INTRODUCTION: T - [1-6]

Digital signal processing is an area of science and engineering that has developed rapidly over the past 40 years. This rapid development is a result of the significant advances in digital computer technology and integrated - circuit fabrication. The rapid developments in IC technology starting with Medium Scale Integration (MSI) and progressing to Large Scale Integration (LSI) and now Very-Large Scale Integration (VLSI) of electronic circuits has spurred the development of powerful, smaller, faster and cheaper digital computers and special-purpose digital hardware.

These digital circuits have made it possible to construct highly sophisticated digital systems capable of performing complex digital signal processing functions. Many of the signal processing tasks that were conventionally performed by analog means are realized today by less expensive and often more reliable digital hardware.

SIGNALS:-

A signal is defined as any physical quantity that varies with time, space or any other independent variable (or) variables. It is described mathematically as a function of one (or) more independent variables.

Ex: $s_1(t) = 5t$

$s_2(t) = 20t^2$

$s(x,y) = 3x + 2xy + 10y^2$

System:-

A system is defined as a physical device (or software realizations) that performs an operation on a signal.

Ex: A filter used to reduce the noise and interference corrupting a desired information bearing signal is called a system.

SIGNAL PROCESSING:-

In general, a system is characterised by the type of operation that it performs on the signal. Such operations are usually referred to as signal processing.

BASIC ELEMENTS OF DIGITAL SIGNAL PROCESSING SYSTEM

Most of the signals encountered in science and engineering are analog in nature. To process these analog signals, appropriate analog systems are used. The illustration of analog processing system is shown below.

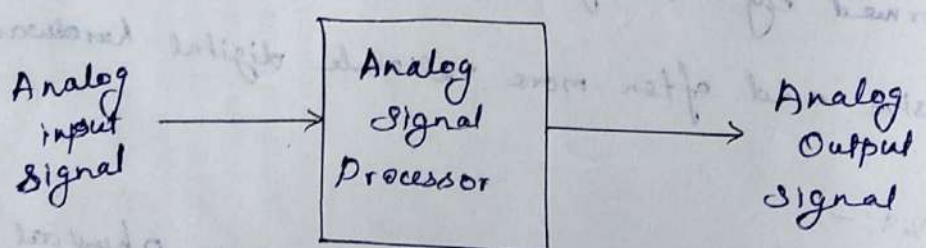


Figure 1.1 Analog signal processing System

Processing of analog signals may include changing their characteristics (or) extracting their desired information. Here both the input and output are in analog form.

(3) Digital signal processing provides an alternative method for processing the analog signal as shown below in figure 1.2.

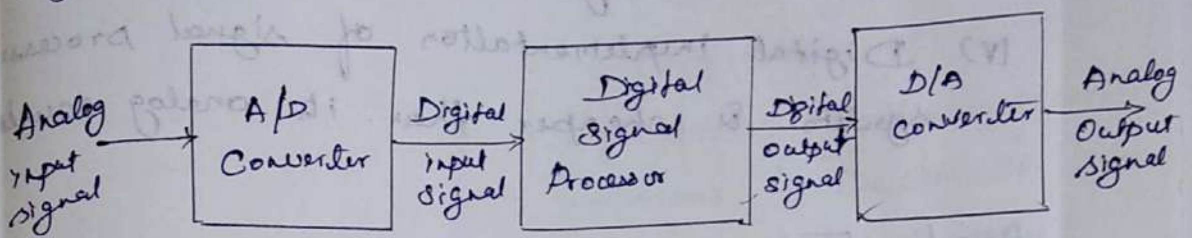


Figure 1.2 Digital Signal Processing System

To process analog signal in digital processor, an interface called A/D converter is to be used. The output of the A/D converter is a digital signal that is appropriate as an input to the digital processor.

The digital signal processor may be a large programmable digital computer or a small microprocessor programmed to perform the desired operations on the input signal.

In applications like speech communications where the digital output from the processor is to be given to the user in analog form, another interface called D/A converter is used.

ADVANTAGES OF DIGITAL OVER ANALOG SIGNAL PROCESSING

(i) A digital programmable system allows flexibility in reconfiguring the digital signal processing operations simply by changing the program.

(ii) Digital systems are more accurate

(iii) Digital signals can easily be stored on magnetic media without loss of signal fidelity.

(iv) Implementation of more sophisticated signal processing algorithms is easy in digital signal processing.

(v) Digital implementation of signal processing system is cheaper than its analog counterpart.

Applications:-

Used in

(i) Speech processing & signal transmission on telephone channels.

(ii) Image processing and transmission

(iii) Seismology and geophysics

(iv) In oil exploration

(v) The detection of nuclear explosions

(vi) processing of signals received from outer space

(vii) Vast variety of other applications.

Limitations:-

(i) Speed of operation of A/D converters and digital signal processors.

(ii) Signals having wide range of bandwidth require fast sampling rate A/D converters and fast digital signal processors.

UNIT - I

DISCRETE FOURIER TRANSFORM

Discrete Signals and Systems - A Review Introduction to DFT - properties of DFT - Circular convolution - Filtering methods based on DFT - FFT algorithms - Decimation in time Algorithms, Decimation in frequency Algorithms - use of FFT in linear filtering.

T - (41-67)

DISCRETE SIGNALS AND SYSTEMS - A REVIEW

Signals that are defined only at certain specific values of time are called discrete-time signals. These time instants need not be equidistant, but in practice they are usually taken at equally spaced intervals for computational convenience and mathematical tractability.

A discrete-time signal having a set of discrete values is called a digital signal.

ie, A signal which is discrete in time and amplitude is called digital signal.

Elementary Discrete-Time Signals :-

(1) Unit sample sequence (or) Unit impulse

$$\delta(n) = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases}$$

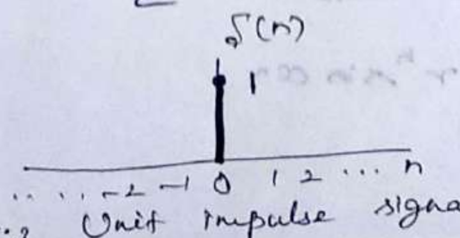


Figure 1.3 Unit impulse signal

② Unit step signal

$$u(n) = \begin{cases} 1, & n \geq 0 \\ 0, & n < 0 \end{cases}$$

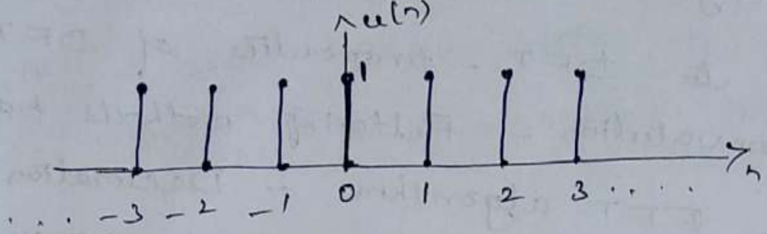


Figure 1.4 unit step signal

③ Unit Ramp signal

$$u_r(n) = \begin{cases} n & n \geq 0 \\ 0 & n < 0 \end{cases}$$

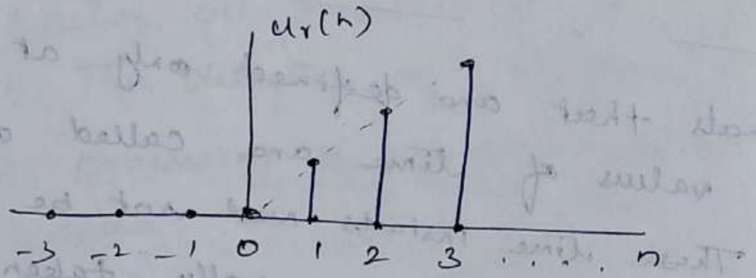


Figure 1.5 Unit ramp signal

④ Exponential signal

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

if a is real then x(n) is a real signal.

When a complex valued then

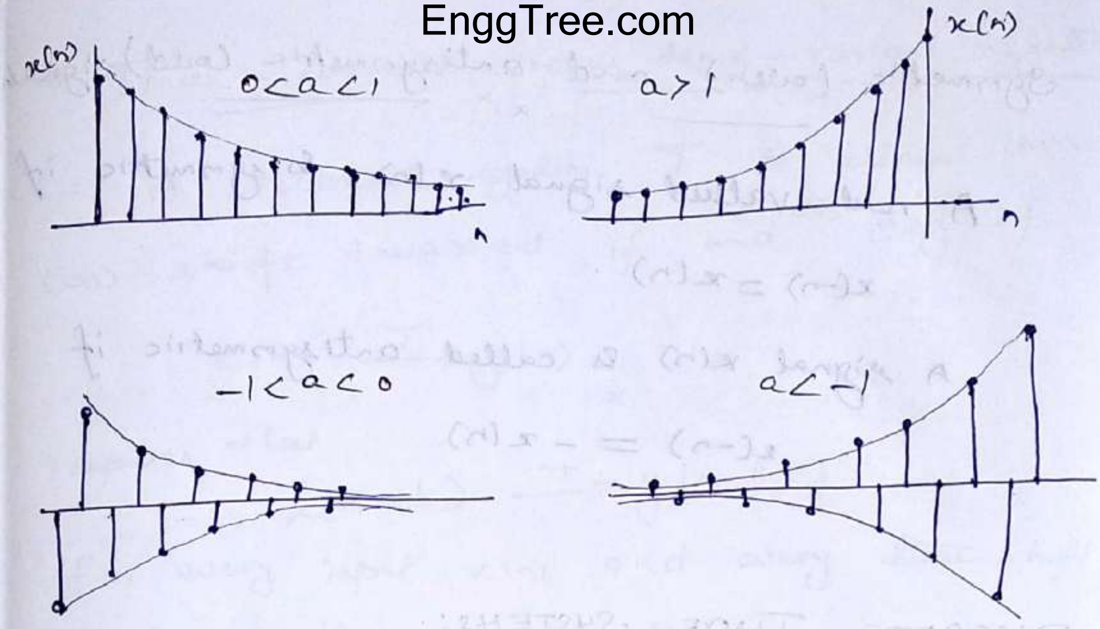
$$a = re^{j\theta}$$

$$x(n) = r^n e^{j\theta n}$$

$$x(n) = r^n (\cos\theta n + j\sin\theta n) \rightarrow \text{complex value}$$

ie $x_R(n) = r^n \cos\theta n$

$$x_I(n) = r^n \sin\theta n$$



Classification of Discrete -Time signals:-

* Energy signals and Power signals

Energy signals $E = \sum_{n=-\infty}^{\infty} |x(n)|^2$

If E is finite, P=0

Power Signals $P = \lim_{N\to\infty} \frac{1}{2N+1} \sum_{n=-N}^{N} |x(n)|^2$

$= \lim_{N\to\infty} \frac{1}{2N+1} E_N$

where $E_N \to$ signal energy over $-N \leq n \leq N$

If E is ∞, the avg. power may be finite (or) infinite.

* Periodic signals and aperiodic signals

A signal $x(n)$ is periodic with period N $(N>0)$ if and only if

$x(n+N) = x(n) \quad \forall n$

If there is no value of N that satisfies the above equation, the signal is called nonperiodic (or) aperiodic.

Symmetric (even) and antisymmetric (odd) signals:-

A real valued signal $x(n)$ is symmetric if

$$x(-n) = x(n).$$

A signal $x(n)$ is called antisymmetric if

$$x(-n) = -x(n)$$

DISCRETE TIME SYSTEMS:

A discrete-time system is a device (or) algorithm that operates on a discrete-time signal called the input (or) excitation, according to some well defined rule, to produce another another discrete time signal called the output (or) response of the system.

Classification of Discrete-Time Systems:

*** Static Versus dynamic Systems:-**

A discrete-time system is called static (or) memoryless if its output at any instant n depends at most on the input sample at the same time, but not on past (or) future samples of the input.

A system is said to be dynamic, if it has a memory.

* Time - invariant versus time - variant systems.

A relaxed system T is time invariant (or) shift invariant if and only if

$$x(n) \xrightarrow{T} y(n)$$

implies that

$$x(n-k) \xrightarrow{T} y(n-k)$$

for every input x(n) and every time shift k.

* Linear versus Non-linear:

A system is linear if and only if

$$T\left[a_1 x_1(n) + a_2 x_2(n)\right] = a_1 T[x_1(n)] + a_2 T[x_2(n)]$$

for any arbitrary input sequence $x_1(n)$ and $x_2(n)$ and any arbitrary constant a_1 and a_2.

* Causal versus Non causal systems

A system is said to be causal if the output of the system at any time n depends only on present and past inputs but not depends on future inputs.

ie $y(n) = F[x(n), x(n-1), x(n-2) \dots]$

* Stable versus Unstable systems

An arbitrary relaxed system is said to be bounded input - bounded output (BIBO) stable if and only if every bounded input produces a bounded output.

DISCRETE FOURIER TRANSFORM

T1 - (454 - 456)

The Discrete Fourier Transform (DFT) of a signal $x(n)$ is given by

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

The inverse DFT (IDFT) is given by

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

DFT is the frequency domain sampling of a signal $x(n)$ of length N.

① Compute the DFT of the sequence $x(n) = \{1, -1, 1, -1\}$ (A/M-2017-R13)

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

The N-point DFT of the sequence $x(n)$ is given by,

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

$$X(0) = \sum_{n=0}^{3} x(n)\, e^{-j\frac{2\pi(0)n}{4}}$$

$$= 1 - 1 + 1 - 1$$

$$X(0) = 0$$

$$X(1) = \sum_{n=0}^{3} x(n)\, e^{-j\frac{2\pi(1)n}{4}}$$

$$= x(0)\, e^{-j\frac{\cancel{2}\pi(0)}{\cancel{4}_2}} + x(1)\, e^{-j\frac{\cancel{2}\pi(1)}{\cancel{4}_2}} + x(2)\, e^{-j\frac{\cancel{2}\pi(2)}{\cancel{4}_2}}$$

$$+ x(3)\, e^{-j\frac{\cancel{2}\pi(3)}{\cancel{4}_2}}$$

$x(1) = 1(0) + (-1)(-1) + (1)(0) + (-1)(1)$

$X(1) = 1 + (-1)(0-j) + 1(-1) - 1(0+j)$

$= 1 + j - 1 - j$

$X(1) = 0$

$$X(2) = \sum_{n=0}^{3} x(n) e^{\frac{-j\cancel{2}\pi(2)n}{\cancel{4}}}$$

$$= x(0)e^{-j\pi(0)} + x(1)e^{-j\pi} + x(2)e^{-j2\pi} + x(3)e^{-j3\pi}$$

$= 1 - 1(-1) + 1(1) - 1(-1)$

$= 1 + 1 + 1 + 1$

$X(2) = 4$

$$X(3) = \sum_{n=0}^{3} x(n) e^{\frac{-j\cancel{2}\pi(3)n}{\cancel{4}2}}$$

$= 1(1) - 1(0+j) + 1(-1) - 1(0-j)$

$= 1 - j - 1 + j$

$= 0$

$$\therefore \boxed{X(k) = \{0, 0, 4, 0\}}$$

2). Find the 4-point DFT of the sequence

$x(n) = \{1, 1, -1, -1\}$ (A/M-17 R-08) (M/J-13 R-08)

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

The N-point DFT of the sequence x(n) is given by

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

$$X(k) = \sum_{n=0}^{3} x(n) e^{-j\frac{2\pi kn}{4}} \qquad k=0,1,2,3$$

$$X(0) = \sum_{n=0}^{3} x(n)$$

$$= 1+1-1-1$$

$$= 0$$

$$X(1) = \sum_{n=0}^{3} x(n) e^{-j\frac{\cancel{2}\pi(1)n}{\cancel{4}2}}$$

$$= 1(1) + 1(0-j) - 1(-1) - 1(0+j)$$

$$= 1 + 1(-j) + 1 - j$$

$$= 2 - 2j$$

$$X(2) = \sum_{n=0}^{3} x(n) e^{-j\frac{\cancel{2}\pi(\cancel{2})n}{\cancel{4}}}$$

$$= 1(1) + 1(-1) - 1(1) - 1(-1)$$

$$= 1 - 1 - 1 + 1$$

$$= 0$$

$$X(3) = \sum_{n=0}^{3} x(n) e^{-j\frac{\cancel{2}\pi(3)n}{\cancel{4}2}}$$

$$= 1(1) + 1(0+j) - 1(-1) - 1(0-j)$$

$$= 1 + j + 1 + j$$

$$= 2 + 2j$$

$$\therefore \boxed{X(k) = \{0, 2-2j, 0, 2+2j\}}$$

3). Find the IDFT of $X(k) = \{1, 2, 3, 4\}$

Given $X(k) = \{1, 2, 3, 4\}$

The IDFT of the sequence $X(k)$ is given by

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

$$x(n) = \frac{1}{4} \sum_{k=0}^{3} X(k) e^{j\frac{2\pi kn}{4}}$$

$$x(0) = \frac{1}{4} \sum_{k=0}^{3} X(k)$$

$$= (1+2+3+4)\frac{1}{4}$$

$$x(0) = \frac{5}{2}$$

$$x(1) = \frac{1}{4} \sum_{k=0}^{3} X(k) e^{j\frac{2\pi k(1)}{4}}$$

$$= \frac{1}{4}\left[1 + 2j + 3(-1) + 4(-j)\right]$$

$$x(1) = -\frac{1}{2} - j\frac{1}{2}$$

$$x(2) = \frac{1}{4} \sum_{k=0}^{3} X(k) e^{j\pi k}$$

$$= \frac{1}{4}(1 - 2 + 3 - 4)$$

$$x(2) = -\frac{1}{2}$$

$$x(3) = \frac{1}{4} \sum_{k=0}^{3} X(k) e^{j3\pi k/2}$$

$$= \frac{1}{4}\left[1 + 2(-j) + 3(-1) + 4j\right]$$

$$x(3) = -\frac{1}{2} + j\frac{1}{2}$$

$$\therefore \boxed{x(n) = \left\{\frac{5}{2}, -\frac{1}{2} - j\frac{1}{2}, -\frac{1}{2}, -\frac{1}{2} + j\frac{1}{2}\right\}}$$

Relationship of the DFT to other transforms

(1) Relationship to the Fourier series coefficients of a periodic sequence

A periodic sequence $[x_p(n)]$ with fundamental period N can be represented in a Fourier series of the form

$$x_p(n) = \sum_{k=0}^{N-1} c_k e^{j\frac{2\pi kn}{N}} \qquad -\infty < n < \infty$$

where Fourier series co-efficients are given by

$$c_k = \frac{1}{N} \sum_{n=0}^{N-1} x_p(n) e^{-j\frac{2\pi nk}{N}} \qquad k=0,1,\dots N-1$$

Comparing these equations with DFT and IDFT equations,

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

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

then $X(k) = N c_k$ and

$x_p(n)$ is same as IDFT.

ie, The N-point DFT provides the exact line spectrum of a periodic sequence with fundamental period N.

② Relationship to Fourier Transform of an aperiodic sequence. (DTFT)

If $x(n)$ is an aperiodic finite energy sequence with Fourier Transform $X(\omega)$ which is sampled at N equally spaced frequencies

$\omega_k = \frac{2\pi k}{N}$, $k = 0, 1, \ldots N-1$ the spectral components

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

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

are the DFT coefficients of the periodic sequence of period N, given by

$$x_p(n) = \sum_{l=-\infty}^{\infty} x(n - lN)$$

③ Relationship to z-transform

The z transform of a sequence $x(n)$ is

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

with ROC that includes the unit circle.

If $X(z)$ is sampled at the N equally spaced points on the unit circle $z_k = e^{j\frac{2\pi k}{N}}$ then

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

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

Fourier transform of the finite duration sequence in terms of its DFT is

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

This expression for the FT is a polynomial (Lagrange) interpolation formula for $X(\omega)$ expressed in terms of the values $\{X(k)\}$ of the polynomial at a set of equally spaced discrete frequencies $\omega_k = \frac{2\pi k}{N}$, $k = 0, 1, \ldots N-1$.

(4) Relationship to the Fourier series co-efficients of a continuous-time signal

Suppose that $x_a(t)$ is a continuous-time periodic signal with fundamental period $T_p = \frac{1}{F_0}$. The signal can be expressed in a Fourier series

$$x_a(t) = \sum_{k=-\infty}^{\infty} c_k e^{j2\pi k t F_0}$$

If $x_a(t)$ is sampled at a uniform rate $F_s = \frac{N}{T_p} = \frac{1}{T}$ then

$$x(n) \equiv x_a(nT) = \sum_{k=-\infty}^{\infty} c_k e^{j2\pi k F_0 n T} \quad \text{— (1)}$$

$$= \sum_{k=-\infty}^{\infty} c_k e^{j\frac{2\pi k n}{N}}$$

$$= \sum_{k=0}^{N-1} \left[\sum_{l=-\infty}^{\infty} c_{k-lN} \right] e^{j\frac{2\pi k n}{N}}$$

eqn (1) is in the form of IDFT where

$$X(k) = N \sum_{l=-\infty}^{\infty} c_{k-lN} \equiv N\tilde{c}_k$$

$$\tilde{c}_k = \sum_{l=-\infty}^{\infty} c_{k-lN}$$

$\{\tilde{c}_k\}$ sequence is a aliased version of the sequence $\{c_k\}$

PROPERTIES OF DFT

The properties of DFT are,

1) periodicity
2) Linearity
3) Circular symmetries of a sequence
4) Symmetry properties of DFT
5). Multiplication of two DFTs and circular convolution
6). Time reversal of a sequence
7). Circular time shift of a sequence
8). Circular frequency shift
9). Complex conjugate properties
10) Circular correlation
11). Multiplication of two sequences
12) Parseval's Theorem

1) Periodicity :-

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

$$x(n+N) = x(n) \quad \forall n$$

$$X(k+N) = X(k) \quad \forall k$$

Proof

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

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

$$= \underbrace{\sum_{n=0}^{N-1} x(n)\, e^{-j\frac{2\pi kn}{N}}}_{}\, \underbrace{e^{-j\frac{2\pi Nn}{N}}}_{=1 \text{ for all values of } n}$$

$$\boxed{X(k+N) = X(k)}$$

② . Linearity.

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

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

then for any real-valued (or) complex-valued constants a_1 and a_2,

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

③ Circular symmetries of a sequence

The N-point DFT of a finite duration sequence $x(n)$, of length $L \le N$, is equivalent to the N-point DFT of a periodic sequence $x_p(n)$, of period N, which is obtained by periodically extending $x(n)$,

ie $$x_p(n) = \sum_{l=-\infty}^{\infty} x(n - lN)$$

When this $x_p(n)$ sequence is shifted by k units to the right, then another periodic sequence is obtained.

ie, $$x_p'(n) = x_p(n-k) = \sum_{l=-\infty}^{\infty} x(n-k-lN)$$

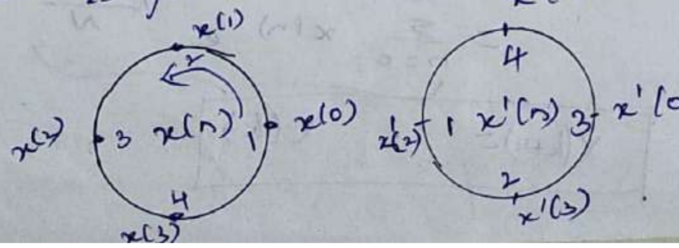
The finite duration sequence

$$x'(n) = \begin{cases} x_p'(n) & 0 \le n \le N-1 \\ 0 & \text{otherwise} \end{cases}$$

is related to the original sequence $x(n)$ by a circular shift.

Ex

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



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

Circular shift of a sequence can be represented as the index modulo N.

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

$x'(n) \equiv x((n-k))_N$

* An N-point sequence is called circularly even if it is symmetric about the point zero on the circle.

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

* An N-point sequence is called circularly odd if it is antisymmetric about the point zero on the circle.

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

* Time reversal of an N-point sequence is attained by reversing its samples about the point zero on the circle. ∴ The sequence $x((-n))_N$ is given as,

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

(4) Symmetry properties of the DFT:-

Let the N-point sequence x(n) and its DFT are both complex valued.

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

$X(k) = X_R(k) + j\, X_I(k) \qquad 0 \le k \le N-1$

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

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

(i) Real-valued sequence:-

If the sequence $x(n)$ is real, then

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

$$|X(N-k)| = |X(k)| \quad \text{and} \quad \angle X(N-k) = -\angle X(k)$$

(a) Real and even sequence:-

If $x(n)$ is real and even,

ie, $x(n) = x(N-n), \quad 0 \le n \le N-1$

then $X_I(k) = 0$

$$\therefore X(k) = \sum_{n=0}^{N-1} x(n) \cdot \cos\frac{2\pi kn}{N}, \quad 0 \le k \le N-1$$

which is real valued and even.

The IDFT is, $[X_I(k) = 0]$

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k)\cos\frac{2\pi kn}{N}, \quad 0 \le n \le N-1$$

b). Real and odd sequence:-

If $x(n)$ is real and odd,

ie, $x(n) = -x(N-n) \quad 0 \le n \le N-1$

then $X_R(k) = 0$.

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

which is purely imaginary and odd.

Since $X_R(k) = 0$,

the IDFT is

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

21

(ii) Purely imaginary sequence:-

If $x(n)$ is purely imaginary,

ie $x(n) = j\, x_I(n)$

then $X_R(k) = \sum_{n=0}^{N-1} x_I(n) \sin \frac{2\pi kn}{N}$

$X_I(k) = \sum_{n=0}^{N-1} x_I(n) \cos \frac{2\pi kn}{N}$

$X_R(k)$ is odd and $X_I(k)$ is even.

If $x_I(n)$ is odd, then $X_I(k) = 0$

$\Rightarrow X(k)$ is purely real

If $x_I(n)$ is even, then $X_R(k) = 0$

$\Rightarrow X(k)$ is purely imaginary.

Symmetry properties - summary

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

$$X(k) = X_R^e(k) + X_R^o(k) + j\, X_I^e(k) + j\, X_I^o(k)$$

5). Multiplication of Two DFTs and Circular Convolution

Suppose that there are two finite -duration sequences of length N, $x_1(n)$ and $x_2(n)$, their respective N-point DFTs are

$$X_1[k] = \sum_{n=0}^{N-1} x_1(n) e^{-j\frac{2\pi nk}{N}}, \quad k=0,1,\ldots N-1 \longrightarrow (1)$$

$$X_2[k] = \sum_{n=0}^{N-1} x_2(n) e^{-j\frac{2\pi nk}{N}} \quad k=0,1,\ldots N-1 \longrightarrow (2)$$

If $X_1[k]$ and $X_2[k]$ are multiplied then,

$$X_3[k] = X_1(k) X_2(k) \qquad k=0,1\ldots N-1$$

The IDFT of $\{X_3[k]\}$ is

$$x_3(m) = \frac{1}{N}\sum_{k=0}^{N-1} X_3[k] e^{j\frac{2\pi km}{N}}$$

$$= \frac{1}{N}\sum_{k=0}^{N-1} X_1[k] X_2[k] e^{j\frac{2\pi km}{N}}$$

by substituting (1) and (2),

$$x_3(m) = \frac{1}{N}\sum_{k=0}^{N-1}\left[\sum_{n=0}^{N-1} x_1(n) e^{-j\frac{2\pi nk}{N}}\right]\left[\sum_{l=0}^{N-1} x_2(l) e^{-j\frac{2\pi lk}{N}}\right] e^{j\frac{2\pi km}{N}}$$

$$x_3(m) = \frac{1}{N}\sum_{n=0}^{N-1} x_1(n)\sum_{l=0}^{N-1} x_2(l)\left[\sum_{k=0}^{N-1} e^{j\frac{2\pi k(m-n-l)}{N}}\right] \longrightarrow (3)$$

The inner sum in the brackets has the form

$$\sum_{k=0}^{N-1} a^k = \begin{cases} N, & a=1 \\ \frac{1-a^N}{1-a} & a\neq 1 \end{cases} \qquad a = e^{j\frac{2\pi(m-n-l)}{N}}$$

$a = 1$ when $m-n-l$ is a multiple of N.

and $a^N = 1$ for any value of $a \neq 0$.

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

substituting this in eq (3),

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

This is in the form of a convolution sum.

The convolution sum involves the index $((m-n))_N$ and is called circular convolution.

Thus,

The multiplication of the DFTs of two sequences is equivalent to the circular convolution of the two sequences in the time domain.

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

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

then, $\quad x_1(n) \, (N) \, x_2(n) \xleftrightarrow[N]{DFT} X_1(k)\, X_2(k)$

(6) Time reversal of a sequence :-

Let $x(n) \xleftrightarrow[N]{DFT} X(k)$

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

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

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

if the index is changed from n to $m=N-n$

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

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

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

$= X(N-k)$

$X(N-k) = X((-k))_N \quad 0 \le k \le N-1.$

7). Circular time shift of a sequence :-

$x(n) \xleftrightarrow[N]{DFT} X(k)$

then $x((n-l))_N \xleftrightarrow[N]{DFT} X(k) e^{-j2\pi kl/N}$

Proof $DFT\{x((n-l))_N\} = \sum_{n=0}^{N-1} x((n-l))_N e^{-j\frac{2\pi kn}{N}}$

(25)

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

But $x((n-l))_N = x(N-l+n)$. Consequently,

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

$$= \sum_{m=N-l}^{N-1} x(m) \, e^{-j\frac{2\pi k(m+l)}{N}}$$

and

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

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

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

8). Circular frequency shift :-

$$x(n) \xleftrightarrow[N]{DFT} X(k)$$

then $$x(n) \, e^{j\frac{2\pi ln}{N}} \xleftrightarrow[N]{DFT} X((k-l))_N$$

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

It is the dual of the circular time shifting property.

9). Complex-conjugate properties:

$$x(n) \xleftrightarrow[N]{DFT} X(k)$$

then

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

The IDFT of $X^*(k)$ is,

$$\frac{1}{N}\sum_{k=0}^{N-1} X^*(k)\, e^{j\frac{2\pi kn}{N}} = \left[\frac{1}{N}\sum_{k=0}^{N-1} X(k)\, e^{\frac{j2\pi k(N-n)}{N}}\right]$$

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

10). Circular correlation

For complex-valued sequences $x(n)$ and $y(n)$,

if $x(n) \xleftrightarrow[N]{DFT} X(k)$

and $y(n) \xleftrightarrow[N]{DFT} Y(k)$

then $\tilde{r}_{xy}(l) \xleftrightarrow[N]{DFT} \tilde{R}_{xy}(k) = X(k)\, Y^*(k)$

where $\tilde{r}_{xy}(l)$ is the unnormalized circular cross correlation sequence, defined as,

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

Proof

$$\tilde{r}_{xy}(l) = x(l) \; (N) \; y^*(-l)$$

the N-point DFT of $\tilde{r}_{xy}$ is

$$\tilde{R}_{xy}(k) = X(k)\, Y^*(k)$$

when $y(n) = x(n)$ then,

$$\tilde{r}_{xx}(l) \xleftrightarrow[N]{DFT} R_{xx}(k) = |X(k)|^2$$

Called as circular autocorrelation of $x(n)$.

11). Multiplication of two sequences

$$x_1(n) \xleftrightarrow[N]{DFT} X_1(k)$$

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

then $x_1(n)\, x_2(n) \xleftrightarrow[N]{DFT} \frac{1}{N} X_1(k) \, \textcircled{N} \, X_2(k)$

This property is the dual of circular convolution property.

12). Parseval's Theorem:

For complex valued sequences $x(n)$ and $y(n)$

if $x(n) \xleftrightarrow[N]{DFT} X(k)$

and $y(n) \xleftrightarrow[N]{DFT} Y(k)$

then $$\sum_{n=0}^{N-1} x(n)\, y^*(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)\, Y^*(k)$$

Proof: From circular correlation property

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

and $$\tilde{r}_{xy}(l) = \frac{1}{N} \sum_{k=0}^{N-1} \tilde{R}_{xy}(k)\, e^{j\frac{2\pi k l}{N}}$$

$$= \frac{1}{N} \sum_{k=0}^{N-1} X(k)\, Y^*(k)\, e^{j\frac{2\pi k l}{N}} \quad \text{--- (1)}$$

By evaluating IDFT at $l=0$, the eqn ⑩ can be obtained. The eqn ⑩ is called the general form of the Parseval's theorem.

When $y(n) = x(n)$ then

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

which is the energy in the finite-duration sequence $x(n)$ in terms of the frequency components $\{X(k)\}$.

4) Find the linear and circular convolution of the two sequence $x_1(n) = \{1, 2, 2, 2\}$ and $x_2(n) = \{1, 2, 3, 4\}$. (A/M-17 - R08)

Linear convolution:-

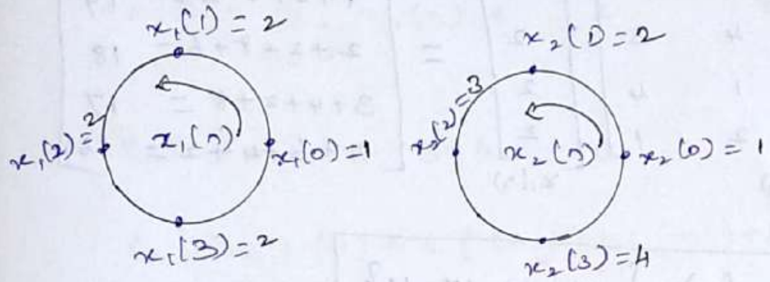
$x_2(n)$ \ $x_1(n)$	1	2	3	4
1	1	2	3	4
2	2	4	6	8
2	2	4	6	8
2	2	4	6	8

$$\boxed{y(n) = \{1, 4, 9, 16, 18, 14, 8\}}$$

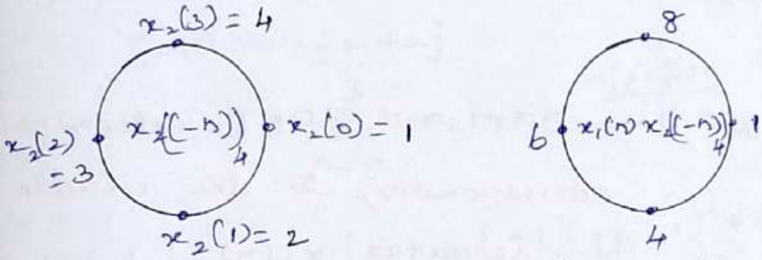
(29)

Circular convolution

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



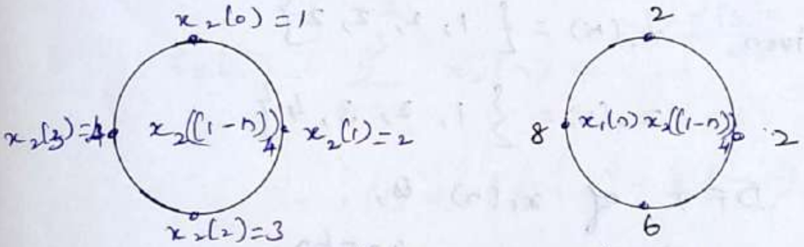
$x_3(0) = \sum_{n=0}^{3} x_1(n)\, x_2((-n))_4$



Folded sequence

Product Sequence

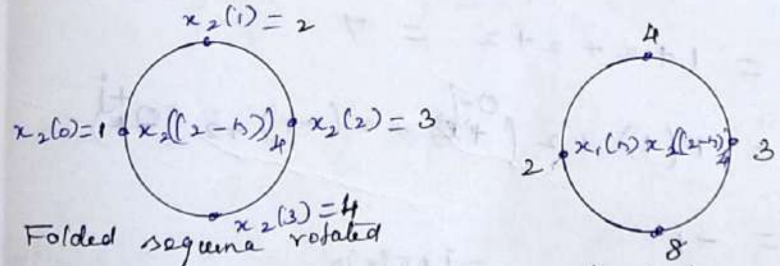
$\boxed{x_3(0) = 19}$



Folded sequence rotated by one unit

Product sequence

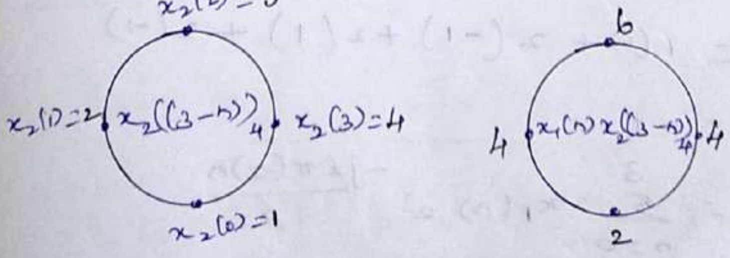
$\boxed{x_3(1) = 18}$



Folded sequence rotated by two units

Product sequence

$\boxed{x_3(2) = 17}$



$\boxed{x_3(3) = 16}$

$\therefore \boxed{x_3(n) = \{19, 18, 17, 16\}}$

Circular convolution of two sequences using matrix method

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

$$\underbrace{\begin{bmatrix} 1 & 4 & 3 & 2 \\ 2 & 1 & 4 & 3 \\ 3 & 2 & 1 & 4 \\ 4 & 3 & 2 & 1 \end{bmatrix}}_{x_2(n)} \underbrace{\begin{bmatrix} 1 \\ 2 \\ 2 \\ 2 \end{bmatrix}}_{x_1(n)} = \begin{bmatrix} 1+8+6+4 = & 19 \\ 2+2+8+6 = & 18 \\ 3+4+2+8 = & 17 \\ 4+6+4+2 = & 16 \end{bmatrix}$$

$$\therefore \boxed{x_3(n) = \{19, 18, 17, 16\}}$$

5) By means of the DFT and IDFT, determine the sequence $x_3(n)$ corresponding to the circular convolution of the sequences $x_1(n) = \{1, 2, 2, 2\}$ and $x_2(n) = \{1, 2, 3, 4\}$.

Soln:

Given, $x_1(n) = \{1, 2, 2, 2\}$

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

4-point DFT of $x_1(n)$ is,

$$X_1(k) = \sum_{n=0}^{3} x_1(n)\, e^{-j\frac{2\pi kn}{4}} \qquad k = 0, 1, 2, 3$$

$$X_1(0) = 1+2+2+2 = 7$$

$$X_1(1) = 1(1) + 2(0-j) + 2(-1) + 2(0+j)$$

$$= -1$$

$$X_1(2) = \sum_{n=0}^{3} x_1(n)\, e^{-j\frac{2\pi(2)n}{4}}$$

$$= 1(1) + 2(-1) + 2(1) + 2(-1)$$

$$X_1(2) = -1$$

$$X_1(3) = \sum_{n=0}^{3} x_1(n)\, e^{-j\frac{2\pi(3)n}{4}}$$

$$= 1(1) + 2(0+j) + 2(-1) + 2(-j)$$

$$= -1$$

$\therefore\ x_1(k) = \{7, -1, -1, -1\}$

4-point DFT of $x_2(n)$ is,

$$X_2(k) = \sum_{n=0}^{3} x_2(n)\, e^{\frac{-j2\pi kn}{4}} \qquad k=0,1,2,3$$

$X_2(0) = 1+2+3+4 = 10$

$X_2(1) = 1(1) + 2(0-j) + 3(-1) + 4(0+j) \qquad e^{-j\frac{\pi n}{2}}$

$= 1 - 2j - 3 + 4j$

$X_2(1) = -2+2j$

$$X_2(2) = \sum_{n=0}^{3} x_2(n)\, e^{\frac{-j2\pi(2)n}{4}}$$

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

$= 1 - 2 + 3 - 4$

$= -2$

$$X_2(3) = \sum_{n=0}^{3} x_2(n)\, e^{\frac{-j2\pi(3)n}{4}}$$

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

$= 1 + 2j - 3 - 4j$

$X_2(3) = -2 - 2j$

$\therefore\ X_2(k) = \{10, -2+2j, -2, -2-2j\}$.

$X_3(k) = X_1(k)\, X_2(k) = \{70, +2-2j, 2, 2+2j\}$

Inverse DFT of $X_3(k)$ is,

$$x_3(n) = \frac{1}{4}\sum_{k=0}^{3} X_3(k)\, e^{\frac{j2\pi kn}{4}} \qquad n=0,1,2,3$$

$x_3(0) = \frac{1}{4}\left[70 + 2 - \cancel{2j} + 2 + 2 + \cancel{2j}\right]$

$x_3(0) = \frac{1}{4}(76) = 19$

$$x_3(1) = \frac{1}{4}\sum_{k=0}^{3} x_3(k)\, e^{j\frac{\pi k}{2}}$$

$$= \frac{1}{4}\Big[70(1) + (2-2j)(0+j) + 2(-1) + (2+2j)(0-j)\Big]$$

$$= \frac{1}{4}\left[70 + \cancel{2}j + 2 - 2 - \cancel{2}j + 2\right]$$

$$= \frac{1}{4}[72]$$

$$= 18$$

$$x_3(2) = \frac{1}{4}\sum_{k=0}^{3} x_3(k)\, e^{\frac{j\cancel{2}\pi k(\cancel{2})}{\cancel{4}}}$$

$$= \frac{1}{4}\Big[70(1) + (2-2j)(-1) + 2(1) + (2+2j)(-1)\Big]$$

$$= \frac{1}{4}\left[70 - 2 + \cancel{2}j + \cancel{2} - \cancel{2} - \cancel{2}j\right]$$

$$= \frac{1}{4}(68)$$

$$x_3(2) = 16$$

$$x_3(3) = \frac{1}{4}\sum_{k=0}^{3} x_3(k)\, e^{\frac{j\cancel{2}\pi k(3)}{\cancel{4}2}}$$

$$= \frac{1}{4}\Big[70(1) + (2-2j)(0-j) + 2(-1) + (2+2j)(0-j)\Big]$$

$$= \frac{1}{4}\left[70 - 2 + \cancel{2}j - 2 - 2 - \cancel{2}j\right]$$

$$= \frac{1}{4}[64]$$

$$x_3(3) = 16$$

$$\therefore \boxed{x_3(n) = \{19, 18, 17, 16\}}$$

6). The first five DFT co-efficients of a sequence $x(n)$ are $X(0) = 20$, $X(1) = 5+j2$, $X(2) = 0$, $X(3) = 0.2+j0.4$, $X(4) = 0$.
Determine the remaining DFT coefficients.
(M/J - 07 - R04)

Soln:

$$X(k) = \{20,\ 5+j2,\ 0,\ 0.2+j0.4,\ 0,\ X(5),\ X(6),\ X(7)\}$$

From the complex conjugate property of DFT,

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

$$X(5) = X^*(8-5) = X^*(3) = 0.2-j0.4$$

$$X(6) = X^*(8-6) = X^*(2) = 0$$

$$X(7) = X^*(8-7) = X^*(1) = 5-j2$$

$$\therefore \boxed{X(k) = \{20,\ 5+j2,\ 0,\ 0.2+j0.4,\ 0,\ 0.2-j0.4,\ 0,\ 5-j2\}}$$

7). Compute 8 point DFT of the sequence
$x(n) = \{1, 0, \underset{\uparrow}{1}, -1, 1, -1, 0, 1\}$ (N/D-12 R08)

Soln:

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

The periodic extension of the sequence $x(n)$ is given by, [Periodicity Property]

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

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

DFT of $x(n)$ is, $X(k) = \sum_{n=0}^{7} x(n)\, e^{-j\frac{2\pi kn}{8}}$, $k = 0, 1, \ldots 7$

$$X(0) = \sum_{n=0}^{7} x(n) = 1-1+1-1+0+1+1+0$$

$$\boxed{X(0) = 2}$$

$$X(1) = \sum_{n=0}^{7} x(n) e^{-j\frac{\pi n}{4}}$$

$$= 1 - 0.707 + j0.707 + 0 - j + 0.707 + j0.707 + 0 - 0.707 + j0.707 + 0 + j + 0$$

$$X(1) = 0.293 + j2.1213$$

$$X(2) = \sum_{n=0}^{7} x(n) e^{-j\frac{\pi (2)n}{4}}$$

$$= 1 + j - 1 - j + 0 - j - 1 + 0$$

$$X(2) = -1 - j$$

$$X(3) = \sum_{n=0}^{7} x(n) e^{-j\frac{2\pi(3)n}{8}}$$

$$X(3) = 1.7071 + j2.1213$$

$$X(4) = \sum_{n=0}^{7} x(n) e^{-j\frac{\pi(4)n}{8}}$$

$$= 1 + 1 + 1 + 1 + 0 + (-1) + 1 + 0$$

$$X(4) = 4$$

$$X(5) = \sum_{n=0}^{7} x(n) e^{-j\frac{2\pi(5)n}{8}}$$

$$X(5) = 1.7071 - j2.1213$$

$$X(6) = \sum_{n=0}^{7} x(n) e^{-j\frac{3\pi n}{2}}$$

$$X(6) = -1 + j$$

$$X(7) = \sum_{n=0}^{7} x(n) e^{-j\frac{7\pi n}{4}}$$

$$X(7) = 0.2929 - j2.1213$$

$$\therefore \boxed{X(k) = \{2,\ 0.2929 + j2.1213,\ -1 - j,\ 1.7071 + j2.1213,\ 4,\ 1.7071 - j2.1213,\ -1 + j,\ 0.2929 - j2.1213\}}$$

FILTERING METHODS BASED OF DFT

Since DFT provides a discrete frequency representation of a finite-duration sequence in the frequency domain, it is used as a computational tool for linear system analysis and especially for linear filtering.

Use of DFT in linear filtering:

The product of two DFTs is equivalent to the circular convolution of the corresponding time-domain sequences. Unfortunately, circular convolution is of no use, if the objective is to find the output of a linear filter to a given input sequence. In this case a frequency domain methodology equivalent to linear convolution is required.

Let $x(n)$ be the finite duration sequence $x(n)$ of length L which excites an FIR filter of length M.

$$\text{Let } x(n) = 0 \quad n<0 \ \& \ n \geq L$$

$$h(n) = 0 \quad n<0 \ \& \ n \geq M$$

where $h(n)$ is the impulse response of FIR filter.

The output sequence,

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

The length of $y(n)$ is $L+M-1$

In frequency domain,

$$Y(\omega) = X(\omega) H(\omega)$$

if $\quad Y(k) \equiv Y(\omega)\Big|_{\omega=\frac{2\pi k}{N}} \qquad k=0,1,\ldots N-1$

$$= X(\omega) H(\omega)\Big|_{\omega=\frac{2\pi k}{N}} \qquad k=0,1\ldots N-1$$

then $\quad Y(k) = X(k) H(k) \qquad k=0,1,\ldots N-1$

where $N \geq L+M-1$ and $X(k)$, $H(k)$ are the N-point DFTs of the sequence $x(n)$ and $h(n)$ respectively. Since $x(n)$ and $h(n)$ have lengths less the N, zero padding can be done here.

Thus with zero padding DFT can be used to perform Linear filtering.

By increasing the length of the sequences $x(n)$ and $h(n)$ to N points and then circularly convoluting the resulting sequences, linear convolution can be obtained.

8). Find the response of linear filter with the sequences $x(n) = \{1, 2, 1\}$ and $h(n) = \{1, -2, 2\}$ using circular convolution method. [N/D – 10 R08]

Given, $\quad x(n) = \{1, 2, 1\} \qquad N_1 = 3$

$$h(n) = \{1, -2, 2\} \qquad N_2 = 3$$

Length of output sequence, $N = N_1 + N_2 - 1$

$$\boxed{N = 5}$$

By zero padding, $\quad x(n) = \{1, 2, 1, 0, 0\}$

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

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

$$\therefore \boxed{y(n) = \{1, 0, -1, 2, +2\}}$$

To verify using linear convolution

$x(n)$ \ $h(n)$	1	-2	2
1	1	-2	2
2	2	-4	4
1	1	-2	2

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

FILTERING OF LONG DATA SEQUENCES [M/J - 4 Rot]

In practical applications involving linear filtering of signals, the input sequence $x(n)$ is often a very long sequence. [Ex real time signal monitoring and analysis].

Since linear filtering performed via the DFT involves operations on a block of data, which is by necessity must be limited in size due to limited memory of a digital computer. A long input signal must be segmented to fixed size by blocks prior to processing. Since the filtering is linear, successive blocks can be processed one at a time via the DFT, and the output blocks are fitted together to form the overall output signal sequence.

Overlap - save and Overlap - add methods are used for long data sequences.

Assume FIR filter has duration M, the input data sequence is segmented into blocks of L points where $L >> M$.

(i) OVERLAP - SAVE METHOD:-

* The size of the input blocks is $N = L + M - 1$ and DFTs and IDFT are of length N.
* Each data block consists of the last $M-1$ data points of the previous data block

followed by L new data points to form a data sequence of length $N = L+M-1$.

* N-point DFT is computed for each data block.

* The impulse response of the FIR filter is increased in length by appending $L-1$ zeros and N-point DFT is computed once and stored.

* The multiplication of the two N-points DFTs $\{H(k)\}$ and $\{X_m(k)\}$ for the mth block of data yields

$$\hat{Y}_m(k) = H(k)\, X_m(k) \qquad k = 0, 1, \ldots N-1$$

Then the N-point IDFT yields the result.

$$\hat{y}_m(n) = \{ y_m(0)\, y_m(1) \ldots y_m(M-1)\, y_m(M) \ldots y_m(N-1) \}$$

* Since the data record is of length N, the first $M-1$ points of $y_m(n)$ are corrupted by aliasing and must be discarded.

* The last L points of $y_m(n)$ are exactly the same as the result from linear convolution

$$\hat{y}_m(n) = y_m(n), \quad n = M, M+1 \ldots N-1$$

* To avoide loss of data due to aliasing, the last $M-1$ points of each data record are saved and these points become the first $M-1$ data points of the subsequent record.

* To start the process, the first $M-1$ points of the first record are set to zero.

Thus the blocks of data sequences are

$$x_1(n) = \{\underbrace{0, 0, \ldots 0}_{M-1 \text{ points}}, x(0), x(1) \ldots x(L-1)\}$$

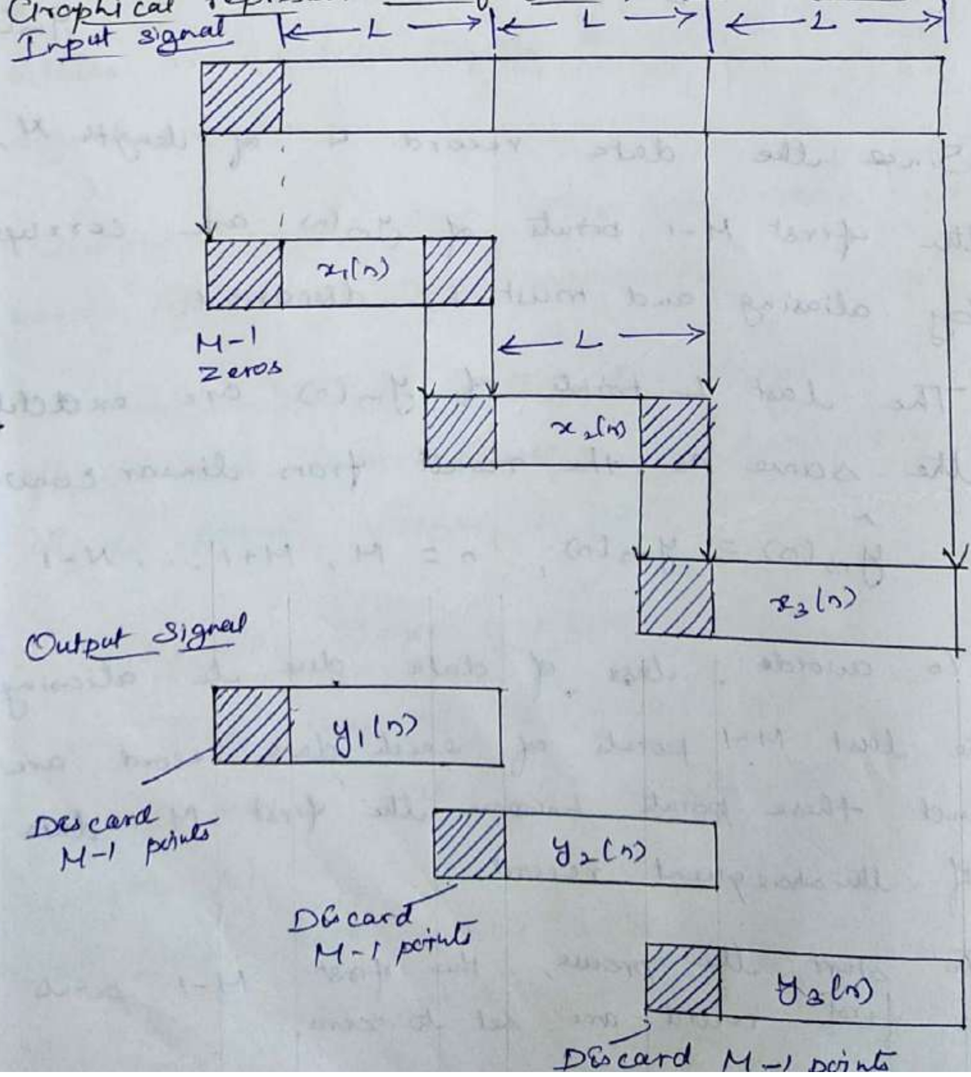
$$x_2(n) = \{\underbrace{x(L-M+1), \ldots x(L-1)}_{M-1 \text{ data points from } x_1(n)}, \underbrace{x(L), \ldots x(2L-1)}_{L \text{ new data points}}\}$$

$$x_3(n) = \{\underbrace{x(2L-M+1), \ldots x(2L-1)}_{M-1 \text{ data points from } x_2(n)}, \underbrace{x(2L), \ldots x(3L-1)}_{L \text{ new data points}}\}$$

and so forth.

In the resulting data sequence from IDFT, the first M−1 points are discarded due to aliasing and the remaining L points constitute the desired result from linear convolution.

Graphical representation of overlap-save method:



OVERLAP - ADD METHOD

* The size of the input data block is L points and the size of the DFTs and IDFT is $N = L + M - 1$.
* To each data block, $M-1$ zeros are appended and N-point DFT is calculated.

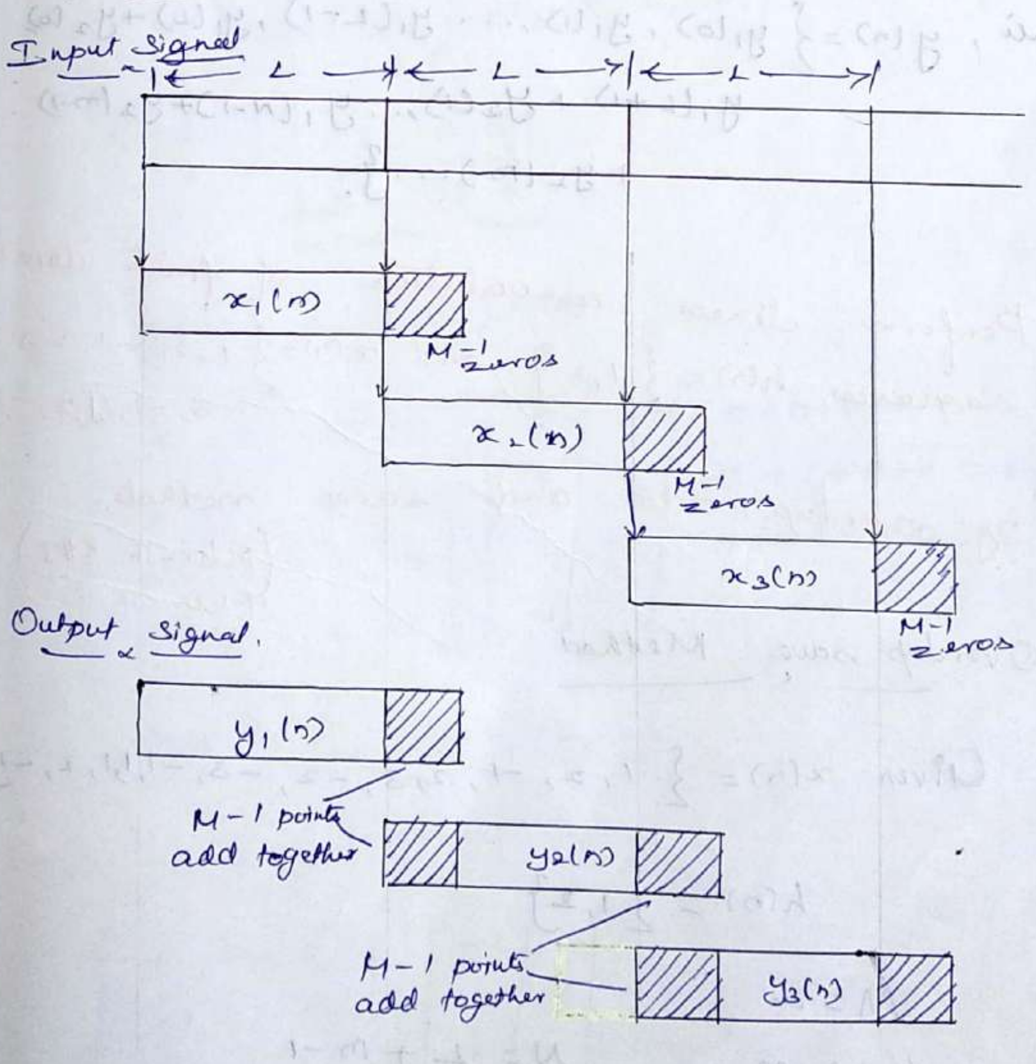
Thus the data block is represented as,

$$x_1(n) = \{x(0), x(1), \dots x(L-1), \underbrace{0, 0, 0 \dots 0}_{M-1 \text{ zeros}}\}$$

$$x_2(n) = \{x(L), x(L+1), \dots x(2L-1), \underbrace{0, 0 \dots 0}_{M-1 \text{ zeros}}\}$$

$$x_3(n) = \{x(2L), \dots x(3L-1), \underbrace{0, 0, \dots 0}_{M-1 \text{ zeros}}\}$$ and so on.

Graphical representation of overlap - add method



The two N-points DFTs are multiplied together to form

$$y_m(k) = H(k) X_m(k) \qquad k=0,1 \ldots N-1$$

The IDFT yields data blocks of length N that are free of aliasing, since the size of the DFTs and IDFT is $N = L+M-1$ and the sequences are increased to N-points by appending zeros to each block.

* Since each data block is terminated with $M-1$ zeros, the last $M-1$ points from each output block must be overlapped and added to the first $M-1$ points of the succeeding block.

ie, $y(n) = \{ y_1(0), y_1(1), \ldots y_1(L-1), y_1(L)+y_2(0), y_1(L+1)+y_2(1), \ldots y_1(N-1)+y_2(M-1) + y_2(M) \ldots \}$.

9). Perform linear convolution of finite duration sequences $h(n) = \{1,2\}$ and $x(n) = \{1,2,-1,2,3,-2,-3,-1,1,1,2,-1\}$ by overlap add and save methods.

(N/D-16 R-13)
(N/D-11 R-08)

Overlap save Method

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

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

$M = 2$

$L >> M \qquad N = L+M-1$

Let $L = 3$ $\therefore N = 3+2-1 = 4$

Divide the input sequence $x(n)$ into data blocks as shown below.

$$x_1(n) = \{ \underset{\substack{\Downarrow \\ M-1 \\ \text{zeros}}}{0}, \underbrace{1, 2, -1}_{3 \text{ i/p data}} \}$$

$$x_2(n) = \{ \underset{\substack{\Downarrow \\ M-1=1 \\ \text{data from} \\ \text{previous block}}}{-1}, \underbrace{2, 3, -2}_{3 \text{ new data}} \}$$

||| ly, $x_3(n) = \{-2, -3, -1, 1\}$

$x_4(n) = \{1, 1, 2, -1\}$

$x_5(n) = \{-1, 0, 0, 0\}$.

$$h(n) = \{ \underbrace{1, 2}_{\text{impulse sequence}}, \underbrace{0, 0}_{L-1 \text{ zeros}} \}$$

$\therefore$ $y_1(n) = x_1(n)$ Ⓝ $h(n)$

$$\begin{bmatrix} 1 & 0 & 0 & 2 \\ 2 & 1 & 0 & 0 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 2 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 2 \\ -1 \end{bmatrix} = \begin{bmatrix} 0+0+0-2 = -2 \\ 0+1+0+0 = 1 \\ 0+2+2+0 = 4 \\ 0+0+4-1 = 3 \end{bmatrix}$$

$$y_1(n) = [-2, +1, 4, 3]$$

$y_2(n) = x_2(n)$ Ⓝ $h(n)$

$$\begin{bmatrix} 1 & 0 & 0 & 2 \\ 2 & 1 & 0 & 0 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 2 & 1 \end{bmatrix} \begin{bmatrix} -1 \\ 2 \\ 3 \\ -2 \end{bmatrix} = \begin{bmatrix} -1+0+0-4 = -5 \\ -2+2+0+0 = 0 \\ 0+4+3+0 = 7 \\ 0+0+6-2 = 4 \end{bmatrix}$$

$$y_2(n) = [-5, 0, 7, 4]$$

$y_3(n) = x_3(n) \circledast h(n)$

$$\begin{bmatrix} 1 & 0 & 0 & 2 \\ 2 & 1 & 0 & 0 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 2 & 1 \end{bmatrix} \begin{bmatrix} -2 \\ -3 \\ -1 \\ 1 \end{bmatrix} = \begin{bmatrix} -2+0+0+2=0 \\ -4-3+0+0=-7 \\ 0-6-1+0=-7 \\ 0+0-2+1=-1 \end{bmatrix}$$

$y_3(n) = [0, -7, -7, -1]$

$$y_4(n) = \begin{bmatrix} 1 & 0 & 0 & 2 \\ 2 & 1 & 0 & 0 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 2 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 2 \\ -1 \end{bmatrix} = \begin{bmatrix} 1+0+0-2=-1 \\ 2+1+0+0=3 \\ 0+2+2+0=4 \\ 0+0+4-1=3 \end{bmatrix}$$

$y_4(n) = [-1, 3, 4, 3]$

$$y_5(n) = \begin{bmatrix} 1 & 0 & 0 & 2 \\ 2 & 1 & 0 & 0 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 2 & 1 \end{bmatrix} \begin{bmatrix} -1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -1 \\ -2 \\ 0 \\ 0 \end{bmatrix}$$

$y_5(n) = [-1, -2, 0, 0]$

$y_1(n) \rightarrow [-2 \quad 1 \quad 4 \quad 3]$, -2 Discard

$y_2(n) \rightarrow [5, 0, 7, 4]$, 5 discard

$y_3(n) \rightarrow [0, -7, -7, -1]$, 0 Discard

$y_4(n) \rightarrow [-1, 3, 4, 3]$, -1 discard

$y_5(n) \rightarrow [-1, -2, 0, 0]$, -1 discard

$$\boxed{y(n) = \{1, 4, 3, 0, 7, 4, -7, -7, -1, 3, 4, 3, -2\}}$$

(45)

Overlap add Method:

$M = 2$ Let $L = 3$.

$N = 4$

The input sequence can be divided into data blocks as shown below.

$x_1(n) = \{\underbrace{1, 2, -1}_{3\text{ data}}, \underset{M-1 \text{ zeros}}{0}\}$

$x_2(n) = \{\underbrace{2, 3, -2}_{3\text{ data}}, \underset{M-1 \text{ zeros}}{0}\}$

||^{ly} $x_3(n) = \{-3, -1, 1, 0\}$

$x_4(n) = \{1, 2, -1, 0\}$

The impulse sequence can be written as

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

$\therefore y_1(n) = x_1(n) \; (N) \; h(n)$

$$= \begin{bmatrix} 1 & 0 & 0 & 2 \\ 2 & 1 & 0 & 0 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 2 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ -1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1+0+0+0 = 1 \\ 2+2+0+0 = 4 \\ 0+4-1+0 = 3 \\ 0+0-2+0 = -2 \end{bmatrix}$$

$y_1(n) = \{1, 4, 3, -2\}$

$y_2(n) = x_2(n) \; (N) \; h(n)$

$$= \begin{bmatrix} 1 & 0 & 0 & 2 \\ 2 & 1 & 0 & 0 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 2 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \\ -2 \\ 0 \end{bmatrix} = \begin{bmatrix} 2+0+0+0 = 2 \\ 4+3+0+0 = 7 \\ 0+6-2+0 = 4 \\ 0+0-4+0 = -4 \end{bmatrix}$$

$y_2(n) = \{2, 7, 4, -4\}$

$$y_3(n) = x_3(n) \; (N) \; h(n)$$

$$= \begin{bmatrix} 1 & 0 & 0 & 2 \\ 2 & 1 & 0 & 0 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 2 & 1 \end{bmatrix} \begin{bmatrix} -3 \\ -1 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} -3+0+0+0 = -3 \\ -6-1+0+0 = -7 \\ 0-2+1+0 = -1 \\ 0+0+2+0 = 2 \end{bmatrix}$$

$$y_3(n) = \{-3, -7, -1, 2\}$$

$$y_4(n) = x_4(n) \; (N) \; h(n)$$

$$= \begin{bmatrix} 1 & 0 & 0 & 2 \\ 2 & 1 & 0 & 0 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 2 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ -1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1+0+0+0 = 1 \\ 2+2+0+0 = 4 \\ 0+4-1+0 = 3 \\ 0+0-2+0 = -2 \end{bmatrix}$$

$$y_4(n) = \{1, 4, 3, -2\}$$

$y_1(n) \rightarrow [1, 4, 3, -2]$

$y_2(n) \rightarrow$ $[2, 7, 4, -4]$

$y_3(n) \rightarrow$ $[-3, -7, -1, 2]$

$y_4(n) \rightarrow$ $[1, 2, -1, 0]$

$$y(n) = \{1, 4, 3, 0, 7, 4, -7, -7, -1, 3, -2, \underbrace{-1, 0}_{\text{not necessary}}$$

$$\boxed{y(n) = \{1, 4, 3, 0, 7, 4, -7, -7, -1, 3, -2\}}$$

FAST FOURIER TRANSFORM

T1-(511 - 512)

* DFT plays an important role in many applications of digital signal processing, including linear filtering, correlation analysis and spectrum analysis.

* The major reason for its importance is the existence of efficient algorithms for computing DFT.

Two approaches for computing DFT:-

(i) Divide-and-conquer approach (DFT of length N).
→ Fast Fourier Transform [FFT]
(When N is a power of 2 (or) 4)

(ii) Based on the formulation of the DFT as a linear filtering operation on the data
→ Goertzel algorithm
→ Chirp-z transform Algorithm

Efficient computation of DFT,

The computational problem for the DFT is to compute the sequence $\{X(k)\}$ of N complex-valued numbers given another sequence of data $\{x(n)\}$ of length N, according to the formula

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

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

The IDFT is given by,

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

For each value of k, direct computation of X(k) involves N complex multiplications [4N real multiplications], (N-1) complex additions [4N-2 real additions].

$\therefore$ To compute N-point DFT, N^2 complex multiplications and N(N-1) complex additions required.

Twiddle Factor

The complex number $W_N = e^{-j\frac{2\pi}{N}}$ (or) $W_N = 1 \angle -2\pi/N$ is called the phase factor (or) twiddle factor. It represents N^{th} root of unity.

Properties of twiddle factor are,

Symmetry Property : $W_N^{k+\frac{N}{2}} = -W_N^k$

Periodicity Property : $W_N^{k+N} = W_N^k$

* FFT - Fast Fourier Transform is the efficient way of calculating DFT.

(49) Radix-2 FFT Algorithm   T1-(519-527)

The length N of DFT is represented as $N = r^{v}$ where $r \to$ radix of DFT.

radix-2 algorithms are most widely used.

I. Radix-2 DIT-FFT [Decimation In Time FFT Algorithm] [N/D-16 R13]

Consider $N = 2^{v}$

* Split the N-point data sequence into two $N/2$ point data sequences $f_1(n)$ and $f_2(n)$, corresponding to the even-numbered and odd-numbered samples of $x(n)$ respectively,

ie $f_1(n) = x(2n)$

$f_2(n) = x(2n+1) \quad n = 0, 1, \ldots \left(\frac{N}{2}-1\right)$ ——(1)

$f_1(n)$ and $f_2(n)$ are obtained by decimating $x(n)$ by a factor of 2, hence the resulting FFT algorithm is called a decimation-in-time algorithm.

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

$$= \sum_{n\ \text{even}} x(n) W_N^{kn} + \sum_{n\ \text{odd}} x(n) W_N^{kn}$$

$$X(k) = \sum_{m=0}^{\left(\frac{N}{2}\right)-1} x(2m) W_N^{2mk} + \sum_{m=0}^{\left(\frac{N}{2}\right)-1} x(2m+1) W_N^{k(2m+1)}$$ ——(2)

WKT $W_N^2 = W_{N/2}$

$$\therefore (2) \Rightarrow X(k) = \sum_{m=0}^{\left(\frac{N}{2}\right)-1} f_1(m) W_{N/2}^{km} + W_N^{k} \sum_{m=0}^{\left(\frac{N}{2}\right)-1} f_2(m) W_{N/2}^{km}$$ ——(3)

$$= F_1(k) + W_N^{k} F_2(k), \quad k = 0, 1, \ldots N-1$$

where $F_1(k)$ and $F_2(k)$ are $\frac{N}{2}$ point DFTs of the sequences $f_1(m)$ and $f_2(m)$ respectively.

$\therefore$ $F_1(k)$ and $F_2(k)$ are periodic with period $\frac{N}{2}$

$$F_1\left(k+\frac{N}{2}\right) = F_1(k) \quad \text{and} \quad F_2\left(k+\frac{N}{2}\right) = F_2(k)$$

also, $W_N^{k+\frac{N}{2}} = -W_N^k$

$\therefore$ (4) $\Rightarrow$ $$X(k) = F_1(k) + W_N^k F_2(k)$$
$$k = 0, 1, \ldots \frac{N}{2}-1 \quad \text{—— (5)}$$

$$X\left(k+\frac{N}{2}\right) = F_1(k) - W_N^k F_2(k) \quad \text{—— (6)}$$
$$k = 0, 1, \ldots \frac{N}{2}-1$$

The direct computation of $F_1(k)$ and $F_2(k)$ requires $\left(\frac{N}{2}\right)^2$ complex multiplications each. In addition to that $\frac{N}{2}$ additional complex multiplications required to compute $W_N^k F_2(k)$.

$\therefore$ The computation of $X(k)$ require

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

complex multiplications.

$\Rightarrow$ Results in reduction of no. of multiplications from N^2 to $\frac{N^2}{2} + \frac{N}{2}$, which is about a factor of 2 for N large.

Let $G_1(k) = F_1(k) \qquad k = 0, 1, \ldots \frac{N}{2}-1$

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

Then, $X(k) = G_1(k) + G_2(k)$

$X\left(k+\frac{N}{2}\right) = G_1(k) - G_2(k)$

Having performed the decimation-in-time once, the process can be repeated for each of the sequences $f_1(n)$ and $f_2(n)$. Thus $f_1(n)$ would result in the two $\frac{N}{4}$ point sequences.

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

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

similarly for $f_2(n)$,

$$v_{21}(n) = f_2(2n) \qquad n=0,1,\ldots\frac{N}{4}-1$$

$$v_{22}(n) = f_2(2n+1) \qquad n=0,1\ldots\frac{N}{4}-1$$

By computing $\frac{N}{4}$-point DFTs, the $\frac{N}{2}$ point DFTs $F_1(k)$ and $F_2(k)$ can be obtained.

$$F_1(k) = V_{11}(k) + W_{N/2}^{k} V_{12}(k) \qquad k=0,1\ldots\frac{N}{4}-1$$

$$F_1\left(k+\frac{N}{4}\right) = V_{11}(k) - W_{N/2}^{k} V_{12}(k)$$

$$F_2(k) = V_{21}(k) + W_{N/2}^{k} V_{12}(k)$$

$$F_2\left(k+\frac{N}{4}\right) = V_{21}(k) - W_{N/2}^{k} V_{22}(k)$$

$\therefore$ The computation of $X(k)$ from $F_1(k)$ and $F_2(k)$ requires,

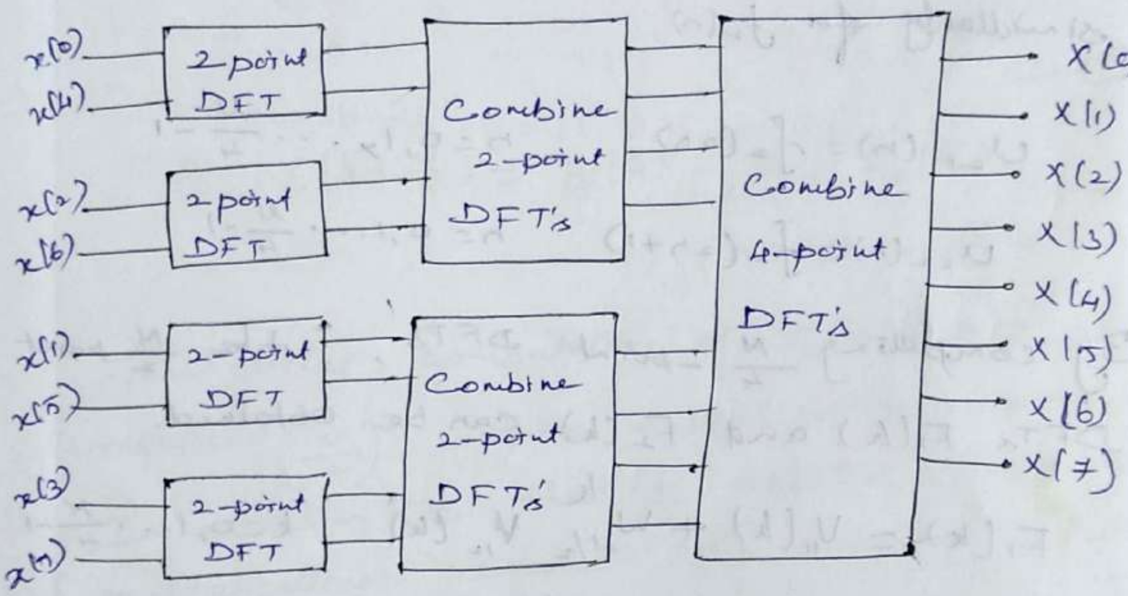
$$\frac{N^2}{4} + \frac{N}{2} + \frac{N}{2} = \frac{N^2}{4} + N.$$

The decimation of a 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.

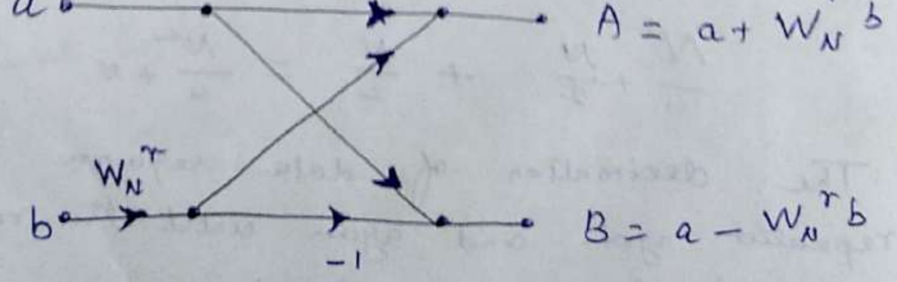
* Thus the total number of complex multiplication required is reduced to $\frac{N}{2} \log_2 N$

Block diagram of 8-point DIT-FFT

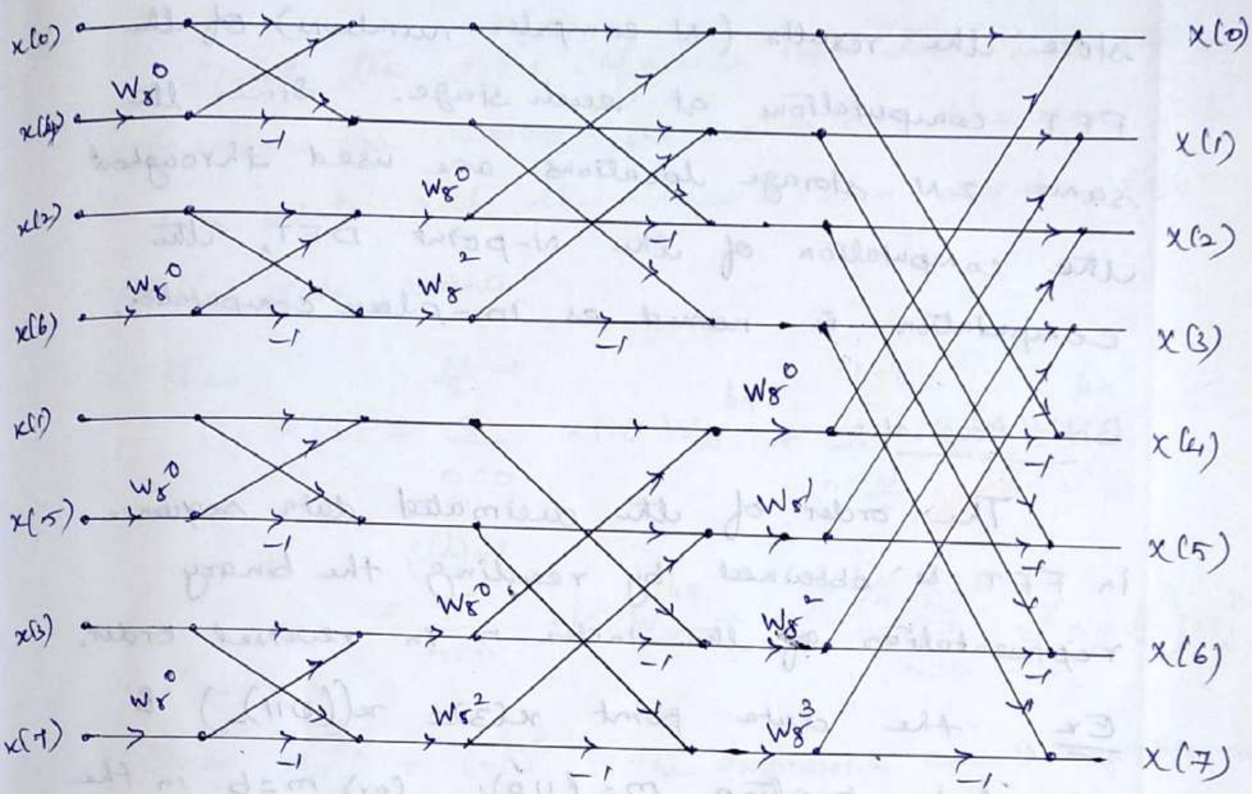


For N = 8 the computation is performed in three stages, beginning with computation of four 2-point DFTs, then two ~~four~~ 4-point DFTs and finally one 8-point DFT.

Basic Butterfly diagram of DIT-FFT



8 - point DIT - FFT Algorithm :-



Each Butterfly involves one complex multiplication and two complex additions. For $N = 2^v$ there are $N/2$ butterflies per stage of the computation process and $\log_2 N$ stages.

$\therefore$ The total number of complex multiplications required is $(N/2)\log_2 N$

The total number of complex additions required is $N \log_2 N$

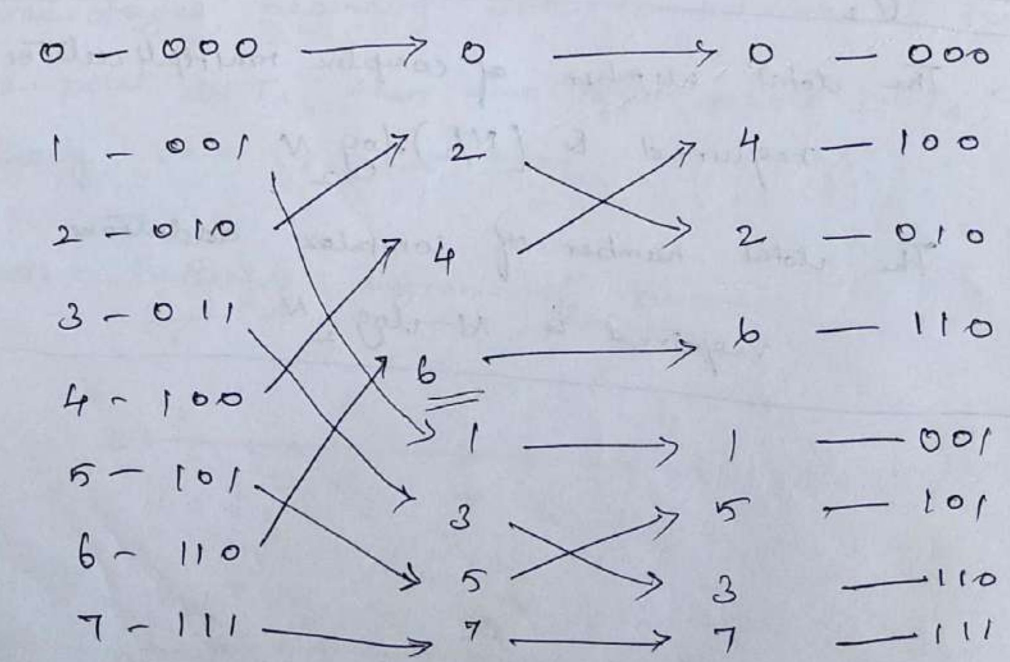
In-place computation

2N storage registers are required to store the results (N complex numbers) of the FFT computations at each stage. Since the same 2N storage locations are used throughout the computation of the N-point DFT, the computation is named as in-place computation.

Bit-Reversal:

The order of the decimated data sequence in FFT is obtained by reading the binary representation of the index n in reversed order.

Ex, the data point $x(3) = x((011)_2)$ is placed in position $m = (110)_2$ (or) $m = 6$ in the decimated array. ∴ The data $x(n)$ after decimation is stored in bit-reversed order. This is called bit reversal.



RADIX -2 DIF - FFT ALGORITHM [M/J-13 R08]

DIF - Decimation In Frequency

Split the DFT formula into two summations of which one involves the sum over the first $N/2$ data points and the second sum over the last $N/2$ data points.

Thus,

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

$$= \sum_{n=0}^{(\frac{N}{2})-1} x(n) W_N^{kn} + W_N^{Nk/2} \sum_{n=0}^{\frac{N}{2}-1} x\left(n+\frac{N}{2}\right) W_N^{kn}$$

Since $W_N^{\frac{kN}{2}} = (-1)^k$, the expression can be written as,

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

Now split $X(k)$ into even- and odd numbered samples.

$$X(2k) = \sum_{n=0}^{\frac{N}{2}-1} \left[x(n) + x\left(n+\frac{N}{2}\right)\right] W_{N/2}^{kn} \quad k=0,1 \ldots \frac{N}{2}-1$$

and

$$X(2k+1) = \sum_{n=0}^{\frac{N}{2}-1} \left\{ \left[x(n) - x\left(n+\frac{N}{2}\right)\right] W_N^{n} \right\} W_{N/2}^{kn}$$

$$k=0,1,\ldots \frac{N}{2}-1$$

$$\because W_N^2 = W_{N/2}$$

If $g_1(n)$ and $g_2(n)$ are the two $\frac{N}{2}$-point sequences,

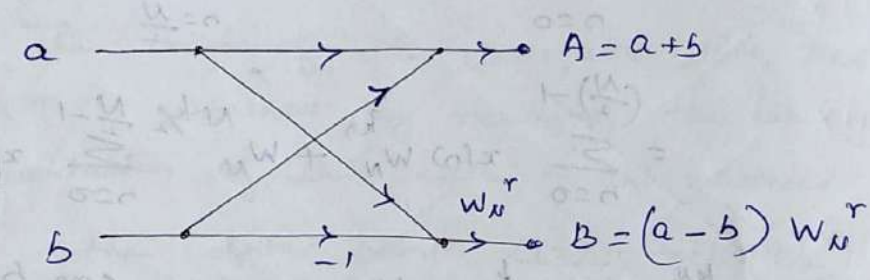
$$g_1(n) = x(n) + x\left(n+\frac{N}{2}\right)$$

$$g_2(n) = \left[x(n) - x\left(n + \frac{N}{2}\right) \right] W_N^{n} \qquad n = 0, 1, \ldots \frac{N}{2} - 1$$

then $$X(2k) = \sum_{n=0}^{\frac{N}{2}-1} g_1(n)\, W_{N/2}^{kn}$$

$$X(2k+1) = \sum_{n=0}^{\frac{N}{2}-1} g_2(n)\, W_{N/2}^{kn}$$

Basic Butterfly diagram :- [DIF-FFT]



The computational procedure can be repeated through decimation of the $N/2$ point DFTs, $X(2k)$ and $X(2k+1)$. The entire process involves, $v = \log_2 N$ stages of decimation, where each stage involves $N/2$ butterflies.

The computation of N-point DFT via the DIF-FFT algorithm requires,

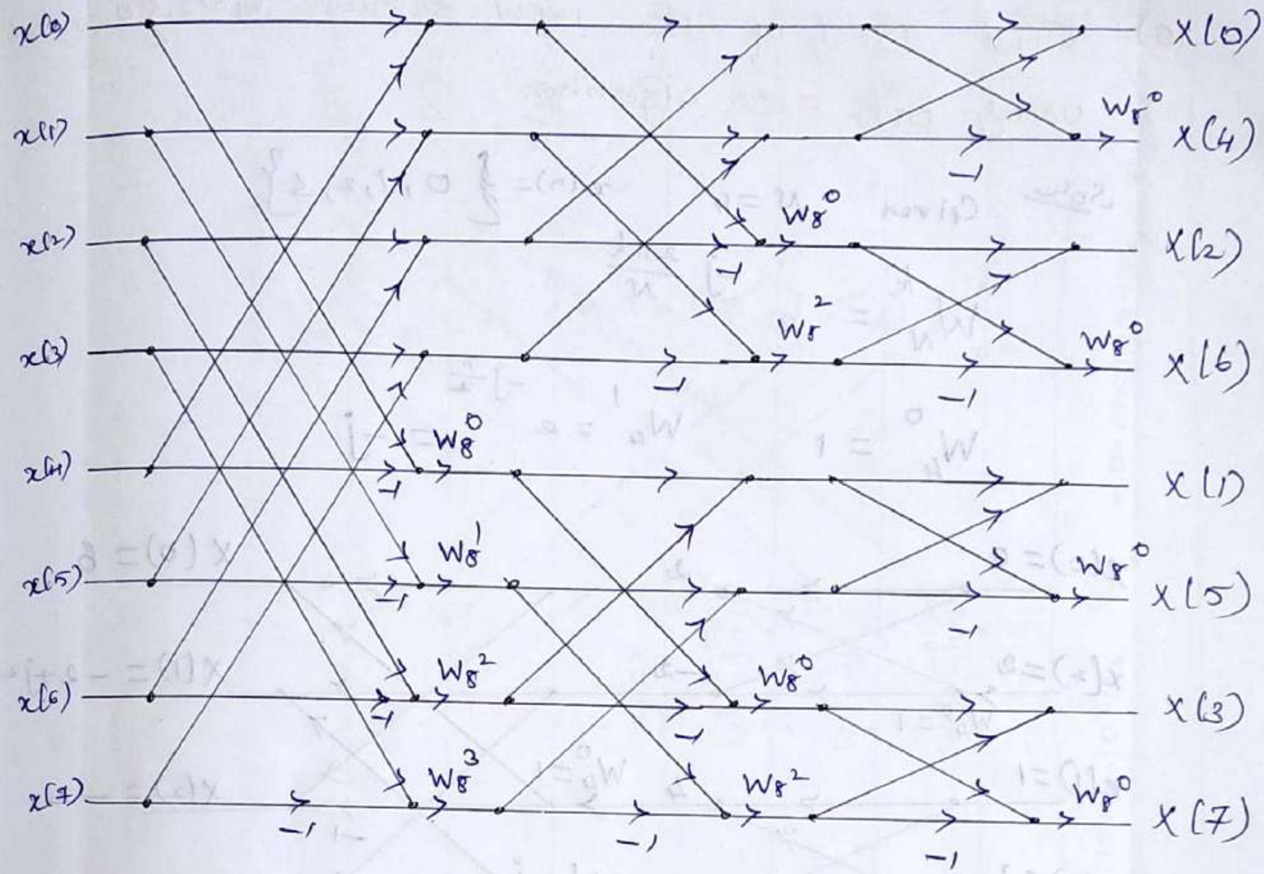
$\frac{N}{2} \log N$ - complex multiplications and

$N \log N$ - complex additions.

For $N = 8$, In DIF FFT algorithm, the input data $x(n)$ occurs in natural order, but the output DFT occurs in bit-reversed order.

→ The computations are performed in place.

8-point DIT-FFT Algorithm:-



USE OF FFT IN LINEAR FILTERING

The important application of FFT algorithm is in FIR linear filtering of long data sequences. The overlap-add and save methods can be used with FFT algorithms for linear filtering of long data sequences.

→ For calculating DFT, FFT algorithms can be used.

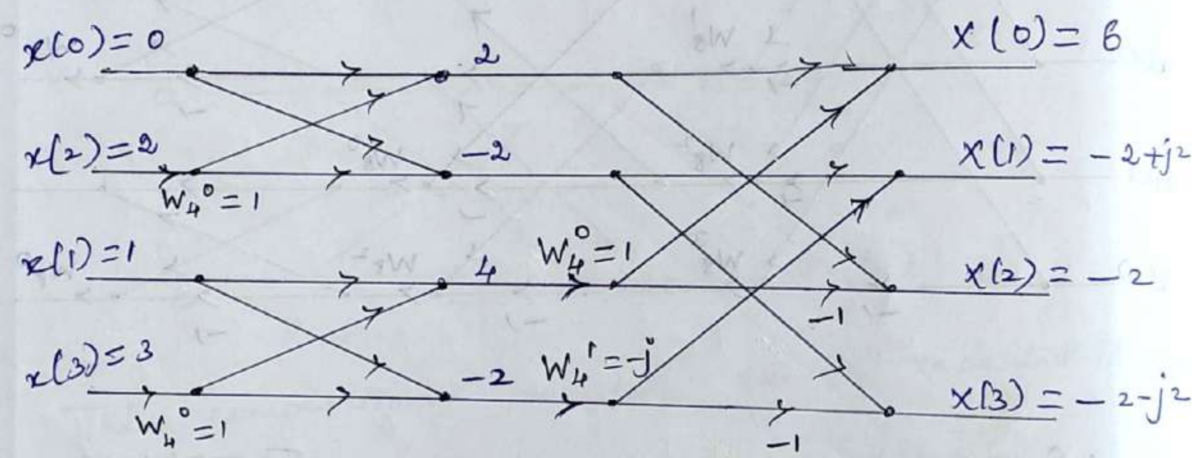
PROBLEMS USING DIT-FFT ALGORITHM

10). Find DFT of the input sequence $x(n) = \{0, 1, 2, 3\}$ using DIT-FFT algorithm.

Soln: Given $N = 4$ $x(n) = \{0, 1, 2, 3\}$

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

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



$$\boxed{X(k) = \{6, -2+j2, -2, -2-j2\}}$$

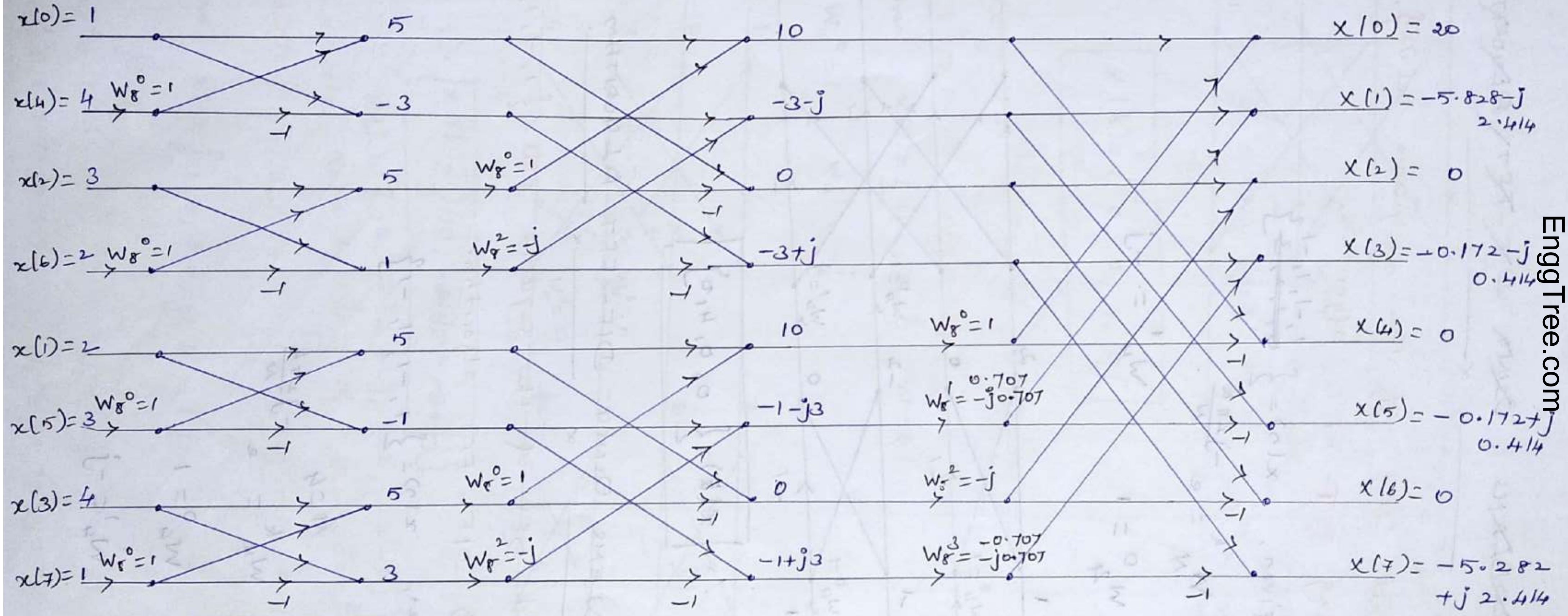
11). Find the DFT of the sequence $x(n) = \{1, 2, 3, 4, 4, 3, 2, 1\}$ using DIT-FFT algorithm. [N/D-16 RB]

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

$N = 8$ $\qquad W_N^k = e^{-j\frac{2\pi}{N}\cdot k}$

$W_8^0 = 1 \qquad W_8^1 = 0.707 - j0.707$

$W_8^2 = -j \qquad W_8^3 = -0.707 - j0.707$



$$\therefore \boxed{X(k) = \{20, -5.828-j2.414, 0, -0.172-j0.414, 0, -0.172+j0.414, 0, -5.282+j2.414\}}$$

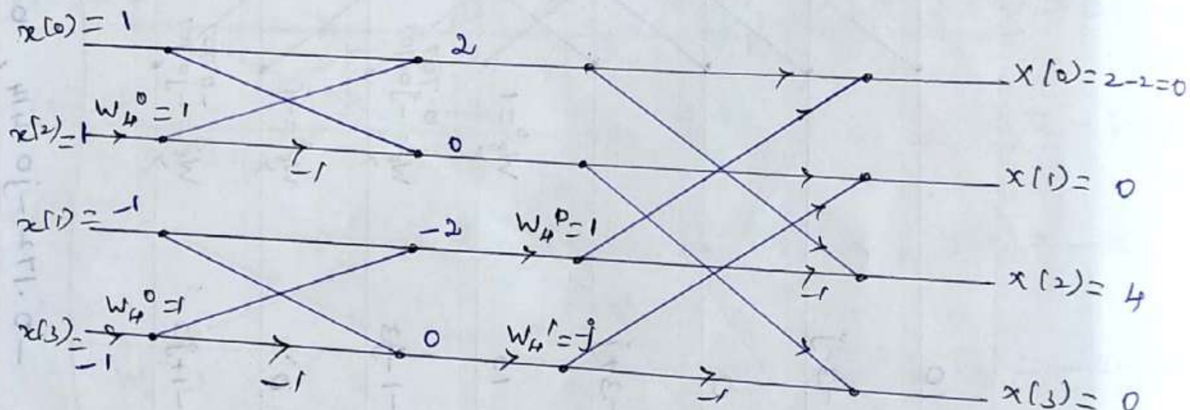
PROBLEMS USING DIT-FFT ALGORITHM

12) Find DFT of the sequence $x(n) = \{1, -1, 1, -1\}$ using DIT-FFT algorithm.

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

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

$$W_4^0 = 1 \qquad W_4^1 = -j$$



$$\therefore \boxed{X(k) = \{0, 0, 4, 0\}}$$

PROBLEMS USING DIF-FFT ALGORITHM

13) Find DFT of the sequence $x(n) = \{1, -1, 1, -1\}$ using DIF-FFT algorithm.

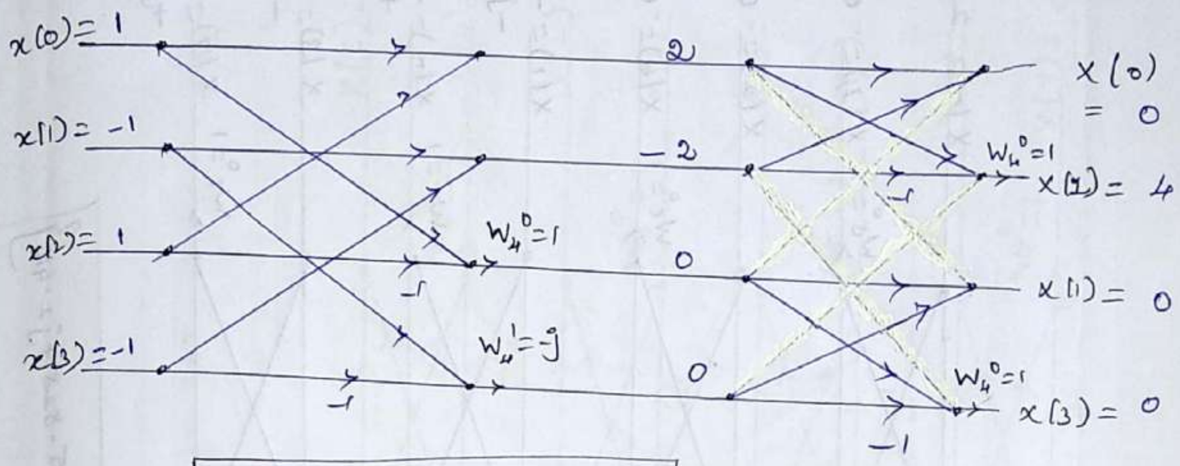
Given, $x(n) = \{1, -1, 1, -1\}$

$N = 4$

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

$$W_4^0 = 1$$

$$W_4^1 = -j$$

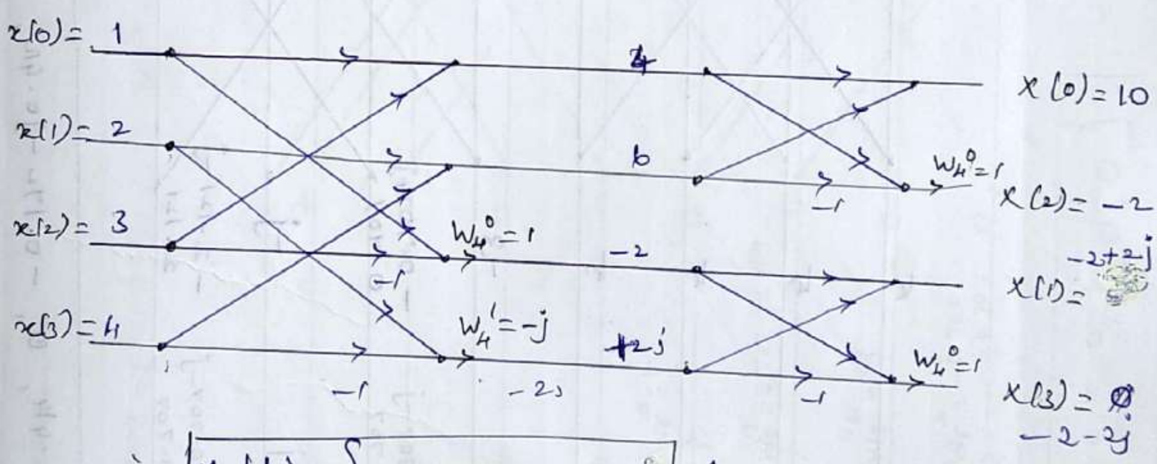


$\therefore$ $\boxed{X(k) = \{0, 0, 4, 0\}}$

14. Find DFT of the input sequence $x(n) = \{1, 2, 3, 4\}$ using DIF-FFT algorithm.

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

$W_4^0 = 1$ $W_4^1 = -j$



$\therefore$ $\boxed{X(k) = \{10, \cancel{-2+2j}, -2, -2-2j\}}$

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

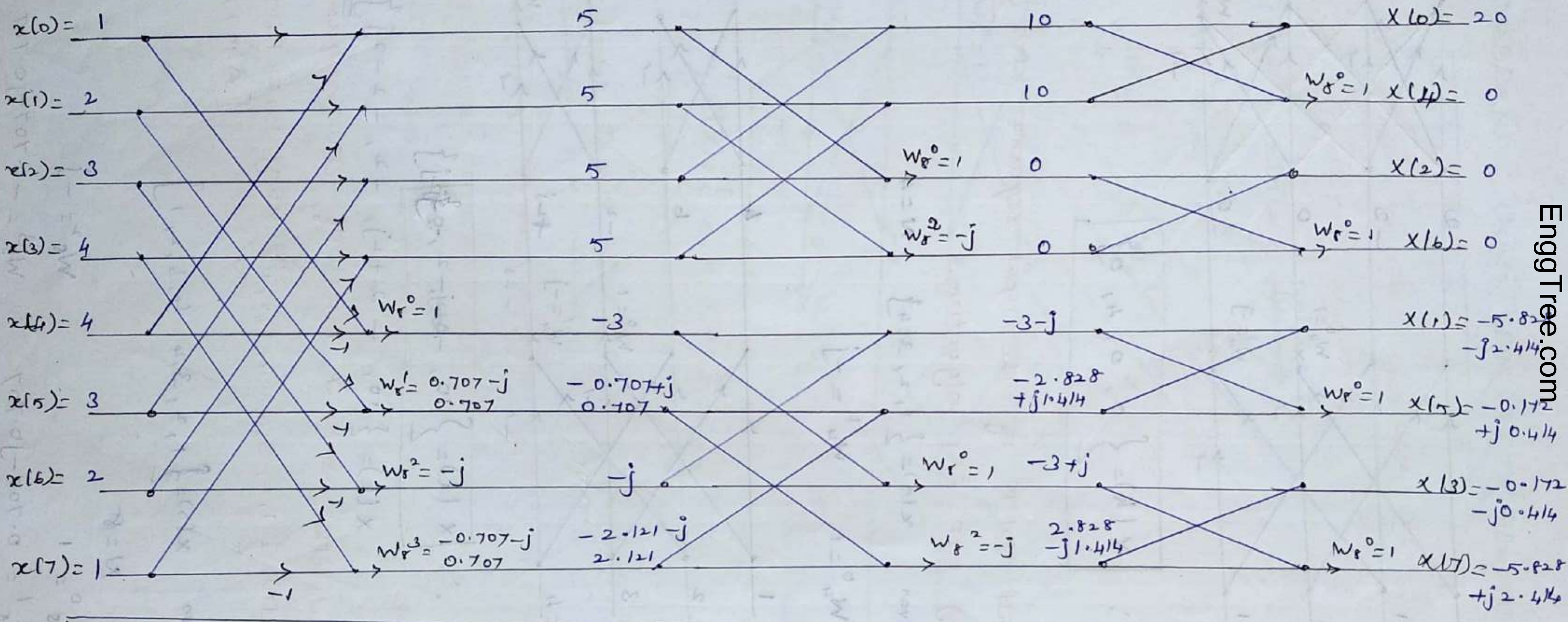
15). Find the 8-point DFT of the sequence $x(n) = \{1, 2, 3, 4, 4, 3, 2, 1\}$ using DIF-FFT algorithm. [A/M 17 R-13]

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

$N = 8$

$W_8^0 = 1$ $W_8^2 = -j$

$W_8^1 = 0.707 - j0.707$ $W_8^3 = -0.707 - j0.707$



$$\boxed{X(k) = \{20,\ -5.828 - j2.414,\ 0,\ -0.172 - j0.414,\ 0,\ -0.172 + j0.414,\ 0,\ -5.828 + j2.414\}}$$

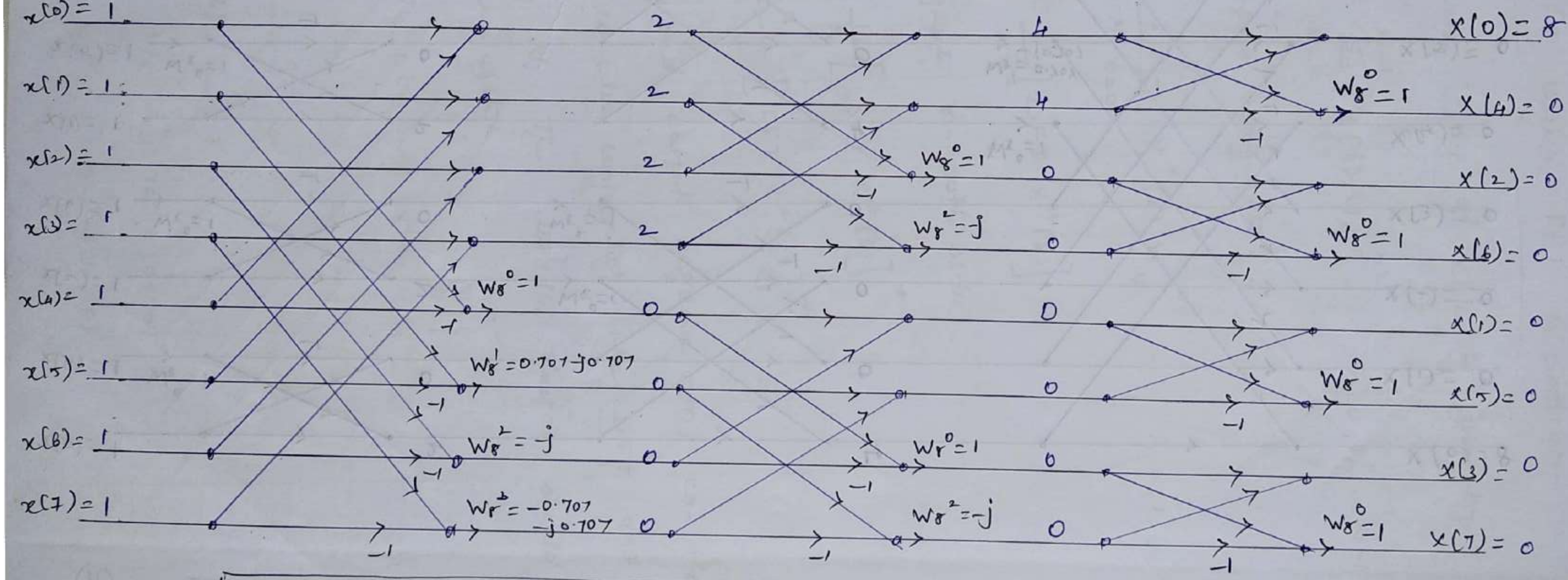
16. Compute the eight-point DFT of the sequence $x(n) = \begin{cases} 1 & 0 \le n \le 7 \\ 0 & \text{otherwise} \end{cases}$ by using DIT, DIF methods.

(i) DIF-FFT

Given, $x(n) = \begin{cases} 1 & 0 \le n \le 7 \\ 0 & \text{otherwise} \end{cases}$

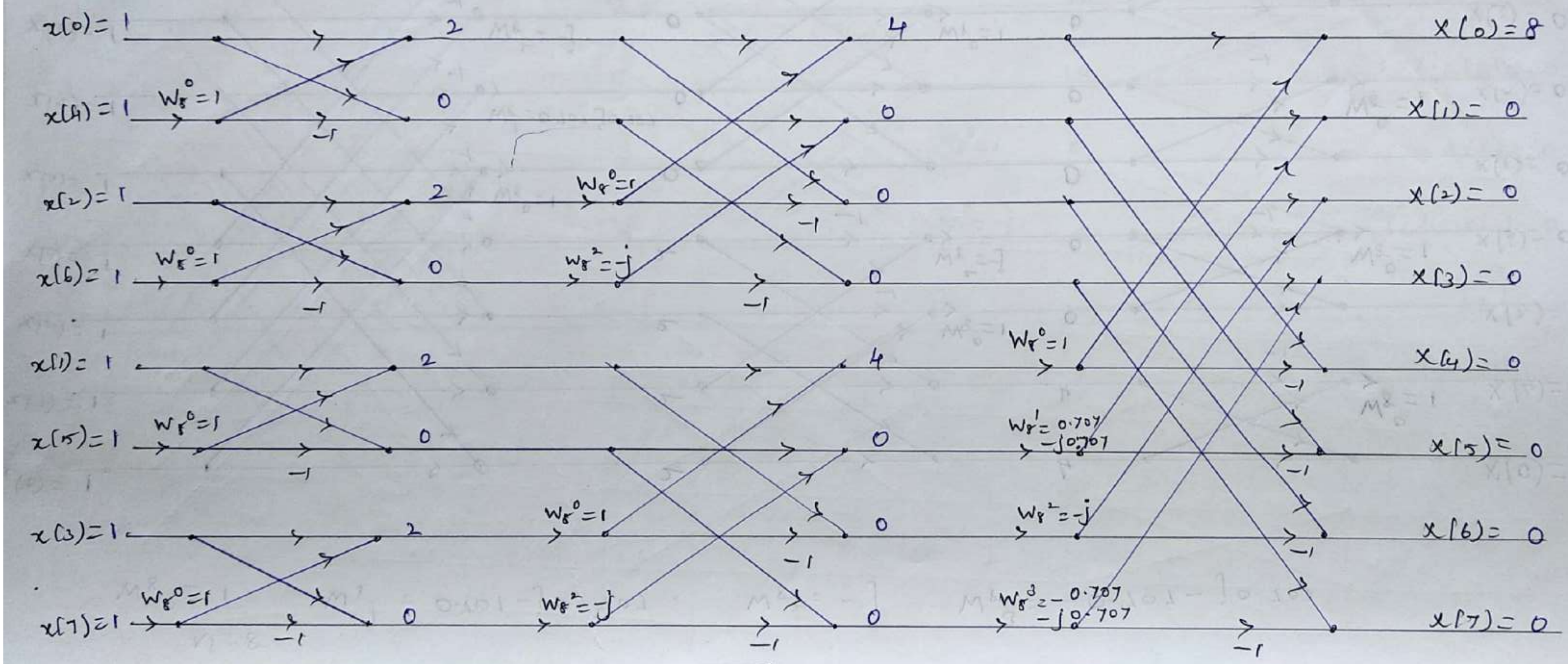
$N = 8$

$W_8^0 = 1 \qquad W_8^1 = 0.707 - j0.707 . \qquad W_8^2 = -j \qquad W_8^3 = -0.707 - j0.707$



$\therefore \boxed{X(k) = \{8, 0, 0, 0, 0, 0, 0, 0\}}$

(ii) DIT - FFT



$$\therefore \boxed{X(k) = \{8, 0, 0, 0, 0, 0, 0, 0\}}$$

INVERSE DFT USING FFT ALGORITHM

The Inverse DFT of $X(k)$ is given by

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

Take complex conjugate

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

$$x^*(n) = \frac{1}{N}\left[DFT\ X^*(k)\right]$$

Take conjugate on both sides,

$$x(n) = \frac{1}{N}\left[DFT\ X^*(k)\right]^*$$

$$\therefore x(n) = \frac{1}{N}\left[FFT\ X^*(k)\right]^*$$

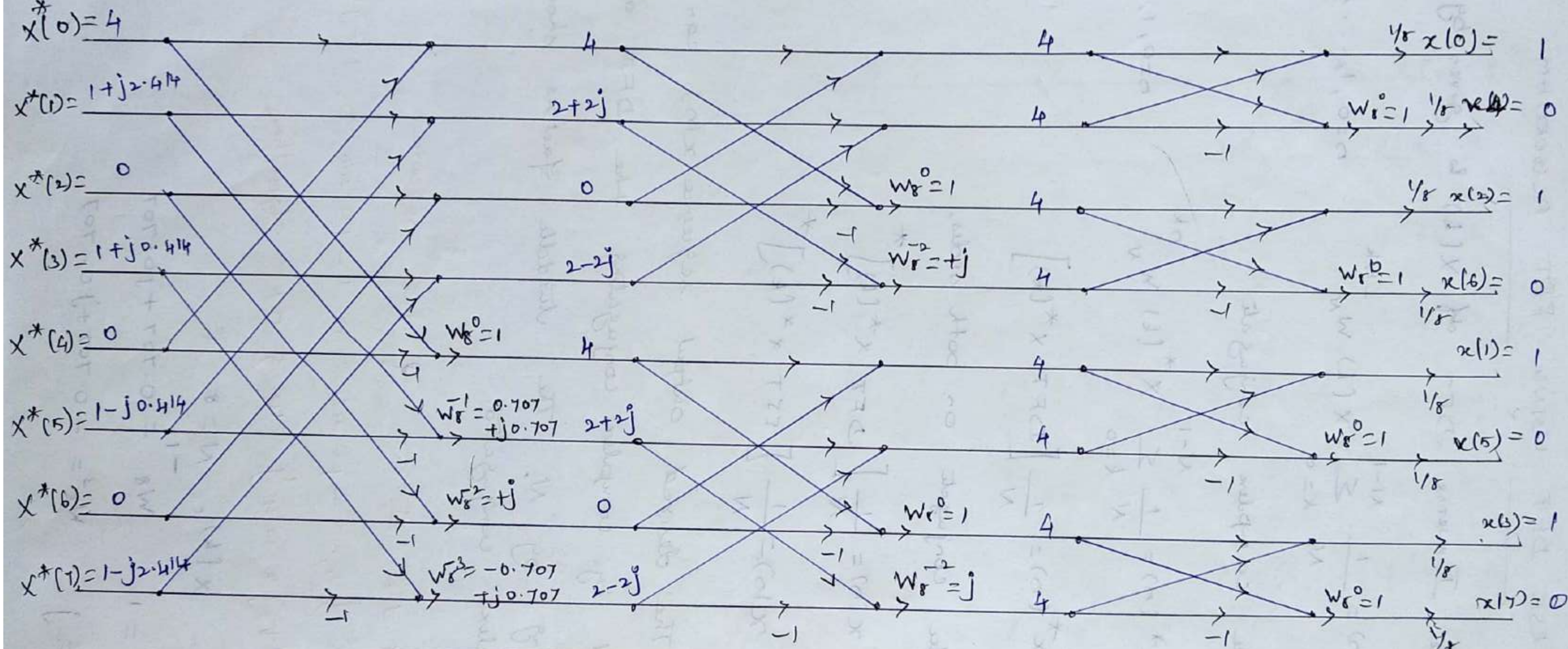
$\therefore$ The desired output sequence $x(n)$ can then be found by complex conjugating the DFT and dividing by N. The twiddle factors should be complex conjugated.

17). Find the IDFT of the sequence $X(k) = \{4,\ 1-j2.414;\ 0,\ 1-j0.414;\ 0,\ 1+j0.414,\ 0,\ 1+j2.414\}$ using DIF algorithm.

Given $X(k)$, $N=8$

$W_8^0 = 1 \qquad W_8^{-1} = 0.707 + j0.707$

$W_8^2 = j \qquad W_8^3 = -0.707 + j0.707$



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

UNIT-II

IIR FILTER DESIGN

Structures of IIR - Analog filter design - Discrete time IIR filter from analog filter - IIR filter design by Impulse Invariance, Bilinear transformation, Approximation of derivatives - (LPF, HPF, BPF, BRF) filter design using frequency translation.

T1-(701-703)

IIR Filters - Infinite Impulse Response Filters

Digital filter is a linear time invariant discrete time system.

FIR filters - Non recursive type - present output sample depends on the present input sample and previous input sample.

IIR filters - Recursive type - present output sample depends on the present input, past input and past output also.

ie $y(n) = x(n) + x(n-1) + \cdots$
$\quad + y(n-1) + y(n-2) + \cdots$

Take z transform,

$$Y(z) = b_0 X(z) + b_1 z^{-1} X(z) + \cdots$$
$$+ a_1 z^{-1} Y(z) + a_2 z^{-2} Y(z) + \cdots$$

$$Y(z) - a_1 z^{-1} Y(z) - a_2 z^{-2} Y(z) + \cdots = b_0 X(z) + b_1 z^{-1} X(z) + b_2 z^{-2} X(z) + \cdots$$

$$\frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots}{1 - a_1 z^{-1} - a_2 z^{-2} + \cdots}$$

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{k=1}^{N} a_k z^{-k}} \begin{array}{l} \longrightarrow x \\ \longrightarrow y \end{array}$$

Then the transfer function of IIR filter will be,

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{k=1}^{N} a_k z^{-k}}$$

The design of IIR filter for the given specification is finding the filter co-eff. a_k and b_k.

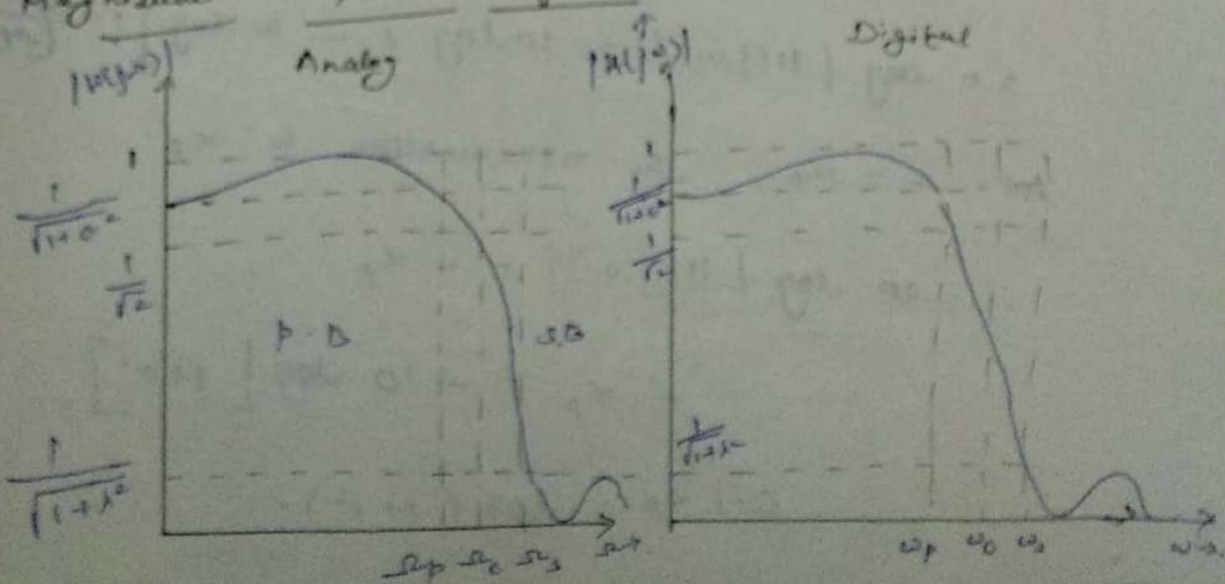
Design of digital filters from Analog filters:

Step-1 Map the desired digital filter specification into those for an equivalent analog filter.

Step-2 Derive the analog transfer function for the analog prototype.

Step-3 Transform the transfer function of the analog prototype into an equivalent digital filter transfer function.

Magnitude Response of LPF:



$\epsilon \rightarrow$ parameter specifying allowable passband $\epsilon = \frac{2\sqrt{\delta_p}}{1-\delta_p}$

$\lambda \rightarrow$ parameter specifying allowable stopband

$$\lambda = \frac{\sqrt{(1+\delta_p)^2 - \delta_s^2}}{\delta_s}$$

Analog filters → Butterworth filter / Chebyshev filter

↳ (i) Order (N)
↳ (ii) Transfer function $\left(H_a(s)\right)$

Analog Butterworth filter design: T1 (717–718)

(i) Order of the filter (N)

The magnitude function is given by,

$$\left| H(j\Omega) \right| = \frac{1}{\left[1 + \epsilon^2 \left(\Omega/\Omega_p\right)^{2N}\right]^{1/2}}$$

$$\left| H(j\Omega) \right|^2 = \frac{1}{1 + \epsilon^2 \left(\frac{\Omega}{\Omega_p}\right)^{2N}}$$

Taking log on both sides,

$$10 \log\left(\left| H(j\Omega) \right|^2\right) = 10 \log 1 - 10 \log\left[1 + \epsilon^2 \left(\frac{\Omega}{\Omega_p}\right)^{2N}\right]$$

$$20 \log \left| H(j\Omega) \right| = 10 \log 1 - 10 \log\left[1 + \epsilon^2 \left(\frac{\Omega}{\Omega_p}\right)^{2N}\right]$$

At $\Omega = \Omega_p$, the attenuation is α_p,

$$\therefore 20 \log \left| H(j\Omega) \right| = -\alpha_p$$

$$\cancel{-} \alpha_p = \cancel{-} 10 \log\left[1 + \epsilon^2\right]$$

$$0.1 \alpha_p = \log\left(1 + \epsilon^2\right)$$

$$10^{0.1 \alpha_p} = 1 + \epsilon^2$$

$$\epsilon^2 = 10^{0.1\alpha_p} - 1$$

$$\boxed{\epsilon = \sqrt{10^{0.1\alpha_p} - 1}}$$

At $\Omega = \Omega_s$, the minimum stopband attenuation is α_s.

$$\therefore 20 \log |H(j\Omega_s)| = 10 \log 1 - 10 \log \left[1 + \epsilon^2 \left(\frac{\Omega_s}{\Omega_p}\right)^{2N}\right]$$

$$\cancel{-} \alpha_s = \cancel{-} 10 \log \left(1 + \epsilon^2 \left(\frac{\Omega_s}{\Omega_p}\right)^{2N}\right)$$

$$0.1 \alpha_s = \log \left[1 + \epsilon^2 \left(\frac{\Omega_s}{\Omega_p}\right)^{2N}\right]$$

$$10^{0.1\alpha_s} = 1 + \epsilon^2 \left(\frac{\Omega_s}{\Omega_p}\right)^{2N}$$

$$\epsilon^2 \left(\frac{\Omega_s}{\Omega_p}\right)^{2N} = 10^{0.1\alpha_s} - 1$$

$$\left(\frac{\Omega_s}{\Omega_p}\right)^{2N} = \frac{10^{0.1\alpha_s} - 1}{10^{0.1\alpha_p} - 1}$$

Take log on both sides,

$$N \log \left(\frac{\Omega_s}{\Omega_p}\right) = \log \sqrt{\frac{10^{0.1\alpha_s} - 1}{10^{0.1\alpha_p} - 1}}$$

$$N = \frac{\log \sqrt{\frac{10^{0.1\alpha_s} - 1}{10^{0.1\alpha_p} - 1}}}{\log \left(\frac{\Omega_s}{\Omega_p}\right)}$$

The above expression doesnot results in an integer value.

$$\therefore \boxed{N \geq \frac{\log \left(\sqrt{\frac{10^{0.1\alpha_s} - 1}{10^{0.1\alpha_p} - 1}}\right)}{\log \left(\frac{\Omega_s}{\Omega_p}\right)}}$$

$$N \geq \frac{\log \frac{\lambda}{\epsilon}}{\log \frac{\Omega_s}{\Omega_p}}$$

where $\lambda = \sqrt{10^{0.1\alpha_s} - 1}$

$$\epsilon = \sqrt{10^{0.1\alpha_p} - 1}$$

if $A = \frac{\lambda}{\epsilon}$ and $k = \frac{\Omega_p}{\Omega_s}$

then, $$\boxed{N \geq \frac{\log A}{\log (1/k)}}$$

(ii) Transfer function of Butterworth filter:

The magnitude square function of a Butterworth filter is,

$$|H(j\Omega)|^2 = \frac{1}{1+\left(\frac{\Omega}{\Omega_c}\right)^{2N}} \quad \text{——— (1)}$$

For normalized cut-off frequency,

$$\Omega_p = \Omega_c = 1 \text{ rad/sec.}$$

$$\therefore \quad |H(j\Omega)|^2 = H(\Omega^2)$$

$$= H(-s^2) \qquad / \because \Omega = \frac{s}{j}$$

$$= H(s)\,H(-s)$$

$$\therefore H(s)\,H(-s) = \frac{1}{1+\left(\frac{s}{j}\right)^{2N}}$$

$$= \frac{1}{1+(-1)^N s^{2N}}$$

$$H(s)\,H(-s) = \frac{1}{1+(-s^2)^N} \quad \text{——— (2)}$$

The above equations tells that it has poles in the LHP (Left Half of a Plane) as well as in the RHP (Right Half of a Plane), because of the two factors $H(s)$ and $H(-s)$.

If $H(s)$ has roots in the LHP then $H(-s)$ has the corresponding roots in the RHP.

The roots can be obtained by equating the denominator to zero.

ie, $1 + (-s^2)^N = 0$. —— (3)

For N-odd, the above eqn reduces to

$$s^{2N} = 1 = e^{j2\pi k}$$

$$s^N = e^{j\pi k}$$

$$\boxed{s_k = e^{j\frac{\pi k}{N}}} \quad k = 1, 2, \ldots 2N$$

For N-even, the eqn (3) reduces to,

$$s^{2N} = -1 = e^{j(2k-1)\pi}$$

$$\therefore \boxed{s_k = e^{j(2k-1)\frac{\pi}{2N}}} \quad \text{for } k = 1, 2, \ldots 2N$$

For $N = 3$, $s^6 = 1$

For N-odd,

$$s_1 = e^{j\frac{\pi}{3}} = \cos\frac{\pi}{3} + j\sin\frac{\pi}{3} = 0.5 + j0.866$$

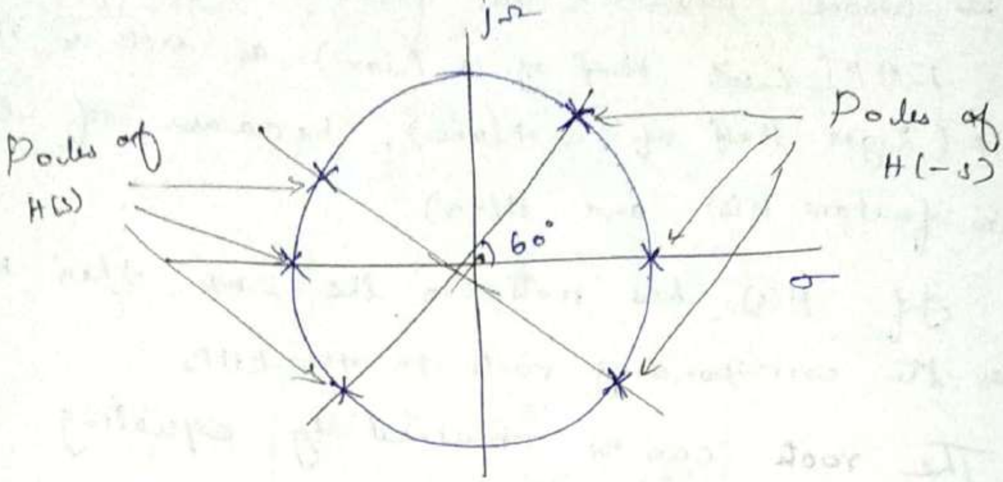
$$s_2 = e^{j\frac{2\pi}{3}} = -0.5 + j0.866$$

$$s_3 = e^{j\pi} = -1$$

$$s_4 = e^{j\frac{4\pi}{3}} = -0.5 - j0.866$$

$$s_5 = e^{j\frac{5\pi}{3}} = 0.5 - j0.866$$

$$s_6 = e^{j2\pi} = 1$$



All the poles are located in the s-plane. The angular seperation between the poles is $\frac{360°}{2N}$ ie 60° and all the poles lie on a unit circle.

To ensure stability, only the poles that lie on the left half of the s-plane are considered.

∴ The denominator of the transfer function H(s) as,

$$(s+1)\{(s+0.5)^2+(0.866)^2\}$$

$$=(s+1)(s^2+s+1)$$

∴ H(s), the transfer function of the 3rd order Butterworth filter for cutoff frequency $\Omega_c = 1$ rad/sec

$$H(s) = \frac{1}{(s+1)(s^2+s+1)}$$

As we are interested on the poles, which lies in the left half of the s-plane, the same can be found by using

$$\boxed{\begin{aligned} s_k &= e^{j\varphi_k} \\ \text{where, } \varphi_k &= \frac{\pi}{2} + \frac{(2k-1)\pi}{2N} \qquad k=1,2,\ldots N \end{aligned}} \quad \text{— (6)}$$

List of Butterworth Polynomials

N	Denominator of $H(s)$
1	$s+1$
2	$s^2+\sqrt{2}s+1$
3	$(s+1)(s^2+s+1)$
4	$(s^2+0.76537s+1)(s^2+1.8477s+1)$
5	$(s+1)(s^2+0.61803s+1)(s^2+1.61803s+1)$
6	$(s^2+1.931855s+1)(s^2+\sqrt{2}s+1)(s^2+0.51764s+1)$
7	$(s+1)(s^2+1.80194s+1)(s^2+1.247s+1)(s^2+0.445s+1)$

These poles are called normalized poles, because $\Omega_c = 1$ rad/sec.

In general, the unnormalized poles are given by $s_k' = \Omega_c s_k$.

The transfer function of such type of Butterworth filter can be obtained by substituting $s \rightarrow \frac{s}{\Omega_c}$ in the transfer function of Butterworth filter.

Frequency Transformation

① Low pass ($\Omega_c = 1$) filter to Low pass filter new Ω_c $\quad s \rightarrow \frac{s}{\Omega_c}$

② Low pass filter to High pass filter $\quad s \rightarrow \frac{\Omega_c}{s}$

③ Low pass filter to Band pass filter $\quad s \rightarrow \Omega_c \frac{s^2+\Omega_1\Omega_2}{s(\Omega_2-\Omega_1)}$

④ Low pass filter to Band stop filter $\quad s \rightarrow \Omega_c \frac{s(\Omega_2-\Omega_1)}{s^2+\Omega_1\Omega_2}$

1) Determine the order and the poles of lowpass Butterworth filter that has a 3dB attenuation at 500Hz and an attenuation of 40dB at 1000Hz.

Soln:

Given $\alpha_p = 3dB$ $f_p = 500Hz$

$\alpha_s = 40dB$ $f_s = 1000Hz$

$\Omega_p = 2 \times \pi \times 500 = 1000\pi$ rad/sec

$\Omega_s = 2 \times \pi \times 1000 = 2000\pi$ rad/sec.

$$\text{Order}, N \geq \frac{\log \sqrt{\frac{10^{0.1\alpha_s}-1}{10^{0.1\alpha_p}-1}}}{\log \frac{\Omega_s}{\Omega_p}}$$

$$\geq \frac{\log \sqrt{\frac{10^4-1}{10^{0.3}-1}}}{\log \frac{2000\pi}{1000\pi}}$$

≥ 6.6

$\boxed{N = 7}$

2). Prove that $\Omega_c = \frac{\Omega_p}{\left(10^{0.1\alpha_p}-1\right)^{1/2N}} = \frac{\Omega_s}{\left(10^{0.1\alpha_s}-1\right)^{1/2N}}$

Magnitude square function of Butterworth analog LPF is,

$$|H(j\Omega)|^2 = \frac{1}{1+\left(\frac{\Omega}{\Omega_c}\right)^{2N}} \quad \text{--- (1)}$$

also,

$$|H(j\Omega)|^2 = \frac{1}{1+\epsilon^2\left(\frac{\Omega}{\Omega_p}\right)^{2N}} \quad \text{--- (2)}$$

compare (1) and (2) we,

$$1+\left(\frac{\Omega}{\Omega_c}\right)^{2N} = 1+\epsilon^2\left(\frac{\Omega}{\Omega_p}\right)^{2N}$$

$$\epsilon^2\left(\frac{\Omega}{\Omega_p}\right)^{2N} = \left(\frac{\Omega}{\Omega_c}\right)^{2N}$$

$$\epsilon^2 = \left(\frac{\Omega_p}{\Omega_c}\right)^{2N}$$

$$10^{0.1\alpha_p}-1 = \left(\frac{\Omega_p}{\Omega_c}\right)^{2N}$$

$$\Omega_c^{2N} = \frac{\Omega_p^{2N}}{10^{0.1\alpha_p}-1}$$

$$\boxed{\Omega_c = \frac{\Omega_p}{\left(10^{0.1\alpha_p}-1\right)^{1/2N}}} \qquad \boxed{\Omega_c = \frac{\Omega_p}{\epsilon^{1/N}}}$$

$$\left(\frac{\Omega_s}{\Omega_p}\right)^{2N} = \frac{10^{0.1\alpha_s}-1}{10^{0.1\alpha_p}-1}$$

$$\Omega_s = \Omega_p\left[\frac{10^{0.1\alpha_s}-1}{10^{0.1\alpha_p}-1}\right]^{1/2N}$$

$$= \Omega_c\left[\cancel{10^{0.1\alpha_p}-1}\right]^{1/2N} \cdot \frac{\left[10^{0.1\alpha_s}-1\right]^{1/2N}}{\left[\cancel{10^{0.1\alpha_p}-1}\right]^{1/2N}}$$

$$\boxed{\Omega_c = \frac{\Omega_s}{\left[10^{0.1\alpha_s}-1\right]^{1/2N}}}$$

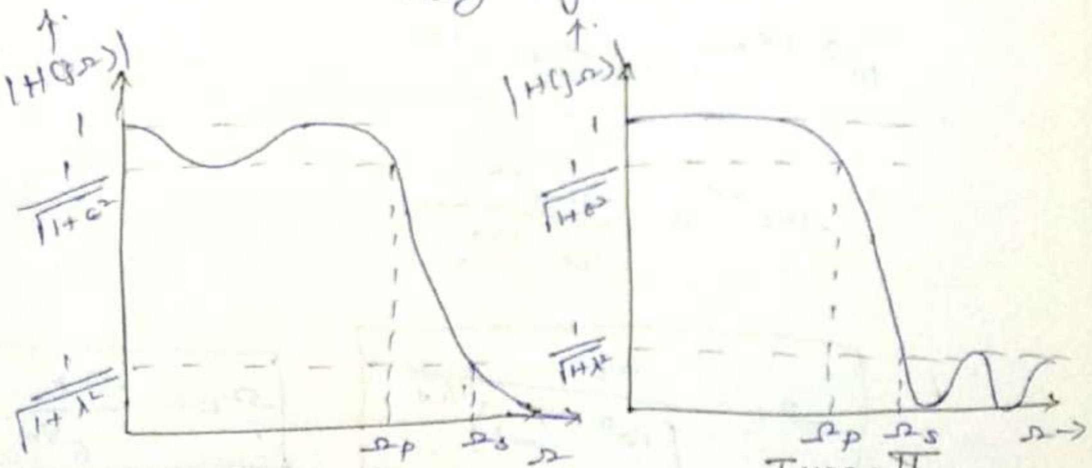
Chebyshev Filters

T1 (419 – 723)

Two types of Chebyshev filters:

Type I — All pole filters that exhibit equiripple behavior in the passband and monotonic characteristics in the stopband.

Type-II — Contains both poles and zeros and exhibits a monotonic behavior in the pass band and an equiripple behavior in the stopband. Zeros lie on the imaginary axis.



Type-I Type-II

The magnitude square function of N^{th} order Type-I filter can be expressed as,

$$|H(j\Omega)|^2 = \frac{1}{1+\epsilon^2 C_N^2\left(\frac{\Omega}{\Omega_p}\right)} \qquad N = 1, 2, \ldots \qquad (1)$$

where $\epsilon \rightarrow$ parameter related to the ripple in the passband

$C_N(x) \rightarrow N^{th}$ order Chebyshev polynomial

$$C_N(x) = \begin{cases} \cos\left[N \cos^{-1} x\right] & |x| < 1 \rightarrow \text{passband} \\ \cosh(N \cosh^{-1} x) & |x| > 1 \rightarrow \text{stopband} \end{cases}$$

$$C_N(x) = 2x\, C_{N-1}(x) - C_{N-2}(x) \qquad N > 1$$

$$C_0(x) = 1 \qquad C_1(x) = x$$

Properties of Chebyshev polinomial:

I
① $C_N(x) = -C_N(-x)$, N odd

② $C_N(x) = C_N(-x)$, N even

③ $C_N(0) = (-1)^{N/2}$, N even

④ $C_N(0) = 0$, N odd

5) $C_N(1) = 1$ $\forall N$

6) $C_N(-1) = 1$ N even

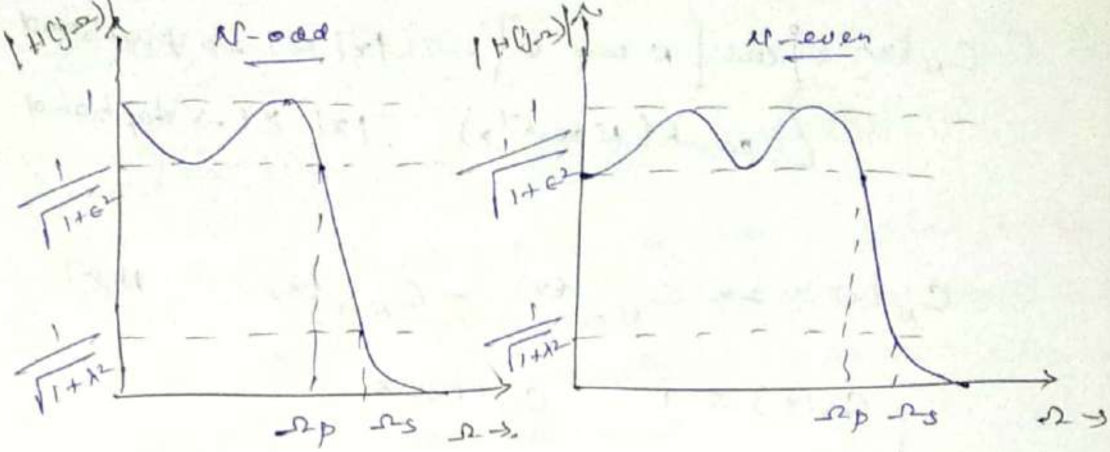
7) $C_N(-1) = -1$ N odd

II $C_N(x)$ oscillates with equal ripple between ± 1 for $|x| \leq 1$

III For all N $0 \leq |C_N(x)| \leq 1$ for $0 \leq |x| \leq 1$

$|C_N(x)| > 1$ for $|x| \geq 1$

IV $C_N(x)$ is monotonically increasing for $|x| > 1$ for all N.



For odd values of N, the oscillatory curve starts from unity.

For even values of N, the oscillatory curve starts from $\frac{1}{\sqrt{1+\epsilon^2}}$

Type-I

Order

$$\boxed{N \geq \frac{\cosh^{-1}\left(\frac{10^{0.1\alpha_s}-1}{10^{0.1\alpha_p}-1}\right)^{1/2}}{\cosh^{-1}\left(\frac{\Omega_s}{\Omega_p}\right)}}$$

(or)

$$\boxed{N \geq \frac{\cosh^{-1} A}{\cosh^{-1}(1/k)}}$$

Pole Locations

$$s_k = a\cos\phi_k + jb\sin\phi_k$$

$$s_k = \sigma_k + j\Omega_k \qquad k = 1, 2, \dots N$$

where, $a = \Omega_p\left[\frac{\mu^{1/N} - \mu^{-1/N}}{2}\right]$ $\qquad \mu = \epsilon^{-1} + \sqrt{\epsilon^{-2}+1}$

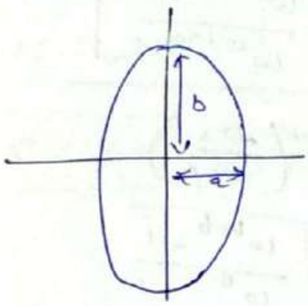
$$b = \Omega_p\left[\frac{\mu^{1/N} + \mu^{-1/N}}{2}\right]$$

$$\phi_k = \frac{\pi}{2} + \frac{(2k-1)\pi}{2N} \qquad k = 1, 2, \dots N$$

The poles of the chebyshev transfer function are located on an ellipse in the s-plane. The equation of the ellipse is given by,

$$\frac{\sigma_k^2}{a^2} + \frac{\Omega_k^2}{b^2} = 1$$

a, b are minor and major axis of the ellipse.



Type-II

It has both poles and zeros.

$$|H(j\Omega)|^2 = \frac{1}{1+\epsilon^2\left[\dfrac{C_N^2\left(\frac{\Omega_s}{\Omega_p}\right)}{C_N^2\left(\frac{\Omega_s}{\Omega}\right)}\right]}$$

The zeros are located on the imaginary axis at the points,

$$s_k = j\frac{\Omega_s}{\sin\phi_k} \qquad k = 1, 2, \dots N$$

Poles are located at the points (x_k, y_k)

$$x_k = \frac{\Omega_s \sigma_k}{\sigma_k^2 + \Omega_k^2} \qquad k = 1, 2, \dots N$$

$$\sigma_k = a\cos\phi_k$$

$$y_k = \frac{\Omega_s \Omega_k}{\sigma_k^2 + \Omega_k^2} \qquad \Omega_k = b\sin\phi_k$$

$$\mu = \lambda + \sqrt{1+\lambda^2}$$

$$\boxed{\text{Order } N = \frac{\cosh^{-1}\left(\frac{\lambda}{\epsilon}\right)}{\cosh^{-1}\left(\frac{\Omega_s}{\Omega_p}\right)}}$$

(3) Given the specifications $\alpha_p = 3\,dB$, $\alpha_s = 6\,dB$, $f_p = 1\,KHz$ and $f_s = 2\,KHz$. Determine the order of the Chebyshev filter.

Soln:

Given $\alpha_p = 3\,dB$ $\quad f_p = 1\,KHz$

$\alpha_s = 6\,dB$ $\quad f_s = 2\,KHz$.

$$\text{order } N \geq \frac{\cosh^{-1}\left[\sqrt{\dfrac{10^{0.1\alpha_s}-1}{10^{0.1\alpha_p}-1}}\right]}{\cosh^{-1}\left(\dfrac{\Omega_s}{\Omega_p}\right)}$$

$$\geq \frac{\cosh^{-1}\sqrt{\dfrac{10^{0.6}-1}{10^{0.3}-1}}}{\cosh^{-1}\left(\dfrac{4000\pi}{2000\pi}\right)}$$

$$\geq \cancel{1.91}\ 0.87$$

$$\boxed{N = 1}$$ $\rightarrow$ Oscillatory curve starts from 1. $\cancel{\frac{1}{\sqrt{1+\epsilon^2}}}$

4). Find the order of a Chebyshev filter with maximum passband attenuation of 2.5 dB at $\Omega_p = 20\,rad/sec$ and the stopband attenuation of 30 dB at $\Omega_s = 50\,rad/sec$.

(M/J - 07)

Soln:

Given,

$\Omega_p = 20\,rad/sec$ $\quad \alpha_p = 2.5\,dB$

$\Omega_s = 50\,rad/sec$ $\quad \alpha_s = 30\,dB$.

$$N \geq \frac{\cosh^{-1} \frac{\lambda}{\epsilon}}{\cosh^{-1} 1/k} \qquad N \geq \frac{\cosh^{-1}\left[\sqrt{\frac{10^{0.1\alpha_s}-1}{10^{0.1\alpha_p}-1}}\right]}{\cosh^{-1}\left(\frac{\Omega_s}{\Omega_p}\right)}$$

$$N = \sqrt{}$$

$$N \geq \frac{\cosh^{-1}\left(\frac{31.607}{0.882}\right)}{\cosh^{-1}\left(\frac{1}{0.4}\right)}$$

$$N \geq 2.726$$

$\boxed{N = 3}$ → Oscillatory curve starts from.

Steps to design an analog Butterworth LPF:-

1. From the given specifications, find the order of the filter N.
2. Round off it to the next higher integer.
3. Find the Transfer Function $H(s)$ for $\Omega_c = 1$ rad/sec for the value of N.
4. Calculate the value of cut-off frequency Ω_c from,

$$\Omega_c = \frac{\Omega_p}{\left[10^{0.1\alpha_p} - 1\right]^{1/2N}}$$

5. Find the transfer function $H_a(s)$ for the above value of Ω_c by substituting $s \to \frac{s}{\Omega_c}$ in $H(s)$

Steps to design an analog Chebyshev LPF:-

① From the given specifications find the order of the filter N.

2). Round off it to the next higher integer.

3). Using the following formulae find the values of a and b, which are minor and major axis of the ellipse respectively.

$$a = \Omega_p \left[\frac{\mu^{1/N} - \mu^{-1/N}}{2} \right] \quad b = \Omega_p \left[\frac{\mu^{1/N} + \mu^{-1/N}}{2} \right]$$

where $\mu = \epsilon^{-1} + \sqrt{\epsilon^{-2}+1}$ $\quad \epsilon = \sqrt{10^{0.1\alpha_p} - 1}$

For normalized Chebyshev filter $\Omega_p = 1$ rad/sec

4). Calculate the poles of Chebyshev filter which lie on an ellipse by using

$$S_k = a \cos \phi_k + j b \sin \phi_k$$

where $\phi_k = \frac{\pi}{2} + \frac{(2k-1)\pi}{2N}$ $\quad k = 1, 2, \ldots N$

5). Find the denominator polynomial of the Transfer Function using the above poles.

6). The numerator of the Transfer Function depends on the value of N.

a) For N-odd, substitute $s=0$ in $D(s)$ and find the value. This value is the numerator of TF.

b). For N-even, substitute $s=0$ in $D(s)$ and divide the result by $\sqrt{1+\epsilon^2}$. This value is the numerator of TF.

Design of digital IIR filters

~~Three~~ Four techniques:

(i) Impulse Invariance

(ii) Bilinear Transformation

(iii) Approximation of derivatives.

(iv) Matched z-transform.

Requirements for digital filters to be stable:

(i) The $j\Omega$ axis in the s-plane should map into the unit circle in the z-plane. Thus there will be a direct relationship b/w the two frequency variables in the two domains.

(ii) The LHP of the s-plane should map into the inside of the unit circle in the z-plane. Thus a stable analog filter will be converted to a stable digital filter.

Steps to design digital filters from Analog filters:-

① Map the desired digital filter specifications into those for an equivalent analog filter.

② Derive the analog transfer function for the analog prototype.

3). Transform the transfer function of the analog prototype into an equivalent digital filter transfer function.

T₁ (707 – 716)

IIR filter design by Impulse Invariance Method

The objective is to design an IIR filter having a unit sample response $h(n)$, that is the sampled version of the impulse response of the analog filter.

ie, $h(n) \equiv h(nT) \quad n = 0, 1, 2, \ldots$

where $T \rightarrow$ sampling interval.

From sampling theorem,

$$X(f) = F_s \sum_{k=-\infty}^{\infty} X_a \left[(f-k) F_s \right]$$

where $f = F/F_s$ is the normalized frequency.

Aliasing occurs if the sampling rate F_s less than twice the highest frequency contained in $X_a(F)$.

For impulse response,

$$H(f) = F_s \sum_{k=-\infty}^{\infty} H_a \left[(f-k) F_s \right] \quad (or)$$

$$H(\omega) = F_s \sum_{k=-\infty}^{\infty} H_a \left[(\omega - 2\pi k) F_s \right] \quad (or)$$

$$H(\Omega T) = \frac{1}{T} \sum_{k=-\infty}^{\infty} H_a \left(\Omega - \frac{2\pi k}{T} \right).$$

It is clear that the digital filter with frequency response $H(\omega)$ has the frequency response characteristics of the corresponding analog filter if the sampling interval T is selected sufficiently small, to completely avoid aliasing.

(x) It is also clear that impulse invariance method is inappropriate for designing highpass filters due to the spectrum aliasing that results from the sampling process.

Mapping of points b/w z-plane and s-plane:-

Let $z = e^{sT}$

$s = \sigma + j\Omega \qquad z = re^{j\omega}$

$\therefore re^{j\omega} = e^{\sigma T + j\Omega T}$

$re^{j\omega} = e^{\sigma T} e^{j\Omega T}$

$\therefore \omega, \; r = e^{\sigma T}$

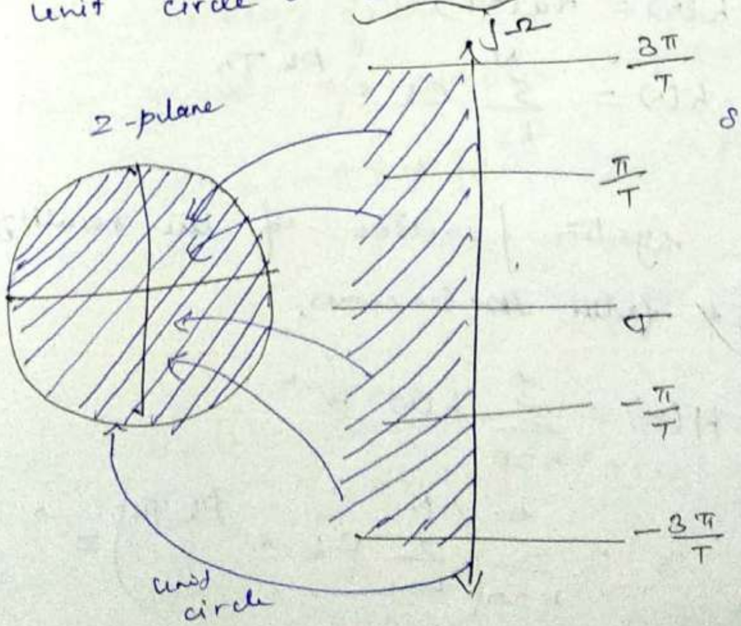
$\omega = \Omega T$

when $\sigma < 0 \Rightarrow 0 < r < 1$

$\sigma > 0 \Rightarrow r > 1$

$\sigma = 0 \Rightarrow r = 1$

ie, LHP in s-plane is mapped inside the unit circle in z and the RHP in s plane is mapped outside the unit circle in z. Also $j\Omega$ axis is mapped into the unit circle in z. However the mapping of the $j\Omega$ axis into the unit circle is $\underbrace{\text{not one-to-one}}$. (Many to one mapping)



Since ω is unique over the range $(-\pi, \pi)$ the mapping $\omega = \Omega T$ implies that the interval $\frac{-\pi}{T} \leq \Omega \leq \frac{\pi}{T}$ maps into the corresponding values of $-\pi \leq \omega \leq \pi$.

The frequency interval $\frac{\pi}{T} \leq \Omega \leq \frac{3\pi}{T}$ also maps into the interval $-\pi \leq \omega \leq \pi$ and so on.

Let us assume the system function of the analog filter as in partial fraction form.

$$H_a(s) = \sum_{k=1}^{N} \frac{C_k}{s - p_k}$$

where $\{p_k\}$ are the poles of the analog filter and $\{C_k\}$ are the co-efficients in the partial fraction expansion.

$$h_a(t) = \sum_{k=1}^{N} C_k e^{p_k t} \qquad t \geq 0$$

If we sample $h_a(t)$ at $t = nT$,

$$h(n) = h_a(nT)$$

$$h(n) = \sum_{k=1}^{N} C_k e^{p_k T n}$$

$\therefore$ The system function of the resulting digital IIR filter ~~des~~ becomes,

$$H(z) = \sum_{n=0}^{\infty} h(n) z^{-n}$$

$$= \sum_{n=0}^{\infty} \left(\sum_{k=1}^{N} C_k e^{p_k T n} \right) z^{-n}$$

$$= \sum_{k=1}^{N} C_k \sum_{n=0}^{\infty} \left(e^{P_k T} z^{-1}\right)^n$$

$$= \sum_{k=1}^{N} C_k \cdot \frac{1}{1 - e^{P_k T} z^{-1}} \qquad \left[\because \sum_{n=0}^{\infty} a^n = \frac{1}{1-a}\right.$$

$\therefore$ The system function of the digital filter is,

$$H(z) = \sum_{k=1}^{N} \frac{C_k}{1 - e^{P_k T} z^{-1}}$$

The digital filter has poles at $z_k = e^{P_k T}$

$k = 1, 2 \ldots N.$

(1) Convert the analog filter with system function

$$H_a(s) = \frac{s + 0.1}{(s+0.1)^2 + 9}$$

into a digital filter by means of the impulse invariance method.

Soln

Given $H_a(s) = \dfrac{s+0.1}{(s+0.1)^2+9} = \dfrac{s+0.1}{s^2 + 0.2s + 0.01 + 9}$

This analog filter has a zero at $s = -0.1$ and a pair of complex conjugate pole at

$$P_k = -0.1 \pm j3$$

$$H_a(s) = \frac{s+0.1}{(s+0.1)^2 + 3^2} = \frac{s+0.1}{(s+0.1+3j)(s+0.1-3j)}$$

$$\therefore H_a(s) = \frac{s+0.1}{(s+0.1-j3)(s+0.1+j3)}$$

$$H_a(s) = \frac{A}{s+0.1-j3} + \frac{B}{s+0.1+j3}$$

$$A = \left.\frac{s+0.1}{s+0.1+j3}\right|_{s=-0.1+j3}$$

$$= \frac{-\cancel{0.1}+j3+\cancel{0.1}}{-\cancel{0.1}+j3+\cancel{0.1}+j3}$$

$$= \frac{j\cancel{3}}{j\cancel{6}_2}$$

$$\boxed{A = \frac{1}{2}}$$

$$B = \left.\frac{s+0.1}{s+0.1-j3}\right|_{s=-0.1-j3}$$

$$= \frac{-\cancel{0.1}-j3+\cancel{0.1}}{-\cancel{0.1}-j3+\cancel{0.1}-j3}$$

$$= \frac{-j3}{-j6}$$

$$\boxed{B = 1/2}$$

$$\therefore H_a(s) = \frac{1/2}{s+0.1-j3} + \frac{1/2}{s+0.1-j3}$$

Then $H(z) = \dfrac{1/2}{1-e^{-0.1T}e^{j3T}z^{-1}} + \dfrac{1/2}{1-e^{-0.1T}e^{-3jT}z^{-1}}$

$$\boxed{H(z) = \frac{1-(e^{-0.1T}\cos 3T)z^{-1}}{1-(2e^{-0.1T}\cos 3T)z^{-1}+e^{-0.2T}z^{-1}}}$$

$$H(z) = \frac{1/2\left[1-e^{0.1T}e^{-j3T}z^{-1}\right]+1/2\left[1-e^{-0.1T}e^{j3T}z^{-1}\right]}{1-e^{-0.1T}e^{-j3T}z^{-1}-e^{-0.1T}e^{j3T}z^{-1}+e^{-0.2T}z^{-2}}$$

$$= \frac{\cancel{\tfrac{1}{2}}\left[\cancel{2}-e^{-0.1T}(\cancel{2}\cos 3T)z^{-1}\right]}{1-2e^{-0.1T}(\cos 3T)z^{-1}+e^{-0.2T}z^{-2}}$$

2). For the analog transfer function $H(s) = \frac{2}{(s+1)(s+2)}$ determine the $H(z)$ using impulse invariance method. Assume $T = 1$ sec. (N/D-14)

Given $H(s) = \frac{2}{(s+1)(s+2)}$

Using partial fraction,

$$H(s) = \frac{A}{s+1} + \frac{B}{s+2}$$

$$A = \frac{2}{s+2}\Big|_{s=-1} = 2$$

$$B = \frac{2}{s+1}\Big|_{s=-2} = -2$$

$$\therefore H(s) = \frac{2}{s+1} + \frac{-2}{s+2}$$

Using impulse invariance technique, we have

if $H(s) = \sum_{k=1}^{N} \frac{C_k}{s-p_k}$ then $H(z) = \sum_{k=1}^{N} \frac{C_k}{1-e^{p_kT}z^{-1}}$

$$\therefore H(z) = \frac{2}{1-e^{-T}z^{-1}} - \frac{2}{1-e^{-2T}z^{-1}}$$

For $T = 1$ sec,

$$H(z) = \frac{2}{1-e^{-1}z^{-1}} - \frac{2}{1-e^{-2}z^{-1}}$$

$$= \frac{2}{1-0.3678z^{-1}} - \frac{2}{1-0.1353z^{-1}}$$

$$\boxed{H(z) = \frac{0.465z^{-1}}{1-0.503z^{-1}+0.04976z^{-2}}}$$

3) An analog filter has a transfer function $H(s) = \frac{10}{s^2+7s+10}$. Design a digital filter equivalent to this using impulse invariant method for $T = 0.2$ sec.

Given $H(s) = \frac{10}{s^2+7s+10}$ $T = 0.2sec.$

$$H(s) = \frac{10}{(s+2)(s+5)}$$

$$= \frac{A}{s+2} + \frac{B}{s+5}$$

$$A = \frac{10}{(s+5)\cancel{(s+2)}}\Bigg|_{s=-2} = \frac{10}{(3)} = 3.33$$

$$B = \frac{10}{s+2}\Bigg|_{s=-5} = \frac{10}{-3} = -3.33$$

$$H(s) = \frac{3.33}{s+2} - \frac{3.33}{s+5}$$

Form impulse invariance method,

If $H(s) = \sum_{k=1}^{N} \frac{Ck}{s-Pk}$ then $H(z) = \sum_{k=1}^{N} \frac{TCk}{1-e^{PkT}z^{-1}}$ For high sampling rate

$$\therefore H(z) = \left[\frac{3.33}{1-e^{-2T}z^{-1}} - \frac{3.33}{1-e^{-5T}z^{-1}}\right]0.2$$

$$= \frac{0.666}{1-0.3678z^{-1}} - \frac{0.666}{1-0.67z^{-1}}$$

$$H(z) = \frac{0.2012z^{-1}}{1-1.0378z^{-1}+0.247z^{-2}}$$

4). Design a digital Butterworth filter satisfying the constraints,

$$0.707 \le |H(e^{j\omega})| \le 1 \qquad 0 \le \omega \le \frac{\pi}{2}$$

$$|H(e^{j\omega})| \le 0.2 \qquad \frac{3\pi}{4} \le \omega \le \pi$$

with T = 1sec using impulse invariance method.

Step-1

Given,

$$\frac{1}{\sqrt{1+\epsilon^2}} = 0.707 = \frac{1}{\sqrt{2}} \qquad \frac{1}{\sqrt{1+\lambda^2}} = 0.2$$

$$1+\epsilon^2 = 2 \qquad \frac{1}{1+\lambda^2} = 0.04$$

$$\epsilon^2 = 1 \qquad 1+\lambda^2 = 25$$

$$\epsilon = \pm 1 \qquad \Rightarrow \lambda^2 = 24$$

Let $\epsilon = 1$ $\qquad \lambda = 4.898$

$$\omega_p = \frac{\pi}{2} \Rightarrow \Omega_p = \frac{\omega_p}{T} = \frac{\pi}{2}$$

$$\omega_s = \frac{3\pi}{4} \Rightarrow \Omega_s = \frac{\omega_s}{T} = \frac{3\pi}{4}$$

Step-2 :-

$$\text{Order } N \ge \frac{\log\left[\sqrt{\dfrac{10^{0.1\alpha_s}-1}{10^{0.1\alpha_p}-1}}\right]}{\log\left[\dfrac{\Omega_s}{\Omega_p}\right]} = \frac{\log(\lambda/\epsilon)}{\log\left(\frac{\Omega_s}{\Omega_p}\right)}$$

$$\ge \frac{\log\left(\frac{4.898}{1}\right)}{\log\left(\frac{3\pi/4}{\pi/2}\right)}$$

$$N \ge \frac{0.69}{0.176} \ge 3.918$$

Step-3 : Round off N to next higher integer

$$\therefore \boxed{N = 4}$$

Step 4 :-

∴ The transfer function $H(s)$ is,

$$H(s) = \frac{1}{(s^2+0.76537s+1)(s^2+1.8477s+1)}$$

Step-5

Cut off frequency

$$\Omega_c = \frac{\Omega_p}{\sqrt[1/N]{10^{0.1\alpha_s}-1}} = \frac{\Omega_p}{\epsilon^{1/N}}$$

$$= 0.5\pi$$

$$\Omega_c = 1.57$$

$$\therefore H_a(s) = H(s)\Big/_{s=\frac{s}{1.57}}$$

$$\therefore H_a(s) = \frac{(1.57)^4}{(s^2+1.202s+2.465)(s^2+2.962s+2.465)}$$

$$H_a(s) = \frac{A}{s+1.45+j0.6} + \frac{A^*}{s+1.45-j0.6} + \frac{B}{s+0.6+j1.45} + \frac{B^*}{s+0.6-j1.45}$$

$A = 0.7253 + j1.754$ $\quad B = -0.7253 - 0.3j$

$$H_a(s) = \frac{0.7253+j1.754}{s-(-1.45-j0.6)} + \frac{0.7253-j1.754}{s-(-1.45+j0.6)} + \frac{-0.7253-0.3j}{s-(-0.6-j1.45)} + \frac{-0.7253+0.3j}{s-(-0.6+j1.45)}$$

Using impulse invariance method,

$$H(z) = \sum_{k=1}^{N} \frac{c_k}{1 - e^{p_k T} z^{-1}} \quad T = 1\,sec$$

$$\therefore H(z) = \frac{0.7253 + j1.754}{1 - e^{-1.45} e^{-j0.6} z^{-1}} + \frac{0.7253 - j1.754}{1 - e^{-1.45} e^{j0.6} z^{-1}}$$

$$+ \frac{-0.7253 - 0.3j}{1 - e^{-0.6} e^{-j1.45} z^{-1}} + \frac{-0.7253 + 0.3j}{1 - e^{-0.6} e^{j1.45} z^{-1}}$$

$$\boxed{\therefore H(z) = \frac{1.454 + 0.1839 z^{-1}}{1 - 0.387 z^{-1} + 0.055 z^{-2}} + \frac{-1.454 + 0.2307 z^{-1}}{1 - 0.1322 z^{-1} + 0.3012 z^{-2}}}$$

8) Des.

5). Design a Butterworth filter using impulse invariance method for the specification

$$0.8 \le |H(\omega)| \le 1 \qquad 0 \le \omega \le 0.2\pi$$

$$|H(\omega)| \le 0.2 \qquad 0.3\pi \le \omega \le \pi$$

(M/J - 13)

Soln

Given $\frac{1}{\sqrt{1+\epsilon^2}} = 0.8 \Rightarrow \epsilon = 0.75$

$$\frac{1}{\sqrt{1+\lambda^2}} = 0.2 \Rightarrow \lambda = 4.898$$

step-1 $\omega_p = 0.2\pi$

$\omega_s = 0.3\pi$

Using Impulse invariance method,

$\omega = \Omega T$

Assume $T = 1$ sec.

$\therefore \Omega_p = 0.2\pi$

$\Omega_s = 0.3\pi$

step-2: Analog filter Transfer Function

step 2.1: Order N

$$N \ge \frac{\log\left(\frac{\lambda}{\epsilon}\right)}{\log\left(\frac{\Omega_s}{\Omega_p}\right)}$$

$$\ge \frac{\log\left(\frac{4.898}{0.75}\right)}{\log\left(\frac{0.3\pi}{0.2\pi}\right)} \ge \frac{0.815}{0.177} \ge 4.6$$

step 2.2: Round-off N to next higher integer.

$\therefore \boxed{N = 5}$

Step 2.3: Denominator polynomial D(s):

$D(s) = (s+1)(s^2 + 0.618034s + 1)(s^2 + 1.618034s + 1)$

Step 2.4: Find cut-off frequency Ω_c:

$$\Omega_c = \frac{\Omega_p}{\epsilon^{1/N}} = \frac{0.2\pi}{(0.75)^{1/5}}$$

$$\boxed{\Omega_c = 0.662}$$

Transfer function:

$$H(s) = \frac{1}{(s+1)(s^2+0.61803s+1)(s^2+1.61803s+1)}$$

$\because \Omega_c = 1\,rad/sec$

Step 2.5:

$$H_a(s) = H(s)\Big/_{s \to \frac{s}{0.662}}$$

$$\therefore H_a(s) = \frac{1}{\left(\frac{s}{0.662}+1\right)\left[\left(\frac{s}{0.662}\right)^2+0.61803\left(\frac{s}{0.662}\right)+1\right]\left[\left(\frac{s}{0.662}\right)^2+1.61803\left(\frac{s}{0.662}\right)+1\right]}$$

$$= \frac{1}{\left(\frac{s+0.662}{0.662}\right)\left[\frac{s^2+(0.61803)(0.662)s+0.662^2}{0.662^2}\right]\left[\frac{s^2+(1.61803)(0.662s)+0.662^2}{0.662^2}\right]}$$

$$\boxed{H_a(s) = \frac{0.662^5}{(s+0.662)(s^2+0.41s+0.438)(s^2+1.07s+0.438)}}$$

Step 3: Express the analog filter TF in partial fraction form [sum of first-order filters]

$$H_a(s) = \frac{0.662^5}{(s+0.662)(s+0.205-j0.629)(s+0.205+j0.629)(s+0.535-j0.389)(s+0.535+j0.389)}$$

$$H_a(s) = \frac{A}{s+0.662} + \frac{B}{s+0.205-j0.629} + \frac{B^*}{s+0.205+j0.629} + \frac{C}{s+0.535-j0.389} + \frac{C^*}{s+0.535+j0.389}$$

$$A = \left.\frac{0.127}{(s^2+0.41s+0.438)(s^2+1.07s+0.438)}\right|_{s=-0.662}$$

$$A = 1.26$$

$$B = \left.\frac{0.127}{(s+0.662)(s+0.205+j0.629)(s^2+1.07s+0.438)}\right|_{s=-0.205+j0.629}$$

$$\therefore B = -0.09+j0.282$$

$$\therefore B^* = -0.09-j0.282$$

$$C = \left.\frac{0.127}{(s+0.662)(s+0.535+j0.389)(s^2+0.41s+0.438)}\right|_{s=-0.535+j0.389}$$

$$C = -0.5336-j0.7409$$

$$C^* = -0.5336+j0.7409$$

$$\therefore H_a(s) = \frac{1.26}{s+0.662} + \frac{-0.09+j0.282}{s+0.205-j0.629} + \frac{-0.09-j0.282}{s+0.205+j0.629} + \frac{-0.5336-j0.7409}{s+0.535-j0.389} + \frac{-0.5336+j0.7409}{s+0.535+j0.389}$$

Step 4: From Impulse invariance method,

If $H_a(s) = \sum_{k=1}^{N} \frac{c_k}{s-p_k}$ then $H(z) = \sum_{k=1}^{N} \frac{c_k}{1-e^{p_kT}z^{-1}}$

$$H(z) = \frac{1.26}{1-e^{-0.662T}z^{-1}} + \frac{-0.09+j0.282}{1-e^{(-0.205+j0.629)T}z^{-1}}$$

$$+ \frac{-0.09-j0.282}{1-e^{(-0.205-j0.629)T}z^{-1}} + \frac{-0.5336-j0.7409}{1-e^{(-0.535+j0.389)T}z^{-1}}$$

$$+ \frac{-0.5336+j0.7409}{1-e^{(-0.535-j0.389)T}z^{-1}}$$

$$H(z) = \frac{1.26}{1-0.5158z^{-1}} + \frac{-0.09+j0.282}{1-(0.8146)(0.8086+j0.588)z^{-1}}$$

$$+ \frac{-0.09-j0.282}{1-(0.8146)(0.8086-j0.588)z^{-1}} + \frac{-0.5336-j0.7409}{1-(0.5856)(0.925+j0.379)z^{-1}}$$

$$+ \frac{-0.5336+j0.7409}{1-(0.5856)(0.925-j0.379)z^{-1}}$$

$$H(z) = \frac{1.26}{1-0.5158z^{-1}} + \frac{-0.09+j0.282}{1-0.65868z^{-1}-j0.47898z^{-1}}$$

$$+ \frac{-0.09-j0.282}{1-0.65868z^{-1}+j0.47898z^{-1}} + \frac{-0.5336-j0.7409}{1-0.54168z^{-1}-j0.2219z^{-1}}$$

$$+ \frac{-0.5336+j0.7409}{1-0.54168z^{-1}+j0.2219z^{-1}}$$

6. Design a Chebyshev filter for the following specification using impulse invariance method.

$$0.8 \le |H(e^{j\omega})| \le 1 \qquad 0 \le \omega \le 0.2\pi$$

$$|H(e^{j\omega})| \le 0.2 \qquad 0.6\pi \le \omega \le \pi$$

Given,

$$\frac{1}{\sqrt{1+\epsilon^2}} = 0.8 \Rightarrow \epsilon = 0.75$$

$$\frac{1}{\sqrt{1+\lambda^2}} = 0.2 \Rightarrow \lambda = 4.898$$

$\omega_p = 0.2\pi \qquad \omega_s = 0.6\pi$

Step-1 $\quad \omega = \Omega T \qquad$ Let T = 1sec.

$\therefore \Omega_p = \omega_p = 0.2\pi$

$\Omega_s = \omega_s = 0.6\pi$

Step-2 Analog filter Transfer function $H_a(s)$

Step-2.1 Order N:

$$N \ge \frac{\cosh^{-1}\left(\frac{\lambda}{\epsilon}\right)}{\cosh^{-1}\left(\frac{\Omega_s}{\Omega_p}\right)} \ge \frac{\cosh^{-1}\left(\frac{4.898}{0.75}\right)}{\cosh^{-1}\left(\frac{0.6\pi}{0.2\pi}\right)}$$

$$\ge 1.45$$

Step 2.2: Round-off N to the next higher integer

$$\boxed{N = 2}$$

Step 2.3: Minor and Major Axes

$$\mu = \epsilon^{-1} + \sqrt{\epsilon^{-2}+1}$$

$$\mu = 3$$

$$a = \Omega_p \left[\frac{\mu^{1/N} - \mu^{-1/N}}{2} \right]$$

$$= 0.3627$$

$$b = \Omega_p \left[\frac{\mu^{1/N} + \mu^{-1/N}}{2} \right] = 0.7255$$

step 2.4: Denominator polynomial D(s):

$$s_k = a\cos\varphi_k + j\, b\sin\varphi_k$$

$$\varphi_k = \frac{\pi}{2} + \frac{(2k-1)\pi}{2N} \qquad k = 1, 2, \cancel{3}, \cancel{4}$$

$$\varphi_1 = 135^\circ, \quad \varphi_2 = 225^\circ$$

$$s_1 = 0.3627\cos(135^\circ) + j(0.7255)\sin(135^\circ)$$

$$s_1 = -0.2564 + j0.513$$

$$s_2 = -0.2564 - j0.513$$

$$D(s) = (s + 0.2564 - j0.513)(s + 0.2564 + j0.513)$$

step 2.5: Numerator of H(s)

For N even,

$$N_r = \frac{D(s)/s=0}{\sqrt{1+\epsilon^2}} = 0.264$$

∴ Analog filter Transfer Function,

$$\boxed{H_a(s) = \frac{0.264}{(s+0.2564-j0.513)(s+0.2564+j0.513)}}$$

Step-3

From Impulse invariance method,

If $H_a(s) = \sum_{k=1}^{N} \frac{C_k}{s-P_k}$ then $H(z) = \sum_{k=1}^{N} \frac{C_k}{1-e^{P_kT}z^{-1}}$

$\therefore H(z) =$

Step-3 Express Ha(s) as a sum of first order filte

$$H_a(s) = \frac{A}{s+0.256-j0.513} + \frac{B}{s+0.256+j0.513}$$

$$A = \left.\frac{0.264}{s+0.256+j0.513}\right|_{s=-0.256+j0.513}$$

$$A = 0.257j$$

$$B = \left.\frac{0.264}{s+0.256-j0.513}\right|_{s=-0.256-j0.513}$$

$$= -0.257j$$

$$\therefore H_a(s) = \frac{0.257j}{s+0.256-j0.513} - \frac{0.257j}{s+0.256+j0.513}$$

Step-4 From Impulse invariance method, If

$H_a(s) = \sum_{k=1}^{N} \frac{C_k}{s-P_k}$ then $H(z) = \sum_{k=1}^{N} \frac{C_k}{1-e^{P_kT}z^{-1}}$

$$\therefore H(z) = \frac{0.257j}{1-e^{-0.256+j0.513}z^{-1}} - \frac{0.257j}{1-e^{-0.256-j0.513}z^{-1}}$$

$$= \frac{0.257j}{1-(0.774)(0.8713+j0.49)z^{-1}} - \frac{0.257j}{1-(0.774)(0.8713-j0.49)z^{-1}}$$

$$\therefore H(z) = \frac{0.257j}{1-0.6743z^{-1}-j0.379z^{-1}}$$

$$- \frac{0.257j}{1-0.6743z^{-1}+j0.379z^{-1}}$$

$$\therefore \boxed{H(z) = \frac{0.1954z^{-1}}{1-1.3483z^{-1}+0.5987z^{-2}}}$$

IIR Filter Design by the Bilinear Transformation

T1-(712-715)

Impulse invariance technique has a severe limitation that they are appropriate only for lowpass filters and a limited classes of bandpass filters.

* The bilinear transformation is a conformal mapping that transforms the $j\Omega$ axis into the unit circle in the z-plane only once, thus avoiding aliasing of frequency components.
* All points in the LHP of s are mapped inside the unit circle in the z-plane and all points in the RHP of s are mapped into corresponding points outside the unit circle in the z-plane.

Consider an analog linear filter with system function,

$$H(s) = \frac{b}{s+a} \quad \text{——} ①$$

$$\frac{dy(t)}{dt} + a\, y(t) = b x(t) \quad \text{--- } ①$$

Integrate the derivative and approximate the integral by the trapezoidal formula.

$$\therefore \; y(t) = \int_{t_0}^{t} y'(\tau)\, d\tau + y(t_0)$$

where $y'(t)$ denotes the derivative of $y(t)$. The approximation of the integral by the trapezoidal formula at $t = nT$ and $t_0 = nT - T$ yields,

$$y(nT) = \frac{T}{2}\left[y'(nT) + y'(nT-T)\right] + y(nT-T)$$

$$① \Rightarrow \int_{nT-T}^{nT} \frac{dy(t)}{dt}\, dt + a\int_{nT-T}^{nT} y(t)\, dt = b\int_{nT-T}^{nT} x(t)\, dt$$

$$y(nT) - y(nT-T) + \frac{aT}{2}\left[y(nT) + y(nT-T)\right] = \frac{bT}{2}\left[x(nT) + x(nT-T)\right]$$

$$y(nT)\left[1 + \frac{aT}{2}\right] - \left(1 - \frac{aT}{2}\right) y(nT-T) = \frac{bT}{2}\left[x(nT) + x(nT-T)\right] \quad \text{--- } ②$$

Assume $y(nT) \equiv y(n)$

$x(nT) \equiv x(n)$

Taking z transform of ②

$$\left[1 + \frac{aT}{2}\right] Y(z) - \left[1 - \frac{aT}{2}\right] z^{-1} Y(z) = \frac{bT}{2}\left[1 + z^{-1}\right] X(z)$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\frac{bT}{2}(1 + z^{-1})}{1 + \frac{aT}{2} - \left(1 - \frac{aT}{2}\right) z^{-1}}$$

$$H(z) = \frac{b}{\frac{2}{T}\left(\frac{1 - z^{-1}}{1 + z^{-1}}\right) + a}$$

∴ the mapping from the s-plane to the z-plane is,

$$\boxed{s = \frac{2}{T}\left[\frac{1-z^{-1}}{1+z^{-1}}\right]}$$ This is called Bilinear Transformation —— (4)

To Investigate the characteristics of the Bilinear transformation,

Let $z = re^{j\omega}$ $s = \sigma + j\Omega$

$$s = \frac{2}{T}\frac{z-1}{z+1}$$

$$\sigma + j\Omega = \frac{2}{T}\frac{re^{j\omega}-1}{re^{j\omega}+1}$$

$$= \frac{2}{T}\frac{r\cos\omega + jr\sin\omega - 1}{r\cos\omega + jr\sin\omega + 1}$$

$$= \frac{2}{T}\frac{r\cos\omega - 1 + jr\sin\omega}{r\cos\omega + 1 + jr\sin\omega} \times \frac{r\cos\omega + 1 - jr\sin\omega}{r\cos\omega + 1 - jr\sin\omega}$$

$$= \frac{2}{T}\frac{r^2\cos^2\omega - 1 - \cancel{jr^2\sin\omega\cos\omega} + \cancel{jr^2\sin\omega\cos\omega} + jr\sin\omega + jr\sin\omega + r^2\sin^2\omega}{r^2\cos^2\omega + 2r\cos\omega + 1 + r^2\sin^2\omega}$$

$$= \frac{2}{T}\cdot\frac{r^2 - 1 + j2r\sin\omega}{r^2+1+2r\cos\omega}$$

$$\sigma + j\Omega = \frac{2}{T}\frac{r^2-1}{r^2+1+2r\cos\omega} + j\frac{2}{T}\frac{2r\sin\omega}{r^2+1+2r\cos\omega}$$

$$\therefore \sigma = \frac{2}{T}\frac{r^2-1}{1+r^2+2r\cos\omega} \quad \text{—— (5)}$$

$$\Omega = \frac{2}{T}\frac{2r\sin\omega}{1+r^2+2r\cos\omega} \quad \text{—— (6)}$$

If $r<1$ then $\sigma<0$

$r>1$ then $\sigma>0$

$\therefore$ the LHP of s-plane maps inside the unit circle and RHP of s-plane maps outside unit circle.

When $r=1$ then $\sigma=0$ and

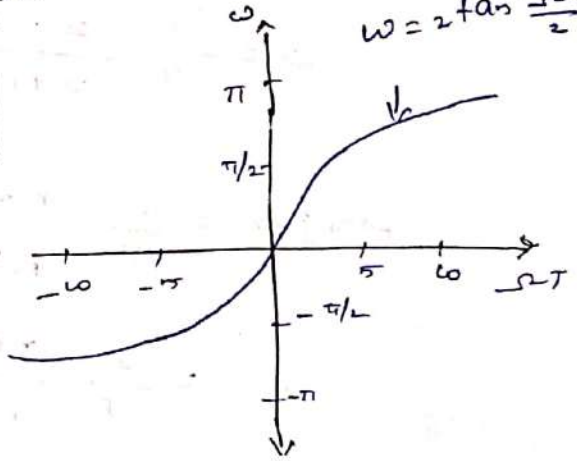
$$\Omega = \frac{2}{T}\frac{\sin\omega}{1+\cos\omega}$$

$$= \frac{2}{T}\frac{\cancel{2}\sin\omega/2\ \cos\omega/2}{\cancel{2}\cos^2\omega/2}$$

$$\boxed{\Omega = \frac{2}{T}\tan\frac{\omega}{2}}$$

(or)

$$\boxed{\omega = 2\tan^{-1}\frac{\Omega T}{2}}$$



Mapping between two frequencies.

The entire range in Ω is mapped only once into the range $-\pi \le \omega \le \pi$. But the mapping is highly nonlinear.

We observe a frequency compression (or) frequency warping as it is usually called, due to the nonlinearity of the arctangent function.

The bilinear transformation maps the point $s=\infty$ into the ~~pole~~ point $z=-1$.

① Apply Bilinear transform to

$$H(s) = \frac{2}{(s+1)(s+3)} \quad \text{with}, \ T = 0.1 \text{ sec}.$$

Soln:

Given $H(s) = \frac{2}{(s+1)(s+3)}$ $\quad T = 0.1 \text{sec}.$

$$H(z) = H(s) \Big/_{s = \frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}}$$

$$= \frac{2}{s^2 + 4s + 3} \Big/_{s = \frac{2}{0.1}\frac{1-z^{-1}}{1+z^{-1}}}$$

$$= \frac{2}{\left[\frac{2^2}{0.1^2}\frac{(1-z^{-1})^2}{(1+z^{-1})^2}\right] + 4\left[\frac{2}{0.1}\frac{1-z^{-1}}{1+z^{-1}}\right] + 3}$$

$$= \frac{2(1+z^{-1})^2}{400(1-z^{-1})^2 + 80(1-z^{-1})(1+z^{-1}) + 3(1+z^{-1})^2}$$

$$= \frac{2(1+z^{-1})^2}{400 + 400z^{-2} - 800z^{-1} + 80 - 80z^{-2} + 3 + 6z^{-1} + 3z^{-2}}$$

$$= \frac{2(1+z^{-1})^2}{483 + 806z^{-1} + 323z^{-2}}$$

$$\boxed{H(z) = \frac{0.004(1+z^{-1})^2}{1 + 1.668z^{-1} + 0.6687z^{-2}}}$$

① Design a digital Butterworth filter satisfying the following specification

$$0.8 \le |H(e^{j\omega})| \le 1 \qquad 0 \le |\omega| \le 0.2\pi$$

$$|H(e^{j\omega})| \le 0.2 \qquad 0.6\pi \le |\omega| \le \pi$$

Assume T = 1 sec. Apply Bilinear transformation.

Soln:

Given $\frac{1}{\sqrt{1+\epsilon^2}} = 0.8 \Rightarrow \epsilon = 0.75$

$$\frac{1}{\sqrt{1+\lambda^2}} = 0.2 \Rightarrow \lambda = 4.898$$

$$\omega_p = 0.2\pi \qquad \omega_s = 0.6\pi$$

Step-1: From Bilinear transformation,

$$\Omega_p = \frac{2}{T}\tan\frac{\omega_p}{2} = \frac{2}{1}\tan\frac{0.2\pi}{2} = 0.6498$$

$$\Omega_s = \frac{2}{T}\tan\frac{\omega_s}{2} = 2.752$$

Step 2 Analog filter Transfer Function $H_a(s)$

Step 2.1 Order N:

$$N \ge \frac{\log\left(\frac{\lambda}{\epsilon}\right)}{\log\left(\frac{\Omega_s}{\Omega_p}\right)}$$

$$\ge \frac{\log\left(\frac{4.898}{0.75}\right)}{\log\left(\frac{2.752}{0.6498}\right)} \ge \frac{0.8149}{0.6268}$$

$$N \ge 1.29$$

Step 2.2 Round-off N to the next higher integer

$$\therefore \boxed{N = 2}$$

Step 2.3 Analog TF $H(s)$ for $\Omega_c = 1$ rad/sec

$$D(s) = s^2 + \sqrt{2}\,s + 1$$

$$\therefore \quad H(s) = \frac{1}{D(s)} = \frac{1}{s^2+\sqrt{2}\,s+1}$$

Step 2.4: Cut-off frequency Ω_c

$$\Omega_c = \frac{\Omega_p}{(\epsilon)^{1/N}} = \frac{0.6498}{(0.75)^{1/2}}$$

$$= 0.7503$$

Step 2.5 Analog T.F. $H_a(s)$

$$H_a(s) = H(s)\Big/_{s \to \frac{s}{0.7503}}$$

$$= \frac{1}{\left(\frac{s}{0.7503}\right)^2 + \sqrt{2}\left(\frac{s}{0.7503}\right) + 1}$$

$$= \frac{(0.7503)^2}{s^2 + \sqrt{2}(0.7503)s + (0.7503)^2}$$

$$H_a(s) = \frac{0.5629}{s^2 + 1.06108\,s + 0.5629}$$

Step 3 Using Bilinear transformation,

$$H(z) = H_a(s)\Big/_{s \to \frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}}$$

$$= \frac{0.5629}{\left(\frac{2}{T}\right)^2\frac{(1-z^{-1})^2}{(1+z^{-1})^2} + 1.06108\left(\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}\right) + 0.5629}$$

$$= \frac{0.5629\,(1+z^{-1})^2}{4\,(1-z^{-1})^2 + 2(1.6108)(1-z^{-1})(1+z^{-1}) + 0.5629\,(1+z^{-1})^2}$$

$$= \frac{0.5629\,(1+z^{-1})^2}{4 - 8z^{-1} + 4z^{-2} + 1.2216 - 1.2216z^{-2} + 0.5629 + 1.1258z^{-1} + \cancel{2/1.28}\ 0.5629z^{-2}}$$

$$= \frac{0.5629\,(1+z^{-1})^2}{5.8745 - \cancel{6.43+1z^{-1}}\ 6.8113z^{-1} + 2.7784}$$

$$\boxed{H(z) = \frac{0.0958\,(1+z^{-1})^2}{1 - 1.07435z^{-1} + 0.47295z^{-2}}}$$

3. Design a Chebyshev filter for the following specification using bilinear transformation

$$0.8 \le |H(e^{j\omega})| \le 1 \qquad 0 \le \omega \le 0.2\pi$$

$$|H(e^{j\omega})| \le 0.2 \qquad 0.6\pi \le \omega \le \pi$$

Given,

$$\frac{1}{\sqrt{1+\epsilon^2}} = 0.8 \Rightarrow \epsilon = 0.75 \qquad \omega_p = 0.2\pi$$

$$\omega_s = 0.6\pi$$

$$\frac{1}{\sqrt{1+\lambda^2}} = 0.2 \Rightarrow \lambda = 4.898$$

Let T = 1 sec

Step-1 Mapping

$$\Omega_p = \frac{2}{T}\tan\frac{\omega_p}{2} = 2\tan\frac{0.2\pi}{2}$$

$$\boxed{\Omega_p = 0.6498}$$

$$\Omega_s = \frac{2}{T} \tan\frac{\omega_s}{2} = 2 \tan\frac{0.6\pi}{2}$$

$$\boxed{\Omega_s = 2.752}$$

Step 2 Analog filter Transfer Function

Step 2.1 Order of the filter:

$$N \geq \frac{\cosh^{-1}\left(\frac{\lambda}{\epsilon}\right)}{\cosh^{-1}\left(\frac{\Omega_s}{\Omega_p}\right)}$$

$$N \geq \frac{\cosh^{-1}\left(\frac{4.898}{0.75}\right)}{\cosh^{-1}\left(\frac{2.752}{0.6498}\right)}$$

step 2.2

$$\geq 1.208$$

$$\boxed{N = 2}$$

Step 2.3 Minor and Major axis of ellipse

$$\mu = \epsilon^{-1} + \sqrt{1+\epsilon^{-2}}$$

$$\mu = 3$$

$$a = \Omega p\left[\frac{\mu^{1/N} - \mu^{-1/N}}{2}\right] = 0.3752$$

$$b = \Omega p\left[\frac{\mu^{1/N} + \mu^{-1/N}}{2}\right] = 0.75$$

Step 2.4 Denominator polynomial D(s)

$$S_k = a\cos\phi_k + jb\sin\phi_k$$

where $\phi_k = \frac{\pi}{2} + \frac{(2k-1)\pi}{2N}$ $\quad k = 1,2$

$$\phi_1 = \frac{\pi}{2} + \frac{\pi}{4} = 135^\circ$$

$$\varphi_2 = \frac{\pi}{2} + \frac{3\pi}{4} = 225^\circ$$

$$S_{k1} = 0.3752 \cos(135^\circ) + j(0.75)\cos(135^\circ)$$

$$s_1 = -0.2653 + j0.53$$

$$s_2 = 0.3752 \cos(225^\circ) + j(0.75)\cos(225^\circ)$$

$$s_2 = -0.2653 - j0.53$$

Step.2.5

$$D(s) = (s + 0.2653 - j0.53)(s + 0.2653 + j0.53)$$

$$D(s) = s^2 + 0.5306s + 0.3516$$

Step 2.6 Numerator of H(s)

$$Nr = \frac{D(s)/_{s=0}}{\sqrt{1+\epsilon^2}} = \frac{0.3516}{\sqrt{1+0.75^2}}$$

$$Nr = 0.28$$

$$\therefore \boxed{H_a(s) = \frac{0.2}{s^2 + 0.5306s + 0.3516}}$$

Step 3 Digital filter Transfer Function:

$$H(z) = H(s)\Big/_{s = \frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}}$$

$$= \frac{0.2}{\left[2\frac{1-z^{-1}}{1+z^{-1}}\right]^2 + 0.5306\left[2\frac{1-z^{-1}}{1+z^{-1}}\right] + 0.3516}$$

$$H(z) = \frac{0.28(1+z^{-1})^2}{5.4128 - 7.298z^{-1} + 3.29z^{-2}}$$

$$\boxed{H(z) = \frac{0.052(1+z^{-1})^2}{1 - 1.348z^{-1} + 0.608z^{-2}}}$$

IIR filter Design by Approximation of Derivatives

One of the simplest methods for converting an analog filter into a digital filter is to approximate the differential equation

$$\sum_{k=0}^{N} \alpha_k \frac{d^k y(t)}{dt^k} = \sum_{k=0}^{M} \beta_k \frac{d^k x(t)}{dt^k} \quad \text{—(1)}$$

$$\frac{Y(s)}{X(s)} = H_a(s) = \frac{\sum_{k=0}^{M} \beta_k s^k}{\sum_{k=0}^{N} \alpha_k s^k}$$

This approach is often used to solve a linear constant - coefficient differential equation numerically on a digital computer.

For the derivative $\frac{dy(t)}{dt}$ at time $t = nT$, substitute the backward difference $\frac{y(nT) - y(nT-T)}{T}$.

$$\therefore \left.\frac{dy(t)}{dt}\right|_{t=nT} = \frac{y(nT) - y(nT-T)}{T}$$

$$= \frac{y(n) - y(n-1)}{T}$$

$T \rightarrow$ sampling interval and

$y(n) \equiv y(nT)$.

The analog differentiator with output $\frac{dy(t)}{dt}$ has the system function $H(s) = s$ while the digital system that produces the output $\frac{[y(n) - y(n-1)]}{T}$ has the system function

$$H(z) = \frac{(1 - z^{-1})}{T}.$$

$$\therefore \quad s = \frac{1 - z^{-1}}{T} \quad \text{---} (2)$$

The second derivative $\frac{d^2 y(t)}{dt^2}$ is replaced by the second difference, which is derived as follows:

$$\left.\frac{d^2 y(t)}{dt^2}\right|_{t = nT} = \frac{d}{dt}\left[\frac{dy(t)}{dt}\right]_{t = nT}$$

$$= \frac{\left[\frac{y(nT) - y(nT - T)}{T}\right] - \left[\frac{y(nT - T) - y(nT - 2T)}{T}\right]}{T}$$

$$= \frac{y(n) - 2y(n-1) + y(n-2)}{T^2}$$

In frequency domain,

$$s^2 = \frac{1 - 2z^{-1} + z^{-2}}{T^2}$$

$$= \left(\frac{1 - z^{-1}}{T}\right)^2 \quad \text{---} (3)$$

For k^{th} derivative,

$$s^k = \left(\frac{1 - z^{-1}}{T}\right)^k \quad \text{---} (4)$$

∴ The system function for the digital IIR filter obtained as a result of the approximation of the derivatives by finite difference is

$$H(z) = H_a(s) \Big|_{s = \frac{1-z^{-1}}{T}}$$

where $H_a(s)$ → Analog filter system function.

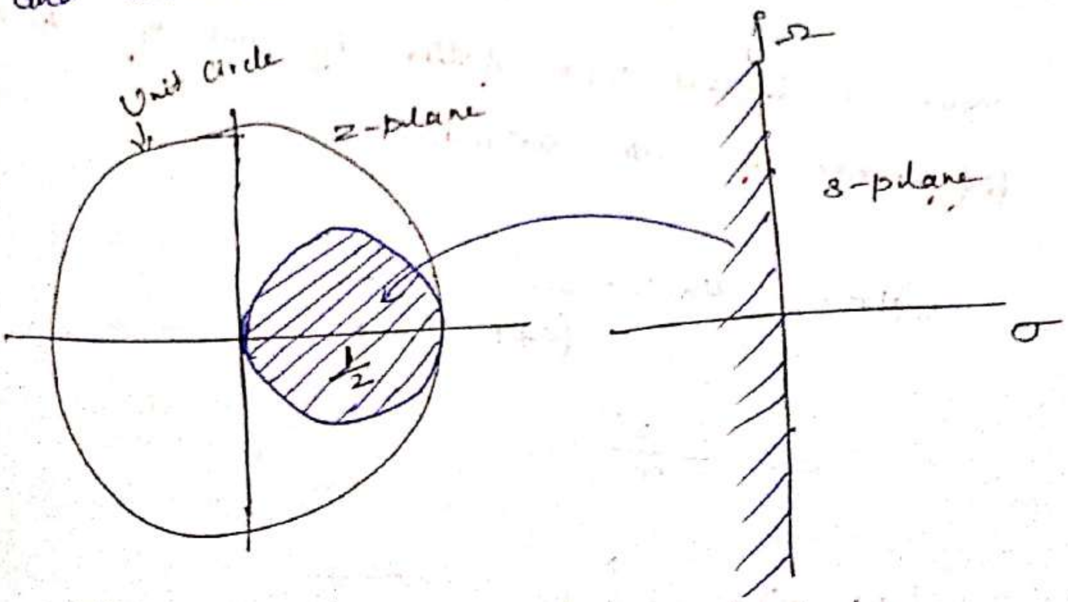
<u>Mapping</u>:

$$z = \frac{1}{1-sT} \quad \text{—— (5)}$$

$$s = j\Omega, \quad z = \frac{1}{1-j\Omega T}$$

$$= \frac{1}{1+\Omega^2 T^2} + j\frac{\Omega T}{1+\Omega^2 T^2}$$

As Ω varies from $-\infty$ to ∞, the corresponding locus of points in the z-plane is a circle of radius $\frac{1}{2}$ and with center at $z = \frac{1}{2}$.



The mapping $s = \frac{1-z^{-1}}{T}$ takes LHP in the s-plane into points inside the circle of radius $\frac{1}{2}$ and center $z = \frac{1}{2}$ in the z-plane.

The mapping takes points in the LHP of the s-plane into corresponding points inside the circle in the z-plane, and points in the RHP of the s-plane are mapped into points outside the circle.

* The mapping has a desirable property that a stable analog filter is transformed into a stable digital filter, the digital filter and co are confined to relatively small frequencies and as a consequence, this mapping is restricted to the design of LPF and BPF having relatively small resonant frequencies.

* It is not possible for example to transform a highpass analog filter into a corresponding highpass digital filter.

① Convert the analog bandpass filter with system function

$$H_a(s) = \frac{1}{(s+0.1)^2+9}$$

into a digital IIR filter by use of the backward difference for the derivatives.

Given, $H_a(s) = \dfrac{1}{(s+0.1)^2+9}$

$$s = \frac{1-z^{-1}}{T}$$

$$H(z) = \frac{1}{\left[\frac{1-z^{-1}}{T}+0.1\right]^2+9}$$

$$= \frac{T^2}{(1-z^{-1})^2+2(1-z^{-1})0.1+0.1^2+9}$$

$$= \frac{T^2}{1 - 2z^{-1} + z^{-2} + 0.2 - 0.2z^{-1} + 0.01 + 9}$$

$$= \frac{T^2 / (1 + 0.2T + 9.01T^2)}{1 - \frac{2(1+0.1T)}{1+0.2T+9.01T^2} z^{-1} + \frac{1}{1+0.2T+9.01T^2} z^{-2}}$$

It has the form of resonator provided that T is selected small enough ($T \leq 0.1$) in order for the poles to be near the unit circle. The condition $a_1^2 < 4a_2$ is satisfied, so that the poles are complex valued.

Structures of IIR systems T1 (582 - 601)

Consider

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

$$H(z) = \frac{\sum_{k=0}^{M} b_k z^{-k}}{1 + \sum_{k=1}^{N} a_k z^{-k}} \quad \text{——} ②$$

Different IIR filter realizations are given based on eqn ① and ②.

Types of structures:-

1. Direct - form I & II
2. Cascade form
3. Parallel form
4. Lattice
5. Lattice - Ladder

Direct form Structures:

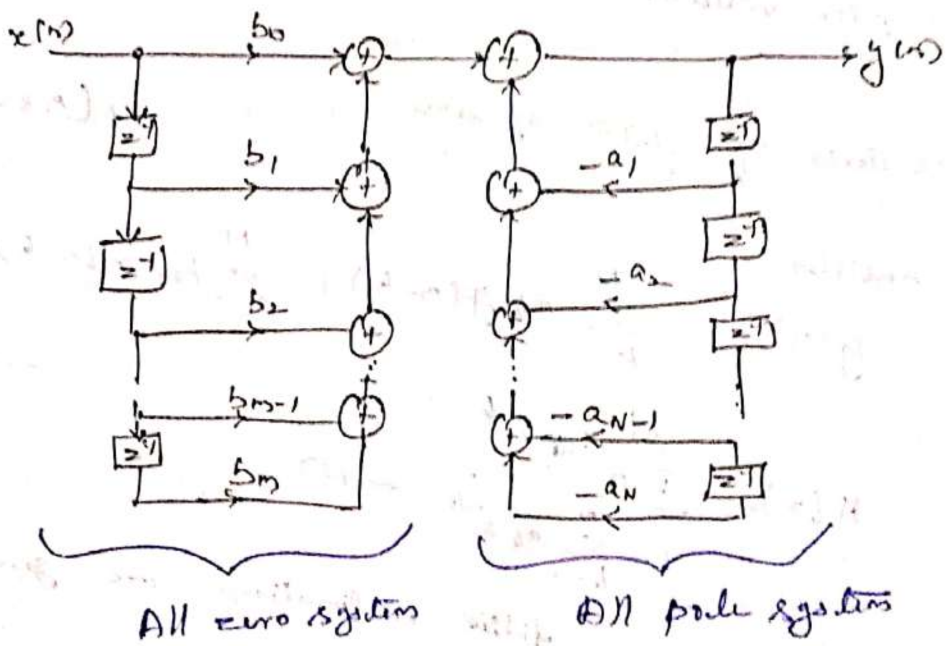
Let $H(z) = H_1(z)\,H_2(z)$

where $H_1(z)$ consists of the zeros of $H(z)$ and $H_2(z)$ consists of the poles of $H(z)$

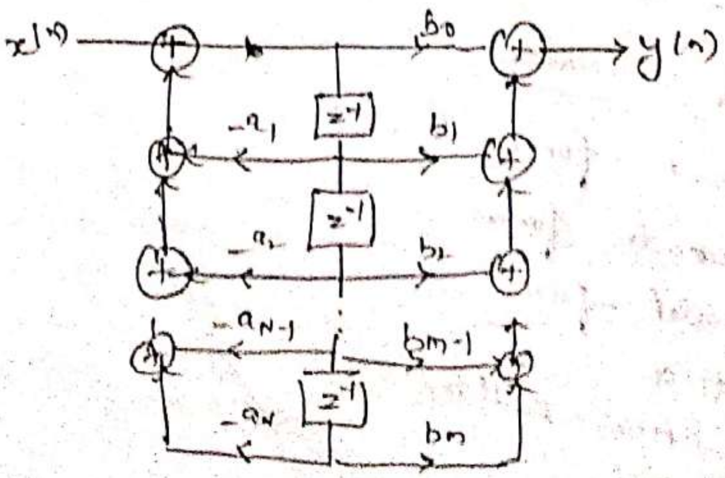
$$H_1(z) = \sum_{k=0}^{M} b_k z^{-k} \quad \text{and}$$

$$H_2(z) = \frac{1}{1+\sum_{k=1}^{N} a_k z^{-k}}$$

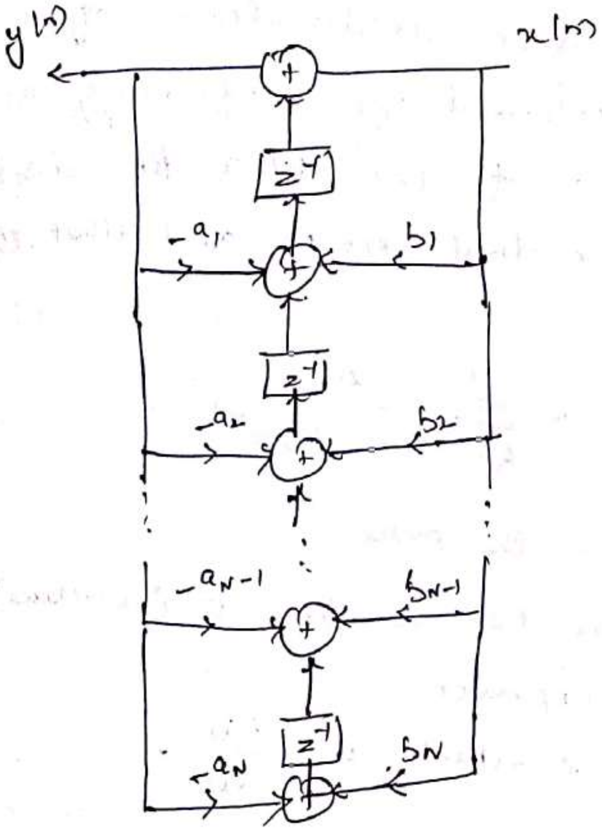
Direct form-I



Direct form-II



Transposed Direct form II structure



* Reverse the directions of all branch
* Exchange the input and output
* Exchange the summing points with branch points.

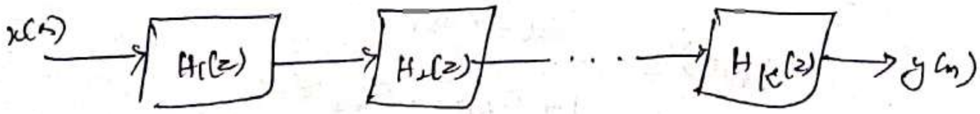
Cascade form structures:

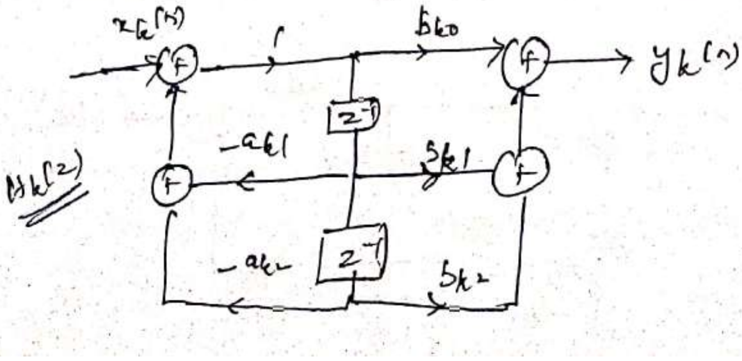
The system function can be factored into a cascade of second-order subsystems.

$$\text{ie}\quad H(z) = \prod_{k=1}^{K} H_k(z)$$

where $k \rightarrow$ Integer part of $\frac{N+1}{2}$.

$$H_k(z) = \frac{b_{k0} + b_{k1} z^{-1} + b_{k2} z^{-2}}{1 + a_{k1} z^{-1} + a_{k2} z^{-2}}$$





Parallel form structure :-

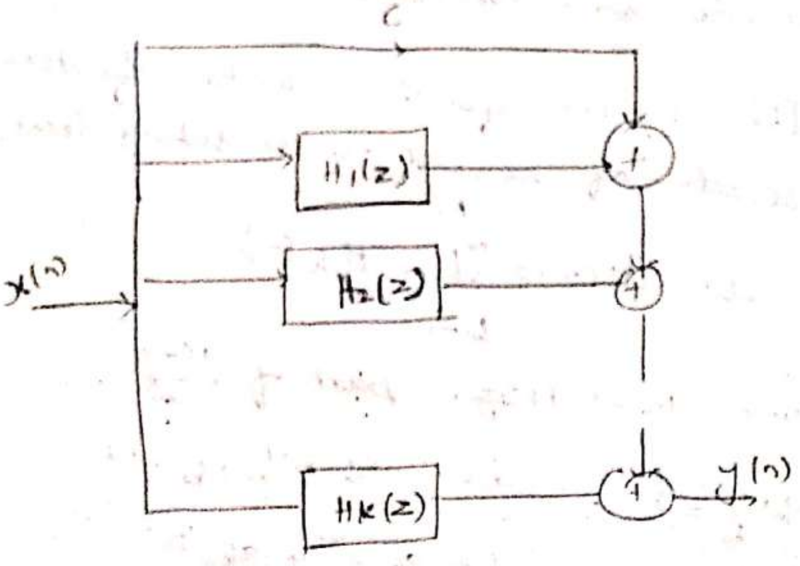
A parallel form realization of an IIR system can be obtained by performing a partial fraction expansion of H(z). With out loss of generality, assume that $N \geq M$ and that the poles are distinct.

Then,
$$H(z) = C + \sum_{k=1}^{N} \frac{A_k}{1 - p_k z^{-1}}$$

where $\{p_k\}$ are the poles.

$\{A_k\}$ are the co-efficients (residues) in partial fraction expansion,

$C \rightarrow$ constant $c = \frac{b_N}{a_N}$.



Subsystems has the form,

$$H_k(z) = \frac{b_{k0} + b_{k1} z^{-1}}{1 + a_{k1} z^{-1} + a_{k2} z^{-2}}$$

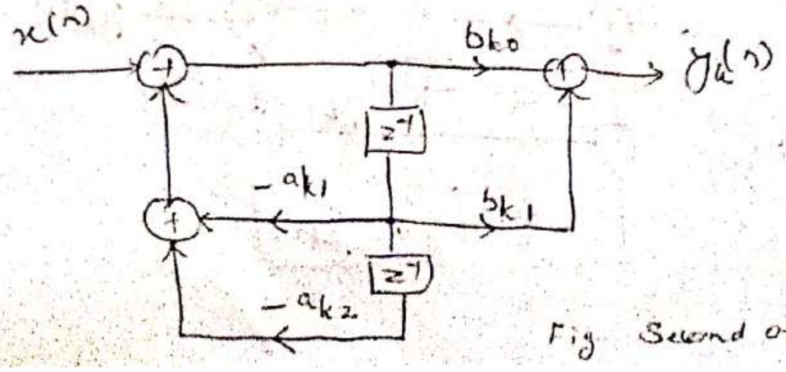
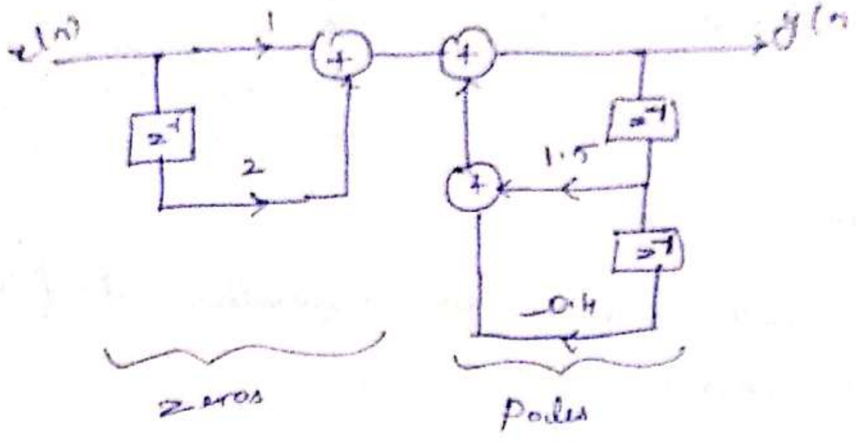


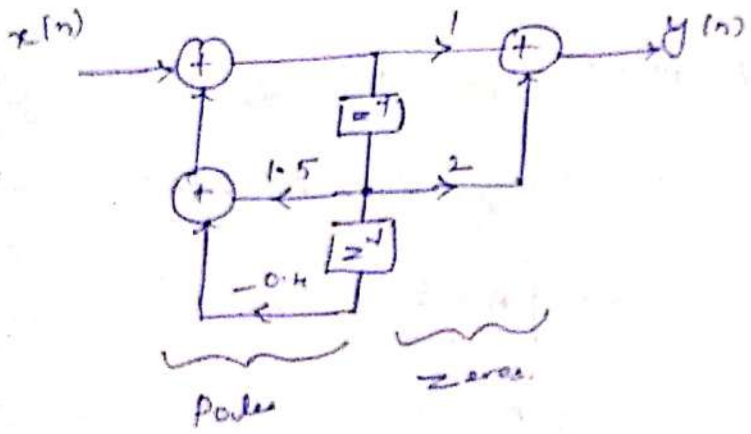
Fig Second order subsystem

1) Obtain Direct form I and Direct II realization of $H(z) = \dfrac{1+2z^{-1}}{1-1.5z^{-1}+0.4z^{-2}}$ (M/J-12)

Direct form I realization:-



Direct form II realization:-



2). Obtain the cascade and parallel realizations for the system function given by

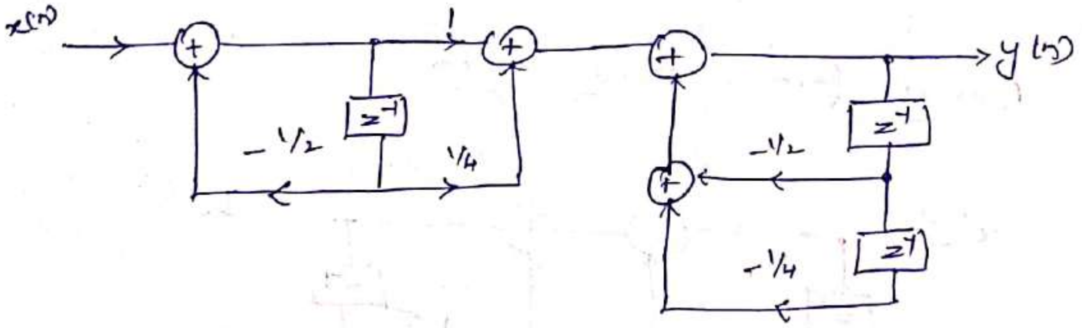
$$H(z) = \frac{1+\frac{1}{4}z^{-1}}{\left(1+\frac{1}{2}z^{-1}\right)\left(1+\frac{1}{2}z^{-1}+\frac{1}{4}z^{-2}\right)}$$ (N/D-14)

Cascade Realization:-

To obtain the cascade realization, the transfer function is broken into a product of two functions as,

$$H(z) = H_1(z) \cdot H_2(z)$$

where $H_1(z) = \dfrac{1+\frac{1}{4}z^{-1}}{1+\frac{1}{2}z^{-1}}$ $\quad H_2(z) = \dfrac{1}{1+\frac{1}{2}z^{-1}+\frac{1}{4}z^{-2}}$



Parallel Realization :-

To obtain parallel form realisation, perform partial fraction expansion of $H(z)$,

$$H(z) = \frac{1+\frac{1}{4}z^{-1}}{\left(1+\frac{1}{2}z^{-1}\right)\left(1+\frac{1}{2}z^{-1}+\frac{1}{4}z^{-2}\right)} = \frac{A}{1+\frac{1}{2}z^{-1}} + \frac{Bz^{-1}+C}{1+\frac{1}{2}z^{-1}+\frac{1}{4}z^{-2}}$$

$$1+\frac{1}{4}z^{-1} = A\left(1+\frac{1}{2}z^{-1}+\frac{1}{4}z^{-2}\right) + Bz^{-1}\left(1+\frac{1}{2}z^{-1}\right) + C\left(1+\frac{1}{2}z^{-1}\right)$$

Let $z^{-1} = -2 \Rightarrow 1+\frac{1}{\cancel{4}_2}(-\cancel{2}) = A\left(\frac{1}{\cancel{4}}(\cancel{4})\right)$

$$\Rightarrow \boxed{A = \frac{1}{2}}$$

Put $z^{-1} = 0 \Rightarrow 1 = A + C$

$$C = 1 - \frac{1}{2} = \frac{1}{2}$$

$$\boxed{C = \frac{1}{2}}$$

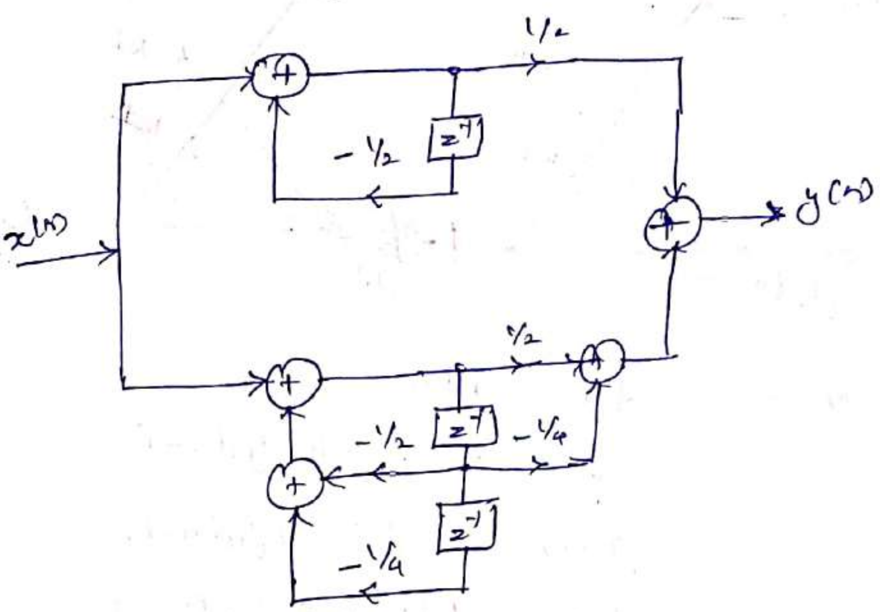
Put $z^{-1} = 1 \Rightarrow 1+\frac{1}{4} = A\left[1+\frac{1}{2}+\frac{1}{4}\right] + B\left[1+\frac{1}{2}\right] + C\left[1+\frac{1}{2}\right]$

$$\frac{5}{4} = \frac{1}{2}\left[\frac{4+2+1}{4}\right] + B\left(\frac{3}{2}\right) + \frac{1}{2}\left(\frac{3}{2}\right)$$

$$\Rightarrow \boxed{B = -\frac{1}{4}}$$

$$\therefore H(z) = \frac{1/2}{1+\frac{1}{2}z^{-1}} + \frac{\frac{1}{2}-\frac{1}{4}z^{-1}}{1+\frac{1}{2}z^{-1}+\frac{1}{4}z^{-2}}$$

The parallel realisation structure is,



Lattice and Lattice-Ladder structures for IIR systems

Consider an all-pole system with system function,

$$H(z) = \frac{1}{1+\sum_{k=1}^{N} a_N(k) z^{-k}}$$

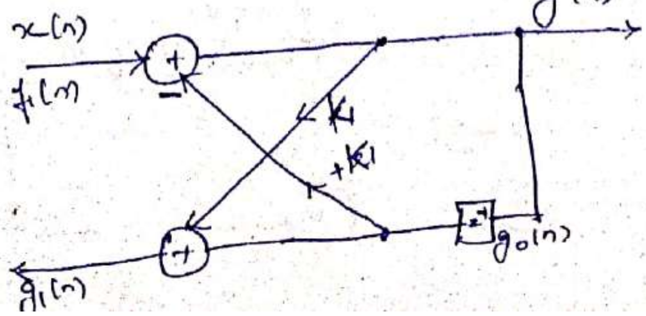
The difference equation for this IIR system is,

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

(or)

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

For first order IIR filter, $x(n) = y(n) + a_1 y(n-1)$



$x(n) = f_1(n)$

$f_0(n) = f_1(n) - K_1 g_0(n-1)$

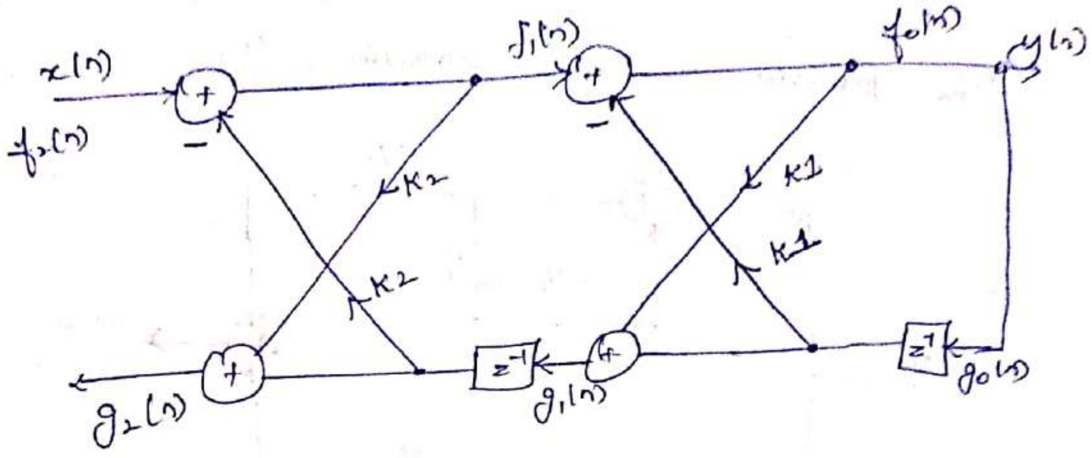
$g_1(n) = K_1 f_0(n) + g_0(n-1)$

$y(n) = f_0(n)$

$y(n) = x(n) - K_1 y(n-1)$

$g_1(n) = K_1 y(n) + y(n-1)$

Two pole Lattice structure

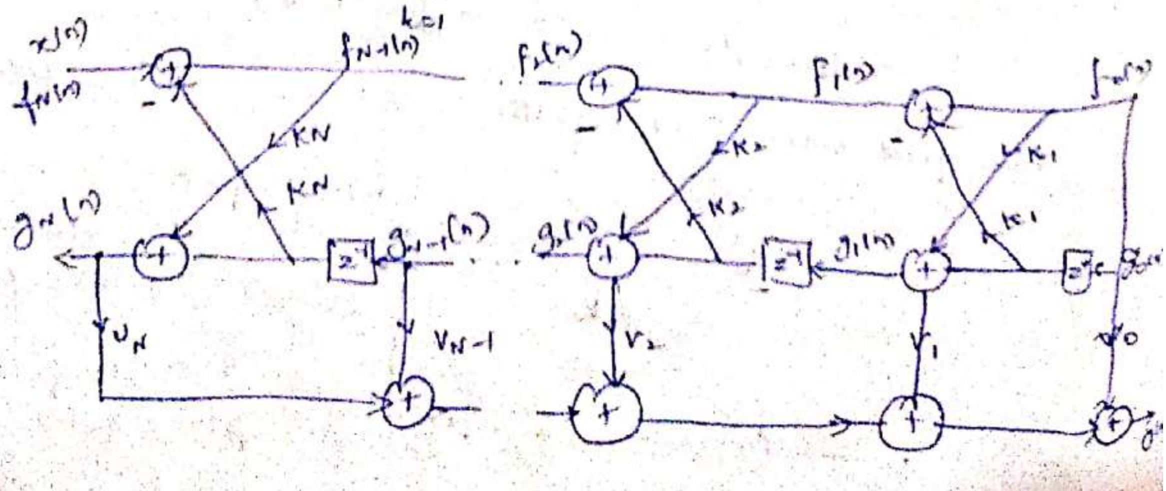


for $N=2$,
$$f_2(n) = x(n)$$
$$f_1(n) = f_2(n) - K_2 g_1(n-1)$$
$$g_2(n) = K_2 f_1(n) + g_1(n-1)$$
$$f_0(n) = f_1(n) - K_1 g_0(n-1)$$
$$g_1(n) = K_1 f_0(n) + g_0(n-1)$$
$$y(n) = f_0(n) = g_0(n)$$

After simplifications,
$$y(n) = -K_1(1+K_2)y(n-2) - K_2 y(n-2) + x(n)$$
$$g_2(n) = K_2 y(n) + K_1(1+K_2)y(n-1) + y(n-2)$$

Lattice - Ladder structure

$$H(z) = \frac{\sum_{k=0}^{M} c_M(k) z^{-k}}{1+\sum_{k=1}^{N} a_N(k) z^{-k}} \qquad y(n) = \sum_{m=0}^{M} v_m g_m(n)$$



where $\{v_m\}$ are the parameters that determine the zeros of the system.

$$H(z) = \frac{Y(z)}{X(z)}$$

$$= \sum_{m=0}^{M} v_m \frac{G_m(z)}{X(z)}$$

since $X(z) = F_N(z)$ and $F_0(z) = G_0(z)$,

$$H(z) = \sum_{m=0}^{M} v_m \frac{G_m(z)}{G_0(z)} \frac{F_0(z)}{F_N(z)}$$

$$= \sum_{m=0}^{M} v_m \frac{B_m(z)}{A_N(z)}$$

$$H(z) = \frac{\sum_{m=0}^{M} v_m B_m(z)}{A_N(z)}$$

$$\therefore C_m(z) = \sum_{m=0}^{m} v_m B_m(z)$$

3). Convert the following pole zero IIR filter into a lattice - ladder structure.

$$H(z) = \frac{1+2z^{-1}+2z^{-2}+z^{-3}}{1+\frac{13}{24}z^{-1}+\frac{5}{8}z^{-2}+\frac{1}{3}z^{-3}}$$

Given,

$$B_m(z) = 1+2z^{-1}+2z^{-2}+z^{-3}$$

$$A_N(z) = 1+\frac{13}{24}z^{-1}+\frac{5}{8}z^{-2}+\frac{1}{3}z^{-3}$$

$a_3(0) = 1 \quad a_3(1) = \frac{13}{24} \quad a_3(2) = \frac{5}{8} \quad a_3(3) = \frac{1}{3}$

$$k_3 = a_3(3) = \frac{1}{3}$$

$$a_{m-1}(k) = \frac{a_m(k) - a_m(m)\, a_m(m-k)}{1 - a_m^2(m)}$$

For $m=3$, $k=1$

$$a_2(1) = \frac{a_3(1) - a_3(3)\, a_3(2)}{1 - a_3^2(3)}$$

$$= \frac{\frac{13}{24} - \frac{1}{3}\left(\frac{5}{8}\right)}{1-\left(\frac{2}{3}\right)^2} = \frac{3}{8}$$

For $m=3$ $k=2$

$$k_2 = a_2(2) = \frac{a_3(2) - a_3(3)\, a_3(1)}{1 - a_3^2(3)}$$

$$= \frac{\frac{5}{8} - \frac{1}{3}\left(\frac{13}{24}\right)}{1-\left(\frac{1}{3}\right)^2} = \frac{1}{2}$$

For $m=2$, $k=1$

$$k_1 = a_1(1) = \frac{a_2(1) - a_2(2)\, a_2(1)}{1 - a_2^2(2)}$$

$$= \frac{3/8 - \frac{1}{2}\left(\frac{3}{8}\right)}{1-\left(\frac{1}{2}\right)^2} = \frac{1}{4}$$

$\therefore$ For Lattice structure,

$$k_1 = \frac{1}{4} \qquad k_2 = \frac{1}{2} \qquad k_3 = \frac{1}{3}$$

For Ladder structure

$$C_m = b_m - \sum_{i=m+1}^{M} c_i\, a_i(i-m) \qquad m = M, M-1, \dots 0$$

$$C_3 = b_3 = 1$$

$$c_2 = b_2 - c_3\, a_3(1)$$
$$= 2 - 1\left(\frac{13}{24}\right) = 1.4583$$

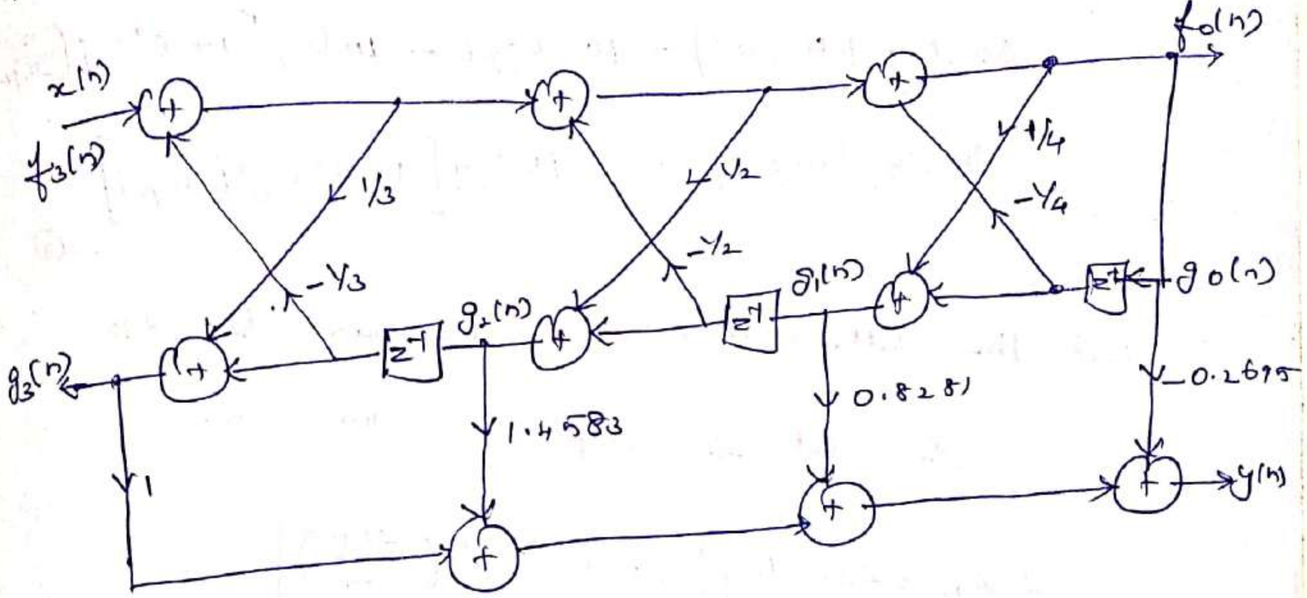
$$c_1 = b_1 - \sum_{i=1}^{3} c_i\, a_i(i-m)$$
$$= b_1 - \left[c_2\, a_2(1) + c_3\, a_3(2)\right] = 0.8281$$

$$c_0 = b_0 - \sum_{i=1}^{3} c_i\, a_i(i-m)$$

$$= b_0 - \left[c_1 a_1(1) + c_2 a_2(2) + c_3 a_3(3)\right]$$

$$= -0.2695 .$$

$\therefore$ The Lattice Ladder structure is,



Chebyshev Filter Design

Order N:

$$|H(j\Omega)|^2 = \frac{1}{1+\epsilon^2 C_N^2\left(\frac{\Omega}{\Omega_p}\right)} \qquad N = 1, 2, \ldots \quad \text{—(1)}$$

Taking 10 log on both sides,

$$20 \log |H(j\Omega)| = 10 \log 1 - 10 \log\left[1+\epsilon^2 C_N^2\left(\frac{\Omega}{\Omega_p}\right)\right]$$

$$20 \log |H(j\Omega)| = -10 \log\left[1+\epsilon^2 C_N^2\left(\frac{\Omega}{\Omega_p}\right)\right] \quad \text{—(2)}$$

Let the attenuation in passband be α_p.

i.e. at $\Omega = \Omega_p \Rightarrow$ gain $= \alpha_p$

$$\therefore \cancel{-}\alpha_p = \cancel{-}10 \log\left[1+\epsilon^2 C_N^2\left(\frac{\Omega_p}{\Omega_p}\right)\right]$$

$$\alpha_p = 10 \log(1+\epsilon^2) \quad / \because C_N(1) = 1$$

$$\Rightarrow \epsilon^2 = 10^{0.1\alpha_p} - 1$$

$$\boxed{\epsilon = \sqrt{10^{0.1\alpha_p} - 1}} \quad \text{—(3)}$$

Let the attenuation in stop band be α_s.

$$\cancel{-}\alpha_s = \cancel{-}10 \log\left[1+\epsilon^2 C_N^2\left(\frac{\Omega_s}{\Omega_p}\right)\right]$$

$$\alpha_s = 10 \log\left[1+\epsilon^2\left\{\cosh\left(N \cosh^{-1}\left(\frac{\Omega_s}{\Omega_p}\right)\right)\right\}^2\right] \quad \text{—(4)}$$

$$/ \because \frac{\Omega_s}{\Omega_p} > 1$$

Substitute (3) in (4) and solve for N.

$$\therefore \alpha_s = 10 \log\left[1+\left(10^{0.1\alpha_p}-1\right)\left\{\cosh\left(N \cosh^{-1}\left(\frac{\Omega_s}{\Omega_p}\right)\right)\right\}^2\right]$$

$$10^{0.1\alpha_s} - 1 = \left(10^{0.1\alpha_p}-1\right)\left\{\cosh\left(N \cosh^{-1}\left(\frac{\Omega_s}{\Omega_p}\right)\right)\right\}^2$$

$$\frac{10^{0.1\alpha_s}-1}{10^{0.1\alpha_p}-1} = \left\{\cosh\left[N\cosh^{-1}\left(\frac{\Omega_s}{\Omega_p}\right)\right]\right\}^2$$

$$\therefore N\cosh^{-1}\left(\frac{\Omega_s}{\Omega_p}\right) = \cosh^{-1}\left[\frac{10^{0.1\alpha_s}-1}{10^{0.1\alpha_p}-1}\right]$$

$$\boxed{N \geq \frac{\cosh^{-1}\left[\frac{10^{0.1\alpha_s}-1}{10^{0.1\alpha_p}-1}\right]}{\cosh^{-1}\left(\frac{\Omega_s}{\Omega_p}\right)}}$$

(or)

$$\boxed{N \geq \frac{\cosh^{-1}\left(\frac{\lambda}{\epsilon}\right)}{\cosh^{-1}\left(\frac{\Omega_s}{\Omega_p}\right)}}$$

(or)

$$\boxed{N \geq \frac{\cosh^{-1}(A)}{\cosh^{-1}(1/k)}}$$

Pole Locations for Chebyshev Filters

To obtain the poles of chebyshev filters,

$$1+\epsilon^2 C_N^2\left(\frac{-js}{\Omega_p}\right)=0 \qquad \Omega=\frac{s}{j}$$

$$C_N^2\left(-\frac{js}{\Omega_p}\right) = -\frac{1}{\epsilon^2}$$

$$C_N\left(\frac{-js}{\Omega_p}\right) = \pm\frac{j}{\epsilon}$$

$$\cos\left(N\cos^{-1}\left(\frac{js}{\Omega_p}\right)\right) = \pm\frac{j}{\epsilon} \quad \text{———} \ (1)$$

Let $\cos^{-1}\left(-\frac{js}{\Omega_p}\right) = \varphi - j\alpha$ ——— (2)

$$\pm \frac{j}{\epsilon} = \cos[N\varphi - jN\alpha]$$

$$= \cos N\varphi \cos jN\alpha + \sin(N\varphi)\sin(jN\alpha)$$

$$= \cos(N\varphi)\cosh(N\alpha) + j\sin N\varphi \sinh N\alpha$$

Equating real and imaginary parts,

$$\cos[N\varphi]\cosh(N\alpha) = 0 \quad \text{——} (3)$$

$$\sin(N\varphi)\sinh(N\alpha) = \pm\frac{1}{\epsilon} \quad \text{——} (4)$$

$\sin\alpha \cosh(N\alpha) > 0$ for α real,

(1) $\Rightarrow$ $\therefore \varphi = \frac{2k-1}{2N}\pi$, $k = 1, 2, \ldots N$ —— (5)

sub (5) in eqn (2), solve for α.

where $\sin N\varphi = \pm 1$

$$\alpha = \pm \frac{1}{N}\sinh^{-1}\left(\frac{1}{\epsilon}\right) \quad \text{——} (6)$$

(2) $\Rightarrow$ $s_k = j\Omega_p \cos(\varphi - j\alpha)$

$$= j\Omega_p[\cos\varphi\cosh\alpha + j\sin\varphi\sinh\alpha]$$

$$S_k = \Omega_p[-\sin\varphi\sinh\alpha + j\cos\varphi\cosh\alpha] \quad \text{——} (7)$$

$$\sinh^{-1}(\epsilon^{-1}) = \ln\left(\epsilon^{-1} + \sqrt{1+\epsilon^{-2}}\right)$$

(or)

$$\mu = e^{\sinh^{-1}(\epsilon^{-1})}$$

$$\boxed{\mu = \epsilon^{-1} + \sqrt{1+\epsilon^{-2}}}$$

$$\sinh \theta = \sinh\left[\frac{1}{N}\sinh^{-1}(\epsilon^{-1})\right]$$

$$= \frac{e^{\frac{1}{N}\sinh^{-1}(\epsilon^{-1})} - e^{-(1/N)\sinh^{-1}(\epsilon^{-1})}}{2}$$

$$= \frac{\left[e^{\sinh^{-1}(\epsilon^{-1})}\right]^{1/N} - \left[e^{-\sinh^{-1}(\epsilon^{-1})}\right]^{1/N}}{2}$$

$$\sinh\theta = \frac{\mu^{1/N} - \mu^{-1/N}}{2}$$

Simillarly, $\cosh\theta = \dfrac{\mu^{1/N} + \mu^{-1/N}}{2}$

$$\therefore\ s_k = \Omega_p\left[-\sin\varphi\left(\frac{\mu^{1/N} - \mu^{-1/N}}{2}\right) + j\cos\varphi\left(\frac{\mu^{1/N} + \mu^{-1/N}}{2}\right)\right]$$

$$s_k = -a\sin\varphi + jb\cos\varphi$$

$$= -a\sin\frac{(2k-1)\pi}{2N} + jb\cos\frac{(2k-1)\pi}{2N}$$

$$= a\cos\left[\frac{\pi}{2} + \frac{(2k-1)\pi}{2N}\right] + jb\sin\left[\frac{\pi}{2} + \frac{(2k-1)\pi}{2N}\right]$$

$$\therefore\ \boxed{s_k = a\cos\varphi_k + jb\sin\varphi_k}$$

where,

$$a = \Omega_p\left[\frac{\mu^{1/N} - \mu^{-1/N}}{2}\right]$$

$$b = \Omega_p\left[\frac{\mu^{1/N} + \mu^{-1/N}}{2}\right]$$

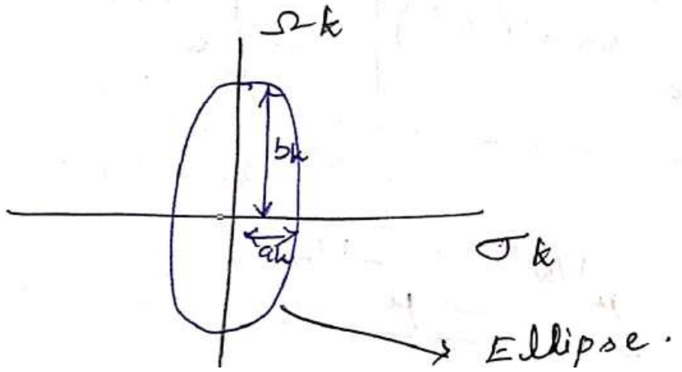
$$\varphi_k = \frac{\pi}{2} + \frac{(2k-1)\pi}{2N} \qquad k = 1, 2, \ldots N$$

Poles of Chebyshev transfer function are located on an ellipse in the s-plane.

The equation of the ellipse is given by,

$$\frac{\sigma_k^2}{a^2} + \frac{\Omega_k^2}{b^2} = 1$$

where a and b are minor and major axis of the ellipse.



UNIT - III

FIR FILTER DESIGN

Structures of FIR - Linear phase FIR filter - Fourier series - Filter design using windowing techniques (Rectangular window, Hamming window, Hanning window), Frequency sampling techniques - Finite word length effects in digital filters: Errors, Limit Cycle, Noise power spectrum.

INTRODUCTION:

T1 (660-664)

FIR filters are non-recursive type. The present output depends on present and previous values of input.

$$y(n) = b_0 x(n) + b_1 x(n-1) + \cdots b_{M-1} x(n-M+1) \quad \text{---(1)}$$

$$y(n) = \sum_{k=0}^{M-1} b_k x(n-k) \quad \text{---(2)}$$

$$\therefore Y(z) = \sum_{k=0}^{M-1} b_k z^{-k} X(z)$$

$$\boxed{H(z) = \frac{Y(z)}{X(z)} = \sum_{k=0}^{M-1} b_k z^{-k}}$$

Linear phase FIR filters:

An FIR filter of length M is given by eqn (1) and (2).

where $\{b_k\}$ set of filter co-efficients.

Alternatively, the output sequence is the convolution of the unit sample response $h(n)$ of the system with the input signal.

$$\therefore \quad y(n) = \sum_{k=0}^{M-1} h(k)\, x(n-k) \quad \text{--- (3)}$$

Eqn (2) and (3) are identical.

The filter can also be characterized by its system fn

$$H(z) = \sum_{k=0}^{M-1} h(k)\, z^{-k} \quad \text{--- (4)}$$

which is viewed as a polynomial of degree M-1 in the variable z^{-1}. The roots of the polynomial constitute the zeros of the filter.

An FIR filter has linear phase if its unit sample response satisfies the condition

$$\boxed{h(n) = \pm h(M-1-n),} \quad n = 0, 1, \ldots M-1 \quad \text{--- (5)}$$

When the symmetry and antisymmetry conditions in eqn (5) are incorporated into eqn (4), we have.

$$H(z) = h(0) + h(1)z^{-1} + h(2)z^{-2} + \ldots + h(M-1)z^{-(m-1)}$$

$$= z^{-(M-1)/2} \left\{ h\left(\frac{M-1}{2}\right) + \sum_{n=0}^{(M-3)/2} h(n) \left[z^{(M-1-2k)/2} \pm z^{-(M-1-2k)/2} \right] \right\} \quad m\text{-odd.}$$

$$= z^{-(M-1)/2} \sum_{n=0}^{\frac{M}{2}-1} h(n) \left[z^{(M-1-2k)/2} \pm z^{-(M-1-2k)/2} \right]. \quad M\text{-even}$$

$$\text{--- (6)}$$

Substitute z^{-1} for z in eqn (4) and multiply both sides of the resulting eqn by $z^{-(M-1)}$,

$$z^{-(M-1)} H(z^{-1}) = \pm H(z) \quad \text{--- (7)}$$

This result implies that the roots of the polynomial $H(z^{-1})$. Consequently, the roots of $H(z)$

must occur in reciprocal pairs. ie, If z_1 is a root (or) a zero of $H(z)$, then $1/z_1$ is also a root. If the unit sample response $h(n)$ of the filter is real, complex valued roots must occur in complex - conjugate pairs.

Hence, If z_1 is a complex - valued root, z_1^* is also a root.

As a consequence of (7), $H(z)$ also has a zero at $\frac{1}{z_1^*}$.

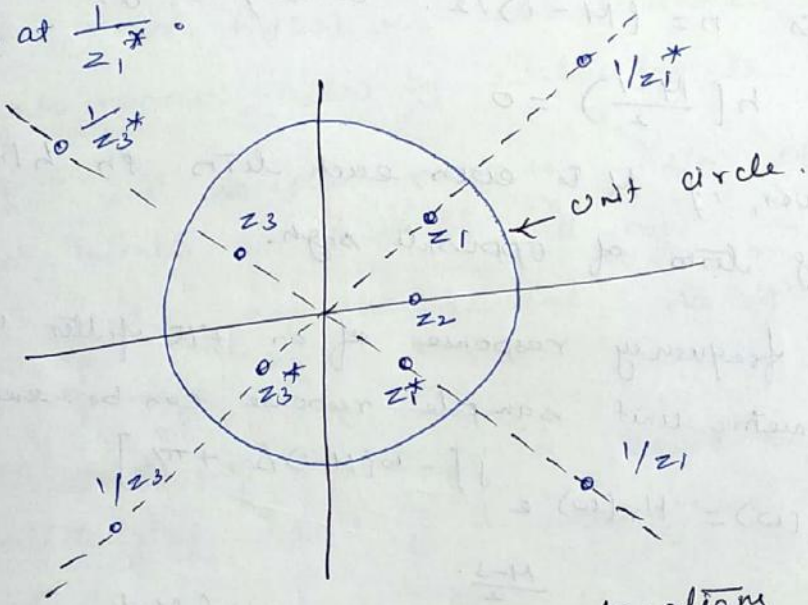


Fig. Symmetry of zero locations for a linear phase FIR filter.

The frequency response characteristics of linear-phase FIR filters are obtained by evaluating (6) on the unit circle. This substitution yields the expression for $H(\omega)$.

When $h(n) = h(M-1-n)$, $H(\omega)$ can be expressed as

$$H(\omega) = H_r(\omega) e^{-j\omega(M-1)/2} \quad \text{—— (8)}$$

where $H_r(\omega)$ is a real function of ω and can be expressed as

$$H_r(\omega) = h\left(\frac{M-1}{2}\right) + 2\sum_{n=0}^{(M-3)/2} h(n)\cos\omega\left(\frac{M-1}{2} - n\right), \quad M\text{-odd}$$

$$H_r(\omega) = 2\sum_{n=0}^{(M/2)-1} h(n)\cos\omega\left(\frac{M-1}{2} - n\right), \quad M\text{ even} \quad \text{—— (9)}$$

The phase characteristics of the filter for both M-odd and M even is,

$$\Theta(\omega) = \begin{cases} -\omega\left(\frac{M-1}{2}\right) & \text{if } H_r(\omega) > 0 \\ -\omega\left(\frac{M-1}{2}\right) + \pi & \text{if } H_r(\omega) < 0 \end{cases}$$

when $h(n) = -h(M-1-n)$

the unit sample response is antisymmetric.

For M-odd, the center point of the anti-symmetric $h(n)$ is $n = (M-1)/2$. consequently,

$$h\left(\frac{M-1}{2}\right) = 0$$

However, if M is even, each term in $h(n)$ has a matching term of opposite sign.

The frequency response of an FIR filter with an antisymmetric unit sample response can be expressed as,

$$H(\omega) = H_r(\omega)\, e^{j[-\omega(M-1)/2 + \pi/2]}$$

where

$$H_r(\omega) = 2 \sum_{n=0}^{\frac{M-3}{2}} h(n) \sin\omega\left(\frac{M-1}{2} - n\right), \quad M\text{-odd}$$

$$= 2 \sum_{n=0}^{\frac{M}{2}-1} h(n) \sin\omega\left(\frac{M-1}{2} - n\right), \quad M\text{-even}$$

The phase characteristic of the filter for both M-odd and M even is

$$\Theta(\omega) = \begin{cases} \frac{\pi}{2} - \omega\left(\frac{M-1}{2}\right), & \text{if } H_r(\omega) > 0 \\ \frac{3\pi}{2} - \omega\left(\frac{M-1}{2}\right), & \text{if } H_r(\omega) < 0 \end{cases}$$

For a symmetric $h(n)$,

the number of filter co-effi that specify the frequency response is $\frac{M+1}{2}$ for M-odd, $\frac{M}{2}$ for M is even.

If the unit sample response is antisymmetric,

$h\left(\frac{M-1}{2}\right) = 0 \Rightarrow \frac{M-1}{2}$ co-eff for M-odd

$M/2$ co-eff for M-even.

DESIGN OF LINEAR PHASE FIR FILTERS USING WINDOWS

This method of design starts with the desired frequency response specification $Hd(\omega)$ and the corresponding unit sample response $hd(n)$ is determined.

$$Hd(\omega) = \sum_{n=0}^{\infty} hd(n) e^{-j\omega n} \quad \text{—— (1)}$$

where $$hd(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} Hd(\omega) e^{j\omega n} d\omega \quad \text{—— (2)}$$

Thus given $Hd(\omega)$, we can determine the unit sample response $hd(n)$ by evaluating the integral in (2)

The unit sample response $hd(n)$ obtained from (2) is infinite in duration and must be truncated at some point, say at $n = M-1$ to yield an FIR filter of length M.

Truncation of $hd(n)$ to a length $M-1$ is equivalent to multiplying $hd(n)$ by a "rectangular window" defined as,

$$w(n) = \begin{cases} 1, & n = 0, 1, \dots M-1 \\ 0, & \text{otherwise} \end{cases} \quad \text{—— (3)}$$

$$\therefore \; h(n) = hd(n)\, w(n)$$

$$= \begin{cases} hd(n) & n = 0, 1, \dots M-1 \\ 0 & \text{otherwise} \end{cases}$$

The multiplication of $hd(n)$ with window function is equivalent to convolution of $Hd(\omega)$ with $W(\omega)$

where $$W(\omega) = \sum_{n=0}^{M-1} w(n) e^{-j\omega n}$$

Thus the convolution of $Hd(\omega)$ with $W(\omega)$ yields the frequency response of the (truncated) FIR filter.

ie, $H(\omega) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_d(\nu)\, W(\omega - \nu)\, d\nu$

The Fourier Transform of rectangular window is,

$$W(\omega) = \sum_{n=0}^{M-1} e^{-j\omega n}$$

$$= \frac{1 - e^{-j\omega M}}{1 - e^{-j\omega}}$$

$$= e^{-j\omega(M-1)/2} \frac{\sin(\omega M/2)}{\sin(\omega/2)}$$

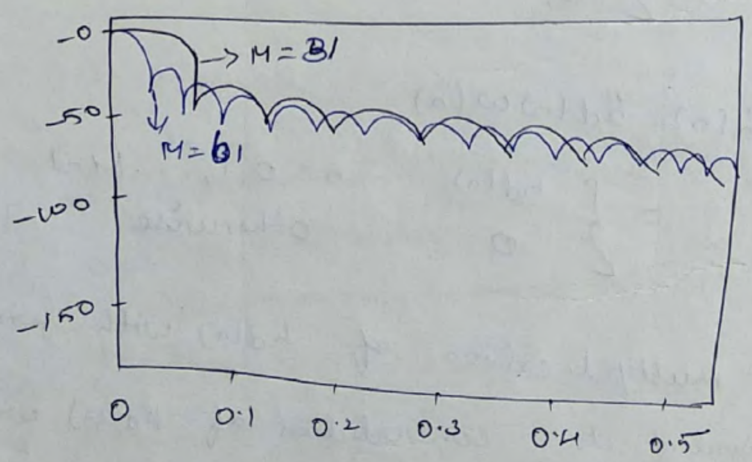
This window function has a magnitude response,

$$|W(\omega)| = \frac{|\sin(\omega M/2)|}{|\sin(\omega/2)|}, \quad \pi \le \omega \le \pi$$

and

$$\Theta(\omega) = \begin{cases} -\omega\left(\frac{M-1}{2}\right) & \text{when } \sin(\omega M/2) \ge 0 \\ -\omega\left(\frac{M-1}{2}\right) + \pi & \text{when } \sin(\omega M/2) < 0 \end{cases}$$

The magnitude response of the window function is,



The width of main lobe is $\frac{4\pi}{M}$. As M increases the width will become narrower. However the sidelobes of $|W(\omega)|$ are relatively high and unaffected by an increase in M.

* The characteristics of rectangular window play a significant role in determining the resulting frequency response of the FIR filter obtained by truncating $hd(n)$ to length M.

* The convolution of $Hd(\omega)$ with $W(\omega)$ has the effect of smoothing $Hd(\omega)$. As M is increased, $W(\omega)$ becomes narrower, and the smoothing provided by $W(\omega)$ is reduced.

* On the otherhand, the large sidelobes of $W(\omega)$ result in some undisirable ringing effects in the FIR filter frequency response $H(\omega)$, and also in relatively larger sidelobes in $H(\omega)$.

* These undesirable effects are best alleviated by the use of windows that do not contain abrupt discontinuities in their time-domain characteristics and have correspondingly low sidelobes in their frequency-domain characteristics.

Window Functions for FIR filter Design

Name of Window	Time domain sequence $h(n)$, $0 \le n \le M-1$
(1) Bartlett (Triangular	$1 - \dfrac{2\left\|n - \frac{M-1}{2}\right\|}{M-1}$
2). Blackman	$0.42 - 0.5\cos\dfrac{2\pi n}{M-1} + 0.08\cos\dfrac{4\pi n}{M-1}$
3) Hamming	$0.54 - 0.46\cos\dfrac{2\pi n}{M-1}$
4) Hanning	$\dfrac{1}{2}\left[1 - \cos\dfrac{2\pi n}{M-1}\right]$

Gibbs phenomenon:-

The multiplication of hd(n) with a rectangular window is identical to truncating the Fourier series representation of the desired filter characteristics Hd(ω). The truncation of the Fourier series is known to introduce ripples in the frequency response characteristic H(ω) due to the nonuniform convergence of the Fourier series at a discontinuity. The oscillatory behavior near the band edge of the Fourier series of filter is called the Gibbs phenomenon.

Important Frequency - Domain Characteristics of some Window Functions

Type of Window	Approximate transition width of main lobe	Peak side lobe (dB)
1) Rectangular	$\frac{4\pi}{M}$	−13
2) Bartlett	$8\pi/M$	−25
3) Hanning	$8\pi/M$	−31
4) Hamming	$8\pi/M$	−41
5) Blackman	$12\pi/M$	−57

① A Low pass filter is to be designed with the following desired frequency response

$$H_d(e^{j\omega}) = \begin{cases} e^{-j2\omega} & -\frac{\pi}{4} \le \omega \le \frac{\pi}{4} \\ 0 & \end{cases}$$

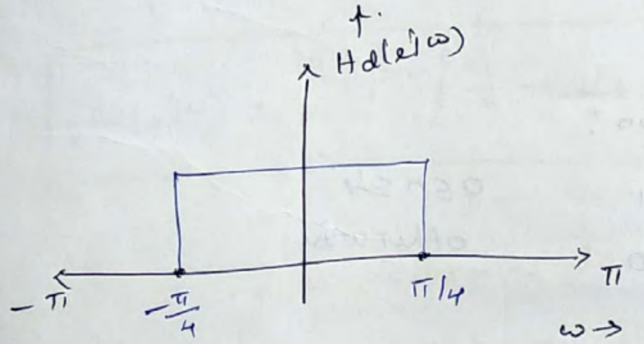
Determine the filter co-efficients $h_d(n)$ if the window function is defined as

$$w(n) = \begin{cases} 1 & 0 \le n \le 4 \\ 0 & \text{otherwise} \end{cases}$$

Also determine the frequency response $H(e^{j\omega})$ of the designed filter.

Solution:

Given $H_d(e^{j\omega}) = \begin{cases} e^{-j2\omega} & -\frac{\pi}{4} \le \omega \le \frac{\pi}{4} \\ 0 & \pi/4 < |\omega| \le \pi \end{cases}$



Step-1

$$h_d(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d(e^{j\omega})\, e^{j\omega n}\, d\omega$$

$$= \frac{1}{2\pi}\int_{-\pi/4}^{\pi/4} e^{-j2\omega}\, e^{j\omega n}\, d\omega$$

$$= \frac{1}{2\pi}\int_{-\pi/4}^{\pi/4} e^{j\omega(n-2)}\, d\omega$$

$$= \frac{1}{2\pi}\left[\frac{e^{j\omega(n-2)}}{j(n-2)}\right]_{-\pi/4}^{\pi/4}$$

$$= \frac{1}{2\pi}\left[\frac{e^{j\frac{\pi}{4}(n-2)} - e^{-j\frac{\pi}{4}(n-2)}}{j(n-2)}\right]$$

$$\boxed{hd(n) = \frac{1}{\pi(n-2)}\left[\sin\left(\frac{\pi}{4}(n-2)\right)\right], \quad n \neq 2}$$

Step-2 hd(n) co-efficients

$$hd(0) = \frac{1}{-2\pi} - \sin\left(\frac{\pi}{2}\right)$$

$$hd(0) = \frac{1}{2\pi} = hd(4)$$

$$hd(1) = \frac{1}{\sqrt{2}\pi} = hd(3)$$

$$hd(2) = \frac{\cos\left(\frac{\pi}{4}(n-2)\right)\cdot\frac{\pi}{4}}{\pi}$$

$$= \frac{1}{4}$$

Step 3

Window Function:

$$w(n) = \begin{cases} 1 & 0 \le n \le 4 \\ 0 & \text{otherwise} \end{cases}$$

Step-4

Window co-efficients:

$w(0) = 1$

$w(1) = 1$

$w(2) = 1$

$w(3) = 1$

$w(4) = 1$

Step-5 FIR filter co-efficients

$h(n) = hd(n)\, w(n)$

$h(0) = hd(0) \cdot w(0) = \frac{1}{2\pi} = h(4)$

$h(1) = \frac{1}{\sqrt{2}\pi} = hd(3)$

$h(2) = 1/4$

step-6: Frequency Response

$$H(e^{j\omega}) = \sum_{n=0}^{4} h(n) e^{-j\omega n}$$

$$= h(0) + h(1) e^{-j\omega} + h(2) e^{-2j\omega} + h(3) e^{-j3\omega} + h(4) e^{-j4\omega}$$

$$= e^{-j2\omega}\left[h(0) e^{j2\omega} + h(4) e^{-j2\omega}\right] + e^{-j2\omega}\left[h(1) e^{j\omega} + h(3) e^{-j\omega}\right] + h(2) e^{-j2\omega}$$

$$= e^{-j2\omega}\left[h(2) + h(0)\left[e^{j2\omega} + e^{-j2\omega}\right] + h(1)\left[e^{j\omega} + e^{-j\omega}\right]\right]$$

$$= e^{-j2\omega}\left[\frac{1}{4} + \frac{1}{\cancel{2}\pi}(\cos 2\omega)\cancel{2} + \frac{1}{\sqrt{2}\pi}(\cos\omega)2\right]$$

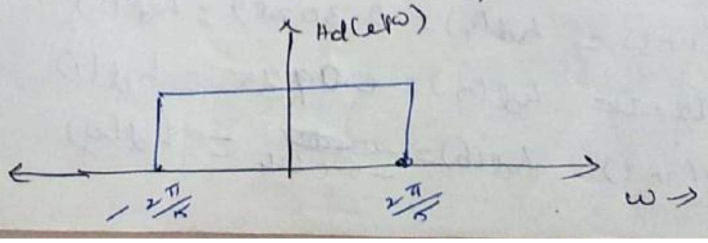
$$\boxed{H(e^{j\omega}) = e^{-j2\omega}\left[\frac{1}{4} + \frac{1}{\pi}\cos 2\omega + \frac{\sqrt{2}}{\pi}\cos\omega\right]}$$

2) Using a rectangular window technique, design a LPF with passband gain of unity, cut-off frequency of 1000 Hz and working at a sampling frequency of 5 kHz. The length of the impulse response should be 7. (A/M-17)

Solution:

Given, $f_c = 1000$ Hz, $F = 5000$ Hz, $a_p = 1$, $M = 7$

Cut off frequency $\omega_c = \frac{2\pi f_c}{F} = \frac{2\pi}{5}$



$$H_d(e^{j\omega}) = \begin{cases} 1 & -\frac{2\pi}{5} \le \omega_c \le \frac{2\pi}{5} \\ 0 & \text{otherwise} \end{cases}$$

$$\therefore h_d(n) = \frac{1}{2\pi} \int_{-\frac{2\pi}{5}}^{\frac{2\pi}{5}} (1)\, e^{j\omega n}\, d\omega$$

$$= \frac{1}{2\pi} \left[\frac{e^{j\omega n}}{jn} \right]_{-\frac{2\pi}{5}}^{\frac{2\pi}{5}}$$

$$= \frac{1}{2\pi} \left[\frac{e^{j\frac{2\pi}{5} n} - e^{-j\frac{2\pi}{5} n}}{jn} \right]$$

$$\boxed{h_d(n) = \frac{1}{\pi(n)} \sin\left[\frac{2\pi}{5} n\right] \qquad n \neq 0}$$

$h_d(0) = 0.4$ $\quad \Big| \because h_d(0) = \dfrac{\cos\left(\frac{2\pi}{5} n\right) \frac{2\pi}{5}}{\pi}$

$h_d(1) = 0.30289$

$h_d(2) = \cancel{0.187}\ 0.0935$

$h_d(3) = \cancel{-0.18719}\ -0.0624$

$h_d(4) = -0.30289$

$h_d(5) = 0$

$h_d(6) = \cancel{0.18719}\ 0.0935$

To design a realizable FIR filter, shift $h_d(n)$ co-efficients to $\frac{M-1}{2}$ units. $\frac{M-1}{2} = \underline{\underline{3}}$

$\therefore h_d(0+3) = h_d(3) = 0.4$

$h_d(1+3) = h_d(4) = 0.30289 = h_d(2)$

$h_d(2+3) = h_d(5) = 0.0935 = h_d(1)$

$h_d(3+3) = h_d(6) = \cancel{0.0935}\ -0.0624 = h_d(0)$

Window Function,

$$\omega(n) = \begin{cases} 1 & 0 \le n \le 6 \\ 0 & \text{else} \end{cases}$$

$\therefore$ FIR filter co-efficients,

$$h(n) = h_d(n)\,\omega(n)$$

$$h(0) = -0.0\cancel{624} = h(6)$$

$$h(1) = \cancel{0.187}\ 0.0935 = h(5)$$

$$h(2) = 0.30289 = h(4)$$

$$h(3) = 0.4$$

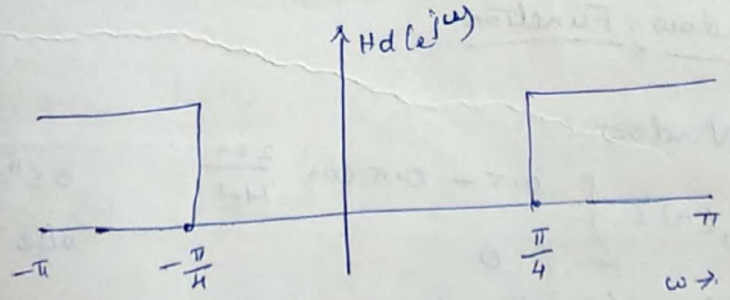
3). Design a FIR filter with the following desired specification

$$H_d(e^{j\omega}) = \begin{cases} 0 & \frac{-\pi}{4} \le \omega \le \frac{\pi}{4} \\ e^{-j2\omega} & \frac{\pi}{4} \le |\omega| \le \pi \end{cases}$$

using a Hanning window with $N=5$. (M/J-13)

Solution:

Given $$H_d(e^{j\omega}) = \begin{cases} 0 & -\frac{\pi}{4} \le \omega \le \frac{\pi}{4} \\ e^{-j2\omega} & \frac{\pi}{4} \le |\omega| \le \pi \end{cases}$$



Step-1

$$h_d(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_d(e^{j\omega})\, e^{j\omega n}\, d\omega$$

$$= \frac{1}{2\pi} \left[\int_{-\pi}^{-\frac{\pi}{4}} e^{-j2\omega}\, e^{j\omega n}\, d\omega + \int_{\pi/4}^{\pi} e^{-j2\omega}\, e^{j\omega n}\, d\omega \right]$$

$$= \frac{1}{2\pi}\left\{\left[\frac{e^{j\omega(n-2)}}{j(n-2)}\right]_{-\pi}^{-\frac{\pi}{4}} + \left[\frac{e^{j\omega(n-2)}}{j(n-2)}\right]_{\frac{\pi}{4}}^{\pi}\right\}$$

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

$$= \frac{1}{(n-2)\pi}\left[\frac{e^{j\pi(n-2)} - e^{-j\pi(n-2)}}{2j} - \frac{e^{j\frac{\pi}{4}(n-2)} - e^{-j\frac{\pi}{4}(n-2)}}{2j}\right]$$

$$\boxed{h_d(n) = \frac{1}{\pi(n-2)}\left[\sin\{\pi(n-2)\} - \sin\{\tfrac{\pi}{4}(n-2)\}\right], \quad n \neq 2}$$

Step-2 $h_d(n)$ co-efficients:

$$h_d(0) = \frac{1}{2\pi} = h_d(4)$$

$$h_d(1) = \frac{1}{\sqrt{2}\pi} = h_d(3)$$

$$h_d(2) = \frac{3}{4}$$

Step-3 Window Function

Hanning Window,

$$w_{Hn}(n) = \begin{cases} 0.5 - 0.5\cos\frac{2\pi n}{M-1} & 0 \le n \le 4 \\ 0 & \text{else} \end{cases}$$

Step-4 Window co-efficients

$w_{Hn}(0) = 0$

$w_{Hn}(1) = 0.5$

$w_{Hn}(2) = 1$

$w_{Hn}(3) = 0.5$

$w_{Hn}(4) = 0$

Step - 5 FIR filter co-efficients

$$h(n) = hd(n)\, \omega_{Hn}(n)$$

$$h(0) = hd(0)\, \omega_{Hn}(0) = \frac{1}{2\pi}(0) = 0 = h(4)$$

$$h(1) = hd(1)\, \omega_{Hn}(1) = \frac{1}{\sqrt{2}\pi}(0.5) = 0.1125 = h(3)$$

$$h(2) = hd(2)\, \omega_{Hn}(2) = \frac{3}{4}(1) = \frac{3}{4} = 0.75$$

4). Design a digital FIR band-pass filter with lower cut-off frequency 2000 Hz and upper cut off frequency 3200 Hz using Hamming window of length N=7. sampling rate is 10000 Hz. (N/D-12)

Solution :-

Given,

Filter Type : Band Pass Filter

Lower cutoff frequency : 2000 Hz

Upper cutoff frequency : 3200 Hz

Sampling rate : 10000 Hz

Length M : 7

window : Hamming window

Step-1 $hd(n)$:

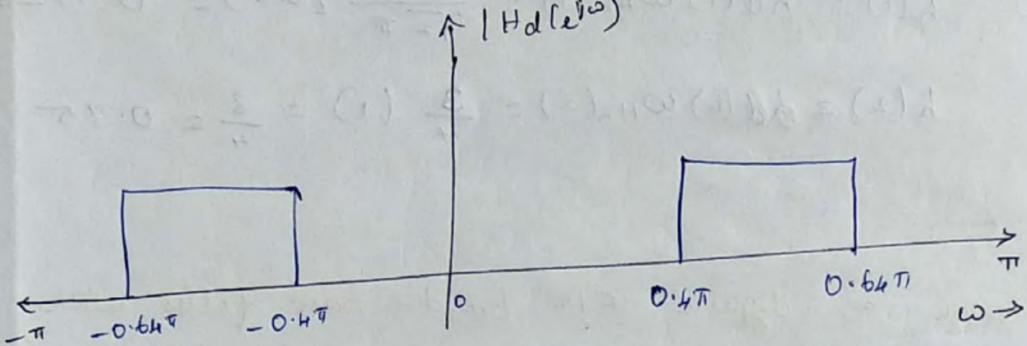
$$H_d(e^{j\omega}) = \begin{cases} e^{-j\alpha\omega} & \omega_{c_1} \le |\omega| \le \omega_{c_2} \\ 0 & \omega_{c_2} \le |\omega| \le \pi \\ 0 & 0 \le |\omega| \le \omega_{c_1} \end{cases}$$

$$\alpha = \frac{M-1}{2} = \frac{6}{2} = 3$$

$$\omega_{c_1} = \frac{2 \times \pi f_{c_1}}{F} = \frac{2 \times \pi \times 2000}{10{,}000} = \frac{2\pi}{5} = 0.4\pi$$

$$\omega_{c_2} = \frac{2\pi f_{c_2}}{F} = \frac{2 \times \pi \times 3200}{10{,}000} = 0.64\pi$$

$$H_d(e^{j\omega}) = \begin{cases} e^{-j3\omega} & 0.4\pi \le |\omega| \le 0.64\pi \\ 0 & 0.64\pi \le |\omega| \le \pi \\ 0 & 0 \le |\omega| \le 0.4\pi \end{cases}$$



$$h_d(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d(e^{j\omega})\, e^{j\omega n}\, d\omega$$

$$= \frac{1}{2\pi}\left[\int_{-0.64\pi}^{-0.4\pi} e^{-j3\omega} e^{j\omega n}\, d\omega + \int_{0.4\pi}^{0.64\pi} e^{-j3\omega} e^{j\omega n}\, d\omega\right]$$

$$= \frac{1}{2\pi}\left\{\left[\frac{e^{j\omega(n-3)}}{j(n-3)}\right]_{-0.64\pi}^{-0.4\pi} + \left[\frac{e^{j\omega(n-3)}}{j(n-3)}\right]_{0.4\pi}^{0.64\pi}\right\}$$

$$= \frac{1}{\pi(n-3)}\left[\frac{e^{-j0.4\pi(n-3)} - e^{-j0.64\pi(n-3)}}{2j} + \frac{e^{j0.64\pi(n-3)} - e^{j0.4\pi(n-3)}}{2j}\right]$$

$$\boxed{h_d(n) = \frac{1}{\pi(n-3)}\left[\sin\{0.64\pi(n-3)\} - \sin\{0.4\pi(n-3)\}\right] \quad n \ne 3}$$

Step-2 $h_d(n)$ co-efficients:

$h_d(0) = \cancel{0.0679}\ 0.03599 = h_d(6)$

$h_d(1) = -0.2162 = h_d(5)$

$h_d(2) = \cancel{-0.06845}\ -0.01472 = h_d(4)$

$$h_d(3) = \frac{\cos[0.64\pi(n-3)] \cdot 0.64\pi - \cos[0.4\pi(n-3)]\ 0.4\pi}{\pi}$$

$= 0.64 - 0.4$

$h_d(3) = 0.24$

Step-3 Window Function:-

$$\omega_H(n) = \begin{cases} 0.54 - 0.46 \cos\frac{2\pi n}{M-1} & 0 \le n \le 6 \\ 0 & \text{else} \end{cases}$$

Step-4 Window co-efficients:

$\omega_H(0) = 0.08 = \omega_H(6)$

$\omega_H(1) = 0.31 = \omega_H(5)$

$\omega_H(2) = 0.77 = \omega_H(4)$

$\omega_H(3) = 1$

Step-5 FIR Filter Co-efficients:

$h(n) = h_d(n)\ \omega_H(n)$

$h(0) = h_d(0) \cdot \omega_H(0) = 0.02879 = h(6)$

$h(1) = h_d(1)\ \omega_H(1) = -0.067 = h(5)$

$h(2) = h_d(2)\ \omega_H(2) = 0.01133 = h(4)$

$h(3) = \cancel{0.24}\ h_d(3)\ \omega_H(3) = 0.24$

DESIGN OF LINEAR PHASE FIR FILTERS BY THE FREQUENCY - SAMPLING METHOD

TI (671 - 672)

In the frequency - sampling method for FIR filter design, we specify the desired frequency response $H_d(\omega)$ at a set of equally spaced frequencies, namely

$$\omega_k = \frac{2\pi}{M}(k+\alpha), \quad k = 0, 1, \ldots \frac{M-1}{2}, \; M \text{ odd}$$

$$k = 0, 1, \ldots \frac{M}{2} - 1, \; M \text{ even}$$

$$\alpha = 0 \text{ (or) } \frac{1}{2} \quad \text{——} (1)$$

and solve for the unit sample response $h(n)$ of the FIR filter from these equally spaced frequency specifications. To reduced sidelobes, it is desirable to optimize the frequency specifications in the transition band of the filter.

A basic symmetry property of the sampled frequency response function to simplify the computations is given here. Let us begin with the desired frequency response of the FIR filter,

$$H(\omega) = \sum_{n=0}^{M-1} h(n) e^{-j\omega n} \quad \text{——} (2)$$

If we specify the frequency response of the filter at the frequencies given by (1), then from (2)

$$H(k+\alpha) \equiv H\left(\frac{2\pi}{M}(k+\alpha)\right)$$

$$H(k+\alpha) \equiv \sum_{n=0}^{M-1} h(n) e^{-j2\pi(k+\alpha)n/M} \quad k = 0, 1, \ldots M-1$$

$$\text{——} (3)$$

It is simple to invert ③ and express $h(n)$ in terms of $H(k+\alpha)$. If we multiply both sides of ③ by the exponential, $\exp(j2\pi km/M)$, $m=0,1,\ldots M-1$ and sum over $k=0,1,\ldots M-1$, the right hand side of ③ reduces to ~~Mh(m)~~ $M\,h(m)\exp(-j2\pi\alpha m/M)$.

$$\therefore\ h(n) = \frac{1}{M}\sum_{k=0}^{M-1} H(k+\alpha)\, e^{2\pi(k+\alpha)n/M}, \quad n=0,1,\ldots M-1 \qquad \text{—— ④}$$

From above eqn ④, we compute the values of the unit sample response $h(n)$ from the specification of the frequency samples $H(k+\alpha)$, $k=0,1,\ldots M-1$.

When $\alpha=0$, eqn ③ reduces to the DFT of the sequence $\{h(n)\}$ and eqn ④ reduces to IDFT.

Since $h(n)$ is real, we can easily show that the frequency samples $\{H(k+\alpha)\}$ satisfy the symmetry condition,

$$H(k+\alpha) = H^*(M-k-\alpha)$$

This symmetry condition, along with the symmetry condition for $h(n)$, can be used to reduce the frequency specifications from M-points to $\frac{M+1}{2}$ points for M odd and $M/2$ points for M even.

Unit sample response $h(n) = \pm h(M-1-n)$

Symmetric

$$H(k) = G(k)\, e^{j\pi k/M} \qquad k = 0,1,\ldots M-1$$

$$G(k) = (-1)^k H_r\left(\frac{2\pi k}{M}\right) \qquad G(k) = -G(M-k)$$

$\alpha = 0$

$$h(n) = \frac{1}{M}\left\{ G(0) + 2\sum_{k=1}^{U} G(k)\cos\frac{2\pi k}{M}\left(n+\frac{1}{2}\right)\right\}$$

$$U = \begin{cases} \frac{M-1}{2} & , M\text{-odd} \\ \frac{M}{2}-1 & M\text{-even}\end{cases}$$

$$H\left(k+\tfrac{1}{2}\right) = G\left(k+\tfrac{1}{2}\right) e^{-j\pi/2}\, e^{j\pi(2k+1)/2M}$$

$$G\left(k+\tfrac{1}{2}\right) = (-1)^k H_r\left[\frac{2\pi}{M}\left(k+\tfrac{1}{2}\right)\right]$$

$\alpha = 1/2$

$$G\left(k+\tfrac{1}{2}\right) = G\left(M-k-\tfrac{1}{2}\right)$$

$$h(n) = \frac{2}{M}\sum_{k=0}^{U} G\left(k+\tfrac{1}{2}\right)\sin\frac{2\pi}{M}\left(k+\tfrac{1}{2}\right)\left(n+\tfrac{1}{2}\right)$$

Antisymmetric

$$H(k) = G(k)\, e^{j\frac{\pi}{2}}\, e^{j\pi k/M} \qquad k = 0,1,\ldots M-1$$

$$G(k) = (-1)^k H_r\left(\frac{2\pi k}{M}\right),$$

$\alpha = 0$

$$G(k) = G(M-k)$$

$$h(n) = -\frac{2}{M}\sum_{k=1}^{\frac{M-1}{2}} G(k)\sin\frac{2\pi k}{M}\left(n+\tfrac{1}{2}\right), \quad M\text{-odd}$$

$$h(n) = \frac{1}{M}\left\{(-1)^{n+1} G(M/2) - 2\sum_{k=1}^{\frac{M}{2}-1} G(k)\sin\frac{2\pi k}{M}\left(n+\tfrac{1}{2}\right)\right\}, \quad M \text{ even}$$

$$H\left(k+\frac{1}{2}\right) = G\left(k+\frac{1}{2}\right) e^{j\pi(2k+1)/2M}$$

$$\alpha = \frac{1}{2} \qquad G\left(k+\frac{1}{2}\right) = (-1)^k H_r\left[\frac{2\pi}{M}\left(k+\frac{1}{2}\right)\right]$$

$$G\left(k+\frac{1}{2}\right) = -G\left(M-k-\frac{1}{2}\right)$$

$$G(M/2) = 0 \quad \text{for } M\text{-odd}$$

$$h(n) = \frac{2}{M}\sum_{k=0}^{V} G\left(k+\frac{1}{2}\right)\cos\frac{2\pi}{M}\left(k+\frac{1}{2}\right)\left(n+\frac{1}{2}\right)$$

$$V = \begin{cases} \frac{M-3}{2} & M\text{-odd} \\ \frac{M}{2}-1 & M\text{-even} \end{cases}$$

---

1. Determine the co-efficients of a linear phase FIR filter of length $M=15$ which has a symmetric unit sample response and a frequency response that satisfies the condition

$$H_r\left(\frac{2\pi k}{15}\right) = \begin{cases} 1 & k=0,1,2,3 \\ 0.4 & k=4 \\ 0 & k=5,6,7 \end{cases}$$

Solution:-

Since $h(n)$ is symmetric and frequencies are selected to correspond to the case $\alpha=0$, use the symmetric formula to find $h(n)$.

$$G(k) = (-1)^k H_r\left(\frac{2\pi k}{M}\right)$$

$G(0) = 1 \quad G(1) = -1 \quad G(2) = 1 \quad G(3) = -1$

$G(4) = 0.4 \quad G(5) = 0 \quad G(6) = 0 \quad G(7) = 0$

$$h(n) = \frac{1}{15}\left\{G(0) + 2\sum_{k=1}^{U} G(k)\cos\left[\frac{2\pi k}{M}\left(n+\frac{1}{2}\right)\right]\right\}$$

$$U = \frac{M-1}{2} = 7$$

$$h(n) = \frac{1}{15}\left\{ 1 + 2\sum_{k=1}^{7} a(k)\cos\left[\frac{2\pi k}{15}\left(n+\frac{1}{2}\right)\right]\right\}$$

$$h(0) = \frac{1}{15}\left\{ 1 + 2\left[(-1)(0.9781) + 1(0.91354) + (-1)(0.809) + (0.4)(0.669)\right]\right\}$$

$$= \frac{1}{15}\left\{-0.21192\right\}$$

$h(0) = -0.01412893 = h(14)$

$h(1) = -0.001945309 = h(13)$

$h(2) = 0.04000004 = h(12)$

$h(3) = 0.01223454 = h(11)$

$h(4) = -0.09138802 = h(10)$

$h(5) = -0.01808986 = h(9)$

$h(6) = 0.3133176 = h(8)$

$h(7) = 0.52$

2). Determine the co-efficients of a linear phase FIR filter of length $N=15$ which has a symmetric unit sample response and a frequency response that satisfies the conditions

$$H\left(\frac{2\pi k}{15}\right) = \begin{cases} 1 & \text{for } k=0,1,2,3 \\ 0 & \text{for } k=4,5,6,7 \end{cases}$$

(A/M–17)

Solution

$$H\left(\frac{2\pi k}{15}\right) = \begin{cases} 1 & k=0,1,2,3 \\ 0 & k=4,5,6,7 \end{cases} \qquad \alpha=0$$

$$G(k) = (-1)^k H\left(\frac{2\pi k}{15}\right)$$

$$G(0) = 1$$

$$G(1) = -1$$

$$G(2) = 1$$

$$G(3) = -1$$

$$G(4) = G(5) = G(6) = G(7) = 0.$$

$$h(n) = \frac{1}{M}\left\{ G(0) + \sum_{k=1}^{U} G(k) \cos\left[\frac{2\pi k}{M}\left(n+\frac{1}{2}\right)\right]\right\}$$

$$h(n) = \frac{1}{15}\left\{ G(0) + \sum_{k=1}^{7} G(k) \cos\left[\frac{2\pi k}{15}\left(n+\frac{1}{2}\right)\right]\right\}$$

$$h(0) = \frac{1}{15}\left\{ 1 + \sum_{k=1}^{7} G(k) \cos\left[\frac{\cancel{2\pi}}{15} k\left(\frac{1}{2}\right)\right]\right\}$$

$$= \frac{1}{15}\left[1 + 2\left(-0.9781476 + 0.913545 - 0.8090169\right)\right]$$

$$h(0) = -0.049815 = h(14)$$

$$h(1) = 0.041 = h(13)$$

$$h(2) = 0.0666 = h(12)$$

$$h(3) = -0.0365 = h(11)$$

$$h(4) = -0.1078 = h(10)$$

$$h(5) = 0.034 = h(9)$$

$$h(6) = 0.3188 = h(8)$$

$$h(7) = 0.466$$

FOURIER SERIES METHOD

The sinusoidal steady-state transfer function of a digital filter is periodic in the sampling frequency and it can be expanded in a Fourier series.

Thus $H(z)\Big|_{z=e^{j\omega}} = H(e^{j\omega})$

$$= \sum_{n=-\infty}^{\infty} h(n) e^{-j\omega n T} \quad \text{——} (1)$$

where $h(n)$ represents the terms of the unit impulse response. In the equation (1) the filter is assumed to be non-causal and it can be modified to yield a causal filter.

Rearranging the transfer function given below into real and imaginary components, we get

$$H(e^{j\omega}) = \sum_{n=-\infty}^{\infty} h(n) \cos \omega n T - j \sum_{n=-\infty}^{\infty} h(n) \sin \omega n T$$

ie, $H(e^{j\omega}) = H_r(f) - j H_i(f)$

where $H_r(f) = \sum_{n=-\infty}^{\infty} h(n) \cos 2\pi f n T$ —— (2)

and $H_i(f) = \sum_{n=-\infty}^{\infty} h(n) \sin 2\pi f n T$ —— (3)

$H_r(f)$ is an even function and $H_i(f)$ is an odd function of frequency. If $h(nT)$ is an even sequence, the imaginary part of the transfer function, $H_i(f)$ will be zero and if $h(nT)$ is an odd sequence, the real part of the transfer function, $H_r(f)$ will be zero. Thus an even unit impulse response yields a real transfer function and an odd unit impulse response yields an imaginary transfer function.

DESIGN EQUATIONS

The term $H(e^{j\omega})$ is periodic in the sampling frequency and hence both $H_r(f)$ and $H_i(f)$ are also periodic in the sampling frequency.

Since the real part of the TF, $H_r(f)$ is an even function of frequency, its Fourier series will be of the form

$$H_r(f) = a_0 + \sum_{n=1}^{\infty} a_n \cos(2\pi f n T)$$

The Fourier co-eff. a_n are given by

$$a_n = \frac{2}{f_s} \int_{-f_s/2}^{f_s/2} H_r(f) \cos(2\pi f n T)\, df \quad , \quad n \neq 0$$

and the a_0 term is given by

$$a_0 = \frac{1}{f_s} \int_{-f_s/2}^{f_s/2} H_r(f)\, df$$

Similarly, the imaginary part of the TF, which is an odd function of frequency can be expanded in the Fourier series

$$H_i(f) = \sum_{n=1}^{\infty} b_n \sin(2\pi f n T)$$

The Fourier coeff. b_n are given by

$$b_n = \frac{2}{f_s} \int_{-f_s/2}^{f_s/2} H_i(f) \sin(2\pi f n T)\, df$$

$b_0 = 0$

For $H_i(f) = 0$

$$H(e^{j\omega}) = a_0 + \sum_{n=1}^{\infty} \frac{a_n}{2} (z^n + z^{-n}) \Big/_{z = e^{j2\pi f T}}$$

The TF can also be written as,

$$H(e^{j\omega}) = h(0) + \sum_{n=1}^{\infty} \left[h(-n) z^n + h(n) z^{-n} \right] \Big/_{z = e^{j2\pi f T}}$$

$$\left.\begin{aligned} h(0) &= a_0 \\ h(-n) &= \frac{1}{2} a_n \\ h(n) &= \frac{1}{2} a_n \end{aligned}\right\} n > 0$$

For $H_r(f) = 0$

$$H(e^{j2\pi fT}) = \sum_{n=1}^{\infty} \frac{b_n}{2}\left[z^{n} - z^{-n}\right] \Big/_{z = e^{j2\pi fT}}$$

$$\left.\begin{aligned} h(-n) &= \tfrac{1}{2} b_n \\ h(n) &= -\tfrac{1}{2} b_n \end{aligned}\right\} n > 0$$

Design Procedure:

1. Decide whether $H_r(f)$ (or) $H_i(f)$ is to be set equal to zero. For filtering applications we typically set $H_i(f) = 0$.
2. Expand $H_r(f)$ or $H_i(f)$ in a Fourier series
3. The unit impulse response is determined from the Fourier co-efficients using eqns

$H_i(f) \equiv 0$
$$\left.\begin{aligned} h(0) &= a_0 \\ h(-n) &= \tfrac{1}{2} a_n \\ h(n) &= \tfrac{1}{2} a_n \end{aligned}\right\} n > 0$$

$H_r(f) \equiv 0$
$$\left.\begin{aligned} h(-n) &= \tfrac{1}{2} b_n \\ h(n) &= -\tfrac{1}{2} b_n \end{aligned}\right\} n > 0$$

1. Use the Fourier series method to design a low-pass digital filter to approximate the ideal specifications given by

$$H(e^{j\omega}) = \begin{cases} 1 & \text{for } |f| \le f_p \\ 0 & f_p < |f| < F/2 \end{cases}$$

where f_p is the passband frequency and F is the sampling frequency.

Solution:

For filtering applications,

$$H(e^{j\omega}) = H_r(f) = \begin{cases} 1 & |f| \le f_p \\ 0 & f_p < |f| \le F/2 \end{cases}$$

$$a_n = \frac{2}{F}\int_{-F/2}^{F/2} H_r(f)\cos(2\pi f n T)\,df, \quad n \neq 0$$

$$= \frac{2}{F}\int_{-f_p}^{f_p} 1 \cdot \cos(2\pi f n T)\,df$$

$$= \frac{2}{T}\left[\frac{\sin 2\pi f n T}{2\pi n T}\right]_{-f_p}^{f_p}$$

Replacing T by $1/F$ and multiplying both numerator and denominator by f_p, we get

$$a_n = \left(\frac{2f_p}{F}\right)\left[\frac{\sin 2\pi n f_p/F}{\pi n (f_p/F)}\right]$$

$$a_0 = \frac{1}{F}\int_{-F/2}^{F/2} H_r(f)\,df = \frac{2f_p}{F}$$

$$\therefore \quad h(0) = a_0 = \frac{2f_p}{F}$$

$$h(-n) = \left(\frac{f_p}{F}\right)\left[\frac{\sin 2\pi n f_p/F}{\pi n (f_p/F)}\right]$$

and $$h(n) = \left(\frac{f_p}{F}\right)\left[\frac{\sin 2\pi n f_p/F}{\pi n (f_p/F)}\right]$$

Structures of FIR filters

FIR system is described by the difference equation,

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

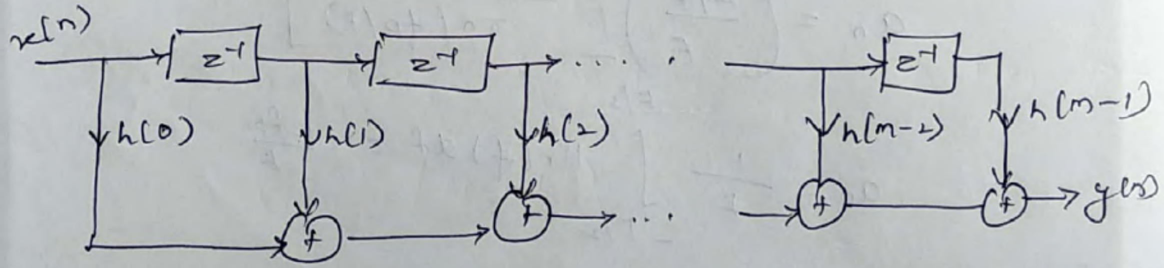
(or)

$$H(z) = \sum_{k=0}^{M-1} b_k z^{-k}$$

and $h(n) = \begin{cases} b_n & 0 \le n \le M-1 \\ 0 & \text{otherwise} \end{cases}$

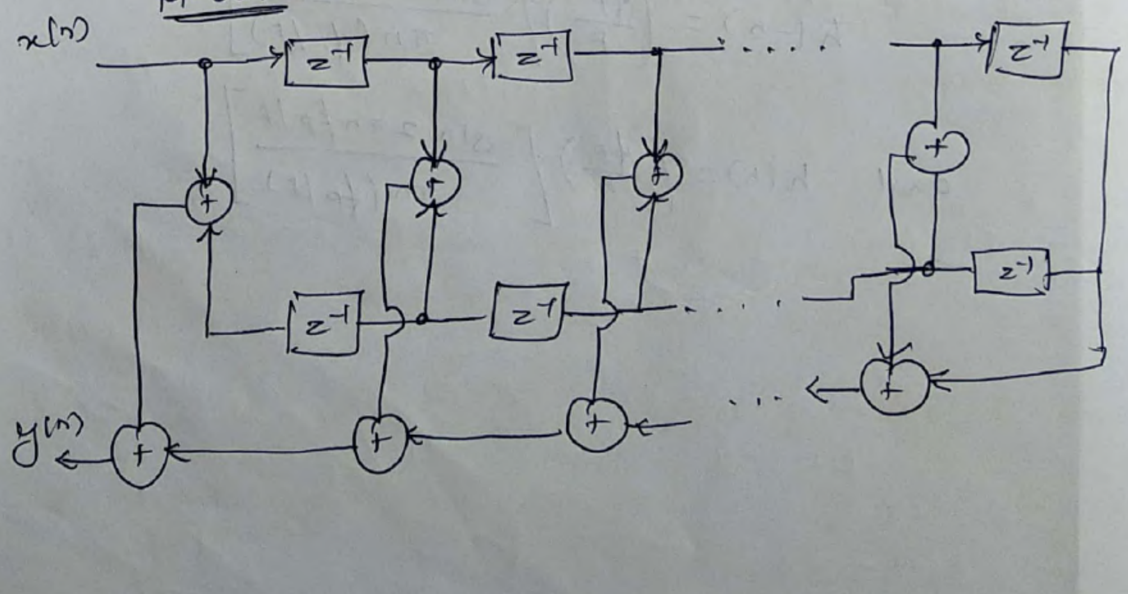
Direct form structure

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



Direct form realization of Linear phase FIR filters:

M-~~odd~~ even



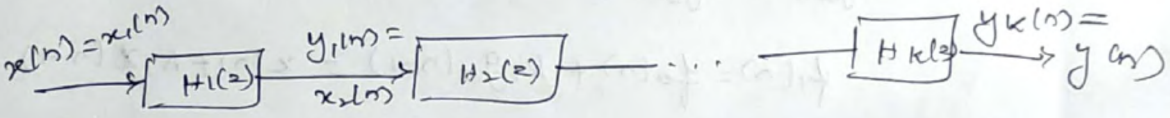
Cascade-Form Structure:

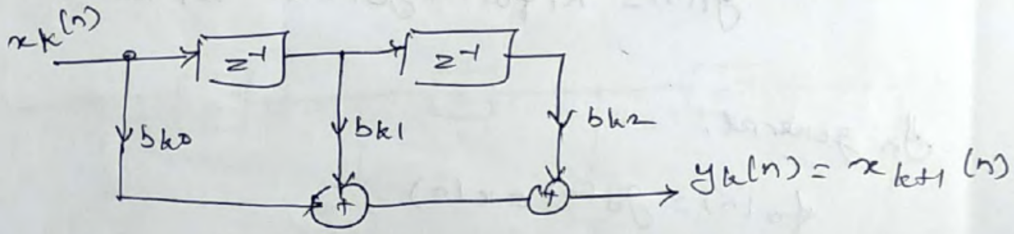
Express $H(z)$ into product of second-order systems,

$$H(z) = \prod_{k=1}^{K} H_k(z)$$

where $H_k(z) = b_{k0} + b_{k1} z^{-1} + b_{k2} z^{-2}$ $\quad k = 1, 2 \ldots K$

K = integer part of $(M+1)/2$

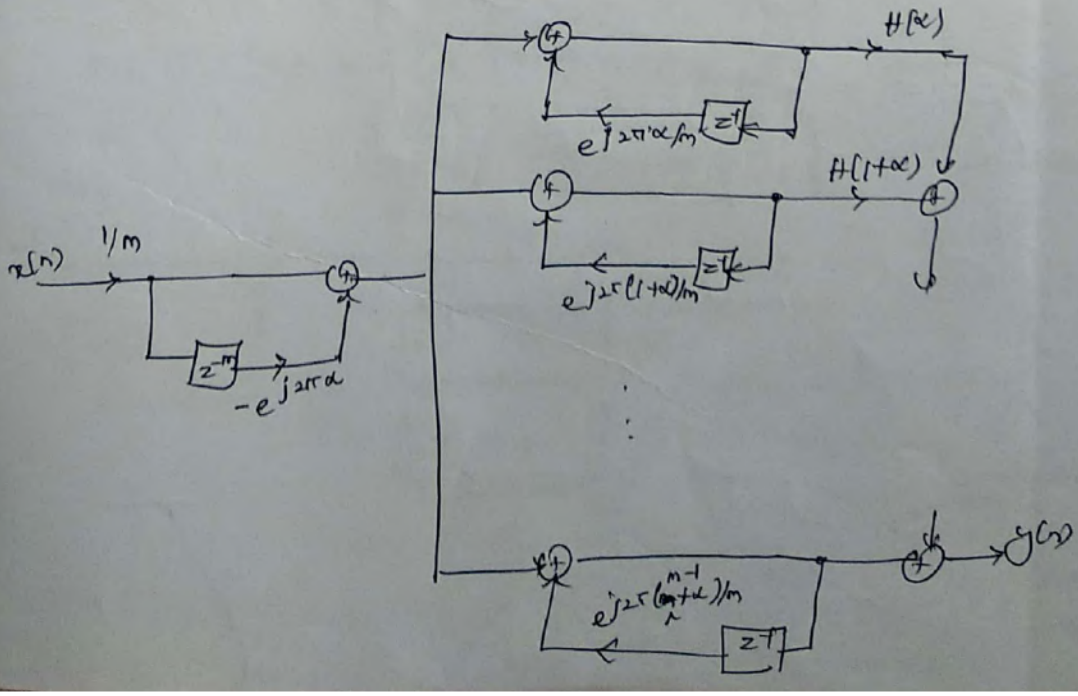




Frequency Sampling Structures

$$H(z) = \sum_{k=0}^{M-1} H(k+\alpha) \left[\frac{1}{M} \sum_{n=0}^{M-1} \left(e^{j2\pi(k+\alpha)/M} z^{-1} \right)^n \right]$$

$$= \frac{1 - z^{-M} e^{j2\pi\alpha}}{M} \sum_{k=0}^{M-1} \frac{H(k+\alpha)}{1 - e^{j2\pi(k+\alpha)/M} z^{-1}}$$

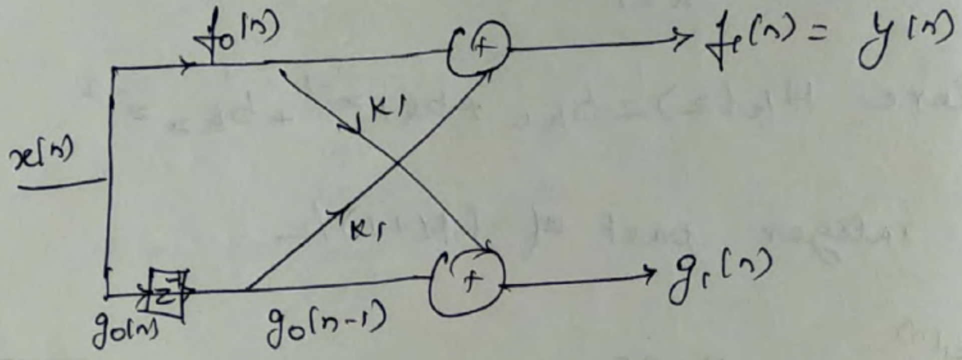


Lattice structure :

For M=1

$$y(n) = x(n) + \alpha_1(1)\, x(n-1)$$

$$k_1 = \alpha_1(1) \rightarrow \text{Reflection co-efficient.}$$



$$f_1(n) = f_0(n) + k_1 g_0(n-1) = x(n) + k_1 x(n-1)$$

$$g_1(n) = k_1 f_0(n) + g_0(n-1) = k_1 x(n) + x(n-1)$$

In general,

$$f_0(n) = g_0(n) = x(n)$$

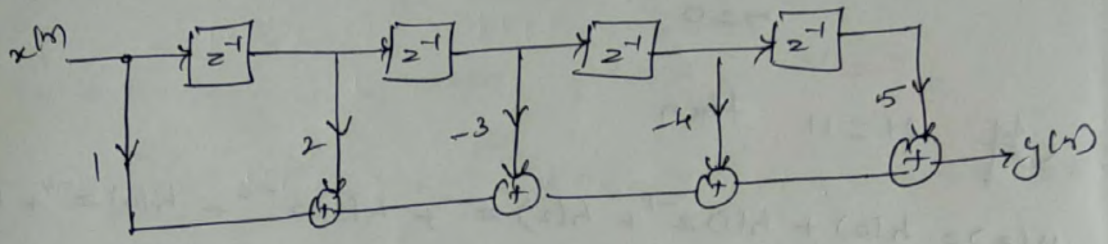
$$f_m(n) = f_{m-1}(n) + k_m g_{m-1}(n-1) \qquad m = 1, 2, \ldots M-1$$

$$g_m(n) = k_m f_{m-1}(n) + g_{m-1}(n-1) \qquad m = 1, 2, \ldots M-1$$

$$y(n) = f_{M-1}(n)$$

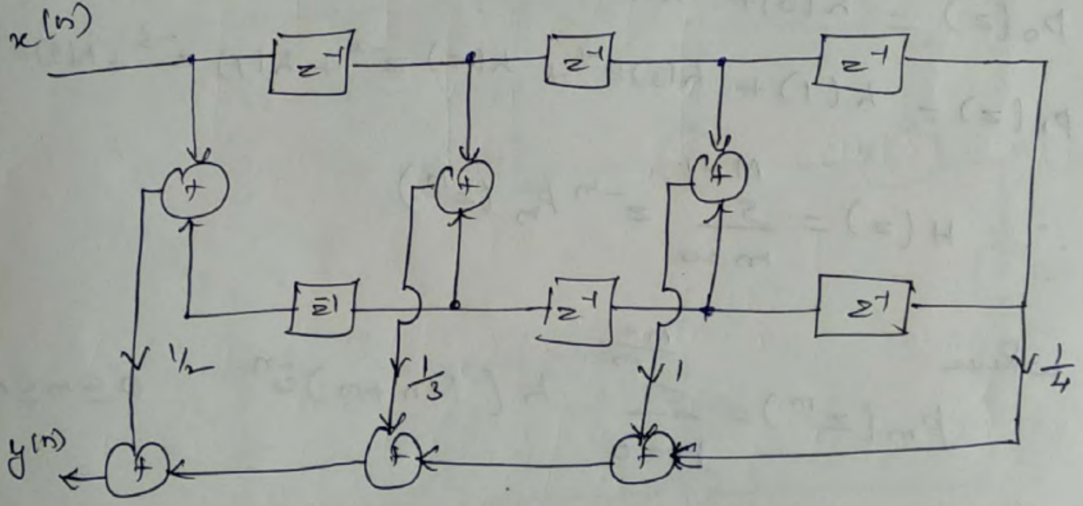
1) Determine the direct form realization of

$$H(z) = 1 + 2z^{-1} - 3z^{-2} - 4z^{-3} + 5z^{-4}$$



2). Draw the linear phase realization of

$$H(z) = \frac{1}{2} + \frac{1}{3}z^{-1} + z^{-2} + \frac{1}{4}z^{-3} + z^{-4} + \frac{1}{3}z^{-5} + \frac{1}{2}z^{-6}$$



3). Obtain the cascade realization of

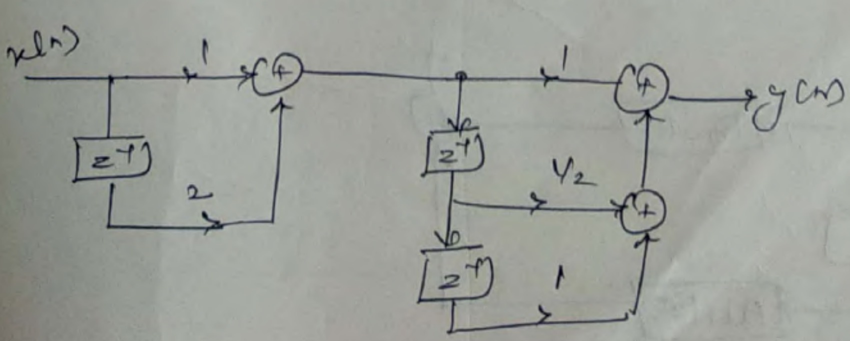
$$H(z) = 1 + \frac{5}{2}z^{-1} + 2z^{-2} + 2z^{-3}$$

$$H(z) = (1 + 2z^{-1})\left(1 + \frac{1}{2}z^{-1} + z^{-2}\right)$$

$\therefore z^{-1} = -2$

$z^{-1} = -0.25 - j\,0.968$

$z^{-1} = -0.25 + j\,0.968$



Poly Phase Realization of FIR filters

$$H(z) = \sum_{n=0}^{M-1} h(n) z^{-n}$$

If $M = 11$ then

$$H(z) = h(0) + h(1)z^{-1} + h(2)z^{-2} + h(3)z^{-3} + h(4)z^{-4} + h(5)z^{-5} + h(6)z^{-6} + h(7)z^{-7} + h(8)z^{-8} + h(9)z^{-9} + h(10)z^{-10}$$

Divide the into 2 subsystems

$$H(z) = P_0(z^2) + z^{-1} P_1(z^2)$$

$$P_0(z) = h(0) + h(2)z^{-1} + h(4)z^{-2} + \dots + h(10)z^{-5}$$

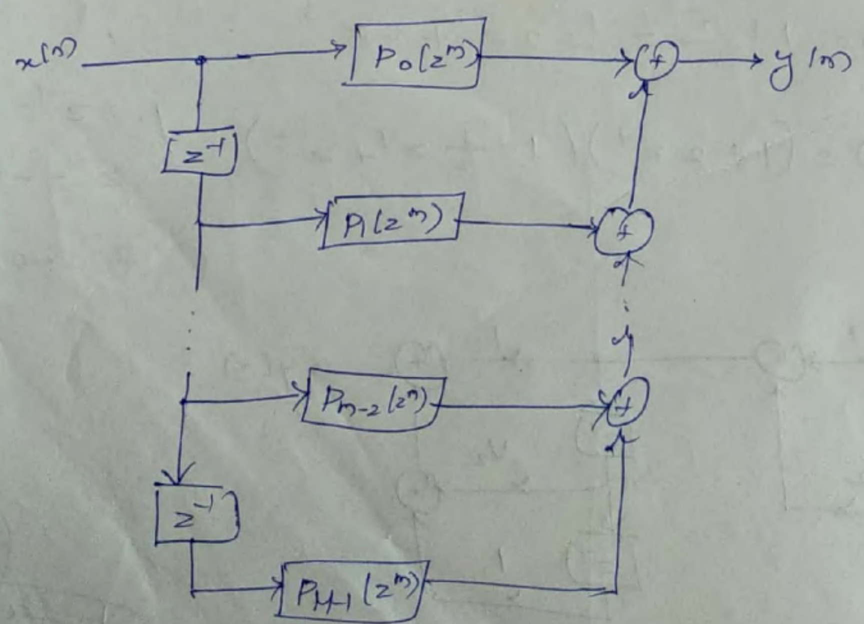
$$P_1(z) = h(1) + h(3)z^{-1} + h(5)z^{-2} + h(7)z^{-3} + h(9)z^{-4}$$

$$\therefore \quad H(z) = \sum_{m=0}^{M-1} z^{-m} P_m(z^M)$$

where

$$P_m(z^M) = \sum_{n=0}^{\frac{M+1}{M}} h(Mn+m) z^{-n} \qquad 0 \le m \le M-1$$

$P_m(z^n)$ can be realized in direct form



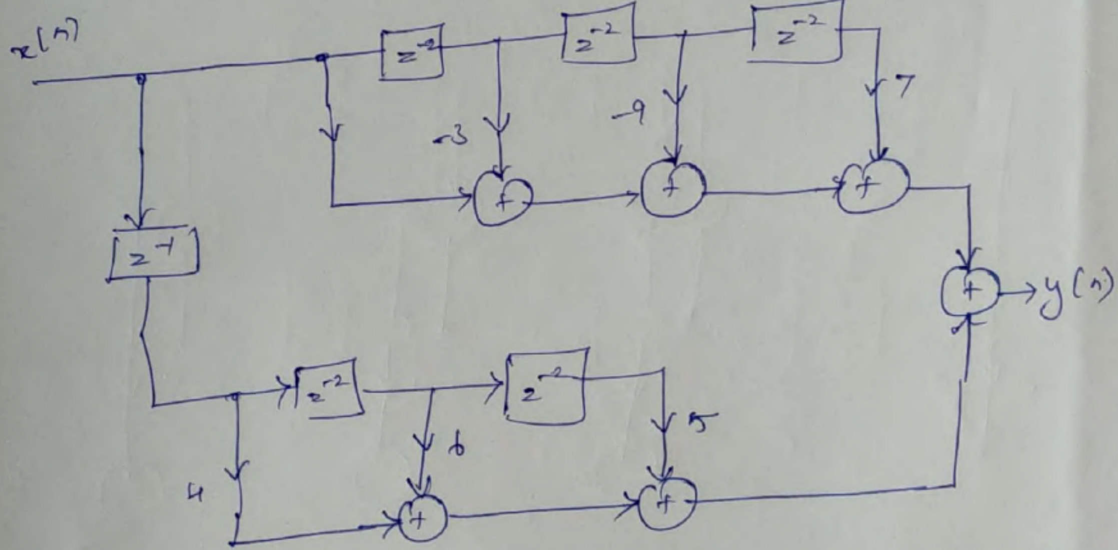
$$H(z) = 1 + 4z^{-1} - 3z^{-2} + 6z^{-3} - 9z^{-4} + 5z^{-5} + 7z^{-6}$$

Divide by 2 groups $M = 2$, $N = 7$

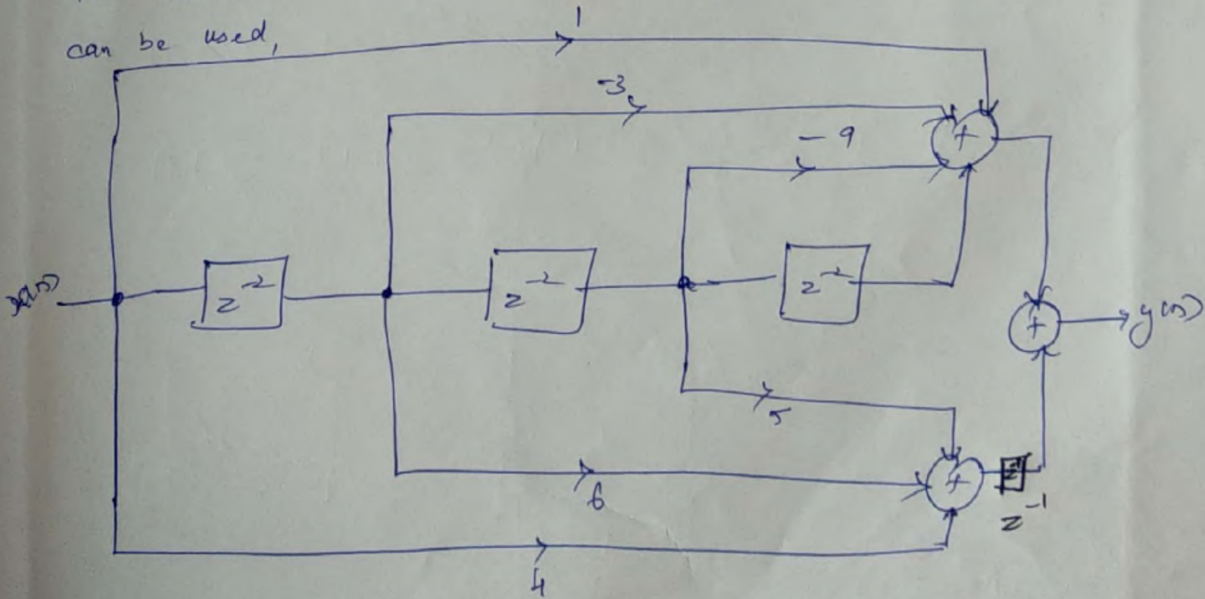
$$\therefore H(z) = (1 - 3z^{-2} - 9z^{-4} + 7z^{-6}) + (4z^{-1} + 6z^{-3} + 5z^{-5})$$

$$= (1 - 3z^{-2} - 9z^{-4} + 7z^{-6}) + z^{-1}(4 + 6z^{-2} + 5z^{-4})$$

$$= p_0(z^2) + z^{-1} p_1(z^2)$$



To reduce the number of delay elements, the following can be used,



UNIT – IV FINITE WORD LENGTH EFFECTS

1. Represent the following numbers in floating point format with five bits in mantissa and three bits in exponent. (8) [M/J – 13 R08]

a) 7_{10}
b) 0.25_{10}
c) -1_{10}
d) -0.25_{10}

Solution:

a) 7_{10}

$(7)_{10} = 0.875 \text{ x } 2^3 = 0.1110 \text{ x } 2^{011}$

b) 0.25_{10}

$(0.25)_{10} = 0.125 \text{ x } 2^1 = 0.0010 \text{ x } 2^{001}$

c) -1_{10}

Assume sign magnitude representation for negative number

$(-1)_{10} = -0.5 \text{ x } 2^1 = 1.1000 \text{ x } 2^{001}$

d) -0.25_{10}

Assume sign magnitude representation for negative number

$(-0.25)_{10} = -0.125 \text{ x } 2^1 = 1.0010 \text{ x } 2^{001}$

2. Distinguish between fixed point and floating point arithmetic. (4) [N/D – 11 R08] [M/J – 12 R08]

S.No.	Fixed Point Arithmetic	Floating Point Arithmetic
1	Fast Operation	Slow Operation
2	Relatively economical	More expensive because of costlier hardware
3	Small dynamic range	Increased dynamic range
4	Round off error occur only for additions	Round off errors can occur with both additions and multiplication
5	Overflow occur in addition	Overflow does not arise
6	Used in small computers	Used in general purpose computers

3. Discuss the various common methods of quantization. (8) [N/D – 13 R08]

The common methods of quantization are

i. Truncation
ii. Rounding

Truncation:

Truncation is process of reducing the size of binary number by discarding all bits less significant than the least significant bit that is retained. In the truncation of a binary number to b bits, all the less significant bits beyond b^{th} bit are discarded.

Examples:

i. The 8 bit binary number 0.00110011 may be truncated to 4 bits as 0.0011
ii. The 8 bit binary number 1.01001001 may be truncated to 4 bits as 1.0100

When a number is truncated, the signal value is approximated by the highest quantization level that is not greater than the signal.

Rounding:

Rounding is the process of reducing the size of a binary number to finite word size of b bits such that, the rounded b bit number is closest to the original unquantized number.

Examples:

i. The binary number 0.11010 may be rounded to three bits as either 0.110 or 0.111.
ii. If the binary number 0.110111111 is rounded to 8 bits then the result may be 0.11011111 or 0.1110000.

Rounding up or down will have negligible effect on accuracy of computation.

4. Explain the errors due to rounding and truncation. [N/D – 10 R08]
Explain the problems due to round off and truncation in converting a decimal fraction. (8) [N/D – 10 R08]
Compare the truncation and rounding errors using fixed point and floating point representation. (8) [M/J – 14 R08]

Rounding or truncation introduces an error whose magnitude depends on the number of bits truncated or rounded-off. Also, the characteristics of the error depend on the form of binary number representation.

Consider a number x, whose original length is 'L' bits, as shown below.

	Sign bit	0	1	2		L–1
x =	X	X	X	X	….	X

Let this number be quantized (truncated or rounded) to 'B' bits, as shown below.

	Sign bit	0	1	2		B–1
Q(x) =	X	X	X	X	…	X

The quantized number is represented as Q(x). Here $B < L$.

i) Truncation error for sign magnitude representation:

When the number x is positive, truncation results in reducing the magnitude of the number. Thus the truncation error is negative and the range is given by,

$$-(2^{-B} - 2^{-L}) \leq \varepsilon_T \leq 0 \qquad (1)$$

The largest error occurs when all the discarded bits are one's.

When the number x is negative, truncation results in reducing the magnitude of the number. Because of the negative number the resulting will be greater than the original number. Thus the truncation error is positive and the range is given by,

$$0 \leq \varepsilon_T \leq (2^{-B} - 2^{-L}) \qquad (2)$$

The overall range for the sign magnitude representation is

$$-(2^{-B} - 2^{-L}) \leq \varepsilon_T \leq (2^{-B} - 2^{-L}) \qquad (3)$$

ii) Truncation error for two's complement representation:

When the number x is positive, truncation results in reducing the magnitude of the number as in the case of sign magnitude numbers. Thus the truncation error is negative. When the number x is negative, truncation results in smaller number. Thus the truncation error is negative. The complete range of truncation error for two's complement is given by,

$$-(2^{-B} - 2^{-L}) \leq \varepsilon_T \leq 0 \qquad (4)$$

iii) Round-off error for sign magnitude and two's complement representation:

The rounding of a binary number involves only the magnitude of the number and is independent of the type of fixed point binary representation. The error due to rounding may be either positive or negative. Its range is

$$-\frac{(2^{-B}-2^{-L})}{2} \leq \varepsilon_R \leq \frac{(2^{-B}-2^{-L})}{2} \qquad (5)$$

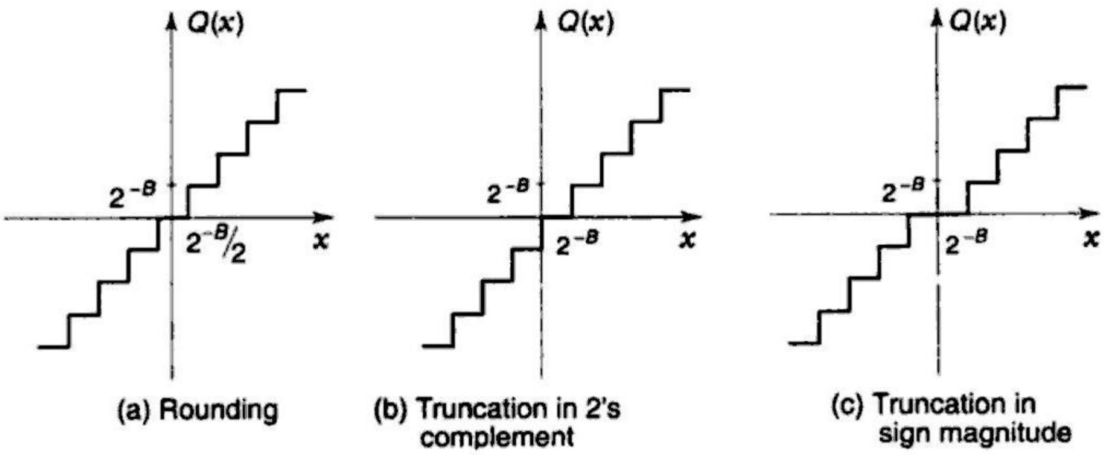


Fig. 1 ***Quantization Error in Rounding and Truncation***

In most of the applications, infinite precision is assumed i.e., $L = \infty$

Therefore,

i) Truncation error for sign magnitude representation

$$-2^{-B} \leq \epsilon_T \leq 2^{-B}$$

ii) Truncation error for two's complement representation

$$-2^{-B} \leq \epsilon_T \leq 0$$

iii) Rounding error for sign magnitude and two's complement representation

$$\frac{-2^{-B}}{2} \leq \epsilon_T \leq \frac{2^{-B}}{2}$$

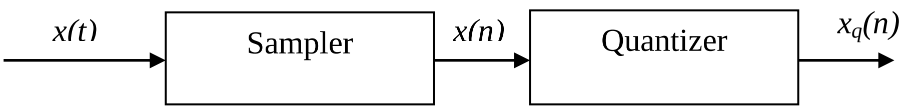
5. What is quantization noise? Derive the expression for quantization noise power. (12) [M/J – 12 R08]

What is meant by quantization? Derive the expression for the quantization error. [N/D – 12 R08]

Explain the quantization noise and derive the expression for finding quantization noise power. (8) [N/D – 10 R08]

QUANTIZATION:

In DSP, the continuous time input signals are converted into digital using ADC. The process of converting analog signal to digital signal is given below.



The signal $x(t)$ is sampled at regular intervals $t = nT$, where $n = 0, 1, 2, ...$ to create a sequence $x(n)$. This is done by a sampler. The numeric equivalent of each sample $x(n)$ is expressed by a finite number of bits giving the sequence $x_q(n)$. The difference signal $e(n) = x_q(n) - x(t)$ is called quantization noise or A/D conversion noise.

The common methods of quantization are

1) Truncation
2) Rounding

EXPRESSION FOR QUANTIZATION NOISE POWER:

Input quantization error:

The quantization error is given by

$$e(n) = x_q(n) - x(n)$$

where

$x_q(n)$ = sampled quantization value

$x(n)$ = sampled unquantized value

Depending on the way in which $x(n)$ is quantized different distributions of quantization noise may be obtained. If rounding is used, the error signal satisfies the relation

$$\frac{-2^{-B}}{2} \le e(n) \le \frac{2^{-B}}{2}$$

The other type of quantization can be obtained by truncation. In truncation the signal is represented by the highest level that is not greater than the signal. In two's complement truncation, the error $e(n)$ is always negative and satisfies the inequality,

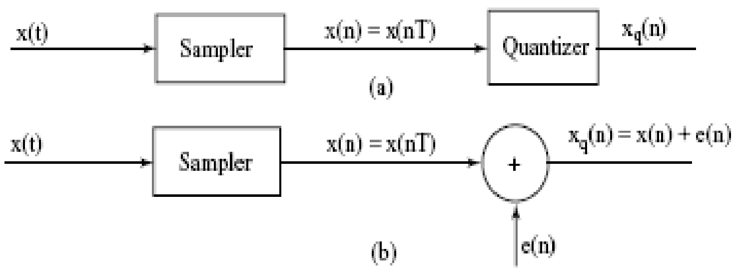
$$-2^{-B} \le e(n) < 0$$

Steady state input noise power:

The quantization error is commonly viewed as an additive noise signal, that is

$$x_q(n) = x(n) + e(n)$$

The A/D converter output is the sum of the input signal $x(n)$ and the error signal $e(n)$ as shown below.



If rounding is used for quantization error $e(n) = x_q(n) - x(n)$ is bounded by $\frac{-2^{-B}}{2} \leq e(n) \leq \frac{2^{-B}}{2}$.

The error $e(n)$ has the following properties:

1. The error sequence $e(n)$ is a sample sequence of a stationary random process.
2. The error sequence is uncorrelated with $x(n)$ and other signals in the system.
3. The error is a white noise process with uniform amplitude probability distribution over the range of quantization error.

The variance of $e(n)$ is given by

$$\sigma_e^2 = E[e^2(n)] - E^2[e(n)]$$

where $E^2[e(n)]$ is the average of $e^2(n)$ and $E[e(n)]$ is mean value of $e(n)$

For rounding

$$\sigma_e^2 = \int_{\frac{-2^{-B}}{2}}^{\frac{2^{-B}}{2}} e^2 p_e(n) de - (0)^2$$

$$\sigma_e^2 = 2^B \left[\frac{e^3}{3}\right]_{\frac{-2^{-B}}{2}}^{\frac{2^{-B}}{2}}$$

Therefore,

$$\sigma_e^2 = \frac{2^{-2B}}{12}$$

6. Derive the signal to quantization noise ratio of A/D converter. (6) [M/J – 14 R08]

The signal to quantization noise ratio of A/D converter is given by

SNR = 10 log[Px(n)/Pe(n)]

= 10 log Px(n) – 10 log Pe(n)

= 10 log Px(n) – 10 log $\frac{2^{-2B}}{12}$

= 10 log Px(n) + 10 log 2^{-2B} + 10 log 12

= 10 log Px(n) + 20B log 2+ 10.8

= 10 log Px(n) + 6.02B + 10.8

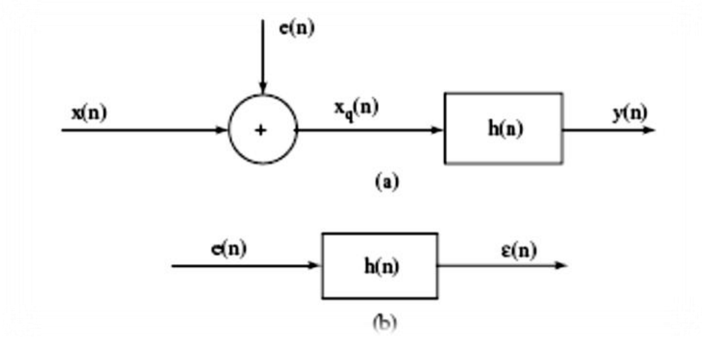
SNR = 10 log Px(n) + 6B + 10.8

For every increase in B bits the signal to quantization noise ratio increases by 6 dB.

7. How is the steady state output noise variance calculated? (8) [N/D – 10 R08]

Steady state output noise variance (or) power:

The quantized input to a digital system with impulse response $h(n)$ due to A/D conversion noise can be represented as shown in figure.



Let $\varepsilon(n)$ be the output noise due to quantization of the input. Then,

$$\varepsilon(n) = e(n) * h(n)$$

$$= \sum_{k=0}^{n} h(k)e(n-k)$$

The variance of any term in above sum in equal to $\sigma^2 h^2(n)$.

The variance of the sum of independent random variable is the sum of their variances. If the quantization errors are assumed to be independent at different sampling instances, then the variance of the output is

$$\sigma_\varepsilon^2(n)(or)\sigma_{e0}^2(n) = \sigma_e^2 \sum_{n=0}^{k} h^2(n)$$

To find the steady state variance, extend the limit k up to infinity. Then,

$$\sigma_\varepsilon^2 = \sigma_e^2 \sum_{n=0}^{\infty} h^2(n)$$

Using Parseval's theorem the steady state output noise variance due to the quantization error is given by

$$\sigma_\varepsilon^2 = \sigma_e^2 \sum_{n=0}^{\infty} h^2(n) = \sigma_e^2 \frac{1}{2\pi j} \oint_c H(z)H(z^{-1})\, z^{-1}\, dz$$

where the closed contour of integration is around the unit circle $|z| = 1$ in which case only the poles that lie inside the unit circle are evaluated using the residue theorem.

8. Determine the steady state output noise variance due to quantization of input, for the first order filter $y(n) = ay(n-1) + x(n)$. (16)

Given $y(n) = ay(n-1) + x(n)$

Taking z transform on both sides

$$y(z) = az^{-1}y(z) + x(z)$$

$$H(z) = \frac{y(z)}{x(z)} = \frac{1}{1-az^{-1}} = \frac{z}{z-a}$$

$$H(z^{-1}) = \frac{z^{-1}}{z^{-1}-a}$$

$$\sigma_\epsilon^2 = \sigma_e^2 \frac{1}{2\pi j}\oint_c H(z)H(z^{-1})\, z^{-1}\, dz$$

$$\sigma_\epsilon^2 = \sigma_e^2 \frac{1}{2\pi j}\oint_c \frac{z}{z-a}\,\frac{z^{-1}}{z^{-1}-a} z^{-1}\, dz$$

$$\sigma_\epsilon^2 = \sigma_e^2 \frac{1}{2\pi j}\oint_c \frac{z^{-1}}{(z-a)(z^{-1}-a)}\, dz$$

$$\sigma_\epsilon^2 = \sigma_e^2\left[residue\ of\ \frac{z^{-1}}{(z-a)(z^{-1}-a)}\ at\ (z=a) + \ residue\ of\ \frac{z^{-1}}{(z-a)(z^{-1}-a)}\ at\ (z=\frac{1}{a})\right]$$

Assume a < 1

$$\sigma_\epsilon^2 = \sigma_e^2\left[(z-a)\frac{z^{-1}}{(z-a)(z^{-1}-a)}\bigg|_{z=a}\right]$$

$$\sigma_\epsilon^2 = \sigma_e^2\frac{a^{-1}}{a^{-1}-a} = \ \sigma_e^2\frac{1}{1-a^2}$$

9. Consider a second order IIR filter with $H(z) = \frac{1.0}{(1-0.5\,z^{-1})\,(1-0.45\,z^{-1})}$. Explain the effect of quantization on pole locations of the system when realized in direct form and in cascade form. Assume b = 3 bits. (10) [N/D – 11 R08]

Solution:

Given: $H(z) = \dfrac{1.0}{(1-0.5\,z^{-1})(1-0.45\,z^{-1})}$

Direct form:

$H(z)$ can be written as,

$$H(z) = \frac{1}{(1 - 0.95z^{-1} + 0.225z^{-2})}$$

The poles are at $p_1 = 0.5$ and $p_2 = 0.45$

Let us quantize the co-efficient by truncation.

$$(0.95)_{10} \xrightarrow{to\ binary} (0.1111)_2 \xrightarrow{truncate\ to\ 3\ bits} (0.111)_2 \xrightarrow{to\ decimal} (0.875)_{10}$$

$$(0.225)_{10} \xrightarrow{to\ binary} (0.0011)_2 \xrightarrow{truncate\ to\ 3\ bits} (0.001)_2 \xrightarrow{to\ decimal} (0.125)_{10}$$

Hence,

$$H(z) = \frac{1}{(1 - 0.875z^{-1} + 0.125z^{-2})}$$

$$H(z) = \frac{z^2}{z^2 - 0.875\ z + 0.125}$$

$$H(z) = \frac{z^2}{(z - 0.695)(z - 0.18)}$$

So, $\overline{p_1} = 0.695$ and $\overline{p_2} = 0.18$

Thus the pole locations are shifted from 0.5, 0.45 to 0.695 and 0.18 respectively.

Cascade Form:

$$H(z) = \frac{1.0}{(1 - 0.5\ \mathrm{z}^{-1})(1 - 0.45\ \mathrm{z}^{-1})}$$

$$(0.5)_{10} \xrightarrow{to\ binary} (0.1000)_2 \xrightarrow{truncate\ to\ 3\ bits} (0.100)_2 \xrightarrow{to\ decimal} (0.5)_{10}$$

$$(0.45)_{10} \xrightarrow{to\ binary} (0.0111)_2 \xrightarrow{truncate\ to\ 3\ bits} (0.011)_2 \xrightarrow{to\ decimal} (0.375)_{10}$$

Hence,

$$H(z) = \frac{1.0}{(1 - 0.5\ \mathrm{z}^{-1})(1 - 0.375\ \mathrm{z}^{-1})}$$

So, $\overline{p_1} = 0.5$ and $\overline{p_2} = 0.375$

There is no change in first pole location and the second pole location is shifted from 0.45 to 0.375.

10. Explain coefficient quantization in IIR filter. (16) [N/D – 12 R08]

The filter coefficients are computed to infinite precision in theory. But, in digital computation the filter coefficients are represented in binary and are stored in registers. If b bit register is used, the filter coefficients must be rounded or truncated to b bits, which produces an error. This error is known as coefficient quantization error.

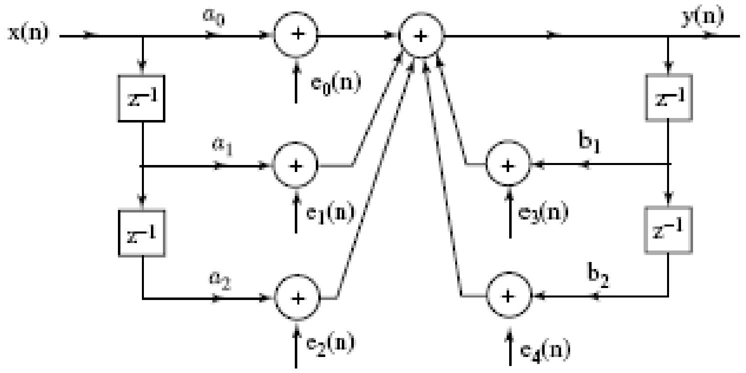
Due to quantization of coefficients, the frequency response of the filter may differ appreciably from the desired response and sometimes the filter may actually fail to meet the desired specifications. If the poles of desired filter are close to the unit circle, then those of the filter with quantized coefficients may lie just outside the unit circle, leading to instability.

Example:

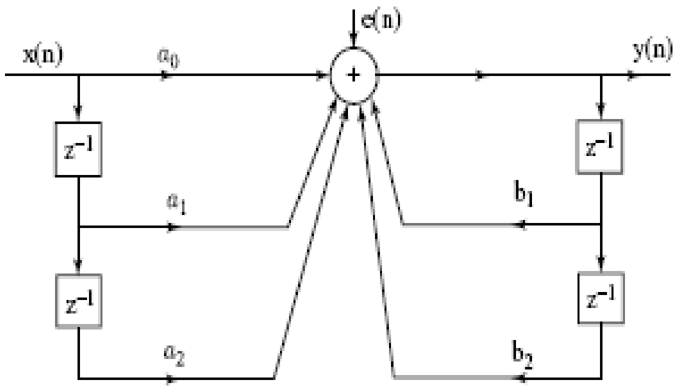
Consider a second order IIR filter with $H(z) = \frac{1.0}{(1-0.5\,z^{-1})\,(1-0.45\,z^{-1})}$. Explain the effect of quantization on pole locations of the system when realized in direct form and in cascade form. Assume b = 3 bits.

Answer: Refer problem No: 9

11. Draw the product quantization noise model of second order IIR system. (8)
[M/J – 13 R08]



Quantization noise model for a second-order system with five noise



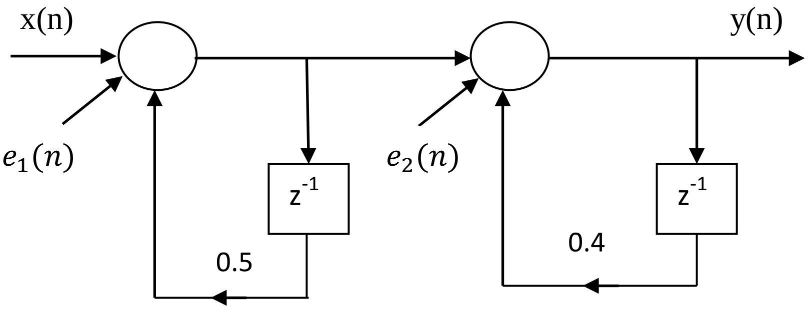
Quantization noise model for a second order system with a single noise

12. Determine the output round-off noise power for the system having transfer function H(z) = $\frac{1}{(1-0.5z^{-1})(1-0.4z^{-1})}$ which is realized in cascade form. Assume word length is 4 bits. (8)

Solution:

Given:

$$\boldsymbol{H(z) = \frac{1}{(1-0.5z^{-1})(1-0.4z^{-1})}}$$



$$\sigma_{e0}^2 = \sigma_{e01}^2 + \sigma_{e02}^2$$

$$\sigma_{e01}^2 = \sigma_e^2 \frac{1}{2\pi j}\oint_c H(z)H(z^{-1})\, z^{-1}\, dz$$

$$\sigma_{e01}^2 = \sigma_e^2 I$$

where

$$I = \frac{1}{2\pi j}\oint_c H(z)H(z^{-1})\, z^{-1}\, dz$$

$H(z)$ is the transfer function seen by the error source $e_1(n)$

$$H(z) = \frac{1}{(1-0.5z^{-1})(1-0.4z^{-1})}$$

I = sum of residues at poles within the unit circle

$$I = \frac{1}{2\pi j}\oint_c \frac{1}{(1-0.5z^{-1})(1-0.4z^{-1})}\,\frac{1}{(1-0.5z)(1-0.4z)} z^{-1}\, dz$$

$$I = \frac{1}{2\pi j}\oint_c \frac{(z)\cancel{(z)}}{(z-0.5)(z-0.4)}\,\frac{1}{(1-0.5z)(1-0.4z)} \cancel{z^{-1}}\, dz$$

$$I = \frac{1}{2\pi j}\oint_c \frac{(z)}{(z-0.5)(z-0.4)}\,\frac{1}{(1-0.5z)(1-0.4z)}\, dz$$

The poles z = 0.5 and z = 0.4 are within unit circle.

$$I = (z-0.5)\frac{(z)}{(z-0.5)(z-0.4)}\,\frac{1}{(1-0.5z)(1-0.4z)}\bigg|_{z=0.5} + (z-0.4)\frac{(z)}{(z-0.5)(z-0.4)}\,\frac{1}{(1-0.5z)(1-0.4z)}\bigg|_{z=0.4}$$

$$I = 8.33 - 5.95 = 2.38$$

$$\sigma_{e01}^2 = \sigma_e^2(2.38)$$

$$\sigma_{e01}^2 = \frac{2^{-2B}}{12}(2.38)$$

$$\sigma_{e01}^2 = \frac{2^{-2(4)}}{12}(2.38)$$

$$\sigma_{e01}^2 = 3.25X10^{-4}(2.38)$$

$$\sigma_{e01}^2 = 7.747X10^{-4}$$

$$\sigma_{e02}^2 = \sigma_e^2\frac{1}{2\pi j}\oint_c H(z)H(z^{-1})\,z^{-1}\,dz$$

$$\sigma_{e02}^2 = \sigma_e^2 I$$

where

$$I = \frac{1}{2\pi j}\oint_c H(z)H(z^{-1})\,z^{-1}\,dz$$

$H(z)$ is the transfer function seen by the error source $e_2(n)$

$$H(z) = \frac{1}{(1-0.4z^{-1})}$$

I = sum of residues at poles within the unit circle

$$I = \frac{1}{2\pi j}\oint_c \frac{1}{(1-0.4z^{-1})}\,\frac{1}{(1-0.4z)}z^{-1}\,dz$$

$$I = \frac{1}{2\pi j}\oint_c \frac{\cancel{(z)}}{(z-0.4)}\,\frac{1}{(1-0.4z)}\cancel{z^{-1}}\,dz$$

$$I = \frac{1}{2\pi j}\oint_c \frac{1}{(z-0.4)}\,\frac{1}{(1-0.4z)}\,dz$$

The pole z = 0.4 is within unit circle.

$$I = (z-0.4)\frac{1}{(z-0.4)}\,\frac{1}{(1-0.4z)}\bigg|_{z=0.4}$$

$$I = 1.1905$$

$$\sigma_{e02}^2 = \sigma_e^2(1.1905)$$

$$\sigma_{e02}^2 = 3.25X10^{-4}(1.1905)$$

$$\sigma_{e02}^2 = 3.869X10^{-4}$$

Therefore,

$$\sigma_{e0}^2 = 7.747X10^{-4} + 3.869X10^{-4} = 11.616X10^{-4}$$

13. Determine the dead band of the system $y(n) = 0.2y(n-1) + 0.5y(n-2) + x(n)$. Assume 8 bits are used for signal representation. (8) [M/J – 13 R08]

Given: $y(n) = 0.2y(n-1) + 0.5y(n-2) + x(n)$

Assume

$$x(n) = \begin{Bmatrix} 0.0125, & n = 0 \\ 0, & otherwise \end{Bmatrix}$$

$$y(-1) = y(-2) = 0$$

$$b = 8\ bits\ (excluding\ sign\ bit)$$

n	$x(n)$	$0.2y(n-1)$	$Q[0.2y(n-1)]$	$0.5y(n-2)$	$Q[0.5y(n-2)]$	$y(n)$
0	0.0125	0	0	0	0	0.0125
1	0	0.0025	0.00390625	0	0	0.00390625
2	0	0.00078125	0	0.00625	0.0078125	0.0078125
3	0	0.0015625	0	0.001953125	0.00390625	0.00390625
4	0	0.00078125	0	0.00390625	0.00390625	0.00390625

$$(0.0025)_{10} \xrightarrow{to\ binary} (0.0000000010)_2 \xrightarrow{round\ off\ to\ 8\ bits} (0.00000001)_2 \xrightarrow{to\ decimal} (0.00390625)_{10}$$

$$(0.00078125)_{10} \xrightarrow{to\ binary} (0.000000000)_2 \xrightarrow{round\ off\ to\ 8\ bits} (0.00000000)_2 \xrightarrow{to\ decimal} (0)_{10}$$

$$(0.00625)_{10} \xrightarrow{to\ binary} (0.000000011)_2 \xrightarrow{round\ off\ to\ 8\ bits} (0.00000010)_2 \xrightarrow{to\ decimal} (0.0078125)_{10}$$

$$(0.0015625)_{10} \xrightarrow{to\ binary} (0.00000000)_{10} + \xrightarrow{round\ off\ to\ 8\ bits} (0.00000000)_2 \xrightarrow{to\ decimal} (0)_{10}$$

$$0.001953125 \xrightarrow{to\ binary} (0.000000001)_{10} \xrightarrow{round\ off\ to\ 8\ bits} (0.00000001)_2 \xrightarrow{to\ decimal} (0.00390625)_{10}$$

$$(0.00390625)_{10} \xrightarrow{to\ binary} (0.000000001)_{10} + \xrightarrow{round\ off\ to\ 8\ bits} (0.00000001)_2 \xrightarrow{to\ decimal} (0.00390625)_{10}$$

$$Dead\ band = \frac{\frac{1}{2}2^{-8}}{1-0.5}$$

$$Dead\ band = 0.00390625$$

14. Describe the quantization in floating point realization of IIR digital filters.

(16) [N/D –13 R08]

Refer Page No: 7.37 in Digital Signal Processing by Ramesh Babu

15. Explain the finite word length effects in FIR digital filters. (8)[N/D – 13 R08]

Explain the round off noise in direct form realization of a linear phase FIR filter with relevant diagrams. (8) [A/M – 11 R08]

Explain the effects of coefficient quantization in FIR filters.

(8)[M/J – 14 R08] [A/M –11 R08]

Refer Page No: 7.41 in Digital Signal Processing by Ramesh Babu

16. Explain the following (8) [M/J – 12 R08]

- **a) Coefficient quantization error**
- **b) Product quantization error**
- **c) Signal scaling**
- **d) Truncation and Rounding**

a) Coefficient quantization error:

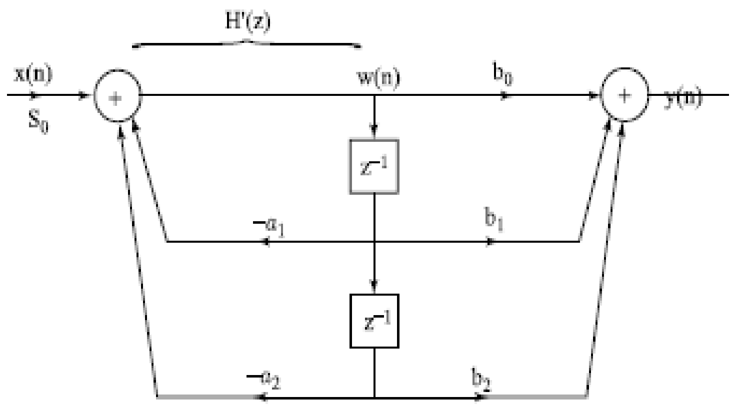
The filter coefficients are computed to infinite precision in theory. But, in digital computation the filter coefficients are represented in binary and are stored in registers. If b bit register is used, the filter coefficients must be rounded or truncated to b bits, which produces an error. This error is known as coefficient quantization error. Due to quantization of coefficients, the frequency response of the filter may differ appreciably from the desired response and sometimes the filter may actually fail to meet the desired specifications. If the poles of desired filter are close to the unit circle, then those of the filter with quantized coefficients may lie just outside the unit circle, leading to instability.

b) Product quantization error:

Product quantization errors arise at the output of a multiplier. Multiplication of b bit data with b bit coefficient results in a product having 2b bits. Since b bit register is used, the multiplier output must be rounded or truncated to b bits, which produces an error. This error is known as product quantization error.

c) Signal scaling:

Saturation arithmetic eliminates limit cycles due to overflow, but it causes undesirable signal distortion due to the nonlinearity of the clipper. In order to limit the amount of non-linear distortion, it is important to scale the input signal (scaling factor S_0) and the unit sample response between the input and any internal summing nodes in the system such that overflow become a rare event.



d) Truncation and Rounding:

Truncation:

Truncation is process of reducing the size of binary number by discarding all bits less significant than the least significant bit that is retained. In the truncation of a binary number to b bits all the less significant bits beyond b^{th} bit are discarded.

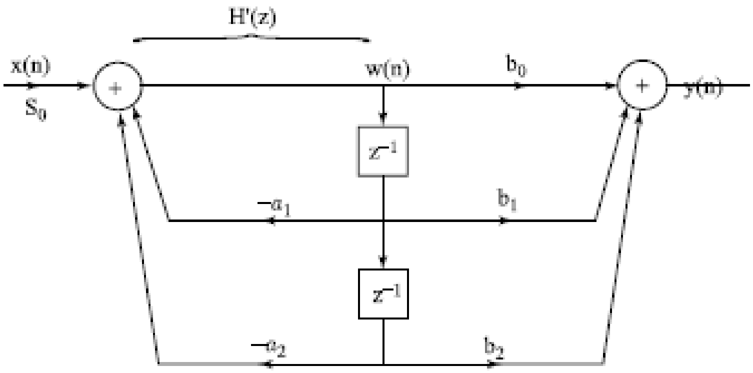
Rounding:

Rounding is the process of reducing the size of a binary number to finite word size of b bits such that, the rounded b bit number is closest to the original unquantized number.

17. a) How is signal scaling used to prevent overflow limit cycle in the digital filter implementation? Explain with an example. (8) (N/D 11 R08) (M/J 13 R08)

Signal Scaling:

Saturation arithmetic eliminates limit cycles due to overflow, but it causes undesirable signal distortion due to the nonlinearity of the clipper. In order to limit the amount of non-linear distortion, it is important to scale the input signal and the unit sample response between the input and any internal summing nodes in the system such that overflow become a rare event.



Consider a second order IIR filter shown in figure. A scale factor S_0 is introduced between the input $x(n)$ and the adder 1, to prevent overflow at the output adder 1.

Now the overall input-output transfer function is

$$H(z) = S_0 \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

$$H(z) = S_0 \frac{N(z)}{D(z)}$$

From the figure,

$$H'(z) = \frac{W(z)}{X(z)} = \frac{S_0}{1 + a_1 z^{-1} + a_2 z^{-2}} = \frac{S_0}{D(z)}$$

If the instantaneous energy in the output sequence $w(n)$ is less than the finite energy in the input sequence then, there will not be any overflow.

$$w(z) = \frac{S_0 X(z)}{D(z)} = S_0 S(z) X(z)$$

where,

$$S_0^2 = \frac{1}{\frac{1}{2\pi j}\oint_c S(z)s(z^{-1})z^{-1}dz}$$

$$S_0^2 = \frac{1}{\frac{1}{2\pi j}\oint_c \frac{z^{-1}dz}{D(z)D(z^{-1})}}$$

$$S_0^2 = \frac{1}{I}$$

where

$$I = \frac{1}{2\pi j}\oint_c \frac{z^{-1}dz}{D(z)D(z^{-1})}$$

18. Describe the effects of quantization in IIR filter. Consider a first order filter with difference equation $y(n) = x(n) + 0.5y(n-1)$. Assume that the data register length is three bits plus a sign bit. The input x(n)=0.875δ(n). Explain the limit cycle oscillations in the above filter, if quantization is performed by means of rounding and signed magnitude representation. (16)

Solution:

Given: $y(n) = x(n) + 0.5y(n-1)$

$$x(n) = 0.875\ \delta(n)$$

$$x(n) = \begin{Bmatrix} 0.875, & n = 0 \\ 0, & otherwise \end{Bmatrix}$$

$$y(-1) = 0$$

$$b = 3\ bits\ (excluding\ sign\ bit)$$

n	$x(n)$	$0.5y(n-1)$	$Q[0.5y(n-1)]$	$y(n)$
0	0.875	0	0	0.875
1	0	0.4375	0.5	0.5
2	0	0.25	0.25	0.25
3	0	0.125	0.125	0.125
4	0	0.0625	0.125	0.125

$$(0.4375)_{10} \xrightarrow{to\ binary} (0.0111)_2 \xrightarrow{round\ off\ to\ 3\ bits} (0.100)_2 \xrightarrow{to\ decimal} (0.5)_{10}$$

$$(0.25)_{10} \xrightarrow{to\ binary} (0.0100)_2 \xrightarrow{round\ off\ to\ 3\ bits} (0.010)_2 \xrightarrow{to\ decimal} (0.25)_{10}$$

$$(0.125)_{10} \xrightarrow{to\ binary} (0.0010)_2 \xrightarrow{round\ off\ to\ 3\ bits} (0.001)_2 \xrightarrow{to\ decimal} (0.125)_{10}$$

$$(0.0625)_{10} \xrightarrow{to\ binary} (0.0001)_2 \xrightarrow{round\ off\ to\ 3\ bits} (0.001)_2 \xrightarrow{to\ decimal} (0.125)_{10}$$

$$Dead\ band = \frac{\frac{1}{2}2^{-3}}{1-0.5}$$

$$Dead\ band = 0.125$$

19. Consider all pole second order IIR digital filter described by difference equation $\boldsymbol{y(n) = -0.5y(n-1) - 0.75y(n-2) + x(n)}$. Assuming 8-bits to represent a pole, determine the dead band region governing the limit cycle. (8)

Given: $y(n) = -0.5y(n-1) + 0.75y(n-2) + x(n)$

Assume

$$x(n) = \begin{Bmatrix} 0.0125, & n = 0 \\ 0, & otherwise \end{Bmatrix}$$

$$y(-1) = y(-2) = 0$$

$$b = 8\ bits\ (excluding\ sign\ bit)$$

$\boldsymbol{n}$	$\boldsymbol{x(n)}$	$\boldsymbol{-0.5y(n-1)}$	$\boldsymbol{Q[-0.5y(n-1)]}$	$\boldsymbol{-0.75y(n-2)}$	$\boldsymbol{Q[-0.75y(n-2)]}$	$\boldsymbol{y(n)}$
0	0.0125	0	0	0	0	0.0125
1	0	0.0025	0.00390625	0	0	0.00390625
2	0	0.00078125	0	0.00625	0.0078125	0.0078125
3	0	0.0015625	0	0.001953125	0.00390625	0.00390625
4	0	0.00078125	0	0.00390625	0.00390625	0.00390625

$$(0.0025)_{10} \xrightarrow{to\ binary} (0.0000000010)_2 \xrightarrow{round\ off\ to\ 8\ bits} (0.00000001)_2 \xrightarrow{to\ decimal} (0.00390625)_{10}$$

$$(0.00078125)_{10} \xrightarrow{to\ binary} (0.000000000)_2 \xrightarrow{round\ off\ to\ 8\ bits} (0.00000000)_2 \xrightarrow{to\ decimal} (0)_{10}$$

$$(0.00625)_{10} \xrightarrow{to\ binary} (0.000000011)_2 \xrightarrow{round\ off\ to\ 8\ bits} (0.00000010)_2 \xrightarrow{to\ decimal} (0.0078125)_{10}$$

$$(0.0015625)_{10} \xrightarrow{to\ binary} (0.00000000)_{10} + \xrightarrow{round\ off\ to\ 8\ bits} (0.00000000)_2 \xrightarrow{to\ decimal} (0)_{10}$$

$$0.001953125 \xrightarrow{to\ binary} (0.000000001)_{10} \xrightarrow{round\ off\ to\ 8\ bits} (0.00000001)_2 \xrightarrow{to\ decimal} (0.00390625)_{10}$$

$$(0.00390625)_{10} \xrightarrow{to\ binary} (0.000000001)_{10} + \xrightarrow{round\ off\ to\ 8\ bits} (0.00000001)_2 \xrightarrow{to\ decimal} (0.00390625)_{10}$$

$$Dead\ band = \frac{\frac{1}{2}2^{-8}}{1 - 0.75}$$

$$Dead\ band = 0.0078125$$

EC 3492 – DIGITAL SIGNAL PROCESSING

UNIT V – DSP APPLICATIONS

1. **A signal x(n) is given by $x(n) = \{0, 1, 2, 3, 4, 5, 6, 0, 1, 2, 3, \ldots\ldots\}$ (8)**

   a) **Obtain the decimated signal with a factor of 2.**

   b) **Obtain the interpolated signal with a factor of 2.**

Solution:

Given: $x(n) = \{0, 1, 2, 3, 4, 5, 6, 0, 1, 2, 3, \ldots\ldots\}$

a) Decimation with a factor of 2:

$$y(n) = \{0, 2, 4, 6, 1, 3, \ldots\ldots\}$$

b) Interpolation with a factor of 2:

$$y(n) = \{0, 0, 1, 0, 2, 0, 3, 0, 4, 0, 5, 0, 6, 0, 0, 0, 1, 0, 2, 0, 3, 0, \ldots\ldots\}$$

2. **How does the sampling rate increase by an integer factor I and derive the input-output relationship in both time and frequency domains.**

An increase in the sampling rate by an integer factor of I can be accomplished by interpolating $I - 1$ new samples between successive values of the signal. The interpolation process can be accomplished in a variety of ways. We shall describe a process that preserves the spectral shape of the signal sequence $x(n)$.

Let $v(m)$ denote a sequence with a rate $F_y = IF_x$, which is obtained from $x(n)$ by adding $I - 1$ zeros between successive values of $x(n)$. Thus

$$v(m) = \begin{cases} x(m/I), & m = 0, \pm I, \pm 2I, \ldots \\ 0, & \text{otherwise} \end{cases} \tag{10.3.1}$$

and its sampling rate is identical to the rate of $y(m)$. This sequence has a z-transform

$$\begin{aligned} V(z) &= \sum_{m=-\infty}^{\infty} v(m) z^{-m} \\ &= \sum_{m=-\infty}^{\infty} x(m) z^{-mI} \qquad (10.3.2) \\ &= X(z^I) \end{aligned}$$

The corresponding spectrum of $v(m)$ is obtained by evaluating (10.3.2) on the unit circle. Thus

$$V(\omega_y) = X(\omega_y I) \tag{10.3.3}$$

where ω_y denotes the frequency variable relative to the new sampling rate F_y (i.e., $\omega_y = 2\pi F/F_y$). Now the relationship between sampling rates is $F_y = IF_x$ and hence, the frequency variables ω_x and ω_y are related according to the formula

$$\omega_y = \frac{\omega_x}{I} \tag{10.3.4}$$

The spectra $X(\omega_x)$ and $V(\omega_y)$ are illustrated in Fig. 10.5. We observe that the sampling rate increase, obtained by the addition of $I - 1$ zero samples between successive values of $x(n)$, results in a signal whose spectrum $V(\omega_y)$ is an I-fold periodic repetition of the input signal spectrum $X(\omega_x)$.

Since only the frequency components of $x(n)$ in the range $0 \le \omega_y \le \pi/I$ are unique, the images of $X(\omega)$ above $\omega_y = \pi/I$ should be rejected by passing the sequence $v(m)$ through a lowpass filter with frequency response $H_I(\omega_y)$ that ideally has the characteristic

$$H_I(\omega_y) = \begin{cases} C, & 0 \le |\omega_y| \le \pi/I \\ 0, & \text{otherwise} \end{cases} \tag{10.3.5}$$

where C is a scale factor required to properly normalize the output sequence $y(m)$. Consequently, the output spectrum is

$$Y(\omega_y) = \begin{cases} CX(\omega_y I), & 0 \le |\omega_y| \le \pi/I \\ 0, & \text{otherwise} \end{cases} \tag{10.3.6}$$

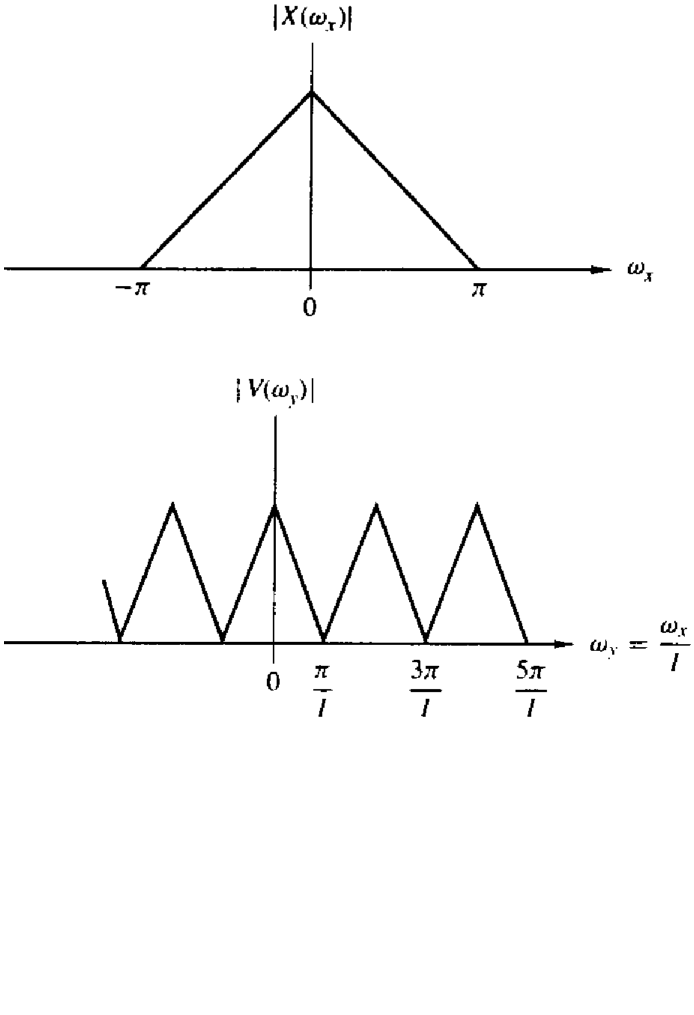


Figure 10.5 Spectra of $x(n)$ and $v(n)$ where $V(\omega_y) = X(\omega_y I)$.

The scale factor C is selected so that the output $y(m) = x(m/I)$ for $m = 0$, $\pm I$, $+2I, \ldots$. For mathematical convenience, we select the point $m = 0$. Thus

$$\begin{aligned} y(0) &= \frac{1}{2\pi}\int_{-\pi}^{\pi} Y(\omega_y)d\omega_y \\ &= \frac{C}{2\pi}\int_{-\pi/I}^{\pi/I} X(\omega_y I)d\omega_y \end{aligned} \tag{10.3.7}$$

Since $\omega_y = \omega_x/I$, (10.3.7) can be expressed as

$$\begin{aligned} y(0) &= \frac{C}{I}\frac{1}{2\pi}\int_{-\pi}^{\pi} X(\omega_x)d\omega_x \\ &= \frac{C}{I}x(0) \end{aligned} \tag{10.3.8}$$

Therefore, $C = I$ is the desired normalization factor.

Finally, we indicate that the output sequence $y(m)$ can be expressed as a convolution of the sequence $v(n)$ with the unit sample response $h(n)$ of the lowpass filter. Thus

$$y(m) = \sum_{k=-\infty}^{\infty} h(m-k)v(k) \tag{10.3.9}$$

Since $v(k) = 0$ except at multiples of I, where $v(kI) = x(k)$, (10.3.9) becomes

$$y(m) = \sum_{k=-\infty}^{\infty} h(m-kI)x(k) \tag{10.3.10}$$

3. **Explain sampling rate reduction by an integer factor 'D'. Derive the relation between input and output frequency. (8)**

DECIMATION BY A FACTOR D

Let us assume that the signal $x(n)$ with spectrum $X(\omega)$ is to be downsampled by an integer factor D. The spectrum $X(\omega)$ is assumed to be nonzero in the frequency interval $0 \le |\omega| \le \pi$ or, equivalently, $|F| \le F_x/2$. We know that if we reduce the sampling rate simply by selecting every Dth value of $x(n)$, the resulting signal will be an aliased version of $x(n)$, with a folding frequency of $F_x/2D$. To avoid aliasing, we must first reduce the bandwidth of $x(n)$ to $F_{max} = F_x/2D$ or, equivalently, to $\omega_{max} = \pi/D$. Then we may downsample by D and thus avoid aliasing.

The decimation process is illustrated in Fig. 10.2. The input sequence $x(n)$ is passed through a lowpass filter, characterized by the impulse response $h(n)$ and a frequency response $H_D(\omega)$, which ideally satisfies the condition

$$H_D(\omega) = \begin{cases} 1, & |\omega| \le \pi/D \\ 0, & \text{otherwise} \end{cases} \tag{10.2.1}$$

Thus the filter eliminates the spectrum of $X(\omega)$ in the range $\pi/D < \omega < \pi$. Of course, the implication is that only the frequency components of $x(n)$ in the range $|\omega| \le \pi/D$ are of interest in further processing of the signal.

The output of the filter is a sequence $v(n)$ given as

$$v(n) = \sum_{k=0}^{\infty} h(k)x(n-k) \tag{10.2.2}$$

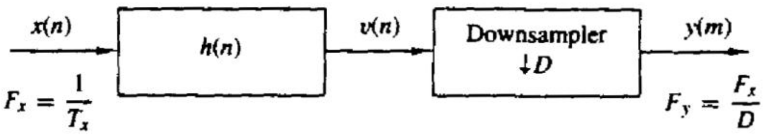


Figure 10.2 Decimation by a factor D.

which is then downsampled by the factor D to produce $y(m)$. Thus

$$\begin{aligned} y(m) &= v(mD) \\ &= \sum_{k=0}^{\infty} h(k)x(mD-k) \end{aligned} \tag{10.2.3}$$

Although the filtering operation on $x(n)$ is linear and time invariant, the downsampling operation in combination with the filtering results in a time-variant system. This is easily verified. Given the fact that $x(n)$ produces $y(m)$, we note that $x(n-n_0)$ does not imply $y(n-n_0)$ unless n_0 is a multiple of D. Consequently, the overall linear operation (linear filtering followed by downsampling) on $x(n)$ is not time invariant.

The frequency-domain characteristics of the output sequence $y(m)$ can be obtained by relating the spectrum of $y(\mathrm{m})$ to the spectrum of the input sequence $x(n)$. First, it is convenient to define a sequence $\tilde{v}(n)$ as

$$\tilde{v}(n) = \begin{cases} v(n), & n = 0, \pm D, \pm 2D, \dots \\ 0, & \text{otherwise} \end{cases} \tag{10.2.4}$$

Clearly, $\tilde{v}(n)$ can be viewed as a sequence obtained by multiplying $v(n)$ with a periodic train of impulses $p(n)$, with period D, as illustrated in Fig. 10.3. The discrete Fourier series representation of $p(n)$ is

$$p(n) = \frac{1}{D} \sum_{k=0}^{D-1} e^{j2\pi kn/D} \tag{10.2.5}$$

Hence

$$\tilde{v}(n) = v(n)p(n) \tag{10.2.6}$$

and

$$y(m) = \tilde{v}(mD) = v(mD)p(mD) = v(mD) \tag{10.2.7}$$

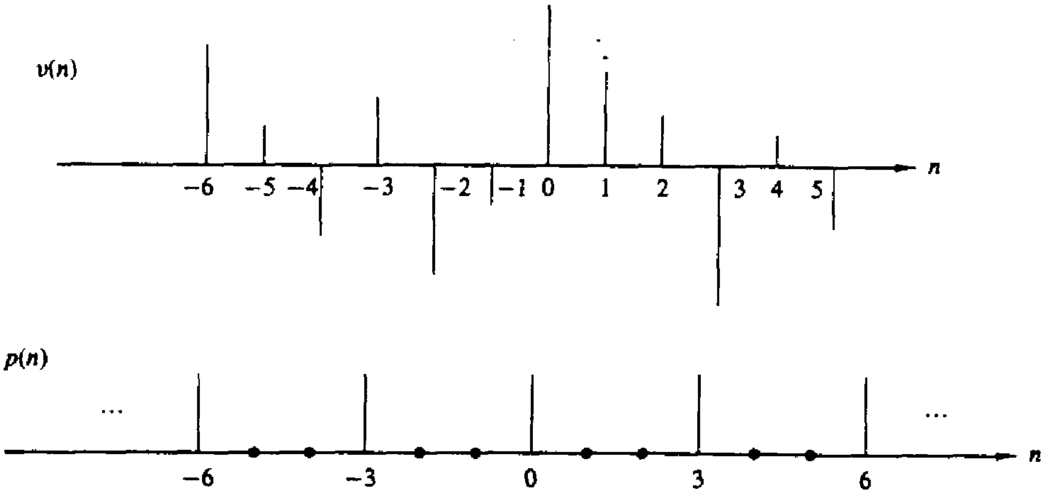


Figure 10.3 Multiplication of $v(n)$ with a periodic impulse train $p(n)$ with period $D = 3$.

Now the z-transform of the output sequence $y(m)$ is

$$\begin{aligned} Y(z) &= \sum_{m=-\infty}^{\infty} y(m) z^{-m} \\ &= \sum_{m=-\infty}^{\infty} \tilde{v}(mD) z^{-m} \end{aligned} \tag{10.2.8}$$

$$Y(z) = \sum_{m=-\infty}^{\infty} \tilde{v}(m) z^{-m/D}$$

where the last step follows from the fact that $\tilde{v}(m) = 0$, except at multiples of D. By making use of the relations in (10.2.5) and (10.2.6) in (10.2.8), we obtain

$$\begin{aligned} Y(z) &= \sum_{m=-\infty}^{\infty} v(m) \left[\frac{1}{D} \sum_{k=0}^{D-1} e^{j2\pi mk/D} \right] z^{-m/D} \\ &= \frac{1}{D} \sum_{k=0}^{D-1} \sum_{m=-\infty}^{\infty} v(m)(e^{-j2\pi k/D} z^{1/D})^{-m} \\ &= \frac{1}{D} \sum_{k=0}^{D-1} V(e^{-j2\pi k/D} z^{1/D}) \\ &= \frac{1}{D} \sum_{k=0}^{D-1} H_D(e^{-j2\pi k/D} z^{1/D}) X(e^{-j2\pi k/D} z^{1/D}) \end{aligned} \tag{10.2.9}$$

where the last step follows from the fact that $V(z) = H_D(z)X(z)$.

By evaluating $Y(z)$ in the unit circle, we obtain the spectrum of the output signal $y(m)$. Since the rate of $y(m)$ is $F_y = 1/T_y$, the frequency variable, which we denote as ω_y, is in radians and is relative to the sampling rate F_y,

$$\omega_y = \frac{2\pi F}{F_y} = 2\pi F T_y \tag{10.2.10}$$

Since the sampling rates are related by the expression

$$F_y = \frac{F_x}{D} \tag{10.2.11}$$

it follows that the frequency variables ω_y and

$$\omega_x = \frac{2\pi F}{F_x} = 2\pi F T_x \tag{10.2.12}$$

are related by

$$\omega_y = D\omega_x \tag{10.2.13}$$

Thus, as expected, the frequency range $0 \le |\omega_x| \le \pi/D$ is stretched into the corresponding frequency range $0 \le |\omega_y| \le \pi$ by the downsampling process.

We conclude that the spectrum $Y(\omega_y)$, which is obtained by evaluating (10.2.9) on the unit circle, can be expressed as

$$Y(\omega_y) = \frac{1}{D} \sum_{k=0}^{D-1} H_D\left(\frac{\omega_y - 2\pi k}{D}\right) X\left(\frac{\omega_y - 2\pi k}{D}\right) \tag{10.2.14}$$

With a properly designed filter $H_D(\omega)$, the aliasing is eliminated and, consequently, all but the first term in (10.2.14) vanish. Hence

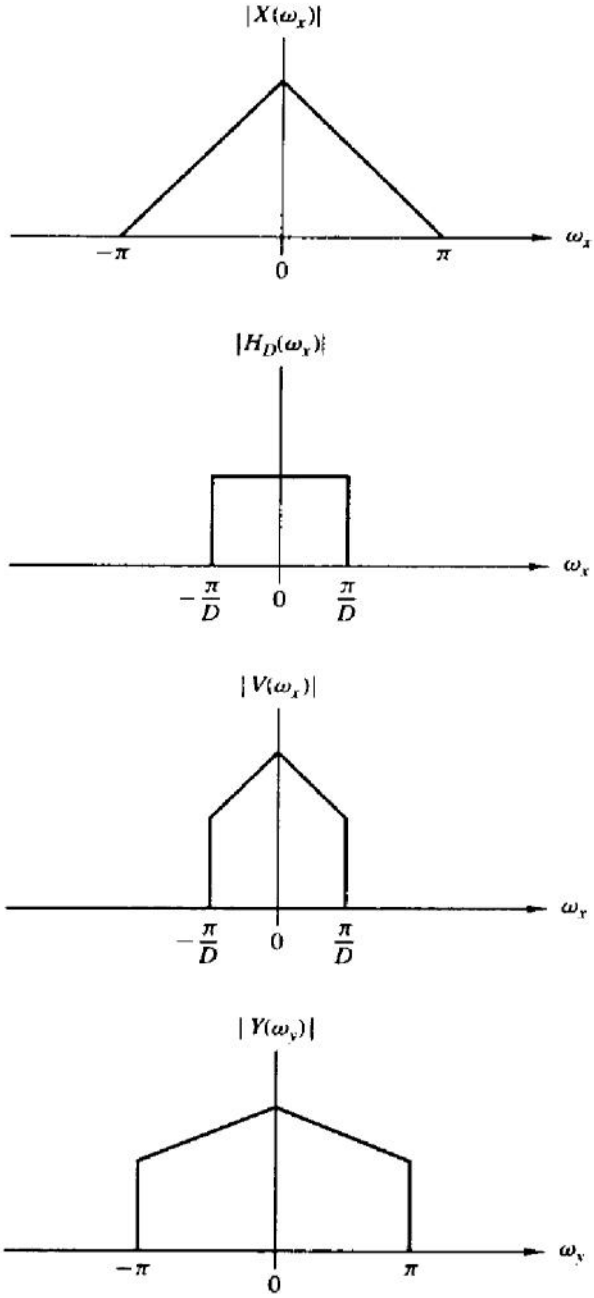


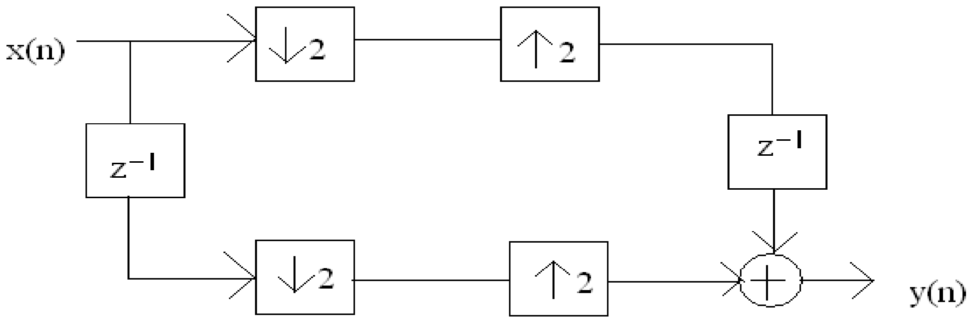
Figure 10.4 Spectra of signals in the decimation of $x(n)$ by a factor D.

$$\begin{aligned} Y(\omega_y) &= \frac{1}{D} H_D\left(\frac{\omega_y}{D}\right) X\left(\frac{\omega_y}{D}\right) \\ &= \frac{1}{D} X\left(\frac{\omega_y}{D}\right) \end{aligned} \tag{10.2.15}$$

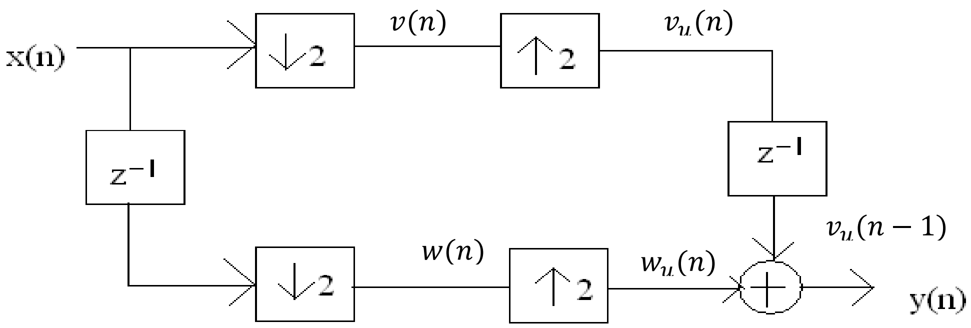
for $0 \le |\omega_y| \le \pi$. The spectra for the sequences $x(n)$, $v(n)$, and $y(m)$ are illustrated in Fig. 10.4.

4. **For the multirate system shown in figure, find the relation between x(n) and y(n).**

(N/D 11 R08)



SOLUTION:



$$v(n) = x(2n)$$

$$v(n) = \{x(0), x(\pm 2), x(\pm 4), \ldots.\}$$

$$v_u(n) = \{\ldots.x(-4), 0, x(-2), 0, x(0), 0, x(2), 0, x(4), 0, \ldots.\}$$

$$v_u(n-1) = \{\ldots.0, x(0), 0, x(2), 0, x(4), 0, \ldots\}$$

$$w(n) = x(2n-1)$$

$$w(n) = \{\ldots x(-1), x(1), x(3), x(5), \ldots\}$$

$$w_u(n) = \{\ldots\ 0, x(-1), 0, x(1), 0, x(3), 0, x(5), 0, \ldots\}$$

$$y(n) = v_u(n-1) + w_u(n)$$

$$y(n) = \{\ldots\ x(-1), x(0), x(1), x(2), x(3), \ldots\}$$

Therefore,

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

5. **Explain sampling rate conversion by a rational factor and derive input-output relation in both time and frequency domain. (10)**

SAMPLING RATE CONVERSION BY A RATIONAL FACTOR I/D

Having discussed the special cases of decimation (downsampling by a factor D) and interpolation (upsampling by a factor I), we now consider the general case of sampling rate conversion by a rational factor I/D. Basically, we can achieve this sampling rate conversion by first performing interpolation by the factor I and then decimating the output of the interpolator by the factor D. In other words, a sampling rate conversion by the rational factor I/D is accomplished by cascading an interpolator with a decimator, as illustrated in Fig. 10.6.

We emphasize that the importance of performing the interpolation first and the decimation second, is to preserve the desired spectral characteristics of $x(n)$. Furthermore, with the cascade configuration illustrated in Fig. 10.6, the two filters with impulse response $\{h_u(l)\}$ and $\{h_d(l)\}$ are operated at the same rate, namely IF_x and hence can be combined into a single lowpass filter with impulse response $h(l)$ as illustrated in Fig. 10.7. The frequency response $H(\omega_v)$ of the combined filter must incorporate the filtering operations for both interpolation and decimation, and hence it should ideally possess the frequency response characteristic

$$H(\omega_v) = \begin{cases} I, & 0 \le |\omega_v| \le \min(\pi/D, \pi/I) \\ 0, & \text{otherwise} \end{cases} \tag{10.4.1}$$

where $\omega_v = 2\pi F/F_v = 2\pi F/I F_x = \omega_x/I$.

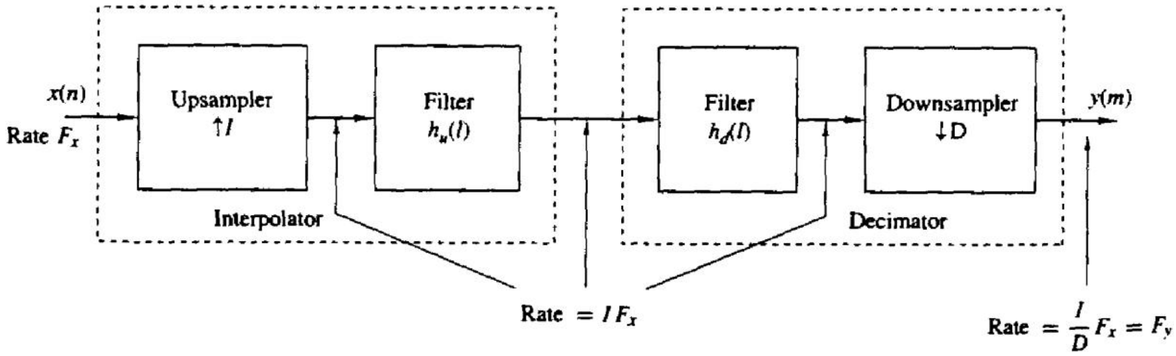


Figure 10.6 Method for sampling rate conversion by a factor I/D.

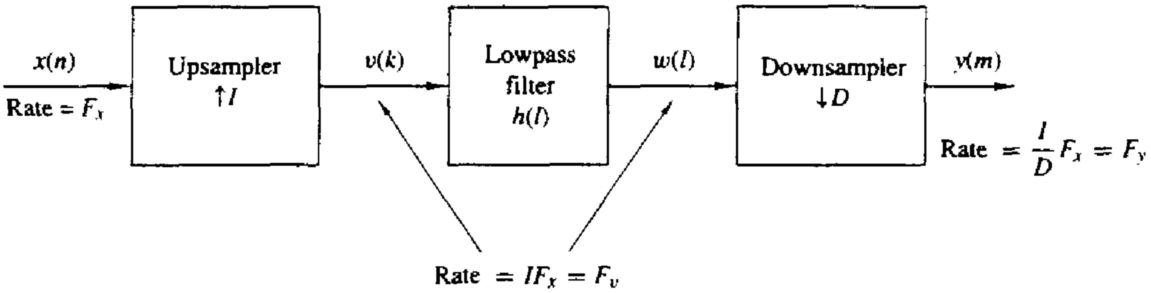


Figure 10.7 Method for sampling rate conversion by a factor I/D.

In the time domain, the output of the upsampler is the sequence

$$v(l) = \begin{cases} x(l/I), & l = 0, \pm I, \pm 2I, \ldots \\ 0, & \text{otherwise} \end{cases} \tag{10.4.2}$$

and the output of the linear time-invariant filter is

$$\begin{aligned} w(l) &= \sum_{k=-\infty}^{\infty} h(l-k)v(k) \\ &= \sum_{k=-\infty}^{\infty} h(l-kI)x(k) \end{aligned} \tag{10.4.3}$$

Finally, the output of the sampling rate converter is the sequence $\{y(m)\}$, which is obtained by downsampling the sequence $\{w(l)\}$ by a factor of D. Thus

$$\begin{aligned} y(m) &= w(mD) \\ &= \sum_{k=-\infty}^{\infty} h(mD-kI)x(k) \end{aligned} \tag{10.4.4}$$

It is illuminating to express (10.4.4) in a different form by making a change in variable. Let

$$k = \left\lfloor \frac{mD}{I} \right\rfloor - n \tag{10.4.5}$$

where the notation $\lfloor r \rfloor$ denotes the largest integer contained in r. With this change in variable, (10.4.4) becomes

$$y(m) = \sum_{n=-\infty}^{\infty} h\left(mD - \left\lfloor \frac{mD}{I} \right\rfloor I + nI\right) x\left(\left\lfloor \frac{mD}{I} \right\rfloor - n\right) \tag{10.4.6}$$

We note that

$$\begin{aligned} mD - \left\lfloor \frac{mD}{I} \right\rfloor I &= mD \quad \text{modulo } I \\ &= (mD)_I \end{aligned}$$

Consequently, (10.4.6) can be expressed as

$$y(m) = \sum_{n=-\infty}^{\infty} h(nI + (mD)_I)x\left(\left\lfloor \frac{mD}{I} \right\rfloor - n\right) \tag{10.4.7}$$

It is apparent from this form that the output $y(m)$ is obtained by passing the input sequence $x(n)$ through a time-variant filter with impulse response

$$g(n,m) = h(nI + (mD)_I) \qquad -\infty < m, n < \infty \tag{10.4.8}$$

where $h(k)$ is the impulse response of the time-invariant lowpass filter operating at the sampling rate IF_x. We further observe, that for any integer k,

$$\begin{aligned} g(n, m + kI) &= h(nI + (mD + kDI)_I) \\ &= h(nI + (mD)_I) \\ &= g(n, m) \end{aligned} \tag{10.4.9}$$

Hence $g(n, m)$ is periodic in the variable m with period I.

The frequency-domain relationships can be obtained by combining the results of the interpolation and decimation processes. Thus the spectrum at the output of the linear filter with impulse response $h(l)$ is

$$\begin{aligned} V(\omega_v) &= H(\omega_v)X(\omega_v I) \\ &= \begin{cases} IX(\omega_v I), & 0 \le |\omega_v| \le \min(\pi/D, \pi/I) \\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{10.4.10}$$

The spectrum of the output sequence $y(m)$, obtained by decimating the sequence $v(n)$ by a factor of D, is

$$Y(\omega_y) = \frac{1}{D}\sum_{k=0}^{D-1} V\left(\frac{\omega_y - 2\pi k}{D}\right) \tag{10.4.11}$$

where $\omega_y = D\omega_v$. Since the linear filter prevents aliasing as implied by (10.4.10), the spectrum of the output sequence given by (10.4.11) reduces to

$$Y(\omega_y) = \begin{cases} \dfrac{I}{D} X\left(\dfrac{\omega_y}{D}\right), & 0 \le |\omega_y| \le \min\left(\pi, \dfrac{\pi D}{I}\right) \\ 0, & \text{otherwise} \end{cases} \tag{10.4.12}$$

UNIT V DIGITAL SIGNAL PROCESSOR

INTRODUCTION

A digital signal processor (DSP) is a specialized microprocessor (or a SIP block), with architecture optimized for the operational needs of digital signal processing.

The goal of DSP is usually to measure, filter or compress continuous real-world analog signals. Most general-purpose microprocessors can also execute digital signal processing algorithms successfully, but may not be able to keep up with such processing continuously in real-time. Also, dedicated DSPs usually have better power efficiency, thus they are more suitable in portable devices such as mobile phones because of power consumption constraints. DSPs often use special memory architectures that are able to fetch multiple data or instructions at the same time.

Circular Buffering

Digital Signal Processors are designed to quickly carry out FIR filters and similar techniques. To understand the *hardware*, we must first understand the *algorithms*. In this section we will make a detailed list of the steps needed to implement an FIR filter. In the next section we will see how DSPs are designed to perform these steps as efficiently as possible.

To start, we need to distinguish between **off-line processing** and **real-time processing**. In off-line processing, the *entire* input signal resides in the computer at the same time. For example, a geophysicist might use a seismometer to record the ground movement during an earthquake. After the shaking is over, the information may be read into a computer and analyzed in some way. Another example of off-line processing is medical imaging, such as computed tomography and MRI. The data set is acquired while the patient is inside the machine, but the image reconstruction may be delayed until a later time. The key point is that *all* of the information is simultaneously available to the processing program. This is common in scientific research and engineering, but not in consumer products. Off-line processing is the realm of personal computers and mainframes.

In real-time processing, the output signal is produced at the same time that the input signal is being acquired. For example, this is needed in telephone communication, hearing aids, and radar. These applications must have the information immediately available, although it can be delayed by a short amount. For instance, a 10 millisecond delay in a telephone call cannot be detected by the speaker or listener. Likewise, it makes no difference if a radar signal is delayed by a few seconds before being displayed to the operator. Real-time applications input a sample, perform the algorithm, and output a sample, over-and-over. Alternatively, they may input a group of samples, perform the algorithm, and output a group of samples. This is the world of Digital Signal Processors.

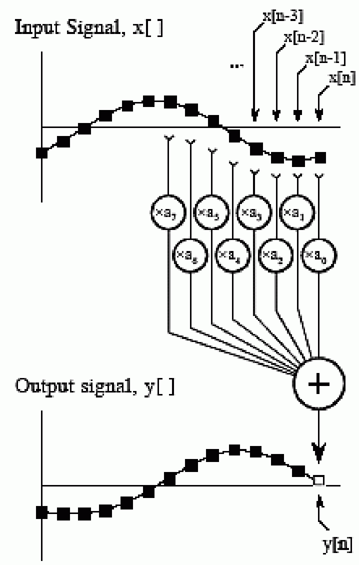


FIGURE 28-2
FIR digital filter. In FIR filtering, each sample in the output signal, y[n], is found by multiplying samples from the input signal, x[n], x[n-1], x[n-2], ..., by the filter kernel coefficients, a_0, a_1, a_2, a_3 ..., and summing the products.

Now look at Fig. 28-2 and imagine that this is an FIR filter being implemented in *real-time*. To calculate the output sample, we must have access to a certain number of the most recent samples from the input. For example, suppose we use eight coefficients in this filter, a_0, a_1, … a_7. This means we must know the value of the eight most recent samples from the input signal, $x[n]$, $x[n-1]$, … $x[n-7]$. These eight samples must be stored in memory and continually updated as new samples are acquired. What is the best way to manage these stored samples? The answer is **circular buffering**.

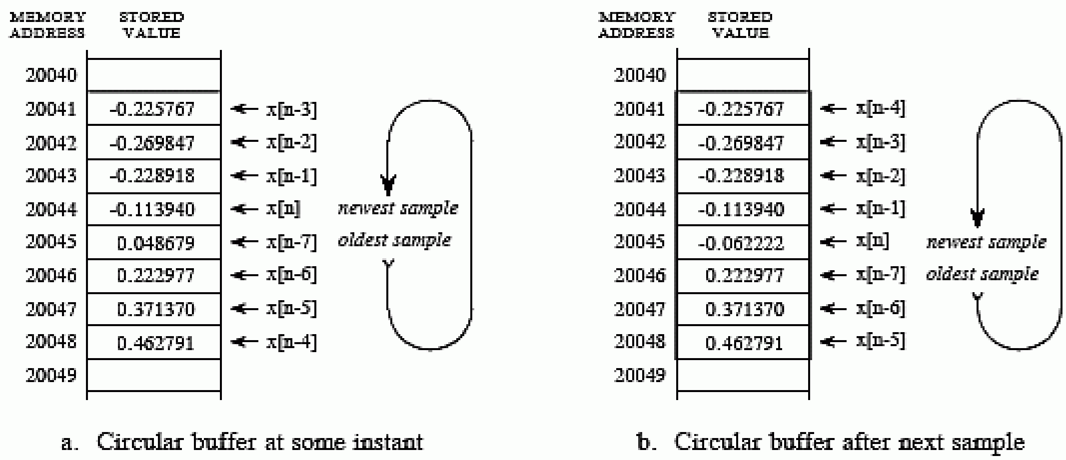


FIGURE 28-3
Circular buffer operation. Circular buffers are used to store the most recent values of a continually updated signal. This illustration shows how an eight sample circular buffer might appear at some instant in time (a), and how it would appear one sample later (b).

Figure 28-3 illustrates an eight sample circular buffer. We have placed this circular buffer in eight consecutive memory locations, 20041 to 20048. Figure (a) shows how the eight samples from the input might be stored at one particular instant in time, while (b) shows the changes after the next sample is acquired. The idea of circular buffering is that the end of this linear array is connected to its beginning; memory location 20041 is viewed as being next to 20048, just as 20044 is next to 20045. You keep track of the array by a **pointer** (a variable whose value is an *address*) that indicates where the most recent sample resides.

For instance, in (a) the pointer contains the address 20044, while in (b) it contains 20045. When a new sample is acquired, it replaces the oldest sample in the array, and the pointer is moved one address ahead. Circular buffers are efficient because only one value needs to be changed when a new sample is acquired.

Four parameters are needed to manage a circular buffer. First, there must be a pointer that indicates the start of the circular buffer in memory (in this example, 20041). Second, there must be a pointer indicating the end of the array (e.g., 20048), or a variable that holds its length (e.g., 8). Third, the step size of the memory addressing must be specified. In Fig. 28-3 the step size is *one*, for example: address 20043 contains one sample, address 20044 contains the next sample, and so on. This is frequently not the case. For instance, the addressing may refer to bytes, and each sample may require two or four bytes to hold its value. In these cases, the step size would need to be two or four, respectively.

These three values define the size and configuration of the circular buffer, and will not change during the program operation. The fourth value, the pointer to the most recent sample, must be modified as each new sample is acquired. In other words, there must be program logic that controls how this fourth value is updated based on the value of the first three values. While this logic is quite simple, it must be very fast. This is the whole point of this discussion; DSPs should be optimized at managing circular buffers to achieve the highest possible execution speed.

As an aside, circular buffering is also useful in *off-line* processing. Consider a program where both the input and the output signals are completely contained in memory. Circular buffering isn't needed for a convolution calculation, because every sample can be immediately accessed. However, many algorithms are implemented in *stages*, with an intermediate signal being created between each stage. For instance, a recursive filter carried out as a series of biquads operates in this way. The brute force method is to store the entire length of each intermediate signal in memory. Circular buffering provides another option: store only those intermediate samples needed for the calculation at hand. This reduces the required amount of memory, at the expense of a more complicated algorithm. The important idea is that circular buffers are *useful* for off-line processing, but *critical* for real-time applications.

Now we can look at the steps needed to implement an FIR filter using circular buffers for both the input signal and the coefficients. This list may seem trivial and overexamined- it's not! The efficient handling of these individual tasks is what separates a DSP from a traditional microprocessor. For each new sample, all the following steps need to be taken:

TABLE 28-1
FIR filter steps.

1. Obtain a sample with the ADC; generate an interrupt
2. Detect and manage the interrupt
3. Move the sample into the input signal's circular buffer
4. Update the pointer for the input signal's circular buffer
5. Zero the accumulator
6. Control the loop through each of the coefficients
   7. Fetch the coefficient from the coefficient's circular buffer
   8. Update the pointer for the coefficient's circular buffer
   9. Fetch the sample from the input signal's circular buffer
   10. Update the pointer for the input signal's circular buffer
   11. Multiply the coefficient by the sample
   12. Add the product to the accumulator
13. Move the output sample (accumulator) to a holding buffer
14. Move the output sample from the holding buffer to the DAC

The goal is to make these steps execute quickly. Since steps 6-12 will be repeated many times (once for each coefficient in the filter), special attention must be given to these operations. Traditional microprocessors must generally carry out these 14 steps in *serial* (one after another), while DSPs are designed to perform them in *parallel*. In some cases, all of the operations within the loop (steps 6-12) can be completed in a *single clock cycle*. Let's look at the internal architecture that allows this magnificent performance.

ARCHITECTURE OF THE DIGITAL SIGNAL PROCESSOR

One of the biggest bottlenecks in executing DSP algorithms is transferring information to and from memory. This includes *data*, such as samples from the input signal and the filter coefficients, as well as *program instructions*, the binary codes that go into the program sequencer. For example, suppose we need to multiply two numbers that reside somewhere in memory. To do this, we must fetch three binary values from memory, the numbers to be multiplied, plus the program instruction describing what to do.

Figure 28-4a shows how this seemingly simple task is done in a traditional microprocessor. This is often called a **Von Neumann architecture**, after the brilliant American mathematician John Von Neumann (1903-1957). Von Neumann guided the mathematics of many important discoveries of the early twentieth century. His many achievements include: developing the concept of a stored program computer, formalizing the mathematics of quantum mechanics, and work on the atomic bomb. If it was new and exciting, Von Neumann was there!

As shown in (a), a Von Neumann architecture contains a single memory and a single bus for transferring data into and out of the central processing unit (CPU). Multiplying two numbers requires at least three clock cycles, one to transfer each of the three numbers over the bus from the memory to the CPU. We don't count the time to transfer the result back to memory, because we assume that it remains in the CPU for additional manipulation (such as the sum of products in an FIR filter). The Von Neumann design is quite satisfactory when you are content to execute all of the required tasks in serial. In fact, most computers today are of the Von Neumann design. We only need other architectures when very fast processing is required, and we are willing to pay the price of increased complexity.

This leads us to the **Harvard architecture**, shown in (b). This is named for the work done at Harvard University in the 1940s under the leadership of Howard Aiken (1900-1973). As shown in this illustration, Aiken insisted on separate memories for data and program instructions, with separate buses for each. Since the buses operate independently, program instructions and data can be fetched at the same time, improving the speed over the single bus design. Most present day DSPs use this dual bus architecture.

Figure (c) illustrates the next level of sophistication, the **Super Harvard Architecture**. This term was coined by Analog Devices to describe the internal operation of their ADSP-2106x and new ADSP -211xx families of Digital Signal Processors. These are called **SHARC ®** DSPs, a contraction of the longer term, Super Harvard ARChitecture. The idea is to build upon the Harvard architecture by adding features to improve the throughput. While the SHARC DSPs are optimized in dozens of ways, two areas are important enough to be included in Fig. 28-4c: an *instruction cache,* and an *I/O controller*.

First, let's look at how the instruction cache improves the performance of the Harvard architecture. A handicap of the basic Harvard design is that the data memory bus is busier than the program memory bus. When two numbers are multiplied, two binary values (the numbers) must be passed over the data memory bus, while only one binary value (the program instruction) is passed over the program memory bus. To improve upon this situation, we start by relocating part of the "data" to program memory. For instance, we might place the filter coefficients in program memory, while keeping the input signal in data memory. (This relocated data is called " secondary data" in the illustration). At first glance, this doesn't seem to help the situation; now we must transfer one value over the data memory bus (the input signal sample), but two values over the program memory bus (the program instruction and the coefficient). In fact, if we were executing random instructions, this situation would be no better at all.

However, DSP algorithms generally spend most of their execution time in loops, such as instructions 6-12 of Table 28-1. This means that the same set of program instructions will continually pass from program memory to the CPU. The Super Harvard architecture takes advantage of this situation by including an **instruction cache** in the CPU. This is a small memory that contains about 32 of the most recent program instructions. The first time through a loop, the program instructions must be passed over the program memory bus. This results in slower operation because of the conflict with the coefficients that must also be fetched along this path. However, on additional executions of the loop, the program instructions can be pulled from the instruction cache. This means that all of the memory to CPU information transfers can be accomplished in a single cycle: the sample from the input signal comes over the data memory bus, the coefficient comes over the program memory bus, and the program instruction comes from the instruction cache. In the jargon of the field, this efficient transfer of data is called a *high memory-access bandwidth.*

Figure 28-5 presents a more detailed view of the SHARC architecture, showing the **I/O controller** connected to data memory. This is how the signals enter and exit the system. For instance, the SHARC DSPs provides both serial and parallel communications ports. These are extremely high speed connections. For example, at a 40 MHz clock speed, there are two serial ports that operate at 40 Mbits/second each, while six parallel ports each provide a 40 Mbytes/second data transfer. When all six parallel ports are used together, the data transfer rate is an incredible 240 Mbytes/second.

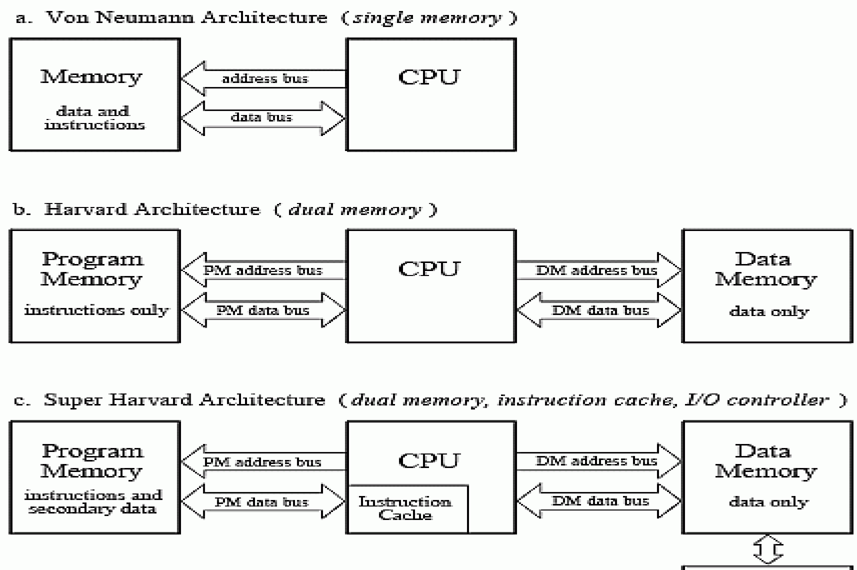


FIGURE 28-4
Microprocessor architecture. The Von Neumann architecture uses a single memory to hold both data and instructions. In comparison, the Harvard architecture uses separate memories for data and instructions, providing higher speed. The Super Harvard Architecture improves upon the Harvard design by adding an instruction cache and a dedicated I/O controller.

This is fast enough to transfer the entire text of this book in only 2 milliseconds! Just as important, dedicated hardware allows these data streams to be transferred directly into memory (Direct Memory Access, or DMA), without having to pass through the CPU's registers. In other words, tasks 1 & 14 on our list happen independently and simultaneously with the other tasks; no cycles are stolen from the CPU. The main buses (program memory bus and data memory bus) are also accessible from outside the chip, providing an additional interface to off-chip memory and peripherals. This allows the SHARC DSPs to use a four Gigaword (16 Gbyte) memory, accessible at 40 Mwords/second (160 Mbytes/second), for 32 bit data. Wow!

This type of high speed I/O is a key characteristic of DSPs. The overriding goal is to move the data in, perform the math, and move the data out before the next sample is available. Everything else is secondary. Some DSPs have on-board analog-to-digital and digital-to-analog converters, a feature called **mixed signal**. However, all DSPs can interface with external converters through serial or parallel ports.

Now let's look inside the CPU. At the top of the diagram are two blocks labeled **Data Address Generator** (DAG), one for each of the two memories. These control the addresses sent to the program and data memories, specifying where the information is to be read from or written to. In simpler microprocessors this task is handled as an inherent part of the program sequencer, and is quite transparent to the programmer. However, DSPs are designed to operate with *circular buffers*, and benefit from the extra hardware to manage them efficiently. This avoids needing to use precious CPU clock cycles to keep track of how the data are stored. For instance, in the

SHARC DSPs, each of the two DAGs can control *eight* circular buffers. This means that each DAG holds 32 variables (4 per buffer), plus the required logic.

Why so many circular buffers? Some DSP algorithms are best carried out in stages. For instance, IIR filters are more stable if implemented as a cascade of biquads (a stage containing two poles and up to two zeros). Multiple stages require multiple circular buffers for the fastest operation. The DAGs in the SHARC DSPs are also designed to efficiently carry out the *Fast Fourier transform*. In this mode, the DAGs are configured to generate **bit-reversed addresses** into the circular buffers, a necessary part of the FFT algorithm. In addition, an abundance of circular buffers greatly simplifies DSP code generation- both for the human programmer as well as high-level language compilers, such as C.

The data register section of the CPU is used in the same way as in traditional microprocessors. In the ADSP-2106x SHARC DSPs, there are 16 general purpose registers of 40 bits each. These can hold intermediate calculations, prepare data for the math processor, serve as a buffer for data transfer, hold flags for program control, and so on. If needed, these registers can also be used to control loops and counters; however, the SHARC DSPs have extra hardware registers to carry out many of these functions.

The math processing is broken into three sections, a **multiplier**, an **arithmetic logic unit (ALU)**, and a **barrel shifter**. The multiplier takes the values from two registers, multiplies them, and places the result into another register. The ALU performs addition, subtraction, absolute value, logical operations (AND, OR, XOR, NOT), conversion between fixed and floating point formats, and similar functions. Elementary binary operations are carried out by the barrel shifter, such as shifting, rotating, extracting and depositing segments, and so on. A powerful feature of the SHARC family is that the multiplier and the ALU can be accessed in parallel. In a single clock cycle, data from registers 0-7 can be passed to the multiplier, data from registers 8-15 can be passed to the ALU, and the two results returned to any of the 16 registers.

There are also many important features of the SHARC family architecture that aren't shown in this simplified illustration. For instance, an 80 bit accumulator is built into the multiplier to reduce the round-off error associated with multiple fixed-point math operations. Another interesting

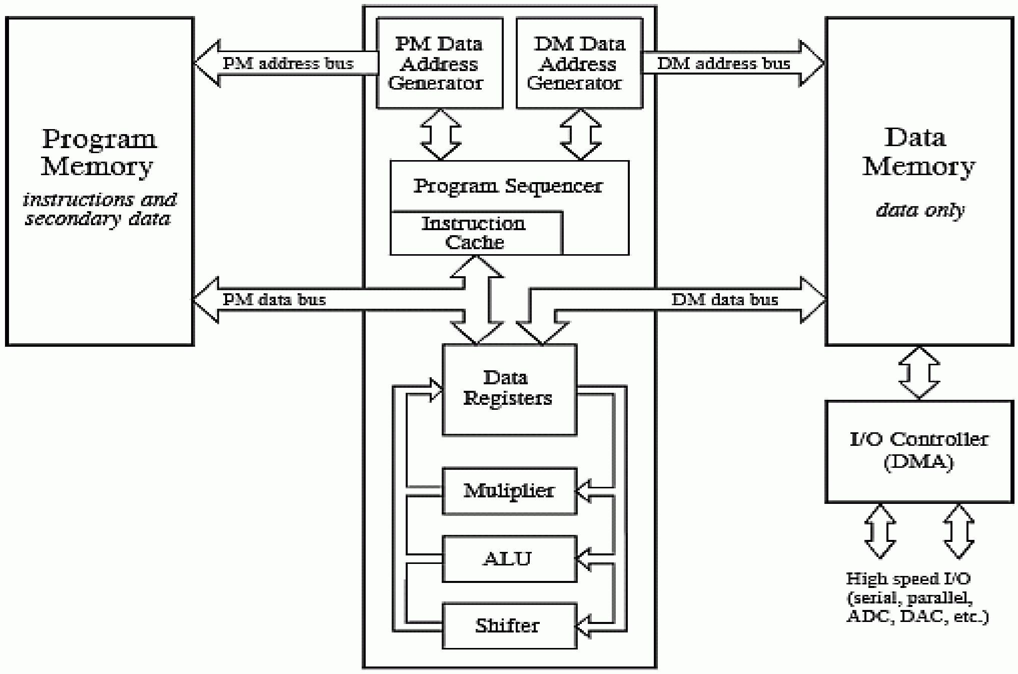


FIGURE 28-5
Typical DSP architecture. Digital Signal Processors are designed to implement tasks in parallel. This simplified diagram is of the Analog Devices SHARC DSP. Compare this architecture with the tasks needed to implement an FIR filter, as listed in Table 28-1. All of the steps within the loop can be executed in a single clock cycle.

feature is the use of **shadow registers** for all the CPU's key registers. These are duplicate registers that can be switched with their counterparts in a single clock cycle. They are used for *fast context switching*, the ability to handle interrupts quickly. When an interrupt occurs in traditional microprocessors, all the internal data must be saved before the interrupt can be handled. This usually involves pushing all of the occupied registers onto the stack, one at a time. In comparison, an interrupt in the SHARC family is handled by moving the internal data into the shadow registers in a *single clock cycle*. When the interrupt routine is completed, the registers are just as quickly restored. This feature allows step 4 on our list (managing the sample-ready interrupt) to be handled very quickly and efficiently.

Now we come to the critical performance of the architecture, how many of the operations within the loop (steps 6-12 of Table 28-1) can be carried out at the same time. Because of its highly parallel nature, the SHARC DSP can simultaneously carry out *all* of these tasks. Specifically, within a single clock cycle, it can perform a multiply (step 11), an addition (step 12), two data moves (steps 7 and 9), update two circular buffer pointers (steps 8 and 10), and control the loop (step 6). There will be extra clock cycles associated with beginning and ending the loop (steps 3, 4, 5 and 13, plus moving initial values into place); however, these tasks are also handled very efficiently. If the loop is executed more than a few times, this overhead will be negligible. As an example, suppose you write an efficient FIR filter program using 100 coefficients. You can expect it to require about 105 to 110 clock cycles per sample to execute (i.e., 100 coefficient loops plus overhead). This is very impressive; a traditional microprocessor requires many thousands of clock cycles for this algorithm.

FIXED VERSUS FLOATING POINT

Digital Signal Processing can be divided into two categories, **fixed point** and **floating point**. These refer to the format used to store and manipulate numbers within the devices.

Fixed point DSPs usually represent each number with a minimum of 16 bits, although a different length can be used. For instance, Motorola manufactures a family of fixed point DSPs that use 24 bits. There are four common ways that these $2^{16} = 65536$ possible bit patterns can represent a number.

In **unsigned integer**, the stored number can take on any integer value from 0 to 65,535. Similarly, **signed integer** uses two's complement to make the range include negative numbers, from -32,768 to 32,767. With **unsigned fraction** notation, the 65,536 levels are spread uniformly between 0 and 1. Lastly, the **signed fraction** format allows negative numbers, equally spaced between -1 and 1.

In comparison, floating point DSPs typically use a minimum of 32 bits to store each value. This results in many more bit patterns than for fixed point, $2^{32} = 4,294,967,296$ to be exact. A key feature of floating point notation is that the represented numbers are *not* uniformly spaced. In the most common format (ANSI/IEEE Std. 754-1985), the largest and smallest numbers are $\pm 3.4\times 10^{38}$ and �1.2�10^{-38}, respectively.

The represented values are unequally spaced between these two extremes, such that the gap between any two numbers is about ten-million times smaller than the value of the numbers. This is important because it places large gaps between large numbers, but small gaps between small numbers. Floating point notation is discussed in more detail in Chapter 4.

All floating point DSPs can also handle fixed point numbers, a necessity to implement counters, loops, and signals coming from the ADC and going to the DAC. However, this doesn't mean that fixed point math will be carried out as quickly as the floating point operations; it depends on the internal architecture. For instance, the SHARC DSPs are optimized for both floating point and fixed point operations, and executes them with equal efficiency. For this reason, the SHARC devices are often referred to as "32-bit DSPs," rather than just "Floating Point."

Figure 28-6 illustrates the primary trade-offs between fixed and floating point DSPs. In Chapter 3 we stressed that fixed point arithmetic is much