UNIT 1

INTRODUCTION

1.1INTRODUCTION TO DIGITAL SIGNAL PROCESSING:

SIGNAL: A signal is defined as any physical quantity that varies with time, space or another independent variable.

SYSTEM: A system is defined as a physical device that performs an operation on a signal.

SIGNAL PROCESSING: System is characterized by the type of operation that performs on the signal. Such operations are referred to as signal processing. This type of processing by Digital systems is called **DIGITAL SIGNAL PROCESSING.**

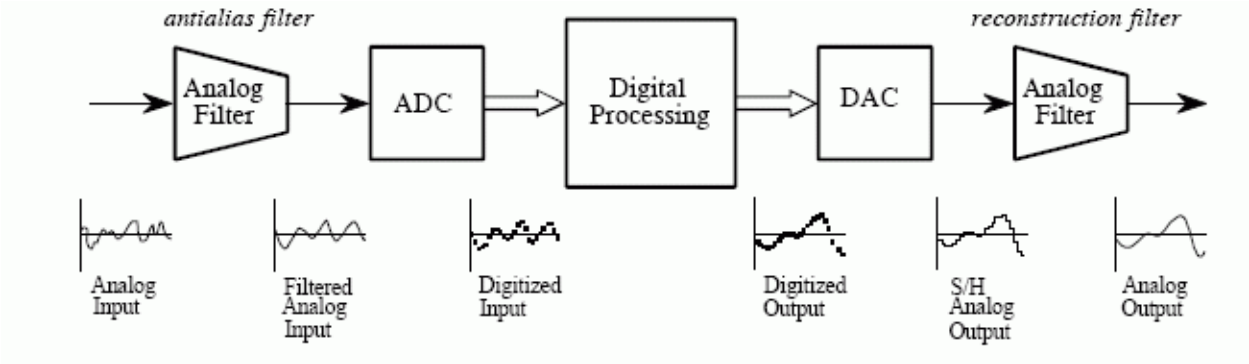


Fig: Block Diagram of DSP

Advantages of DSP

1. A digital programmable system allows flexibility in reconfiguring the digital signal processing operations by changing the program. In analog redesign of hardware is required.
2. In digital accuracy depends on word length, floating Vs fixed point arithmetic etc. In analog depends on components.
3. Can be stored on disk.
4. It is very difficult to perform precise mathematical operations on signals in analog form but these operations can be routinely implemented on a digital computer using software.
5. Cheaper to implement.
6. Small size.

7. Several filters need several boards in analog, whereas in digital same DSP processor is used for many filters.

Disadvantages of DSP

1. When analog signal is changing very fast, it is difficult to convert digital form (beyond 100KHz range)
2. w=1/2 Sampling rate.
3. Finite word length problems.
4. When the signal is weak, within a few tenths of mill volts, we cannot amplify the signal after it is digitized.
5. DSP hardware is more expensive than general purpose microprocessors & micro controllers.
6. Dedicated DSP can do better than general purpose DSP.

Applications of DSP

1. Filtering.
2. Speech synthesis in which white noise (all frequency components present to the same level) is filtered on a selective frequency basis in order to get an audio signal.
3. Speech compression and expansion for use in radio voice communication.
4. Speech recognition.
5. Signal analysis.
6. Image processing: filtering, edge effects, enhancement.
7. PCM used in telephone communication.
8. High speed MODEM data communication using pulse modulation systems such as FSK, QAM etc. MODEM transmits high speed (1200-19200 bits per second) over a band limited (3-4 KHz) analog telephone wire line.

1.2 DISCRETE TIME SIGNALS AND SEQUENCES:

DISCRETE TIME SIGNAL: A signal that has values at discrete instants of time which is obtained by sampling a continuous time signal.

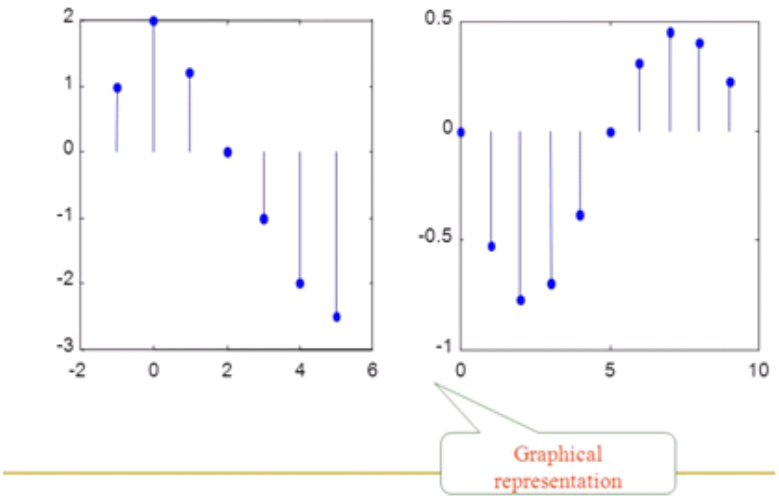
Representation of discrete time signals:

There are 4 types:

1. Graphical representation
2. Functional representation
3. Tabular representation
4. Sequence representation

Graphical representation

Consider a discrete time signal x(n) with values x(-1)=1,x(0)=2,x(1)=3,x(2)=4……..,this can be represented as shown in fig.



Functional representation: the signal is represented as

$$X(n)=\begin{pmatrix} 1\ for\ n=-1 \\ 2\ for\ n=0,1 \\ 0.5\ for\ n=2 \\ 1.5\ for\ n=3 \\ 0 \quad else \end{pmatrix}$$

Tabular representation: The signal is represented as

n	…	-2	-1	0	1	2
$x(n)$	…	0.12	2.01	1.78	5.23	0.12

Sequence representation: The signal is represented as sequence with time origin indicated by symbol ↑.

$x(n) = \{ \ldots, 0, 0, \underset{\uparrow}{1}, 4, 1, 0, 0, \ldots \} \ldots\ldots$

$x(n) = \{ \underset{\uparrow}{0}, 1, 4, 1, 0, 0, \ldots \} \ldots\ldots\ldots\ldots$

$x(n) = \{ 3, -1, -2, \underset{\uparrow}{5}, 0, 4, -1 \} \ldots\ldots\ldots\ldots$

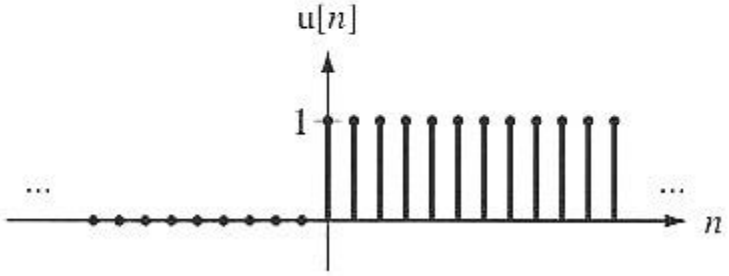
Discrete Time signals and sequences:

1.2.1 Discrete Time sequences

Unit step sequence:

It is defined as

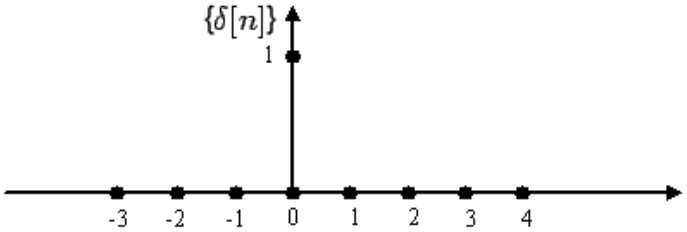
$$u[n] = \begin{cases} 1, & n \geq 0 \\ 0, & n < 0 \end{cases}$$



Unit impulse sequence :

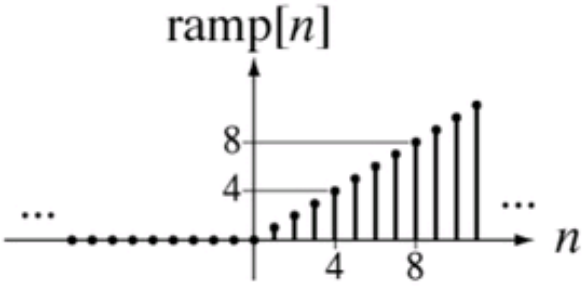
δ(n)=1 n=0

0 n≠0



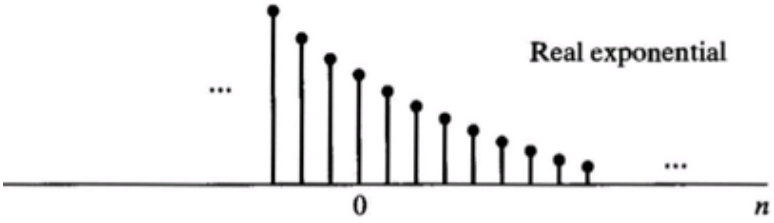
Unit Ramp sequence: It is defined as

$$\text{ramp}[n] = \begin{Bmatrix} n & , & n \geq 0 \\ 0 & , & n < 0 \end{Bmatrix} = \sum_{m=-\infty}^{n} \text{u}[m-1]$$



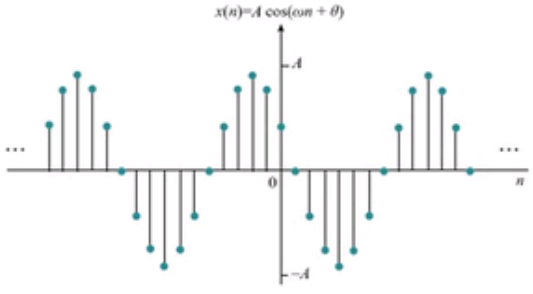
Exponential sequence: it is a sequence of the form

$x(n)=a^n$



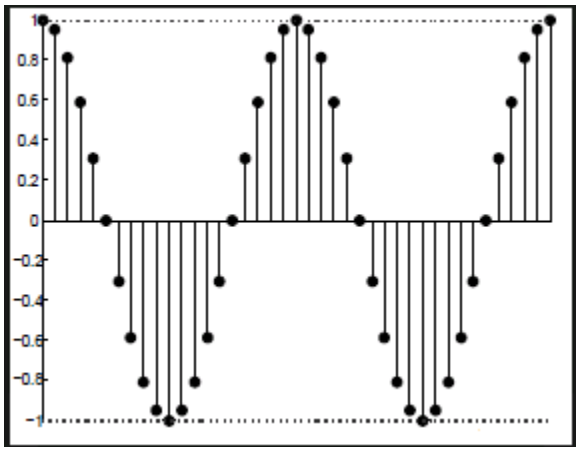
Sinusoidal signal:

It is represented as $x(n)=A\cos(w_o n+\Phi)$



Complex exponential signal: It is represented as $x(n)= \mathbf{a^n e^{j(won+\Phi)}}$

$$\mathbf{i.e\ x(n)= a^n} \cos(w_o n+\Phi) \mathbf{+ja^n} \sin(w_o n+\Phi)$$



1.2.2 Classification of Discrete time signals:

1. Energy and power signals.
2. Periodic and aperiodic signals.
3. Symmetric and asymmetric signals.
4. Causal and non causal signals.
5. Energy and power signals.

Energy and power signal: For a discrete time signal x(n) the energy E is given by

$$E_x = \sum_{n=-\infty}^{\infty} |x[n]|^2$$

The average power of a Discrete time signal x(n) is

$$P_x = \lim_{N\to\infty} \frac{1}{2N} \sum_{n=-\infty}^{\infty} |x[n]|^2.$$

A signal is energy signal iff the total energy is finite.

Periodic and aperiodic signal:

A discrete time signal x(n) is said to be periodic with period N iff

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

the smallest value of n for which the signal is periodic is called <u>fundamental period.</u>

If the above condition is valid then the signal is a periodic.

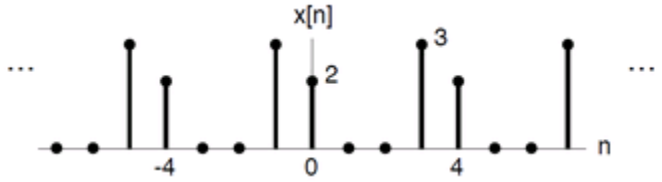


Fig: Periodic sequence

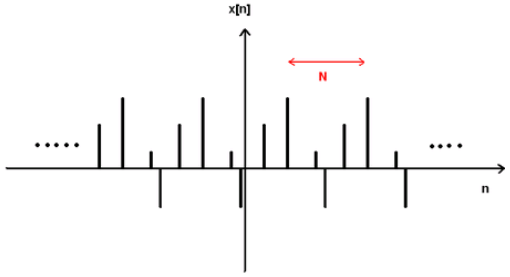


fig: aperiodic sequence

Example: Show that the exponential sequence $x(n)=e^{jw0n}$ is periodic if $w_0/2\pi$ is rational number.

Solution: Given $x(n) = e^{j\omega_0 n}$

$x(n)$ will be periodic if $x(n + N) = x(n)$

i.e. $e^{j[\omega_0(n+N)]} = e^{j\omega_0 n}$

i.e. $e^{j\omega_0 N} e^{j\omega_0 n} = e^{j\omega_0 n}$

This is possible only if $e^{j\omega_0 N} = 1$

This is true only if $\omega_0 N = 2\pi k$
where k is an integer.

$\therefore$ $\dfrac{\omega_0}{2\pi} = \dfrac{k}{N}$ Rational number

This shows that the complex exponential sequence $x(n) = e^{j\omega_0 n}$ is periodic π is a rational number.

Symmetric (even) and asymmetric (odd) signals:

A discrete time signal x(n) is said to be even it satisfies the condition

x(-n)=x(n) for all n

A discrete time signal x(n) is said to be odd it satisfies the condition

x(-n)=-x(n) for all n

this signals are represented as shown in fig.

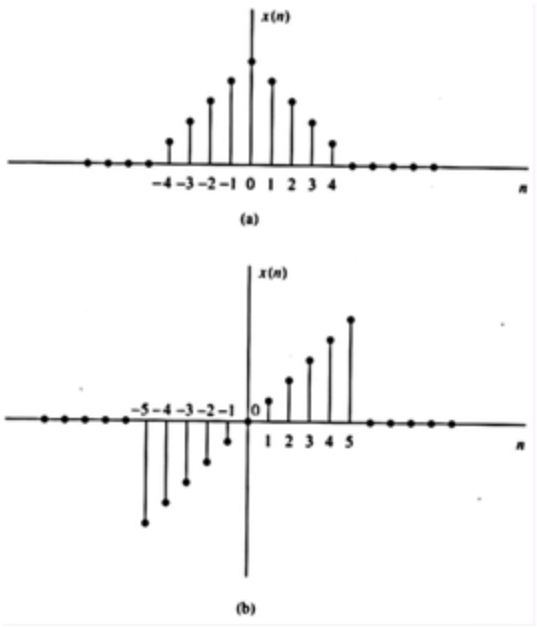
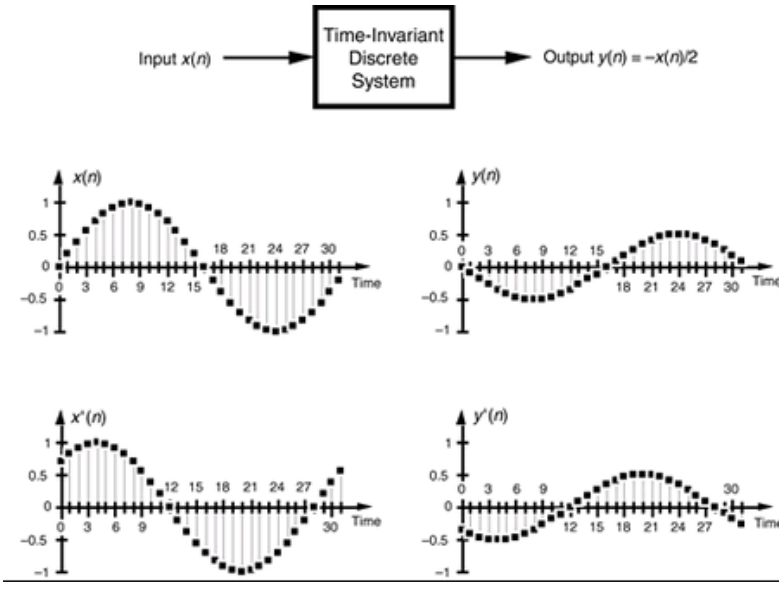


Fig a.even signal b.odd signal

Causal and non causal signals: A signal x(n) is said to be causal if its value is zero for n<0, otherwise the signal is non causal. A signal that is zero for all n≥0 is called anti causal signal.

1.3 Linear Shift Invariant Systems:

A system is said to be linear shift(time) invariant if the characteristics of the system does not change with time. In other words if the input sequence is shifted by k samples, the generated output sequence is the original sequence shifted by k samples.



To test if any given system is time invariant first apply a sequence x(n) and find y(n). Now delay the input sequence by k samples and find the output sequence.

Note: A linear time invariant system satisfies both linearity and time invariance property.

If the input to the system is unit impulse i.e x(n)=δ(n) then the output of the system is called **impulse response** denoted by h(n).

$$h(n)=T[\delta(n)]$$

for an LTI system if the input and impulse response are known then output y(n) is given by

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

the above equation represents output is the convolution sum of input sequence x(n) and impulse response h(n) represented as

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

1.4 STABILITY AND CAUSALITY

1.4.1 Stable and unstable systems

An LTI system is said to be **stable** if it produces bounded output sequence for every bounded input sequence. If the input is bounded and output is unbounded then it is unstable system. The necessary and sufficient condition for stability is

$$\sum_{n=-\infty}^{\infty} |h(n)| < \infty$$

i.e., impulse response is stable if the impulse response is summable.

1.4.2 Causal System

Generally a causal system is a system whose output depends on only past and present values of input. The output of an LTI system is given by

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

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

=…….h(-2)x(n+2)+h(-1)x(n+1)+h(0)x(n)+h(1)x(n-1)+……..

As the causal system output does not depends on future inputs so neglect the terms then y(n) reduces to

$$y(n)= h(0)x(n)+h(1)x(n-1)+\ldots\ldots\ldots\ldots$$

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

i.e. h(k)=0 for k<0.

Therefore an LTI system is causal if and only if its impulse response is zero for negative values of n.

1.5 LINEAR CONSTANT COEFFICIENT DIFFERENCE EQUATION.

There are different methods of analyzing the behavior or response of LTI system.

1. Direct solution of difference equation.
2. Discrete convolution.
3. Z transform

Direct Solution Of Difference Equation: the input and output relation of LTI system is governed by constant coefficient difference equation of form

$$y(n)= -\sum_{k=1}^{N} aky(n-k)+\sum_{k=0}^{M} bkx(n-k)$$

Mathematically the direct solution of above equation can be obtained to determine the response of the system.

Discrete convolution: The output is convolution of input and impulse response.

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

Z transform: The convolution property of z transform of the convolution of input and impulse response is equa to the product of their individual z transforms.

$$\text{i.e. } Z[x(n)*h(n)]=X(Z)H(Z)$$

$$\text{but } y(n)=x(n)*h(n)$$

$$\text{so } Z[y(n)]= X(Z)H(Z)$$

$$\text{therefore } y(n)=Z^{-1}(X(Z)H(Z))$$

i.e the response y(n) of an LTI system is obtained by taking inverse Z transform of X(Z) and H(Z). Conversely if the transfer function of the system is known then we can determine the impulse response of system by taking inverse Z transform of transfer function.

$$\text{i.e. } h(n)=Z^{-1}[H(Z)]=Z^{-1}\{Y(Z)/X(Z)\}$$

1.6 FREQUENCY DOMAIN REPRESENTATION OF DISCRETE TIME SIGNALS and SEQUENCES.

Fourier transform gives an effective representation of signals and systems in frequency domain. The Fourier transform of discrete time signal is given as

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

w is the frequency and it varies continuously from o to 2π. The magnitude of X(w) gives frequency spectrum of x(n).

$$Y(w)=X(w)H(w)$$

$$H(w)= Y(w) / X(w)$$

H(w) is system function and h(n)=IFT[H(w)]

Discrete Time Fourier series:

Consider a periodic sequence x(n) with period N and this is expressed in discrete fourier series as

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

the values of C_k k=0,1,2,3…….N-1 are called discrete spectra of x(n). Each C_k appears at frequency $w_k=2\pi k/N$.

Discrete Time Fourier Transform:

DFT is given by $\mathbf{X}(e^{jw}) =\sum_{n=-\infty}^{\infty} x(n)e^{-jwn}$

$$x[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(w)e^{jwn}dw$$

Where $\mathbf{X}(e^{jw})$ is called DTFT of x(n)

x(n) is called IDTFT of $\mathbf{X}(e^{jw})$

a sufficient condition for the existence of DTFT for a periodic sequence x(n) is

$$\sum_{n=-\infty}^{\infty} |x[n]| < \infty$$

i.e sequence is absolutely summable.

1.7 Properties of DTFT

	$x(n)$	$X(\omega)$
1. Linearity	$a_1x_1(n)+a_2x_2(n)$	$a_1x_1(\omega)+a_2x_2(\omega)$
2. Time reversal	$x(-n)$	$X(-\omega)$
3. Time shift	$x(n-n_0)$	$X(\omega)e^{-j\omega n_0}$
4. Frequency shift	$x(n)e^{j\omega_0 n}$	$X(\omega-\omega_0)$
5. Time convolution	$x_1(n)*x_2(n)$	$X_1(\omega)X_2(\omega)$
6. Frequency convolution	$x_1(n)x_2(n)$	$X_1(\omega)*X_2(\omega)$

1.8 APPLICATIONS OF Z-TRANSFORMS:

- The z-transform is a powerful mathematical tool used for the analysis of linear-time-invariant discrete systems in frequency domain.
- The z-transform has imaginary and real parts like fourier tansform .A plot of imaginary part Vs real part is called Z-plane .This is also called complex Z-plane.
- The poles and zeros of discrete LTI systems are plotted in the complex Z-plane. The stability of LTI systems can also be determined from pole-zero plot.

DEFINATION OF Z-TRANSFORM AND REGION OF CONVERGENCE:

The Z-transform of a discrete time signal x(n) is denoted by X(Z)

Z-transform:

'Z' is the complex variable

$$X(Z) = Z[x(n)]$$

$$x(n) \xleftrightarrow{Z} X(Z)$$

This Z-transform is also called as bilateral or two sided Z-transform

1.9 REGION OF CONVERGENCE:

Region of Convergence is the range of complex variable Z in the Z-plane. The Z-transformation of the signal is finite or convergent. So, ROC represents those set of values of Z, for which X(Z) has a finite value.

Z-transform is an “infinite series” (infinite power series) is not convergent for all values of Z always.

PROPERTIES OF ROC

1.ROC does not include any pole.

2.For right-sided signal,ROC will be outside the circle in Z-plane.

3.For left sided signal,ROC will be inside the circle in Z-plane.

4.For stability,ROC includes unit circle in Z-plane.

5.For Both sided signal,ROC is a ring in Z-plane.

6.For finite-duration signal,ROC is entire Z-plane.

The Z-transform is uniquely characterized by:

1.Expression of X(Z)

2.ROC of X(Z)

Sequence	Transform	ROC
$\delta[n]$	1	All z
$u[n]$	$\frac{1}{1-z^{-1}}$	$\|z\| > 1$
$-u[-n-1]$	$\frac{1}{1-z^{-1}}$	$\|z\| < 1$
$\delta[n-m]$	z^{-m}	All z except 0 or ∞
$a^n u[n]$	$\frac{1}{1-az^{-1}}$	$\|z\| > \|a\|$
$-a^n u[-n-1]$	$\frac{1}{1-az^{-1}}$	$\|z\| < \|a\|$
$na^n u[n]$	$\frac{az^{-1}}{(1-az^{-1})^2}$	$\|z\| > \|a\|$
$-na^n u[-n-1]$	$\frac{az^{-1}}{(1-az^{-1})^2}$	$\|z\| < \|a\|$
$\begin{cases} a^n & 0 \le n \le N-1, \\ 0 & \text{otherwise} \end{cases}$	$\frac{1-a^N z^{-N}}{1-az^{-1}}$	$\|z\| > 0$
$\cos(\omega_0 n)u[n]$	$\frac{1-\cos(\omega_0)z^{-1}}{1-2\cos(\omega_0)z^{-1}+z^{-2}}$	$\|z\| > 1$
$r^n \cos(\omega_0 n)u[n]$	$\frac{1-r\cos(\omega_0)z^{-1}}{1-2r\cos(\omega_0)z^{-1}+r^2 z^{-2}}$	$\|z\| > r$

Example: Determine Z-transform and ROC of the signal $x(n)=a^n u(n)+b^n(-n-1)$

Solution:

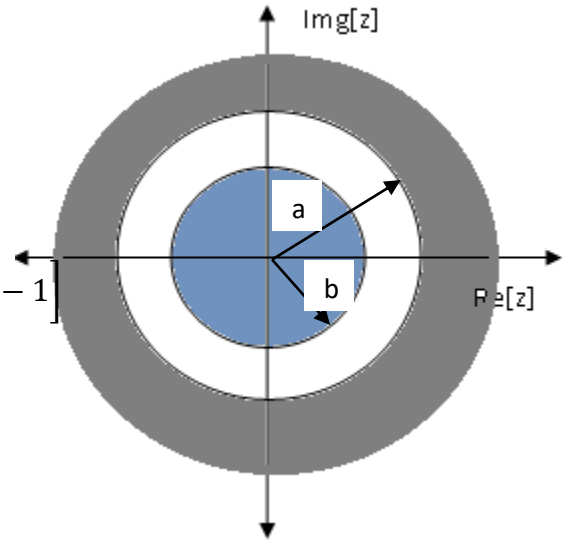
Given $x(n)=a^n u(n)+b^n(-n-1)$

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

$= \left[1 + \frac{a}{2} + \left(\frac{a}{2}\right)^2 + \cdots\right] + \sum_{n=1}^{\infty} (b^{-1}z)^n$

$= \left[1 + \frac{a}{2} + \left(\frac{a}{2}\right)^2 + \cdots\right] + [\sum_{n=0}^{\infty} (b^{-1}z)^n - 1]$

$= \left[1 + \frac{a}{2} + \left(\frac{a}{2}\right)^2 + \cdots\right] + \left[1 + \frac{z}{b} + \left(\frac{z}{b}\right)^2 + \cdots - 1\right]$



$= \frac{1}{1-\frac{a}{z}} + \frac{1}{1-\frac{z}{b}} - 1$

$= \left|\frac{a}{z}\right| < 1 + \left|\frac{z}{b}\right| < 1$

ROC: $|z| > |a|$ & $|z| < |b|$ ➔ $|b| > |z| > |a|$

$X(Z) = \frac{z}{z-a} + \frac{b}{b-z} - 1$

$= \frac{z}{z-a} + \frac{z}{b-z}$

***Example**: Determine Z-transform of the signal*

$x(n) = u(n)$

Sol. (a) $z[u(n)] = ?$

$u(n) = 1 \quad \text{for} \quad k \geq 0.$

$= 0 \quad \text{for} \quad k < 0.$

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

$$X(z) = \frac{1}{1-z^{-1}} = \frac{z}{z-1}.$$

1.10 SOLUTION OF DIFFERENCE EQUATIONS OF DIGITAL FILTERS

The N^{th} order system or digital filters are described by a general form of linear constant coefficient difference equation as

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

Where y(n) is output ,x(n) is input and

a_k , b_k are linear constant coefficients

Taking $a_0 = 1$, $y(n) = -\sum_{k=0}^{N} a_k y(n-k) + \sum_{k=0}^{N} b_k x(n-k)$

RESPONSE OF SYSTEM WITH ZERO INITIAL CONDITIONS:

System function H(Z) of system ,represent H(Z) as a ratio of two polynomials B(Z)/A(Z),

Where B(Z) is the numerator that contains zeros of H(Z) and

A(Z) is the denominator polynomial that determines poles of H(Z) input signal x(n) has a rational z-transform X(Z).

$$\text{ie,. } X(Z) = \frac{N(Z)}{Q(Z)}$$

Z-transform of the output of system has the form

$$Y(Z) = H(Z)\, X(Z) = \frac{B(Z)N(Z)}{A(Z)Q(Z)}$$

Suppose that system contains simple poles P_1, P_2, $P_{3,\ldots}$ P_N and Z-transform of the input signal contains poles q_1, q_2, $q_{3,\ldots}$ q_L, where $P_k \neq q_m$ for all k = 1,2,…N and m=1,2,…L.

The partial fraction expansions of Y(Z) yields as,

$$Y(Z) = \sum_{k=1}^{N} \frac{A_k}{1-P_k Z^{-1}} + \sum_{k=1}^{L} \frac{Q_k}{1-q_k Z^{-1}}$$

Inverse transform as $y(n) = \sum_{k=1}^{N} A_k\ (P_k)^n u(n) + \sum_{k=1}^{N} Q_k\ (q_k)^n u(n)$

P_k : Function of poles (P_k) of system is called natural response.

q_k : of the input signal is called forced response of the system.

At initial conditions are zero the response y(n) is called zero state response

$$y_{zs}(n) = y_n(n) + y_f(n)$$

Problem *Determine the system function H(z) of*

$$y(n) + \frac{3}{4}y(n-1) + \frac{1}{8}y(n-2) = x(n) + x(n-1).$$

Sol. (a) Taking z-transform of both sides,

$$Y(z) + \frac{3}{4}z^{-1}Y(z) + \frac{1}{8}z^{-2}Y(z) = X(z) + z^{-1}X(z)$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1+z^{-1}}{1+\frac{3}{4}z^{-1}+\frac{1}{8}z^{-2}}.$$

Example: Solve the following difference equation using Z-transform method

$x(n-2) - 9x(n-1) + 18x(n) = 0$

Initial conditions are x(-1)=1, x(-2)=9.

Solution : Consider the difference equation,

$$x(n-2) - 9x(n-1) + 18x(n) = 0$$

Taking unilateral z- transform of above equation,

$$\left[z^{-2}X(z) + x(-1)z^{-1} + x(-2)\right] - 9\left[z^{-1}X(z) + x(-1)\right] + 18X(z) = 0$$

$$\left[z^{-2}X(z) + z^{-1} + 9\right] - 9\left[z^{-1}X(z) + 1\right] + 18X(z) = 0$$

$$\therefore \quad X(z) = -\frac{z^{-1}}{z^{-2} - 9z^{-1} + 18} = -\frac{z}{1 - 9z + 18z^2}$$

$$\therefore \quad \frac{X(z)}{z} = -\frac{1}{18\left(z^2 - \frac{1}{2}z + \frac{1}{18}\right)} = -\frac{1}{18\left(z-\frac{1}{3}\right)\left(z-\frac{1}{6}\right)} = \frac{1/3}{z-\frac{1}{6}} - \frac{1/3}{z-\frac{1}{3}}$$

$$\therefore \quad X(z) = \frac{1/3}{1-\frac{1}{6}z^{-1}} - \frac{1/3}{1-\frac{1}{3}z^{-1}}$$

Taking inverse z-transform of above,

$$x(n) = \frac{1}{3}\left(\frac{1}{6}\right)^n u(n) - \frac{1}{3}\left(\frac{1}{3}\right)^n u(n) = \left[2\left(\frac{1}{6}\right)^{n+1} - \left(\frac{1}{3}\right)^{n+1}\right]u(n)$$

This is the solution of given difference equation.

1.11 STABILITY CRITERION:

A necessary and sufficient condition for linear time invariant system to be BIBO stable is $\sum_{n=-\infty}^{\infty}|h(n)| < \infty$

In turn, this condition implies that H(Z) most contain the unit circle within its ROC

Since H(Z) = $\sum_{n=-\infty}^{\infty} h(n)\, z^{-n}$

Take modulus on both sides $|H(Z)| \leq \sum_{n=-\infty}^{\infty}|h(n)z^{-n}|$

$= \sum_{n=-\infty}^{\infty}|h(n)|\, z^{-n}$

When its evaluated on the unit circle $|z| = 1$

$$|H(Z)| \leq \sum_{n=-\infty}^{\infty}|h(n)|$$

Hence if the system is BIBO stable, the unit circle is constrained in the ROC of H(Z). This can also be stated like "A linear time-invariant system is BIBO stable if and only if the ROC of the system function includes the unit circle.

$$|z|=b_k,\ k=1,2,\ldots M$$

Unit sample response $h(n) = \sum_{k=1}^{M} h_k(n)$

Where $h_k(n) = a_k\,(b_k)^n\, u(n)$

For the system to be stable,

Each component of the sequence $h_k(n)$ must satisfy the condition $\sum_{n=0}^{\infty}|h(n)| < \infty$

$= \sum_{n=0}^{\infty}|a_k\,(b_k)^n|$

$= a_k \sum_{n=0}^{\infty}|(b_k)^n|$

For the above system to be finite ,the magnitude of each term must be less than unity , ie,.. each $|b_k| < 1$, where b_k is a pole ie,.. $|Z| < 1$.So , all poles of the system must lie inside the unit circle, for the system to be stable.

1.12 SCHUR-COHN STABILITY TEST:

When the denominator polynomial of a transfer function of the system is large and which cannot be factorized, it is not possible to find the poles of the system .Consequently we cannot

decide whether the system is stable or not. In such cases stability can be decided by using Schur-Cohn Stability Test.

Let us consider transfer function of a system ,whose stability to be decided ,

$H(Z) = \frac{1}{1-\frac{7}{4}z^{-1}-\frac{1}{2}z^{-2}}$ consider only the denominator polynomial, here order of the denominator polynomial is 2 .So denote the polynomial as $D_2(Z) = 1 - \frac{7}{4}z^{-1} - \frac{1}{2}z^{-2}$.

Let $k_2 = -\frac{1}{2}$ and $k_2 = \left|\frac{1}{2}\right|$,If k_2 is greater than or equal to 1,system is unstable .

If k_2 is less than 1,then find k_1 by forming reverse polynomial $R_2(Z)$ from which $D_1(Z)$.

Can be found by using $D_1(Z) = \frac{D_2(Z) - k_2 R_2(Z)}{1 - k_2^2}$

Here $|k_2| < 1$ So form the reverse polynomial $R_2(Z) = \frac{-1}{2} - \frac{7}{4}z^{-1} - z^{-2}$

Therefore, $D_1(Z) = \frac{D_2(Z) - k_2 R_2(Z)}{1 - k_2^2}$

On applying in the above formulae $D_1(Z) = 1 - \frac{7}{2}z^{-1}$; $k_{1=} -\frac{7}{2}$; $|k_1| \left|\frac{7}{2}\right|$

Here $|k_1| > 1$,so the system is unstable. If $D_N(Z)$ is given from $R_N(Z)$ use recursive equation

$D_{N-1}(Z) = \frac{D_N(Z) - k_N R_N(Z)}{1 - k_N^2}$, to get $k_{N-1....,....} k_1$

If anyone of $k_{N-1,.....} k_1$ is greater than 1,stop calculating remaining K values and decide system as unstable.

Example: Find the stability of the following transfer function $H(Z) = \frac{Z^2+Z+1}{Z^4+2Z^3+3Z^2+4Z+6}$

Solution: given $H(Z) = \frac{Z^2+Z+1}{Z^4+2Z^3+3Z^2+4Z+6}$

$= \frac{Z^{-2}+Z^{-3}+Z^{-4}}{1+2Z^{-1}+3Z^{-2}+4Z^{-3}+6Z^{-4}}$

Since $k_4 = 6$; greater than 1 ; System unstable.

1.13FREQUENCY RESPONSE OF STABLE SYSTEM:

Discrete time Fourier transform and Z-transform s are used to obtain frequency response of discrete time systems. If we set $z = e^{j\omega t}$ ie,.. evaluate z-transform around unit circle ,we get the Fourier transform of the system with sampling time period,T.

$$H(e^{j\omega t}) = H(\omega) = \sum_{n=-\infty}^{\infty} h(n) e^{-jn\omega}$$

H(ω) is the frequency response of the system ,Its modulus gives the magnitude response and its phase is the phase response

Magnitude/Phase Transfer Function using Fourier Transform:

We know that the output of LTI system is given by linear convolution. i.e.,

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

Let the system be excited by the sinusoid or phasor $e^{j\omega n}$ i.e.,

$$x(n) = e^{j\omega n} \quad \text{for} \quad -\infty < n < \infty$$

Here the sinusoid is complex in nature. It has unit amplitude and frequency is 'ω'. Then output $y(n)$ becomes,

$$y(n) = \sum_{k=-\infty}^{\infty} h(k)\, e^{j\omega(n-k)} = \sum_{k=-\infty}^{\infty} h(k)\, e^{j\omega n}\, e^{-j\omega k}$$

$$= \left[\sum_{k=-\infty}^{\infty} h(k)\, e^{-j\omega k}\right] e^{j\omega n}$$

$$= H(\omega)\, e^{j\omega n} \qquad \ldots (4.3.1)$$

Here, $$H(\omega) = \sum_{k=-\infty}^{\infty} h(k)\, e^{-j\omega k} \qquad \ldots (4.3.2)$$

Thus $H(\omega)$ is the fourier transform of $h(k)$. And $h(k)$ is the unit sample response. $H(\omega)$ is called Transfer function of the system. $H(\omega)$ is complex valued function of ω in the range $-\pi \le \omega \le \pi$. The transfer function $H(\omega)$ can be expressed in polar form as,

$$H(\omega) = |H(\omega)|\, e^{j\angle H(\omega)} \qquad \ldots (4.3.3)$$

Here $|H(\omega)|$ is magnitude of $H(\omega)$

and $\angle H(\omega)$ is angle of $H(\omega)$.

By euler's identity we can write $e^{\theta} = \cos\theta + j\sin\theta$. Hence equation (4.3.2) can be written as,

$$H(\omega) = \sum_{k=-\infty}^{\infty} h(k)\left[\cos(\omega k) - j\sin(\omega k)\right]$$

$$= \sum_{k=-\infty}^{\infty} h(k)\cos(\omega k) - j\sum_{k=-\infty}^{\infty} h(k)\sin(\omega k) \qquad \ldots (4.3.4)$$

Here $H_R(\omega)$ = real part of $H(\omega) = \sum_{k=-\infty}^{\infty} h(k)\cos(\omega k)$ ... (4.3.5)

and $H_I(\omega)$ = Imaginary part of $H(\omega) = -\sum_{k=-\infty}^{\infty} h(k)\sin(\omega k)$ ... (4.3.6)

and $$|H(\omega)| = \sqrt{H_R^2(\omega) + H_I^2(\omega)} \quad \text{... (4.3.7)}$$

and $$\angle H(\omega) = \tan^{-1}\frac{H_I(\omega)}{H_R(\omega)} \quad \text{... (4.3.8)}$$

Example: Calculate the frequency response for the LTI system representation

$y(n) + \frac{1}{4}y(n-1)=x(n)-x(n-1)$

Solution: Given $y(n) + \frac{1}{4}y(n-1)=x(n)-x(n-1)$

Taking Fourier transform on both sides

$$Y(e^{j\omega}) + \frac{1}{4}e^{-j\omega}Y(e^{j\omega}) = X(e^{j\omega}) - e^{-j\omega}X(e^{j\omega})$$

$$Y(e^{j\omega})[1+\frac{1}{4}e^{-j\omega}] = X(e^{j\omega})(1-e^{-j\omega})$$

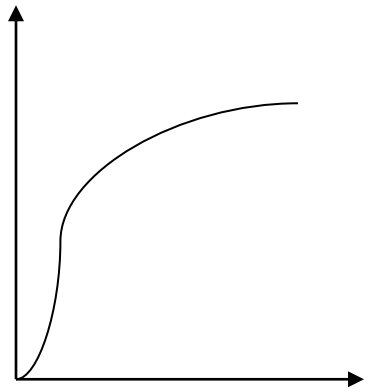
$$H(e^{j\omega}) = \frac{Y(ej\omega)}{X(ej\omega)} = \frac{1-e-j\omega}{1+\frac{1}{4}e-j\omega}$$

$$|H(e^{j\omega})| = \frac{1-\cos\omega+j\sin\omega}{1+\frac{1}{4}\cos\omega-j/4\sin\omega} = \frac{2\sin\frac{w}{2}}{(1.062+0.5cos\omega)}^{1/2}$$

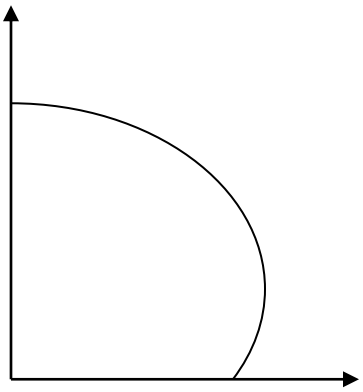
Phase Response : $\angle H(e^{j\omega}) = \tan^{-1}\left(\frac{\sin\omega}{1-\cos\omega}\right) - \tan^{-1}\left(\frac{-0.25\sin\omega}{1+0.25\cos\omega}\right)$

Ω	0	$\frac{\pi}{6}$	$\frac{\pi}{4}$	$\frac{\pi}{3}$
$\lvert H(e^{j\omega})\rvert$	0	0.4	0.6	0.8
$\angle H(e^{j\omega})$	0.5π	0.49π	0.42π	0.3π

$|H(e^{j\omega})|$



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



1.14 REALIZATION OF DIGITAL FILTERS :

A digital filter transfer function can be realized in a variety of ways .There are two types 0f realization **1. Recursive 2. Non Recursive**

For recursive realization the current output y(n) is a function of past outputs ,past and present inputs. This form corresponds to an Infinite Impulse Response (IIR) digital filter. For a Non recursive realization current output sample y(n) is a function of only past and present inputs. This form corresponds to a Finite Impulse Response (FIR) digital filter.
IIR filter can be realized in many forms. They are :

1. Direct Form-I realization
2. Direct Form-II (Canonic) realization
3. Cascade Form
4. Parallel Form.

1.Direct Form-I realization :

IIR systems can be described by a generalized equations as

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

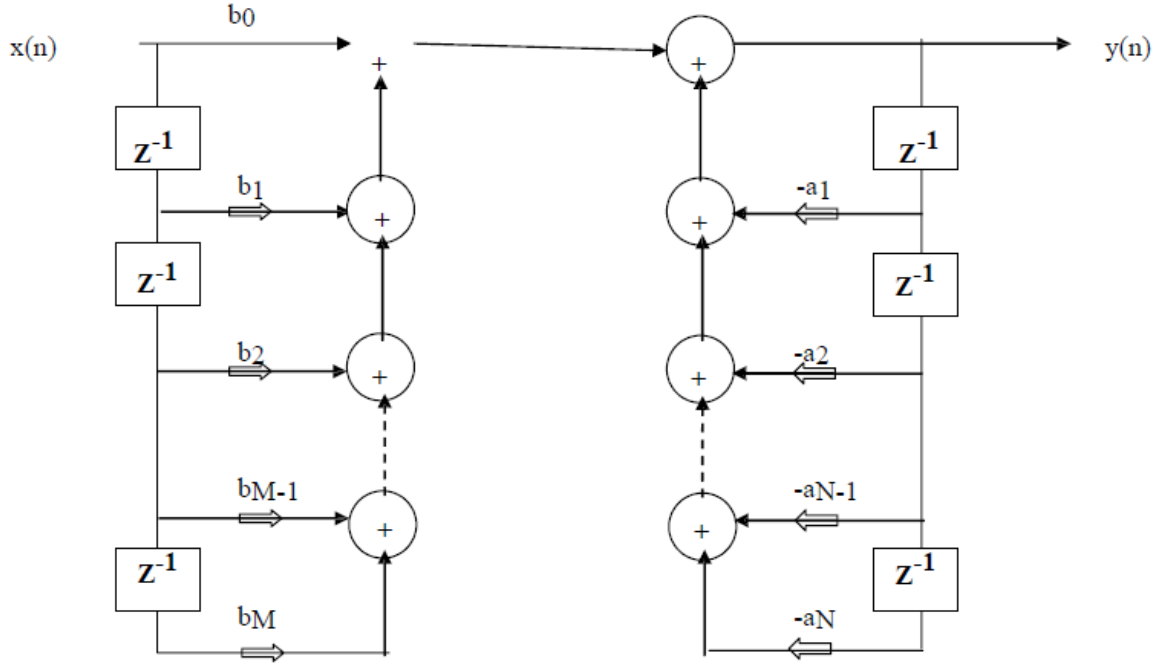
Z transform is given as

$$H(z) = \sum_{K=0}^{M} b_k\, z^{-k} \;/\; 1+\sum_{k=1}^{N} a_k\, z^{-k} \qquad (2)$$

Here $H1(z) = \sum_{K=0}^{M} b_k\, z^{-k}$ And $H2(z) = 1+\sum_{k=0}^{N} a_k\, z^{-k}$

Overall IIR system can be realized as cascade of two function H1(z) and H2(z). Here H1(z) represents zeros of H(z) and H2(z) represents all poles of H(z).

DIRECT FORM - I



1. Direct form I realization of H(z) can be obtained by cascading the realization of H1(z) which is all zero system first and then H2(z) which is all pole system.

2. There are M+N-1 unit delay blocks. One unit delay block requires one memory location. Hence direct form structure requires M+N-1 memory locations.

3. Direct Form I realization requires M+N+1 number of multiplications and M+N number of additions and M+N+1 number of memory locations.

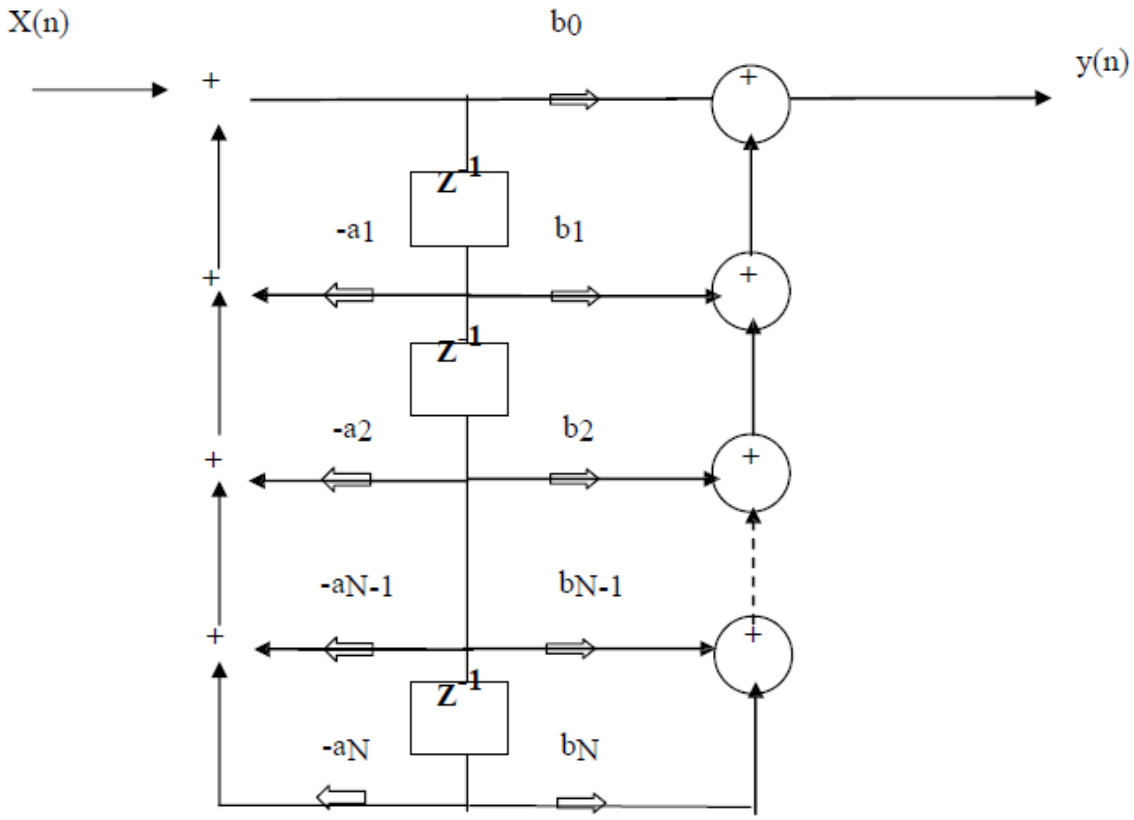
1.14.1DIRECT FORM –II REALIZATION:

1. Direct form realization of H(z) can be obtained by cascading the realization of H1(z) which is all pole system and H2(z) which is all zero system.

2. Two delay elements of all pole and all zero system can be merged into single delay element.

3. Direct Form II structure has reduced memory requirement compared to Direct form I structure. Hence it is called canonic form.

4. The direct form II requires same number of multiplications(M+N+1) and additions (M+N) as that of direct form I.



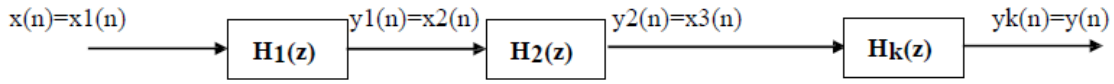
1.14.2 CASCADE FORM REALIZATION:

In cascade form, stages are cascaded (connected) in series. The output of one system is input to another. Thus total K number of stages are cascaded. The total system function 'H' is given by

$$H = H_1(z) \cdot H_2(z) \ldots\ldots\ldots\ldots\ldots\ldots H_k(z) \quad (1)$$

$$H = Y_1(z)/X_1(z) \cdot Y_2(z)/X_2(z) \cdot \ldots\ldots\ldots\ldots Y_k(z)/X_k(z) \quad (2)$$

$$H(z) = \prod_{k=1}^{k} H_k(z) \quad (3)$$



Each H1(z), H2(z)… etc is a second order section and it is realized by the direct form as shown in below figure.

System function for IIR systems

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

Expanding the above terms we have

$$H(z) = H_1(z) \cdot H_2(z) \ldots\ldots\ldots\ldots\ldots\ldots H_k(z)$$

where $H_K(z) = b_{k0} + b_{k1} z^{-1} + b_{k2} z^{-2} / 1 + a_{k1} z^{-1} + a_{k2} z^{-2}$ (2)

Thus Direct form of second order IIR system is shown as

1.14.3 PARALLEL FORM REALIZATION:

System function for IIR systems is given as

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

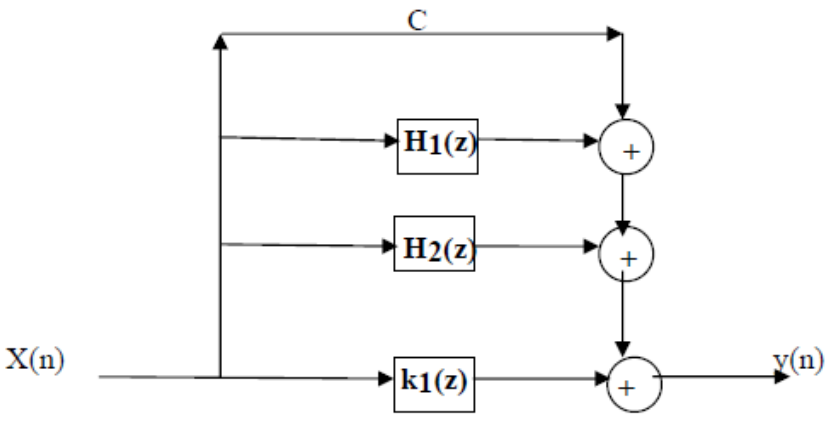
$$= b_0 + b_1 z^{-1} + b_2 z^{-2} + \ldots\ldots\ldots + b_M z^{-M} / 1 + a_1 z^{-1} + a_2 z^{-2} + \ldots\ldots + a_N z^{-N} \quad (2)$$

The above system function can be expanded in partial fraction as follows

$$H(z) = C + H_1(z) + H_2(z) \ldots\ldots\ldots\ldots\ldots\ldots + H_k(z) \quad (3)$$

Where C is constant and Hk(z) is given as

$$H_k(z) = b_{k0} + b_{k1} z^{-1} / 1 + a_{k1} z^{-1} + a_{k2} z^{-2} \quad (4)$$



Example:

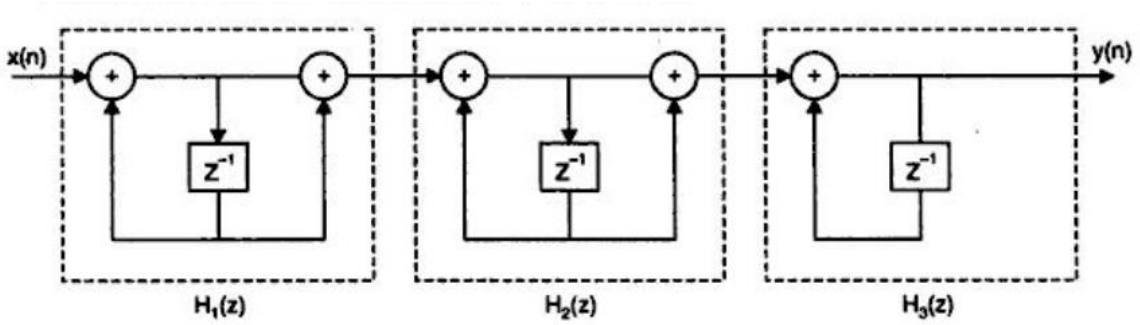
. Using first order section, obtain a cascade realization for

$$H(z) = \frac{\left(1+\frac{1}{8}z^{-1}\right)\left(1+\frac{1}{4}z^{-1}\right)}{\left(1-\frac{1}{2}z^{-1}\right)\left(1-\frac{1}{4}z^{-1}\right)\left(1-\frac{1}{10}z^{-1}\right)}.$$

Sol. The H(z) can be decomposed into the three section as $H(z) = H_1(z)\, H_2(z)\, H_3(z)$.

where, $$H_1(z) = \frac{1+\frac{1}{8}z^{-1}}{1-\frac{1}{2}z^{-1}}$$

$$H_2(z) = \frac{1+\frac{1}{4}z^{-1}}{1-\frac{1}{4}z^{-1}} \quad \text{and} \quad H_3(z) = \frac{1}{1-\frac{1}{10}z^{-1}}.$$



Example: Obtain direct form I for the system described

$$y(n) = -0.1\, y(n-1) + 0.72\, y(n-2) + 0.7x(n) - 0.252\, x(n-2).$$

Sol. We take z-transform both sides of difference eqn., we have

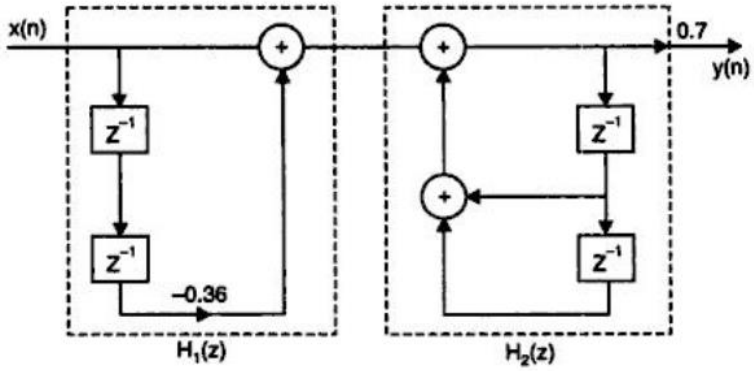
$$Y(z) = -0.1\, z^{-1}\, Y(z) + 0.72\, z^{-2}\, Y(2) + 0.7X(z) - 0.252z^{-2}X(z)$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{0.7(1-0.36z^{-2})}{1+0.1z^{-1}-0.72z^{-2}} = H_1(z)\, H_2(z)$$

$$H_1(z) = (1 - 0.36\, z^{-2}) \quad \text{and} \quad H_2(z) = \frac{0.7}{1+0.1z^{-1}-0.72z^{-2}}.$$

Direct form I

illustrates direct form I realization.

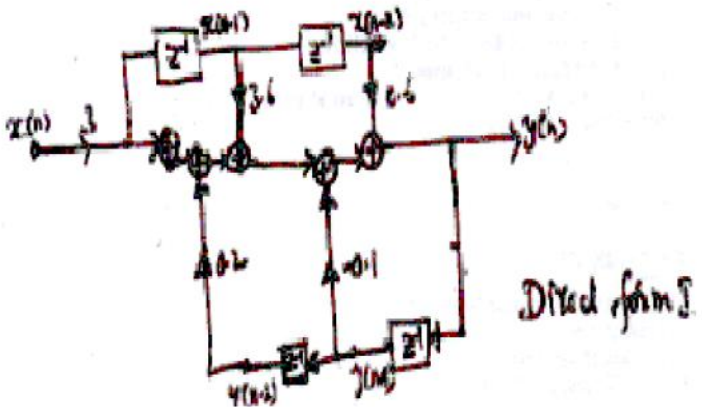


EXAMPLE:

Obtain the direct form – I, direct form-II, Cascade and parallel form realization for the following system, y(n)=-0.1 y(n-1)+0.2y(n-2)+3x(n)+3.6 x(n-1)+0.6 x(n-2) (12). (May/june-07)

Solution:

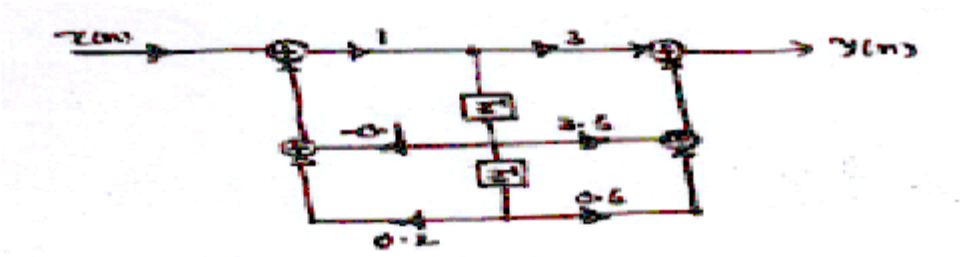
The Direct form realization is done directly from the given i/p – o/p equation, show in below diagram



Direct form –II realization

Taking ZT on both sides and finding H(z)

$$H(z)=\frac{Y(z)}{X(z)}=\frac{3+3.6z^{-1}+0.6z^{-2}}{1+0.1z^{-1}-0.2z^{-2}}$$



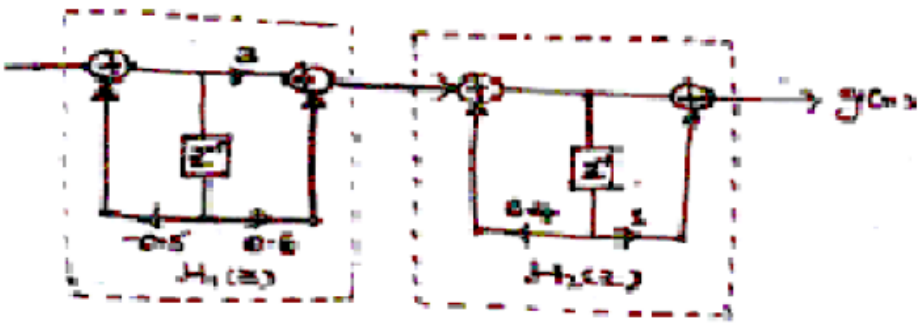
Cascade form realization

The transformer function can be expressed as:

$$H(z)=\frac{(3+0.6z^{-1})(1+z^{-1})}{(1+0.5z^{-1})(1-0.4z^{-1})}$$

which can be re written as

where $H_1(z)=\frac{3+0.6z^{-1}}{1+0.5z^{-1}}$ and $H_2(z)=\frac{1+z^{-1}}{1-0.4z^{-1}}$



Parallel Form realization

The transfer function can be expressed as

$H(z) = C + H_1(z) + H_2(z)$ where $H_1(z)$ & $H_2(z)$ is given by,

$$H(z) = -3 + \frac{7}{1-0.4z^{-1}} - \frac{1}{1+0.5z^{-1}}$$

EXAMPLE:

Convert the following pole-zero IIR filter into a lattice ladder structure,(Apr-10)

$$H(Z) = \frac{1+2Z^{-1}+2Z^{-2}+Z^{-3}}{1+\frac{13}{24}Z^{-1}+\frac{5}{8}Z^{-2}+\frac{1}{3}Z^{-3}}$$

Given $b_M(Z) = 1 + 2Z^{-1} + 2Z^{-2} + Z^{-3}$

And $A_N(Z) = 1 + \frac{13}{24}Z^{-1} + \frac{5}{8}Z^{-2} + \frac{1}{3}Z^{-3}$

$a_3(0) = 1; \quad a_3(1) = \frac{13}{24}; \quad a_3(2) = \frac{5}{8}; \quad a_3(3) = \frac{1}{3}$

$$k_3 = a_3(3) = \frac{1}{3}$$

Using the equation

$$a_{m-1}(k) = \frac{a_m(k) - a_m(m)a_m(m-k)}{1 - a^2 m(m)}$$

For m=3, k=1

$$a_2(1) = \frac{a_3(1) - a_3(3)a_3(2)}{1 - a_3^2(3)} = \frac{\frac{13}{24} - \frac{1}{3}\cdot\frac{5}{8}}{1 - \left(\frac{1}{3}\right)^2} = \frac{3}{8}$$

For m=3, & k=2

$$a_2(2) = k_2 = \frac{a_3(2) - a_3(3)a_3(1)}{1 - a_3^2(3)}$$

$$\frac{\frac{5}{8} - \frac{1}{3}\cdot\frac{13}{24}}{1 - \frac{1}{9}} = \frac{\frac{45-13}{72}}{\frac{8}{9}} = \frac{1}{2}$$

for m=2, & k=1

$$a_1(1) = k_1 = \frac{a_2(1) - a_2(2)a_2(1)}{1 - a_2^2(2)}$$

$$\frac{\frac{3}{8} - \frac{1}{2}\cdot\frac{3}{8}}{1 - \left(\frac{1}{2}\right)^2} = \frac{\frac{3}{8} - \frac{3}{16}}{1 - \frac{1}{4}} = \frac{1}{4}$$

for lattice structure $k_1 = \frac{1}{4}, \quad k_2 = \frac{1}{2}, \quad k_3 = \frac{1}{3}$

For ladder structure

$$C_m = b_m - \sum_{i=m+1}^{M} C_1.a_1(1-m) \qquad \text{m=M, M-1,1,0}$$

M=3 $\quad C_3 = b_3 = 1; \quad C_2 = b_2 - C_3 a_3(1)$

$= 2 - 1.(\frac{13}{24}) = 1.4583$

$$C_1 = b_1 - \sum_{i=2}^{3} c_1 a_1(i-m) \qquad \text{m=1}$$

$$= b_1 - \left[c_2 a_2(1) + c_3 a_{3(2)}\right]$$

$$= 2 - \left[(1.4583)(\tfrac{3}{8}) + \tfrac{5}{8}\right] = 0.8281$$

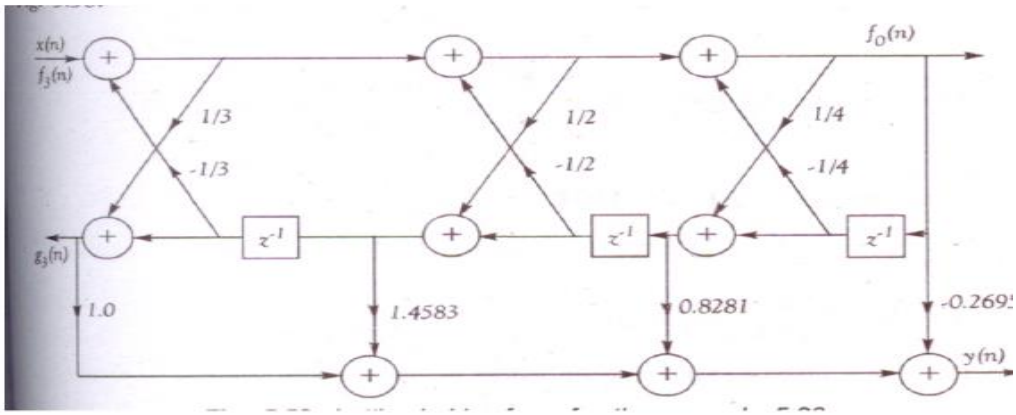
$$c_0 = b_0 - \sum_{i=1}^{3} c_1 a_1(i-m)$$

$$= b_0 - \left[c_1 a_1(1) + c_2 a_2(2) + c_3 a_3(3)\right]$$

$$= 1 - \left[08281(\tfrac{1}{4}) + 1.4583(\tfrac{1}{2}) + \tfrac{1}{3}\right] = -02695$$

To convert a lattice- ladder form into a direct form, we find an equation to obtain

$a_N(k)$ From k_m (m=1,2,……N) then equation for c_m is recursively used to compute b_m (m=0,1,2,……M).



UNIT 2

DISCRETE FOURIER SERIES

2.1 DFS REPRESENTATION OF PERIODIC SEQUENCE:

Consider a sequence $x_p(n)$ with a period of N samples so that $x_p(n)=x_p(n+lN)$.since $x_p(n)$ is a periodic ,it can be represented as a weighted sum of complex exponentials whose frequencies are integer multiples of fundamental frequency.

These periodic complex exponentials are of the form

$$e^{\frac{j2\pi}{N}kn}, \quad k = 0, \pm 1, \pm 2, \ldots$$

Any periodic sequence x(n) can be written as

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

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

2.2 PROPERTIES OF DISCRETE FOURIER SERIES:

1.LINEARITY OF DFS:

If

$$\{\tilde{x}[n]\} \leftrightarrow \{\tilde{X}[k]\}$$
$$\{\tilde{y}[n]\} \leftrightarrow \{\tilde{Y}[k]\}$$

If both the signals are periodic with same period N then

$$A\{\tilde{x}[n]\} + B\{\tilde{y}[n]\} \leftrightarrow A\{\tilde{X}[k]\} + B\{\tilde{Y}[k]\}$$

2. SHIFT OF A SEQUENCE:

$$\{\tilde{x}[n-m]\} \leftrightarrow \{e^{-j\frac{2\pi}{N}mk}\,\tilde{X}[k]\}$$
$$\{e^{j\frac{2\pi}{N}ln}\,\tilde{x}[n]\} \leftrightarrow \{\tilde{X}[k-l]\}$$

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

let $n-m=l$, we get

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

since $\tilde{x}[l]$ is periodic we can use any N consecutive values, then

$$= e^{-j\frac{2\pi}{N}km} \sum_{l=0}^{N-1} \tilde{x}[l]e^{-j\frac{2\pi}{N}kl}$$

$$= e^{-j\frac{2\pi}{N}km}\, \tilde{X}[n]$$

3.COMPLEX CONJUGATION OF A PERIODIC SEQUENCE:

$$\{\tilde{x}^*[n]\} \leftrightarrow \{\tilde{X}^*[-k]\}$$

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

$$= \tilde{X}^*[-k]$$

4.TIME REVERSAL:

$$\{\tilde{x}[-n]\} \leftrightarrow \{\tilde{X}[-k]\}$$

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

putting $m=-n$ we get

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

Since $\tilde{x}[m]$ is periodic, we can use any N consecutive values

$$= \sum_{m=0}^{N-1} \tilde{x}[m]e^{j\frac{2\pi}{N}km}$$

$$= \tilde{X}[-k]$$

5.TIME SCALING:

Let us define

$$\widetilde{x}_{(m)}[n] = \begin{cases} x[n/m], & \text{if n is multiple of m} \\ 0, & \text{if n is not a multiple of m} \end{cases}$$

sequence $\{\widetilde{x}_{(m)}[n]\}$ is obtained by inserting $(m-1)$ zeros between two consecutive values of $\widetilde{x}[n]$. Thus $\{\widetilde{x}_{(m)}[n]\}$ is also periodic, but period is mN. The DFS coefficients are given by

$$\sum_{n=0}^{mN-1} \widetilde{x}_{(m)}[n] e^{-j\frac{2\pi}{mN}kn}$$

putting $n = lm + r,\ 0 \le l \le N-1, 0 \le r < m$

$$= \sum_{l=0}^{N-1} \widetilde{x}[l] e^{-j\frac{2\pi}{N}\frac{k}{m}(lm)}$$

as non zero terms occur only when $r = 0$

$$= \widetilde{x}[h].$$

6.DIFFERENCE:

$$\{(\widetilde{x}[n] - \widetilde{x}[n-1])\} \longleftrightarrow \{(1 - e^{-j\frac{2\pi}{N}kn})\widetilde{X}[k]\}$$

This follows from linearity property.

7.ACCUMULATION:

Let us define

$$\widetilde{y}[n] = \sum_{k=-\infty}^{n} \widetilde{x}[k]$$

$\{\widetilde{y}[n]\}$ will be bounded and periodic only if the sum of terms of $\widetilde{x}[n]$ over one period is zero, i.e. $\sum_{n=0}^{N-1} \widetilde{x}[n] = 0$, which is equivalent to $\widetilde{X}[0] = 0$. Assuming this to be true

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

Example: Determine the spectra of the signals

a. $x(n) = \text{Cos } \sqrt{2}\,\pi n$

$w_o = \sqrt{2}\,\pi$

$f_o = \frac{1}{\sqrt{2}}$ is not rational number

$\therefore$ Signal is not periodic.

Its spectra content consists of the single frequency

2.3 DISCRETE FOURIER TRANSFORM:

The DFT of a finite duration sequence x(n) is obtained by sampling the fourier transform $X(e^{jw})$ at N equally spaced points over the interval $0 \le w \le 2\pi$ with a spacing of $2\pi/N$.The DFT is denoted by X(K),and is given by

$$X[k] = \begin{cases} \sum_{n=0}^{N-1} \tilde{x}[n] e^{-j\frac{2\pi}{N}kn} = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn}, & 0 \le k \le N-1 \\ 0, & \text{otherwise} \end{cases}$$

$$x[n] = \begin{cases} \frac{1}{N} \sum_{k=0}^{N-1} \tilde{X}[k] e^{j\frac{2\pi}{N}hn} = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j\frac{2\pi}{N}hn}, & 0 \le N \le N-1 \\ 0, & \text{otherwise} \end{cases}$$

For convenience

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

With this notation DFT analysis and synthesis equation is given by

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

$$\text{Synthesis equation:} \quad x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] W_N^{-kn}, \ 0 \le n \le N-1$$

2.4 SAMPLING OF THE FOURIER TRANSFORM:

The DFT values X(K) can be considered as samples of $X(e^{jw})$.

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

$(\text{as } x[n] = 0 \text{ for } n < 0, \text{ and } n > N-1)$

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

Thus is $X[k]$ is obtained by sampling $X(e^{jw})$ at $w = \frac{2\pi}{N}k,\ k = 0, 1..N-1.$

2.5 PROPERTIES OF DFT:

1.LINEARITY:

If two finite length sequence have length M and N, we can consider both of them with length greater than or equal to maximum of M and N. Thus if

$$\{x[n]\} \longleftrightarrow \{X[k]\}$$

$$\{y[n]\} \longleftrightarrow \{Y[k]\}$$

then

$$a\{x[n]\} + b\{y[n]\} \longleftrightarrow a\{X[k]\} + b\{Y[k]\}$$

2.CIRCULAR SHIFT OF A SEQUENCE:

$$\widetilde{x}[n] = x[((n))_N]$$

We can shift this sequence by m to get

$$\{\widetilde{y}[n]\} = \widetilde{x}[n-m]$$

$$\widetilde{x}[n-m] = x[((n-m))_N]$$

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

3.SHIFT PROPERTY OF A DFT:

From the definition of the circular shift, it is clear that it corresponds to linear shift of the associated periodic sequence and so the shift property of the DFS coefficient will hold for the circular shift. Hence

$$\{x[((n-m))_N],\ 0 \le n \le N-1\} \longleftrightarrow \{W_N^{km} X[k]\}$$

and

$$\{W_N^{-nl} x[n]\} \longleftrightarrow \{X[((k-l))_N\ 0 \le k \le N-1\}$$

4.DUALITY:

We have the duality for the DFS coefficient given by $\{\widetilde{X}[n]\} \longleftrightarrow \{N\widetilde{X}[-k]\}$, retaining one period of the sequences the duality property for the DFT coefficient will become

$$\{X[n]\} \longleftrightarrow \{N\, x[((-k))_N,\ 0 \le k \le N-1\}$$

5.SYMMETRY PROPERTIES:

We can infer all the symmetry properties of the DFT from the symmetry properties of the associated periodic sequence $\{\widetilde{x}[n]\}$ and retaining the first period. Thus we have

$$\{\widetilde{x}^*[n]\} \longleftrightarrow \{X^*[((-k))_N],\ 0 \le k \le N-1\}$$

and

$$\{X^*[((-n))_N],\ 0 \le n \le N-1\} \longleftrightarrow \{X^*[k]\}$$

6.CIRCULAR CONVOLUTION:

We saw that multiplication of DFS coefficients corresponds of periodic convolution of the sequence. Since DFT coefficients are DFS coefficients in the interval, $0 \le k \le N-1$, they will correspond to DFT of the sequence retained by periodically convolving associated periodic sequences and retaining their first period.

$$\widetilde{x}[n] = x[((n))_N]$$
$$\widetilde{y}[n] = y[((n))_N]$$

Periodic convolution is given by

$$\tilde{z}[n] = \sum_{k=0}^{N-1} \tilde{x}[k]\tilde{y}[n-k]$$

using properties of the modulo arithmetic

$$\tilde{z}[n] = \sum_{k=0}^{N-1} x[((k))_N]y[((n-k))_N]$$

and then

$$z[n] = \begin{cases} \tilde{z}[n], & 0 \le n \le N-1 \\ 0, & otherwise \end{cases}$$

Since $((k))_N = k, \quad 0 \le k \le N-1$ we get

$$z[n] = \sum_{l=0}^{N-1} x[k]y[((n-k))_N], \qquad 0 \le n \le N-1$$

The convolution defined by equation (6.28) is known as N-point-circular convolution of sequence $\{x[n]\}$ and $\{y[n]\}$, where both the sequence are considered sequence of length N. From the periodic convolution property of DFS it is clear that DFT of $\{z[n]\}$ is $\{X[k]Y[k]\}$. If we use the notation
$\{x[n]\} \textcircled{N} \{y[n]\}$

to denote the N point circular convolution we see that

$$\{x[n]\} \textcircled{N} \{y[n]\} \longleftrightarrow \{X[h]Y[k]\}$$

In view of the duality property of the DFT we have

$$\{x[n]y[n]\} \longleftrightarrow \frac{1}{N}\{X[k]\} \textcircled{N} \{Y[k]\}$$

2.6 LINEAR CONVOLUTION USING DFT:

If we have sequence x(n) of length L and a sequence y(n) of length M, the sequence z(n) obtained by linear convolution has length (L+M-1).This can be seen from the definition

$$Z[n] = \sum_{k=-\infty}^{\infty} x[k]y[n-k]$$
$$= \sum_{k=0}^{L-1} x[k]y[n-k]$$

From the circular convolution property of the DFT we have

$$\{v[n]\} = \{x[n]\} \textcircled{N} \{y[n]\}$$

Thus, the circular convolution of two-finite length sequences can be viewed as linear convolution, followed time aliasing, defined by equation (6.32). If N is greater than or equal to $(L + M - l)$, then there will be no time aliasing as the linear convolution produces a sequence of length $(L + M - l)$. Thus we can use circular convolution for linear convolution by padding sufficient number of zeros at the end of a finite length sequence. We can use DFT algorithm for calculating the circular convolution.

Example : Find the DFT of given sequence

$$x[k] = \begin{cases} 2 & k=0 \\ 3 & k=1 \\ -1 & k=2 \\ 1 & k=3. \end{cases}$$

$$X[r] = \sum_{k=0}^{3} x[k]e^{-j(2\pi kr/4)}$$
$$= 2 + 3 \times e^{-j(2\pi r/4)} - 1 \times e^{-j(2\pi(2)r/4)} + 1 \times e^{-j(2\pi(3)r/4)},$$

for $0 \leq r \leq 3$. On substituting different values of r, we obtain

$r = 0 \quad X[0] = 2 + 3 - 1 + 1 = 5;$

$r = 1 \quad X[1] = 2 + 3 \times e^{-j(2\pi/4)} - 1 \times e^{-j(2\pi(2)/4)} + 1 \times e^{-j(2\pi(3)/4)}$
$= 2 + 3(-j) - 1(-1) + 1(j) = 3 - 2j;$

$r = 2 \quad X[2] = 2 + 3 \times e^{-j(2\pi(2)/4)} - 1 \times e^{-j(2\pi(2)(2)/4)} + 1 \times e^{-j(2\pi(3)(2)/4)}$
$= 2 + 3(-1) - 1(1) + 1(-1) = -3;$

$r = 3 \quad X[3] = 2 + 3 \times e^{-j(2\pi(3)/4)} - 1 \times e^{-j(2\pi(2)(3)/4)} + 1 \times e^{-j(2\pi(3)(3)/4)}$
$= 2 + 3(j) - 1(-1) + 1(-j) = 3 + j2.$

Example :

Calculate the inverse DFT of

$$X[r] = \begin{cases} 5 & r = 0 \\ 3 - \mathrm{j}2 & r = 1 \\ -3 & r = 2 \\ 3 + \mathrm{j}2 & r = 3. \end{cases}$$

$$x[k] = \frac{1}{4}\sum_{k=0}^{3} X[r]\mathrm{e}^{\mathrm{j}(2\pi kr/4)} = \frac{1}{4}\left[5 + (3 - \mathrm{j}2) \times \mathrm{e}^{\mathrm{j}(2\pi k/4)} - 3 \times \mathrm{e}^{\mathrm{j}(2\pi(2)k/4)} + (3 + \mathrm{j}2) \times \mathrm{e}^{\mathrm{j}(2\pi(3)k/4)}\right],$$

for $0 \leq k \leq 3$. On substituting different values of k, we obtain

$$x[0] = \frac{1}{4}[5 + (3 - \mathrm{j}2) - 3 + (3 + \mathrm{j}2)] = 2;$$

$$x[1] = \frac{1}{4}\left[5 + (3 - \mathrm{j}2)\mathrm{e}^{\mathrm{j}(2\pi/4)} - 3\mathrm{e}^{\mathrm{j}(2\pi(2)/4)} + (3 + \mathrm{j}2)\mathrm{e}^{\mathrm{j}(2\pi(3)/4)}\right]$$

$$= \frac{1}{4}[5 + (3 - \mathrm{j}2)(\mathrm{j}) - 3(-1) + (3 + \mathrm{j}2)(-\mathrm{j})] = 3;$$

$$x[2] = \frac{1}{4}\left[5 + (3 - \mathrm{j}2)\mathrm{e}^{\mathrm{j}(2\pi(2)/4)} - 3\mathrm{e}^{\mathrm{j}(2\pi(2)(2)/4)} + (3 + \mathrm{j}2)\mathrm{e}^{\mathrm{j}(2\pi(3)(2)/4)}\right]$$

$$= \frac{1}{4}[5 + (3 - \mathrm{j}2)(-1) - 3(1) + (3 + \mathrm{j}2)(-1)] = -1;$$

$$x[3] = \frac{1}{4}\left[5 + (3 - \mathrm{j}2)\mathrm{e}^{\mathrm{j}(2\pi(3)/4)} - 3\mathrm{e}^{\mathrm{j}(2\pi(2)(3)/4)} + (3 + \mathrm{j}2)\mathrm{e}^{\mathrm{j}(2\pi(3)(3)/4)}\right]$$

$$= \frac{1}{4}[5 + (3 - \mathrm{j}2)(-\mathrm{j}) - 3(-1) + (3 + \mathrm{j}2)(\mathrm{j})] = 1.$$

Overlap-save method. In this method the size of the input data blocks is $N = L + M - 1$ and the 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 $N = L + M - 1$. An 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 an N-point DFT of the sequence is computed once and stored. The multiplication of the two N-point 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

$$\hat{Y}_m(n) = \{\hat{y}_m(0)\hat{y}_m(1)\cdots\hat{y}_m(M-1)\hat{y}_m(M)\cdots\hat{y}_m(N-1)\}$$

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

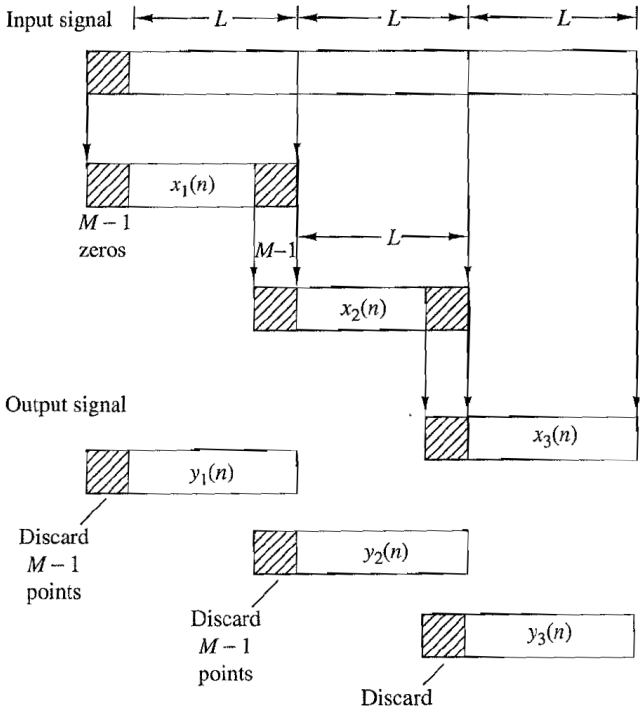
Since the data record is of length N, the first $M-1$ points of $y_m(n)$ are corrupted by aliasing and must be discarded. The last L points of $y_m(n)$ are exactly the same as the result from linear convolution and, as a consequence,

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

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

$$x_3(n) = \{x(2L-M+1) \quad x(2L-1) \quad x(2L) \quad x(3L-1)\}$$

and so forth. The resulting data sequences from the IDFT are given by (7.3.8), where the first $M-1$ points are discarded due to aliasing and the remaining L points constitute the desired result from linear convolution. This segmentation of the input data and the fitting of the output data blocks together to form the output sequence



Overlap-add method. In this 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 we append $M-1$ zeros and compute the N-point DFT. Thus the data blocks may be represented as

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

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

$$x_3(n) = \{x(2L), \ldots, x(3L-1), \underbrace{0, 0, \ldots, 0}_{M-1 \text{ zeros}}\}$$

and so on. The two N-point DFTs are multiplied together to form

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

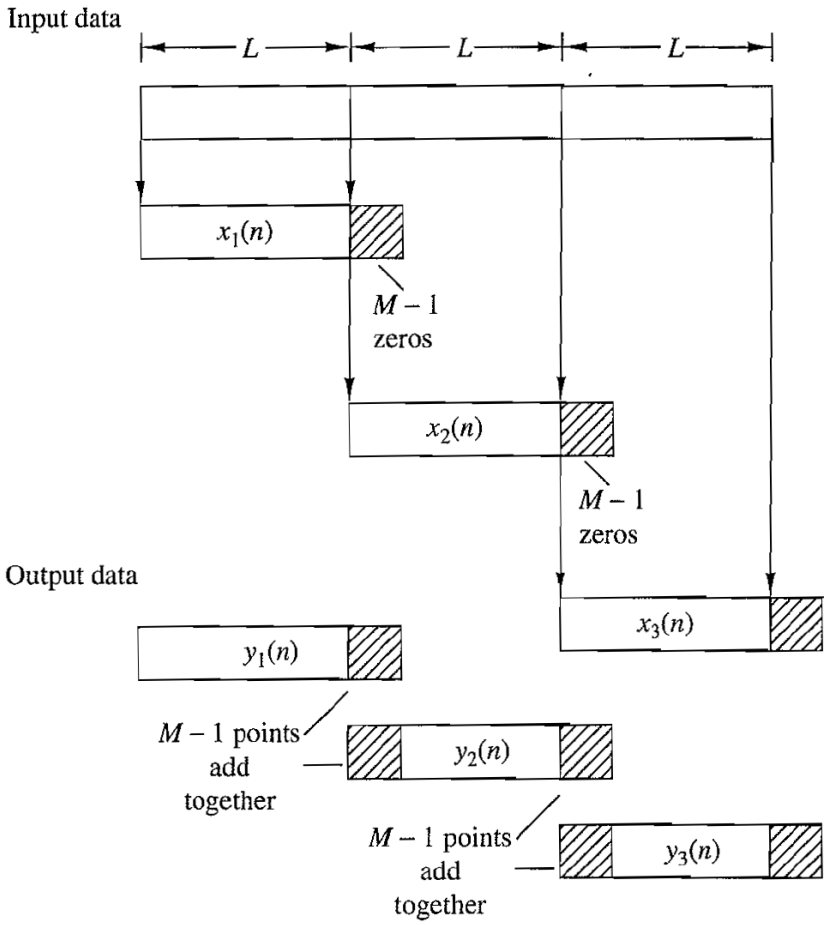
The IDFT yields data blocks of length N that are free of aliasing, since the size of the DFTs and IDFT is $N = L + M - 1$ and the sequences are increased to N-points by appending zeros to each block.

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. Hence this method is called the overlap-add method. This overlapping and adding yields the output sequence

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

The segmentation of the input data into blocks and the fitting of the output data blocks to form the output sequence are graphically illustrated in Fig. 7.3.2.

At this point, it may appear to the reader that the use of the DFT in linear FIR filtering not only is an indirect method of computing the output of an FIR filter, but also may be more expensive computationally, since the input data must first be converted to the frequency domain via the DFT, multiplied by the DFT of the FIR filter, and finally, converted back to the time domain via the IDFT. On the contrary, however, by using the fast Fourier transform algorithm, as will be shown in Chapter 8, the DFTs and IDFT require fewer computations to compute the output sequence than the direct realization of the FIR filter in the time domain. This computational efficiency is the basic advantage of using the DFT to compute the output of an FIR filter.



2.7 RELATION BETWEEN Z-TRANSFORM AND DTFT:

The Z-transform of a signal x(n) is given by

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

The DTFT of a signal x(n) is given by

$$X(w)=\sum_{n=-\infty}^{\infty} x[n]e^{-jwn},$$

Relation is given by

$Z=e^{jw}$

2.8 RELATION BETWEEN DTFT AND DFT:

The DTFT of a signal x(n) is given by

$$X(w) = \sum_{n=-\infty}^{\infty} x[n]e^{-jwn},$$

The DFT of a signal x(n) is given by

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

$$(\text{as } x[n] = 0 \text{ for } n < 0, \text{ and } n > N-1)$$

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

2.9 FAST FOURIER TRANSFORM

COMPUTATIONAL COMPLEXITY OF DFT AND FFT:

DFT:

No.of complex multiplications=N^2

No.of complex additions=N(N-1)

FFT:

No.of complex multiplications=$N\log_2 N$

No.of complex additions=$N/2\log_2 N$

INTRODUCTION:

In this section we present several methods for computing the DFT efficiently. In view of the importance of the DFT in various digital signal processing applications, such as linear filtering, correlation analysis, and spectrum analysis, its efficient computation is a topic that has received considerable attention by many mathematicians, engineers, and applied scientists.

From this point, we change the notation that *X(k)*, instead of *y(k)* in previous sections, represents the Fourier coefficients of *x(n)*.

Basically, 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

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

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

In general, the data sequence $x(n)$ is also assumed to be complex valued. Similarly, The IDFT becomes

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

Since DFT and IDFT involve basically the same type of computations, our discussion of efficient computational algorithms for the DFT applies as well to the efficient computation of the IDFT.

We observe that for each value of k, direct computation of $X(k)$ involves N complex multiplications ($4N$ real multiplications) and N-1 complex additions ($4N$-2 real additions). Consequently, to compute all N values of the DFT requires N^2 complex multiplications and N^2-N complex additions.

Direct computation of the DFT is basically inefficient primarily because it does not exploit the symmetry and periodicity properties of the phase factor W_N. In particular, these two properties are :

$$\text{Symmetry property} : W_N^{k+N/2} = -W_N^k$$

$$\text{Periodicit y property} : W_N^{k+N} = W_N^k$$

The computationally efficient algorithms described in this sectio, known collectively as fast Fourier transform (FFT) algorithms, exploit these two basic properties of the phase factor.

2.10 RADIX-2 DIT-FFT ALGORITHM:

Let us consider the computation of the $N = 2^v$ point DFT by the divide-and conquer approach. We 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, that is,

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

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

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

Now the N-point DFT can be expressed in terms of the DFT's of the decimated sequences as follows:

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn}, \qquad k = 0,1,\ldots, N-1$$
$$= \sum_{n \text{ even}} x(n) W_N^{kn} + \sum_{n \text{ odd}} x(n) W_N^{kn}$$
$$= \sum_{m=0}^{(N/2)-1} x(2m) W_N^{2mk} + \sum_{m=0}^{(N/2)-1} x(2m+1) W_N^{k(2m+1)}$$

But $W_N^2 = W_{N/2}$. With this substitution, the equation can be expressed as

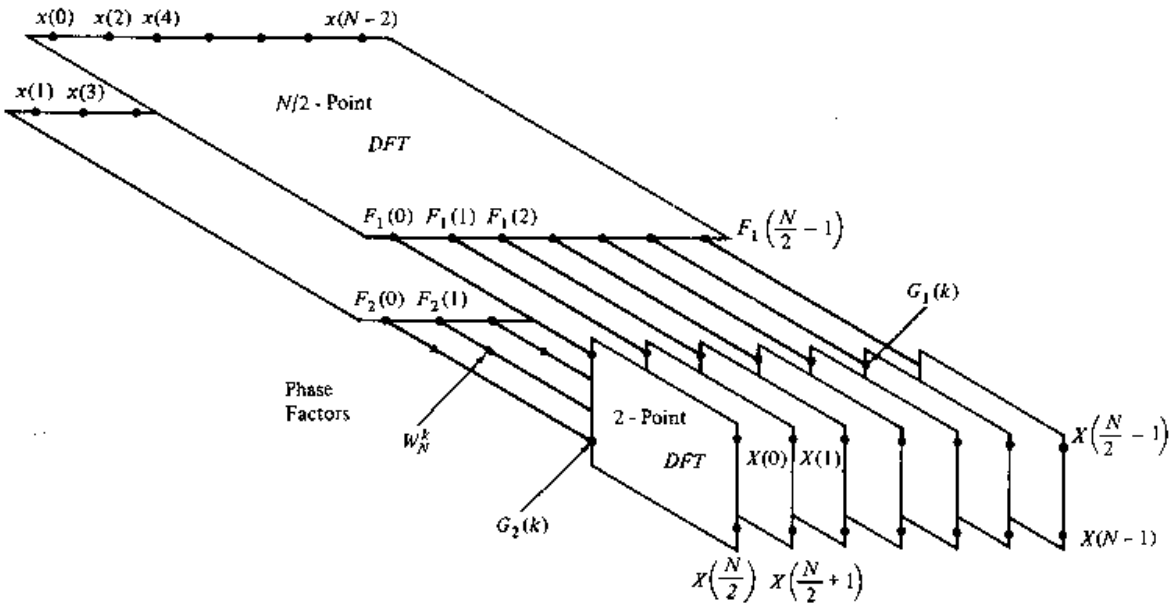
$$X(k) = \sum_{m=0}^{(N/2)-1} f_1(m) W_{N/2}^{km} + W_N^k \sum_{m=0}^{(N/2)-1} f_2(m) W_{N/2}^{km}$$
$$= F_1(k) + W_N^k F_2(k), \qquad k = 0,1,\ldots, N-1$$

where $F_1(k)$ and $F_2(k)$ are the $N/2$-point DFTs of the sequences $f_1(m)$ and $f_2(m)$, respectively.

Since $F_1(k)$ and $F_2(k)$ are periodic, with period $N/2$, we have $F_1(k+N/2) = F_1(k)$ and $F_2(k+N/2) = F_2(k)$. In addition, the factor $W_N^{k+N/2} = -W_N^k$. Hence the equation may be expressed as

$$X(k) = F_1(k) + W_N^k F_2(k), \qquad k = 0,1,\ldots, \frac{N}{2} - 1$$
$$X(k + \frac{N}{2}) = F_1(k) - W_N^k F_2(k), \qquad k = 0,1,\ldots, \frac{N}{2} - 1$$

We observe that the direct computation of $F_1(k)$ requires $(N/2)^2$ complex multiplications. The same applies to the computation of $F_2(k)$. Furthermore, there are $N/2$ additional complex multiplications required to compute $W_N^k F_2(k)$. Hence the computation of $X(k)$ requires $2(N/2)^2 + N/2 = N^2/2 + N/2$ complex multiplications. This first step results in a reduction of the number of multiplications from N^2 to $N^2/2 + N/2$, which is about a factor of 2 for N large.



By computing $N/4$-point DFTs, we would obtain the $N/2$-point DFTs $F_1(k)$ and $F_2(k)$ from the relations

$$F_1(k) = \mathrm{F}\{f_1(2n)\} + W_{N/2}^k \mathrm{F}\{f_1(2n+1)\}, \qquad k = 0,1,\ldots,\frac{N}{4}-1; \qquad n = 0,1,\ldots,\frac{N}{4}-1$$

$$F_1\left(k+\frac{N}{4}\right) = \mathrm{F}\{f_1(2n)\} - W_{N/2}^k \mathrm{F}\{f_1(2n+1)\}, \qquad k = 0,1,\ldots,\frac{N}{4}-1; \qquad n = 0,1,\ldots,\frac{N}{4}-1$$

$$F_2(k) = \mathrm{F}\{f_2(2n)\} + W_{N/2}^k \mathrm{F}\{f_2(2n+1)\}, \qquad k = 0,1,\ldots,\frac{N}{4}-1; \qquad n = 0,1,\ldots,\frac{N}{4}-1$$

$$F_2\left(k+\frac{N}{4}\right) = \mathrm{F}\{f_2(2n)\} - W_{N/2}^k \mathrm{F}\{f_2(2n+1)\}, \qquad k = 0,1,\ldots,\frac{N}{4}-1; \qquad n = 0,1,\ldots,\frac{N}{4}-1$$

$\mathrm{F}\{*\}$ represents Fourier tr ansform

The decimation of the data sequence can be repeated again and again until the resulting sequences are reduced to one-point sequences. For $N = 2^v$, this decimation can be performed $v = \log_2 N$ times. Thus the total number of complex multiplications is reduced to $(N/2)\log_2 N$. The number of complex additions is $N\log_2 N$.

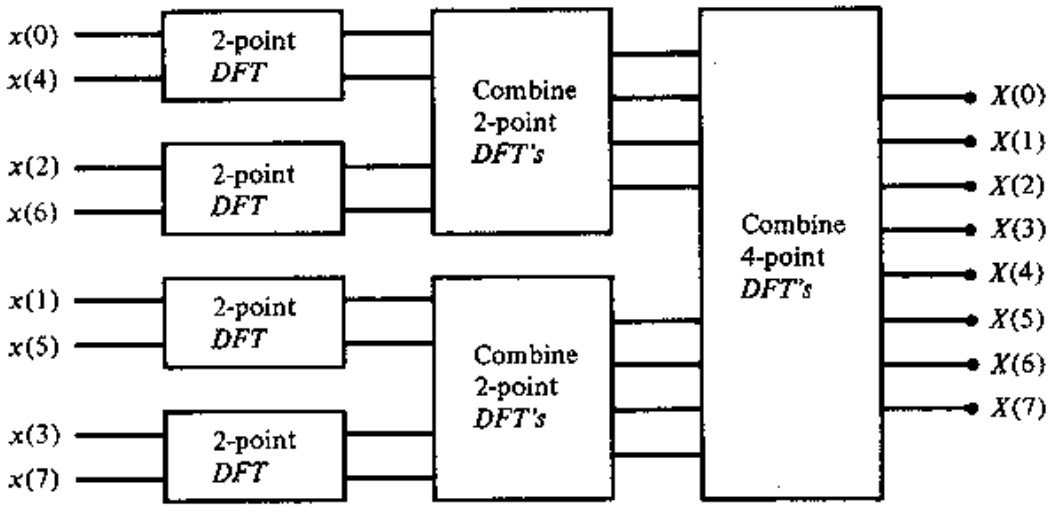
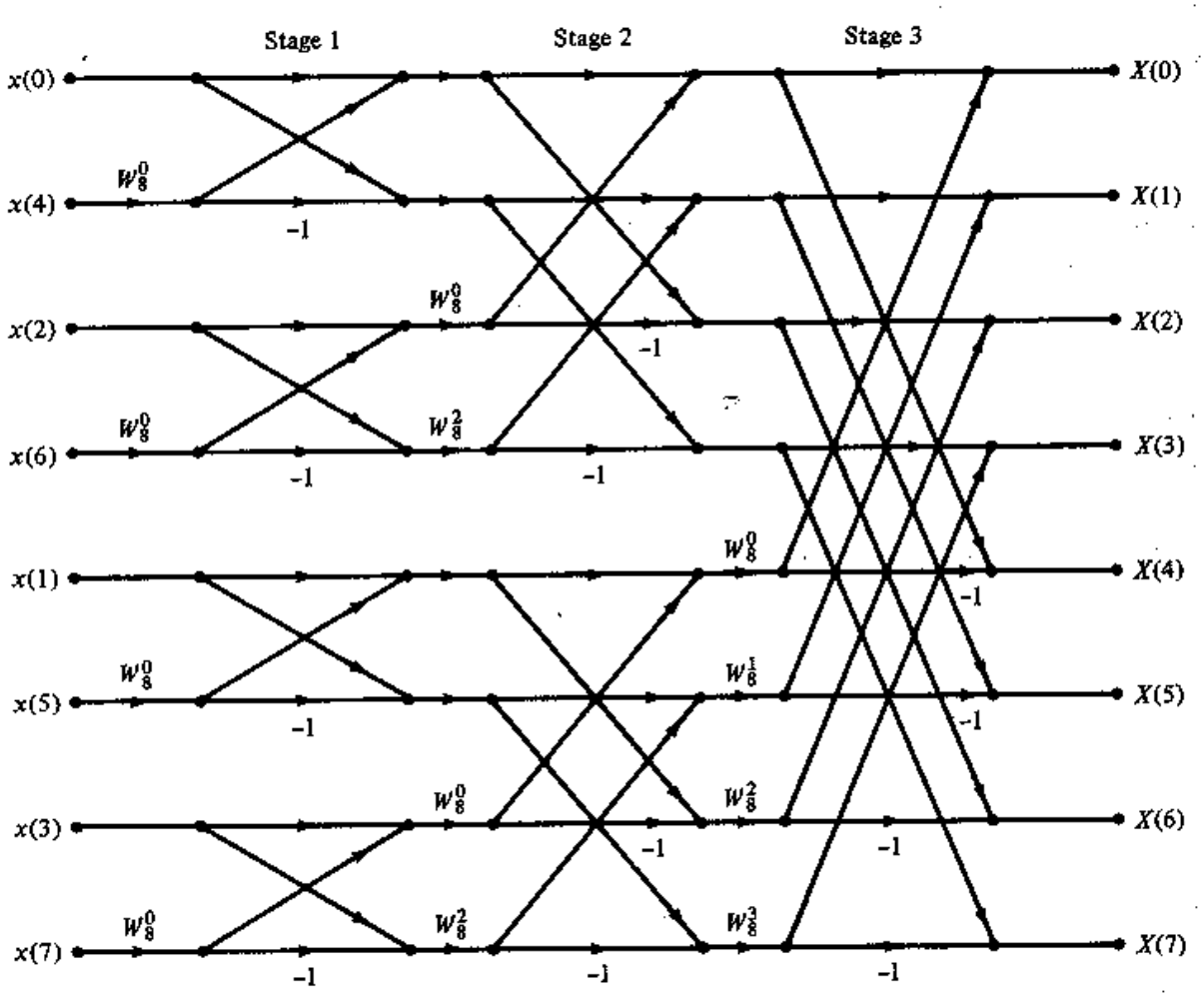


FIG.8-point fft using four 2-point Dfts



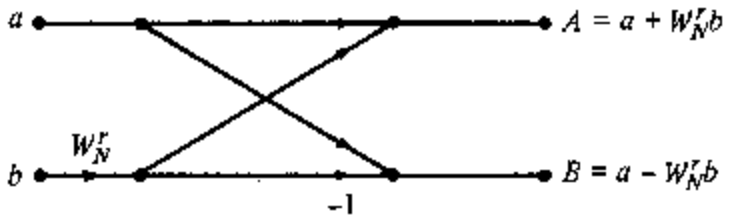
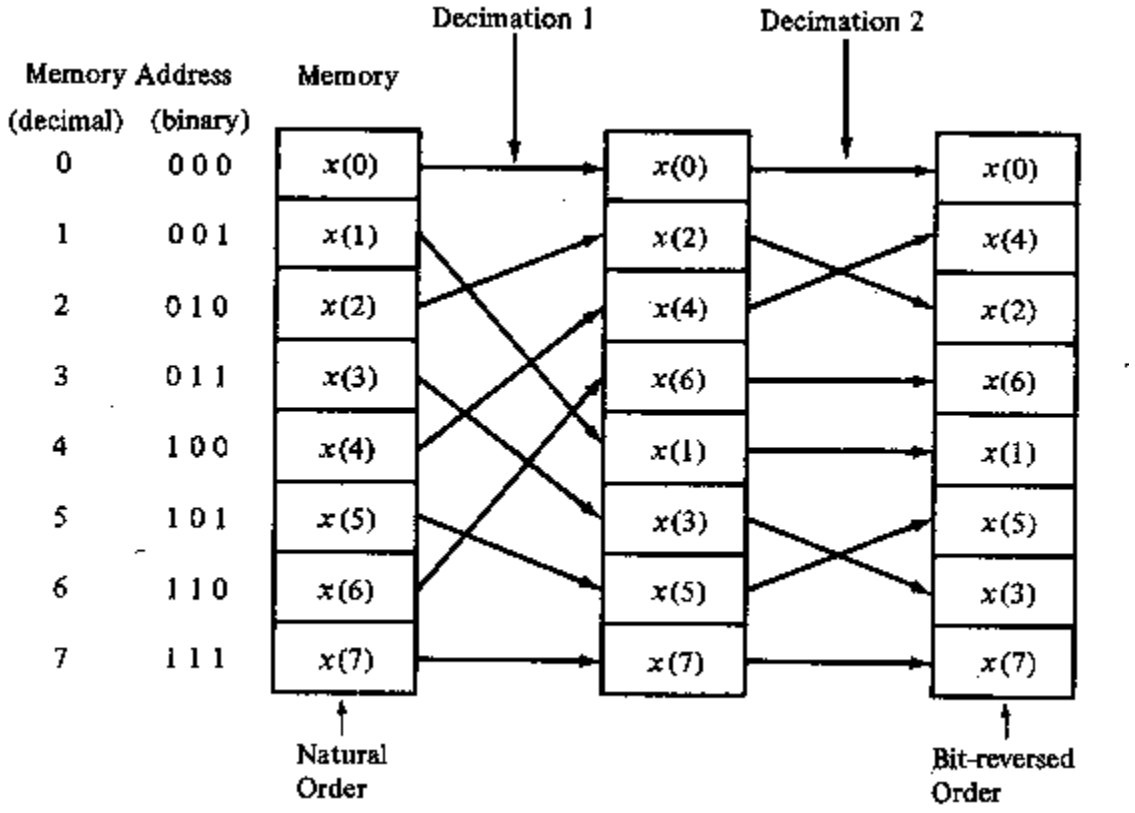


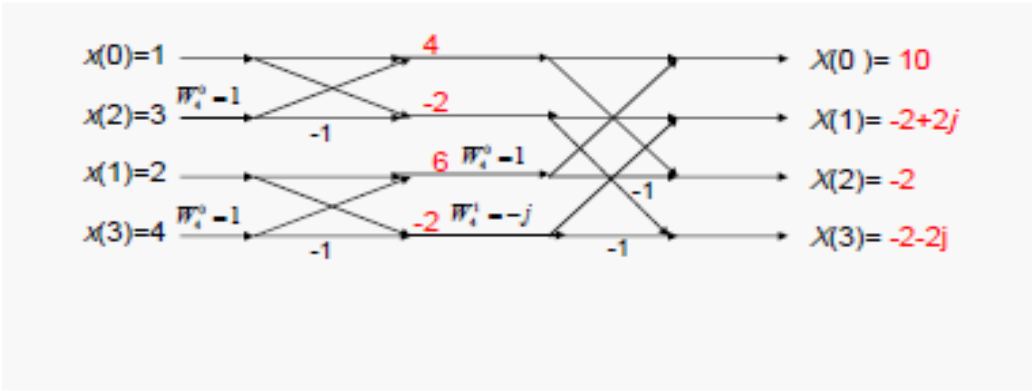
Fig.Butterfly diagram for DIT-FFT Algorithm

An important observation is concerned with the order of the input data sequence after it is decimated (v-1) times. For example, if we consider the case where N = 8, we know that the first decimation yeilds the sequence *x(0), x(2), x(4), x(6), x(1), x(3), x(5), x(7),* and the second decimation results in the sequence *x(0), x(4), x(2), x(6), x(1), x(5), x(3), x(7).* This *shuffling* of the input data sequence has a well-defined order as can be ascertained from observing Figure TC.3.5, which illustrates the decimation of the eight-point sequence.

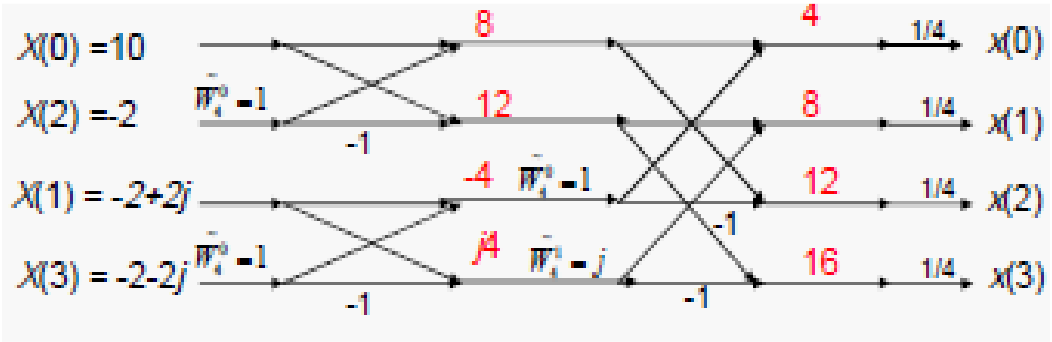


$(n_2 n_1 n_0)$	→	$(n_0 n_2 n_1)$	→	$(n_0 n_1 n_2)$
(0 0 0)	→	(0 0 0)	→	(0 0 0)
(0 0 1)	→	(1 0 0)	→	(1 0 0)
(0 1 0)	→	(0 0 1)	→	(0 1 0)
(0 1 1)	→	(1 0 1)	→	(1 1 0)
(1 0 0)	→	(0 1 0)	→	(0 0 1)
(1 0 1)	→	(1 1 0)	→	(1 0 1)
(1 1 0)	→	(0 1 1)	→	(0 1 1)
(1 1 1)	→	(1 1 1)	→	(1 1 1)

Example : Find the FFT of a given sequence x(n)={1,2,3,4}by using DIT-FFT Algorithm?



Example : Find the IFFT of a given sequence x(k)={10,-2+2j,-2,-2-2j}by using DIT-IFFT Algorithm?



2.11 RADIX-2 DIF-FFT ALGORITHM:

Another important radix-2 FFT algorithm, called the decimation-in-frequency algorithm, is obtained by using the divide-and-conquer approach. To derive the algorithm, we begin by splitting the DFT formula into two summations, one of which involves the sum over the first *N*/2 data points and the second sum involves the last N/2 data points. Thus we obtain

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

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

Since $W_N^{kN/2} = (-1)^k$

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

Now, let us split (decimate) X(k) into the even- and odd-numbered samples. Thus we obtain

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

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

where we have used the fact that $W_N^2 = W_{N/2}$

The computational procedure above 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 of the type shown in Figure TC.3.7. Consequently, the computation of the N-point DFT via the decimation-in-frequency FFT requires $(N/2)\log_2 N$ complex multiplications and $N\log_2 N$ complex additions, just as in the decimation-in-time algorithm. For illustrative purposes, the eight-point decimation-in-frequency algorithm is given in Figure.

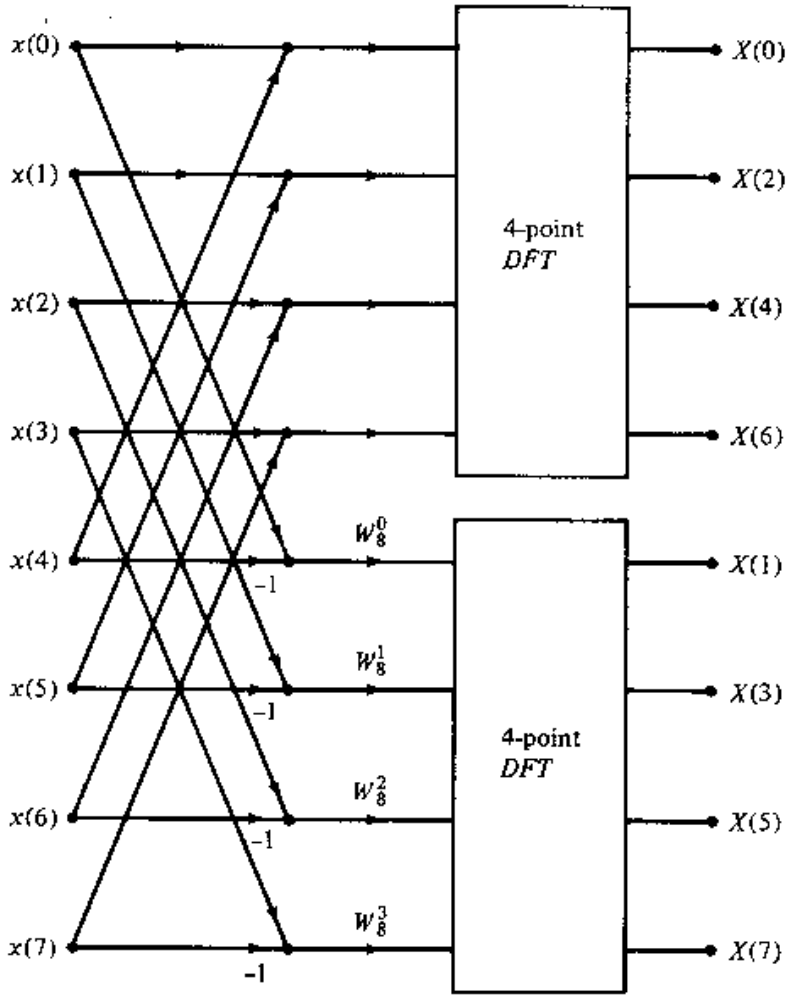


FIG.8-point FFT using two 4-point DFT

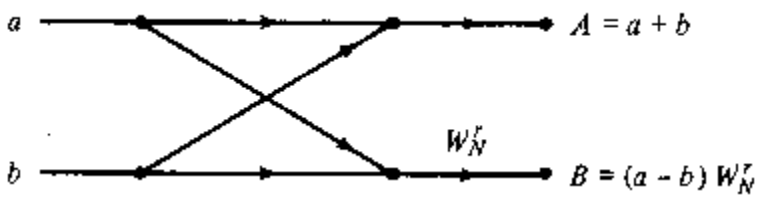
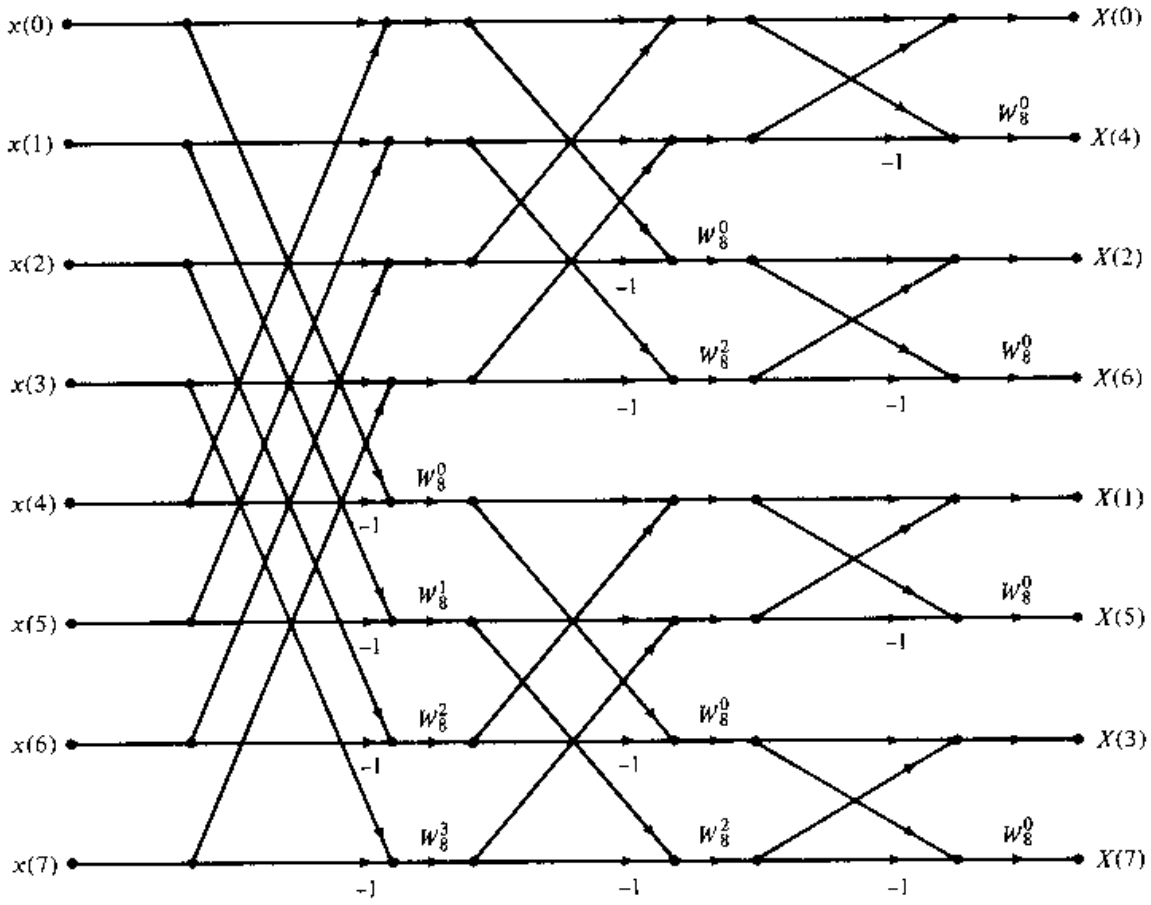
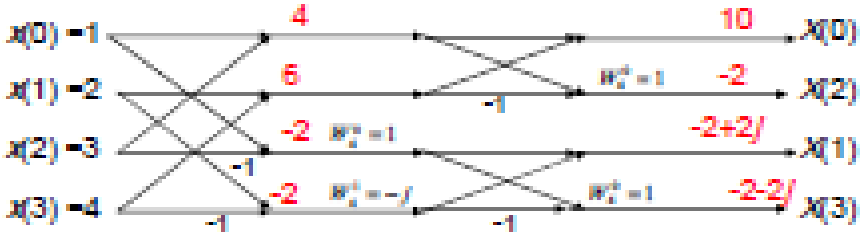


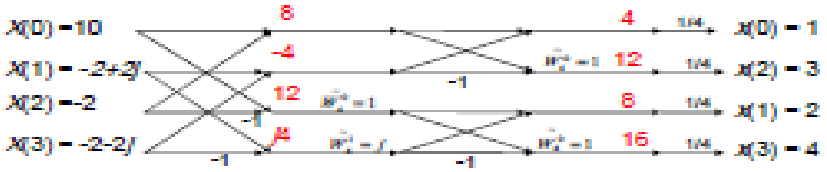
Fig. Butterfly diagram for DIF-FFT Algorithm



Example : Find the FFT of a given sequence x(n)={1,2,3,4}by using DIF-FFT Algorithm?



Example : Find the IFFT of a given sequence x(k)={10,-2+2j,-2,-2-2j}by using DIF-IFFT Algorithm?



Unit 3
IIR FILTERS

ANALOG FILTER APPROXIMATIONS:

Basically a digital filter is a linear time invariant discrete time system. The terms infinite impulse response (IIR) and finite impulse response (FIR) are used to distinguish filter types. The FIR filters are of non-recursive type, where the present output sample depends on present and previous input samples. IIR are of non-recursive type where the present output depends on previous and past input samples and output samples.

FREQUENCY SELECTIVE FILTERS:

A filter is one which rejects unwanted frequencies from the input and allows the desired frequencies to obtain the required shape of output signal. The range of frequencies that are passed through the filter is called pass band and those frequencies that are blocked are called stop band. The filters are of different types :

1.Low Pass Filter 2.High Pass Filter 3.Band Pass Filter 4.Band Reject Filter

DESIGN OF DIGITAL FILTERS FROM ANALOG FILTERS:

For the given specifications of a digital filter the derivation of digital filter transfer function requires three steps:

1. Map the desired digital filter specifications into equivalent analog filter.
2. Obtain the analog transfer function.
3. Transfer the analog transfer function to equivalent digital filter transfer function.

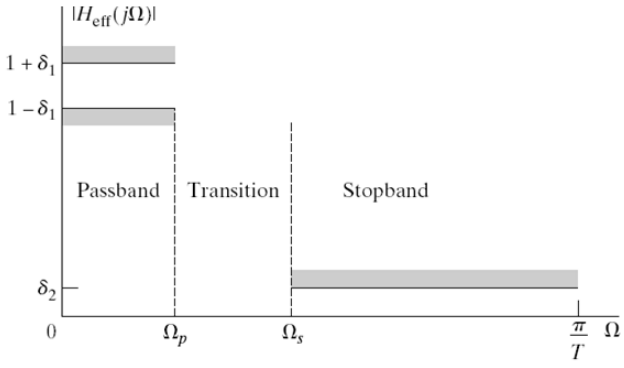


Fig (a): Magnitude response of analog LPF

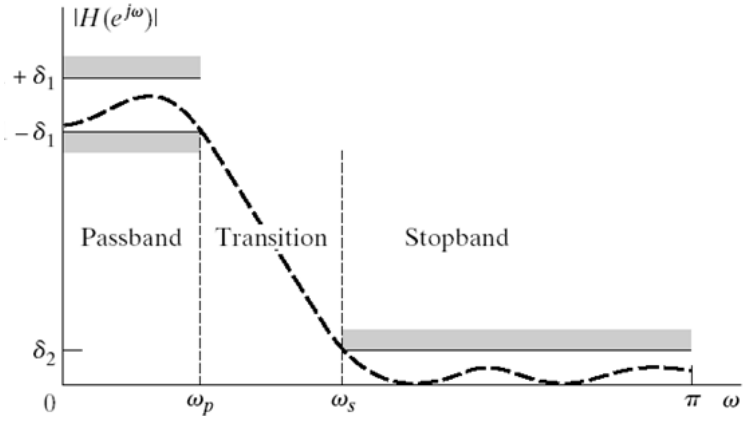


Fig (b): Magnitude response of digital LPF

Where,

ω_p = Pass band frequency in radians

ω_s = Stop band frequency in radians

ω_c = 3dB cutoff frequency in radians

ε = Parameter specifying allowable Pass band

λ = Parameter specifying allowable Stop band

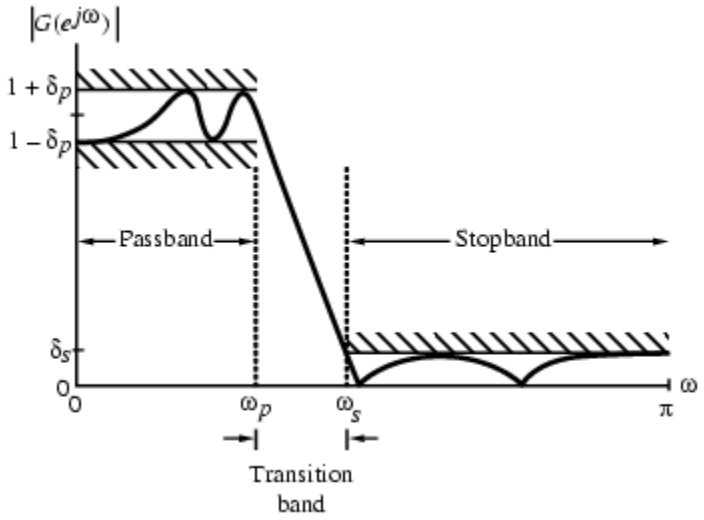


Fig : Alternate specification of magnitude response of LPF

δ_p = Pass band error tolerance

δ_s = Stop band error tolerance

The relation between parameters are

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

ANALOG LOW PASS FILTER DESIGN:

The general form of analog filter transfer function is

$$H(s) = \frac{N(s)}{D(s)} = \frac{\sum_{i=0}^{M} a_i\, s^i}{1+\sum_{i=1}^{N} b_i\, s^i}$$

Where H(s) is the Laplace Transform of impulse response of h(t) and N≥M must be satisfied.

For a stable analog filter the poles of H(s) lies in the left half of the s-plane.

The two types of analog filters we design are: 1.Butterworth Filter 2.Chebyshev Filter.

ANALOG LOW PASS BUTTERWORTH FILTER:

The squared magnitude response of butterworth low pass filter is given by

$$|H_c(j\Omega)|^2 = \frac{1}{1 + (j\Omega / j\Omega_c)^{2N}}$$

Where, N= Order of the filter

Ωc = Cutoff frequency

The magnitude response decrease monotonically as shown in figure and the maximum response is unity at Ω=0 ie,.. as the N increases the response approaches ideal low pass characteristics.

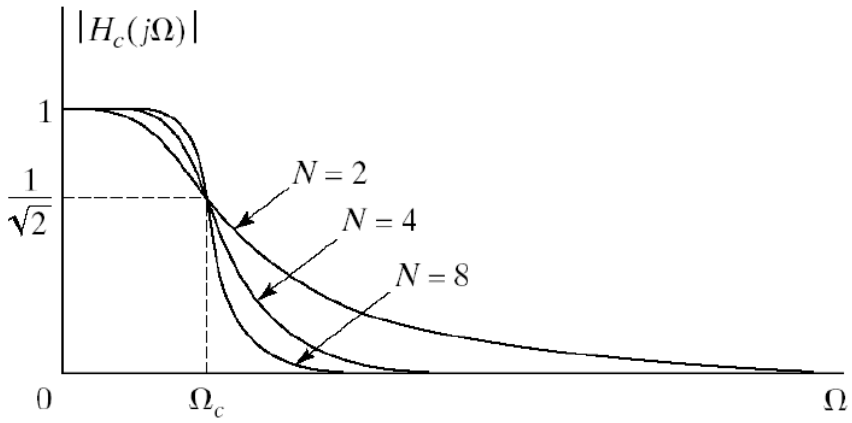


Fig: Low Pass butterworth magnitude response.

We can get magnitude square function of normalized butterworth filter(1 rad/sec cut off frequency) as

$$|H(j\Omega)|^2=\frac{1}{1+\Omega^{2N}} \qquad N=1,2,3\ldots$$

For transfer function of stable filter substitute Ω=s/j in above eqn.

$$|H(j\Omega)|^2=|H(\Omega)|^2=H(-s^2)=H(j\Omega)H(-j\Omega)= \frac{1}{1+(\frac{s}{j})^{2N}}$$

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

Equating denominator to zero, poles are obtained

$$\text{i.e } 1 + (-s^2)^N=0$$

for **N odd**, the above equation reduces to $s^{2N} = 1=e^{j2\pi k}$

now the roots can be found as $s_k = e^{j\pi k/N}$ k=1,2,3,……2N

for **N even,** the above equation reduces to $s^{2N} = -1 = e^{j(2k-1)\pi}$

which gives $s_k = e^{j(2k-1)\pi/2N}$ k=1,2,3….

Note: The poles which lie in the left half of the s plane, the same can be found using the formula

$s_k = e^{j\varphi_k}$ where $\varphi_k = \frac{\pi}{2} + \frac{(2K-1)\pi}{2N}$ k=1,2,3……..N

LIST OF BUTTERWORTH POLYNOMIALS:

n	
1	$s+1$
2	$s^2+\sqrt{2}s+1$
3	$(s^2+s+1)(s+1)$
4	$(s^2+0.76536s+1)(s^2+1.84776s+1)$
5	$(s+1)(s^2+0.6180s+1)(s^2+1.6180s+1)$
6	$(s^2+0.5176s+1)(s^2+\sqrt{2}s+1)(s^2+1.9318s+1)$
7	$(s+1)(s^2+0.4450s+1)(s^2+1.2456s+1)(s^2+1.8022s+1)$
8	$(s^2+0.3986s+1)(s^2+1.1110s+1)(s^2+1.6630s+1)(s^2+1.9622s+1)$

The unnormalised poles are given by $s_k' = \Omega_c s_k$.

The transfer function of butterworth filter can be obtained by substituting $s \longrightarrow s/\Omega_c$

ORDER OF FILTER

The magnitude function is given as

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

Taking log on both sides

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

at pass band frequency the attenuation is equal to α_p

$$20 \log |H(j\Omega p)| = -\alpha_p = -10\log[1+\varepsilon^2]$$

$$\alpha_p = 10\log[1+\varepsilon^2]$$

$$0.1\,\alpha_p = \log[1+\varepsilon^2]$$

By taking antilog on both sides

$$\varepsilon = (10^{0.1\,\alpha_p} - 1)^{1/2}$$

at stop band frequency the minimum attenuation is equal to α_s

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

$$20 \log |H(j\Omega p)| = -\alpha_s = -10\log[1+\varepsilon^2(\frac{\Omega_s}{\Omega_p})^{2N}]$$

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

$$(\frac{\Omega_s}{\Omega_p})^{2N} = \frac{10^{0.1\,\alpha s}-1}{10^{0.1\,\alpha p}-1}$$

Taking log on both sides

$$N = \frac{log\sqrt{\frac{10^{0.1\,\alpha s}-1}{10^{0.1\,\alpha p}-1}}}{log\frac{\Omega_s}{\Omega_p}}$$

Round off N to next higher integer.

$$N \geq \frac{log\sqrt{\frac{10^{0.1\,\alpha s}-1}{10^{0.1\,\alpha p}-1}}}{log\frac{\Omega_s}{\Omega_p}}$$

$$N \geq \frac{\log(\frac{\lambda}{\varepsilon})}{log\frac{\Omega_s}{\Omega_p}} \quad \text{where } \lambda^2=[10^{0.1\,\alpha s}-1] \; and \; \varepsilon^2=[10^{0.1\,\alpha p}-1]$$

For simplicity $A=\frac{\lambda}{\varepsilon}$ and $k=\frac{\Omega_p}{\Omega_s}$ the transition ratio

Therefore, the order of the low pass butterworth analog filter $N=\frac{\log A}{\log\frac{1}{k}}$

ANALOG LOW PASS CHEBYSHEV FILTER

The magnitude squared response of the analog lowpassType I Chebyshevfilter of Nth order is given by:

|H(W)|2= 1/[1 + e2TN2(W/Wp)].

where TN(W) is the Chebyshev polynomial of order N:

TN(W) = cos(Ncos-1W),|W| ≤1,

= cosh(Ncosh-1W),|W| > 1.

The polynomial can be derived via a recurrence relation given by

Tr(W) = 2WTr-1(W) –Tr-2(W),r ≥2, with T0(W) = 1 and T1(W) = W.

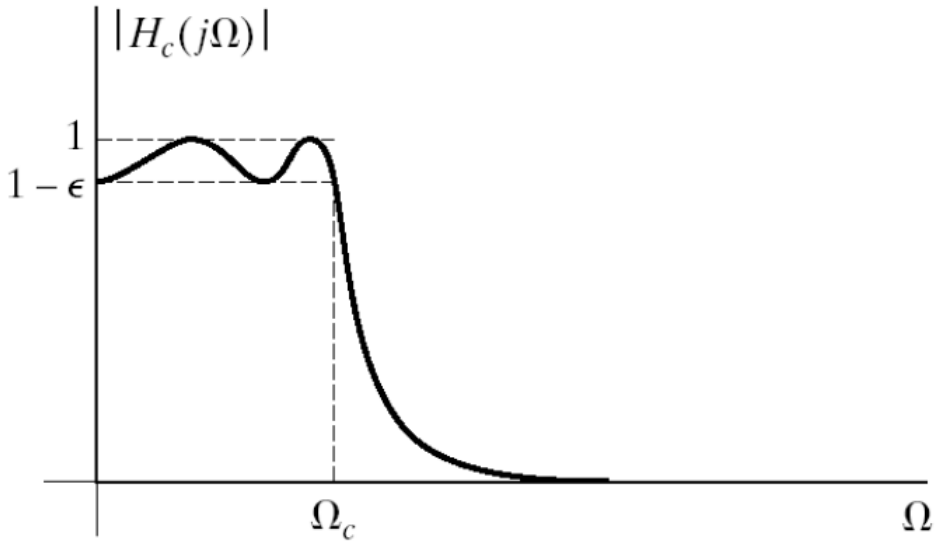
The magnitude squared response of the analog lowpassType II or inverse Chebyshevfilter of Nth order is given by:

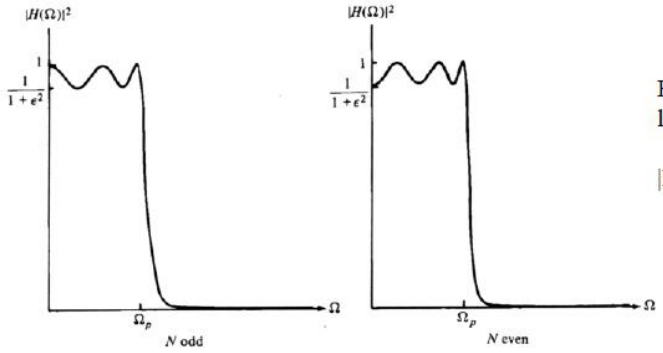
|H(W)|2= 1/[1 + e2{TN(Ws/Wp)/ TN(Ws/W)}2].

Equiripple in the passband and monotonic in the stopband.

Or equiripple in the stopband and monotonic in the passband.

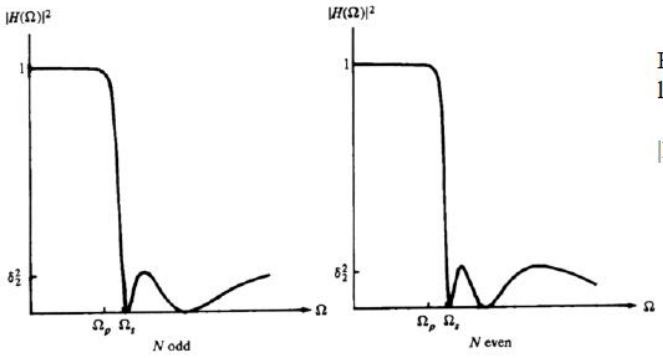
$$|H_c(j\Omega)|^2 = \frac{1}{1+\varepsilon^2 V_N^2(\Omega/\Omega_c)} \qquad V_N(x) = \cos(N\cos^{-1}x)$$





Frequency response of
lowpass Type I Chebyshev filter

$|H(W)|^2 = 1/[1 + e^2 T_N^2(W/W_p)]$



Frequency response of
lowpass Type II Chebyshev filter

$|H(W)|^2 = 1/[1 + e^2\{T_N^2(W_s/W_p)/T_N^2(W_s/W)\}]$

$$N = \log_{10}[(\sqrt{1 - d_2^2} + \sqrt{1 - d_2^2(1 + e^2)})/ed_2]/\log_{10}[(W_s/W_p) + \sqrt{(W_s/W_p)^2 - 1}\,]$$
$$= [\cosh^{-1}(d/e)]/[\cosh^{-1}(W_s/W_p)]$$

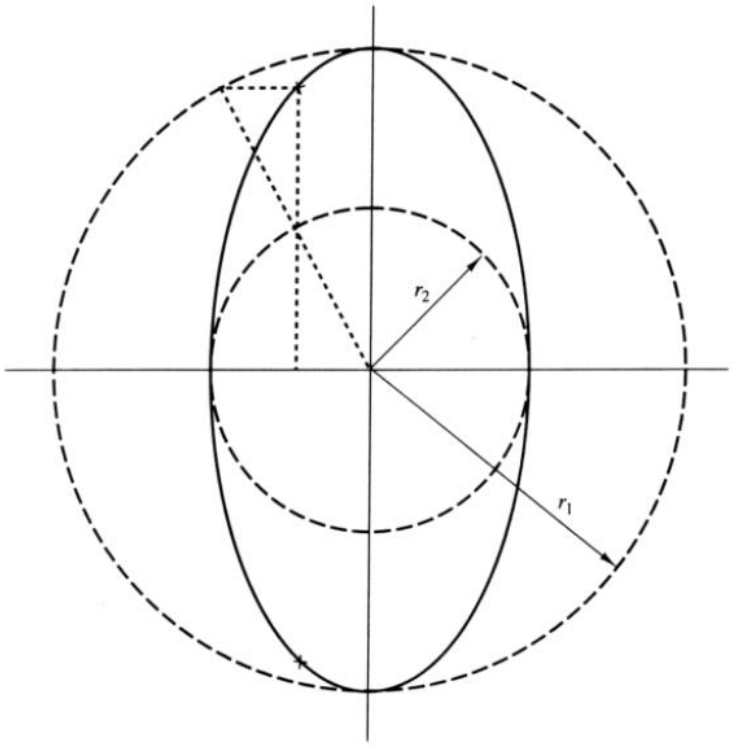
for both Type I and II Chebyshev filters, and where

$$d_2 = 1/\sqrt{1 + d^2}.$$

- The poles of a Type I Chebyshev filter lie on an ellipse in the s-plane with major axis $r_1 = W_p\{(b^2 + 1)/2b]$ and minor axis $r_1 = W_p\{(b^2 - 1)/2b]$ where b is related to e according to

  $$b = \{[\sqrt{1 + e^2} + 1]/e\}^{1/N}$$

- The zeros of a Type II Chebyshev filter are located on the imaginary axis.



Determination of the pole locations for a Chebyshev filter.

Type I: pole positions are

$x_k = r_2 \cos f_k$

$y_k = r_1 \sin f_k$

$f_k = [p/2] + [(2k + 1)p/2N]$

$r_1 = W_p[b_2 + 1]/2b$

$r_2 = W_p[b^2 - 1]/2b$

$b = \{[\sqrt{1 + e^2} + 1]/e\}^{1/N}$

Type II: zero positions are

$s_k = jW_s/\sin f_k$

and pole positions are

$v_k = W_s x_k / \sqrt{x_k^2 + y_k^2}$

$w_k = W_s y_k / \sqrt{x_k^2 + y_k^2}$

$b = \{[1 + \sqrt{1 - d_2^2}]/d_2\}^{1/N}$

$k = 0, 1, \ldots, N\text{-}1.$

DESIGN OF IIR FILTER FROM ANALOG FILTERS

1. IMPULSE INVARIANCE
2. STEP INVARIANT
3. BILINEAR TRANSFORMATION

Impulse Invariance Method is simplest method used for designing IIR Filters. Important Features of this Method are

1. In impulse variance method, Analog filters are converted into digital filter just by replacing unit sample response of the digital filter by the sampled version of impulse response of analog filter. Sampled signal is obtained by putting t=nT hence h(n) = ha(nT) n=0,1,2. ……. where h(n) is the unit sample response of digital filter and T is sampling interval.

2. But the main disadvantage of this method is that it does not correspond to simple algebraic mapping of S plane to the Z plane. Thus the mapping from analog frequency to digital frequency

is many to one. The segments $(2k-1)\Pi/T \leq \Omega \leq (2k+1)\Pi/T$ of $j\Omega$ axis are all mapped on the unit circle $\Pi \leq \omega \leq \Pi$. This takes place because of sampling.

3. Frequency aliasing is second disadvantage in this method. Because of frequency aliasing, the frequency response of the resulting digital filter will not be identical to the original analog frequency response.

4. Because of these factors, its application is limited to design low frequency filters like LPF or a limited class of band pass filters.

RELATIONSHIP BETWEEN Z PLANE AND S PLANE

In impulse invariant method the IIR filter is designed such that unit impulse response h(n) of digital filter is the sampled version of the impulse response of analog filter.

The Z transform of IIR is given by

$$H(Z)=\sum_{n=0}^{\infty} h(n)z^{-n}$$

$$H(Z)/z=e^{sT}=\sum_{n=0}^{\infty} h(n)e^{-sTn}$$

Z is represented as $re^{j\omega}$ in polar form and relationship between Z plane and S plane is given as

$$Z=e^{ST} \text{ where } s=\sigma + j\,\Omega.$$

$$Z= e^{ST} \text{ (Relationship Between Z plane and S plane)}$$

$$Z= e^{(\sigma + j\,\Omega)\,T}$$

$$= e^{\sigma T} . e^{j\,\Omega\,T}$$

Comparing Z value with the polar form we have.

$$r= e^{\sigma T} \text{ and } \omega = \Omega\, T$$

Here we have three condition

1) If $\sigma = 0$ then $r=1$

2) If $\sigma < 0$ then $0 < r < 1$

3) If $\sigma > 0$ then $r > 1$

Thus

1) Left side of s-plane is mapped inside the unit circle.

2) Right side of s-plane is mapped outside the unit circle.

3) $j\Omega$ axis is in s-plane is mapped on the unit circle.

Im(z)

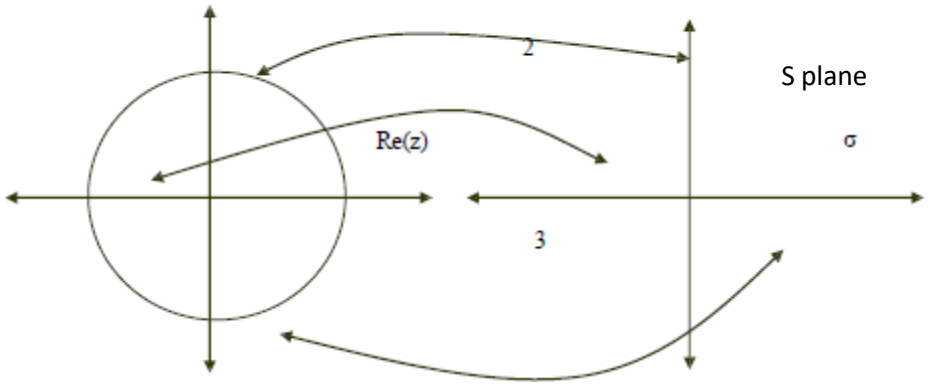


Fig: Impulse invariant pole mapping

Let the system function of analog filter is n

$$Ha(s)= \sum_{k=1} C_k / s-p_k \qquad (1)$$

where pk are the poles of the analog filter and ck are the coefficients of partial fraction expansion. The impulse response of the analog filter ha(t) is obtained by inverse Laplace transform and given as

$$ha(t) = \sum_{k=1}^{n} C_k \, e^{pkt} \qquad (2)$$

The unit sample response of the digital filter is obtained by uniform sampling of ha(t). $h(n) = h_a(nT)$ n=0,1,2. ……

$$h(n) = \sum_{k=1}^{n} C_k \, e^{pknT} \qquad (3)$$

System function of digital filter H(z) is obtained by Z transform of h(n).

$$H(z) = \sum_{k=1}^{N} C_k \sum_{n=0}^{\infty} \left[e^{pkT} z^{-1}\right]^n \qquad (4)$$

$$= \sum_{k=1}^{N} \frac{c_k}{1-e^{p_{kT}} Z^{-1}}$$

Using the standard relation and comparing equ 1 and 4

If $H_a(s)=\sum_{k=1}^{N} \frac{c_k}{s-p_k}$ then $H(z)= \sum_{k=1}^{N} \frac{c_k}{1-e^{p_{kT}} Z^{-1}}$

Steps to design Digital IIR Filter using impulse invariant technique:

1. For the given specifications, find Ha(s), transfer function of an analog filter.
2. Select the sampling rate of the digital filter .
3. Express the analog filter transfer function as the sum of single pole filters.

$$\text{Ha(s)}= \sum_{k=1}^{N} \frac{c_k}{s-p_k}$$

4. Compute the z transform of the digital filter by using the formula.

$$\text{H(z)}= \sum_{k=1}^{N} \frac{c_k}{1-e^{p_k T Z^{-1}}}$$

Example:

Convert the analog filter with system function

$$H_a(s) = [s + 0.1]/[(s + 0.1)^2 + 9]$$

into a digital IIR filter by means of the impulse invariance method.

The analog filter has a zero at s = - 0.1 and a pair of complex conjugate poles at p_k = - 0.1 ± j3.

Thus,

$$H_a(s) = \frac{\frac{1}{2}}{s+0.1-j3} + \frac{\frac{1}{2}}{s+0.1+j3}$$

Then

$$H(z) = \frac{\frac{1}{2}}{1-e^{-0.1T}e^{j3T}z^{-1}} + \frac{\frac{1}{2}}{1-e^{-0.1T}e^{-j3T}z^{-1}}$$

STEP INVARIANT METHOD:

The step response y(t) is defined as the output of a LTI system due to a unit step input signal x(t)=u(t).Then

$$\text{X(s)}= \frac{1}{s} \text{ and Y(s)= X(s)H(s)= } \frac{1}{s} \text{ H(s).}$$

We know that a digital filter is equivalent to an analog filter in the sense of time domain invariance, if equivalent input yield equivalent outputs.

Therefore the sampled input to digital filter is x(nT)=x(n)=u(n)Then

$$\text{X(z)= } \frac{1}{1-z-1} \text{ and y(n)=y(nT).}$$

The transfer function of the digital filter is given by

$$\text{H(z)= Y(z)/X(z)=(1-z}^{-1}\text{)Y(z).}$$

BILINEAR TRANSFORMATION METHOD:

The method of filter design by impulse invariance suffers from aliasing. Hence in order to overcome this drawback Bilinear transformation method is designed. In analogue domain frequency axis is an infinitely long straight line while sampled data z plane it is unit circle radius. The bilinear transformation is the method of squashing the infinite straight analog frequency axis so that it becomes finite.

Important Features of Bilinear Transform Method are

1. Bilinear transformation method (BZT) is a mapping from analog S plane to digital Z plane. This conversion maps analog poles to digital poles and analog zeros to digital zeros. Thus all poles and zeros are mapped.

2. This transformation is basically based on a numerical integration techniques used to simulate an integrator of analog filter.

3. There is one to one correspondence between continuous time and discrete time frequency points. Entire range in Ω is mapped only once into the range $-\prod \leq \omega \leq \prod$.

4. Frequency relationship is non-linear. Frequency warping or frequency compression is due to non-linearity. Frequency warping means amplitude response of digital filter is expanded at the lower frequencies and compressed at the higher frequencies in comparison of the analog filter.

5. But the main disadvantage of frequency warping is that it does change the shape of the desired filter frequency response. In particular, it changes the shape of the transition bands.

Bilinear transformation

$$s = \frac{2}{T_d}\left(\frac{1 - z^{-1}}{1 + z^{-1}}\right)$$

Transformed system function

$$H(z) = H_c\left[\frac{2}{T_d}\left(\frac{1 - z^{-1}}{1 + z^{-1}}\right)\right]$$

$$S = \frac{2}{T}\,\frac{re^{j\omega} - 1}{re^{j\omega} + 1}$$

$$S = \frac{2}{T}\,\frac{r(\cos\omega + j\sin\omega) - 1}{r(\cos\omega + j\sin\omega) + 1}$$

$$S = \frac{2}{T}\left[\frac{r^2 - }{1+r^2+2r\cos\omega} + \frac{2r \quad j\,2\,r\sin\omega}{+r^2+2r\cos\omega}\right]$$

Comparing the above equation with S= σ + j Ω. We have

$$\sigma = \frac{2}{T}\frac{r^2 - 1}{1 + r^2 + 2r\cos\omega}$$

$$\Omega = \frac{2}{T}\frac{2r\sin\omega}{1 + r^2 + 2r\cos\omega}$$

$$\Omega = \frac{2}{T}\frac{\sin\omega}{1+\cos\omega}$$

Up on simplification, we get

$$\Omega = (2/T)\tan(\omega/2)$$

$$\omega = 2\tan^{-1}(\Omega T/2)$$

WARPING EFFECT

Let Ω and ω represents the frequency variables in the analog filter and the derived digital filter resp.

$$\Omega = \frac{2}{T}\tan\frac{\omega}{2}$$

For small value of ω

$$\Omega = \frac{2}{T}\frac{\omega}{2} = \frac{\omega}{T}$$

$$\omega = \Omega T$$

For low frequencies the relation between ω and Ω are linear, as a result the digital filter have the same amplitude response as analog filter. For high frequencies however the relation between ω and Ω becomes nonlinear and distortion is introduced in the frequency scale of digital filter to that of analog filter. This is known as **warping effect.**

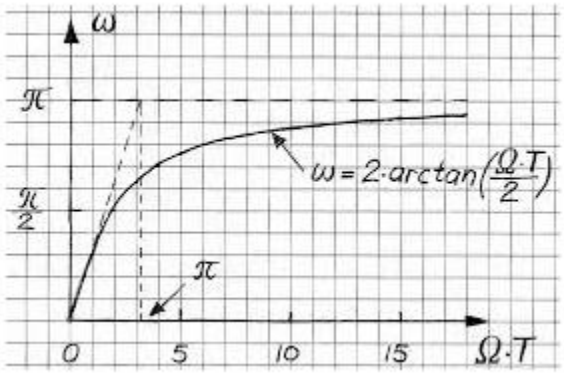


Fig: Relationship between ω and Ω

The influence of the warping effect on amplitude response is shown in figure below. The analog filter with a number of pass bands centered at regular intervals. The derived digital filter will have same number of pass bands. But the centre frequencies and bandwidth of higher frequency pass band will tend to reduce disproportionately.

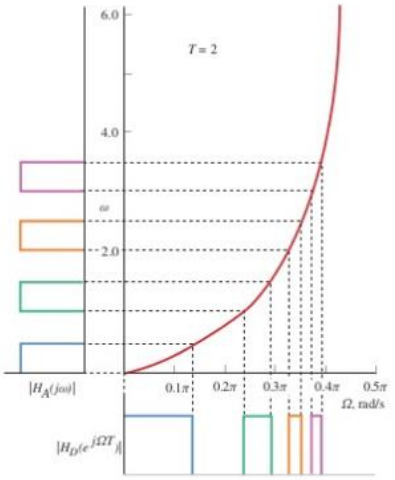
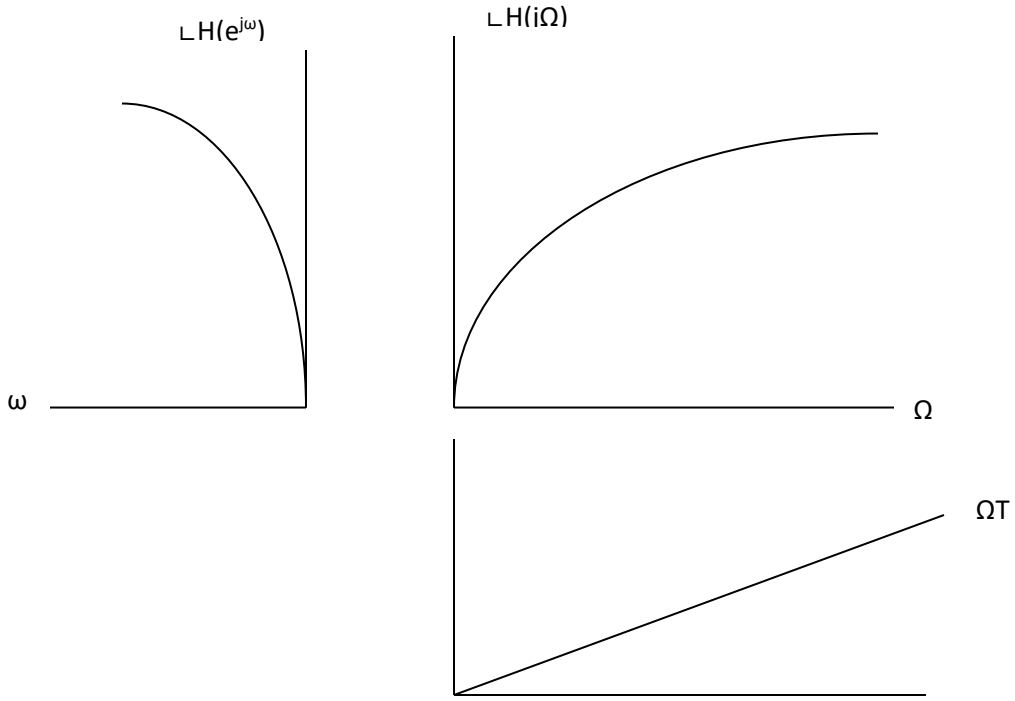


Fig : Effect on magnitude response due to warping effect

The influence of warping effect on the phase response is as shown below ,Considering an analog filter with linear phase response, the phase response of derived digital filter will be non linear.



Prewarping

The prewarping effect can be eliminated by prewarping the analog filter. This can be done by finding prewarping analog frequencies using the formula

$$\Omega=\frac{2}{T}\tan\frac{\omega}{2}$$

Therefore we have $\Omega p=\frac{2}{T}\tan\frac{\omega_p}{2}$

And $\Omega s=\frac{2}{T}\tan\frac{\omega_s}{2}$

STEPS TO DESIGN DIGITAL FILTER USING BILINEAR TRANSFORM TECHNIQUE:

1. From the given specifications, find prewarping analog frequencies using formula $\Omega=\frac{2}{T}\tan\frac{\omega}{2}$.
2. Using the analog frequencies find H(s) of the analog filter.
3. Select the sampling rate of the digital filter, call it T sec/sample.
4. Substitute $Z = \frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}$ into the transfer function.

SPECTRAL TRANSFORMATIONS:

IN ANALOG DOMAIN : A analog low pass filter can be converted into a analog High Pass, Band Stop, Band Pass or another Low Pass digital filter as given below

Low Pass to Low Pass:

$$S \longrightarrow \frac{s}{\Omega_c}$$

Low Pass to High Pass:

$$S \longrightarrow \frac{\Omega_c}{s}$$

Low Pass to Band Pass:

$$S \longrightarrow \frac{s^2+\Omega_l\Omega_u}{s(\Omega_u-\Omega_l)}$$

Low Pass to Band Stop:

$$S \longrightarrow \frac{s(\Omega_u-\Omega_l)}{s^2+\Omega_l\Omega_u}$$

IN DIGITAL DOMAIN:

A digital low pass filter can be converted into a digital High Pass, Band Stop, Band Pass or another Low Pass digital filter as given below

Low Pass to Low Pass:

$$\mathbf{Z^{-1}} \longrightarrow \frac{z^{-1}-\alpha}{1-\alpha z^{-1}}$$

$$\textbf{Where } \alpha = \frac{\sin[(\omega p-\omega p')/2]}{\sin[(\omega p+\omega p')/2]}$$

ω_p = Pass band frequency of low pass filter

ω_p' = Pass band frequency of new filter

Low Pass to High Pass:

$$\mathbf{Z^{-1}} \longrightarrow \text{---}\left[\frac{z^{-1}+\alpha}{1+\alpha z^{-1}}\right]$$

$$\textbf{Where } \alpha = -\frac{\cos[(\omega p+\omega p')/2]}{\cos[(\omega p'-\omega p)/2]}$$

ω_p = Pass band frequency of low pass filter

ω_p' = Pass band frequency of high pass filter

Low Pass to Band Pass:

$$\mathbf{Z^{-1}} \longrightarrow \text{---}\left[\frac{z^{-2}-\frac{2\propto k}{1+k}z^{-1}+\frac{k-1}{k+1}}{\frac{k-1}{k+1}z^{-2}-\frac{2\propto k}{K+1}z^{-1}+1}\right]$$

Where $\alpha = \frac{\cos(\omega_u+\omega_l)/2}{\cos(\omega_u-\omega_l)/2}$ and $k = [\cot(\omega_u-\omega_l)/2][\tan(\omega_p/2)]$

ω_u = Upper cutoff frequency

ω_l = Lower cutoff frequency

Low Pass to Band Stop:

$$\mathbf{Z^{-1}} \longrightarrow \text{---}\left[\frac{z^{-2}-\frac{2\propto k}{1+k}z^{-1}+\frac{1-k}{k+1}}{\frac{1-k}{k+1}z^{-2}-\frac{2\propto k}{K+1}z^{-1}+1}\right]$$

Where $\alpha = \frac{\cos(\omega_u+\omega_l)/2}{\cos(\omega_u-\omega_l)/2}$ and $k = [\tan(\omega_u-\omega_l)/2][\tan(\omega_p/2)]$

ω_u = Upper cutoff frequency

ω_l = Lower cutoff frequency

PROBLEMS

1.

Represents the transfer function of a low-pass filter (not butterworth) with a pass-band of 1 rad/sec. Use freq transformation to find the transfer function of the following filters:

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

Represents the transfer function of a low pass filter (not butterworth) with a passband of 1 rad/sec. Use freq transformation to find the transfer function of the following filters: (Apr/May-08) (12)

1. **A LP filter with a pass band of 10 rad/sec**
2. **A HP filter with a cutoff freq of 1 rad/sec**
3. **A HP filter with a cutoff freq of 10 rad/sec**
4. **A BP filter with a pass band of 10 rad/sec and a corner freq of 100 rad/sec**
5. **A BS filter with a stop band of 2 rad/sec and a center freq of 10 rad/sec**

Solution:

Given

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

a. LP – LP Transform

replace

$$s \rightarrow \frac{s}{\Omega'_p} = \frac{s}{10}$$

$$sub \quad H_a(s) = H(s)\,|_{s \rightarrow \frac{s}{10}} = \frac{1}{\left(\left(\frac{s}{10}\right)^2 + \left(\frac{s}{10}\right) + 1\right)}$$

$$= \frac{100}{s^2 + 10s + 100}$$

b. LP – HP(normalized) Transform

$$s \to \frac{\Omega_u}{s} = \frac{1}{s}$$

$$sub \quad H_a(s) = H(s)\big|_{s \to \frac{1}{s}} = \frac{1}{\left((\frac{1}{s})^2 + (\frac{1}{s}) + 1\right)} \qquad = \frac{s^2}{s^2 + s + 1}$$

c. LP – HP(specified cutoff) Transform

$$s \to \frac{\Omega_u}{s} = \frac{10}{s}$$

$$sub \quad H_a(s) = H(s)\big|_{s \to \frac{10}{s}} = \frac{1}{\left((\frac{10}{s})^2 + (\frac{10}{s}) + 1\right)}$$

d. LP – BP Transform

replace

$$s \to \frac{s^2 + \Omega_u \Omega_l}{s(\Omega_u - \Omega_l)} = \frac{s^2 + \Omega_o^2}{sB_0} \quad where \quad \Omega_o = \sqrt{\Omega_u \Omega_l}$$

$$and \quad B_o = (\Omega_u - \Omega_l)$$

$$sub \quad H_a(s) = H(s)\big|_{s \to \frac{s^2+10^4}{10s}}$$

$$= \frac{100s^2}{s^4 + 10s^3 + 20100s^2 + 10^5 s + 10^8}$$

e. LP – BS Transform

replace

$$s \to \frac{s(\Omega_u - \Omega_l)}{s^2 + \Omega_u \Omega_l} = \frac{sB_0}{s^2 + \Omega_o^2} \quad where \quad \Omega_o = \sqrt{\Omega_u \Omega_l}$$

$$and \quad B_o = (\Omega_u - \Omega_l)$$

$$sub \quad H_a(s) = H(s)\big|_{s \to \frac{2s}{s^2+100}}$$

$$= \frac{(s^2 + 100)^2}{s^4 + 2s^3 + 204s^2 + 200s + 10^4}$$

2.

Convert single pole LP Bufferworth filter with system function $H(z)=\frac{0.245(1+z^{-1})}{1+0.509z^{-1}}$ **into BPF with upper & lower cutoff frequency** ω_u & ω_l **respectively , The LPF has 3-dB bandwidth** $\omega_p = 0.2\pi$ **(Nov-**

Solution:

$$z^{-1} \rightarrow \frac{z^{-2}-\alpha_1 z^{-1}+\alpha_2}{\alpha_2 z^{-2}-\alpha_1 z^{-1}+1}$$

applying this to the given transfer function,

$$H(Z)=\frac{0.245\left(1+\frac{z^{-2}-\alpha_1 z^{-1}+\alpha_2}{\alpha_2 z^{-2}-\alpha_1 z^{-1}+1}\right)}{1+0.509\left(\frac{z^{-2}-\alpha_1 z^{-1}+\alpha_2}{\alpha_2 z^{-2}-\alpha_1 z^{-1}+1}\right)}$$

$$H[z]=\frac{0.245(1-\alpha_2)(1-z^{-2})}{(1+0.509\alpha_2)-1.509\alpha_1 z^{-1}+(\alpha_2+0.509)z^{-2}}$$

Note that the resulting filter has zeros at z=±1 and a pair of poles that depend on the choice of ωl and ωu

Ex : $\omega_u=\frac{3\pi}{5}$ $\omega_l=\frac{2\pi}{5}$

$\omega_p = 0.2\pi$

Then k=1, $\alpha_2=0, \alpha_1=0$

$$\therefore H[z]=\frac{0.245\,(1-z^{-2})}{1+0.509\,z^{-2}}$$

This filter has poles at z=±j0.713 and hence resonates at ω=π/2

The following observations are made,

- It is shown here that how easy to convert one form of filter design to another form.
- What we require is only prototype low pass filter design steps to transform to any other form.

UNIT 4

FIR FILTERS

4.1 INTRODUCTION

The FIR Filters can be easily designed to have perfectly linear Phase. These filters can be realized recursively and Non-recursively. There is greater flexibility to control the Shape of their Magnitude response. Errors due to round off noise are less severe in FIR Filters, mainly because Feedback is not used.

4.2 FEATURES OF FIR FILTER:

1. FIR filter always provides linear phase response. This specifies that the signals in the pass band will suffer no dispersion Hence when the user wants no phase distortion, then FIR filters are preferable over IIR. Phase distortion always degrades the system performance. In various applications like speech processing, data transmission over long distance FIR filters are more preferable due to this characteristic.

2. FIR filters are most stable as compared with IIR filters due to its non feedback nature.

3. Quantization Noise can be made negligible in FIR filters. Due to this sharp cutoff FIR filters can be easily designed.

4. Disadvantage of FIR filters is that they need higher ordered for similar magnitude response of IIR filters.

4.3 FIR SYSTEM ARE ALWAYS STABLE. Why?

Proof: Difference equation of FIR filter of length M is given as

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

And the coefficient bk are related to unit sample response as

$H(n) = b_n$ for $0 \le n \le M-1$
$= 0$ otherwise.

We can expand this equation as

$Y(n)= b_0 x(n) + b_1 x(n-1) + \ldots\ldots + b_{M-1} x(n-M+1)$

System is stable only if system produces bounded output for every bounded input. This is stability definition for any system.
Here h(n)={b0, b1, b2, } of the FIR filter are stable. Thus y(n) is bounded if input x(n) is

bounded. This means FIR system produces bounded output for every bounded input. Hence FIR systems are always stable.

4.4 SYMMETRIC AND ANTI-SYMMETRIC FIR FILTERS:

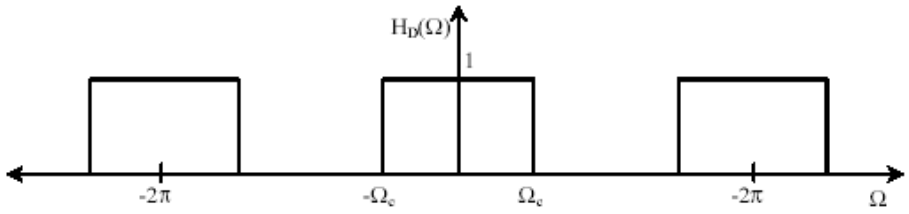
1. Unit sample response of FIR filters is symmetric if it satisfies following condition.
h(n)= h(M-1-n) n=0,1,2…….M-1
2. Unit sample response of FIR filters is Anti-symmetric if it satisfies following condition
h(n)= -h(M-1-n) n=0,1,2…….M-1

4.5 FIR FILTER DESIGN METHODS:

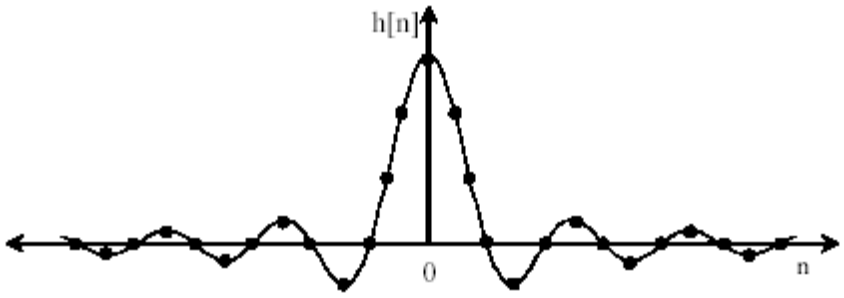
The various method used for FIR Filer design are as follows
1. Fourier Series method
2. Windowing Method
3. DFT method
4. Frequency sampling Method. (IFT Method)

GIBBS PHENOMENON:

Consider the ideal LPF frequency response as shown in Fig 1 with a normalizing angular cut off frequency Ωc.



Impulse response of an ideal LPF is as shown in Fig 2.

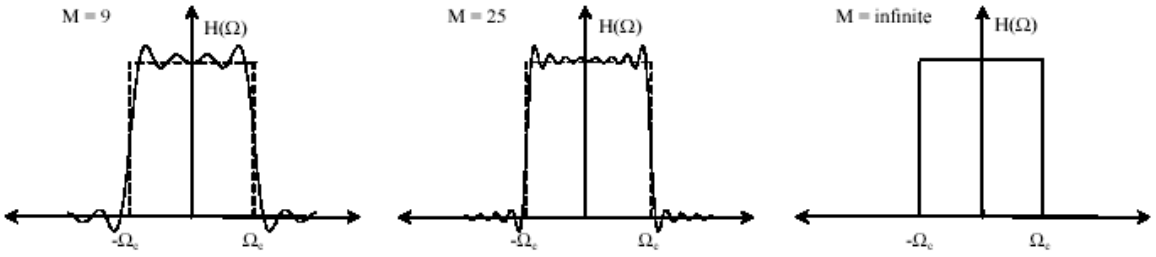


1. In Fourier series method, limits of summation index is -∞ to ∞. But filter must have finite terms.Hence limit of summation index change to -Q to Q where Q is some finite integer. But this type of truncation may result in poor convergence of the series. Abrupt truncation of infinite

series is equivalent to multiplying infinite series with rectangular sequence. i.e at the point of discontinuity some oscillation may be observed in resultant series.
2. Consider the example of LPF having desired frequency response Hd (ω) as shown in figure. The oscillations or ringing takes place near band-edge of the filter.

3. This oscillation or ringing is generated because of side lobes in the frequency response W(ω) of the window function. This oscillatory behavior is called "Gibbs Phenomenon".

Truncated response and ringing effect is as shown in fig 3.



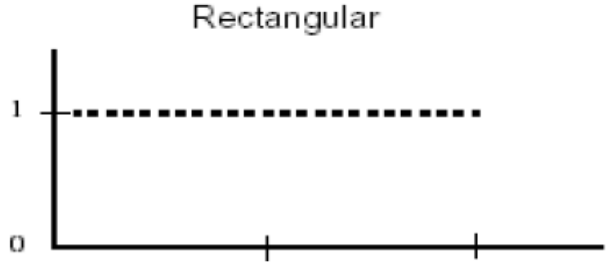
4.6 WINDOWING TECHNIQUE:

Windowing is the quickest method for designing an FIR filter. A windowing function simply truncates the ideal impulse response to obtain a causal FIR approximation that is non causal and infinitely long. Smoother window functions provide higher out-of band rejection in the filter response.

However this smoothness comes at the cost of wider stopband transitions.Various windowing method attempts to minimize the width of the main lobe (peak) of the frequency response. In addition, it attempts to minimize the side lobes (ripple) of the frequency response.

$$\text{since } h[n] = h_d[n]w[n],$$
$$H(e^{j\omega}) = H_d(e^{j\omega}) * W(e^{j\omega}).$$

Rectangular Window: Rectangular This is the most basic of windowing methods. It does not require any operations because its values are either 1 or 0. It creates an abrupt discontinuity that results in sharp roll-offs but large ripples.

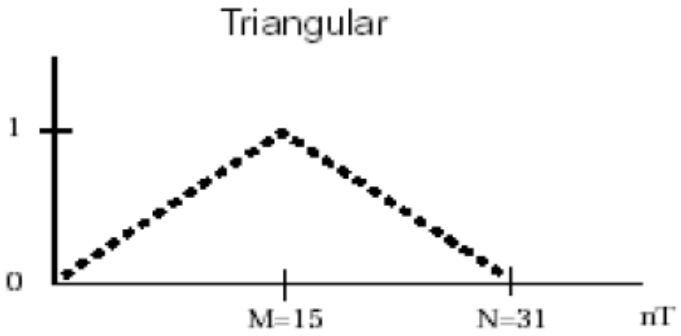


Rectangular window is defined by the following equation.

- **Rectangular:**

$$w[n] = \begin{cases} 1 & 0 \le n \le N \\ 0 & \text{otherwise} \end{cases}$$

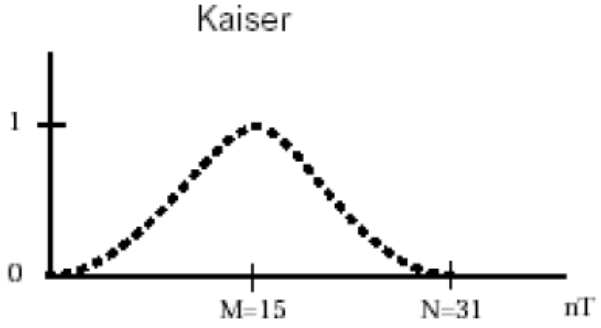
Triangular Window: The computational simplicity of this window, a simple convolution of two rectangle windows, and the lower sidelobes make it a viable alternative to the rectangular window.



- **Bartlett (triangular):**

$$w[n] = \begin{cases} 2n/N & 0 \le n \le N/2 \\ 2 - 2n/N & N/2 < n \le N \\ 0 & \text{otherwise} \end{cases}$$

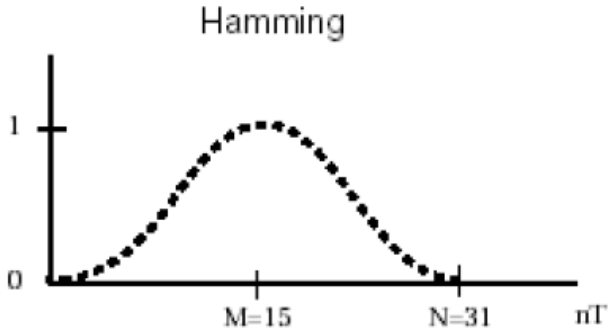
Kaiser Window: This windowing method is designed to generate a sharp central peak. It has reduced side lobes and transition band is also narrow. Thus commonly used in FIR filter design.



- **Kaiser:**

$$w[n] = \begin{cases} I_0[\beta(1 - [(n-\alpha)/\alpha]^2)^{1/2}] & 0 \le n \le N \\ 0 & \text{otherwise} \end{cases}$$

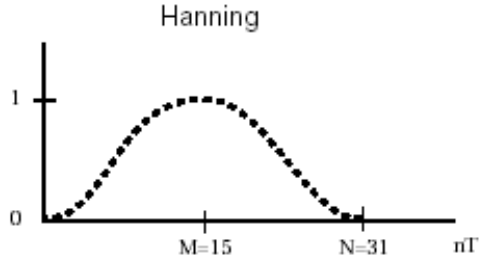
Hamming Window: This windowing method generates a moderately sharp central peak. Its ability to generate a maximally flat response makes it convenient for speech processing filtering.



- **Hamming:**

$$w[n] = \begin{cases} 0.54 - 0.46\cos(2\pi n/N) & 0 \le n \le N \\ 0 & \text{otherwise} \end{cases}$$

Hanning Window: This windowing method generates a maximum flat filter design.



- **Hanning:**

$$w[n] = \begin{cases} 0.5 - 0.5\cos(2\pi n/N) & 0 \le n \le N \\ 0 & \text{otherwise} \end{cases}$$

WINDOWING FUNCTIONS FOR RECTANGULAR,HANNING,HAMMING,BLACKMAN WINDOWS

Name of window function $w(n)$	Mathematical definition
Rectangular	1
Hanning	$0.5 - 0.5\cos\left[\frac{2\pi n}{N-1}\right]$
Hamming	$0.54 - 0.46\cos\left[\frac{2\pi n}{N-1}\right]$
Blackman	$0.42 - 0.5\cos\left[\frac{2\pi n}{N-1}\right] + 0.08\cos\left[\frac{2\pi n}{N-1}\right]$

Window	Peak sidelobe amplitude (dB)	Mainlobe transition width	Peak approximation error (dB)
Rectangular	−13	$4\pi/(N+1)$	−21
Bartlett	−25	$8\pi/N$	−25
Hanning	−31	$8\pi/N$	−44
Hamming	−41	$8\pi/N$	−53

4.7 DESIGNING FILTER FROM POLE ZERO PLACEMENT:

Filters can be designed from its pole zero plot. Following two constraints should be imposed while designing the filters.

1. All poles should be placed inside the unit circle on order for the filter to be stable. However zeros can be placed anywhere in the z plane. FIR filters are all zero filters hence they are always stable. IIR filters are stable only when all poles of the filter are inside unit circle.

2. All complex poles and zeros occur in complex conjugate pairs in order for the filter coefficients to be real.

In the design of low pass filters, the poles should be placed near the unit circle at points corresponding to low frequencies (near ω=0)and zeros should be placed near or on unit circle at points corresponding to high frequencies (near ω=Π). The opposite is true for high pass filters.

4.8 FREQUENCY SAMPLING METHOD FOR DESIGNING FIR DIGITAL FILTERS:

In this design method, the desired frequency response $H_d(e^{j\omega})$ is sampled at equally-spaced points, and the result is inverse discrete Fourier transformed.

Specifically, letting

$$H[k] = H_d(e^{j\omega})\big|_{\omega=\frac{2\pi k}{N}}, \qquad k = 0, \ldots, N-1,$$

the unit sample response of the filter is $h[n] = \text{IDFT}(H[k])$, so

$$h[n] = \frac{1}{N}\sum_{k=0}^{N-1} H[k]e^{j2\pi nk/N}.$$

The resulting filter will have a frequency response that is exactly the same as the original response at the sampling instants. Note that it is also necessary to specify the *phase* of the desired response $H_d(e^{j\omega})$, and it is usually chosen to be a linear function of frequency to ensure a linear phase filter. Additionally, if a filter with real-valued coefficients is required, then additional constraints have to be enforced.

The *actual* frequency response $H(e^{j\omega})$ of the filter $h[n]$ still has to be determined. The z-transform of the impulse response is

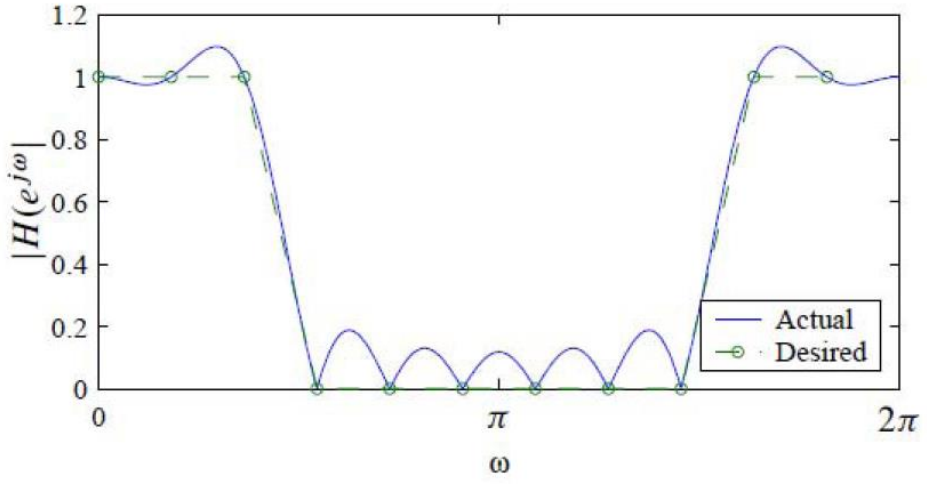
$$\begin{aligned} H(z) &= \sum_{n=0}^{N-1} h[n]z^{-n} = \sum_{n=0}^{N-1}\left[\frac{1}{N}\sum_{k=0}^{N-1} H[k]e^{j2\pi nk/N}\right]z^{-n} \\ &= \frac{1}{N}\sum_{k=0}^{N-1} H[k]\sum_{n=0}^{N-1} e^{j2\pi nk/N}z^{-n} \\ &= \frac{1}{N}\sum_{k=0}^{N-1} H[k]\left[\frac{1-z^{-N}}{1-e^{j2\pi k/N}z^{-1}}\right]. \end{aligned}$$

Evaluating on the unit circle $z = e^{j\omega}$ gives the frequency response

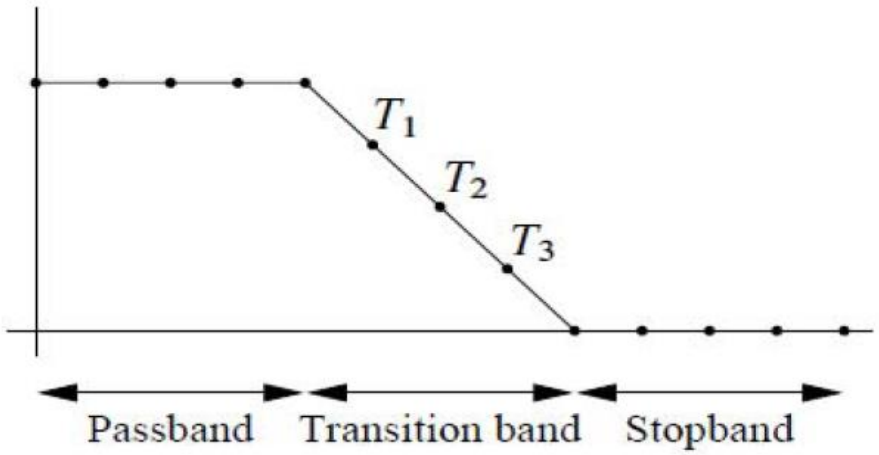
$$H(e^{j\omega}) = \frac{1 - e^{-j\omega N}}{N} \sum_{k=0}^{N-1} \frac{H[k]}{1 - e^{j2\pi k/N} e^{-j\omega}}.$$

This expression can be used to find the actual frequency response of the filter obtained, which can be compared with the desired response.

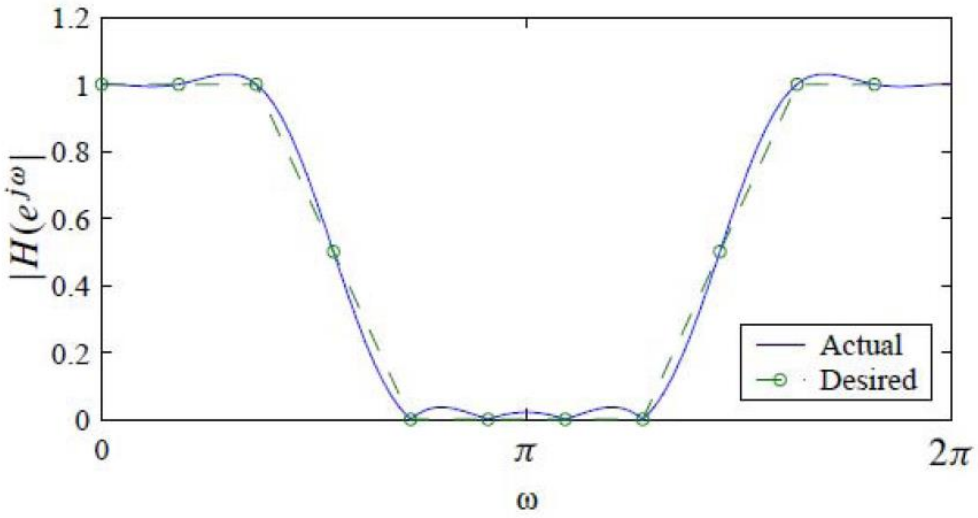
The method described only guarantees correct frequency response values at th points that were sampled. This sometimes leads to excessive ripple at intermediate points:



One way of addressing this problem is to allow **transition samples** in the region where discontinuities in $H_d(e^{j\omega})$ occur:



This effectively increases the transition width and can decrease the ripple, as observed below:



By leaving the value of the transition sample unconstrained, one can to some extent optimise the filter to minimise the ripple. Empirically, with three transition samples a stopband attenuation of 100dB is achievable. Recall however that for $h[n]$ real we require even or odd symmetry in the impulse response, so the values are not entirely unconstrained.

4.9 COMPARISON BETWEEN FIR AND IIR DIGITAL FILTERS:

Sr No	FIR Digital Filter	IIR Digital Filter
1	FIR system has finite duration unit sample response. i.e $h(n) = 0$ for $n<0$ and $n \geq M$ Thus the unit sample response exists for the duration from 0 to M-1.	IIR system has infinite duration unit sample response. i. e $h(n) = 0$ for $n<0$ Thus the unit sample response exists for the duration from 0 to ∞.
2	FIR systems are non recursive. Thus output of FIR filter depends upon present and past inputs.	IIR systems are recursive. Thus they use feedback. Thus output of IIR filter depends upon present and past inputs as well as past outputs
3	Difference equation of the LSI system for FIR filters becomes $y(n)=\sum_{k=0}^{M} b_k\, x(n-k)$	Difference equation of the LSI system for IIR filters becomes $y(n)=-\sum_{k=1}^{N} a_k\, y(n-k)+\sum_{k=0}^{M} b_k\, x(n-k)$
4	FIR systems has limited or finite memory requirements.	IIR system requires infinite memory.
5	FIR filters are always stable	Stability cannot be always guaranteed.
6	FIR filters can have an exactly linear phase response so that no phase distortion is introduced in the signal by the filter.	IIR filter is usually more efficient design in terms of computation time and memory requirements. IIR systems usually requires less processing time and storage as compared with FIR.
7	The effect of using finite word length to implement filter, noise and quantization errors are less severe in FIR than in IIR.	Analogue filters can be easily and readily transformed into equivalent IIR digital filter. But same is not possible in FIR because that have no analogue counterpart.
8	All zero filters	Poles as well as zeros are present.
9	FIR filters are generally used if no phase distortion is desired. Example: System described by $Y(n) = 0.5\, x(n) + 0.5\, x(n-1)$ is FIR filter. $h(n)=\{0.5,0.5\}$	IIR filters are generally used if sharp cutoff and high throughput is required. Example: System described by $Y(n) = y(n-1) + x(n)$ is IIR filter. $h(n)=a^{n} u(n)$ for $n\geq 0$

FIR Systems are represented in four different ways

1. Direct Form Structures
2. Cascade Form Structure
3. Frequency-Sampling Structures
4. Lattice structures.

1. DIRECT FORM STRUCTURE OF FIR SYSTEM

The convolution of h(n) and x(n) for FIR systems can be written as

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

The above equation can be expanded as,

$$Y(n)= h(0)\, x(n) + h(1)\, x(n-1) + h(2)\, x(n-2) + \ldots\ldots\ldots\ldots + h(M-1)\, x(n-M+1) \qquad (2)$$

Implementation of direct form structure of FIR filter is based upon the above equation.

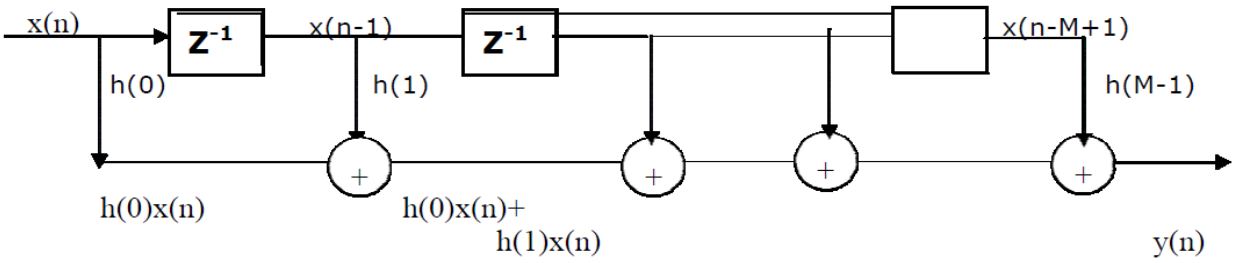


FIG - DIRECT FORM REALIZATION OF FIR SYSTEM

1) There are M-1 unit delay blocks. One unit delay block requires one memory location. Hence direct form structure requires M-1 memory locations.

2) The multiplication of h(k) and x(n-k) is performed for 0 to M-1 terms. Hence M multiplications and M-1 additions are required.

3) Direct form structure is often called as transversal or tapped delay line filter.

2. CASCADE FORM STRUCTURE OF FIR SYSTEM

In cascade form, stages are cascaded (connected) in series. The output of one system is input to another. Thus total K number of stages are cascaded. The total system function 'H' is given by

$$H = H_1(z) \cdot H_2(z) \ldots\ldots\ldots\ldots\ldots\ldots H_k(z) \quad (1)$$

$$H = Y_1(z)/X_1(z) \cdot Y_2(z)/X_2(z) \cdot \ldots\ldots\ldots\ldots Y_k(z)/X_k(z) \quad (2)$$

$$H(z) = \prod_{k=1}^{k} H_k(z) \quad (3)$$

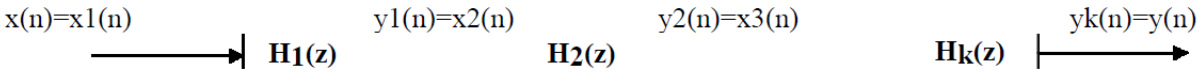


FIG- CASCADE FORM REALIZATION OF FIR SYSTEM

Each H1(z), H2(z)… etc is a second order section and it is realized by the direct form as shown in below figure.

System function for FIR systems

$$H(z) = \sum_{k=0}^{M-1} b_k z^{-k} \quad (1)$$

Expanding the above terms we have

$$H(z) = H_1(z) \cdot H_2(z) \ldots\ldots\ldots\ldots\ldots\ldots H_k(z)$$

where $H_K(z) = b_{k0} + b_{k1} z^{-1} + b_{k2} z^{-2}$ (2)

Thus Direct form of second order system is shown as

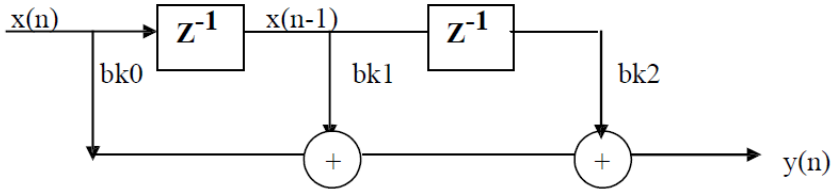


FIG - DIRECT FORM REALIZATION OF FIR SECOND ORDER SYSTEM

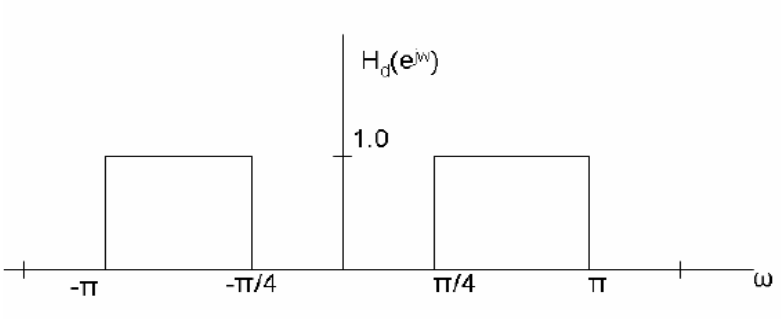
PROBLEMS

1. Design an ideal high-pass filter with a frequency response using a hanning window with M = 11 and plot the frequency response. (12)(Nov-07,09)

$$H_d(e^{j\omega}) = 1 \qquad \text{for } \frac{\pi}{4} \le |\omega| \le \pi$$

$$= 0 \qquad |\omega| < \frac{\pi}{4}$$

solution:



$$h_d(n) = \frac{1}{2\pi}\left[\int_{-\pi}^{-\pi/4} e^{j\omega n} d\omega + \int_{\pi/4}^{\pi} e^{j\omega n} d\omega\right]$$

$$h_d(n) = \frac{1}{\pi n}\left[\sin \pi n - \sin \frac{\pi n}{4}\right] \quad for \quad -\infty \le n \le \infty \quad and \quad n \ne 0$$

$$h_d(0) = \frac{1}{2\pi}\left[\int_{-\pi}^{-\pi/4} d\omega + \int_{\pi/4}^{\pi} d\omega\right] = \frac{3}{4} = 0.75$$

$h_d(1) = h_d(-1) = -0.225$

$h_d(2) = h_d(-2) = -0.159$

$h_d(3) = h_d(-3) = -0.075$

$h_d(4) = h_d(-4) = 0$

$h_d(5) = h_d(-5) = 0.045$

The hamming window function is given by

$$w_{hm}(n) = 0.5 + 0.5\cos\frac{2\pi n}{M-1} \qquad -(\frac{M-1}{2}) \le n \le (\frac{M-1}{2})$$
$$= 0 \qquad otherwise$$

$for \quad N = 11$

$$w_{hm}(n) = 0.5 + 0.5\cos\frac{\pi n}{5} \qquad -5 \le n \le 5$$

$w_{hn}(0) = 1$

$w_{hn}(1) = w_{hn}(-1) = 0.9045$

$w_{hn}(2) = w_{hn}(-2) = 0.655$

$w_{hn}(3) = w_{hn}(-3) = 0.345$

$w_{hn}(4) = w_{hn}(-4) = 0.0945$

$w_{hn}(5) = w_{hn}(-5) = 0$

$h(n) = w_{hn}(n)h_d(n)$

h(n)=[0 0 -0.026 -0.104 -0.204 0.75 -0.204 -0.104 -0.026 0 0]

2. Design a filter with a frequency response: using a Hanning window with M = 7

$$H_d(e^{j\omega}) = e^{-j3\omega} \qquad for -\frac{\pi}{4} \le \omega \le \frac{\pi}{4}$$
$$= 0 \qquad \frac{\pi}{4} < |\omega| \le \pi$$

(8)(Apr/08)

Solution:

The freq resp is having a term $e^{-j\omega(M-1)/2}$ which gives h(n) symmetrical about n = M-1/2 = 3 i.e we get a causal sequence.

$$h_d(n) = \frac{1}{2\pi}\int_{-\pi/4}^{\pi/4} e^{-j3\omega} e^{j\omega n} d\omega$$

$$= \frac{\sin\frac{\pi}{4}(n-3)}{\pi(n-3)}$$

$this \quad gives \quad h_d(0) = h_d(6) = 0.075$

$h_d(1) = h_d(5) = 0.159$

$h_d(2) = h_d(4) = 0.22$

$h_d(3) = 0.25$

The Hanning window function values are given by

$w_{hn}(0) = w_{hn}(6) = 0$

$w_{hn}(1) = w_{hn}(5) = 0.25$

$w_{hn}(2) = w_{hn}(4) = 0.75$

$w_{hn}(3) = 1$

$h(n) = h_d(n)\, w_{hn}(n)$

h(n)=[0 0.03975 0.165 0.25 0.165 0.3975 0]

3. Design a LP FIR filter using Freq sampling technique having cutoff freq of π/2 rad / sample. The filter should have linear phase and length of 17. (12)(May-07)

The desired response can be expressed as

$$H_d(e^{j\omega}) = e^{-j\omega(\frac{M-1}{2})} \quad for \quad |\omega| \le \omega c$$
$$= 0 \quad otherwise$$
$$with \quad M = 17 \quad and \quad \omega c = \pi/2$$

$$H_d(e^{j\omega}) = e^{-j\omega 8} \quad for \quad 0 \le \omega \le \pi/2$$
$$= 0 \quad for \quad \pi/2 \le \omega \le \pi$$

$$\text{Selecting } \omega_k = \frac{2\pi k}{M} = \frac{2\pi k}{17} \quad for \quad k = 0,1,.....16$$

$$H(k) = H_d(e^{j\omega})\big|_{\omega=\frac{2\pi k}{17}}$$

$$H(k) = e^{-j\frac{2\pi k}{17}8} \quad for \quad 0 \le \frac{2\pi k}{17} \le \frac{\pi}{2}$$
$$= 0 \quad for \quad \pi/2 \le \frac{2\pi k}{17} \le \pi$$

$$H(k) = e^{-j\frac{16\pi k}{17}} \qquad for \quad 0 \le k \le \frac{17}{4}$$

$$= 0 \qquad for \quad \frac{17}{4} \le k \le \frac{17}{2}$$

The range for "k" can be adjusted to be an integer such as

$0 \le k \le 4$

and $5 \le k \le 8$

The freq response is given by

$$H(k) = e^{-j\frac{2\pi k}{17}8} \qquad for \quad 0 \le k \le 4$$

$$= 0 \qquad for \qquad 5 \le k \le 8$$

Using these value of H(k) we obtain h(n) from the equation

$$h(n) = \frac{1}{M}(H(0) + 2\sum_{k=1}^{(M-1)/2} \mathrm{Re}(H(k)e^{j2\pi kn/M}))$$

i.e.,

$$h(n) = \frac{1}{17}(1 + 2\sum_{k=1}^{4} \mathrm{Re}(e^{-j16\pi k/17} e^{j2\pi kn/17}))$$

$$h(n) = \frac{1}{17}(H(0) + 2\sum_{k=1}^{4} \cos(\frac{2\pi k(8-n)}{17}) \quad for \quad n = 0,1,.....16$$

- Even though k varies from 0 to 16 since we considered ω varying between 0 and π/2 only k values from 0 to 8 are considered
- While finding h(n) we observe symmetry in h(n) such that n varying 0 to 7 and 9 to 16 have same set of h(n)

UNIT-5
MULTIRATE SIGNAL PROCESSING

INTRODUCTION:

Multirate means "multiple sampling rates". A multirate DSP system uses multiple sampling rates within the system. Whenever a signal at one rate has to be used by a system that expects a different rate, the rate has to be increased or decreased, and some processing is required to do so. Therefore "Multirate DSP" really refers to the art or science of changing sampling

Need of Multirate DSP:

The most immediate reason is when you need to pass data between two systems which use incompatible sampling rates. For example, professional audio systems use 48 kHz rate, but consumer CD players use 44.1 kHz; when audio professionals transfer their recorded music to CDs, they need to do a rate conversion.But the most common reason is that multirate DSP can greatly increase processing efficiency (even by orders of magnitude!), which reduces DSP system cost. This makes the subject of multirate DSP vital to all professional DSP practitioners

Categories of Multirate: Multirate consists of:

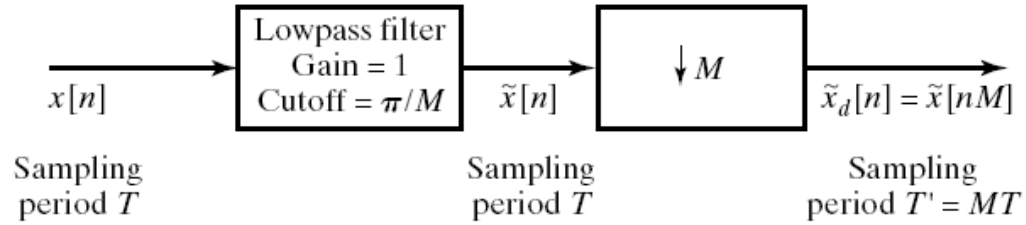
1. **Decimation:** To decrease the sampling rate,
2. **Interpolation:** To increase the sampling rate,
3. **Resampling:**To combine decimation and interpolation in order to change the sampling rate by a fractional value that can be expressed as a ratio. For example, to resample by a factor of 1.5, you just interpolate by a factor of 3 then decimate by a factor of 2 (to change the sampling rate by a factor of 3/2=1.5.)

APPLICATIONS:

1. Used in A/D and D/A converters.
2. Used to change the rate of a signal. When two devices that operate at different rates are to be interconnected, it is necessary to use a rate changer between them.
3. In transmultiplexers
4. In speech processing to reduce the storage space or the transmitting rate of speech data.
5. Filter banks and wavelet transforms depend on multi rate methods.

DOWN SAMPLING:

The process of reducing a sampling rate by an integer factor is referredto as down sampling of a data sequence. We also refer to down sampling as "decimation". To down sample a data sequence $x(n)$ by an integer factor of M, we use the following notation:



$$y(m) = x(mM)$$

Where $y(m)$ is the down sampled sequence, Obtained by taking a sample from the data sequence $x(n)$ for every M samples (discarding $M - 1$ samples for every M samples). As an example, if the original sequence with a sampling period $T = 0.1$ second (sampling rate = 10 samples per sec) is given by

Consider $x(n)$:8 7 4 8 9 6 4 2 –2 –5 –7 –7 –6 –4 …
and we down sample the data sequence by a factor of 3, we obtain the down sampled sequence as

$$y(m):8 \quad 8 \quad 4 \quad -5 \quad -6 \ldots,$$

with the resultant sampling period $T = 3 \times 0.1 = 0.3$ second (the sampling rate now is 3.33 samples per second).

From the Nyquist sampling theorem, it is known that aliasing can occur in the down sampled signal due to the reduced sampling rate. After down sampling by a factor of M, the new sampling period becomes MT, and therefore the new sampling frequency is

$$f_{sM} = 1/(MT) = f_s/M,$$

where f_s is the original sampling rate.

Hence, the folding frequency after down sampling becomes

$$f_{sM}/2 = f_s/(2M).$$

This tells us that after down sampling by a factor of M, the new folding frequency will be decreased M times. If the signal to be down sampled has frequency components larger than the new folding frequency, $f > f_s/(2M)$, aliasing noise will be introduced into the down sampled data.

To overcome this problem, it is required that the original signal $x(n)$ be processed by a low pass filter $H(z)$ before down sampling, which should have a stop frequency edge at $f_s/(2M)$ (Hz). The corresponding normalized stop frequency edge is then converted to be

$$\Omega_{stop} = 2\pi\ (f_s/(2M))\ T = \pi/M \text{ radians.}$$

In this way, before down sampling, we can guarantee that the maximum frequency of the filtered signal satisfies $f_{max} < f_s/(2M)$,

such that no aliasing noise is introduced after down sampling. A general block diagram of decimation is given in Figure, where the filtered output in terms of the z-transform can be written as $W(z) = H(z)X(z)$,

where $X(z)$ is the z-transform of the sequence to be decimated,$x(n)$, and $H(z)$ is the lowpass filter transfer function. After anti-aliasing filtering, the down sampled signal $y(m)$ takes its value from the filter output as: $y(m) = w(mM)$.

The process of reducing the sampling rate by a factor of 3 is shown in Figure The corresponding spectral plots for $x(n)$,$w(n)$, and $y(m)$ in general are shown in Figure

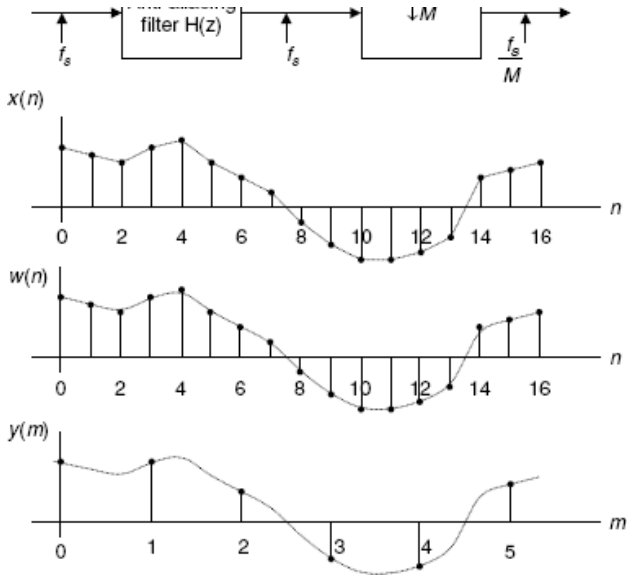


Figure. Block diagram of the downsampling process with M = 3

Example:1: If x(n) = {1, -1, 2, 4, 0, 3, 2, 1, 5,….}

Then y(m)= x(mM) for M = 2 is

Y (m) = {1, 2, 0, 2, 5,….}

i.e if we left M-1 samples inbetween samples of x(n) to generate y(m).

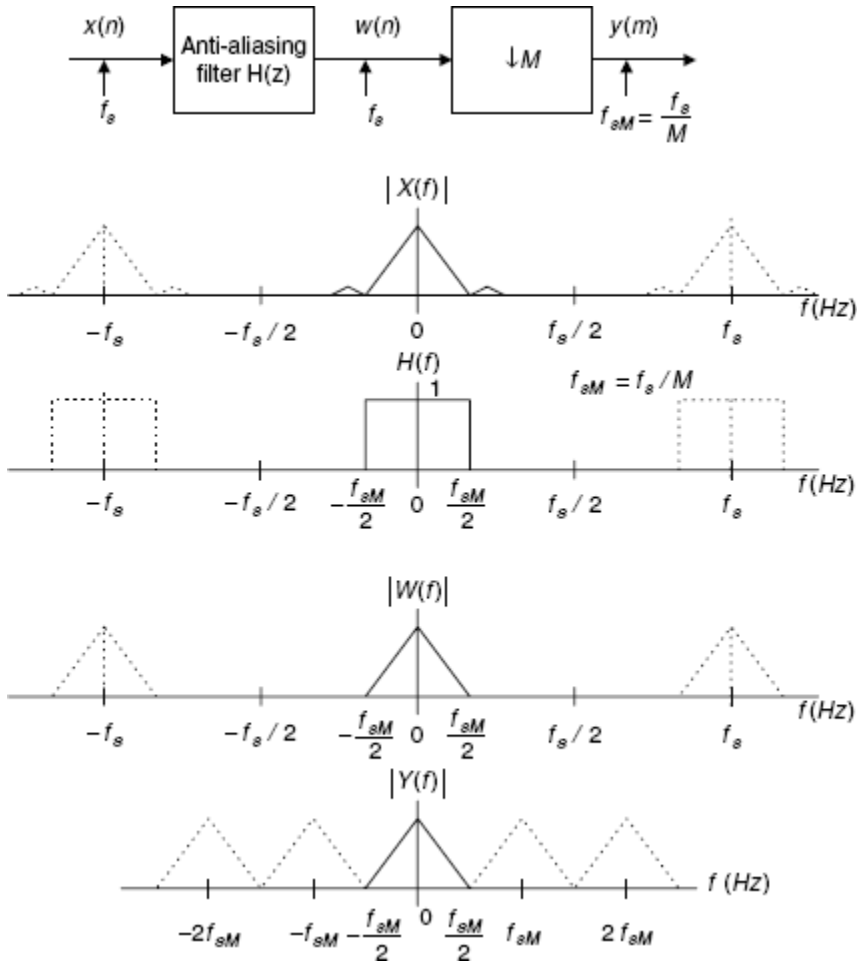
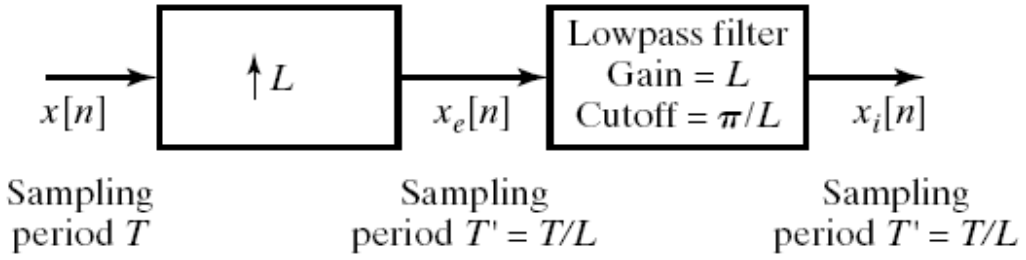


Figure Spectrum after down sampling.

UP-SAMPLER :

Increasing a sampling rate is a process of upsampling by an integer factor of *L*. This process is described as follows:



$$y(m) = x(m/L)$$

where $n = 0, 1, 2, \ldots$, $x(n)$ is the sequence to be up sampled by a factor of L, and $y(m)$ is the up sampled sequence. As an example, suppose that the data sequence is given as follows:

$x(n)$: 8 8 4 –5 –6 …

After up sampling the data sequence $x(n)$ by a factor of 3 (adding L– 1 zeros for each sample), we have the up sampled data sequence $w(m)$ as:

$w(m)$: 8 0 0 8 0 0 4 0 0 –5 0 0 –6 0 0 …

The next step is to smooth the up sampled data sequence via an interpolation filter. The process is illustrated in Figure

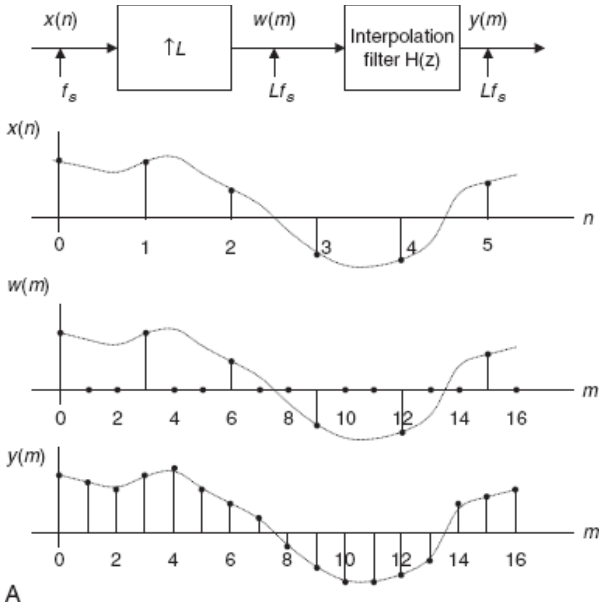


Figure Block diagram for the upsampling process with L = 3.

Similar to the downsampling case, assuming that the data sequence has the current sampling period of T, the Nyquist frequency is given by $f_{max} = f_s/2$. After upsampling by a factor of$L$, the new sampling period becomes T/L, thus the new sampling frequency is changed to be

$$f_{sL} = Lf_s. \quad (12.10)$$

This indicates that after up sampling, the spectral replicas originally centered at f_s, $2f_s$, … are included in the frequency range from 0 Hz to the new Nyquist limit Lf_s=2 Hz, as shown in Figure. To remove those included spectral replicas, an interpolation filter with a stop frequency edge of f_s=2 in Hz must be attached, and the normalized stop frequency edge is given by

$$\Omega_{stop} = 2\pi\,(f_s/2) \times (T/L) = \pi/L \text{ radians.}$$

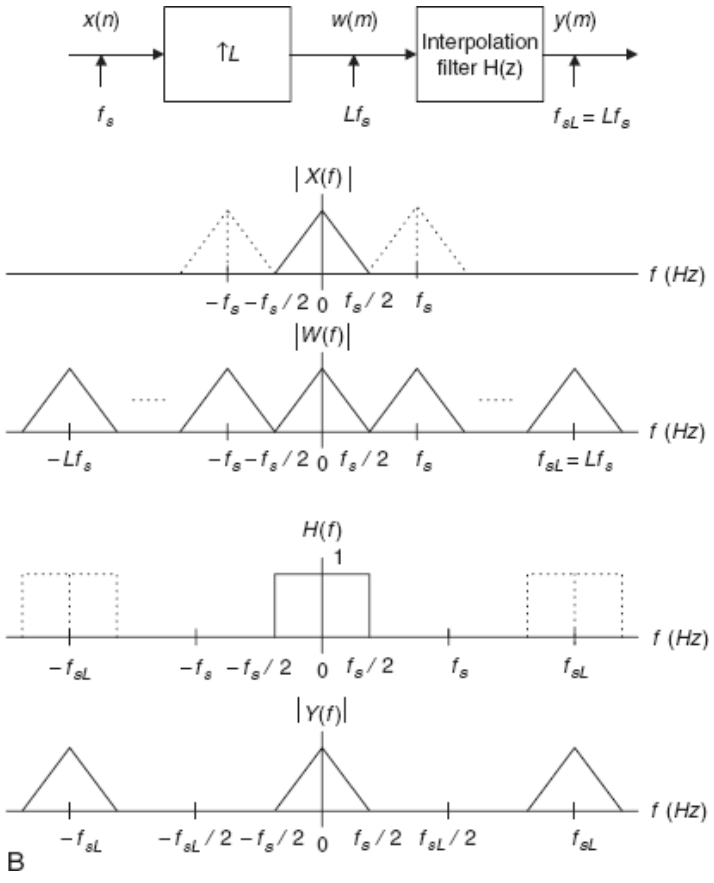


Figure Spectra before and after upsampling.

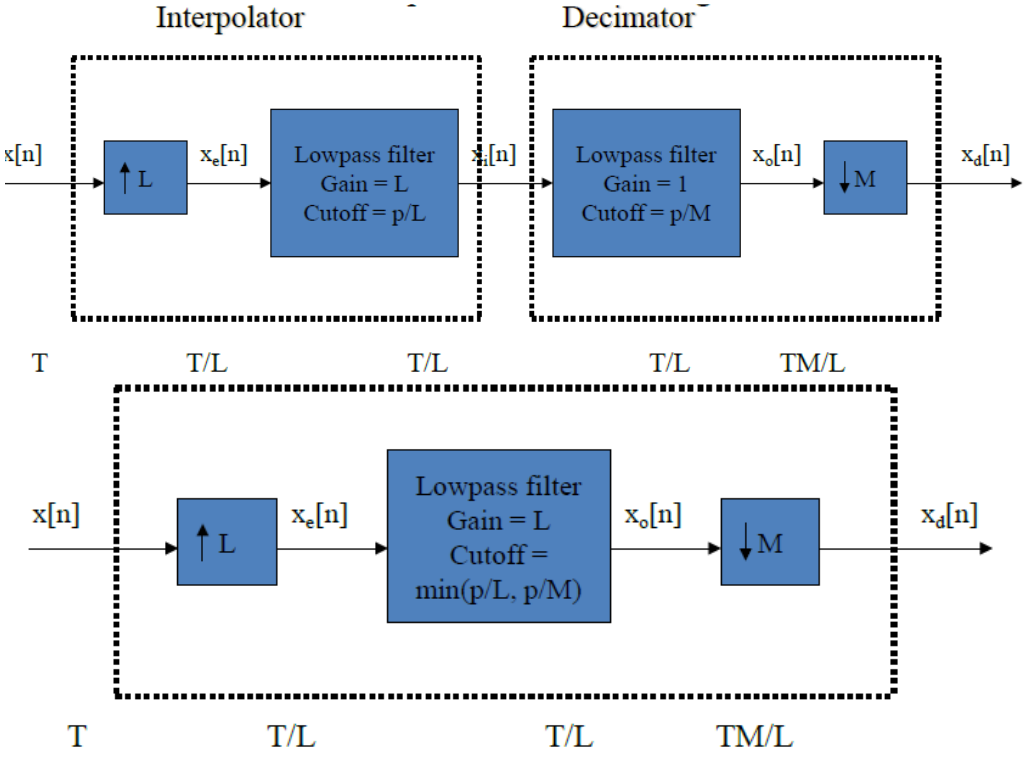
After filtering via the interpolation filter, we will achieve the desired spectrum for *y*(*n*), as shown in Figure 5.2.b. Note that since the interpolation is to remove the high-frequency images that are aliased by the upsampling operation, it is essentially an anti-aliasing lowpass filter.

Example: If x(n) = {1, -1, 2, 4, 3, ,….}
Then y(m)= x(m/L) for L = 3 is
Y (m) = {1, 0, 0,-1,0, 0,2,0, 0, 4, 0, 0, 3, 0, 0,,….}

SAMPLE RATE CONVERSION:

In Decimation and Interpolation, sampling rate conversion is achieved by Integer Factor. When sampling rate conversion requires by non integer factor, we need to perform sampling rate conversion by rational factor I/D .

- Perform Interpolation by a Factor I.
- Filter the output of interpolator using a Low Pass (Anti Imaging Filter) with the Bandwidth of π/I.
- The output of Anti Imaging Filter is Passed through a another Low Pass Filter (Anti Aliasing Filter) to limit the bandwidth of signal to π/D.
- Finally Signal Band limited to π/D is decimated by factor D.



- The anti Imaging Filter and anti Aliasing Filter are operated at same sampling rate and hence can be replaced by simple lowpass filter with cut off frequency,

$$W_c = \min[\pi/I, \pi/D]$$

It is Important to note that, in order to preserve the spectral characteristics of x(n), the interpolation has to be performed first and decimation is to performed next

Example: Show that the upsampler and down sampler are time variant systems.

Consider a factor of L upsampler defined by

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

The o/p due to delayed i/p is

$y(n, k) = x(n/L - k)$

the delayed output is

$y(n-k) = x[(n-k)/L]$

$y(n, k) \neq y(n-k)$

therefore up sampler is a time variant systems.

Similarly for down sampler

$Y(n) = x(nM)$

$y(n,k) = x(nM-k)$

$y(n-k) = x(M(n-k))$

$y(n, k) \neq y(n-k)$

Therefore down sampler is a time variant systems.

FINITE WORD LENGTH EFFECTS

5.6 NUMBER REPRESENTATION:

In digital signal processing, (B +1)-bit fixed-point numbers are usually represented as two's-complementsigned fractions in the format bo b-ib-2 …… b-B

The number represented is then

$$X = -bo + b\text{-}i2^{-1} + b\text{-}22^{-2} + \bullet\bullet\bullet + b\text{-}B\ 2^{-B}$$

where bo is the sign bit and the number range is —1 <X <1. The advantage of this representation is that the product of two numbers in the range from — 1 to 1 is another number in the same range. Floating-point numbers are represented as

$$X = (-1)^s m 2^c$$

where s is the sign bit, mis the mantissa, and cis the characteristic or exponent.To make the representation of a number unique, the mantissa is normalized so that 0.5 <m <1.

Although floating-point numbers are always represented in the form of , the way in which this representation is actually storedin a machine may differ. Since m >0.5, it is not necessary to store the 2^{-1}-weight bit of m, which is always set. Therefore, in practice numbers are usually stored as

$$X = (-1)^s(0.5 + f)2^c$$

where fis an unsigned fraction, 0 <f <0.5.

Most floating-point processors now use the IEEE Standard 754 32-bit floating point format for storing numbers. According to this standard the exponent is stored as an unsigned integer pwhere

$$p = c + 126$$

Therefore, a number is stored as

$$X = (-1)^s(0.5 + f)2^{p-126}$$

where s is the sign bit, fis a 23-b unsigned fraction in the range 0 <f <0.5, and p is an 8-b unsigned integer in the range 0 <p <255. The total number of bits is 1 + 23 + 8 = 32. For example, in IEEE format 3/4 is written $(-1)^0(0.5 + 0.25)2°$ so s =0, p =126, and f =0.25. The value X =0 is a unique case and is represented by all bits zero (i.e., s = 0, f =0, and p =0). Although the 2^{-1}-weight mantissa bit is not actually stored, it does exist so the mantissa has 24 b plus a sign bit.

5.7 FIXED-POINT QUANTIZATION ERRORS :

In fixed-point arithmetic, a multiply doubles the number of significant bits. For example, the product of the two 5-b numbers 0.0011 and 0.1001 is the 10-b number 00.000 110 11. The extra bit to the left of the decimal point can be discarded without introducing any error. However, the

least significant four of the remaining bits must ultimately be discarded by some form of quantization so that the result can be stored to 5 b for use in other calculations. In the example above this results in 0.0010 (quantization by rounding) or 0.0001(quantization by truncating). When a sum of products calculation is performed, the quantization can be performed either after each multiply or after all products have been summed with double length precision.

We will examine three types of fixed-point quantization—rounding, truncation, and magnitude truncation. If X is an exact value, then the rounded value will be denoted $Q_r(X)$,the truncated value $Q_t(X)$,and the magnitude truncated value $Q_{mt}(X)$.If the quantized value has Bbits to the right of the decimal point, the quantization step size is

$$A = 2^{-B}$$

Since rounding selects the quantized value nearest the unquantized value, it gives a value which is never more than ± A /2 awayfrom the exact value. If we denote the rounding error by

$$fr = Qr(X) - X$$

$$\frac{A}{2} < f_r < \frac{A}{2} - 2$$

Truncation simply discards the low-order bits, giving a quantized value that is always less than or equal to the exact value so

$$-A < f_t < 0$$

Magnitude truncation chooses the nearest quantized value that has a magnitude less than or equal to the exact value so

$$-A < f_{mt} < A$$

The error resulting from quantization can be modeled as a random variable uniformly distributed over the appropriate error range. Therefore, calculations with roundoff error can be considered error-free calculations that have been corrupted by additive white noise. The meanof this noise for rounding is

$$m_{(r} — E\{fr\} = x/\ frdfr — 0$$

where E{}represents the operation of taking the expected value of a random variable. Similarly, the variance of the noise for rounding is

$$a_{\epsilon}^{2} — E\{(fr - m_{\epsilon r})^{2}\} — — \int_{J-A/2}^{A/2} (fr - m_{\epsilon r})^{2}\, dfr = \frac{A^{2}}{12}$$

Like wise for truncation,

$$m_{f_t} = E\{f_t\} = \bar{y}$$

$$a^{2}_{m_{f_{mt}}} = E\{(ft - mft)^{2}\} = \frac{A^{2}}{0}$$

$$E\{f_{mt}\} —$$

And for magnitude truncation,

$$a^{2}_{f\text{-}mt} = E\{(f_{mt} - m_m)^{2}\} = \frac{A^{2}}{}$$

5.8 FLOATING-POINT QUANTIZATION ERRORS:

With floating-point arithmetic it is necessary to quantize after both multiplications and additions. The addition quantization arises because, prior to addition, the mantissa of the smaller number in the sum is shifted right until the exponent of both numbers is the same. In general, this gives a sum mantissa that is too long and so must be quantized. We will assume that quantization in floating-point arithmetic is performed by rounding. Because of the exponent in floating-point arithmetic, it is the relative error that is important. The relative error is defined as

$$e_r = \frac{Qr(X) - X}{XX} = \frac{e_r}{}$$

5.9 ROUNDOFF NOISE:

To determine the roundoff noise at the output of a digital filter we will assume that the noise due to a quantization is stationary, white, and uncorrelated with the filter input, output, and internal

variables. This assumption isgood if the filter input changes from sample to sample in a sufficiently complex manner. It is not valid for zero or constant inputs for which the effects of rounding are analyzed from a limit cycle perspective.

To satisfy the assumption of a sufficiently complex input, roundoff noise in digital filters is often calculated for the case of a zero-mean white noise filter input signal x(n)of variance a^1. This simplifies calculation of the output roundoff noise because expected values of the form E{x(n)x(n — k)}are zero for k =0 and give a^2 when k =0. This approach to analysis has been found to give estimates of the output roundoff noise thatare close to the noise actually observed for other input signals.

Another assumption that will be made in calculating roundoff noise is that the product of two quantization errors is zero. To justify this assumption, consider the case of a 16-b fixed-point processor. In thiscase a quantization error is of the order 2^{-15}, while the product of two quantization errors is of the order 2^{-30}, which is negligible by comparison.

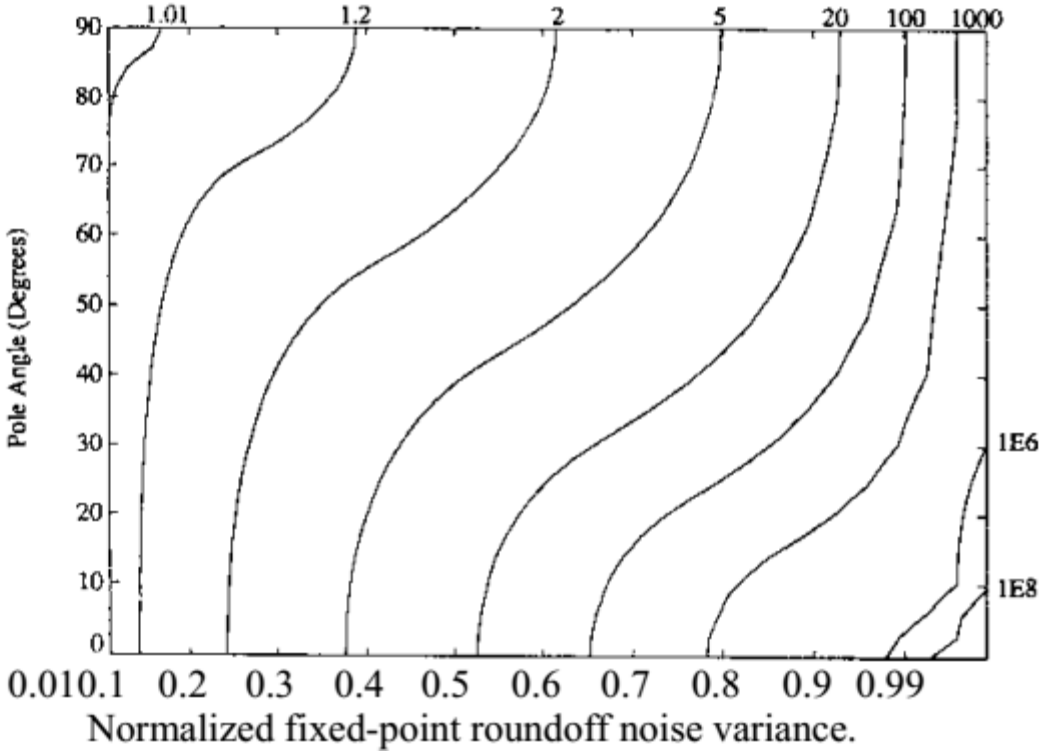
If a linear system with impulse response g(n)is excited by white noise with mean m_x and variance a^2, the output is noise of mean

$$my = mx \sum_{n=-\infty}^{\infty} g(n)$$

And variance

$$ay = al \sum_{n=-\infty}^{\infty} g^2(n)$$

Therefore, if g(n)is the impulse response from the point where a round off takes place to the filter output, the contribution of that round off to the variance (mean square value) of the output round off noise is given by with a^2 replaced with the variance of the round off. If there is more than one source of round off error in the filter, it is assumed that the errors are uncorrelated so the output noise variance is simply the sum of the contributions from each source.



5.10 LIMIT CYCLE OSCILLATIONS:

A limit cycle, sometimes referred to as a multiplier round off limit cycle, is a low level oscillation that can exist in an otherwise stable filter as a result of the nonlinearity associated with rounding (or truncating) internal filter calculations . Limit cycles require recursion to exist and do not occur in non recursive FIR filters. As an example of a limit cycle, consider the second-order filter realized by

$$y(n) = Qr\{ \hat{}y(n — 1) — 8y(n — ^{2}) + ^{x}(n)$$

where Qr{} represents quantization by rounding. This is stable filter with poles at 0.4375 ± j0.6585. Consider the implementation of this filter with 4-b (3-b and a sign bit) two's complement fixed-point arithmetic, zero initial conditions (y(—1) = y(—2) = 0), and an input sequence x(n) =|S(n), where S(n)is the unit impulse or unit sample. The following sequence is obtained.

Notice that while the input is zero except for the first sample, the output oscillates with amplitude 1/8 and period 6. Limit cycles are primarily of concern in fixed-point recursive filters. As long as floating-point filters are realized as the parallel or cascade connection of first- and second-order sub filters, limit cycles will generally not be a problem since limit cycles are practically not observable in first and second-order systems implemented with 32-bit floating-point arithmetic . It has been shown that such systems must have an extremely small margin of

stability for limit cycles to exist at anything other than underflow levels, which are at an amplitude of less than . There are at least three ways of dealing with limit cycles when fixed-point arithmetic is used. One is to determine a bound on the maximum limit cycle amplitude, expressed as an integral number of quantization steps . It is then possible to choose a word length that makes the limit cycle amplitude acceptably low. Alternately, limit cycles can be prevented by randomly rounding calculations up or down. However, this approach is complicated to implement. The third approach is to properly choose the filter realization structure and then quantize the filter calculations using magnitude truncation . This approach has the disadvantage of producing more round off noise than truncation or rounding .

5.11 OVERFLOW OSCILLATIONS:

With fixed-point arithmetic it is possible for filter calculations to overflow. This happens when two numbers of the same sign add to give a value having magnitude greater than one. Since numbers with magnitude greater than one are not representable, the result overflows. For example, the two's complement numbers 0.101 (5/8) and 0.100 (4/8) add togive 1.001 which is the two's complement representation of -7/8.

The overflow characteristic of two's complement arithmetic can be represented as R{} where

$$X \quad - \quad X>1$$

An overflow oscillation, sometimes also referred to as an adder overflow limit cycle, is a high-level oscillation that can exist in an otherwise stable fixed-point filter due to the gross nonlinearity associated with the overflow of internal filter calculations .Like limit cycles, overflow oscillations require recursion to exist and do not occur in non recursive FIR filters. Overflow oscillations also do not occur with floating-point arithmetic due to the virtual impossibility of overflow.

Quantization:

Total number of bits in x is reduced by using two methods namely Truncation and Rounding. These are known as quantization Processes.

Input Quantization Error:

The Quantized signal are stored in a b bit register but for nearest values the same digital equivalent may be represented. This is termed as Input Quantization Error.

Product Quantization Error:

The Multiplication of a b bit number with another b bit number results in a 2b bit number but it should be stored in a b bit register. This is termed as Product Quantization Error.

Co-efficient Quantization Error:

The Analog to Digital mapping of signals due to the Analog Co-efficient Quantization results in error due to the Fact that the stable poles marked at the edge of the jΩ axis may be marked as an unstable pole in the digital domain.

Limit Cycle Oscillations:

If the input is made zero, the output should be made zero but there is an error occur due to the quantization effect that the system oscillates at a certain band of values.

Overflow limit Cycle oscillations:

Overflow error occurs in addition due to the fact that the sum of two numbers may result in overflow. To avoid overflow error saturation arithmetic is used.

Dead band:

The range of frequencies between which the system oscillates is termed as Deadband of the Filter. It may have a fixed positive value or it may oscillate between a positive and negative value**.**

Signal scaling:

The inputs of the summer is to be scaled first before execution of the addition operation to find for any possibility of overflow to be occurredafter addition. The scaling factor s0is multiplied with the inputs to avoid overflow.