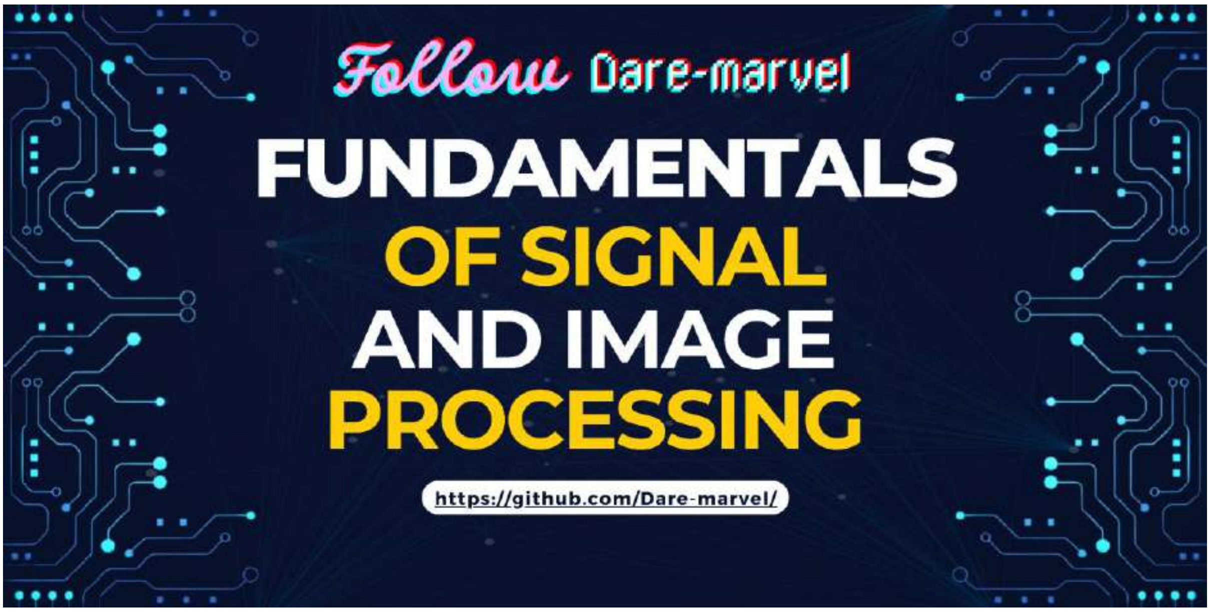
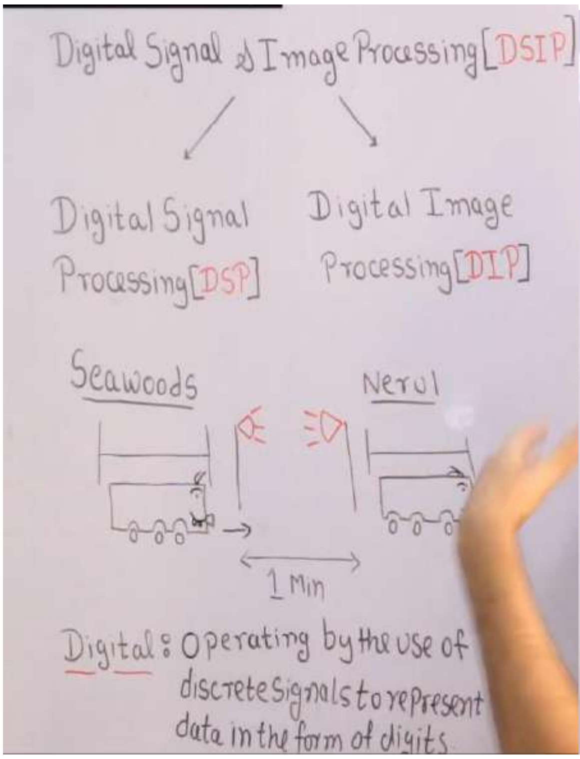
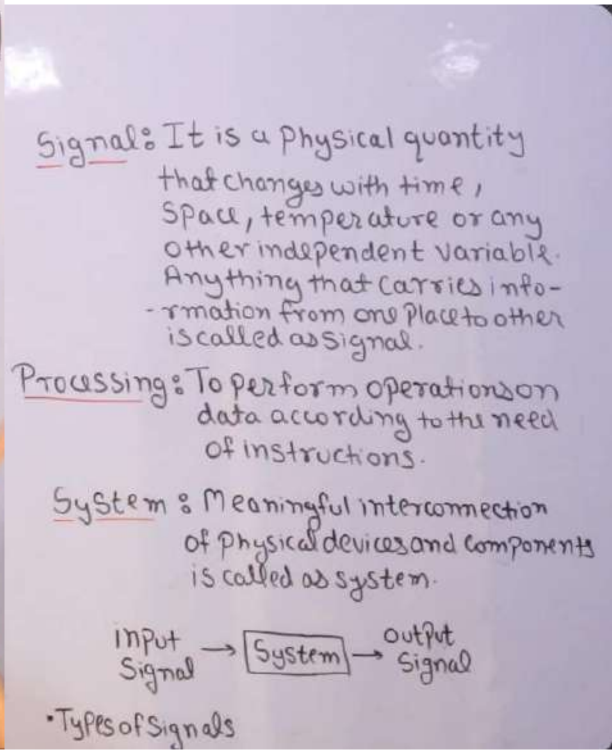


Introduction to Digital Signal Processing





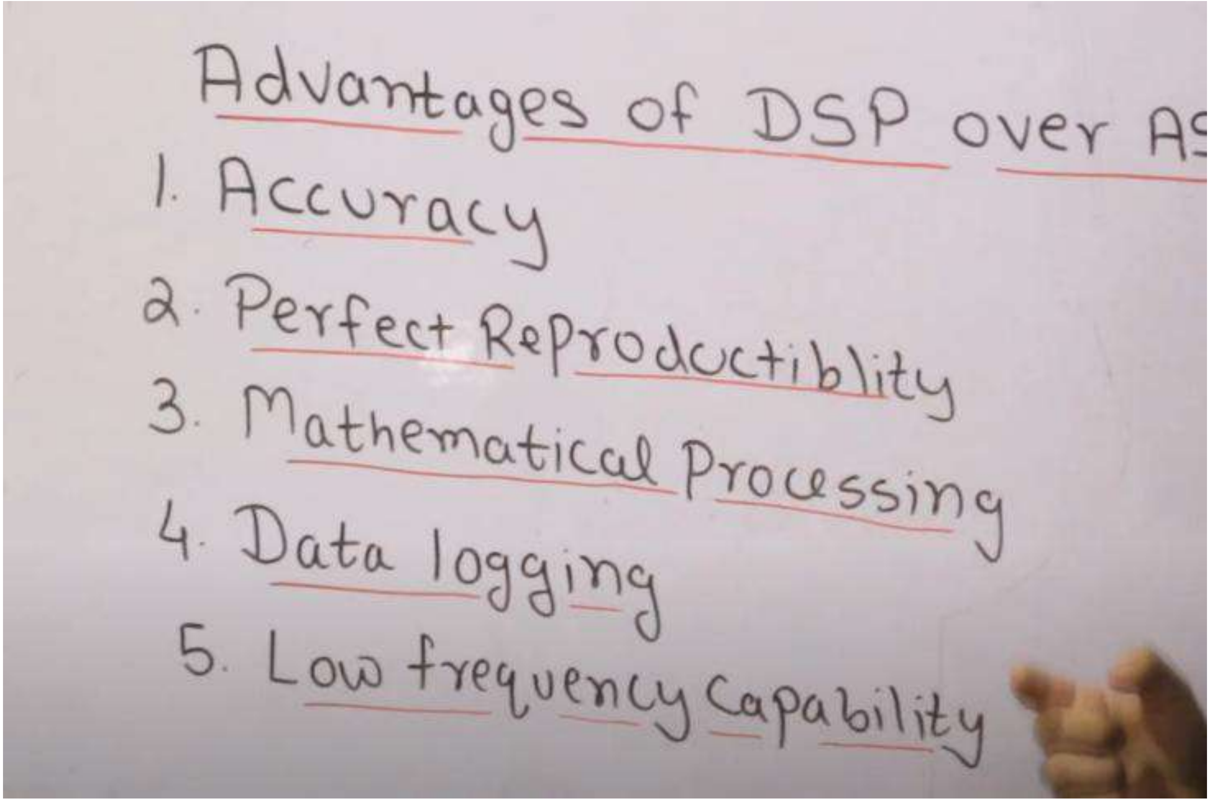
Difference between Analog and Digital Signal

	Analog Signals	Digital Signals
1.	It is a continues signal which represents a measurement	They are discrete time signals which are generated by digital modulation.
2.	More affected by noise	Less affected by noise
3.	Analog signals are represented by sign waves	They are represented by square waves
4.	less accurate	More accurate
5.	less costly	More costly
	eg Voice signals, audio, video	eg wifi, remote signals

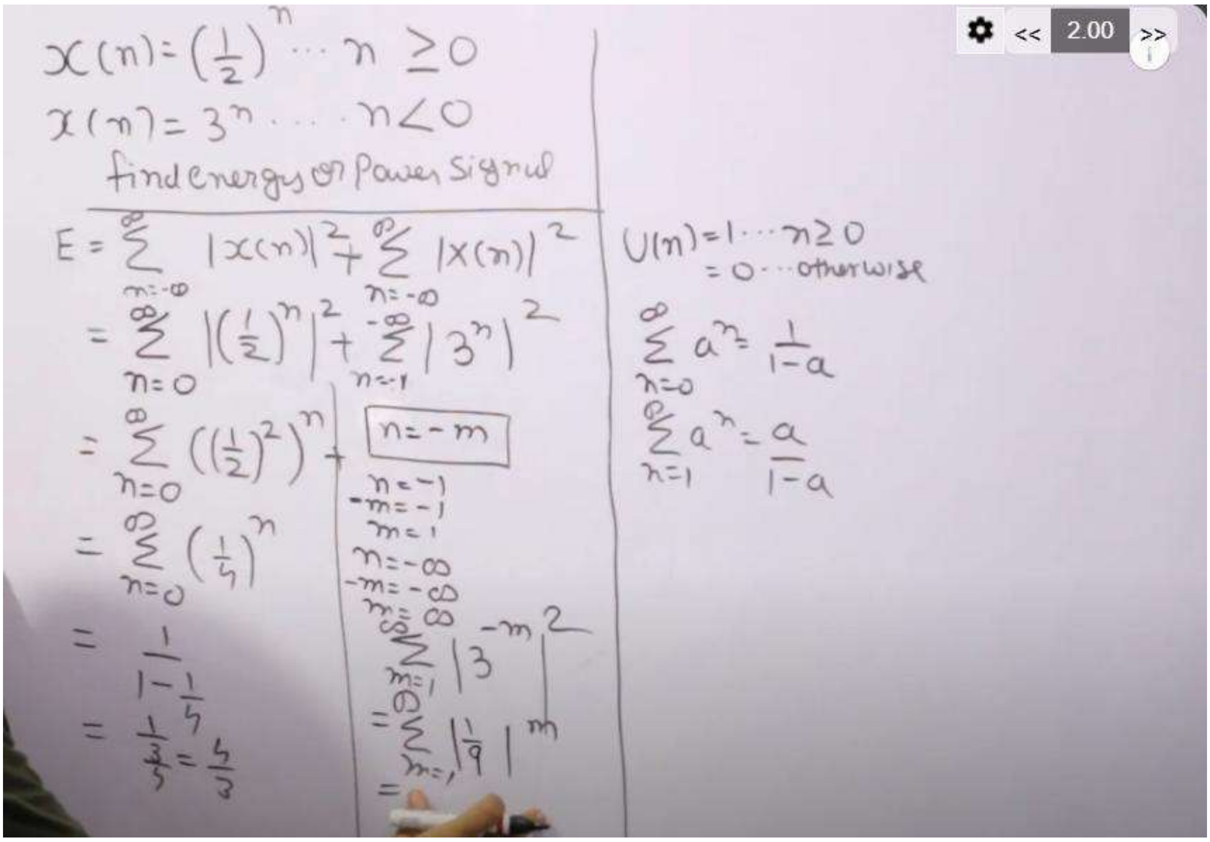
applications of dsp

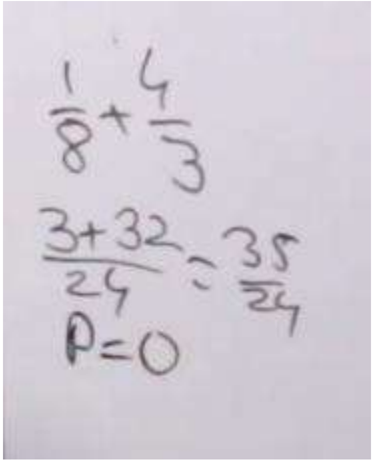
1. SONAR
2. Seismology
3. Military radar
4. Mobile & Telecommunication
5. Speech Processing

advantages of dsp over asp



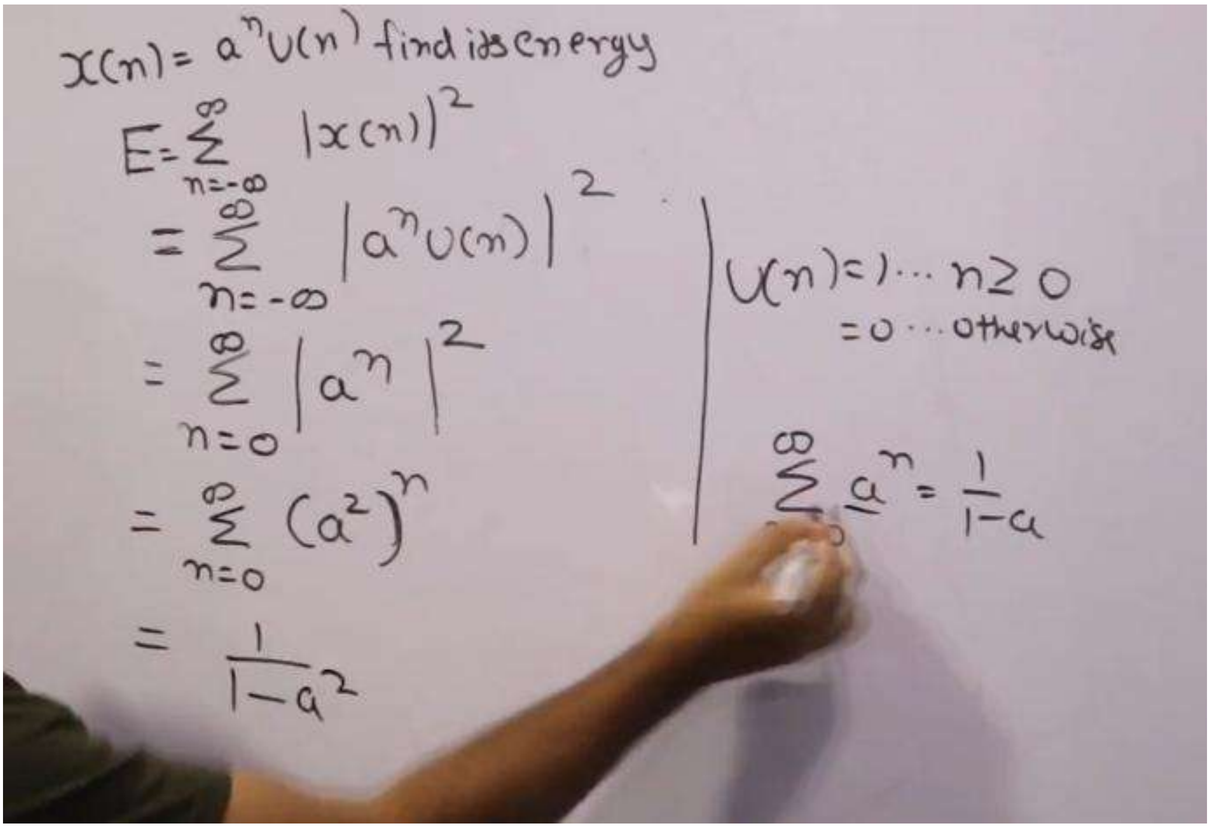
energy and power signals examples

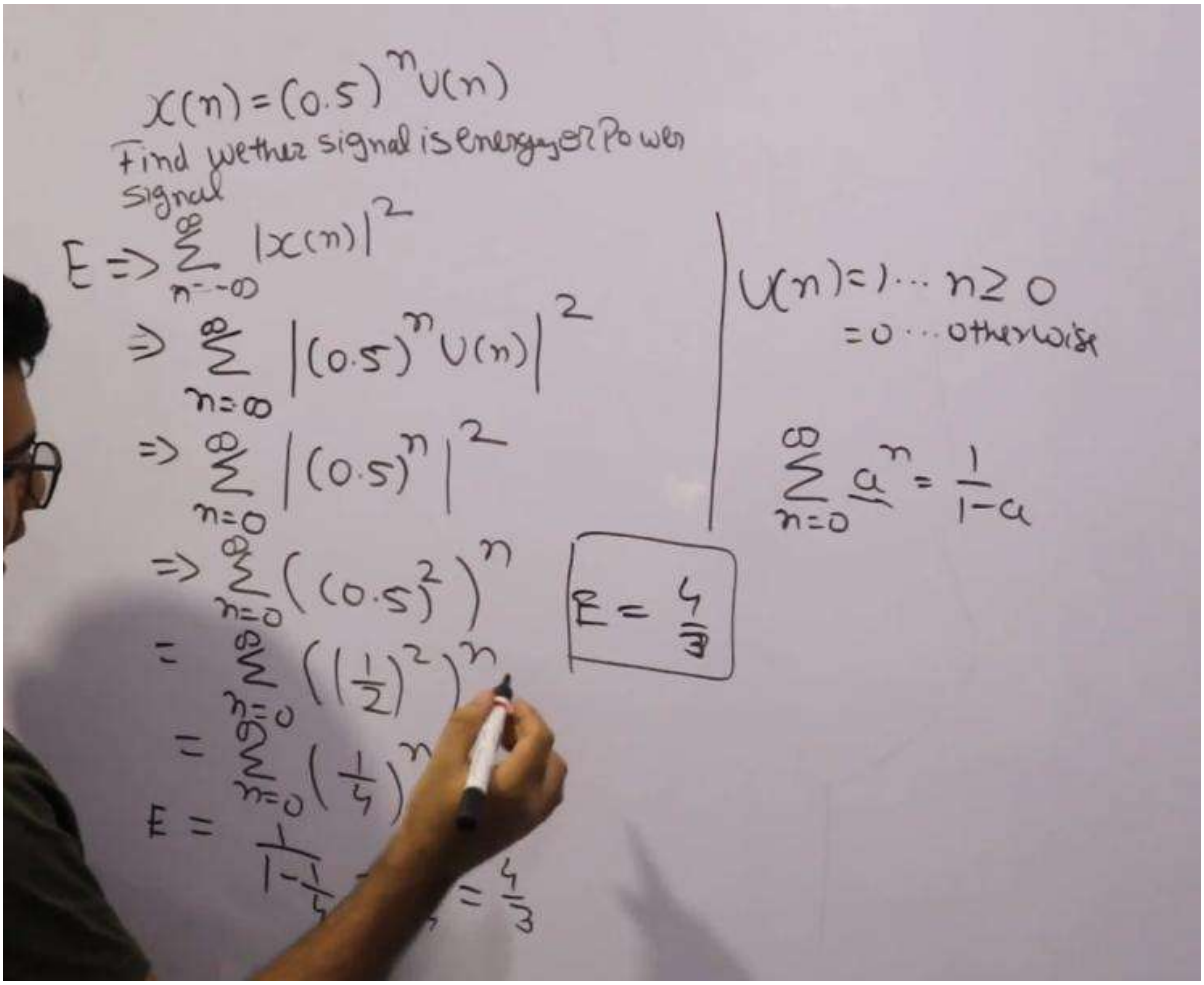




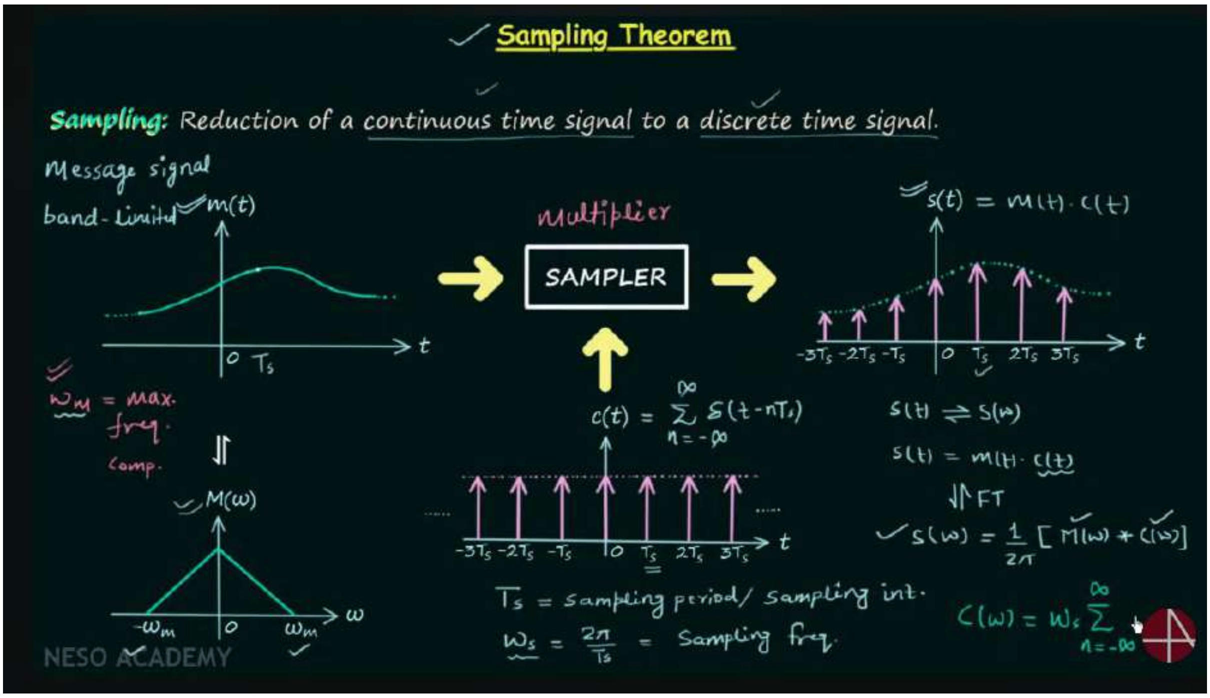
As Energy is finite power is 0

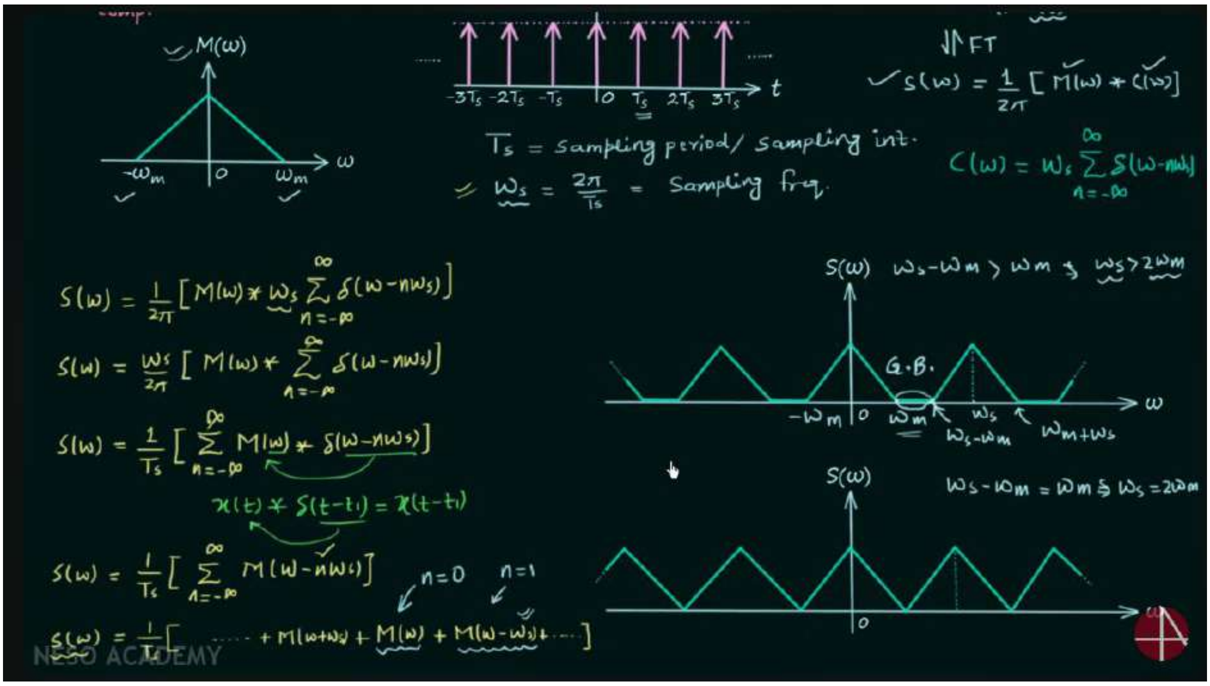
energy and power signals- SOLVED problems/examples





Sampling Theorem



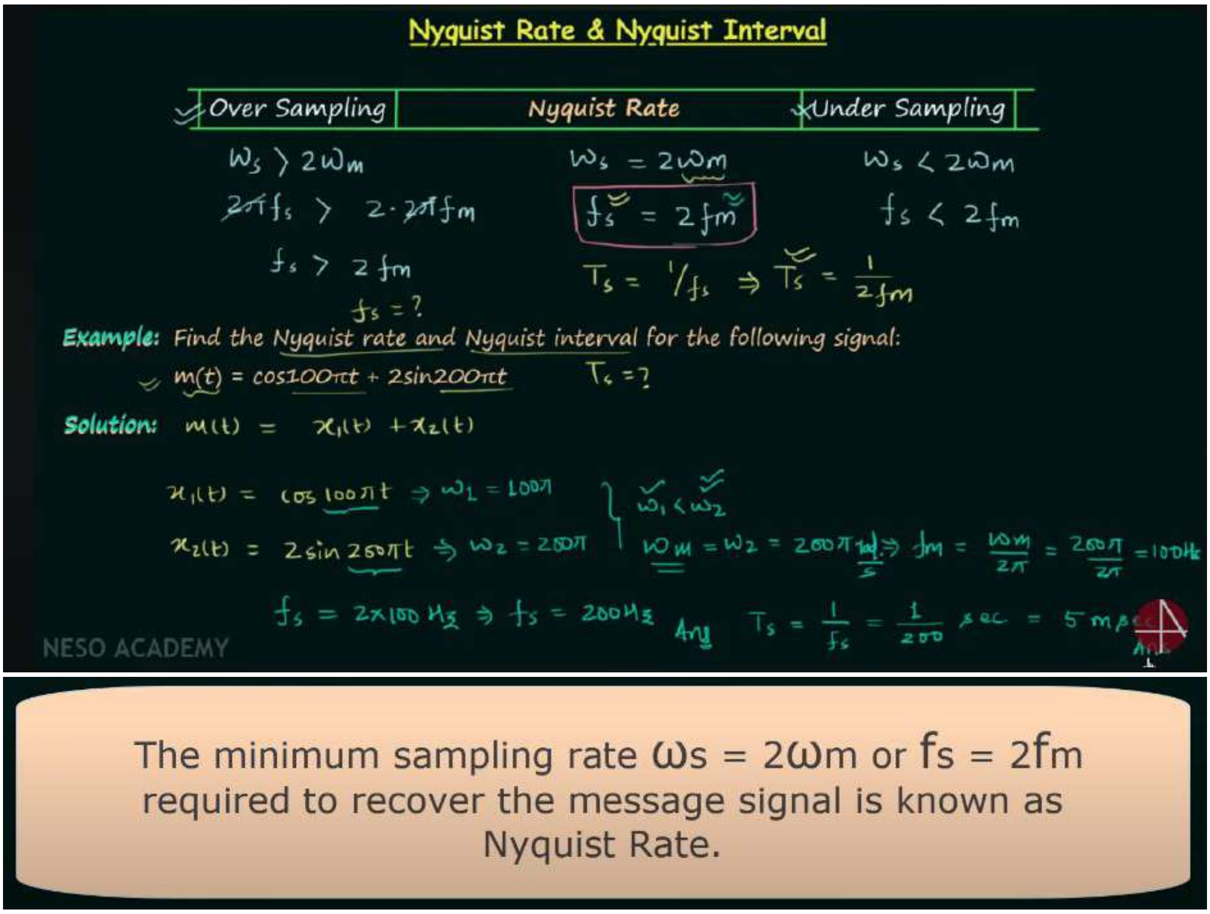


Sampling Theorem:

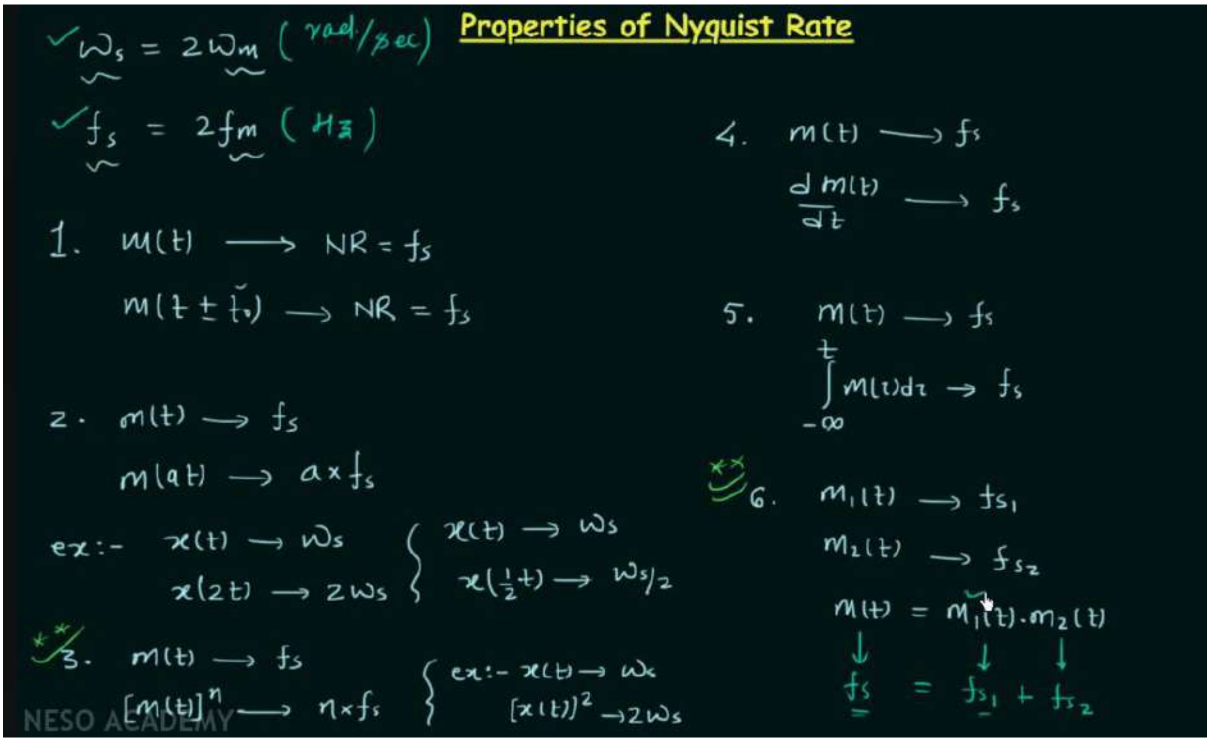
A signal can be represented in its samples and can be recovered back when sampling frequency ω_s is greater or equal to twice of maximum frequency component present in the signal $2\omega_m$

$$\boxed{\omega_s \geq 2\omega_m}$$

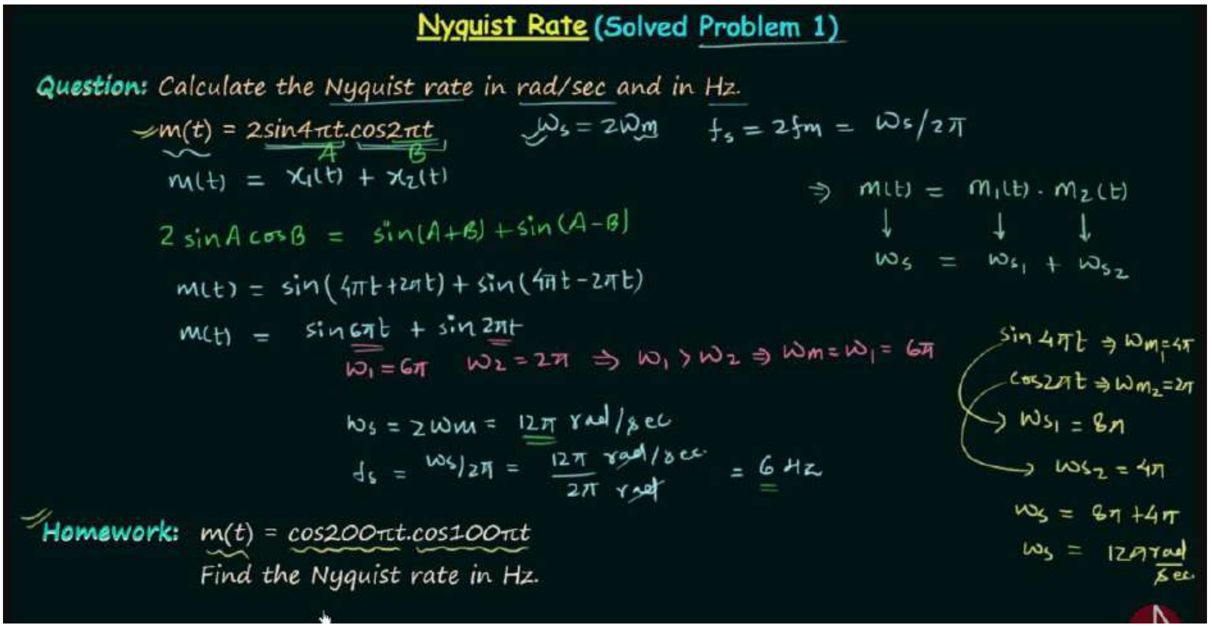
Nyquist Rate & Nyquist Interval



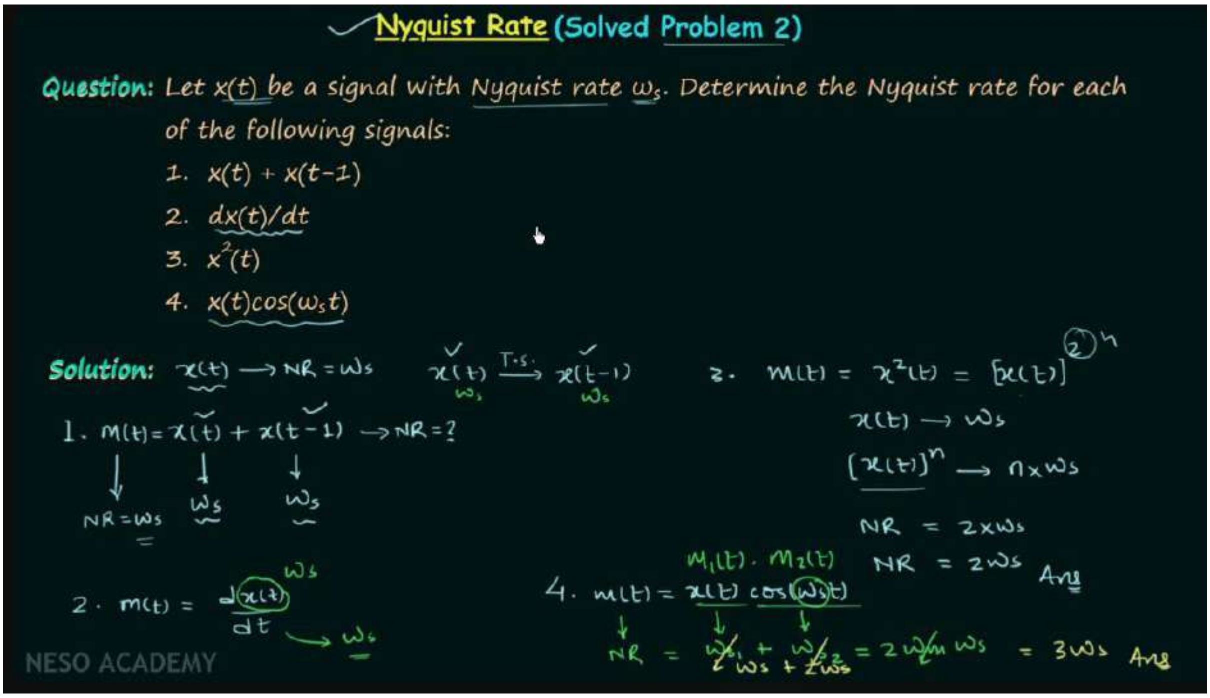
Properties of Nyquist Rate



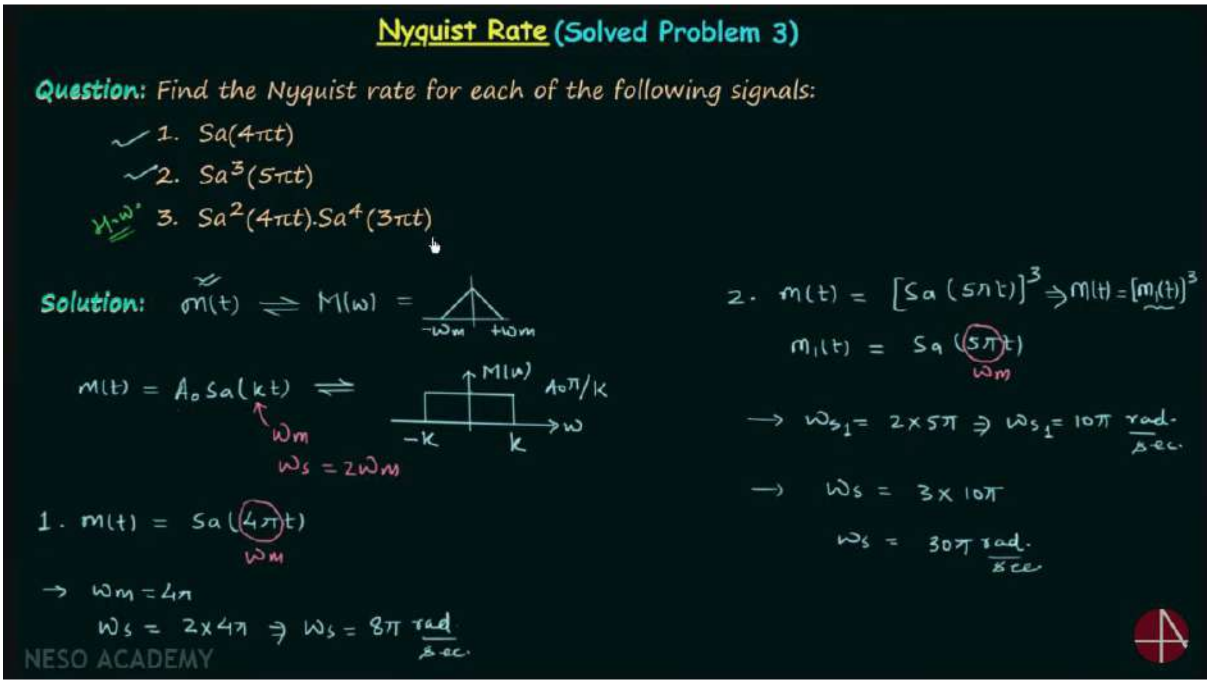
Nyquist Rate (Solved Problem 1)



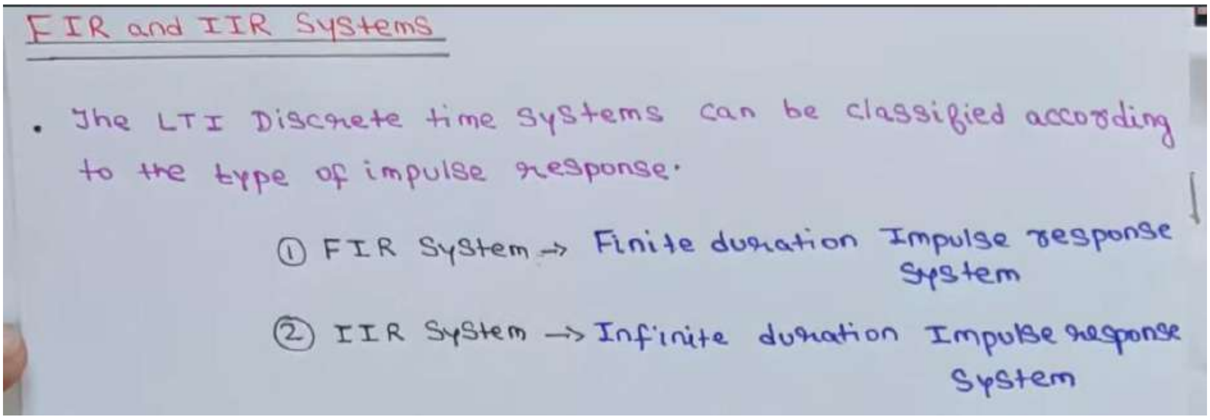
Nyquist Rate (Solved Problem 2)



Nyquist Rate (Solved Problem 3)



FIR and IIR Systems



• FIR System (Finite impulse response System)

→ In FIR System, the impulse response consists of finite number of samples.

→ If the impulse response sequence is of finite duration, the system is called a finite impulse response system.

→ Example:

$$h(n) = \begin{cases} -1 & n = 3,4 \\ 1 & n = 1,2 \\ 0 & \text{otherwise} \end{cases}$$

→ The convolution formula for FIR System:

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

where, $h(n) = 0$; for $n < 0$ and $n \geq N$ ✓

→ FIR System is described by the difference equation:

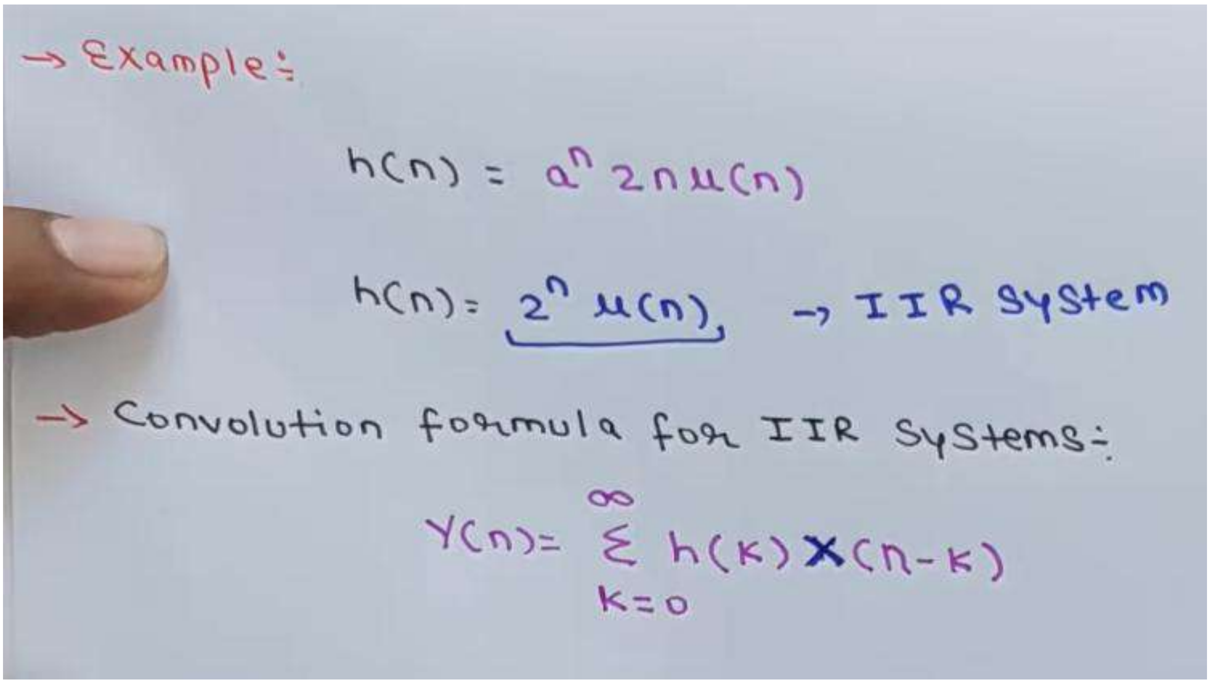
$$y(n) = \sum_{K=0}^{N-1} b_K\, x(n-K)$$

where $b_K = h(K)$ for $K = 0$ to $K = N-1$

• IIR System (Infinite duration Impulse Response System)

→ If the impulse response sequence is of infinite duration then the system is called an IIR System. ✓

→ In IIR System, the impulse response has infinite number of Samples.



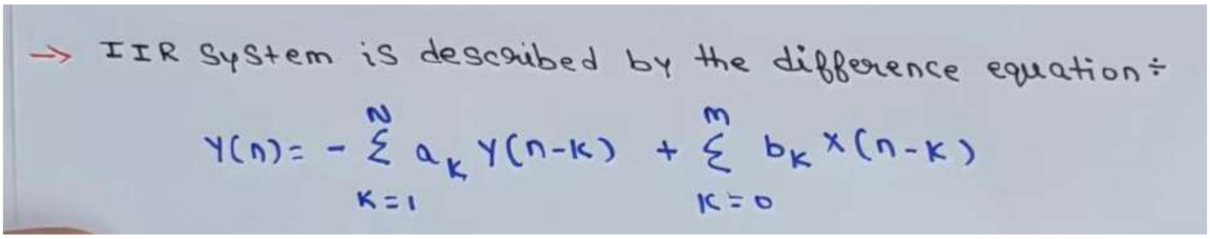
→ Example:

$$h(n) = a^n 2n\,u(n)$$

$$h(n) = \underbrace{2^n\,u(n),} \rightarrow \text{IIR System}$$

→ Convolution formula for IIR Systems:

$$y(n) = \sum_{k=0}^{\infty} h(k)\,x(n-k)$$

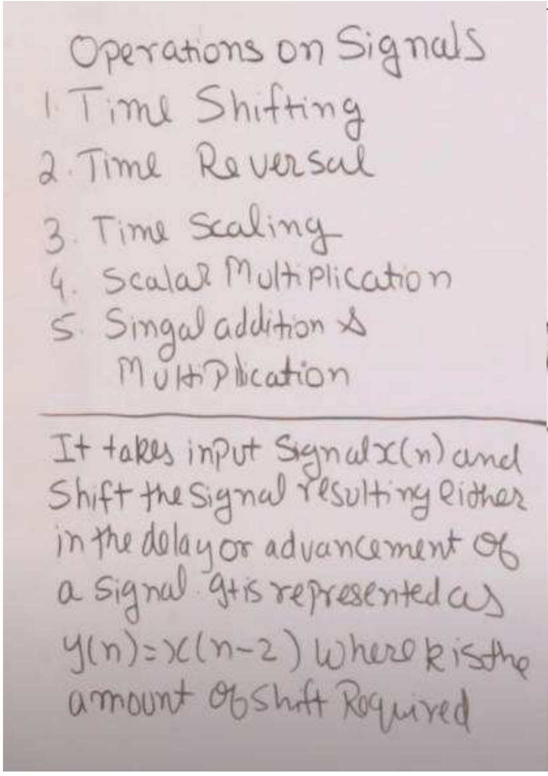


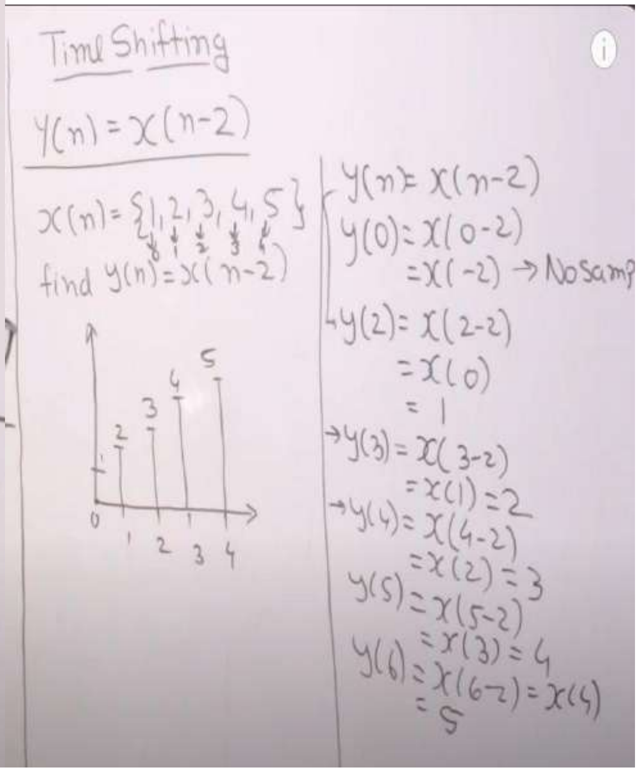
→ IIR System is described by the difference equation:

$$y(n) = -\sum_{k=1}^{N} a_k\,y(n-k) + \sum_{k=0}^{m} b_k\,x(n-k)$$

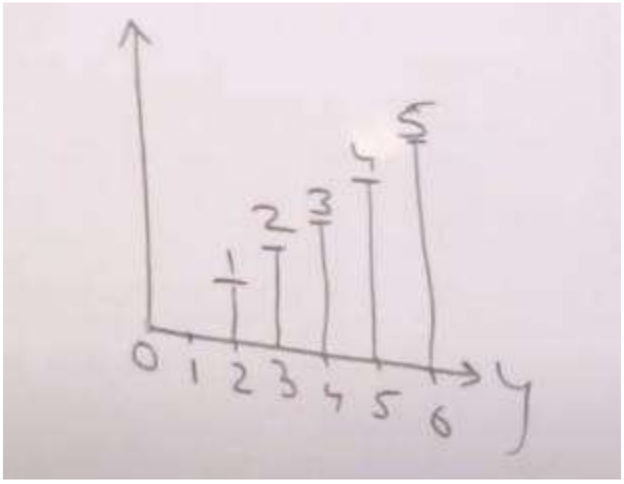
Operations On Signals

Time Shifting

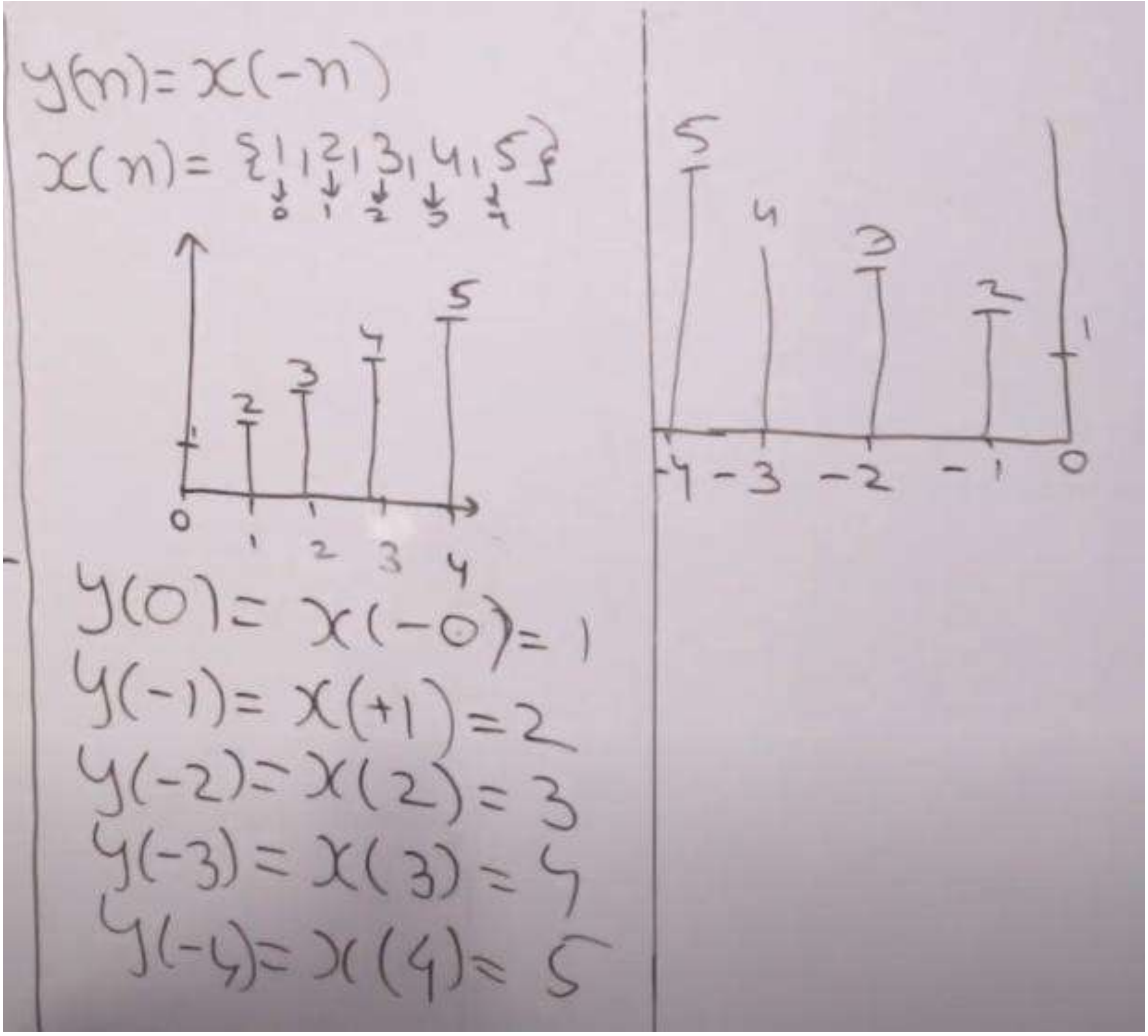




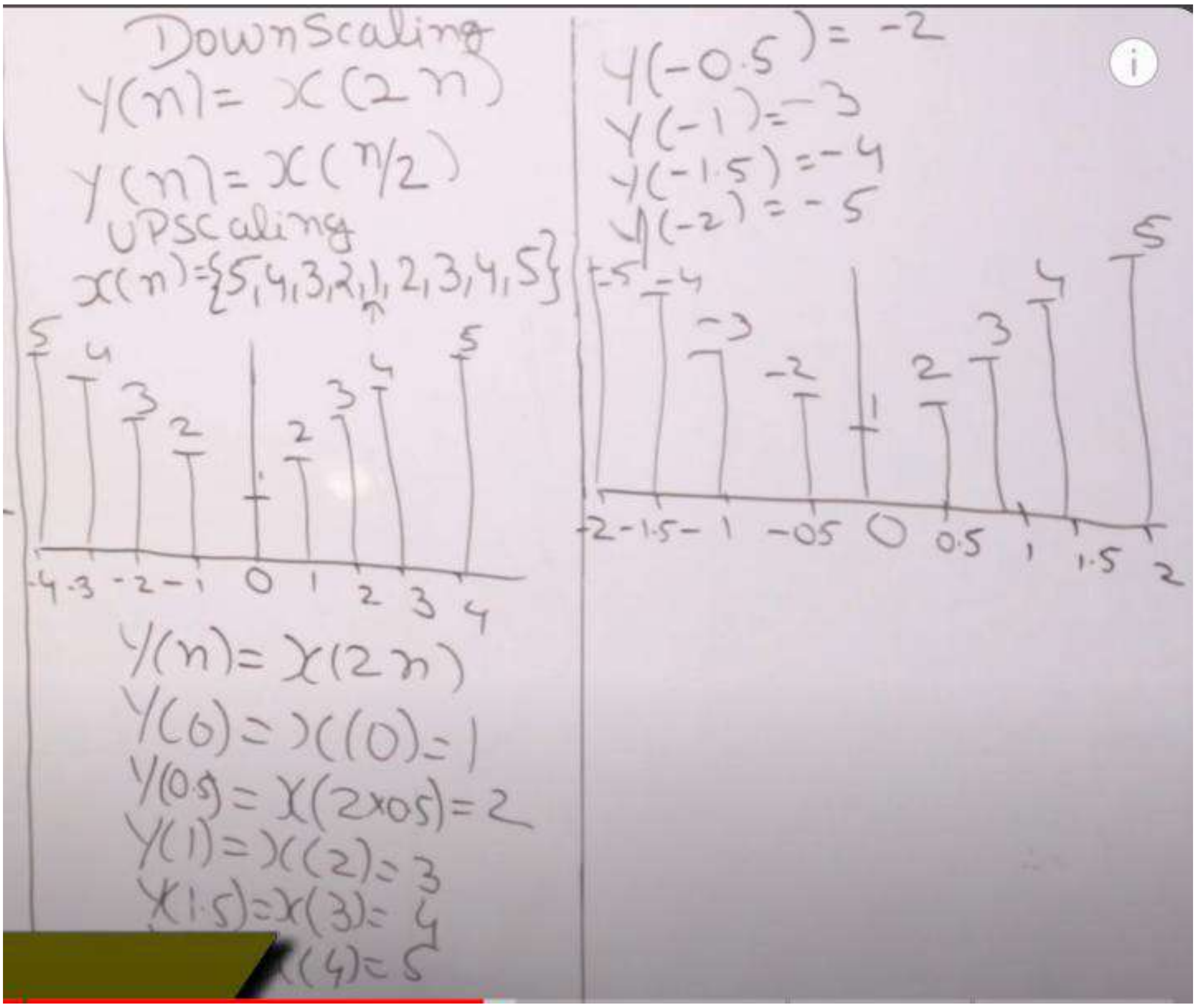
Final Output



Time Reversal



Time Scaling



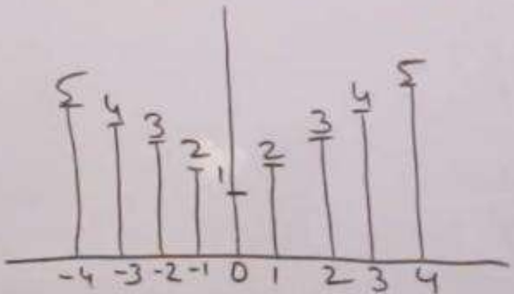
DownScaling

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

$y(n) = x(n/2)$

upscaling

$x(n) = \{5, 4, 3, 2, 1, 2, 3, 4, 5\}$



$y(n) = x(n/2)$

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

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

$y(4) = x(2) = 3$

$y(6) = x(3) = 4$

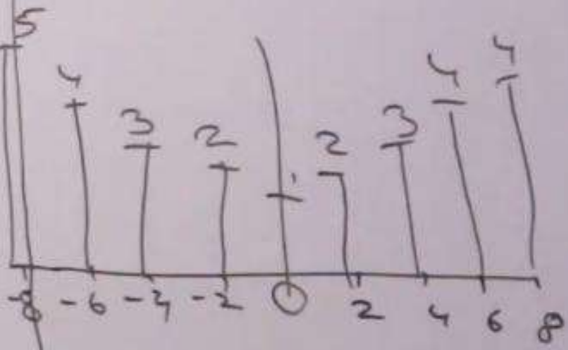
$y(8) = x(4) = 5$

$y(-2) = 2$

$y(-4) = 3$

$y(-6) = 4$

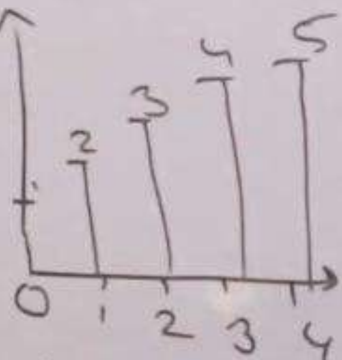
$y(-8) = 5$



Scalar Multiplication

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

$x(n) = \{\underset{\uparrow}{1}, 2, 3, 4, 5\}$



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

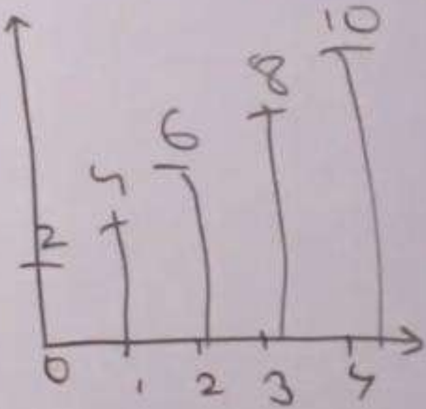
$y(0) = 2 \times 1 = 2$

$y(1) = 2 \times 2 = 4$

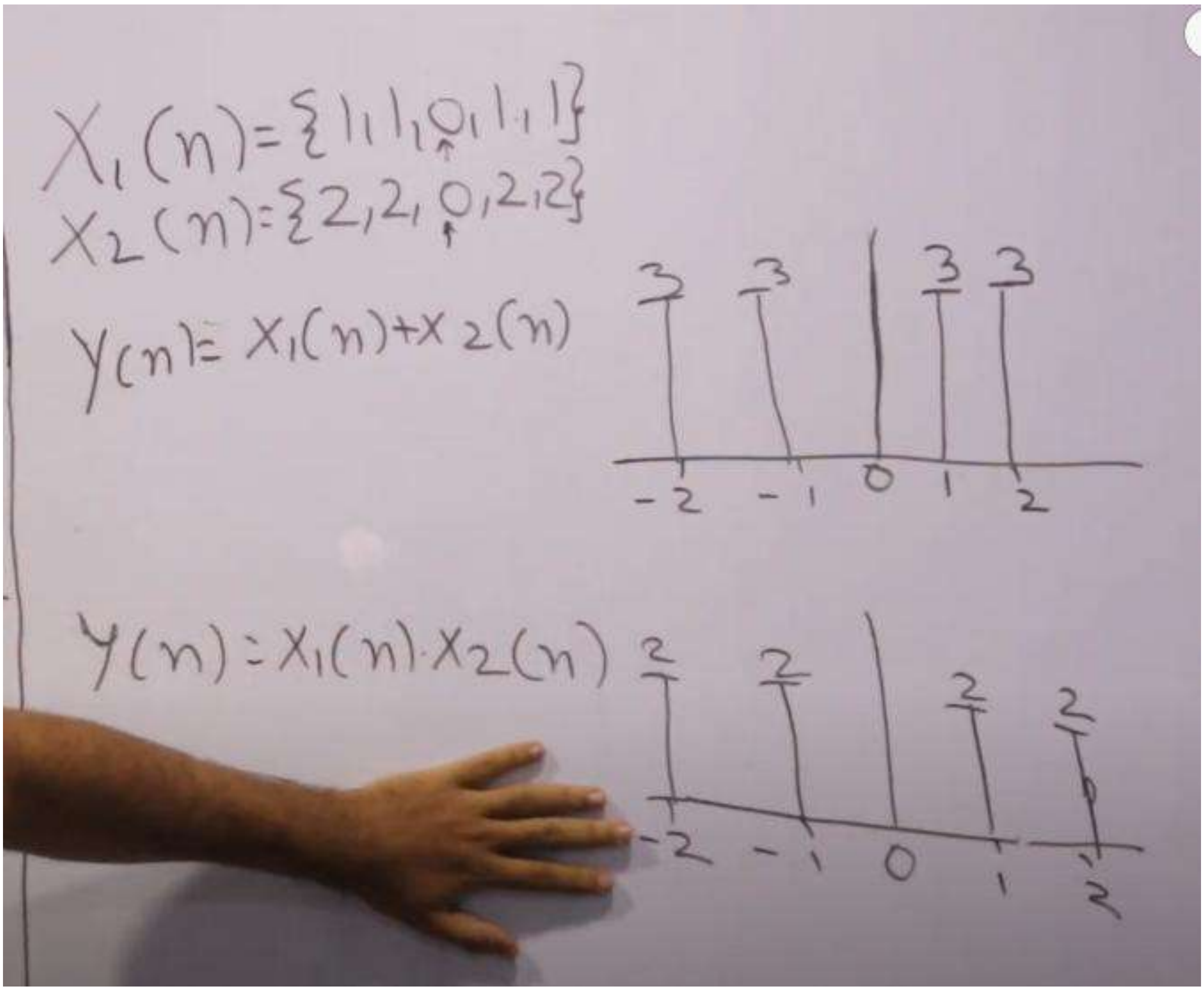
$y(2) = 2 \times 3 = 6$

$y(3) = 2 \times 4 = 8$

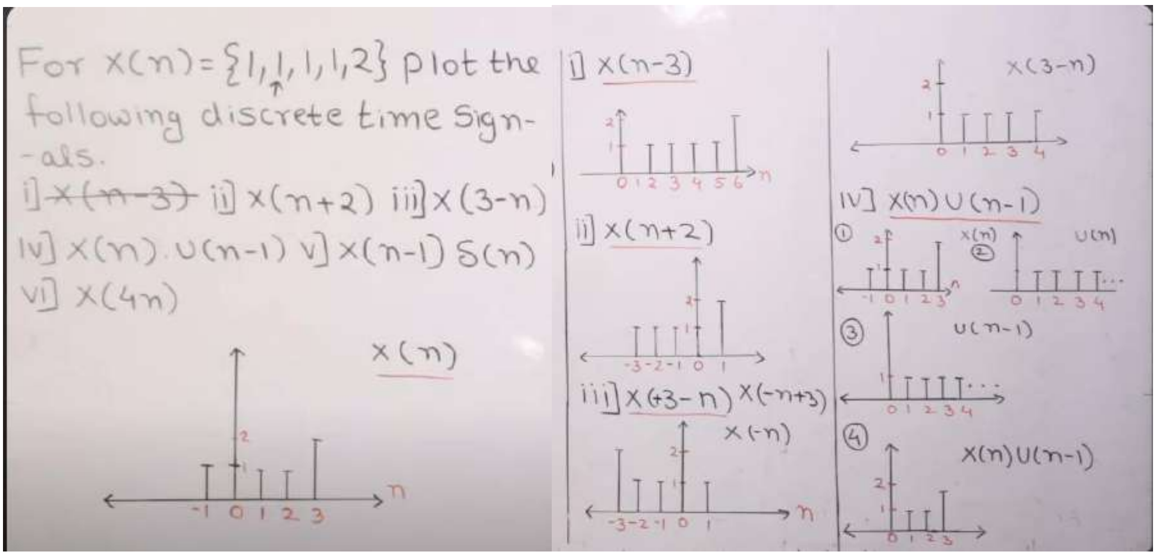
$y(4) = 2 \times 5 = 10$

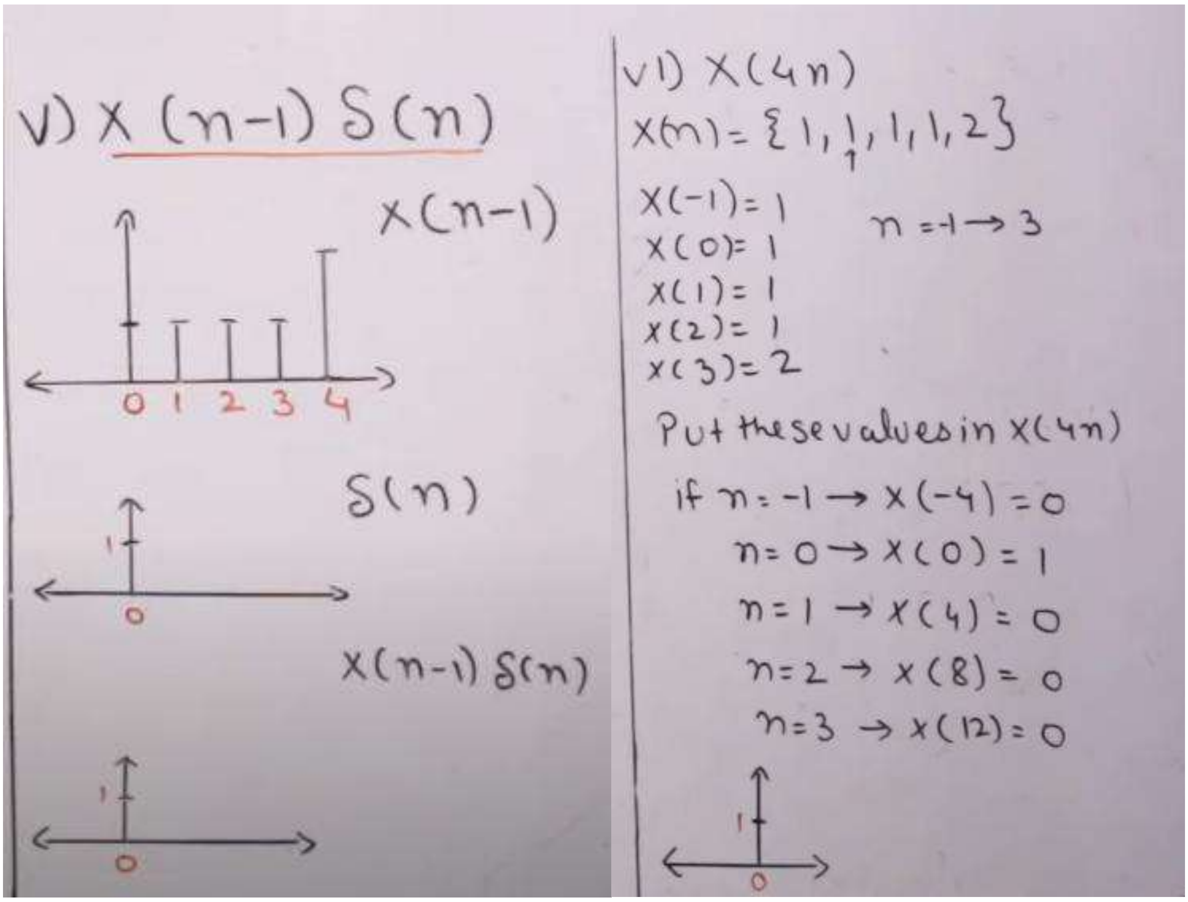


Addition and Multiplication



plot discrete time signals





Auto Correlation

Auto Correlation

- Autocorrelation is also known as Serial Correlation.
- Autocorrelation provides a measure of similarity between a signal and itself at a given lag.
- It is the correlation of a signal with a delayed copy of itself as a function of delay.
- It is used for finding repeating patterns, such as the presence of a periodic signal concealed by noise.

Formula of Auto correlation:

$$r_{xx}(n) = \sum_{k=-\infty}^{k=\infty} x(k) * x(k-n)$$

No. of Terms & Range:

Let x(n) be a signal having N length.

Total number of terms in auto correlation are **2*N-1**

Range: **-(N-1) to (N-1)**

Ex.1 Perform Auto correlation for the given sequence $x(n)=\{\underset{\uparrow}{1}, 2, 3, 4\}$.

No. of Terms & Range:

Total number of terms in auto correlation are 4*2-1=7

Location	**0**	1	2	3
Values	1 ↑	2	3	4
Location	**3**	2	1	0

0+3=3.

Range: -3 to 3

Auto Correlation formula

$$y(n) = \sum_{k=-\infty}^{k=\infty} x(k) * x(k-n)$$

1) Matrix Method:

$$
\begin{array}{r|cccc}
 & \downarrow & & & \\
 & 1 & 2 & 3 & 4 \\
\hline
4 & 4 & 8 & 12 & 16 \\
3 & 3 & 6 & 9 & 12 \\
2 & 2 & 4 & 6 & 8 \\
\rightarrow 1 & 1 & 2 & 3 & 4
\end{array}
$$

$$y(n) = \{4, 11, 20, \underset{\uparrow}{30}, 20, 11, 4\}$$

There are various methods to put the arrow, As we can see the range of the signal is from -3 to 3 therefore counting from 4 which is index -3 we go ahead and stop at 30 which is at index 0.

Another method is, we look in the matrix the position of first arrow is at 0 and the second arrow is at index 3, therefore we take the number at index 0 + 3 in y(n)

2) Graphical Method:

$$y(n) = \sum_{k=-\infty}^{k=\infty} x(k) * x(k-n)$$

Range: -3 to 3

$$y(-3) = \sum_{k=-\infty}^{k=\infty} x(k) * x(k+3)$$

$$\begin{array}{cccccccc} 0 & 0 & 0 & 0 & 1 & 2 & 3 & 4 \\ & & & & \uparrow & & & \end{array}$$

$$y(-3) = \begin{array}{ccccccc} & & & 1 & 2 & 3 & 4 \\ & & & \uparrow & & & \\ 1 & 2 & 3 & 4 & & & \\ & & & \uparrow & & & \end{array}$$

$$y(-3) = 4$$

$$y(-2) = \begin{array}{cccccc} & & 1 & 2 & 3 & 4 \\ & & \uparrow & & & \\ 1 & 2 & 3 & 4 & & \\ & & \uparrow & & & \end{array}$$

$$y(-2) = 1 \times 3 + 2 \times 4 = 11$$

Range: -2 to 3

$$y(-2) = \sum_{k=-\infty}^{k=\infty} x(k) * x(k+2)$$

$$\begin{array}{cccccccc} 0 & 0 & 0 & 0 & 1 & 2 & 3 & 4 \\ & & & & \uparrow & & & \end{array}$$

$$y(-1) = \begin{array}{ccccc} & 1 & 2 & 3 & 4 \\ & \uparrow & & & \\ 1 & 2 & 3 & 4 & \\ & \uparrow & & & \end{array}$$

$$y(-1) = 1 \times 2 + 2 \times 3 + 3 \times 4 = 20$$

Range: -1 to 3

$$y(-1) = \sum_{k=-\infty}^{k=\infty} x(k) * x(k+1)$$

$$\begin{array}{l} \quad\quad 1 \quad 2 \quad 3 \quad 4 \\ \quad\quad \uparrow \\ y(0) = \\ \quad\quad 1 \quad 2 \quad 3 \quad 4 \\ \quad\quad \uparrow \end{array}$$

$$y(0) = 1\times 1 + 2\times 2 + 3\times 3 + 4\times 4 = 30$$

Range: 0 to 3

$$y(0) = \sum_{k=-\infty}^{k=\infty} x(k) * x(k)$$

$$\begin{array}{cccccccc} 0 & 0 & 0 & 0 & 1 & 2 & 3 & 4 \\ & & & & \uparrow & & & \end{array}$$

$$\begin{array}{l} \quad\quad 1 \quad 2 \quad 3 \quad 4 \\ \quad\quad \uparrow \\ y(1) = \\ \quad\quad 0 \quad 1 \quad 2 \quad 3 \quad 4 \\ \quad\quad \uparrow \end{array}$$

$$y(1) = 1\times 0 + 2\times 1 + 3\times 2 + 4\times 3 = 20$$

Range: 0 to 4

$$y(1) = \sum_{k=-\infty}^{k=\infty} x(k) * x(k-1)$$

$$\begin{array}{l} \quad\quad 1 \quad 2 \quad 3 \quad 4 \\ \quad\quad \uparrow \\ y(2) = \\ \quad\quad 0 \quad 0 \quad 1 \quad 2 \quad 3 \quad 4 \\ \quad\quad \uparrow \end{array}$$

$$y(2) = 1\times 3 + 4\times 2 = 11$$

Range: 0 to 5

$$\begin{array}{l} \quad\quad 1 \quad 2 \quad 3 \quad 4 \\ \quad\quad \uparrow \\ y(3) = \\ \quad\quad 0 \quad 0 \quad 0 \quad 1 \quad 2 \quad 3 \quad 4 \\ \quad\quad \uparrow \end{array}$$

$$y(3) = 4\times 1 = 4$$

Range: 0 to 6

$$\begin{array}{cccccccc} 0 & 0 & 0 & 0 & 1 & 2 & 3 & 4 \\ & & & & \uparrow & & & \end{array}$$

$$y(2) = \sum_{k=-\infty}^{k=\infty} x(k) * x(k-2) \qquad y(3) = \sum_{k=-\infty}^{k=\infty} x(k) * x(k-3)$$

3) Tabular Method:

	-4	-3	-2	-1	0	1	2	3	4			
$x(k)$					1	2	3	4				
					↑							
$x(k+3)$		1	2	3	4							y(-3)=4
					↑							
$x(k+2)$			1	2	3	4						y(-2)=11
					↑							
$x(k+1)$				1	2	3	4					y(-1)=20
					↑							
$x(k)$					1	2	3	4				y(0)=30
					↑							
$x(k-1)$					0	1	2	3	4			y(1)=20
					↑							
$x(k-2)$					0	0	1	2	3	4		y(2)=11
					↑							
$x(k-3)$					0	0	0	1	2	3	4	y(3)=4

4) Summation Method:

$$y(n) = \sum_{k=-\infty}^{k=\infty} x(k) * x(k-n)$$

Range: -3 to 3 **Range: 0 to 4**

Range: -2 to 3 **Range: 0 to 5**

Range: -1 to 3 **Range: 0 to 6**

$$y(-3) = \sum_{k=-\infty}^{k=\infty} x(k) * x(k+3)$$

Range: 0 to 3 Signal coincides exactly

$$r_{xx}(-3) = x(-3) * x(0) + x(-2) * x(1) + x(-1) * x(2) + \mathbf{x(0) * x(3)} + x(1) * x(4) + x(2) * x(5) + x(3) * x(6)$$

$$r_{xx}(-3) = 1 \times 4 = 4$$

$$\begin{matrix} 0 & 0 & 0 & 0 & 1 & 2 & 3 & 4 \\ & & & & \uparrow & & & \end{matrix}$$

Ex.2 Perform Auto correlation for the given sequence

$$x(n) = \{2, 4, \underset{\uparrow}{6}, 8\}$$

No. of Terms & Range:

Total number of terms in auto correlation are 4*2-1=7

Location	0	1	2	3
Values	1	2	3 ↑	4
Location	3	2	1	0

2+1=3.

Range: -3 to 3

Ex.2 Perform Auto correlation for the given sequence

$$x(n) = \{2, 4, \underset{\uparrow}{6}, 8\}$$

No. of Terms & Range:

Total number of terms in auto correlation are 4*2-1=7

Location	0	1	2	3
Values	1	2	3 ↑	4
Location	3	2	1	0

2+1=3.

Range: -3 to 3

The values in the table are wrong they are actually 2,4,6,8

1) Matrix Method:

$$
\begin{array}{c|cccc}
 & 2 & 4 & \overset{\downarrow}{6} & 8 \\
\hline
8 & 16 & 32 & 48 & 64 \\
\rightarrow 6 & 12 & 24 & 36 & 48 \\
4 & 8 & 16 & 24 & 32 \\
2 & 4 & 8 & 12 & 64
\end{array}
$$

$$y(n) = \{16, 44, 80, 120, 80, 44, 16\}$$

2) Graphical Method:

$$0 \quad 0 \quad 2 \quad 4 \quad \underset{\uparrow}{6} \quad 8 \quad 0 \quad 0$$

$$y(n) = \sum_{k=-\infty}^{k=\infty} x(k) * x(k-n)$$

$$y(-3) = \sum_{k=-\infty}^{k=\infty} x(k) * x(k+3)$$

Range: -5 to 1

$$
\boxed{
\begin{array}{l}
\qquad\qquad\qquad\qquad\qquad 2 \quad 4 \quad \underset{\uparrow}{6} \quad 8 \\
y(-3) = 2 \quad 4 \quad 6 \quad 8 \quad 0 \quad \underset{\uparrow}{0} \quad 0 \\
y(-3) = 8 \times 2 = 16
\end{array}
}
$$

$$
y(-2) = \begin{array}{cccccc}
 & & 2 & 4 & 6 & 8 \\
 & & & & \uparrow & \\
2 & 4 & 6 & 8 & 0 & \\
 & & & & \uparrow &
\end{array}
$$

$$y(-2) = 2\times 6 + 4\times 8 = 12 + 32 = 44$$

Range: -4 to 1

$$
\begin{array}{cccccccc}
0 & 0 & 2 & 4 & 6 & 8 & 0 & 0 \\
 & & & & \uparrow & & &
\end{array}
$$

$$y(-2) = \sum_{k=-\infty}^{k=\infty} x(k) * x(k+2)$$

$$y(-1) = \sum_{k=-\infty}^{k=\infty} x(k) * x(k+1)$$

$$
y(-1) = \begin{array}{ccccc}
 & 2 & 4 & 6 & 8 \\
 & & & \uparrow & \\
2 & 4 & 6 & 8 & \\
 & & & \uparrow &
\end{array}
$$

$$y(-1) = 2\times 4 + 4\times 6 + 6\times 8$$

$$y(-1) = 8 + 24 + 48$$

$$y(-1) = 80$$

Range: -3 to 1

$$
y(0) = \begin{array}{cccc}
2 & 4 & 6 & 8 \\
 & & \uparrow & \\
2 & 4 & 6 & 8 \\
 & & \uparrow &
\end{array}
$$

$$y(0) = 2\times 2 + 4\times 4 + 6\times 6 + 8\times 8$$

$$y(0) = 4 + 16 + 36 + 64$$

$$y(0) = 120$$

Range: -2 to 1

$$
\begin{array}{cccccccc}
0 & 0 & 2 & 4 & 6 & 8 & 0 & 0 \\
 & & & & \uparrow & & &
\end{array}
$$

$$y(0) = \sum_{k=-\infty}^{k=\infty} x(k) * x(k)$$

$$y(1) = \sum_{k=-\infty}^{k=\infty} x(k) * x(k-1)$$

$$
y(1) = \begin{array}{ccccc}
2 & 4 & 6 & 8 & \\
 & & \uparrow & & \\
 & 2 & 4 & 6 & 8 \\
 & & \uparrow & &
\end{array}
$$

$$y(1) = 2\times 4 + 6\times 4 + 8\times 6$$

$$y(1) = 8 + 24 + 48$$

$$y(1) = 80$$

Range: -2 to 2

$$\begin{array}{l} \quad\quad\quad 2 \quad 4 \quad 6 \quad 8 \\ \quad\quad\quad\quad\quad\;\; \uparrow \\ y(2) = \\ \quad\quad\quad\quad\quad\;\; 2 \quad 4 \quad 6 \quad 8 \\ \quad\quad\quad\quad\quad\;\; \uparrow \end{array}$$

$y(2) = 2 \times 6 + 8 \times 4$

$y(2) = 12 + 32$

$y(2) = 44$

Range: -2 to 3

$$\begin{array}{cccccccc} 0 & 0 & 2 & 4 & 6 & 8 & 0 & 0 \\ & & & & \uparrow & & & \end{array}$$

$$y(2) = \sum_{k=-\infty}^{k=\infty} x(k) * x(k-2)$$

$$\begin{array}{l} \quad\quad\quad 2 \quad 4 \quad 6 \quad 8 \\ \quad\quad\quad\quad\quad\;\; \uparrow \\ y(3) = \\ \quad\quad\quad\quad\quad\;\; 0 \quad 2 \quad 4 \quad 6 \quad 8 \\ \quad\quad\quad\quad\quad\;\; \uparrow \end{array}$$

$y(3) = 8 \times 2 = 16$

Range: -2 to 4

$$y(3) = \sum_{k=-\infty}^{k=\infty} x(k) * x(k-3)$$

3) Tabular Method:

	-5	-4	-3	-2	-1	0	1	2	3			
$x(k)$	0	0	0	2	4	6	8	0	0	0		
						↑						
$x(k+3)$	2	4	6	8	0	0						y(-3)=16
						↑						
$x(k+2)$	0	2	4	6	8	0						y(-2)=44
						↑						
$x(k+1)$	0	0	2	4	6	8						y(-1)=80
						↑						
$x(k)$		0	0	2	4	6	8					y(0)=120
						↑						
$x(k-1)$					2	4	6	8				y(1)=80
						↑						
$x(k-2)$						2	4	6	8			y(2)=44
						↑						
$x(k-3)$						0	2	4	6	8		y(3)=16
						↑						

4) Summation Method:

Range: -5 to 1 **Range: -2 to 2**

Range: -4 to 1 **Range: -2 to 3**

Range: -3 to 1 **Range: -2 to 4**

Range: -2 to 1 Signal coincides exactly

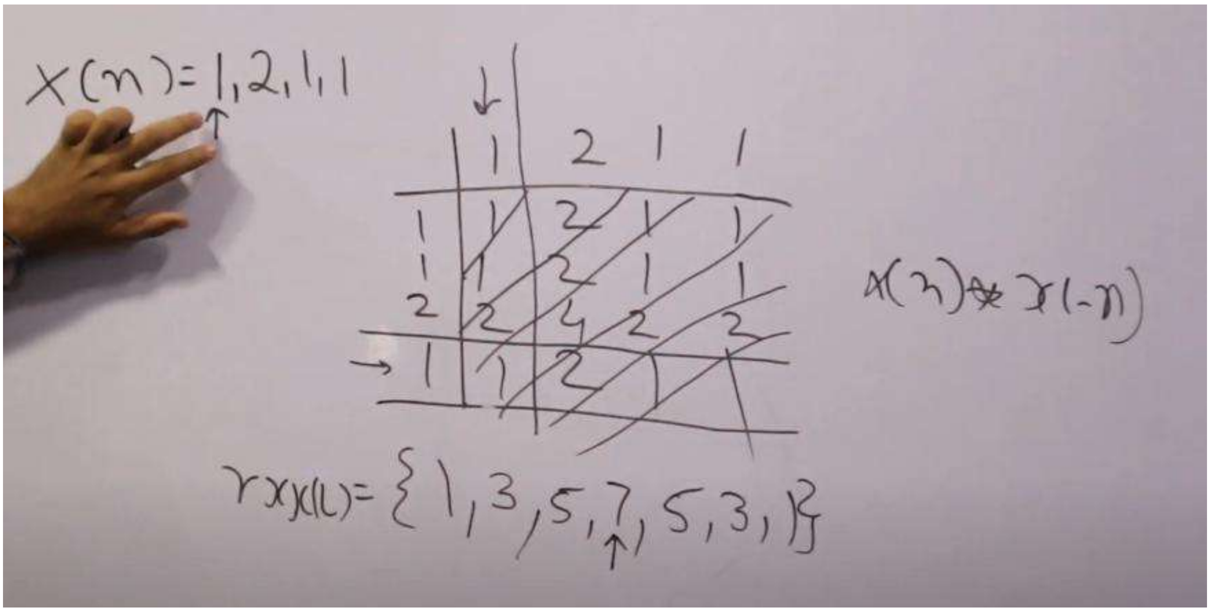
$$r_{xx}(n) = \sum_{k=-\infty}^{k=\infty} x(k) * x(k-n)$$

$$r_{xx}(-3) = \sum_{k=-\infty}^{k=\infty} x(k) * x(k+3)$$

$$r_{xx}(-3) = x(-5) * x(-2) + x(-4) * x(-1) + x(-3) * x(0) + \boldsymbol{x}(-\mathbf{2}) * \boldsymbol{x}(\mathbf{1}) + x(-1) * x(2) + x(0) * x(3) + x(1) * x(4)$$

$$r_{xx}(-3) = 2 \times 8 = 16$$

$$\begin{array}{cccccccc} 0 & 0 & 2 & 4 & 6 & 8 & 0 & 0 \\ & & & & \uparrow & & & \end{array}$$



Cross Correlation

Cross Correlation

In cross correlation, two different signal are correlated to get a third signal.

Cross-correlation measures the similarity between a signal x and shifted (lagged) copies of a signal y as a function of the lag.

Formula of Cross Correlation:

$$r_{xy}(n) = \sum_{k=-\infty}^{k=\infty} x(k) * y(k-n)$$

$$r_{yx}(n) = \sum_{k=-\infty}^{k=\infty} y(k) * x(k-n)$$

$$r_{xy}(n) \neq r_{yx}(n)$$

No. of Terms & Range:

Let x(n) and y(n) be a signal having length N1 and N2 respectively.

Total number of terms in cross correlation are **(N1+N2)-1**

Ex.1 Perform Cross correlation for the given sequence

$x(n)=\{1, 2, \underset{\uparrow}{3}, 4\}$ and $y(n)=\{2, 4, \underset{\uparrow}{6}\}$.

No. of Terms & Range r_{xy} :

Total number of terms in cross correlation are (4+3)-1=6

Location	0	1	2	3
Values	1	2	3 ↑	4

Location	**2**	1	0
Values	2	4	6 ↑

0+2=2.

Range: -2 to 3

No. of Terms & Range for r_{yx}:

Total number of terms in cross correlation are (4+3)-1=6

Location	**0**	1	2
Values	2	4	6 ↑

Location	3	2	1	0
Values	1	2	3 ↑	4

2+1=3.

Range: -3 to 2

1) Matrix Method:

$$
\begin{array}{cc|cccc}
 & & & & \downarrow & \\
 & & 1 & 2 & 3 & 4 \\
\hline
\rightarrow & 6 & 6 & 12 & 18 & 24 \\
 & 4 & 4 & 8 & 12 & 16 \\
 & 2 & 2 & 4 & 6 & 8
\end{array}
$$

$$r_{xy}(n) = \{6, 16, \underset{\uparrow}{28}, 40, 22, 8\}$$

2) Graphical Method:

$$r_{xy}(n) = \sum_{k=-\infty}^{k=\infty} x(k) * y(k-n) \quad r_{xy}(-2) = \sum_{k=-\infty}^{k=\infty} x(k) * y(k+2)$$

Range: -4 to 1

$$\begin{matrix} 0 & 0 & 0 & 2 & 4 & \underset{\uparrow}{6} & 0 & 0 & 0 \end{matrix}$$

$$\begin{matrix} 0 & 0 & 0 & 1 & 2 & \underset{\uparrow}{3} & 4 & 0 & 0 \\ 0 & 2 & 4 & 6 & 0 & \underset{\uparrow}{0} & 0 & 0 & 0 \end{matrix}$$

$$r_{xy}(-2) = 6 \times 1 = 6$$

$$\begin{matrix} 0 & 0 & 1 & 2 & \underset{\uparrow}{3} & 4 & 0 & 0 \\ 0 & 2 & 4 & 6 & \underset{\uparrow}{0} & 0 & 0 & 0 \end{matrix}$$

$$r_{xy}(-1) = 4 \times 1 + 6 \times 2 = 16$$

Range: -3 to 1

$$r_{xy}(-1) = \sum_{k=-\infty}^{k=\infty} x(k) * y(k+1)$$

$$\begin{matrix} 0 & 0 & 0 & 2 & 4 & \underset{\uparrow}{6} & 0 & 0 & 0 \end{matrix}$$

$$\begin{matrix} 0 & 1 & 2 & \underset{\uparrow}{3} & 4 & 0 & 0 \\ 0 & 2 & 4 & \underset{\uparrow}{6} & 0 & 0 & 0 \end{matrix}$$

$$r_{xy}(0) = 1 \times 2 + 2 \times 4 + 3 \times 6 = 28$$

Range: -2 to 1

$$r_{xy}(0) = \sum_{k=-\infty}^{k=\infty} x(k) * y(k)$$

0 1 2 3 4 0 0

↑

0 0 2 4 6 0 0

↑

$r_{xy}(1) = 2\times2 + 3\times4 + 4\times6 = 40$

Range: -1 to 1

$$r_{xy}(1) = \sum_{k=-\infty}^{k=\infty} x(k) * y(k-1)$$

0 0 0 2 4 6 0 0 0

↑

1 2 3 4 0 0

↑

0 0 2 4 6 0

↑

$r_{xy}(2) = 2\times3 + 4\times4 = 22$

Range: 0 to 2

$$r_{xy}(2) = \sum_{k=-\infty}^{k=\infty} x(k) * y(k-2)$$

1 2 3 4 0 0

↑

0 0 0 2 4 6

↑

$r_{xy}(3) = 4\times2 = 8$

Range: 0 to 3

0 0 0 2 4 6 0 0 0

↑

$$r_{xy}(3) = \sum_{k=-\infty}^{k=\infty} x(k) * y(k-3)$$

3) Tabular Method:

	-6	-5	-4	-3	-2	-1	0	1	2	3	4			
$x(k)$					1	2	3	4						
							↑							
$y(k)$			0	0	2	4	6	0	0					
							↑							
$y(k+2)$	0	0	2	4	6	0	0							$r_{xy}(-2)=6$
							↑							
$y(k+1)$		0	0	2	4	6	0	0						$r_{xy}(-1)=16$
							↑							
$y(k)$			0	0	2	4	6	0	0					$r_{xy}(0)=28$
							↑							
$y(k-1)$				0	0	2	4	6	0	0				$r_{xy}(1)=40$
							↑							
$y(k-2)$					0	0	2	4	6	0	0			$r_{xy}(2)=22$
							↑							
$y(k-3)$					0	0	0	2	4	6	0	0		$r_{xy}(3)=8$
							↑							

4) Summation Method:

$$r_{xy}(n) = \sum_{k=-\infty}^{k=\infty} x(k) * y(k-n)$$

Range: -4 to 1 **Range: 0 to 2**

Range: -3 to 1 **Range: 0 to 2**

$$r_{xy}(-2) = \sum_{k=-\infty}^{k=\infty} x(k) * y(k+2)$$

Range: -2 to 1

Range: -1 to 1 Signal coincides exactly

$$r_{xy}(-2) = x(-4) * y(-2) + x(-3) * y(-1) + \mathbf{x(-2) * y(0)} + x(-1) * y(1) + x(0) * y(3)$$

$$r_{xy}(-2) = 1 \times 6 = 6$$

x signal

$$\begin{matrix} 1 & 2 & 3 & 4 & 0 & 0 \\ & & \uparrow & & & \end{matrix}$$

y signal

$$\begin{matrix} 0 & 0 & 0 & 2 & 4 & 6 & 0 & 0 & 0 \\ & & & & & \uparrow & & & \end{matrix}$$

Ex.2 Perform Cross Correlation for the given sequence

$$x(n) = \{3, \underset{\uparrow}{2}, 4, 1\} \qquad\qquad y(n) = \{\underset{\uparrow}{2}, 1, 3\}$$

No. of Terms & Range:

Total number of terms in cross correlation are (4+3)-1=7

Location	0	1	2	3
Values	3	2 ↑	4	1

1+2=3.

Range: -3 to 2

Location	**2**	1	**0**
Values	2 ↑	1	3

1) Matrix Method:

$$\begin{array}{r|cccc} & & \downarrow & & \\ & 3 & 2 & 4 & 1 \\ \hline 3 & 9 & 6 & 12 & 3 \\ 1 & 3 & 2 & 4 & 1 \\ \rightarrow 2 & 6 & 4 & 8 & 2 \end{array}$$

$$r_{xy}(n) = \{9, 9, 20, \underset{\uparrow}{11}, 9, 2\}$$

2) Graphical Method:

$$r_{xy}(n) = \sum_{k=-\infty}^{k=\infty} x(k) * y(k-n) \qquad r_{xy}(-3) = \sum_{k=-\infty}^{k=\infty} x(k) * y(k+3)$$

Range: -3 to 2

$$0 \quad 0 \quad \underset{\uparrow}{2} \quad 1 \quad 3 \quad 0 \quad 0$$

$$\boxed{\begin{array}{c} \begin{array}{cccccc} & & 3 & \underset{\uparrow}{2} & 4 & 1 \\ 2 & 1 & 3 & \underset{\uparrow}{0} & & \end{array} \\ r_{xy}(-3) = 3 \times 3 = 9 \end{array}}$$

$$\begin{array}{ccccc} & & 3 & 2 & 4 & 1 \\ & & & \uparrow & & \\ 0 & 2 & 1 & 3 & 0 & \\ & & & \uparrow & & \end{array}$$

$$r_{xy}(-2) = 3\times 1 + 2\times 3 = 9$$

$$r_{xy}(-2) = \sum_{k=-\infty}^{k=\infty} x(k) * y(k+2)$$

Range: -2 to 2

$$\begin{array}{ccccccc} 0 & 0 & 2 & 1 & 3 & 0 & 0 \\ & & \uparrow & & & & \end{array}$$

$$r_{xy}(-1) = \sum_{k=-\infty}^{k=\infty} x(k) * y(k+1)$$

$$\begin{array}{ccccc} & 3 & 2 & 4 & 1 \\ & & \uparrow & & \\ 0 & 2 & 1 & 3 & \\ & & \uparrow & & \end{array}$$

$$r_{xy}(-1) = 3\times 2 + 2\times 1 + 4\times 3 = 20$$

Range: -1 to 2

$$\begin{array}{cccc} 3 & 2 & 4 & 1 \\ & \uparrow & & \\ 0 & 2 & 1 & 3 \\ & \uparrow & & \end{array}$$

$$r_{xy}(0) = 2\times 2 + 4\times 1 + 1\times 3 = 11$$

$$r_{xy}(0) = \sum_{k=-\infty}^{k=\infty} x(k) * y(k)$$

Range: 0 to 2

$$\begin{array}{ccccccc} 0 & 0 & 2 & 1 & 3 & 0 & 0 \\ & & \uparrow & & & & \end{array}$$

$$r_{xy}(1) = \sum_{k=-\infty}^{k=\infty} x(k) * y(k-1)$$

$$\begin{array}{ccccc} 3 & 2 & 4 & 1 & \\ & \uparrow & & & \\ & 0 & 2 & 1 & 3 \\ & \uparrow & & & \end{array}$$

$$r_{xy}(-1) = 4\times 2 + 1\times 1 = 9$$

Range: 0 to 3

$$\begin{array}{ccccc} 3 & 2 & 4 & 1 & \\ & \uparrow & & & \\ & 0 & 0 & 2 & 1 & 3 \\ & \uparrow & & & \end{array}$$

$$r_{xy}(-2) = 1 \times 2 = 2$$

$$r_{xy}(2) = \sum_{k=-\infty}^{k=\infty} x(k) * y(k-2)$$

Range: 0 to 4

$$\begin{array}{ccccccc} 0 & 0 & 2 & 1 & 3 & 0 & 0 \\ & & \uparrow & & & & \end{array}$$

3) Tabular Method:

	-6	-5	-4	-3	-2	-1	0	1	2	3	4			
$x(k)$						3	2	4	1					
							↑							
$y(k)$							2	1	3					
							↑							
$y(k+3)$				2	1	3	0	0	0	0				$r_{xy}(-3)=9$
							↑							
$y(k+2)$		0	0	0	2	1	3	0						$r_{xy}(-2)=9$
							↑							
$y(k+1)$			0	0	0	2	1	3	0					$r_{xy}(-1)=20$
							↑							
$y(k)$				0	0	0	2	1	3	0				$r_{xy}(1)=11$
							↑							
$y(k-1)$					0	0	0	2	1	3	0			$r_{xy}(2)=9$
							↑							
$y(k-2)$					0	0	0	0	2	1	3	0		$r_{xy}(3)=2$
							↑							

4) Summation Method:

Range: -3 to 2 **Range: 0 to 3**

Range: -2 to 2 **Range: 0 to 4**

Range: -1 to 2 Signal coincides exactly

Range: 0 to 2

$$r_{xy}(n) = \sum_{k=-\infty}^{k=\infty} x(k) * y(k-n)$$

$$r_{xy}(-3) = \sum_{k=-\infty}^{k=\infty} x(k) * y(k+3)$$

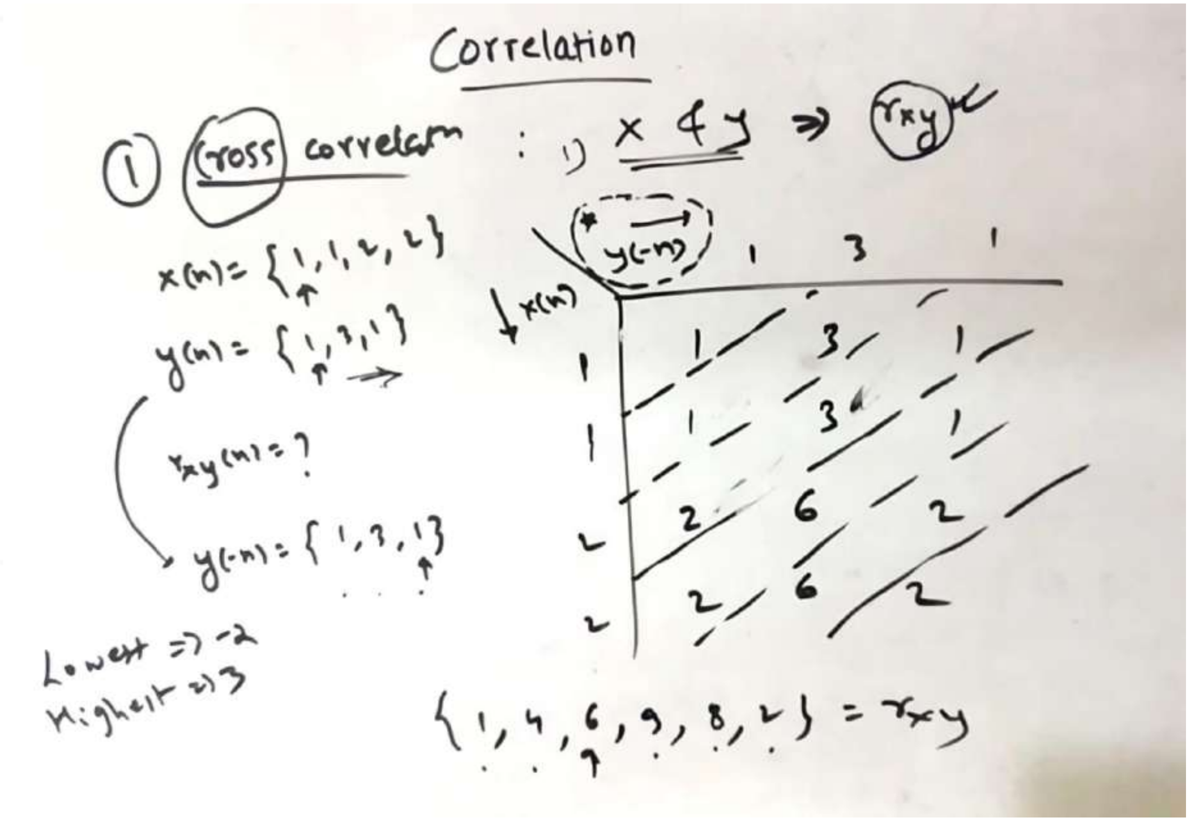
$$r_{xy}(-3) = x(-3) * y(0) + x(-2) * y(1) + \boldsymbol{x(-1) * y(2)} + x(0) * y(3) + x(1) * y(4) + x(2) * y(5)$$
$$r_{xy}(-3) = 3 \times 3 = 9$$

x signal

$$\begin{matrix} 3 & 2 & 4 & 1 \\ & \uparrow & & \end{matrix}$$

y signal

$$\begin{matrix} 0 & 0 & 2 & 1 & 3 & 0 & 0 \\ & & \uparrow & & & & \end{matrix}$$



Determine cross-correlation between two sequences

$$x_1[n] = \{\underset{\uparrow}{1}, 2, 3, 4\} \quad x_2[n] = \{\underset{\uparrow}{0}, 1, 2, 3\}$$

Solution :-

Cross Correlation is defined by

$$R_{x_1x_2}[K] = \sum_{n=-\infty}^{+\infty} x_1[n]\, x_2[n-K]$$

Direct Computation :

we can start K arbitrarily

If $K < -3$, e.g. $K = -4$.

$$R_{x_1x_2}[K] = \sum_{n=-\infty} x_1[n]\, x_2[n-K]$$

Direct Computation :

we can start K arbitrarily

If $K < -3$, e.g. $K = -4$.

$$R_{x_1x_2}[-4] = \sum_{n=-\infty}^{\infty} x_1[n]\, x_2[n+4]$$

As $x_1[n] = 0$ for $n < 0$ and $n > 3$,

Limits for summation $n = 0$ to 3.

$$\therefore \; R_{x_1x_2}[-4] = \sum_{n=0}^{3} x_1[n]\, x_2[n+4]$$

Direct Computation :

we can start K arbitrarily

If $K < -3$, e.g. $K = -4$.

$$R_{x_1x_2}[-4] = \sum_{n=-\infty}^{\infty} x_1[n]\, x_2[n+4]$$

As $x_1[n] = 0$ for $n < 0$ and $n > 3$,

Limits for summation $n = 0$ to 3.

$$\therefore\ R_{x_1x_2}[-4] = \sum_{n=0}^{3} x_1[n]\, x_2[n+4]$$

$$= x_1(0)x_2(4) + x_1(1)x_2(5)$$

As $x_1[n] = 0$ for $n < 0$ and $n > 3$,

Limits for summation $n = 0$ to 3.

$$\therefore\ R_{x_1x_2}[-4] = \sum_{n=0}^{3} x_1[n]\, x_2[n+4]$$

$$= x_1(0)x_2(4) + x_1(1)x_2(5)$$

$$+ x_1(2)x_2(6) + x_1(3)x_2(7)$$

$$= (1)(0) + (2)(0) + (3)(0) + (4)(0)$$

$$= 0 + 0 + 0 + 0 = 0$$

$K = -3$.

$$R_{x_1x_2}[-3] = \sum_{n=0}^{3} x_1[n] \, x_2[n+3]$$

$$= x_1(0)\,x_2(3) + x_1(1)\,x_2(4) + x_1(2)\,x_1(5) + x_1(3) \cdot x_2(6)$$

$$= (1)(3) + 0 + 0 + 0 = 3$$

$K = -2$

$$R_{x_1x_2}[-2] = \sum_{n=0}^{3} x_1(n)\,x_2(n+2)$$

$$= x_1(0)\,x_2(2) + x_1(1)\,x_2(3) + x_1(2)\,x_2(4) + x_1(3)\,x_2(5)$$

$$= (1)(2) + (2)(3) + (3)(0) + (4)(0)$$

$$= 8$$

$K = -1$

$$R_{x_1x_2}[-1] = \sum_{n=0}^{3} x_1[n]\,x_2[n+1]$$

$$= x_1(0)\,x_2(1) + x_1(1) \cdot x_2(2) + x_1(2)\,x_2(3) + x_1(3)\,x_2(4)$$

$$= (1)(1) + (2)(2) + (3)(3) + (4)(0)$$

$$= 1 + 6 + 9 = 14$$

$K = 0$

$$R_{x_1x_2}[0] = \sum_{n=0}^{3} x_1[n] \cdot x_2[n]$$

$$= x_1(0)\, x_2(0) + x_1(1) \cdot x_2(1) + x_1(2)\, x_2(2) + x_1(3) \cdot x_2(3)$$

$$= 0 + (1)(2) + (3)(2) + (4)(3)$$

$$= 0 + 2 + 6 + 12 = 20$$

$K = 1$

$$R_{x_1x_2}[1] = \sum_{n=0}^{3} x_1[n]\, x_2[n-1]$$

$$= x_1(0) \cdot x_2(-1) + x_1(1)\, x_2(0) + x_1(2)\, x_2(1) + x_1(3)\, x_2(2)$$

$$= (1)(0) + (2)(0) + (3)(1) + (4)(2)$$

$$= 0 + 0 + 3 + 8 = 11$$

$K = 2$

3

$$= 0+0+3+8 = 11$$

$K=2$

$$R_{x_1x_2}[2] = \sum_{n=0}^{3} x_1[n]\, x_2[n-2]$$

$$= x_1(0)x_2(-2) + x_1(1)x_2(-1) + x_1(2)x_2(0) + x_1(3)x_2(1)$$

$$= 0+0+(3)(0)+(4)(1) = 4$$

$K=3$

$$R_{x_1x_2}[3] = \sum x_1[n]\, x_2[n-3]$$

$K=3$

$$R_{x_1x_2}[3] = \sum_{n=0}^{3} x_1[n]\, x_2[n-3]$$

$$= x_1(0)x_2(-3) + x_1(1)x_2(-2) + x_1(2)x_2(-1) + x_1(3)x_2(0)$$

$$= 0+0+0+0 = 0$$

$$\therefore R_{x_1x_2}[K] = \{3, 8, 14, \underset{\uparrow}{20}, 11, 4\}$$

Linear convolution and circular convolution

Graphical Method

Q) Determine the response of the LTI System whose i

and impulse response h(n) are given by:-

$$x(n) = \{\underset{\uparrow}{\overset{n=0}{1}}, \overset{1}{2}, \overset{2}{0.5}, \overset{3}{1}\}$$

$$h(n) = \{\overset{-1}{1}, \underset{\uparrow}{\overset{n=0}{2}}, \overset{1}{1}, \overset{2}{-1}\}$$

Response, $y(n) = x(n) * h(n)$

$$= \sum_{m=-\infty}^{+\infty} x(m)\, h(n-m)$$

Graphical representation method:-

Input $x(n) = 4 \; (N_1 = 4)$

Impulse Response $h(n) = 4 \; (N_2 = 4)$

Output Sequence $y(n) = N_1 + N_2 - 1$

$$= 4 + 4 - 1 = 8 - 1 = 7$$

Input Sequence $x(n)$ Start at $n = 0 \; (n_1 = 0)$

Impulse response sequence Start at $n = -1 \; (n_2 = -1)$

Output Sequence $y(n)$ Starts at $n = n_1 + n_2 = 0 + (-1)$

$$= -1$$

End $\rightarrow (n_1 + n_2) + (N_1 + N_2 - 2) = 0 + (-1) + (4 + 4 - 2)$

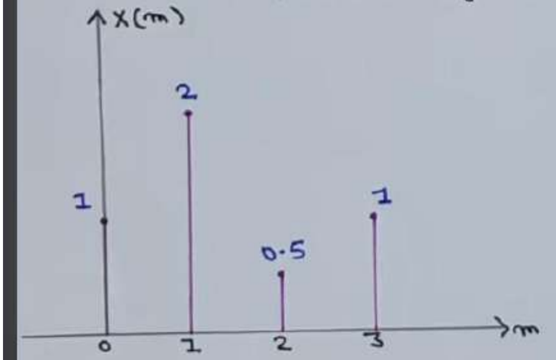
$$= -1 + 6 = 5$$

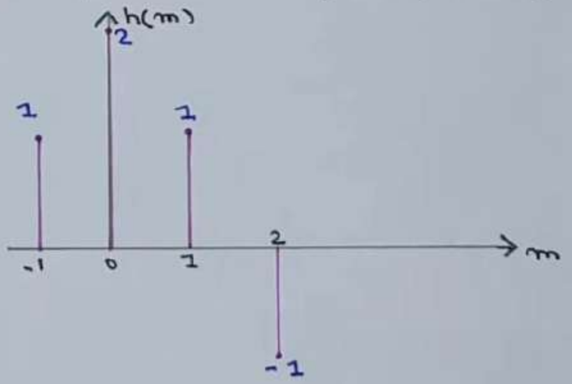
Step-1> Change the index from n to m in the Sequences x(n) & h(n).

$$x(n) \rightarrow x(m)$$

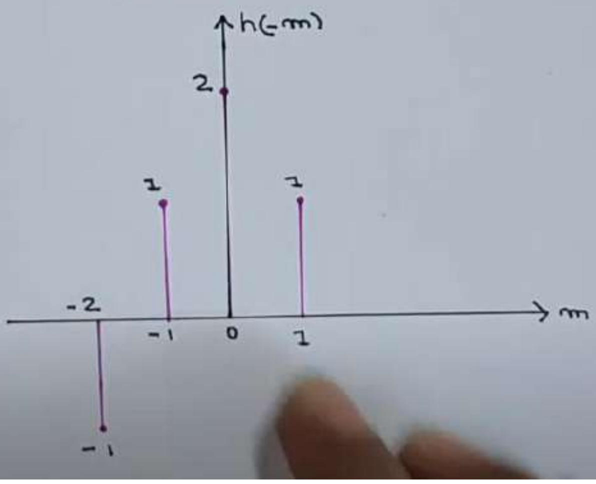
$$h(n) \rightarrow h(m)$$

Step-2> Now Sketch the graphical representation of x(m) & h(m).





ep-3> Fold h(m) about m=0 to obtain h(-m) and Sketch
graph of the sequence h(-m).



$$y(n) = \sum_{m=-\infty}^{+\infty} x(m)h(n-m)$$

Step-4) Shift $h(-m)$ by n to the right if n is positive and to the left if n is negative in order to obtain $h(\underline{n}-m)$.

① When $n = -1$

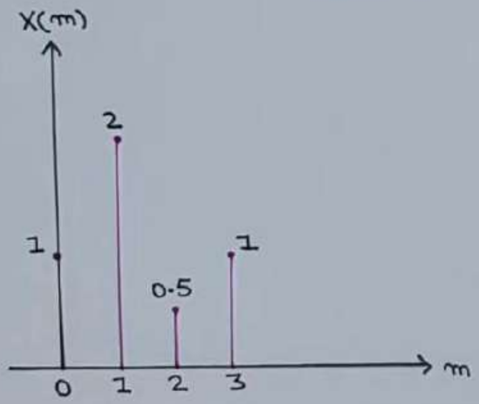
$$Y(-1) = \sum_{m=-\infty}^{+\infty} x(m)\, h(-1-m)$$

$$Y(-1) = \sum_{m=-\infty}^{+\infty} x(m)\, h(-1-m)$$

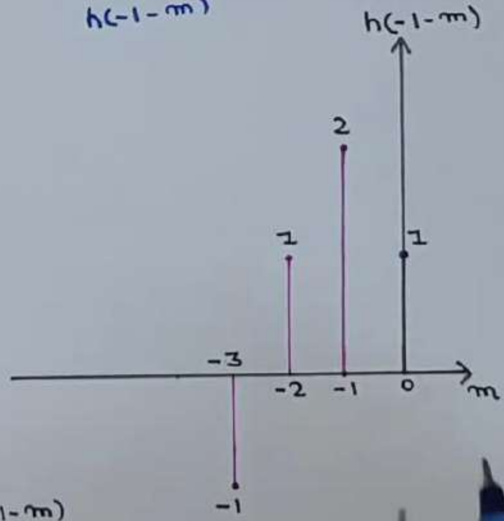
$h(-1-m)$

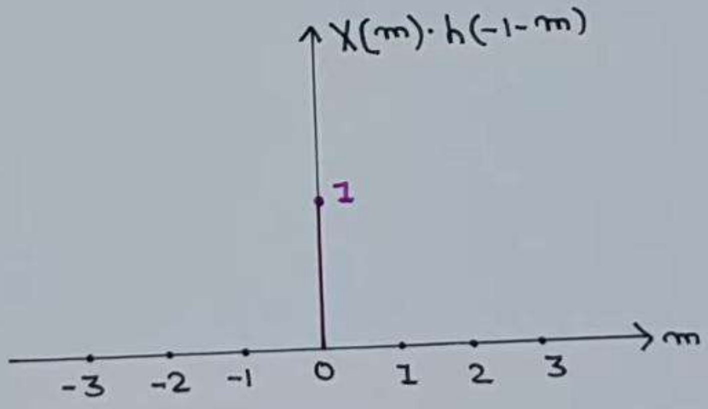
$h(-m) \rightarrow$ 1 unit left shift

$h(-1-m)$



$\times$



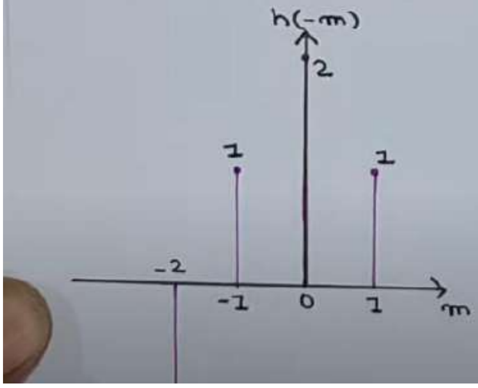


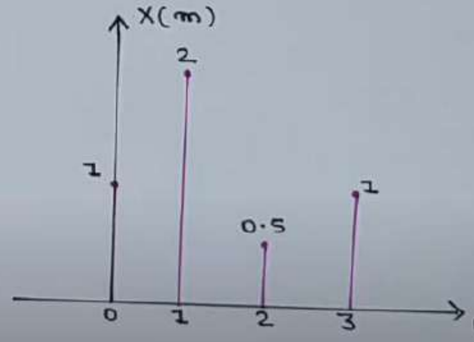
The sum of product sequence $x(m)h(-1-m)$

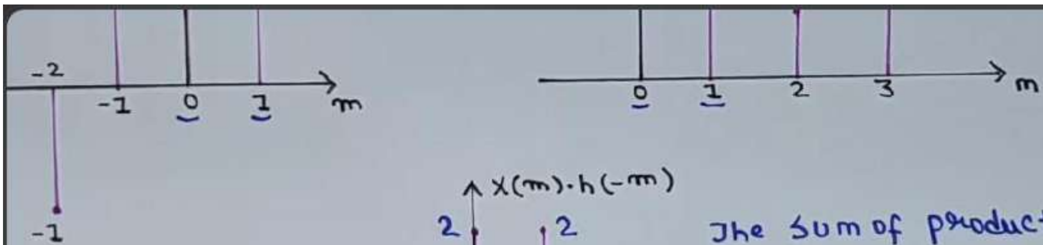
$Y(-1) = 1$

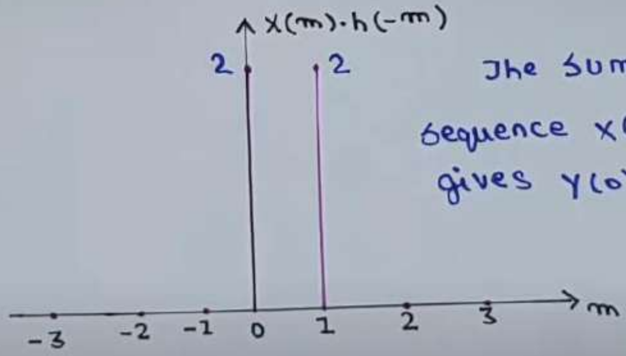
② When $n=0$;

$$y(0) = \sum_{m=-\infty}^{+\infty} x(m)\,h(0-m)$$







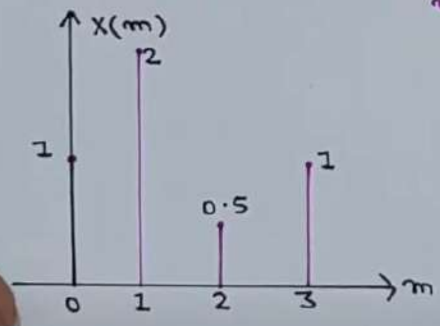


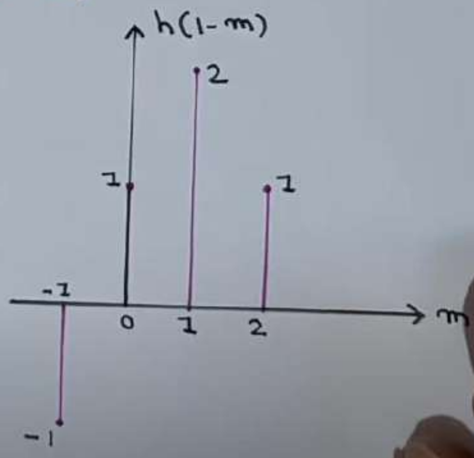
The sum of product sequence $x(m)\cdot h(-m)$ gives $y(0) = 2+2 = 4$

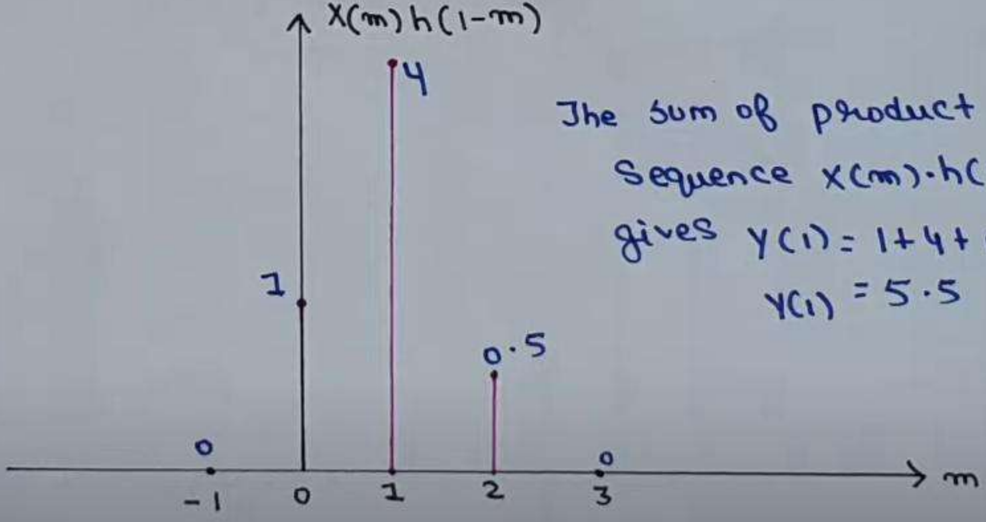
$y(0) = 4$ ✓

③ When $n=1$

$$y(1) = \sum_{m=-\infty}^{+\infty} x(m)\,h(1-m)$$







The sum of product Sequence $x(m)\cdot h(1-m)$ gives $y(1) = 1+4+0.5$

$y(1) = 5.5$

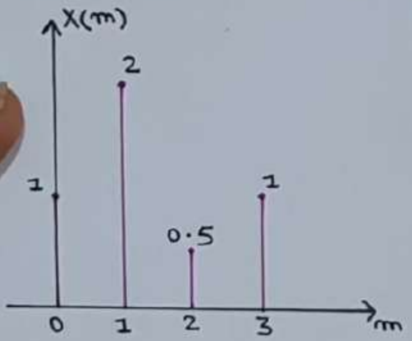
④ When $n=2$

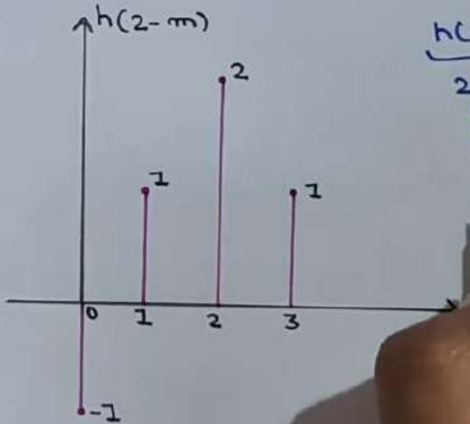
$$y(2) = \sum_{m=-\infty}^{+\infty} x(m)\,h(2-m)$$

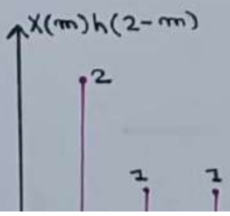
$h(2-m)$

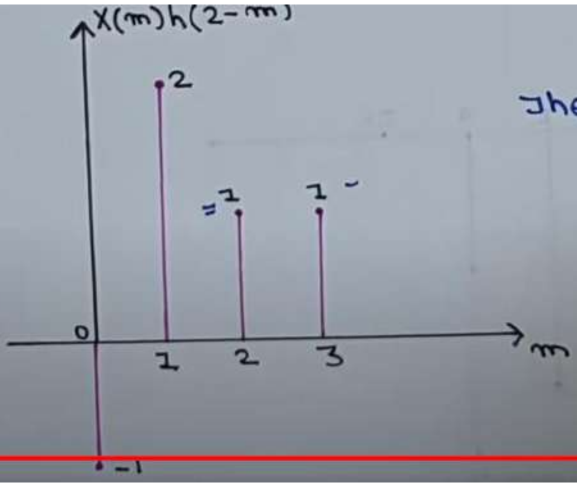
$\underline{\underline{n=2}}$

$\underbrace{h(-m)}$ 2 unit Right shift





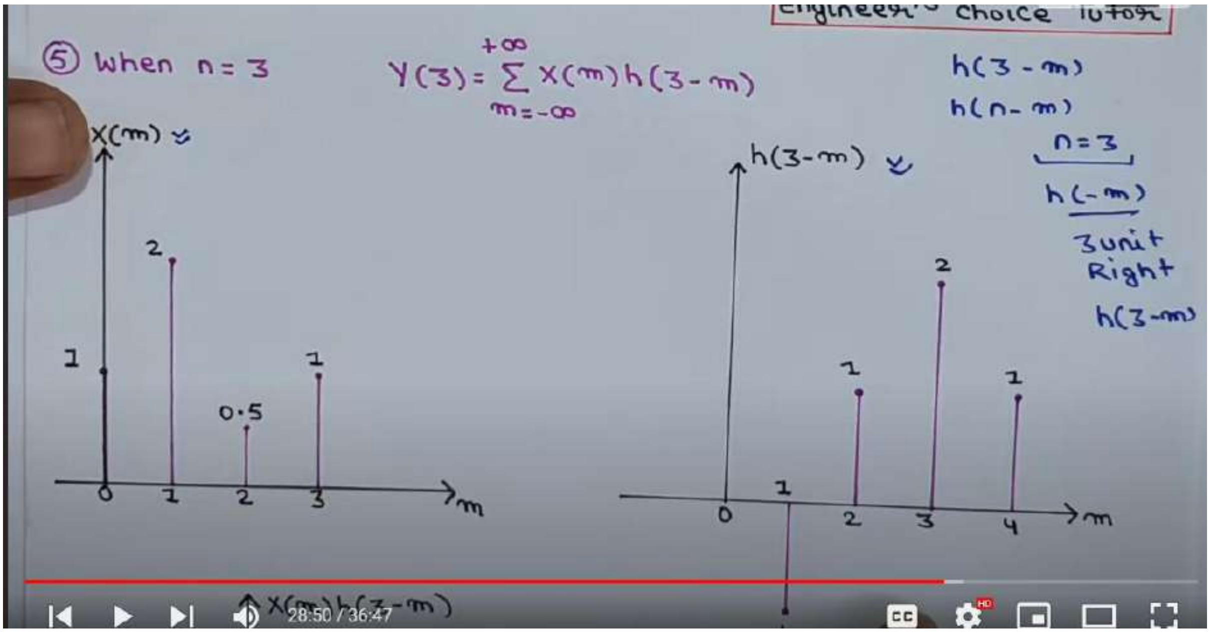


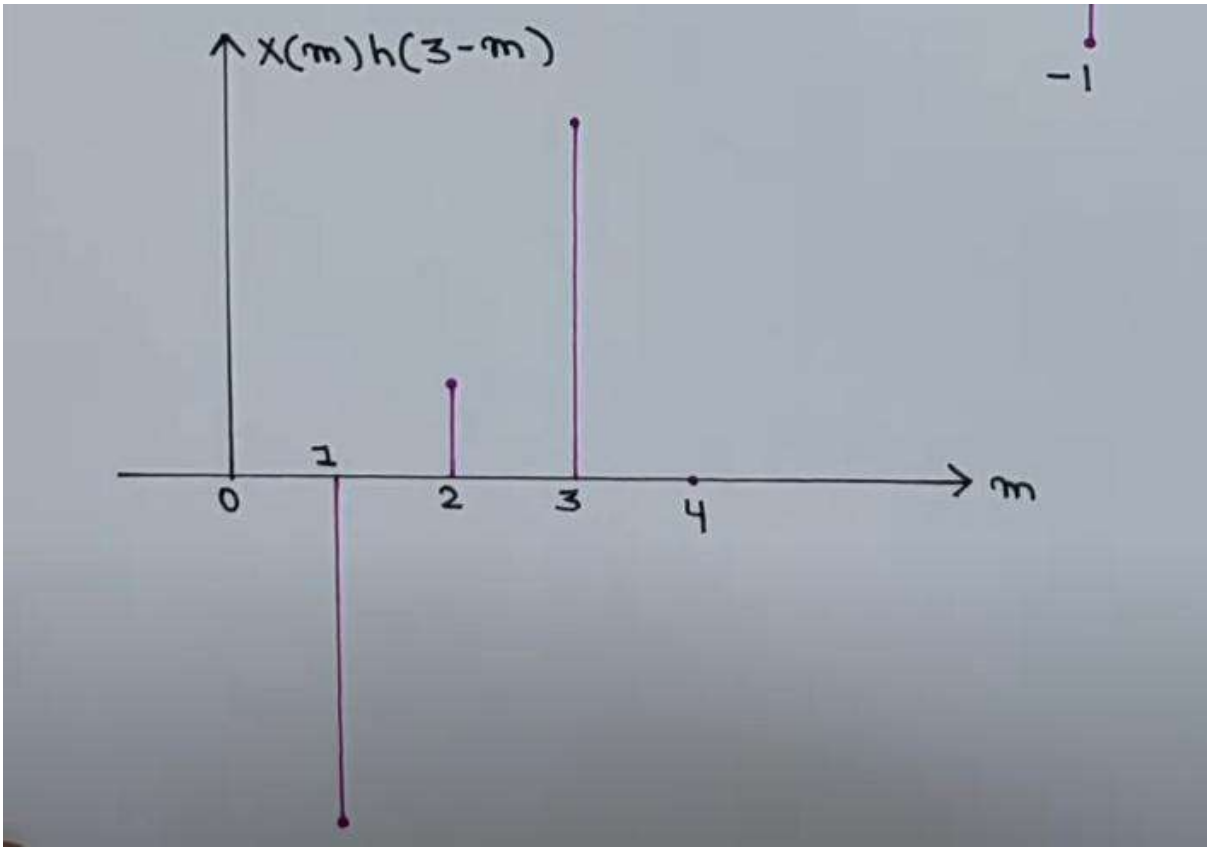


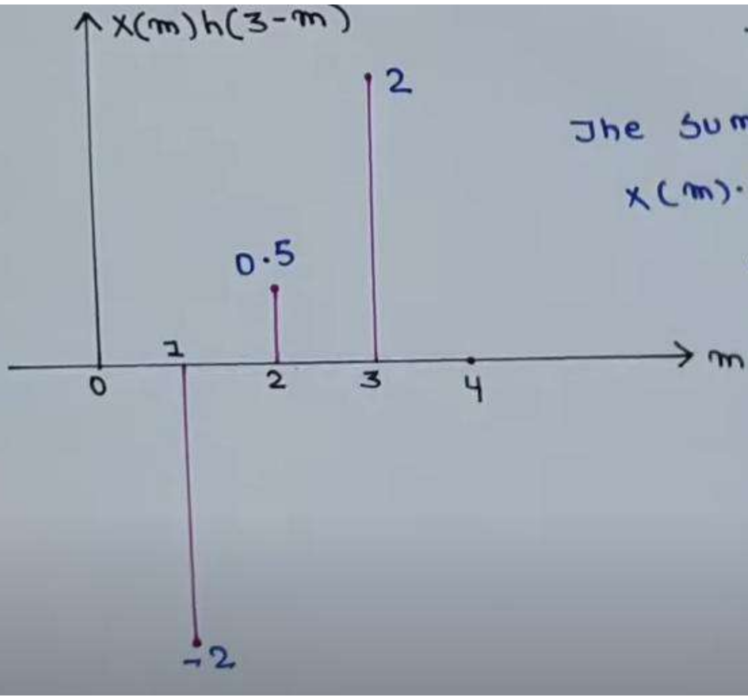
The sum of product seq $x(m)\cdot h(2-m) \rightarrow y(2)=$

$y(2) = -\cancel{1}+2+\cancel{1}+1$

$y(2) = 3$


Engineer's Choice Tutor
⑤ When n = 3
Y(3) = $\sum_{m=-\infty}^{+\infty}$ x(m)h(3-m)
h(3-m)
h(n-m)
n=3
h(-m)
3 unit Right
h(3-m)
x(m)
h(3-m)
2
1
0.5
1
0 1 2 3 m
2
1
1
1
0 2 3 4 m
x(m)h(3-m)
28:50 / 36:47


x(m)h(3-m)
-1
1
0 2 3 4
m



-1

The sum of product seq

$x(m)\cdot h(3-m) \rightarrow y(3)$

$y(3) = -2 + 0.5 + 2$

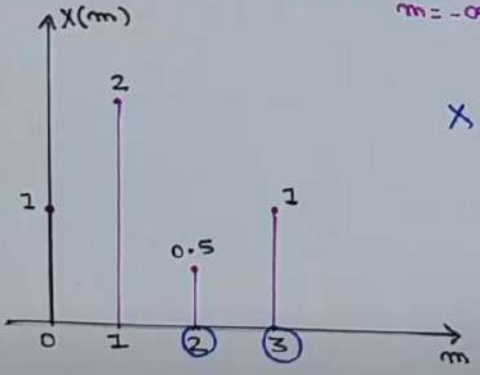
$y(3) = 0.5$

⑥ When $n = 4$ $\quad y(4) = \sum_{m=-\infty}^{+\infty} x(m)h(4-m)$

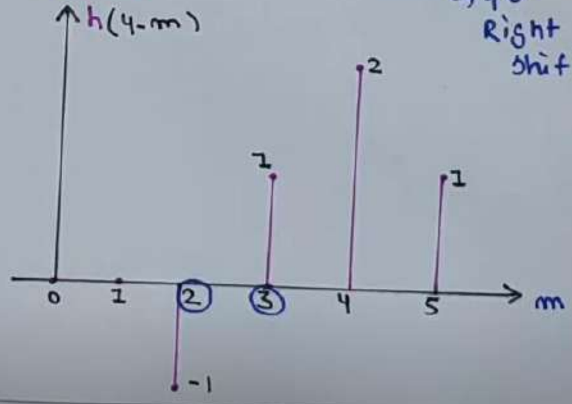
$h(4-m)$

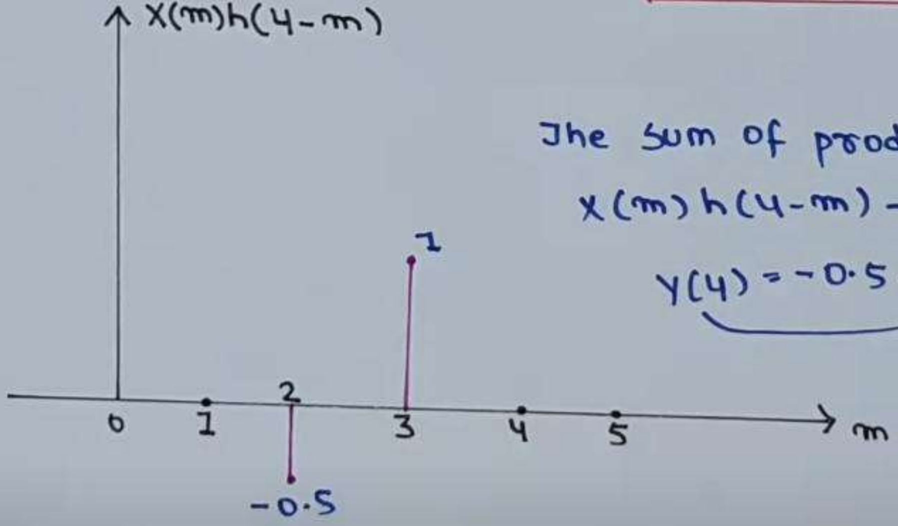
$h(-m)$

↳ 4U Right shift



X





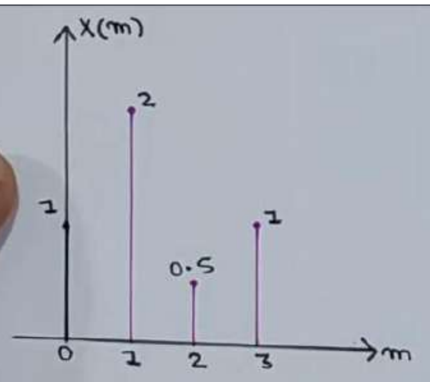
The sum of prod seq

$x(m)h(4-m) \rightarrow y(4)$

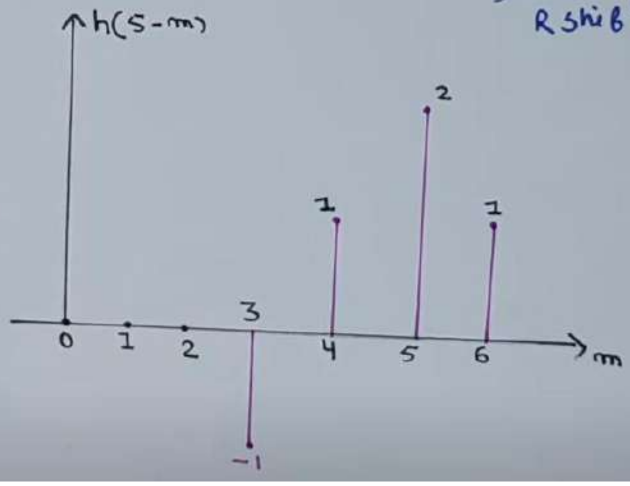
$y(4) = -0.5 + 1 = 0.5$

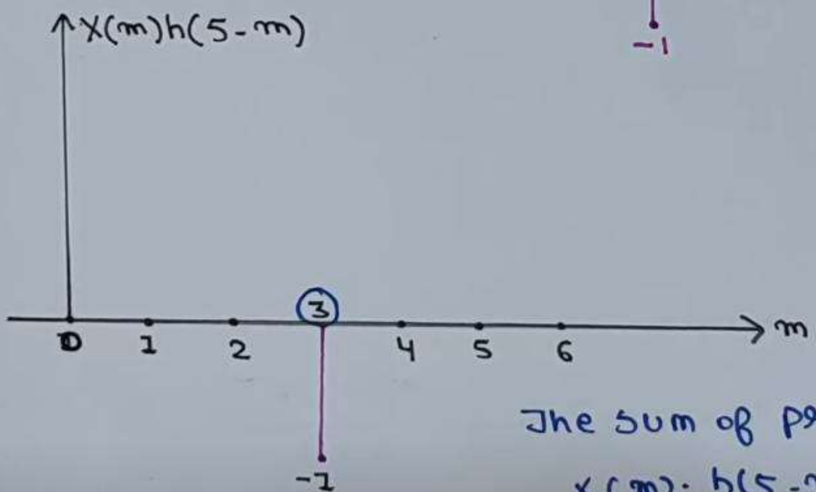
(7) When $n = 5$

$$y(5) = \sum_{m=-\infty}^{+\infty} x(m)\,h(5-m)$$



5 units R shift

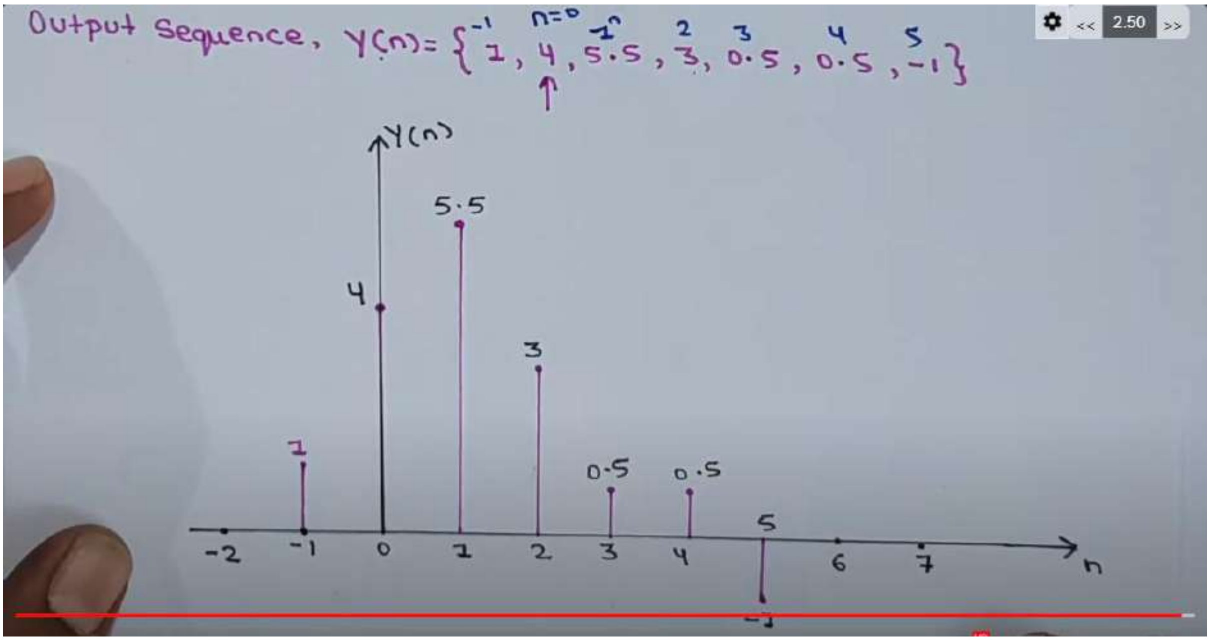




The sum of product sequence

$x(m) \cdot h(5-m) \rightarrow y(5)$

$y(5) = -1$



Tabular Method

Q) Determine the response of the LTI System whose input x(n) & impulse response h(n) are :-

$$x(n) = \{ \underset{\uparrow}{1}, 2, 0.5, 1 \}$$

$$h(m) = \{ 1, \underset{\uparrow}{2}, 1, -1 \}$$

$$Y(n) = x(n) * h(n)$$

$$= \sum_{m=-\infty}^{+\infty} x(m) h(n-m)$$

Tabular method :-

Step-1> Change the index from n to m in $x(n)$ & $h(n)$.

$$x(n) \to x(m)$$

$$h(n) \to h(m)$$

Input $x(n) = 4$ $(N_1 = 4)$

Impulse Response $h(n) = 4$ $(N_2 = 4)$

Output Sequence $y(n) = N_1 + N_2 - 1 = 4 + 4 - 1 = 7$ Sample

Input Sequence $x(n)$ Start at $n = 0$ $(n_1 = 0)$

Impulse response sequence Start at $n = -1$ $(n_2 = -1)$

Output Sequence $y(n)$ Starts at $n = n_1 + n_2 = 0 + (-1)$

$$= -1$$

$$\text{End} = (n_1 + n_2) + (N_1 + N_2 - 2) = -1 + (4 + 4 - 2)$$

$$n = -1 + 6 = 5$$

Step-2> Represent the sequences $x(m)$ and $h(m)$ as two rows of tabular array.

Step-3> Fold one of the Sequences. Let us fold $h(m)$ to get $h(-m)$.

m	-3	-2	-1	0	1	2	3	4	5	6
$x(m)$				1	2	0.5	1			
$h(m)$			1	2	1	-1				
$h(-m)$		-1	1	2	1					
$n=-1$ $h(-1-m)$	-1	1	2	1						
$n=0$ $h(0-m)$		-1	1	2	1					

$n=0$ $h(0-m)$		-1	1	2	1					
$n=1$ $h(1-m)$			-1	1	2	1				
$n=2$ $h(2-m)$				-1	1	2	1			
$n=3$ $h(3-m)$					-1	1	2	1		
$n=4$ $h(4-m)$						-1	1	2	1	
$n=5$ $h(5-m)$							-1	1	2	1

For $n=0$:-

$$y(0) = \sum_{m=-\infty}^{+\infty} x(m)\,h(0-m) = \sum_{m=-\infty}^{+\infty} x(m)\cdot h(-m) = 1\cdot 2 + 2\cdot 1$$

$$y(0) = 2+2 = 4$$

For $n=1$:-

$$y(1) = \sum_{m=-\infty}^{+\infty} x(m)\,h(1-m) = 1\cdot 1 + 2\cdot 2 + 0.5\times 1$$

$$y(1) = 5.5$$

For $n=2$:-

$$y(2) = \sum_{m=-\infty}^{+\infty} x(m)\,h(2-m) = 1\times -1 + 2\times 1 + 0.5\times 2 + 1\times 1$$

$$y(2) = \cancel{-1} + 2 + \cancel{1} + 1 = 3$$

For $n=3$:-

$$y(3) = \sum_{m=-\infty}^{+\infty} x(m)\,h(3-m) = 2\times -1 + 0.5\times 1 + 1\times 2$$

$$y(3) = -2 + 0.5 + 2 = 0.5$$

For $n=4$:-

$$y(4) = \sum_{m=-\infty}^{+\infty} x(m)\,h(4-m) = 0.5\times -1 + 1\times 1$$

$$y(4) = -0.5 + 1 = 0.5$$

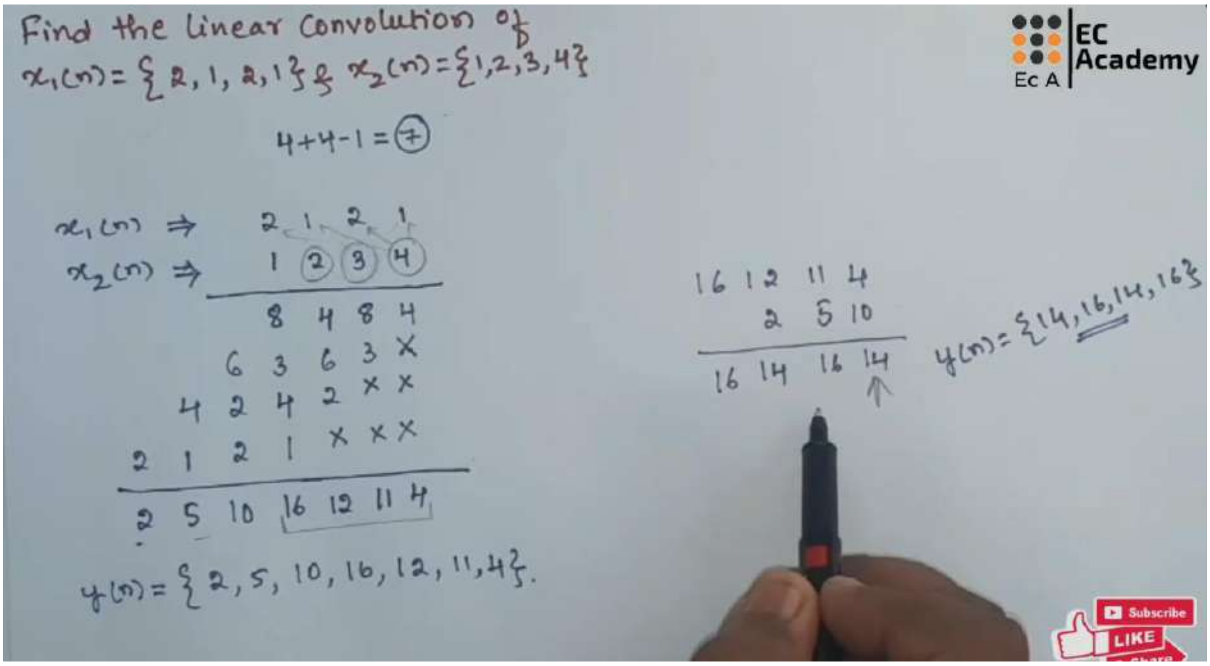
For $n=5$:-

$$y(5) = \sum_{m=-\infty}^{+\infty} x(m)\,h(5-m) = 1\cdot -1 = -1$$

$$y(5) = -1$$

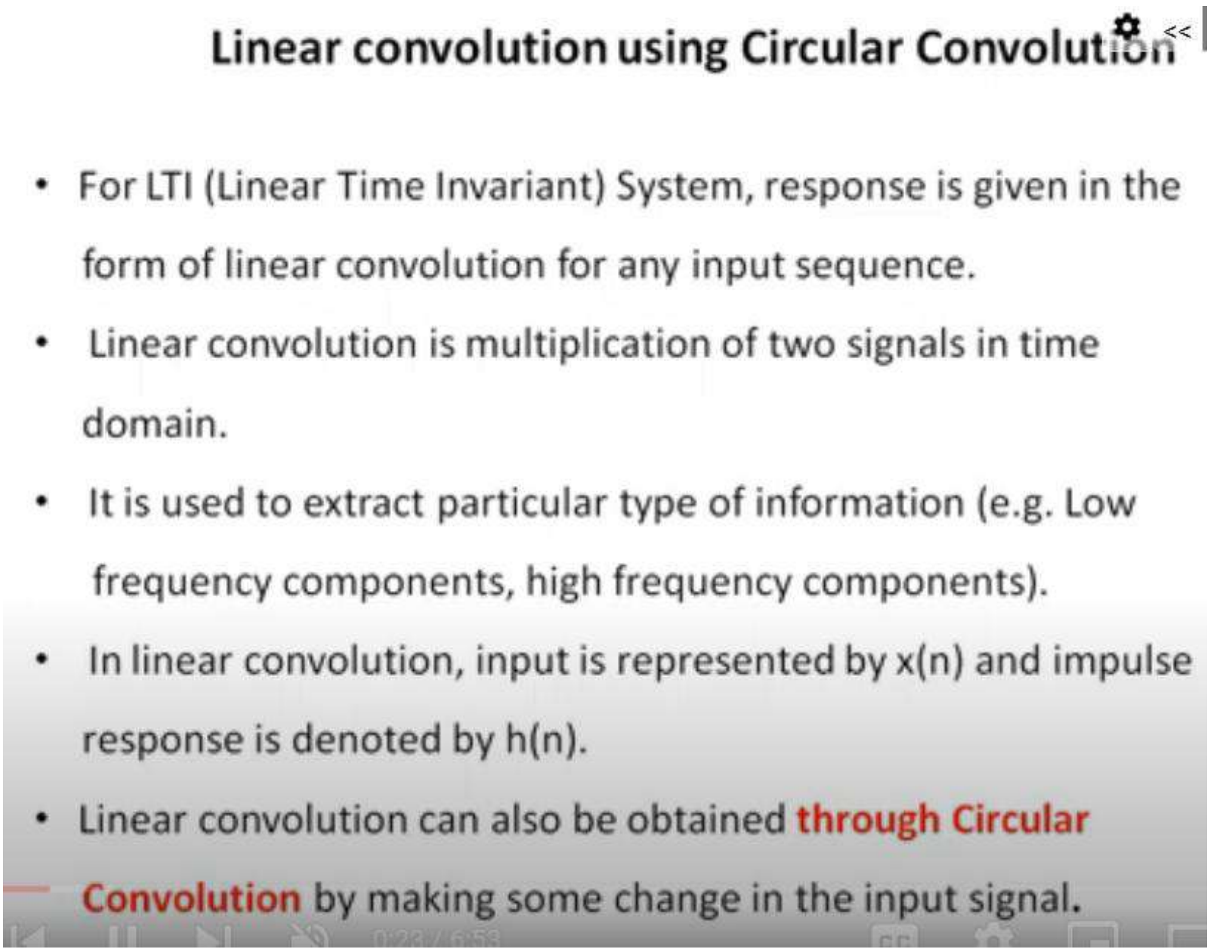
$$\text{O/P sequence } \quad y(n) = \{\begin{array}{ccccccc} n=-1 & 0 & 1 & 2 & 3 & 4 & 5 \\ 1, & \underset{\uparrow}{4}, & 5.5, & 3, & 0.5, & 0.5, & -1 \end{array}\}$$

Linear Convolution



Linear convolution can be converted into circular convolution by taking the first n values , here 4 as it’s a 4 point sequence and adding it with the rest of the values

Linear Convolution using Circular Convolution



- Output of Circular convolution is not equal to Linear Convolution.
- Let x1(n) be a signal of M length and x2(n) be a signal of N length.
- No. of terms are **M+N-1** in Linear Convolution.
- In Circular Convolution, for x1(n) and x2(n) having length of M. No. of terms are **M**.

$$M + N - 1 \neq M$$

- Hence Output of Linear Convolution and Circular Convolution is not same.
- Let x1(n) and x2(n) be a signal of 4 length.
- No. of terms are 4+4-1=7, in Linear Convolution.
- In Circular Convolution, for x1(n) and x2(n) having length of 4. No. of terms are 4.

Linear convolution using Circular Convolution

- Let x1(n) be a signal having length M1 and x2(n) be a signal having M2 length.
- No. of terms are M1+M2-1 in Linear Convolution.
- To obtain same result that is linear convolution using circular convolution, make the length of the signal as M1+M2-1 in circular convolution.
- This is done by padding extra zeros.
- By this method, no. of terms in circular and linear convolution are same. Thus Linear convolution is obtained using Circular Convolution.

Ex.1 Overlap Add Method, To perform convolution of long sequence usi...

i) Find the circular convolution and linear convolution of the following sequences. Show that their results are not same.

ii) Find the circular convolution for the same sequences so that result of linear and circular convolution is same.

$$x1(n) = \{\underset{\uparrow}{5}, 6, 2, 1\} \qquad x2(n) = \{\underset{\uparrow}{3}, 2, 1, 4\}$$

	5	6	2	1
3	15	18	6	3
2	10	12	4	2
1	5	6	2	1
4	20	24	8	4

Linear Convolution

$$y(n) = \{\underset{\uparrow}{15}, 28, 23, 33, 28, 9, 4\}$$

Circular Convolution

$$\begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ y(3) \end{bmatrix} = \begin{bmatrix} 5 & 1 & 2 & 6 \\ 6 & 5 & 1 & 2 \\ 2 & 6 & 5 & 1 \\ 1 & 2 & 6 & 5 \end{bmatrix} \times \begin{bmatrix} 3 \\ 2 \\ 1 \\ 4 \end{bmatrix} = \begin{bmatrix} 43 \\ 37 \\ 27 \\ 33 \end{bmatrix} \qquad y(n) = \{\underset{\uparrow}{43}, 37, 27, 33\}$$

Linear Convolution $\neq$ Circular Convolution

$$y(n) = \{\underset{\uparrow}{15}, 28, 23, 33, 28, 9, 4\} \neq y(n) = \{\underset{\uparrow}{43}, 37, 27, 33\}$$

No of Terms **4+4-1=7** **4**

ii) Find the circular convolution for the same sequences so that result of linear and circular convolution is same. Or Find linear convolution using Circular Convolution

$$x1(n) = \{\underset{\uparrow}{5}, 6, 2, 1\} \qquad x2(n) = \{\underset{\uparrow}{3}, 2, 1, 4\}$$

$$x1(n) = \{5, 6, 2, 1, 0, 0, 0\} \qquad x2(n) = \{\underset{\uparrow}{3}, 2, 4, 1, 0, 0, 0\}$$

$$\begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ y(3) \\ y(4) \\ y(5) \\ y(6) \end{bmatrix} = \begin{bmatrix} 5 & 0 & 0 & 0 & 1 & 2 & 6 \\ 6 & 5 & 0 & 0 & 0 & 1 & 2 \\ 2 & 6 & 5 & 0 & 0 & 0 & 1 \\ 1 & 2 & 6 & 5 & 0 & 0 & 0 \\ 0 & 1 & 2 & 6 & 5 & 0 & 0 \\ 0 & 0 & 1 & 2 & 6 & 5 & 0 \\ 0 & 0 & 0 & 1 & 2 & 6 & 5 \end{bmatrix} \times \begin{bmatrix} 3 \\ 2 \\ 1 \\ 4 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 15 \\ 28 \\ 23 \\ 33 \\ 28 \\ 9 \\ 4 \end{bmatrix}$$

$$y(n) = \{\underset{\uparrow}{15}, 28, 23, 33, 28, 9, 4\}$$

Ex.2

Find the circular convolution of the following casual sequence in time domain so that the result of linear and circular convolution will be same.

$$x1(n) = \{\underset{\uparrow}{1}, 2, 5\} \qquad x2(n) = \{\underset{\uparrow}{4}, 7\}$$

Linear Convolution **No of Terms** **3+2-1=4**

$$x1(n) = \{1, 2, 5, 0\} \qquad x2(n) = \{4, 7, 0, 0\}$$

$$\begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ y(3) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 5 & 2 \\ 2 & 1 & 0 & 5 \\ 5 & 2 & 1 & 0 \\ 0 & 5 & 2 & 1 \end{bmatrix} \times \begin{bmatrix} 4 \\ 7 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 4 \\ 15 \\ 34 \\ 35 \end{bmatrix} \qquad y(n) = \{\underset{\uparrow}{4}, 15, 34, 35\}$$

	1	2	5
4	4	8	20
7	7	14	35

$$y(n) = \{\underset{\uparrow}{4}, 15, 34, 35\}$$

Linear Convolution = Circular Convolution

Ex.3

Find the circular convolution of the following casual sequence in time domain so that the result of linear and circular convolution will be same.

$$x1(n) = \{\underset{\uparrow}{1}, -1, 2, -4\} \qquad x2(n) = \{\underset{\uparrow}{1}, 2\}$$

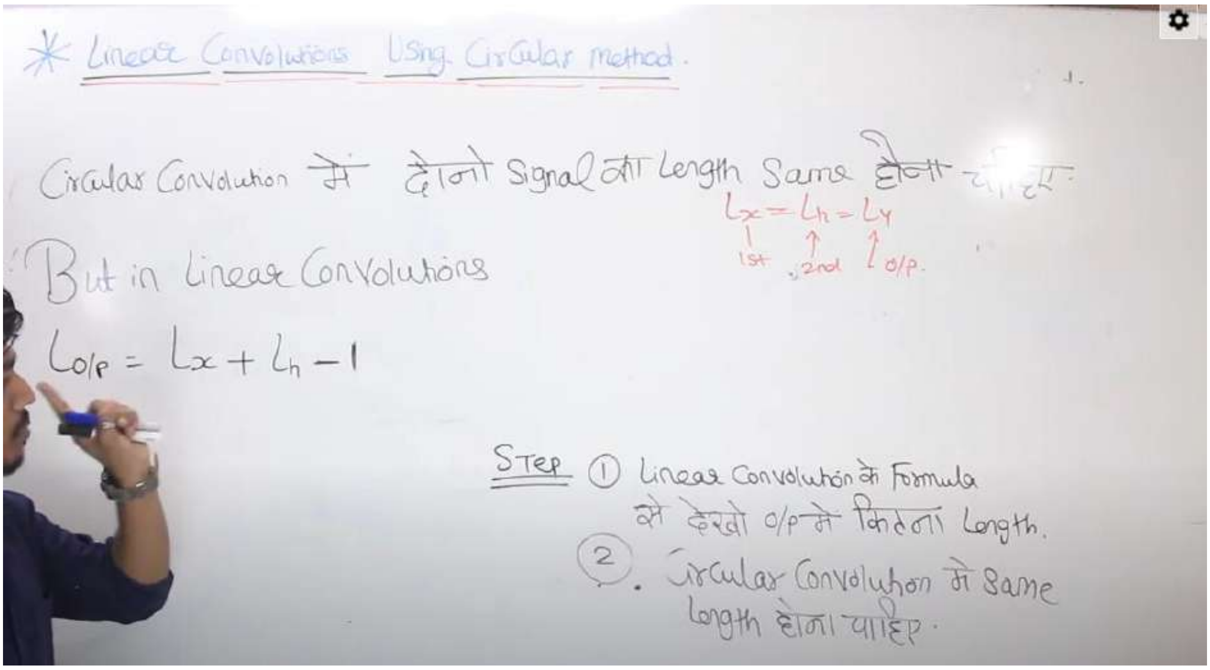
Linear Convolution **No of Terms** **4+2-1=5**

$$x1(n) = \{1, -1, 2, -4, 0\} \quad x2(n) = \{1, 2, 0, 0, 0\}$$

$$
\begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ y(3) \\ y(4) \end{bmatrix} = \begin{bmatrix} 1 & 0 & -4 & 2 & -1 \\ -1 & 1 & 0 & -4 & 2 \\ 2 & -1 & 1 & 0 & -4 \\ -4 & 2 & -1 & 1 & 0 \\ 0 & -4 & 2 & -1 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 2 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ -8 \end{bmatrix}
$$

$$
y(n) = \{\underset{\uparrow}{1}, 1, 0, 0, -8\}
$$

	1	−1	2	−4
1	1	−1	2	−4
2	2	−2	4	−8



Find Linear Convolutions Using Circular Convolution

$x(n) = \{2, 3, 4, 5\}$ & $y(n) = \{5, 2, 3, 4\}$

Sol Step 1 $L_x = 4$, $L_y = 4$

$L_0 = L_x + L_y - 1$

$= 4 + 4 - 1$

$\boxed{L_0 = 7}$

Step 2 Circular convolution

$L_0 = L_x = L_y$

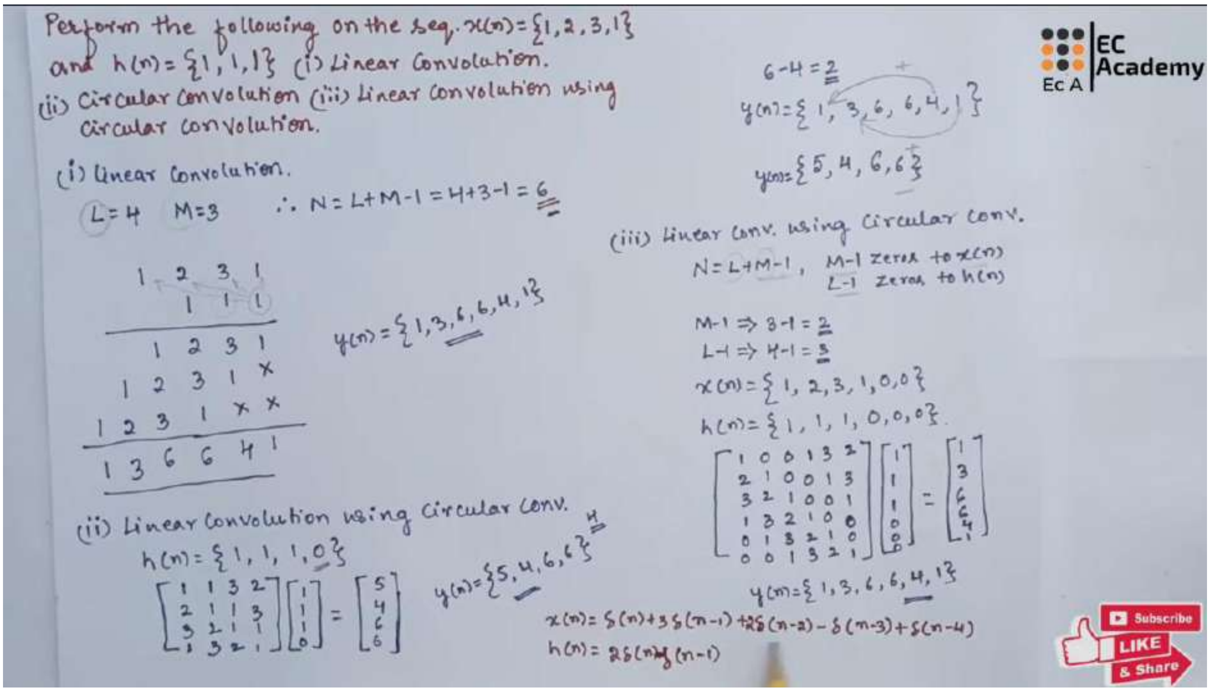
$\boxed{7 = L_x = L_y}$

$x(n) = \{2, 3, 4, 5, 0, 0, 0\}$

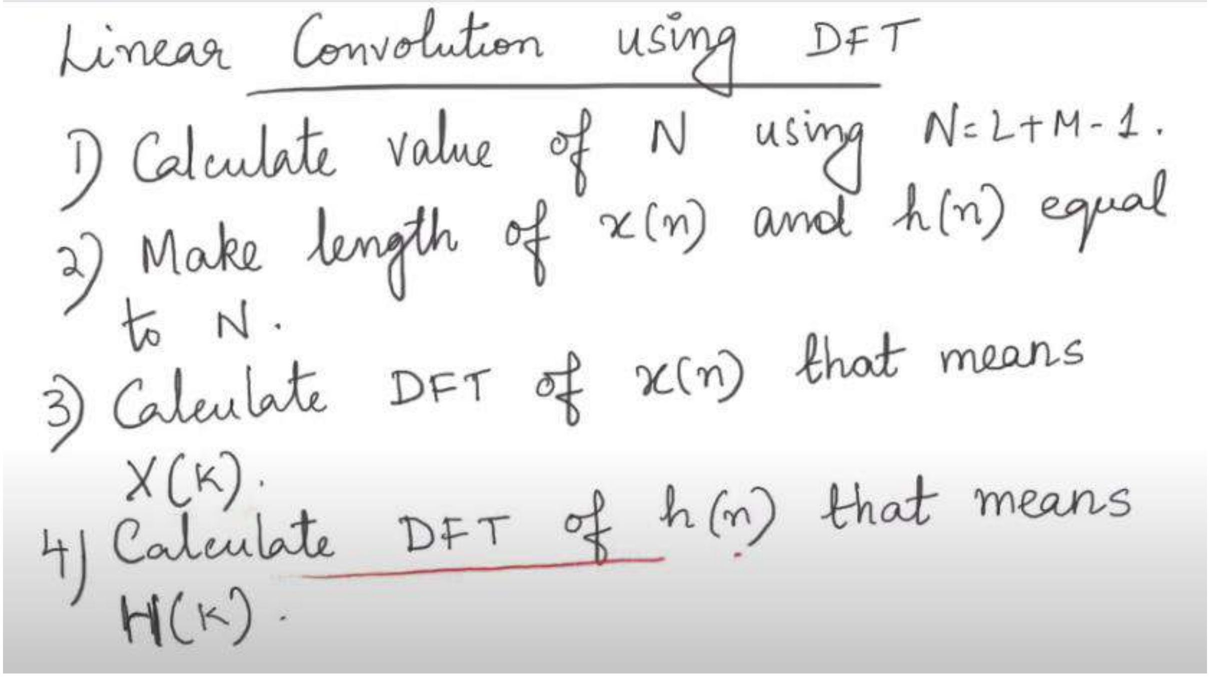
$y(n) = \{5, 2, 3, 4, 0, 0, 0\}$

$x(n) \circledast y(n) = z(n)$

$$\begin{bmatrix} 2 & 0 & 0 & 0 & 5 & 4 & 3 \\ 3 & 2 & 0 & 0 & 0 & 5 & 4 \\ 4 & 3 & 2 & 0 & 0 & 0 & 5 \\ 5 & 4 & 3 & 2 & 0 & 0 & 0 \\ 0 & 5 & 4 & 3 & 2 & 0 & 0 \\ 0 & 0 & 5 & 4 & 3 & 2 & 0 \\ 0 & 0 & 0 & 5 & 4 & 3 & 2 \end{bmatrix} \cdot \begin{bmatrix} 5 \\ 2 \\ 3 \\ 4 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 10 \\ 19 \\ 32 \\ 50 \\ 34 \\ 31 \\ 20 \end{bmatrix}$$



Linear Convolution using DFT



Linear Convolution using DFT

1) Calculate value of N using $N=L+M-1$.
2) Make length of $x(n)$ and $h(n)$ equal to N.
3) Calculate DFT of $x(n)$ that means $X(K)$.
4) Calculate DFT of $h(n)$ that means $H(K)$.

5) Multiply $X(k)$ and $H(k)$ to get

$$Y(k) = X(k) \cdot H(k)$$

6) obtain IDFT of $Y(k)$ that means $y(n)$.

eg:- $x(n) = \{1, 2\}$ and $h(n) = \{2, 2\}$

$$N = 2 + 2 - 1 = 3 \qquad \underline{\underline{N = 4}}$$

$$x(n) = \{1, 2, 0, 0\} \qquad h(n) = \{2, 2, 0, 0\}$$

$$X(k) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1+2+0+0 \\ 1-2j+0+0 \\ 1-2+0+0 \\ 1+2j+0+0 \end{bmatrix} = \begin{bmatrix} 3 \\ 1-2j \\ -1 \\ 1+2j \end{bmatrix}$$

$$H(k) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 2 \\ 2 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 4 \\ 2-2j \\ 0 \\ 2+2j \end{bmatrix}$$

$$Y(k) = X(k) \cdot H(k)$$

$$Y(0) = X(0) \cdot H(0) = 3 \times 4 = 12$$

$$Y(1) = (1-2j)(2-2j) = -2-6j$$

$$Y(2) = -1 \times 0 = 0$$

$$Y(3) = (1+2j)(2+2j) = -2+6j$$

$$Y(k) = \{12, -2-6j, 0, -2+6j\}$$

$$y(n) = \frac{1}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix}\begin{bmatrix} 12 \\ -2-6j \\ 0 \\ -2+6j \end{bmatrix}$$

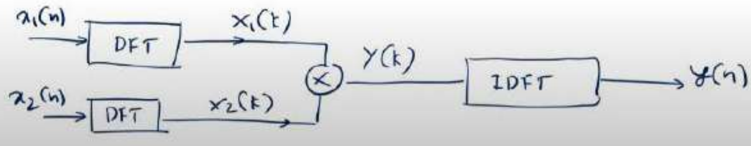
$$= \begin{bmatrix} 2 \\ 6 \\ 4 \\ 0 \end{bmatrix} \qquad y(n) = \{2, 6, 4\}$$

$$\begin{array}{c|cc} & 1 & 2 \\ \hline 2 & 2 & 4 \\ 2 & 2 & 4 \end{array} = 2, 6, 4\}$$

Compute the circular convolution using DFT and IDFT method for the following sequences

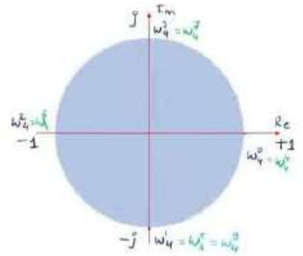
$x_1(n) = \{1, 2, 3, 1\}$ and $x_2(n) = \{4, 3, 2, 2\}$

$\Rightarrow \quad y(n) = x_1(n) \; Ⓝ \; x_2(n)$



$x_1(n) = [1 \quad 2 \quad 3 \quad 1] \qquad N=4$

using matrix method DFT is calculated as follows



$$\begin{bmatrix} X_1(0) \\ X_1(1) \\ X_1(2) \\ X_1(3) \end{bmatrix} = \begin{bmatrix} W_4^0 & W_4^0 & W_4^0 & W_4^0 \\ W_4^0 & W_4^1 & W_4^2 & W_4^3 \\ W_4^0 & W_4^2 & W_4^4 & W_4^6 \\ W_4^0 & W_4^3 & W_4^6 & W_4^9 \end{bmatrix} \begin{bmatrix} x_1(0) \\ x_1(1) \\ x_1(2) \\ x_1(3) \end{bmatrix}$$

$$\begin{bmatrix} X_1(0) \\ X_1(1) \\ X_1(2) \\ X_1(3) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & +j \\ 1 & -1 & +1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \\ 1 \end{bmatrix} = \begin{bmatrix} 1+2+3+1 \\ 1-2j-3+j \\ 1-2+3-1 \\ 1+2j-3-j \end{bmatrix} = \begin{bmatrix} 7 \\ -2-j \\ 1 \\ -2+j \end{bmatrix}$$

$x_2[n] = [4, 3, 2, 2]$

$$\begin{bmatrix} X_2(0) \\ X_2(1) \\ X_2(2) \\ X_2(3) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 4 \\ 3 \\ 2 \\ 2 \end{bmatrix} = \begin{bmatrix} 4+3+2+2 \\ 4-3j-2+2j \\ 4-3+2-2 \\ 4+3j-2-2j \end{bmatrix} = \begin{bmatrix} 11 \\ 2-j \\ 1 \\ 2+j \end{bmatrix}$$

$X_1(k) = [7, -2-j, 1, -2+j]$

$X_2(k) = [11, 2-j, 1, 2+j]$

$Y(k) = X_1(k) \cdot X_2(k) = [77, -5, 1, -5]$

$y(n) = \text{IDFT}[Y(k)]$

$$\begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ y(3) \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & +j \end{bmatrix} \begin{bmatrix} 77 \\ -5 \\ 1 \\ -5 \end{bmatrix} = \frac{1}{4} \begin{bmatrix} 77-5+1-5 \\ 77-5j-1+5j \\ 77+5+1+5 \\ 77+5j-1-5j \end{bmatrix}$$

$$= \frac{1}{4} \begin{bmatrix} 68 \\ 76 \\ 88 \\ 76 \end{bmatrix} = \begin{bmatrix} 17 \\ 19 \\ 22 \\ 19 \end{bmatrix} \qquad \therefore \; y(n) = [\underset{\uparrow}{17}, 19, 22, 19]$$

Circular Convolution

Circular Convolution

- Circular convolution (Cyclic Convolution) is a periodic convolution.
- In this x(n) and h(n) are periodic signals having same period.
- For periodic signals, x(n+N)=x(n)
- No of terms in the output is same as input.
- We can perform Linear Convolution using Circular Convolution.
- In Overlap Add and Overlap save method, it is used during intermediate calculations.
- Circular convolution can be performed in four ways.

1) Graphical 2) Tabular 3) Matrix 4) Summation

Circular Convolution Formula

If x1(n) and x2(n) are the input signal and y(n) is the output signal.

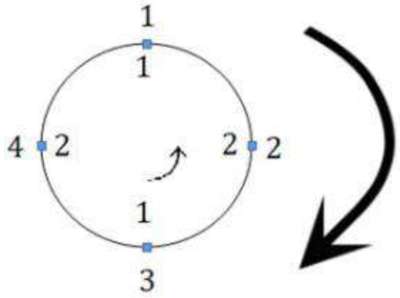
N is the period of the signal.

$$y(n) = \sum_{m=0}^{N-1} x1(m) * x2(n-m)$$

Ex. 1 Perform Circular Convolution

$$x1(n) = \{\underset{\uparrow}{1},2,3,4\} \qquad\qquad x2(n) = \{\underset{\uparrow}{1},2,1,2\}$$

1) Graphical Method



Keep x1(n) clockwise.

Keep x2(n) anticlockwise

$x1(n) = \{1,2,3,4\}$ $\curvearrowright$

$x2(n) = \{1,2,1,2\}$ $\curvearrowleft$

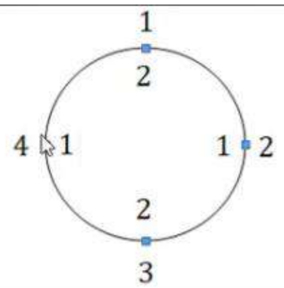
$y(0) = 1 \times 1 + 2 \times 2 + 3 \times 1 + 4 \times 2$

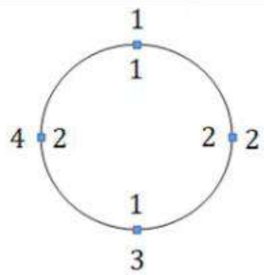
$y(0) = 1 + 4 + 3 + 8$

$y(0) = 16$

Keep x1(n) as a constant.

Move x2(n) in clockwise direction





$y(1) = 1 \times 2 + 2 \times 1 + 3 \times 2 + 4 \times 1$

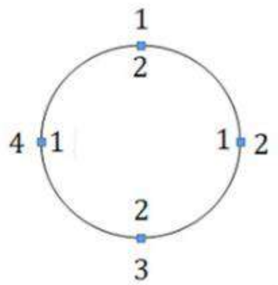
$y(1) = 2 + 2 + 6 + 4$

$y(1) = 14$

$y(2) = 1 \times 1 + 2 \times 2 + 3 \times 1 + 4 \times 2$

$y(2) = 1 + 4 + 3 + 8$

$y(2) = 16$



$y(3) = 1 \times 2 + 2 \times 1 + 3 \times 2 + 4 \times 1$

$y(3) = 2 + 2 + 6 + 4$

$y(3) = 14$

$$y(n) = \{\underset{\uparrow}{16}, 14, 16, 14\}$$

2) Matrix Method

$$y(n) = x1(n) \circledast x2(n)$$

$$\begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ y(3) \end{bmatrix} = \begin{bmatrix} 1 & 4 & 3 & 2 \\ 2 & 1 & 4 & 3 \\ 3 & 2 & 1 & 4 \\ 4 & 3 & 2 & 1 \end{bmatrix} \times \begin{bmatrix} 1 \\ 2 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 16 \\ 14 \\ 16 \\ 14 \end{bmatrix}$$

$$y(n) = \{\underset{\uparrow}{16}, 14, 16, 14\}$$

The * insided the circle is the symbol for circular convolution

3) Tabular Method

$$y(n) = \sum_{m=0}^{N-1} x1(m) * x2(n-m) \qquad y(0) = \sum_{m=0}^{N-1} x1(m) * x2(-m)$$

$$y(1) = \sum_{m=0}^{N-1} x1(m) * x2(1-m) \qquad y(2) = \sum_{m=0}^{N-1} x1(m) * x2(2-m)$$

$$y(3) = \sum_{m=0}^{N-1} x1(m) * x2(3-m)$$

	-4	-3	-2	-1	0	1	2	3	4	$y(n)$
$x1(m)$					1	2	3	4		
					↑					
$x2(m)$	1	2	1	2	1	2	1	2		
					↑					
$x2(-m)$	1	2	1	2	1	2	1	2		$=16$
					↑					
$x2(1-m)$					2	1	2	1		$=14$
					↑					
$x2(2-m)$					1	2	1	2		$=16$
					↑					
$x2(3-m)$					2	1	2	1		$=14$
					↑					

$$y(n) = \{16, 14, 16, 14\}$$

$$\uparrow$$

4) Summation Method

$$x1(n) \quad \begin{array}{cccc} 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 4 \\ \uparrow & & & \end{array}$$

$$x2(n) \quad \begin{array}{ccccccccc} -4 & -3 & -2 & -1 & 0 & 1 & 2 & 3 & 4 \\ 1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 & 1 \\ & & & & \uparrow & & & & \end{array}$$

N=4

$$y(n) = \sum_{m=0}^{N-1} x1(m) * x2(n-m) \qquad y(0) = \sum_{m=0}^{N-1} x1(m) * x2(-m)$$

$$y(0) = x1(0) * x2(-0) + x1(1) * x2(-1) + x1(2) * x2(-2) + x1(3) * x2(-3)$$

$$y(0) = 1 \times 1 + 2 \times 2 + 3 \times 1 + 4 \times 2$$

$$y(0) = 1 + 4 + 3 + 8$$

$$y(0) = 16$$

$$x1(n) \quad \begin{array}{cccc} 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 4 \\ \uparrow & & & \end{array}$$

$$x2(n) \quad \begin{array}{ccccccccc} -4 & -3 & -2 & -1 & 0 & 1 & 2 & 3 & 4 \\ 1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 & 1 \\ & & & & \uparrow & & & & \end{array}$$

$$y(1) = \sum_{m=0}^{N-1} x1(m) * x2(1-m)$$

$$y(1) = x1(0) * x2(1-0) + x1(1) * x2(1-1) + x1(2) * x2(1-2) + x1(3) * x2(1-3)$$

$$y(1) = x1(0) * x2(1) + x1(1) * x2(0) + x1(2) * x2(-1) + x1(3) * x2(-2)$$

$$\boxed{\begin{aligned} y(1) &= 1 \times 2 + 2 \times 1 + 3 \times 2 + 4 \times 1 \\ y(1) &= 2 + 2 + 6 + 4 \\ y(1) &= 14 \end{aligned}}$$

$$x1(n) \quad \begin{array}{cccc} 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 4 \\ \uparrow & & & \end{array}$$

$$x2(n) \quad \begin{array}{ccccccccc} -4 & -3 & -2 & -1 & 0 & 1 & 2 & 3 & 4 \\ 1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 & 1 \\ & & & & \uparrow & & & & \end{array}$$

$$y(n) = \sum_{m=0}^{N-1} x1(m) * x2(n-m) \qquad y(2) = \sum_{m=0}^{N-1} x1(m) * x2(2-m)$$

$$y(2) = x1(0) * x2(2-0) + x1(1) * x2(2-1) + x1(2) * x2(2-2) + x1(3) * x2(2-3)$$

$$y(2) = x1(0) * x2(2) + x1(1) * x2(1) + x1(2) * x2(0) + x1(3) * x2(-1)$$

$$\boxed{\begin{aligned} y(2) &= 1 \times 1 + 2 \times 2 + 3 \times 1 + 4 \times 2 \\ y(2) &= 1 + 4 + 3 + 8 \\ y(2) &= 16 \end{aligned}}$$

$x1(n)$	0	1	2	3
	1	2	3	4
	$\uparrow$			

$x2(n)$	-4	-3	-2	-1	0	1	2	3	4
	1	2	1	2	1	2	1	2	1
					$\uparrow$				

$$y(n) = \sum_{m=0}^{N-1} x1(m) * x2(n-m) \qquad y(3) = \sum_{m=0}^{N-1} x1(m) * x2(3-m)$$

$$y(3) = x1(0) * x2(3-0) + x1(1) * x2(3-1) + x1(2) * x2(3-2) + x1(3) * x2(3-3)$$

$$y(3) = x1(0) * x2(3) + x1(1) * x2(2) + x1(2) * x2(1) + x1(3) * x2(0)$$

$$y(3) = 1 \times 2 + 2 \times 1 + 3 \times 2 + 4 \times 1$$

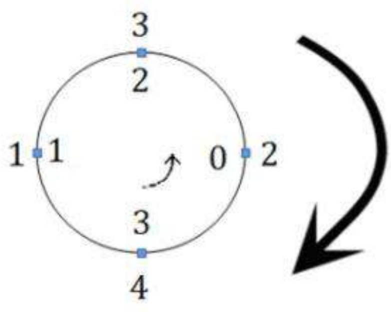
$$y(3) = 2 + 2 + 6 + 4$$

$$y(3) = 14$$

Ex. 2 Perform Circular Convolution

$$x1(n) = \{3, 2, 4, 1\} \qquad x2(n) = \{2, 1, 3\} \qquad N = 4$$

1) Graphical Method



Keep x1(n) clockwise.

Keep x2(n) anticlockwise

$$x1(n) = \{3, 2, 4, 1\}$$

$$x2(n) = \{2, 1, 3, 0\}$$

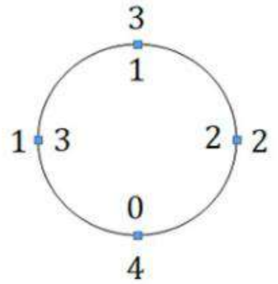
$$y(0) = 3 \times 2 + 2 \times 0 + 4 \times 3 + 1 \times 1$$

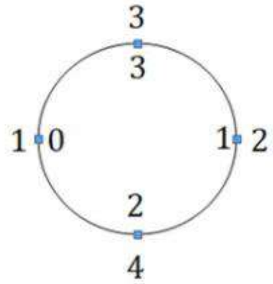
$$y(0) = 6 + 0 + 12 + 1$$

$$y(0) = 19$$

Keep x1(n) as a constant.

Move x2(n) in clockwise direction





$$y(1) = 3 \times 1 + 2 \times 2 + 4 \times 0 + 1 \times 3$$

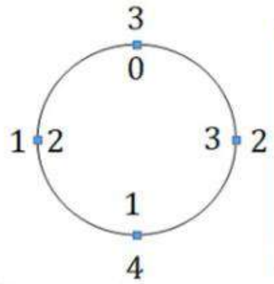
$$y(1) = 3 + 4 + 0 + 3$$

$$y(1) = 10$$

$$y(2) = 3 \times 3 + 2 \times 1 + 4 \times 2 + 1 \times 0$$

$$y(2) = 9 + 2 + 8 + 0$$

$$y(2) = 19$$



$$y(3) = 3 \times 0 + 2 \times 3 + 4 \times 1 + 1 \times 2$$

$$y(3) = 0 + 6 + 4 + 2$$

$$y(3) = 12$$

$$y(n) = \{\underset{\uparrow}{19}, 10, 19, 12\}$$

2) Matrix Method

$$y(n) = x1(n) \circledast x2(n)$$

$$\begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ y(3) \end{bmatrix} = \begin{bmatrix} 3 & 1 & 4 & 2 \\ 2 & 3 & 1 & 4 \\ 4 & 2 & 3 & 1 \\ 1 & 4 & 2 & 3 \end{bmatrix} \times \begin{bmatrix} 2 \\ 1 \\ 3 \\ 0 \end{bmatrix} = \begin{bmatrix} 19 \\ 10 \\ 19 \\ 12 \end{bmatrix}$$

$$y(n) = \{\underset{\uparrow}{19}, 10, 19, 12\}$$

3) Tabular Method

	-4	-3	-2	-1	0	1	2	3	4	$y(n)$
$x1(m)$					3	2	4	1		
					$\uparrow$					
$x2(m)$	2	1	3	0	2	1	3	0		
					$\uparrow$					
$x2(-m)$	2	0	3	1	2	0	3	1		$=19$
					$\uparrow$					
$x2(1-m)$					1	2	0	3		$=10$
					$\uparrow$					
$x2(2-m)$					3	1	2	0		$=19$
					$\uparrow$					
$x2(3-m)$					0	3	1	2		$=12$
					$\uparrow$					

$$y(n) = \{\underset{\uparrow}{19}, 10, 19, 12\}$$

4) Summation Method

$$x1(n) \qquad \begin{array}{cccc} 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 4 \\ \uparrow & & & \end{array}$$

$$x2(n) \qquad \begin{array}{ccccccccc} -4 & -3 & -2 & -1 & 0 & 1 & 2 & 3 & 4 \\ 1 & 2 & 1 & 2 & 1 & 2 & 1 & 2 & 1 \\ & & & & \uparrow & & & & \end{array}$$

N=4

$$y(n) = \sum_{m=0}^{N-1} x1(m) \ast x2(n-m) \qquad y(0) = \sum_{m=0}^{N-1} x1(m) \ast x2(-m)$$

$$y(0) = x1(0) \ast x2(-0) + x1(1) \ast x2(-1) + x1(2) \ast x2(-2) + x1(3) \ast x2(-3)$$

$$y(0) = 1 \times 1 + 2 \times 2 + 3 \times 1 + 4 \times 2$$

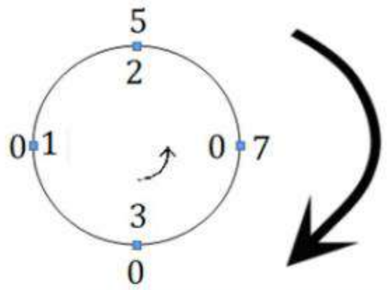
$$y(0) = 1 + 4 + 3 + 8$$

$$y(0) = 16$$

Ex. 3 **Perform 4 point Circular Convolution**

$$x1(n) = \{\underset{\uparrow}{5},7\} \qquad\qquad x2(n) = \{\underset{\uparrow}{2},1,3\}$$

1) Graphical Method



Keep x1(n) clockwise.

Keep x2(n) anticlockwise

$$x1(n) = \{5,7,0,0\}$$

$$x2(n) = \{2,1,3,0\}$$

$$y(0) = 5 \times 2 + 0 \times 7 + 0 \times 3 + 0 \times 1$$
$$y(0) = 10 + 0 + 0 + 0$$
$$y(0) = 10$$

Keep x1(n) as a constant.

Move x2(n) in clockwise direction

2) Matrix Method

$$y(n) = x1(n) \circledast x2(n)$$

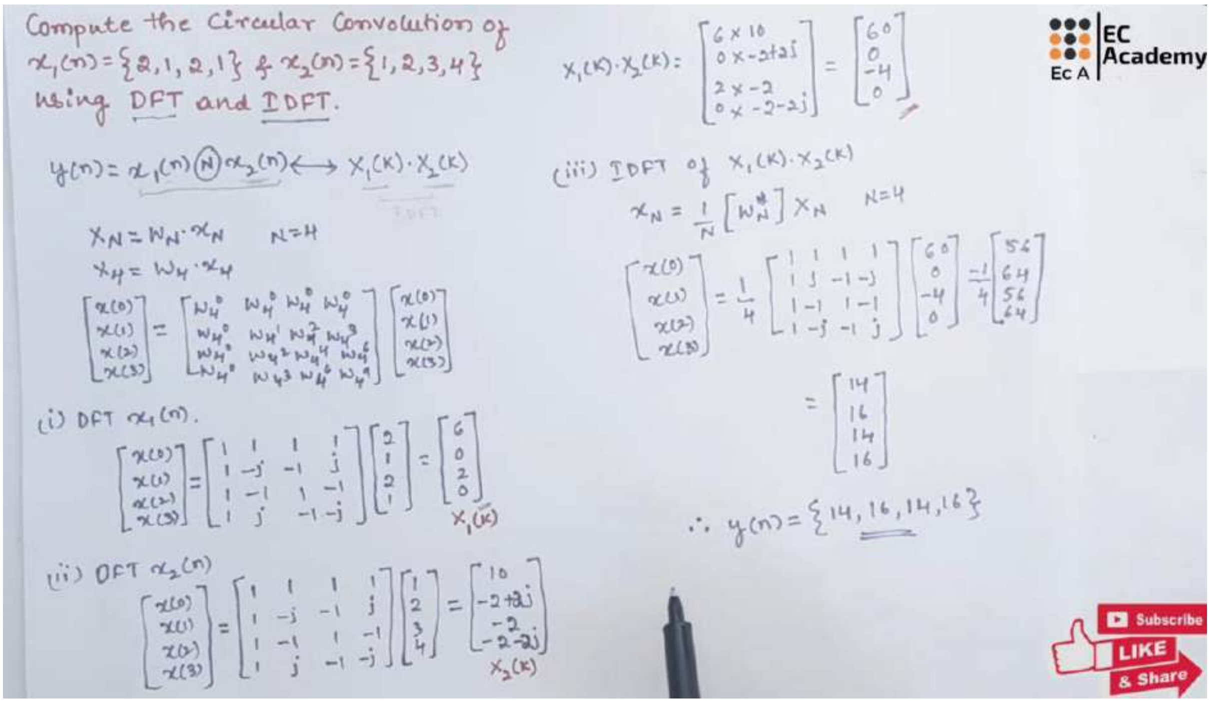
$$\begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ y(3) \end{bmatrix} = \begin{bmatrix} 5 & 0 & 0 & 7 \\ 7 & 5 & 0 & 0 \\ 0 & 7 & 5 & 0 \\ 0 & 0 & 7 & 5 \end{bmatrix} \times \begin{bmatrix} 2 \\ 1 \\ 3 \\ 0 \end{bmatrix} = \begin{bmatrix} 10 \\ 19 \\ 22 \\ 21 \end{bmatrix}$$

$$y(n) = \{\underset{\uparrow}{19},20,22,21\}$$

3) Tabular Method

	-4	-3	-2	-1	0	1	2	3	4	$y(n)$
$x1(m)$					2	1	3	0		
					↑					
$x2(m)$	5	7	0	0	5	7	0	0	5	
					↑					
$x2(-m)$	5	0	0	7	5	0	0	7		$= 10$
					↑					
$x2(1-m)$					7	5	0	0		$= 19$
					↑					
$x2(2-m)$					0	7	5	0		$= 22$
					↑					
$x2(3-m)$					0	0	7	5		$= 21$
					↑					

$$y(n) = \{\underset{\uparrow}{19}, 20, 22, 21\}$$


Compute the Circular Convolution of

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

using DFT and IDFT.

$y(n) = x_1(n) \circledN x_2(n) \longleftrightarrow X_1(K)\cdot X_2(K)$

$X_N = W_N \cdot x_N \quad N=4$

$X_4 = W_4 \cdot x_4$

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} W_4^0 & W_4^0 & W_4^0 & W_4^0 \\ W_4^0 & W_4^1 & W_4^2 & W_4^3 \\ W_4^0 & W_4^2 & W_4^4 & W_4^6 \\ W_4^0 & W_4^3 & W_4^6 & W_4^9 \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix}$$

(i) DFT $x_1(n)$.

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 2 \\ 1 \\ 2 \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} 6 \\ 0 \\ 2 \\ 0 \end{bmatrix}}_{X_1(K)}$$

(ii) DFT $x_2(n)$

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} = \underbrace{\begin{bmatrix} 10 \\ -2+2j \\ -2 \\ -2-2j \end{bmatrix}}_{X_2(K)}$$

$$X_1(K)\cdot X_2(K) = \begin{bmatrix} 6\times 10 \\ 0\times -2+2j \\ 2\times -2 \\ 0\times -2-2j \end{bmatrix} = \begin{bmatrix} 60 \\ 0 \\ -4 \\ 0 \end{bmatrix}$$

(iii) IDFT of $X_1(K)\cdot X_2(K)$

$x_N = \frac{1}{N}\left[W_N^*\right] X_N \quad N=4$

$$\begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix} = \frac{1}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & j \end{bmatrix} \begin{bmatrix} 60 \\ 0 \\ -4 \\ 0 \end{bmatrix} = \frac{1}{4}\begin{bmatrix} 56 \\ 64 \\ 56 \\ 64 \end{bmatrix}$$

$$= \begin{bmatrix} 14 \\ 16 \\ 14 \\ 16 \end{bmatrix}$$

$\therefore y(n) = \{14, 16, 14, 16\}$

Numerical on Discrete Convolution

Discrete Time Systems

- Find the Convolution of two finite duration sequences:

$$h(n) = a^n u(n) \quad \text{for all } n$$

$$x(n) = b^n u(n) \quad \text{for all } n$$

(i) When $a \neq b$

(ii) When $a = b$

$h(n) \rightarrow$ impulse response

$x(n) \rightarrow$ i/p

$$h(n) = a^n \underbrace{u(n)}$$

$$= a^n \cdot 1 \qquad n \geq 0$$

$$h(n) = a^n \qquad n \geq 0$$

$$x(n) = b^n \underbrace{u(n)}$$

$$= b^n \cdot 1 \qquad n \geq 0$$

$$x(n) = b^n \qquad n \geq 0$$

$h(n)$ & $x(n) = 0 \quad n < 0$

$h(n) = 0 \quad n < 0 \rightarrow$ causal system

$x(n) = 0 \quad n < 0 \rightarrow$ causal i/p

$$y(n) = x(n) * h(n) = \sum_{m=-\infty}^{+\infty} x(m)\, h(n-m)$$

For a causal system excited by a causal i/p.

$$y(n) = \sum_{m=0}^{n} x(m)\, h(n-m)$$

$x(n) = b^n \quad n \geq 0$

$h(n) = a^n \quad n \geq 0$

$$y(n) = \sum_{m=0}^{n} x(m)\, h(n-m)$$

$$= \sum_{m=0}^{n} b^m \cdot a^{(n-m)}$$

$$= \sum_{m=0}^{n} b^m \cdot \frac{a^n}{a^m}$$

$$= \sum_{m=0}^{n} a^n \cdot \frac{b^m}{a^m} = a^n \sum_{m=0}^{n} \left(\frac{b}{a}\right)^m$$

(1) when $a \neq b$

$$y(n) = a^n \underbrace{\sum_{m=0}^{n} \left(\frac{b}{a}\right)^m}$$

geometric sum

$$\sum_{i=0}^{n} r^i = \frac{1 - r^{n+1}}{1 - r}, \quad r \neq 1$$

$$y(n) = a^n \left[\frac{1 - \left(\frac{b}{a}\right)^{m+1}}{1 - \frac{b}{a}} \right]$$

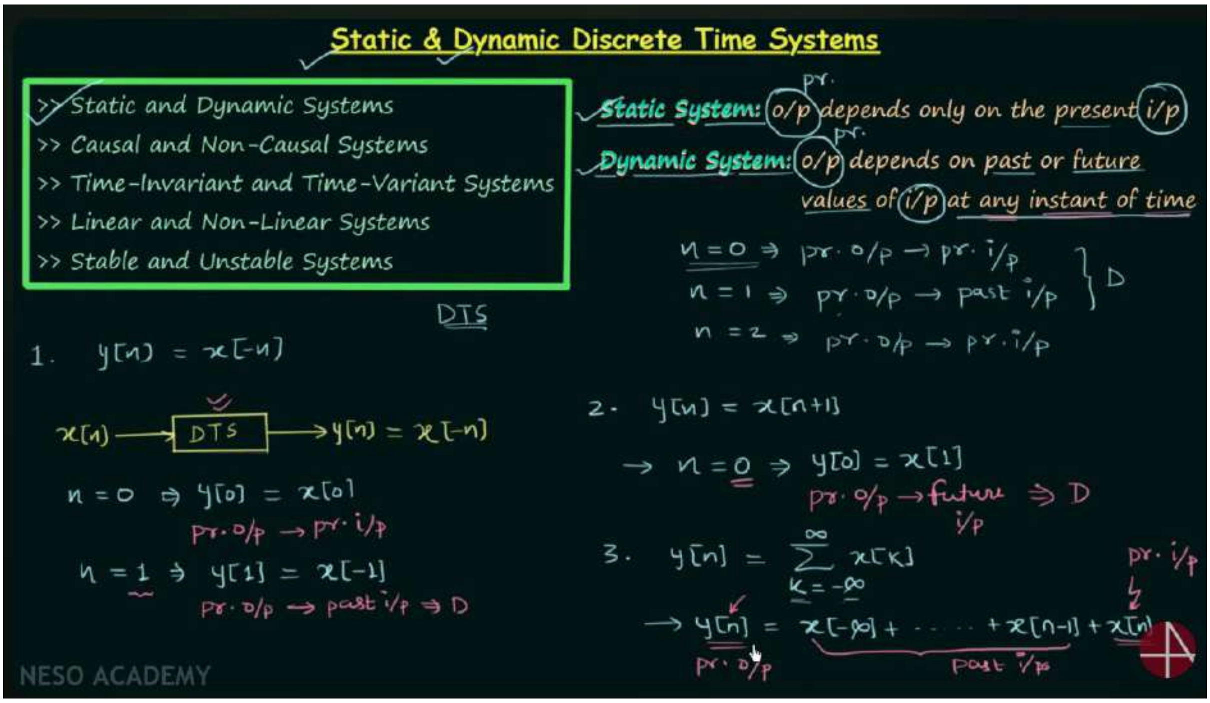
② when $a = b$

$$y(n) = a^n \sum_{m=0}^{n} \left[\frac{a}{a}\right]^m = a^n \sum_{m=0}^{n} (1)^m$$

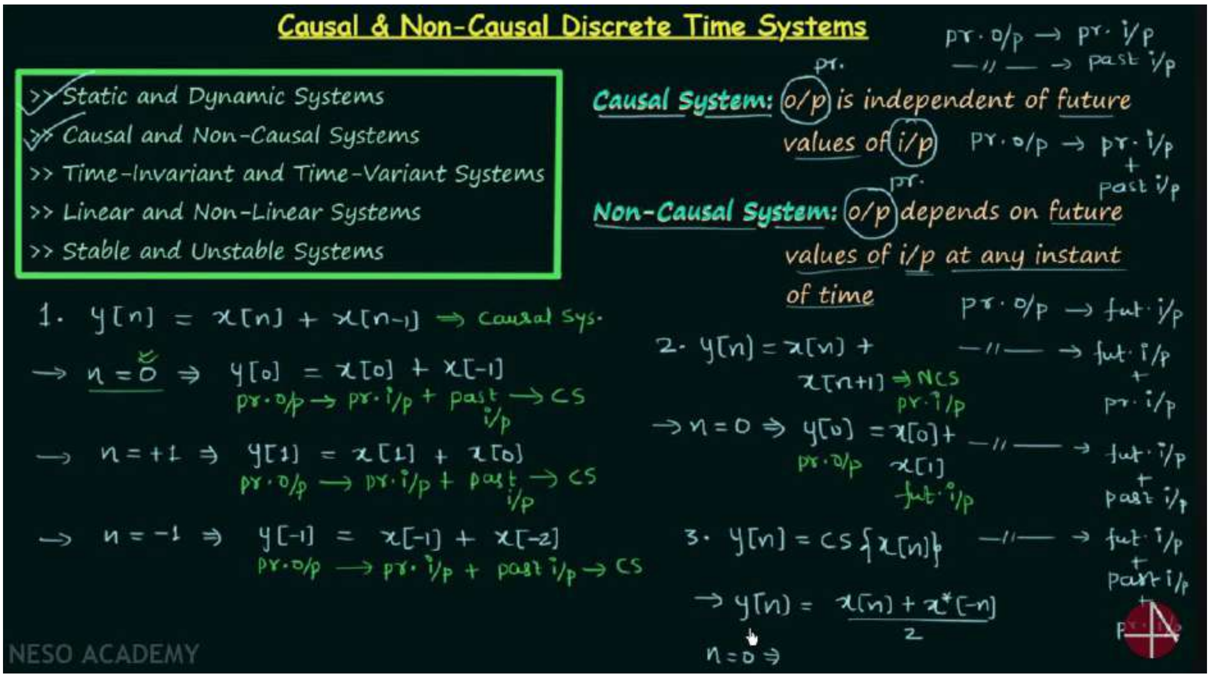
$$y(n) = a^n \left[1 + 1 + 1 + 1 + 1 + \cdots (n+1)\right]$$

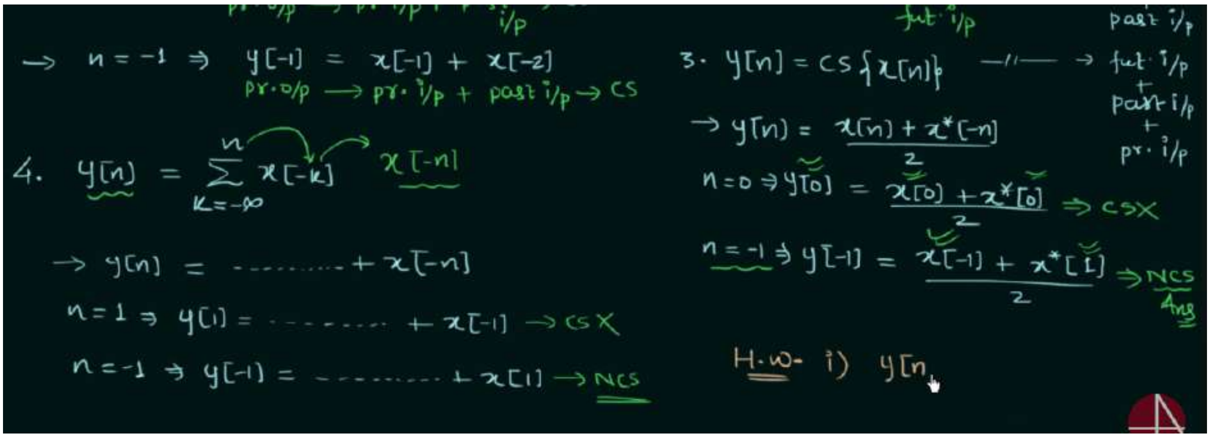
$$y(n) = a^n \left[n + 1\right]$$

Static & Dynamic Discrete Time Systems

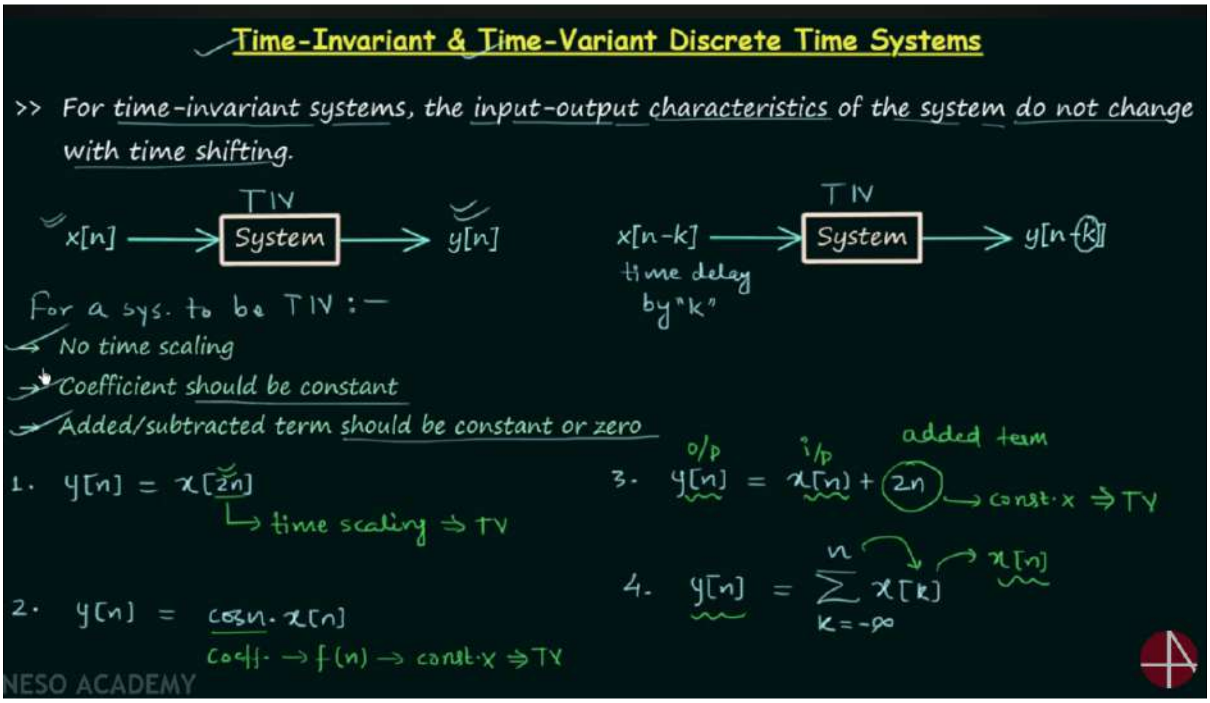


Causal and Non-Causal Discrete Time Systems

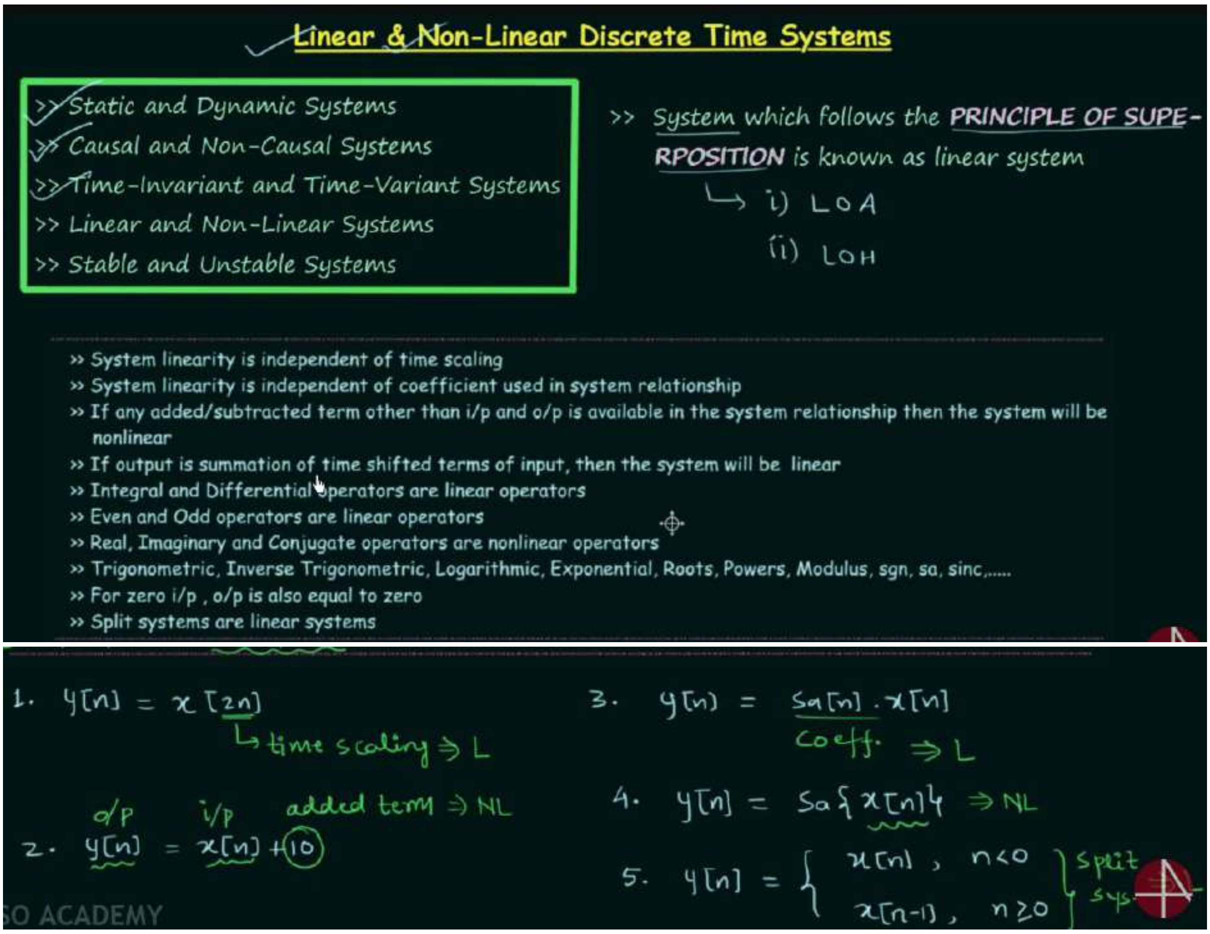




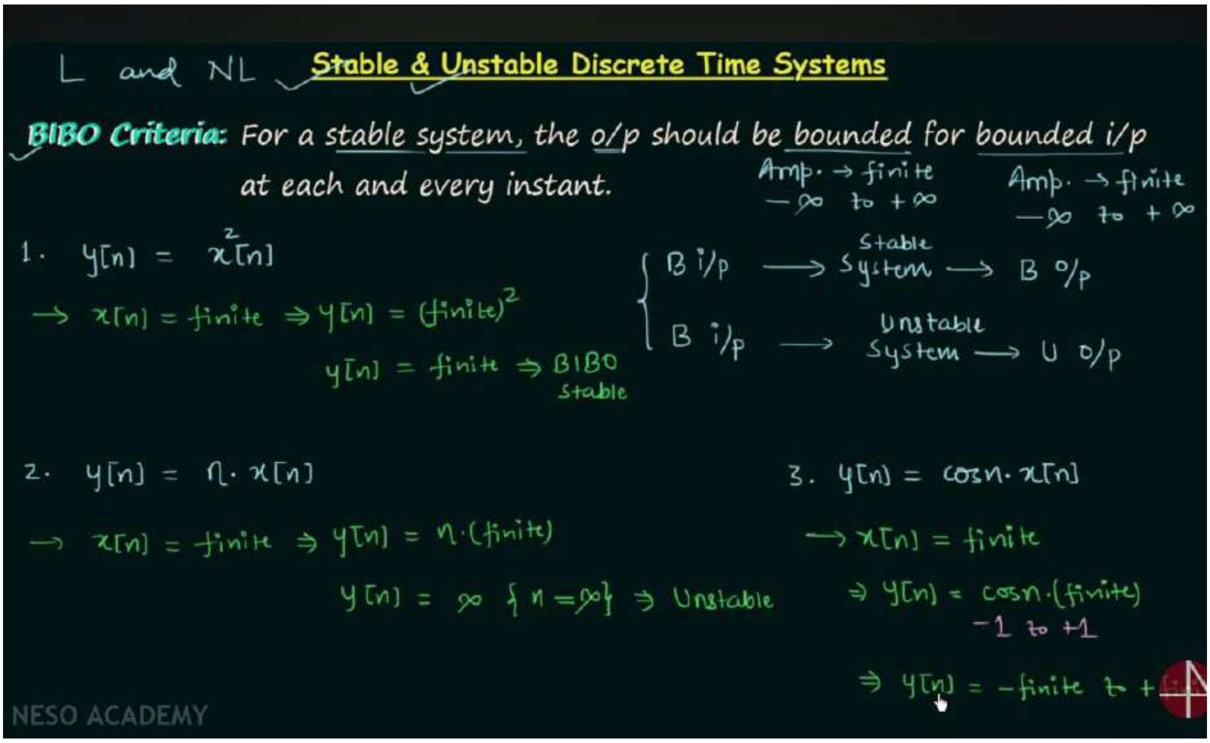
Time−Invariant & Time−Variant Discrete Time Systems

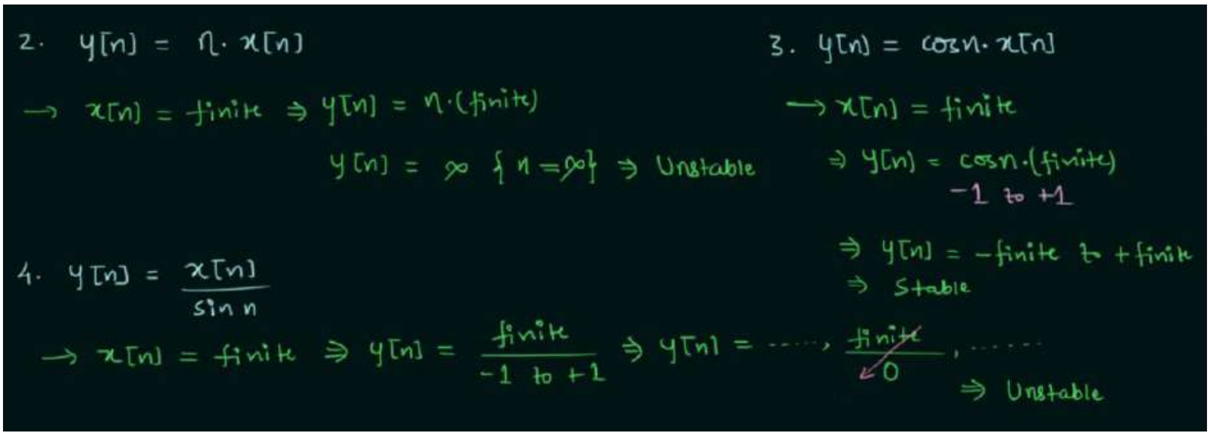


Linear and Non−Linear Discrete Time Systems

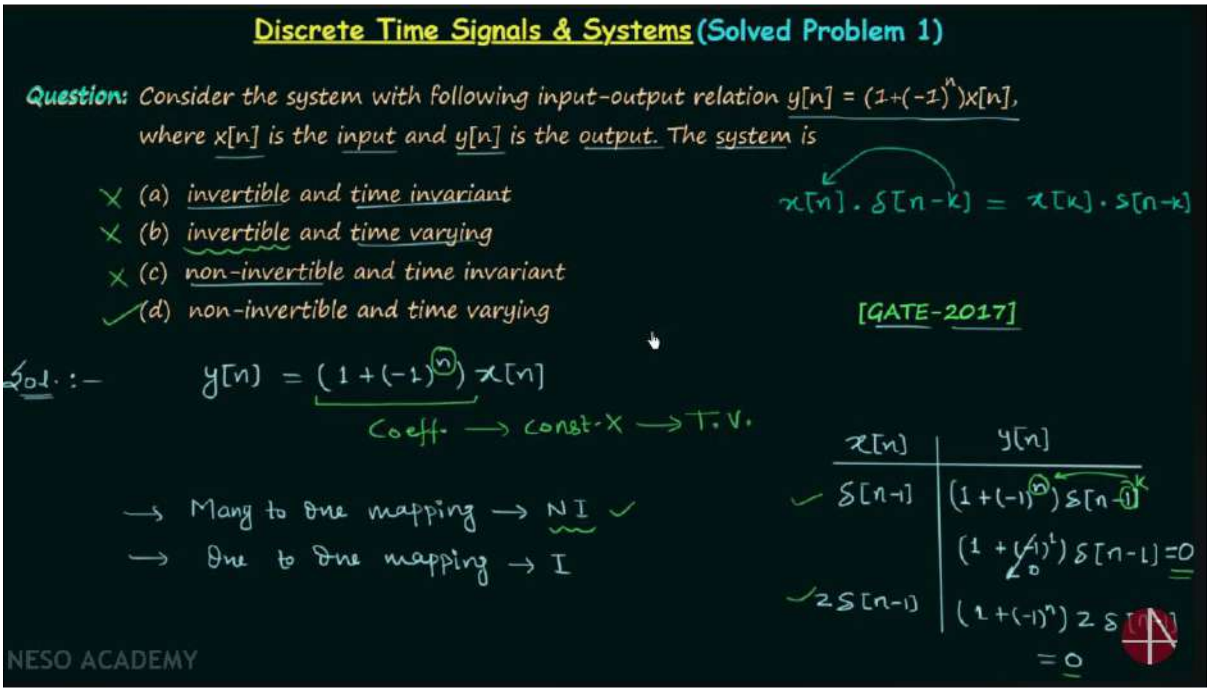


Stable & Unstable Discrete Time Systems





Solved Problems



Discrete Time Signals & Systems (Solved Problem 2)

Question: Two sequences $x_1[n]$ and $x_2[n]$ have the same energy. Suppose $x_1[n] = \alpha\, 0.5^n u[n]$, where α is a positive real number and $u[n]$ is the unit step sequence. Assume

$$x_2[n] = \begin{cases} \sqrt{1.5} & \text{for } n = 0, 1 \\ 0 & \text{otherwise} \end{cases}$$

EC

Then the value of α is ________. [GATE-2015]

Sol. :-

$$E_{x_1[n]} = E_{x_2[n]}$$

$$\sum_{n=-\infty}^{\infty} |x_1[n]|^2 = \sum_{n=-\infty}^{\infty} |x_2[n]|^2$$

$$\sum_{n=-\infty}^{\infty} |\alpha\, 0.5^n u[n]|^2 = \underbrace{\cancel{\sum_{n=-\infty}^{-1} |0|^2}}_{0} + \sum_{n=0}^{1} |\sqrt{1.5}|^2 + \underbrace{\cancel{\sum_{n=2}^{\infty} |0|^2}}_{0}$$

$$\sum_{n=0}^{\infty} |\alpha\, 0.5^n|^2 = \sum_{n=0}^{1} 1.5 \Rightarrow \sum_{n=0}^{\infty} \alpha^2\, 0.5^{2n} = 1.5 + 1.5$$

EC

Then the value of α is 1.5. [GATE-2015]

Sol. :-

$$E_{x_1[n]} = E_{x_2[n]}$$

$$\sum_{n=-\infty}^{\infty} |x_1[n]|^2 = \sum_{n=-\infty}^{\infty} |x_2[n]|^2$$

$$\sum_{n=-\infty}^{\infty} |\alpha\, 0.5^n u[n]|^2 = \underbrace{\cancel{\sum_{n=-\infty}^{-1} |0|^2}}_{0} + \sum_{n=0}^{1} |\sqrt{1.5}|^2 + \underbrace{\cancel{\sum_{n=2}^{\infty} |0|^2}}_{0}$$

$$\sum_{n=0}^{\infty} |\alpha\, 0.5^n|^2 = \sum_{n=0}^{1} 1.5 \Rightarrow \sum_{n=0}^{\infty} \alpha^2\, 0.5^{2n} = 1.5 + 1.5$$

$$\alpha^2 \underbrace{\{ \overset{a}{1} + 0.5^2 + 0.5^{2\cdot 2} + 0.5^{2\cdot 3} + \dots \}}_{r = 0.5^2 \Rightarrow -1 < r < 1 \Rightarrow S_\infty = \frac{a}{1-r}} = 3 \Rightarrow \alpha^2 \times \frac{1}{1 - 0.5^2} = 3$$

$$\alpha^2 \times \frac{1}{0.75} = 3 \Rightarrow \alpha = \pm$$

Discrete Time Signals & Systems (Solved Problem 3)

Question: Consider a discrete-time signal

$$x[n] = \begin{cases} n & \text{for } 0 \le n \le 10 \\ 0 & \text{otherwise} \end{cases} \Rightarrow x[k] = \begin{cases} k & , \ 0 \le k \le 10 \\ 0 & , \ \text{otherwise} \end{cases}$$

EC

If y[n] is the convolution of x[n] with itself, the value of y[4] is ___?___. [GATE-2014]

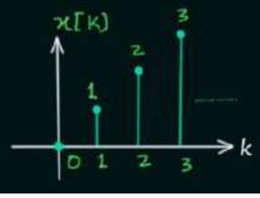
Sol:- $y[n] = x[n] * x[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot x[n-k]$

$x[n] \quad x[n]$

$\downarrow n \to k$

$x[k] \quad x[k] \xrightarrow{T.R.} x[-k] \xrightarrow[\text{"}n\text{"}]{T.S.} x[n-k]$

×



$y[n] = \sum_{k=1}^{n} k \cdot (n-k)$

$y[4] = \sum_{k=1}^{4} k \cdot (4-k)$

$y[4] = 3 + 4 + 3 + 0$

$y[4] = 10$ Ans

NESO ACADEMY

Discrete Time Signals & Systems (Solved Problem 4)

Question: Let y[n] denote the convolution of h[n] and g[n], where $h[n] = (1/2)^n u[n]$ and g[n] is a causal sequence. If y[0] = 1 and y[1] = 1/2, then g[1] equals ?

(a) 0 (b) 1/2 (c) 1 (d) 3/2 [GATE-2012]

Sol:- $y[n] = h[n] * g[n]$

$h[n] = (1/2)^n u[n]$

$g[n] \to$ Causal seq. $\Rightarrow g[n] = 0$ for $n < 0$

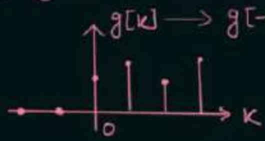
$y[n] = \sum_{k=-\infty}^{\infty} h[k] \cdot g[n-k]$

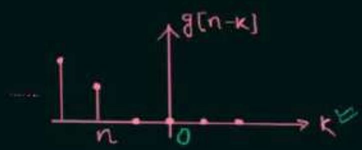
$= \sum_{k=-\infty}^{\infty} (1/2)^k u[k] \cdot g[n-k]$

$= \sum_{k=0}^{n} (1/2)^k g[n-k]$

$g[k] \to$ c.s. $\Rightarrow g[k] = 0 \ , \ k < 0$

$g[k] \longrightarrow g[-k] \longrightarrow g[n-k]$



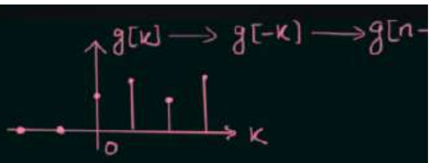


NESO ACADEMY

(a) 0 (b) 1/2 (c) 1 (d) 3/2 [GATE-2012]

$h[n] = (1/2)^n u[n]$

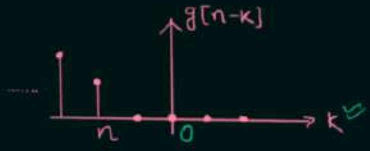
$g[n] \longrightarrow$ Causal seq. $\Rightarrow g[n] = 0$ for $n<0$



$$y[n] = \sum_{k=-\infty}^{\infty} h[k]\cdot g[n-k]$$

$$= \sum_{k=-\infty}^{\infty} (1/2)^k u[k]\cdot g[n-k]$$

$$= \sum_{k=0}^{n} (1/2)^k g[n-k]$$



$n=0$

$$y[0] = (1/2)^0 g[0-0]$$

$$1 = g[0]$$

$n=1$

$$y[1] = \sum_{k=0}^{1} (1/2)^k g[1-k]$$

$$= (1/2)^0 g[1-0] + (1/2)^1 g[1-1]$$

$$y[1] = g[1] + 1/2$$

$$1/2 = g[1] + 1/2 \Rightarrow g[1] = 0$$

Discrete Time Signals & Systems (Solved Problem 5)

Question: A system is defined by its impulse response $h[n] = 2^n u[n-2]$. The system is ?

(a) stable and causal
(b) causal and not stable
× (c) stable but not causal
× (d) unstable and non-causal

Causal or Non-causal? AND Stable or Unstable? [GATE-2011]

Sol.:- Causality :- $h[n] = 0,\ n<0 \Rightarrow$ C.S.

Stability :- $\underbrace{\sum_{n=-\infty}^{\infty} |h[n]|}_{\text{finite}} < \infty \Rightarrow$ S.S.

$u[n-2] = 0,\ n<0$

$\rightarrow h[n] = 2^n u[n-2] = 0,\ n<0 \Rightarrow$ causal ✓

(c)× (d)×

$$\rightarrow \sum_{n=-\infty}^{\infty} |h[n]| = \sum_{n=-\infty}^{\infty} |2^n u[n-2]|$$

$\rightarrow h[n] = 2^n u[n-2] = 0,\ n<0 \Rightarrow$ causal ✓

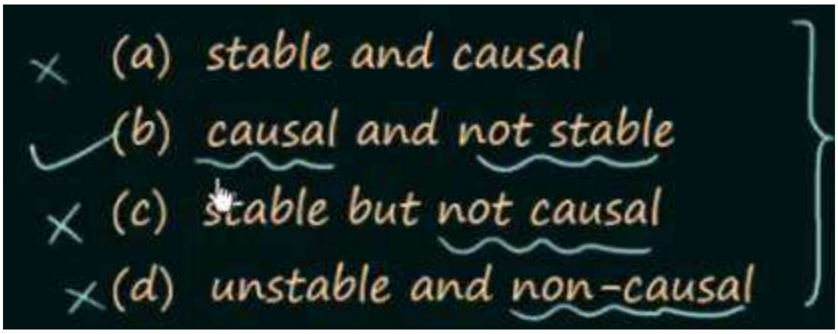
(c)× (d)×

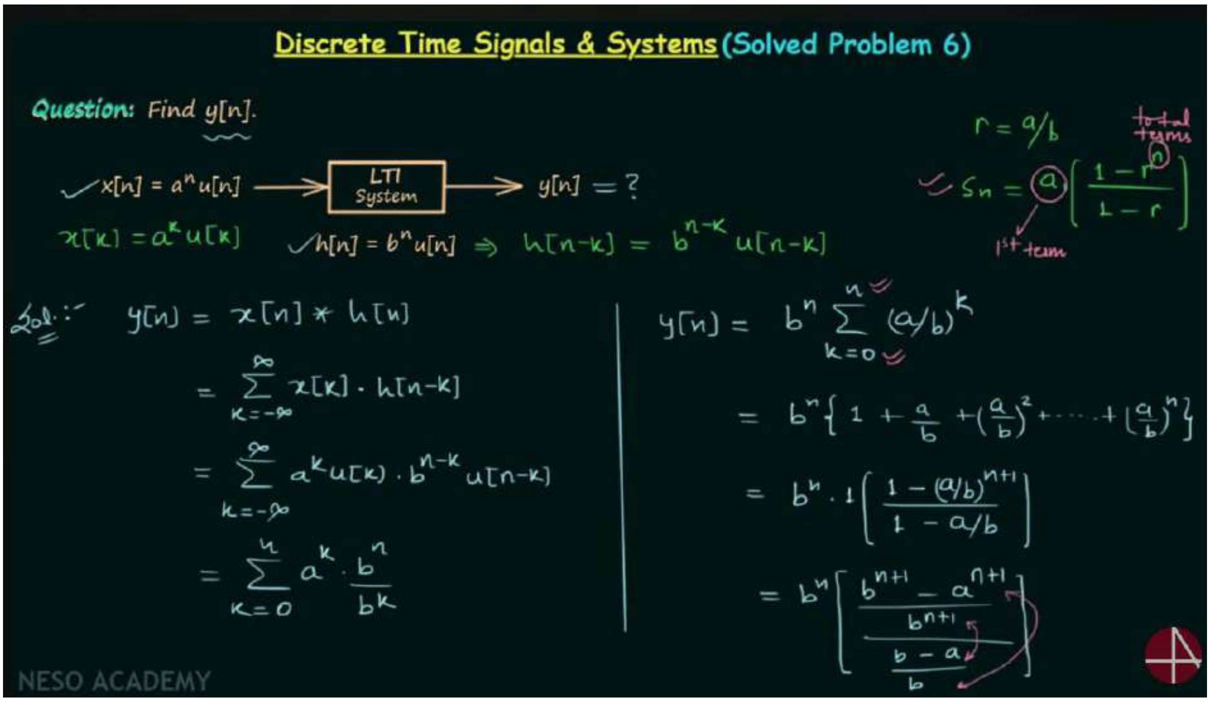
$$\rightarrow \sum_{n=-\infty}^{\infty} |h[n]| = \sum_{n=-\infty}^{\infty} |2^n u[n-2]|$$

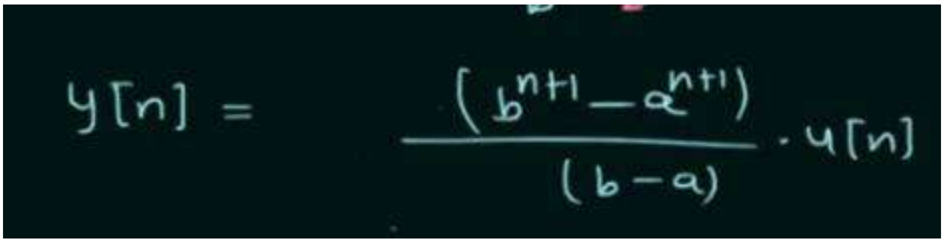
$$= \sum_{n=2}^{\infty} 2^n$$

$$= 2^2 + 2^3 + \cdots + 2^{\infty}$$

$$= \infty$$



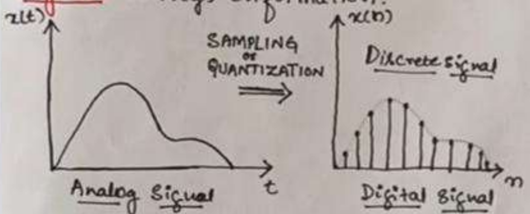




Introduction to Digital Signal Processing

Introduction - Digital Signal Processing

Signal: Conveys information.



Signal Processing:

Analyse, Modify and Sinthesize.

(i) Analog signal processing.

(ii) Digital signal processing.

Block diagram - DSP:

Analog input Signal x(t) → Pre-Filter → A/D Converter → Digital Signal Processor → D/A Converter → Post-Filter → Analog O/P Signal y(t)

Applications - DSP:

→ Speech & Audio.
→ Image & video.
→ Military & space.
→ Biomedical & Heath care.
→ Consumer electronics.

Advantages - DSP:

→ Flexible in operation.
→ Accurate results.
→ Stable sym.
→ Data storage - less expensive.
→ Low cost.

Disadvantages - DSP:

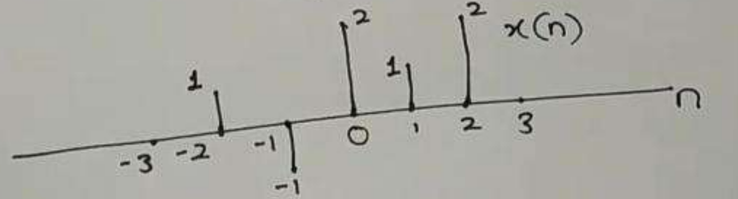
→ Limited speed of operation.

REPRESENTATION OF DISCRETE TIME LTI SYSTEMS IN TERMS OF IMPULSE RESPONSES

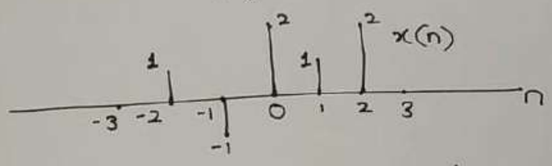
Representation for discrete time LTI Systems in terms of Impulse Responses.

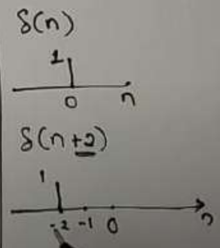
→ we are going to construct discrete time signal "x(n)" in terms of discrete time shifted "unit Impulse"

→ Let us consider a discrete time signal x(n) shown in figure below.



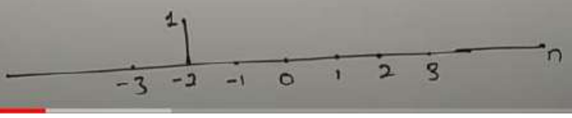
→ Let us consider a discrete time signal $x(n)$ shown in figure below.



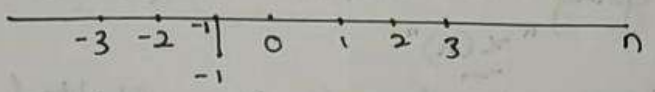


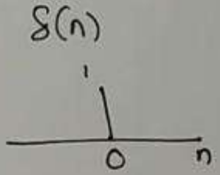
The signal $x(n)$ can be splitted as

$$x_1(n) = x(-2)\,\delta(n+2)$$

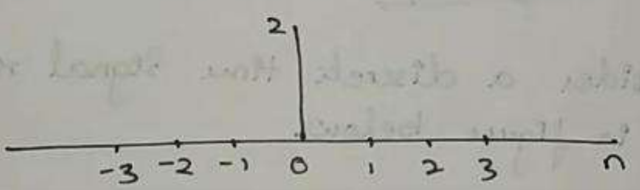


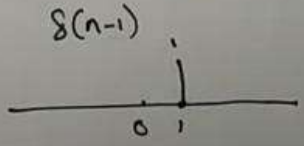
$$x_2(n) = x(-1)\,.\,\delta(n+1)$$



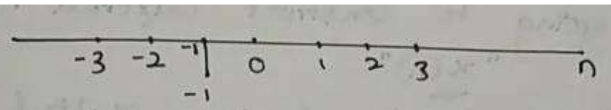


$$x_3(n) = x(0)\,.\,\delta(n)$$

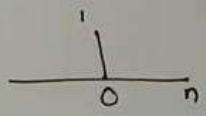




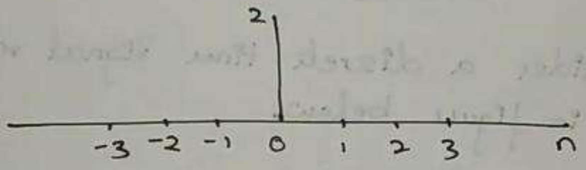
$$x_4(n) = x(1)\,.\,\delta(n-1)$$



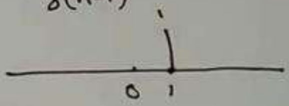
$\delta(n)$



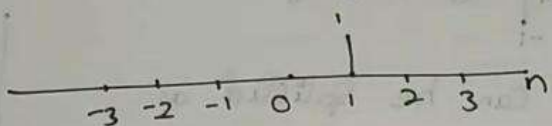
$x_3(n) = x(0).\ \delta(n)$ —ⓒ



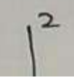
$\delta(n-1)$



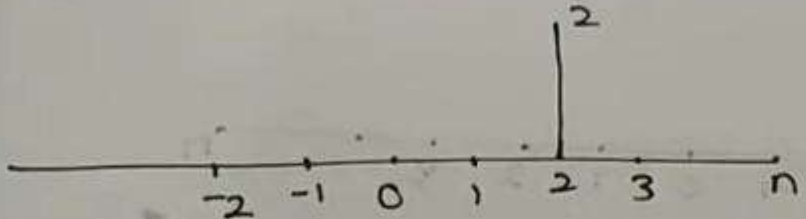
$x_4(n) = x(1).\ \delta(n-1)$ —ⓓ



$x_5(n) = x(2).\ \delta(n-2)$ —ⓔ



$x_5(n) = x(2).\ \delta(n-2)$



from these eqn ⓐ, ⓑ, ⓒ, ⓓ and ⓔ

we can write

$$x(n) = x_1(n) + x_2(n) + x_3(n) + x_4(n) + x_5(n)$$

$$\therefore x(n) = x(-2)\,\delta(n+2) + x(-1)\,\delta(n+1) + x(0).\,\delta(n) + x(1).\,\delta(n-1) + x(2).\,\delta(n-2) \quad \text{—①}$$

$$x(n) = \sum_{k=-2}^{2} x(k)\,\delta(n-k) \quad \text{—②}$$

In general, $x(n) = \sum_{k=-\infty}^{\infty} x(k).\,\delta(n-k)$ —③

$$x(n) \longrightarrow \boxed{H} \longrightarrow y(n)$$

$\therefore x(n) = x(-2)\,\delta(n+2) + x(-1)\,\delta(n+1) + x(0).\delta(n) + x(1).\delta(n-1) + x(2).\delta(n-2)$ — ①

$$x(n) = \sum_{k=-2}^{2} x(k)\,\delta(n-k) \quad -②$$

In general, $x(n) = \sum_{k=-\infty}^{\infty} x(k).\delta(n-k) \quad -③$

$$x(n) \longrightarrow \boxed{H} \longrightarrow y(n)$$

$$y(n) = H\{x(n)\} \quad -④$$

$$y(n) = H\left\{\sum_{k=-\infty}^{\infty} x(k).\delta(n-k)\right\}$$

$$y(n) = \sum_{k=-\infty}^{\infty} x(k).H\{\delta(n-k)\} \quad -⑤$$

In eqn ⑤ $y(n) = \sum_{k=-\infty}^{\infty} x(k).H\{\delta(n-k)\}$

$H\{\delta(n-k)\}$ is the operation performed on time shifted Impulse $\delta(n-k)$

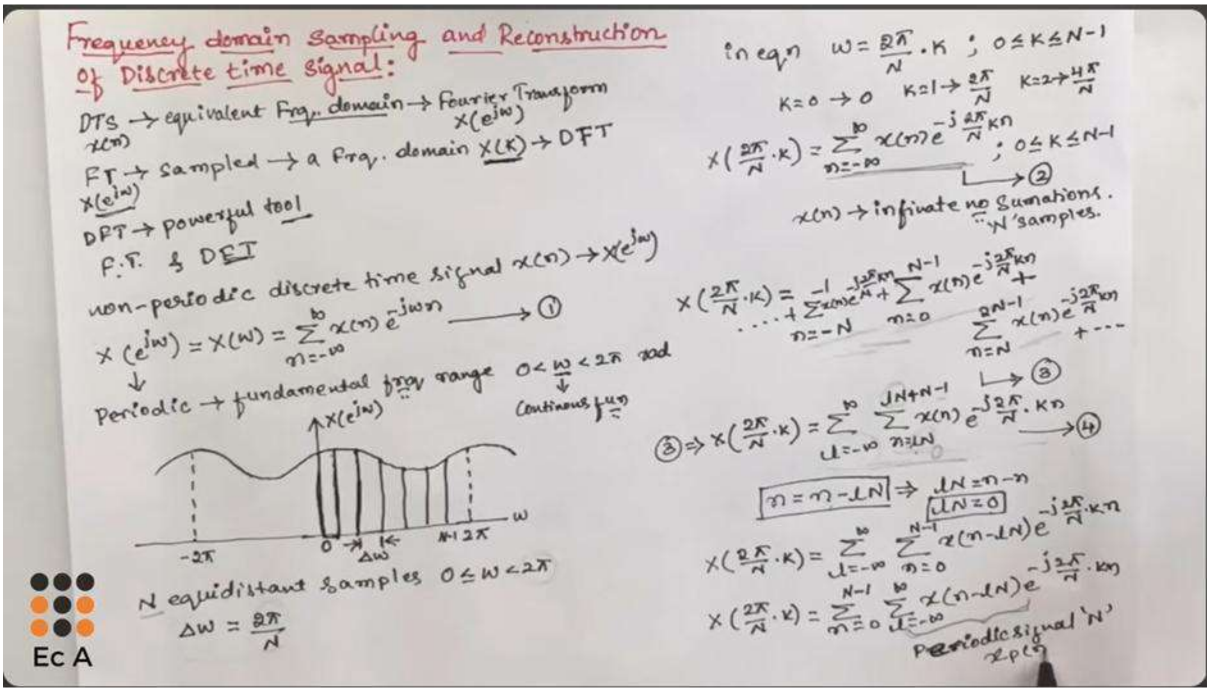
$$H\{\delta(n-k)\} = h(n-k)$$

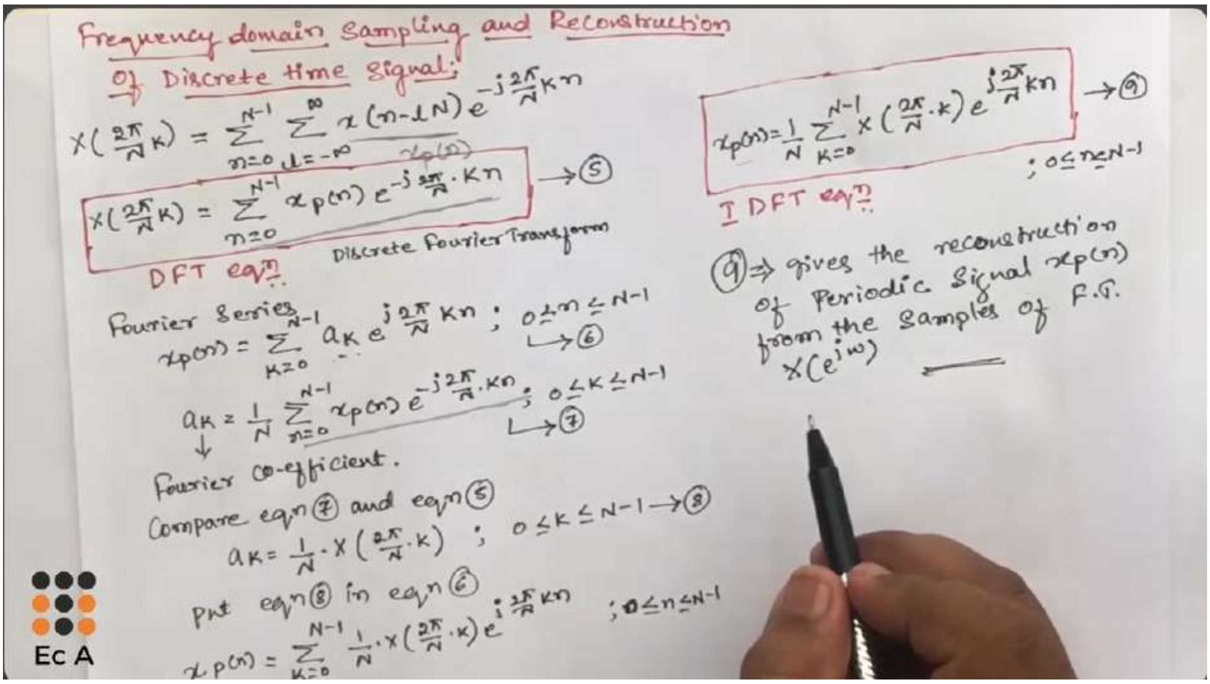
$$y(n) = \sum_{k=-\infty}^{\infty} x(k).h(n-k) \quad -⑥$$

is known as Convolution Som.

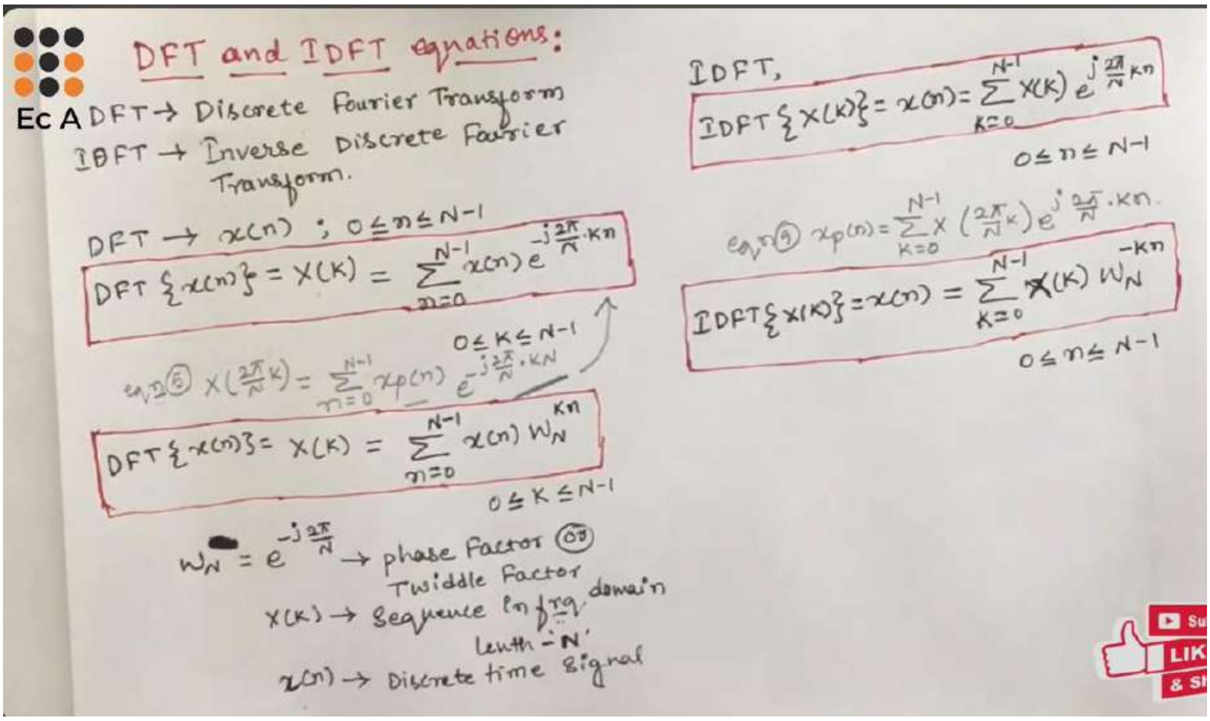
$$y(n) = x(n) * h(n) \quad -⑦$$

Frequency domain sampling and reconstruction of discrete time signals

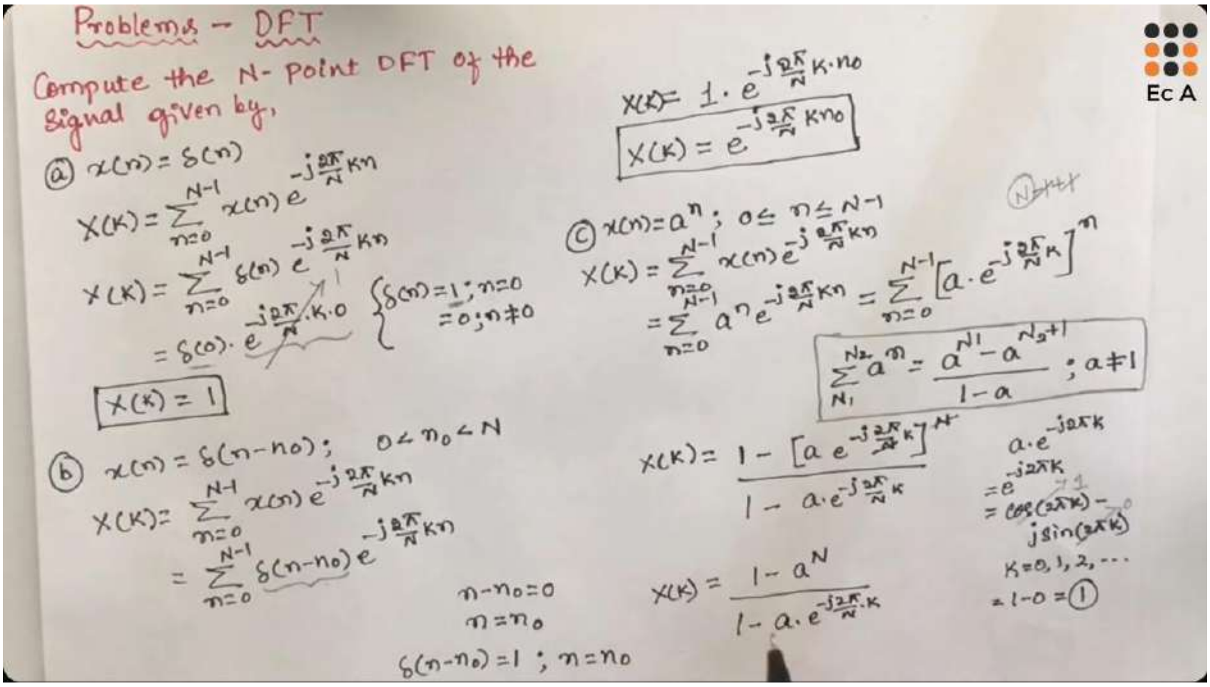




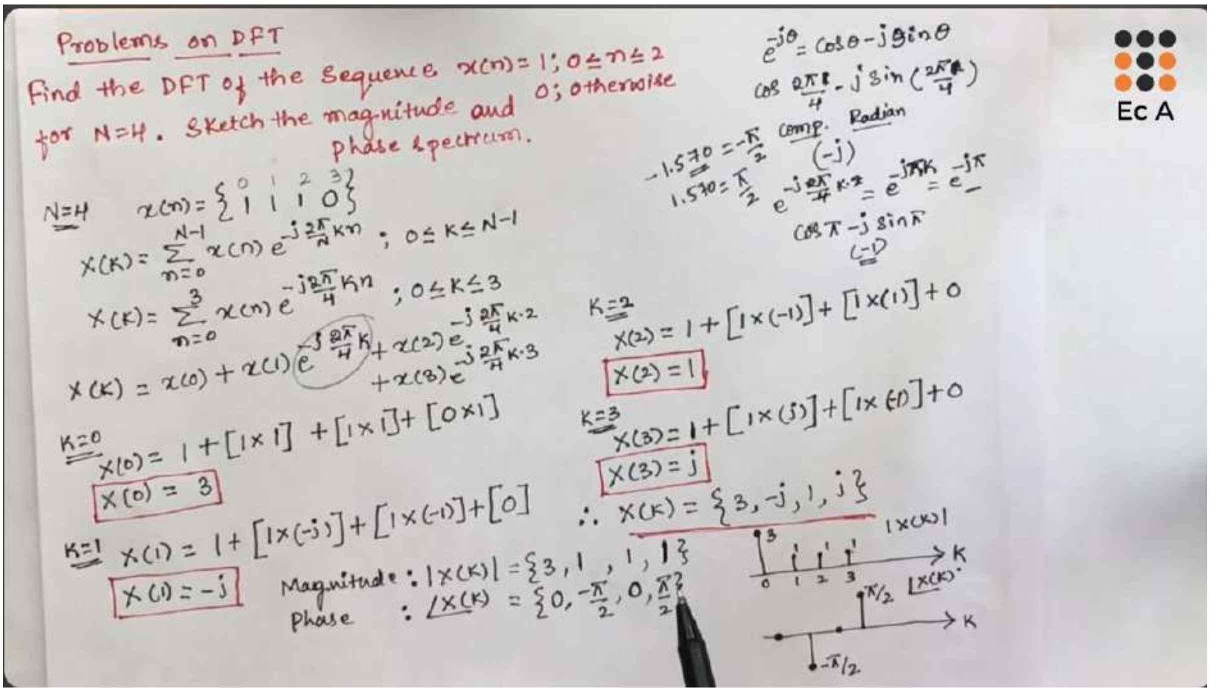
Discrete Fourier Transform (DFT) and Inverse Discrete Fourier Transform (IDFT)



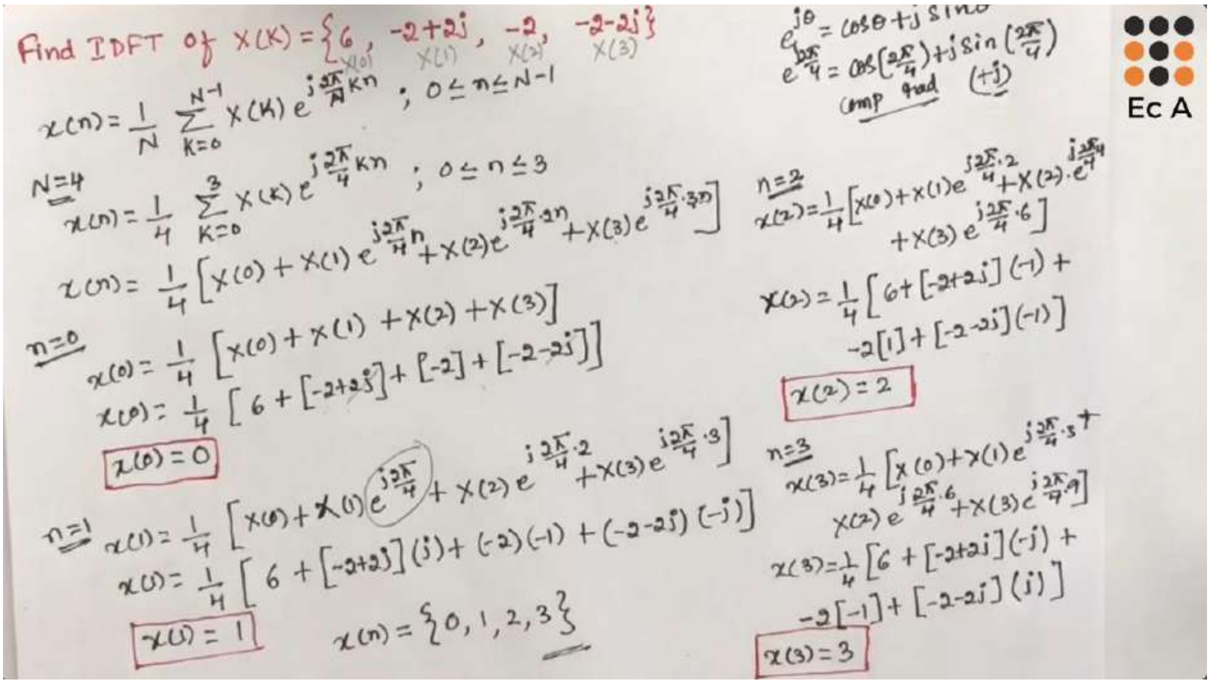
Problems on Discrete Fourier Transform (DFT)



Problem to find DFT, Magnitude and phase spectrum



Problem on Inverse Discrete Fourier Transform (IDFT)



Discrete Fourier transform as linear function (matrix form)

DFT as Linear Transformation [Matrix Method]

$$X(K) = \sum_{n=0}^{N-1} x(n)\, W_N^{Kn} \quad ; \quad 0 \le K \le N-1$$

$W_N \to$ Twiddle factor (or) phase factor

$W_N = e^{-j\frac{2\pi}{N}}$; $W_N^K \to$ periodic 'N'

Put $n = 0, 1, 2 \ldots\ldots N-1$

$$X(K) = x(0)\cdot 1 + x(1)\, W_N^{K\cdot 1} + x(2)\, W_N^{K\cdot 2} + \cdots\cdots + x(N-1)\, W_N^{K\cdot(N-1)}$$

K=0

$$X(0) = x(0) + x(1)\cdot 1 + x(2)\cdot 1 + \cdots\cdots + x(N-1)\cdot 1$$

K=1

$$X(1) = x(0) + x(1)\, W_N^{1} + x(2)\, W_N^{2} + \cdots\cdots + x(N-1)\, W_N^{1\cdot(N-1)}$$

$\vdots$

K=N-1

$$X(N-1) = x(0) + x(1)\, W_N^{(N-1)\cdot 1} + x(2)\cdot W_N^{(N-1)2} + \cdots\cdots + x(N-1)\, W_N^{(N-1)(N-1)}$$

Matrix

$$\begin{bmatrix} X(0) \\ X(1) \\ \vdots \\ X(N-1) \end{bmatrix}_{N\times 1} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & W_N & W_N^2 & \cdots & W_N^{(N-1)} \\ \vdots & \vdots & & & \vdots \\ 1 & W_N^{(N-1)} & W_N^{2(N-1)} & \cdots & W_N^{(N-1)(N-1)} \end{bmatrix}_{N\times N} \begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(N-1) \end{bmatrix}_{N\times 1}$$

$$\boxed{X_N = W_N \cdot x_N} \to ① \text{ DFT}$$

Pre multiply by W_N^{-1}

$$W_N^{-1} X_N = W_N^{-1} \cdot W_N\, x_N$$

$$W_N^{-1} X_N = x_N \Rightarrow \boxed{x_N = W_N^{-1} X_N} \to ②$$

IDFT.

$$x(n) = \frac{1}{N} \sum_{n=0}^{N-1} X(K)\, W_N^{-Kn}$$

$$x(n) = \frac{1}{N} \sum_{n=0}^{N-1} X(K)\left(W_N^{Kn}\right)^* \qquad W_N^* \to \text{complex conjugate}$$

$$\boxed{x_N = \frac{1}{N} X_N\, W_N^*} \to ③ \text{ IDFT}$$

compare ② & ③

$$W_N^{-1} = \frac{1}{N} W_N^*$$

Find 4 point DFT using matrix method or Linear Transformation method

Find 4-Point DFT of $x(n) = \{1, 0, 0, 1\}$ using Linear Transform | Matrix Method.

$X_N = W_N\, x_N$ $\quad \therefore N = 4$

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{array}{c|c} & \begin{array}{cccc} 0 & 1 & 2 & 3 \end{array} \\ \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \end{array} & \begin{bmatrix} W_4^0 & W_4^0 & W_4^0 & W_4^0 \\ W_4^0 & W_4^1 & W_4^2 & W_4^3 \\ W_4^0 & W_4^2 & W_4^4 & W_4^6 \\ W_4^0 & W_4^3 & W_4^6 & W_4^9 \end{bmatrix} \end{array} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix}$$

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Method 1

$W_N = e^{-j\frac{2\pi}{4}}$ $\qquad W_N^0 = 1 \Rightarrow W_4^0 = 1$

$W_4^1 = e^{-j\frac{2\pi}{4}\cdot 1} = \cos\left(\frac{2\pi}{4}\right) - j\sin\left(\frac{2\pi}{4}\right) = -j$

$W_4^2 = -1$

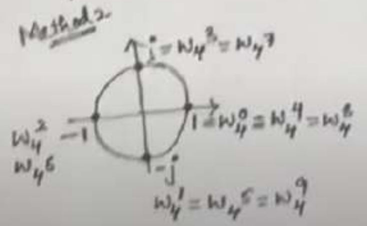
$W_4^3 = +j$

$W_4^4 = 1$

$W_4^6 = -1$

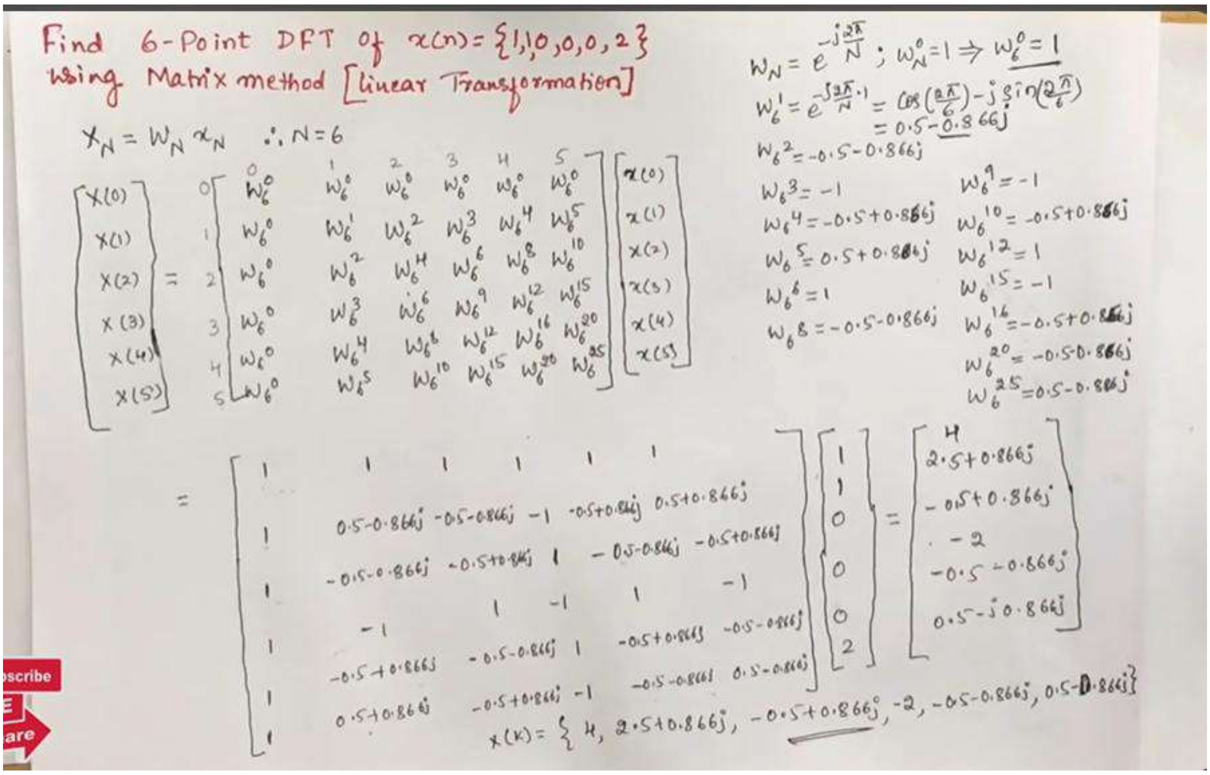
$W_4^9 = -j$

Method 2



$$X(K) = \{2,\ 1+j,\ 0,\ 1-j\}$$

find 6 point DFT using matrix method or Linear Transformation



Find 6-Point DFT of $x(n) = \{1,1,0,0,0,2\}$ using Matrix method [Linear Transformation]

$X_N = W_N x_N \quad \therefore N=6$

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \\ X(4) \\ X(5) \end{bmatrix} = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} W_6^0 & W_6^0 & W_6^0 & W_6^0 & W_6^0 & W_6^0 \\ W_6^0 & W_6^1 & W_6^2 & W_6^3 & W_6^4 & W_6^5 \\ W_6^0 & W_6^2 & W_6^4 & W_6^6 & W_6^8 & W_6^{10} \\ W_6^0 & W_6^3 & W_6^6 & W_6^9 & W_6^{12} & W_6^{15} \\ W_6^0 & W_6^4 & W_6^8 & W_6^{12} & W_6^{16} & W_6^{20} \\ W_6^0 & W_6^5 & W_6^{10} & W_6^{15} & W_6^{20} & W_6^{25} \end{bmatrix} \end{array} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \\ x(4) \\ x(5) \end{bmatrix}$$

$W_N = e^{-j\frac{2\pi}{N}}$; $W_N^0 = 1 \Rightarrow W_6^0 = 1$

$W_6^1 = e^{-j\frac{2\pi}{N}\cdot 1} = \cos\left(\frac{2\pi}{6}\right) - j\sin\left(\frac{2\pi}{6}\right) = 0.5 - 0.866j$

$W_6^2 = -0.5-0.866j$

$W_6^3 = -1$

$W_6^4 = -0.5+0.866j$

$W_6^5 = 0.5+0.866j$

$W_6^6 = 1$

$W_6^8 = -0.5-0.866j$

$W_6^9 = -1$

$W_6^{10} = -0.5+0.866j$

$W_6^{12} = 1$

$W_6^{15} = -1$

$W_6^{16} = -0.5+0.866j$

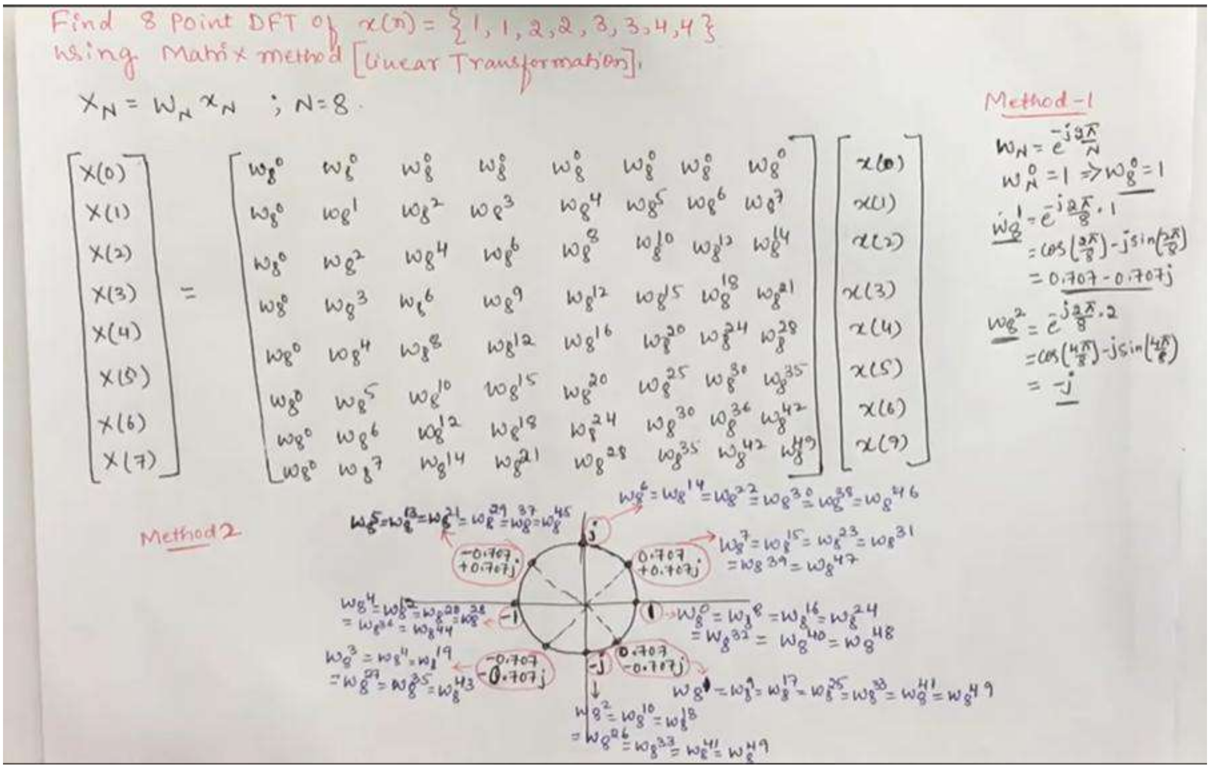
$W_6^{20} = -0.5-0.866j$

$W_6^{25} = 0.5-0.866j$

$$= \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0.5-0.866j & -0.5-0.866j & -1 & -0.5+0.866j & 0.5+0.866j \\ 1 & -0.5-0.866j & -0.5+0.866j & 1 & -0.5-0.866j & -0.5+0.866j \\ 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & -0.5+0.866j & -0.5-0.866j & 1 & -0.5+0.866j & -0.5-0.866j \\ 1 & 0.5+0.866j & -0.5+0.866j & -1 & -0.5-0.866j & 0.5-0.866j \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 2 \end{bmatrix} = \begin{bmatrix} 4 \\ 2.5+0.866j \\ -0.5+0.866j \\ -2 \\ -0.5-0.866j \\ 0.5-j0.866 \end{bmatrix}$$

$X(k) = \{4,\ 2.5+0.866j,\ -0.5+0.866j,\ -2,\ -0.5-0.866j,\ 0.5-0.866j\}$

find 8 point DFT using matrix method or Linear Transformation



Find 8 Point DFT of $x(n) = \{1,1,2,2,3,3,4,4\}$ using Matrix method [Linear Transformation].

$X_N = W_N x_N \quad ; N=8.$

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \\ X(4) \\ X(5) \\ X(6) \\ X(7) \end{bmatrix} = \begin{bmatrix} W_8^0 & W_8^0 & W_8^0 & W_8^0 & W_8^0 & W_8^0 & W_8^0 & W_8^0 \\ W_8^0 & W_8^1 & W_8^2 & W_8^3 & W_8^4 & W_8^5 & W_8^6 & W_8^7 \\ W_8^0 & W_8^2 & W_8^4 & W_8^6 & W_8^8 & W_8^{10} & W_8^{12} & W_8^{14} \\ W_8^0 & W_8^3 & W_8^6 & W_8^9 & W_8^{12} & W_8^{15} & W_8^{18} & W_8^{21} \\ W_8^0 & W_8^4 & W_8^8 & W_8^{12} & W_8^{16} & W_8^{20} & W_8^{24} & W_8^{28} \\ W_8^0 & W_8^5 & W_8^{10} & W_8^{15} & W_8^{20} & W_8^{25} & W_8^{30} & W_8^{35} \\ W_8^0 & W_8^6 & W_8^{12} & W_8^{18} & W_8^{24} & W_8^{30} & W_8^{36} & W_8^{42} \\ W_8^0 & W_8^7 & W_8^{14} & W_8^{21} & W_8^{28} & W_8^{35} & W_8^{42} & W_8^{49} \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \\ x(4) \\ x(5) \\ x(6) \\ x(7) \end{bmatrix}$$

Method-1

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

$W_N^0 = 1 \Rightarrow W_8^0 = 1$

$W_8^1 = e^{-j\frac{2\pi}{8}\cdot 1} = \cos\left(\frac{2\pi}{8}\right) - j\sin\left(\frac{2\pi}{8}\right) = 0.707 - 0.707j$

$W_8^2 = e^{-j\frac{2\pi}{8}\cdot 2} = \cos\left(\frac{4\pi}{8}\right) - j\sin\left(\frac{4\pi}{8}\right) = -j$

Method 2

$W_8^5 = W_8^{13} = W_8^{21} = W_8^{29} = W_8^{37} = W_8^{45}$

$-0.707+0.707j$

j

$W_8^6 = W_8^{14} = W_8^{22} = W_8^{30} = W_8^{38} = W_8^{46}$

$0.707+0.707j$

$W_8^7 = W_8^{15} = W_8^{23} = W_8^{31} = W_8^{39} = W_8^{47}$

$W_8^4 = W_8^{12} = W_8^{20} = W_8^{28} = W_8^{36} = W_8^{44}$

-1

1

$W_8^0 = W_8^8 = W_8^{16} = W_8^{24} = W_8^{32} = W_8^{40} = W_8^{48}$

$W_8^3 = W_8^{11} = W_8^{19} = W_8^{27} = W_8^{35} = W_8^{43}$

$-0.707-0.707j$

$-j$

$0.707-0.707j$

$W_8^1 = W_8^9 = W_8^{17} = W_8^{25} = W_8^{33} = W_8^{41} = W_8^{49}$

$W_8^2 = W_8^{10} = W_8^{18} = W_8^{26} = W_8^{33} = W_8^{41} = W_8^{49}$

$$
\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \\ X(4) \\ X(5) \\ X(6) \\ X(7) \end{bmatrix}
=
\begin{bmatrix}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 0.707-0.707j & -j & -0.707-0.707j & -1 & -0.707+0.707j & j & 0.707+0.707j \\
1 & -j & -1 & j & 1 & -j & -1 & j \\
1 & -0.707+0.707j & j & 0.707-0.707j & -1 & 0.707+0.707 & -j & -0.707-0.707j \\
1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\
1 & -0.707-0.707j & -j & 0.707+0.707j & -1 & 0.707-0.707j & j & -0.707+0.707j \\
1 & j & -1 & -j & 1 & j & -1 & j \\
1 & 0.707+0.707j & j & -0.707-0.707j & -1 & -0.707+0.707j & -j & 0.707-0.707j
\end{bmatrix}
\begin{bmatrix} 1 \\ 1 \\ 2 \\ 2 \\ 3 \\ 3 \\ 4 \\ 4 \end{bmatrix}
=
\begin{bmatrix} 20 \\ -2+4.82j \\ -2+2j \\ -2+0.828j \\ 0 \\ -2-0.828j \\ -2-2j \\ -2-4.82j \end{bmatrix}
$$

$$X(K) = \{20, -2+4.82j, -2+2j, -2+0.828j, 0, -2-0.828j, -2-2j, -2-4.82j\}$$

find Inverse Discrete Fourier transform (IDFT) using matrix method

For $X(K) = \{\underset{X(0)}{2}, \underset{X(1)}{1+j}, \underset{X(2)}{0}, \underset{X(3)}{1-j}\}$, find 4-point IDFT using Matrix method.

$$\underset{4\times1}{x_N} = \frac{1}{N} \underset{4\times4}{W_N^*} \underset{4\times1}{X_N} \qquad \therefore N=4$$

$$W_N = e^{\frac{-j2\pi}{N}}$$

$$W_N^* = e^{\frac{j2\pi}{N}} \; ; \; W_N^{*0} = 1 \Rightarrow W_4^{*0} = 1$$

$$W_4^{*1} = e^{\frac{j2\pi}{4}\cdot 1} = \cos\left(\frac{2\pi}{4}\right) + j\sin\left(\frac{2\pi}{4}\right) = j$$

Comp rad

$$W_4^{*2} = e^{\frac{j2\pi}{4}\cdot 2} = \cos(\pi) + j\sin(\pi) = -1$$

$$W_4^{*3} = -j \quad W_4^{*4} = 1 \quad W_4^{*6} = -j$$

$$W_4^{*9} = j$$

$$
\begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix}
= \frac{1}{4}
\begin{array}{c c}
 & \begin{array}{cccc} 0 & 1 & 2 & 3 \end{array} \\
\begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \end{array} &
\begin{bmatrix}
W_4^{*0} & W_4^{*0} & W_4^{*0} & W_4^{*0} \\
W_4^{*0} & W_4^{*1} & W_4^{*2} & W_4^{*3} \\
W_4^{*0} & W_4^{*2} & W_4^{*4} & W_4^{*6} \\
W_4^{*0} & W_4^{*3} & W_4^{*6} & W_4^{*9}
\end{bmatrix}
\end{array}
\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix}
$$

$$
= \frac{1}{4}
\begin{bmatrix}
1 & 1 & 1 & 1 \\
1 & j & -1 & -j \\
1 & -1 & 1 & -1 \\
1 & -j & -1 & j
\end{bmatrix}
\begin{bmatrix} 2 \\ 1+j \\ 0 \\ 1-j \end{bmatrix}
= \frac{1}{4}
\begin{bmatrix} 4 \\ 0 \\ 0 \\ 4 \end{bmatrix}
=
\begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}
$$

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

Assignment:

① 6-Point IDFT:

$X(K) = \{4, 2.5+0.866j, -0.5+0.866j, -2, -0.5-0.866j, 2.5-0.866j\}$

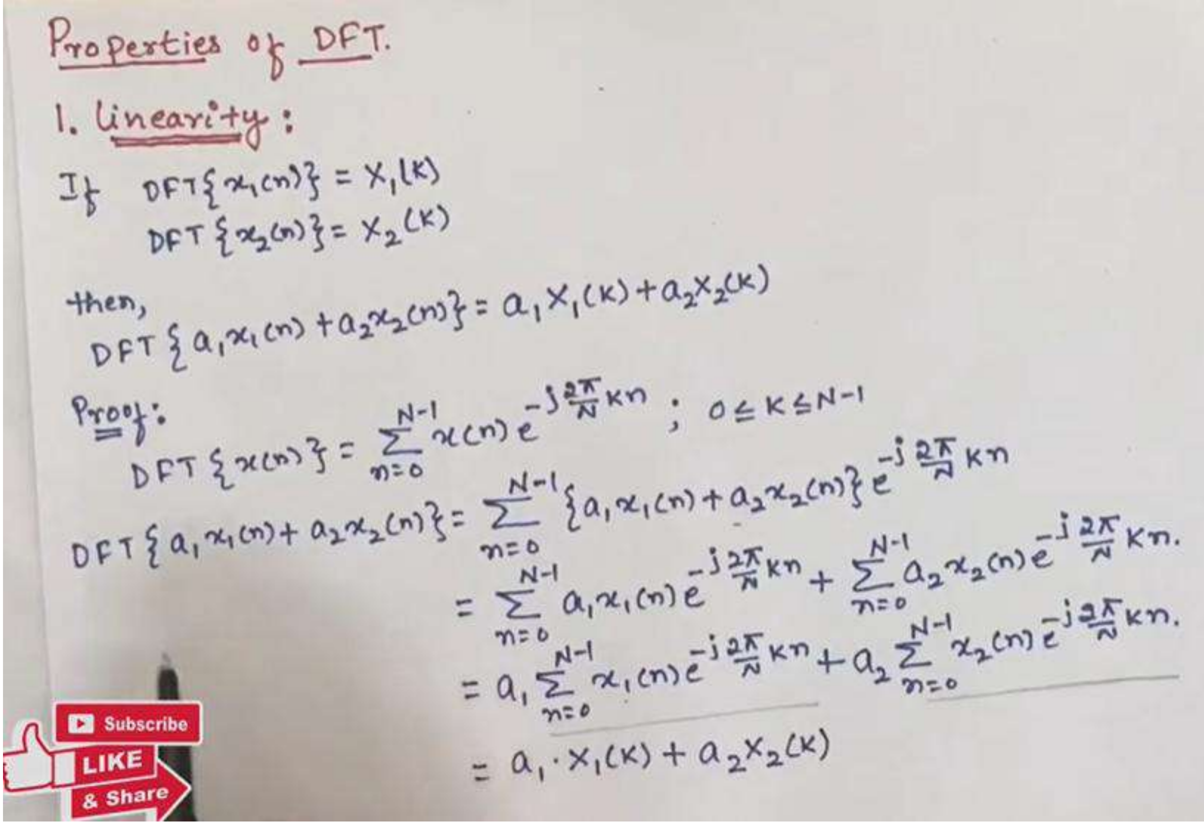
$\boxed{x(n) = \{1, 1, 0, 0, 0, 2\}}$

② 8-Point IDFT:

$X(K) = \{20, -2+4.828j, -2+2j, -2+0.828j, 0, -2-0.828j, -2-2j, -2-4.828j\} \rightarrow \boxed{x(n) = \{1, 1, 2, 2, 3, 3, 4, 4\}}$

Properties of DFT

Linearity property of DFT



Properties of DFT.

1. Linearity:

If $\text{DFT}\{x_1(n)\} = X_1(K)$

$\text{DFT}\{x_2(n)\} = X_2(K)$

then,

$$\text{DFT}\{a_1x_1(n) + a_2x_2(n)\} = a_1X_1(K) + a_2X_2(K)$$

Proof:

$$\text{DFT}\{x(n)\} = \sum_{n=0}^{N-1} x(n)\, e^{-j\frac{2\pi}{N}kn} \; ; \quad 0 \le K \le N-1$$

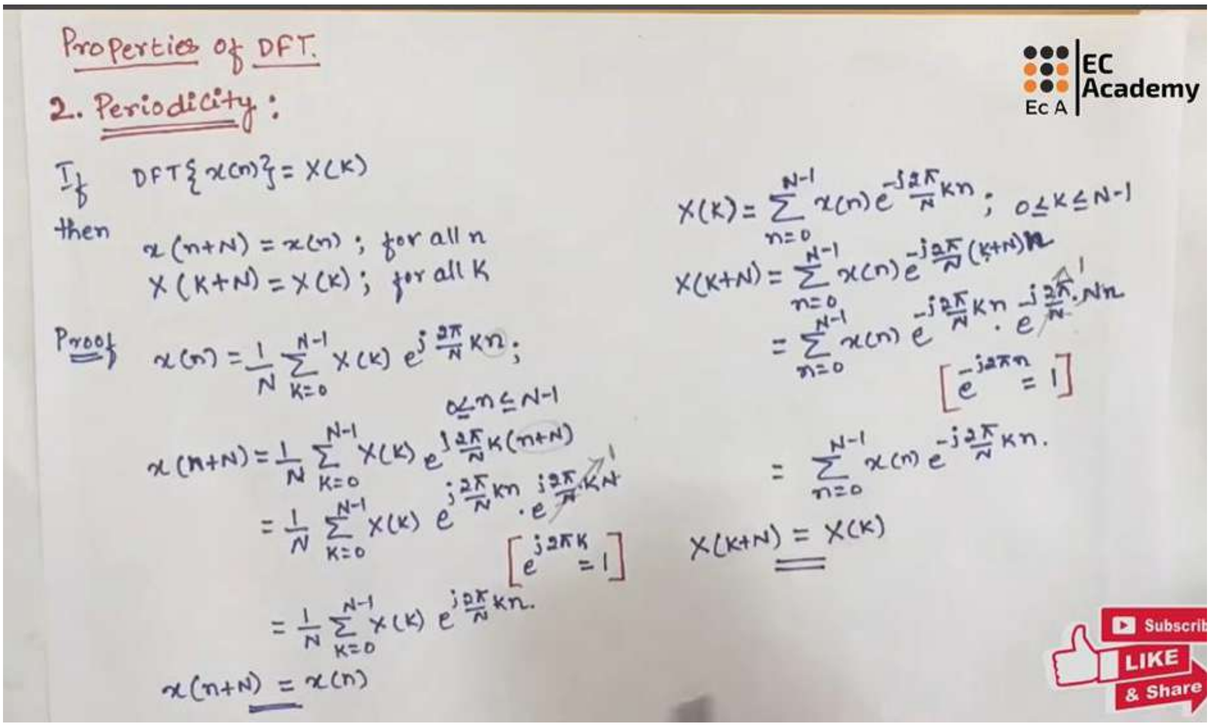
$$\text{DFT}\{a_1x_1(n) + a_2x_2(n)\} = \sum_{n=0}^{N-1} \{a_1x_1(n) + a_2x_2(n)\}\, e^{-j\frac{2\pi}{N}kn}$$

$$= \sum_{n=0}^{N-1} a_1x_1(n)\, e^{-j\frac{2\pi}{N}kn} + \sum_{n=0}^{N-1} a_2x_2(n)\, e^{-j\frac{2\pi}{N}kn}.$$

$$= a_1 \sum_{n=0}^{N-1} x_1(n)\, e^{-j\frac{2\pi}{N}kn} + a_2 \sum_{n=0}^{N-1} x_2(n)\, e^{-j\frac{2\pi}{N}kn}.$$

$$= a_1 \cdot X_1(K) + a_2X_2(K)$$

Periodicity property of DFT



Properties of DFT.

2. Periodicity:

If $\text{DFT}\{x(n)\} = X(K)$

then $x(n+N) = x(n)$; for all n

$X(K+N) = X(K)$; for all K

Proof

$$x(n) = \frac{1}{N} \sum_{K=0}^{N-1} X(K)\, e^{j\frac{2\pi}{N}kn} ; \quad 0 \le n \le N-1$$

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

$$= \frac{1}{N} \sum_{K=0}^{N-1} X(K)\, e^{j\frac{2\pi}{N}kn} \cdot \overbrace{e^{j\frac{2\pi}{N}KN}}^{1}$$

$$\left[e^{j2\pi K} = 1\right]$$

$$= \frac{1}{N} \sum_{K=0}^{N-1} X(K)\, e^{j\frac{2\pi}{N}kn}.$$

$$x(n+N) = x(n)$$

$$X(K) = \sum_{n=0}^{N-1} x(n)\, e^{-j\frac{2\pi}{N}kn} ; \quad 0 \le K \le N-1$$

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

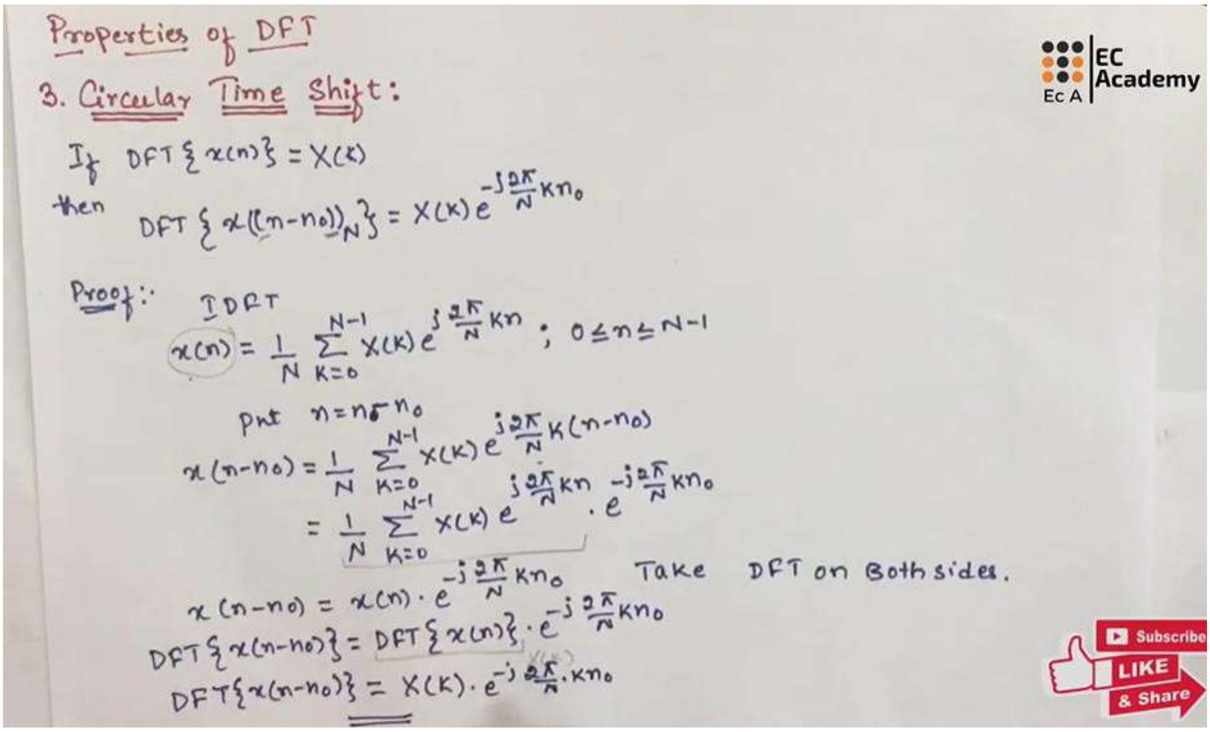
$$= \sum_{n=0}^{N-1} x(n)\, e^{-j\frac{2\pi}{N}kn} \cdot \overbrace{e^{-j\frac{2\pi}{N}Nn}}^{1}$$

$$\left[e^{-j2\pi n} = 1\right]$$

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

$$X(K+N) = X(K)$$

Circular Time shift property



Properties of DFT

3. Circular Time Shift:

If $DFT\{x(n)\} = X(k)$

then $DFT\{x((n-n_0))_N\} = X(k)\, e^{-j\frac{2\pi}{N}kn_0}$

Proof: IDFT

$$x(n) = \frac{1}{N}\sum_{K=0}^{N-1} X(K)\, e^{j\frac{2\pi}{N}kn} \; ; \; 0 \le n \le N-1$$

put $n = n - n_0$

$$x(n-n_0) = \frac{1}{N}\sum_{K=0}^{N-1} X(K)\, e^{j\frac{2\pi}{N}K(n-n_0)}$$

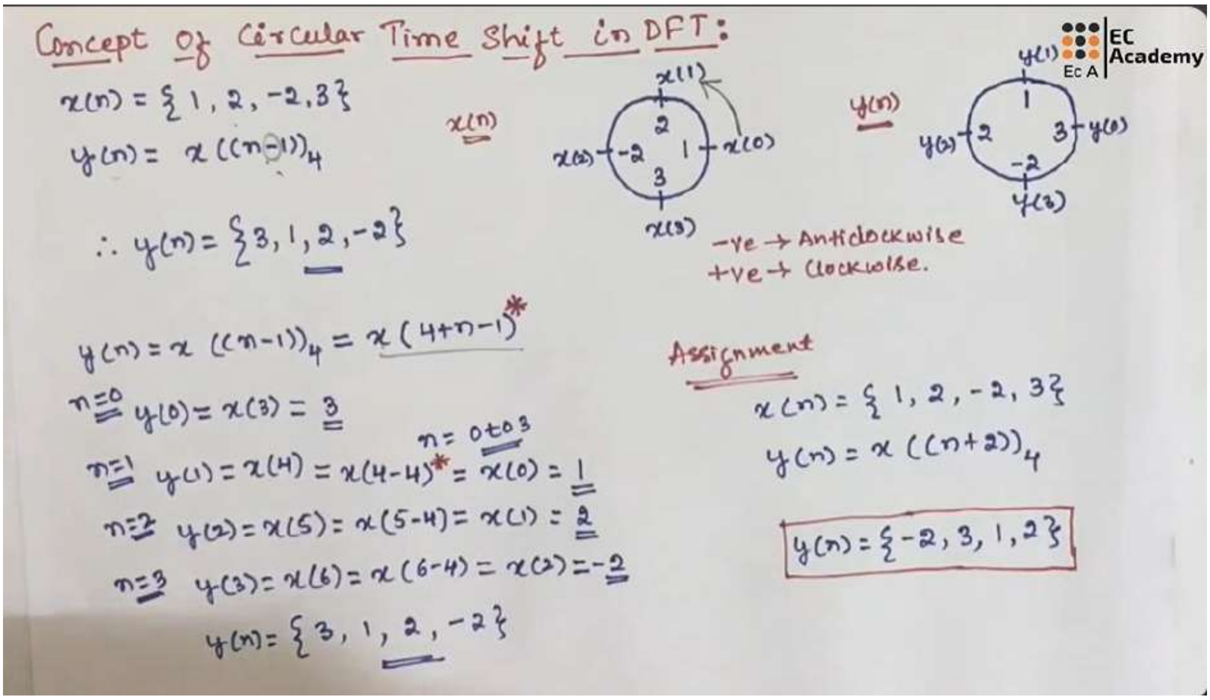
$$= \underbrace{\frac{1}{N}\sum_{K=0}^{N-1} X(K)\, e^{j\frac{2\pi}{N}kn}}\cdot e^{-j\frac{2\pi}{N}kn_0}$$

$x(n-n_0) = x(n)\cdot e^{-j\frac{2\pi}{N}kn_0}$ Take DFT on Both sides.

$DFT\{x(n-n_0)\} = DFT\{x(n)\}\cdot e^{-j\frac{2\pi}{N}kn_0}$

$DFT\{x(n-n_0)\} = X(K)\cdot e^{-j\frac{2\pi}{N}\cdot kn_0}$

concept of circular Time shift



Concept of Circular Time Shift in DFT:

$x(n) = \{1, 2, -2, 3\}$

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

$\therefore y(n) = \{3, 1, 2, -2\}$

-ve → Anticlockwise

+ve → Clockwise.

$y(n) = x((n-1))_4 = x(4+n-1)^*$

$n=0$ $y(0) = x(3) = 3$

$n = 0 \text{ to } 3$

$n=1$ $y(1) = x(4) = x(4-4)^* = x(0) = 1$

$n=2$ $y(2) = x(5) = x(5-4) = x(1) = 2$

$n=3$ $y(3) = x(6) = x(6-4) = x(2) = -2$

$y(n) = \{3, 1, 2, -2\}$

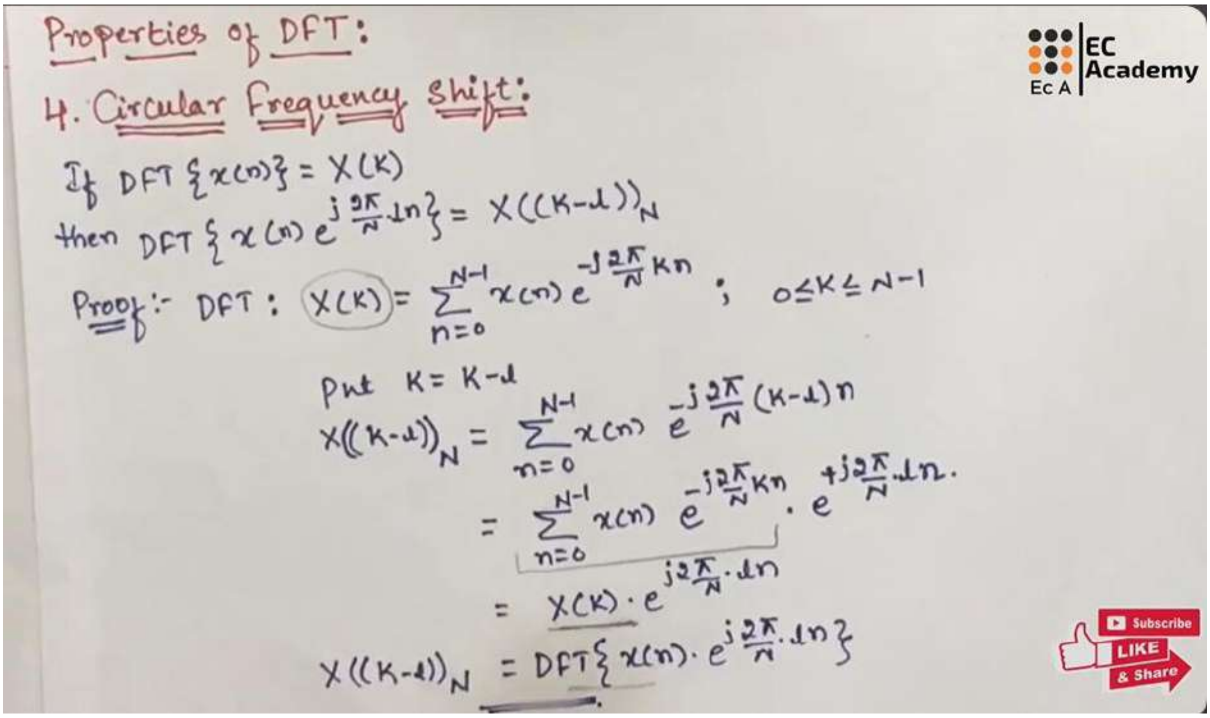
Assignment

$x(n) = \{1, 2, -2, 3\}$

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

$\boxed{y(n) = \{-2, 3, 1, 2\}}$

Circular frequency shift property


Properties of DFT:

4. Circular Frequency Shift:

If $DFT\{x(n)\} = X(K)$

then $DFT\{x(n)\, e^{j\frac{2\pi}{N} l n}\} = X((K-l))_N$

Proof:- DFT : $X(K) = \sum_{n=0}^{N-1} x(n)\, e^{-j\frac{2\pi}{N}Kn}$; $0 \le K \le N-1$

Put $K = K-l$

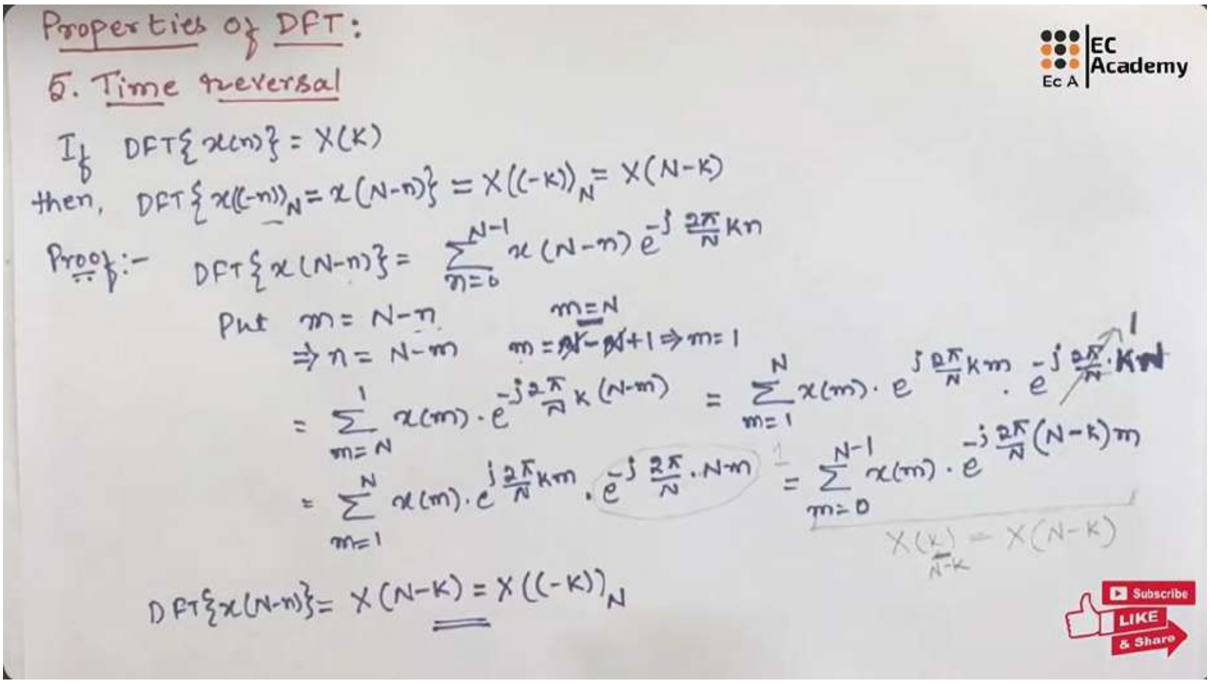
$$X((K-l))_N = \sum_{n=0}^{N-1} x(n)\, e^{-j\frac{2\pi}{N}(K-l)n}$$

$$= \underbrace{\sum_{n=0}^{N-1} x(n)\, e^{-j\frac{2\pi}{N}Kn}} \cdot e^{+j\frac{2\pi}{N} l n}.$$

$$= X(K) \cdot e^{j\frac{2\pi}{N} l n}$$

$$X((K-l))_N = DFT\{x(n) \cdot e^{j\frac{2\pi}{N} l n}\}$$

Time reversal property of DFT


Properties of DFT:

5. Time reversal

If $DFT\{x(n)\} = X(K)$

then, $DFT\{x((-n))_N = x(N-n)\} = X((-K))_N = X(N-K)$

Proof:- $DFT\{x(N-n)\} = \sum_{n=0}^{N-1} x(N-n)\, e^{-j\frac{2\pi}{N}Kn}$

Put $m = N-n$ $\Rightarrow n = N-m$ $\quad m = N$ $\quad m = \cancel{N} - \cancel{N} + 1 \Rightarrow m = 1$

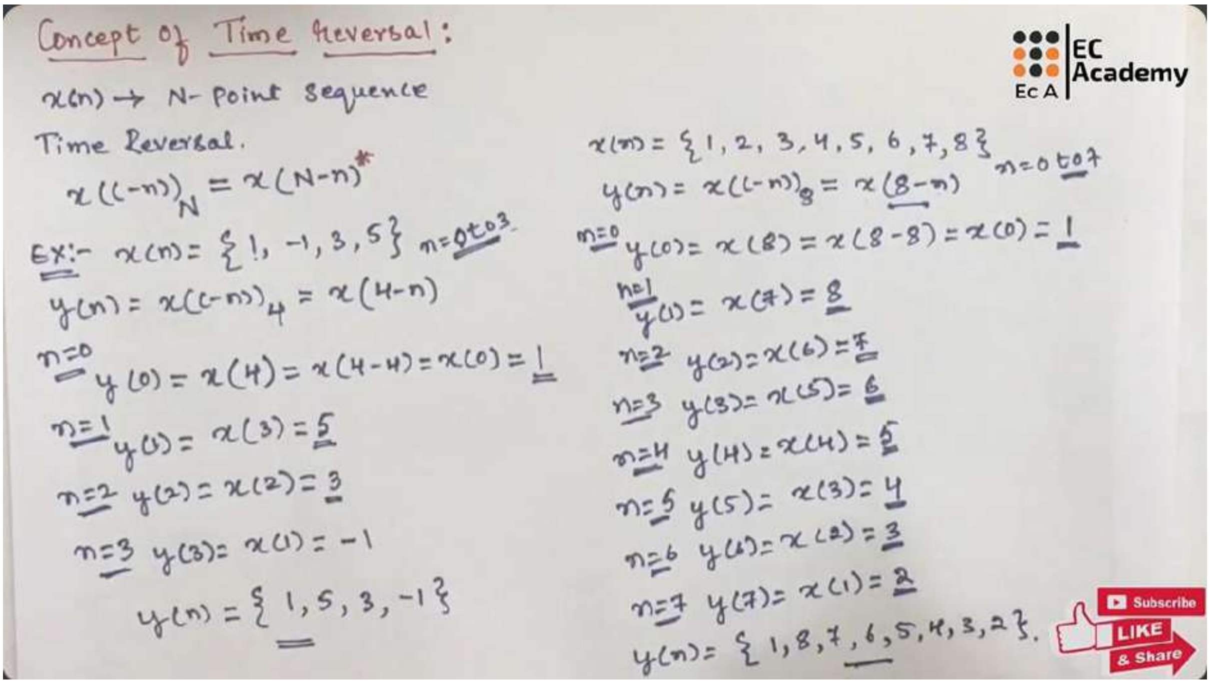
$$= \sum_{m=N}^{1} x(m) \cdot e^{-j\frac{2\pi}{N}K(N-m)} = \sum_{m=1}^{N} x(m) \cdot e^{j\frac{2\pi}{N}Km} \cdot e^{-j\frac{2\pi}{N} \cdot KN}$$

$$= \sum_{m=1}^{N} x(m) \cdot e^{j\frac{2\pi}{N}Km} \cdot e^{-j\frac{2\pi}{N} \cdot Nm} = \sum_{m=0}^{N-1} x(m) \cdot e^{-j\frac{2\pi}{N}(N-K)m}$$

$$X(K)\big|_{N-K} = X(N-K)$$

$$DFT\{x(N-n)\} = X(N-K) = X((-K))_N$$

Concept of Time reversal



Concept of Time Reversal:

$x(n) \rightarrow$ N-Point sequence

Time Reversal.

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

Ex:- $x(n) = \{1, -1, 3, 5\}$ $n = 0$ to 3

$y(n) = x((-n))_4 = x(4-n)$

$n=0$ $y(0) = x(4) = x(4-4) = x(0) = 1$

$n=1$ $y(1) = x(3) = 5$

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

$n=3$ $y(3) = x(1) = -1$

$y(n) = \{1, 5, 3, -1\}$

$x(n) = \{1, 2, 3, 4, 5, 6, 7, 8\}$ $n = 0$ to 7

$y(n) = x((-n))_8 = x(8-n)$

$n=0$ $y(0) = x(8) = x(8-8) = x(0) = 1$

$n=1$ $y(1) = x(7) = 8$

$n=2$ $y(2) = x(6) = 7$

$n=3$ $y(3) = x(5) = 6$

$n=4$ $y(4) = x(4) = 5$

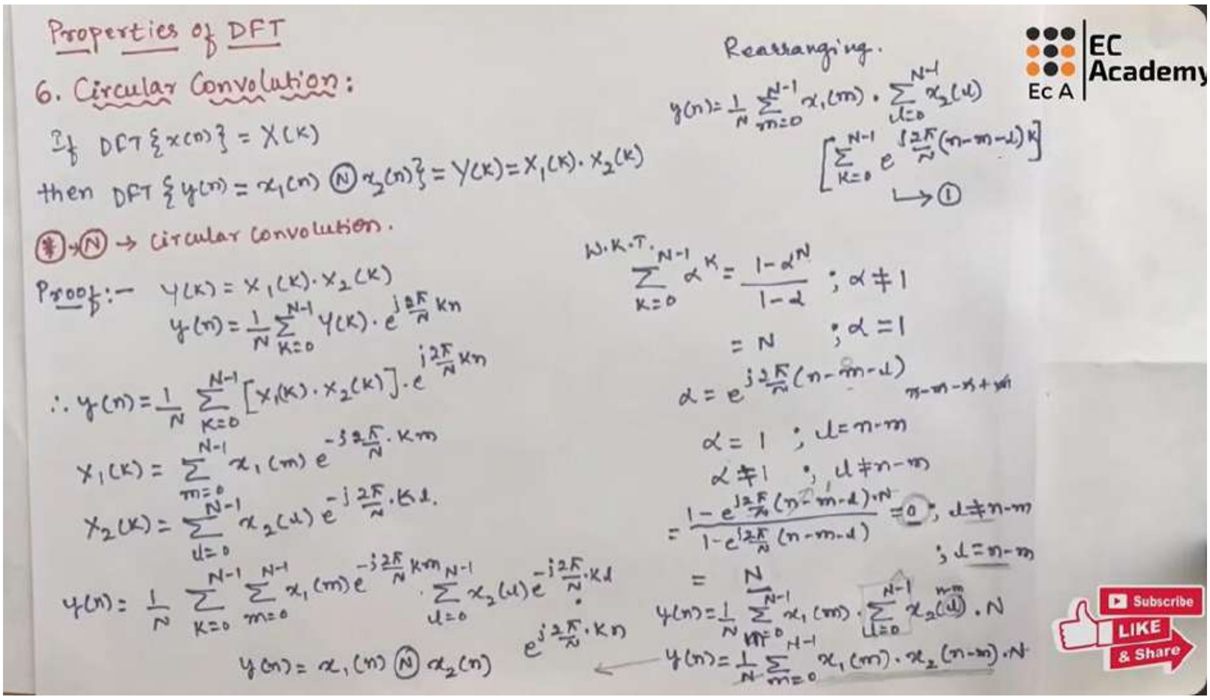
$n=5$ $y(5) = x(3) = 4$

$n=6$ $y(6) = x(2) = 3$

$n=7$ $y(7) = x(1) = 2$

$y(n) = \{1, 8, 7, 6, 5, 4, 3, 2\}$.

Circular convolution property of DFT



Properties of DFT

6. Circular Convolution:

If $DFT\{x(n)\} = X(K)$

then $DFT\{y(n) = x_1(n) \text{ Ⓝ } x_2(n)\} = Y(K) = X_1(K) \cdot X_2(K)$

Ⓝ → circular convolution.

Proof:- $Y(K) = X_1(K) \cdot X_2(K)$

$$y(n) = \frac{1}{N}\sum_{K=0}^{N-1} Y(K) \cdot e^{j\frac{2\pi}{N}Kn}$$

$$\therefore y(n) = \frac{1}{N}\sum_{K=0}^{N-1} [X_1(K) \cdot X_2(K)] \cdot e^{j\frac{2\pi}{N}Kn}$$

$$X_1(K) = \sum_{m=0}^{N-1} x_1(m) e^{-j\frac{2\pi}{N} \cdot Km}$$

$$X_2(K) = \sum_{l=0}^{N-1} x_2(l) e^{-j\frac{2\pi}{N} \cdot Kl}$$

$$y(n) = \frac{1}{N}\sum_{K=0}^{N-1} \sum_{m=0}^{N-1} x_1(m) e^{-j\frac{2\pi}{N}Km} \cdot \sum_{l=0}^{N-1} x_2(l) e^{-j\frac{2\pi}{N} \cdot Kl} \, e^{j\frac{2\pi}{N} \cdot Kn}$$

$$y(n) = x_1(n) \text{ Ⓝ } x_2(n)$$

Rearranging.

$$y(n) = \frac{1}{N}\sum_{m=0}^{N-1} x_1(m) \cdot \sum_{l=0}^{N-1} x_2(l) \left[\sum_{K=0}^{N-1} e^{j\frac{2\pi}{N}(n-m-l)K}\right] \rightarrow ①$$

W.K.T.

$$\sum_{K=0}^{N-1} \alpha^K = \frac{1-\alpha^N}{1-\alpha} \; ; \alpha \neq 1$$

$$= N \; ; \alpha = 1$$

$$\alpha = e^{j\frac{2\pi}{N}(n-m-l)} \qquad n-m-n+m$$

$\alpha = 1$; $l = n-m$

$\alpha \neq 1$; $l \neq n-m$

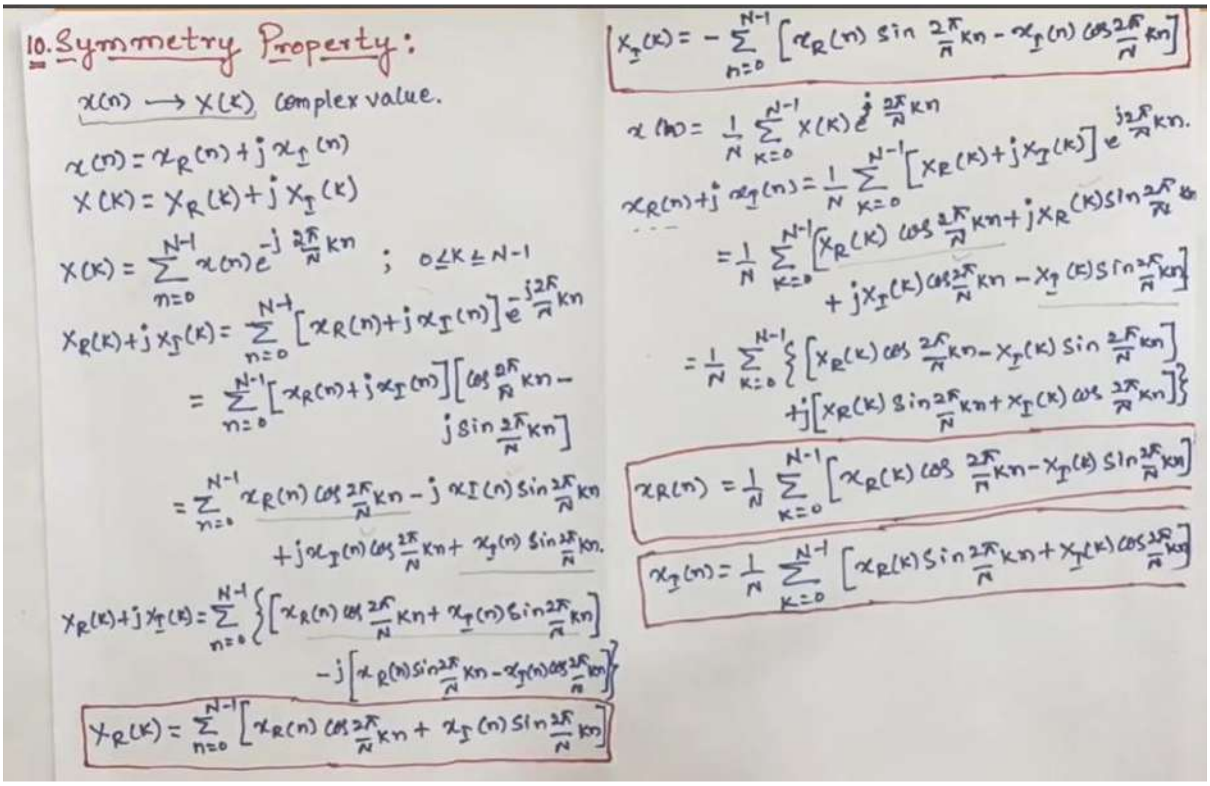
$$= \frac{1 - e^{j\frac{2\pi}{N}(n-m-l) \cdot N}}{1 - e^{j\frac{2\pi}{N}(n-m-l)}} = 0 \; ; l \neq n-m$$

$$= N \; ; l = n-m$$

$$y(n) = \frac{1}{N}\sum_{m=0}^{N-1} x_1(m) \cdot \sum_{l=0}^{N-1} x_2(\overset{n-m}{l}) \cdot N$$

$$y(n) = \frac{1}{N}\sum_{m=0}^{N-1} x_1(m) \cdot x_2(n-m) \cdot N$$

Symmetry property of dft



10. Symmetry Property:

$x(n) \longrightarrow X(K)$ complex value.

$$x(n) = x_R(n) + j\,x_I(n)$$

$$X(K) = X_R(K) + j\,X_I(K)$$

$$X(K) = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi}{N}kn} \quad ; \quad 0 \le K \le N-1$$

$$X_R(K) + jX_I(K) = \sum_{n=0}^{N-1} [x_R(n) + jx_I(n)] e^{-j\frac{2\pi}{N}kn}$$

$$= \sum_{n=0}^{N-1} [x_R(n) + jx_I(n)]\left[\cos\frac{2\pi}{N}kn - j\sin\frac{2\pi}{N}kn\right]$$

$$= \sum_{n=0}^{N-1} x_R(n)\cos\frac{2\pi}{N}kn - j\,x_R(n)\sin\frac{2\pi}{N}kn + j\,x_I(n)\cos\frac{2\pi}{N}kn + x_I(n)\sin\frac{2\pi}{N}kn.$$

$$X_R(K) + jX_I(K) = \sum_{n=0}^{N-1} \left\{\left[x_R(n)\cos\frac{2\pi}{N}kn + x_I(n)\sin\frac{2\pi}{N}kn\right] - j\left[x_R(n)\sin\frac{2\pi}{N}kn - x_I(n)\cos\frac{2\pi}{N}kn\right]\right\}$$

$$\boxed{X_R(K) = \sum_{n=0}^{N-1}\left[x_R(n)\cos\frac{2\pi}{N}kn + x_I(n)\sin\frac{2\pi}{N}kn\right]}$$

$$\boxed{X_I(K) = -\sum_{n=0}^{N-1}\left[x_R(n)\sin\frac{2\pi}{N}kn - x_I(n)\cos\frac{2\pi}{N}kn\right]}$$

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

$$x_R(n) + j\,x_I(n) = \frac{1}{N}\sum_{K=0}^{N-1} [X_R(K) + jX_I(K)] e^{j\frac{2\pi}{N}kn}.$$

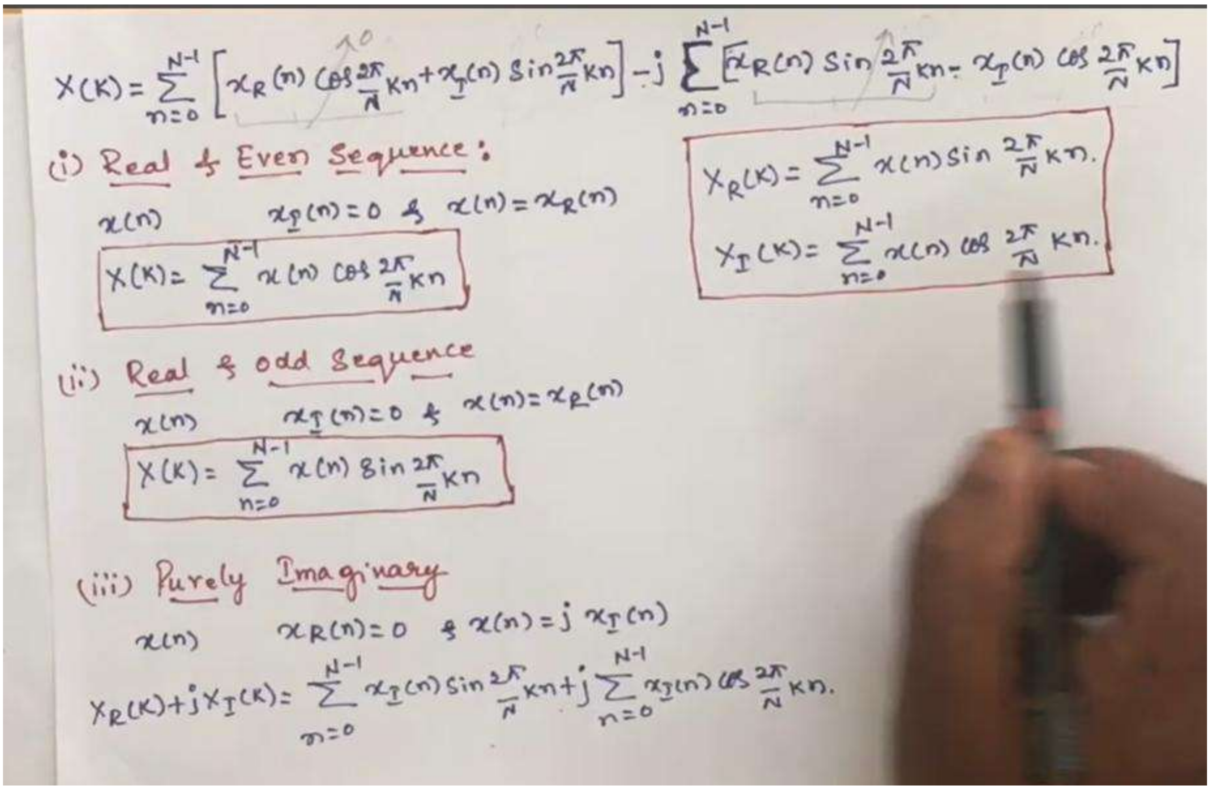
$$= \frac{1}{N}\sum_{K=0}^{N-1}\left[X_R(K)\cos\frac{2\pi}{N}kn + jX_R(K)\sin\frac{2\pi}{N}kn + jX_I(K)\cos\frac{2\pi}{N}kn - X_I(K)\sin\frac{2\pi}{N}kn\right]$$

$$= \frac{1}{N}\sum_{K=0}^{N-1}\left\{\left[X_R(K)\cos\frac{2\pi}{N}kn - X_I(K)\sin\frac{2\pi}{N}kn\right] + j\left[X_R(K)\sin\frac{2\pi}{N}kn + X_I(K)\cos\frac{2\pi}{N}kn\right]\right\}$$

$$\boxed{x_R(n) = \frac{1}{N}\sum_{K=0}^{N-1}\left[X_R(K)\cos\frac{2\pi}{N}kn - X_I(K)\sin\frac{2\pi}{N}kn\right]}$$

$$\boxed{x_I(n) = \frac{1}{N}\sum_{K=0}^{N-1}\left[X_R(K)\sin\frac{2\pi}{N}kn + X_I(K)\cos\frac{2\pi}{N}kn\right]}$$

Hence for N point sequence x(n) which is real and even it’s n point DFT X(K) is also real and even.



$$X(K) = \sum_{n=0}^{N-1}\left[x_R(n)\cos\frac{2\pi}{N}kn + x_I(n)\sin\frac{2\pi}{N}kn\right] - j\sum_{n=0}^{N-1}\left[x_R(n)\sin\frac{2\pi}{N}kn - x_I(n)\cos\frac{2\pi}{N}kn\right]$$

(i) Real & Even Sequence:

$x(n)$ $\quad x_I(n) = 0$ & $x(n) = x_R(n)$

$$\boxed{X(K) = \sum_{n=0}^{N-1} x(n)\cos\frac{2\pi}{N}kn}$$

(ii) Real & odd Sequence

$x(n)$ $\quad x_I(n) = 0$ & $x(n) = x_R(n)$

$$\boxed{X(K) = \sum_{n=0}^{N-1} x(n)\sin\frac{2\pi}{N}kn}$$

(iii) Purely Imaginary

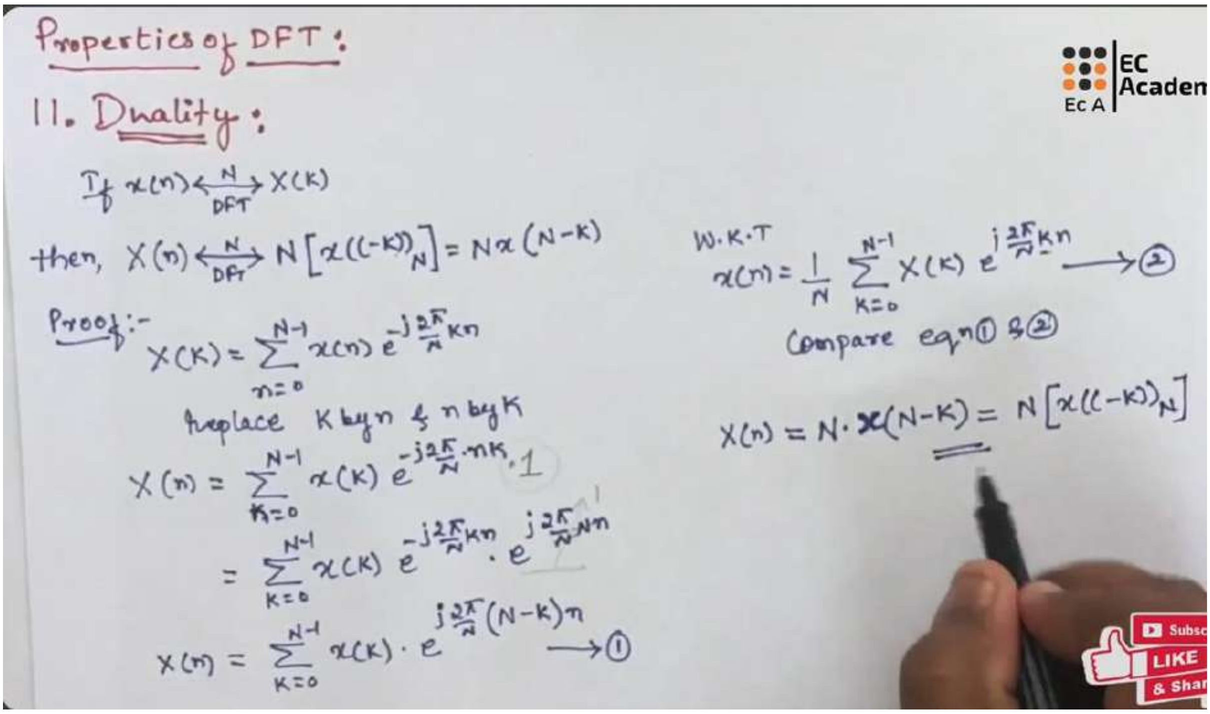
$x(n)$ $\quad x_R(n) = 0$ & $x(n) = j\,x_I(n)$

$$X_R(K) + jX_I(K) = \sum_{n=0}^{N-1} x_I(n)\sin\frac{2\pi}{N}kn + j\sum_{n=0}^{N-1} x_I(n)\cos\frac{2\pi}{N}kn.$$

$$\boxed{\begin{aligned} X_R(K) &= \sum_{n=0}^{N-1} x(n)\sin\frac{2\pi}{N}kn. \\ X_I(K) &= \sum_{n=0}^{N-1} x(n)\cos\frac{2\pi}{N}kn. \end{aligned}}$$

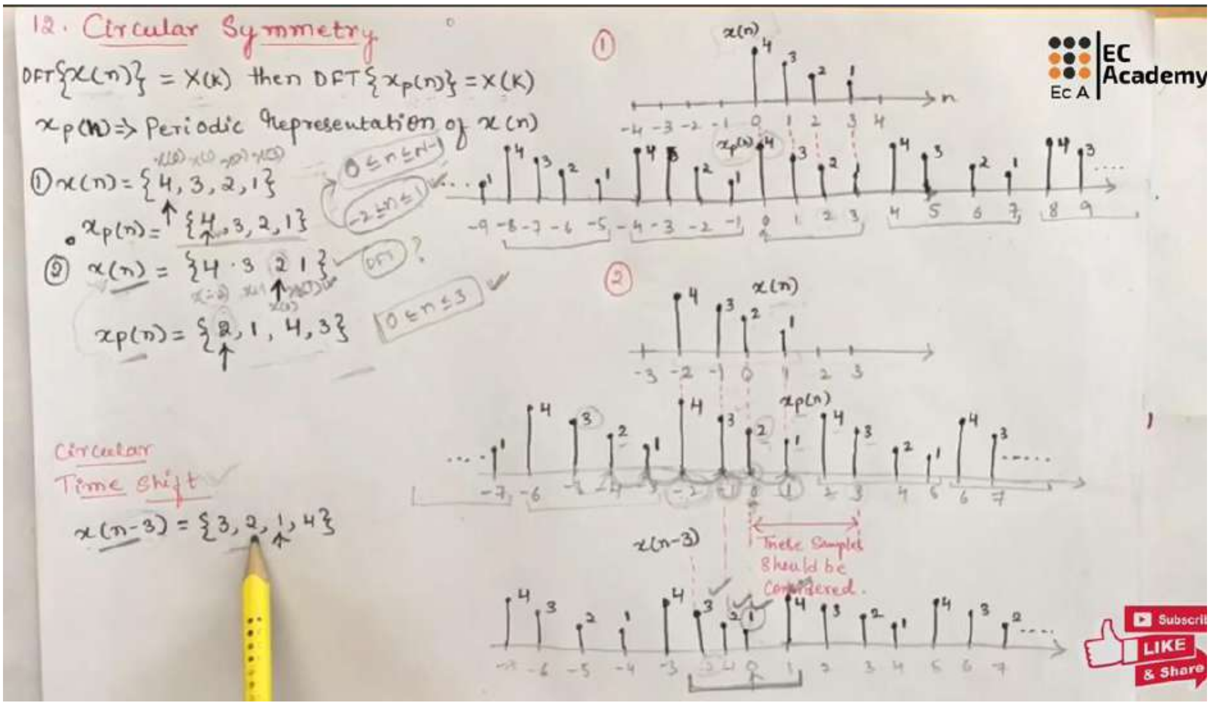
The N point sequence x(n) which is real and odd, its DFT X(k) is pure and imaginary.
For N point sequence x(n) the DFT Xr(K) is odd and DFT Xi(K) is even.

Duality property of DFT

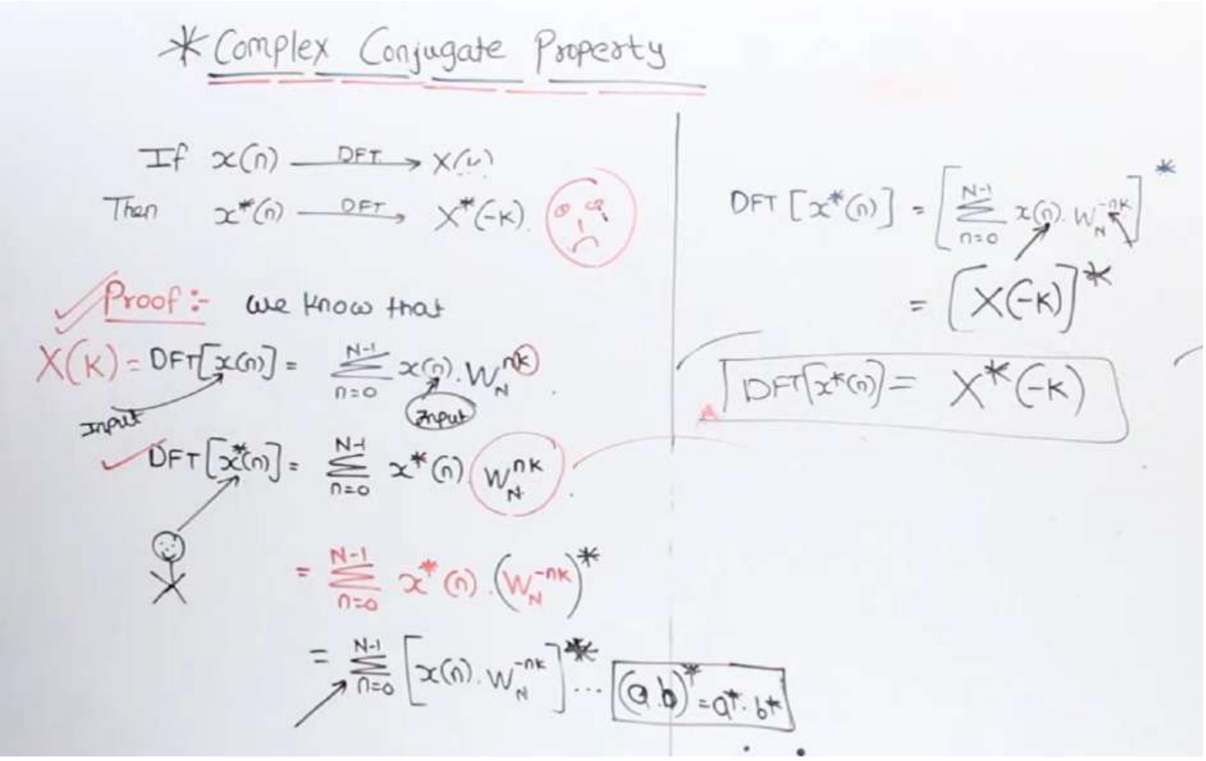


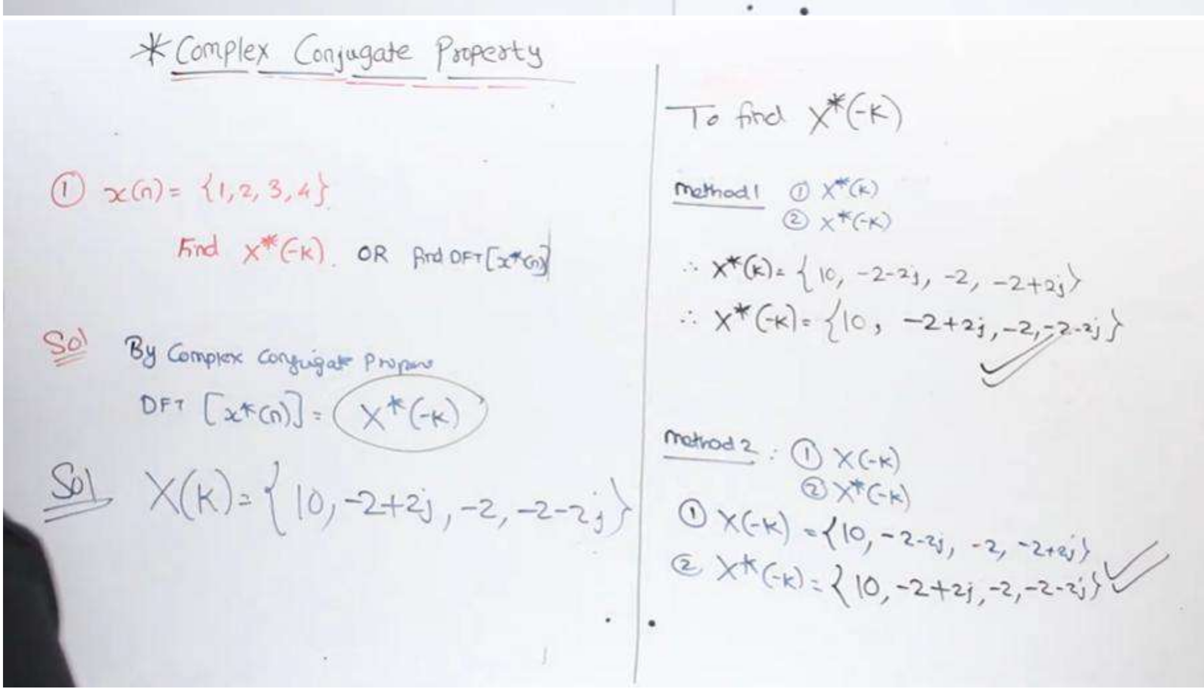
Eqn 1 is N times Eqn 2 that's why we write Nx(N-K)

Circular symmetry property of DFT

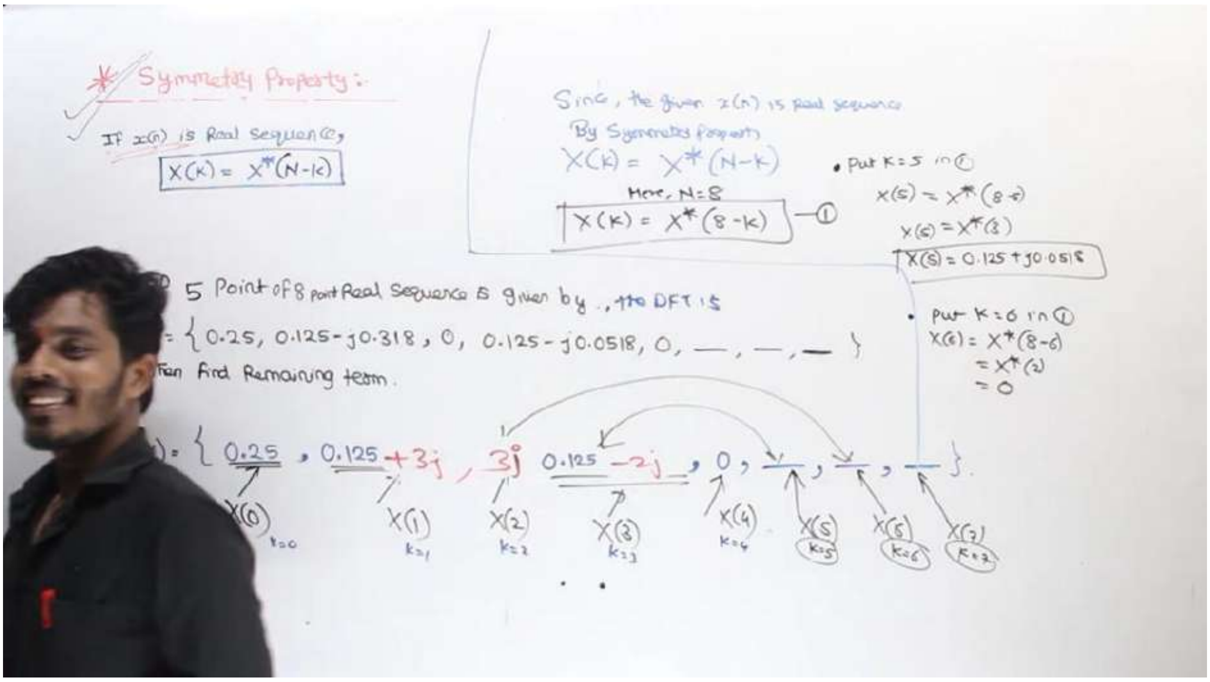


Proof of Complex Conjugate Property

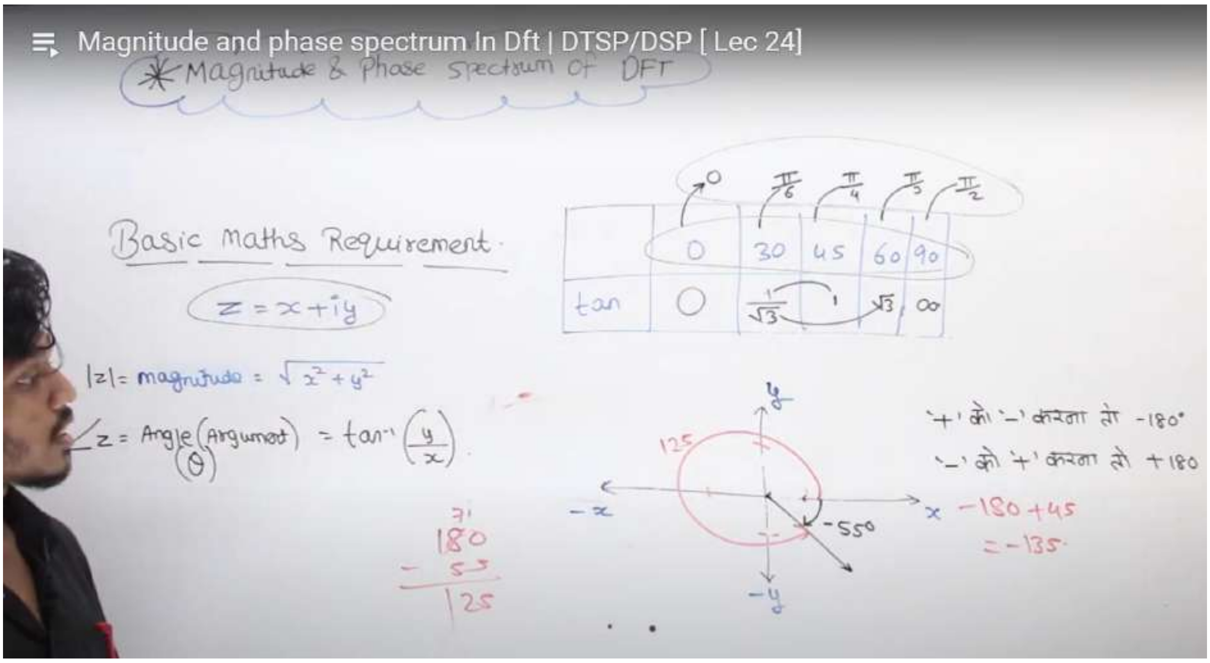


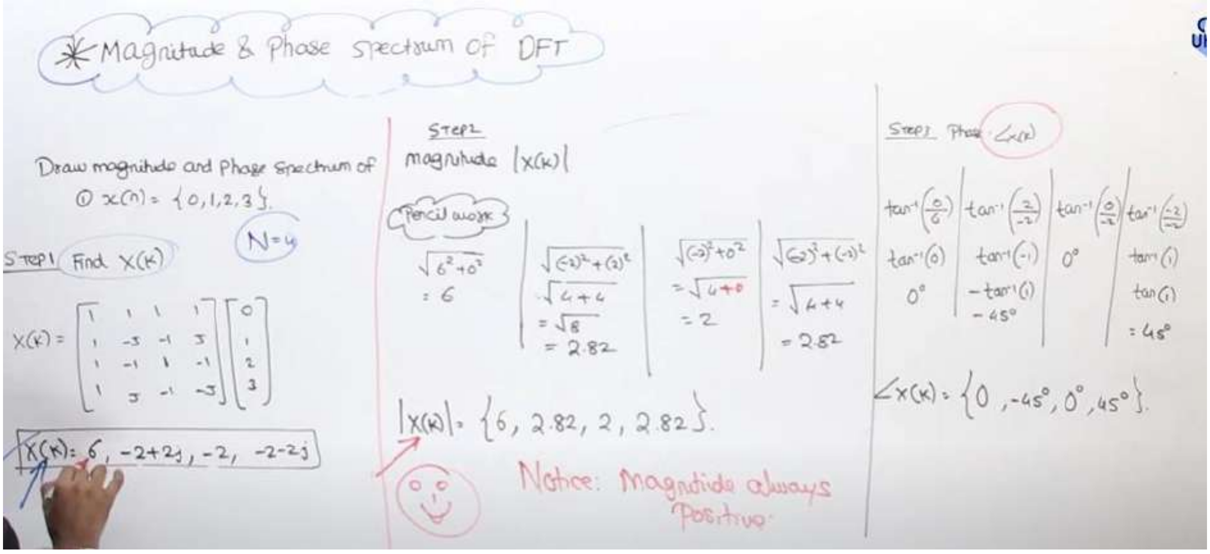


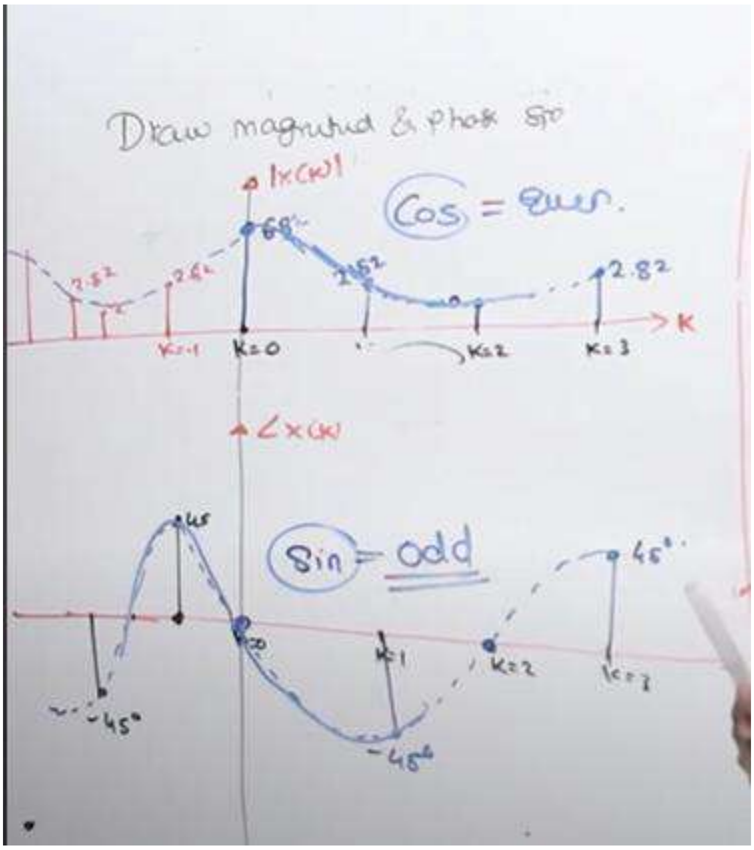
Symmetry property in DFT



Magnitude and phase spectrum In Dft







Sums using Properties of DFT

• Find DFT of $x(n) = \{1, 2, 3, 4\}$

Not otherwise

Sol $x(n) = \{1, 2, 3, 4\}$

$n=0 \quad 1 \quad 2 \quad 3$

Here, $N=4$, $n=0$ to $N-1$

$n=0$ to 3

we know that,

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

By matrix method

$$\begin{bmatrix} X[0] \\ X[1] \\ X[2] \\ X[3] \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -J & -1 & J \\ 1 & -1 & 1 & -1 \\ 1 & J & -1 & -J \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$$

$$= \begin{bmatrix} 10 \\ -2+2J \\ -2 \\ -2-2J \end{bmatrix}$$

$$X(k) = \{10, -2+2J, -2, -2-2J\}$$

May-18 (10 marks) • Using Result obtained in part 1 & Not otherwise Find DFT of Following Sequence.

Not otherwise "Use Property"

$a(n) = \{4, 1, 2, 3\}$

$b(n) = \{2, 3, 4, 1\}$

$c(n) = \{3, 4, 1, 2\}$

$d(n) = \{4, 6, 4, 6\}$

$e(n) = \{1, 4, 3, 2\}$

$f(n) = \{1, -2, 3, -4\}$

•1) $a(n) =$

we have

$x(n) =$

$a(n) =$

Note :-

① Circular Time shift Property.

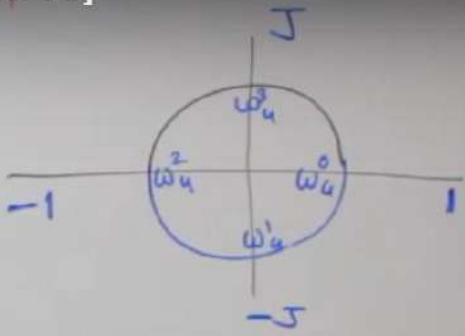
$$x(n+l) = W_N^{-lk} X(k)$$

$$x(n-l) = W_N^{lk} X(k)$$

$$X(k) = \{\underset{k=0}{10}, \underset{1}{-2+2j}, \underset{2}{-2}, \underset{3}{-2-2j}\}$$

tsp/Dsp [Lec 15]

Not otherwise



•1) $a(n) = \{4, 1, 2, 3\}$

we have

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

$$\boxed{a(n) = x(n-1)}$$

By Circular time shift Property

$$\boxed{A[k] = W_4^{1k} X(k)} \quad (*)$$

Note, $k = 0, 1, 2, 3$

$k=0$

$A[0] = W_4^0 X(0)$

$= 1 \times 10$

$\boxed{A[0] = 10}$

$k=1$

$A[1] = W_4^1 X(1)$

$= -j(-2+2j)$

$= 2j - 2j^2$

$= 2j + 2$

$k=2$

$A[2] = W_4^2 X(2)$

$= -1(-2)$

$= 2$

$K = 3$

$-j$

$$A[3] = W_4^3\, x(3)$$

$$= j(-2-2j)$$

$$= -2j - 2j^2$$

$$= -2j + 2$$

$$= 2 - 2j$$

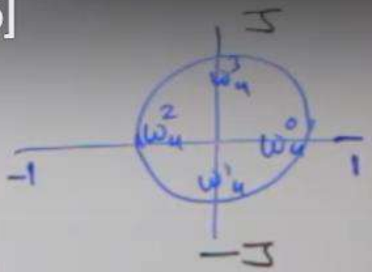
$$A(k) = \{10,\ 2+2j,\ 2,\ 2-2j\}$$

3 step verification for it now, check if the given sequence a(n) is real or not….. If it’s real, then

1) Add all the terms of a(n) and check you must get the first term of A[k] i.e. A[0]
2) Then if n is the length of sequence then check the number at n/2 position, here 4/2 i.e. 2nd position the number must be real
3) And the number at 1st and 3rd position should be complex conjugate of each other.

tsp/Dsp [Lec 15]

Not otherwise



2) $b(n) = \{2, 3, 4, 1\}$

we have

$x(n) = \{1, 2, 3, 4\}$

Focus.

$$\boxed{b(n) = x(n+1)}$$

By Circular Time shift Property.

$$B[k] = W_4^{-1k} X(k)$$

K=0

$$B(0) = W_4^0 X(0)$$
$$= 1 \times 10 = 10$$

K=1

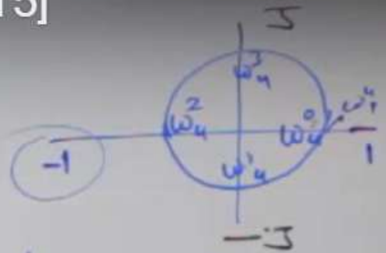
$$B(1) = W_4^{-1} X(1)$$
$$= j(-2+2j)$$
$$= -2-2j$$

$$B(2) = 2$$

$$B(3) = -2+2j$$

tsp/Dsp [Lec 15]

Not otherwise



3) $c(n) = \{3, 4, 1, 2\}$

we have

$$x(n) = \{1, 2, 3, 4\}$$

Focus.

$$\boxed{c(n) = x(n-2)}$$

By Circular Time shift Property.

$$C(k) = W_4^{2k} X(k)$$

K=0

$$C(0) = W_4^0 X(0)$$
$$= 1 \times [illegible]$$
$$\boxed{C(0) = 10}$$

K=1

$$C(1) = W_4^2 X(1)$$
$$= -1(-2+2j)$$
$$= 2-2j$$

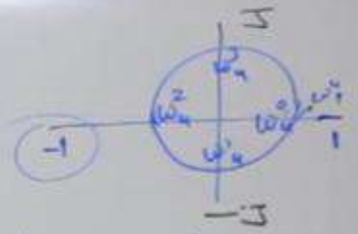
K=2

$$C(2) = W_4^4 X(2)$$
$$= 1(-2) = -2$$

K=3

$$C(3) = W_4^6 X(3)$$
$$= -1(-2-2j)$$
$$= 2+2j.$$

Not otherwise



• 4> $d(n) = \{4, 6, 4, 6\}$

we have

$$x(n) = \{1, 2, 3, 4\}$$

Focus

Rough work

$$x(n) + \text{कुछ} = d(n)$$

$$\text{कुछ} = d(n) - x(n)$$

$$= \{4,6,4,6\} - \{1,2,3,4\}$$

$$\therefore \{3, 4, 1, 2\}$$

$$= c(n)$$

$$= x(n-2)$$

• $d(n) = x(n) + x(n-2)$

By Linearity Property

$$D(k) = X(k) + W_4^{2k} X(k)$$

$\boxed{K=0}$

$$D(0) = X(0) + W_4^0 X(0)$$
$$= 20$$

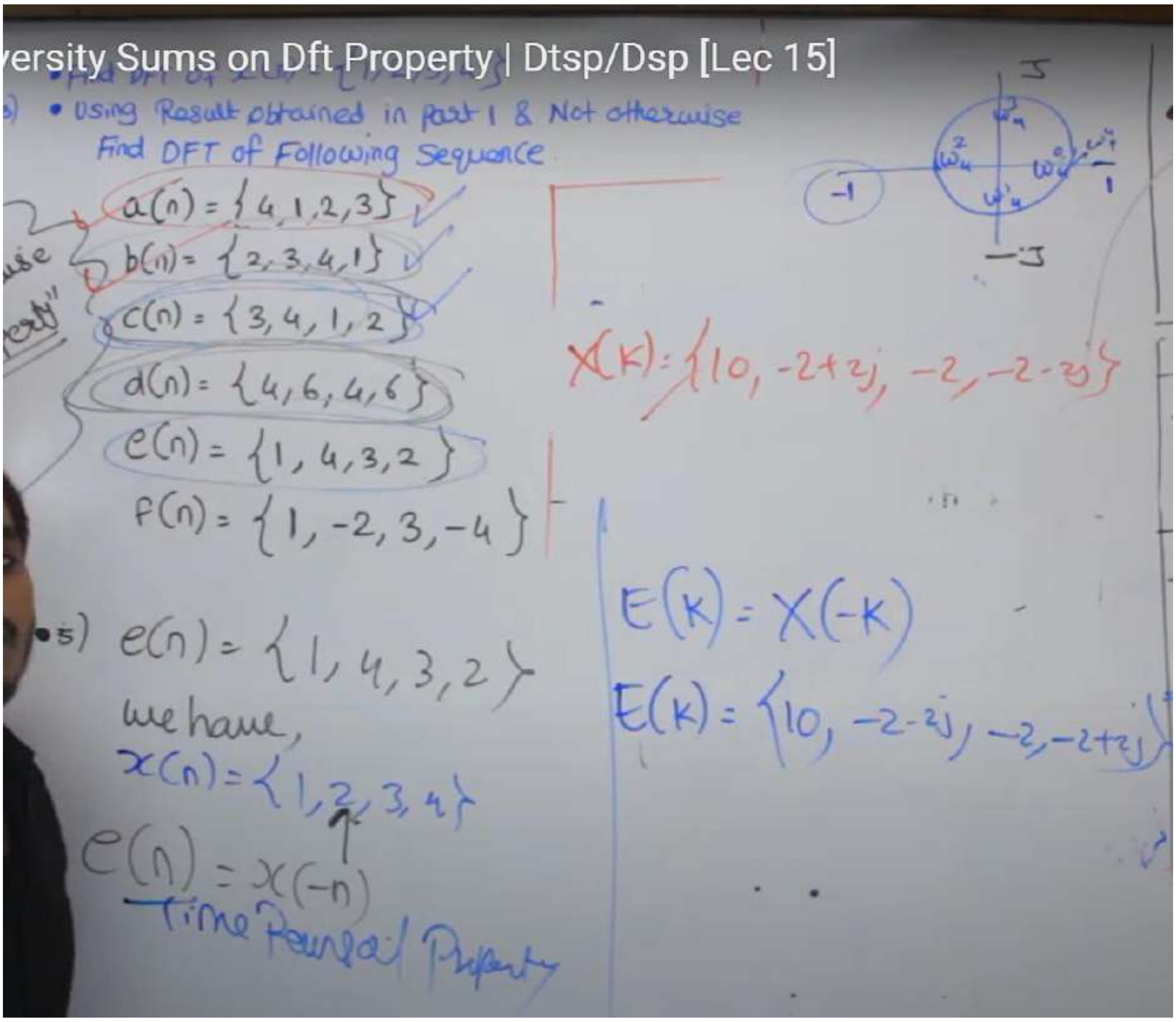
$\boxed{K=1}$

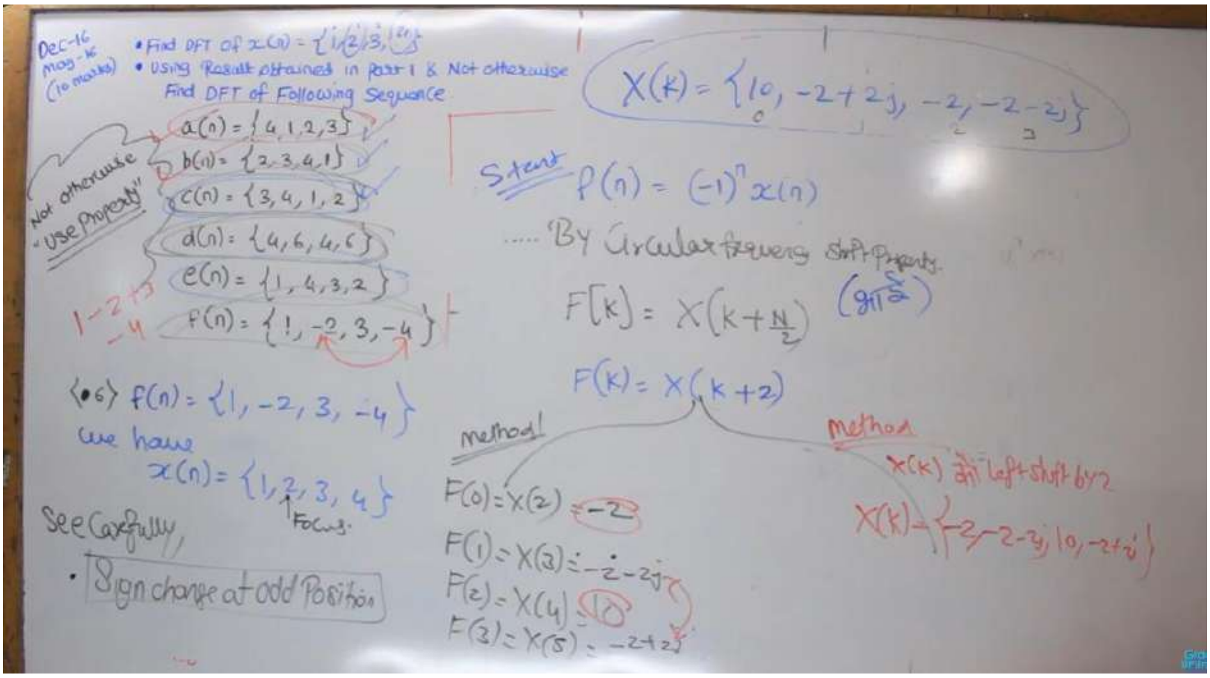
$$D(1) = X(1) + W_4^2 X(1)$$
$$= -2+2j + (-1)(-2+2j)$$
$$= -2+2j+2-2j$$
$$= 0$$

Similarly $D(2) = -4$

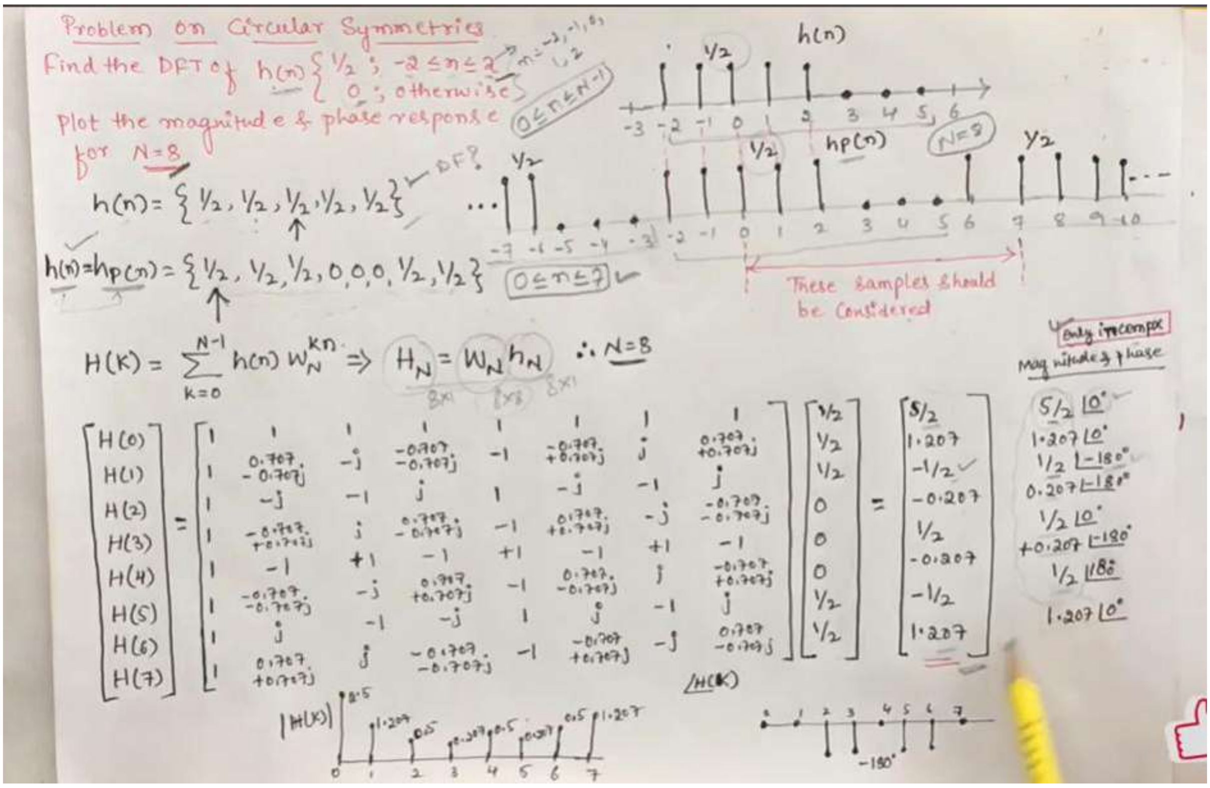
$D(3) = 0$

$$D(k) = \{20, 0, -4, 0\}$$


ersity Sums on Dft Property | Dtsp/Dsp [Lec 15]
• Using Result obtained in part 1 & Not otherwise
Find DFT of Following Sequence
a(n) = {4, 1, 2, 3}
b(n) = {2, 3, 4, 1}
c(n) = {3, 4, 1, 2}
d(n) = {4, 6, 4, 6}
e(n) = {1, 4, 3, 2}
f(n) = {1, -2, 3, -4}
X(k) = {10, -2+2j, -2, -2-2j}
5) e(n) = {1, 4, 3, 2}
we have,
x(n) = {1, 2, 3, 4}
e(n) = x(-n)
Time Reversal Property
E(k) = X(-k)
E(k) = {10, -2-2j, -2, -2+2j}
-1
1
j
-j


Dec-16
May-16
(10 marks)
• Find DFT of x(n) = {1, 2, 3, 4}
• Using Result obtained in part 1 & Not otherwise
Find DFT of Following Sequence
a(n) = {4, 1, 2, 3}
b(n) = {2, 3, 4, 1}
c(n) = {3, 4, 1, 2}
d(n) = {4, 6, 4, 6}
e(n) = {1, 4, 3, 2}
f(n) = {1, -2, 3, -4}
Not otherwise "Use Property"
X(k) = {10, -2+2j, -2, -2-2j}
Start f(n) = (-1)^n x(n)
..... By Circular frequency shift Property
F[k] = X(k + N/2)
F(k) = X(k+2)
6) f(n) = {1, -2, 3, -4}
we have
x(n) = {1, 2, 3, 4}
Focus.
See Carefully,
• Sign change at odd Position
method 1
F(0) = X(2) = -2
F(1) = X(3) = -2-2j
F(2) = X(4) = 10
F(3) = X(5) = -2+2j
method 2
X(k) को left shift by 2
X(k) = {-2, -2-2j, 10, -2+2j}

find DFT using circular symmetry property



Problem on Circular Symmetries

Find the DFT of $h(n) = \begin{cases} 1/2 ; & -2 \le n \le 2 \\ 0 ; & \text{otherwise} \end{cases}$ $\quad n = -2,-1,0,1,2$ $\quad (0 \le n \le N-1)$

Plot the magnitude & phase response for $N=8$

$h(n) = \{1/2, 1/2, \underset{\uparrow}{1/2}, 1/2, 1/2\}$ DFT?

$h(n) = h_p(n) = \{\underset{\uparrow}{1/2}, 1/2, 1/2, 0, 0, 0, 1/2, 1/2\}$ $\boxed{0 \le n \le 7}$

These samples should be considered

$$H(K) = \sum_{k=0}^{N-1} h(n) W_N^{kn} \Rightarrow H_N = W_N h_N \quad \therefore N=8$$

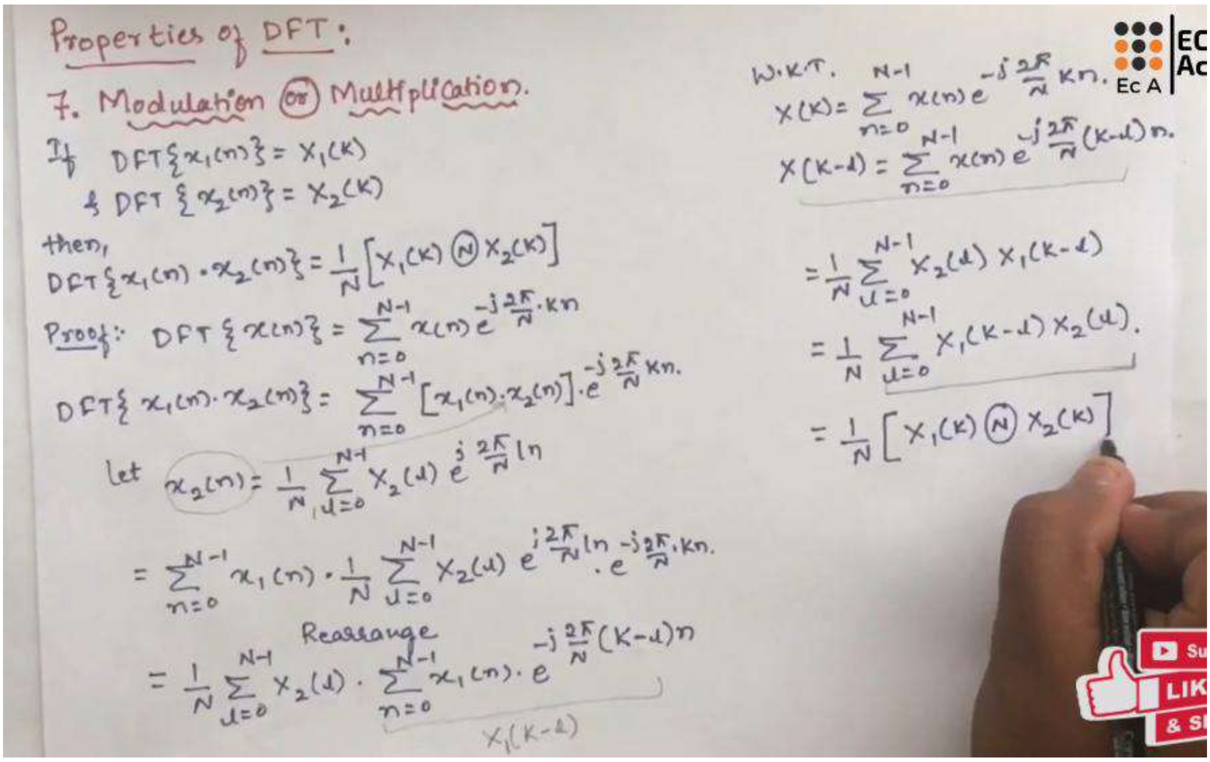
$$\begin{bmatrix} H(0) \\ H(1) \\ H(2) \\ H(3) \\ H(4) \\ H(5) \\ H(6) \\ H(7) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0.707-0.707j & -j & -0.707-0.707j & -1 & -0.707+0.707j & j & 0.707+0.707j \\ 1 & -j & -1 & j & 1 & -j & -1 & j \\ 1 & -0.707-0.707j & j & 0.707-0.707j & -1 & 0.707+0.707j & -j & -0.707+0.707j \\ 1 & -1 & +1 & -1 & +1 & -1 & +1 & -1 \\ 1 & -0.707+0.707j & -j & 0.707+0.707j & -1 & 0.707-0.707j & j & -0.707-0.707j \\ 1 & j & -1 & -j & 1 & j & -1 & -j \\ 1 & 0.707+0.707j & j & -0.707+0.707j & -1 & -0.707-0.707j & -j & 0.707-0.707j \end{bmatrix} \begin{bmatrix} 1/2 \\ 1/2 \\ 1/2 \\ 0 \\ 0 \\ 0 \\ 1/2 \\ 1/2 \end{bmatrix} = \begin{bmatrix} 5/2 \\ 1.207 \\ -1/2 \\ -0.207 \\ 1/2 \\ -0.207 \\ -1/2 \\ 1.207 \end{bmatrix}$$

Only in complex: Magnitude & phase

$5/2 \angle 0°$, $1.207 \angle 0°$, $1/2 \angle -180°$, $0.207 \angle -180°$, $1/2 \angle 0°$, $+0.207 \angle -180°$, $1/2 \angle 180°$, $1.207 \angle 0°$

$|H(K)|$ $\quad \angle H(K)$ $\quad -180°$

Multiplication or Modulation property of DFT



Properties of DFT:

7. Modulation or Multiplication.

If $DFT\{x_1(n)\} = X_1(K)$

& $DFT\{x_2(n)\} = X_2(K)$

then,

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

Proof: $DFT\{x(n)\} = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi}{N} \cdot kn}$

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

let $x_2(n) = \frac{1}{N} \sum_{l=0}^{N-1} X_2(l) e^{j\frac{2\pi}{N}ln}$

$$= \sum_{n=0}^{N-1} x_1(n) \cdot \frac{1}{N} \sum_{l=0}^{N-1} X_2(l) e^{j\frac{2\pi}{N}ln} \cdot e^{-j\frac{2\pi}{N} \cdot kn}$$

Rearrange

$$= \frac{1}{N} \sum_{l=0}^{N-1} X_2(l) \cdot \underbrace{\sum_{n=0}^{N-1} x_1(n) \cdot e^{-j\frac{2\pi}{N}(K-l)n}}_{X_1(K-l)}$$

W.K.T. $X(K) = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi}{N}kn}$.

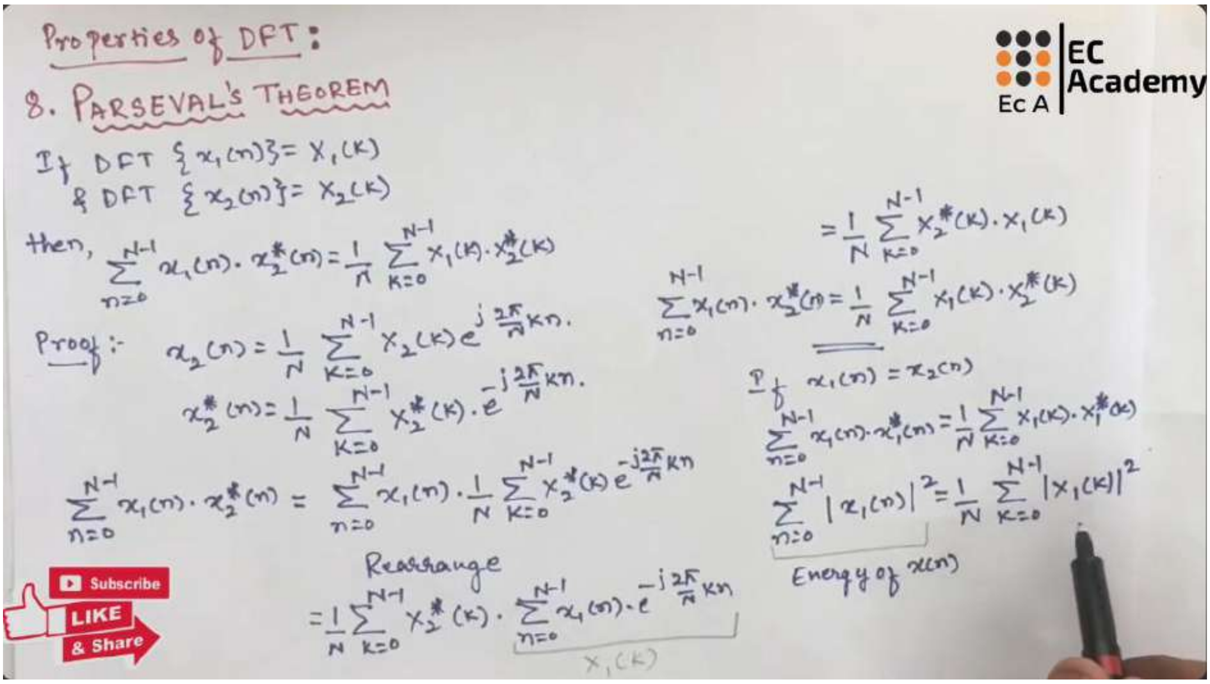
$X(K-l) = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi}{N}(K-l)n}$.

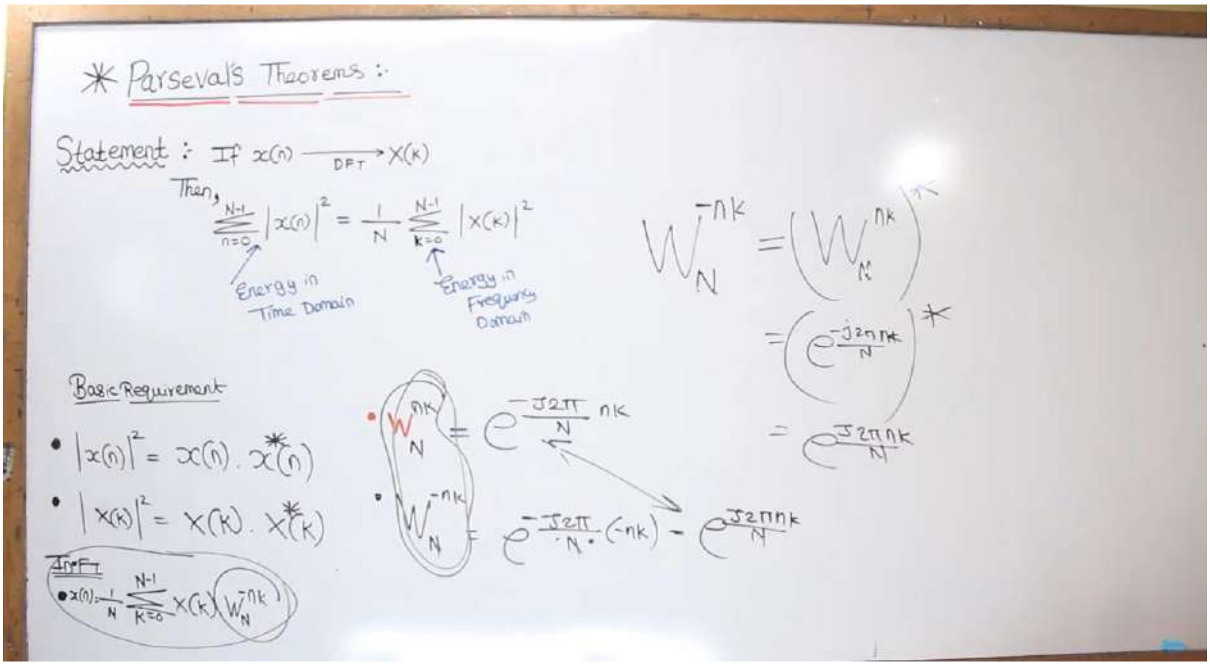
$$= \frac{1}{N} \sum_{l=0}^{N-1} X_2(l) X_1(K-l)$$

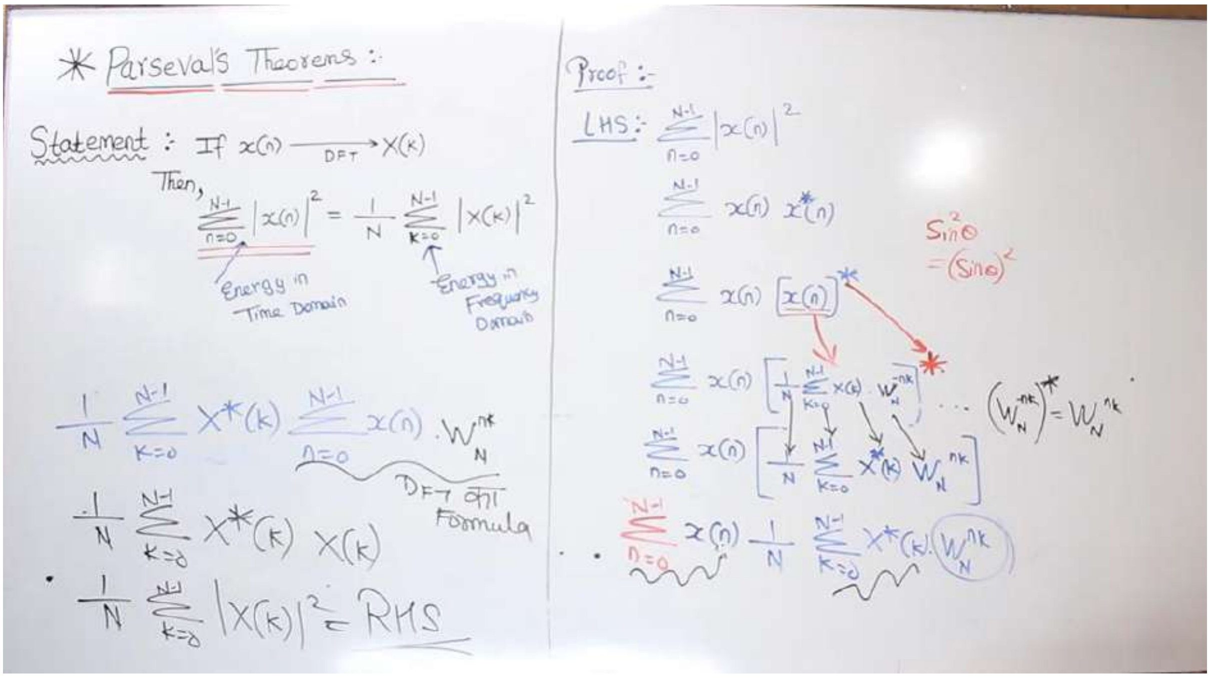
$$= \frac{1}{N} \sum_{l=0}^{N-1} X_1(K-l) X_2(l).$$

$$= \frac{1}{N}\left[X_1(K) \, \textcircled{N} \, X_2(K)\right]$$

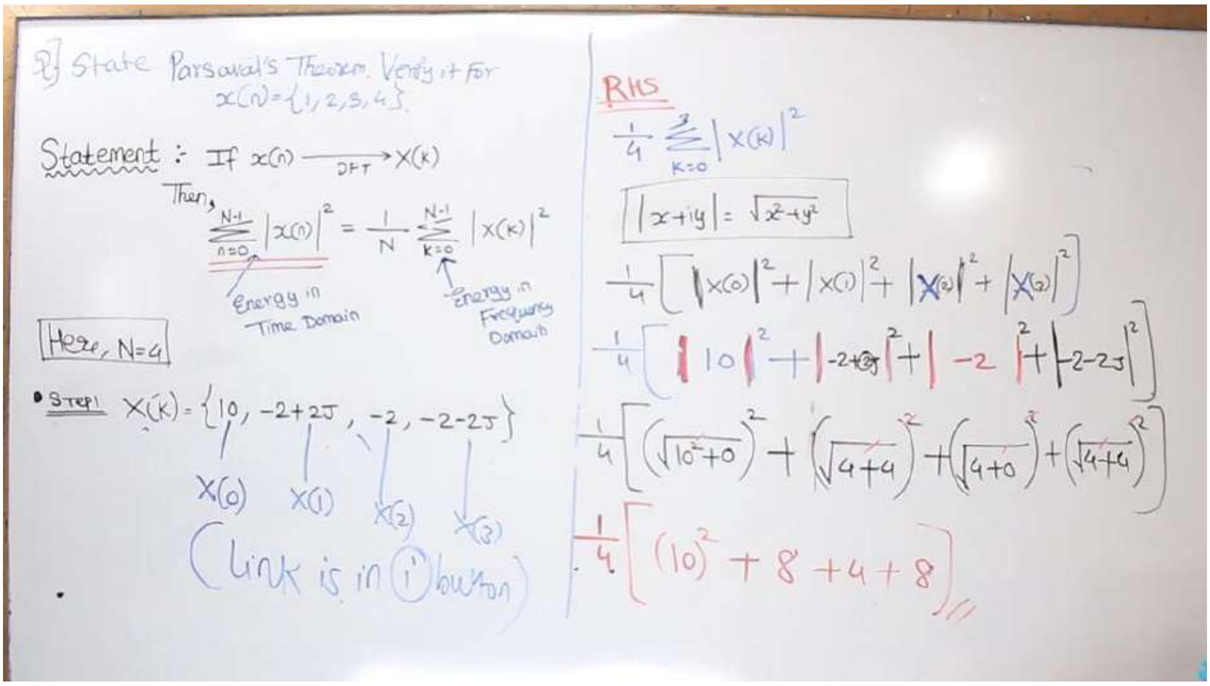
Parseval's theorem







Sum on Parsevals Theorem



RHS $\frac{1}{4}\left[100+8+4+8\right]$

$\frac{1}{4}\left[120\right]$

$=30J$

LHS $\sum_{n=0}^{3}|x(n)|^2$

$|x(0)|^2+|x(1)|^2+|x(2)|+|x(2)|^2$

$|1|^2+|2|^2+|3|^2+|4|^2$

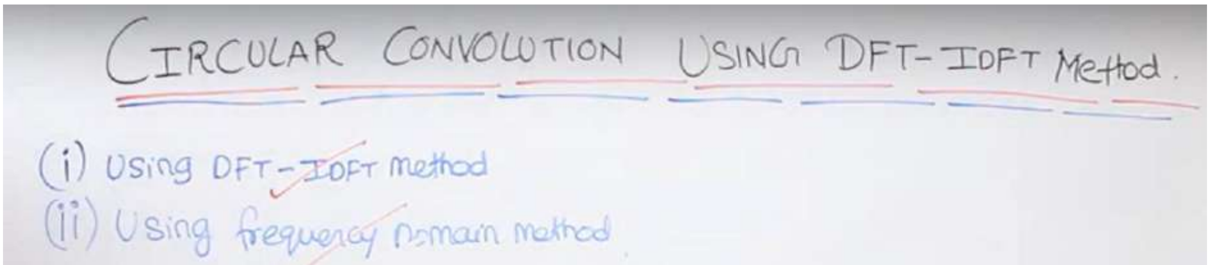
$1+4+9+16$

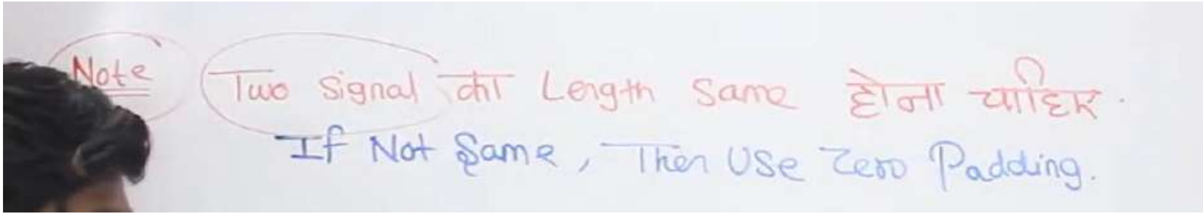
$=30J.$

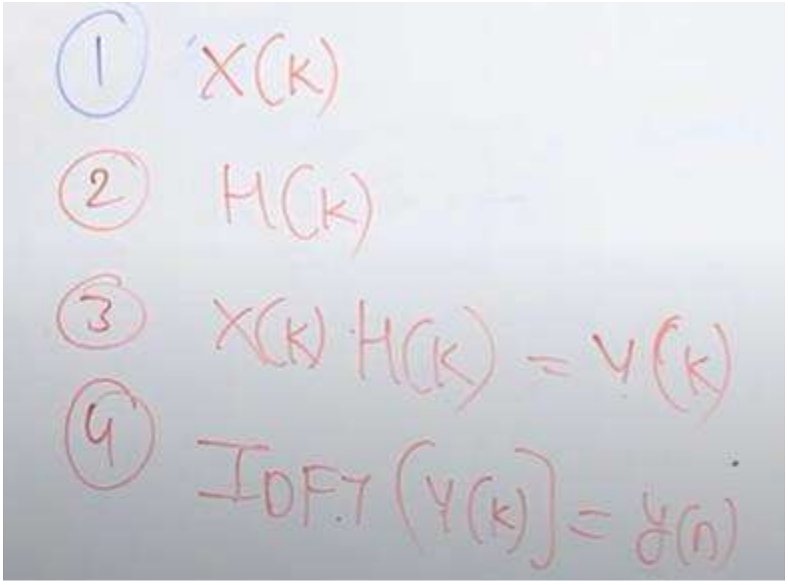
Hence we can say

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

Circular Convolution using Dft-Idft method







* Find Circular Convolutions Using DFT – IDFT method / frequency Domain method.

$x(n) = \{2, 3, 4, 5\}$ & $y(n) = \{5, 2, 3, 4\}$

Step1

$$X(K) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 2 \\ 3 \\ 4 \\ 5 \end{bmatrix}$$

$$\begin{bmatrix} 2+3+4+5 \\ 2-3j-4+5j \\ 2-3+4-5 \\ 2+3j-4-5j \end{bmatrix}$$

$$= \begin{bmatrix} 14 \\ -2+2j \\ -2 \\ -2-2j \end{bmatrix}$$

Step2

$$Y(K) = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} 5 \\ 2 \\ 3 \\ 4 \end{bmatrix}$$

$$Y(K) = \{14, 2+2j, 2, 2-2j\}$$

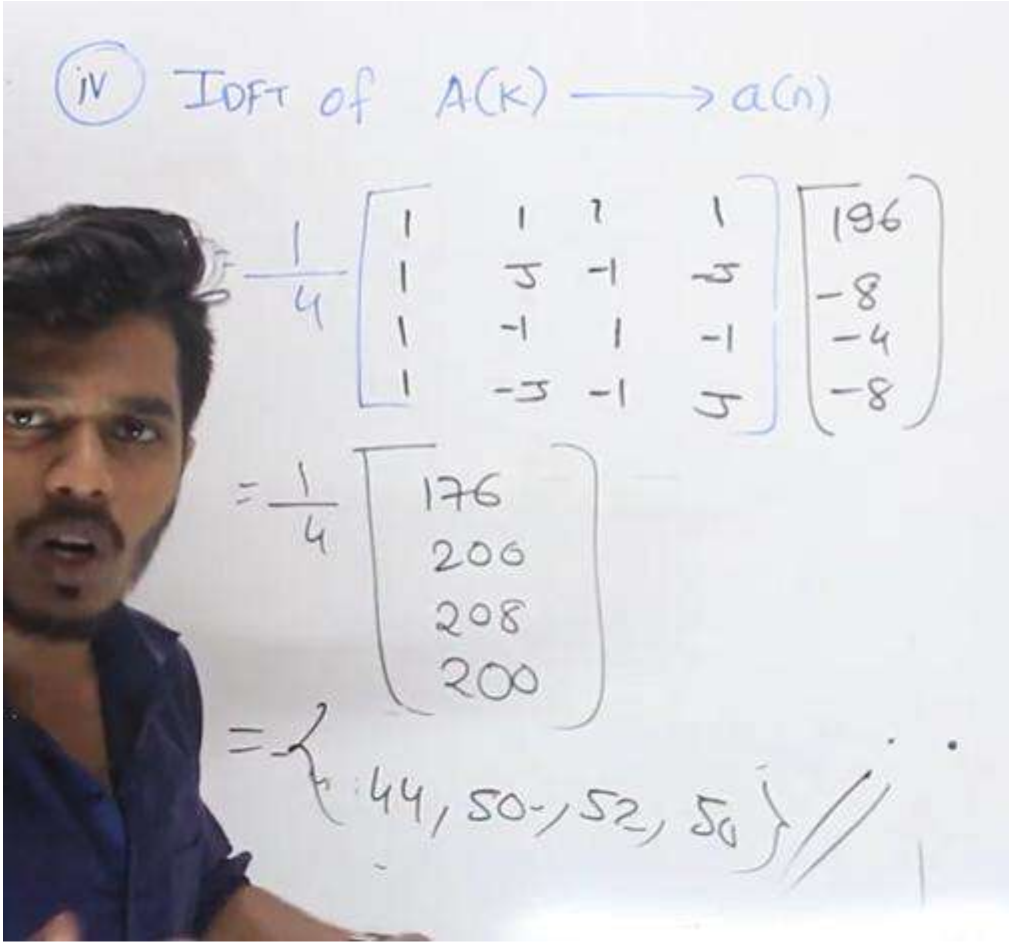
* Find Circular Convolutions Using DFT – IDFT method / frequency Domain method.

$x(n) = \{2, 3, 4, 5\}$ & $y(n) = \{5, 2, 3, 4\}$

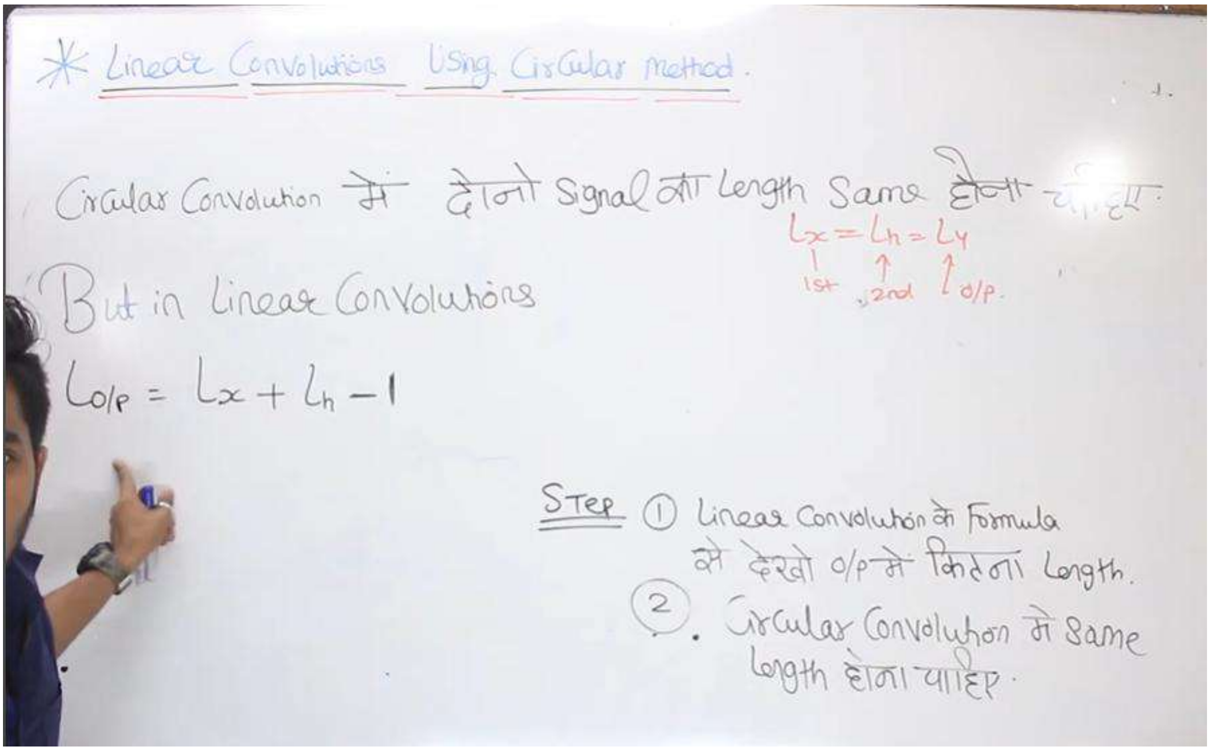
~~Step3~~ Step3 $X(k) \cdot Y(k) = A(k)$

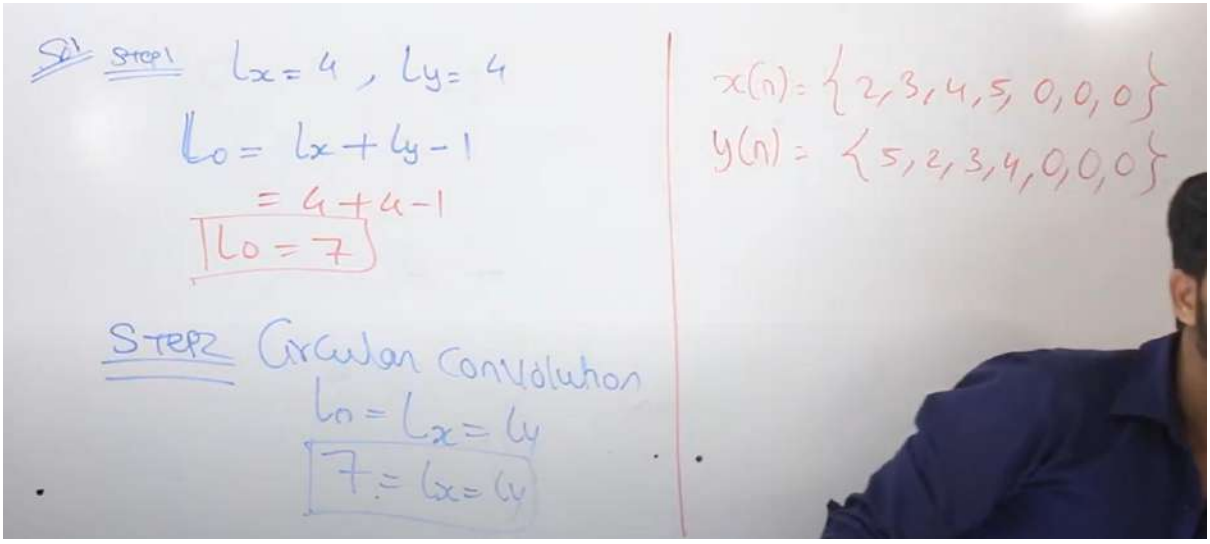
$$\{14, -2+2j, -2, -2-2j\} \cdot \{14, 2+2j, 2, 2-2j\}$$

$$A(k) = \{196, -8, -4, -8\}$$


iv IDFT of A(K) → a(n)
= 1/4 [1 1 1 1; 1 j -1 -j; 1 -1 1 -1; 1 -j -1 j] [196; -8; -4; -8]
= 1/4 [176; 200; 208; 200]
= {44, 50, 52, 50}

Linear convolution using Circular Convolution using Dft-Idft method



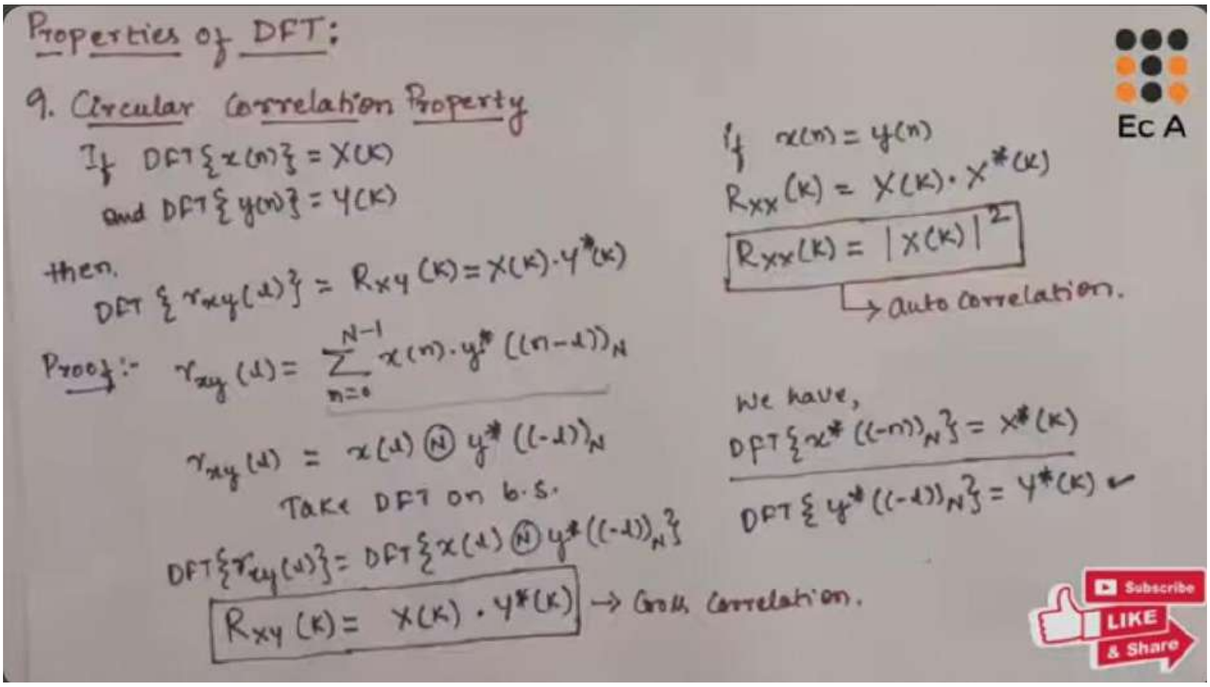


$$x(n) \circledast y(n) = z(n)$$

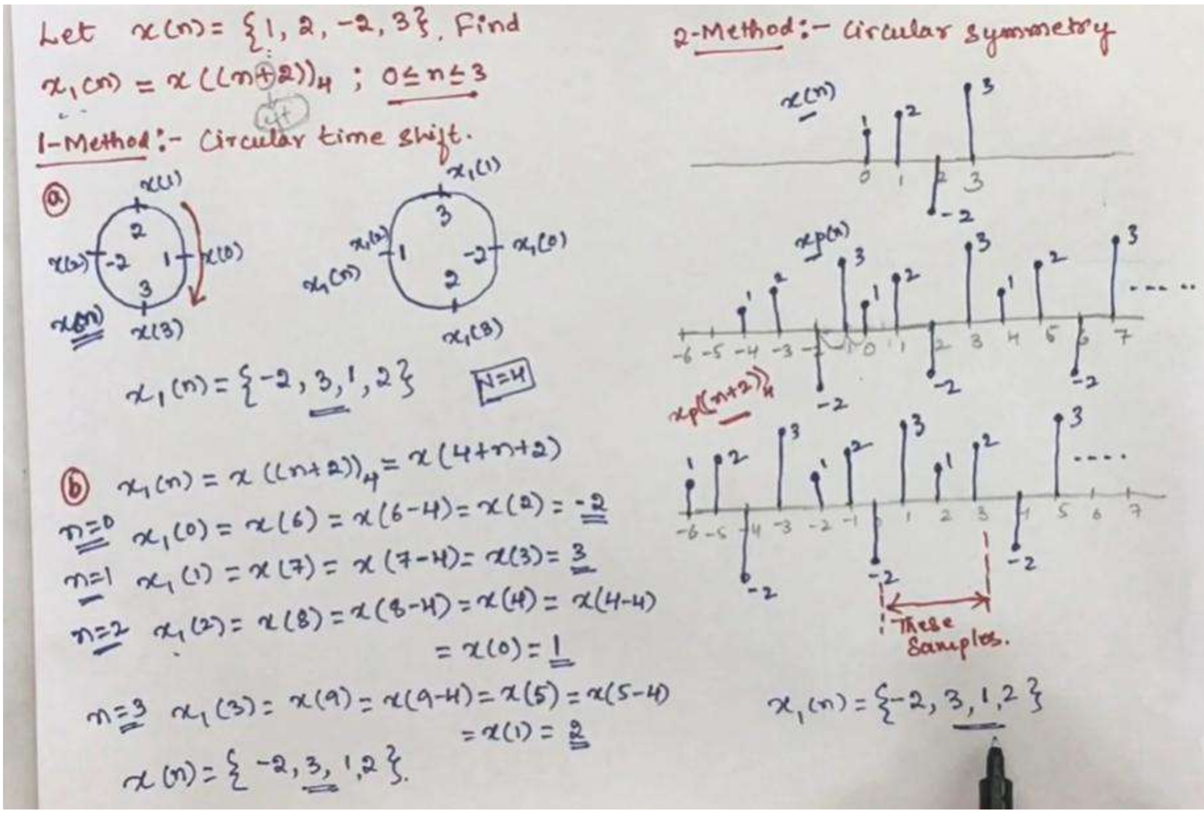
$$\begin{bmatrix} 2 & 0 & 0 & 0 & 5 & 4 & 3 \\ 3 & 2 & 0 & 0 & 0 & 5 & 4 \\ 4 & 3 & 2 & 0 & 0 & 0 & 5 \\ 5 & 4 & 3 & 2 & 0 & 0 & 0 \\ 0 & 5 & 4 & 3 & 2 & 0 & 0 \\ 0 & 0 & 5 & 4 & 3 & 2 & 0 \\ 0 & 0 & 0 & 5 & 4 & 3 & 2 \end{bmatrix} \begin{bmatrix} 5 \\ 2 \\ 3 \\ 4 \\ 0 \\ 0 \\ 0 \end{bmatrix} =$$

$$\begin{bmatrix} 10+0+0\cdots \\ 15+4 \\ 20+6+6 \\ 25+8+9+8 \\ 10+12+12 \\ 15+16 \\ 20 \end{bmatrix} = \begin{bmatrix} 10 \\ 19 \\ 32 \\ 50 \\ 34 \\ 31 \\ 20 \end{bmatrix}$$

Circular correlation property of DFT



Problem on circular time shift and circular symmetry



Problem on symmetry property of dft

The First five points of 8-point DFT $X(K)$ are

$\{0.25, 0.125-j0.3018, 0, 0.125-j0.518, 0\}$

Determine the remaining three point.

Estimate the value of $x(0)$.

$X(0) = 0.25$

$X(1) = 0.125 - j0.3018$

$X(2) = 0$

$X(3) = 0.125 - j0.518$ $\quad \boxed{N=8}$

$X(4) = 0$

$X(5) = ?$

$X(6) = ?$

$X(7) = ?$

Symmetry property

$$\boxed{X(N-K) = X^*(K)}$$

$\underline{K=3}$

$X(8-K) = X^*(K)$

$X(8-3) = X^*(3)$

$X(5) = X^*(3)$

$$\boxed{X(5) = 0.125 + j0.518}$$

$\underline{K=2}$ $\quad X(8-2) = X^*(2)$

$X(6) = X^*(2) \Rightarrow \boxed{X(6) = 0}$

$\underline{K=1}$ $\quad X(8-1) = X^*(1)$

$X(7) = X^*(1)$

$$\boxed{X(7) = 0.125 + j0.3018}$$

IDFT,

$$x(n) = \frac{1}{N} \sum_{K=0}^{N-1} X(K) e^{j2\pi Kn/N}$$

$$x(0) = \frac{1}{N} \sum_{K=0}^{N-1} X(K) \cdot 1$$

$$x(0) = \frac{1}{8} \sum_{K=0}^{7} X(K)$$

$$x(0) = \frac{1}{8}\left[0.25 + 0.125 - j0.3018 + 0 + 0.125 - j0.518 + 0 + 0.125 + j0.518 + 0 + 0.125 + j0.3018\right]$$

$$\boxed{x(0) = 0.09375}$$

Assignment:- $X(0) = 0.25$, $X(1) = 0.125 - j0.3018$, $X(6) = X(4) = 0$, $X(5) = 0.125 - j0.0518$. Determine the remaining Samples.

Problem on symmetry property of dft

Let $X(K)$ denotes N-Point DFT of the N-Point sequence $x(n)$

(i) Show that if $x(n)$ Satisfies the relation $x(n) = -x(N-1-n)$ then $X(0) = 0$

(ii) Show that with N even & if, $x(n) = x(N-1-n)$ then $X\left(\frac{N}{2}\right) = 0$

(i) $x(n) = -x(N-1-n) \rightarrow$ odd sequence

$$X(K) = -j\sum_{n=0}^{N-1} x(n)\sin\left[\frac{2\pi K n}{N}\right] \text{---- Symmetry Property.}$$

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

$$\boxed{X(0) = 0}$$

(ii) $x(n) = x(N-1-n) \rightarrow$ even Sequence

$$X(K) = \sum_{n=0}^{N-1} x(n)\cos\left[\frac{2\pi K n}{N}\right] \text{--- Symmetry Property.}$$

$$X\left(\frac{N}{2}\right) = \sum_{n=0}^{N-1} x(n)\cos\left[\frac{2\pi \frac{N}{2}\cdot n}{N}\right] = \sum_{n=0}^{N-1} x(n)\cos(\pi n)$$

$\underline{N=4}$

$$X\left(\frac{N}{2}\right) = \sum_{n=0}^{3} x(n)\cos(\pi n)$$

$$= x(0) - x(1) + x(2) - x(3)$$

$x(n) = x(N-1-n)$ $\quad \underline{N=4}$

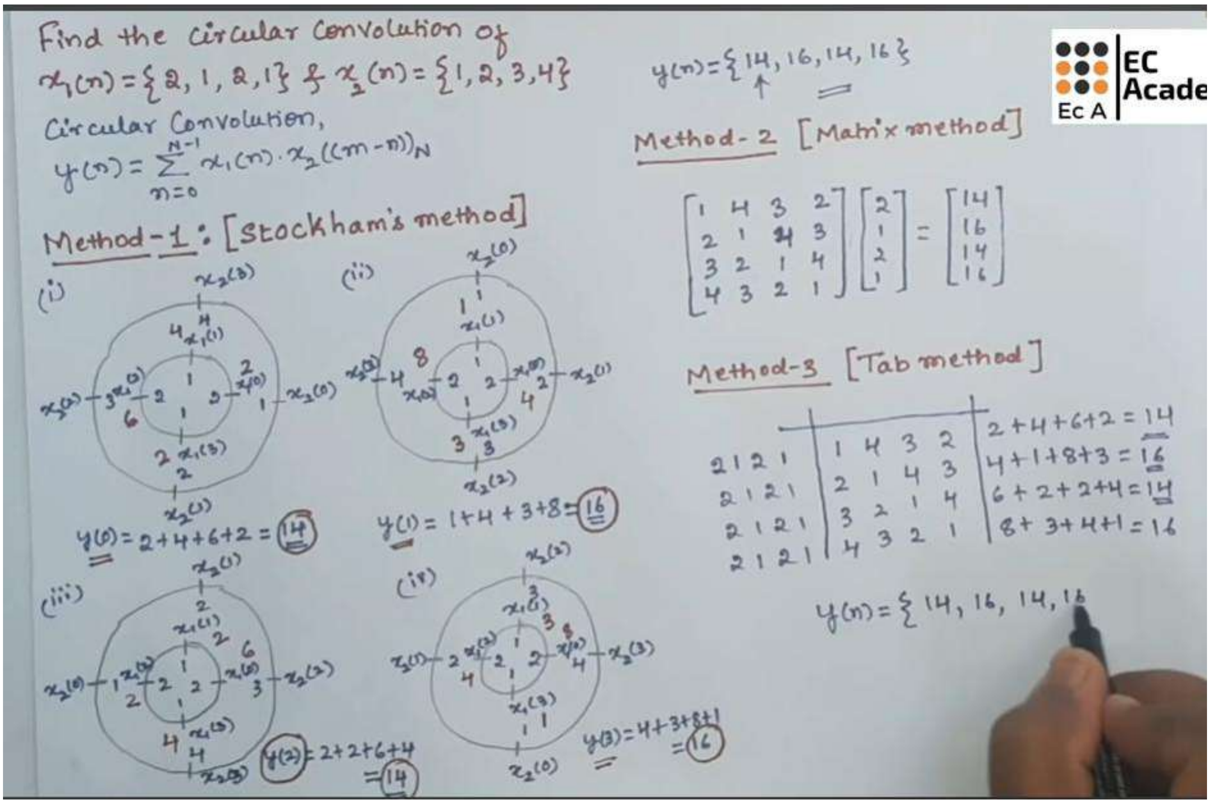
$x(n) = x(3-n)$

$\underline{n=0}$ $\quad x(0) = x(3)$

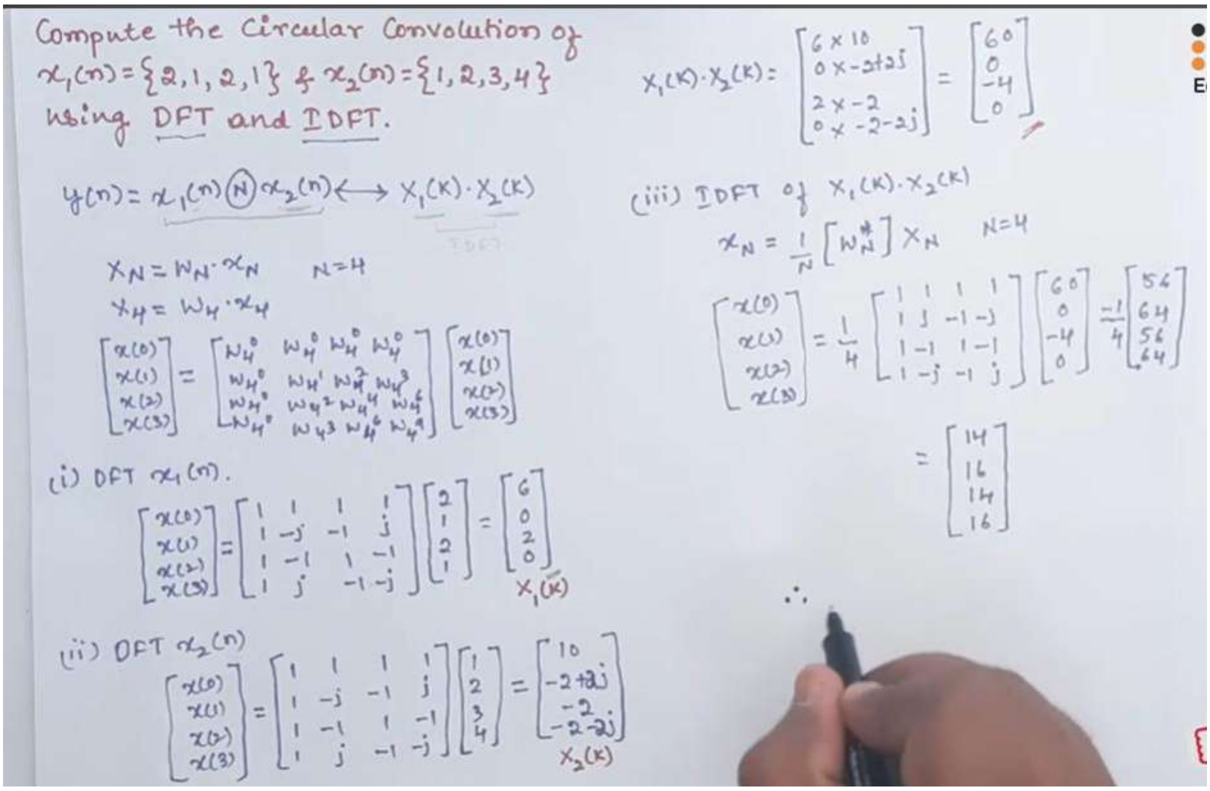
$\underline{n=1}$ $\quad x(1) = x(2)$

$$\boxed{X\left(\frac{N}{2}\right) = 0}$$

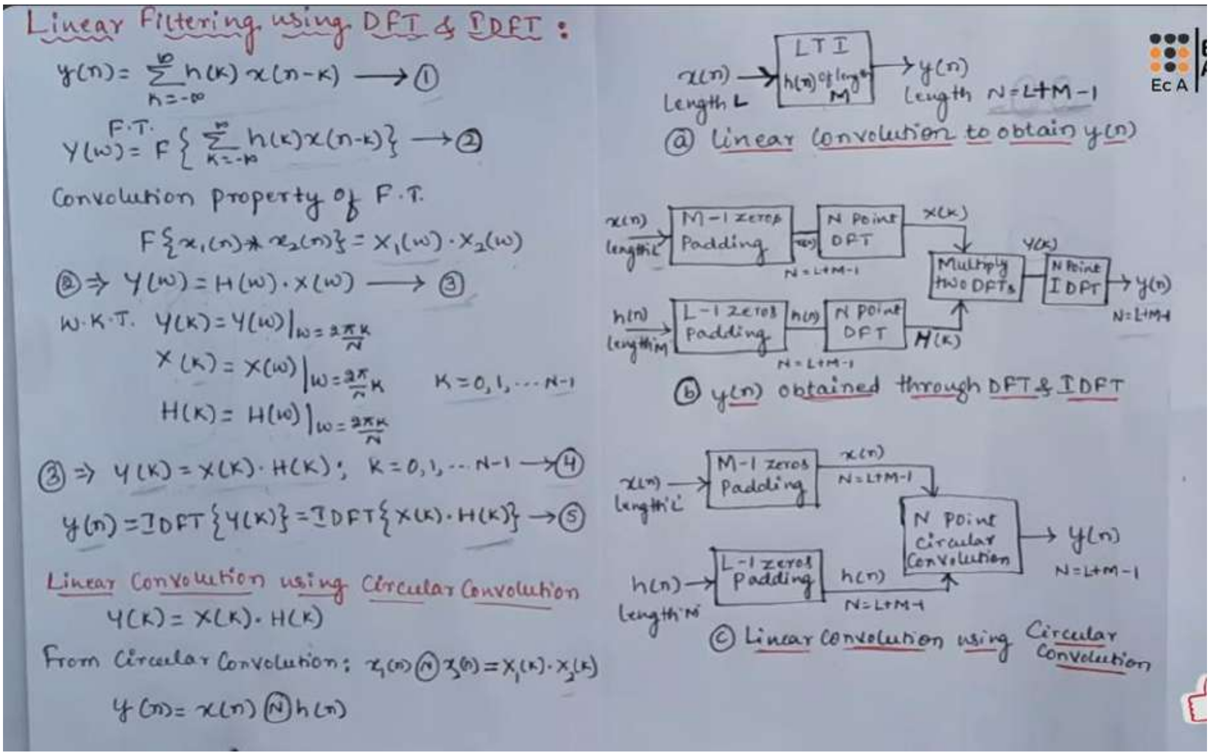
Problem on circular convolution using stockhams method, matrix method and Tab method



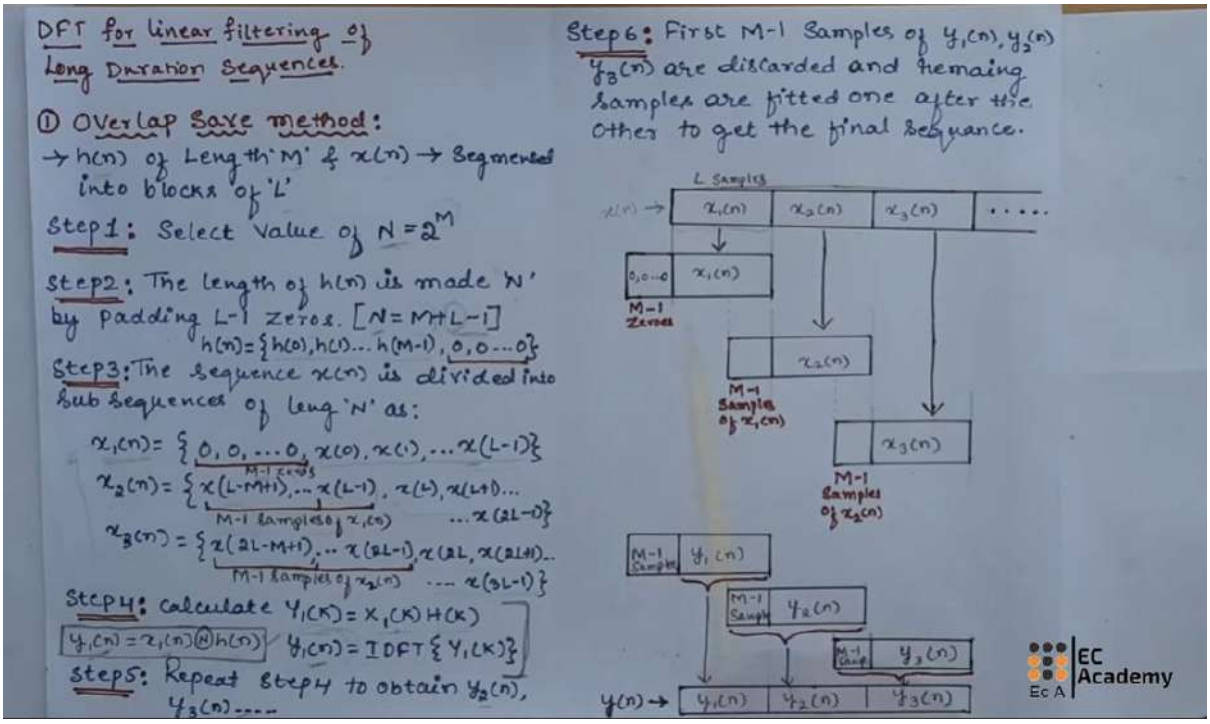
Problem on circular convolution using dft & idft



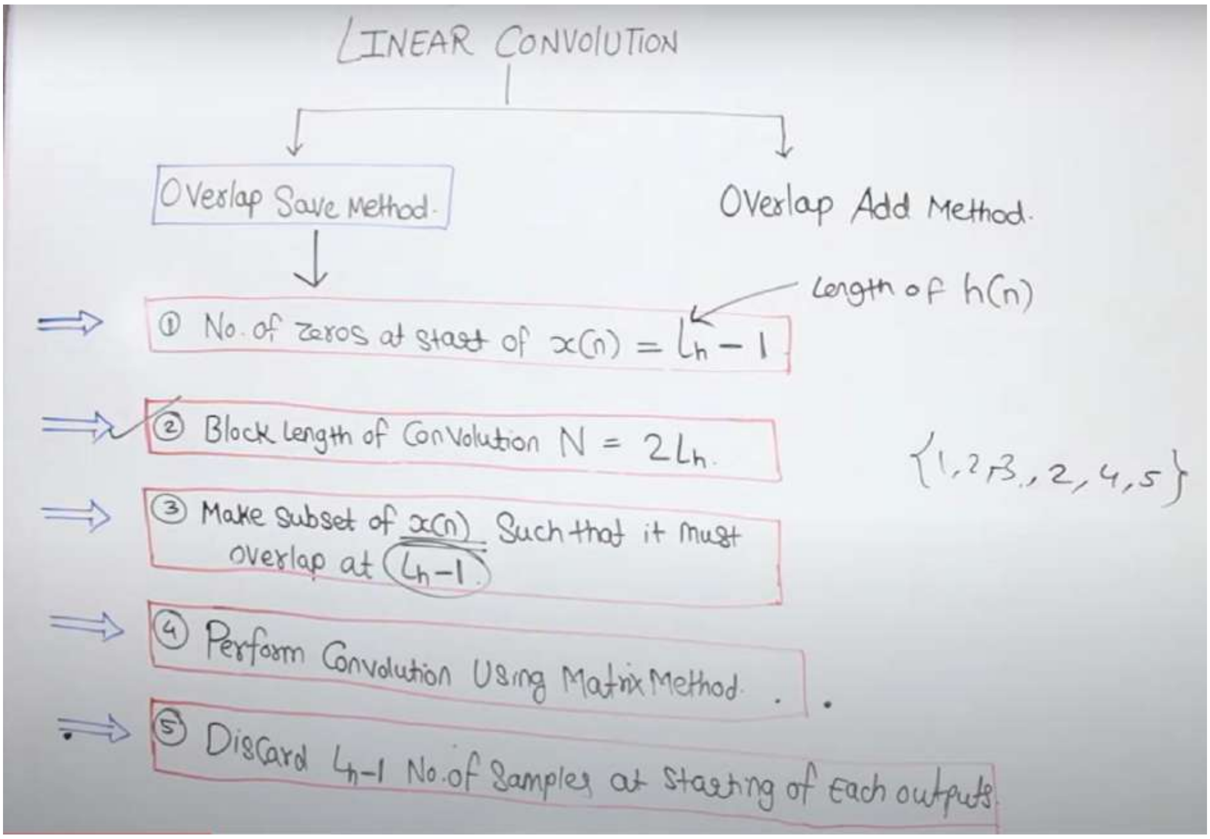
linear filtering method using dft and circular convolution



Overlap Save method for linear filtering of long duration sequence



Problem on Overlap save method



* Find Response of the System where Input $x(n) = \{1,2,-1,2,3,-2,-3,-1,1,1,2,-1\}$ and $h(n) = \{1,2,3\}$ Using Overlap save method (10 Marks)

Sol Here, $L_h = 3$.

STEP1 :- No. of zeros at start of $x(n) = L_h - 1 = 3 - 1 = 2$.

$$x(n) = \{0,0,1,2,-1,2,3,-2,-3,-1,1,1,2,-1\}$$

STEP2 : Block Length of Convolutions. $N = 2L_h = 6$.

STEP3 :- Make subset of $x(n)$ such that it must overlap at $L_h - 1 = 3-1 = 2$

2 का overlap.

$x_1(n) = \{0,0,1,2,-1,2\}$

$x_2(n) = \{-1,2,3,-2,-3,-1\}$

$x_3(n) = \{-3,-1,1,1,2,-1\}$

$x_4(n) = \{2,-1,0,0,0,0\}$

$h(n) = \{1,2,3,0,0,0\}$

$$\begin{array}{c} x_1(n) \\ y_1(n) = \begin{bmatrix} 0 & 2 & -1 & 2 & 1 & 0 \\ 0 & 0 & 2 & -1 & 2 & 1 \\ 1 & 0 & 0 & 2 & -1 & 2 \\ 2 & 1 & 0 & 0 & 2 & -1 \\ -1 & 2 & 1 & 0 & 0 & 2 \\ 2 & -1 & 2 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \\ 0 \\ 0 \\ 0 \end{bmatrix} \end{array}$$

$$= \begin{bmatrix} 4-3 \\ 6 \\ 1 \\ 2+2 \\ -1+4+3 \\ \text{[illegible]} \end{bmatrix} \qquad y_1(n) = \begin{bmatrix} 1 \\ 6 \\ 1 \\ 4 \\ 6 \\ 6 \end{bmatrix}$$

Step 4 Perform Convolution using Matrix method.

$y_2(n) = x_2(n) \circledast h(n)$ ← Circular Convolution.

$$y_2(n) = \begin{bmatrix} -1 & -1 & -3 & -2 & 3 & 2 \\ 2 & -1 & -1 & -3 & -2 & 3 \\ 3 & 2 & -1 & -1 & -3 & -2 \\ -2 & 3 & 2 & -1 & -1 & -3 \\ -3 & -2 & 3 & 2 & -1 & -1 \\ -1 & -3 & -2 & 3 & 2 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$y_2(n) = \begin{bmatrix} -1-2-9 \\ \cancel{2}\cancel{-2}-3 \\ \cancel{3}+4\cancel{-3} \\ -2+6+6 \\ -3-4+9 \\ -1-6-6 \end{bmatrix} \qquad y_2(n) = \begin{bmatrix} -12 \\ -3 \\ 4 \\ 10 \\ 2 \\ -13 \end{bmatrix}$$

Step 4 Perform Convolution using Matrix method.

- $y_1(n) = \{\underline{1, 6}, 1, 4, 6, 6\}$
- $y_2(n) = \{\underline{-12, -3}, 4, 10, 2, -13\}$
- $y_3(n) = \{\underline{1, -10}, -10, 0, 7, 6\} \Leftarrow$
- $y_4(n) = \{\underline{2, 3}, 4, -3, 0, 0\} \Leftarrow$

Step 5:- Discard $\underbrace{L_h - 1}_{=2}$ No. of samples at starting of each output

$$y(n) = \{1, 4, 6, 6, 4, 10, 2, -13, -10, 0, 7, 6, 4, -3, 0, 0\}$$

ademy

Find $y(n)$ for $h(n) = \{1,1,1\}$ and $x(n) = \{\underbrace{3,-1,0,1,3,2}, 0,1,2,1\}$ using Overlap-Save method.

$h(n) = \{1,1,1\} \quad \therefore M = 3$

Step 1: $N = 2^M = 2^3 \Rightarrow \boxed{N=8}$

$N = M+L-1 \Rightarrow L = N-M+1 = 8-3+1 \Rightarrow \boxed{L=6}$

$6-1=5$

Step 2: $h(n) = \{1,1,1,\underbrace{0,0,0,0,0}_{L-1 \text{ zeros}}\}$

$3-1=2$

Step 3:

$x_1(n)\ \{\underbrace{0,0}_{M-1 \text{ zeros}},3,-1,0,1,\underbrace{3,2}\}$

$x_2(n)\ \{\underbrace{3,2}_{M-1\ x_1(n)},0,1,2,1,0,0\}$

Step 4:- $y_1(n) = x_1(n) \circledN h(n)$

$$\begin{bmatrix} 0 & 2 & 3 & 1 & 0 & -1 & 3 & 0 \\ 0 & 0 & 2 & 3 & 1 & 0 & -1 & 3 \\ 3 & 0 & 0 & 2 & 3 & 1 & 0 & -1 \\ -1 & 3 & 0 & 0 & 2 & 3 & 1 & 0 \\ 0 & -1 & 3 & 0 & 0 & 2 & 3 & 1 \\ 1 & 0 & -1 & 3 & 0 & 0 & 2 & 3 \\ 3 & 1 & 0 & -1 & 3 & 0 & 0 & 2 \\ 2 & 3 & 1 & 0 & -1 & 3 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 5 \\ 2 \\ 3 \\ 2 \\ 2 \\ 0 \\ 4 \\ 6 \end{bmatrix}$$

$y_1(n) = \{\cancel{5,2},3,2,2,0,4,6\}$

Step 5:- $y_2(n) = x_2(n) \circledN h(n)$

$$\begin{bmatrix} 3 & 0 & 0 & 1 & 2 & 1 & 0 & 2 \\ 2 & 3 & 0 & 0 & 1 & 2 & 1 & 0 \\ 0 & 2 & 3 & 0 & 0 & 1 & 2 & 1 \\ 1 & 0 & 2 & 3 & 0 & 0 & 1 & 2 \\ 2 & 1 & 0 & 2 & 3 & 0 & 0 & 1 \\ 1 & 2 & 1 & 0 & 2 & 3 & 0 & 0 \\ 0 & 1 & 2 & 1 & 0 & 2 & 3 & 0 \\ 0 & 0 & 1 & 2 & 1 & 0 & 2 & 3 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 3 \\ 5 \\ 5 \\ 3 \\ 3 \\ 4 \\ 3 \\ 1 \end{bmatrix}$$

$y_2(n) = \{\cancel{3,5},5,3,3,4,3,1\}$

Step 6:

$y(n) = \{3,2,2,0,4,6,5,3,3,4,3,1\}$

Overlap Add method for linear filtering of long duration sequence

DFT for linear Filtering of Long duration Sequence

2. Overlap Add method

$h(n) \rightarrow$ Length 'M' & $x(n) \rightarrow$ Segmented into blocks.

Step 1: Select $N = 2^M$ $\quad h(n) = \{h(0), h(1) \ldots, \underbrace{0,0 \ldots 0}_{L-1}\}$

Step 2: Length of $h(n)$ is made 'N' by Padding $L-1$ zeros $[N = M+L-1]$

Step 3: The Seq. $x(n)$ is divided into Sub seq. of length 'N'

$x_1(n) = \{x(0), x(1) \ldots x(L-1), \underbrace{0,0 \ldots 0}_{M-1}\}$

$x_2(n) = \{x(L), x(L+1), \ldots x(2L-1), \underbrace{0,0 \ldots 0}\}$

$x_3(n) = \{x(2L), x(2L+1), x(3L-1), \underbrace{0,0 \ldots 0}\}$

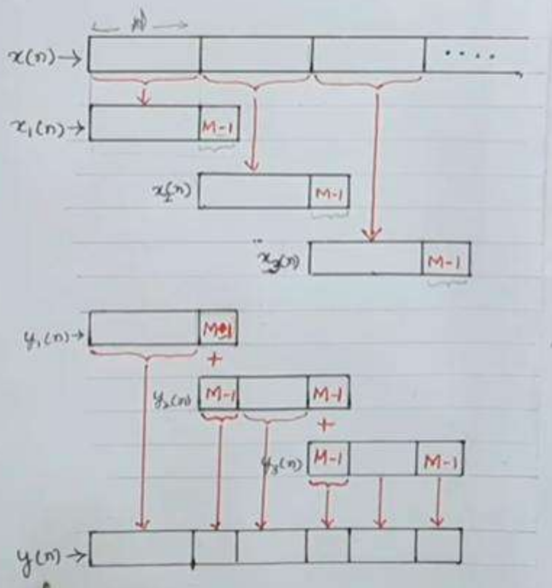
Step 4: Calculate $Y_1(K) = X_1(K) \cdot H(K)$

$y_1(n) = IDFT\{Y_1(K)\}$

(or) $y_1(n) = x_1(n) \circledN h(n)$

Step 5: Repeat Step 4 to obtain $y_2(n), y_3(n) \ldots$

Step 6: Add all $M-1$ samples of each o/p Sequence to first $M-1$ samples of succeeding o/p seq. Such seq. are fitted one after another to get final seq.



Problem on Overlap Add method

Overlap Add Method.

⇒ 1 Block length of Convolutions. $N = 2L_h$.

⇒ 2 Length of Subset from $x(n)$

$$L = N - L_h + 1$$

⇒ 3 No. of zeros at End of Subset : $L_h - 1$.

⇒ 4 Perform Convolution Using matrix Method.

⇒ 5 Overlap $L_h - 1$ Samples at starting of Each output & Add them.

* Find Response of the System where Input $x(n) = \{1,2,-1,2,3,-2,-3,-1,1,1,2,-1\}$ and $h(n) = \{1,2,3\}$ Using Overlap Add method (10 Marks)

Sol. Here, $L_h = 3$

Step 1 : Block Length of Convolutions $N = 2L_h$

$$\boxed{N = 6.}$$

Step 2 Length of Subset from $x(n)$

$$L = N - L_h + 1$$
$$= 6 - 3 + 1$$
$$\boxed{L = 4}$$

Step 3 No. of Zeros at End (subset) $L_h - 1$, $3 - 1 = 2$

$x_1(n) = \{1, 2, -1, 2, 0, 0\}$

$x_2(n) = \{3, -2, -3, -1, 0, 0\}$

$x_3(n) = \{1, 1, 2, -1, 0, 0\}$

$h(n) = \{1, 2, 3, 0, 0, 0\}$

$x(n) = \{\underbrace{1, 2, -1, 2}, \underbrace{3, -2, -3, -1}, \underbrace{1, 1, 2, -1}\}$.

* Find Response of the System where Input $x(n) = \{1,2,-1,2,3,-2,-3,-1,1,1,2,-1\}$ and $h(n) = \{1,2,3\}$ Using Overlap Add method (10 Marks)

Step 4 : Perform Convolution Using Matrix Method.

$y_1(n) = x_1(n) \circledast h(n)$

$$y_1(n) = \begin{bmatrix} 1 & 0 & 0 & 2 & -1 & 2 \\ 2 & 1 & 0 & 0 & 2 & -1 \\ -1 & 2 & 1 & 0 & 0 & 2 \\ 2 & -1 & 2 & 1 & 0 & 0 \\ 0 & 2 & -1 & 2 & 1 & 0 \\ 0 & 0 & 2 & -1 & 2 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 3 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$y_1(n) = \begin{bmatrix} 1 \\ 2+2 \\ -1+4+3 \\ \cancel{2-2}+6 \\ 4-3 \\ 6 \end{bmatrix} \qquad y_1(n) = \begin{bmatrix} 1 \\ 4 \\ 6 \\ 6 \\ 1 \\ 6 \end{bmatrix}$$

* Find Response of the System where Input $x(n) = \{1,2,-1,2,3,-2,-3,-1,1,1,2,-1\}$ and $h(n) = \{1,2,3\}$ Using Overlap Add method. (10 Marks)

Step 4 : Perform Convolution Using Matrix Method.

$y_1(n) = \{1,4,6,6,1,6\}$

$y_2(n) = \{3,4,2,-13,-11,-3\}$

$y_3(n) = \{1,3,7,6,4,-3\}$.

Request

Step 5 Overlap $\underset{\substack{3-1 \\ =2}}{L_h - 1}$ samples at Ending of Each output & Add them

$$
\begin{array}{l}
y(n) = \{1,4,6,6,\underline{1,6} \\
\qquad\qquad\qquad\quad 3\ \ 4\ \ 2\ \ -13\ \ \underline{-11\ \ -3} \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad 1\ \ 3\ \ 7\ \ 6\ \ 4\ \ -3\}
\end{array}
$$

$y(n) = \{1,4,6,6,4,10,2,-13,-10,0,7,6,4,-3\}$

For $h(n) = \{3,2,1,1\}$ & $x(n) = \{1,2,3,3,2,1,-1,-2,-3,5,6,-1,2,0,2,1\}$

Find the o/p using overlap Add method assume block length as 7.

EC Academy

$h(n) = \{3,2,1,1\}$ $\therefore M=4$

step 1:- $N=7$ $\qquad N \neq 2^M$

$N = M+L-1 \Rightarrow 7 = 4+L-1 \Rightarrow L=4$

$4-1 \Rightarrow 3$

step 2:- $h(n) = \{3,2,1,1,\underbrace{0,0,0}_{L-1}\}$

$4-1 \Rightarrow 3$

step 3: $x_1(n) = \{1,2,3,3,\underbrace{0,0,0}_{M-1}\}$

$x_2(n) = \{2,1,-1,-2,0,0,0\}$

$x_3(n) = \{-3,5,6,-1,0,0,0\}$

$x_4(n) = \{2,0,2,1,0,0,0\}$

Step 4:- $y_1(n) = x_1(n) \, \textcircled{N} \, h(n)$

$$
\begin{bmatrix}
1 & 0 & 0 & 0 & 3 & 3 & 2 \\
2 & 1 & 0 & 0 & 0 & 3 & 3 \\
3 & 2 & 1 & 0 & 0 & 0 & 3 \\
3 & 3 & 2 & 1 & 0 & 0 & 0 \\
0 & 3 & 3 & 2 & 1 & 0 & 0 \\
0 & 0 & 3 & 3 & 2 & 1 & 0 \\
0 & 0 & 0 & 3 & 3 & 2 & 1
\end{bmatrix}
\begin{bmatrix} 3 \\ 2 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}
=
\underbrace{\begin{bmatrix} 3 \\ 8 \\ 14 \\ 18 \\ 11 \\ 6 \\ 3 \end{bmatrix}}_{y_1(n)}
$$

Step 5 $y_2(n) = x_2(n) \, \textcircled{N} \, h(n)$

$= \{2,1,-1,-2,0,0,0\} \, \textcircled{N} \, \{3,2,1,1,0,0,0\}$

$y_2(n) = \{6,7,1,-5,-4,-3,-2\}$

$y_3(n) = x_3(n) \, \textcircled{N} \, h(n)$

$= \{-3,5,6,-1,0,0,0\} \, \textcircled{N} \, \{3,2,1,1,0,0,0\}$

$y_3(n) = \{-9,9,25,11,9,5,-1\}$.

$y_4(n) = x_4(n) \, \textcircled{N} \, h(n)$

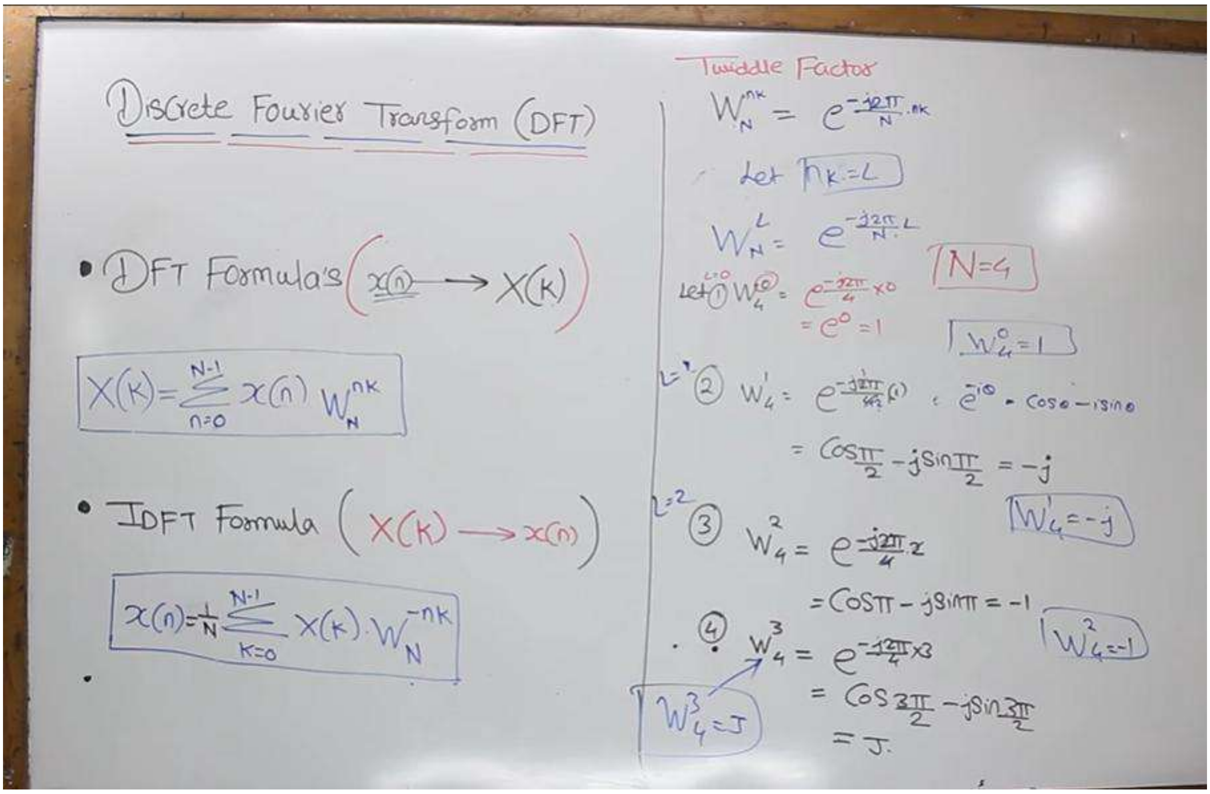
$= \{2,0,2,1,0,0,0\} \, \textcircled{N} \, \{3,2,1,1,0,0,0\}$

$y_4(n) = \{6,4,8,9,4,3,1\}$

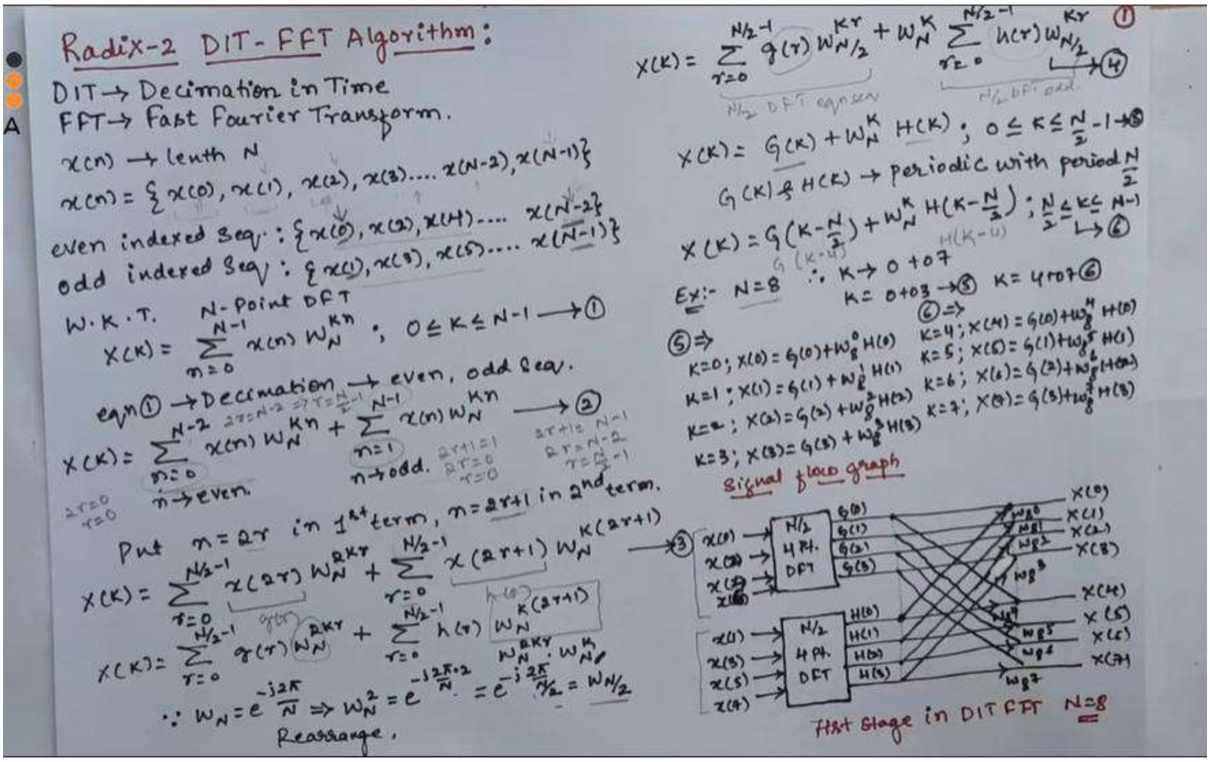
Step 6 $\qquad M=4 \quad M-1 \Rightarrow 3$

$$
\begin{array}{l}
y_1(n) = 3\ \ 8\ \ 14\ \ 18\ \ \overline{11\ \ 6\ \ 3} \\
y_2(n) = \qquad\qquad\qquad 6\ \ 7\ \ 1\ \ -5\ \ \underline{-4\ \ -3\ \ -2} \\
y_3(n) = \qquad\qquad\qquad\qquad\qquad\qquad -9\ \ 9\ \ 25\ \ 11\ \ \underline{9\ \ 5\ \ -1} \\
y_4(n) = \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad 6\ \ 4\ \ 8\ \ 9\ \ 4\ \ 3\ \ 1 \\
\hline
y(n) = 3\ \ 8\ \ 14\ \ 18\ \ 17\ \ 13\ \ 4\ \ -5\ \ -13\ \ 6\ \ 23\ \ 11\ \ 15\ \ 9\ \ 7\ \ 9\ \ 4\ \ 3\ \ 1
\end{array}
$$

Formulas of DFT and IDFT



Radix 2 dit fft algorithm (Part-1)



(2)

$G(K)$ & $H(K) \rightarrow \frac{N}{2}$ point

Combination of two $\frac{N}{4}$ points.

$$G(K) = \sum_{r=0}^{N/2-1} g(r)\, W_{N/2}^{Kr} \rightarrow (7)$$

$$G(K) = \sum_{r=0}^{N/2-2} g(r)\, W_{N/2}^{Kr} + \sum_{r=0}^{N/2-1} g(r)\, W_{N/2}^{Kr}$$

$2l = \frac{N}{2}-2 \Rightarrow l = \frac{N}{4}-1$ $\quad 2l+1 = \frac{N}{2}-1$, $2l = \frac{N}{2}-2$, $l = \frac{N}{4}-1$ $\quad 2l = 0 \Rightarrow l = 0$ $\quad 2l+1 = 1 \Rightarrow l = 0$

$$g(r) = \{ g(0), g(1), g(2) \ldots g(\tfrac{N}{2}-2), g(\tfrac{N}{2}-1) \}$$

Put $r = 2l$ in 1st term, $r = 2l+1$ in 2nd (7)

$$G(K) = \sum_{l=0}^{N/4-1} \underbrace{g(l)}_{a(l)} W_{N/2}^{2Kl} + \sum_{l=0}^{N/4-1} \underbrace{g(2l+1)}_{b(l)} \boxed{W_{N/2}^{K(2l+1)}}$$

$W_{N/2}^{2Kl} \cdot W_{N/2}^{K}$

write here as----> $$G(K) = \underbrace{\sum_{l=0}^{N/4-1} a(l)\, W_{N/4}^{2Kl}}_{A(K)} + W_{N/2}^{K} \underbrace{\sum_{l=0}^{N/4-1} b(l)\, W_{N/4}^{2Kl}}_{B(K)}$$

a(l)WN/4^kl & b(l)WN/4^kl

$W_{N/2}^{2} = W_{N/4}$

$$G(K) = \underbrace{A(K)}_{N/4} + W_{N/2}^{K} \underbrace{B(K)}_{N/4} \; ; \; 0 \le K \le \frac{N}{4}-1 \rightarrow (8)$$

$|||^{ly}$ $$H(K) = C(K) + W_{N/2}^{K} D(K) \; ; \; 0 \le K \le \frac{N}{4}-1 \rightarrow (9)$$

$A(K), B(K), C(K)$ & $D(K) \rightarrow$ periodic $\frac{N}{4}$.

$x(0) \quad x(2) \quad x(4) \quad x(6)$

$x(1) \quad x(3) \quad x(5) \quad x(7)$

(8) $\Rightarrow$ $$G(K) = A(K - \tfrac{N}{4}) + W_{N/2}^{K} B(K - \tfrac{N}{4})$$
$A(K-2)$ $\quad B(K-2)$
$$; \; \frac{N}{4} \le K \le \frac{N}{2}-1 \rightarrow (10)$$

(9) $\Rightarrow$ $$H(K) = C(K - \tfrac{N}{4}) + W_{N/2}^{K} D(K - \tfrac{N}{4}) \; ;$$
$C(K-2)$ $\quad D(K-2)$
$$\frac{N}{4} \le K \le \frac{N}{2}-1 \rightarrow (11)$$

$K \Rightarrow 0$ & $1 \rightarrow$ eqn (8) & (9)

2 & $3 \rightarrow$ eqn (10) & (11)

eqn (8) $\Rightarrow$

$K=0;\ G(0) = A(0) + W_4^0 B(0)$

$K=1;\ G(1) = A(1) + W_4^1 B(1)$

eqn (10)

$K=2;\ G(2) = A(0) + W_4^2 B(0)$

$K=3;\ G(3) = A(1) + W_4^3 B(1)$

eqn (9)

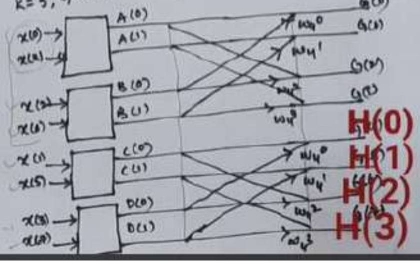
$K=0;\ H(0) = C(0) + W_4^0 D(0)$

$K=1;\ H(1) = C(1) + W_4^1 D(1)$

eqn (11)

$K=2;\ H(2) = C(0) + W_4^2 D(0)$

$K=3;\ H(3) = C(1) + W_4^3 D(1)$



Radix-2 DIT-FFT Algorithm:

Each $\frac{N}{4}$ DFT as two $\frac{N}{8}$ point DFTs.

the 2-point DFT of $x(0)$ & $x(4)$

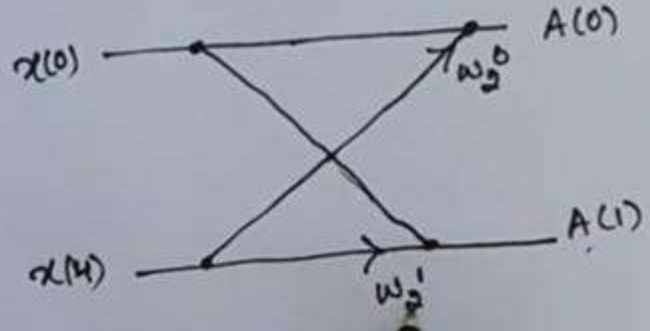
$$A(K) = \sum_{n=0}^{N/4-1} x(n)\, W_{N/4}^{Kn} \; ; \; 0 \le K \le \frac{N}{4} - 1$$

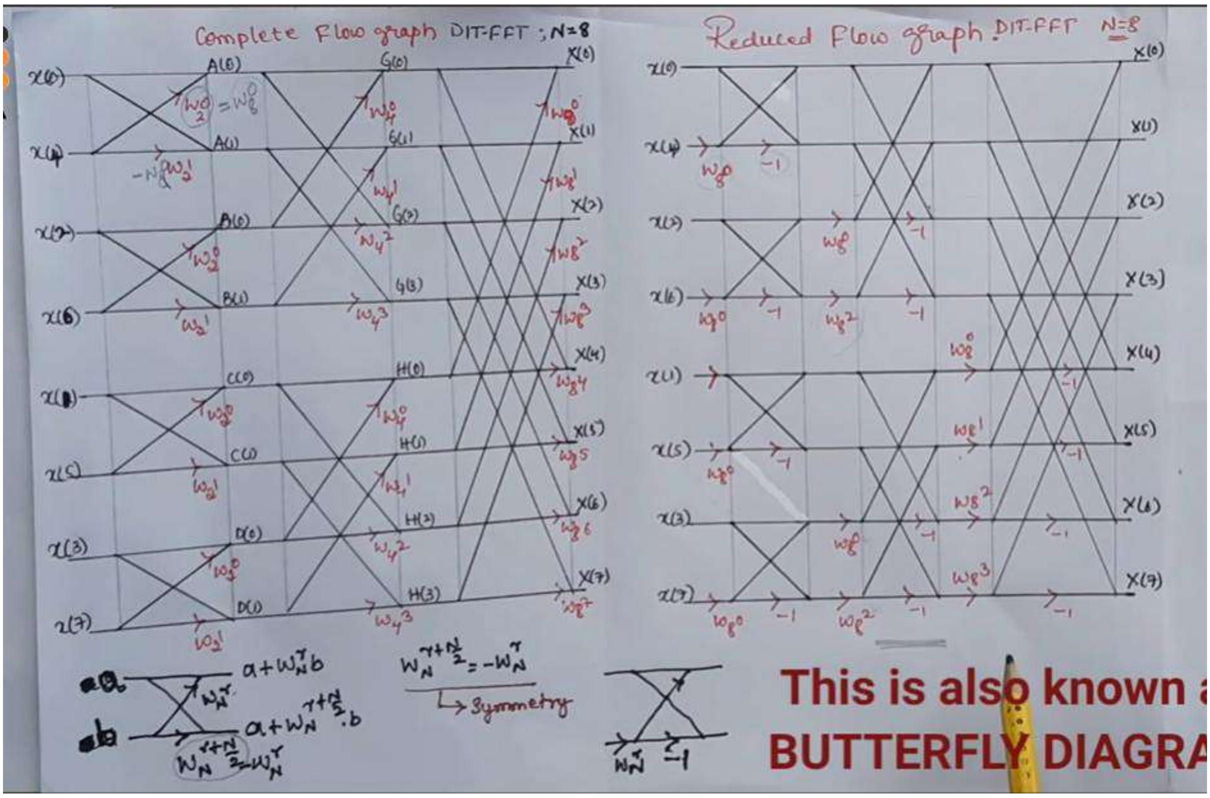
$\boxed{N=8}$

$$A(K) = \sum_{n=0}^{1} x(n)\, W_{2}^{Kn} \; ; \; 0 \le K \le 1$$

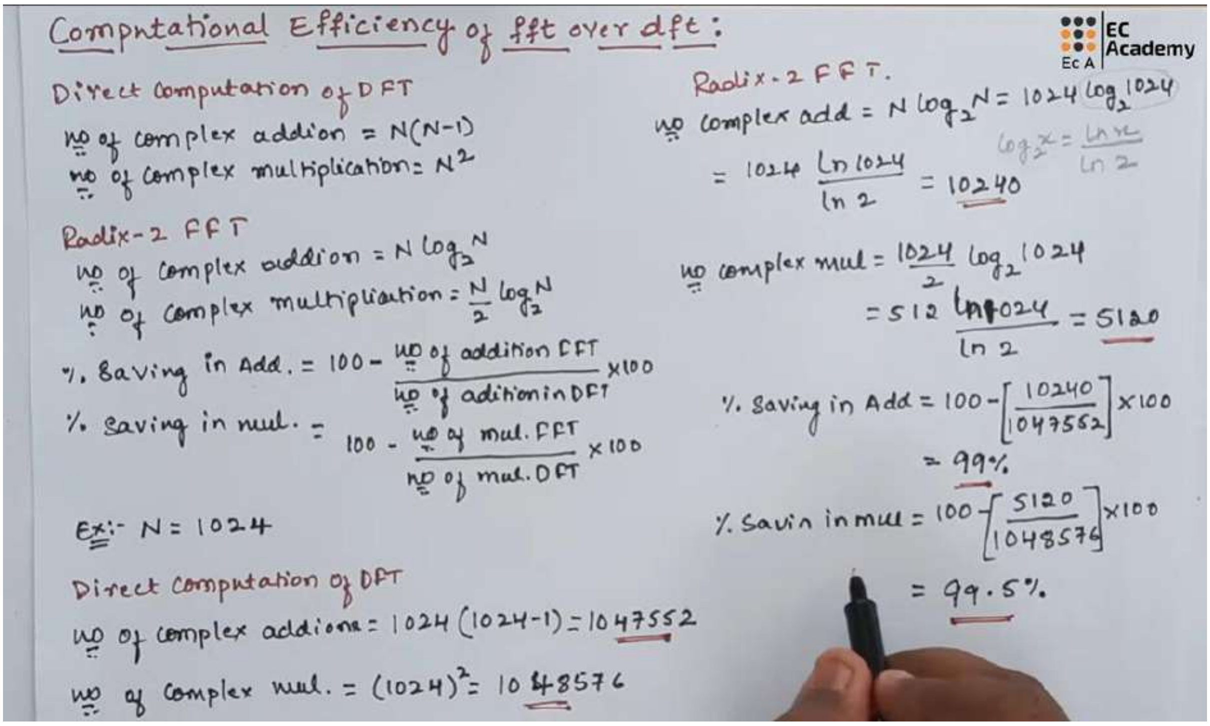
For; $K = 0 \Rightarrow A(0) = x(0) + W_2^0\, x(4)$

for; $K = 1 \Rightarrow A(1) = x(1) + W_2^1\, x(4)$

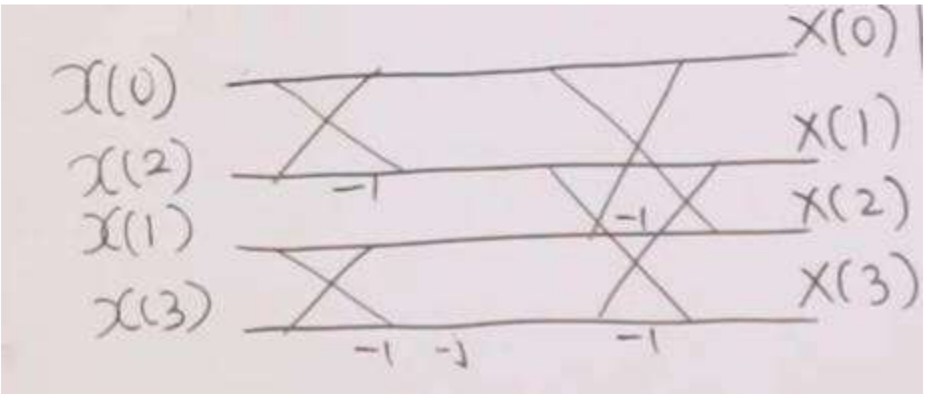


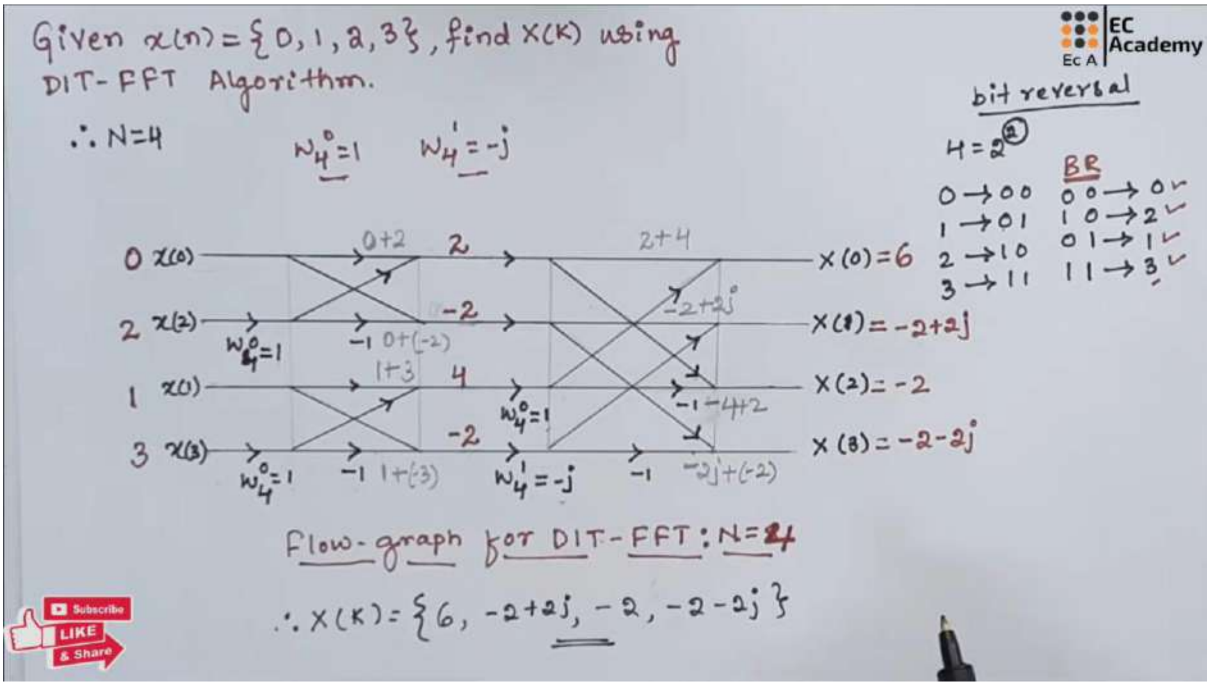


Computational efficiency of DFT over FFT



problem on 4 point DFT using DIT FFT





problem on 8 point DFT using DIT FFT

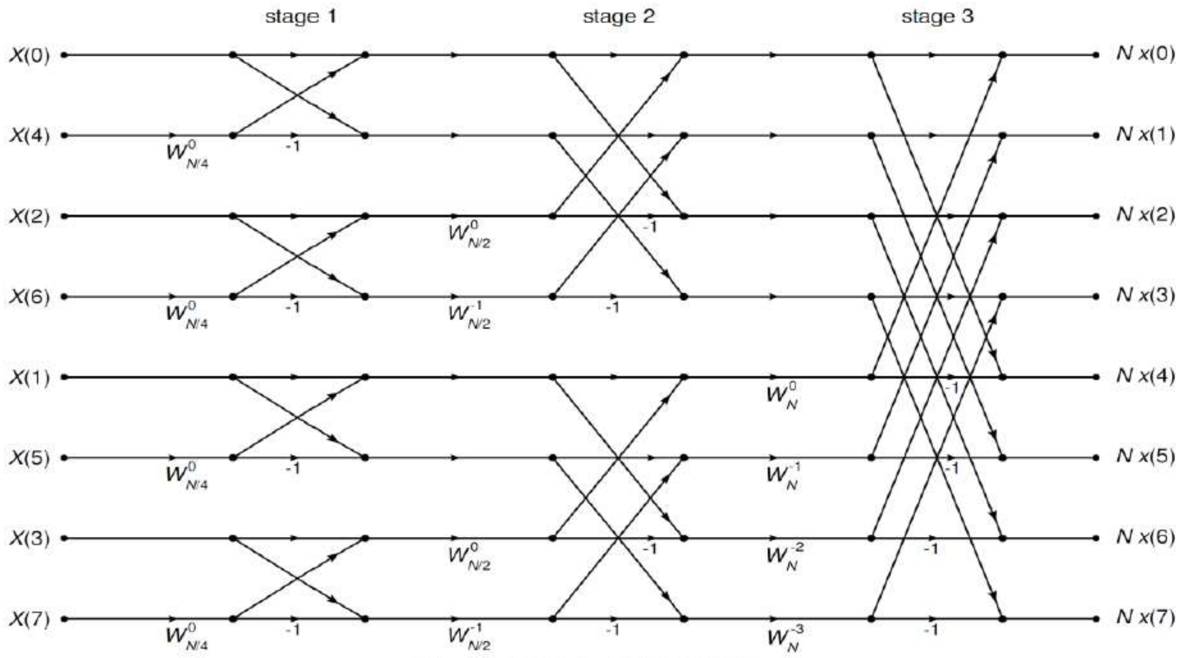
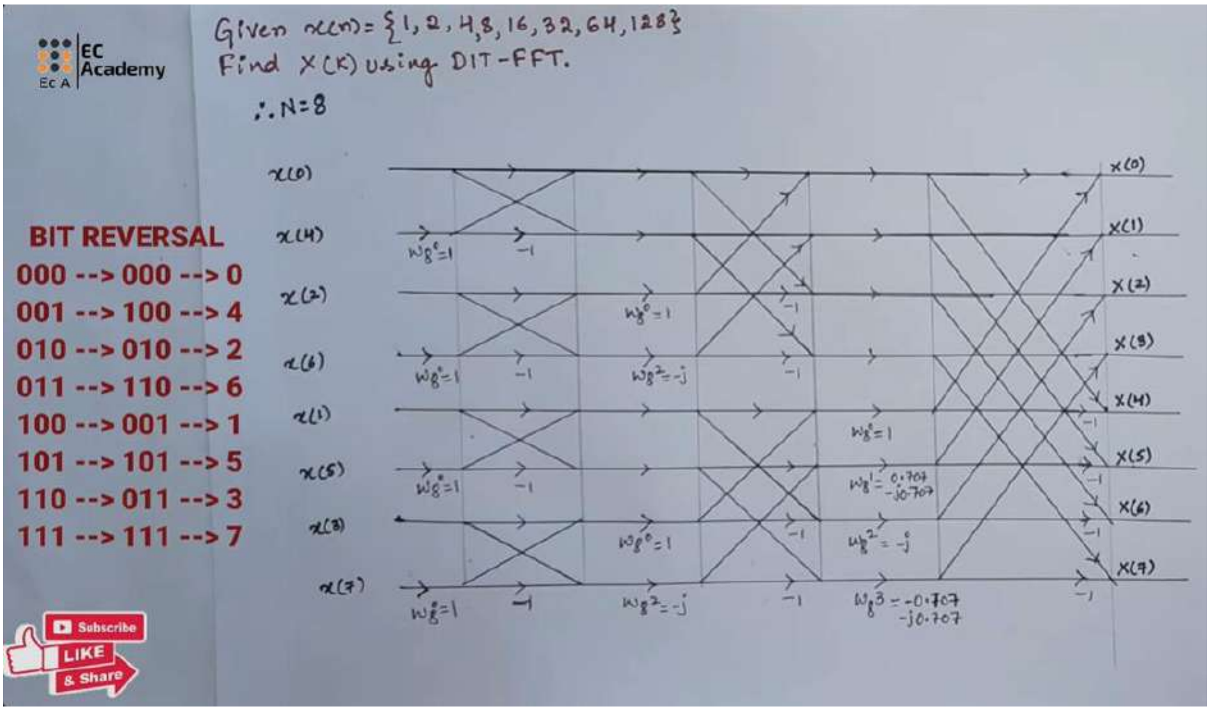


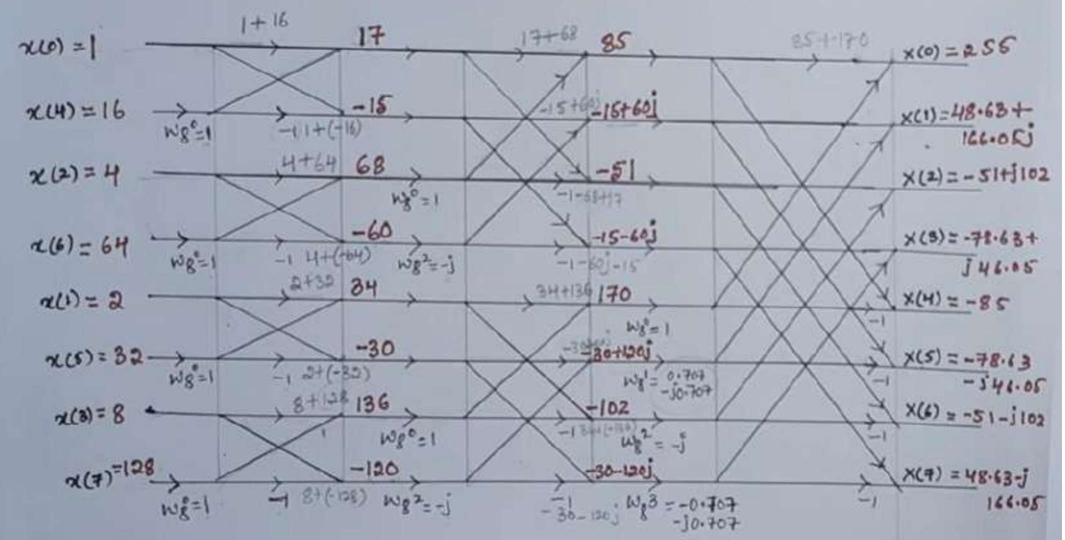
Figure 1: 8 Point DIT-FFT [1]

1. *Butterfly Unit*



Given $x(n) = \{1, 2, 4, 8, 16, 32, 64, 128\}$
Find $X(K)$ using DIT-FFT.

$\therefore N = 8$



$\therefore X(K) = \{256, 48.63+j166.05, -51+j102, -78.63+j46.05, -85, -78.63-j46.05, -51-j102, 48.63-j166.05\}$

Radix 2 DIF FFT algorithm

Radix-2 DIF FFT Algorithm:

DIF → Decimation in Frequency.

$$X(K) = \sum_{n=0}^{N-1} x(n) W_N^{Kn}, \; 0 \le K \le N-1 \longrightarrow (1)$$

$$(1) \Rightarrow X(K) = \sum_{n=0}^{N/2-1} x(n) W_N^{Kn} + \sum_{n=\frac{N}{2}}^{N-1} x(n) W_N^{Kn}. \quad n = n + \frac{N}{2}$$

$$= \sum_{n=0}^{N/2-1} x(n) W_N^{Kn} + \sum_{n=0}^{N/2-1} x\left(n+\frac{N}{2}\right) W_N^{K\left(n+\frac{N}{2}\right)}$$

$$= \sum_{n=0}^{N/2-1} x(n) W_N^{Kn} + W_N^{K\frac{N}{2}} \sum_{n=0}^{N/2-1} x\left(n+\frac{N}{2}\right) W_N^{Kn}$$

$$\because W_N^{K\frac{N}{2}} = (-1)^K$$

$$= \sum_{n=0}^{N/2-1} x(n) W_N^{Kn} + (-1)^K \sum_{n=0}^{N/2-1} x\left(n+\frac{N}{2}\right) W_N^{Kn}$$

$$\therefore X(K) = \sum_{n=0}^{N/2-1} \left[x(n) + (-1)^K x\left(n+\frac{N}{2}\right)\right] W_N^{Kn} \longrightarrow (2)$$

Decompose $X(K)$ as $\underset{K=2r}{\underline{\text{even}}}$ & $\underset{K=(2r+1)}{\underline{\text{odd}}}$ index seq.

$$X(2r) = \sum_{n=0}^{N/2-1} \left[x(n) + (-1)^{2r} x\left(n+\frac{N}{2}\right)\right] W_N^{2rn}$$

$$X(2r) = \sum_{n=0}^{N/2-1} \underbrace{\left[x(n) + x\left(n+\frac{N}{2}\right)\right]}_{g(n)} W_{N/2}^{rn}$$

$$X(2r) = \sum_{n=0}^{N/2-1} g(n) W_{N/2}^{rn} \; ; \; 0 \le r \le \frac{N}{2}-1 \longrightarrow (3)$$

$$X(2r+1) = \sum_{n=0}^{N/2-1} \left[x(n) + (-1)^{2r+1} x\left(n+\frac{N}{2}\right)\right] W_N^{(2r+1)n}$$

$$= \sum_{n=0}^{N/2-1} \underbrace{\left[x(n) - x\left(n+\frac{N}{2}\right)\right]}_{h(n)} W_N^{n} \cdot W_{N/2}^{rn} \longrightarrow (4)$$

$$\therefore X(2r+1) = \sum_{n=0}^{N/2-1} h(n) W_N^{n} \cdot W_{N/2}^{rn}$$

$$\Rightarrow g(n) = x(n) + x\left(n+\frac{N}{2}\right)$$

$$h(n) = x(n) - x\left(n+\frac{N}{2}\right)$$

N = 8

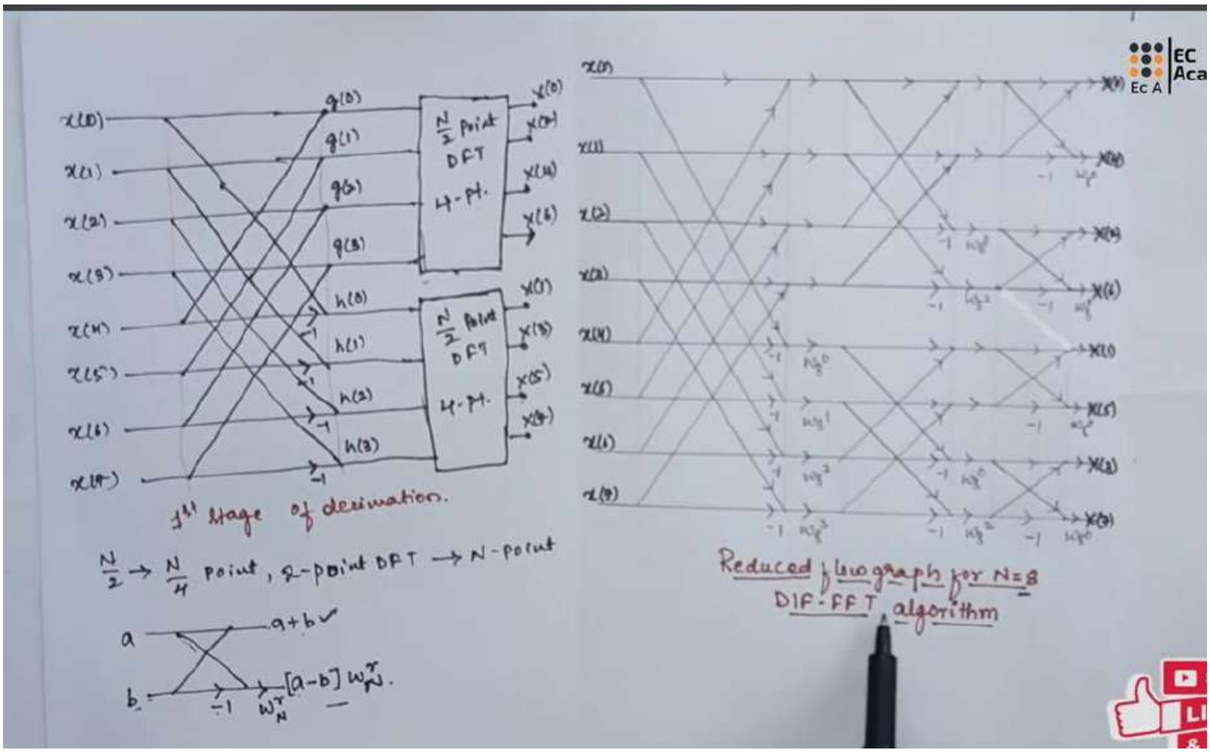
Put $n \to \underline{0 \text{ to } 3}$

$n=0$: $g(0) = x(0) + x(4)$ $\quad h(0) = x(0) - x(4)$

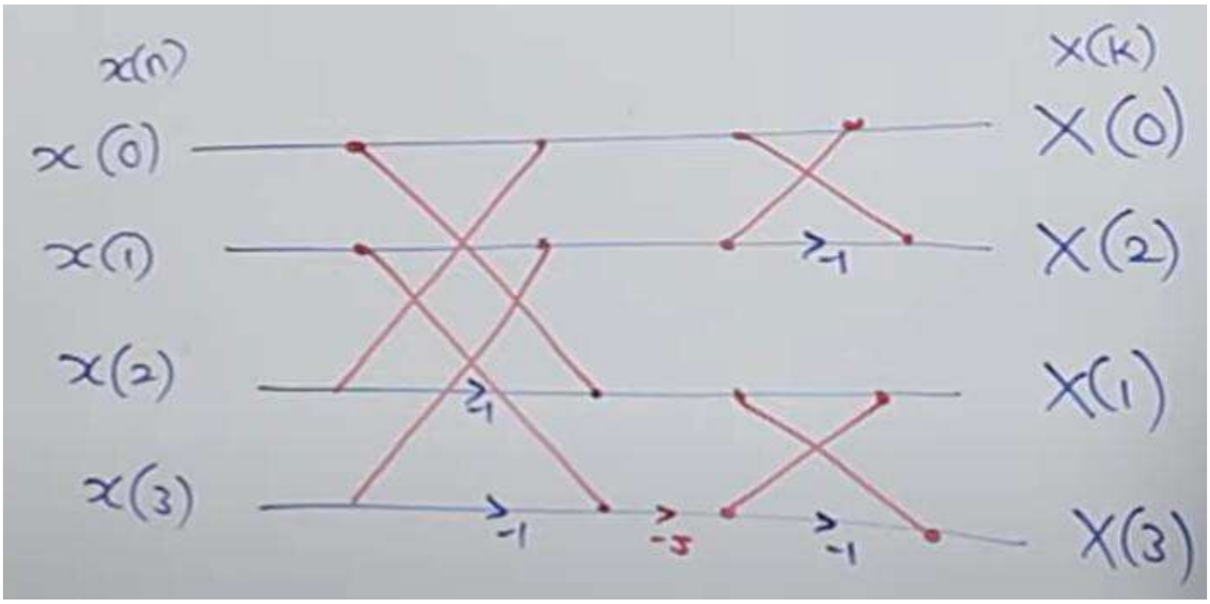
$n=1$: $g(1) = x(1) + x(5)$ $\quad h(1) = x(1) - x(5)$

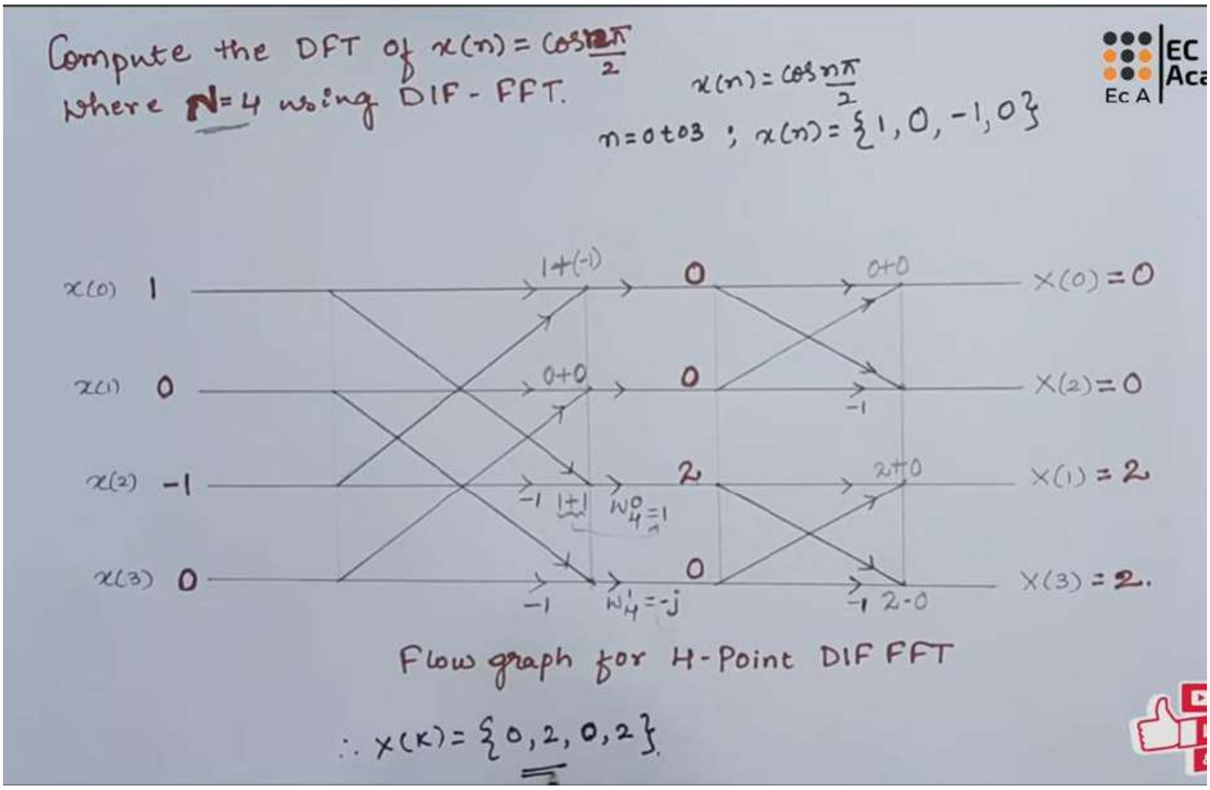
$n=2$: $g(2) = x(2) + x(6)$ $\quad h(2) = x(2) - x(6)$

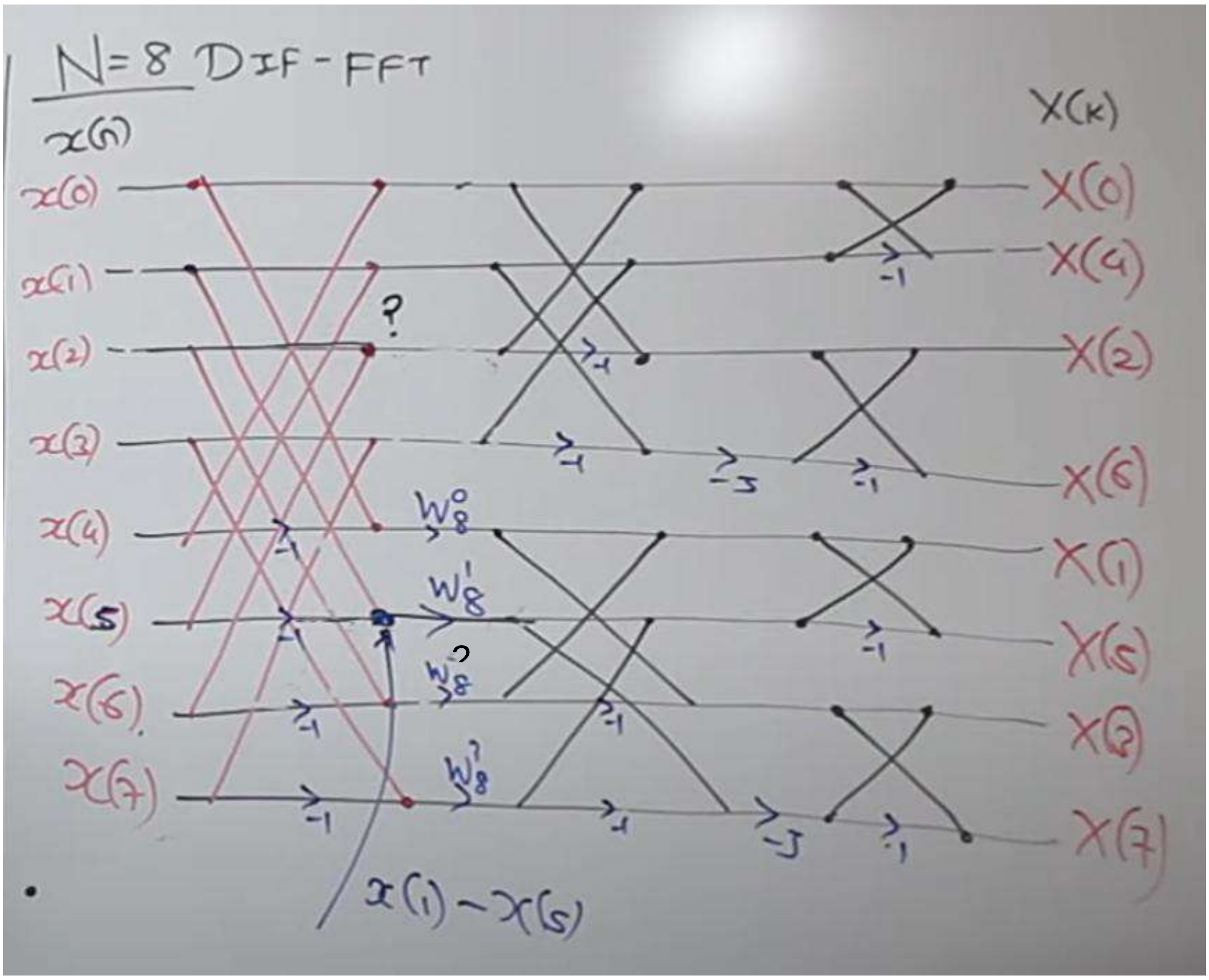
$n=3$: $g(3) = x(3) + x(7)$ $\quad h(3) = x(3) - x(7)$

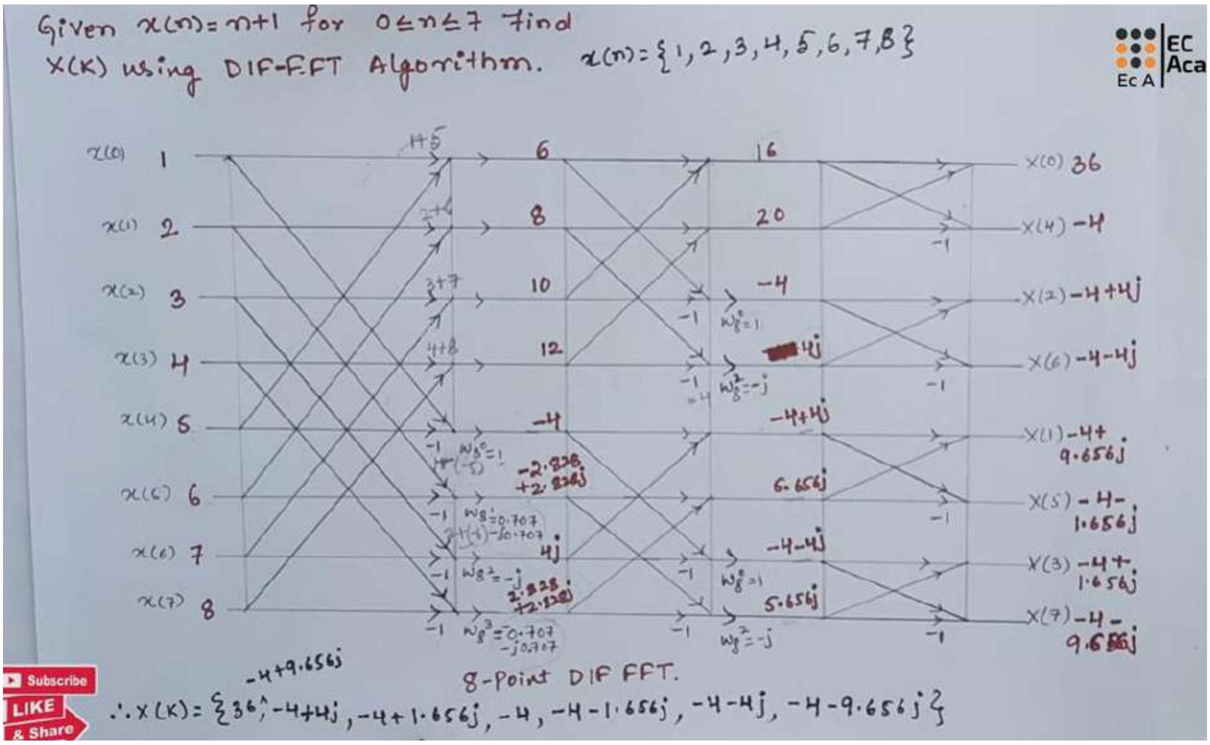


problem on 4 point DFT using DIF FFT

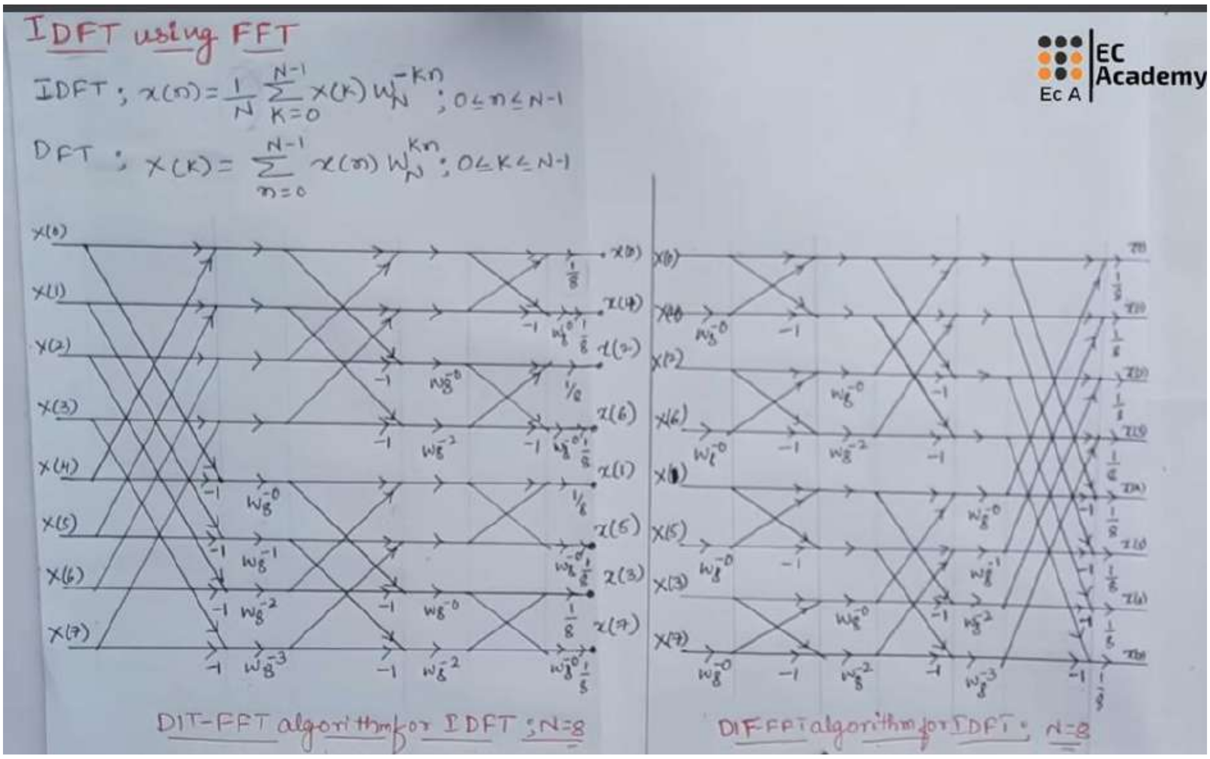



Compute the DFT of x(n) = cos nπ/2
where N=4 using DIF - FFT.
x(n) = cos nπ/2
n=0 to 3 ; x(n) = {1, 0, -1, 0}
x(0) 1
x(1) 0
x(2) -1
x(3) 0
1+(-1)
0+0
0
0
2
0
W_4^0 = 1
W_4^1 = -j
0+0
2+0
2-0
X(0) = 0
X(2) = 0
X(1) = 2
X(3) = 2.
Flow graph for 4-Point DIF FFT
∴ X(K) = {0, 2, 0, 2}

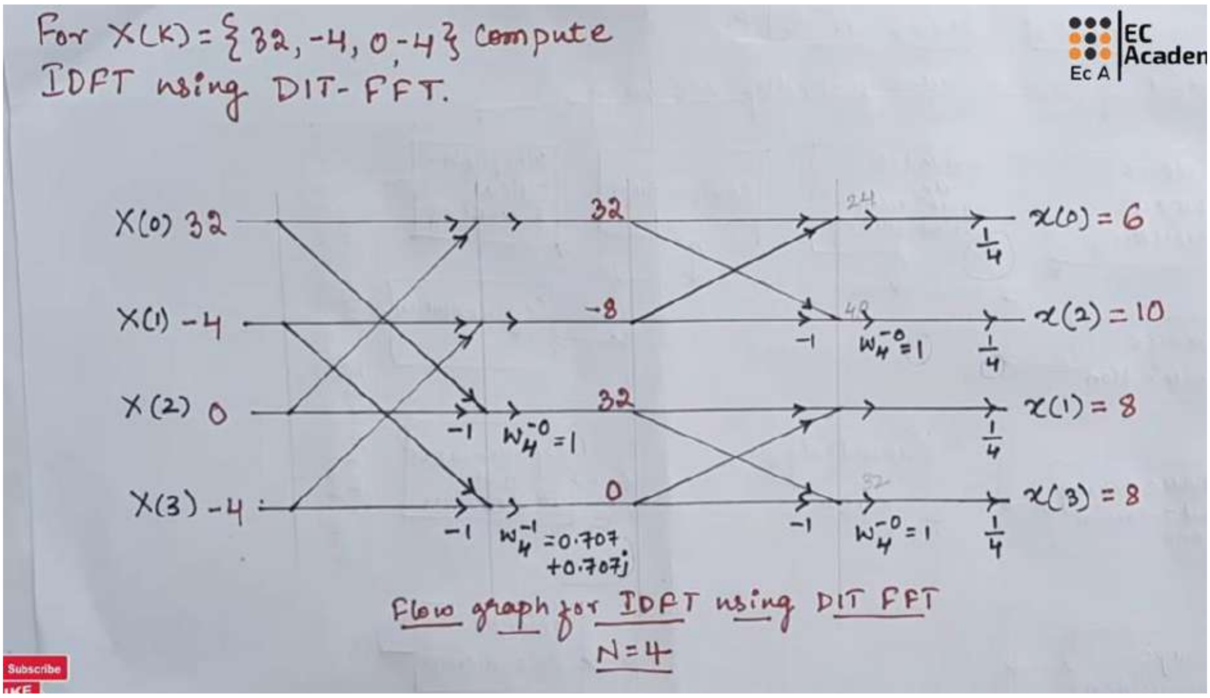

N = 8 DIF - FFT
x(n)
X(k)
x(0)
x(1)
x(2)
x(3)
x(4)
x(5)
x(6)
x(7)
W_8^0
W_8^1
W_8^2
W_8^3
X(0)
X(4)
X(2)
X(6)
X(1)
X(5)
X(3)
X(7)
x(1) − x(5)



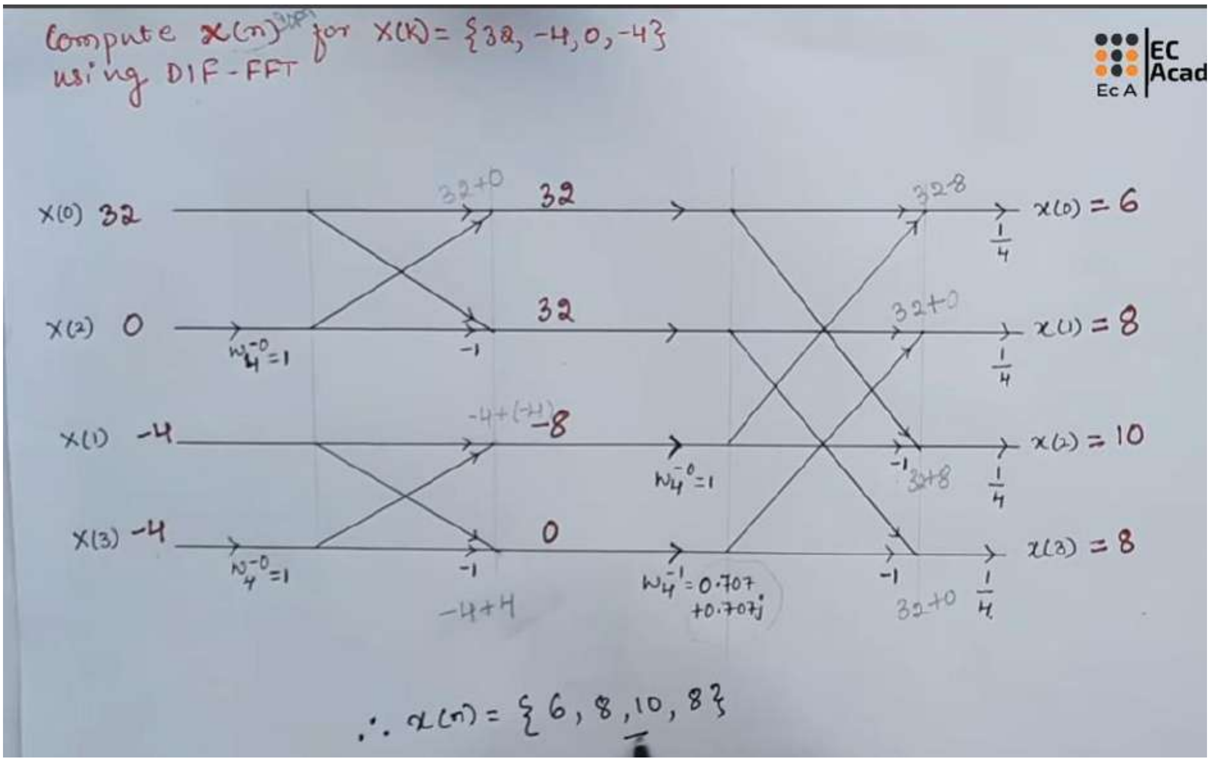
IDFT using FFT



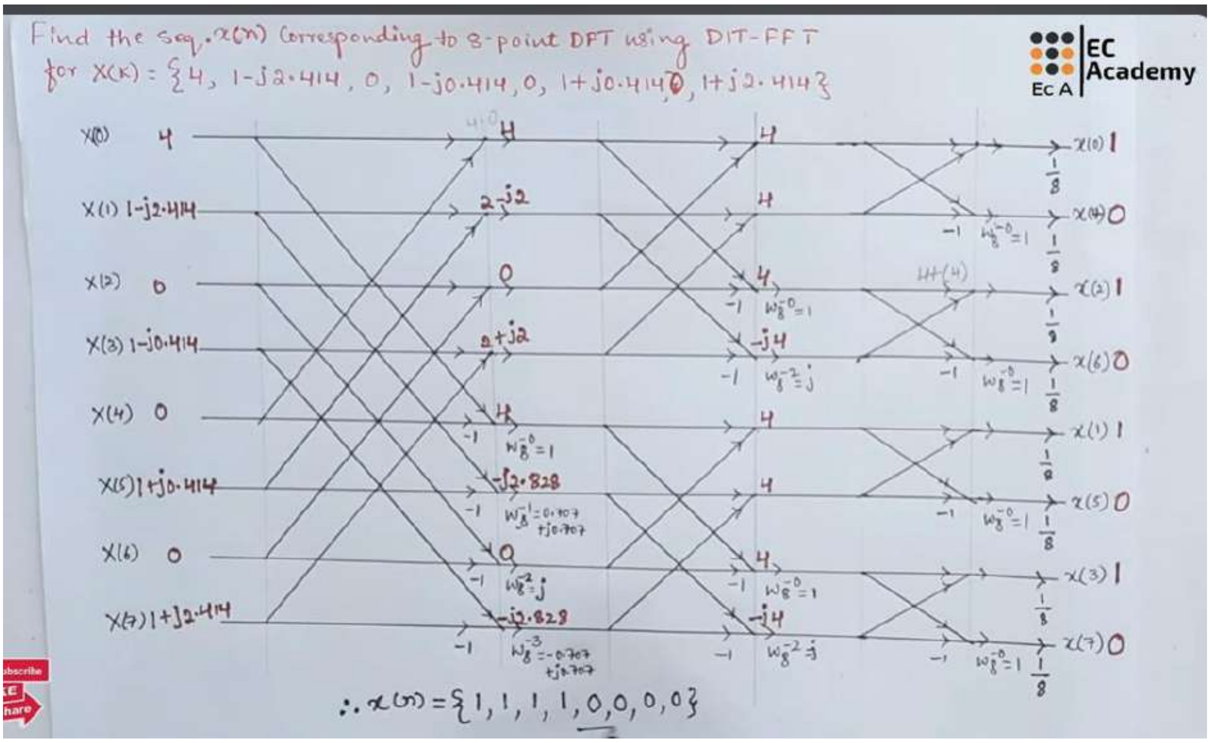
problem on 4 point IDFT using DIT FFT



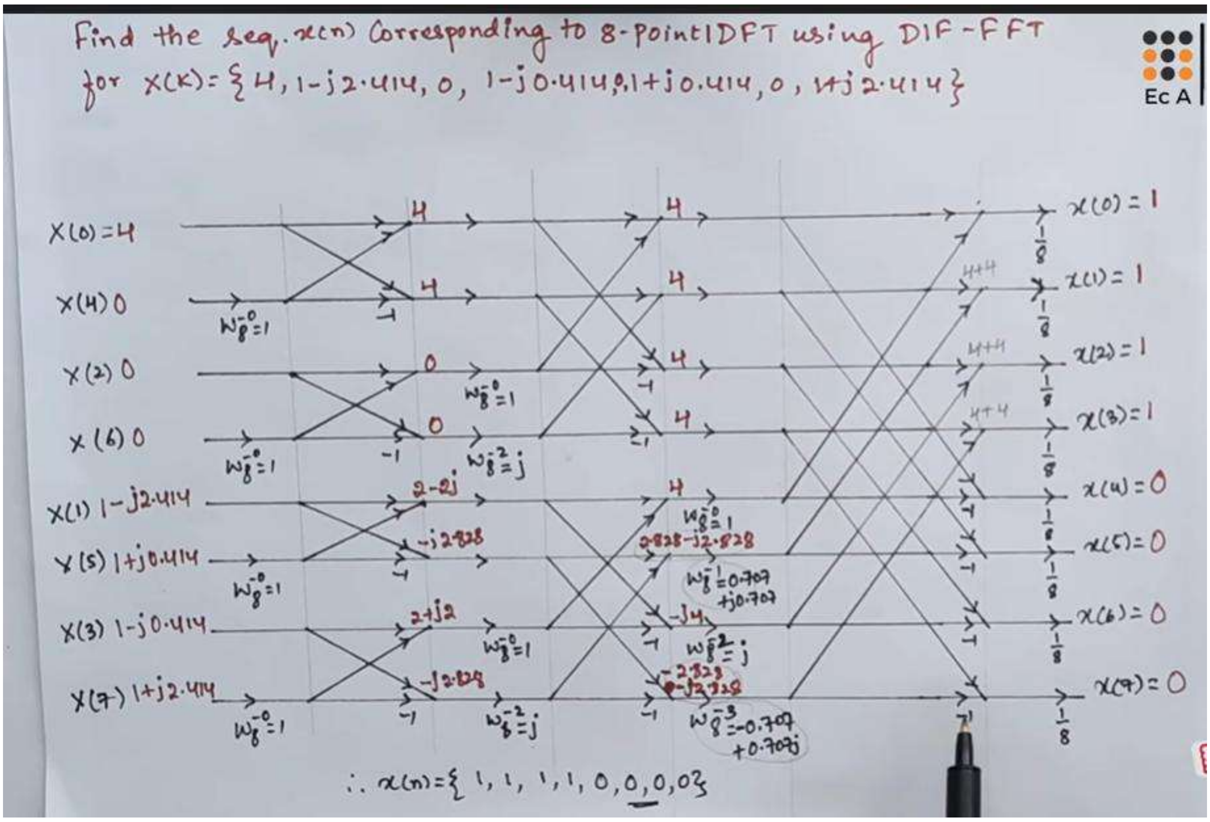
problem on 4 point IDFT using DIF FFT



problem on 8 point IDFT using DIT FFT



problem on 8 point IDFT using DIF FFT



introduction to FIR filter

Introduction – Design of FIR Filter

EC Academy

→ Filter → Impulse response is finite.

→ output → depends only on present and (or) past values.

→ Applications → where linear phase is important.

Ex:– Data transmission, Speech processing, Correlation processing, Interpolation.

Characteristics:

(i) Impulse response → Finite length.

(ii) non-recursive FIR filter → stable.

(iii) phase distortion of Frq response can be eliminated by FIR filter.

(iv) Implement a recursive FIR Filters

(v) Effect of Start-up transient have small duration.

(vi) Quantization noise can be made negligible

Advantages:

→ Stable

→ Can be realized in both recursive & non-recursive.

→ exact linear phase.

→ Flexible.

→ low sensitive to Quantization noise.

→ Efficiently realized in H/w.

Disadvantages:

→ complex

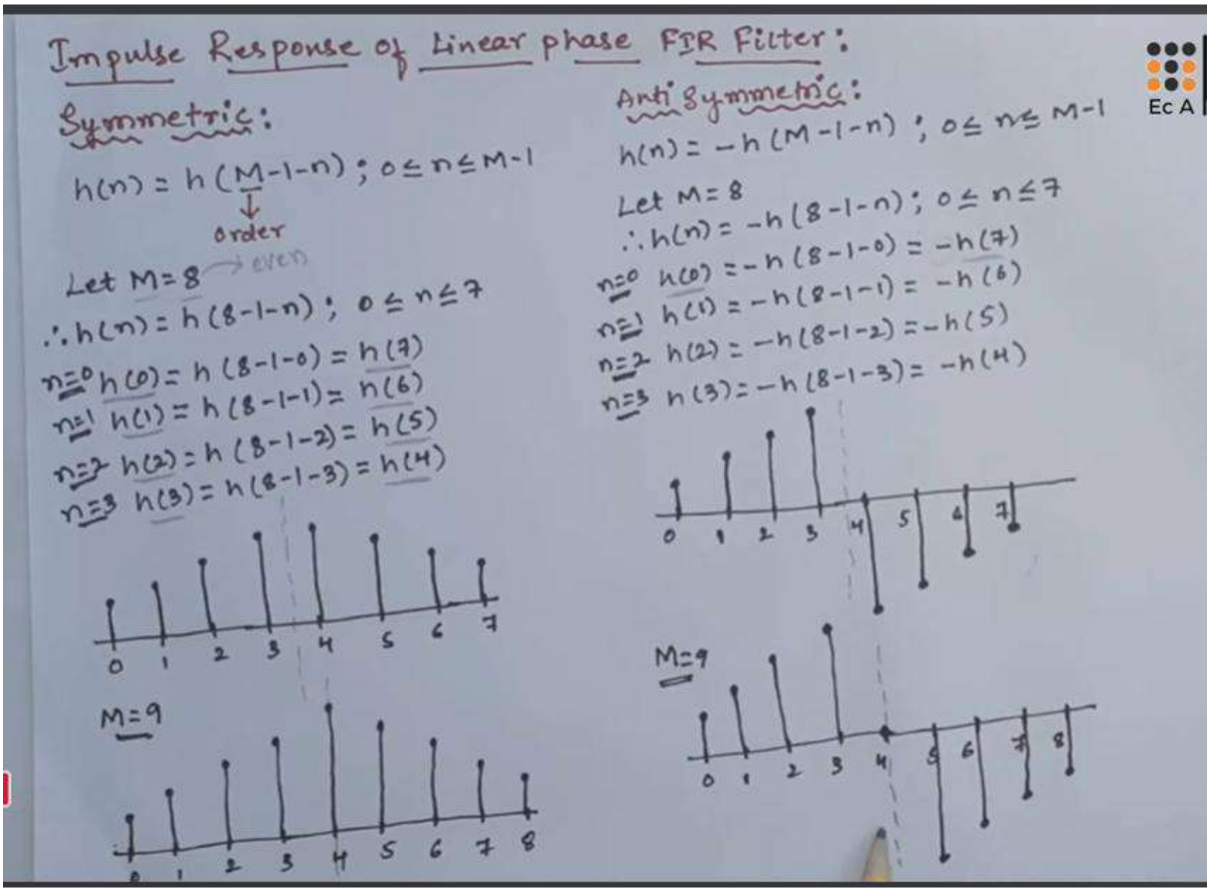
→ requires more filter co-efficients to be stored.

→ long duration impulse response require large amount of processing.

→ narrow transition band FIR Filter requires more arithmetic operations & H/w components

↳ Costly.

Impulse response of linear phase FIR filters



Impulse Response of Linear phase FIR Filter:

Symmetric:

$h(n) = h(\underbrace{M}_{\text{order}}-1-n)\ ;\ 0 \le n \le M-1$

Let $M = 8 \rightarrow$ even

$\therefore h(n) = h(8-1-n)\ ;\ 0 \le n \le 7$

$n=0\quad h(0) = h(8-1-0) = h(7)$

$n=1\quad h(1) = h(8-1-1) = h(6)$

$n=2\quad h(2) = h(8-1-2) = h(5)$

$n=3\quad h(3) = h(8-1-3) = h(4)$

0 1 2 3 4 5 6 7

M=9

0 1 2 3 4 5 6 7 8

Anti Symmetric:

$h(n) = -h(M-1-n)\ ;\ 0 \le n \le M-1$

Let $M = 8$

$\therefore h(n) = -h(8-1-n)\ ;\ 0 \le n \le 7$

$n=0\quad h(0) = -h(8-1-0) = -h(7)$

$n=1\quad h(1) = -h(8-1-1) = -h(6)$

$n=2\quad h(2) = -h(8-1-2) = -h(5)$

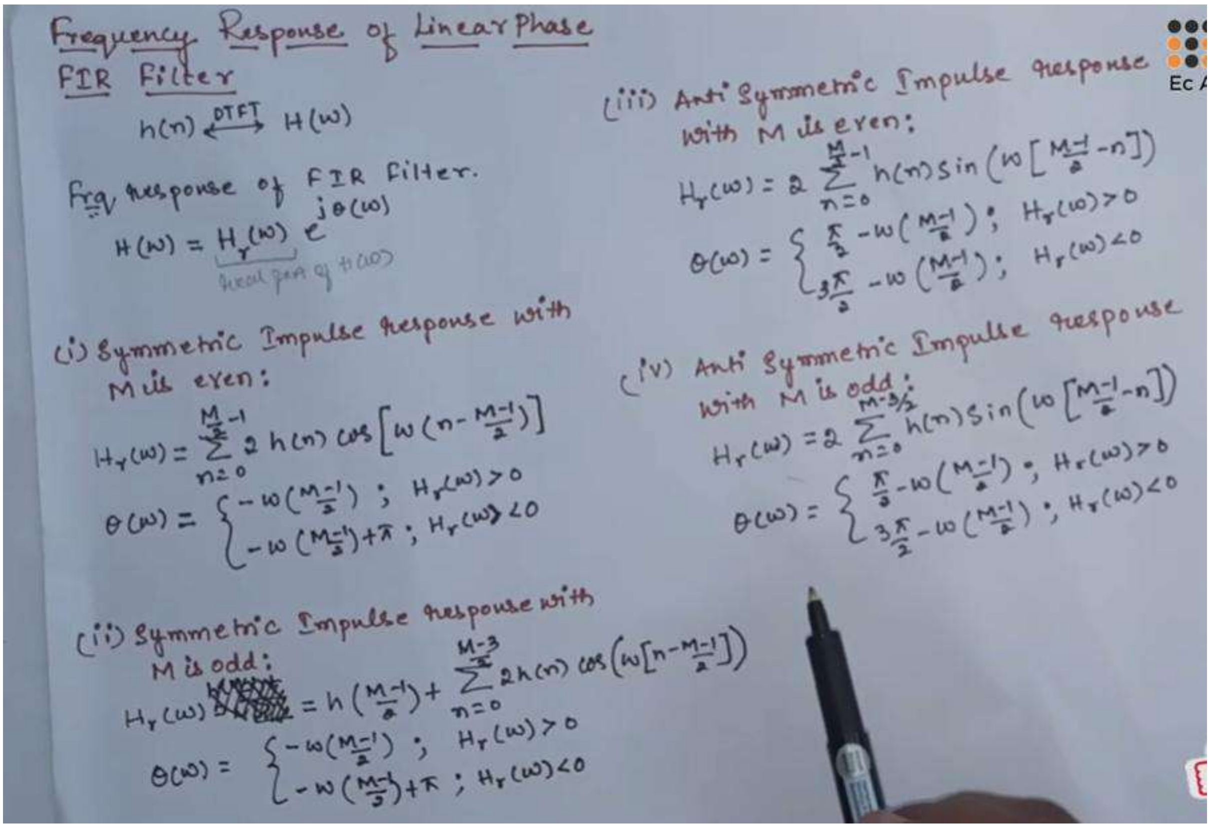
$n=3\quad h(3) = -h(8-1-3) = -h(4)$

0 1 2 3 4 5 6 7

M=9

0 1 2 3 4 5 6 7 8

Frequency response of linear phase FIR filter

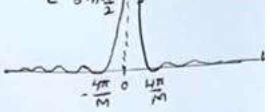


Different types of windows to design linear phase FIR filter

Design of Linear phase FIR filter using window method

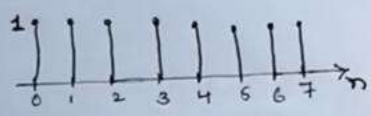
(i) Different types of windows

$W_T(e^{j\omega}) = \left[\frac{\sin\left(\frac{M-1}{4}\right)\omega}{\sin\frac{\omega}{2}}\right]^2$

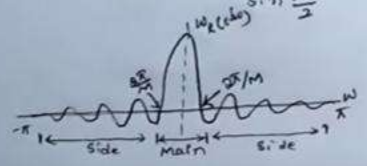


ⓐ Rectangular window

$W_R(n) = 1 \; ; \; 0 \le n \le M-1$
$\quad = 0 \; ; \; \text{otherwise}.$



Spectrum: $W_R(e^{j\omega}) = \frac{\sin\frac{\omega M}{2}}{\sin\frac{\omega}{2}}$

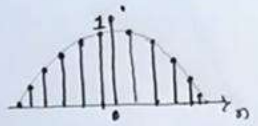


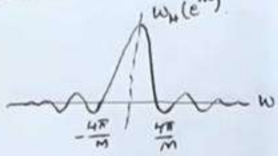
ⓑ Bartlett window [Triangular]

$W_T(n) = 1 - \frac{2\left|n - \left(\frac{M-1}{2}\right)\right|}{M-1} \; ; \; 0 \le n \le M-1$
$\quad = 0 \; ; \; \text{otherwise}$

ⓒ Hanning window

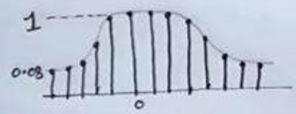
$W_H(n) = 0.5 - 0.5\cos\frac{2\pi n}{M-1} \; , \; 0 \le n \le M-1$
$\quad = 0 \; ; \; \text{otherwise}$

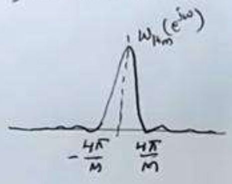




ⓓ Hamming window

$W_{Hm}(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{M-1}\right) \; ; \; 0 \le n \le M-1$
$\quad = 0 \; ; \; \text{otherwise}.$





Procedure to design linear phase FIR filter

Procedure to design Linear phase FIR Filters using windows.

$H_d(\omega) \rightarrow$ desired freq response

$H_d(\omega) \rightarrow$ F.T. of $h_d(n)$

$\therefore H_d(\omega) = \sum_{n=0}^{\infty} h_d(n) e^{-j\omega n} \longrightarrow ①$

$h_d(n) \rightarrow$ Inverse F.T. of $H_d(\omega)$

$\therefore h_d(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d(\omega) e^{j\omega n} \cdot d\omega \longrightarrow ②$

infinite duration.

$h_d(n) \rightarrow$ finite $\rightarrow$ multiply $h_d(n)$ by window sequence. 'M'

Lenth M

Example: Rectangular window.

$W_R(n) = \begin{cases} 1 \; ; \; n = 0,1,2,\ldots M-1 \\ 0 \; ; \; \text{otherwise}. \end{cases} \longrightarrow ③$

$h_d(n) \rightarrow$ Sample response $\rightarrow$ Infinite

$\therefore h(n) = h_d(n) W_R(n) \longrightarrow ④$

$h_d(n) \rightarrow$ desired sample response.

finite

$\therefore h(n) = \underbrace{\begin{cases} h_d(n) \; ; \; n = 0,1,2\ldots M-1 \\ 0 \; ; \; \text{otherwise} \end{cases}}_{\text{Windowing}} \longrightarrow ⑤$

④ $\Rightarrow h(n) = h_d(n) \cdot W_R(n)$

generally.

$h(n) = h_d(n) \cdot \underbrace{W(n)}_{\text{generalized window}} \rightarrow$ unit sample response FIR.

Freq response,

$H(\omega) = \text{F.T.}\{h_d(n) \cdot W(n)\}$

$H(\omega) = H_d(\omega) * W(\omega) \longrightarrow ⑥$

Problem on FIR filter using Rectangular window

Design the Symmetric FIR lowpass Filter whose

$$H_d(w) = \begin{cases} e^{-jw\tau} & ; |W| \le W_c \\ 0 & ; \text{otherwise} \end{cases}$$

with $M=7$ & $W_c = 1$ rad/Sam.

use Rectangular Window.

(i) obtain $h_d(n)$:

$$h_d(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d(w)\, e^{jwn}\, dw \longrightarrow (1)$$

$$H_d(w) = \begin{cases} e^{-jw\tau} & ; -1 \le w \le 1 \\ 0 & ; \text{Otherwise.} \end{cases} \longrightarrow (2)$$

$$\therefore h_d(n) = \frac{1}{2\pi}\int_{-1}^{1} e^{-jw\tau} \cdot e^{jwn}\, dw = \frac{1}{2\pi}\underline{\int_{-1}^{1} e^{jw(n-\tau)}\, dw}$$

$$= \frac{1}{2\pi}\left[\frac{e^{jw(n-\tau)}}{j(n-\tau)}\right]_{-1}^{1} = \frac{1}{2\pi}\left[\frac{e^{j(n-\tau)} - e^{-j(n-\tau)}}{j(n-\tau)}\right]$$

$$= \frac{1}{\pi(n-\tau)}\left[\frac{e^{j(n-\tau)} - e^{-j(n-\tau)}}{2j}\right] \qquad \because \sin\theta = \frac{e^{j\theta} - e^{-j\theta}}{2j}$$

$$h_d(n) = \frac{\sin(n-\tau)}{\pi(n-\tau)} \qquad n \ne \tau \longrightarrow (3)$$

if $n = \tau$

$$h_d(n) = \frac{1}{2\pi}\int_{-1}^{1} 1 \cdot dw = \frac{1}{2\pi}[2] = \frac{1}{\pi} \longrightarrow (4)$$

$$h_d(n) = \begin{cases} \dfrac{\sin(n-\tau)}{\pi(n-\tau)} & ; n \ne \tau \\ \dfrac{1}{\pi} & ; n = \tau \end{cases} \longrightarrow (5)$$

determine the Value of τ

$$h(n) = h(M-1-n)$$

$$\because h(n) = h_d(n) \cdot w(n)$$

$$h_d(n)\, w(n) = h_d(M-1-n)\, w(n)$$

$$h_d(n) = h_d(M-1-n)$$

$$\frac{\sin(n-\tau)}{\pi(n-\tau)} = \frac{\sin(M-1-n-\tau)}{\pi(M-1-n-\tau)}$$

If we don't multiply numerator and denominator in the first step by -1 we won't get the value.

$$\frac{-\sin(n-\tau)}{-\pi(n-\tau)} = \frac{\sin(M-1-n-\tau)}{\pi(M-1-n-\tau)} \qquad \because -\sin\theta = \sin(-\theta)$$

$$\frac{\sin[-(n-\tau)]}{\pi(-(n-\tau))} = \frac{\sin(M-1-n-\tau)}{\pi(M-1-n-\tau)}$$

$$-(n-\tau) = M-1-n-\tau$$

$$-\cancel{n}+\tau = M-1-\cancel{n}-\tau$$

$$2\tau = M-1$$

$$\boxed{\tau = \frac{M-1}{2}}$$

$$(5) \Rightarrow h_d(n) = \begin{cases} \dfrac{\sin\left(n - \frac{M-1}{2}\right)}{\pi\left(n - \frac{M-1}{2}\right)} & ; n \ne \dfrac{M-1}{2} \\ \dfrac{1}{\pi} & ; n = \dfrac{M-1}{3} \end{cases}$$

$\because M = 7$

$$h_d(n) = \begin{cases} \dfrac{\sin(n-3)}{\pi(n-3)} & ; n \ne 3 \\ \dfrac{1}{\pi} & ; \underline{n=3} \end{cases}$$

Put $n = 0$ to 6

$n=0$; $h_d(0) = 0.01497$

$n=1$; $h_d(1) = 0.14472$

$n=2$; $h_d(2) = 0.26785$

$n=3$; $h_d(3) = \frac{1}{\pi}$

$n=4$; $h_d(4) = 0.26785$

$n=5$; $h_d(5) = 0.14472$

$n=6$; $h_d(6) = 0.01497$

$\because h(n) = h_d(n) \cdot \underline{w(n)}$ $\qquad w_n(n) = \begin{cases} 1 & ; 0 \le n \le 6 \\ 0 & ; \text{other} \end{cases}$

$$h(n) = \begin{cases} h_d(n) & ; 0 \le n \le 6 \\ 0 & ; \text{otherwise} \end{cases}$$

Co-efficients of FIR Filter

→ $h(0) = 0.01497$

→ $h(1) = 0.14472$

→ $h(2) = 0.26785$

$h(3) = \frac{1}{\pi}$

→ $h(4) = 0.26785$

→ $h(5) = 0.14472$

→ $h(6) = 0.01497$

Symmetric $h(n) = h(6-n)$

Problem on FIR filter using Hanning window

Design the Symmetric FIR lowpass filter whose

$$H_d(\omega) = \begin{cases} e^{-j\omega n} ; & |\omega| \le \omega_c \\ 0 ; & \text{Otherwise} \end{cases}$$ with $M=7$ & $\omega_c = 1\,\text{rad/sam}$.

Use Hanning window.

$n=0$; $h_d(0) = 0.01497$
$n=1$; $h_d(1) = 0.14472$
$n=2$; $h_d(2) = 0.26785$
$n=3$; $h_d(3) = \frac{1}{\pi}$
$n=4$; $h_d(4) = 0.26785$
$n=5$; $h_d(5) = 0.14472$
$n=6$; $h_d(6) = 0.1497$

Hanning window $W(n) = 0.5\left[1-\cos\left(\frac{2\pi n}{M-1}\right)\right]$

$\because M=7$ $\quad W(n) = 0.5\left[1-\cos\left(\frac{2\pi n}{6}\right)\right]$

$\Rightarrow W(n) = 0.5\left[1-\cos\left(\frac{\pi n}{3}\right)\right]$ $\quad n = 0$ to 6 $\quad M=7$

Rad $\quad 0.5 \times (1-\cos(\pi A \div 3)$

$n=0$; $w(0) = 0$
$n=1$; $w(1) = 0.25$
$n=2$; $w(2) = 0.75$
$n=3$; $w(3) = 1$
$n=4$; $w(4) = 0.75$
$n=5$; $w(5) = 0.25$
$n=6$; $w(6) = 0$

$h(n) = h_d(n) \cdot w(n)$

$\therefore h(0) = 0$
$h(1) = 0.03618$
$h(2) = 0.20089$
$h(3) = \frac{1}{\pi}$
$h(4) = 0.20089$
$h(5) = 0.03618$
$h(6) = 0$

This sum is almost same as the previous one, just the notable difference is in the concluding steps.

Problem on FIR filter using hamming window

Determine the filter coefficients $h_d(n)$ for desired frq. response of a low pass filter given by,

$$H_d(e^{jw}) = \begin{cases} e^{-j2w}; & -\frac{\pi}{4} \le w \le \frac{\pi}{4} \\ 0; & \frac{\pi}{4} \le |w| \le \pi \end{cases}$$

If we define new filter co-efficients by $h_d(n)\cdot w(n)$ where $w(n) = \begin{cases} 1 : & 0 \le n \le 4 \\ 0 ; & \text{elsewhere} \end{cases}$

Determine $h(n)$ and also the frq. response $H(e^{jw})$ and compare with $H_d(e^{jw})$. Determine $H(e^{jw})$ using the Hamming window.

(i) obtain $h_d(n)$:

$$h_d(n) = \frac{1}{2\pi}\int_{-2\pi}^{\pi} H_d(e^{jw}) e^{jwn}\, dw = \frac{1}{2\pi}\int_{-\frac{\pi}{4}}^{\frac{\pi}{4}} e^{-j2w}\cdot e^{jwn}\, dw$$

$$= \frac{1}{2\pi}\int_{-\pi/4}^{\pi/4} e^{jw(n-2)}\, dw = \frac{1}{2\pi}\left[\frac{e^{jw(n-2)}}{j(n-2)}\right]_{-\frac{\pi}{4}}^{\frac{\pi}{4}}$$

$$= \frac{1}{2\pi}\left[\frac{e^{j\frac{\pi}{4}(n-2)} - e^{-j\frac{\pi}{4}(n-2)}}{j(n-2)}\right] = \frac{1}{\pi(n-2)}\left[\frac{e^{j\frac{\pi}{4}(n-2)} - e^{-j\frac{\pi}{4}(n-2)}}{2j}\right]$$

$$h_d(n) = \frac{\sin\frac{\pi}{4}(n-2)}{\pi(n-2)}\ ;\ n \ne 2$$

if $n=2$

$$h_d(n) = \frac{1}{2\pi}\int_{-\pi/4}^{\pi/4} 1\, dw = \frac{1}{2\pi}\left[\frac{2\pi}{4}\right] = \frac{1}{4}$$

$$\therefore h_d(n) = \begin{cases} \dfrac{\sin\frac{\pi}{4}(n-2)}{\pi(n-2)} ; & n \ne 2 \\ \dfrac{1}{4} ; & n = 2 \end{cases}$$

(ii) obtain $h(n)$:

$h(n) = h_d(n)\cdot w(n)\ ;\ 0 \le n \le 4$

$n=0;\ h(0) = h_d(0) = 0.159091$

$n=1;\ h(1) = h_d(1) = 0.224989$

$n=2;\ h(2) = h_d(2) = \frac{1}{4}$

$n=3;\ h(3) = h_d(3) = 0.224989$

$n=4;\ h(4) = h_d(4) = 0.159091$

(iii) obtain $H(e^{jw})$:

$\because M = 5\ ;\ 0 \le n \le 4$

↳ odd.

$$\therefore H(w) = e^{-jw\left(\frac{M-1}{2}\right)}\left\{h\left(\frac{M-1}{2}\right) + 2\sum_{n=0}^{\frac{M-3}{2}} h(n)\cos w\left(n - \frac{M-1}{2}\right)\right\}$$

$$H(w) = e^{-j2w}\left[h(2) + 2\sum_{n=0}^{1} h(n)\cos w(n-2)\right]$$

$$= e^{-j2w}\left[h(2) + 2h(0)\cos w(-2) + 2h(1)\cos w(-1)\right]$$

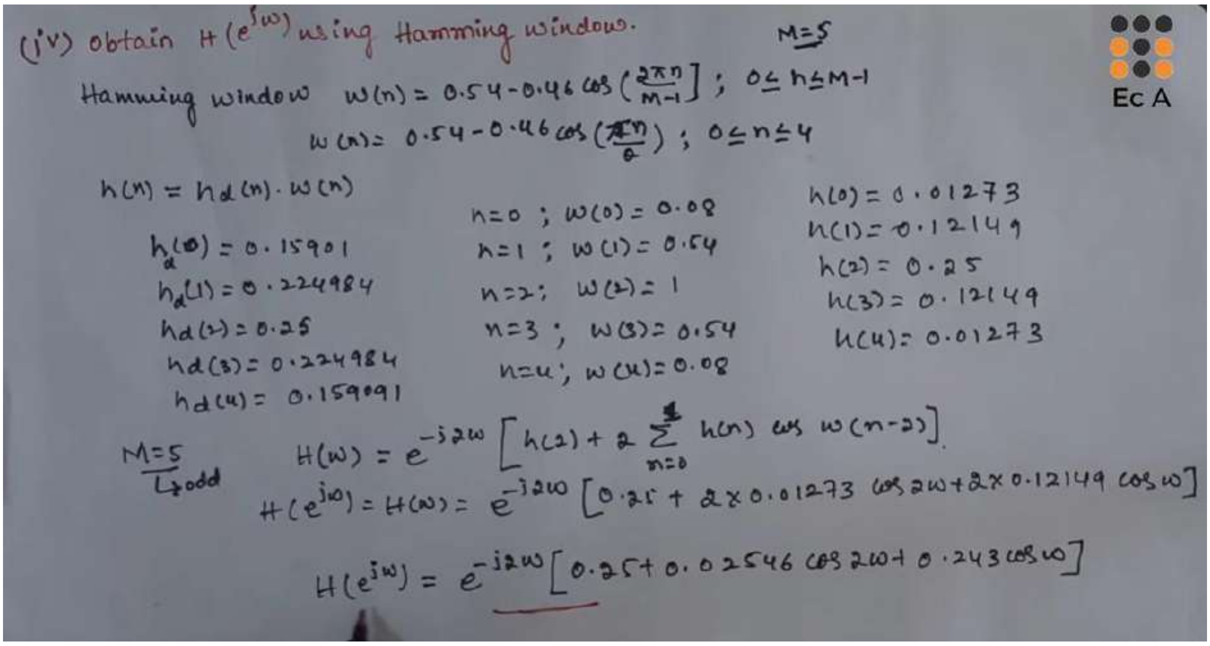
$$= e^{-j2w}\left[0.25 + 2\times 0.159091\cos 2w + 2\times 0.224989\cos w\right]$$

$$= e^{-j2w}\left[0.25 + 0.318\cos 2w + 0.45\cos w\right]$$

$$\boxed{|H(w)| = 0.25 + 0.318\cos 2w + 0.45\cos w}$$

$$\angle H(w) = \begin{cases} -2w ; & |H(w)| > 0 \\ -2w + \pi ; & |H(w)| < 0 \end{cases}$$

$H(e^{jw})$ with $H_d(e^{jw}) \rightarrow$ different & will not same.



(iv) obtain $H(e^{j\omega})$ using Hamming window. $M=5$

Hamming window $w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{M-1}\right)$; $0 \le n \le M-1$

$w(n) = 0.54 - 0.46\cos\left(\frac{\pi n}{2}\right)$; $0 \le n \le 4$

$h(n) = h_d(n) \cdot w(n)$

$h_d(0) = 0.15901$
$h_d(1) = 0.224984$
$h_d(2) = 0.25$
$h_d(3) = 0.224984$
$h_d(4) = 0.159091$

$n=0$; $w(0) = 0.08$
$n=1$; $w(1) = 0.54$
$n=2$; $w(2) = 1$
$n=3$; $w(3) = 0.54$
$n=4$; $w(4) = 0.08$

$h(0) = 0.01273$
$h(1) = 0.12149$
$h(2) = 0.25$
$h(3) = 0.12149$
$h(4) = 0.01273$

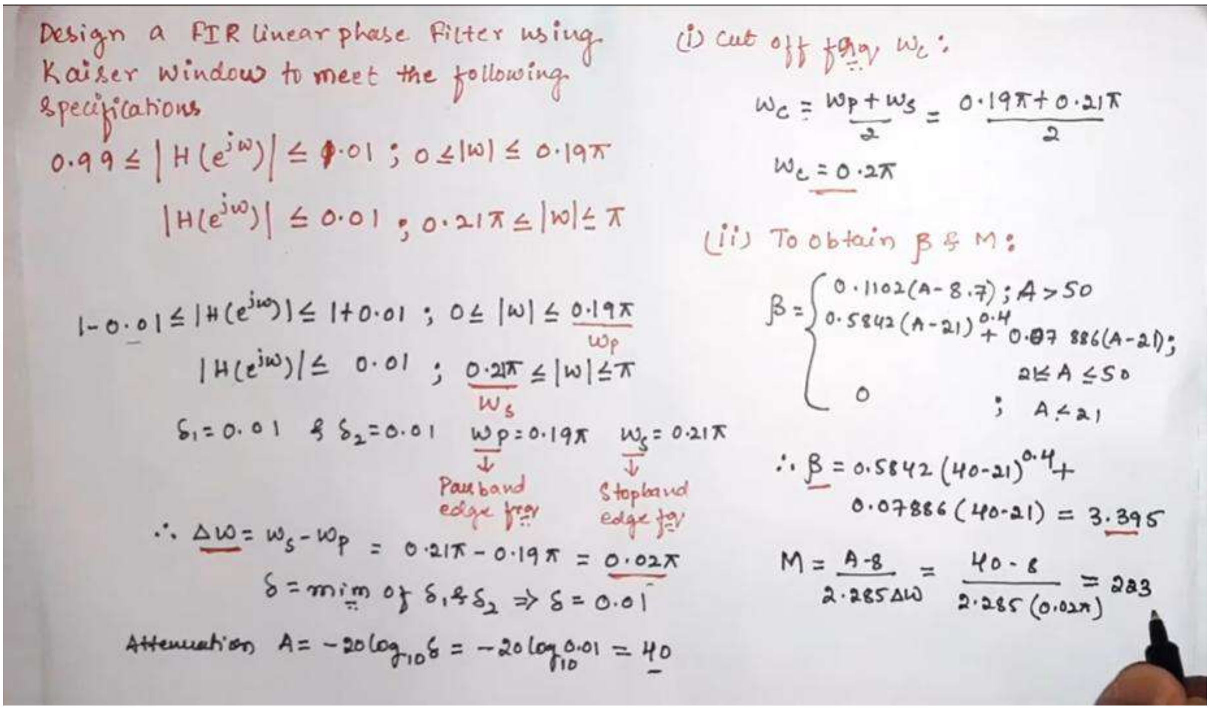
$M=5$ → odd

$$H(\omega) = e^{-j2\omega}\left[h(2) + 2\sum_{n=0}^{1} h(n)\cos\omega(n-2)\right]$$

$$H(e^{j\omega}) = H(\omega) = e^{-j2\omega}\left[0.25 + 2\times 0.01273\cos 2\omega + 2\times 0.12149\cos\omega\right]$$

$$H(e^{j\omega}) = e^{-j2\omega}\left[0.25 + 0.02546\cos 2\omega + 0.243\cos\omega\right]$$

Problem on FIR filter using Kaiser window



Design a FIR linear phase Filter using Kaiser window to meet the following specifications

$0.99 \le |H(e^{j\omega})| \le 1.01$; $0 \le |\omega| \le 0.19\pi$

$|H(e^{j\omega})| \le 0.01$; $0.21\pi \le |\omega| \le \pi$

$1 - 0.01 \le |H(e^{j\omega})| \le 1 + 0.01$; $0 \le |\omega| \le \underbrace{0.19\pi}_{\omega_p}$

$|H(e^{j\omega})| \le 0.01$; $\underbrace{0.21\pi}_{\omega_s} \le |\omega| \le \pi$

$\delta_1 = 0.01$ & $\delta_2 = 0.01$ $\quad \omega_p = 0.19\pi$ ↓ Pass band edge freq $\quad \omega_s = 0.21\pi$ ↓ Stopband edge freq

$\therefore \Delta\omega = \omega_s - \omega_p = 0.21\pi - 0.19\pi = 0.02\pi$

$\delta = \min$ of δ_1 & $\delta_2 \Rightarrow \delta = 0.01$

Attenuation $A = -20\log_{10}\delta = -20\log_{10} 0.01 = 40$

(i) cut off freq ω_c:

$$\omega_c = \frac{\omega_p + \omega_s}{2} = \frac{0.19\pi + 0.21\pi}{2}$$

$\omega_c = 0.2\pi$

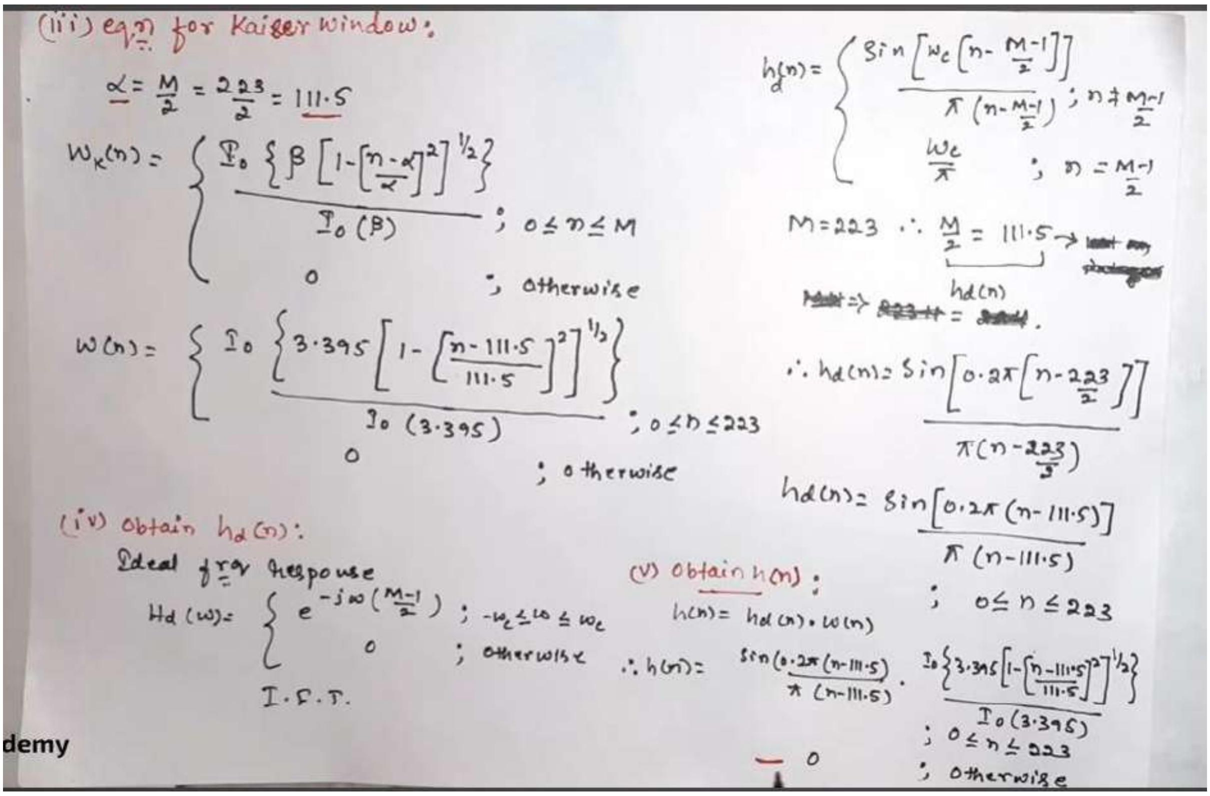
(ii) To obtain β & M:

$$\beta = \begin{cases} 0.1102(A-8.7) & ; A > 50 \\ 0.5842(A-21)^{0.4} + 0.07886(A-21) & ; 21 \le A \le 50 \\ 0 & ; A < 21 \end{cases}$$

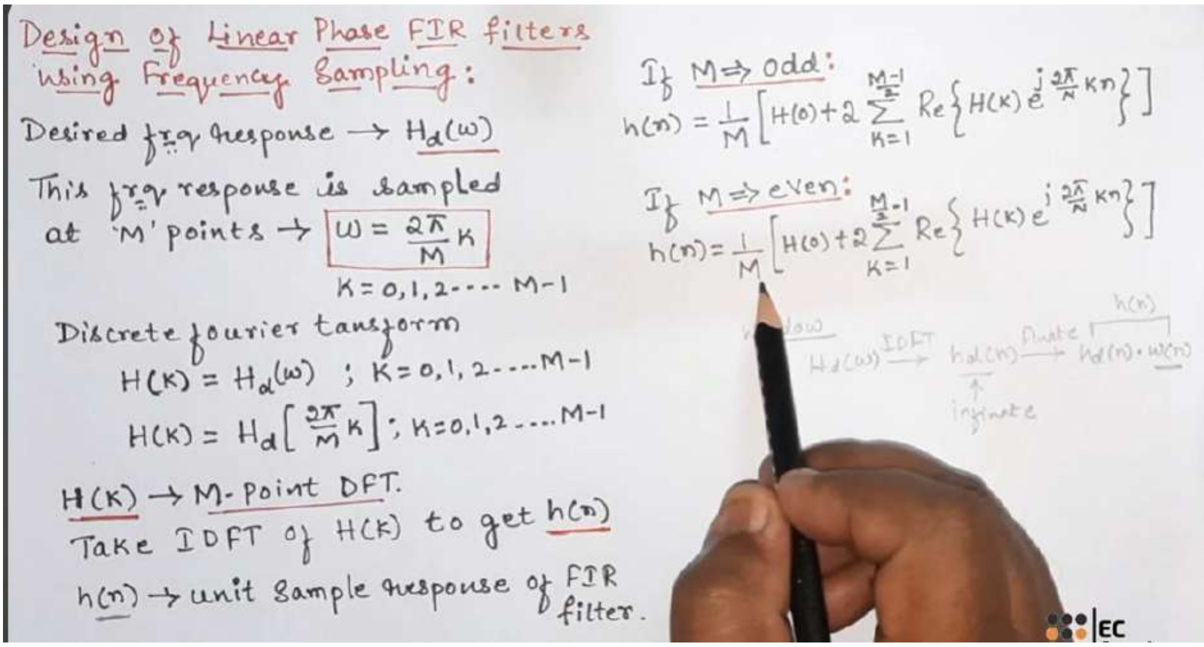
$\therefore \beta = 0.5842(40-21)^{0.4} + 0.07886(40-21) = 3.395$

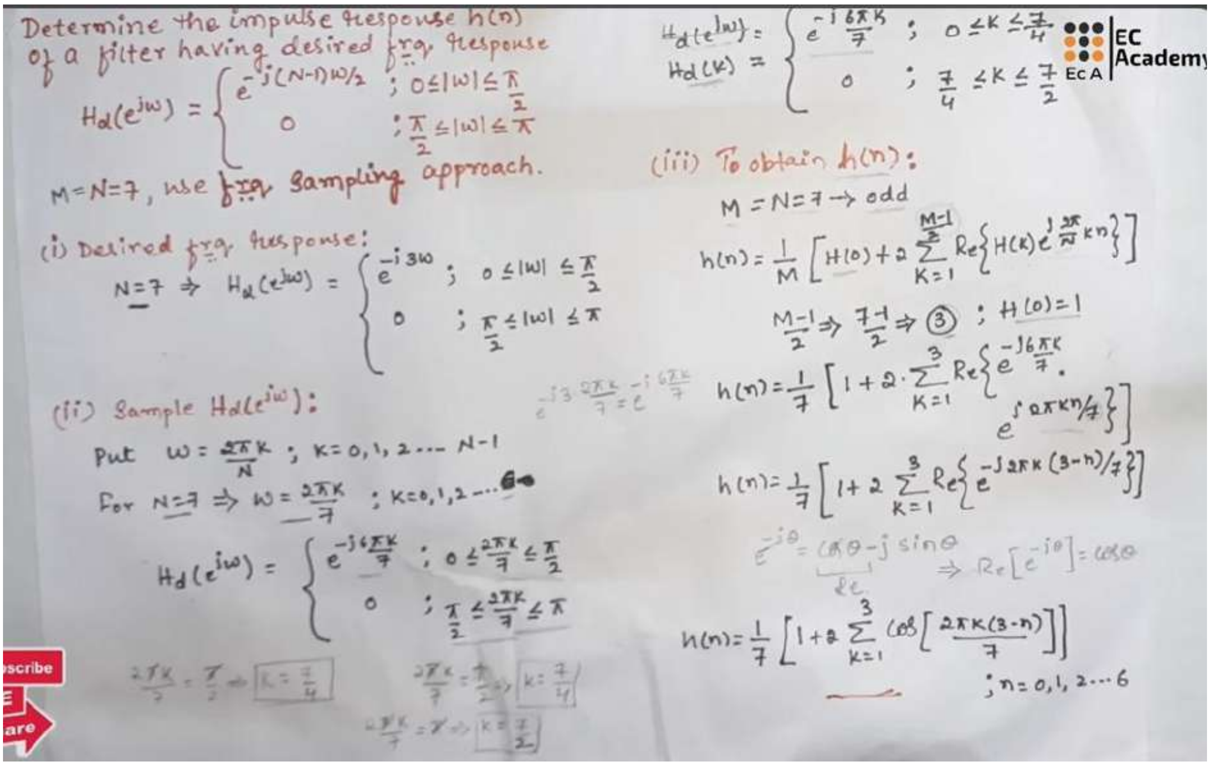
$$M = \frac{A-8}{2.285\Delta\omega} = \frac{40-8}{2.285(0.02\pi)} = 223$$

In this equation we have taken length of window as M+1, hence we have taken M/2 i.e. 223/2



Design of linear phase FIR filter using frequency sampling method





Determine the impulse response $h(n)$ of a filter having desired frq. response

$$H_d(e^{j\omega}) = \begin{cases} e^{-j(N-1)\omega/2} & ; 0 \le |\omega| \le \frac{\pi}{2} \\ 0 & ; \frac{\pi}{2} \le |\omega| \le \pi \end{cases}$$

$M = N = 7$, use frq. sampling approach.

(i) Desired frq. response:

$$N=7 \Rightarrow H_d(e^{j\omega}) = \begin{cases} e^{-j3\omega} & ; 0 \le |\omega| \le \frac{\pi}{2} \\ 0 & ; \frac{\pi}{2} \le |\omega| \le \pi \end{cases}$$

(ii) Sample $H_d(e^{j\omega})$:

Put $\omega = \frac{2\pi k}{N}$; $k = 0, 1, 2 \dots N-1$

For $N=7 \Rightarrow \omega = \frac{2\pi k}{7}$; $k = 0, 1, 2 \dots 6$

$$H_d(e^{j\omega}) = \begin{cases} e^{-j\frac{6\pi k}{7}} & ; 0 \le \frac{2\pi k}{7} \le \frac{\pi}{2} \\ 0 & ; \frac{\pi}{2} \le \frac{2\pi k}{7} \le \pi \end{cases}$$

$\frac{2\pi k}{7} = \frac{\pi}{2} \Rightarrow \boxed{k = \frac{7}{4}}$ $\quad \frac{2\pi k}{7} = \frac{\pi}{2} \Rightarrow \boxed{k = \frac{7}{4}}$ $\quad \frac{2\pi k}{7} = \pi \Rightarrow \boxed{k = \frac{7}{2}}$

$e^{-j3 \cdot \frac{2\pi k}{7}} = e^{-j\frac{6\pi k}{7}}$

$$H_d(e^{j\omega}) = H_d(k) = \begin{cases} e^{-j\frac{6\pi k}{7}} & ; 0 \le k \le \frac{7}{4} \\ 0 & ; \frac{7}{4} \le k \le \frac{7}{2} \end{cases}$$

(iii) To obtain $h(n)$:

$M = N = 7 \rightarrow$ odd

$$h(n) = \frac{1}{M}\left[H(0) + 2\sum_{k=1}^{\frac{M-1}{2}} Re\left\{H(k) e^{j\frac{2\pi}{N}kn}\right\}\right]$$

$\frac{M-1}{2} \Rightarrow \frac{7-1}{2} \Rightarrow 3$; $H(0) = 1$

$$h(n) = \frac{1}{7}\left[1 + 2 \cdot \sum_{k=1}^{3} Re\left\{e^{-j\frac{6\pi k}{7}} \cdot e^{j2\pi kn/7}\right\}\right]$$

$$h(n) = \frac{1}{7}\left[1 + 2\sum_{k=1}^{3} Re\left\{e^{-j2\pi k(3-n)/7}\right\}\right]$$

$e^{-j\theta} = \underbrace{\cos\theta}_{Re} - j\sin\theta \Rightarrow Re\left[e^{-j\theta}\right] = \cos\theta$

$$h(n) = \frac{1}{7}\left[1 + 2\sum_{k=1}^{3} \cos\left[\frac{2\pi k(3-n)}{7}\right]\right] \quad ; n = 0, 1, 2 \cdots 6$$

Direct form representation of filter

Digital Filter Structure:

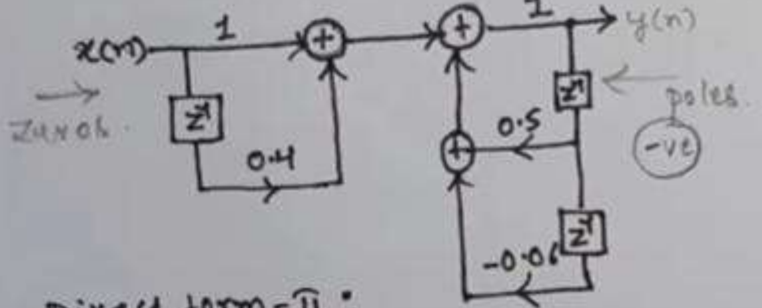
I. Direct Form Structure:

ⓐ Direct form - I
ⓑ Direct form - II

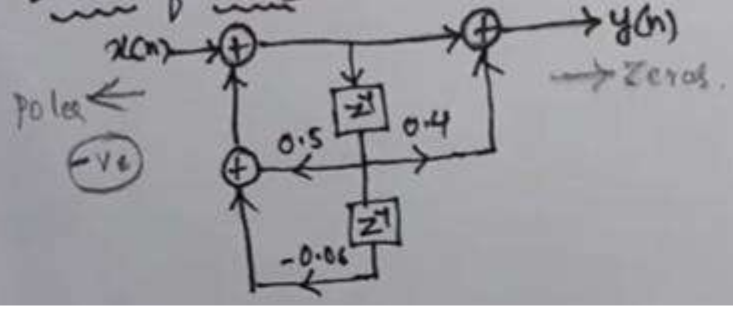
① $H(z) = \dfrac{1+0.4z^{-1}}{1-0.5z^{-1}+0.06z^{-2}}$ zeros, poles

z^{-n} ↳ delay.

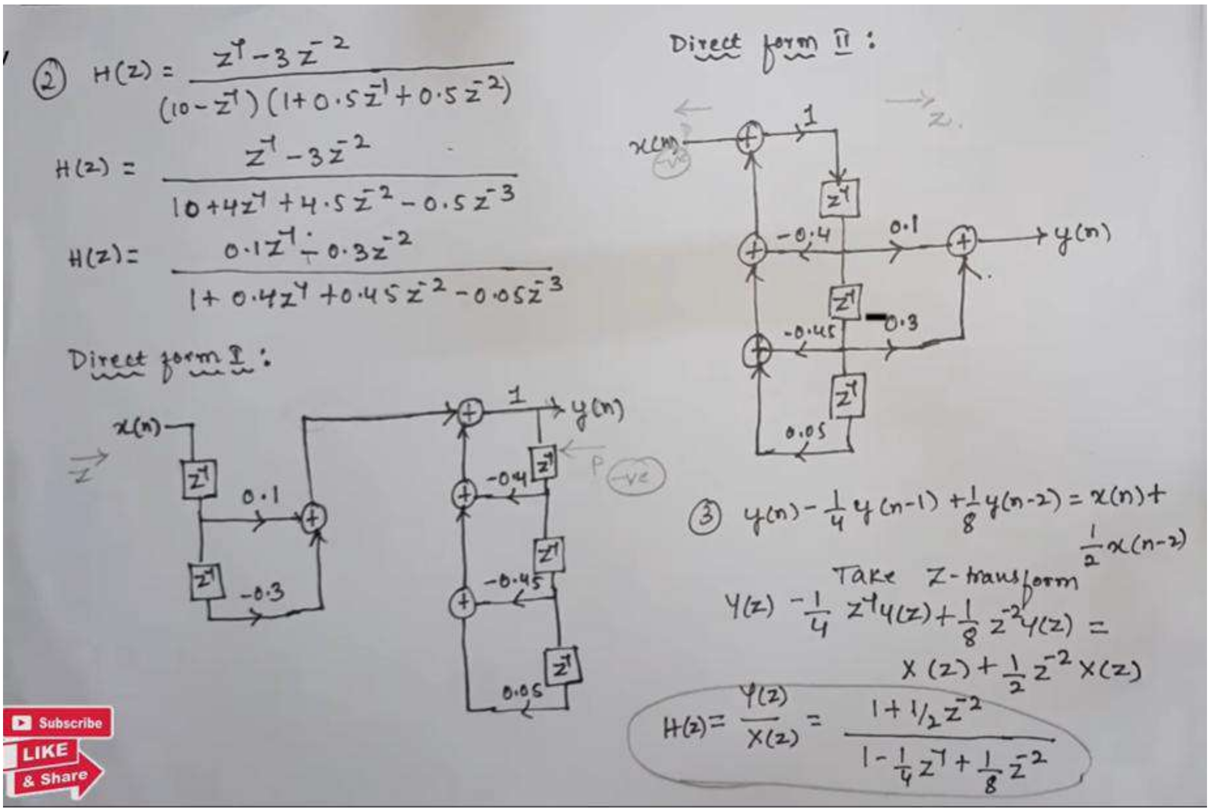
Direct form-I:



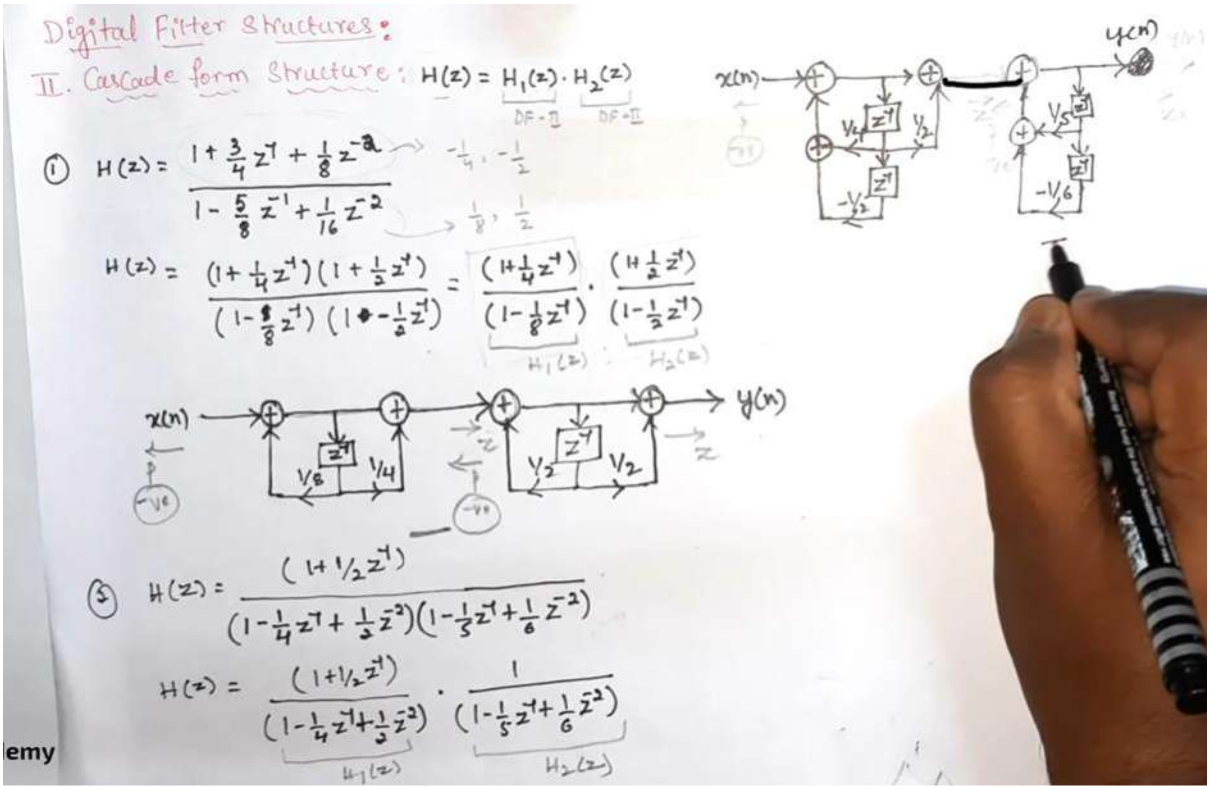
Direct form-II:



We take negative coefficients of poles.

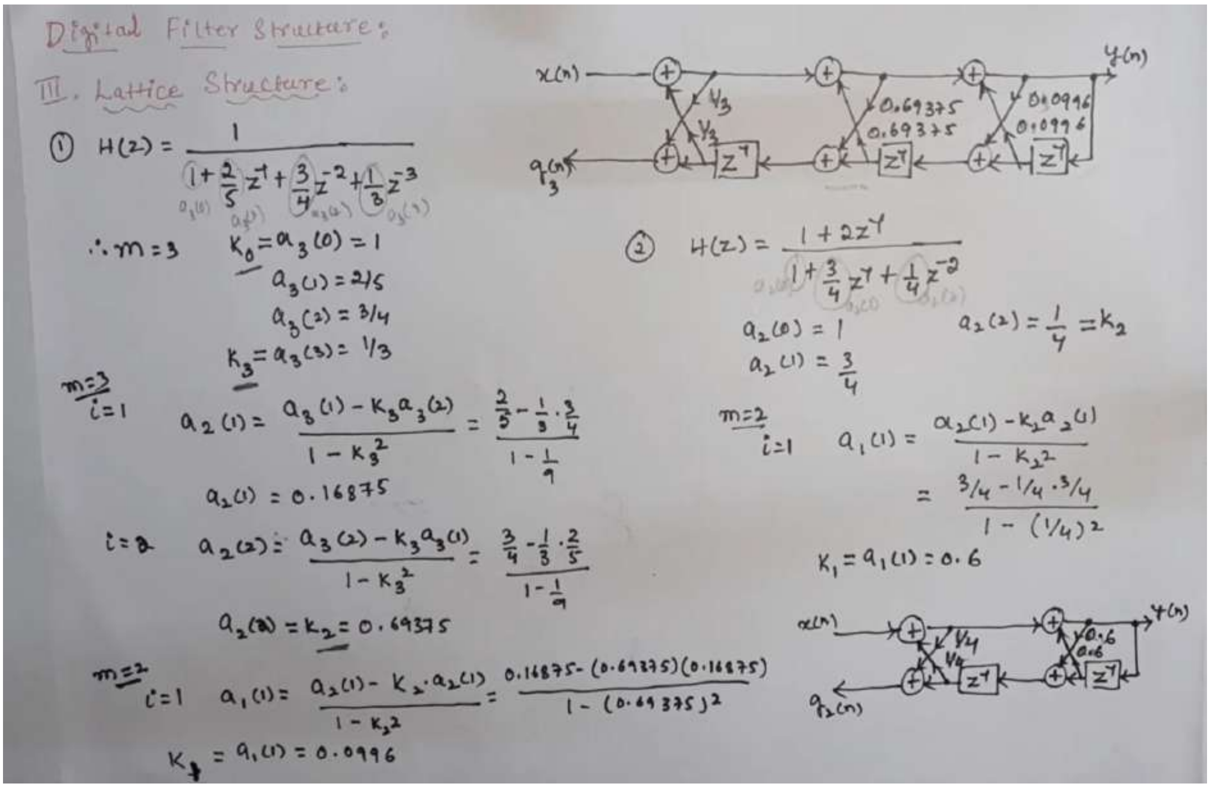


Cascade form structure representation



LATTICE form representation

We need to remember that formula given below for m=n



Digital Filter Structure:

III. Lattice Structure:

(1) $H(z) = \dfrac{1}{1+\frac{2}{5}z^{-1}+\frac{3}{4}z^{-2}+\frac{1}{3}z^{-3}}$

$\therefore m=3$ $\quad K_0 = a_3(0) = 1$

$a_3(1) = 2/5$

$a_3(2) = 3/4$

$K_3 = a_3(3) = 1/3$

$m=3$, $i=1$: $a_2(1) = \dfrac{a_3(1) - K_3 a_3(2)}{1-K_3^2} = \dfrac{\frac{2}{5}-\frac{1}{3}\cdot\frac{3}{4}}{1-\frac{1}{9}}$

$a_2(1) = 0.16875$

$i=2$: $a_2(2) = \dfrac{a_3(2) - K_3 a_3(1)}{1-K_3^2} = \dfrac{\frac{3}{4}-\frac{1}{3}\cdot\frac{2}{5}}{1-\frac{1}{9}}$

$a_2(2) = K_2 = 0.69375$

$m=2$, $i=1$: $a_1(1) = \dfrac{a_2(1) - K_2 \cdot a_2(1)}{1-K_2^2} = \dfrac{0.16875-(0.69375)(0.16875)}{1-(0.69375)^2}$

$K_1 = a_1(1) = 0.0996$

(2) $H(z) = \dfrac{1+2z^{-1}}{1+\frac{3}{4}z^{-1}+\frac{1}{4}z^{-2}}$

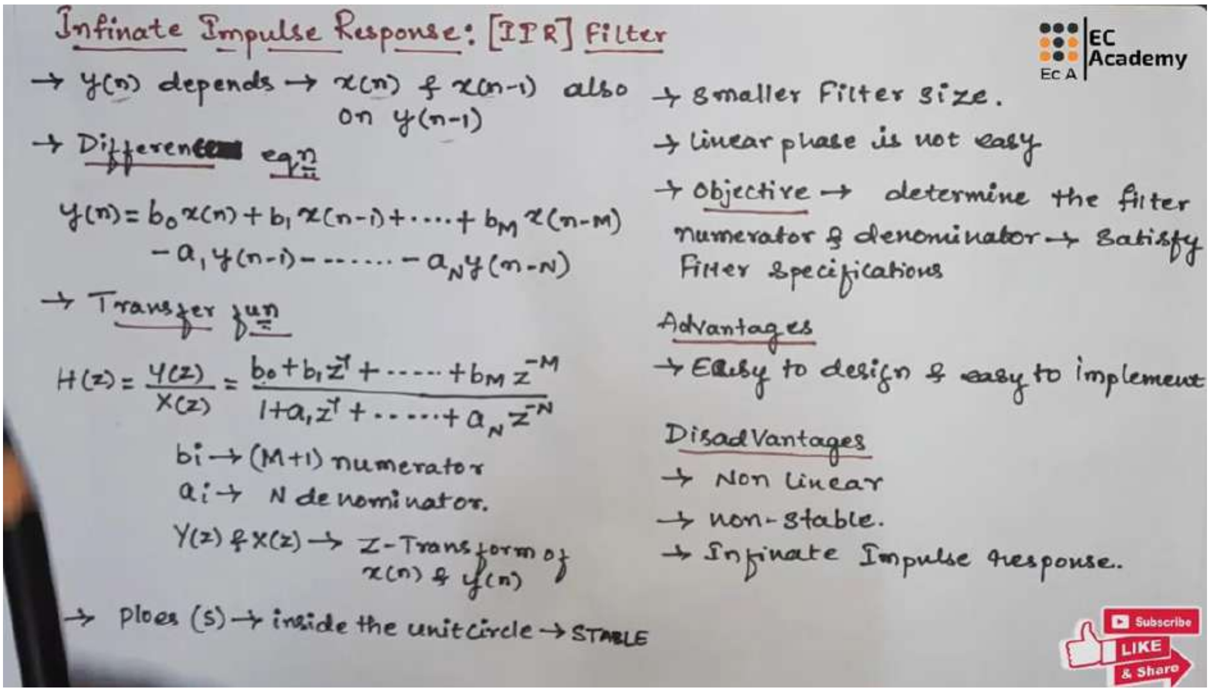
$a_2(0) = 1$ $\quad a_2(2) = \frac{1}{4} = K_2$

$a_2(1) = \frac{3}{4}$

$m=2$, $i=1$: $a_1(1) = \dfrac{a_2(1) - K_2 a_2(1)}{1-K_2^2} = \dfrac{3/4 - 1/4\cdot 3/4}{1-(1/4)^2}$

$K_1 = a_1(1) = 0.6$

Introduction to infinite impulse response (IIR) Filter



Infinate Impulse Response: [IIR] Filter

→ y(n) depends → x(n) & x(n-1) also on y(n-1)

→ Difference eqn

$y(n) = b_0 x(n) + b_1 x(n-1) + \cdots + b_M x(n-M) - a_1 y(n-1) - \cdots - a_N y(n-N)$

→ Transfer fun

$H(z) = \dfrac{Y(z)}{X(z)} = \dfrac{b_0 + b_1 z^{-1} + \cdots + b_M z^{-M}}{1 + a_1 z^{-1} + \cdots + a_N z^{-N}}$

b_i → (M+1) numerator

a_i → N denominator.

Y(z) & X(z) → Z-Transform of x(n) & y(n)

→ ploes (s) → inside the unit circle → STABLE

→ Smaller Filter size.

→ linear phase is not easy

→ Objective → determine the filter numerator & denominator → Satisfy Filter Specifications

Advantages

→ Easy to design & easy to implement

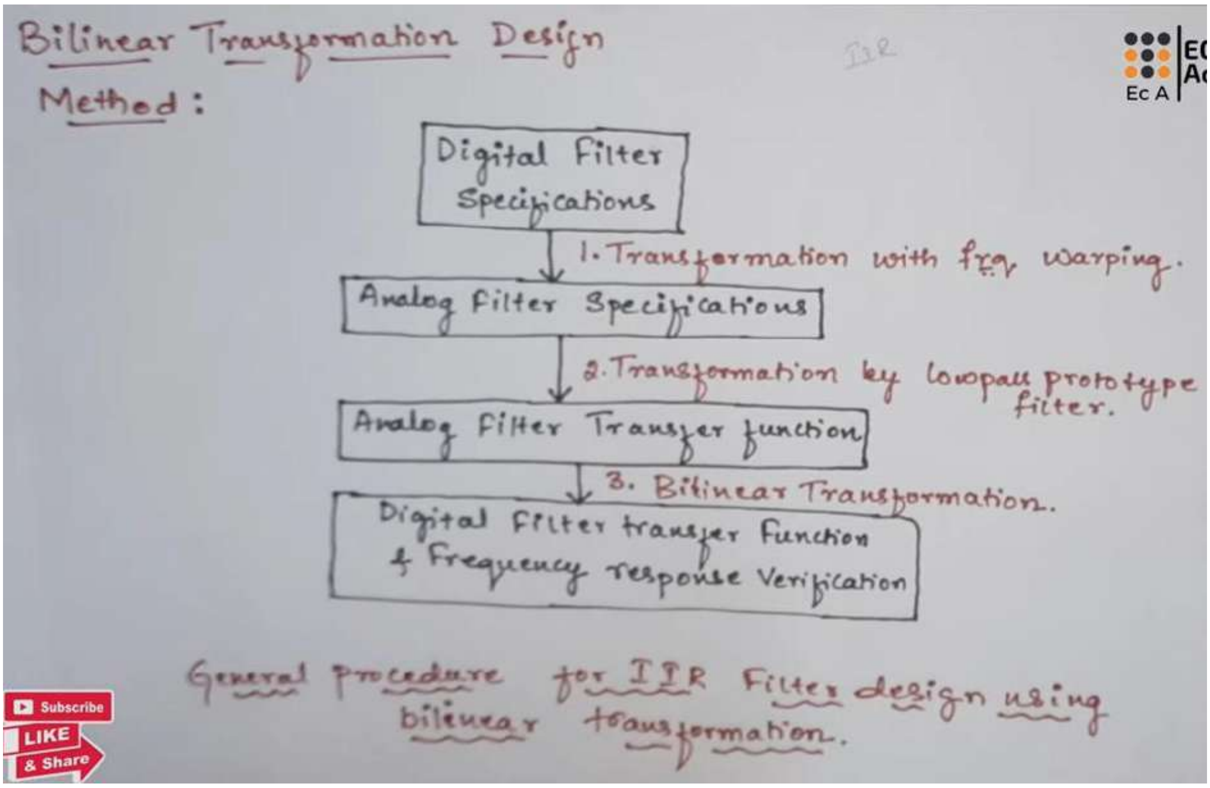
DisadVantages

→ Non Linear

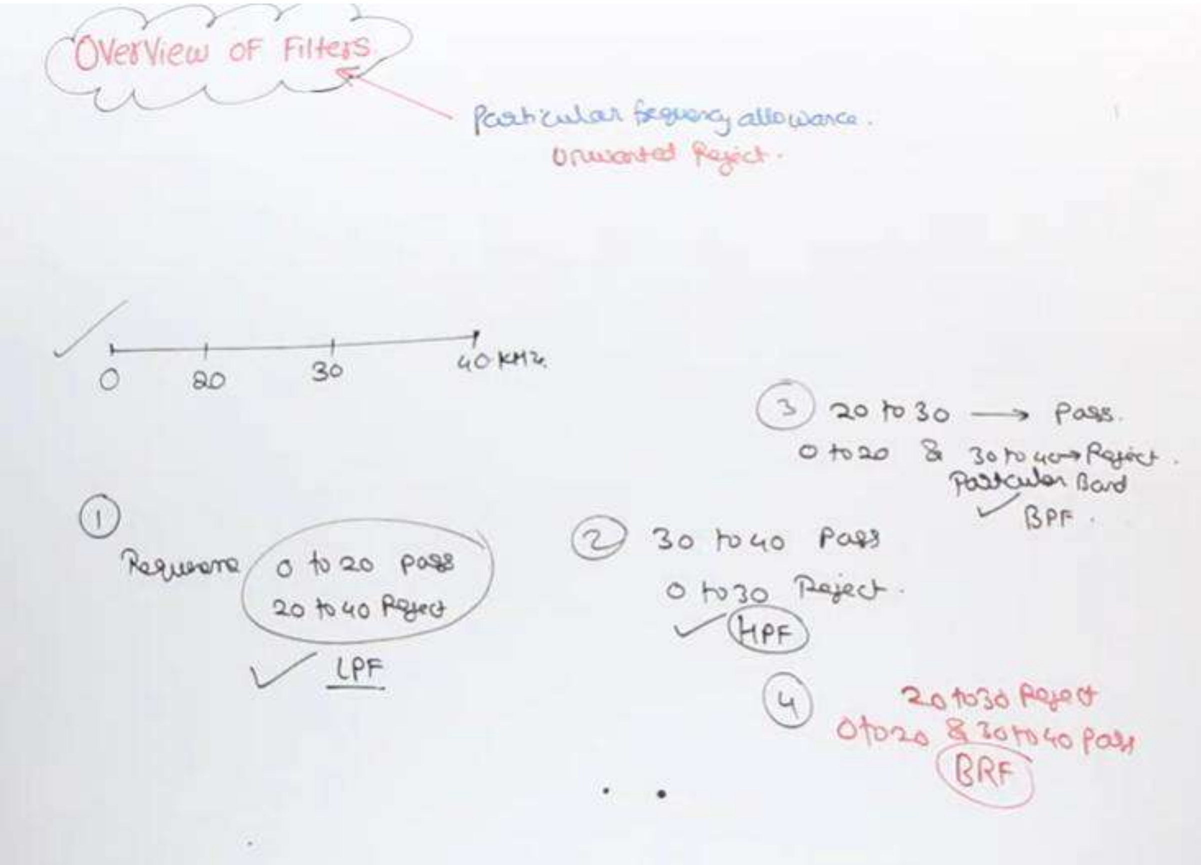
→ non-stable.

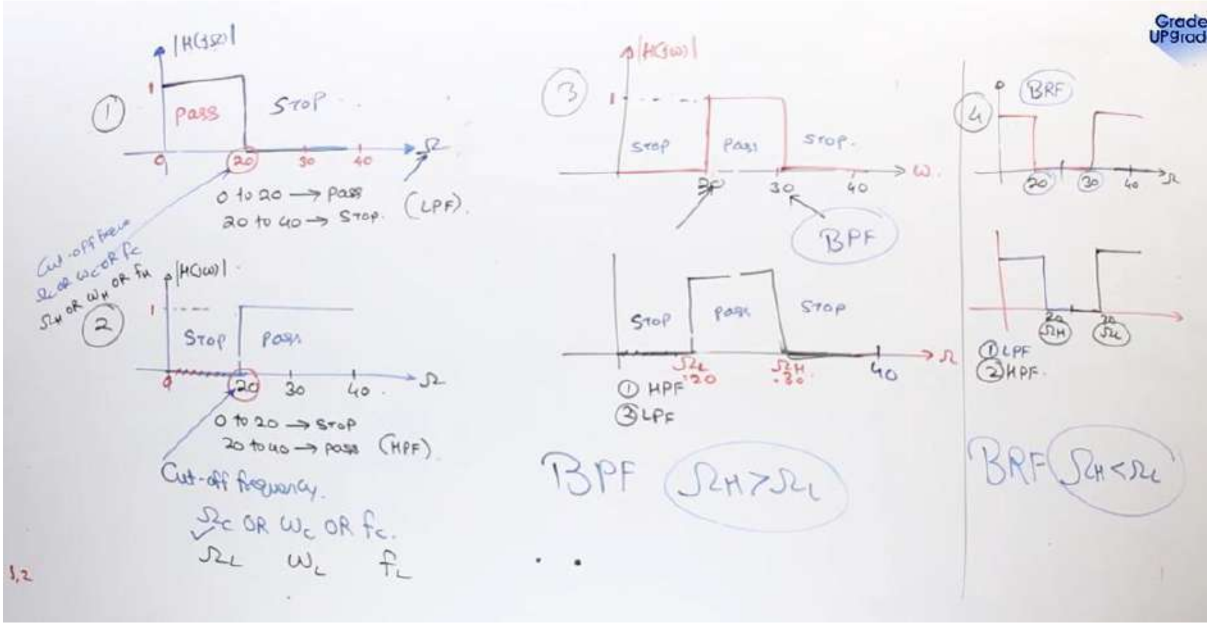
→ Infinate Impulse response.

Bilinear Transformation design method for IIR filter design



Overview Of Ideal Filters





Analog freq (Ω).

$$\Omega = 2\pi f$$

Unitless Radian Hz OR $\frac{1}{sec}$

$\Omega =$ Radian . Hz.

OR

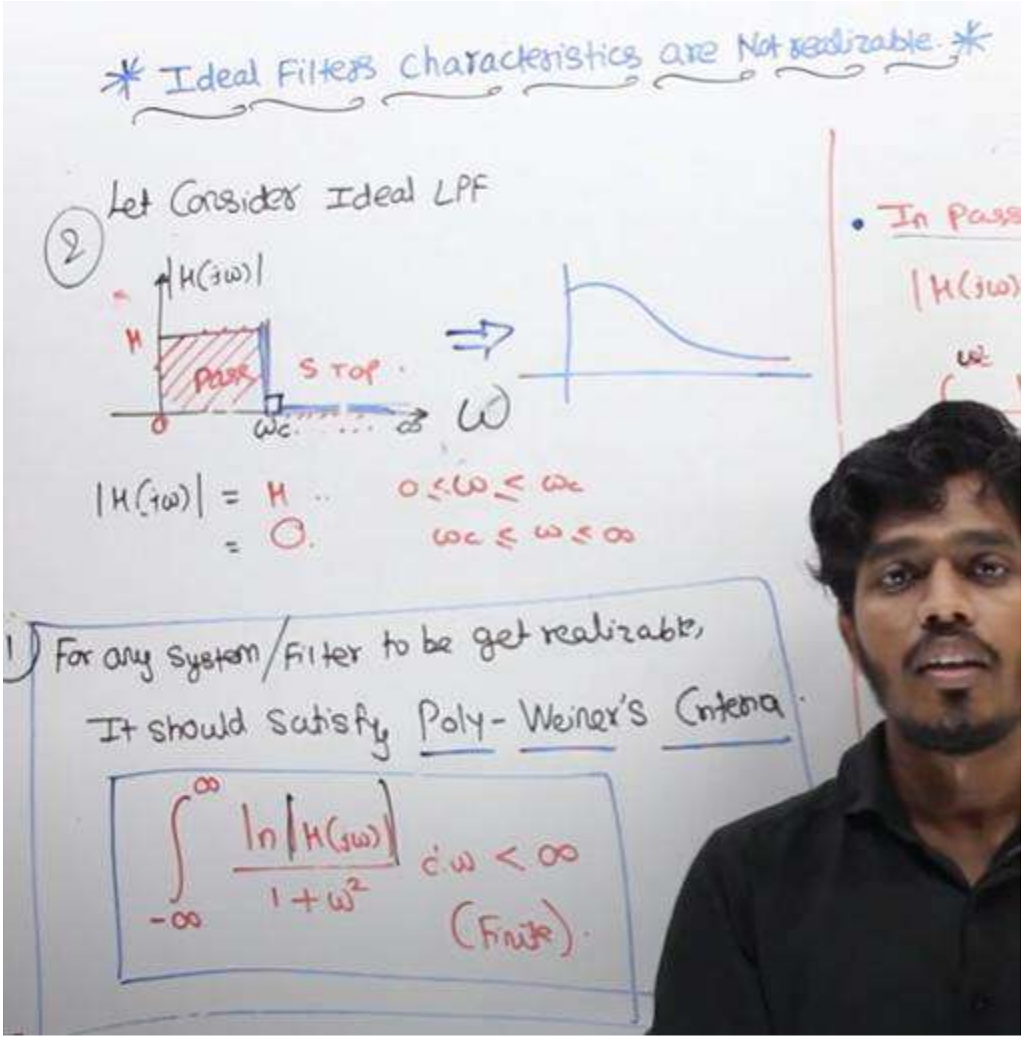
$\Omega =$ Radian $\frac{1}{sec}$

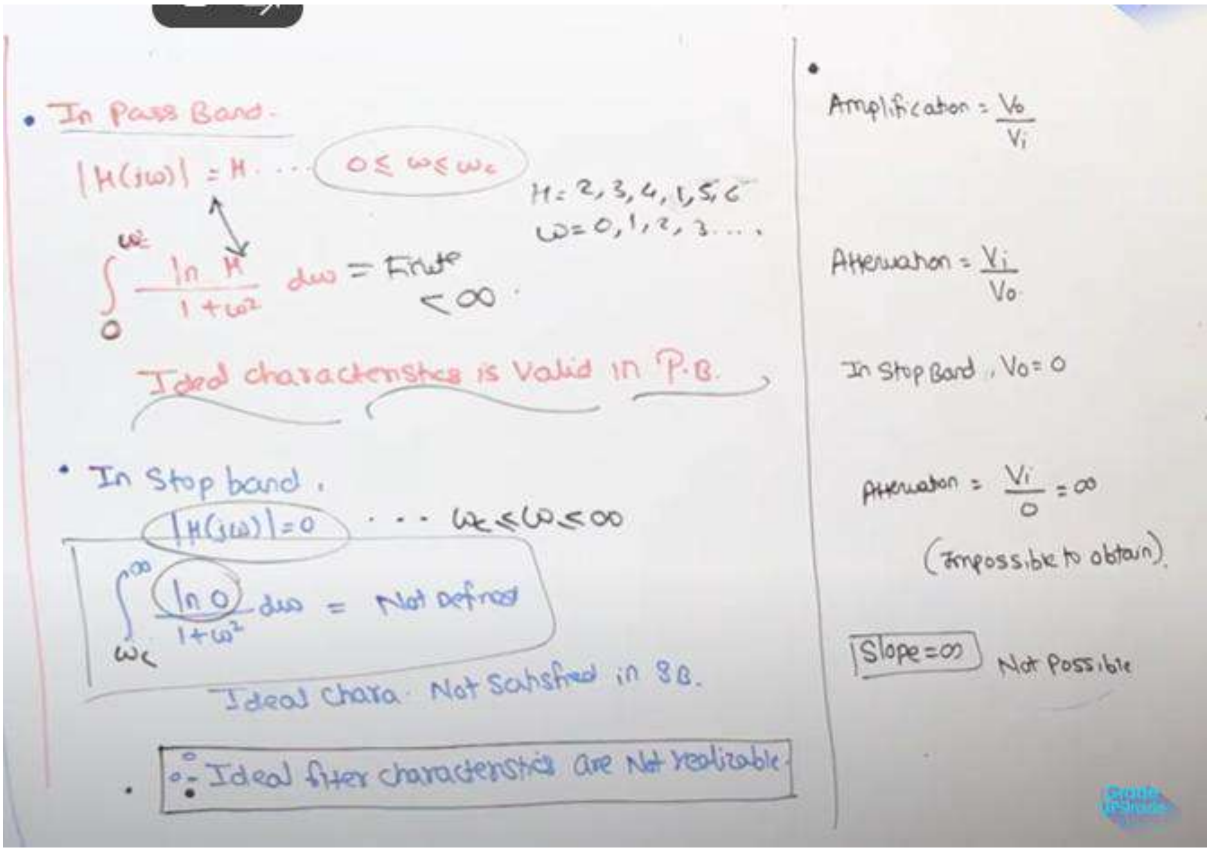
$$\boxed{\Omega = Rad/sec}$$

$$\omega = \frac{\Omega}{f_s} = \frac{2\pi f}{f_s} = \frac{Rad\,\cancel{Hz}}{\cancel{Hz}} = Rad.$$

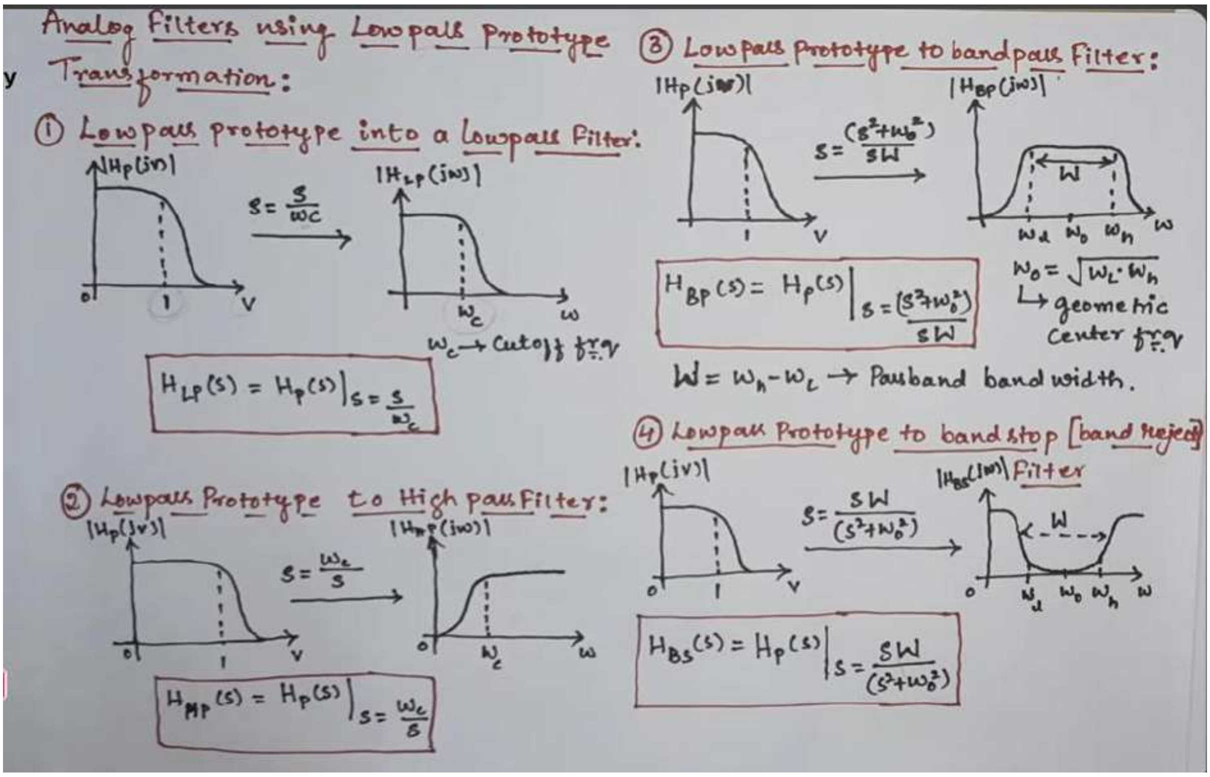
f_s Sampling freq.

Why Ideal Filters Not used

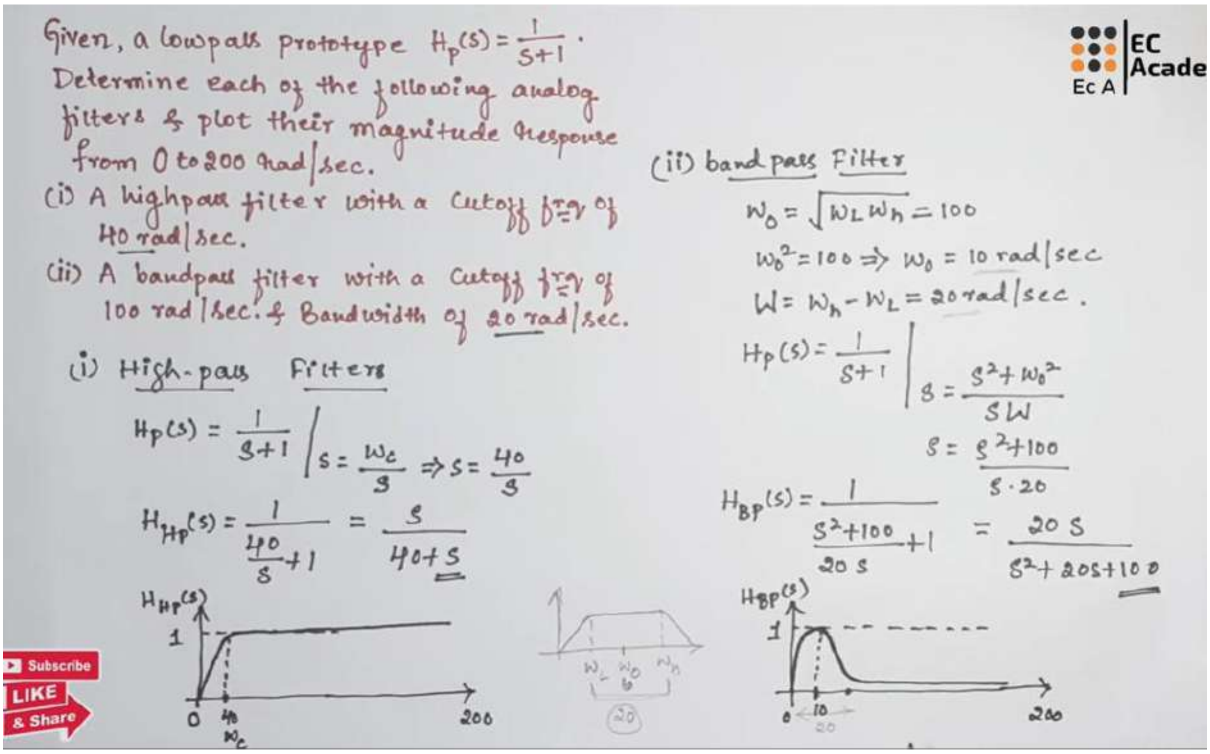




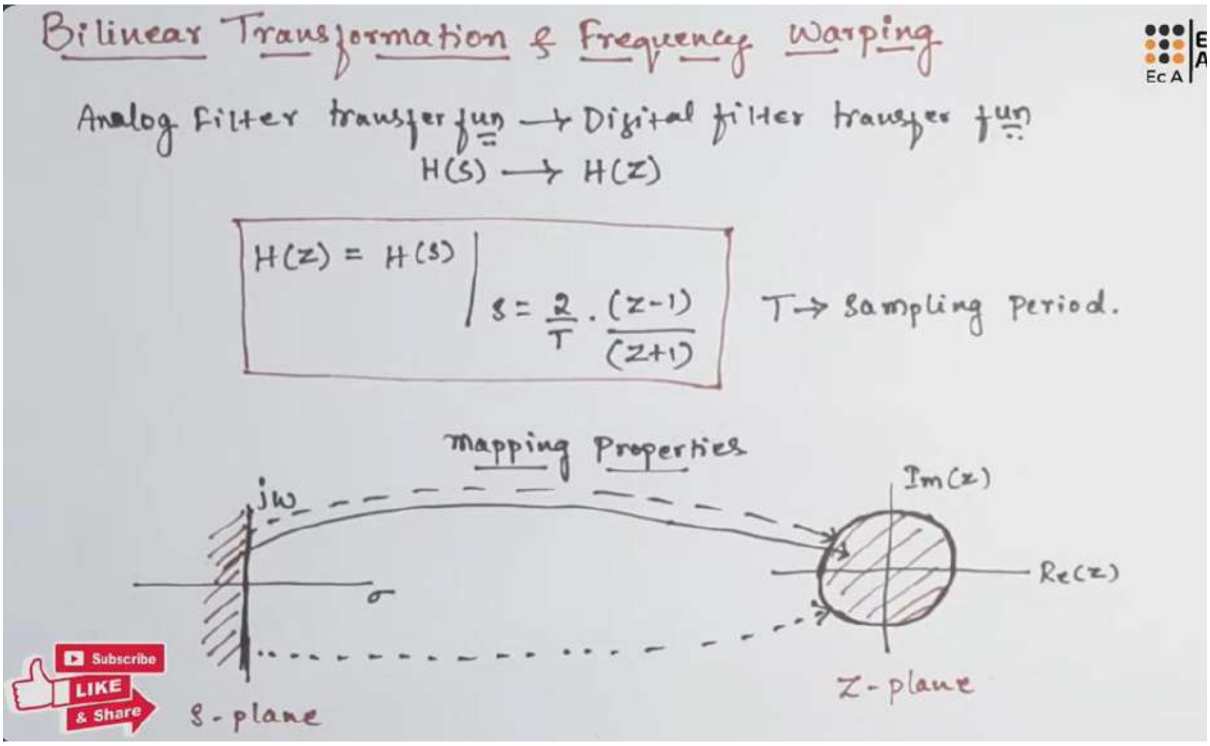
Analog filters using Lowpass prototype Transformation



Problem on Analog filters using Lowpass prototype Transformation



bilinear transformation and frequency warping in IIR filters



DFT Proofs

Let $x(n)$ be a real sequence of length N and its N-point DFT is X(k), show that

i. $X(N-K) = X^*(K)$

ii. $X(0)$ is real

iii. If N is even, then $X\left(\frac{N}{2}\right)$ is real

$$DFT[x(n)] = X(k) = \sum_{n=0}^{N-1} x(n)\, e^{-j\frac{2\pi}{N}nk} \quad : k = 0 \text{ to } N-1$$

$x(n)$ is Real

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

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

$$e^{-j\frac{2\pi}{N}\cdot nN} = e^{-j2\pi n}$$

$$= \cos 2\pi n - j\sin 2\pi n$$

$$= 1$$

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

$$\boxed{X(N-k) = X^*(k)}$$

[because here there exist $+j$ instead of $-j$]

The * symbol here means it’s complex conjugate

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

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

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

Here there is no any j term because all samples of $x(n)$ are real

$\therefore$ $X(0)$ is Real

iii) $$X\left(\frac{N}{2}\right) = \sum_{n=0}^{N-1} x(n)\, e^{-j\frac{2\pi}{N}\cdot n\left(\frac{N}{2}\right)}$$

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

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

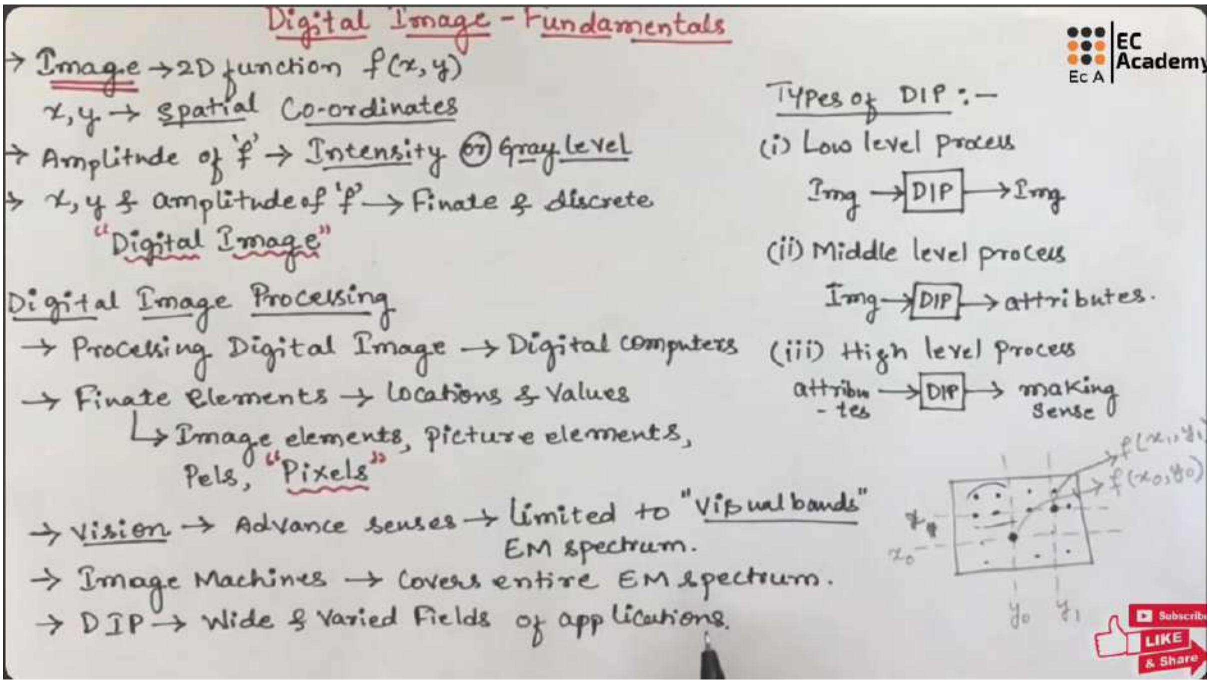
$$\left(e^{-j\pi}\right)^n = \left(\cos\pi - j\sin\pi\right)^n = (-1)^n$$

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

Here there is no any j term because all samples of $x(n)$ is real

$\therefore$ $X\left(\frac{N}{2}\right)$ is Real

Introduction to Digital Image Processing



Digital Image - Fundamentals

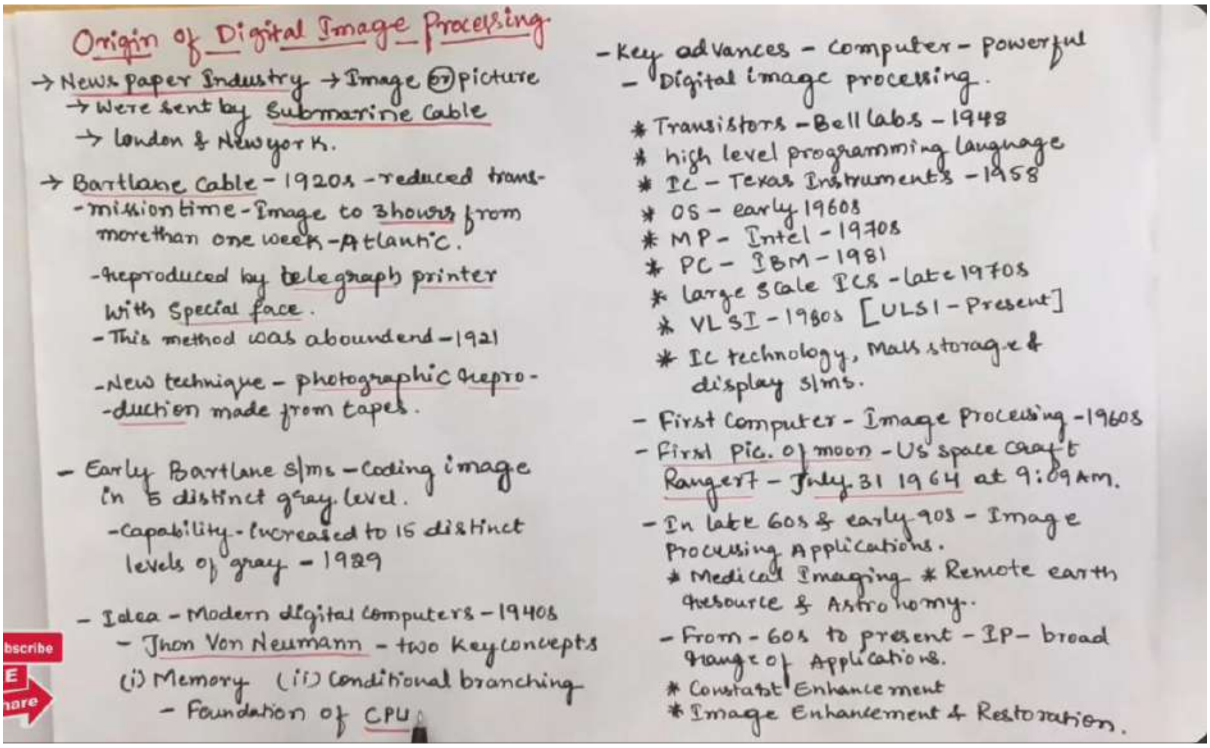
→ Image → 2D function $f(x, y)$
x, y → Spatial Co-ordinates
→ Amplitude of 'f' → Intensity (or) Gray level
→ x, y & amplitude of 'f' → Finite & discrete
"Digital Image"

Digital Image Processing
→ Processing Digital Image → Digital computers
→ Finite elements → locations & Values
↳ Image elements, picture elements, Pels, "Pixels"
→ Vision → Advance senses → limited to "Visual bands" EM spectrum.
→ Image Machines → Covers entire EM spectrum.
→ DIP → Wide & Varied Fields of applications.

Types of DIP :—
(i) Low level process
Img → DIP → Img
(ii) Middle level process
Img → DIP → attributes.
(iii) High level process
attributes → DIP → making sense

$f(x_1, y_1)$
$f(x_0, y_0)$
x_0 x_1
y_0 y_1

Origin of Digital Image Processing



Origin of Digital Image Processing

→ News paper Industry → Image (or) picture
→ were sent by Submarine Cable
→ london & New york.

→ Bartlane Cable - 1920s - reduced trans-mission time - Image to 3 hours from more than one week - Atlantic.
- reproduced by telegraph printer with special face.
- This method was aboundend - 1921
- New technique - photographic repro-duction made from tapes.

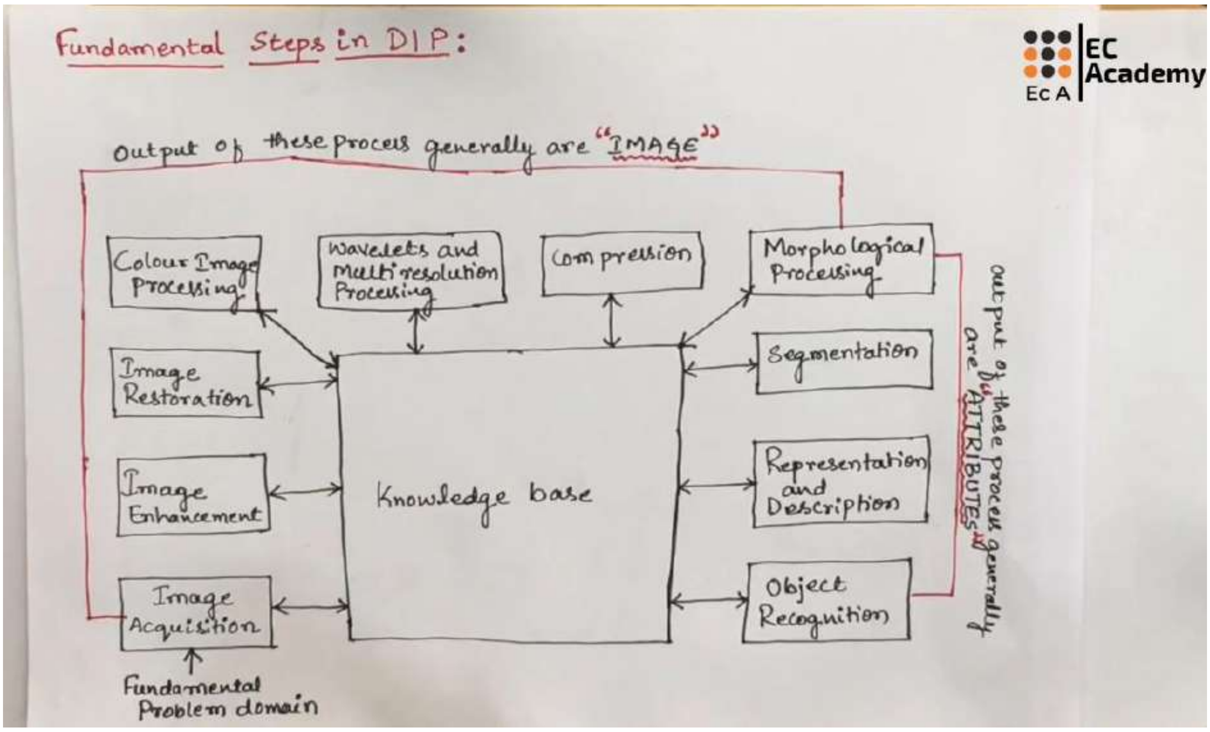
- Early Bartlane s/ms - Coding image in 5 distinct gray level.
- Capability - increased to 15 distinct levels of gray - 1929

- Idea - Modern digital computers - 1940s
- Jhon Von Neumann - two key concepts
(i) Memory (ii) Conditional branching
- Foundation of CPU

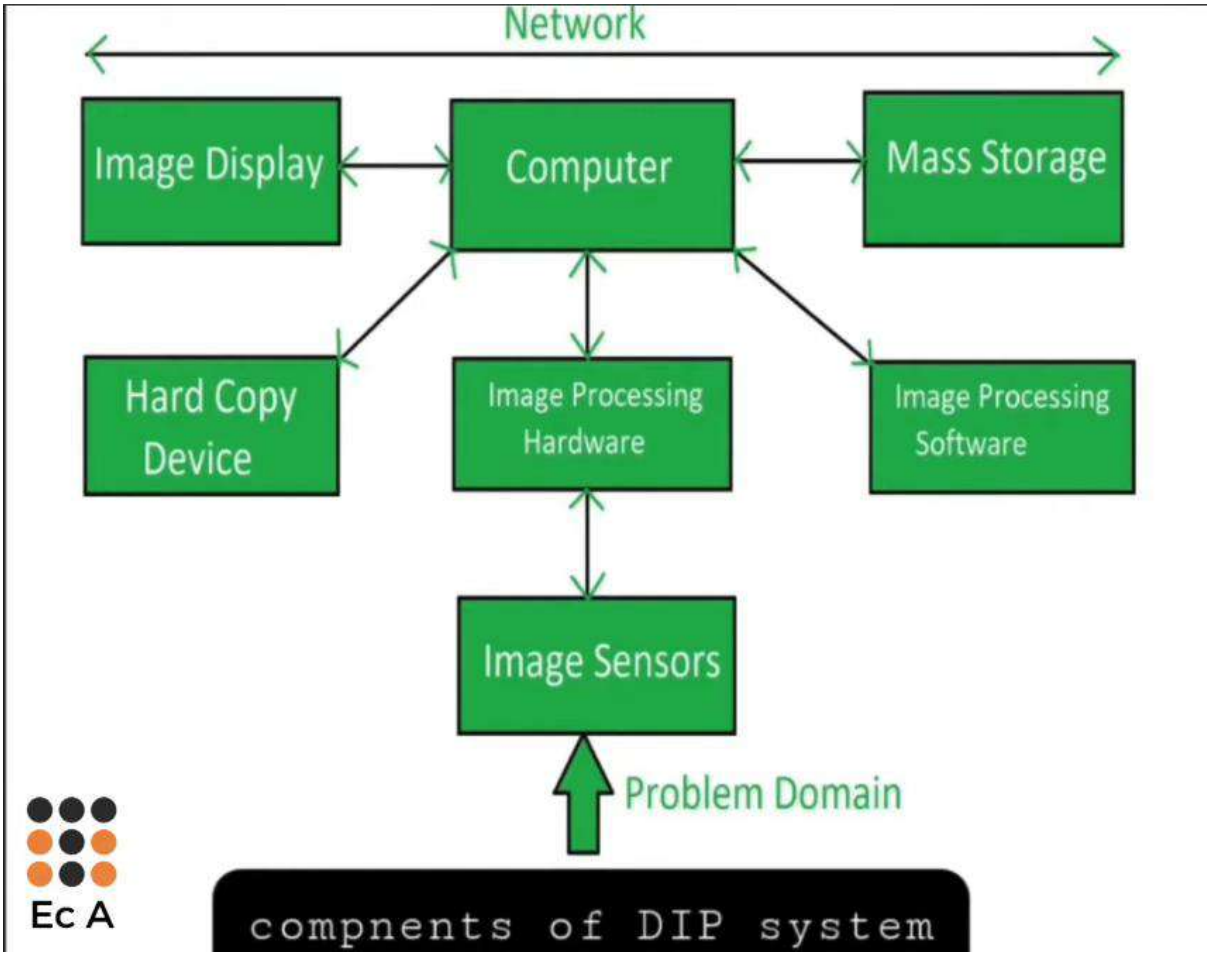
- Key advances - computer - powerful
- Digital image processing.
* Transistors - Bell labs - 1948
* high level programming language
* IC - Texas Instruments - 1958
* OS - early 1960s
* MP - Intel - 1970s
* PC - IBM - 1981
* large scale ICs - late 1970s
* VLSI - 1980s [ULSI - Present]
* IC technology, Mass storage & display s/ms.

- First Computer - Image Processing - 1960s
- First Pic. of moon - US space craft Ranger 7 - July 31 1964 at 9:09 Am.
- In late 60s & early 90s - Image Processing Applications.
* Medical Imaging * Remote earth resource & Astronomy.
- From - 60s to present - IP - broad range of Applications.
* Constrast Enhancement
* Image Enhancement & Restoration.

Fundamental steps in Digital image processing



Components of Digital image processing system



Structure of a human eye

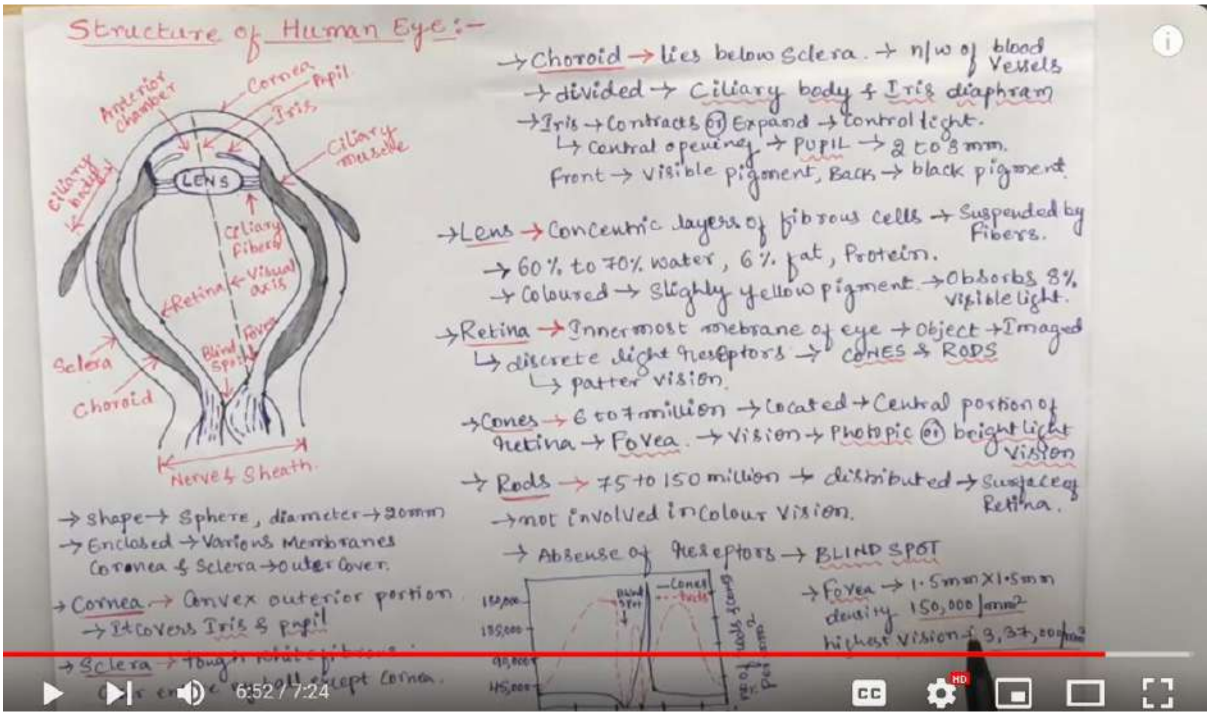


Image formation in an human eye

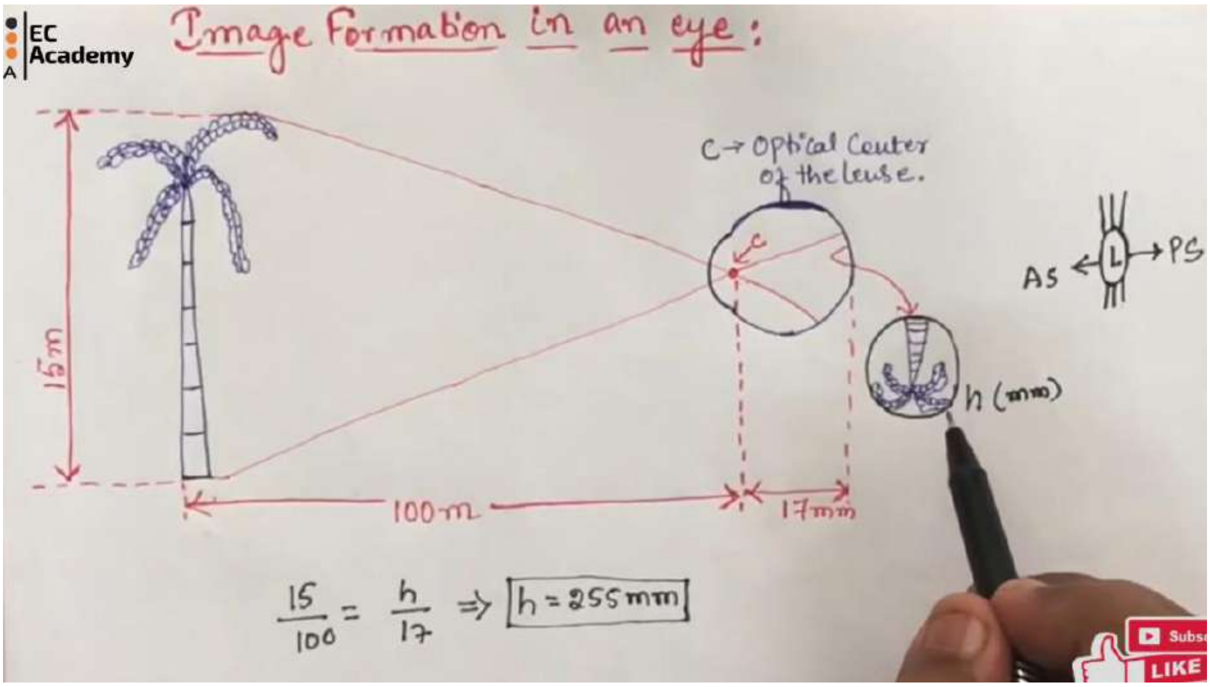
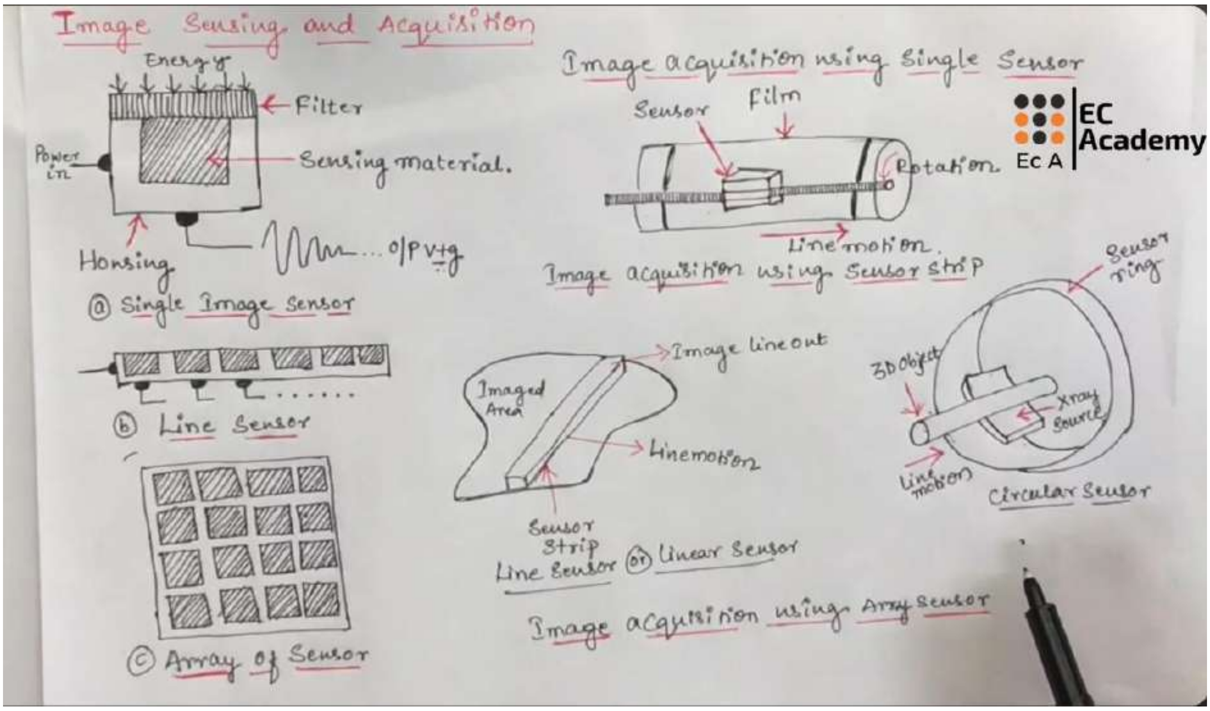
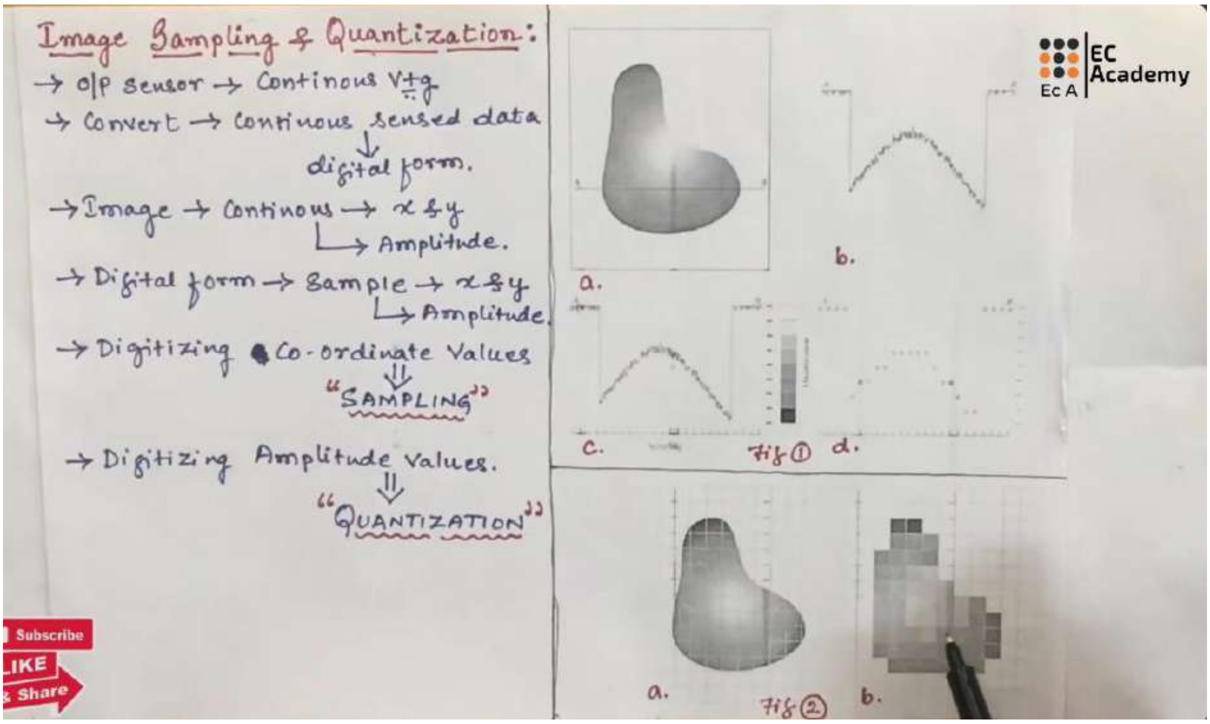


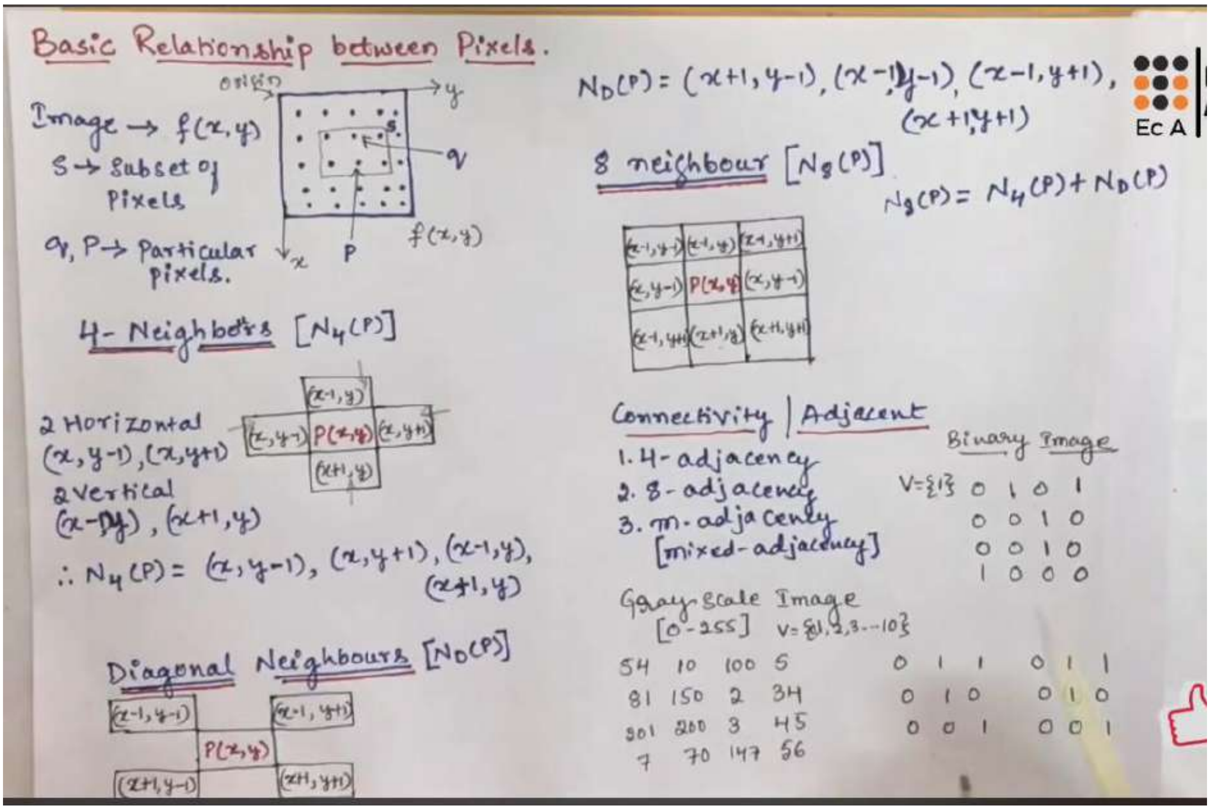
Image sensing and acquisition



Sampling and Quantisation of Digital image



Basic relation between pixels



Distance measure between pixels

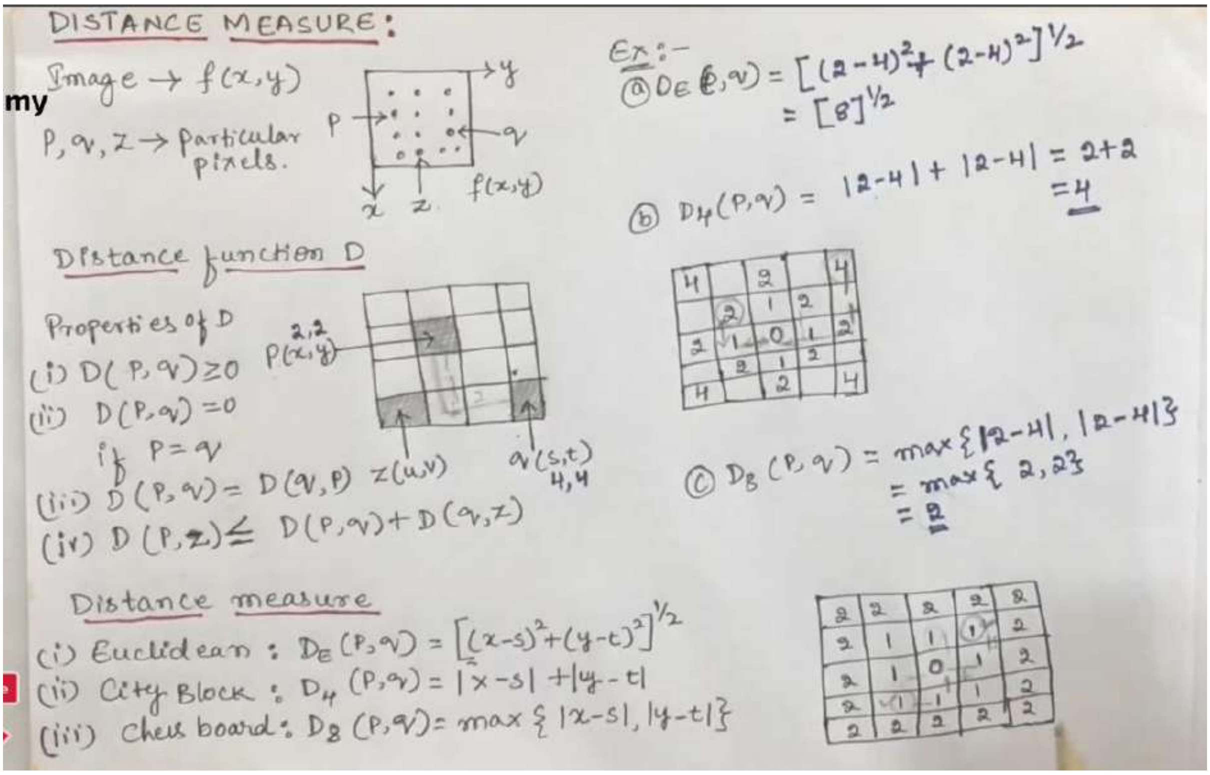


Image Quality | Need for Image Enhancement

❑ Image Quality Factors

❑ Image Quality Assessment Tool

❑ Image Quality Metrics

Image enhancement can be understood with the concept of image quality. Several factors are responsible for image quality but some essential factors are:

- ❑ Contrast
- ❑ Brightness
- ❑ Spatial Resolution
- ❑ Noise

Contrast

- It refers to the finer details of an image.
- Contrast can be simply explained as the difference between maximum and minimum pixel intensity in an image.

When an image is formed, the magnitude of the intensity differences between the different surface of an object is recorded which is known as contrast.

Contrast can be measured in many ways. A common measure of contrast involving foreground and background objects, given as:

$$C = \frac{f_{object} - f_{background}}{f_{object} + f_{background}}$$

Here, f_{object} and $f_{background}$ are the average pixel intensity of object and background.

Another Expression may be written as:

$$C_{Webber} = \frac{f_{object} - f_{background}}{f_{background}}$$

It is often known as Simultaneous Contrast.

One more way to compute contrast on sinusoidal grating, written as:

$$C_{Michelson} = \frac{f_{max} - f_{min}}{f_{max} + f_{min}}$$

Here, f_{max} and f_{min} are the maximum and minimum intensities of the image.

Contrast can be increased by multiplying every pixel of the image by constant, a.

New obtained image can be written as:

new $\boxed{g(x, y) = a \times f(x, y)}$

i/p image

Brightness

- Brightness is a relative term. It depends on your visual perception.
- Since brightness is a relative term, so brightness can be defined as the amount of energy output by a source of light relative to the source we are comparing it to.
- In some cases we can easily say that the image is bright, and in some cases, its not easy to perceive.

How to make an image brighter?

Brightness can be simply increased or decreased by simple addition or subtraction, to the image matrix.

The brightness can be increased by adding a constant b to every pixel of the image. New obtained image can be written as:

$$\boxed{g(x, y) = f(x, y) + b}$$

The brightness can be decreased by subtracting a constant b to every pixel of the image. New obtained image can be written as:

$$\boxed{g(x, y) = f(x, y) - b}$$

Spatial Resolution

- Spatial resolution is a term that refers to the number of pixels, pixel density and quantization levels.
- This depends on sampling and quantization process.
- If Spatial resolution is not satisfactory then image quality will not be satisfactory.
- Images having higher spatial resolution are composed with a greater number of pixels than those of lower spatial resolution.

Measuring spatial resolution

Since the spatial resolution refers to clarity, so for different devices, different measure has been made to measure it.

Some of the examples are:

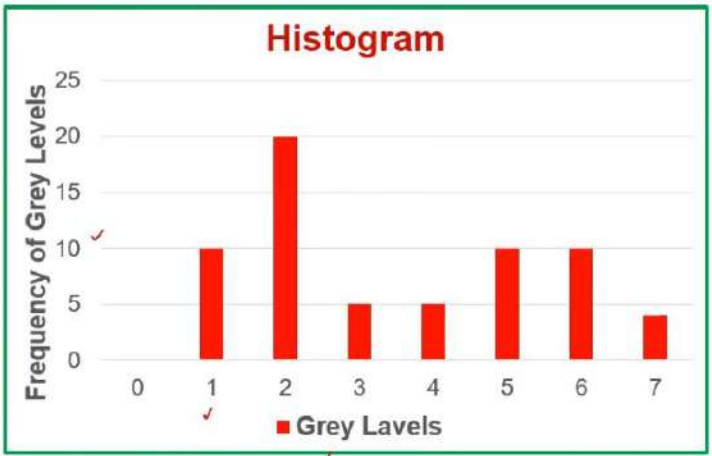
- Dots per inch (usually used in monitors)
- Lines per inch (usually used in laser printers)
- Pixels per inch (used to measure for different devices such as tablets , Mobile phones etc.)

Noise

- Image applications are frequently affected by the noise present in the image. Noise is an unwanted disturbance that causes fluctuations in the pixel value.
- It is a random or stochastic process therefore its true value can not be predicted accurately.
- Noise obeys all the statistical properties.

Frequency distribution of grey levels

Pixels	Frequency
0	0
1	10
2	20
3	5
4	5
5	10
6	10
7	4



Advantages of Histogram

- The histogram of an image indicates the dynamic range of the image.
- Dynamic range is an indicator of contrast and how good the image is.

Dynamic Range is a useful metric that is expressed as the difference between the maximum and minimum pixel value found in the histogram.

$$\text{Dynamic Range} = f_{max} - f_{min}$$

Dynamic Range can also be expressed in db:

$$\text{Dynamic Range} = 20 \log (f_{max} - f_{min}) \text{ db}$$

Image Quality Metric

- It is necessary to quantify the quality of an image.
- The metric used to quantify the image quality can be divided into two categories.

❖ Objective Fidelity Criteria

❖ Subjective Fidelity Criteria

Objective Fidelity Criteria

It provides equation that help us quantify the error i.e. helping in characterizing image quality.

$f(x,y)$ → **Image Task** **(Ex: Image Compression)** → $\hat{f}(x,y)$

Error can now be written as:

$$\text{Error} = e(x,y) = \hat{f}(x,y) - f(x,y)$$

Error may be negative so to avoid negative number, mean square error (MSE) is commonly used. It can be expressed as:

$$\text{MSE} = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}[f(x,y) - \hat{f}(x,y)]^2$$

Image quality is also measured as the signal to noise ratio (SNR). It can be expressed as:

$$\text{SNR} = 20\log_{10}\left(\frac{\text{Signal amplitude}}{\text{Noise amplitude}}\right)\text{dB}$$

Signal to noise ratio (SNR) of an image can be expressed as:

$$\text{SNR} = 20\log_{10}\frac{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}[f(x,y)]^2}{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}[f(x,y) - \hat{f}(x,y)]^2}\text{dB}$$

M×N

Another useful metric is peak PSNR, which can be described for an 8-bit image as follows:

$$\text{PSNR} = 20\log_{10}\frac{255^2 MN}{\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}[f(x,y) - \hat{f}(x,y)]^2}\text{dB}$$

Subjective Fidelity Criteria

- It is not based on any metric.
- It depend on the perception of human observers and their visual systems.
- Rating by a number of human observers, based on typical decompressed images, are averaged to obtain this subjective fidelity criteria.

Example of an absolute comparison scale is give in table.

Value	Rating	Description
1	Excellent	An image of extremely high quality --- as good as desired.
2	Fine	An image of high quality, providing enjoyable viewing. Interference is not objectionable.
3	Passable	An image of acceptable quality. Interference is not objectionable.
4	Marginal	An image of poor quality; you wish you could improve it. Interference is somewhat objectionable.
5	Inferior	A very poor image, but you could watch it. Objectionable interference is definitely present.
6	Unusable	An image so bad that you cold not watch it.

Introduction to image enhancement using spatial domain

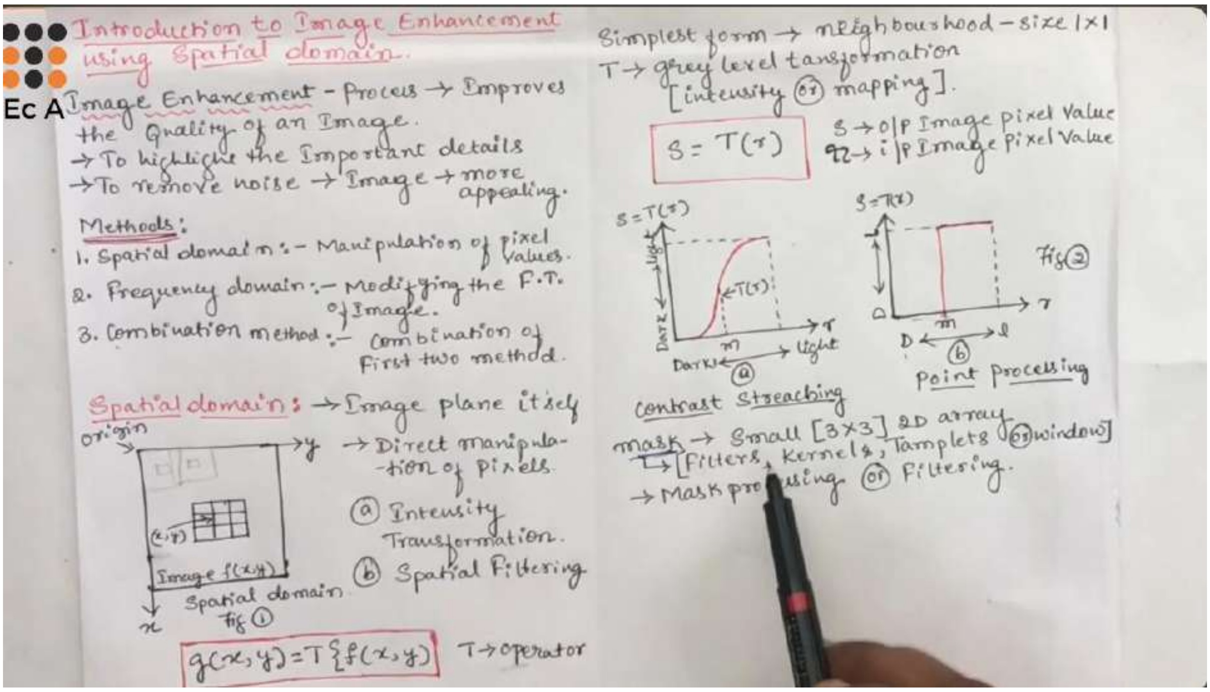


Image Enhancement in Spatial Domain

- **Linear Point Transformations**
- **Non Linear Point Transformations**
  - ❑ **Square Function**
  - ❑ **Square Root Function**
  - ❑ **Logarithmic Function**
  - ❑ **Exponential Function**
  - ❑ **Power Function**
  - ❑ **Gamma Correction**

Image Enhancement in Spatial Domain

Process of improving quality & information content of original image.

ex. contrast, brightness, slicing, filtering etc.

Enhancement domain → Procedures directly applied on pixels —

$$\boxed{g(x,y) = T[f(x,y)]}$$

$g(x,y)$ → Transformed image

T → tr

$f(x,y)$ → input image

Point Transform

- Linear Point Transforms
- Non linear Point Transforms
- Piecewise linear Transforms

$\boxed{a \times f(x,y) + b}$ → linear

Linear Point Transformations

* Identity or Inverse

Inversion ← imp linear transform

↓

digital negative operation

binary image → 0 & 1

$\boxed{\begin{array}{l} 0 \rightarrow 1 \\ 1 \rightarrow 0 \end{array}}$

back → white
white → black

* NOT operation
* 8 bit grey scale image

$\boxed{2^8 = 256}$

black → white
0 → 1

mathematically,

$$\boxed{g(x,y) = L - 1 - f(x,y)}$$

$g(x,y)$ → Transformed image

$L - 1$ → no. of grey levels

$f(x,y)$ → i/p image

Non-Linear Point Transformations

i/p $\xrightarrow[\text{not linear}]{}$ o/p

- * Square Function
- * Square Root
- * Logarithmic f^n
- * Exponential f^n
- * Power Function
- * Gamma Correction

① Square f^n →

* enhance the contrast of image.
* Saturation → all pixels have high values

② Square root →

* Expand the grey scale to dark areas of the image.
* also reduces the dynamic range of light areas of image.

$$\boxed{g(x,y) = \sqrt{255 \times f(x,y)}}$$

(3) Logarithmic function →

* compress the dynamic range of image.

$$\boxed{g(x,y) = a \log[f(x,y)]}$$

$\log(0)$ → not defined

Transformation can be written as—

$$\boxed{g(x,y) = a \log\left[1+(e^{\alpha}-1)\, f(x,y)\right]}$$

a → constant

α → used to control the i/p range

$$\boxed{a = \frac{255}{\log\, 1 + \max . f(x,y)}}$$

(4) Exponential function →

→ Reverse of logarithmic function

$$\boxed{g(x,y) = e^{f(x,y)}}$$

modified exponential f^n can be written as –

$$\boxed{g(x,y) = c\left[(1+\alpha)^{f(x,y)} - 1\right]}$$

(c ↓ scaling factor)

* if $f(x,y) = 0$ then $g(x,y) = 0$
* Application ⇒ ✓ used to enhance the details in high value region of image

✓ Decreases dynamic range in low value region.

(5) Power function →

$$\boxed{g(x,y) = c \times f(x,y)^{\gamma}}$$

c & γ → positive constants

* if $\gamma > 1$, contrast of high value portion is increased.
* if $\gamma < 1$, contrast of high value portion is reduced.
* $\gamma = 1$, operation performs range scaling

(6) Gamma correlation →

↑ describe the nonlinearity of computer display monitor.

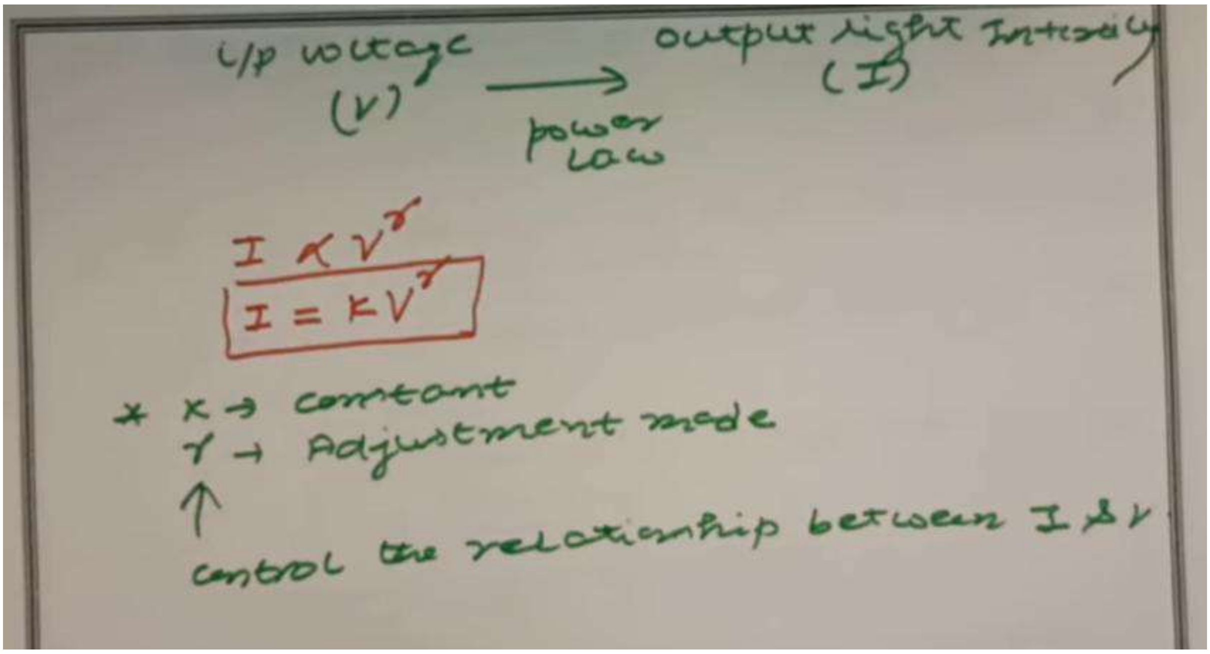


Image Enhancement in Spatial Domain (Part-2)

Image Enhancement in Spatial Domain

- **Piecewise Linear Transforms**
    - **Contrast Stretching & Its Variants**
    - **Grey Level Slicing**
    - **Bit Plane Slicing**

Piecewise Linear Functions

* premitive ←
* used to manipulate the contrast of image
PLT

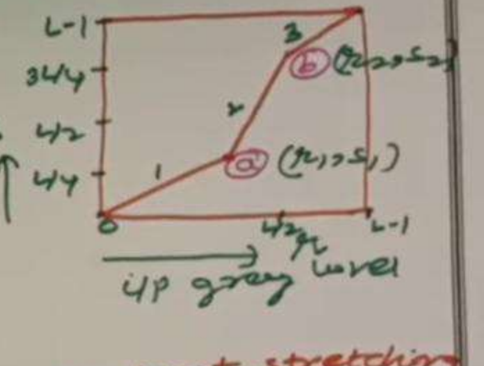
* Advantage → use simple functions to represent complex functions
* Disadvantage → they require many i/p parameters for image.

* contrast stretching & its variants
* Grey level slicing
* Bit Plane slicing

r → grey level of i/p image
s → Transformed function

① contrast stretching & its variants

darker → more dark
lighter → more light



* if $r_1 = s_1$ & $r_2 = s_2$
  Transformation is a linear f^n
  ↓
  no change in grey levels.

contrast stretching
(form of transformation f^n)

* if $r_1 = r_2$ & $s_1 = 0$, $s_2 = L-1$

Transformation becomes a thresholding function
→ creates binary image

* Intermediate values of (r_1, s_1) & (r_2, s_2)
→ produces various degrees of spread in grey level of o/p image.

* in general

$\boxed{r_1 \le r_2}$ & $\boxed{s_1 \le s_2}$

f^n is single valued & monotonically increasing.

The formulation of contrast stretching can be given as —

$$s = \begin{cases} l \cdot r & \text{for } 0 \le r \le a \\ m \cdot (r-a) + v & \text{for } a \le r \le b \\ n(r-b) + w & \text{for } b < r \le L-1 \end{cases}$$

where, $l, m, n \rightarrow$ slopes

$l, n < 1$
$m > 1$

many variants are available →

* range normalization
* clipping
* binarization i.e. thresholding
* posterizing i.e. multiple thresholding

(a) Range normalization →

* used to enhance image quality by altering range of grey levels present.

image range grey level $a \rightarrow b$

contrast Range $c \rightarrow d$

* subtract a from each grey level so range becomes $0 — (b-a)$.
* multiply the result by $\left(\frac{d-c}{b-a}\right)$. Now range becomes $0 — (d-c)$.
* Add c to the result. Now range becomes $c-d$.

This transformation can be described as—

$$\boxed{g(x,y) = \frac{(d-c)}{(b-a)} \times \left[f(x,y) - a\right] + c}$$

* **Clipping or Thresholding** $[0, L]$

* very light & dark areas

The grey scaling function can be written as—

$$G = \begin{cases} 0 & \text{if } 0 \le f(x,y) \le t_0 \\ \frac{255}{t_1 - t_0} & \text{if } t_0 \le f(x,y) < t_1 \\ L & \text{if } f(x,y) \ge t_1 \end{cases}$$

threshold t_0 & t_1 are used to determine the histogram of given image.

transformation can be written as →

$$g(x,y) = \begin{cases} 0 & \text{if } f(x,y) \ge T \\ 1 & \text{otherwise} \end{cases}$$

* multiple thresholding →

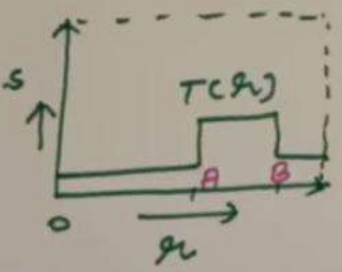
extension of thresholding.

$$g = \begin{cases} g_1 & \text{if } 0 \le f_i < t_1 \\ g_2 & \text{if } t_1 \le f_i < t_2 \\ \vdots & \\ g_n & \text{if } t_{n-1} \le f_i \le 255 \end{cases}$$

Grey Level slicing

* highlight specific range of grey values
* Two approaches →

① First approach → display high value for all grey levels in the range of interest & low level for remaining one.



$$s = \begin{cases} L-1 & \text{if } a \le r \le b \\ 0 & \text{otherwise} \end{cases}$$

This method is known as grey level slicing without background

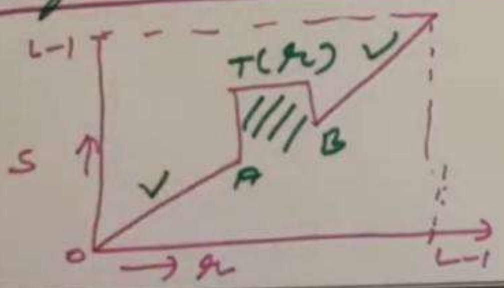
* Second method is based on transformation,
* it preserves background also.

Formulation can be –

$$s = \begin{cases} L-1 & \text{if } a \le r \le b \\ r & \text{otherwise} \end{cases}$$

↑ original content as it is

$\boxed{\text{grey level slicing with background.}}$

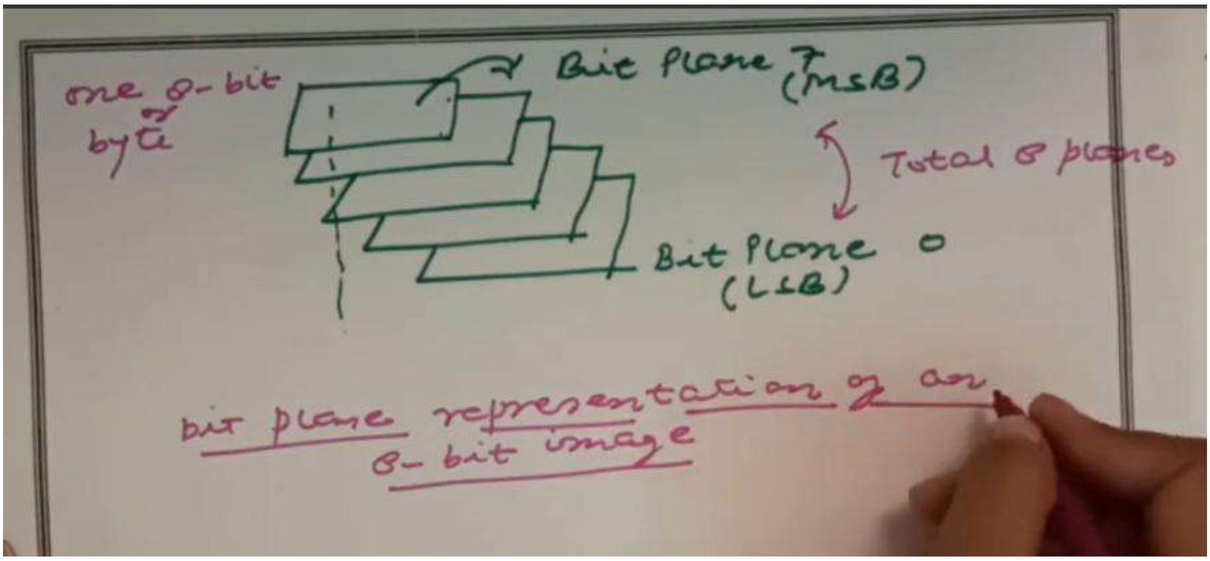


Bit Plane slicing

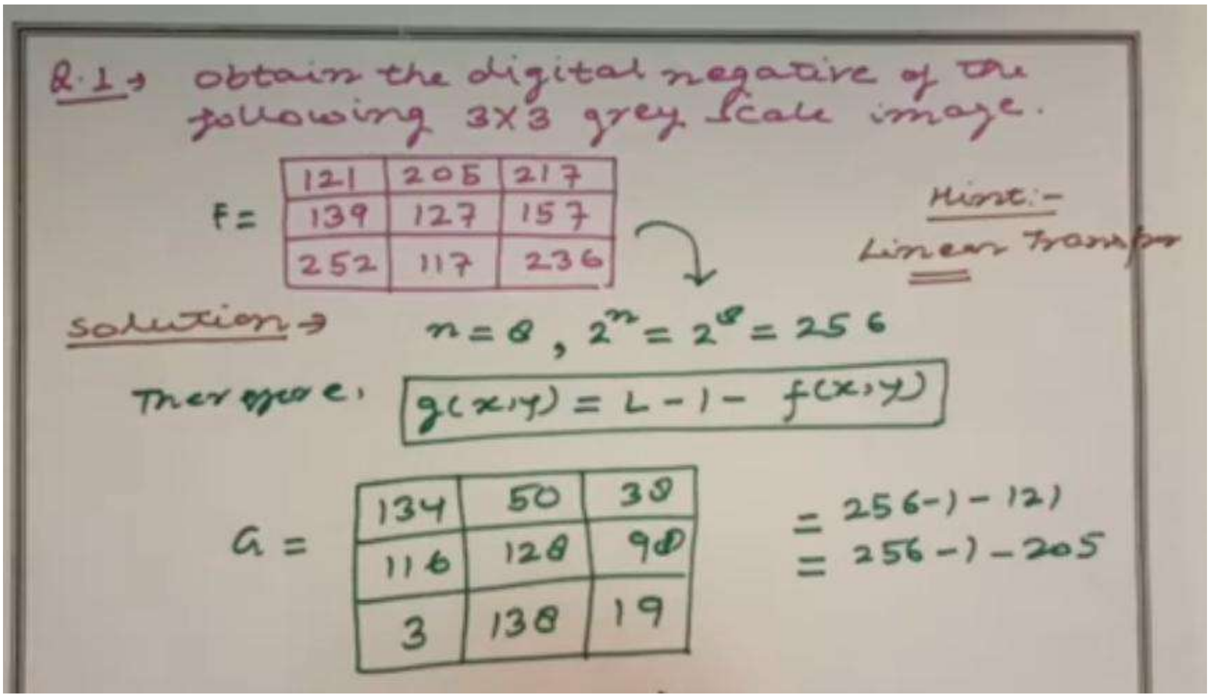
* contribution made by each bit to grey image need to be understood.
* image ⇒ $256 \times 256 \times 8$
  ↑ no. of bits required to represent each pixel.
* 8 bit → 256 grey levels

Black → 0000 0000 ⟵ 256 grey levels
white → 1111 1111 ⟵

LSB value of each pixel & plot the image using LSBs → continues for each bit till MSB



Numericals on Image Enhancement in Spatial Domain



Q.1 → obtain the digital negative of the following 3X3 grey scale image.

F =

121	205	217
139	127	157
252	117	236

Hint:- Linear Transfor

Solution → $n = 8$, $2^n = 2^8 = 256$

Therefore, $\boxed{g(x,y) = L - 1 - f(x,y)}$

G =

134	50	38
116	128	98
3	138	19

$= 256 - 1 - 121$
$= 256 - 1 - 205$

Q-2 → Given that grey level is 0-7 i.e. 8. Apply the following transformation —

- Inversion
- Square Root
- Square function
- Logarithm function

$c = 1$
$r = 1.2$

$f(x,y) =$ (given image)

1	2	3	4
5	5	6	6
6	7	6	6
6	7	2	3

Solution → **Inversion**

$g(x,y) = L - 1 - f(x,y)$

$L = 8$

$L - 1 = 7$

$$\boxed{g(x,y) = 7 - f(x,y)}$$

$g(x,y) =$

6	5	4	3
2	2	1	1
1	0	1	1
1	0	5	4

Square Root

$g(x,y) = \sqrt{7 \times f(x,y)}$

$g(x,y) =$

2.6	3.7	4.5	5.2
5.9	5.9	6.4	6.4
6.4	7	6.4	6.4
6.4	7	3.7	4.5

$g(x,y) =$

3	4	5	5
6	6	6	6
6	7	6	6
6	7	4	5

Square Function

$g(x,y) = [f(x,y)]^2$

$g(x,y) =$

1	4	9	16
25	25	36	36
36	49	36	36
36	49	4	9

$g(x,y) =$

1	4	7	7
7	7	7	7
7	7	7	7
7	7	4	7

Power Function

$g(x,y) = [c\, f^{r}(x,y)]$

$c = 1, \; r = 1.2$

$g(x,y) =$

1	2	4	5
7	7	9	9
9	10	9	9
9	10	2	4

$\Rightarrow g(x,y) =$

1	2	4	5
7	7	7	7
7	7	7	7
7	7	2	4

Logarithm Function

$g(x,y) = a \log [f(x,y) + 1]$

$a = 0.5$

$g(x,y) =$

1	1.15	1.23	1.3
1.35	1.35	1.34	1.39
1.39	1.42	1.39	1.39
1.39	1.42	1.15	1.23

$g(x,y) =$

1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1

Q-3 → A grey level image is ⇒ $F = \begin{bmatrix} 10 & 15 \\ 20 & 50 \end{bmatrix}$

Its range is given as 10 – 60. Transform this image to another image (120 – 100). What should be the grey level of transformation.

Hint:- Range Normalization

Solution → $\boxed{g(x,y) = \frac{(d-c)}{(b-a)}[f(x,y)-a]+c}$

$d = 100,\ c = 120,\ b = 60,\ a = 10$

$$g(x,y) = \left[\frac{100-120}{60-10}\left[f(x,y)-10\right]+120\right]$$

$g(x,y) =$

120	126
132	168

Q-4 → Apply grey level slicing on image —

$F =$

3	4	5
6	6	7
1	2	2

Let $r_1 = 3$, $r_2 = 6$

$\boxed{L = 8}$

HINT:-

* without background

$$s = \begin{cases} L-1 & r_1 \le r \le r_2 \\ 0 & \text{otherwise} \end{cases}$$

* with background

$$s = \begin{cases} L-1 & r_1 \le r \le r_2 \\ 1 & \text{otherwise} \end{cases}$$

Solution →

* without Background →

$g(x,y) =$

7	7	7
7	7	0
0	0	0

0 ← [3 – 6] → 0

L-1

7

* with Background →

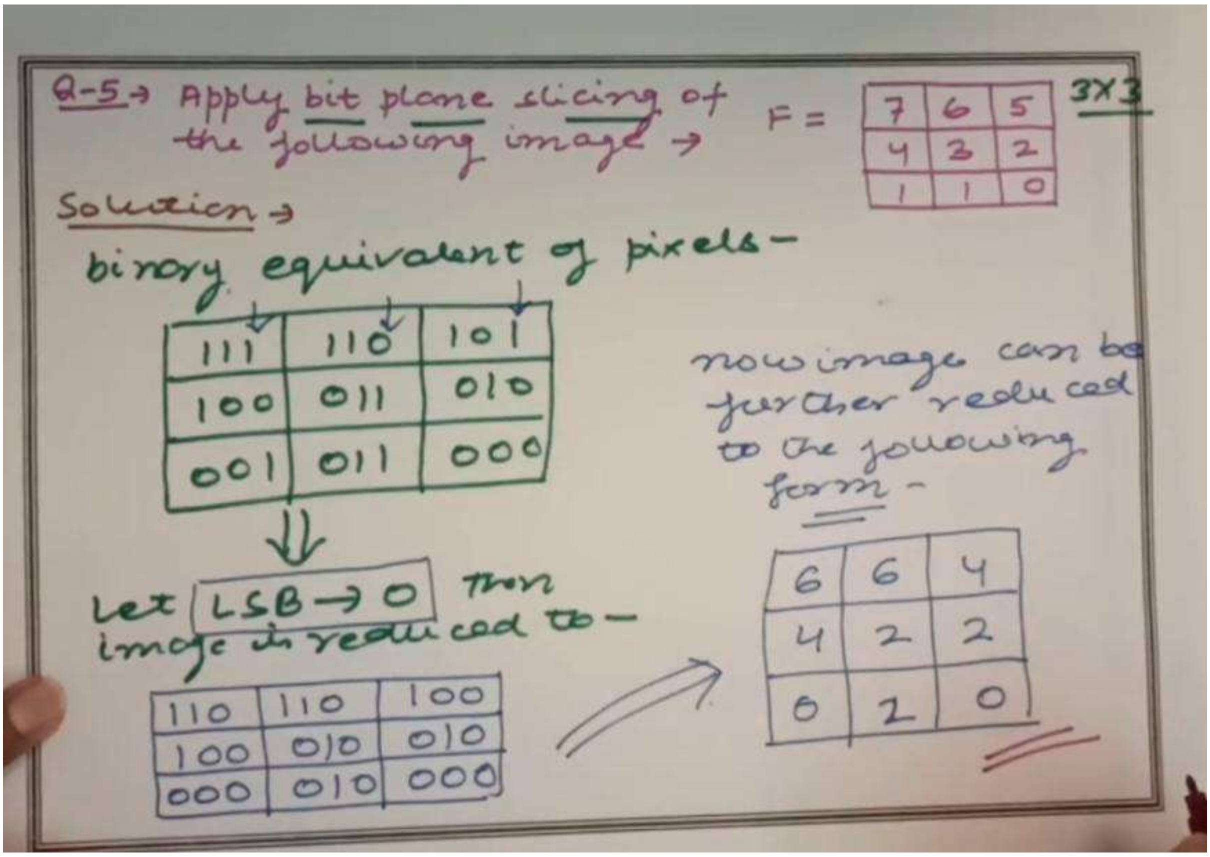
$g(x,y) =$

7	7	7
7	7	1
1	1	1

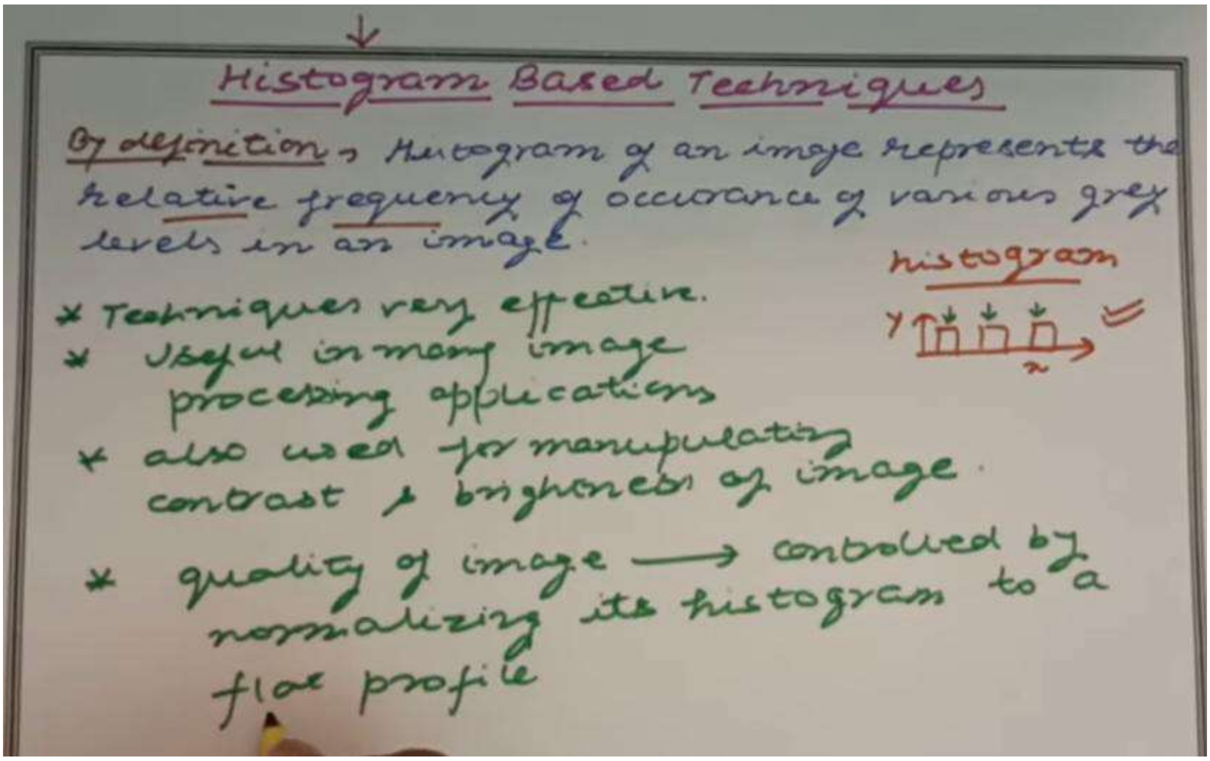
1 ← [3 – 6] → 1

L-1

7



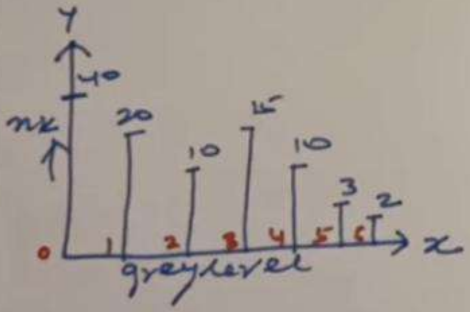
Histogram based Techniques | Histogram Stretching & Sliding



Histogram of an image can be Plotted in Two ways:—

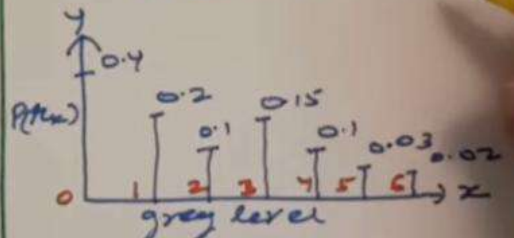
METHOD-1

Grey level	No. of Pixels (n_k)
0	40
1	20
2	10
3	15
4	10
5	3
6	2



METHOD-2

grey level	no. of Pixels (n_k)	Probability of occurance $P(r_k) = \frac{n_k}{n}$
0	40	0.4
1	20	0.2
2	10	0.1
3	15	0.15
4	10	0.1
5	3	0.03
6	2	0.02
	$n \rightarrow \boxed{100}$	

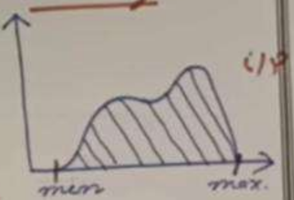


normalized histogram

Histogram stretching

* It is used to increase dynamic range.
* Here histogram is spread to cover entire dynamic range.

It is also known as histogram scaling.



linear stretching

mapping function for histogram stretching -

$$\frac{S_{max} - S_{min}}{r_{max} - r_{min}}$$

$r_{min} \rightarrow$ min grey level value of i/p

r_{max} = max grey level value of o/p

& Transform is given as —

$$\boxed{S = \frac{S_{max} - S_{min}}{r_{max} - r_{min}} (r - r_{min}) + S_{min}}$$

$S_{max} \rightarrow$ max. pixel value in stretched histogram

Ex. → Apply histogram stretching on 8×8, 8-level grey image. The gray level distribution is—

Grey level (r_k)	0	1	2	3	4	5	6	7
no. of Pixels (p_k)	0	0	20	20	5	19	0	0

Solution → Transform can be given as—

$$S = \frac{S_{max.} - S_{min}}{r_{max.} - r_{min}}(r - r_{min}) + S_{min.}$$

$r_{min} = 2$ $r_{max} = 5$

$S_{min} = 0$ $S_{max.} = 7$

$$\boxed{S = \frac{7}{3}(r-2)}$$

$r \to 2, 3, 4, 5$

when $r = 2$, $S = 0$

$r = 3$, $S = 2.33 \simeq 2$

$r = 4$, $S = 4.66 \simeq 5$

$r = 5$, $S = 7$

$r = 6$, $S = 7$

$r = 7$, $S = 7$

grey level S_k	no. of Pixels P_k
0	20
1	0
2	20
3	0
4	0
5	5
6	0
7	19

Histogram sliding

* This operation can make an image either darker or lighter.
* It retains the relationship between grey level values.

It can be represented as—

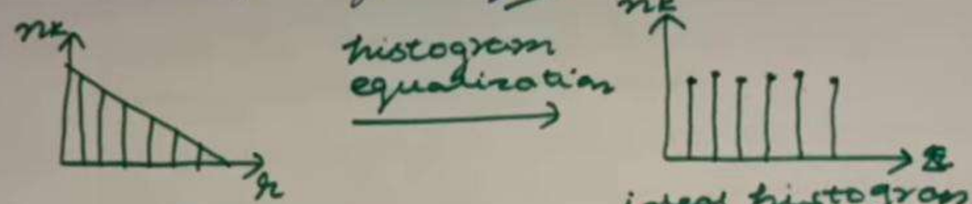
$$\boxed{S = \text{slide}(r) = r + \text{Offset}}$$

offset is the extent image or its histogram needs to be slide.

* +ve → Increases brightness
* −ve → creates a darker image

Histogram Equalization

* It is similar to histogram stretching.
* It tries to flatten the histogram to create better quality image.



ideal histogram equalization

$$\boxed{s = T(r)}$$

$0 \le r \le 1$

* $T(r)$ must be single valued & monotonically increasing in nature.
* $0 \le T(r) \le 1$ for $0 \le r \le 1$

Histogram equalization process is as: —

(1) Form the cumulative histogram.

(2) Normalize the value by dividing it by the total number of pixels.

(3) Multiply the value by the max. grey level value & round off value.

(4) Map the original value to the result of step 3 by one to one correspondence.

Ex:- Perform histogram equalization for the 8×8, 8-level image -

r_k	0	1	2	3	4	5	6	7
p_k	8	10	10	2	12	16	4	2

solution → Equalization process for given image is -

r_k	p_k	cumulative running of pixels	$\left(\frac{\text{cumulative}}{\text{Total}}\right) \times (L-1)$	Round off
0	8	8	$8/64 \times 7 = 0.87$	1
1	10	18	$18/64 \times 7 = 1.96$	2
2	10	28	$= 3.06$	3
3	2	30	$= 3.28$	3
4	12	42	$= 4.59$	5
5	16	58	$= 6.34$	6
6	4	62	$= 6.78$	7
7	2	64	$= 7$	7

Final Image

grey level r_k	no. of pixels p_k
0	1
1	2
2	3
3	3
4	5
5	6
6	7
7	7

Histogram specification ←

* It is used for compressing the dynamic range of an image to avoid pixels having less information.
* It makes an image easier to view on video monitor.
* grey level of an image which have uniform PDF.

original image → $\boxed{r = T^{-1}(S)}$ → Transformed image

* $P_r(r)$ & $P_s(s)$ → grey levels PDF of i/p & o/p

K → some intermediate result i.e.

$K = T_1(r) = \int_0^r P_r(r)\,dr$

$K = T_2(S) = \int_0^s P_s(s)\,dr$

$P_K(K)$ → uniform

Transformation

$\boxed{S = T_2^{-1}(T_2(r))}$

The algorithm for histogram specification is as:-

(1) Find the mapping table of histogram equalization.

(2) specify the desired histogram. Equalize the desired histogram.

(3) Perform the mapping process so that the values of step 1 can be mapped to the result of step 2.

Ex. Perform histogram specification on 8×8, 8 level image -

p_k	8	10	10	2	12	16	4	2
r_k	0	1	2	3	4	5	6	7

* equalization process already explained.

Solution → Target histogram-

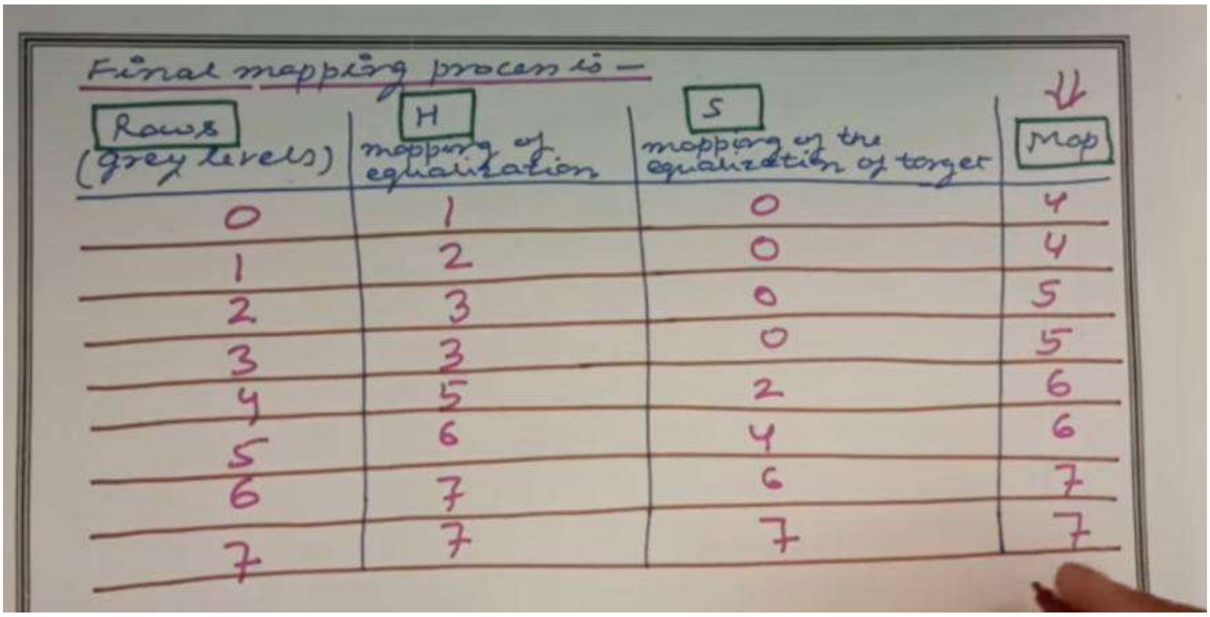
r_k	0	1	2	3	4	5	6	7
p_k	0	0	0	0	20	20	16	8

equalization of target histogram →

r_k	p_k	cumulative pixels	$\frac{\text{cumulative}}{\text{Total}} \times (L-1)$ (7)	Round off
0	0	0	0	0
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	20	20	$20/64 \times 7 = 2.18$	2
5	20	40	$= 4.3$	4
6	16	56	$= 6.1$	6
7	8	64	$= 7$	7

mapping

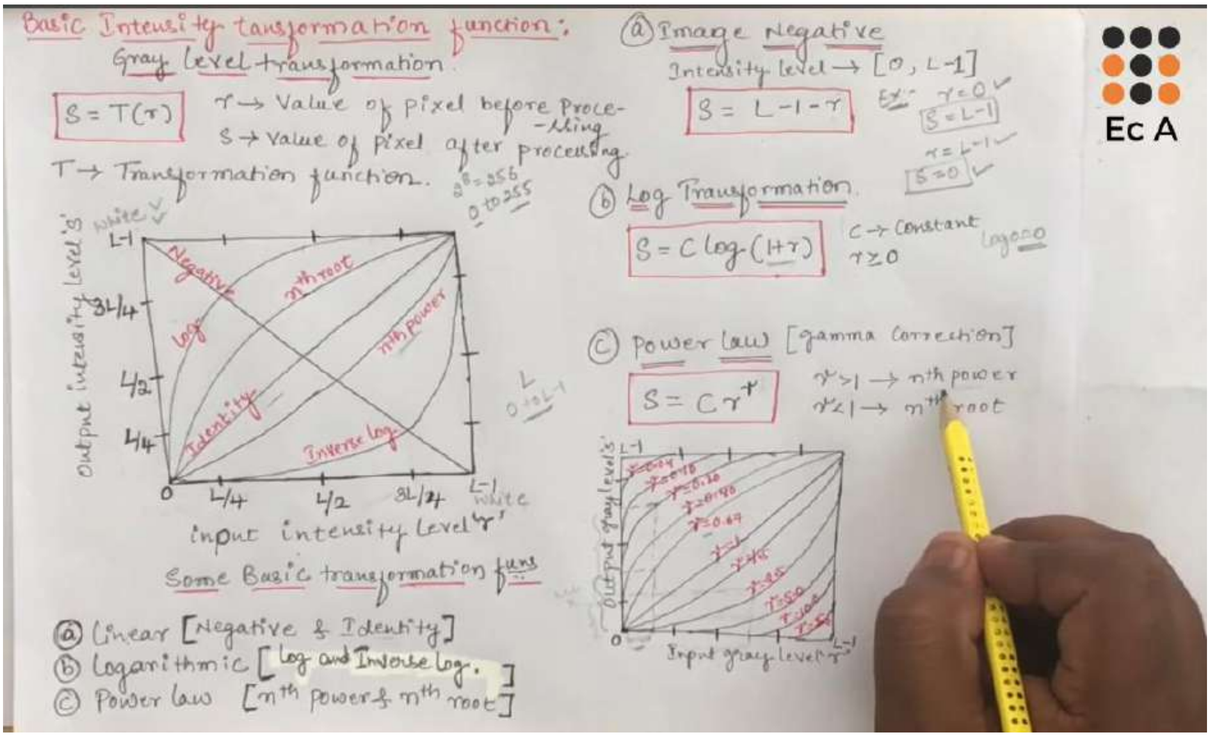
r_k	Closest level
0	0
1	0
2	0
3	0
4	2
5	4
6	6
7	7



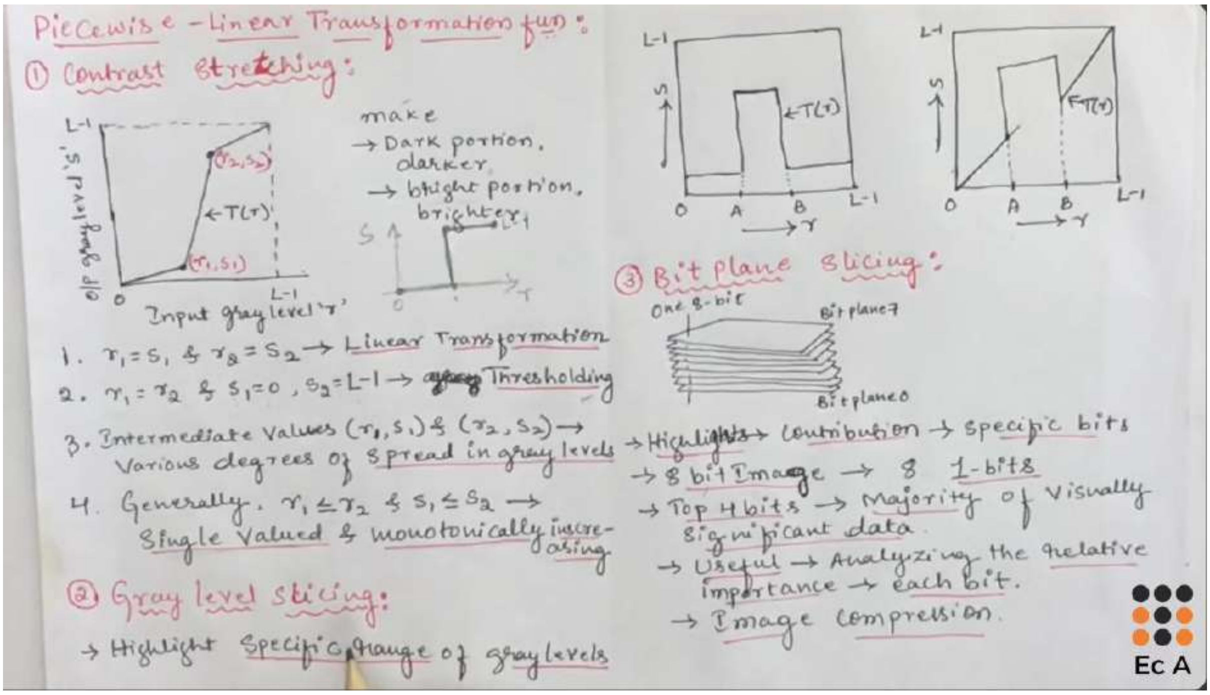
Final mapping process is —

Rows (grey levels)	H mapping of equalization	S mapping of the equalization of target	Map
0	1	0	4
1	2	0	4
2	3	0	5
3	3	0	5
4	5	2	6
5	6	4	6
6	7	6	7
7	7	7	7

Basic intensity (gray level) transformation in Digital Image processing



Piecewise linear transformation



Piecewise - Linear Transformation fun:

① Contrast Stretching:

make
→ Dark portion, darker
→ bright portion, brighter

1. $r_1 = s_1$ & $r_2 = s_2$ → Linear Transformation
2. $r_1 = r_2$ & $s_1 = 0$, $s_2 = L-1$ → Thresholding
3. Intermediate Values (r_1, s_1) & (r_2, s_2) → Various degrees of spread in gray levels
4. Generally, $r_1 \le r_2$ & $s_1 \le s_2$ → Single valued & monotonically increasing

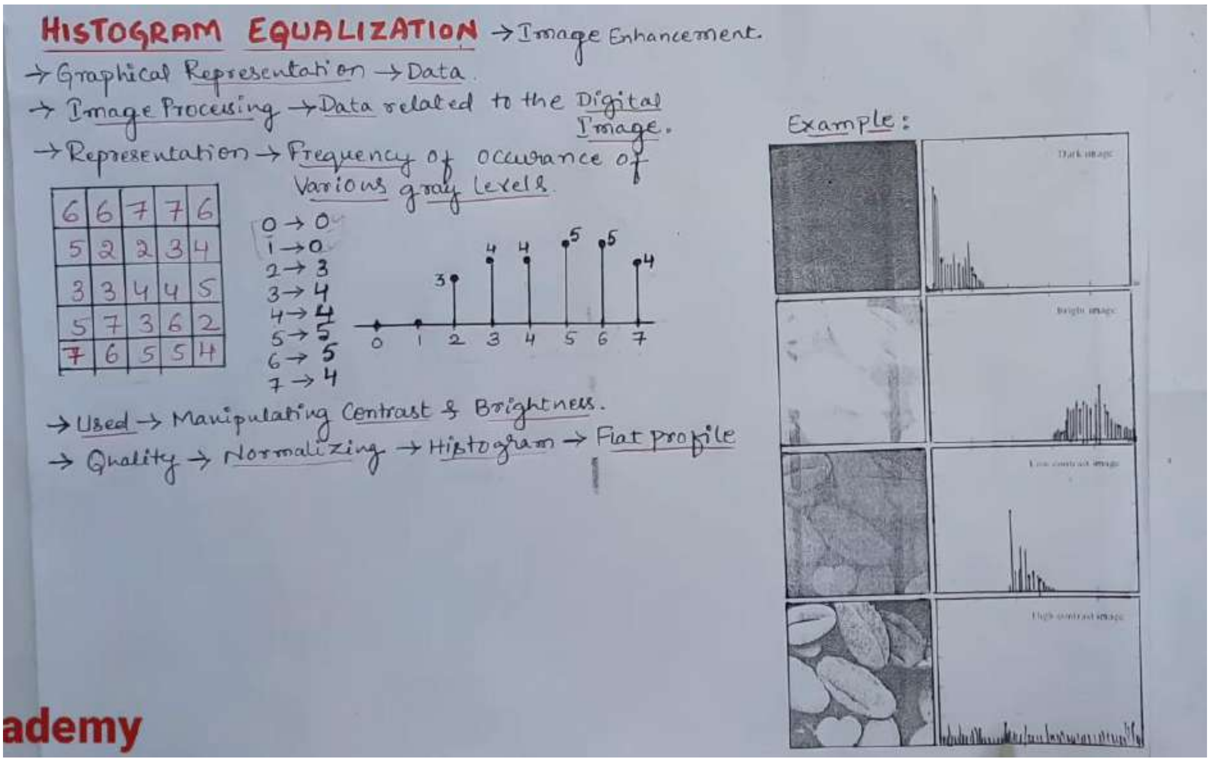
② Gray level slicing:

→ Highlight specific range of gray levels

③ Bit Plane Slicing:

→ Highlights → contribution → specific bits
→ 8 bit Image → 8 1-bits
→ Top 4 bits → Majority of visually significant data
→ Useful → Analyzing the relative importance → each bit.
→ Image compression.

Histogram equalization



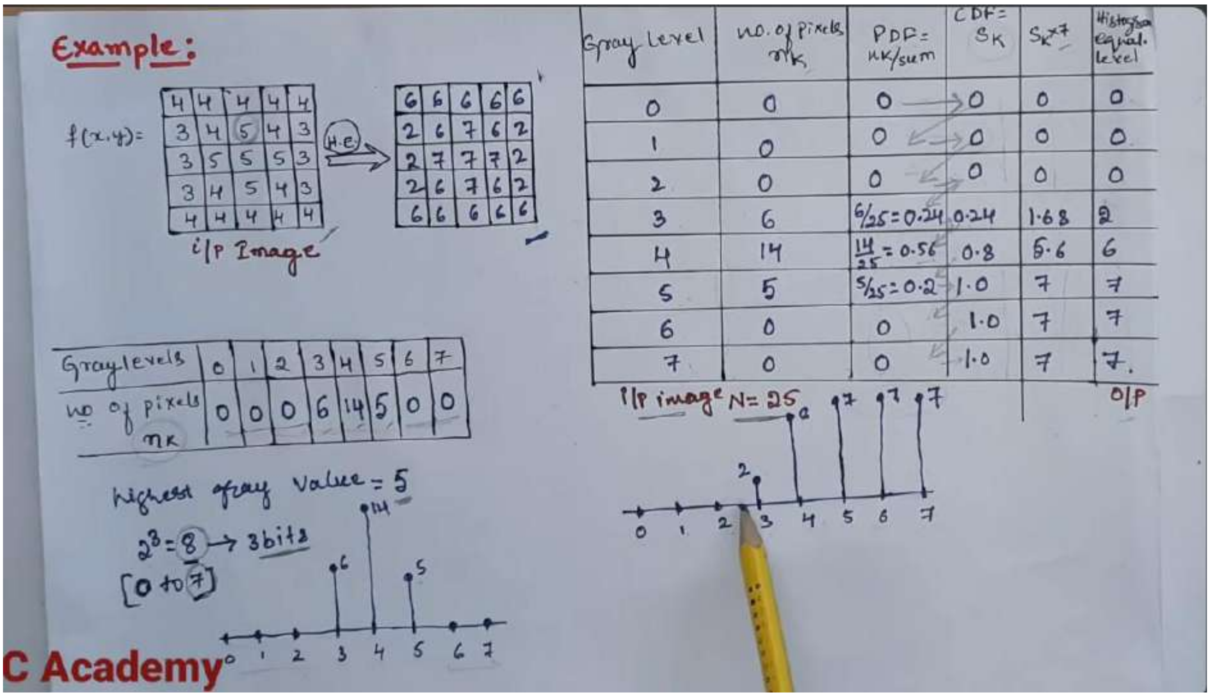
HISTOGRAM EQUALIZATION → Image Enhancement.
→ Graphical Representation → Data
→ Image Processing → Data related to the Digital Image.
→ Representation → Frequency of occurance of Various gray levels

6	6	7	7	6
5	2	2	3	4
3	3	4	4	5
5	7	3	6	2
7	6	5	5	4

0 → 0
1 → 0
2 → 3
3 → 4
4 → 4
5 → 5
6 → 5
7 → 4

→ Used → Manipulating Contrast & Brightness.
→ Quality → Normalizing → Histogram → Flat profile

Example:



Example:

$f(x,y) =$

4	4	4	4	4
3	4	5	4	3
3	5	5	5	3
3	4	5	4	3
4	4	4	4	4

i/p Image $\xrightarrow{H.E}$

6	6	6	6	6
2	6	7	6	2
2	7	7	7	2
2	6	7	6	2
6	6	6	6	6

Graylevels	0	1	2	3	4	5	6	7
no. of pixels n_K	0	0	0	6	14	5	0	0

highest gray value = 5

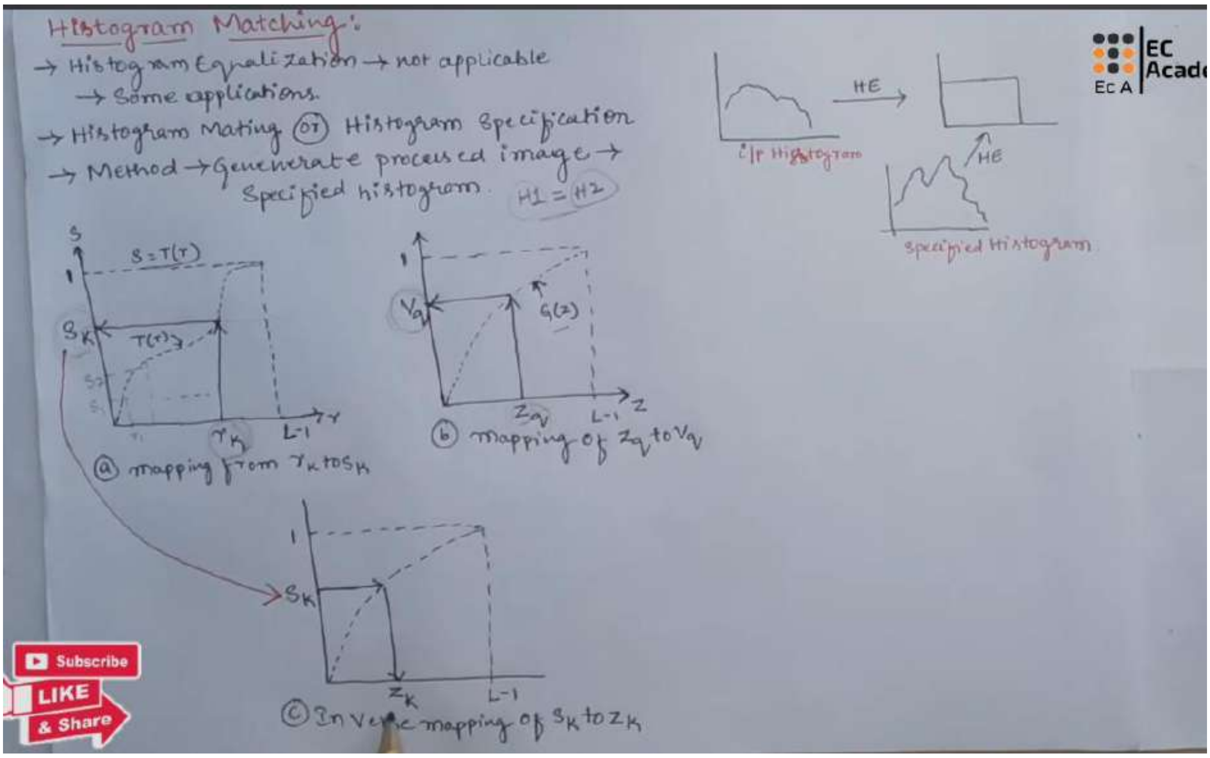
$2^3 = 8 \rightarrow$ 3bits

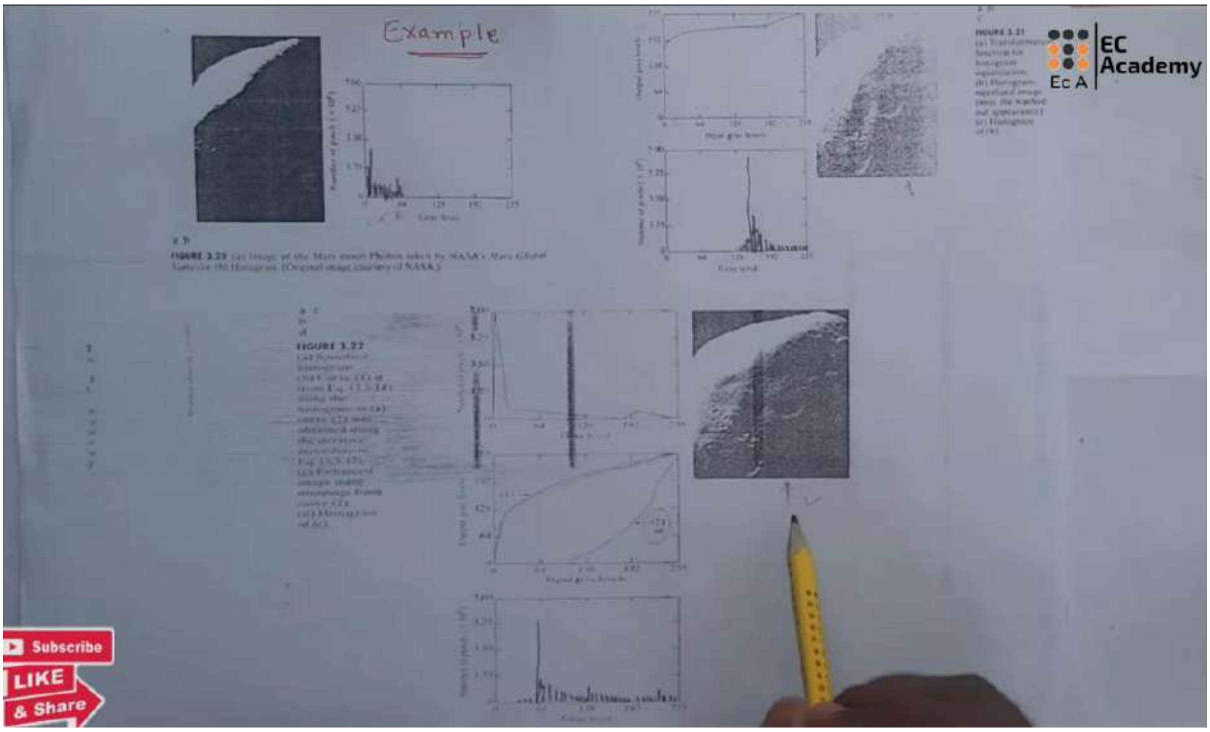
[0 to 7]

Gray Level	no. of pixels n_K	PDF = n_K/sum	CDF = S_K	$S_K \times 7$	Histogram Equal. level
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	6	6/25 = 0.24	0.24	1.68	2
4	14	14/25 = 0.56	0.8	5.6	6
5	5	5/25 = 0.2	1.0	7	7
6	0	0	1.0	7	7
7	0	0	1.0	7	7

i/p image N = 25 — o/p

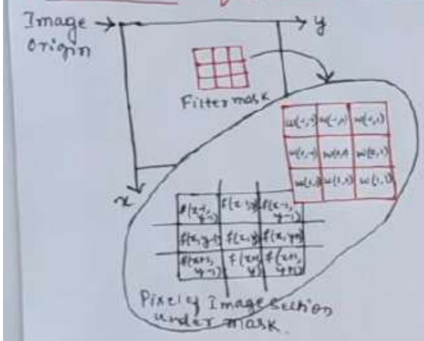
Histogram matching





Fundamentals of spatial filtering

Fundamentals of Spatial Filtering:



→ Response of Linear Filter

$$R = w(-1,-1)f(x-1,y-1) + w(-1,0)f(x-1,y) + \cdots\cdots w(0,0)f(x,y) + \cdots\cdots + w(1,0)f(x+1,y) + w(1,1)f(x+1,y+1) \longrightarrow ①$$

→ mask $m \times n \Rightarrow m = 2a+1$ & $n = 2b+1$

Linear Filtering of Image of size $M \times N$ and mask size $m \times n$

$$g(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t) \rightarrow ②$$

where $a = \frac{m-1}{2}$ & $b = \frac{n-1}{2}$

eqn ② => Convolution mask (or) Convolution Kernel.

by simplifying

$$R = w_1 z_1 + w_2 z_2 + \cdots\cdots + w_{mn} z_{mn}$$

w → mask co-efficients

z → the values of the image gray level.

$m \times n$ → total no of co-efficients

$$R = \sum_{i=1}^{mn} w_i z_i \longrightarrow ③$$

For 3×3 general mask

$$R = w_1 z_1 + w_2 z_2 + \cdots\cdots + w_9 z_9$$

$$R = \sum_{i=1}^{9} w_i z_i \longrightarrow ④$$

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

mask Co-efficients

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Sub image

Spatial Filtering Concepts

Spatial Filtering Concepts

The word "filtering" comes from the frequency domain where "filters" are classified as:

- ❑ Low-pass (i.e., preserve low frequencies)
- ❑ High-pass (i.e., preserve high frequencies)
- ❑ Band-pass (i.e., preserve frequencies within a band)
- ❑ Band-reject (i.e., reject frequencies within a band)

Many image enhancement techniques are based on spatial operations which are performed on local neighborhood of the input pixels.

- ❑ Spatial masks are also known as window, filter, kernel, template.
- ❑ They are used and convolved over the entire image for local enhancement (spatial filtering)

Filters in image processing can be characterized into the following three categories:

Convolution based Filters → They use spatial masks which is convolved with given image to achieve required effect.

Order-statistics Filters → They simply arrange the pixels that are under the mask in a desired order.

Hybrid Filters → They use the concepts of ranking and convolution.

Spatial Filtering Concepts

A spatial filter is characterized by the following two aspects:

- Concept of neighborhood
- Convolution operation over neighborhood

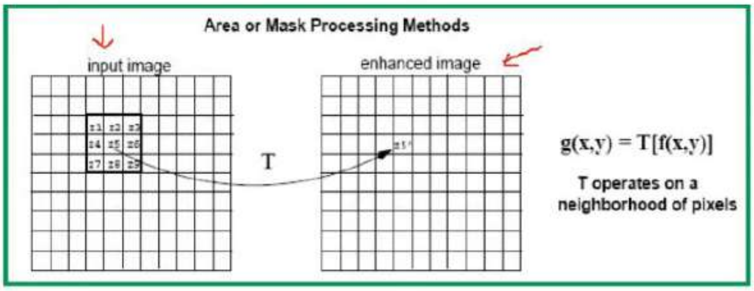
It can be further classified on the basis of nature of response

- Linear Spatial Filters
- Non-linear Spatial Filters

Linear Filters

A filter is called linear when its output is a linear combination of the inputs, e.g.,

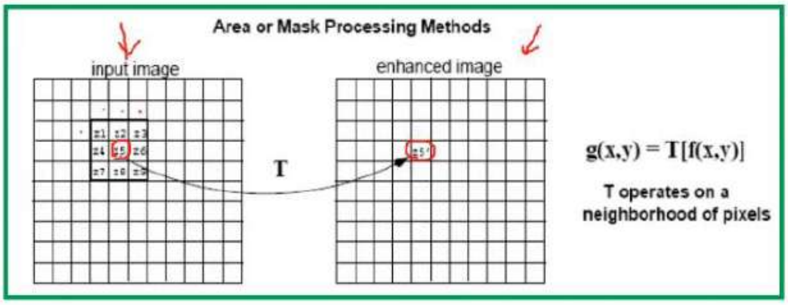
$$\boxed{z'_5 = 5z_1 - 3z_2 + z_3 - z_4 - 2z_5 - 3z_6 + z_8 - z_9 - 9z_7}$$



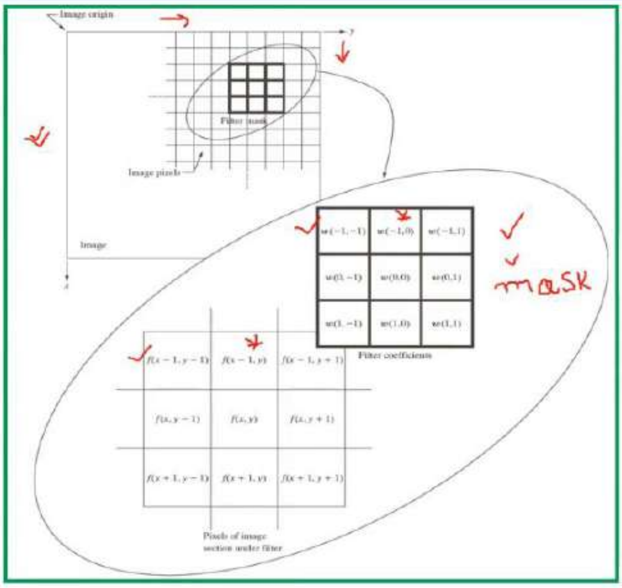
Non-linear Filters

A filter is called non-linear when its output is not a linear combination of the inputs, e.g.,

$$\boxed{z'_5 = \max(z_1, z_2, z_3, z_4, z_5, z_6, z_7, z_8, z_9)}$$



Spatial Filtering Concepts



The mechanics of spatial filtering

- an image of size M x N
- mask of size m x n

$m = 2a+1$
$n = 2b+1$

The resulting output gray level for any coordinates x and y is given by:

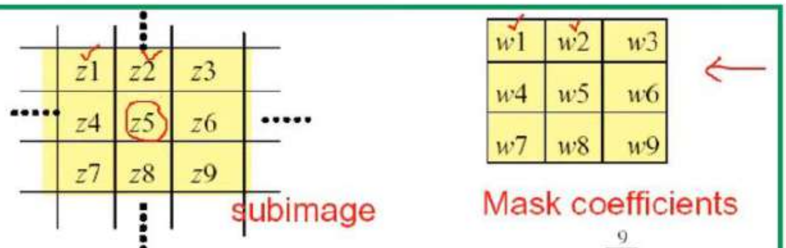
$$g(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)$$

where $a = (m-1)/2, \quad b = (n-1)/2$

$x = 0, 1, 2, \ldots\ldots, M-1, \quad y = 0, 1, 2, \ldots\ldots, N-1$

- ❑ Given the 3 × 3 mask with coefficients: $w_1, w_2, \ldots, w_9$
- ❑ The mask cover the pixels with gray levels: $z_1, z_2, \ldots, z_9$

Example



$$z_5 \leftarrow z = w_1 z_1 + w_2 z_2 + \cdots + w_9 z_9 = \sum_{i=1}^{9} w_i z_i$$

z gives the output intensity value for the processed image (to be stored in a new array) at the location of z_5 in the input image

- ❑ The size of the masks determines the number of neighboring pixels which influence the output value at (x,y).
- ❑ The values (coefficients) of the mask determine the nature and properties of enhancing technique.

Spatial Correlation

The correlation of a filter $w(x, y)$ of size $m \times n$ with an image $f(x, y)$, denoted as $w(x, y) \quad f(x, y)$

$$w(x, y) \bullet f(x, y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s, t) f(x+s, y+t)$$

Spatial Convolution

The convolution of a filter $w(x, y)$ of size $m \times n$ with an image $f(x, y)$, denoted as $w(x, y) \;\; f(x, y)$

$$w(x,y) * f(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x-s, y-t)$$

Two types of spatial filters are there:

- ❑Smoothing (low-pass) filters
- ❑Sharpening (high-pass) filters

Smoothing spatial filters

- ❑ **Smoothing Linear Filters**
- ❑ **Order-Statistics Filters**

- Smoothing filters are used for blurring and for noise reduction
- Blurring is used in preprocessing steps, such as removal of small details and bridging of small gaps in lines or curves
- Smoothing spatial filters include linear filters and nonlinear filters.

Mean Filter / Box Filter

Smoothing filter is a linear filter that creates an image with smooth appearance by blurring the image and removing noise.

- The output (response) of smoothing, linear spatial filter is simply the average of the pixels contained in the neighbourhood of the filter mask.
- These filters are also known as averaging filters.

Consider the output pixel is positioned at the center

$$\frac{1}{9}\times\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix} \qquad \frac{1}{16}\times\begin{bmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{bmatrix}$$

Box filter all coefficients are equal

Weighted average give more (less) weight to pixels near (away from) the output location

Consider mask size:

$m\times n$

$$w_i = \frac{1}{mn}, i = 1, \ldots, mn$$

Averaging process eliminates the extreme values of a neighbourhood in the spatial domain.

Few more examples of box filters

5×5

$$\frac{1}{25} \times \begin{array}{|c|c|c|c|c|}\hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 \\ \hline \end{array}$$

7×7

$$\frac{1}{49} \times \begin{array}{|c|c|c|c|c|c|c|}\hline 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline \end{array}$$

The general implementation for filtering an $M \times N$ image with a weighted averaging filter of size $m \times n$ is given

$$g(x,y) = \frac{\sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)}{\sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t)}$$

where $m = 2a+1, \quad n = 2b+1.$

m n ↓ odd

Example:

If an image is given as:

What would be the output of box filter?

1	5	7
2	4	8
3	6	9

3×3

Solution:

$\frac{1}{9}(1+5+7+2+4+8+3+6+9)$

$= \frac{45}{9} = 5$

1	5	7
2	5	8
3	6	9

Gaussian Filters

To reduce the loss of visual information, it as possible to assign different weights to mask coefficients so that more importance is attached to some pixels. This is to ensure that filter has one peak.

Let $f(x-1)$, $f(x)$ & $f(x+1)$ ← 3 points

$$f(x) = \frac{1}{4} f(x-1) + \frac{1}{2} f(x) + \frac{1}{4} f(x+1)$$

This equation is equivalent to applying the following 1D mask:

1/4	1/2	1/4

Design of Discrete Gaussian Masks

The weights are samples of a 2D Gaussian function:

$$G_\sigma(x, y) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{x^2+y^2}{2\sigma^2}}$$

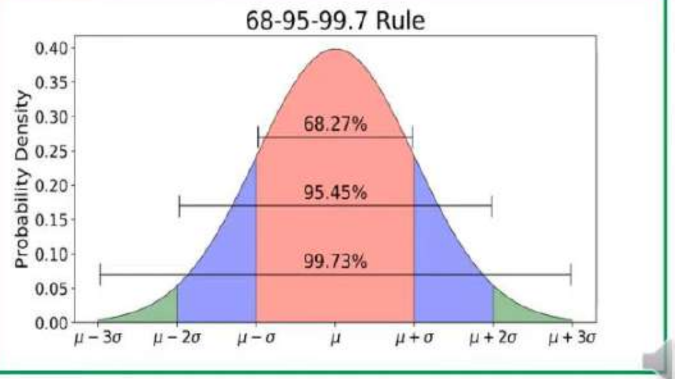
σ is the width of Gaussian function

7 × 7 Gaussian Mask

1	1	2	2	2	1	1
1	2	2	4	2	2	1
2	2	4	8	4	2	2
2	4	8	16	8	4	2
2	2	4	8	4	2	2
1	2	2	4	2	2	1
1	1	2	2	2	1	1

Mask size depends on **σ**

$height = width = 5\sigma$ (subtends 98.76% of the area)



Smoothing Spatial Filters:

→ Used for Blurring & Noise reduction

→ Blurring: Removal of small details from an Image. Prior to object extraction.

→ Noise reduction: blurring with a linear (or) Non linear filters.

SMOOTHING LINEAR FILTERS:

→ O/P → average of pixels contained in the neighborhood → Filter mask.

→ Averaging Filters (or) low pass Filters

Ex:-

(a) $\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ (b) $\frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$

3×3 Smoothing Filter mask

→ Replacing → each pixel → by avg. of gray levels.

→ Application → Noise reduction.

→ Side effect → blur edges.

→ fig (a) → Standard average of pixel values

→ m×n mask → $(1/m\times n)$

→ Box Filters

→ fig (b) → weighted avg

→ Pixel at the center of mask → more importance

→ This is to reduce blurring during smoothing process

→ general Implementation for Image → M×N & mask → m×n

$$g(x,y) = \frac{\sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)}{\sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t)}$$

$x = 0,1,2,\ldots M-1$ & $y = 0,1,2,\ldots N-1$

EC Academy

Order Statistics filters

Order Statistics Filters

- Order Statistics Filters are nonlinear spatial filters.
- Its response is based on ordering $\boxed{\text{(ranking)}}$ the pixels contained in the filter mask.
- Replacing the value of the center pixel with the value determined by the ranking result.
- Examples: Median filter, Max filter & Min filter

Median Filters are quite popular

- It assigns the mid value of all the gray levels in the mask to the center of mask
- Particularly effective when
  - the noise pattern consists of strong, spiky components (known as impulse noise, or salt-and-pepper noise)
  - edges are to be preserved
  - force points with distinct gray levels to be more like their neighbors

10	20	20
20	15	20
20	25	100

Output = ? **20**

Example: a 3 × 3 image is given.

10, 15, 20, 20, 20, 20, 20, 25, 100

Max. = 100

Min = 10

ORDER-STATISTICS Filters:

→ non linear Spatial Filters

→ Response → ordering [Ranking] the pixels → Image

→ Replacing the Center pixel value with value determined by ranking result.

1. Median Filter:

→ Replaces the value of a pixel by the median of the gray level.

→ Most popular → excellent noise reduction

→ less blurring

→ Effective for Impulse noise → Salt and pepper noise

Ex:

10	20	20
20	15	20
20	25	100

⇒

10	20	20
20	**20**	20
20	25	100

10, 15, 20, 20, 20, 20, 20, 25, 100

2. Max filter:

→ Finding the brightest point.

$$R = \max\{Z_k \mid k = 1, 2, 3 \ldots 9\}$$

3. Min filter:

→ Finding the darkest point.

$$R = \min\{Z_k \mid k = 1, 2, 3 \ldots 9\}$$

Ex:

max. Value : 100 – brightest point.

min. Value : 10 – darkest point.

Image sharpening

- ❑ Introduction
- ❑ Gradient Filter
- ❑ Laplace Filter
- ❑ High-boost Filter
- ❑ Unsharp Masking

Sharpening Spatial Filters

- ❑ Image sharpening filters highlight the details of an image.
- ❑ High spatial frequency components have detailed information in the form of edges and boundaries.

- ❑ Image sharpening algorithms are used to separate object outlines.
- ❑ Therefore Image sharpening filters are known as edge enhancement or edge crispening algorithms.

Blurring vs Sharpening

- ❑ Blurring/smooth is done in spatial domain by pixel averaging in a neighbours, it is a process of integration
- ❑ Sharpening is an inverse process, to find the difference by the neighborhood, done by spatial differentiation.

Derivative operator

- This operator calculates the gradient of the image intensity at each point, giving direction of the largest possible increase from light to dark and rate of change in that direction.
- Image differentiation
  - enhances edges and other discontinuities (noise)
  - deemphasizes area with slowly varying gray-level values.

Foundation of sharpening spatial filters

The basic definition of the first-order derivative of a one dimensional function f(x) is the difference

$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

The second-order derivative of a one-dimensional function f(x) is the difference

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$$

First and Second-order derivative of 2D

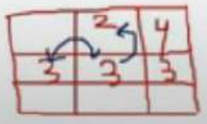
Let us consider an image function of two variables, f(x, y), at which time it will be dealed with partial derivatives along the two spatial axes.

Gradient operator (linear operator) $$\nabla \mathrm{f} = \frac{\partial f(x,y)}{\partial x \partial y} = \frac{\partial f(x,y)}{\partial x} + \frac{\partial f(x,y)}{\partial y}$$

Laplacian operator (non-linear) $$\nabla^2 f = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2}$$

Gradient Filter

- ❑ Edges can be extracted by taking the gradient of the image.
- ❑ Gradient refers to the difference between the pixels of an image.
  - ❖ If neighbouring pixels have same intensity, difference is zero and hence there is no edge.
  - ❖ Edges exit when there is a significant local intensity variation.



For function $f(x, y)$, the gradient of f at coordinates (x, y) is defined as

$$\nabla f \equiv \operatorname{grad}(f) \equiv \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix}$$

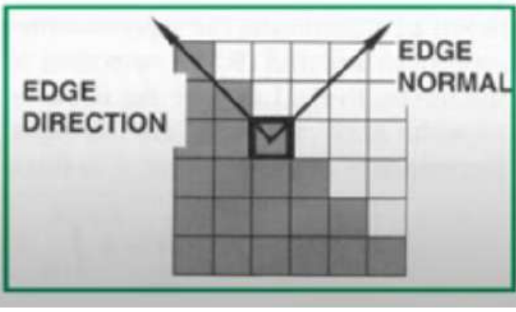
The *magnitude* of vector ∇f, denoted as $M(x, y)$

$$M(x, y) = \operatorname{mag}(\nabla f) = \sqrt{g_x{}^2 + g_y{}^2}$$

Gradient Image

❑ **Gradient magnitude:** provides information about edge strength.

❑ **Gradient direction:** perpendicular to the direction of the edge. It is useful in detecting sudden change in image intensity.



Laplacian Filter

❑ It is a derivative operator

- ❖ it highlights gray-level discontinuities in an image
- ❖ it deemphasizes regions with slowly varying gray levels it deemphasizes regions with slowly varying gray levels

❑ It tends to produce images that have

- ❖ grayish edge lines and other discontinuities, all superimposed on a dark
- ❖ featureless background

Laplacian Filter

The second-order isotropic derivative operator is the Laplacian for a function (image) f(x,y)

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

$$\frac{\partial^2 f}{\partial x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y)$$

$$\frac{\partial^2 f}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y)$$

$$\nabla^2 f = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y)$$

Laplacian Filter

Sample Laplacian Mask

A high pass filter can be implemented using a Laplacian mask.

0	1	0
1	−4	1
0	1	0

1	1	1
1	−8	1
1	1	1

0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

- Image enhancement of the image f(x, y) can be done by adding the Laplacian edge result.
- It enhances the high frequency contents and thereby gives an edge enhancement image.

Mathematically it can be written as:

$$g(x,y)=\begin{cases} f(x,y)-\nabla^2 f(x,y) & \text{if the center coefficient of the Laplacian mask is negative} \\ f(x,y)+\nabla^2 f(x,y) & \text{if the center coefficient of the Laplacian mask is positive} \end{cases}$$

High-boost Filters

- A high-boost filter is used to prevent the loss of low frequency components and restore the visual details of an image.
- This is done by adding an offset to the filtered image by retaining some low frequency components.
- A high pass filtered image can be computed as the difference between the original image and a low pass filtered version of image.

$$\begin{aligned} Highboost &= A\ Original - Lowpass \\ &= (A-1)\ Original + Original - Lowpass \\ &= (A-1)\ Original + Highpass \end{aligned}$$

Multiplying the original image by an amplification factor "A" gives high boost image.

if we use Laplacian filter to create sharpen image $f_s(x, y)$ with addition of original image

$$f_s(x,y)=\begin{cases} f(x,y)-\nabla^2 f(x,y) \\ f(x,y)+\nabla^2 f(x,y) \end{cases}$$

$$f_{hb}(x,y)=\begin{cases} Af(x,y)-\nabla^2 f(x,y) \\ Af(x,y)+\nabla^2 f(x,y) \end{cases}$$

High-boost mask

- $A \geq 1$
- If A = 1, it becomes standard laplacian sharpening
- Any N × N high boost mask can be created using the value of A = (N × N)-1

0	−1	0
−1	$A+4$	−1
0	−1	0

−1	−1	−1
−1	$A+8$	−1
−1	−1	−1

Unsharp Masking

- It is hybrid filter technique.
- This filter is useful for removing both impulse and Gaussian noise present in am image.

- An unsharp filter is simple sharpening operator. It enhances edges.
- This technique is commonly used in photographic and printing industries.

Sharpen images consists of subtracting an unsharp version of an image from the original image

$$f_s(x,y)=f(x,y)-\bar{f}(x,y)$$

Unsharp Masking

Procedure for implementing an unsharp mask is:

- ❑ Read the image
- ❑ Blur the original image
- ❑ Subtract the blurred image from the original
- ❑ Add the mask to the original

Let $\overline{f}(x, y)$ denote the blurred image, unsharp masking is

$$g_{mask}(x, y) = f(x, y) - \overline{f}(x, y)$$

Then add a weighted portion of the mask back to the original

$$g(x, y) = f(x, y) + k * g_{mask}(x, y) \qquad k \geq 0$$

when $k > 1$, the process is referred to as highboost filtering.

Unsharp masking is at k = 1

Sharpening Spatial Filters:

→ highlight the finate detail (or) to enhance details

→ Applications → Electronic printing, Medical imaging, Industrial Inspections and autonomous Guidance in military s/ms.

→ Image blurring → pixel averaging
  ↳ Integration

→ Sharpening → "Spatial differentiation"

→ Image differentiation → enhances edges and noise & deemphasizes areas with slowly varying gray-level values.

Foundation:

→ First order and second order derivatives.

→ Derivatives → defined interms of differences

→ definition for first derivative

→ (a) must be zero in flat segments

(b) must be nonzero at onset of a gray level step (or) ramp

(c) must be non zero along ramp.

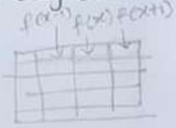
→ Similarly, definition for second derivative

→ (a) must be zero in flat area.

(b) must be non zero at the onset & end of gray-level step (or) ramp.

(c) must be zero along ramps of constant slope.

→ The shortest distance over which change can occur is b/w adjecent pixels.



The basic definition

1st order derivative

$$\frac{\delta f}{\delta x} = f(x+1) - f(x)$$

2nd order derivative

$$\frac{\delta^2 f}{\delta x^2} = f(x+1) + f(x-1) - 2f(x)$$

EC Academy

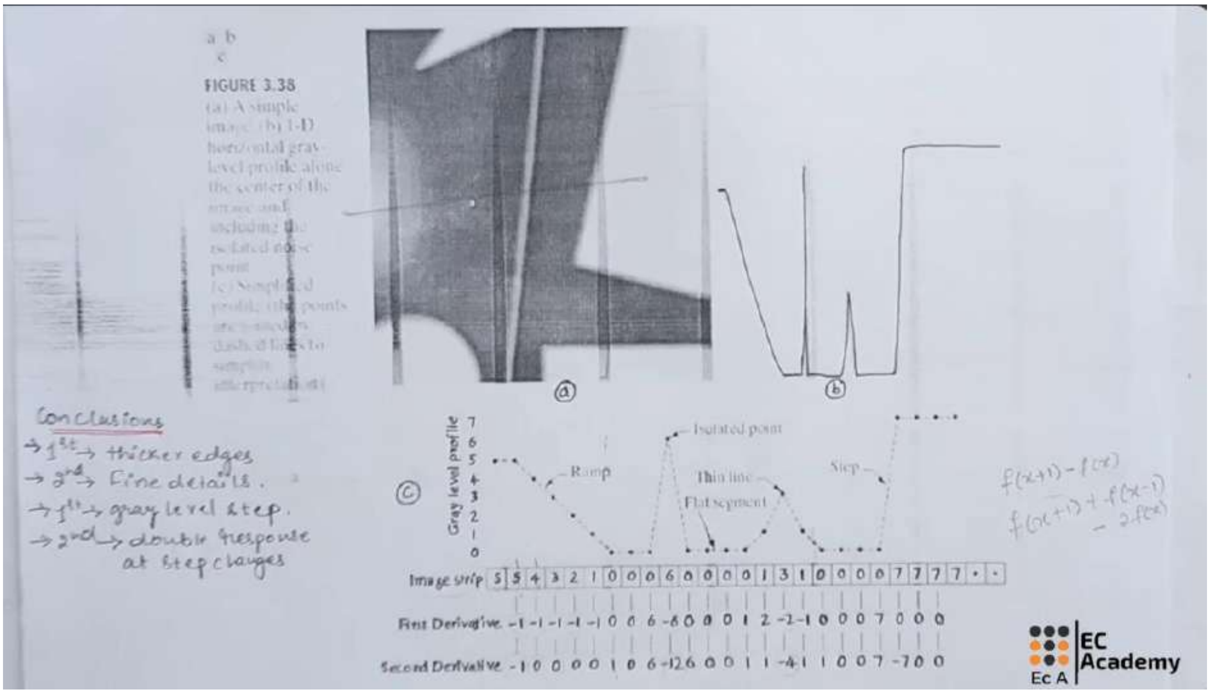
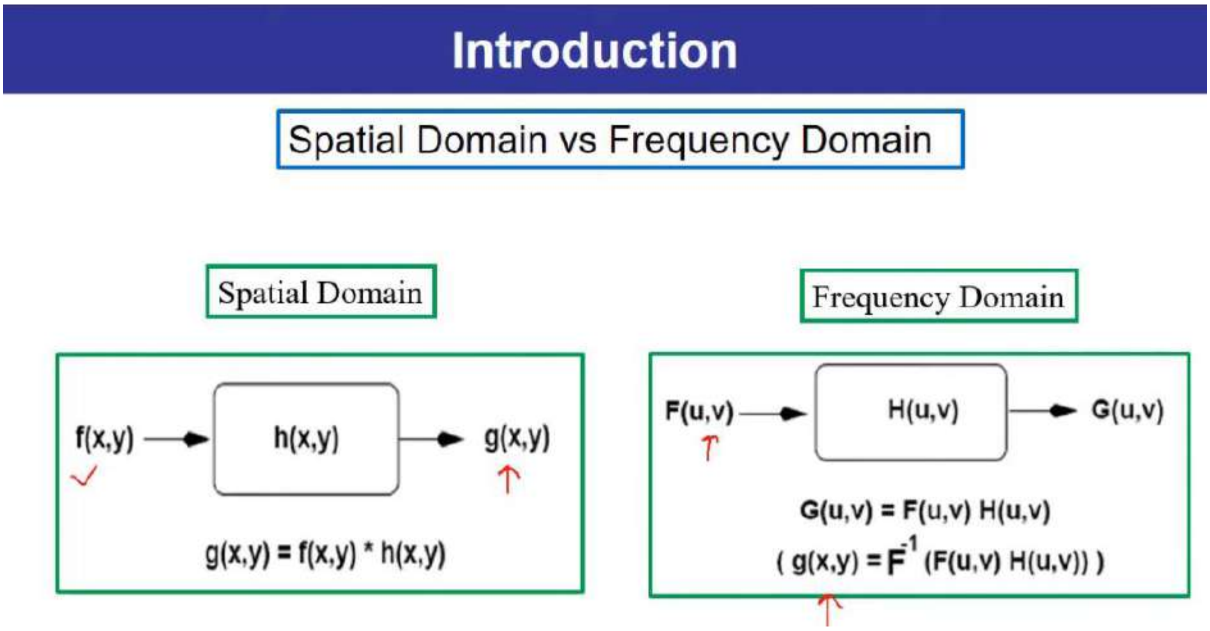


Image Smoothing in Frequency Domain Filtering



Filtering process can also be perform in frequency domain.

- Frequency Domain Filtering process is based on image transforms.
- Frequency Domain Filters are used for smoothing and sharpening of image by removal of high or low frequency components.
- Sometimes it is possible of removal of very high and very low frequency.
- Frequency domain filters are different from spatial domain filters as it basically focuses on the frequency of the images.

Frequency Domain Filtering is preferred to Spatial Domain because of less computation involved.

Image Compression Model

Image compression is very important task in image processing
Images and Videos require lots of space and large transmission time

Data compression is the process of encoding data so that it takes less storage space or less transmission time than it would if it were not compressed

Data Compression:

It is the Mathematical process of transforming data to a smaller representation from the original

Original Data N1 Bits → Compressed Data N2 Bits

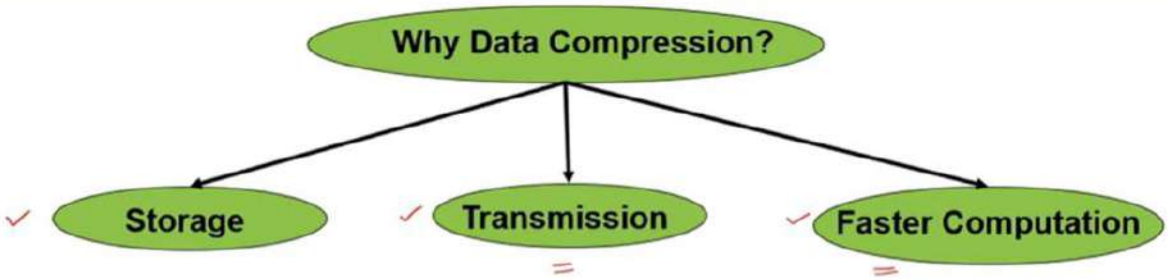
- If N1 = = N2 There is no compression
- If N2 << N1 There is significant Compression
- If N1 << N2 Data Explosion

Data and Information are two different things:

- ❑ Data is raw facts which are encountered in image processing
- ❑ Information is an interpretation of the data in a meaningful way
- ❑ Data is the means by which information is conveyed

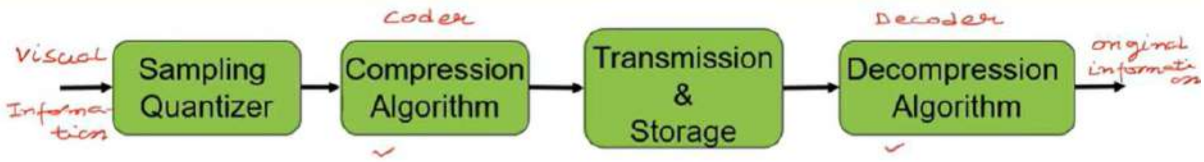
Types of Data

Text Data	Binary Data	Image Data	Graphics Data	Sound Data	Video Data
• Data present in flat files • It can be read and understood by human beings	• Machine can interpret it • Ex: metadata present in database files	• This is pixel data • It contains intensity and color information of image	• This data is in vector form	• This is audio information	• This represents video information



Applications of Data Compression:

- ❑ Personal communication like Facsimile, Voice mail and telephony
- ❑ Computer networks – Internet
- ❑ Multimedia applications
- ❑ Image and signal processing
- ❑ Digital and Satellite TV
- ❑ Video conferencing and Digital Library



Compression Scheme

Two main components of image compressor model are: Encoder and Decoder

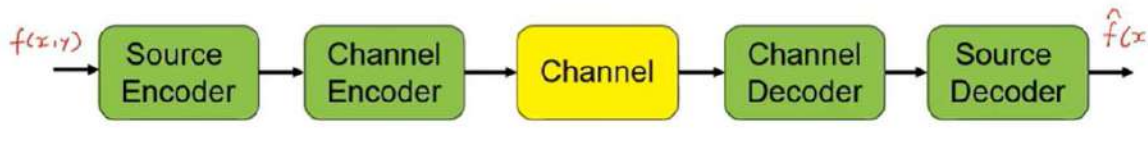


Image Compression Model

Stages of encoding:

$f(x,y) \rightarrow$ Mapper $\rightarrow$ Quantizer $\rightarrow$ Symbol Encoder $\rightarrow \hat{f}(x,y) \rightarrow$ channel

Mapper:
- Reduces Interpixel Redundancy
- Reversible operation

Quantizer:
- Phycovisual Redundancy
- Not Reversible operation

Symbol Encoder:
- Create a fixed or variable length code
- Reversible operation

Source Decoder:

$\rightarrow$ Source Decoder $\rightarrow$ Inverse Mapper $\rightarrow$

Compression Measures

Data Compression:

It is the Mathematical process of transforming data to a smaller representation from the original

Ex:- JPEG — USA

JPEG → Joint photographic Experts group

✓ → Logical compression

✓ → Physical compression

✓ Compression Ratio $(CR) = \frac{N_1}{N_2}$

Original Data N1 Bits $\xrightarrow{\text{Compression Algorithms}}$ Compressed Data N2 Bits

Relative Redundancy $(R_D) = 1 - \frac{1}{CR}$

Data Compression algorithms can be visualized as the mapping of as set of message symbols to codes using some logical rules of conversation

✓ Pixels or block of Pixels → set of symbols

✓ Code → sequence of symbols or numbers

✓ Code word → string of codes

Original Data N1 Bits → Compressed Data N2 Bits

1. ❑ If N1 = N2 There is no compression
2. ❑ If N2 << N1 There is significant Compression
3. ❑ If N1 << N2 Data Explosion

① $N_1 = N_2$

$CR = \frac{N_1}{N_2} = 1$ (compression ratio)

Relative Redundancy $(R_D) = 1 - \frac{1}{CR} = 0$

i/p message is same as o/p message

② $CR = \frac{N_1}{N_2} \rightarrow \infty$

$R_D = 1$

Compression is done

③ $N_1 < N_2$ (Reverse compression) ×

$CR = \frac{N_1}{N_2} \rightarrow 0$, $R_D = 1 - \frac{1}{CR} \rightarrow \infty$

Data in Transformed Set > Original Data

Some of the Metrics used to quantify compression measures are:

Compression Ratio (CR)

Compression Ratio = Uncompressed file size / Compressed file size $= \frac{N_1}{N_2}$

Saving Percentage (SP)

Saving Percentage = 1 – (Compressed file size / Uncompressed file size) $= 1 - \frac{N_2}{N_1}$

Bit Rate (BR) → indicates effeciency of compression Algorithm

Bit Rate = Size of Compressed File / Number of pixels in Image $= \frac{N_2}{N}$ (bits/pixel) — bps, kbps, Mbps

Example: original image is of size 256 x 256 pixels, gray-scale i.e. 8 bits/pixel, file is 65536 bytes (64K). After compression image file is of 6554 bytes

Compression Ratio = N1 / N2
= 65536/6554
= 10
= 10:1
= 10 to 1 compression
= 10 times compression

It means image has been compressed to 1/10 original size

Compression Algorithm & Types

Compression Algorithms and Its Types

The objective of **Compression Algorithm** is to reduce the source data to a compressed form and decompress it to get the original data

Any Compression Algorithms has two components:

① **Modeller**

- It is used to condition the image data for compression using the knowledge of data
- It is present in both sender and receiver
- It can be either static or dynamic

② **Coder**

- Sender side coder is known as encoder
- This codes the symbols independently or using the model
- Receiver side coder is known as decoder
- Decoder, decodes the message from the compressed data

Symmetric scheme → models at sender and receiver side are same

Static → do not change

Dynamic → models change depending upon the change of data

Compression Algorithms and Its Types

Compression Algorithms

Lossless Compression

- Reconstructed data is identical to the original data **(Entropy Coding)**
- Example techniques:
  - ✓ Huffman coding
  - ✓ Arithmetic coding
  - ✓ Shanon-Fano coding

Legal & medical domains

Lossy Compression

- Reconstructed data is an approximation to the original data **(Source Coding)**
- Example techniques:
  - ✓ Linear Prediction
  - ✓ Transform Coding

Difference between Lossless and Lossy Compression

Lossless Compression	Lossy Compression
Reversible Process	Non-reversible Process
No information is lost	Some information is lost
Compression Ratio is less	Compression Ratio is very high
It is used for that data which human can handle directly like Text data	It is used for diffused data which human can not understand or interpret
Compression is independent on psychovisual system	Compression is dependent on psychovisual system

Another way of classifying Image Compression Algorithm is:

- Entropy Coding
- Predictive Coding
- Transform Coding
- Layered Encoding

Entropy Coding

- The average information in an image is known as its entropy
- Coding is based on the entropy of the source and on the possibility of occurrence of the symbols
- An event that is less likely to occur is said to contain more information than an event that is more likely to occur

Set of symbols (alphabet) $S=\{s_1, s_2, \dots, s_N\}$,
N is number of symbols in the alphabet
Probability distribution of the symbols: $P=\{p_1, p_2, \dots, p_N\}$

According to Shannon, the entropy H of an nformation source S is defined as follows:

$$H = -\sum_{i=1}^{N} p_i \cdot \log_2(p_i)$$

Predictive Coding

- The idea is to remove the mutual dependency between the successive pixels and then perform the encoding
- Normally the samples would be very large but the difference would be small

Ex.

Pixels → 400 405 420 430

Difference → 5 15 10

ex. 300 4096 120 5081 10

××

Transform Coding

- Objective is to exploit the information packing capability of the transform
- Energy is packed into fewer components and only these components are encoded and transmitted
- Idea is to remove the redundant high frequency components to create compression
- Removal of these frequency components leads to loss of information
- This loss of information, if tolerable, can be used for imaging and video applications

Layered Coding

- It is very useful in case of layered images
- Data structures like pyramids are useful to represent an image in multiresolution form
- Sometimes these images are segmented as foreground and background, and based on the application requirement encoding is done
- It is also in the form of selected frequency coefficients or selected bits of pixels in an image

Introduction to Redundancy | Types of Redundancy

Redundancy means repetitive data

This may be data that share some common characteristics or overlapped information

Redundancy may be implicitly and explicitly

Ex: AAAABB

$\{(A,4),(B,2)\}$

Ex. Image $\begin{bmatrix} 10 & 10 & 20 \\ 20 & 20 & 20 \\ 20 & 20 & 10 \end{bmatrix}$

10 → 3

20 → 6

Explicit

Implicit

$I = \begin{bmatrix} 00 & 11 \\ 00 & 10 \end{bmatrix}$

$I = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$ and $\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$

Types of Redundancy

Redundancy

- **Coding Redundancy**
- **Inter-pixel Redundancy**
- **Psycho-visual Redundancy**
- **Chromatic Redundancy**

Coding Redundancy

- Coding redundancy is caused due to poor selection of coding technique
- Coding techniques assigns a unique code for all symbols of message
- Wrong choice of coding technique creates unnecessary additional bits. These extra bits are called redundancy

Amount of uncertainty

$\downarrow$

Self information of event

$$I = \log_2\left(\frac{1}{P}\right) = -\log_2(P) \text{ bits}$$

$$I \propto \left(\frac{1}{P}\right)$$

Coding Redundancy = Average bits used to Code- Entropy

$$\boxed{L_{avg.} = \sum_{k=0}^{n} l(r_k) \cdot P(r_k)}$$

$$\boxed{H = -\sum_{i=1}^{n} P_i \log_2(P_i)}$$

$\Rightarrow$ $P(r_k) \rightarrow$ Probability of pixels

$r_k \rightarrow$ grey level

$l_k \rightarrow$ length of the code

$\Rightarrow$ Total number of bits required to code an image of size $M \times N$

$$= M \times N \times L_{avg}$$

Inter-pixel Redundancy

- This type of redundancy is related with the inter-pixel correlations within an image.
- Much of the visual contribution of a single pixel is redundant and can be guessed from the values of its neighbors.

- **Example: consider an image with a constant background**
- The visual nature of the image background is given by many pixels that are not actually necessary
- This is known as **Spatial Redundancy** or **Geometrical Redundancy**

Spatial Redundancy may be present in:

- single frame (intra-frame)
- or among multiple frames (inter-frame or temporal redundancy)

Inter-pixel dependency is solved by algorithms like: Predictive Coding, Bit Plane Algorithm, Run Length Coding and Dictionary based Algorithms

Psycho-visual Redundancy

- The eye and the brain do not respond to all visual information with same sensitivity.
- Some information is neglected during the processing by the brain. Elimination of this information does not affect the interpretation of the image by the brain.
- Edges and textural regions are interpreted as important features and the brain groups and correlates such grouping to produce its perception of an object.

- **Psycho visual redundancy** is distinctly vision related, and its elimination does result in loss of information.
- **Quantization** is an example. When 256 levels are reduced by grouping to 16 levels, objects are still recognizable. The compression is 2:1, but an objectionable graininess and contouring effect results.

Run-length Coding (RLC)

Run-length coding (RLC) exploits the repetitive nature of the image

- RLC tries to identify the **length of the pixel values** and encodes the image in the form of a run
- Each row of the image is written as a sequence
- Then **length** is represented as a **run of black and white pixels**. This is known as Run-length coding
- It is very effective way compressing an image
- If required, **further compression** can be done using **variable length coding** to code the run lengths themselves

RLC is a CCITT (Consultative Committee of the International Telegraph and Telephone) standard that is used to encode binary and grey level images

Do from ppt

Huffman Coding

Huffman Coding

- **It is a type of variable length coding**
- Here coding redundancy can be eliminated by choosing a better way of assigning the codes

The Huffman Coding algorithm is given as:

1) List the symbols and sort probabilities per symbol
2) Combine the lowest two probabilities of symbols and label the new code with it
3) Newly created item is given priority and placed at the highest position in the sorted list
4) Repeat *Step2* until only one node remain
5) Assign code 0 to higher up symbol and 1 to the lower down symbol
6) Now trace the code symbols going backwards

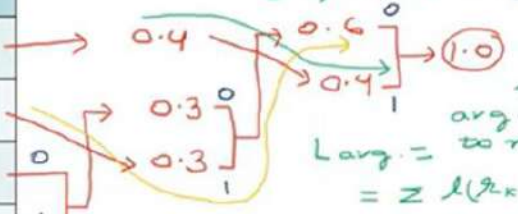
Huffman Coding

Example: Calculate the Huffman Codes for the set of symbols as shown in table.

Symbols	A	B	C	D
Probability	0.4	0.3	0.2	0.1

Solution:

Symbol	Probability
A	0.4
B	0.3
C	0.2
D	0.1



$l(x_K)$

Codewords → length of code word

A → 1 → 1

B → 01 → 2

C → 000 → 3

D → 001 → 3

no. avg. of bits used to represent message

$L_{avg.} = \sum l(x_K) P(x_K)$

$= 1 \times 0.4 + 2 \times 0.3 + 3 \times 0.2 + 3 \times 0.1$

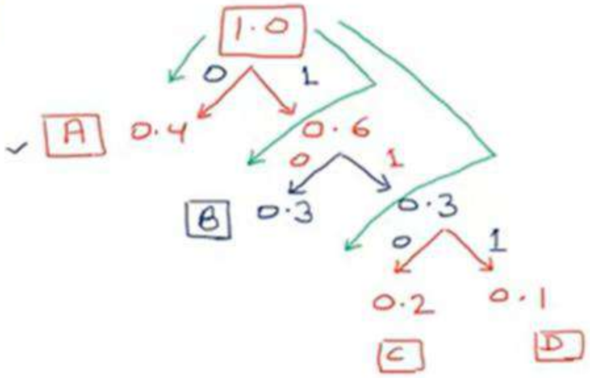
$= 1.9$ bits/symbols

Huffman Code Tree Approach

Example: Calculate the Huffman Coding for the set of symbols as shown in table.

Symbols	A	B	C	D
Probability	0.4	0.3	0.2	0.1

Solution:



code words

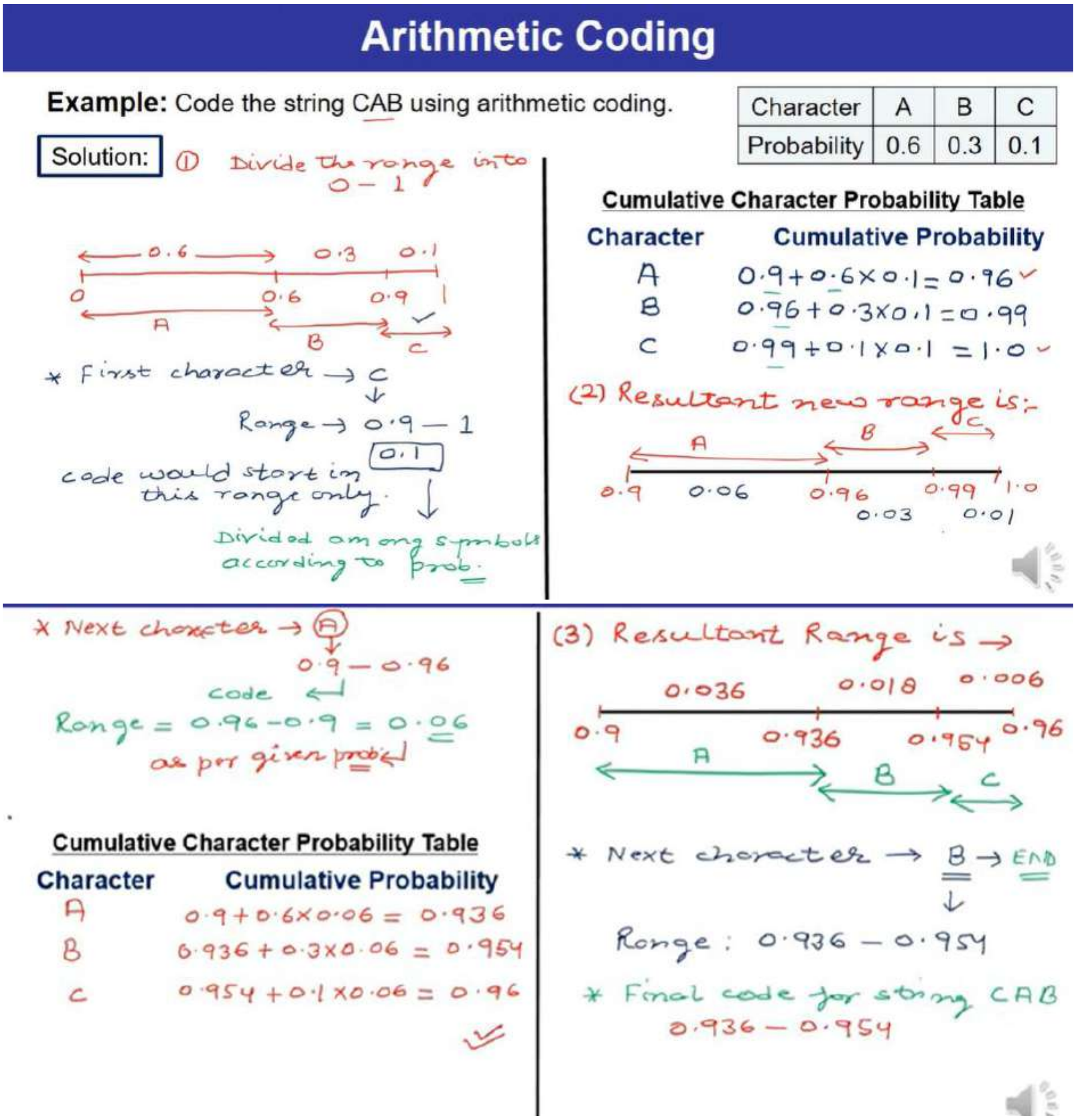
A → 0

B → 10

C → 110

D → 111

Arithmetic Coding



Dictionary based Coding | Lossless Compression Algorithm | LZW

Dictionary-based Coding

- **Dictionary-based coding** techniques replace input strings with a code to an entry in a **dictionary**
- The most well known **dictionary-based** techniques is Lempel-Ziv Algorithms (Ziv and Lempel, 1977, 1978)

- The idea behind Lempel-Ziv-Welch (LZW) coding is to **use a dictionary** to store the string patterns that have already been encountered
- Indices are used to encode the repeated patterns
- Encoder reads the input string and identifies the **recurrent words**, and outputs their indices from the dictionary
- **If a new word in encountered, the word is sent as output in the uncompressed form and is entered into dictionary as a new entry**

Dictionary-based Coding

Advantages

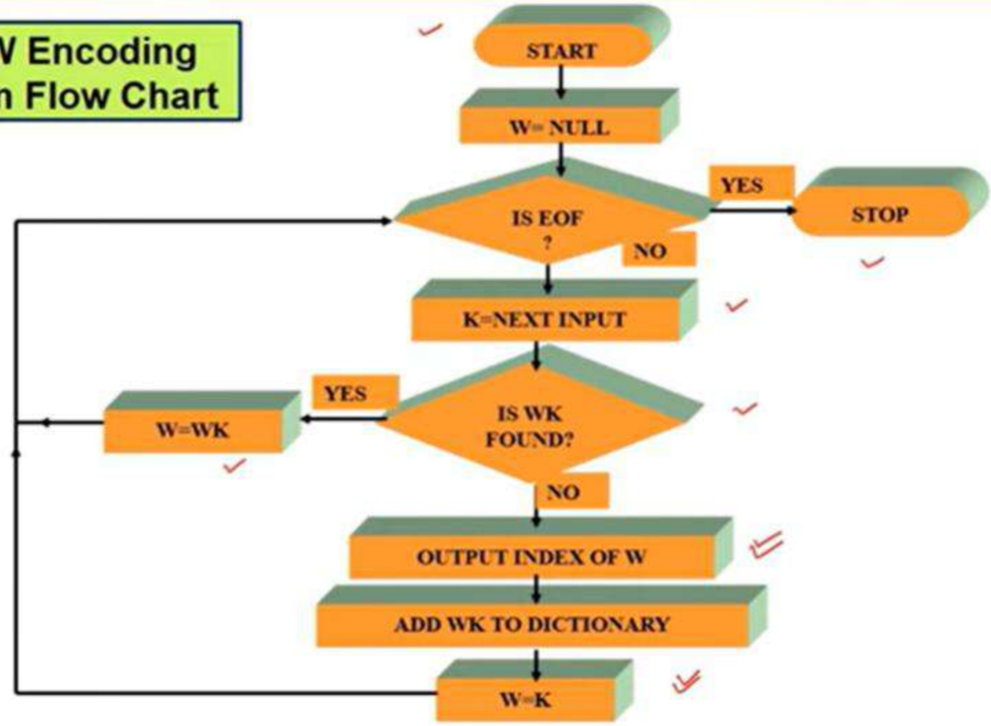
- These methods are **faster**
- They are **adaptive** in nature
- These methods are not based on statistics. Hence there is no dependency of the quality of the model on the distribution of data

Encoding

- Objective is to identify the longest pattern for each collected segment of the input string
- Then it is checked in the dictionary
- If there is no match, the segment becomes a new entry in the dictionary

Dictionary-based Coding

The LZW Encoding Algorithm Flow Chart

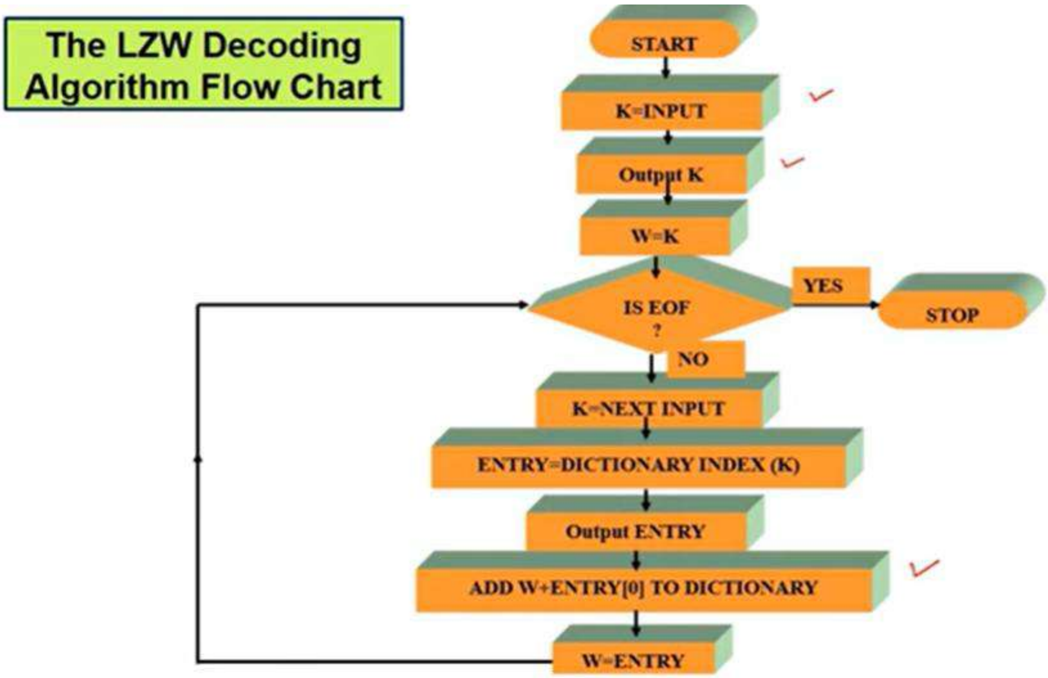


Example: Consider the string ADBB. Explain the working of Dictionary-based technique.

1) Word = Null
2) Read the string
   K ← A
   word + K = A → already in Dictionary
   word = word + A = Null + A = A
3) Read the string again : DBB
   Hence K = D
   word + K = AD → not in dictionary
   Then send output = 1
   New entry = A + D = AD (add it to dictionary)
   New word = D
4) Read the string again : BB
   Next character → B
   word + K = DB
   Send output → 3
   New entry = DB

	Initial Dictionary	Dictionary with new entry	Dictionary with new entry	Dictionary with new entry
A	1	1	1	1
B	2	2	2	2
D	3	3	3	3
AD	–	4	4	4
DB	–	–	5	5
BB	–	–	–	6

5) Read the string : B
   K = B
   word + K = BB , output → 2
6) Read = Null



JPEG encoding algorithm example

Example:- How Jpeg Compression works:-

① Divide an image into blocks, having dimensions of 8x8

Let us consider 8x8 image contains values

$$\begin{bmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 55 & 90 & 109 & 85 & 69 & 72 \\ 62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\ 67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\ 79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\ 87 & 79 & 69 & 68 & 65 & 76 & 78 & 94 \end{bmatrix}$$

⇒ The range of the pixels intensities now are from 0 to 255.
We will change the range from -128 to 127.

Subtracting 128 from each pixel value yields pixel value from -128 to 127. After ~~transfo~~ subtracting 128 from each of the pixel value, we got the result as,

$$\begin{bmatrix} -76 & -73 & -67 & -62 & -58 & -67 & -64 & -55 \\ -65 & -69 & -73 & -38 & -19 & -43 & -59 & -56 \\ -66 & -69 & -60 & -15 & 16 & -24 & -62 & -55 \\ -65 & -70 & -57 & -6 & 26 & -22 & -58 & -59 \\ -61 & -67 & -60 & -24 & -2 & -40 & -60 & -58 \\ -49 & -63 & -68 & -58 & -51 & -60 & -70 & -53 \\ -43 & -57 & -64 & -69 & -73 & -67 & -63 & -45 \\ -41 & -49 & -59 & -60 & -63 & -52 & -50 & -34 \end{bmatrix}$$

Now we will compute using the formula,

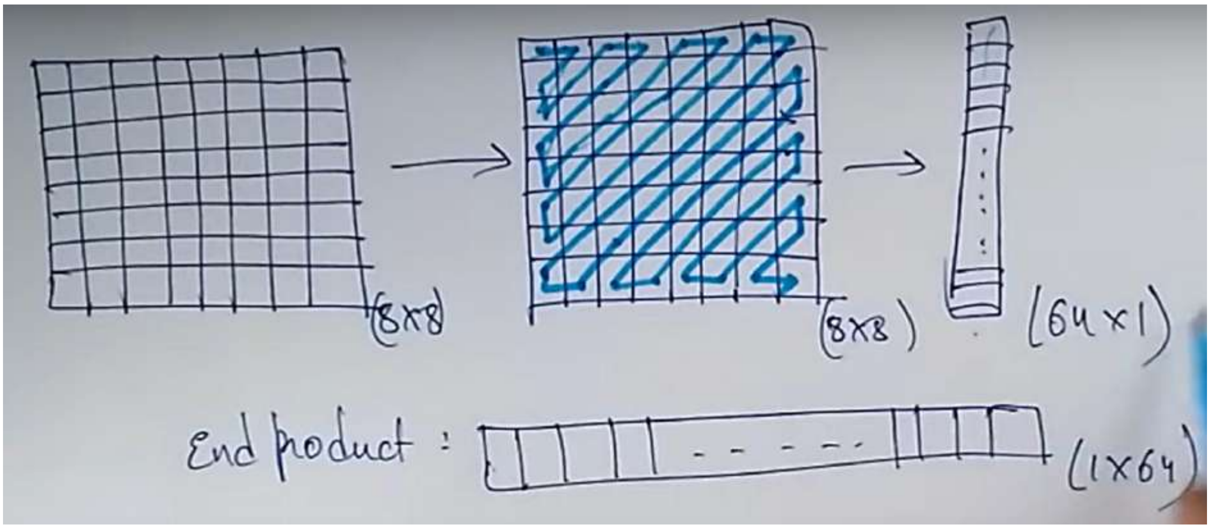
$$G_{u,v} = \alpha(u)\alpha(v) \sum_{x=0}^{7} \sum_{y=0}^{7} g_{x,y} \cos\left[\frac{\pi}{8}\left(x+\frac{1}{2}\right)u\right] \cos\left[\frac{\pi}{8}\left(y+\frac{1}{2}\right)v\right]$$

Apply the formula,

$$B_{j,k} = \text{round}\left(\frac{A_{j,k}}{Q_{j,k}}\right)$$

$$\Rightarrow B_{j,k} = \begin{bmatrix} -26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\ 0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -4 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Now, we will perform the real trick which is done in JPEG Compression i.e ZIG-ZAG movement.



Introduction to Image Segmentation | Segmentation Classification

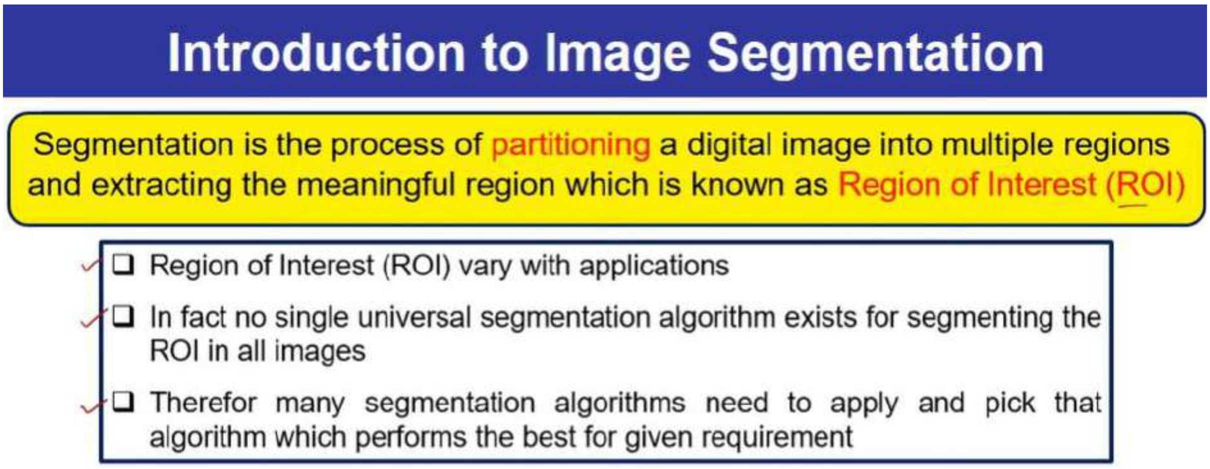


Image Segmentation Algorithms are based on:

Similarity Principle
(Region Approach)

Objective is to group pixels based on common property to extract a coherent region

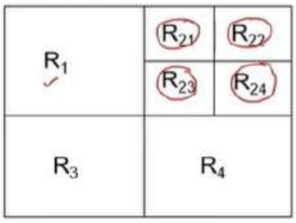
Discontinuity Principle
(Boundary Approach)

Objective is to extract regions that differ in properties like intensity, color, texture etc.

Definition of Image Segmentation

An image can be portioned into many regions R_1, R_2, R_3…R_n

$R = 7$



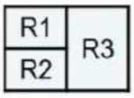
Example

5×5

R_1, R_2, R_3

10	10	20	20	20
10	10	20	20	20
10	10	20	20	20
30	30	20	20	20
30	30	20	20	20

10	10	20	20	20
10	10	20	20	20
10	10	20	20	20
30	30	20	20	20
30	30	20	20	20



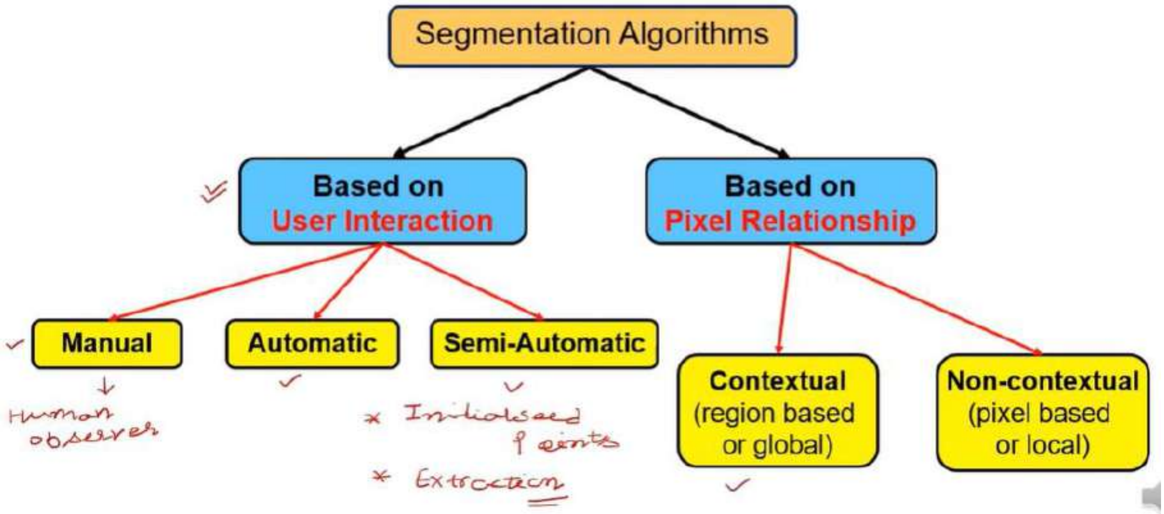
Characteristics of Segmentation Process

Let R represent the entire image region and Segmentation is partitioning R into n subgroups R_i

- $\bigcup_{i=1}^{n} Ri = R$ $i = 1, 2, \ldots n$ R
- Ri should be connected region :i=1,2,3,........,n
- $Ri \cap R_1 = \emptyset \ (for\ all\ i\ and\ j): i \neq j$
- $P(R_i) = TRUE\ for\ i = 1,2,3, \ldots \ldots n$
- $P(R_i \cup R_j) = FALSE\ for\ i \neq j$

Here P(R_i) is a predicate that indicates some property over the region

Classification of Image Segmentation Algorithms



Detection of Discontinuities | Point, Line & Edge Detection

Types of Grey Level discontinuities are:

- **Point**
  - * Isolated Point
- **Line**
  - * Horizontal
  - * Vertical
  - * Slanted
- **Edge**
  - * Object outline

Point Detection

An isolated point is a point whose grey level is significantly different from its background in a homogeneous area

3x3

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Mask

3x3

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Image

Response of the mask:

$$R = \sum_{i=1}^{9} w_i z_i$$

If,
✓ **$|R| \geq T$**, a point is detected
where,
T is a non negative integer

−1	−1	−1
−1	8	−1
−1	−1	−1

Sample Mask for Point Detection

Line Detection

In line detection, **four types of masks** are used to get the responses i,e, R_1, R_2, R_3 and R_4 for the directions vertical, horizontal, $+45^0$ and -45^0 respectively

−1	−1	−1
2	2	2
−1	−1	−1

Horizontal

−1	2	−1
−1	2	−1
−1	2	−1

Vertical

−1	−1	2
−1	2	−1
2	−1	−1

+45⁰

2	−1	−1
−1	2	−1
−1	−1	2

-45⁰

R_1
R_2
R_3
R_4

Response of the mask:

$$R_K = \sum_{k=1}^{4} \omega_K z_K$$

If, at a certain point in the image, **$|R_i|>|R_j|$** for all **$j \neq i$**, that **point** is said to be more likely associated with a **line** in the direction of mask i

Edge Detection

- An edge is a set of connected pixels that lies on the boundary between two regions which differ in grey value. Pixels on edge is known as edge points
- **Edges provide an outline of the object**

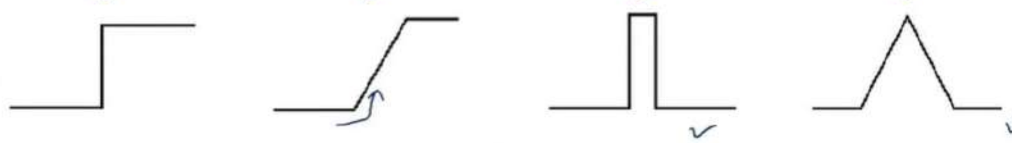
In physical plane, **edge corresponds** to the **discontinuities in** depth, Surface orientation, change in material properties, light variations etc.

- It locates sharp changes in the intensity function
- Edges are pixels where brightness changes abruptly
- An edge can be extracted by computing the derivative of the image function
  - ✓ Magnitude of the derivative, indicates the strength or contrast of edge
  - ✓ Direction of the derivative vector, indicates the edge orientation

Edge Detection

Some of the commonly encountered edges in image processing are:

Step Edge | **Ramp Edge** | **Spike Edge** | **Roof Edge**



- **Step Edge**: An abrupt change in intensity
- **Ramp Edge**: A slow and gradual change in intensity
- **Spike Edge**: Quick change or immediately returns to original intensity level
- **Roof Edge**: It is not instantaneous over short distance

Stages in Edge Detection

A B

Start ⇨ Input Image ⇨ Filtering ⇨ Differn -tiation ⇨ Locali- zation ⇨ Display ⇨ End

- **Filtering**: It involves smoothing. This stage may be performed explicitly or implicitly
- **Differentiation**: It distinguishes the edge pixels from other pixels
- **Localization**: It determine the exact location of edge

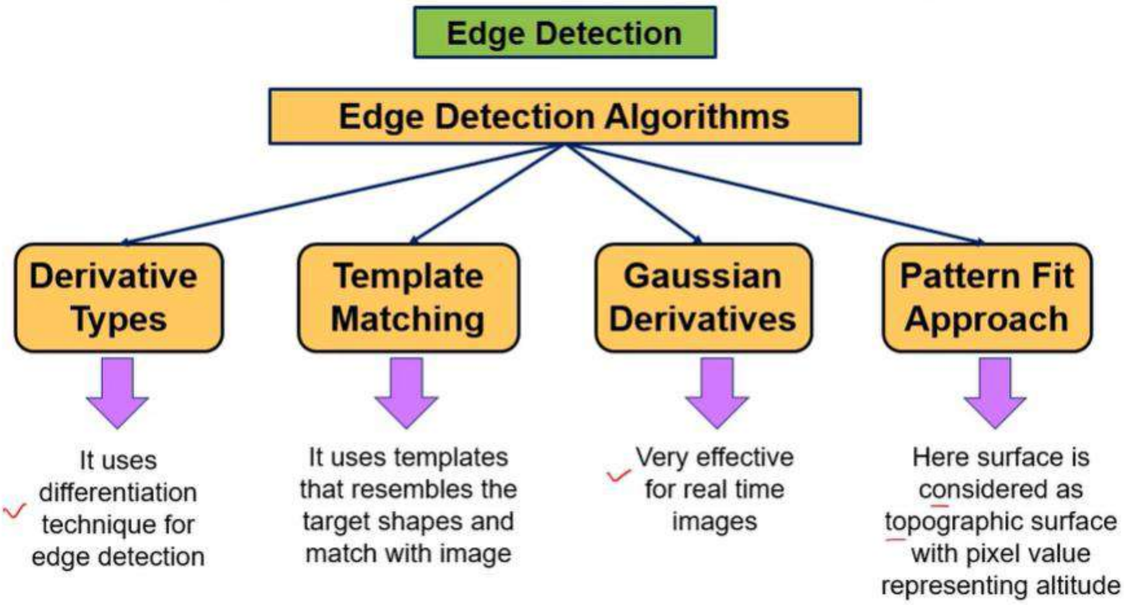
f(x,y)

$$\text{Vector: } \nabla \mathbf{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \partial f/\partial x \\ \partial f/\partial y \end{bmatrix}$$

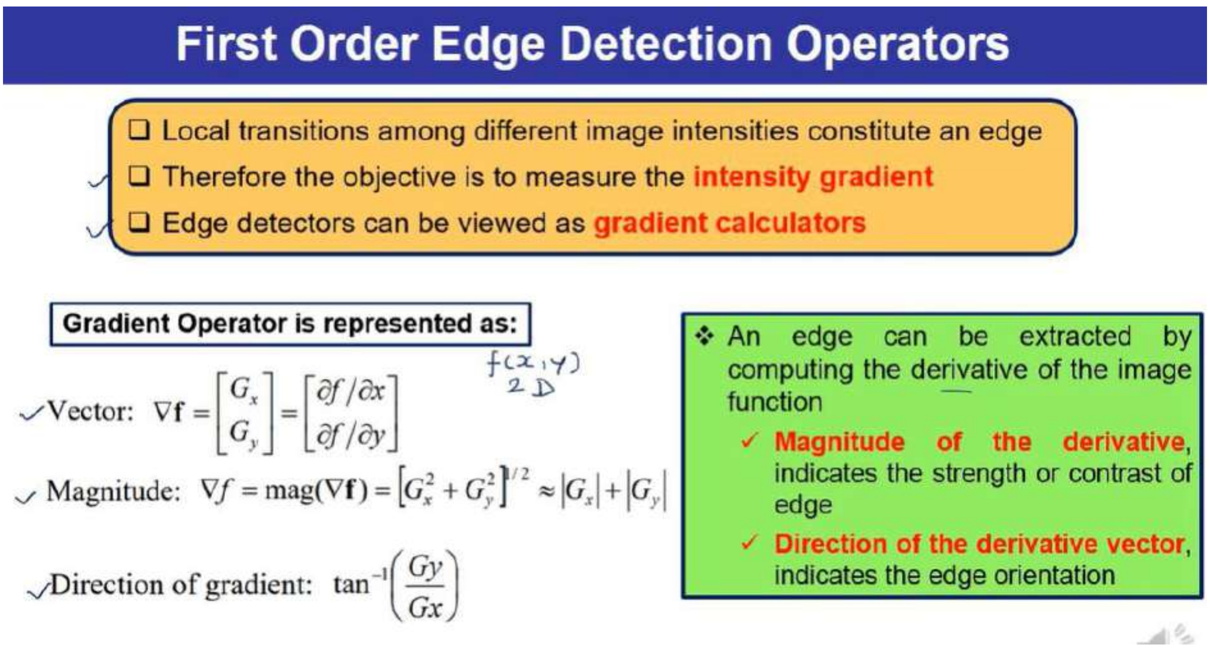
$$\text{Magnitude: } \nabla f = \text{mag}(\nabla \mathbf{f}) = \left[G_x^2 + G_y^2\right]^{1/2} \approx |G_x| + |G_y|$$

$$\text{Direction of gradient: } \tan^{-1}\left(\frac{Gy}{Gx}\right)$$

θ



First Order Edge Detection Operators in Image Segmentation



Backward Difference:

$2 \quad 1$
$x-\Delta x \quad x$

$= [f(x) - f(x-\Delta x)] / \Delta x$

Forward Difference:

$1 \rightarrow 2$
$x \quad x+\Delta x$

$= [f(x+ \Delta x) - f(x)] / \Delta x$

Central Difference:

$2 \quad 0 \quad 1$
$x-\Delta x \quad x \quad x+\Delta x$

$= [f(x+ \Delta x) - f(x-\Delta x)] / 2\Delta x$

These differences can be obtained by applying the following masks, assuming $\Delta x=1$:

Backward Difference = f(x) – f(x-1) = [1 -1]

Forward Difference: = f(x+1) – f(x) = [-1 +1]

Robert Operator

- Robert Kernels are derivatives with respect to the diagonal elements
- They are known as **Cross-Gradient Operators**
- They are based on **Cross Diagonal differences**

Generic gradient based algorithm can be:

- Read the image and smooth it
- Convolve the image f with g_x $\quad \hat{f}(x) = f * g_x$
- Convolve the image f with g_y
- Compute the edge magnitude and edge orientation
- Compare the edge magnitude with a threshold value
  - If edge magnitude is higher, assign it as a possible edge point

This algorithm can be applied to other masks also

Let $f(x,y)$ & $f(x+1,y)$ be neighbouring pixels, then

$$\frac{\partial f}{\partial x} = f(x+1,y) - f(x,y)$$

Robert masks may be of:

G_x

1	0
0	−1

G_y

0	1
−1	0

Prewitt Operator

① $x-\Delta x$ 2 x ③ $x+\Delta x$

The Prewitt Method takes the central difference of the neighbouring pixels; This difference can be represented mathematically as:

$$\frac{\partial f}{\partial x} = [f(x+1) - f(x-1)]/2$$

$\Rightarrow$ For 2D $\rightarrow (x,y)$

$$[f(x+1,y) - f(x-1,y)]/2$$

central difference is obtained by mask $\rightarrow [-1 \quad 0 \quad +1]$

G_x

−1	−1	−1
0	0	0
1	1	1

G_y

−1	0	1
−1	0	1
−1	0	1

Additional mask can be used to detect edges in diagonal as:

0	1	1
−1	0	1
−1	−1	0

−1	−1	0
−1	0	1
0	1	1

3 × 3 digital approximation of Prewitt Operator is given as:

$$G_x = \frac{\partial f}{\partial x} = (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)$$

$$G_y = \frac{\partial f}{\partial y} = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)$$

$$\nabla f \cong |G_x| + |G_y|$$

Sobel Operator

- ❑ It provides both a differentiating and a smoothing effect
- ❑ Sobel Operator relies on the central differences
- ❑ It can be viewed as an approximation of first Gaussian Derivative
- ❑ Here convolution is both commutative an associative

$$\frac{\partial}{\partial x}(f * G) = f * \frac{\partial}{\partial x}(G)$$

G_x

−1	−2	−1
0	0	0
1	2	1

G_y (3x3)

−1	0	1
−2	0	2
−1	0	1

3 × 3 digital approximation of Sobel Operator is given as:

$$G_x = \frac{\partial f}{\partial x} = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$$

$$G_y = \frac{\partial f}{\partial y} = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$$

$$\nabla f \cong |G_x| + |G_y|$$

Additional mask can be used to detect edges in diagonal as:

0	1	2
−1	0	1
−2	−1	0

−2	−1	0
−1	0	1
0	1	2

Template Matching Masks

- ❑ Gradient masks are isotropic and insensitive to direction
- ❑ Sometimes it is necessary to design direction sensitive filters, such type of filters are known as Template Matching Filters

Kirsch Masks

- ✓ Kirsch masks are known as compass masks because they are obtained by taking one mask and rotating it to the eight major directions
- ✓ Directions are: East, North East, North, North West, West, South West, South and South East

-3 -3 5 5 5

-3 5 5 -3

θ

$$M_0 = \begin{bmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{bmatrix} \quad M_1 = \begin{bmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{bmatrix} \quad M_2 = \begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} \quad M_3 = \begin{bmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix}$$

$$M_4 = \begin{bmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{bmatrix} \quad M_5 = \begin{bmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{bmatrix} \quad M_6 = \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{bmatrix} \quad M_7 = \begin{bmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{bmatrix}$$

Template Matching Masks

Robinson Compass Masks

The spatial masks for the Robinson edge operator for all the directions are as:

- ✓ Similar to Kirsch masks, the mask that produces the maximum value defines the **direction of the edge**

M_1 (→)

-1	0	1
-2	0	2
-1	0	1

M_2 (↗)

0	1	2
-1	0	1
-2	-1	0

M_3 (↑)

1	2	1
0	0	0
-1	-2	-1

M_4 (↖)

2	1	0
1	0	-1
0	-1	-2

(←)

1	0	-1
2	0	-2
1	1	-1

(↙)

0	-1	-2
-1	0	-1
2	1	0

(↓)

-1	-2	-1
0	0	0
1	2	1

(↘)

-2	-1	0
-1	0	1
0	1	2

Template Matching Masks

Frei-Chen Masks

- Any image can be considered as the weighted sum of the nine Frei-Chen masks
- Weights are obtained by a process known as projecting process by overlaying a 3×3 image onto each mask and by summing the multiplication of coincident terms
- First four masks represent the edge space, next four represent the line subspace and last one represent the average subspace

$$W_1=\frac{1}{2\sqrt{2}}\begin{bmatrix}1 & \sqrt{2} & 1\\ 0 & 0 & 0\\ -1 & -\sqrt{2} & -1\end{bmatrix} \quad W_2=\frac{1}{2\sqrt{2}}\begin{bmatrix}1 & 0 & -1\\ \sqrt{2} & 0 & -\sqrt{2}\\ 1 & 0 & -1\end{bmatrix}$$

$$W_3=\frac{1}{2\sqrt{2}}\begin{bmatrix}0 & -1 & \sqrt{2}\\ 1 & 0 & -1\\ -\sqrt{2} & 1 & 0\end{bmatrix} \quad W_4=\frac{1}{2\sqrt{2}}\begin{bmatrix}\sqrt{2} & -1 & 0\\ -1 & 0 & 1\\ 0 & 1 & -\sqrt{2}\end{bmatrix}$$

$$W_5=\frac{1}{2}\begin{bmatrix}-1 & 0 & 1\\ 0 & 0 & 0\\ 1 & 0 & -1\end{bmatrix} \quad W_6=\frac{1}{2}\begin{bmatrix}0 & 1 & 0\\ -1 & 0 & -1\\ 0 & 1 & 0\end{bmatrix} \quad W_9=\frac{1}{3}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

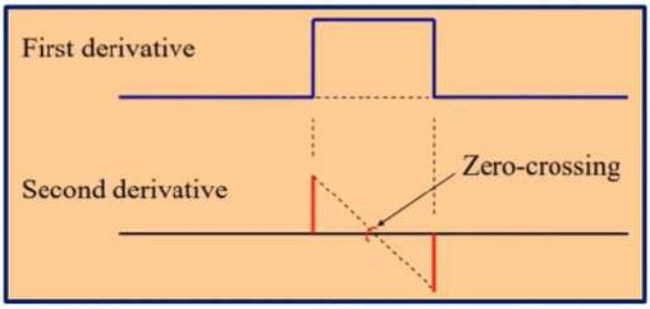
$$W_7=\frac{1}{6}\begin{bmatrix}1 & -2 & 1\\ -2 & 4 & -2\\ 1 & -2 & 1\end{bmatrix} \quad W_8=\frac{1}{6}\begin{bmatrix}-2 & 1 & -2\\ 1 & 4 & 1\\ -2 & 1 & -2\end{bmatrix}$$

Second Order Derivative Filters in Image Segmentation

Second Order Derivative Filters

$\underbrace{20 \quad 40}_{20} \underbrace{\quad 20}_{-20}$

- In first derivative, edges are considered to be present when edge magnitude is large compared to the threshold value
- In case of **second derivative**, edge is present at that location where the **second derivative is zero**
- It is like zero crossing, which can be observed as a sign change



Laplacian Operator

- Laplacian Algorithm is one of the **zero-crossing algorithm**
- Laplacian masks are **very sensitive to noise** because there is no magnitude checking, even a small ripple looks like edge
- Therefore image must be filter first and then edge detection process is applied
- Advantage of Laplacian operator is that they are **rotationally invariant**

The Laplacian of a 2D function f(x,y) is a 2nd order derivative is defined as:

$$\Rightarrow \quad \nabla^2 f(x,y) = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2}$$

$$\frac{\partial^2 f(x,y)}{\partial x^2} = f(x+1,y) - 2f(x,y) + f(x-1,y)$$

$$\frac{\partial^2 f(x,y)}{\partial y^2} = f(x,y+1) - 2f(x,y) + f(x,y-1)$$

$$\nabla^2 f(x,y) = f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y)$$

x y

$(x-1,y)$ $f(x,y)$ $(x+1,y)$

$(x,y+1)$ y $(x,y-1)$

Laplacian Operator

Algorithm can be written as:

- Generate the mask
- Apply the mask
- Detect the zero crossing
  - Zero crossing is a situation where pixels in neighbourhood differ from each other pixel in sign

p & q $|\nabla^2 f(p)| \le |\nabla^2 f(q)|$

0	1	0
1	−4	1
0	1	0

1	1	1
1	−8	1
1	1	1

0	0	-1	0	0
0	-1	-2	-1	0
-1	-2	16	-2	-1
0	-1	-2	-1	0
0	0	-1	0	0

5×5

Different Laplacian Masks

0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

3×3 →

① $\nabla^2 f = 4z_5 - (z_2 + z_4 + z_6 + z_8)$

② for other mask,

$$\nabla^2 f = 8z_5 - (z_1 + z_2 + z_3 + z_4 + z_6 + z_7 + z_8 + z_9)$$

More Advanced Techniques for Edge Detection

- The basic edge detection method is based on simple filtering without taking note of image characteristics and other information
- **More advanced techniques make attempt to improve the simple detection by taking into account factors such as noise, scaling etc.**

Laplacian of Gaussian (Marr-Hildrith) Operator

Discussion of Marr and Hildreth was about:

- Intensity of changes is not independent of image scale
- Sudden intensity change will cause a zero crossing of the second derivative

Therefore, an edge detection operator should:

- Be capable of being tuned to any scale
- Be capable of computing the first and second derivatives

To minimize the noise susceptibility of the Laplacian Operator, Laplacian of Gaussian (LoG) Operator is often preferred

Laplacian of Gaussian (Marr-Hildrith) Operator

The LoG Algorithm can be written as:

- Generate the mask and apply LoG to the image
- Detect the Zero Crossing

* For 1D,

$$\nabla^2(f * g) = f * \nabla^2 g = f * LoG$$

Let the 2D, Gaussian f^n is given as →

$$G_\sigma(x,y) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{x^2+y^2}{2\sigma^2}\right] \quad \text{—①}$$

To suppress the noise, the image is convolved with the Gaussian smoothing function before using the Laplacian for edge detection

$$\nabla\left[G_\sigma(x,y) * f(x,y)\right] = \left[\nabla\, G_\sigma(x,y)\right] * f(x,y) = LoG * f(x,y)$$

The LoG function can be derived as:

$$\frac{\partial}{\partial x} G_\sigma(x,y) = \frac{\partial}{\partial x} e^{\left[-\frac{x^2+y^2}{2\sigma^2}\right]} = -\frac{x}{\sigma^2} e^{\left[-\frac{x^2+y^2}{2\sigma^2}\right]} \quad \text{—②}$$

Laplacian of Gaussian (Marr-Hildrith) Operator

Similarly,

$$\frac{\partial^2}{\partial x^2} G_\sigma(x,y) = \frac{x^2}{\sigma^4} e^{\left(-\frac{x^2+y^2}{2\sigma^2}\right)} - \frac{1}{\sigma^2} e^{\left(-\frac{x^2+y^2}{2\sigma^2}\right)} = \frac{x^2-\sigma^2}{\sigma^4} e^{\left(-\frac{x^2+y^2}{2\sigma^2}\right)}$$

Similarly, by ignoring the normalization constant $1/\sqrt{2\pi\sigma^2}$, we get

$$\frac{\partial^2}{\partial y^2} G_\sigma(x,y) = \frac{y^2}{\sigma^4} e^{\left(-\frac{x^2+y^2}{2\sigma^2}\right)} - \frac{1}{\sigma^2} e^{\left(\frac{x^2+y^2}{2\sigma^2}\right)} = \frac{y^2-\sigma^2}{\sigma^4} e^{\left(-\frac{x^2+y^2}{2\sigma^2}\right)}$$

The LoG kernel can be described as:

$$LoG \triangleq \frac{\partial^2}{\partial x^2} G_\sigma(x,y) + \frac{\partial^2}{\partial y^2} G_\sigma(x,y) = \frac{x^2+y^2-2\sigma^2}{\sigma^4} e^{\left(-\frac{x^2+y^2}{2\sigma^2}\right)}$$

σ ↑

Difference of Gaussian (DoG) Filter

The DoG Filter can be written as:

$$G_{\sigma_1}(x,y) = \frac{1}{\sqrt{2\pi\sigma_1^2}} \exp\left(-\frac{x^2+y^2}{2\sigma_1^2}\right) \text{ with gaussian width } \sigma_1$$

The width of the Gaussian is changed and a new kernel is:

$$G_{\sigma_2}(x,y) = \frac{1}{\sqrt{2\pi\sigma_2^2}} \exp\left(-\frac{x^2+y^2}{2\sigma_2^2}\right) \text{ with gaussian width } \sigma_2$$

The DoG is expressed as the difference between these two Gaussian kernels:

$$G_{\sigma_1}(x,y) - G_{\sigma_2}(x,y) = \left(G_{\sigma_1} - G_{\sigma_2}\right) * f(x,y)$$
$$= DoG * f(x,y)$$

Difference of Gaussian (DoG) Filter

The DoG as a kernel can be written as:

$$DoG = G_{\sigma_1} - G_{\sigma_2} = \frac{1}{\sqrt{2\pi}}\left[\frac{1}{\sigma_1} e^{\left(\frac{x^2+y^2}{2\sigma_1^2}\right)} - \frac{1}{\sigma_2} e^{\left(-\frac{x^2+y^2}{2\sigma_2^2}\right)}\right]$$

$\sigma_1 \quad \sigma_2 \qquad \frac{\sigma_1}{\sigma_2} \rightarrow$ in between 1 & 2 $\rightarrow$ good

The DoG Algorithm can be written as:

- Generate the mask and apply DoG to the image
- Detect the Zero Crossing and apply the threshold to suppress the weak zero-crossing
- Display and Exit

Canny Edge Detection Algorithm

- The Canny edge detector is an edge detection operator that uses a multistage algorithm to detect a wide range of edges in images
- This algorithm was developed by **John F. Canny** in **1986**

Canny approach is based on optimizing the trade off between two performance criteria and can be described as:

- **Good Edge Detection:** should be capable to detect only real edge points and discard all false edge points
- **Good Edge Localization:** should have the ability to produce edge points which are close to real edges
- **Only one response to each edge:** should not produce any false, double or spurious edges

Canny Edge Detection Algorithm

Step-1
- Convolve the image with Gaussian Filter
- Compute the gradient of resultant smooth image
- Store the edge magnitude and edge orientation in two separate arrays

Smoothing image and computing coefficients

$$S = G_\sigma * I \qquad G_\sigma = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

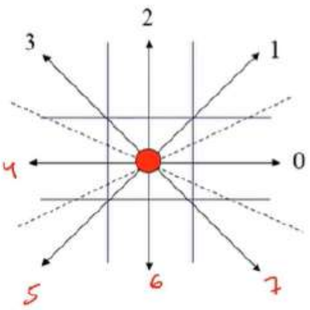
$$\nabla S = \left[\frac{\partial}{\partial x} S \quad \frac{\partial}{\partial y} S\right]^T = \left[S_x \quad S_y\right]^T$$

Computing gradient magnitude and direction

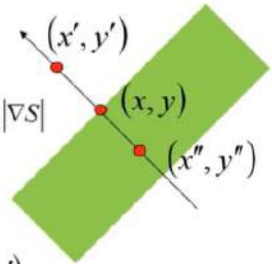
$$|\nabla S| = \sqrt{S_x^2 + S_y^2}$$

$$\theta = \tan^{-1} \frac{S_y}{S_x}$$

Step-2
- Thinning of edges by a process known as Non-maxima Suppression



(x',y') and (x'',y'') are the neighbors of (x,y) in $|\nabla S|$ along the direction normal to an edge

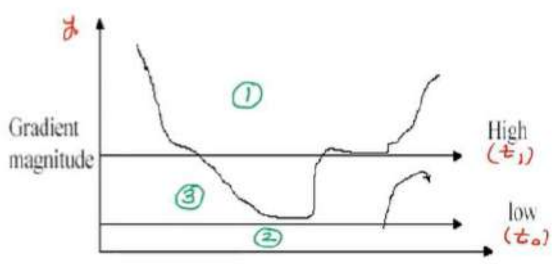


$$M(x,y) = \begin{cases} |\nabla S|(x,y) & \text{if } |\nabla S|(x,y) > |\nabla S|(x',y') \\ & \& \, |\nabla S|(x,y) > |\nabla S|(x'',y'') \\ 0 & \text{otherwise} \end{cases}$$

Canny Edge Detection Algorithm

Step-3
- Apply Hysteresis Thresholding
  (objective is that only a large amount of change in gradient magnitude)

- If the gradient at a pixel is above **'High'**, declare it an **'edge pixel'**
- If the gradient at a pixel is below **'Low'**, declare it a **'non-edge-pixel'**
- If the gradient at a pixel is between 'Low' and 'High' then declare it an 'edge pixel' if and only if it is connected to an 'edge pixel' directly or via pixels between 'Low' and ' High'



Hough Transform and Shape Detection

Hough Transform and Shape Detection

Hough transform is a feature extraction method for detecting simple shapes such as circles, lines etc. in an image

- Hough Transform takes the images created be edge detection operators but most of the time, edge map is disconnected
- Therefore Hough Transform is used to connect the disjoined edge points

Equation of Line:

$y = mx + c$

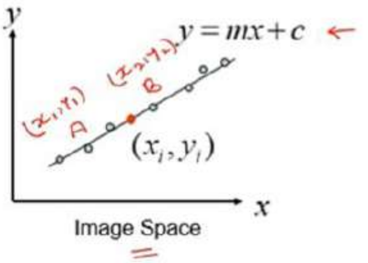
Where,
m = slope
c = intercept of the line

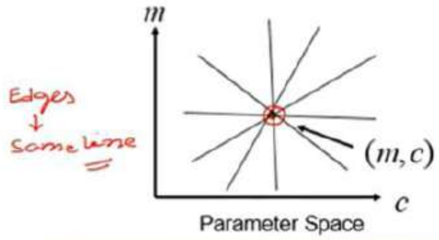
Therefore an edge point in the x-y plane is transformed to the c-m

Point (x_i, y_i)

$y_i = mx_i + c \quad or \quad \boxed{c = -x_i m + y_i}$

Problem is that, Infinite lines can be drawn connecting these points



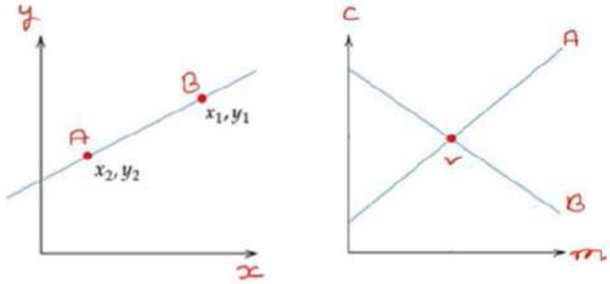


Parameter space also called Hough Space

If A and B are two points connected by a line in spatial domain

⇓

They will be intersecting lines in Hough Space



m', c'

$y = m'x + c'$

Algorithm

Load the image

↓

Determine the image edges using any edge detector

↓

Quantize the Parameter Space, P

↓

Repeat the process for all pixels of image
If the pixel is an edge pixel, then

$$c = (-x)m + y$$

$$P(c.m) = P(c.m) + 1$$

↓

Show the Hough Space

↓

Find the local maxima in parameter space

↓

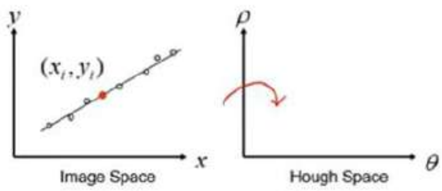
Draw the line using local maxima

- Limitation of **previous algorithm** is that it does not work for vertical lines because they have infinity slope
- Therefore this **line** must be converted into **polar coordinates**

Equation of Line in polar:

$$\rho = x\cos\theta + y\sin\theta$$

Where,
ϴ = angle between the line
ϱ = diameter



Load the image

↓

Determine the image edges using any edge detector

↓

Quantize the Parameter Space, P

↓

Repeat the process for all pixels of image
If the pixel is an edge pixel, then for all ϴ

- Calculate ϱ for pixel (x,y) and ϴ
- Increment the position (ϱ, ϴ) in accumulator array

↓

Show the Hough Space

↓

Find the local maxima in parameter space

↓

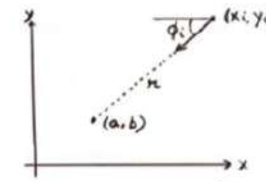
Draw the line using local maxima

Hough Transform for other shapes can also be found

For a Circle Detection, it can be given as

$$(x_i - a)^2 + (y_i - b)^2 = r^2$$

- Edge Location (x_i, y_i)
- Edge Direction ϕ_i (a, b, r)



- $a = x - r\cos\phi$
- $b = y - r\sin\phi$

Load the image
↓
Determine the image edges using any edge detector
↓
Quantize the Parameter Space, P
↓
Repeat the process for all pixels of image
If the pixel is an edge pixel, then calculate for all r

$a = x - r\cos\phi$
$b = y - r\sin\phi$
$P(a,b,r) = P(a,b,r) + 1$

↓
Show the Hough Space
↓
Find the local maxima in parameter space
↓
Draw the Circle using local maxima

- Using hough transform show that the points (1,1), (2,2) and (3,3) are collinear. Find the equation of line?

⇒ The equation of line is: $y = mx + c$

In order to perform hough transform we need to convert line from (x, y) plane to (m, c) plane

Equation of line in (m, c) plane is

$$c = -mx + y$$

For $(x, y) = (1, 1)$, $c = -m + 1$

if $c = 0$, $\begin{bmatrix} 0 = -m + 1 \\ m = 1 \end{bmatrix}$ $m = 1$

if $m = 0$, $c = 1$ $c = 1$

Thus $(m, c) = (1, 1)$

$$c = -mx + y$$

For $(x, y) = (2, 2)$

$c = -2m + 2$

if $c = 0$ $\left[\begin{array}{l} 0 = -2m + 2 \\ -2 = -2m \\ m = \frac{-2}{-2} \end{array}\right.$, $m = 1$

if $m = 0$, $c = 2$, $c = 2$

Thus $(m, c) = (1, 2)$

For $(x, y) = (3, 3)$

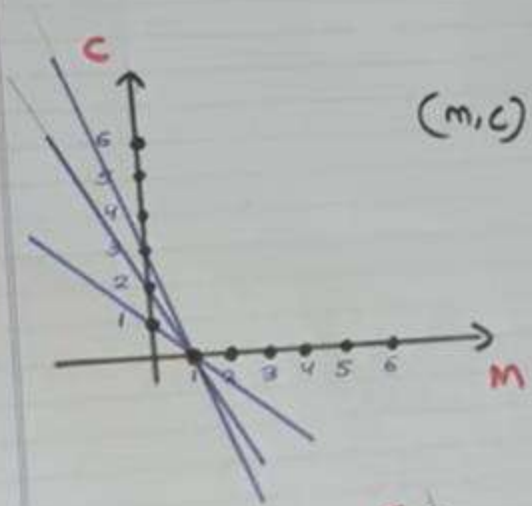
$c = -3m + 3$

if $c = 0$ $\left[\begin{array}{l} 0 = -3m + 3 \\ -3 = -3m \\ m = \frac{-3}{-3} \end{array}\right.$, $m = 1$

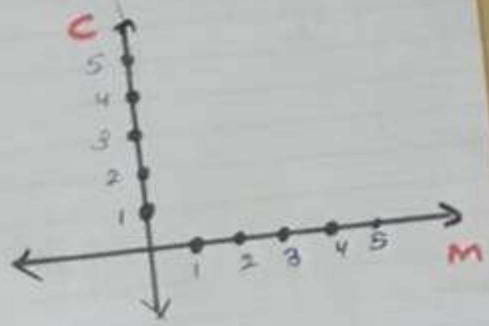
if $m = 0$, $c = 3$

Thus $(m, c) = (1, 3)$

$(m, c) = (1,1), (1,2), (1,3)$



$(m, c) = (1,0)$



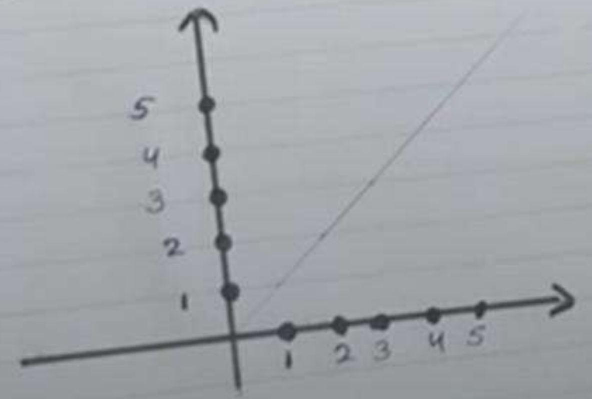
$(m, c) = (1, 0)$

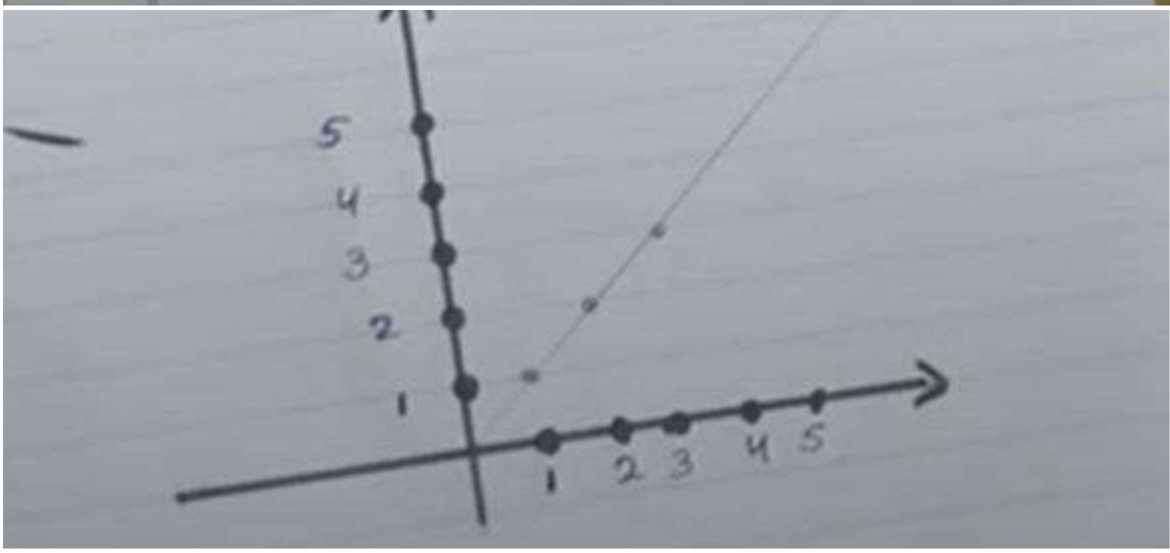
The orignal equation of line

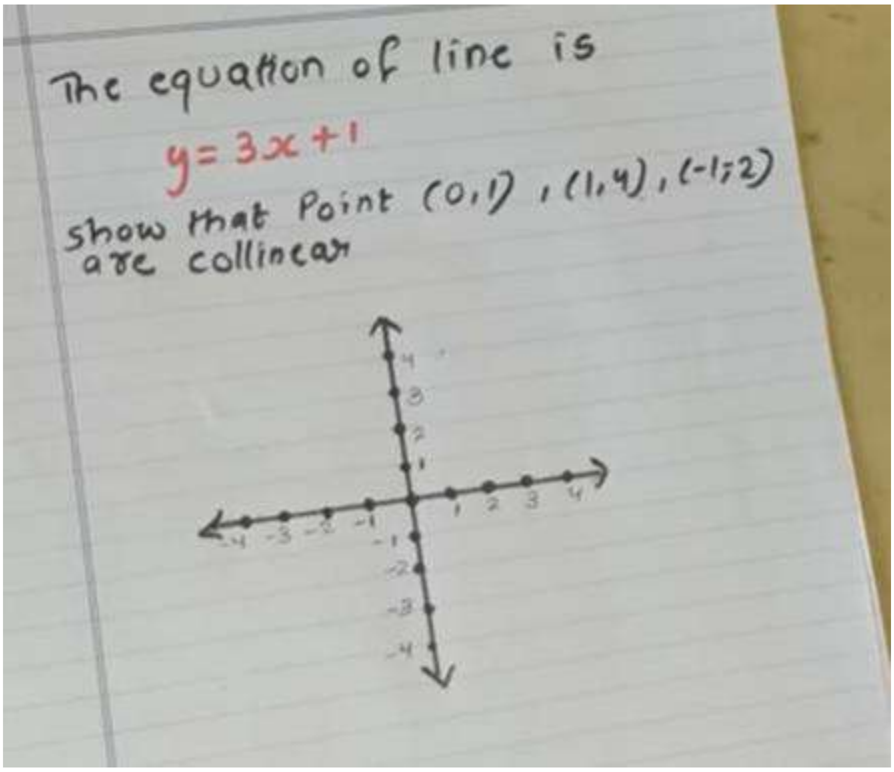
$$y = mx + c$$

$y = x$ ← equation of line

Show that Point $(1,1)$, $(2,2)$, $(3,3)$ are collinear







❖ Edge Linking

- An edge detection algorithm (Roberts, Sobel, Prewitt, LoG etc.) enhance the edges. When implemented, there are normally breaks in lines. Due to this reason, these are generally followed by linking procedures to assemble edge pixels into meaningful edges.

Input Image → Gradient Operator → Edge Linking → A good edge output image

- There are two basic approaches for edge linking:

(1) Local Processing. (This is a simplest approach for linking pixels in a small neighborhood)

(2) Global Processing via the Hough Transform. (Here, we attempt to link edge pixels that lie on specified curves. The Hough transform is designed to detect lines, using the parametric representation of a line.)

❖ Hough Transform (Global Processing)

- Taking this one step further, all pixels which lie on the same line in (x,y) space are represented by lines which all pass through a single point in (m,c) space.
- The single point through which they all pass gives the values of m and c in the equation of the line $y=mx+c$.

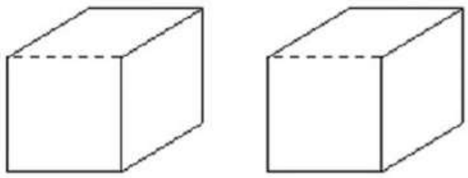
To detect straight lines in an image, we do:

1. Quantize (m,c) space into a two-dimensional array A for appropriate steps of m and c.
2. Initialize all elements of $A(m,c)$ to zero.
3. For each pixel (x',y') which lies on some edge in the image, we add 1 to all elements of $A(m,c)$ whose indices m and c satisfy $y'=mx'+c$.
4. Search for elements of $A(m,c)$ which have large values. Each one found corresponds to a line in the original image.

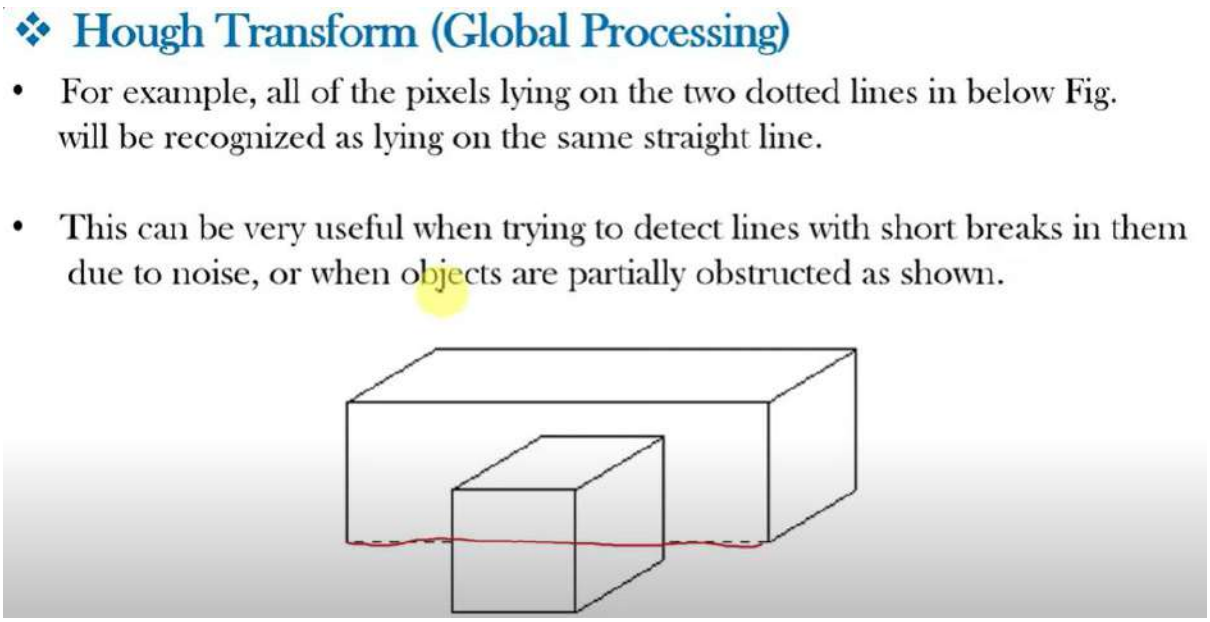
- One useful property of the Hough transform is that the pixels which lie on the line need not all be contiguous.

❖ Hough Transform (Global Processing)

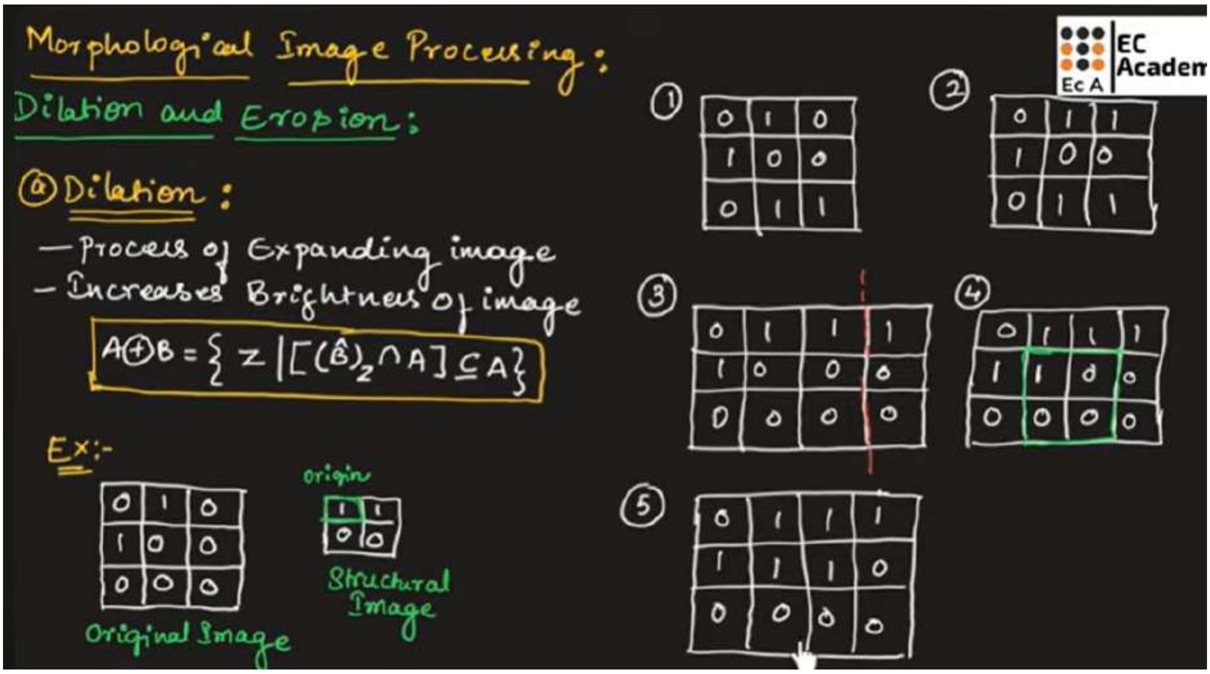
- On the other hand, it can also give misleading results when objects happen to be aligned by chance, as shown by the two dotted lines in Fig. below,



- Indeed, this clearly shows that one disadvantage of the Hough transform method is that it gives an ***infinite line*** as expressed by the pair of m and c values, rather than a finite ***line segment*** with two well-defined endpoints.
- One practical detail is that the $y=mx+c$ form for representing a straight line breaks down for vertical lines, when m becomes infinite.



Dilation and Erosion, Opening and Closing in Image morphology



Check in the original image if the origin of structured image and original image match then replace original image values with the values of mask for Dilation.

Morphological Image Processing:

Dilation and Erosion:

b) Erosion:

- Opposite process of Dilation
- Image shrinking is Obtained

Ex:-

Original Image

1	1	1	1
1	1	0	1
0	1	1	1
0	1	0	0

origin

Structural Image

1	1
1	0

0	1	0	0
0	0	0	0
0	1	0	0
0	0	0	0

Morphological Image processing:

Opening & Closing:

Opening:-

- Erosion is followed by dilation operation.

$A \circ B = (A \ominus B) \oplus B$

Original Image (A), Structural Image (B).

- Identify gaps in an Image
- Edges become sharp (or) smooth
- Isolates objects which are touching one another.

Ex:-

(A) Original Image

1	1	1	0	1	1	1
1	1	1	1	1	1	1
1	1	1	0	1	1	1

(B) Structural Image

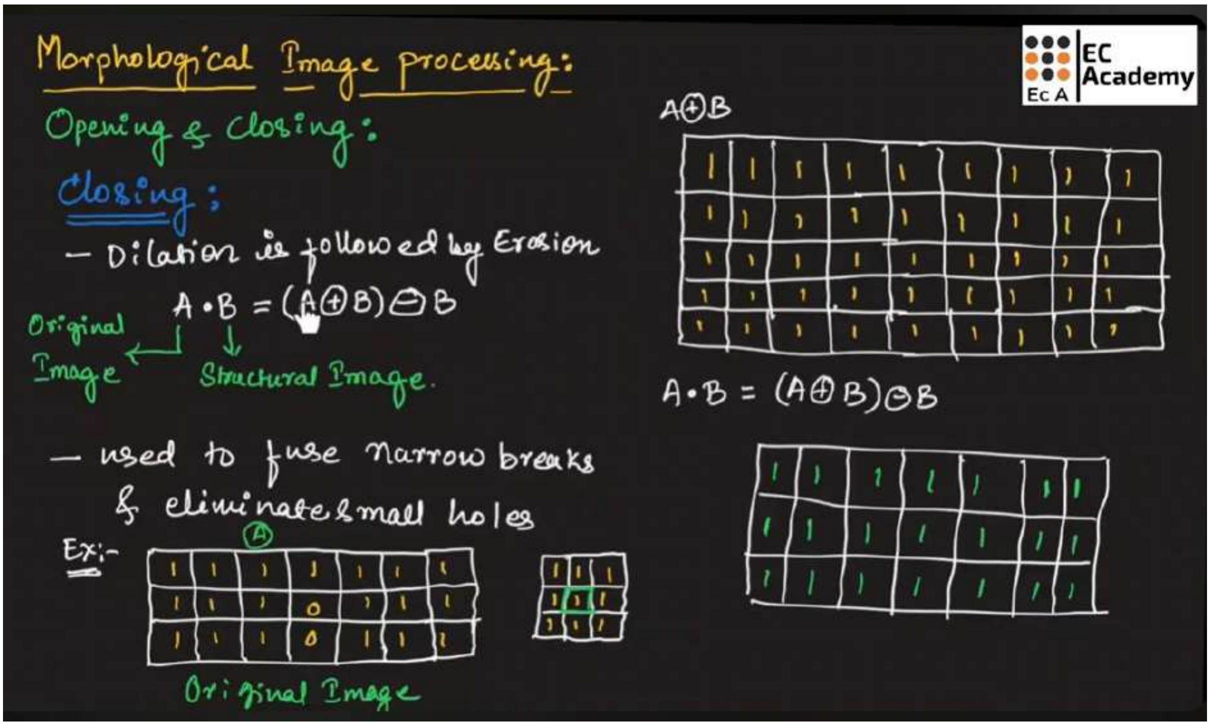
1	1	1
1	1	1
1	1	1

$A \ominus B =$

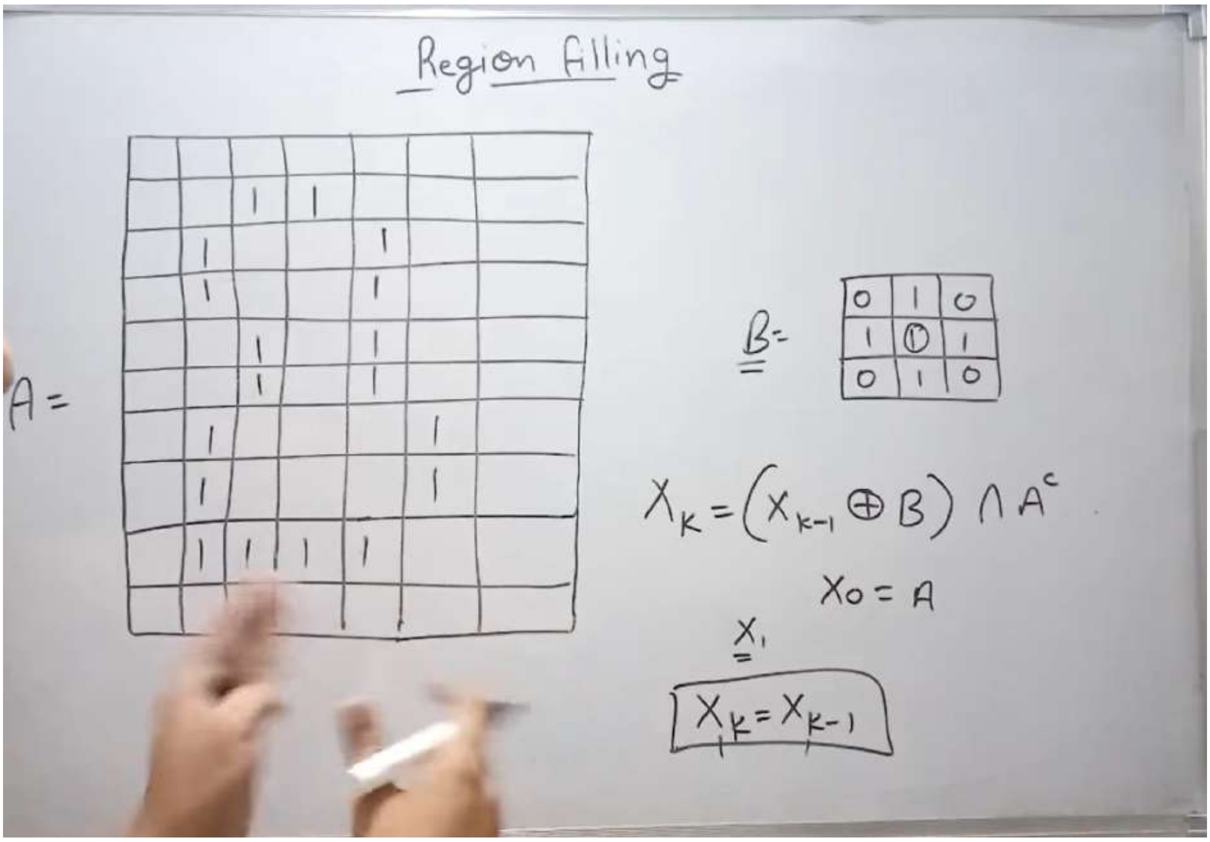
0	0	0	0	0	0	0
0	1	0	1	0	1	0
0	0	0	0	0	0	0

$A \circ B = (A \ominus B) \oplus B$

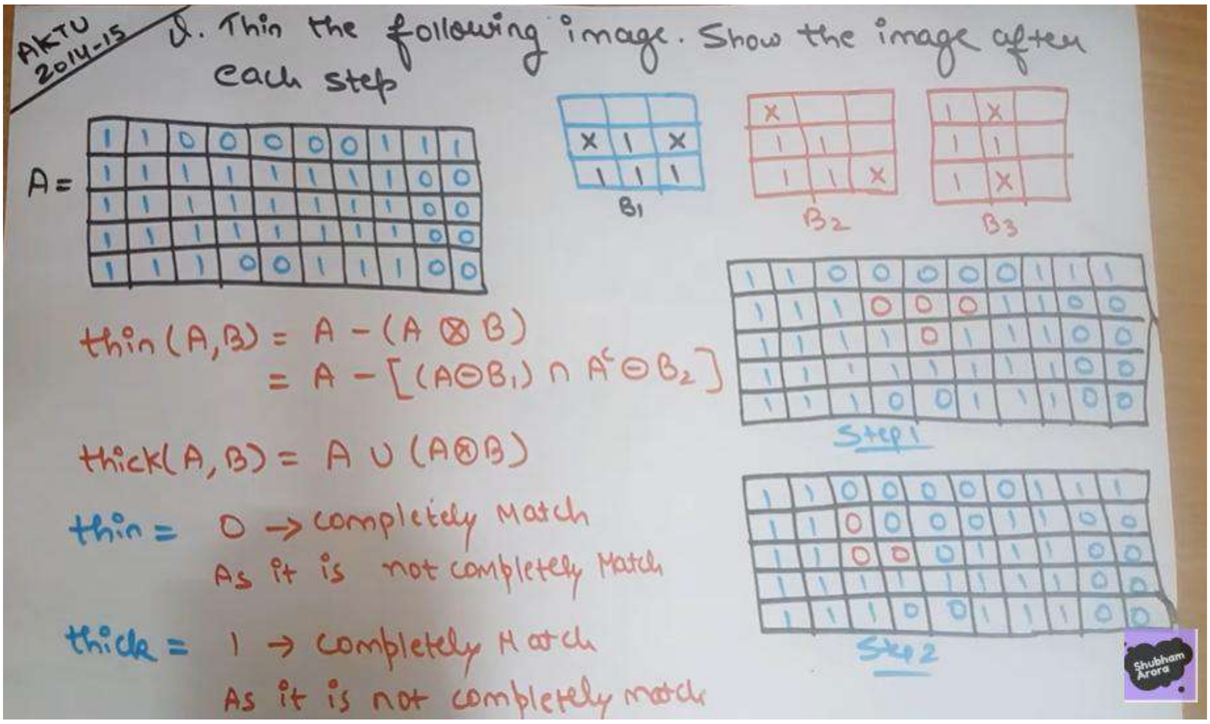
1	1	1	0	1	1	1
1	1	1	1	1	1	1
1	1	1	0	1	1	1



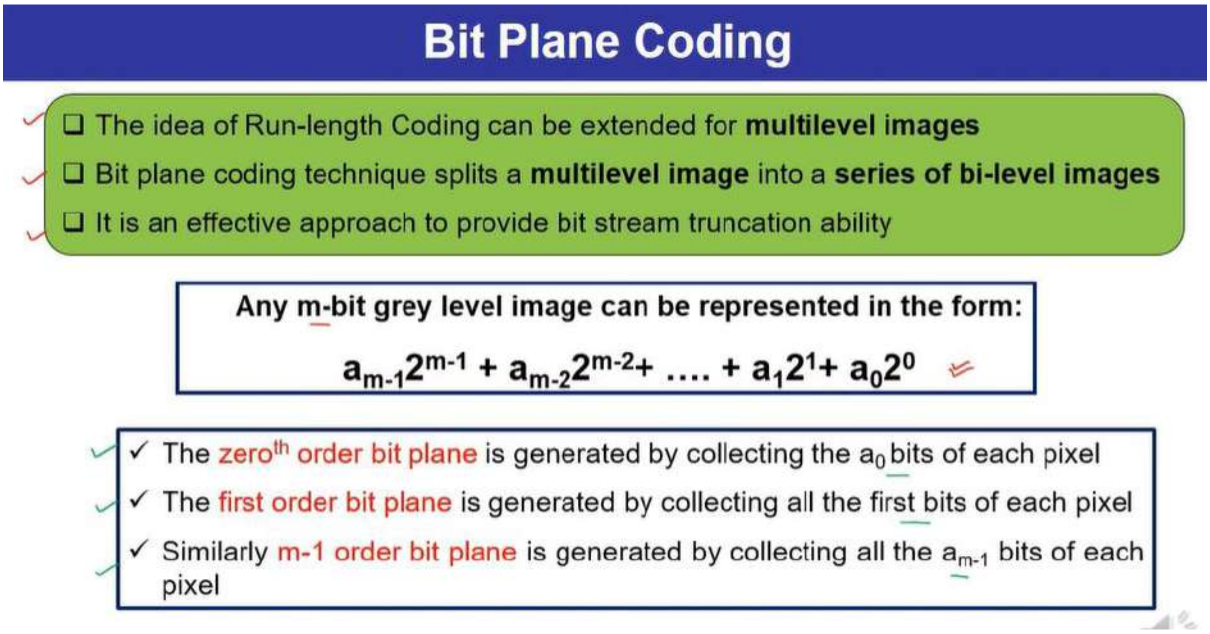
Region Filling



Thinning and Thickening in Image Processing



Bit Plane Coding



Bit Plane Coding

Example: In a given grey scale image is given as apply Bit Plane Coding algorithm.

Discussion:

A =	2	6	6
	6	7	7
	1	2	4

3×3

The individual plane of the image can be compressed using RLC technique

The binary equivalent of image is:

A =	010	110	110
	110	111	111
	001	010	100

$$\Rightarrow A(MSB) = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \qquad A(LSB) = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

$$A(Middle) = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Three Planes
* MSB
* Middle Bit
* LSB

Individual Planes of image can be compressed by using RLC.

Bit Plane Coding

The **disadvantage** of this scheme is that the neighborhoods in the spatial domain like 3 and 4 having binary coded 011 and 100 are not present together in any of the planes

- To avoid this problem, **grey codes** can be used instead of binary codes
- In Grey codes, **successive codes** differ by only one bit

Algorithm for generating grey code can be written as –

$$g_i = \begin{cases} a_i \oplus a_{i+1} & 0 \le i \le m-2 \\ a_i & i = m-1 \end{cases}$$

Bit Plane Coding

Another Bit Plane coding Scheme is
Constant Area Coding (CAC)

- The bit planes have uniform regions of 1's and 0's. a constant portion of the plane can be uniquely coded using less number of bits
- CAC divided the image into a set of blocks of size (m × m) like (8 × 8) or (16 × 16)

Three types of blocks are available:

- A block of all white pixels
- A block of all black pixels
- A block with mixed pixels

Most probable blocks is assigned a single code of either 0 or 1. the remaining blocks are assigned a 2-bit code

White block skipping (WBS) is a scheme where a majority of blocks (white) are assigned a single bit. Rest of blocks including mixed pixel blocks are encoded together.