DIGITAL SIGNAL PROCESSING
UEC – 502

From My Lecture Notes

Lecture No. – 01

for

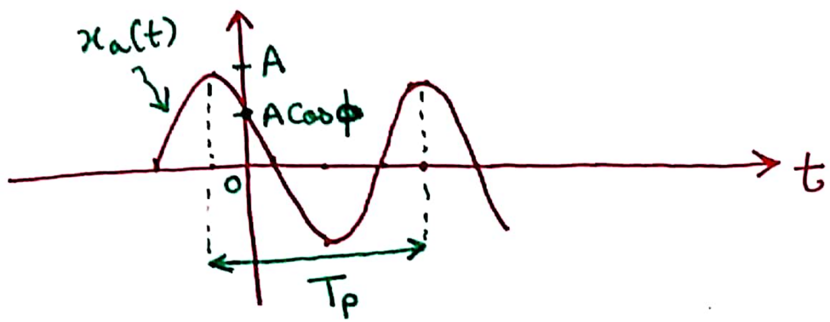
Electronics and Communication Engineering Students
&
Electronics and Computer Engineering Students

Dr. Amit Kumar Kohli
Electronics and Communication Engineering Department
Thapar Institute of Engineering & Technology, Patiala, Punjab, India

"Concept of frequency in Continuous-time & Discrete-time signals"

$$x_a(t) = A \cos(\Omega t + \phi) \qquad -\infty < t < +\infty$$

⇒ Continuous-time sinusoidal signal



For a simple harmonic oscillation

$$\boxed{\begin{aligned} T_p &= 1/F \\ \Omega &= 2\pi F \end{aligned}}$$

in continuous-time domain

T_p is the fundamental period of sinusoidal signal

A is the amplitude of this analog signal

Ω is the angular frequency in radians per second

ϕ is the phase in radians

F is the frequency in cycles per second (Hz)

For F = any fixed value = $1/T_p$, the sinusoidal signal $x_a(t+T_p) = x_a(t)$

Continuous-time sinusoidal signals with distinct frequencies are distinct.

Increasing the frequency F leads to an increase in the rate of oscillations of signal $x_a(t)$

Let $\bar{x}_a(t) = A\, e^{j(\Omega t+\phi)}$ (Complex exponential signal)

$$x_a(t) = \mathrm{Re}\{\bar{x}_a(t)\} = A \cos(\Omega t + \phi)$$

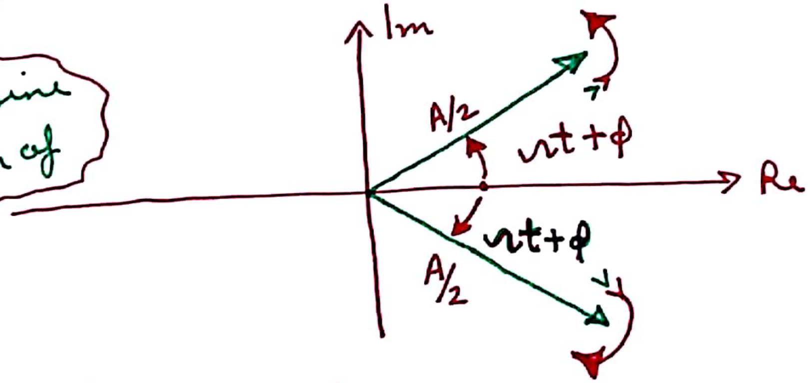
$$= \underbrace{\frac{A}{2} e^{+j(\Omega t+\phi)} + \frac{A}{2} e^{-j(\Omega t+\phi)}}$$

A sinusoidal signal can be obtained by adding two equal-amplitude complex-conjugate exponential signals, which are known as phasors.

Euler Identity

$$\boxed{e^{\pm j\phi} = \cos\phi \pm j\sin\phi}$$

Representation of cosine function by a pair of phasors.



A positive freq. corresponds to counterclockwise uniform angular motion

. A negative freq. corresponds to clockwise angular motion.

⇒ As the time progresses, the phasors rotate in opposite directions with angular frequencies $\pm\Omega$ radians per second.

⇒ Frequency range for analog sinusoids is $-\infty < F < +\infty$

$$-\infty < \Omega < +\infty$$

⇒ Discrete-time sinusoidal signal

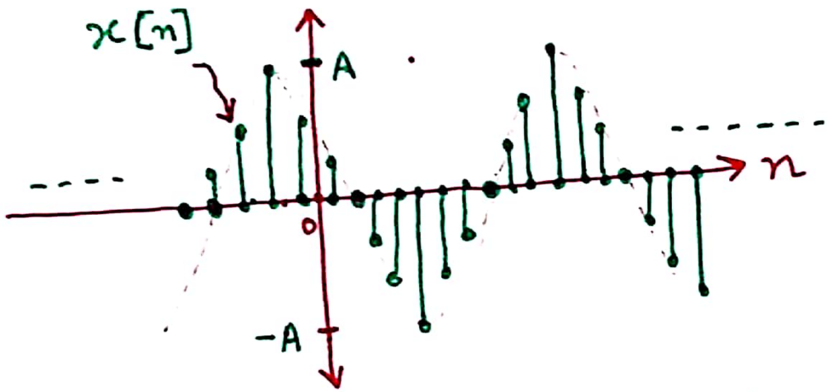
$$x[n] = A\cos(\omega n + \phi) \qquad -\infty < n < +\infty$$

n is an integer variable, which is called the sample number

A is the amplitude of sinusoid.

ω is the angular frequency in radians per sample

ϕ is the phase in radians.



$$\omega = 2\pi f$$

f is the frequency with dimensions cycles per sample

⇒ A discrete-time sinusoid is periodic only if its frequency "f" is a rational number (with a period $N > 0$). Such that

$$x[n+N] = x[n] \quad \text{for all } n$$

The smallest value of N, for which this relation is true, is called fundamental period.

Let $x[n] = \cos(2\pi f n + \phi)$

If it fulfils the conditions for periodicity, then

$$x[n] = \cos(2\pi f n + \phi) = \cos(2\pi f (n+N) + \phi)$$

This relation is correct, if and only if $2\pi f N = 2\pi k$ (k is an integer)

$$f = \frac{k}{N}$$ $N \rightsquigarrow$ fundamental period

$\Rightarrow$ A discrete-time sinusoidal signal is periodic sinusoid, only if its frequency "f" can be represented as the ratio of two integers.

A small change in frequency "f" can lead to a large change in period.

$f_1 = \frac{31}{90} = \frac{31}{90}$	$f_2 = \frac{30}{90} = \frac{1}{3}$
$\Downarrow$	$\Downarrow$
$N_1 = 90$	$N_2 = 3$

⇒ Discrete-time sinusoids whose frequencies are separated by an integer multiple of 2π are identical.

\# $\cos[\omega n + \phi] = x_1[n]$

\# $\cos[(\omega + 2\pi k)n + \phi] = \cos[\omega n + 2\pi k n + \phi] = x_2[n]$

If k is an integer, then $x_2[n] = \cos[\omega n + \phi]$

and, therefore $x_1[n] = x_2[n]$

$$\omega_k = \omega_0 + 2\pi k \quad \text{with } k = 0, 1, 2, \ldots$$
$$\text{with } -\pi \leq \omega_0 \leq +\pi$$

Hence, all sinusoidal sequences

$$x_k[n] = A\cos[\omega_k n + \phi] \quad \text{with } k = 0, 1, 2 \ldots$$

are identical (indistinguishable)

⇒ All frequencies in the range $-\pi \leq \omega < +\pi$ are regarded as unique
$-\frac{1}{2} \leq f < +\frac{1}{2}$

⇒ All frequencies in the range where $|\omega| > \pi$ or $|f| > 1/2$ are regarded as alias.

Any sequence resulting from a sinusoid with a frequency $|\omega| > \pi$ or $|f| > \frac{1}{2}$ is identical to a sequence obtained from a sinusoidal signal with frequency $|\omega| < \pi$.

$\Rightarrow$ The highest rate of oscillations in a discrete-time sinusoid is attained when $\omega = \pi$ or $\omega = -\pi$

i.e., $f = \frac{1}{2}$ or $f = -\frac{1}{2}$

$\Rightarrow$ The discrete-time sinusoidal signal is found to be constant signal at $\omega = 2\pi$ (also at $\omega = 0$). Therefore, if we increase ω of sinusoidal discrete-time signal from π to 2π, then its rate of oscillation decreases.

$\Rightarrow$ The frequency range for discrete-time sinusoids is finite with duration 2π. (which is known as fundamental range)

$$x[n] = A\cos[\omega n + \phi] = \frac{A}{2} e^{+j(\omega n+\phi)} + \frac{A}{2} e^{-j(\omega n+\phi)}$$

$\Rightarrow$ Choose fundamental frequency range as

$-\pi \leq \omega < +\pi$ or $-\frac{1}{2} \leq f < +\frac{1}{2}$

$0 \leq \omega < 2\pi$ or $0 \leq f < 1$

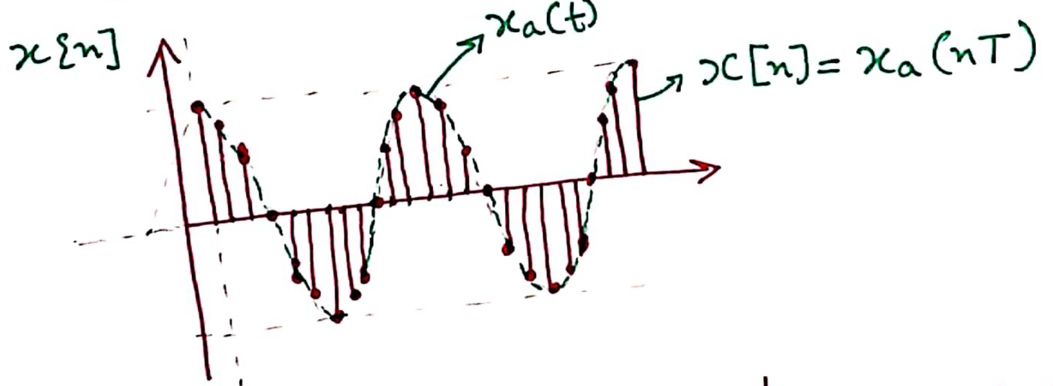
"Sampling of analog signals $\rightarrow$ Periodic or Uniform Sampling"

$\Rightarrow$ $x[n]$ is the discrete-time signal obtained by "taking samples" of the analog signal $x_a(t)$ every T seconds.

$\Rightarrow$ The interval T between successive samples is called the sampling period or sample interval.

$\Rightarrow$ $F_s = 1/T$ is called the sampling rate (samples per second) or the sampling frequency (Hz).

Let Analog signal $x_a(t) = A\cos(2\pi F t + \phi)$



Discrete-time signal (Sampled periodically)

$$\text{Sampled signal } x_a(nT) = x_a(t)\Big|_{t=nT} = A\cos(2\pi F nT + \phi)$$

$$= A\cos\left(2\pi n \frac{F}{F_s} + \phi\right)$$

$x[n] = A \cos(2\pi f n + \phi)$ with $f = \frac{F}{F_s}$ (Relative or normalized freq.)

Frequency variables F and f are linearly related, which leads to

$$\omega = \Omega T$$

For continuous-time signals, $-\infty < F < \infty$; $-\infty < \Omega < +\infty$

For discrete-time signals, $-\frac{1}{2} \leq f < +\frac{1}{2}$; $-\pi \leq \omega < \pi$

$-\frac{1}{2} < f \leq +\frac{1}{2}$; $-\pi < \omega \leq \pi$

$\Downarrow f = F/F_s$ $\qquad\qquad$ $\Downarrow \omega = \Omega T$

$-\frac{1}{2T} < F \leq +\frac{1}{2T}$ $\qquad\qquad$ $-\frac{\pi}{T} < \Omega \leq +\frac{\pi}{T}$

$\Downarrow$

$$-\frac{F_s}{2} < F \leq +\frac{F_s}{2}$$

Therefore, $F_{max} = \frac{F_s}{2} = \frac{1}{2T}$ & $\Omega_{max} = \pi F_s = \pi/T$

The frequency of continuous-time sinusoidal signal when sampled at rate $F_s = 1/T$ must fall in the range $|\Omega| \leq \pi/T$.

In general, the sampling of a continuous-time signal $x_a(t) = A\cos(2\pi F_0 t + \phi)$ with a sampling rate $F_s = 1/T$ leads to a discrete-time signal

$$\underline{x[n] = A\cos[2\pi f_0 n + \phi]}$$

where, $f_0 = F_0/F_s$. If we consider that $F_s/2 \geq |F_0|$, then the relative frequency is $|f_0| \leq 1/2$, which is the frequency range for discrete-time signals.

⇒ As relationship between F_0 and f_0 is one-to-one, therefore it is possible to reconstruct the analog signal $x_a(t)$ from the samples $x[n]$.

"Identify the frequency components, that can cause aliasing": ⇒ ?

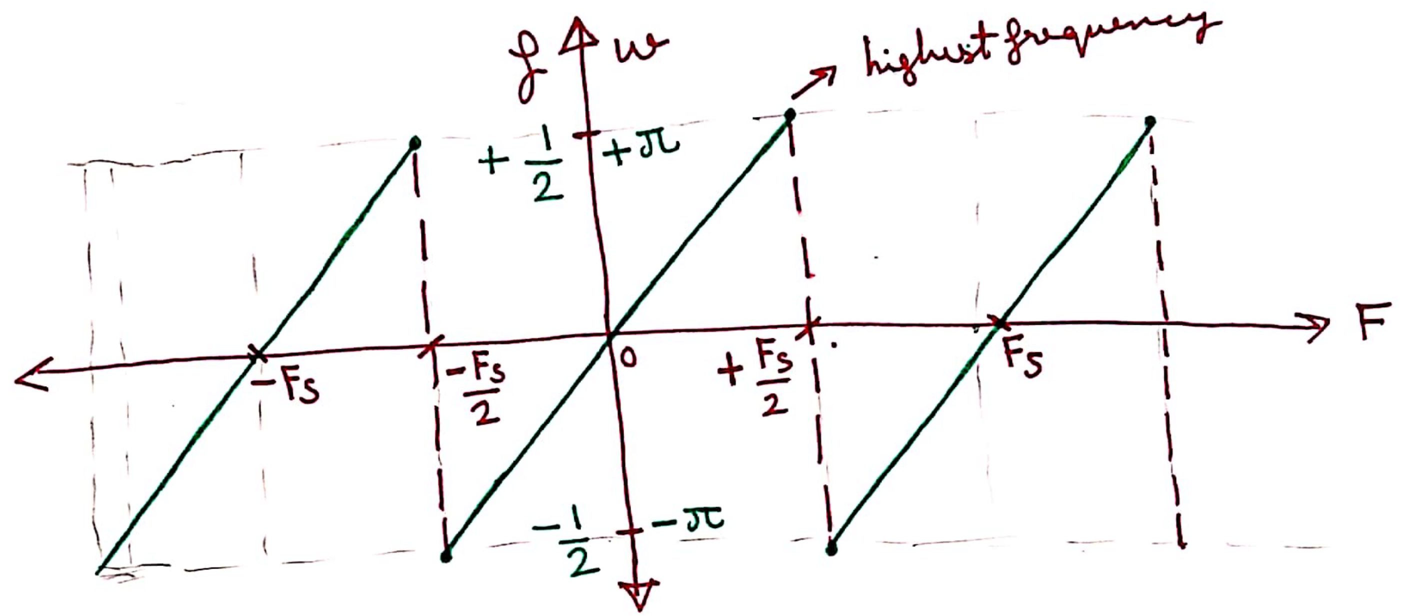
Let $x_a(t) = A\cos[2\pi F_k t + \phi]$ with $F_k = F_0 + kF_s$; $k = \pm 1, \pm 2, \ldots$

$$x[n] = x_a(nT) = A\cos\left\{2\pi\left(\frac{F_0 + kF_s}{F_s}\right)n + \phi\right\}$$

$$= A\cos\left\{2\pi \frac{F_0}{F_s} n + \phi + 2\pi k n\right\} = \underline{A\cos(2\pi f_0 n + \phi)}$$

$F_k = F_0 + kF_s$ (freq.) are indistinguishable from the frequency F_0 after sampling, and hence these are aliases of F_0.

Relationship between Continuous-time & Discrete-time Frequency variables in the case of periodic sampling



⇒ $F_s/2$ or $\omega = \pi$ is considered to be the pivotal point, and reflect or fold the alias frequency to the range $0 \leq \omega \leq \pi$.
Therefore, the frequency $F_s/2$ $(\omega = \pi)$ is called the folding freq. and $F_s/2$ is the point of reflection in case of periodic sampling.

Reference Books ⟹

⟹ "Digital Signal Processing: Principles, Algorithms and Applications," by J. G. Proakis and D. G. Manolakis; Prentice-Hall, 2007.

⟹ "Theory and Application of Digital Signal Processing," by L.R. Rabiner and B. Gold; Prentice-Hall, 2003.

⟹ "Discrete Time Signal Processing," by A.V. Oppenheim and R.W. Schafer; Prentice-Hall, 2001.

⟹ "Digital Signal Processing," by S.K. Mitra; Tata McGraw-Hill, 2006.

⟹ "Digital Signal Processing," by S. Salivahanan; McGraw-Hill, 2019.

⟹ "Digital Filters: Analysis and Design," by A. Antoniou; Tata McGraw-Hill, 1979.

⟹ "Digital Signal Processing," by E. Ifeachor and B.W. Jervis; Pearson Education, 2007.

⟹ "Signals and Systems," by A.V. Oppenheim, A.S. Willsky and S.H. Nawab; Prentice-Hall of India, 1999. (Prerequisite)

Thanks for attending this session on DSP

DIGITAL SIGNAL PROCESSING
UEC – 502

From My Lecture Notes

Lecture No. – 02

for

Electronics and Communication Engineering Students

&

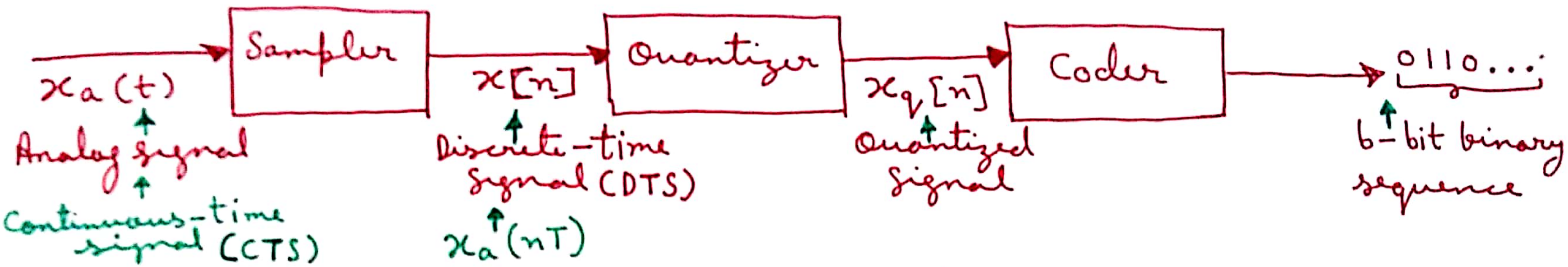
Electronics and Computer Engineering Students

Dr. Amit Kumar Kohli

Electronics and Communication Engineering Department

Thapar Institute of Engineering & Technology, Patiala, Punjab, India

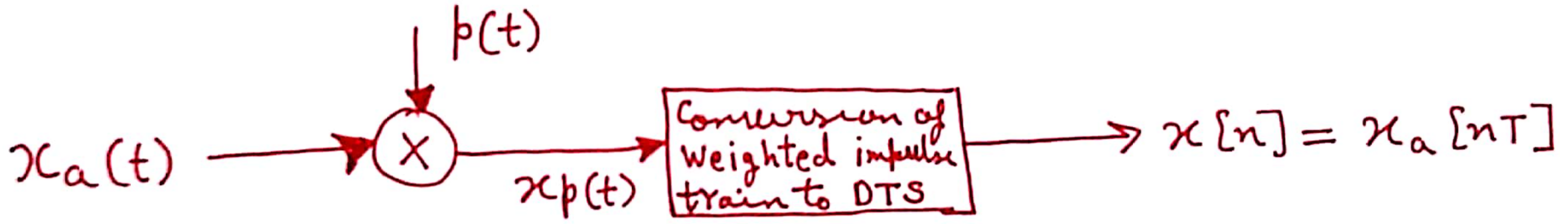
— Analog-to-Digital Conversion —



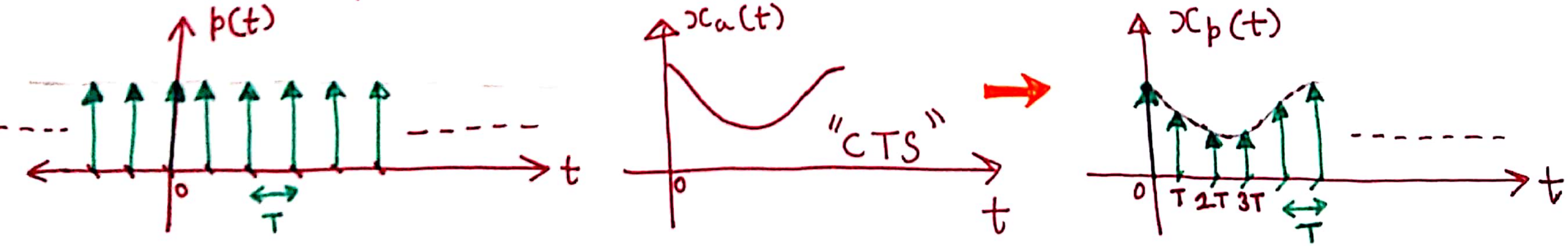
$$e_q[n] = x[n] - x_q[n] = \text{quantization error}$$

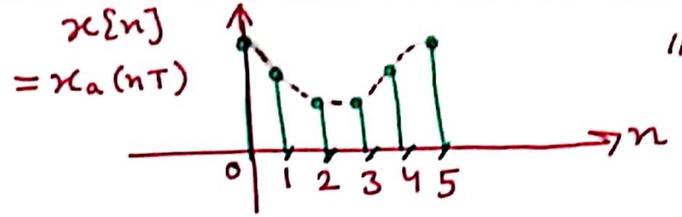
⇒ We shall focus on the frequency-domain analysis of sampling process

⇓



$p(t)$ represents a Dirac-Comb, consisting of an infinite sequence of uniformly spaced delta functions.





"DTS"

$$\boxed{\Omega_s = 2\pi F_s = \frac{2\pi}{T}}$$

F_s is the Sampling rate

$$x_p(t) = x_a(t) \times p(t)$$ (multiplication operation)

$$p(t) = \sum_{n=-\infty}^{+\infty} \delta(t - nT)$$

$$x_p(t) = \sum_{n=-\infty}^{\infty} x_a(nT)\,\delta(t - nT)$$

$$X_p[j\Omega] = \text{CTFT}\{x_p(t)\}$$

CTFT → Continuous-time Fourier-transform

$$X_p[j\Omega] = \sum_{n=-\infty}^{+\infty} x_a(nT)\, e^{-j\Omega nT}$$

→ B

$$p(t) = \sum_{k=-\infty}^{+\infty} a_k\, e^{j\frac{2\pi}{T}kt}$$

with $a_k = 1/T$ {Fourier-series representation}

$$X_p[j\Omega] = X_a[j\Omega] * P[j\Omega] / 2\pi$$

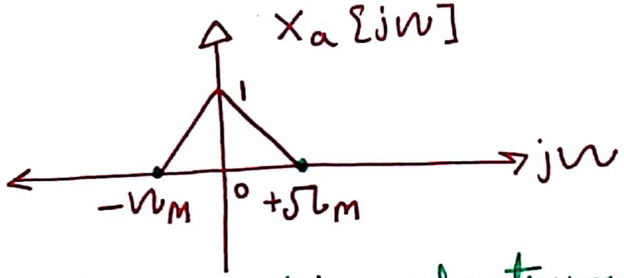
$*$ → linear convolution operator

$$P[j\Omega] = \text{CTFT}\{p(t)\}$$

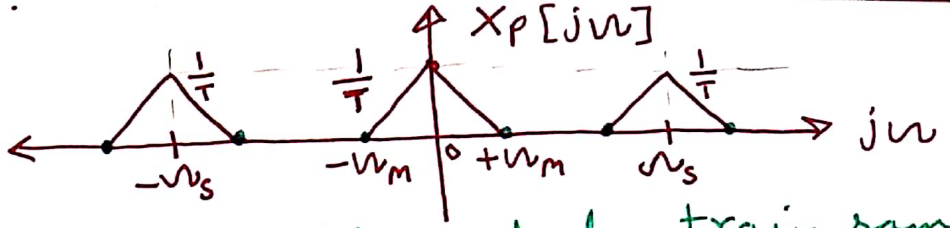
$$= \sum_{k=-\infty}^{+\infty} \frac{2\pi}{T}\, \delta\left(j\Omega - j\frac{2\pi}{T}k\right)$$

$$X_p[j\Omega] = \sum_{k=-\infty}^{\infty} \frac{1}{T} X_a\left[j\Omega - j\frac{2\pi}{T}k\right]$$

→ A



Continuous-time spectrum of $X_a[j\Omega]$



Spectrum after impulse-train sampling (No aliasing, if $\Omega_s \geq 2\Omega_M$)

DTFT → Discrete-time Fourier-transform.

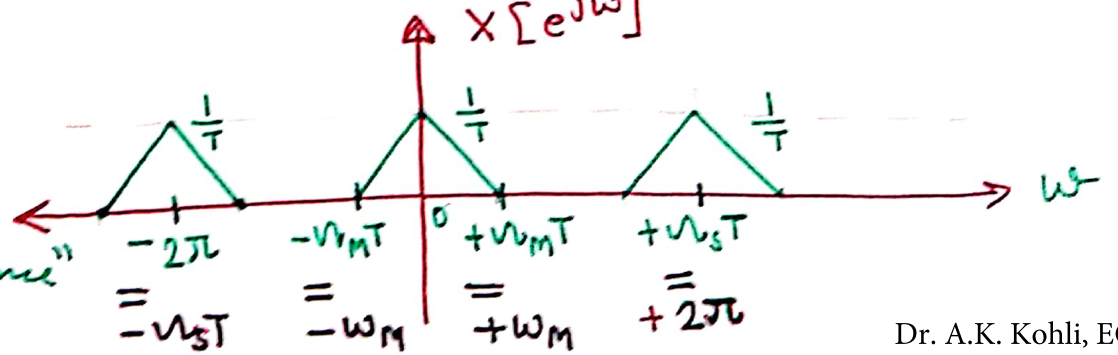
$$X[e^{j\omega}] = \text{DTFT}\{x[n]\} = \sum_{n=-\infty}^{\infty} x[n]\, e^{-j\omega n} \longrightarrow \text{(C)}$$

⇒ As $x[n] = x_a[nT]$ and $\Omega T = \omega$, therefore $\underbrace{X[e^{j\omega}]}_{\text{(C)}}$ and $\underbrace{X_p[j\Omega]}_{B}$ are related through

$$X[e^{j\omega}] = X_p\left[j\frac{\omega}{T}\right]$$

⇒ Consequently, the equation (A) can be used to express $X[e^{j\omega}]$ as

$$X[e^{j\omega}] = \sum_{k=-\infty}^{\infty} \frac{1}{T} X_a\left[j(\omega - 2\pi k)/T\right]$$

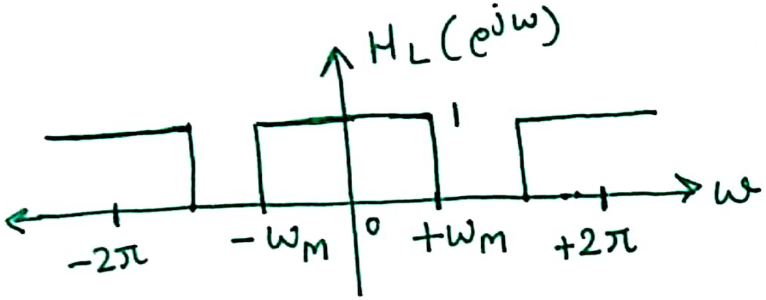


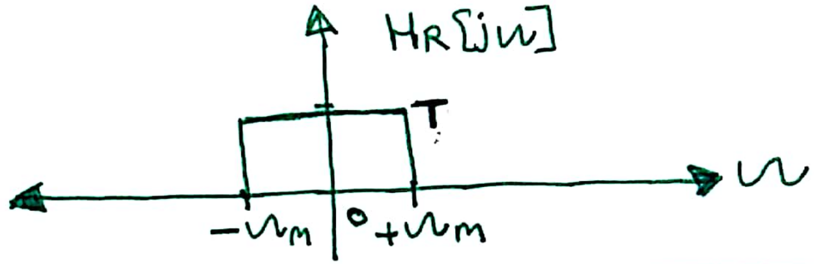
"Spectrum of discrete-time sequence" $x[n]$

Signal Reconstruction Process $\rightarrow$

$$x[n] \rightarrow \boxed{H_L(e^{j\omega})} \rightarrow \boxed{\text{Conversion of sequence to weighted impulse train}} \xrightarrow{x_p(t)} \boxed{\text{Reconstruction filter } H_R[j\Omega]} \rightarrow \hat{x}_a(t)$$

Reconstructed signal





From equation (A),

$$\boxed{\frac{2\pi}{\Omega_s} X_p[j\Omega] = X_a[j\Omega] + \sum_{k\neq 0} X_a\left[j\Omega - j\frac{2\pi k}{T}\right]} \Rightarrow \text{(D)}$$

$$\frac{2\pi}{\Omega_s} X_p[j\Omega] = \sum_{k=-\infty}^{\infty} X_a\left[j\Omega - j\frac{2\pi k}{T}\right]$$

$\Rightarrow$ If we consider $F_s = 2F_M$ or $\Omega_s = 2\Omega_M$, then $\Omega_M = \frac{\Omega_s}{2}$ in $H_R[j\Omega]$.

It follows that

$$\hat{X}_a[j\Omega] = X_p[j\Omega] \times H_R[j\Omega] = \frac{2\pi}{\Omega_s} X_p[j\Omega] \longrightarrow \text{(E)}$$

From equation (B) and (E)

$$\hat{X}_a[j\Omega] = \frac{2\pi}{\Omega_s} \sum_{n=-\infty}^{\infty} x_a[nT]\, e^{-j\Omega nT}$$

$$\hat{x}_a(t) = \text{Inverse CTFT}\{\hat{X}_a[j\Omega]\} = \frac{1}{2\pi}\int_{-\Omega_m}^{+\Omega_m} \hat{X}_a[j\Omega]\, e^{j\Omega t}\, d\Omega$$

$$= \frac{1}{\Omega_s}\sum_{n=-\infty}^{\infty}\int_{-\Omega_m}^{+\Omega_m} x_a[nT]\, e^{j\Omega(t-nT)}\, d\Omega$$

$$= \frac{1}{2\Omega_m}\sum_{n=-\infty}^{+\infty} \frac{2\sin[\Omega_m(t-nT)]}{(t-nT)}\, x_a[nT]$$

$$= \sum_{n=-\infty}^{\infty} \frac{\sin[\pi\, 2F_m(t-nT)]}{\pi\, 2F_m(t-nT)}\, x_a[nT]$$

$$\boxed{\hat{x}_a(t) = \sum_{n=-\infty}^{\infty} x_a[nT]\, \text{sinc}\,(2F_M t - n)} \quad -\infty < t < \infty$$

$$= \sum_{n=-\infty}^{\infty} x_a\left[\frac{n}{2F_M}\right] \text{sinc}\,(2F_M t - n)$$

✻

$$\boxed{T = \frac{1}{F_S} = \frac{1}{2F_M} \quad \& \quad \frac{\sin \pi x}{\pi x} = \text{sinc}(x)}$$

Therefore, $\hat{x}_a(t)$ can be recovered (approx.) from its sample values using the interpolation function $\text{sinc}\,(2F_M t)$.

$\Rightarrow$ A band-limited signal of finite energy that has no frequency components higher than $F_M = \frac{\Omega_M}{2\pi}$ (Hz) is completely described by specifying the values of the signal instants of time separated by $T = \frac{1}{2F_M}$ seconds.

$\Rightarrow$ A band-limited signal of finite-energy that has no frequency components higher than $F_M = \frac{\Omega_M}{2\pi}$ (Hz) is completely recovered from a knowledge of its samples taken at the rate of $2F_M$ samples per second.

$\Rightarrow$ $F_s = F_N = 2F_M$ is called Nyquist rate

$\Rightarrow$ $T = 1/2F_M = 1/F_N$ is called Nyquist interval $= T_N$

$\Rightarrow$ A simple remedy that avoids aliasing and any other potentially troublesome situation is to sample the analog signal at a rate higher than the Nyquist rate.

— Introduction to Laplace Transform —

The Laplace transform of a general signal $x(t)$ is defined as

$$X(s) \triangleq \int_{-\infty}^{+\infty} x(t)\, e^{-st}\, dt \quad \text{with } s = \underset{\text{complex variable}}{\sigma + j\Omega} \longrightarrow \text{(A1)}$$

$$x(t) \overset{L}{\longleftrightarrow} X(s)$$

$$X(s)\Big|_{s=j\Omega} = \text{CTFT}\{x(t)\} \quad \left\{\begin{matrix}\text{Continuous-time} \\ \text{Fourier Transform}\end{matrix}\right\}$$

$$X(s) = X(\sigma + j\Omega) = \int_{-\infty}^{\infty} \left[x(t)\, e^{-\sigma t}\right] e^{-j\Omega t}\, dt \longrightarrow \text{(B1)}$$

$$= \text{CTFT}\{x(t)\, e^{-\sigma t}\}$$

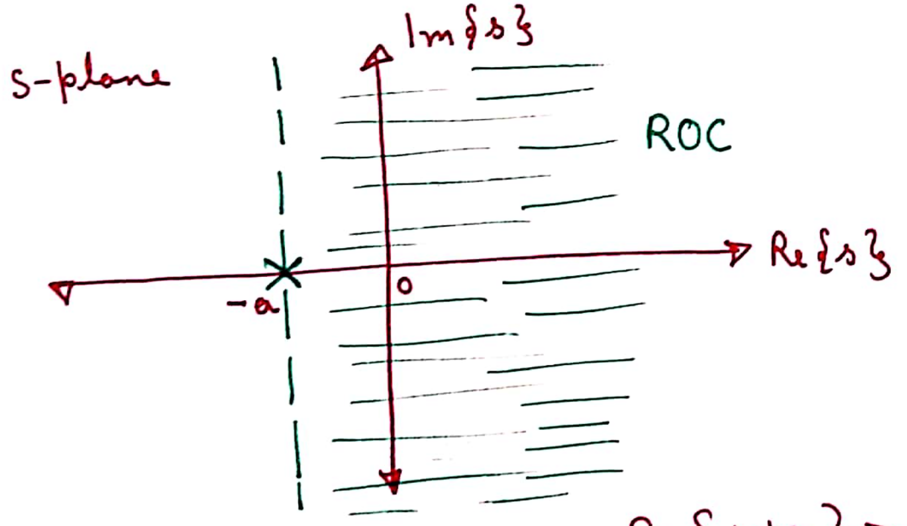
⇛ The range of values of "s" for which the integral in equation (A1) converges is referred to as region of convergence (ROC) of Laplace transform.
The ROC consists of those values of $s = \sigma + j\Omega$ for which the continuous-time Fourier Transform of $x(t)\, e^{-\sigma t}$ converges.

⇛ In some cases, the Laplace transform exists, but CTFT doesn't.

Example: Let the signal $x(t) = e^{-at}u(t)$ for $a>0$

$$X(s) = \int_{-\infty}^{\infty} e^{-at}u(t)\, e^{-st}\,dt = \int_{0}^{\infty} e^{-at}e^{-st}\,dt$$

$$= \frac{1}{s+a} \qquad Re\{s\} > -a$$



$Re\{s+a\} > 0$

Let the signal $x(t) = -e^{-at}u(-t)$ for $a>0$

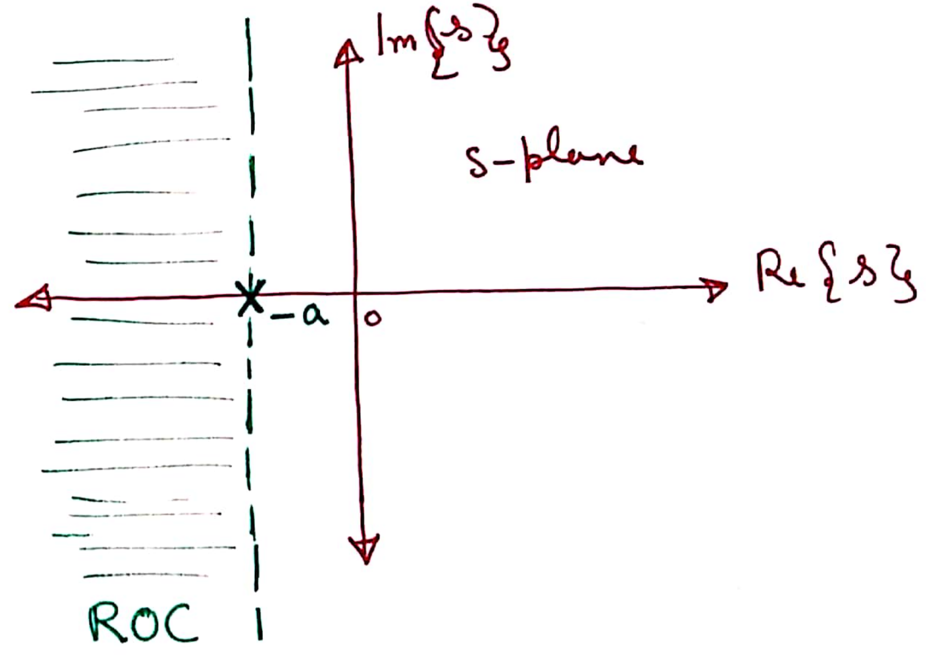
$$X(s) = -\int_{-\infty}^{\infty} e^{-at}e^{-st}u(-t)\,dt$$

$$= -\int_{-\infty}^{0} e^{-at}e^{-st}\,dt$$

$$= -\int_{-\infty}^{0} e^{-(s+a)t}\,dt$$

$$= \frac{1}{s+a} \qquad Re\{s+a\} < 0$$

$Re\{s\} < -a$



⇒ If Laplace transform $X(s)$ of $x(t)$ is rational, then if $x(t)$ is right sided, the ROC is the region in the s-plane to the right of the rightmost pole. If $x(t)$ is left sided, the ROC is the region in the s-plane to the left of leftmost pole.

⇒ For a system with a rational system function, causality of the system is equivalent to the ROC being the right-half plane to the rightmost pole.

⇒ An LTI system is stable, if and only if the ROC of its system function $H(s)$ includes the $j\Omega$-axis i.e., $Re\{s\}=0$.

"The stability of an LTI system is equivalent to its impulse response being absolutely integrable, in that case, the continuous-time Fourier-transform of the impulse response converges

⇒ LTI Systems Characterized by Linear Constant-Coefficient differential Equation →

$$\sum_{k=0}^{N} a_k \frac{d^k y(t)}{dt^k} = \sum_{k=0}^{M} b_k \frac{d^k x(t)}{dt^k}$$

where, $x(t)$ is input signal

$y(t)$ is output of LTI system

Applying the Laplace transform to both sides and using the linearity and differentiation properties repeatedly, we obtain

$$\left[\sum_{k=0}^{N} a_k s^k\right] Y(s) = \left[\sum_{k=0}^{M} b_k s^k\right] X(s)$$

$$\frac{Y(s)}{X(s)} = \frac{\left[\sum_{k=0}^{M} b_k s^k\right]}{\left[\sum_{k=0}^{N} a_k s^k\right]} = H(s)$$

The system function for an LTI system specified by a differential equation is always rational, with zeros at the solutions of

$$\sum_{k=0}^{M} b_k s^k = 0$$

and poles at the solutions of

$$\sum_{k=0}^{N} a_k s^k = 0$$

The linear constant-coefficient differential equation by itself does not constrain the ROC, however with additional information about the stability or causality of the system, the region of convergence can be inferred.

"A generalization of CTFT is known as the Laplace transform"

⇒ Inverse Laplace transform equation is

$$h(t) = \frac{1}{2\pi j} \int_{\sigma - j\infty}^{\sigma + j\infty} H(s)\, e^{st}\, ds$$

Assuming no multiple-order poles, and assuming that the order of the denominator polynomial is greater than the order of the numerator polynomial, we first perform a partial-fraction expansion to obtain

$$H(s) = \sum_{i=1}^{m} \frac{A_i}{(s+a_i)}$$

From ROC of $H(s)$, the ROC of each of the individual terms in above equation can be inferred, and then the inverse Laplace transform of each of these terms can be determined.

Reference Books ⇒

⇒ "Digital Signal Processing: Principles, Algorithms and Applications," by J. G. Proakis and D. G. Manolakis; Prentice-Hall, 2007.

⇒ "Theory and Application of Digital Signal Processing," by L.R. Rabiner and B. Gold; Prentice-Hall, 2003.

⇒ "Discrete Time Signal Processing," by A.V. Oppenheim and R.W. Schafer; Prentice-Hall, 2001.

⇒ "Digital Signal Processing," by S.K. Mitra; Tata McGraw-Hill, 2006.

⇒ "Digital Signal Processing," by S. Salivahanan; McGraw-Hill, 2019.

⇒ "Digital Filters: Analysis and Design," by A. Antoniou; Tata McGraw-Hill, 1979.

⇒ "Digital Signal Processing," by E. Ifeachor and B.W. Jervis; Pearson Education, 2007.

⇒ "Signals and Systems," by A.V. Oppenheim, A.S. Willsky and S.H. Nawab; Prentice-Hall of India, 1999. (Prerequisite)

Thanks for attending this session on DSP

DIGITAL SIGNAL PROCESSING
UEC – 502

From My Lecture Notes

Lecture No. – 03

for

Electronics and Communication Engineering Students

&

Electronics and Computer Engineering Students

Dr. Amit Kumar Kohli
Electronics and Communication Engineering Department
Thapar Institute of Engineering & Technology, Patiala, Punjab, India

— Introduction to z-transform —

≡ "Z-transform is the discrete-time counterpart of Laplace transform" ≡

The z-transform of a general discrete-time signal $x[n]$ is defined as

$$X[z] \triangleq \sum_{n=-\infty}^{+\infty} x[n]\, z^{-n}$$; where $z = re^{j\omega}$ is a complex variable
– polar form –

$$x[n] \xleftrightarrow{z} X[z]$$

r is the magnitude of z
ω is the angle of z

$$X[z] = X(re^{j\omega}) = \sum_{n=-\infty}^{+\infty} \{x[n]\, r^{-n}\}\, e^{-j\omega n}$$

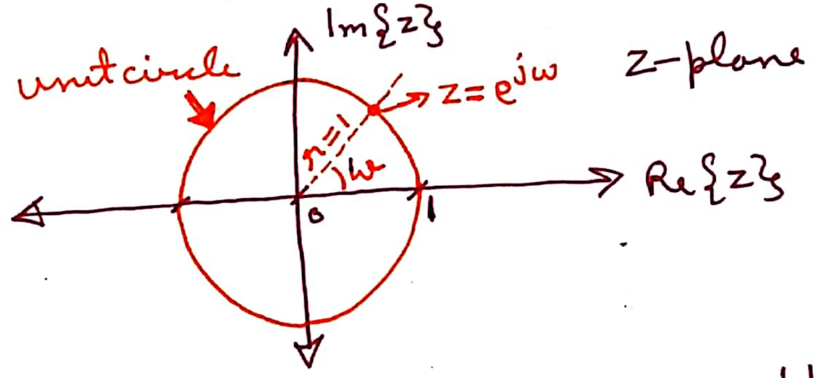
⇒ $X(re^{j\omega})$ is the discrete-time Fourier-transform (DTFT) of the sequence $x[n]$ multiplied by a real exponential r^{-n}; that is

$$X(re^{j\omega}) = \text{DTFT}\{x[n]\, r^{-n}\}$$

$$X[z]\Big|_{z=e^{j\omega}} = X(re^{j\omega})\Big|_{r=1} = \text{DTFT}\{x[n]\}$$

⇒ The Laplace-transform reduces to CTFT on the imaginary axis (i.e., for $s=j\omega$)

⇒ The Z-transform reduces to DTFT on the contour in the complex Z-plane corresponding to a circle with a radius of unity.



⇒ For convergence of z-transform, we require that DTFT of $x[n]r^{-n}$ converges.

⇒ Z-transform of a sequence has associated with it a range of values of z for which $X[z]$ converges, which is referred to as "region of convergence".

⇒ If ROC of z-transform includes the unit circle, then DTFT also converges. In some cases, z-transform exists, but DTFT doesn't.

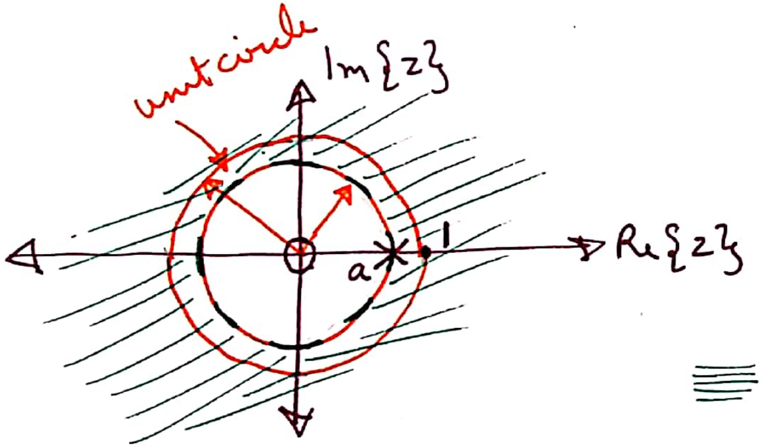
Example

Consider the signal $x[n] = a^n u[n]$

for $0 < a < 1$

$$X[z] = \sum_{n=\infty}^{\infty} a^n u[n]\, z^{-n} = \sum_{n=0}^{\infty} (a z^{-1})^n$$

$$= \frac{1}{1 - a z^{-1}} = \boxed{\frac{z}{z-a}} \quad ; \quad \underset{\text{ROC}}{|z| > |a|}$$



≡ ROC

one zero at $z = 0$
one pole at $z = a$ } Pole-zero plot and ROC

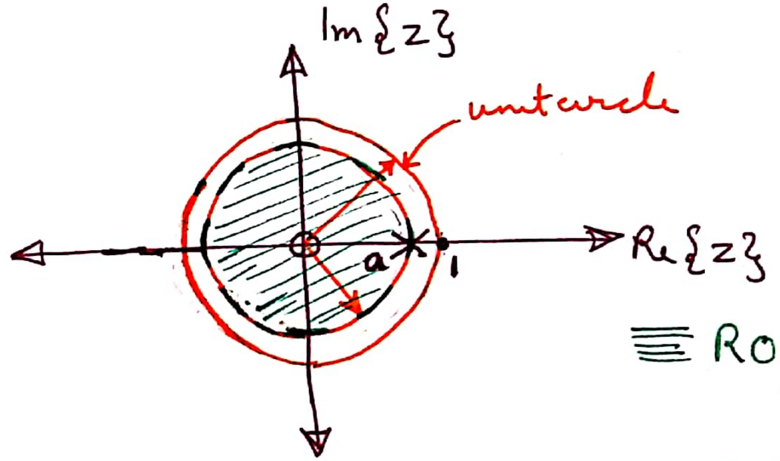
Consider the signal $x[n] = -a^n u[-n-1]$

for $0 < a < 1$

$$X[z] = \sum_{n=-\infty}^{\infty} - a^n u[-n-1]\, z^{-n}$$

$$= -\sum_{n=-\infty}^{-1} a^n z^{-n} = 1 - \sum_{n=0}^{\infty} (a^{-1} z)^n$$

$$= \boxed{\frac{z}{z-a}} \quad ; \quad \underset{\text{ROC}}{|z| < |a|}$$



≡ ROC

one zero at $z = 0$
one pole at $z = a$ } Pole-zero plot and ROC

" The z-transforms differ only in their regions of convergence "

⇒ If the Z-transform $X[z]$ of $x[n]$ is rational, and if $x[n]$ is right sided, then the ROC is the region in the Z-plane outside the outermost pole — i.e., outside the circle of radius equal to the largest magnitude of the poles of $X[z]$. Furthermore, if $x[n]$ is causal (i.e., if it is right sided and equal to 0 for $n<0$), then ROC also includes $z=\infty$.

⇒ If the Z-transform $X[z]$ of $x[n]$ is rational, and if $x[n]$ is left sided, then the ROC is the region in the Z-plane inside the innermost nonzero pole — i.e., inside the circle of radius equal to the smallest magnitude of the poles of $X[z]$ other than any at $z=0$ and extending inward to and possibly including $z=0$. Furthermore, if $x[n]$ is anticausal (i.e., if it is left sided and equal to 0 for $n>0$), then ROC also includes $z=0$.

⇒ LTI Systems characterized by Linear Constant-Coefficient Difference Equation

Consider an LTI system for which input $x[n]$ and output $y[n]$ satisfy a linear constant-coefficient difference-equation (Nth-order) of the form

$$\sum_{k=0}^{N} a_k\, y[n-k] = \sum_{k=0}^{M} b_k\, x[n-k]$$

Taking z-transform of both sides, and using the linearity as well as time-shifting properties, we obtain

$$\sum_{k=0}^{N} a_k z^{-k} Y[z] = \sum_{k=0}^{M} b_k z^{-k} X[z]$$

$$H[z] = \frac{Y[z]}{X[z]} = \frac{\sum_{k=0}^{M} b_k z^{-k}}{\sum_{k=0}^{N} a_k z^{-k}} \quad \text{(System function)}$$

⇒ The difference equation by itself does not provide information about which ROC to associate with the algebraic expression $H[z]$. Additional constraint, such as the causality or stability of system, however, serves to specify the region of convergence.

⇒ A discrete-time LTI system with rational system function $H[z]$ is causal if and only if: (a) ROC is the exterior of a circle outside the outermost pole, including infinity; and (b) with $H[z]$ expressed as a ratio of polynomials in z, the order of the numerator cannot be greater than the order of the denominator.

"Equivalently, the limit of $H[z]$ as $z \to \infty$ must be finite."

⇒ An LTI system is stable if and only if the ROC of its system function $H[z]$ includes the unit circle, $|z|=1$.

⇒ A causal LTI system with rational system function $H[z]$ is stable if and only if all the poles of $H[z]$ lie inside the unit circle – i.e., they must all have magnitude smaller than 1.

"Equivalently, the DTFT of $h[n]$ converges"

The Inverse Z-Transform →

$$h[n] = \frac{1}{2\pi j} \oint H[z]\, z^{n-1}\, dz$$

Where, the symbol $\oint$ denotes integration around a counterclockwise closed circular contour centered at the origin and with radius r. Any value of r such that the circular contour of integration $|z|=r$ is in the ROC. # One useful procedure relies on expressing z-transform as a linear combination of simpler terms, through the partial-fraction expansion of $H[z]$ as $H[z] = \sum_{i=1}^{m} A_i/(1-a_i z^{-1})$

And then, the inverse transform of each term can be obtained by inspection.

Introduction to Continuous-time Fourier transform {CTFT}

$\xrightarrow{\text{CTFT}}$ $$H[j\Omega] = \int_{-\infty}^{\infty} h(t)\, e^{-j\Omega t}\, dt$$ &

$\xrightarrow{\text{Inverse CTFT}}$ $$h(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} H[j\Omega]\, e^{j\Omega t}\, d\Omega$$

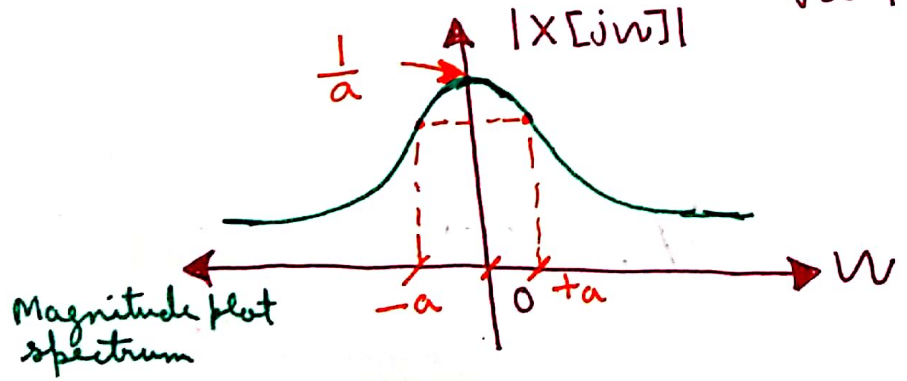
"For periodic and aperiodic signals"

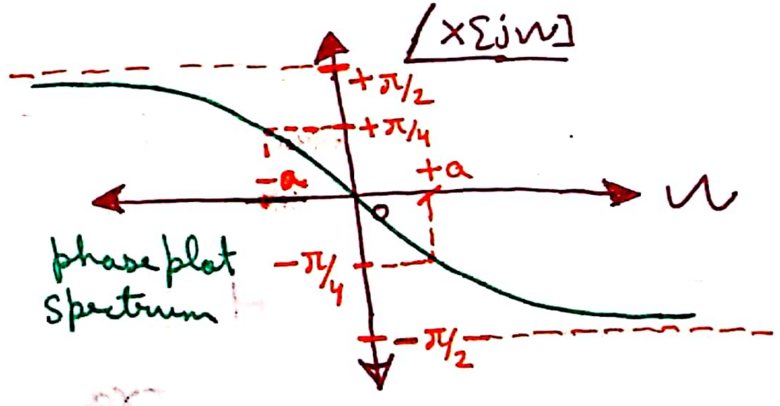
Example:-

Consider the signal $x(t) = e^{-at}u(t)$ $\quad a>0$

$$X[j\Omega] = \int_{0}^{\infty} e^{-at} e^{-j\Omega t}\, dt = \frac{1}{a+j\Omega}$$

We express CTFT $X[j\Omega]$ in terms of its magnitude as $|X[j\Omega]| = \frac{1}{\sqrt{a^2+\Omega^2}}$ and phase as $\angle X[j\Omega] = -\tan^{-1}\left(\frac{\Omega}{a}\right)$





— Introduction to Discrete-time Fourier-transform (DTFT) —

DTFT → Analysis Equation

$$H[e^{j\omega}] = \sum_{n=-\infty}^{+\infty} h[n]\, e^{-j\omega n}$$

Inverse DTFT → Synthesis Equation

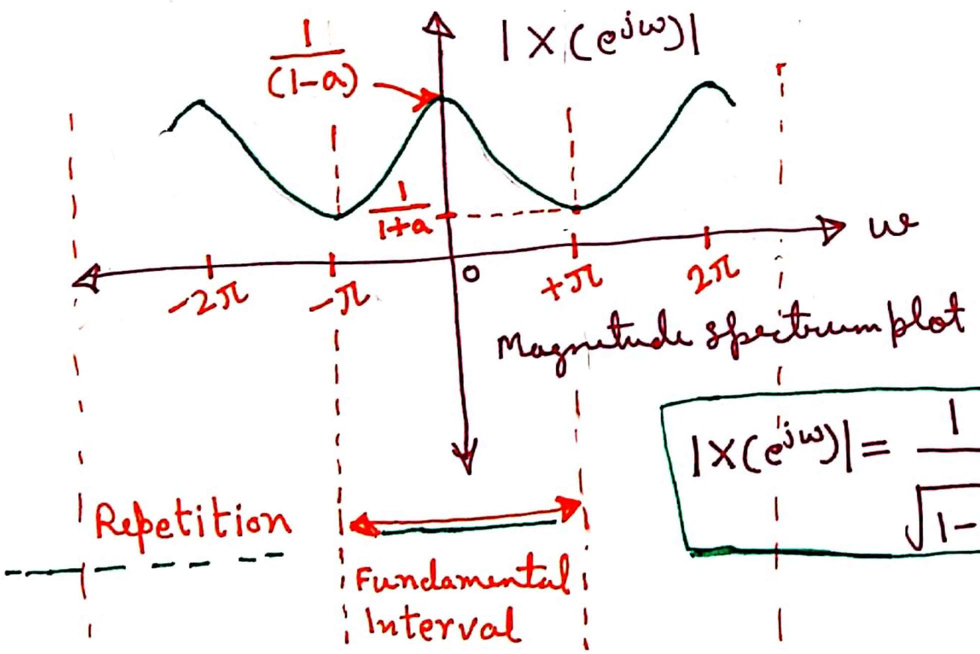
$$h[n] = \frac{1}{2\pi} \int_{2\pi} H[e^{j\omega}]\, e^{j\omega n}\, d\omega$$

" For periodic and aperiodic signals"

Example:-

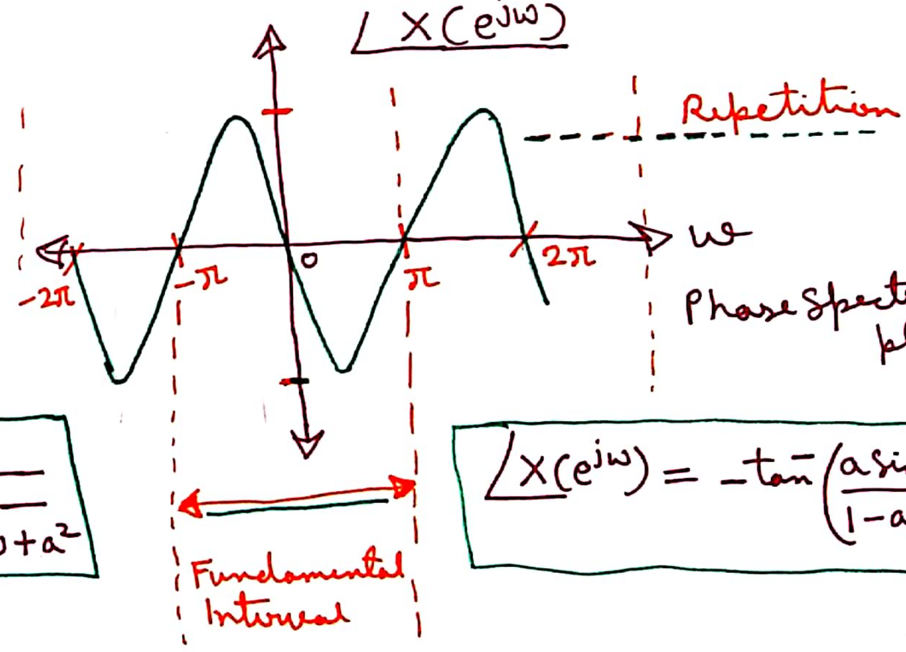
Consider the signal $x[n] = a^n u[n]$ for $0<a<1$

$$X[e^{j\omega}] = \sum_{n=-\infty}^{+\infty} x[n]\, e^{-j\omega n} = \sum_{n=0}^{\infty} a^n e^{-j\omega n} = \frac{1}{1-a\,e^{-j\omega}}$$



Magnitude spectrum plot

$$\boxed{|X(e^{j\omega})| = \frac{1}{\sqrt{1-2a\cos\omega + a^2}}}$$



Phase Spectrum plot

$$\boxed{\angle X(e^{j\omega}) = -\tan^{-1}\left(\frac{a\sin\omega}{1-a\cos\omega}\right)}$$

⇒ "$|X[e^{j\omega}]|$ is periodic with period 2π. Consequently, any interval of length 2π is sufficient for the specification of the spectrum." Usually we plot the spectrum in the fundamental interval $[-\pi, +\pi]$.

⇒ "We emphasize that all the spectral information contained in the fundamental interval is necessary for the complete description or characterization of the signal. For this reason, the range of integration in the synthesis equation $h[n]$ (as discussed above) is always 2π, independent of the specific characteristics of the signal with in the fundamental interval."

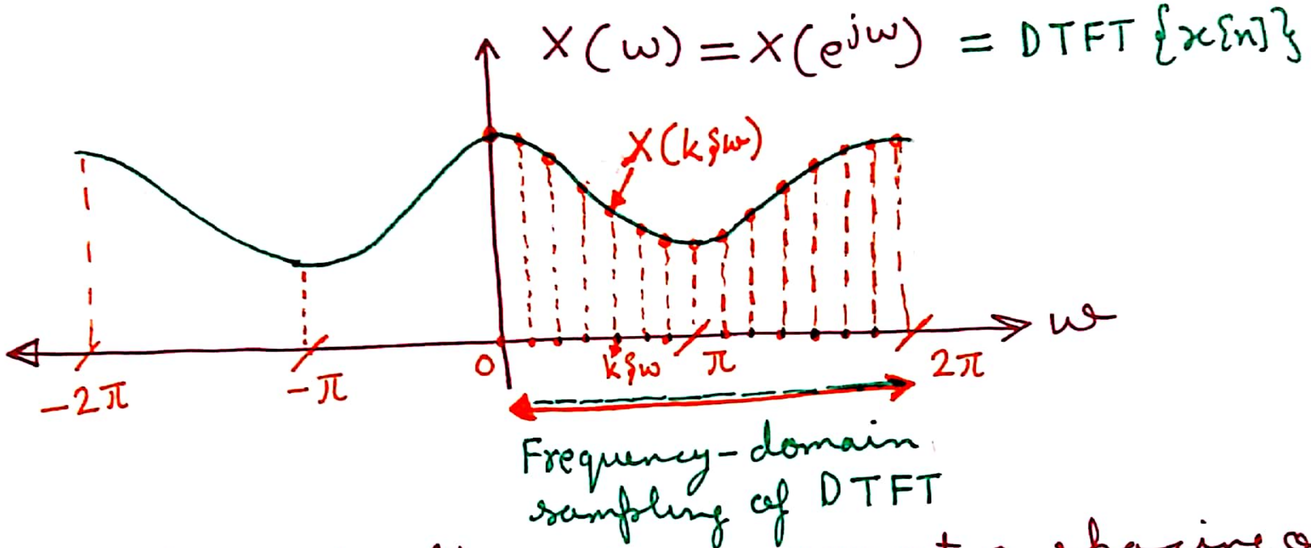
⇒ In discrete-time, a logarithmic frequency scale is not typically used, since the range of frequencies to be considered is always limited. For typical graphical representations of the magnitude and phase of a discrete-time frequency response, we may use

$$20\log_{10}|H(e^{j\omega})| \text{ versus } \omega \text{ plot}$$

$$\angle H(e^{j\omega}) \text{ versus } \omega \text{ plot}$$

Note:- For continuous-time systems, it is quite useful to use Bode plots.

— Introduction to Discrete Fourier Transform (DFT) —



We sample $X(\omega)$ periodically in frequency at a spacing of $\delta\omega$ radians between successive samples. As $X(\omega)$ is periodic with period 2π, only samples in the fundamental frequency range are necessary.
For convenience, we take "N" equidistant samples in the interval "$0 \leq \omega < 2\pi$" with spacing "$\delta\omega = 2\pi/N$" (as shown in above Figure).

Let a finite-duration sequence $x[n]$ of length L [i.e., $x[n] = 0$ for $n < 0$ and $n \geq L$] has DTFT

$$X(\omega) = \sum_{n=0}^{L-1} x(n)\, e^{-j\omega n} \qquad 0 \leq \omega < 2\pi$$

When we sample $X(\omega)$ at equally spaced frequencies $\omega_k = 2\pi k/N$, $k = 0, 1, \ldots, N-1$

where $\underbrace{N \geq L}$, the resultant samples are

$$X(\omega)\Big|_{\omega_k = \frac{2\pi k}{N}} = X[k] = \sum_{n=0}^{L-1} x[n]\, e^{-j2\pi kn/N}$$

For convenience, the upper index in the summation can be increased from $(L-1)$ to $(N-1)$, as $x[n] = 0$ for $n \geq L$. It follows that

$$\boxed{X[k] = \sum_{n=0}^{N-1} x[n]\, e^{-j2\pi kn/N} \; ; \; k = 0, 1, 2, ---, N-1} \longrightarrow (A)$$

$\equiv$ The frequency samples are obtained by evaluating DTFT $X(e^{j\omega})$ at a set of N (equally spaced) discrete frequencies, and the relation in equation (A) is called the "Discrete-Fourier-Transform"

$\Downarrow$

N-point DFT

We can view this computation as expanding the size of the sequence from L-points to N-points by appending (N-L) zeros to the sequence $x[n]$, that is, zero padding. The N-point DFT provides finer interpolation than the L-point DFT.

The sequence $x[n]$ can be recovered from $X[k]$ (frequency samples) by using the following relation

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

It is called the inverse DFT or IDFT
$\Downarrow$
N-point IDFT

<u>Example</u>:- A finite-duration sequence of length L is given as

$$x[n] = \begin{cases} 1, & 0 \leq n \leq L-1 \\ 0, & \text{otherwise} \end{cases}$$

Determine the N-point DFT of this sequence for $N \geq L$.

<u>Solution</u>:- The DTFT (discrete-time Fourier transform) of this sequence is

$$X(\omega) = \sum_{n=0}^{L-1} x[n]\, e^{-j\omega n} = \sum_{n=0}^{L-1} e^{-j\omega n} = \frac{\sin(\omega L/2)}{\sin(\omega/2)} \left[e^{-j\omega \frac{(L-1)}{2}} \right]$$

The N-point DFT of $x[n]$ is simply $X(\omega)$ evaluated at the set of N equally spaced frequencies $\omega_k = 2\pi k/N$, $k = 0, 1, --, N-1$

$$X[k] = \frac{1 - e^{-j2\pi kL/N}}{1 - e^{-j2\pi k/N}} \quad , \quad k = 0, 1, \ldots, N-1$$

$$= \frac{\sin(\pi kL/N)}{\sin(\pi k/N)} \exp\{-j\pi k(L-1)/N\}$$

If N is considered to be L i.e., $N=L$, then DFT appears to be

$$X[k] = \begin{cases} L, & k = 0 \\ 0, & k = 1, 2, \ldots, N-1 \end{cases}$$

There is only one nonzero value in the DFT.

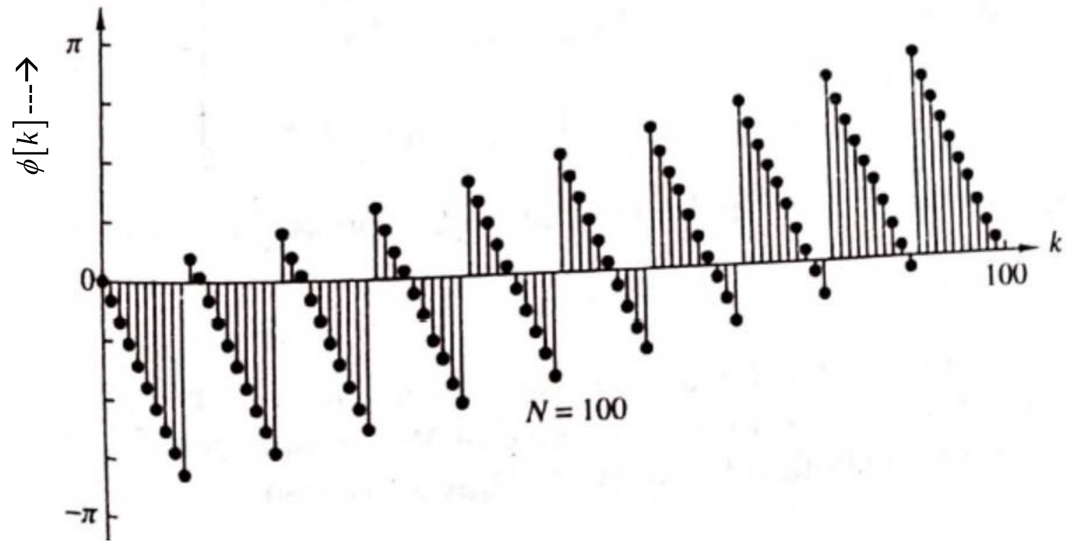
Therefore, the student must verify that $x[n]$ can be recovered from $X[k]$ by performing an N-point IDFT.

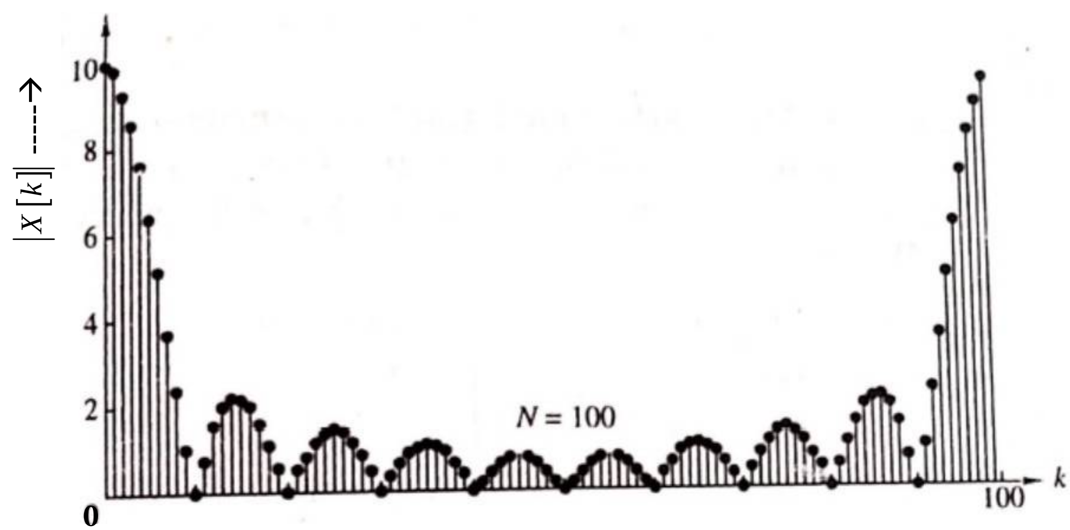
$\Rightarrow$ We must evaluate $X(\omega)$ at more closely spaced frequencies, i.e., $\omega_k = 2\pi k/N$, where $N > L$.

$$|X[k]| = \left[\frac{\sin(\pi kL/N)}{\sin(\pi k/N)}\right] \qquad \phi[k] = -\left[\frac{\pi k(L-1)}{N}\right]$$

— "Magnitude of N-point DFT of $X[k]$" — — "phase of N-point DFT of $X[k]$" —

L = 10





Reference Books ⇒

⇒ "Digital Signal Processing: Principles, Algorithms and Applications," by J. G. Proakis and D. G. Manolakis; Prentice-Hall, 2007.

⇒ "Theory and Application of Digital Signal Processing," by L.R. Rabiner and B. Gold; Prentice-Hall, 2003.

⇒ "Discrete Time Signal Processing," by A.V. Oppenheim and R.W. Schafer; Prentice-Hall, 2001.

⇒ "Digital Signal Processing," by S.K. Mitra; Tata McGraw-Hill, 2006.

⇒ "Digital Signal Processing," by S. Salivahanan; McGraw-Hill, 2019.

⇒ "Digital Filters: Analysis and Design," by A. Antoniou; Tata McGraw-Hill, 1979.

⇒ "Digital Signal Processing," by E. Ifeachor and B.W. Jervis; Pearson Education, 2007.

⇒ "Signals and Systems," by A.V. Oppenheim, A.S. Willsky and S. H. Nawab; Prentice-Hall of India, 1999. (Prerequisite)

Thanks for attending this session on DSP

DIGITAL SIGNAL PROCESSING
UEC – 502

From My Lecture Notes

Lecture No. – 04

for

Electronics and Communication Engineering Students

&

Electronics and Computer Engineering Students

Dr. Amit Kumar Kohli

Electronics and Communication Engineering Department

Thapar Institute of Engineering & Technology, Patiala, Punjab, India

"DFT as a linear transformation"

Using formulas for the DFT & IDFT, we can represent DFT/IDFT pair as

$$X[k] = \sum_{n=0}^{N-1} x(n) W_N^{kn} \qquad k = 0, \ldots, N-1 \qquad \text{DFT} — (A)$$

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X[k] W_N^{-kn} \qquad n = 0, \ldots, N-1 \qquad \text{IDFT} — (B)$$

where,

$$W_N = e^{-j2\pi/N} \Rightarrow \text{"Twiddle Factor"}$$

(an N^{th} root of unity)

⇒ Each point of DFT can be computed by
- → N complex multiplications
- → $(N-1)$ complex additions

⇒ N-point DFT values can be computed in
- → Total N^2 complex multiplications
- → Total $N(N-1)$ complex additions

"Twiddle factor is used to reduce the computational complexity of DFT and IDFT operation" — We can exploit the properties of twiddle factor to serve this purpose.

We can also view DFT and IDFT as linear transformation on the sequences $x[n]$ and $X[k]$ respectively.

First, we define an N-point vector $\vec{x}_N$ of the signal sequence $x(n)$, $n = 0, 1, ---, N-1$ as

$$\vec{x}_N = \begin{bmatrix} x(0) \\ x(1) \\ \vdots \\ x(N-2) \\ x(N-1) \end{bmatrix}_{N \times 1} \qquad \vec{X}_N = \begin{bmatrix} X[0] \\ X[1] \\ \vdots \\ X[N-2] \\ X[N-1] \end{bmatrix}_{N \times 1}$$

and an N-point vector $\vec{X}_N$ of frequency samples as an $N \times N$ matrix $\overrightarrow{W_N}$ as

$$\overrightarrow{W_N} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & W_N^1 & W_N^2 & \cdots & W_N^{N-1} \\ 1 & W_N^2 & W_N^4 & \cdots & W_N^{2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & W_N^{N-1} & W_N^{2(N-1)} & \cdots & W_N^{(N-1)(N-1)} \end{bmatrix} \Rightarrow$$

" $\overrightarrow{W_N}$ is the matrix of the linear transformation "

By using equation (A), we can express N-point DFT in matrix form as

$$\overrightarrow{X_N} = \overrightarrow{W_N}\, \overrightarrow{x_N} \quad \text{——— (C)}$$

If and only if the inverse of $\overrightarrow{W_N}$ exists, then by premultiplying both sides in above equation by $\overrightarrow{W_N}^{-1}$, we obtain

$$\overrightarrow{x_N} = \overrightarrow{W_N}^{-1}\, \overrightarrow{X_N} \quad \text{——— (D)}$$

{ Appearing to be IDFT }

However, from equation (B), we can express N-point IDFT in matrix form as

$$\overrightarrow{x_N} = \frac{1}{N}\, \overrightarrow{W_N}^{*}\, \overrightarrow{X_N} \quad \text{——— (E)}$$

where, $\vec{W}_N^*$ indicates the complex conjugate of the matrix $\vec{W}_N$

From equations (D) & (E), it is apparent that

$$\underbrace{\vec{W}_N^{-1} = \frac{1}{N}\vec{W}_N^*}$$

It results in

$$\vec{W}_N \times \vec{W}_N^* = N\,I_N$$

where, I_N is an $N \times N$ identity matrix, and its
and $\vec{W}_N$ is an orthogonal (unitary) matrix, and its inverse exists as $\vec{W}_N^{-1}$.

Example:- Compute DFT of the four-point sequence $x(n)$, such that

$$\vec{x} = \begin{bmatrix} -2 \\ -1 \\ +1 \\ +2 \end{bmatrix}_{4\times 1}$$

<u>Solution</u>:- First, we determine the matrix $\vec{W}_4$ (with $N=4$)

⇒ By periodicity property of $W_4^k \Longrightarrow W_4^{k+4} = \boxed{W_4^k = W_N^{k+N}}$

⇒ By symmetry property of $W_4^k \Longrightarrow W_4^{k+2} = \boxed{-W_4^k = +W_N^{k+N/2}}$

$$\vec{W}_4 = \begin{bmatrix} W_4^0 & W_4^0 & W_4^0 & W_4^0 \\ W_4^0 & W_4^1 & \boxed{W_4^2} & \boxed{W_4^3} \\ W_4^0 & \boxed{W_4^2} & W_4^4 & \boxed{W_4^6} \\ W_4^0 & \boxed{W_4^3} & \boxed{W_4^6} & W_4^9 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & \boxed{W_4^1} & W_4^2 & W_4^3 \\ 1 & W_4^2 & \boxed{W_4^0} & \boxed{W_4^2} \\ 1 & W_4^3 & \boxed{W_4^2} & \boxed{W_4^1} \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & +j \\ 1 & -1 & 1 & -1 \\ 1 & +j & -1 & -j \end{bmatrix}$$

$$\vec{X}_4 = \vec{W}_4 \vec{x}_4 = \begin{bmatrix} 0 \\ -3+3j \\ -2 \\ -3-3j \end{bmatrix}$$

However,

IDFT of $\vec{X}_4$ can be determined by conjugating the elements in $\vec{W}_4$ to obtain $\vec{W}_4^*$ as

$$\vec{W}_4^* = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & +j & -1 & -j \\ 1 & -1 & 1 & -1 \\ 1 & -j & -1 & +j \end{bmatrix} = 4 \times \vec{W}_4^{-1}$$

$$\vec{x}_4 = \vec{x}_N = \frac{1}{N} \vec{W}_N^* \vec{X}_N = \frac{1}{4} \vec{W}_4^* \vec{X}_4 = \begin{bmatrix} -2 \\ -1 \\ +1 \\ +2 \end{bmatrix}$$

Properties of the DFT $\longrightarrow$

Periodicity :- If $x(n)$ and $X[k]$ are an N-point DFT pair, then

$$x(n) = x(n+N) \quad \text{for all } n$$

$$X[k] = X[k+N] \quad \text{for all } k$$

Linearity :- If $x_1(n) \xleftrightarrow[N]{DFT} X_1[k]$

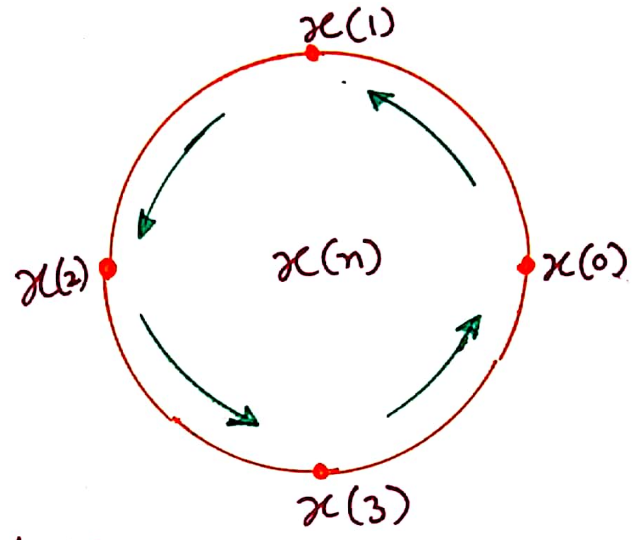
$x_2(n) \xleftrightarrow[N]{DFT} X_2[k]$

then

$$a_1 x_1(n) + a_2 x_2(n) \xleftrightarrow[N]{DFT} a_1 X_1[k] + a_2 X_2[k]$$

for any real-valued or complex-valued constants a_1 and a_2

Circular Symmetries of a sequence :- N-point DFT of a finite duration sequence $x(n)$ of length $L \leq N$ is equivalent to the N-point DFT of a periodic sequence $x_p(n)$ of period N. It gives us an opportunity to represent sequence $x(n)$ as

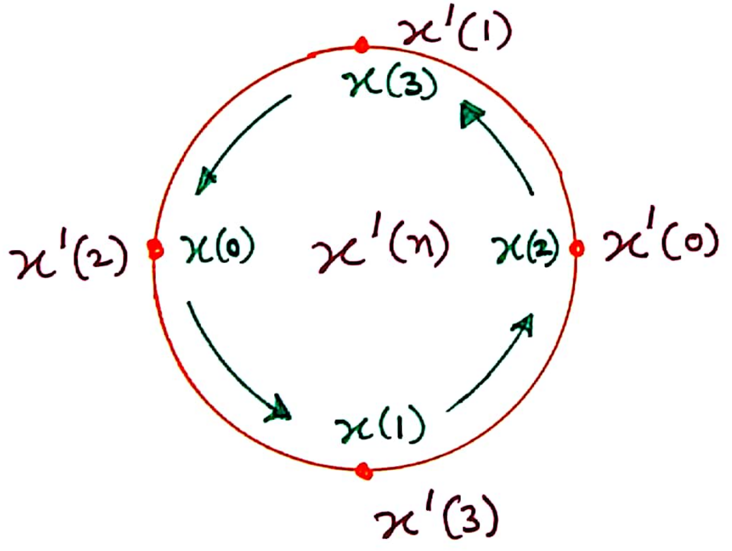


$$\vec{x}_4 = \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix}_{4\times 1} \quad \text{for } N=4$$

In general, the circular shift of the sequence can be represented as the index modulo N. It follows that

$$x'(n) = x(n-k, \text{modulo } N) = x((n-k))_N$$

If $k=2$ and $N=4$, then $x'(n) = x((n-2))_4$



$$\left\{\begin{aligned} x'(0) &= x((-2))_4 = x(2) \\ x'(1) &= x((-1))_4 = x(3) \\ x'(2) &= x((0))_4 = x(0) \\ x'(3) &= x((1))_4 = x(1) \end{aligned}\right\}$$

⇒ An N-point sequence is called circularly even, if it is symmetric about the point zero on the circle. It follows that

$$x(N-n) = x(n) \qquad 1 \leq n \leq N-1$$

⇒ An N-point sequence is called circularly odd, if it is asymmetric about the point zero on the circle; such that

$$x(N-n) = -x(n) \qquad 1 \leq n \leq N-1$$

⇒ The time reversal of an N-point sequence is attained by reversing its samples about the point zero on the circle

$$x((-n))_N = x(N-n) \qquad 0 \leq n \leq N-1$$

" This time reversal is equivalent to plotting $x(n)$ in a clockwise direction on a circle"

⇒ Now, if the sequence is complex valued, we have

Conjugate even :— $x(n) = +x^*(N-n)$

Conjugate odd :— $x(n) = -x^*(N-n)$

The above relationships suggest that we can decompose $x(n)$ as

$$x(n) = x_{ce}(n) + x_{co}(n)$$

where

$$x_{ce}(n) = \frac{1}{2}\left\{x(n) + x^*(N-n)\right]$$

$$x_{co}(n) = \frac{1}{2}\left[x(n) - x^*(N-n)\right]$$

⇒ Let us consider that N-point sequence $x(n)$ and its DFT are both complex valued. Then, these sequences can be represented as

$$x(n) = x_R(n) + j\,x_I(n) \qquad 0 \leq n \leq N-1$$

where, $R \rightarrow$ real part
$I \rightarrow$ imaginary part

$$X[k] = X_R[k] + j\,X_I[k] \qquad 0 \leq k \leq N-1$$

But,

$$X[k] = \sum_{n=0}^{N-1} x(n)\, e^{-j2\pi kn/N} = \sum_{n=0}^{N-1} \left[x_R(n) + j\,x_I(n)\right] e^{-j2\pi kn/N}$$

It results in

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

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

Similarly,

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

$$\left\{ \begin{array}{l} x(n) = \frac{1}{N} \sum_{k=0}^{N-1} \left(X_R[k] + j X_I[k] \right) e^{j2\pi kn/N} \\ \\ x(n) = x_R(n) + j\, x_I(n) \end{array} \right.$$

It results in

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

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

⇒ Some important results are as follows. ⇒

→ $x^*(n) \xleftrightarrow[\text{N-point}]{\text{DFT}} X^*[N-k]$

→ $x^*(N-n) \longleftrightarrow X^*[k]$

→ $x_R(n) \longleftrightarrow X_{ce} = \frac{1}{2}\left[X[k] + X^*[N-k]\right]$

→ $j\, x_I(n) \longleftrightarrow X_{co} = \frac{1}{2}\left[X[k] - X^*[N-k]\right]$

$$x_{ce}(n) = \frac{1}{2}\left[x(n) + x^*(N-n)\right] \xleftrightarrow[\text{N-point}]{\text{DFT}} X_R[k]$$

$$x_{co}(n) = \frac{1}{2}\left[x(n) - x^*(N-n)\right] \longleftrightarrow jX_I[k]$$

Multiplication of Two DFTs and Concept of Circular Convolution ⇒

Let us consider two finite-duration sequences of length N i.e.,s $x_1(n)$ and $x_2(n)$.

$$x_1(n) \xleftrightarrow{\text{N-point DFT}} X_1[k] \qquad x_2(n) \xleftrightarrow{\text{N-point DFT}} X_2[k]$$

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

$$X_2[k] = \sum_{l=0}^{N-1} x_2(l)\, e^{-j2\pi lk/N}$$

$k = 0, 1, \ldots, N-1$ & $l ; n$ **(time-domain index)**

Now through multiplication process $X_3[k] = X_1[k] \times X_2[k]$ → multiplication operator

$$X_3[k] \xleftrightarrow{\text{N-point IDFT}} x_3(m) = ?$$

$m = 0, 1, \ldots, N-1$

Scanned with CamScanner

Dr. A.K. Kohli, ECED, TIET

Taking IDFT of $X_3[k]$, we find that

$$x_3(m) = \frac{1}{N}\sum_{k=0}^{N-1} X_3[k]\, e^{j2\pi km/N} = \frac{1}{N}\sum_{k=0}^{N-1} X_1[k]\, X_2[k]\, e^{j2\pi km/N}$$

$m = 0, 1, 2 - - N-1$

Substituting the $X_1[k]$ & $X_2[k]$ in above equation,

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

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

$$\sum_{k=0}^{N-1} a^k = \begin{cases} N & \text{for } a = 1 \\ \dfrac{1-a^N}{1-a} & \text{for } a \neq 1 \end{cases}$$

where, $a = e^{j2\pi(m-n-l)/N}$

$a = 1$ for $(m-n-l)$ is a multiple of N

It leads to

$$\sum_{k=0}^{N-1} a^k = \begin{cases} N & \text{for } l = m-n+pN = ((m-n))_N \text{ ; } p \text{ is an integer} \\ 0 & \text{otherwise} \end{cases}$$

Consequently, IDFT of $X_3[k]$ (Frequency-domain) $\longleftrightarrow x_3(m) = ?$ (Time-domain)

$$\Downarrow$$

$$x_3(m) = \frac{1}{N}\sum_{n=0}^{N-1} x_1(n)\, x_2(m-n+pN) \times N$$

multiplication operator

$$= \underbrace{\sum_{n=0}^{N-1} x_1(n)\, x_2\left((m-n)\right)_N}$$

⇒ Expression has the form of a convolution sum.

⇒ But, it is not the traditional linear convolution sum.

⇒ This convolution sum formula involving the index $((m-n))_N$ is referred to as the "Circular Convolution in discrete-time domain"

It may be inferred that multiplication of the DFTs of two sequences is equivalent to the circular convolution of these two sequences in the time-domain.

Thanks for attending this session on DSP

Reference Books ⇒

⇒ "Digital Signal Processing: Principles, Algorithms and Applications," by J. G. Proakis and D. G. Manolakis; Prentice-Hall, 2007.

⇒ "Theory and Application of Digital Signal Processing," by L.R. Rabiner and B. Gold; Prentice-Hall, 2003.

⇒ "Discrete Time Signal Processing," by A.V. Oppenheim and R.W. Schafer; Prentice-Hall, 2001.

⇒ "Digital Signal Processing," by S.K. Mitra; Tata McGraw-Hill, 2006.

⇒ "Digital Signal Processing," by S. Salivahanan; McGraw-Hill, 2019.

⇒ "Digital Filters: Analysis and Design," by A. Antoniou; Tata McGraw-Hill, 1979.

⇒ "Digital Signal Processing," by E. Ifeachor and B.W. Jervis; Pearson Education, 2007.

⇒ "Signals and Systems," by A.V. Oppenheim, A.S. Willsky and S. H. Nawab; Prentice-Hall of India, 1999. (Prerequisite)

Thanks for attending this session on DSP

DIGITAL SIGNAL PROCESSING
UEC – 502

From My Lecture Notes

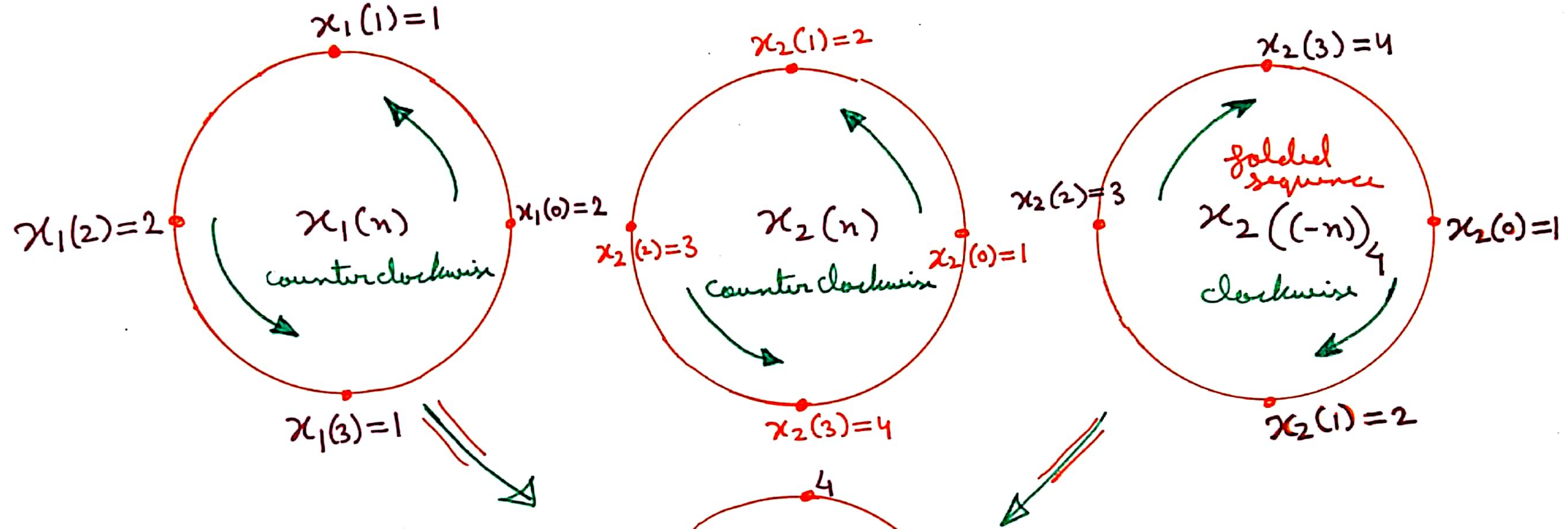
Lecture No. – 05

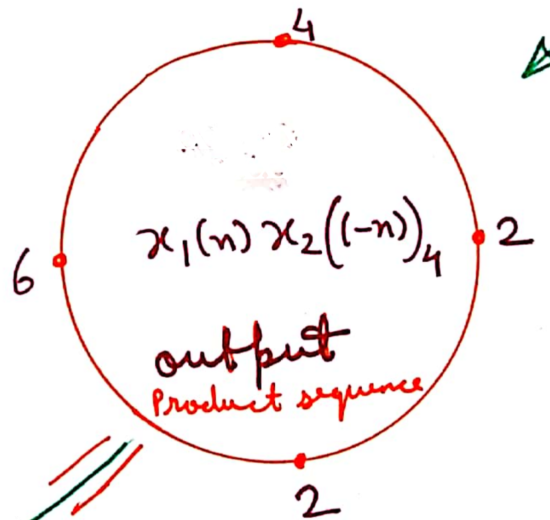
for
Electronics and Communication Engineering Students
&
Electronics and Computer Engineering Students

Dr. Amit Kumar Kohli
Electronics and Communication Engineering Department
Thapar Institute of Engineering & Technology, Patiala, Punjab, India

Perform the circular convolution of the following two sequences

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

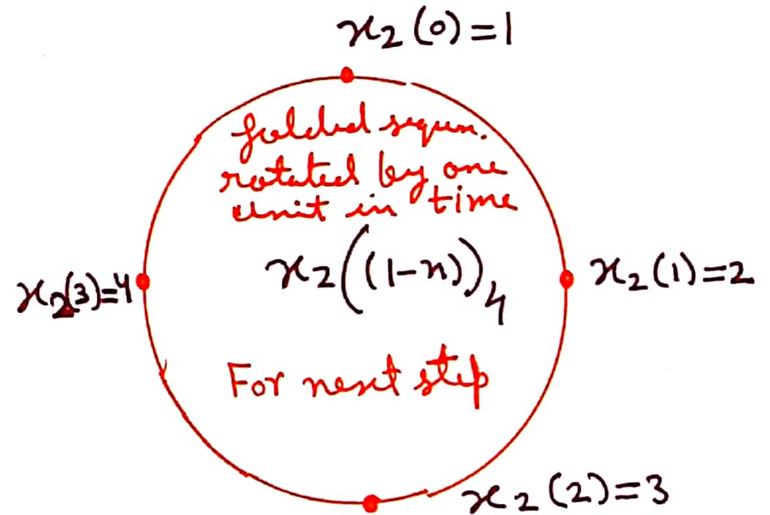




$$x_3(m) = \sum_{n=0}^{N-1} x_1(n)\, x_2((m-n))_N$$

$m = 0, 1, ---, N-1$

$$x_3(0) = \sum_{n=0}^{N-1} x_1(n)\, x_2((-n))_4 = 2+4+6+2$$
$$= 14$$



Similarly, we can calculate $x_3(m)$ from graphed sequences as

$$x_3(1) = \sum_{n=0}^{3} x_1(n)\, x_2\big((1-n)\big)_4 = 16$$

$$x_3(2) = \sum_{n=0}^{3} x_1(n)\, x_2\big((2-n)\big)_4 = 14$$

$$x_3(3) = \sum_{n=0}^{3} x_1(n)\, x_2\big((3-n)\big)_4 = 16$$

$$x_3(m) = \big[\underset{\uparrow}{14}, 16, 14, 16\big] \qquad m = 0, 1, 2, 3$$

In circular convolution, the folding and shifting (rotating) operations are performed in a circular fashion by computing the index of one of the sequences by using modulo N operation.

Circular Convolution is expressed as

$$x_1(n) \; \underset{\text{circular convolution operator}}{\textcircled{N}} \; x_2(n) \xleftrightarrow[N]{DFT} X_1[k]\, X_2[k]$$

An alternative method to solve circular convolution is by means of DFT & IDFT (as shown below)

$$x_1(n) \xleftrightarrow{DFT} X_1[k] \quad \text{and} \quad x_2(n) \xleftrightarrow{DFT} X_2[k]$$

$$X_3[k] = X_1[k]\, X_2[k]$$

$$\text{or} \quad X_3[k] \xleftrightarrow{IDFT} x_3(n) = x_1(n) \; Ⓝ \; x_2(n)$$

Therefore,

$$X_1[k] = \sum_{n=0}^{3} x_1(n)\, e^{-j2\pi kn/4} = 2 + e^{-j\pi k/2} + 2e^{-j\pi k} + e^{-j3\pi k/2}$$

$$\Rightarrow X_1[0] = 6 \; ; \; X_1[1] = 0 \; ; \; X_1[2] = 2 \; ; \; X_1[3] = 0$$

$$X_2[k] = \sum_{n=0}^{3} x_2(n)\, e^{-j2\pi kn/4} = 1 + 2e^{-j\pi k/2} + 3e^{-j\pi k} + 4e^{-j3\pi k/2}$$

$$\Rightarrow X_2[0] = 10 \; ; \; X_2[1] = -2+2j \; ; \; X_2[2] = -2 \; ; \; X_2[3] = -2-2j$$

As $X_3[k] = X_1[k]\, X_2[k]$

$$\Rightarrow X_3[0] = 60 \; ; \; X_3[1] = 0 \; ; \; X_3[2] = -4 \; ; \; X_3[3] = 0$$

$$\Rightarrow x_3(n) = \frac{1}{4}\sum_{k=0}^{3} X_3[k]\, e^{j2\pi nk/4} = \frac{1}{4}\left[60 - 4e^{+j\pi n}\right]$$

Ans $\boxed{x_3(0) = 14 \; ; \; x_3(1) = 16 \; ; \; x_3(2) = 14 \; ; \; x_3(3) = 16}$

Time reversal of a sequence $\longrightarrow$

$$x(n) \xleftrightarrow[N]{DFT} X[k] \qquad x((-n))_N = x(N-n) \xleftrightarrow[n]{DFT} ?$$

$$\Rightarrow DFT\{x(N-n)\} = \sum_{n=0}^{N-1} x(N-n)\, e^{-j2\pi kn/N}$$

let $m = N-n$

$$= \sum_{m=1}^{N} x(m)\, e^{-j2\pi k(N-m)/N}$$

$$= \sum_{m=1}^{N} x(m)\, e^{+j2\pi km/N} = \sum_{m=0}^{N-1} x(m)\, e^{j2\pi km/N}$$

$$= \sum_{m=0}^{N-1} x(m)\, e^{-\frac{j2\pi m}{N}(N-k)} = X(N-k)$$

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

Reversing the N-point sequence in time is equivalent to reversing the DFT values.

Circular time shift of sequence $\longrightarrow$

$$x((n-l))_N \xleftrightarrow[N]{DFT} ? = DFT\left[x((n-l))_N\right] = \sum_{n=0}^{N-1} x((n-l))_N\, e^{-j2\pi kn/N}$$

$$\mathrm{DFT}\{x((n-l))_N\} = \underbrace{\sum_{n=0}^{l-1} x((n-l))_N \, e^{-j2\pi kn/N}}_{} + \underbrace{\sum_{n=l}^{N-1} x(n-l)\, e^{-j2\pi kn/N}}_{}$$

$$\Rightarrow \sum_{n=0}^{l-1} x((n-l))_N e^{-j2\pi kn/N} = \sum_{n=0}^{l-1} x(\underbrace{N-l+n}_{\downarrow m})\, e^{-j2\pi kn/N} = \sum_{m=N-l}^{N-1} x(m)\, e^{-j2\pi k(m+l)/N}$$

$$\Rightarrow \sum_{n=l}^{N-1} x(\underbrace{n-l}_{\downarrow m})\, e^{-j2\pi kn/N} = \sum_{m=0}^{N-1-l} x(m)\, e^{-j2\pi k(m+l)/N}$$

Now, $$\mathrm{DFT}\{x((n-l))_N\} = \sum_{m=N-l}^{N-1} x(m)\, e^{-j2\pi k(m+l)/N} + \sum_{m=0}^{N-1-l} x(m)\, e^{-j2\pi k\frac{(m+l)}{N}}$$

$$= \left[\sum_{m=0}^{N-1} x(m)\, e^{-j2\pi km/N}\right] e^{-j2\pi \frac{kl}{N}}$$

$$= X[k]\, e^{-j2\pi kl/N}$$

Circular frequency shift $\longrightarrow$

$$x(n)\, e^{j2\pi ln/N} \overset{\mathrm{DFT}}{\longleftrightarrow} ? = \sum_{n=0}^{N-1} x(n)\, e^{j2\pi ln/N}\, e^{-j2\pi kn/N}$$

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

$$= X[((k-l))_N]$$

"Multiplication of $x(n)$ with the complex exponential sequence $e^{j2\pi l n/N}$ is equivalent to the circular shift of the DFT by l units in frequency"

Complex-conjugate properties →

$$x^*(n) \xleftrightarrow{DFT} ? = \sum_{n=0}^{N-1} x^*(n)\, e^{-j2\pi kn/N} = \left[\sum_{n=0}^{N-1} x(n)\, e^{+j2\pi kn/N}\, e^{-j2\pi Nn/N}\right]^*$$

$$= \left[\sum_{n=0}^{N-1} x(n)\, e^{-j2\pi n(N-k)}\right]^* = X^*(N-k)$$

$$x^*((-n))_N = x^*(N-n) \xleftrightarrow{DFT} ? = X^*[k]$$

$$\boxed{\begin{aligned} \text{IDFT of } X^*[k] &= \frac{1}{N}\sum_{k=0}^{N-1} X^*[k]\, e^{j2\pi kn/N} = \left[\frac{1}{N}\sum_{k=0}^{N-1} X[k]\, e^{-j2\pi kn/N}\, e^{+j2\pi \frac{kN}{N}}\right]^* \\ &= \left[\frac{1}{N}\sum_{k=0}^{N-1} X[k]\, e^{j2\pi k(N-n)/N}\right]^* = x^*(N-n) \end{aligned}}$$

<u>Circular Correlation</u> $\longrightarrow$ $x(n) \xleftrightarrow{DFT} X[k]$ & $y(n) \xleftrightarrow{DFT} Y[k]$

$\hat{r}_{xy}(l) \xleftrightarrow{DFT} \hat{R}_{xy}[k] = ?$ $\qquad k = 0, 1, \ldots, N-1$

$\hat{r}_{xy}(l)$ is defined as the unnormalized circular correlation, such that

$$\hat{r}_{xy}(l) = \sum_{n=0}^{N-1} x(n)\, y^*((n-l))_N$$

$$= x(l) \, (N) \, y^*(-l) \qquad \text{(circular convolution sum)}$$

Therefore, $\hat{r}_{xy}(l) \xleftrightarrow{DFT} DFT\{x(l)\} \times DFT\{y^*(-l)\}$

$$= X[k]\, Y^*[k] = \hat{R}_{xy}[k]$$

<u>Parseval's theorem</u> $\longrightarrow$ For complex valued sequences $x(n)$ and $y(n)$,

(A) $\Leftarrow$ $\boxed{\hat{r}_{xy}(0) = \sum_{n=0}^{N-1} x(n)\, y^*(n)}$ And $\hat{r}_{xy}(l) = \frac{1}{N} \sum_{k=0}^{N-1} \hat{R}_{xy}(k)\, e^{j2\pi kl/N}$

$$= \frac{1}{N} \sum_{k=0}^{N-1} X[k]\, Y^*[k]\, e^{j2\pi kl/N}$$

(B) $\Leftarrow$ $\boxed{\hat{r}_{xy}(l)\Big|_{l=0} = \frac{1}{N} \sum_{k=0}^{N-1} X[k]\, Y^*[k]}$

From equation (A) and (B), it is apparent that

$$\sum_{n=0}^{N-1} x(n)\,y^*(n) = \frac{1}{N}\sum_{k=0}^{N-1} X[k]\,Y^*[k]$$ (General form of Parseval's theorem)

In special case, when $x(n) = y(n)$, then

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

It expresses the energy in the finite-duration sequence $x(n)$ in terms of the frequency components $|X[k]|$

<u>Multiplication of two sequences</u> $\rightarrow$ $x_1(n) \overset{DFT}{\underset{N}{\longleftrightarrow}} X_1[k]$ and $x_2(n) \overset{DFT}{\underset{N}{\longleftrightarrow}} X_2[k]$

$$x_1(n)\,x_2(n) \overset{DFT}{\underset{N}{\longleftrightarrow}} ? = X_3[p] = \sum_{n=0}^{N-1} x_3[n]\,e^{-j2\pi pn/N}$$

$= x_3(n)$

$$= \sum_{n=0}^{N-1} x_1(n)\,x_2(n)\,e^{-j2\pi pn/N}$$

$p = 0, 1, --, N-1$

$$X_3[p] = \sum_{n=0}^{N-1}\left[\frac{1}{N}\sum_{k=0}^{N-1} X_1[k]e^{j2\pi kn/N}\right]\left[\frac{1}{N}\sum_{l=0}^{N-1} X_2[l]\, e^{\frac{j2\pi l n}{N}}\right] e^{-j2\pi pn/N}$$

$$= \sum_{k=0}^{N-1}\frac{X_1[k]}{N}\sum_{l=0}^{N-1}\frac{X_2[l]}{N}\underbrace{\left[\sum_{n=0}^{N-1} e^{-j2\pi n(p-k-l)/N}\right]}$$

$$\Downarrow$$

$$\begin{cases} N & \text{for } l = p-k+qN \text{ with integer } q \\ 0 & \text{otherwise} \end{cases}$$

$$= \frac{1}{N}\sum_{k=0}^{N-1} X_1[k]\, X_2[p-k+qN]$$

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

$$\boxed{X_3[p] = \frac{1}{N}\, X_1[p] \; N \; X_2[p]} \quad p = 0, 1, 2, ---, N-1$$

Multiplication of two sequences in time-domain is equivalent to the circular convolution of their DFTs in the frequency-domain.

Thanks for attending this session on DSP

Reference Books ⇒

⇒ "Digital Signal Processing : Principles, Algorithms and Applications," by J. G. Proakis and D. G. Manolakis ; Prentice-Hall, 2007.

⇒ " Theory and Application of Digital Signal Processing," by L. R. Rabiner and B. Gold; Prentice-Hall, 2003.

⇒ " Discrete Time Signal Processing," by A.V. Oppenheim and R.W. Schafer; Prentice-Hall, 2001.

⇒ " Digital Signal Processing," by S. K. Mitra; Tata McGraw-Hill, 2006.

⇒ " Digital Signal Processing," by S. Salivahanan; McGraw-Hill, 2019.

⇒ " Digital Filters: Analysis and Design," by A. Antoniou ; Tata McGraw-Hill, 1979.

⇒ " Digital Signal Processing," by E. Ifeachor and B.W. Jervis; Pearson Education, 2007.

⇒ " Signals and Systems," by A.V. Oppenheim, A.S. Willsky and S. H. Nawab; Prentice-Hall of India, 1999. (Prerequisite)

Thanks for attending this session on DSP

DIGITAL SIGNAL PROCESSING
UEC – 502

From My Lecture Notes

Lecture No. – 06

for

Electronics and Communication Engineering Students

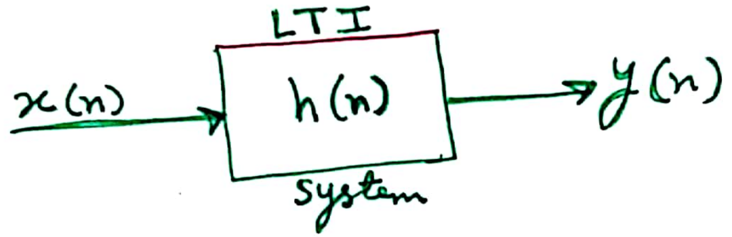
&

Electronics and Computer Engineering Students

Dr. Amit Kumar Kohli
Electronics and Communication Engineering Department
Thapar Institute of Engineering & Technology, Patiala, Punjab, India

Use of the DFT in linear filtering $\longrightarrow$

"Circular convolution" is quite different from "Linear Convolution"



System

$\boxed{\text{FIR Filter}}$

Let input to an LTI system be $x(n)$ of length L (in discrete-time domain)
the impulse response of this system be $h(n)$ of length M

where,
$$x(n) = 0 \quad \text{for } n<0 \text{ and } n \geq L \quad \text{(input signal sequence)}$$
$$h(n) = 0 \quad \text{for } n<0 \text{ and } n \geq M$$

The output signal sequence $y(n)$ can be expressed in the time-domain by using the linear convolution sum formula as

"When $L \geq M$"

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

$$n = 0, 1, 2, ----, L+M-2$$

"Duration of $y(n)$ is $L+M-1$"

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

"Linear convolution operator $*$"

Equivalently in frequency-domain

$$\underset{\text{DTFT}}{Y[e^{jw}]} = \underset{\text{DTFT}}{H[e^{jw}]} \times \underset{\text{DTFT}}{X[e^{jw}]}$$

"If the sequence $y(n)$ is to be expressed uniquely in frequency-domain by samples of its spectrum $Y[e^{jw}]$ at a set of discrete frequencies, then the number of distinct samples must be equal to or must exceed $L+M-1$."

Therefore, a DFT of size $N \geq L+M-1$ is required to represent $\{y(n)\}$ in the frequency-domain.

It follows that

$$Y[e^{jw}]\Big|_{w=\frac{2k\pi}{N}} = Y[w]\Big|_{\frac{2\pi k}{N}=w} = Y[k] = X[k]\,H[k] \qquad k = 0, 1, --, N-1$$

$$\begin{cases} \vec{x} = \{x_0 \;\; x_1 \;\ldots\ldots\; x_{L-1}\} \\ \vec{h} = \{h_0 \;\; h_1 \ldots h_{M-1}\} \end{cases}$$

$$\Rightarrow \vec{x} = \{\underbrace{x_0 \;\; x_1 \ldots x_{L-1}}_{L} \;\underbrace{0\;0\;0 \ldots 0}_{(M-1)}\}_{1\times N}$$

Zero-padding

$$\Rightarrow \vec{h} = \{\underbrace{h_0 \;\; h_1 \ldots h_{M-1}}_{M} \;\underbrace{0\;0\;0 \ldots\ldots 0}_{(L-1)}\}_{1\times N}$$

The increase in the size of the sequences does not alter their (continuous) spectra $X(\omega)$ and $H(\omega)$, as the sequences are aperiodic.

As $N >> L \geq M$

$$Y[k] = X[k]\,H[k] \xleftrightarrow[N\ (=L+M-1)]{\text{IDFT}} y(n)$$

"It implies that the $N(=L+M-1)$-point circular convolution of $x(n)$ and $h(n)$ is equivalent to the linear convolution of $x(n)$ and $h(n)$, when length of the sequences $x(n)$ and $h(n)$ is increased to N point by following the zero-padding procedure."

Example → Determine the response of an LTI system (FIR filter) with impulse response $h(n)$ to the input sequence $x(n)$; where

$$h(n) = \{\underset{\uparrow}{1}\ \ 2\ \ 3\} \Rightarrow M=3 \quad \text{and} \quad x(n) = \{\underset{\uparrow}{1}, 2, 2, 1\} \Rightarrow L=4$$

We can clearly show that the linear convolution of $x(n)$ and $h(n)$ produces a sequence of length $N = L+M-1 = 6$.

⇒ However, it is preferred that the length N is a power of 2.

"Therefore, choose $N = 8$ for DFT & IDFT operations"

CASE-1

$$X[k] = \sum_{n=0}^{7} x(n)\, e^{-j2\pi kn/8} = 1 + 2\,e^{-j2\pi k/8} + 2\,e^{-j4\pi k/8} + 1\,e^{-j6\pi k/8}$$

$$= 1 + 2\,e^{-j\pi k/4} + 2\,e^{-j\pi k/2} + 1\,e^{-j3\pi k/8}$$

$$H[k] = \sum_{n=0}^{7} h(n)\, e^{-j2\pi kn/8} = 1 + 2\,e^{-j1\pi k/4} + 3\,e^{-j\pi k/2}$$

for $k = 0, 1, \ldots, 7$

$X[k] \Rightarrow X[0] = 6$; $X[1] = \frac{2+\sqrt{2}}{2} - j\left(\frac{4+3\sqrt{2}}{2}\right)$; $X[2] = -1-j$; $X[3] = \frac{2-\sqrt{2}}{2} + j\left(\frac{4-3\sqrt{2}}{2}\right)$

$X[4] = 0$; $X[5] = \frac{2-\sqrt{2}}{2} - j\left(\frac{4-3\sqrt{2}}{2}\right)$; $X[6] = -1+j$; $X[7] = \frac{2+\sqrt{2}}{2} + j\left(\frac{4+3\sqrt{2}}{2}\right)$

$H[k] \Rightarrow H[0] = 6$; $H[1] = (1+\sqrt{2}) - j(3+\sqrt{2})$; $H[2] = -2-j2$; $H[3] = (1-\sqrt{2}) + j(3-\sqrt{2})$

$H[4] = 2$; $H[5] = (1-\sqrt{2}) - j(3-\sqrt{2})$; $H[6] = -2+j2$; $H[7] = (1+\sqrt{2}) + j(3+\sqrt{2})$

$Y_8[k] \Rightarrow Y_8[0] = 36$; $Y_8[1] = -14.071 - j17.485$; $Y_8[2] = +j4$; $Y_8[3] = 0.0711 + j0.5147$

$Y_8[4] = 0$; $Y_8[5] = 0.0711 - j0.5147$; $Y_8[6] = -4j$; $Y_8[7] = -14.071 + j17.485$

The corresponding eight-point IDFT is calculated as

$$y_8(n) = \sum_{k=0}^{7} \frac{Y_8[k]}{8} e^{j2\pi kn/8} \quad \text{for } n = 0, 1, ---, 7$$

$$\Downarrow$$

$$y_8(n) = \{\underset{\uparrow}{1}, 4, 9, 11, 8, 3, \underbrace{0, 0}\} \quad \text{in discrete-time domain}$$

We shall now try to determine the sequence $y(n)$ with $N = 4$ $\Longrightarrow$ $N=\text{Max}\{L ; M\}$

Conventional Circular Convolution Approach

CASE-2

$$X[k] = 1 + 2e^{-j2\pi k/4} + 2e^{-j2\pi k2/4} + 1e^{-j2\pi k3/4} = 1 + 2e^{-j\pi k/2} + 2e^{-j\pi k} + e^{-j\pi k3/2}$$

$\Rightarrow X[0] = 6$; $X[1] = -1-j$; $X[2] = 0$; $X[3] = -1+j$ for $k = 0, 1, 2, 3$

$$H[k] = 1 + 2e^{-j2\pi k/4} + 3e^{-j2\pi k2/4} = 1 + 2e^{-j\pi k/2} + 3e^{-j\pi k}$$

$\Rightarrow H[0] = 6$; $H[1] = -2-2j$; $H[2] = 2$; $H[3] = -2+2j$ for $k = 0, 1, 2, 3$

$$\Downarrow$$

$Y_4[0] = 36$; $Y_4[1] = +j4$; $Y_4[2] = 0$; $Y_4[3] = -j4$

The corresponding four-point IDFT is calculated as

$$y_4(n) = \sum_{k=0}^{3} \frac{Y_4[k]}{4} e^{j2\pi kn/4} \qquad n = 0,1,2,3$$

$$= \frac{1}{4}\left(36 + 4j\, e^{j\pi n/2} - 4j\, e^{j3\pi n/2}\right)$$

$$y_4(n) = \{\underset{\uparrow}{9}, 7, 9, 11\}$$ in discrete-time domain

We now compare the result $y_8(n)$ obtained by using eight-point DFTs with the sequence $y_4(n)$ obtained by using four-point DFTs (as discussed above)

Case1 output $\Rightarrow$ $$y_8(n) = \{\underbrace{\underset{\uparrow}{1}, 4, 9, 11, 8, 3}_{\leftarrow L+M-1=6 \text{ terms} \rightarrow}, \underbrace{0, 0}\}$$ "due to zero-padding"

$$\Downarrow \qquad\qquad \Downarrow$$

$$y_6(n) = \{\underbrace{\underset{\uparrow}{1}, 4, 9, 11}_{L=4}, \underbrace{8, 3}_{M-1=2}\} \qquad n = 0,1,2,3,4,5.$$

Case2 output $\Rightarrow$ $$y_4(n) = \{\underset{\uparrow}{9}, 7, 9, 11\} \qquad n = 0,1,2,3$$

Here, we observe that

$$\left.\begin{aligned} y_4(0) &= y_6(0) + y_6(4) = 1+8=9 \\ y_4(1) &= y_6(1) + y_6(5) = 4+3=7 \end{aligned}\right\}$$

Only first two points, $\{(M-1)=2\}$, are corrupted by the effect of aliasing.

$$\Downarrow$$

$$y_4(0) \neq y_6(0)$$

$$y_4(1) \neq y_6(1)$$

" Time-domain aliasing effects are evident,
When we use $N < L+M-1$ in above scenario"

— "Frequency Analysis of Signals using DFT" —

Let us do it by using an example, as given below.

\# The exponential signal $x_a(t) = \begin{cases} e^{-t}, & t \geq 0 \\ 0, & t < 0 \end{cases}$

is sampled at the rate $F_s = 25$ samples per second, and a block of 125 samples is used to estimate the spectrum.

Determine the spectral characteristics of the signal $x_a(t)$ by computing the DFT of the finite duration sequence. Compare the spectrum of the truncated discrete-time signal to the spectrum of the analog signal.

$\Rightarrow$ The spectrum of analog signal is

$$X_a[j\Omega] = \frac{1}{1+j\Omega} \quad \Rightarrow \quad X_a[F] = \frac{1}{1+j2\pi F}$$

$\Rightarrow$ $|X_a[F]| \Rightarrow$ Magnitude spectrum

$$|X_a[F]| = \frac{1}{\sqrt{1+(2\pi F)^2}}$$

The exponential analog signal sampled at the rate of 25 samples per second results in the sequence

$$x(n) = x_a(nT) = e^{-nT} = e^{-n/25} \quad n \geq 0$$

$$= \left(e^{-1/25}\right)^n \quad n \geq 0$$

$$= (0.9608)^n \quad n \geq 0$$

⇒ Therefore, consider the following sequence

$$\hat{x}(n) = \begin{cases} (0.9608)^n & \rightarrow \text{for } 0 \leq n \leq 124 \\ 0 & \rightarrow \text{otherwise} \end{cases}$$

Truncated sequence with window function of length $L = 125$.
⇒ Rectangular Window

⇒ length of sample block $= 125 = L$

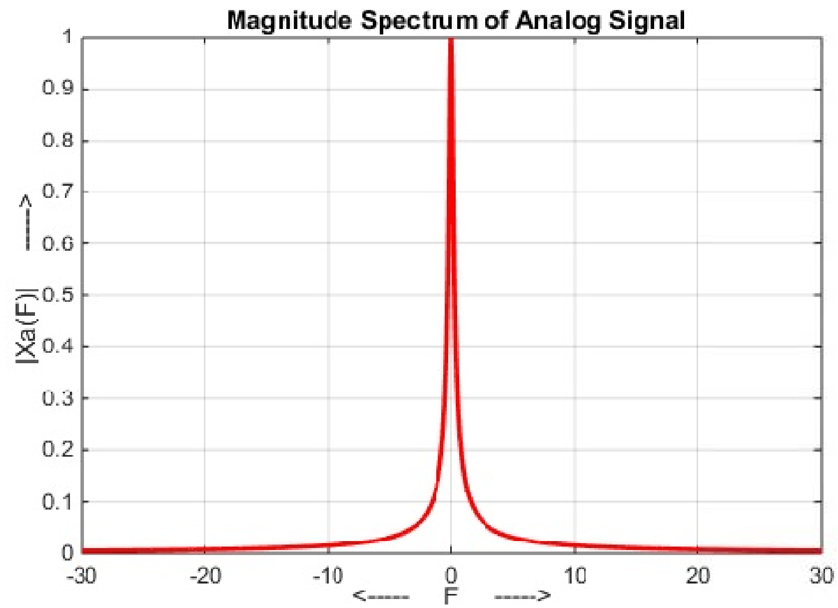
⇒ The N-point DFT of $L = 125$ points is

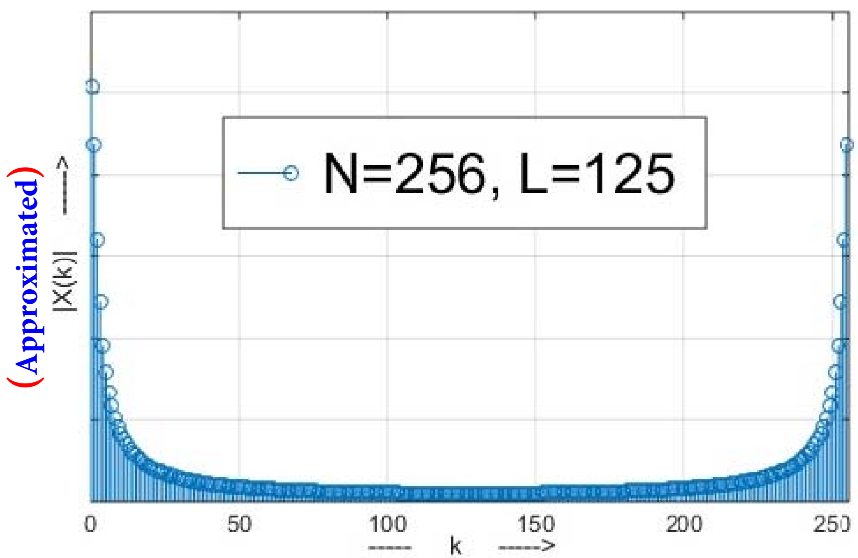
$$\hat{X}[k] = \sum_{l=0}^{N-1} \hat{x}(n)\, e^{-j2\pi l k/N} \quad \text{for } k = 0, 1, \ldots, N-1$$

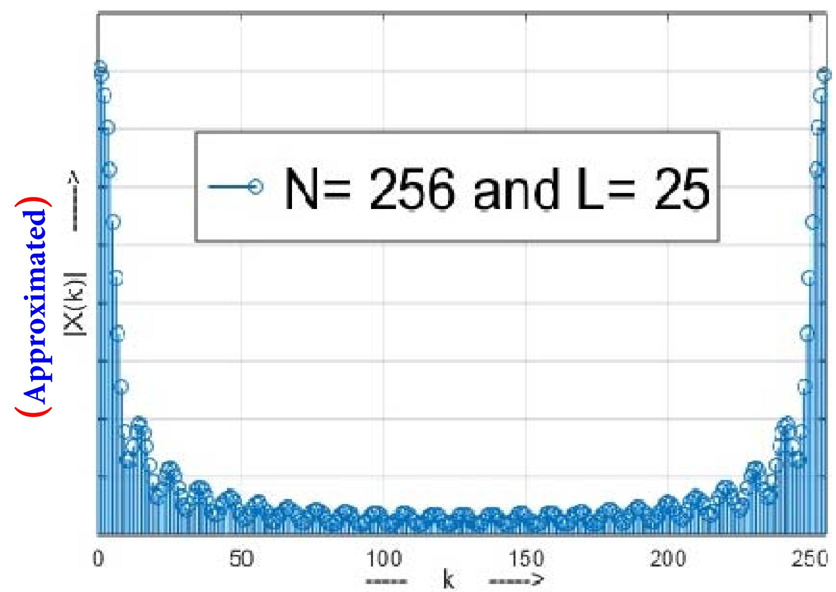
$$= \sum_{l=0}^{L-1} \hat{x}(n)\, e^{-j2\pi k l/N} = \sum_{l=0}^{124} \hat{x}(n)\, e^{-j2\pi k l/N}$$

⇒ To obtain sufficient details in the spectrum, we keep $N = 256$. Use zero-padding

⇒ Magnitude spectrum plots are as follows.


Magnitude Spectrum of Analog Signal
|Xa(F)| ------>
<------ F ------>


(Approximated)
|X(k)| ------>
N=256, L=125
------ k ------>


(Approximated)
|X(k)| ------>
N= 256 and L= 25
------ k ------>

If the window function (rectangular) of length $L = 25$ is selected, then the truncated sequence is

$$\hat{x}(n) = \begin{cases} (0.9608)^n \xrightarrow[\text{for}]{} 0 \leq n \leq 24 \\ 0 \longrightarrow \text{otherwise} \end{cases}$$

It may be inferred from the magnitude spectrum for $N = 256$-point DFT (as shown above) that the sinusoidal envelope variations in the spectrum away from the main peak are due to the large side lobes of the the rectangular window spectrum for $L = 25$.

⇛ Eventually, the DFT for $L = 25$ is no longer a good approximation of the analog signal spectrum

⇛ However for $L = 125$, the magnitude spectrum $|X[k]|$ bears a close resemblance to the spectrum of the analog signal.

Thanks for attending this session on DSP

Reference Books ⇒

⇒ "Digital Signal Processing: Principles, Algorithms and Applications," by J. G. Proakis and D. G. Manolakis; Prentice-Hall, 2007.

⇒ "Theory and Application of Digital Signal Processing," by L.R. Rabiner and B. Gold; Prentice-Hall, 2003.

⇒ "Discrete Time Signal Processing," by A.V. Oppenheim and R.W. Schafer; Prentice-Hall, 2001.

⇒ "Digital Signal Processing," by S.K. Mitra; Tata McGraw-Hill, 2006.

⇒ "Digital Signal Processing," by S. Salivahanan; McGraw-Hill, 2019.

⇒ "Digital Filters: Analysis and Design," by A. Antoniou; Tata McGraw-Hill, 1979.

⇒ "Digital Signal Processing," by E. Ifeachor and B.W. Jervis; Pearson Education, 2007.

⇒ "Signals and Systems," by A.V. Oppenheim, A.S. Willsky and S. H. Nawab; Prentice-Hall of India, 1999. (Prerequisite)

Thanks for attending this session on DSP

DIGITAL SIGNAL PROCESSING
UEC – 502

From My Lecture Notes

Lecture No. – 07

for
Electronics and Communication Engineering Students
&
Electronics and Computer Engineering Students

Dr. Amit Kumar Kohli
Electronics and Communication Engineering Department
Thapar Institute of Engineering & Technology, Patiala, Punjab, India

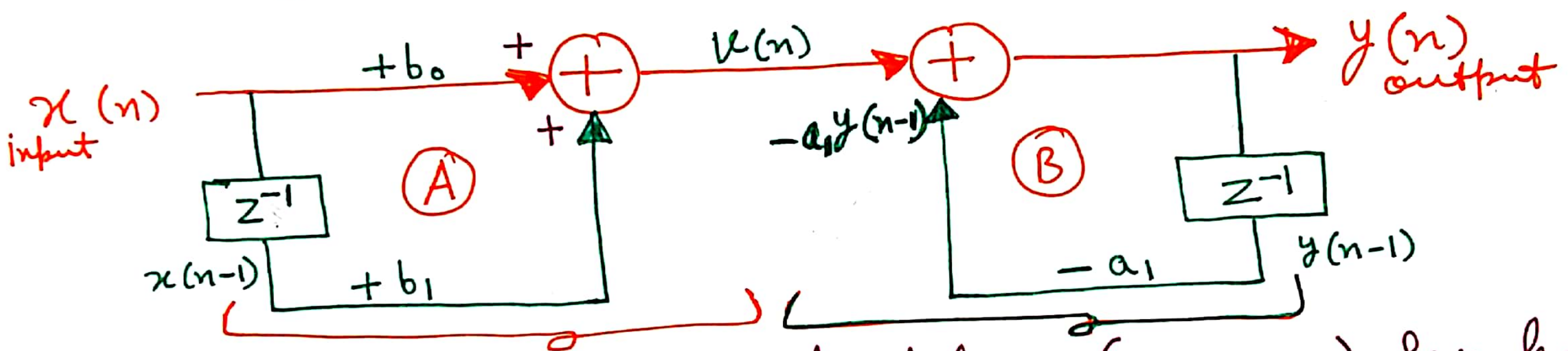
"Implementation of Discrete-time Systems"

— Structures for the realization of LTI systems —

Let us consider the first-order system

$$y(n) = -a_1 y(n-1) + b_0 x(n) + b_1 x(n-1)$$

It is realized as



This realization uses separate delays (memory) for both the input and output signal samples, and it is called a "direct form I structure". This system can be viewed as two LTI systems in cascade.

(A) ⟹ The first is a "nonrecursive system" described by the equation

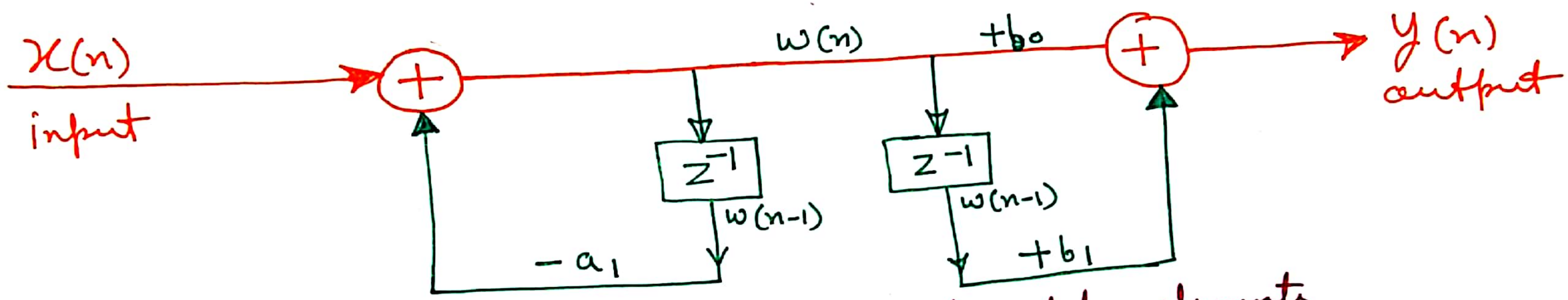
$$v(n) = b_0 x(n) + b_1 x(n-1)$$

(B) ⟹ The second is a "recursive system" described by the equation

$$y(n) = -a_1 y(n-1) + v(n)$$

By using the basic concepts of "signals & systems", if we interchange the order of the cascaded linear time-invariant systems, the overall system response remains the same.

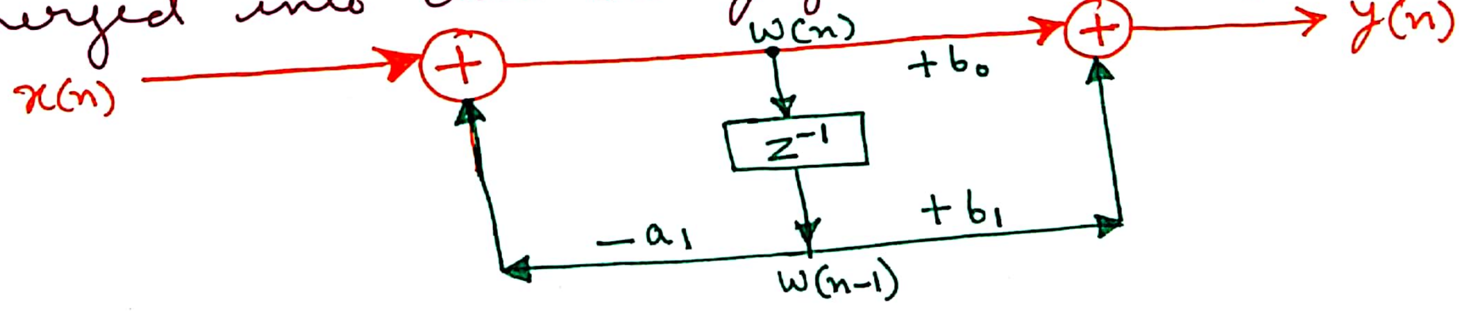
It leads to



Clearly, this structure requires two delay elements

Now,
$$w(n) = -a_1 w(n-1) + x(n)$$
$$y(n) = b_0 w(n) + b_1 w(n-1)$$

However, the above shown two delay elements in this structure can be merged into one delay for the auxiliary quantity $w(n)$.



Dr. A.K. Kohli, ECED, TIET

This new realization requires only one delay, and hence it is more efficient in terms of memory requirements. It is referred to as the "direct form II structure", which has found extensive applications in practical scenarios.

These structures can be generalized for the general LTI recursive system described by the difference equation

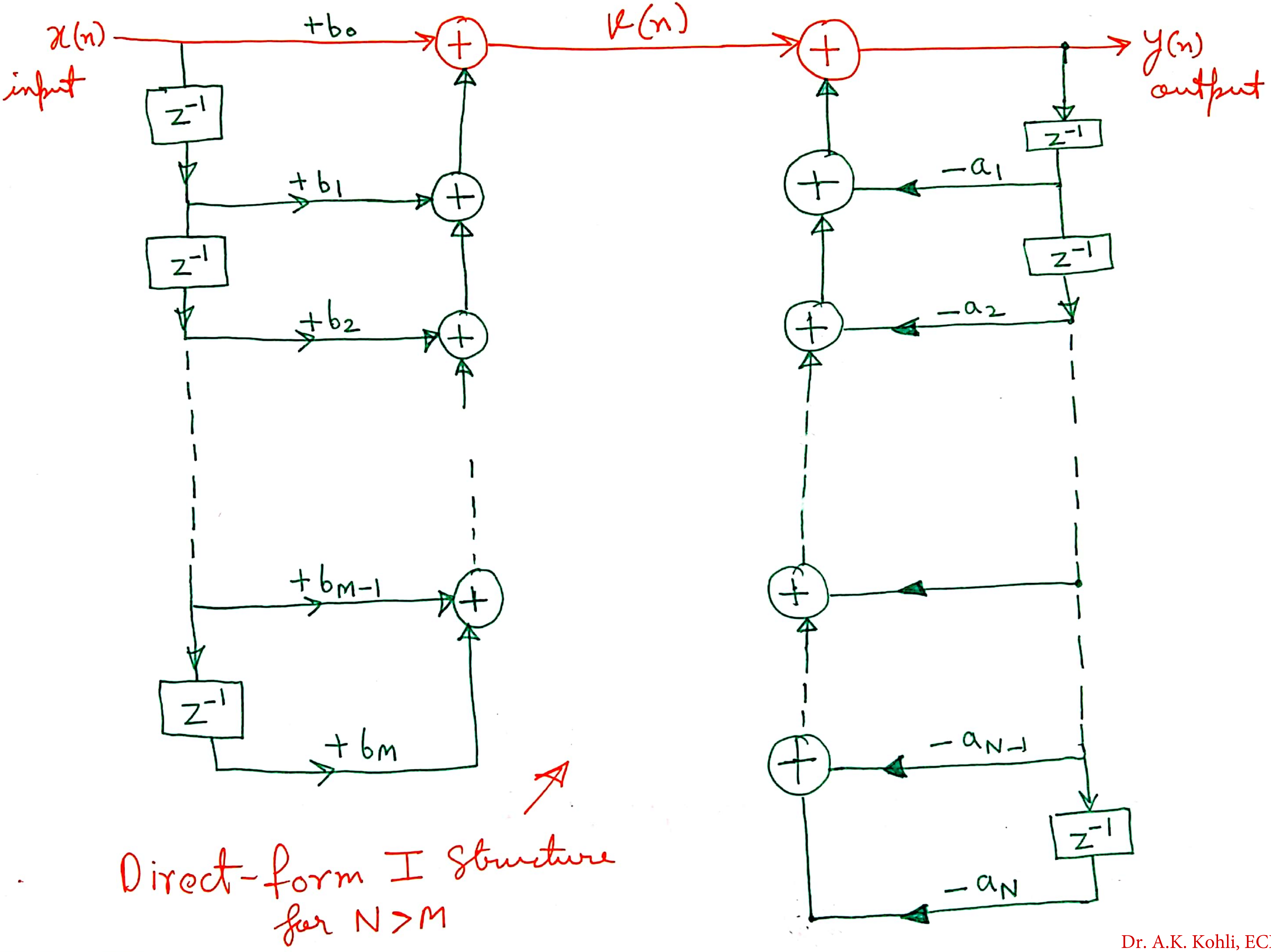
$$y(n) = -\sum_{k=1}^{N} a_k y(n-k) + \sum_{k=0}^{M} b_k x(n-k)$$

or $$y(n) = -\sum_{k=1}^{N} a_k y(n-k) + v(n) \qquad \text{(recursive system)}$$

where,

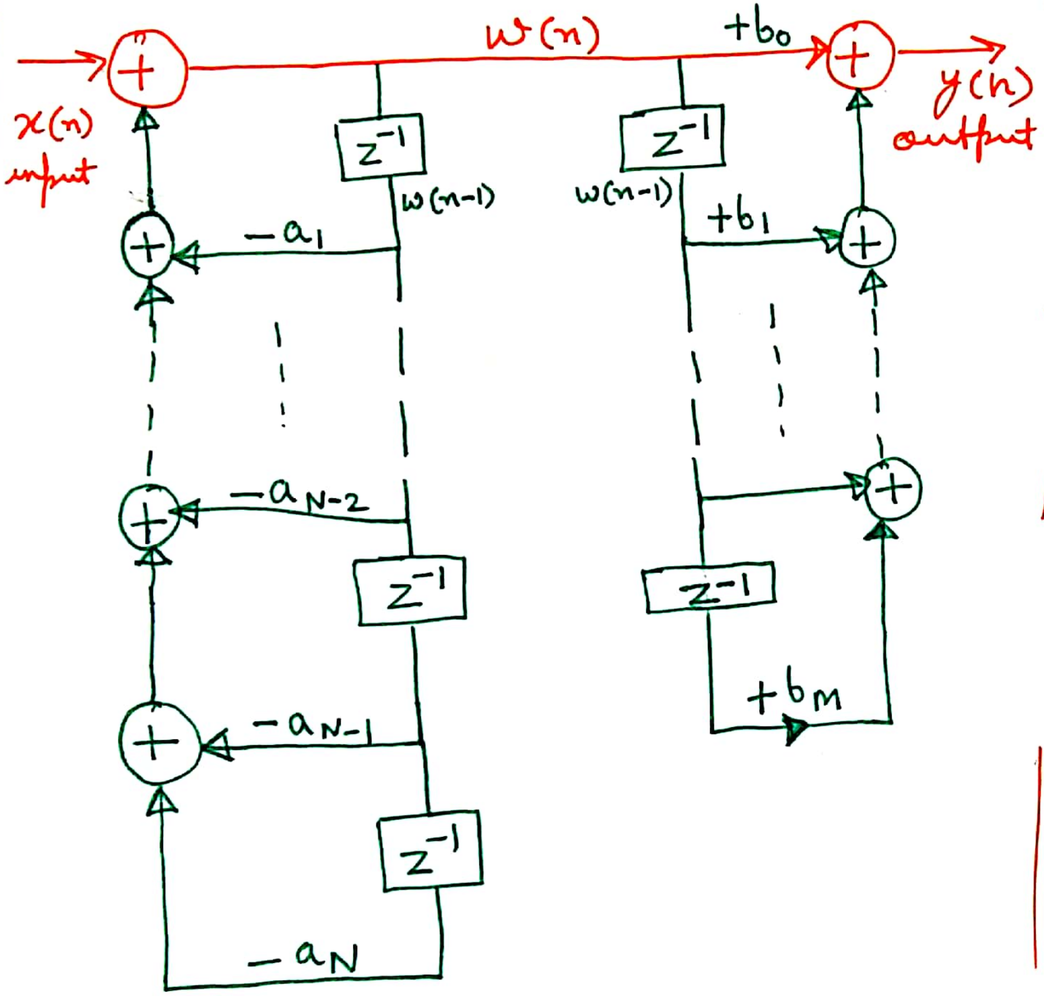
$$v(n) = \sum_{k=0}^{M} b_k x(n-k) \qquad \text{(nonrecursive system)}$$

Just as previous case for first-order system, We can reverse the order of these two systems, to obtain the direct form II structure for $N \geq M$.



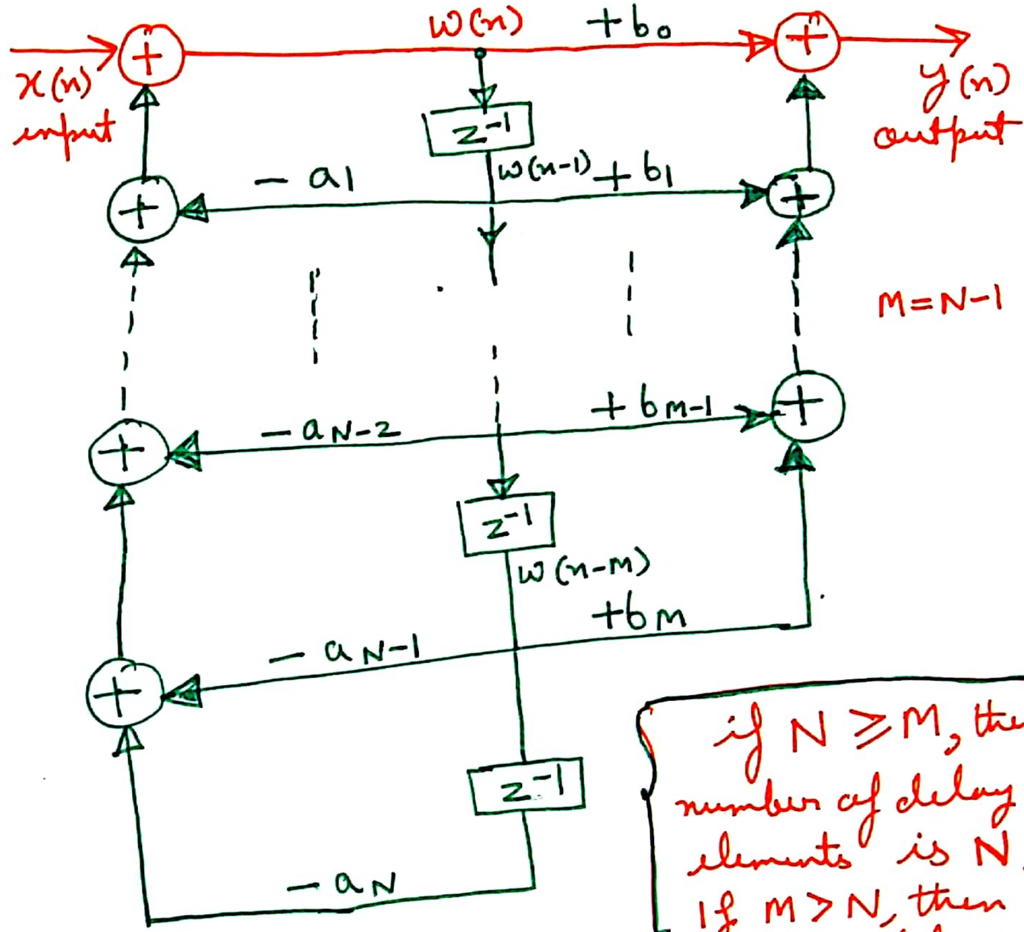
Direct-form I Structure for $N > M$

Direct-form II Structure



— Intermediate-stage —

⇓ By the merger of delay elements



if $N \geq M$, then number of delay elements is N.
If $M > N$, then number of delay elements is M.

— Direct-form II Structure —

⇒ The direct-form II structure requires $M+N+1$ multiplications

⇒ It requires $\max\{M, N\}$ delay elements

The direct-form II structure requires the minimum number of delays for the realization of the above system, therefore it is sometimes referred to as a canonic-form. This structure is the cascade of a recursive system

$$w(n) = -\sum_{k=1}^{N} a_k w(n-k) + x(n)$$

followed by a nonrecursive system

$$y(n) = +\sum_{k=0}^{M} b_k w(n-k)$$

Special Case 1 →

Let us consider that $a_k = 0$ for all k i.e., $k = 1 --- N$

then

$$\underbrace{y(n)}_{\text{Weighted moving average of the input signal}} = \sum_{k=0}^{M} \underbrace{b_k}_{\text{weight}} \underbrace{x(n-k)}_{\text{This system considers most recent } M+1 \text{ input signal samples}}$$

(nonrecursive LTI system)

This system can also be viewed as an FIR system with an impulse response $h(k) = b_k$, where

$$h(k) = \begin{cases} b_k & \text{for } 0 \le k \le M \\ 0 & \text{otherwise} \end{cases}$$

Such that $y(n) = h(n) * x(n)$

Special Case 2 →

Let us assume that $M = 0$ or $b_k = 0$ for $k = 1, --, M$

then

$$y(n) = -\sum_{k=1}^{N} a_k y(n-k) + b_0 x(n)$$

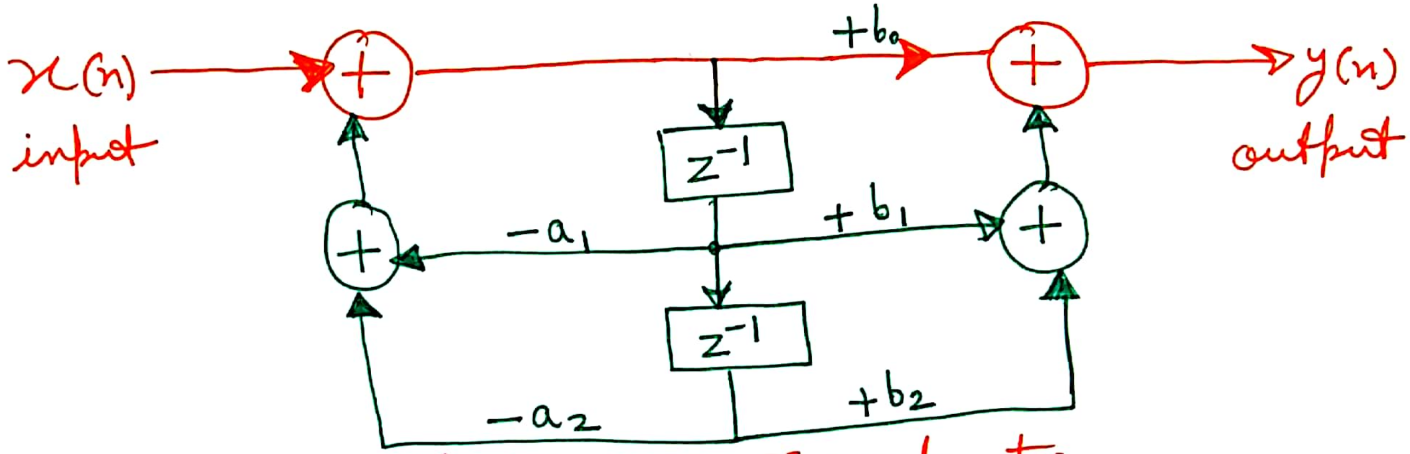
The system output is a weighted linear combination of N past outputs and the present input.

" Second-order systems are usually used as basic building blocks for realizing the higher-order systems, which (LTI systems described by second-order difference equation) is an important subclass of the aforementioned general system."

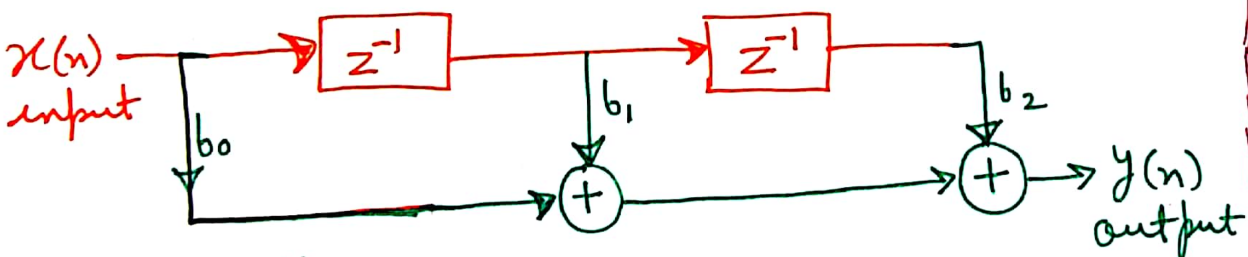
The most general second-order system is described by the difference equation

$$y(n) = -a_1 y(n-1) - a_2 y(n-2) + b_0 x(n) + b_1 x(n-1) + b_2 x(n-2)$$

with
$N = 2$
$M = 2$



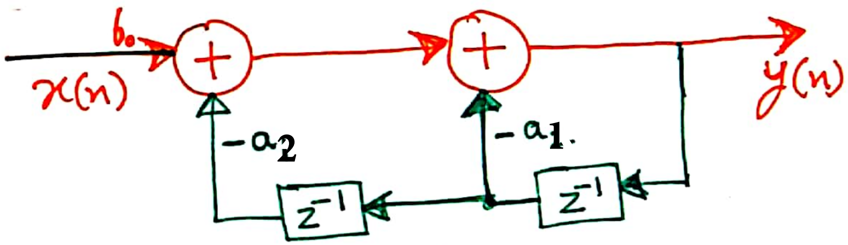
Direct form-II realization



With $a_1 = 0 = a_2$

$$y(n) = b_0 x(n) + b_1 x(n-1) + b_2 x(n-2)$$

It is a special case of the FIR system



with $b_1 = b_2 = 0$

$$y(n) = -a_1 y(n-1) - a_2 y(n-2) + b_0 x(n)$$

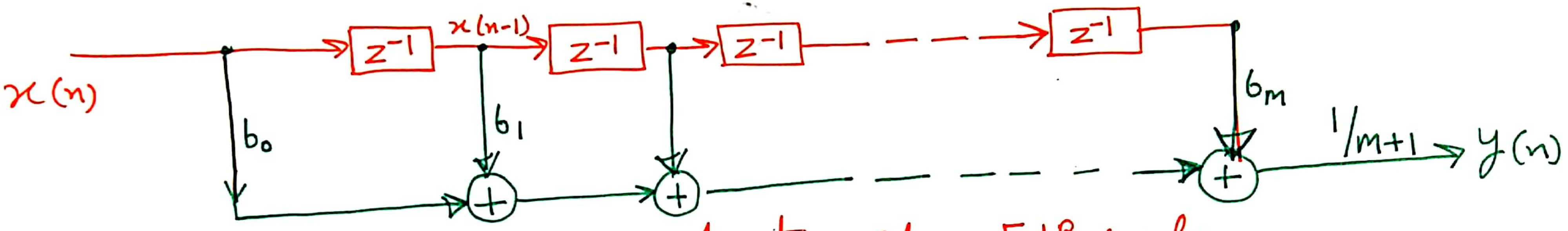
It is a purely recursive second-order system.

Dr. A.K. Kohli, ECED, TIET

Example →

We have a system with an input-out equation

$$y(n) = \sum_{k=0}^{M} h(k)\, x(n-k) \quad \text{(nonrecursive \& FIR Sys)}$$



Nonrecursive realization of an FIR system

If $$h(k) = \begin{cases} b_k/(M+1) & \text{for } 0 \le k \le M \\ 0 & \text{otherwise} \end{cases} \quad \text{(impulse response of FIR system)}$$

then the above shown FIR system is used for computing the moving average of $x(n)$ by using $b_0 = b_1 = b_2 = \cdots = b_k = 1$. It follows that

$$y(n) = \sum_{k=0}^{M} x(n-k)/(M+1)$$

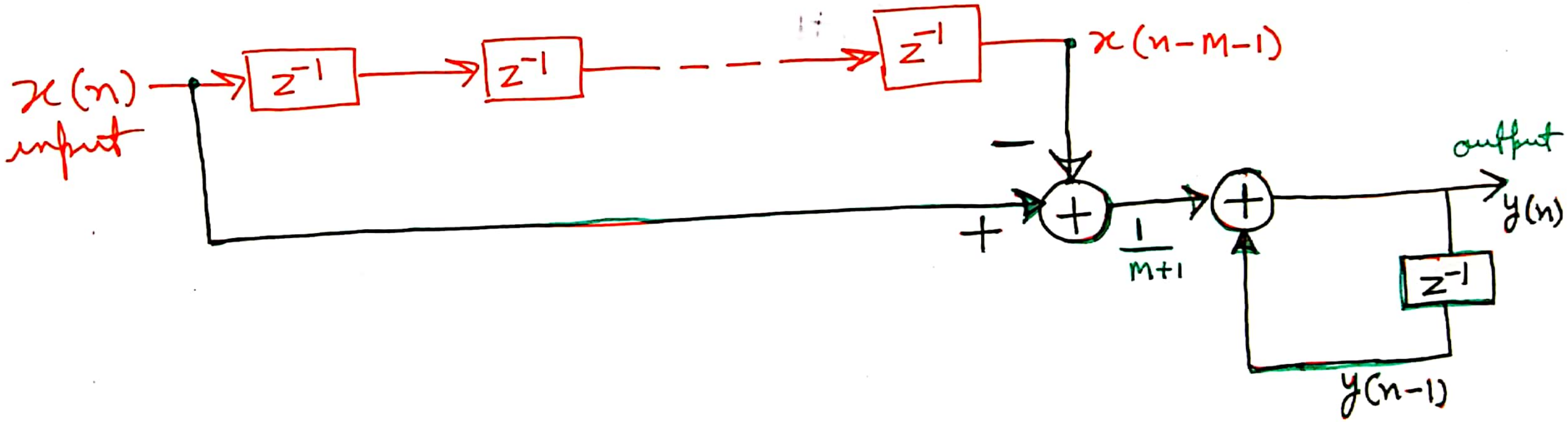
Now, we can express the above equation as

$$y(n) = \frac{1}{M+1} \sum_{k=0}^{M} x(n-1-k) + \frac{1}{M+1}\left[x(n) - x(n-1-M)\right]$$

$$= \frac{1}{M+1}\left\{ x(n) + \boxed{x(n-1) + x(n-2) + \ldots + x(n-M) + x(n-M-1)} - x(n-M-1) \right.$$

$$\left. + \frac{1}{M+1}\left[x(n) - x(n-1-M)\right]\right\}$$

$\begin{Bmatrix}\text{A recursive}\\ \text{realization}\\ \text{of the system}\end{Bmatrix} \Rightarrow y(n) = y(n-1) + \frac{1}{M+1}\left[x(n) - x(n-1-M)\right]$



Thanks for attending this session on DSP

Reference Books ⟹

⟹ "Digital Signal Processing : Principles, Algorithms and Applications," by J. G. Proakis and D. G. Manolakis ; Prentice-Hall, 2007.

⟹ "Theory and Application of Digital Signal Processing," by L.R. Rabiner and B. Gold; Prentice-Hall, 2003.

⟹ "Discrete Time Signal Processing," by A.V. Oppenheim and R.W. Schafer; Prentice-Hall, 2001.

⟹ "Digital Signal Processing," by S.K. Mitra; Tata McGraw-Hill, 2006.

⟹ "Digital Signal Processing," by S. Salivahanan; McGraw-Hill, 2019.

⟹ "Digital Filters: Analysis and Design," by A. Antoniou; Tata McGraw-Hill, 1979.

⟹ "Digital Signal Processing," by E. Ifeachor and B.W. Jervis; Pearson Education, 2007.

⟹ "Signals and Systems," by A.V. Oppenheim, A.S. Willsky and S. H. Nawab; Prentice-Hall of India, 1999. (Prerequisite)

Thanks for attending this session on DSP

DIGITAL SIGNAL PROCESSING
UEC – 502

From My Lecture Notes

Lecture No. – 08

for

Electronics and Communication Engineering Students

&

Electronics and Computer Engineering Students

Dr. Amit Kumar Kohli

Electronics and Communication Engineering Department

Thapar Institute of Engineering & Technology, Patiala, Punjab, India

"A linear filtering approach to the computation of the DFT"

- There are a few applications, where only a selected number of values of the DFTs are desired, but the entire DFT is not required. In such cases, fast-Fourier-transform algorithms may no longer be more efficient than a direct computation of the desired values of DFT.
- The direct computation of the DFT can be formulated as a linear filtering operation on the input data sequence. The linear filter takes the form of a parallel bank of resonators, where each resonator selects one of the frequencies $\omega_k = 2\pi k/N$, $k = 0, 1, \ldots, (N-1)$ corresponding the N frequencies in the DFT.

The **Goertzel algorithm** exploits the periodicity of the phase factors $\{W_N^k\}$, which allows us to represent the computation of the DFT as a linear filtering operation.

We first multiply the DFT by the factor W_N^{-kN}, and it follows that

$$X[k] = W_N^{-kN} \sum_{m=0}^{N-1} x(m)\, W_N^{km} \qquad \boxed{W_N^{-kN} = 1}$$

$$X[k] = \sum_{m=0}^{N-1} x(m)\, W_N^{-k(N-m)} \longrightarrow (A')$$

→ It is in the form of a convolution sum

We next define the sequence $y_k(n)$ as the output of an LTI (Linear filter) system

$$y_k(n) = \sum_{m=0}^{N-1} x(m)\, W_N^{-k(n-m)} \longrightarrow (B)$$

Equivalently, $y_k(n) = x(n) * h_k(n)$

→ $y_k(n)$ is the convolution of the finite-duration input sequence $x(n)$ of length N with a filter that has an impulse response $h_k(n)$.

where, $h_k(n) = W_N^{-kn}\, u(n)$

The output of this filter at $n = N$ yields the value of the DFT at the particular angular frequency $\omega_k = 2\pi k/N$, such that

(Similar to (A'))

$$y_k(n)\Big|_{n=N} = X[k] = \sum_{m=0}^{N-1} x(m)\, W_N^{-k(N-m)}$$

The impulse response $h_k(n)$ of the concerned filter can be expressed in the corresponding frequency domain (system function) as

$$\text{C} \longrightarrow \quad H_k[z] = \frac{1}{1 - W_N^{-k} z^{-1}} \qquad \left\{ \begin{matrix} \text{z-transform} \\ \text{of } h_k(n) \end{matrix} \right\}$$

This filter has a pole on the unit circle at the frequency $\omega_k = \frac{2\pi k}{N}$

The entire DFT can be computed by passing the block of input data into a parallel bank of "N" single-pole filters (resonators), where each filter has a pole at the corresponding frequency of DFT.

Using equation C, we can write that

$$H_k[z] - W_N^{-k} z^{-1} H_k[z] = 1$$

$$\boxed{\frac{Y_k[z]}{X_k[z]} = H_k[z]}$$

$$\frac{Y_k[z]}{X_k[z]} - W_N^{-k} z^{-1} \frac{Y_k[z]}{X_k[z]} = 1$$

$$Y_k[z] - W_N^{-k} z^{-1} Y_k[z] = X_k[z]$$

Recursive Equation with $y_k(-1) = 0$

By inverse z-transform, we can get $\boxed{y_k(n) = W_N^{-k} y_k(n-1) + x(n)}$ D

The desired output is $X[k] = y_k(N)$; for $k = 0, 1, ---, N-1$.

To perform this computation, we need to compute the phase factor W_N^{-k} once for any value of k in the range $k = 0, 1 -- N-1$.

→ The complex multiplications and additions inherent in equation (D) can be avoided by combining the pairs of resonators possessing complex conjugate poles. It results in two-pole filters with system functions of the form

$$\frac{Y_k[z]}{X_k[z]} = H_k[z] = \frac{1}{1-W_N^{-k}z^{-1}} \times \frac{1-W_N^{k}z^{-1}}{1-W_N^{k}z^{-1}} = \frac{1-W_N^{k}z^{-1}}{1-2\left(\frac{W_N^{k}+W_N^{-k}}{2}\right)z^{-1}+z^{-2}} = \boxed{\frac{1-W_N^{k}z^{-1}}{1-2\cos\left(\frac{2\pi k}{N}\right)z^{-1}+z^{-2}}}$$

Its further analysis results in

$$Y_k[z] - 2\cos\left(\frac{2\pi k}{N}\right)z^{-1}Y_k[z] + z^{-2}Y_k[z] = X_k[z] - W_N^{k}z^{-1}X_k[z]$$

or we can use another approach as

$$\frac{Y_k[z]}{X_k[z]} = \frac{Y_k[z]}{V_k[z]} \times \frac{V_k[z]}{X_k[z]} = (1-W_N^{k}z^{-1}) \times \frac{1}{1-2\cos\left(\frac{2\pi k}{N}\right)z^{-1}+z^{-2}}$$

Dr. A.K. Kohli, ECED, TIET

$$H_k[z] = \underbrace{H_k^1[z]} \times \underbrace{H_k^2[z]}$$

$$H_k^1[z] = \frac{V_k[z]}{X_k[z]} = \frac{1}{1 - 2\cos\left(\frac{2\pi k}{N}\right) z^{-1} + z^{-2}} \xrightarrow{\text{Sub-system}} (E)$$

$$H_k^2[z] = \frac{Y_k[z]}{V_k[z]} = \left(1 - W_N^k z^{-1}\right) \xrightarrow{\text{Sub-system}} (F)$$

From equation (E), by using Inverse z-transform, we can show that

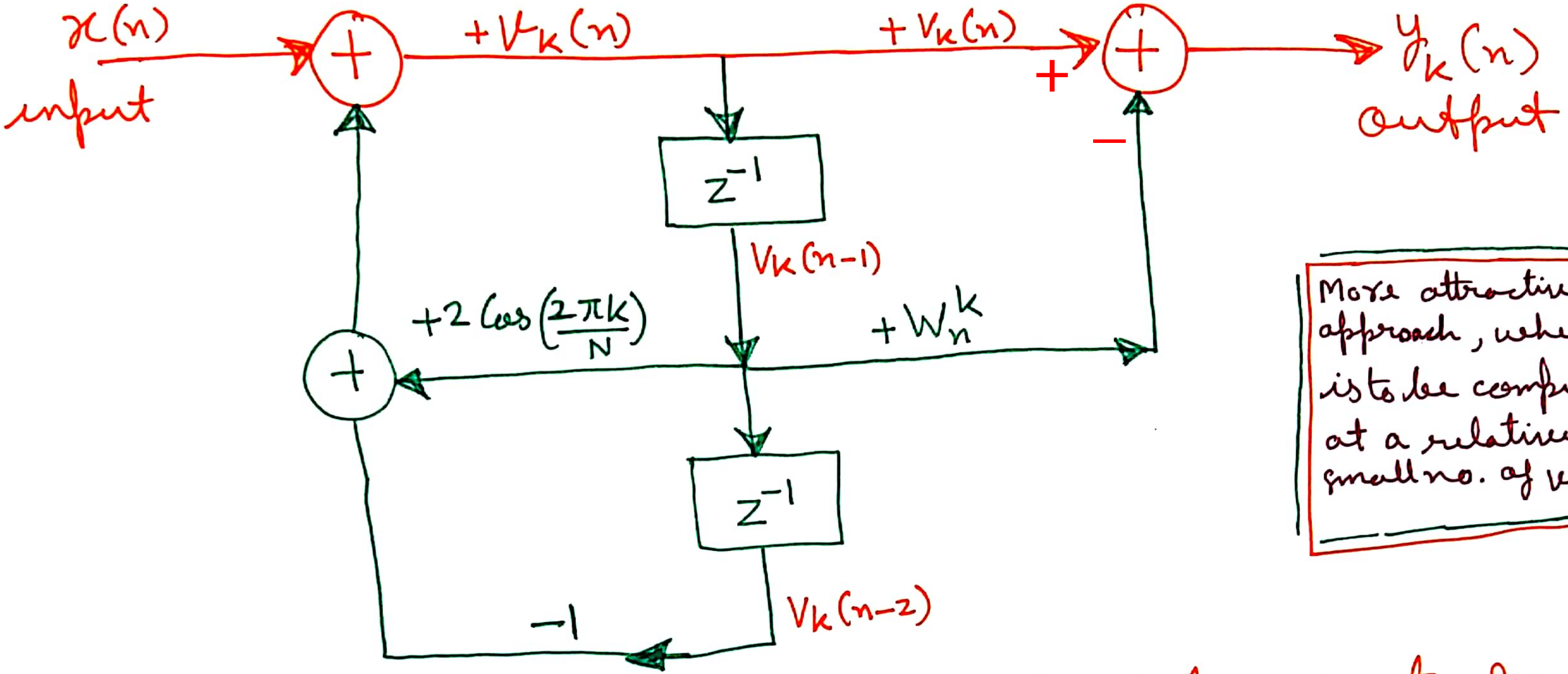
$$v_k(n) - 2\cos\left(\frac{2\pi k}{N}\right) v_k(n-1) + v_k(n-2) = x(n)$$

$$\boxed{v_k(n) = 2\cos\left(\frac{2\pi k}{N}\right) v_k(n-1) - v_k(n-2) + x(n)} \longrightarrow (G)$$

From equation (F), by using Inverse z-transform, it is clear that

$$(H) \longrightarrow \boxed{y_k(n) = v_k(n) - W_N^k v_k(n-1)}$$

with initial conditions $v_k(-1) = v_k(-2) = 0$



More attractive approach, when DFT is to be computed at a relatively small no. of values.

Direct form II realization of two-pole resonator for computing the DFT

- The recursive relation (G) is iterated for $n = 0, 1, ---, N$; however, the equation (H) is computed once at time $n = N$.
- Each iteration requires one real multiplication and two additions.
- For a real input $x(n)$, this approach needs $(N+1)$ real multiplications to compute $X[k]$, however, it also provides the value of $X[N-k]$ due to symmetry property.

"A linear filtering approach to computation of DFT — Chirp-Z Transform Algorithm"

If the set of desired points in Z-plane possesses some regularity, its is possible to express the computation of the Z-transform as a linear filtering operation

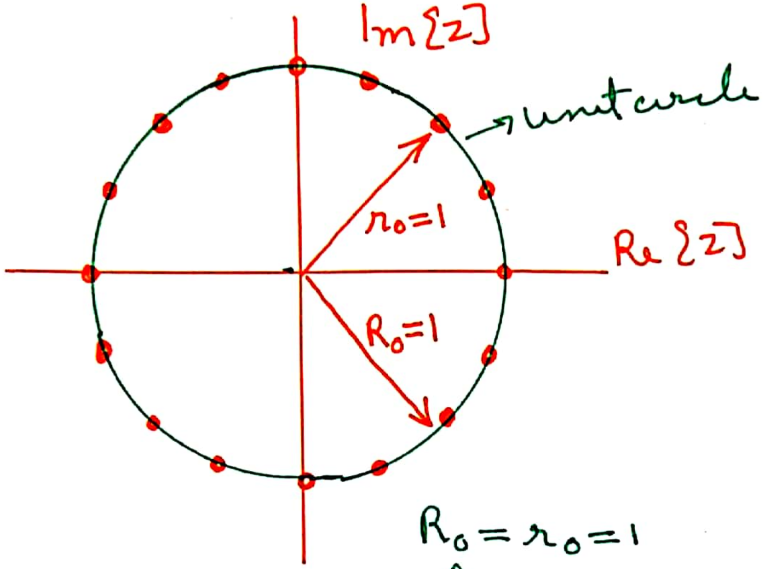
The DFT of an N-point data sequence $x(n)$ can be viewed as the Z-transform of $x(n)$ evaluated at N equally spaced points on the unit circle. (i.e., N-equally spaced samples of DTFT)

If we wish to compute the values of Z-transform of $x(n)$ at a set of points $\{Z_k\}$, then

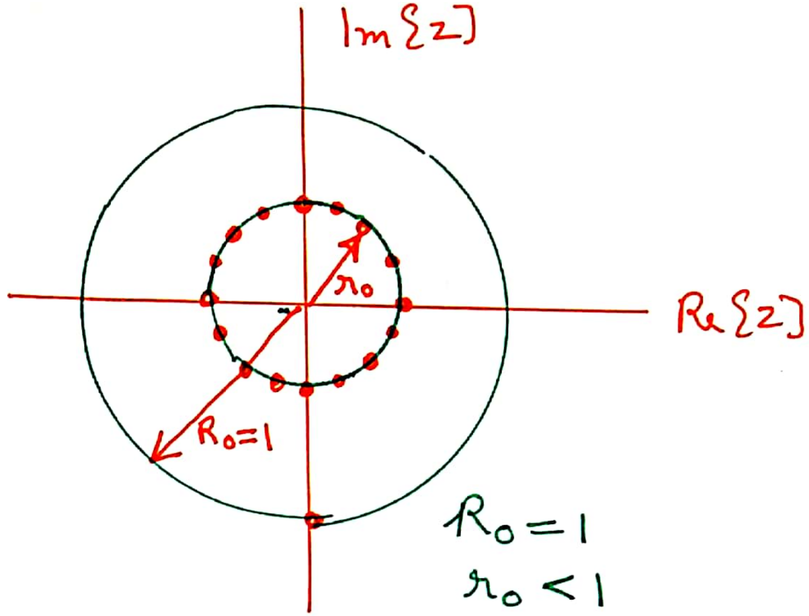
$$X[Z_k] = \sum_{n=0}^{N-1} x(n)\, Z_k^{-n} \qquad k = 0, 1, 2, ---, N-1$$

Let us assume that the points Z_k in the Z-plane fall on an arc, which begins at some point $Z_0 = r_0 e^{j\theta_0}$, and spirals either towards origin or out away from origin, such that the points $\{Z_k\}$ are defined as

$$Z_k = r_0 e^{j\theta_0} \left(R_0 e^{j\phi_0}\right)^k \qquad k = 0, 1, 2 ---, N-1$$



$R_0 = r_0 = 1$
$\phi_0 = \theta_0 = 0$



$R_0 = 1$
$r_0 < 1$
$\phi_0 = \theta_0 = 0$

Thus,

$$X[z_k] = \sum_{n=0}^{N-1} x(n) z_k^{-n} = \sum_{n=0}^{N-1} x(n) (r_0 e^{j\theta_0})^{-n} (R_0 e^{j\phi_0})^{-nk}$$

$$= \sum_{n=0}^{N-1} x(n) (r_0 e^{j\theta_0})^{-n} V^{-nk} \quad \text{with} \quad V = R_0 e^{j\phi_0}$$

We know that $nk = \frac{1}{2}[n^2 + k^2 - (k-n)^2]$

$$X[z_k] = V^{-k^2/2} \sum_{n=0}^{N-1} \Big[\underbrace{x(n) (r_0 e^{j\theta_0})^{-n} V^{-n^2/2}}_{g(n)} \Big] \underbrace{V^{(k-n)^2/2}}_{h(k-n)}$$

We now define a new sequence $g(n) = x(n) (r_0 e^{j\theta_0})^{-n} V^{-n^2/2}$

$$X[Z_k] = V^{-k^2/2} \sum_{n=0}^{N-1} g(n)\, V^{(k-n)^2/2}$$

It can be viewed as the convolution of the sequence $g(n)$ with impulse response $h(n)$ of a filter, such that $h(n) = V^{n^2/2}$

$$X[Z_k] = V^{-k^2/2}\, y(k)$$

where, $y(k) = \sum_{n=0}^{N-1} g(n)\, V^{(k-n)^2/2} = g(k) * h(k)$

$$= \sum_{n=0}^{N-1} g(n)\, h(k-n)$$

$$X[Z_k] = y[k]/h(k) \quad , \text{ as } h(k) = V^{k^2/2}$$

$$k = 0, 1, 2, \ldots, N-1$$

Here, $y(k)$ is interpreted as the output of a linear filter. $h(n)$ and $g(n)$ are both complex valued sequences.

If sequence $x(n)$ exhibits length equal to L, then we use the zero-padding procedure to extend the length to N.

Finally, $X[Z_k]$ are computed by dividing $y(k)$ by $h(k)$ for $k = 0, 1, 2 \ldots L-1$

The sequence $h(n)$ with $R_0 = 1$ has the form of a complex exponential with argument $\omega n = n^2 \phi_0/2 = (n\phi_0/2)n$. The quantity $n\phi_0/2$ represents the frequency of the complex exponential signal, which increases linearly with time.

Such signals are used in radar systems, and are called "chirp signals".

Hence, the z-transform evaluated as

$$X[z_k] = V^{-k^2/2} \sum_{n=0}^{N-1} g(n)\, V^{(k-n)^2/2}$$

is called the chirp z-transform

Now, for the computation of $\underline{\underline{\text{DFT}}}$, we select $r_0 = R_0 = 1$ and $\theta_0 = 0$, $\phi_0 = 2\pi/N$, and $L = N$. In this case, we consider

$$h(n) = V^{+n^2/2} = e^{+j(2\pi/N)n^2/2} = e^{j\pi n^2/N} = \cos\left(\frac{\pi n^2}{N}\right) + j\sin\left(\frac{\pi n^2}{N}\right)$$

$$= h_R(n) + j\, h_I(n)$$

$\Rightarrow$ $h(n)$ is the impulse response of the linear filter, (chirp filter).

Thanks for attending this session on DSP

<u>Reference Books</u> ⇒

⇒ "Digital Signal Processing : Principles, Algorithms and Applications," by J. G. Proakis and D. G. Manolakis ; Prentice-Hall, 2007.

⇒ "Theory and Application of Digital Signal Processing," by L.R. Rabiner and B. Gold; Prentice-Hall, 2003.

⇒ "Discrete Time Signal Processing," by A.V. Oppenheim and R.W. Schafer; Prentice-Hall, 2001.

⇒ "Digital Signal Processing," by S.K. Mitra; Tata McGraw-Hill, 2006.

⇒ "Digital Signal Processing," by S. Salivahanan; McGraw-Hill, 2019.

⇒ "Digital Filters: Analysis and Design," by A. Antoniou; Tata McGraw-Hill, 1979.

⇒ "Digital Signal Processing," by E. Ifeachor and B.W. Jervis; Pearson Education, 2007.

⇒ "Signals and Systems," by A.V. Oppenheim, A.S. Willsky and S.H. Nawab; Prentice-Hall of India, 1999. (Prerequisite)

Thanks for attending this session on DSP

DIGITAL SIGNAL PROCESSING
UEC – 502

From My Lecture Notes

Lecture No. – 09

for

Electronics and Communication Engineering Students

&

Electronics and Computer Engineering Students

Dr. Amit Kumar Kohli

Electronics and Communication Engineering Department

Thapar Institute of Engineering & Technology, Patiala, Punjab, India

— Fast Fourier Transform Algorithms — (FFT)

for efficient computation of DFT

As DFT is used for linear filtering, correlation analysis and spectrum analysis etc., therefore its efficient computation is a topic of prime interest.

Let $x(n)$ be the data sequence of length N. The DFT is computed to generate the sequence $X[k]$, such that its $\{\vec{X}\}$ elements are N complex valued numbers. We also assume that $x(n)$ is also complex valued

(A) → DFT ⟹ $X[k] = \sum_{n=0}^{N-1} x(n) W_N^{kn}$ with $W_N = e^{-j2\pi/N}$ for $0 \leq k \leq N-1$

(B) → IDFT ⟹ $x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X[k] W_N^{-kn}$ for $0 \leq n \leq N-1$

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

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

Represented in terms of real & imaginary components

The direct computation of equations (C) and (D) requires

Inefficient Approach

$\Rightarrow$ $N \times N$ complex multiplications $\Rightarrow$ $4N \times N$ real multiplications

$\Rightarrow$ $(N-1) \times N$ complex additions $\Rightarrow$ $4(N-1) \times N$ real additions

$\Rightarrow$ $2N^2$ evaluations of trigonometric functions

$\Rightarrow$ Many indexing and addressing operations

for $k = 0, 1, 2, ----, N-1$

But, we have an option that we can exploit the symmetry and periodicity properties of the phase factor W_N (twiddle factor); and it follows that

$$\underbrace{W_N^{k+N/2} = -W_N^k}_{-SP-} \quad \& \quad \underbrace{W_N^{k+N} = W_N^k}_{-PP-}$$

The fast Fourier transform algorithms mainly exploit these two fundamental properties of twiddle factor/phase factors.

Basics of Divide-and-Conquer Approach for DFT computation →

$N = L \times M$ ⇒ factorization as a product of two integers

If N is a prime number, then we can use zero-padding to ensure appropriate factorization.

0	1			N−1
$x(0)$	$x(1)$	− − −	− − −	$x(N-1)$

→ $x(n)$ one dimensional array

It can be stored in two-dimensional array by using

$x(l,m)$ ⇒

row index l \ column index m	0	1	…	M−1
0	$x(0,0)$	$x(0,1)$		$x(0,M-1)$
1	$x(1,0)$	$x(1,1)$		$x(1,M-1)$
2				
⋮				
L−1	$x(L-1,0)$	$x(L-1,1)$		$x(L-1,M-1)$

Mapping of index n to indices (l,m)

$n = Ml + m$ or $n = l + mL$

for Row wise entry — for column wise entry

Rowwise entry

$x(l,m) \Rightarrow$ 2-D

$l \backslash m$	0	1		$M-1$
0	$x(0)$	$x(1)$		$x(M-1)$
1	$x(M)$	$x(M+1)$		$x(2M-1)$
			$n = Ml + m$	
$L-1$	$x((L-1)M)$			$x(LM-1)$

first row consists of the first M elements of $x(n)$, and so on.....

Columnwise entry

$x(l,m) \Rightarrow$ 2-D

$l \backslash m$	0	1		$M-1$
0	$x(0)$	$x(L)$		$x((M-1)L)$
1	$x(1)$	$x(L+1)$		
			$n = l + mL$	
$L-1$	$x(L-1)$	$x(2L-1)$		$x(LM-1)$

The mapping stores first L elements of $x(n)$ in first column, and so on.....

A similar arrangement can be used to store the computed DFT values. It involves the mapping of index k to a pair of indices (p, q).

$$p = 0, 1, \ldots L-1$$
$$q = 0, 1, \ldots M-1$$

If $k = Mp + q$, then the DFT values are stored on row basis that is, first row contains first M elements of DFT $X[k]$, and so on

If $k = p + qL$, then the DFT values are stored on column basis that is, first L elements of $X[k]$ are stored in first column, and so on

Therefore, $x(n)$ is mapped to $\rightarrow x(l, m)$ array
$X[k]$ is mapped to $\rightarrow X(p, q)$ array

Let us assume the column wise mapping for $x(n)$, and the row wise mapping for $X[k]$, which results in

$$X[p,q] = \sum_{l=0}^{L-1} \sum_{m=0}^{M-1} x(l,m)\, e^{-j\frac{2\pi}{N}(mL+l)(Mp+q)}$$

$$X[p,q] = \sum_{l=0}^{L-1}\sum_{m=0}^{M-1} x(l,m)\, \underbrace{W_N^{(mL+l)(Mp+q)}}$$

$$W_N^{(mL+l)(Mp+q)} = W_N^{MLmp} \times W_N^{mLq} \times W_N^{Mpl} \times W_N^{lq}$$

$$= \cancel{W_N^{Nmp}} \rightarrow 1 \times W_{N/L}^{mq} \times W_{N/M}^{pl} \times W_N^{lq}$$

$$= W_M^{mq} \times W_L^{pl} \times W_N^{lq}$$

$$X[p,q] = \underbrace{\sum_{l=0}^{L-1}}_{\text{L-pt DFT}} \underbrace{\left[W_N^{lq} \underbrace{\left[\underbrace{\sum_{m=0}^{M-1}}_{\text{M-pt DFT}} x(l,m)\, W_M^{mq} \right]}_{\text{DFT}} \right]}_{\text{DFT}} W_L^{lp}$$

Now, three step computation procedure is as follows.

① ⟹ First, we compute the M-point DFTs for $0 \leq q \leq M-1$

$$F(l,q) = \sum_{m=0}^{M-1} x(l,m)\, W_M^{mq}$$

for each of the rows $l = 0, 1, ---, L-1$

L-DFTs each of M-points
LM^2 Complex multiplications
$LM(M-1)$ Complex additions

② ⟹ Next, we compute a new rectangular array $G(l,q)$

$$G(l,q) = W_N^{lq}\, F(l,q) \qquad 0 \leq q \leq M-1, \quad 0 \leq l \leq L-1$$

LM Complex multiplications

③ ⟹ Finally, we compute the L-point DFTs

$$X(p,q) = \sum_{l=0}^{L-1} G(l,q)\, W_L^{lp}$$

ML^2 Complex multiplications
$ML(L-1)$ Complex additions
in M-DFTs each of L-points

for each of the columns $q = 0, 1, ---, M-1$ of the array $G(l,q)$

Equivalently
$$X(p,q) = \sum_{l=0}^{L-1} \left[W_N^{lq} F(l,q) \right] W_L^{lp}$$
$p = 0, 1, ---, L-1$

Finally, its computational complexity is → No. of complex multiplications $= N(M+L+1)$
No. of complex additions $= N(M+L-2)$

where $N = LM$.

Therefore, no. of multiplications has been reduced from N^2 to $N(L+M+1)$

and, no. of additions has been reduced from $N(N-1)$ to $N(L+M-2)$

"This divide-and-conquer approach is based on the decomposition of an N-point DFT into successively smaller DFTs; which leads to a family of computationally efficient algorithms referred to as FFT algorithms (collectively)."

Example:- Let $N = 1024$, and we select $L = 4$ and $M = 256$,

⇒ then the direct computation of N-point DFT requires

- (CM) No. of complex multiplications = 1048576 ✓
- (CA) No. of complex additions = 1047552 ✓

⇒ then the divide-and-conquer approach results in

- (CM) No. of complex multiplications = 267264 ✓
- (CA) No. of complex additions = 264192 ✓

Advantage

CM and CA are reduced by approx. a factor of 4.

— Radix – 2 FFT algorithm —

We can exploit the divide-and-conquer approach to derive fast algorithms when the size of DFT is restricted to be a power of 2 or a power of 4. This approach is quite efficient when N is highly composite i.e., when N can be factored as $N = r_1 . r_2 . r_3 . \; . r_\gamma$ (where r_j is prime).

If $r_1 = r_2 = \ldots\ldots = r_v = r$, then $N = r^v$. Here, r is called the radix of FFT algorithm. DFTs are of size r, so that the computation of N-point DFT has a regular pattern.

— "Radix-2 is the most widely used FFT algorithm" —

Radix-2 FFT algorithm with "Decimation in time" approach →

We select $M = N/2$ and $L = 2$ in divide-and-conquer approach

$\Rightarrow$ Splitting N-point data sequence into two N/2-point data sequences $f_1(n) = x(2n)$ and $f_2(n) = x(2n+1)$ $\quad n = 0, 1, \ldots, \frac{N}{2}-1$

$f_1(n)$ (Even-samples) and $f_2(n)$ (Odd-samples) are generated by decimating $x(n)$ by a factor of 2 i.e., decimation-in-time process. It follows that

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

$$= \sum_{n\text{-Even}} x(n)\, W_N^{kn} + \sum_{n\text{-Odd}} x(n)\, W_N^{kn}$$

$$= \sum_{m=0}^{N/2-1} x(2m)\, W_N^{2km} + \sum_{m=0}^{N/2-1} x(2m+1)\, W_N^{(2m+1)k}$$

We know that $W_N^2 = W_{N/2}$, and it leads to

$$X[k] = \underbrace{\sum_{m=0}^{N/2-1} f_1(m)\, W_{N/2}^{km}}_{N/2\text{-point DFT}} + \underbrace{\sum_{m=0}^{N/2-1} f_2(m)\, W_{N/2}^{km}}_{N/2\text{-point DFT}} \times W_N^k$$

$$\boxed{X[k] = F_1[k] + W_N^k F_2[k]} \qquad k = 0, 1, ---, N-1$$

$F_1[k]$ & $F_2[k]$ are periodic with period $N/2$, which results in

$$F_1[k+N/2] = F_1[k] \qquad \& \qquad W_N^{k+N/2} = -W_N^k$$
$$F_2[k+N/2] = F_2[k]$$

Therefore, we can show that

$$\left.\begin{aligned} X[k] &= F_1[k] + W_N^k F_2[k] \\ X[k+N/2] &= F_1[k] - W_N^k F_2[k] \end{aligned}\right\} k = 0, 1, \ldots, N/2 - 1$$

$\Rightarrow$ $F_1[k]$ requires $N/2 \times N/2$ complex multiplications

$\Rightarrow$ $F_2[k]$ requires $N/2 \times N/2$ complex multiplications

$\Rightarrow$ $(N/2)$ additional multiplications required to compute $W_N^k F_2[k]$

$\Rightarrow$ Total no. of complex multiplications in computation of $X[k]$ is

$$= \left(\frac{N}{2}\right)^2 + N/2 + (N/2)^2 = \frac{N^2}{2} + N/2$$
$$= \frac{N^2}{2}\left(1 + \frac{1}{N}\right)$$

Therefore, the reduction of complex multiplications is observed to be from $N^2 \longrightarrow$ to $\rightarrow \frac{N^2}{2}\left(1+\frac{1}{N}\right)$; this reduction is by approx. a factor of 2 for large N.

Now, we may define

$$G_1[k] = F_1[k] \qquad k = 0, 1, \ldots \frac{N}{2}-1$$

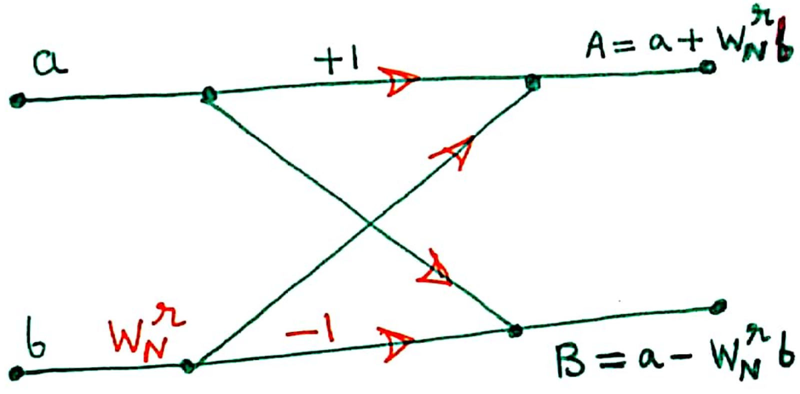
$$G_2[k] = W_N^k F_2[k] \qquad k = 0, 1, \ldots \frac{N}{2}-1$$

Hence, DFT $X[k]$ can be expressed as

$$X[k] = G_1[k] + G_2[k]$$

$$X[k+\tfrac{N}{2}] = G_1[k] - G_2[k]$$

Basic butterfly computation in decimation-in-time FFT algo.



$\Rightarrow$ Repeat decimation-in-time once more, $f_1(n)$ results in two $N/4$-point sequences as

$$V_{11}(n) = f_1(2n) \qquad n = 0, 1, \ldots, \frac{N}{4}-1$$

$$V_{12}(n) = f_1(2n+1) \qquad n = 0, 1, \ldots, N/4-1$$

and $f_2(n)$ results in two $N/4$-point sequences as

$$\left.\begin{aligned} V_{21}(n) &= f_2(2n) \\ V_{22}(n) &= f_2(2n+1) \end{aligned}\right\} \quad n = 0, 1, \ldots, N/4-1$$

Now $v_{ij}(n) \xleftrightarrow[N/4\text{-point}]{\text{DFT}} V_{ij}[k]$

Following the same butterfly computation procedure, it is clear that

$$F_1[k] = V_{11}[k] + W_{N/2}^{k} V_{12}[k]$$

$$F_1[k+\tfrac{N}{4}] = V_{11}[k] - W_{N/2}^{k} V_{12}[k]$$

$$k = 0, 1, \ldots, \tfrac{N}{4}-1$$

$$F_2[k] = V_{21}[k] + W_{N/2}^{k} V_{22}[k]$$

$$F_2[k+\tfrac{N}{4}] = V_{21}[k] - W_{N/2}^{k} V_{22}[k]$$

$$k = 0, 1, \ldots, \tfrac{N}{4}-1$$

By computing $\frac{N}{4}$-point DFTs, we can obtain $N/2$-point DFTs $F_1[k]$ & $F_2[k]$.

$\Rightarrow$ $V_{ij}[k]$ requires $4 \times \left(\frac{N}{4}\right)^2$ complex multiplications i.e., $N^2/4$

Now, computation of $F_1[k]$ and $F_2[k]$ needs complex multiplications in total as

$$4 \times \left(\frac{N}{4}\right)^2 + 2 \times \frac{N}{4} = \frac{N^2}{4}\left(1+\frac{2}{N}\right) = \left(\frac{N^2}{4} + \frac{N}{2}\right)$$

\# Consequently, the total no. of CM is reduced by about a factor of 2 again for large N, when we view the total no. of CM required to compute $X[k]$ from $F_1[k]$ and $F_2[k]$, that is

$$\text{Total no. of complex multiplications} = \frac{N^2}{4}\left(1+\frac{4}{N}\right) = \left(\frac{N^2}{4}+\frac{N}{2}\right)+\frac{N}{2}$$

$$= \left(\frac{N^2}{4}+N\right)$$

The decimation of the data sequence can be repeated again and again until the resulting sequences are reduced to one-point sequences.

For $N = 2^{\nu}$, the decimation can be performed $\nu = \log_2 N$ times.

Therefore, the total no. of complex multiplications is reduced to

$$(N/2) \log_2 N = \left(\frac{N}{2}\right)\nu$$

and, the total no. of complex additions is reduced to

$$(N) \log_2 N = N\nu$$

" In general, each butterfly involves one CM and two CAs.
For $N = 2^{\nu}$, there are $N/2$ butterflies per stage of the computation process and $\log_2 N$ stages."

Thanks for attending this session on DSP

Reference Books ⇒

⇒ "Digital Signal Processing: Principles, Algorithms and Applications," by J. G. Proakis and D. G. Manolakis; Prentice-Hall, 2007.

⇒ "Theory and Application of Digital Signal Processing," by L.R. Rabiner and B. Gold; Prentice-Hall, 2003.

⇒ "Discrete Time Signal Processing," by A.V. Oppenheim and R.W. Schafer; Prentice-Hall, 2001.

⇒ "Digital Signal Processing," by S.K. Mitra; Tata McGraw-Hill, 2006.

⇒ "Digital Signal Processing," by S. Salivahanan; McGraw-Hill, 2019.

⇒ "Digital Filters: Analysis and Design," by A. Antoniou; Tata McGraw-Hill, 1979.

⇒ "Digital Signal Processing," by E. Ifeachor and B.W. Jervis; Pearson Education, 2007.

⇒ "Signals and Systems," by A.V. Oppenheim, A.S. Willsky and S. H. Nawab; Prentice-Hall of India, 1999. (Prerequisite)

Thanks for attending this session on DSP

DIGITAL SIGNAL PROCESSING
UEC – 502

From My Lecture Notes

Lecture No. – 10

for

Electronics and Communication Engineering Students

&

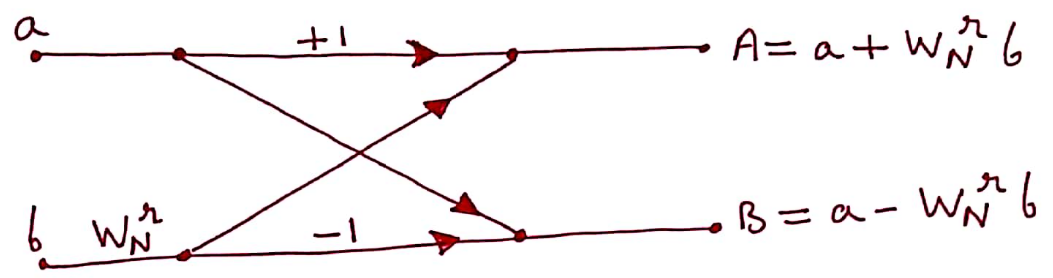
Electronics and Computer Engineering Students

Dr. Amit Kumar Kohli

Electronics and Communication Engineering Department

Thapar Institute of Engineering & Technology, Patiala, Punjab, India

Example of radix-2 (decimation-in-time) FFT algorithm for N=8

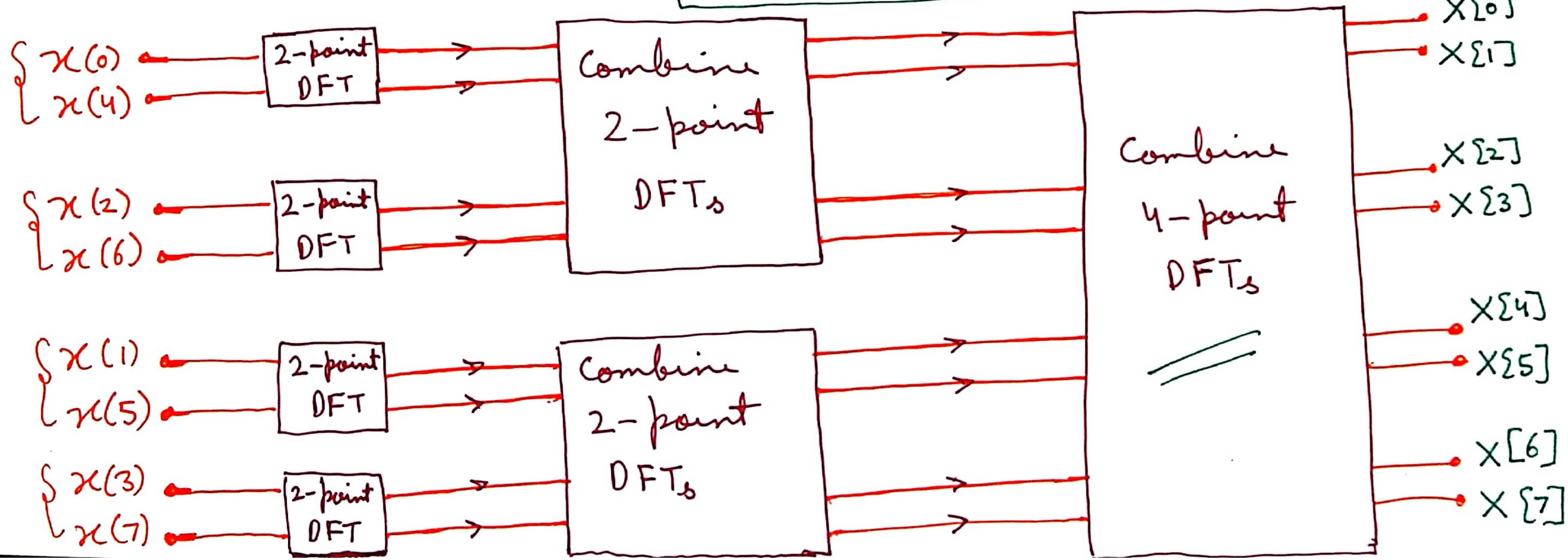


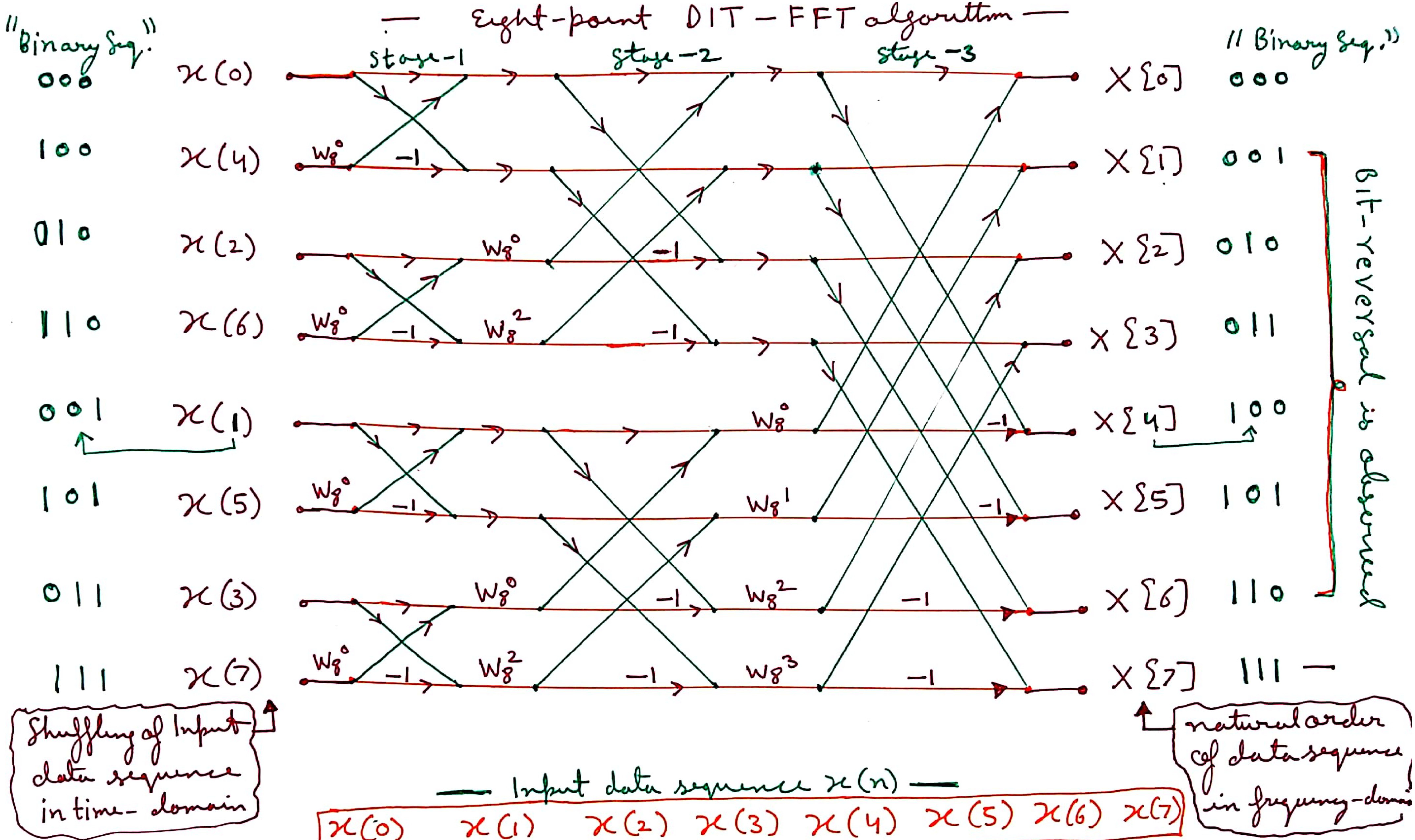
→ Basic butterfly computation in decimation-in-time FFT algorithm

⇓

→ $\boxed{\text{Three-stages}}$ in the computation of an N = 8 - point DFT

$$\boxed{v = \log_2 N = \log_2 8 = 3}$$





— Input data sequence $x(n)$ —

$x(0)$ $x(1)$ $x(2)$ $x(3)$ $x(4)$ $x(5)$ $x(6)$ $x(7)$

— FIRST DECIMATION YIELDS SEQUENCE —

$x(0)$ $x(2)$ $x(4)$ $x(6)$ $x(1)$ $x(3)$ $x(5)$ $x(7)$

— SECOND DECIMATION RESULTS IN SEQUENCE —

$x(0)$ $x(4)$ $x(2)$ $x(6)$ $x(1)$ $x(5)$ $x(3)$ $x(7)$

Vice-versa is also true

— Radix-2 FFT Algorithm With Decimation-in-Frequency Approach —

We begin by splitting DFT formula into two summations, of which one involves the sum over the first $(N/2)$ data points and the second the sum over the last $(N/2)$ data points.

i.e., choosing $M=2$ and $L=N/2$ in the divide-and-conquer approach.

therefore, $$X[k] = \sum_{n=0}^{N-1} x(n) W_N^{kn} \qquad \text{(general formula)}$$

$$= \sum_{n=0}^{N/2-1} x(n) W_N^{kn} + \sum_{n=N/2}^{N-1} x(n) W_N^{kn}$$

$$= \sum_{n=0}^{N/2-1} x(n) W_N^{kn} + \sum_{l=0}^{N/2-1} \underbrace{x(l+N/2)\, W_N^{k(l+N/2)}}$$

$\boxed{\text{if } n = l + N/2}$

$$= \sum_{n=0}^{N/2-1} x(n) W_N^{kn} + \sum_{l=0}^{N/2-1} \underbrace{x(l+N/2)\, W_N^{kl}}\, W_N^{kN/2}$$

$\boxed{W_N^{kN/2} = (-1)^k}$

$$= \sum_{n=0}^{N/2-1} x(n) W_N^{kn} + (-1)^k \sum_{l=0}^{N/2-1} \underbrace{x(l+N/2)\, W_N^{kl}}$$

Without any loss we can replace l with variable n

$$X[k] = \sum_{n=0}^{N/2-1}\left[x(n) + (-1)^k x(n+N/2)\right] W_N^{kn}$$

$\Rightarrow$ Split (decimate) $X[k]$ into even and odd numbered samples

$$X[2k] = \sum_{n=0}^{N/2-1}\left[x(n) + x(n+N/2)\right] W_{N/2}^{kn} \qquad k = 0, 1, ---, N/2-1$$

$$X[2k+1] = \sum_{n=0}^{N/2-1}\left\{\left[x(n) - x(n+N/2)\right] W_N^{n}\right\} W_{N/2}^{kn}$$

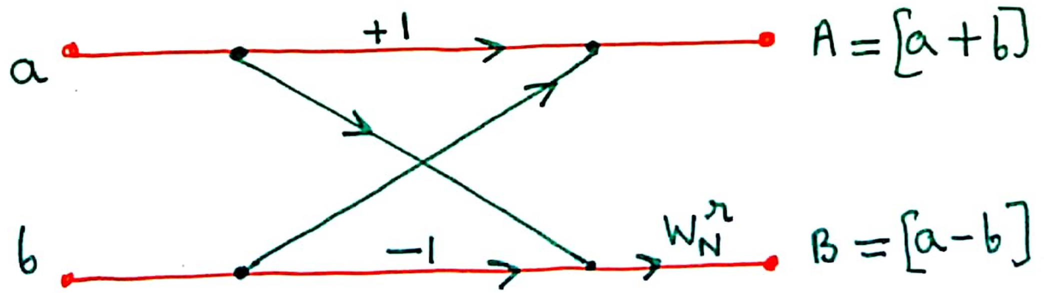
If we define $N/2$-point sequences $g_1(n)$ and $g_2(n)$ as

$$g_1(n) = x(n) + x(n+N/2) \quad \text{and} \quad g_2(n) = \left[x(n) - x(n+N/2)\right] W_N^{n}$$

$n = 0, 1, 2 - - \frac{N}{2} - 1$

then

$$X[2k] = \sum_{n=0}^{N/2-1} g_1(n) W_{N/2}^{kn} \quad \text{and} \quad X[2k+1] = \sum_{n=0}^{N/2-1} g_2(n) W_{N/2}^{kn}$$



"Basic butterfly computation in the decimation-in-frequency FFT algorithm"

This computation procedure can be repeated through the decimation of $N/2$-point DFTs $X[2k]$ and $X[2k+1]$.

⇒ The entire process involves $\nu = \log_2 N$ stages of decimation, where each stage involves $N/2$ butterflies.

⇒ Thus computation of N-point DFT via decimation-in-frequency FFT algorithm requires

$1 \times \frac{N}{2} \log_2 N \rightarrow$ Complex multiplications ✓ $= \frac{N}{2} \log_2 N$

$2 \times \frac{N}{2} \log_2 N \rightarrow$ Complex additions ✓ $= N \log_2 N$

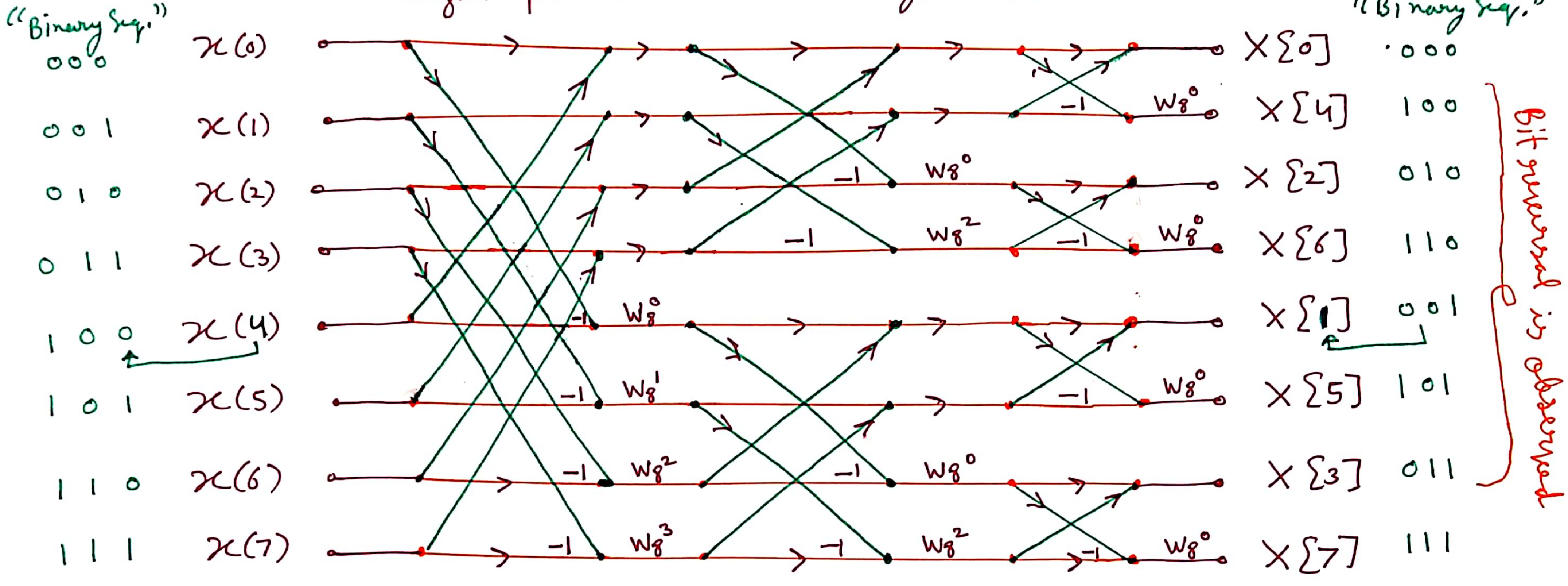
" Its computational complexity is similar to DIT–FFT "

Example of radix-2 (decimation-in-frequency) FFT algorithm for N=8

⇒ Here, we shall observe that the time-domain input data sequence $x(n)$ occurs in the natural order, but output DFT occurs in bit-reversed order.

⇒ It is also possible to reconfigure DIF–FFT algorithm with output DFT in natural order, and input sequence in bit-reversed order. ⇒

— Eight-point DIF-FFT algorithm —



natural order of data sequence in time-domain

Shuffling of output sequence in frequency-domain

Vice-Versa is also true

Brief details about Radix-4 FFT algorithms →

$$N = r^{\nu} \quad \text{with} \quad r = 4$$

⇒ Radix-4 FFT algorithm is found to be more efficient than Radix-2 algo.

For Radix-4 decimation-in-time FFT algorithm, we select $L=4$ and $M=N/4$ in the divide-and-conquer approach.

$$X[p,q] = \sum_{l=0}^{3} \left[W_N^{lq} F(l,q) \right] W_4^{lp} \qquad \begin{matrix} p = 0,1,2,3 \\ m = 0,1,2,\ldots, \frac{N}{4}-1 \end{matrix} \rightarrow \text{(A)}$$

where,

$$F(l,q) = \sum_{m=0}^{N/4-1} x(l,m)\, W_{N/4}^{mq} \quad (N/4\text{-point DFT}) \qquad \begin{matrix} l = 0,1,2,3 \\ q = 0,1,2,\ldots, N/4-1 \end{matrix} \rightarrow \text{(B)}$$

➡ $n = 4m + l$ ✓ $\quad k = (N/4)p + q$ ✓ $\Rightarrow$ $x(l,m) = x(4m+l)$, $X(p,q) = X[\frac{N}{4}p + q]$ through mapping

➡ We split or decimate N-point input sequence into four subsequences

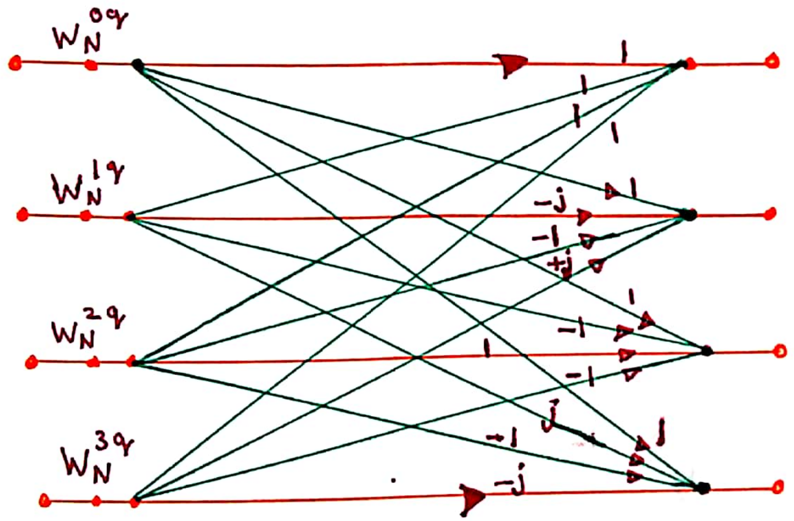
$$x(4n),\ x(4n+1),\ x(4n+2),\ x(4n+3)$$

$$n = 0,1,\ldots\ldots, N/4-1$$

➡ Four N/4-point DFTs (B) are combined according to (A) to generate N-point DFT

The equation (A) defines a radix-4 decimation-in-time butterfly, in matrix form as

$$\begin{bmatrix} X(0,q) \\ X(1,q) \\ X(2,q) \\ X(3,q) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & +j \\ 1 & -1 & +1 & -1 \\ 1 & +j & -1 & -j \end{bmatrix} \begin{bmatrix} W_N^0 F(0,q) \\ W_N^q F(1,q) \\ W_N^{2q} F(2,q) \\ W_N^{3q} F(3,q) \end{bmatrix}$$



— "Each butterfly involves 03 CMs and 12 CAs" —

— FFT (DIT) consists of ν stages

— Each stage contains $N/4$ butterflies

— Total/Net computational burden of FFT(DIT) includes

$$CMs = 3\nu N/4 = \frac{3N}{8}\log_2 N$$

$$CAs = 12\nu N/4 = \frac{3}{2}N\log_2 N$$

$$\Rightarrow 4^\nu = N \Rightarrow 2^{2\nu} = N \Rightarrow \log_2 N = 2\nu$$

$$\Rightarrow \nu = \frac{1}{2}\log_2 N$$

However,
the computation of N-point DFT through radix-2 FFT algorithm involves

$\Rightarrow$ Total $CM_s = N/2 \log_2 N$

$\Rightarrow$ Total $CA_s = N \log_2 N$

"It can be inferred that if we use radix-4 FFT algorithm instead of radix-2 FFT algorithm, then the number of complex multiplications is reduced by 25%, but the number of complex additions has increased by 50%."

Thanks for attending this session on DSP

Reference Books ⇒

⇒ "Digital Signal Processing: Principles, Algorithms and Applications," by J. G. Proakis and D. G. Manolakis; Prentice-Hall, 2007.

⇒ "Theory and Application of Digital Signal Processing," by L.R. Rabiner and B. Gold; Prentice-Hall, 2003.

⇒ "Discrete Time Signal Processing," by A.V. Oppenheim and R.W. Schafer; Prentice-Hall, 2001.

⇒ "Digital Signal Processing," by S.K. Mitra; Tata McGraw-Hill, 2006.

⇒ "Digital Signal Processing," by S. Salivahanan; McGraw-Hill, 2019.

⇒ "Digital Filters: Analysis and Design," by A. Antoniou; Tata McGraw-Hill, 1979.

⇒ "Digital Signal Processing," by E. Ifeachor and B.W. Jervis; Pearson Education, 2007.

⇒ "Signals and Systems," by A.V. Oppenheim, A.S. Willsky and S. H. Nawab; Prentice-Hall of India, 1999. (Prerequisite)

Thanks for attending this session on DSP