Y[k] = \sum_{n=0}^{N-1} y[n]W_N^{kn} = y[m]W_N^{km} = \underbrace{W_N^{km}}_{e^{-j\cdot\frac{2\pi}{N}km}}, 0 \le k \le N-1$$

Discrete Fourier Transform

N=100

N=100

- **Example:** Determine DFT of the length-N sequence

$$g[n] = \cos\left(\frac{2\pi rn}{N}\right), 0 \le r \le N-1$$

g[n]=2cos(2π·0.02n) +10cos(2π·0.05n)

$r_1=2$

$r_2=5$

- Using the trigonometric identity, we can obtain

$$g[n] = \frac{1}{2}\left(e^{j2\pi rn/N} + e^{-j2\pi rn/N}\right) = \frac{1}{2}\left(W_N^{-rn} + W_N^{rn}\right)$$

Sum and complex exp.

- The N-point DFT is given by $G[k] = \sum_{n=0}^{N-1} g[n] W_N^{kn} = \frac{1}{2}\left(\sum_{n=0}^{N-1} W_N^{-(r-k)n} + \sum_{n=0}^{N-1} W_N^{(r+k)n}\right)$ ✓

2, 5, 95, 98

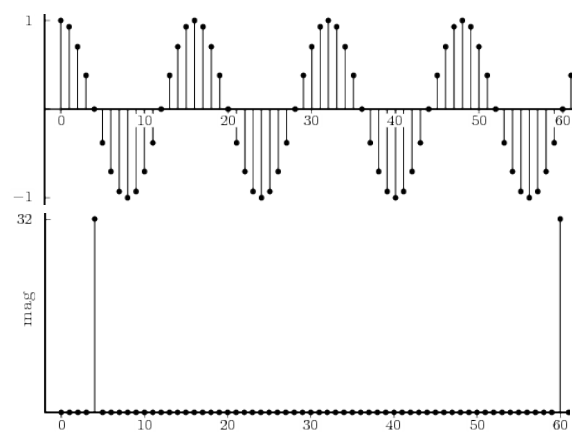
=N   =N for r+k=N

- Using the identity, $\sum_{n=0}^{N-1} W_N^{-(k-l)n} = \begin{cases} N, \text{for } k-l = rN, r \text{ an integer} \\ 0, \text{otherwise} \end{cases}$

only when r=k, non-zero value

only one non-zero value!

we get $G[k] = \begin{cases} \frac{N}{2}, \text{for } k = r \\ \frac{N}{2}, \text{for } k = N-r \\ 0, \text{otherwise} \end{cases}$ ✓



Matrix Relations

$\vec{X}$

$[X[0], \ldots x[N-1]]$

$$\begin{bmatrix} W_N^{0} \\ W_N^{1n} \\ \vdots \\ W_N^{(N-1)n} \end{bmatrix} = \vec{W_k}$$

- The DFT samples defined by

$\vec{X} = \vec{x} \cdot D_N$

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

$D_N = [\vec{W_0}, \vec{W_1}, \vec{W_2} \ldots \vec{W_{N-1}}]$

$= \vec{x} \cdot \vec{W_k}$

can be expressed in matrix form as $\mathbf{X} = \mathbf{D}_N \mathbf{x}$ where

$$\mathbf{X} = [X[0]\ X[1]\ \ldots X[N-1]]^T, \mathbf{x} = [x[0]\ x[1] \ldots x[N-1]]^T$$ and

$\mathbf{D}_N$ is the $N \times N$ DFT matrix given by

$X^T = x^T \cdot D_N \quad x \to X$

$$\mathbf{D}_N = \begin{array}{cc} \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & W_N^1 & W_N^2 & \cdots & W_N^{(N-1)} \\ 1 & W_N^2 & W_N^4 & \cdots & W_N^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & W_N^{(N-1)} & W_N^{2(N-1)} & \cdots & W_N^{(N-1)^2} \end{bmatrix} & \begin{array}{l} k=0 \\ k=1 \\ k=2 \\ \vdots \\ k=N-1 \end{array} \end{array}$$

$$X[k] = \vec{x} \cdot \vec{W_k} = [x[0]\ x[1] \cdots] \begin{bmatrix} W_N^{0} \\ W_N^{k} \\ \vdots \\ W_N^{(N-1)k} \end{bmatrix}$$

Matrix Relations

- The IDFT relation can be expressed as Since $\vec{X} = D\vec{x}$

$$\mathbf{x} = \mathbf{D}_N^{-1}\mathbf{X}$$

where $\mathbf{D}_N^{-1}$ is the $N \times N$ IDFT matrix with $\mathbf{D}_N^{-1} = \frac{1}{N}\mathbf{D}_N^*$ where

Just take conjugate!

$$\mathbf{D}_N^* = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & W_N^{-1} & W_N^{-2} & \cdots & W_N^{-(N-1)} \\ 1 & W_N^{-2} & W_N^{-4} & \cdots & W_N^{-2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & W_N^{-(N-1)} & W_N^{-2(N-1)} & \cdots & W_N^{-(N-1)^2} \end{bmatrix}$$

Special matrix!

Same as doing calculation!

DFT Properties: Symmetry Relations

Length-N Sequence	N-point DFT
$x[n]$	$X[k]$
$x^*[n]$	$X^*[\langle -k \rangle_N]$
$x^*[\langle -n \rangle_N]$	$X^*[k]$
$\mathrm{Re}\{x[n]\}$	$X_{\text{pcs}}[k] = \frac{1}{2}\{X[\langle k \rangle_N] + X^*[\langle -k \rangle_N]\}$
$j\,\mathrm{Im}\{x[n]\}$	$X_{\text{pca}}[k] = \frac{1}{2}\{X[\langle k \rangle_N] - X^*[\langle -k \rangle_N]\}$
$x_{\text{pcs}}[n]$	$\mathrm{Re}\{X[k]\}$
$x_{\text{pca}}[n]$	$j\,\mathrm{Im}\{X[k]\}$

Periodic conjugate symmetric!

Note: $x_{\text{pcs}}[n]$ and $x_{\text{pca}}[n]$ are the periodic conjugate-symmetric and periodic conjugate-antisymmetric parts of $x[n]$, respectively. Likewise, $X_{\text{pcs}}[k]$ and $X_{\text{pca}}[k]$ are the periodic conjugate-symmetric and periodic conjugate-antisymmetric parts of $X[k]$, respectively.

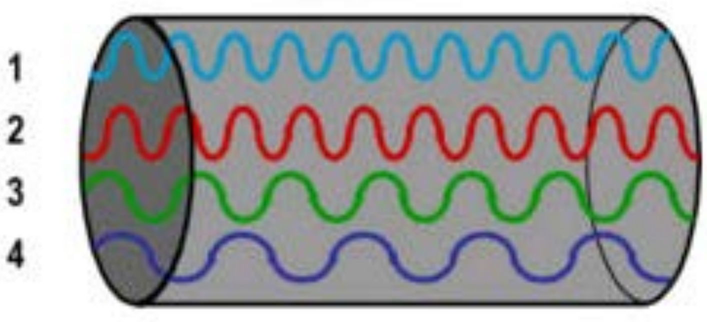
General Properties of DFT

Type of Property	Length-N Sequence	N-point DFT
	$g[n]$ $h[n]$	$G[k]$ $H[k]$
Linearity	$\alpha g[n] + \beta h[n]$	$\alpha G[k] + \beta H[k]$
Circular time-shifting	$g[\langle n - n_o \rangle_N]$	$W_N^{kn_o} G[k]$
Circular frequency-shifting	$W_N^{-k_o n} g[n]$	$G[\langle k - k_o \rangle_N]$
Duality	$G[n]$	$N g[\langle -k \rangle_N]$
N-point circular convolution	$\sum_{m=0}^{N-1} g[m]h[\langle n - m \rangle_N]$	$G[k]H[k]$ → like DTFT
Modulation	$g[n]h[n]$	$\frac{1}{N} \sum_{m=0}^{N-1} G[m]H[\langle k - m \rangle_N]$
Parseval's relation	$\sum_{n=0}^{N-1} \lvert x[n] \rvert^2 = \frac{1}{N} \sum_{k=0}^{N-1} \lvert X[k] \rvert^2$	

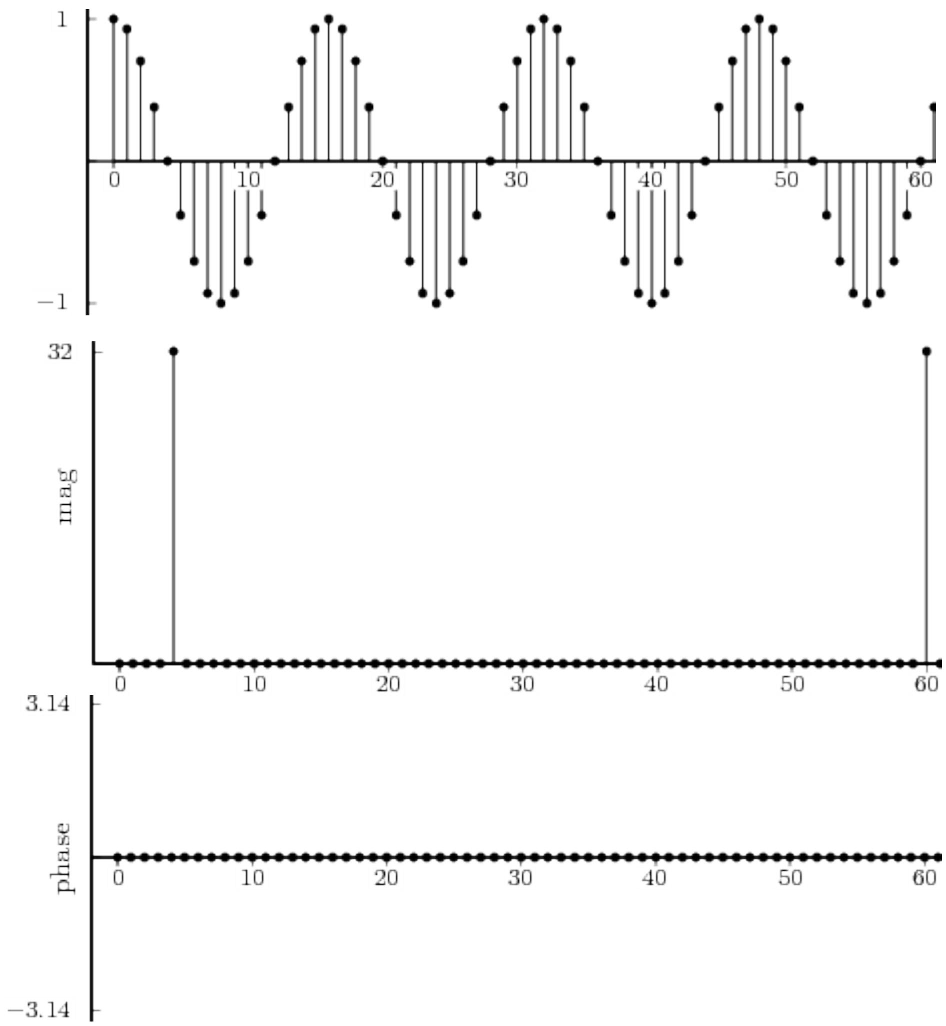
why not linear convolution? The length is N-1, linear convolution is length: 2N-1, which is outside this window!

Physical Interpretation

- Decomposition of a finite-length signal into a set of *N* sinusoidal components
  - Take an array of *N* complex sinusoidal generators;
  - Set the frequency of the *k*-th generator to $(2\pi/N)k$;
  - Set the amplitude of the *k*-th generator to $X[k]$, i.e. to the magnitude of the *k*-th DFT coefficient;
  - Set the phase of the *k*-th generator to $\measuredangle X[k]$, i.e. to the phase of the *k*-th DFT coefficient;
  - Start the generators at the same time and sum their outputs.
- The first *N* output values of this "machine" are exactly *x*[*n*].



Physical Interpretation: Example

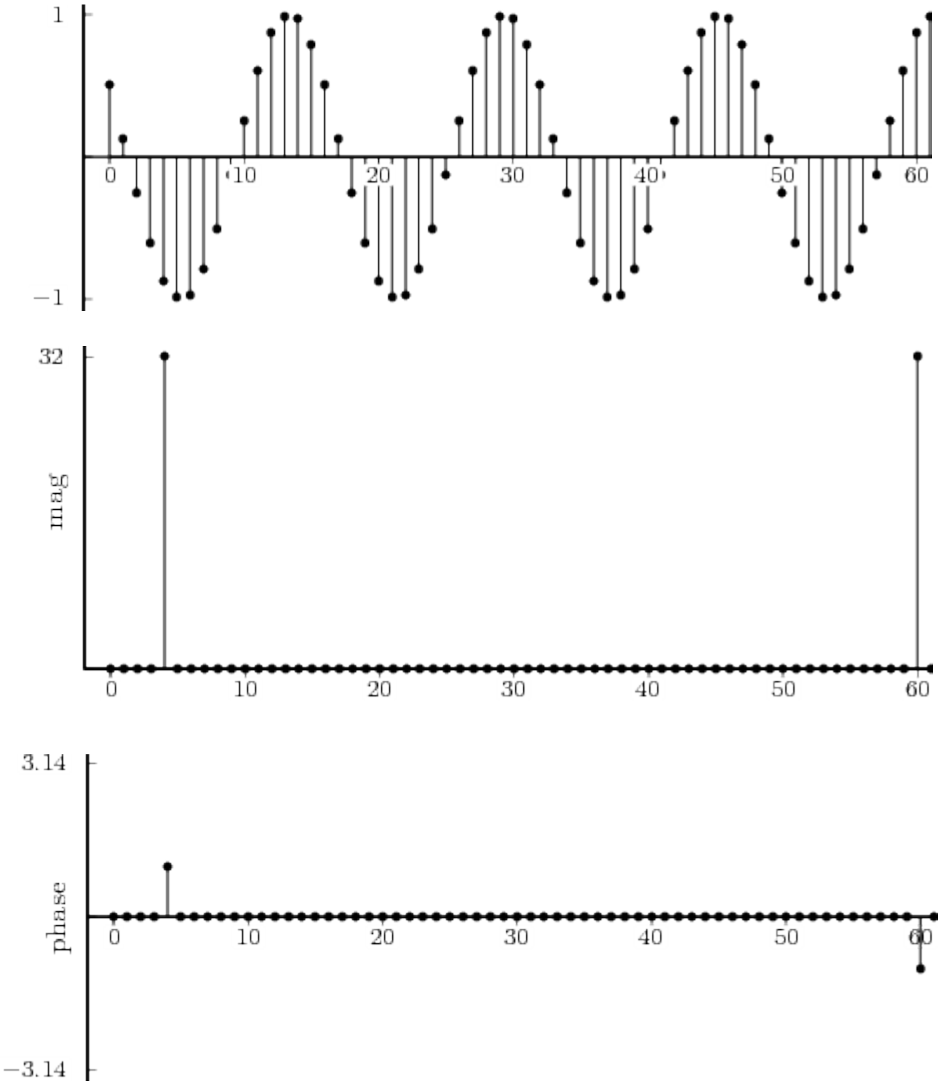


$N=64$

$$x[n] = \cos\left(\frac{\pi}{8}n\right) \quad n = 0, \dots, 63$$

2-non zero in
freq. domain

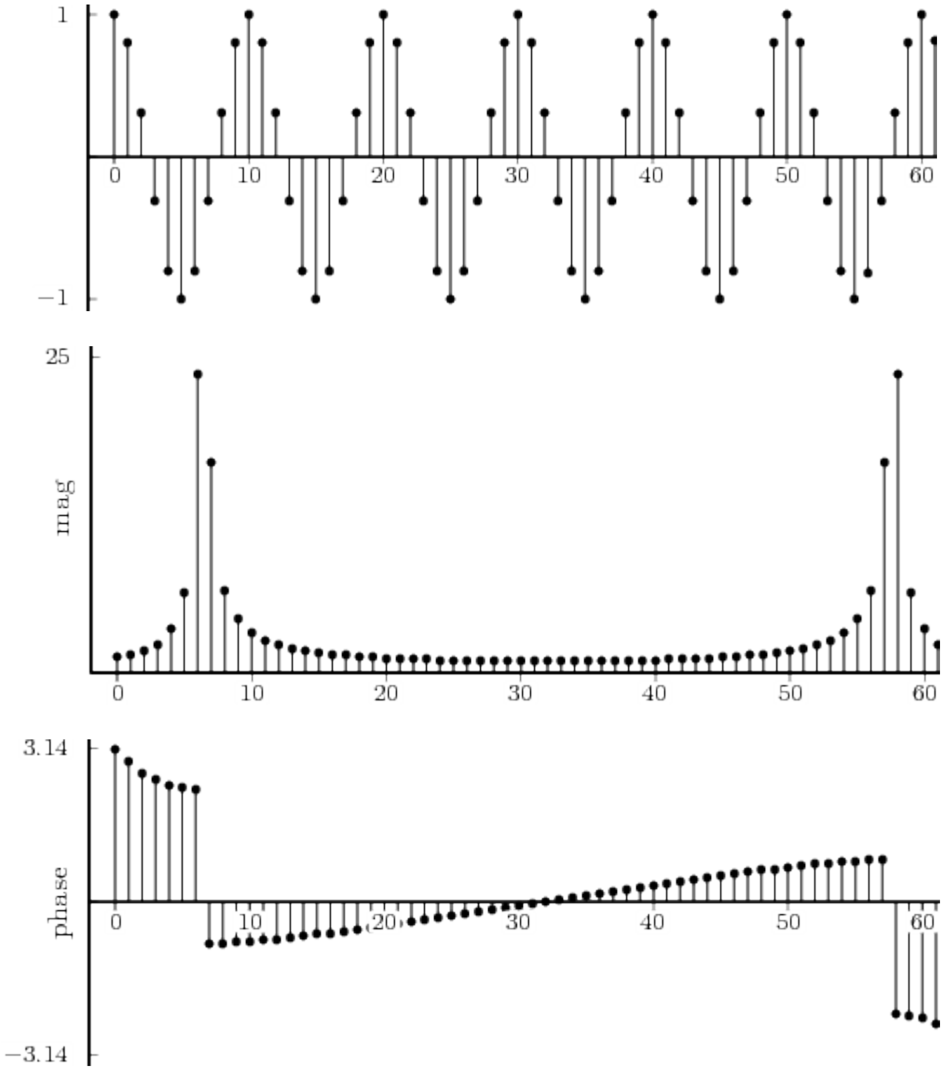
Physical Interpretation: Example



$$x[n] = \cos\left(\frac{\pi}{8}n + \frac{\pi}{3}\right) \quad n = 0, \ldots, 63$$

$64 / 8 = 8$

Physical Interpretation: Example

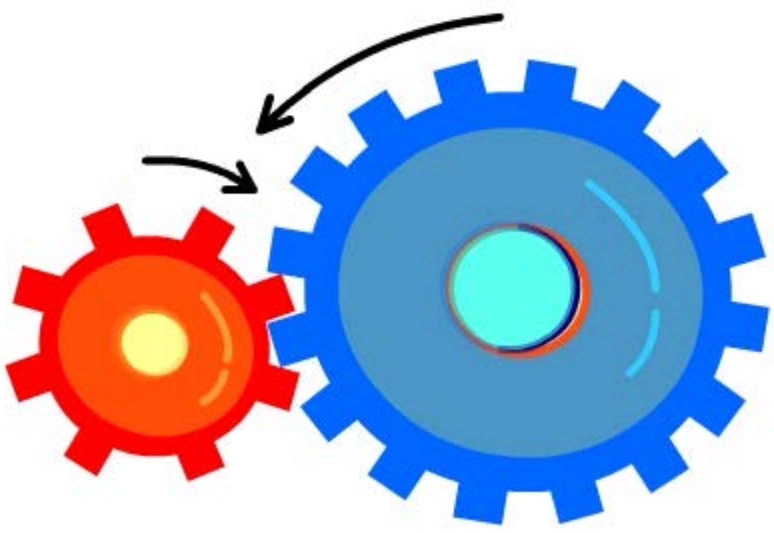


$$x[n] = \cos\left(\frac{\pi}{5} n\right) \quad n = 0, \ldots, 63$$

$N = 64$

Ch3.2: Frequency Analysis for DT Signals

- Discrete Fourier Transform
- **Circular Convolution**



Before is linear convolution!

Circular Shift of a Sequence

- Consider a length-N sequence $x[n]$ defined for $0 \leq n \leq N-1$. Sample values are equal to zero for values of $n < 0$ and $n \geq N$.

- For any arbitrary integer n_o, the shifted sequence $x_1[n] = x[n - n_o]$, is no longer defined for the range $0 \leq n \leq N-1$.

  no longer in the range

- Thus, we need to define another type of "shift" that will always keep the shifted sequence in the range $0 \leq n \leq N-1$.

- The desired shift, called the **circular shift**, is defined using a modulo operation:

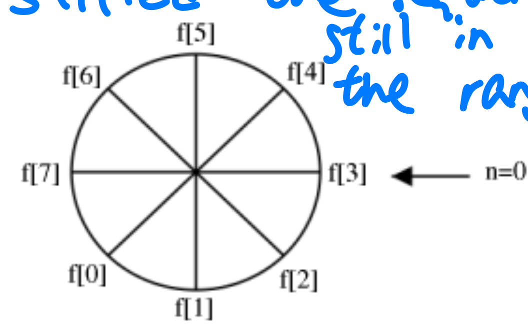
$$x_c[n] = x[< n - n_o >_N]$$

shifted the sequence still in the range

For $n_o > 0$ (right circular shift), the above equation implies

$$x_c[n] = \begin{cases} x[n - n_o], \text{for } n_o \leq n \leq N-1 \\ x[n - n_o + N], \text{for } 0 \leq n < n_o \end{cases}$$

state in the window!



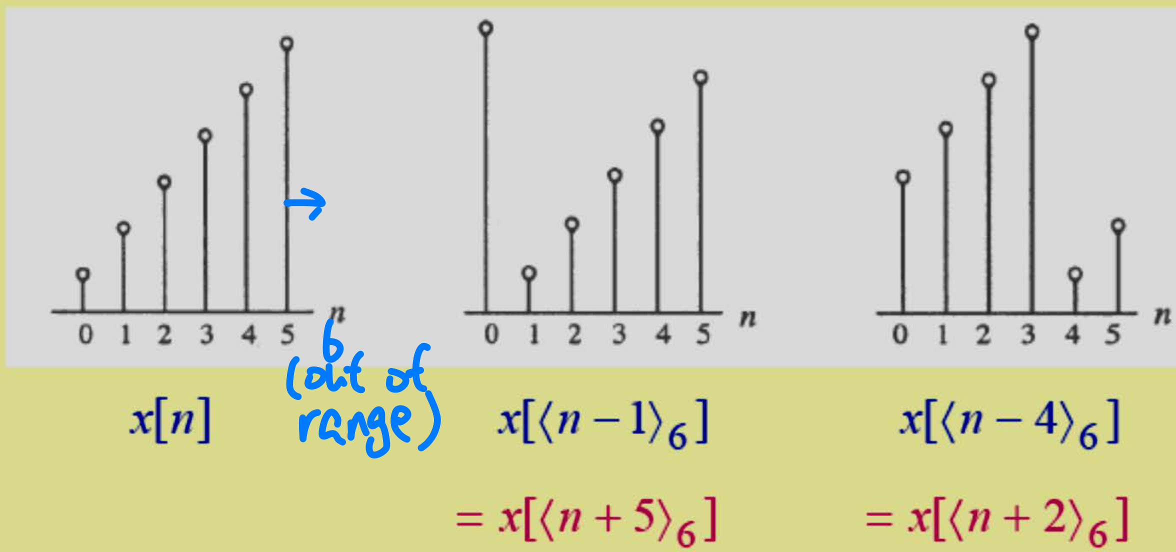
linear convolution

↓

filp but out of range!

Circular Shift of a Sequence

- Illustration of the concept of a circular shift



- A right circular shift by n_o is equivalent to a left circular shift by $N - n_o$ sample periods.
- A circular shift by an integer number n_o greater than *N* is equivalent to a circular shift $\langle n_o \rangle_N$.

Circular Convolution

- Circular convolution is analogous to linear convolution, but with a subtle difference.
- Consider two length-*N* sequences, $g[n]$ and $h[n]$, respectively.
- Their linear convolution results in a length $(2N-1)$ sequence $y_L[n]$ given by ✓

$$y_L[n] = \sum_{m=0}^{N-1} g[m]h[n-m], 0 \leq n \leq 2N-2$$

- The longer form results from the time-reversal of the sequence $h[n]$ and its linear shift to the right

Drew pictures!
Maybe good to understand

Circular Convolution

- To develop a convolution-like operation resulting in a **length-*N*** sequence , we need to define a circular time-reversal, and then apply a circular time-shift.

- Resulting operation, called a **circular convolution**, is defined by

$$y_C[n] = \sum_{m=0}^{N-1} g[m]h[< n - m >_N]\,, 0 \le n \le N - 1$$

- Since the operation defined involves two length-*N* sequences, it is often referred to as an *N*-point circular convolution, denoted as

$$y_C[n] = g[n] \, N \, h[n].$$

design N!

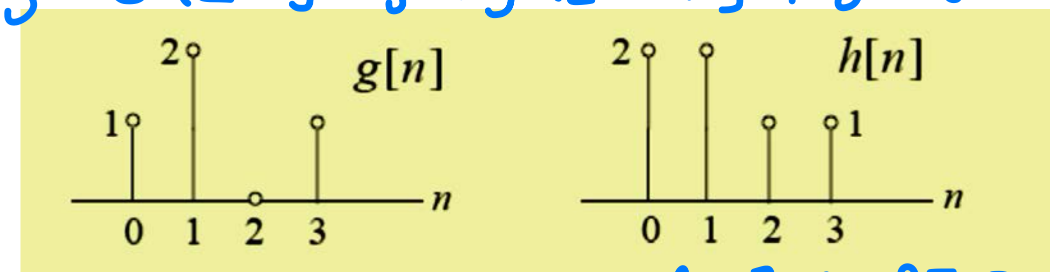
- The circular convolution is commutative,

$$g[n] \, N \, h[n] = h[n] \, N \, g[n]$$

Circular Convolution

- **Example –** Determine the 4-point circular convolution of two length-4 sequences: $g[n] = \{1\ 2\ 0\ 1\}$ and $h[n] = \{2\ 2\ 1\ 1\}$ as sketched below

$y[0] = g[0]h[0-0] + g[1]h[0-1] + g[2]h[0-2] + g[3]h[0-3]$



$2\delta[n] + 2\delta[n-1] + \delta[n-2] + \delta[n-3]$

- The result is a length-4 sequence

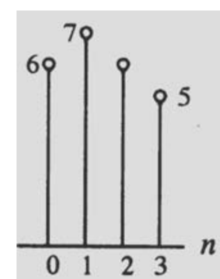
$$y_C[n] = g[n] \circledN h[n] = \sum_{m=0}^{3} g[m]h[< n - m >_4]\,,\ 0 \le n \le 3$$

- For example,

$n=1$

$$y_C[0] = \sum_{m=0}^{3} g[m]h[< -m >_4]$$
$$= \underline{g[0]h[0]} + g[1]h[3] + g[2]h[2] + g[3]h[1]$$
$$= 6$$

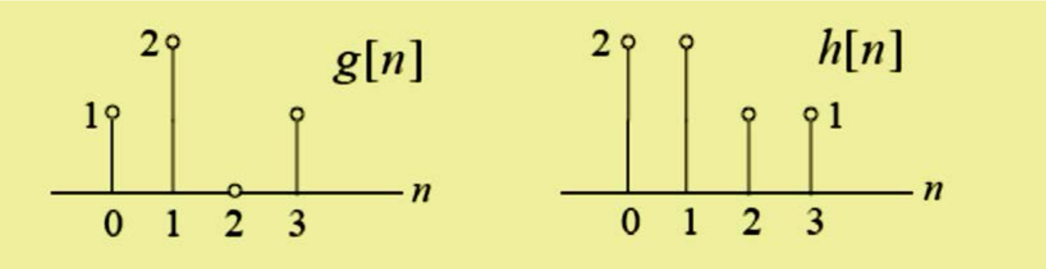
$h[<n-0>]$ $h[<n-1>_4]$ $<0-1>_4 = 3$ $\rightarrow 3$ $h[<3-1>_4]$ $\rightarrow 2$



Circular Convolution: DFT Method

可用FFT代替

- **Example –** Determine the 4-point circular convolution of two length-4 sequences: $g[n] = \{1\ 2\ 0\ 1\}$ and $h[n] = \{2\ 2\ 1\ 1\}$ as sketched below



- The 4-point DFT $G[k]$ *and* $H[k]$ are given by

$$G[k] = \{4, 1 - j, -2, 1 + j\} \text{ and } H[k] = \{6, 1 - j, 0, 1 + j\}.$$

- Thus, the 4-point DFT of $y_C[n]$ is given by

$$Y_C[k] = G[k]H[k] = \{24, -j2, 0, j2\}.$$

- A 4-point IDFT of $Y_C[k]$ yields

$$y_C[n] = \{6\ 7\ 6\ 5\}.$$

Circular Convolution

- **Example –** Now let us extend the two length-4 sequences to length-7 by appending each with three zero-valued samples

$$g_e[n] = \begin{cases} g[n], & 0 \le n \le 3 \\ 0, & 4 \le n \le 6 \end{cases}$$

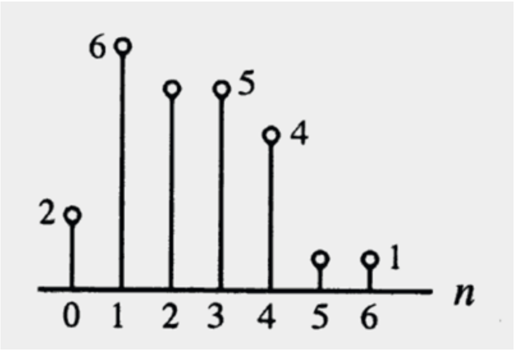
$$h_e[n] = \begin{cases} h[n], & 0 \le n \le 3 \\ 0, & 4 \le n \le 6 \end{cases}$$

$N = 7$

- We next determine the 7-point circular convolution $g_e[n]$ and $h_e[n]$:

$$y[n] = \sum_{m=0}^{6} g_e[m] h_e[< n - m >_7] = \{2, 6,5, 5, 4,1,1\}$$

- It can be checked that $y[n]$ is precisely the sequence $y_L[n]$ obtained by a **linear convolution of $g[n]$ and $h[n]$**



Circular Convolution

- The N-point circular convolution can be written in matrix form as

$$\begin{bmatrix} y_C[0] \\ y_C[1] \\ y_C[2] \\ \vdots \\ y_C[N-1] \end{bmatrix} = \begin{bmatrix} h[0] & h[N-1] & h[N-2] & \cdots & h[1] \\ h[1] & h[0] & h[N-1] & \cdots & h[2] \\ h[2] & h[1] & h[0] & \cdots & h[3] \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ h[N-1] & h[N-2] & h[N-3] & \cdots & h[0] \end{bmatrix} \begin{bmatrix} g[0] \\ g[1] \\ g[2] \\ \vdots \\ g[N-1] \end{bmatrix}$$

- Note: The elements of each diagonal of the $N \times N$ matrix are equal
- Such a matrix is called a **circulant matrix**

Circular Convolution

- <u>Tabular Method</u>
- We illustrate the method by an example
- Consider the evaluation of $y[n] = h[n] \circledast g[n]$ where $\{g[n]\}$ and $\{h[n]\}$ are length-4 sequences
- First, the samples of the two sequences are multiplied using the conventional multiplication method as shown on the next slide

Circular Convolution

n:	0	1	2	3	$\langle 4\rangle_4$	$\langle 5\rangle_4$	$\langle 6\rangle_4$
$g[n]$:	$g[0]$	$g[1]$	$g[2]$	$g[3]$			
$h[n]$	$h[0]$	$h[1]$	$h[2]$	$h[3]$			
	$g[0]h[0]$	$g[1]h[0]$	$g[2]h[0]$	$g[3]h[0]$			
		$g[0]h[1]$	$g[1]h[1]$	$g[2]h[1]$	$g[3]h[1]$		
			$g[0]h[2]$	$g[1]h[2]$	$g[2]h[2]$	$g[3]h[2]$	
				$g[0]h[3]$	$g[1]h[3]$	$g[2]h[3]$	$g[3]h[3]$

The partial products generated in the 2nd, 3rd, and 4th rows are circularly shifted to the left as indicated above

Circular Convolution

Same as definition

- The modified table after circular shifting is shown below

n:	0	1	2	3
$g[n]$:	$g[0]$	$g[1]$	$g[2]$	$g[3]$
$h[n]$:	$h[0]$	$h[1]$	$h[2]$	$h[3]$
	$g[0]h[0]$	$g[1]h[0]$	$g[2]h[0]$	$g[3]h[0]$
	$g[3]h[1]$	$g[0]h[1]$	$g[1]h[1]$	$g[2]h[1]$
	$g[2]h[2]$	$g[3]h[2]$	$g[0]h[2]$	$g[1]h[2]$
	$g[1]h[3]$	$g[2]h[3]$	$g[3]h[3]$	$g[0]h[3]$
$y_C[n]$:	$y_C[0]$	$y_C[1]$	$y_C[2]$	$y_C[3]$

- The samples of the sequence $\{y_c[n]\}$ are obtained by adding the 4 partial products in the column above of each sample system → linear convolution

Linear Convolution Using the DFT

- Linear convolution is a key operation in many signal processing applications.
- Since a DFT can be efficiently implemented using FFT algorithms, it is of interest to develop methods for the implementation of linear convolution using the DFT.
- Let $g[n]$ and $h[n]$ be two finite-length sequences of length *N* and *M*, respectively. Denote *L*=*N*+*M*-1. Define two length-*L* sequences

$$g_e[n] = \begin{cases} g[n], & 0 \le n \le N-1 \\ 0, & N \le n \le L-1 \end{cases}$$

$$h_e[n] = \begin{cases} h[n], & 0 \le n \le M-1 \\ 0, & M \le n \le L-1 \end{cases}$$

Linear Convolution Using the DFT

linear convolution　　　　circular convolution!

this two are the same

- Then

$$y_L[n] = g[n] \circledast h[n] = y_C[n] = g_e[n] \textcircled{\scriptsize L} h_e[n]$$

- The corresponding implementation scheme is illustrated below