LECTURE NOTES

ON

DIGITAL SIGNAL PROCESSING

DIGITAL SIGNAL PROCESSING UNIT-1

- A signal is defined as any physical quantity that varies with time, space or another independent variable.
- A system is defined as a physical device that performs an operation on a signal.
- System is characterized by the type of operation that performs on the signal. Such operations are referred to as signal processing.

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 millivolts, 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.
9. Wave form generation.

Classification of Signals

I. Based on Variables:

1. f(t)=5t : single variable
2. f(x,y)=2x+3y : two variables
3. S_1= A Sin(wt) : real valued signal
4. S_2 = A e^{jwt} : A Cos(wt)+j A Sin(wt) : Complex valued signal
5. $S_4(t)=\begin{bmatrix} S1(t) \\ S2(t) \\ \\ S3(t) \end{bmatrix}$: Multichannel signal

Ex: due to earth quake, ground acceleration recorder

6. $I(x,y,t)=\begin{bmatrix} Ir(x,y,t) \\ Ig(x,y,t) \\ \\ Ib(x,y,t) \end{bmatrix}$ multidimensional

II. Based on Representation:

III. Based on duration.

1. right sided: x(n)=0 for n<N
2. left sided :x(n)=0 for n>N
3. causal : x(n)=0 for n<0
4. Anti causal : x(n)=0 for $n \geq 0$
5. Non causal : x(n)=0 for n >N

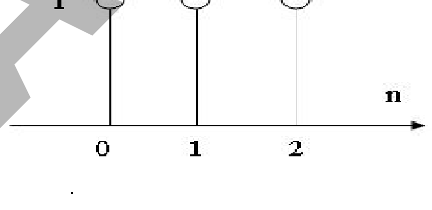
IV. Based on the Shape.

1. δ(n)=0 $\quad n \neq 0$

   =1 $\quad$ n=0

2. u (n) =1 $\quad n \geq 0$

   =0 $\quad$ n<0



Arbitrary sequence can be represented as a sum of scaled, delayed impulses.

$P(n) = a_{-3}\,\delta(n+3) + a_1\,\delta(u-1) + a_2\,\delta(u-2) + a_7\,\delta(u-7)$

Or

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

$$u(n) = \sum_{k=-\infty}^{n} \delta(k) = \delta(n) + \delta(n-1) + \delta(n-2)\ldots..$$

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

3.Discrete pulse signals.

Rect (n/2N) =1 $\quad\quad n \le N$

= 0 $\quad\quad$ else where.

5.Tri (n/N) = 1- n /N $\quad n \le N$

= 0 $\quad\quad$ else where.

1. Sinc (n/N)= Sa(n∏/N) = Sin(n∏/N) / (n∏/N), Sinc(0)=1

Sinc (n/N) =0 at n=kN, k= ± 1, ± 2…

Sinc (n) = δ(n) for N=1; (Sin (n∏) / n∏=1= δ(n))

6.Exponential Sequence

$x(n) = A\,\alpha^n$

If A & α are real numbers, then the sequence is real. If $0<\alpha<1$ and A is +ve, then sequence values are +ve and decreases with increasing n.

For $-1<\alpha<0$, the sequence values alternate in sign but again decreases in magnitude with increasing n. If $\alpha>1$, then the sequences grows in magnitude as n increases.

7.Sinusoidal Sequence

$x(n) = A\,\text{Cos}(w_o n+\phi)$ for all n

8.Complex exponential sequence

If $\alpha = |\alpha| e^{jwo}$

$A = |A| e^{j\phi}$

$x(n) = |A| e^{j\phi} |\alpha|^n e^{jwon}$

$= |A| |\alpha|^n \mathrm{Cos}(w_o n+\phi) + j |A| |\alpha|^n \mathrm{Sin}(w_o n+\phi)$

If $\alpha>1$, the sequence oscillates with exponentially growing envelope.

If $\alpha<1$, the sequence oscillates with exponentially decreasing envelope.

So when discussing complex exponential signals of the form $x(n)= A\ e^{jwon}$ or real sinusoidal signals of the form $x(n)= A\ \mathrm{Cos}(w_o n+\phi)$, we need only consider frequencies in a frequency internal of length 2Π such as $\Pi < Wo < \Pi$ or $0 \leq Wo<2\Pi$.

V. Deterministic ($x(t) = \alpha^t$ $x(t) = A\ \mathrm{Sin}(wt)$)

& Non-deterministic Signals. (Ex: Thermal noise.)

VI. Periodic & non periodic based on repetition.

VII. Power & Energy Signals

Energy signal: E = finite, P=0

- Signal with finite energy is called energy signal.

Power signal: $E = \infty$, $P \neq 0$, $P \neq \infty$ Ex: All periodic waveforms

Neither energy nor power: $E=\infty$, $P=0$ √

Based on Symmetry

1.	Even	$x(n)=x_e(n)+x_o(n)$
2.	Odd	$x(-n)=x_e(-n)+x_o(-n)$
3.	Hidden	$x(-n)=x_e(n)-x_o(n)$
4.	Half-wave symmetry.	$x_e(n)= \frac{1}{2}[x(n)+x(-n)]$

$$x_o(n) = \frac{1}{2}[x(n)-x(-n)]$$

Operation on Signals:

1. Shifting.

x(n) $\rightarrow$ shift right or delay = x(n-m)

x(n) $\rightarrow$ shift left or advance = x(n+m)

2. Time reversal or fold.

x(-n+2) is x(-n) delayed by two samples.

x(-n-2) is x(-n) advanced by two samples.

Or

x(n) is right shift x(n-2), then fold x(-n-2)

x(n) fold x(-n) shift left x(-(n+2)) = x(-n-2)

Ex:

x(n) = 2, 3 , 4 , 5, 6, 7 .

↑

Find 1. y(n)=x(n-3) 2. x(n+2) 3. x(-n) 4. x(-n+1) 5. x(-n-2)

1. y(n)= x(n-3) = { 0 ,2,3,4,5,6,7} shift x(n) right 3 units.
↑

2. $x(n+2) = \{ 2,3,4,5,\underset{\uparrow}{6},7\}$ shift x(n) left 2 units.

3. $x(-n) = \{ 7,6,5,\underset{\uparrow}{4},3,2\}$ fold x(n) about n=0.

4. $x(-n+1) = \{ 7,6,\underset{\uparrow}{5},4,3,2\}$ fold x(n), delay by 1.

5. $x(-n-2) = \{ 7,6,5,4,3,2\}$ fold x(n), advanced by 2.

3. a. Decimation.

Suppose x(n) corresponds to an analog signal x(t) sampled at intervals Ts. The signal y(n) = x(2n) then corresponds to the compressed signal x(2t) sampled at Ts and contains only alternate samples of x(n)(corresponding to x(0), x(2), x(4)…). We can also obtain directly from x(t) (not in compressed version). If we sample it at intervals 2Ts (or at a sampling rate $Fs = \frac{1}{2Ts}$). This means a two fold reduction in the sampling rate. Decimation by a factor N is equivalent to sampling x(t) at intervals NTs and implies an N-fold reduction in the sampling rate.

b. Interpolation.

y(n) = x(n/2) corresponds to x(t) sampled at Ts/2 and has twice the length of x(n) with one new sample between adjacent samples of x(n).

The new sample value as ‘0’ for Zero interpolation.

The new sample constant = previous value for step interpolation.

The new sample average of adjacent samples for linear interpolation.

Interpolation by a factor of N is equivalent to sampling x(t) at intervals Ts/N and implies an N-fold increase in both the sampling rate and the signal length.

Ex:

$$\{\underset{\uparrow}{1}, 2, 6, 4, 8\} \xrightarrow[n \to 2n]{\text{Decimation}} \{\underset{\uparrow}{1}, 6, 8\} \xrightarrow[n \to n/2]{\text{Step interpolation}} \{\underset{\uparrow}{1}, 1, 6, 6, 8, 8\}$$

$$\{\underset{\uparrow}{1}, 2, 6, 4, 8\} \xrightarrow[n \to n/2]{\text{Step interpolation}} \{1, \underset{\uparrow}{1},2,2,6, 6,4,4,8, 8\} \xrightarrow[n \to 2n]{\text{Decimation}} \{\underset{\uparrow}{1}, 2, 6, 4, 8\}$$

Since Decimation is indeed the inverse of interpolation, but the converse is not necessarily true. First Interpolation & Decimation.

Ex: $x(n) = \{ \underset{\uparrow}{1}\ 1, 2, 5, -1\}$

x(n/3) = { 1,0,0, $\underset{\uparrow}{2}$ 2,0,0,5,0,0,-1,0,0} Zero interpolation.

= { 1,1,1, $\underset{\uparrow}{2}$,2,2,5,5,5,-1,-1,-1} Step interpolation.

= { 1, $\frac{4}{3}$, $\frac{5}{3}$, $\underset{\uparrow}{2}$, 3,4,5,3,1,-1, - $\frac{2}{3}$,- $\frac{1}{3}$} Linear interpolation.

4. Fractional Delays.

It requires interpolation (N), shift (M) and Decimation (n): x (n - $\frac{M}{N}$) = x ($\frac{(Nn-M)}{N}$)

x(n) = {2, 4, $\underset{\uparrow}{6}$, 8}, find y(n)=x(n-0.5) = x ($\frac{2n-1}{2}$)

g(n) = x (n/2) = {2, 2, 4, 4, $\underset{\uparrow}{6}$, 6, 8,8} for step interpolation.

h(n) =g(n-1) = x($\frac{n-1}{2}$) = {2, 2, 4, $\underset{\uparrow}{4}$, 6, 6,8,8}

y(n) = h(2n) = x(n-0.5) = x($\frac{2n-1}{2}$) = {2, $\underset{\uparrow}{4}$, 6, 8}

OR

g(n) = x(n/2) = {2,3,4,5, $\underset{\uparrow}{6}$,7,8,4} linear interpolation.

h(n) = g(n-1) = {2,3,4, $\underset{\uparrow}{5}$, 6, 7,8,4}

g (n) = h(2n)={3,5,7,4}

Classification of Systems

1. a. Static systems or memory less system. (Non Linear / Stable)

Ex. y(n) = a x (n)

= n x(n) + b x^3(n)

= $[x(n)]^2$ = a(n-1) x(n)

y(n) = τ [x(n), n]

If its o/p at every value of 'n' depends only on the input x(n) at the same value of 'n'

Do not include delay elements. Similarly to combinational circuits.

b. Dynamic systems or memory.

If its o/p at every value of 'n' depends on the o/p till (n-1) and i/p at the same value of 'n' or previous value of 'n'.

Ex. y(n) = x(n) + 3 x(n-1)

= 2 x(n) - 10 x(n-2) + 15 y(n-1)

Similar to sequential circuit.

2. Ideal delay system. (Stable, linear, memory less if nd=0)

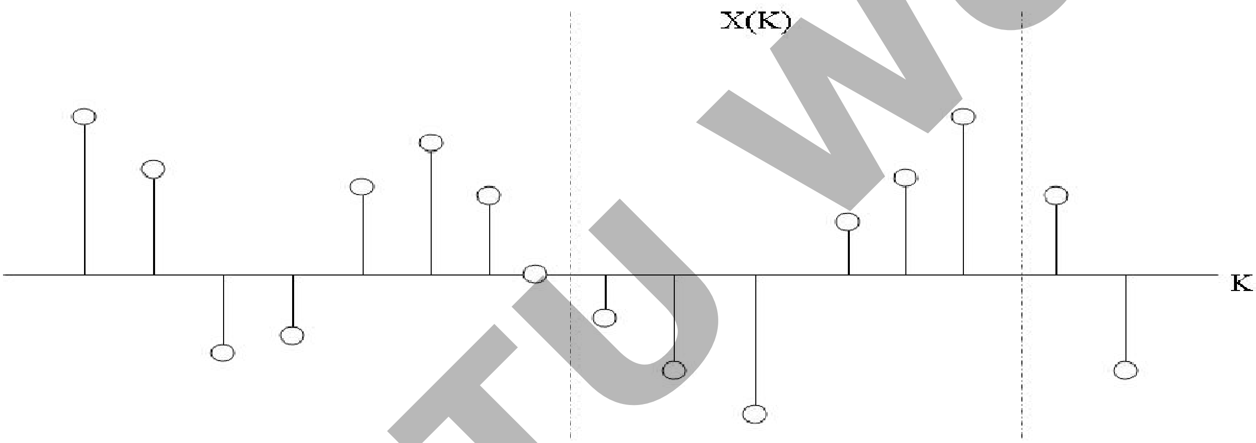
Ex. y (n) = x(n-nd)

nd is fixed = +ve integer.

3. Moving average system. (LTIV ,Stable)

$$y(n) = 1/(m_1+m_2+1) \sum_{k=-m1}^{m2} x(n-k)$$

This system computes the n^{th} sample of the o/p sequence as the average of (m_1+m_2+1) samples of input sequence around the n^{th} sample.



If M1=0; M2=5

$$y(7) = 1/6 \left[\sum_{k=0}^{5} x(7-k) \right]$$

$= 1/6\, [x(7) + x(6) + x(5) + x(4) + x(3) + x(2)]$

$y(8) = 1/6\, [x(8) + x(7) + x(6) + x(5) + x(4) + x(3)]$

So to compute y (8), both dotted lines would move one sample to right.

4. Accumulator. (Linear , Unstable)

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

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

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

$x(n) = \{ \ldots 0,3,2,1,0,1,2,3,0,\ldots.\}$

$y(n) = \{ \ldots 0,3,5,6,6,7,9,12,12\ldots\}$

O/p at the n^{th} sample depends on the i/p's till n^{th} sample

Ex:

x(n) = n u(n) ; given y(-1)=0. i.e. initially relaxed.

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

$$= y(-1) + \sum_{k=0}^{n} x(k) = 0 + \sum_{k=0}^{n} n = \frac{n(n+1)}{2}$$

5. Linear Systems.

If $y_1(n)$ & $y_2(n)$ are the responses of a system when $x_1(n)$ & $x_2(n)$ are the respective inputs, then the system is linear if and only if

$$\tau[x1(n) + x2(n)] = \tau[x1(n)] + \tau[x2(n)]$$

$$= y_1(n) + y_2(n) \quad \text{(Additive property)}$$

$$\tau[ax(n)] = a\,\tau[x(n)] = a\,y(n) \quad \text{(Scaling or Homogeneity)}$$

The two properties can be combined into principle of superposition stated as

$$\tau[ax1(n) + bx2(n)] = a\,\tau[x1(n)] + b\,\tau[x2(n)]$$

Otherwise non linear system.

6. Time invariant system.

Is one for which a time shift or delay of input sequence causes a corresponding shift in the o/p sequence.

$y(n\text{-}k) = \tau[x(n-k)]$ TIV

$\neq$ TV

7. Causality.

A system is causal if for every choice of n_o the o/p sequence value at index $n= n_o$ depends only on the input sequence values for $n \leq n_o$.

y(n) = x(n) + x(n-1) causal.

y(n) = x(n) + x(n+2) + x(n-4) non causal.

8. Stability.

For every bounded input $|x(n)| \le B_x < \infty$ for all n, there exists a fixed +ve finite value

By such that $|y(n)| \le B_y < \infty$.

PROPERTIES OF LTI SYSTEM.

1. $x(n) = \sum_{k=-\infty}^{\infty} x(k)\delta(n-k)$

$y(n) = \tau[\sum_{k=-\infty}^{\infty} x(k)\delta(n-k)]$ for linear

$\sum_{k=-\infty}^{\infty} x(k)\ \tau[\delta(n-k)]$ for time invariant

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

Therefore o/p of any LTI system is convolution of i/p and impulse response.

$y(n_o) = \sum_{k=-\infty}^{\infty} h(k)x(no-k)$

$= \sum_{k=-\infty}^{-1} h(k)x(no-k) + \sum_{k=0}^{\infty} h(k)x(no-k)$

$= h(-1)\,x(n_0+1) + h(-2)\,x(n_0+2)\ldots\ldots\ldots + h(0)\,x(n_0) + h(1)\,x(n_0-1) + \ldots.$

y(n) is causal sequence if h(n) =0 n<0

y(n) is anti causal sequence if h(n) =0 n≥ 0

y(n) is non causal sequence if h(n) =0 |n|>N

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

If i/p is also causal $y(n) = \sum_{k=0}^{n} h(k)x(n-k)$

2. Convolution operation is commutative.

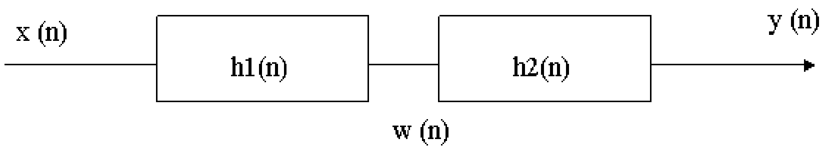
$x(n) * h(n) = h(n) * x(n)$

3. Convolution operation is distributive over additive.

$x(n) * [h_1(n) + h_2(n)] = x(n) * h_1(n) + x(n) * h_2(n)$

4. Convolution property is associative.

$x(n) * h_1(n) * h_2(n) = [x(n) * h_1(n)] * h_2(n)$



5 $y(n) = h_2 * w(n) \quad = \quad h2(n)*h1(n)*x(n) = h3(n)*x(n)$

6

$h(n) = h_1(n) + h_2(n)$

7 LTI systems are stable if and only if impulse response is absolutely summable.

$$|y(n)| = \left|\sum_{k=-\infty}^{\infty} h(k)x(n-k)\right| \le \sum_{k=-\infty}^{\infty} h(k) \quad x(n-k)$$

Since x (n) is bounded $x(n) \le b_x < \infty$

$$\therefore |y(n)| \le B_x \sum_{k=-\infty}^{\infty} |h(k)|$$

$\therefore S = \sum_{k=-\infty}^{\infty} |h(k)|$ is necessary & sufficient condition for stability.

8 $\delta(n) * x(n) = x(n)$

9 Convolution yields the zero state response of an LTI system.

10 The response of LTI system to periodic signals is also periodic with identical period.

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

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

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

put n-k = m

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

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

m=k

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

Q. y (n)-0.4 y(n-1) =x (n). Find causal impulse response? h(n)=0 n<0.

$h(n) = 0.4\, h(n-1) + \delta(n)$

$h(0) = 0.4\, h(-1) + \delta(0) = 1$

$h(1) = 0.4\, h(0) = 0.4$

$h(2) = 0.4^2$

$h(n) = 0.4^n$ for $n \geq 0$

Q. y(n)-0.4 y(n-1) = x(n). find the anti-causal impulse response? h(n)=0 for n≥0

$h(n-1) = 2.5\, [h(n) - \delta(n)]$

$h(-1) = 2.5\, [h(0) - \delta(0)] = -2.5$

$h(-2) = -2.5^2$. …….. $h(n) = -2.5^n$ valid for $n \leq -1$

Q. x(n)={1,2,3} y(n)={3,4} Obtain difference equation from i/p & o/p information

y(n) + 2 y(n-1) + 3 y(n-2) = 3 x(n) + 4 x(n-1) (Ans)

Q. x(n) = {4,4,}, y(n)= x(n)- 0.5x(n-1). Find the difference equation of the inverse system. Sketch the realization of each system and find the output of each system.

Solution:

The original system is y(n)=x(n)-0.5 x(n-1)

The inverse system is x(n)= y(n)-0.5 y(n-1)

y (n) = x (n) – 0.5 x(n-1)

$Y(z) = X(z)\,[1-0.5Z^{-1}]$

$\frac{Y(z)}{X(z)} = 1-0.5\,Z^{-1}$ System

Inverse System

y (n) – 0.5 y(n-1) =x(n)

$Y(z)\,[1-0.5\,Z^{-1}] = X(z)$

$\frac{Y(z)}{X(z)} = [1-0.5\,Z^{-1}]^{-1}$

$g(n) = 4\,\delta(n) - 2\delta(n-1) + 4\delta(n-1) - 2\delta(n-2) = 4\delta(n) + 2\delta(n-1) - 2\delta(n-2)$

$y(n) = 0.5\,y(n-1) + 4\delta(n) + 2\delta(n-1) - 2\delta(n-2)$

$y(0) = 0.5y(-1) + 4\delta(0) = 4$

y(1) = 4

$y(2) = 0.5\,y(1) - 2\delta(0) = 0$

y(n) = {4, 4} same as i/p.

Non Recursive filters	Recursive filters
$y(n) = \sum_{k=-\infty}^{\infty} a_k\,x(n-k)$ for causal system $= \sum_{k=0}^{\infty} a_k\,x(n-k)$ For causal i/p sequence	$y(n) = \sum_{k=0}^{N} a_k\,x(n-k) - \sum_{k=1}^{N} b_k\,y(n-k)$ Present response is a function of the present and past N values of the excitation as well as the past N values of response. It gives IIR o/p but not

$y(n) = \sum_{k=0}^{N} a_k x(n-k)$ Present response depends only on present i/p & previous i/ps but not future i/ps. It gives FIR o/p.	always. $y(n) - y(n-1) = x(n) - x(n-3)$

Q. $y(n) = \frac{1}{3}[x(n+1) + x(n) + x(n-1)]$ Find the given system is stable or not?

Let $x(n) = \delta(n)$

$h(n) = \frac{1}{3}[\delta(n+1) + \delta(n) + \delta(n-1)]$

$h(0) = \frac{1}{3}$

$h(-1) = \frac{1}{3}$

$h(1) = \frac{1}{3}$

$S = \sum h(n) < \infty$ therefore Stable.

Q. $y(n) = a\, y(n-1) + x(n)$ given $y(-1) = 0$

Let $x(n) = \delta(n)$

$h(n) = y(n) = a\, y(n-1) + \delta(n)$

$h(0) = a\, y(-1) + \delta(0) = 1 = y(0)$

$h(1) = a\, y(0) + \delta(1) = a$

$h(2) = a\, y(1) + \delta(2) = a^2$ . . . . $h(n) = a^n u(n)$ stable if $a<1$.

$y(n-1) = \frac{1}{a}[y(n) - x(n)]$

$y(n) = \frac{1}{a}[y(n+1) - x(n+1)]$

$y(-1) = \frac{1}{a}[\ y(0) - x(0)]=0$

$y(-2) = 0$

Q. $y(n) = \frac{1}{n+1}\ y(n-1) + x(n)$ for $n \geq 0$

$= 0$ otherwise. Find whether given system is time variant or not?

Let $x(n) = \delta(n)$

$h(0) = 1\ y(-1) + \delta(0) = 1$

$h(1) = ½\ y(0) + \delta(1) = ½$

$h(2) = 1/6$

$h(3) = 1/24$

if $x(n) = \delta(n-1)$

$y(n) = h(n-1)$

$h(n-1) = y(n) = \frac{1}{n+1}\ h(n-2) + \delta(n-1)$

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

n=1 $h(0) = y(1) = ½ \times 0 + \delta(0) = 1$

n=2 $h(1) = y(2) = 1/3 \times 1 + 0 = 1/3$

$h(2) = 1/12$

$\therefore h(n, 0) \neq h(n,1)$ $\therefore$ TV

Q. $y(n) = 2n\ x(n)$ Time varying

Q. $y(n) = \frac{1}{3}[x(n+1) + x(n) + x(n-1)]$ Linear

Q. $y(n) = 12\ x(n-1) + 11\ x(n-2)$ TIV

Q. $y(n) = 7\ x^2(n-1)$ non linear

Q. $y(n) = x^2(n)$ non linear

Q. $y(n) = n^2\ x(n+2)$ linear

Q. $y(n) = x(n^2)$ linear

Q. $y(n) = e^{x(n)}$ non linear

Q. $y(n) = 2^{x(n)}\ x(n)$ non linear, TIV

(If the roots of characteristics equation are a magnitude less than unity. It is a necessary & sufficient condition)

Non recursive system, or FIR filter are always stable.

Q. $y(n) + 2y^2(n) = 2x(n) - x(n-1)$ non linear, TIV

Q. $y(n) - 2y(n-1) = 2^{x(n)} x(n)$ non linear, TIV

Q. $y(n) + 4y(n)y(2n) = x(n)$ non linear, TIV

Q. $y(n+1) - y(n) = x(n+1)$ is causal

Q. $y(n) - 2y(n-2) = x(n)$ causal

Q. $y(n) - 2y(n-2) = x(n+1)$ non causal

Q. $y(n+1) - y(n) = x(n+2)$ non causal

Q. $y(n-2) = 3x(n-2)$ is static or Instantaneous.

Q. $y(n) = 3x(n-2)$ dynamic

Q. $y(n+4) + y(n+3) = x(n+2)$ causal & dynamic

Q. $y(n) = 2x(\alpha n)$

If $\alpha=1$ causal, static

$\alpha<1$ causal, dynamic

$\alpha>1$ non causal, dynamic

$\alpha \neq 1$ TV

Q. $y(n) = 2(n+1)x(n)$ is causal & static but TV.

Q. $y(n) = x(-n)$ TV

Solution of linear constant-co-efficient difference equation

Q. $y(n) - 3y(n-1) - 4y(n-2) = 0$ determine zero-input response of the system;

Given $y(-2) = 0$ & $y(-1) = 5$

Let solution to the homogeneous equation be

$y_h(n) = \lambda^n$

$\lambda^n - 3\lambda^{n-1} - 4\lambda^{n-2} = 0$

$\lambda^{n-2}[\lambda^2 - 3\lambda - 4] = 0$

$\lambda = -1, 4$

$y_h(n) = C_1\lambda_1^n + C_2\lambda_2^n = C_1(-1)^n + C_2 4^n$

$y(0) = 3y(-1) + 4y(-2) = 15$

$\therefore\ C_1 + C_2 = 15$

$y(1) = 3y(0) + 4y(-1) = 65$

$\therefore\ -C_1 + 4C_2 = 65$ Solve: $C_1 = -1$ & $C_2 = 16$

$y(n) = (-1)^{n+1} + 4^{n+2}$ (Ans)

If it contain multiple roots $y_h(n) = C_1 \lambda_1^n + C_2 n\lambda_1^n + C_3 n^2 \lambda_1^n$

or $\lambda_1^n [C_1 + nC_2 + n^2 C_3 \ldots.]$

Q. Determine the particular solution of $y(n) + a_1 y(n-1) = x(n)$

$x(n) = u(n)$

Let $y_p(n) = k\, u(n)$

$k\, u(n) + a_1\, k\, u(n-1) = u(n)$

To determine the value of k, we must evaluate this equation for any $n \geq 1$

$k + a_1 k = 1$

$k = \frac{1}{1+a1}$

$y_p(n) = \frac{1}{1+a1} u(n)$ Ans

x(n)	$y_p(n)$
1. A	K
2. Am^n	Km^n
3. An^m	$K_o n^m + K_1 n^{m-1} + \ldots.\ K_m$
4. A $\text{Cos}w_o n$ or A $\text{Sin}w_o n$	$K_1 \text{ Cos}w_o n + K_2 \text{ Sin}w_o n$

Q. $y(n) = \frac{5}{6} y(n-1) - \frac{1}{6} y(n-2) + x(n)$ $x(n) = 2^n$ $n \geq 0$

Let $y_p(n) = K2^n$

$K2^n u(n) = \frac{5}{6} K\, 2^{n-1} u(n-1) - \frac{1}{6} K\, 2^{n-2} u(n-2) + 2^n u(n)$

For $n \geq 2$

$4K = \frac{5}{6}(2K) - \frac{1}{6}K + 4$ Solve for $K = 8/5$

$\therefore\ y_p(n) = \frac{8}{5} 2^n$ Ans

Q. $y(n) - 3y(n-1) - 4y(n-2) = x(n) + 2x(n-1)$ Find the h(n) for recursive system.

We know that $y_h(n) = C_1(-1)^n + C_2 4^n$

$y_p(n) = 0$ when $x(n) = \delta(n)$

for n=0

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

$\therefore y(0) = 1$

$y(1) = 3y(0) + 2 = 5$

$C_1 + C_2 = 1$

$-C_1 + C_2 = 5$ Solving $C_1 = -\frac{1}{5}$; $C_2 = \frac{6}{5}$

$\therefore h(n) = [-\frac{1}{5}(-1)^n + \frac{6}{5}4^n]\,u(n)$ Ans

OR

$h(n) - 3h(n-1) - 4h(n-2) = \delta(n) + 2\delta(n-1)$

$h(0) = 1$

$h(1) = 3h(0) + 2 = 5$

plot for h(n) in both the methods are same.

Q. $y(n) - 0.5y(n-1) = 5\cos 0.5n\Pi$ $n \geq 0$ with $y(-1) = 4$

$y_h(n) = \lambda^n$

$\lambda^n - 0.5\lambda^{n-1} = 0$

$\lambda^{n-1}[\lambda - 0.5] = 0$

$\lambda = 0.5$

$\therefore y_h(n) = C(0.5)^n$

$y_p(n) = K_1\cos 0.5n\Pi + K_2\sin 0.5n\Pi$

$y_p(n-1) = K_1\cos 0.5(n-1)\Pi + K_2\sin 0.5(n-1)\Pi$

$= -K_1\sin 0.5n\Pi - K_2\cos 0.5n\Pi$

$y_p(n) - 0.5y_p(n-1) = 5\cos 0.5n\Pi$

$= (K_1 + 0.5K_2)\cos 0.5n\Pi - (0.5K_1 - K_2)\sin 0.5n\Pi$

$K_1 + 0.5K_2 = 5$

$0.5K_1 - K_2 = 0$ Solving we get: $K_1 = 4$ & $K_2 = 2$

$\therefore y_p(n) = 4\cos 0.5n\Pi + 2\sin 0.5n\Pi$

The final response

$y(n) = C(0.5)^n + 4\cos 0.5n\Pi + 2\sin 0.5n\Pi$

with $y(-1) = 4$

$4 = 2C-2$

$C=3$

$\therefore y(n) = 3(0.5)^n + 4\cos 0.5n\Pi + 2\sin 0.5n\Pi \quad \text{for } n\geq 0$

Concept of frequency in continuous-time and discrete-time.

1) $x_a(t) = A\cos(\Omega t)$

$x(nTs) = A\cos(\Omega nTs)$

$= A\cos(wn)$

$w = \Omega Ts$

Ω = rad / sec w = rad / Sample

F = cycles / sec f = cycles / Sample

2) A Discrete- time – sinusoid is periodic only of its f is a Rational number.

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

$\cos 2\pi f_0(n+N) = \cos 2\pi f_0 n$

$2\pi f_0 N = 2\pi K \Rightarrow f_0 = \frac{K}{N}$

Ex: $A\cos(\frac{\Pi}{6})n$

$w = \frac{\Pi}{6} = 2\pi f$

$f = \frac{1}{12}$ N=12 Samples/Cycle ; Fs= Sampling Frequency; Ts = Sampling Period

Q. Cos (0.5n) is not periodic

Q. x (n) = 5 Sin (2n)

$2\pi f = 2 \Rightarrow f = \frac{1}{\Pi}$ Non-periodic

Q. x (n) = 5 Cos (6πn)

$2\pi f = 6\pi \Rightarrow f = 3$ N=1 for K=3 Periodic

Q. x (n) = 5 Cos $\frac{6\Pi n}{35}$

$2\pi f = \frac{6\Pi}{35} \Rightarrow f = \frac{3}{35}$ for N=35 & K=3 Periodic

Q. x (n) = Sin (0.01πn)

$2\pi f = 0.01\pi \Rightarrow f = \frac{0.01}{2}$ for N=200 & K=1 Periodic

Q. x (n) = Cos (3πn) for N=2 Periodic

f_o = GCD (f_1, f_2) & T = LCM (T_1, T_2) ------- For Analog/digital signal

[Complex exponential and sinusoidal sequences are not necessarily periodic in 'n' with period ($\frac{2\Pi}{Wo}$) and depending on Wo, may not be periodic at all]

N = fundamental period of a periodic sinusoidal.

3. The highest rate of oscillations in a discrete time sinusoid is obtained when w = π or -π

Discrete-time sinusoidal signals with frequencies that are separated by an integral multiple of 2π are Identical.

4. $-\frac{Fs}{2} \leq F \leq \frac{Fs}{2}$

$-\pi Fs \leq 2\pi F \leq \pi Fs$

$-\frac{\Pi}{Ts} \leq \Omega \leq \frac{\Pi}{Ts}$

$-\pi \leq \Omega\, Ts \leq \pi$

Therefore $-\pi \leq w \leq \pi$

5. Increasing the frequency of a discrete- time sinusoid does not necessarily decrease the period of the signal.

$x_1(n) = \mathrm{Cos}\left(\frac{\Pi n}{4}\right)$ N=8

$x_2(n) = \mathrm{Cos}\left(\frac{3\Pi n}{8}\right)$ N=16 3/8 > 1/4

$2\pi f = 3\pi/8$

$\Rightarrow f = \frac{3}{16}$

6. If analog signal frequency = F = $\frac{1}{Ts}$ samples/Sec = Hz then digital frequency f = 1

$W = \Omega\, T_s$

$2\pi f = 2\pi F\, T_s$ => f =1

$2\pi F = \frac{\Pi}{4}$; $2\pi f = \pi/4$

$F = \frac{1}{8}$; $T = 8$; $f = \frac{1}{8}$ N=8

7. Discrete-time sinusoids are always periodic in frequency.

Q. The signal x (t) = 2 Cos (40 π t) + Sin (60π t) is sampled at 75Hz. What is the common period of the sampled signal x (n), and how many full periods of x (t) does it take to obtain one period of x(n)?

F_1 = 20Hz F_2 = 30Hz

$f_1 = \frac{20}{75} = \frac{4}{15} = \frac{K1}{N1}$ $f_2 = \frac{30}{75} = \frac{2}{5} = \frac{K2}{N2}$

The common period is thus N=LCM (N_1, N2) = LCM (15, 5) = 15

The fundamental frequency F_o of x (t) is GCD (20, 30) = 10Hz

And fundamental period $T = \frac{1}{Fo} = 0.1s$

Since N=15

1sample ---------- $\frac{1}{75}$ sec

15 sample ----------- ? => $\frac{15}{75} = 0.2S$

$\therefore$ So it takes two full periods of x (t) to obtain one period of x (n) or GCD (K_1, K_2) = GCD (4, 2) = 2

Frequency Domain Representation of discrete-time signals and systems

For LTI systems we know that a representation of the input sequence as a weighted sum of delayed impulses leads to a representation of the output as a weighted sum of delayed responses.

Let x (n) = e^{jwn}

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

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

$$= e^{jwn} \sum_{k=-\infty}^{\infty} h(k)\, e^{-jwk}$$

Let H (e^{jw}) = $\sum_{k=-\infty}^{\infty} h(k)\, e^{-jwk}$ is the frequency domain representation of the system.

$\therefore$ y (n) = H (e^{jw}) e^{jwn}

e^{jwn} = eigen function of the system.

H (e^{jw}) = eigen value

UNIT 2

DISCRETE FOURIER TRANSFORMS (DFT)

1.1 Introduction:

Before we introduce the DFT we consider the sampling of the Fourier transform of an aperiodic discrete-time sequence. Thus we establish the relation between the sampled Fourier transform and the DFT.A discrete time system may be described by the convolution sum, the Fourier representation and the z transform as seen in the previous chapter. If the signal is periodic in the time domain DTFS representation can be used, in the frequency domain the spectrum is discrete and periodic. If the signal is non-periodic or of finite duration the frequency domain representation is periodic and continuous this is not convenient to implement on the computer. Exploiting the periodicity property of DTFS representation the finite duration sequence can also be represented in the frequency domain, which is referred to as Discrete Fourier Transform DFT.

DFT is an important mathematical tool which can be used for the software implementation of certain digital signal processing algorithms .DFT gives a method to transform a given sequence to frequency domain and to represent the spectrum of the sequence using only k frequency values, where k is an integer that takes N values, K=0, 1, 2,…..N-1.

The advantages of DFT are:

1. It is computationally convenient.
2. The DFT of a finite length sequence makes the frequency domain analysis much simpler than continuous Fourier transform technique.

1.2 FREQUENCY DOMAIN SAMPLING AND RECONSTRUCTION OF DISCRETE TIME SIGNALS:

Consider an aperiodic discrete time signal x (n) with Fourier transform, an aperiodic finite energy signal has continuous spectra. For an aperiodic signal x[n] the spectrum is:

$$X[w] = \sum_{n=-\infty}^{\infty} x[n]e^{-jwn} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(1.1)$$

Suppose we sample X[w] periodically in frequency at a sampling of δw radians between successive samples. We know that DTFT is periodic with 2π, therefore only samples in the fundamental frequency range will be necessary. For convenience we take N equidistant samples in the interval (0<=w<2π). The spacing between samples will be $\delta w = \frac{2\pi}{N}$ as shown below in Fig.1.1.

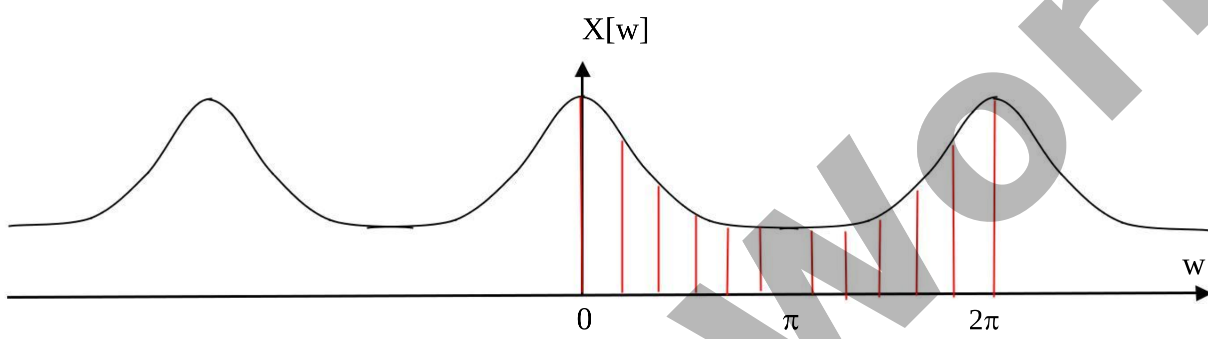


Fig 1.1 Frequency Domain Sampling

Let us first consider selection of N, or the number of samples in the frequency domain.

If we evaluate equation (1) at $w = \frac{2\pi k}{N}$

$$X\left[\frac{2\pi k}{N}\right] = \sum_{n=-\infty}^{\infty} x[n] e^{-j2\pi kn/N} \qquad k = 0,1,2,\ldots\ldots,(N-1) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1.2)$$

We can divide the summation in (1) into infinite number of summations where each sum contains N terms.

$$X\left[\frac{2\pi k}{N}\right] = \ldots\ldots + \sum_{n=-N}^{-1} x[n] e^{-j2\pi kn/N} + \sum_{n=0}^{N-1} x[n] e^{-j2\pi kn/N} + \sum_{n=N}^{2N-1} x[n] e^{-j2\pi kn/N}$$

$$= \sum_{l=-\infty}^{\infty} \sum_{n=lN}^{lN+N-1} x[n] e^{-j2\pi kn/N}$$

If we then change the index in the summation from n to n-l N and interchange the order of summations we get:

$$X\left[\frac{2\pi k}{N}\right] = \sum_{n=0}^{N-1}\left[\sum_{l=-\infty}^{\infty} x[n-lN]\right] e^{-j2\pi kn/N} \qquad for\ \ k = 0,1,2,......,(N-1) \ ....(1.3)$$

Denote the quantity inside the bracket as $x_p[n]$. This is the signal that is a repeating version of x[n] every N samples. Since it is a periodic signal it can be represented by the Fourier series.

$$x_p[n] = \sum_{k=0}^{N-1} c_k\, e^{j2\pi kn/N} \qquad n = 0,1,2,........,(N-1)$$

With FS coefficients:

$$c_k = \frac{1}{N}\sum_{n=0}^{N-1} x_p[n]\, e^{-j2\pi kn/N} \qquad k = 0,1,2,.......,(N-1) \ .... (1.4)$$

Comparing the expressions in equations (1.4) and (1.3) we conclude the following:

$$c_k = \frac{1}{N} X\left[\frac{2\pi}{N} k\right] \qquad k = 0,1,.......,(N-1) \ .... (1.5)$$

Therefore it is possible to write the expression $x_p[n]$ as below:

$$x_p[n] = \frac{1}{N}\sum_{k=0}^{N-1} X\left[\frac{2\pi}{N} k\right] e^{j2\pi kn/N} \qquad n = 0,1,.....,(N-1) \ .... (1.6)$$

The above formula shows the reconstruction of the periodic signal $x_p[n]$ from the samples of the spectrum X[w]. But it does not say if X[w] or x[n] can be recovered from the samples.

Let us have a look at that:

Since $x_p[n]$ is the periodic extension of x[n] it is clear that x[n] can be recovered from $x_p[n]$ if there is no aliasing in the time domain. That is if x[n] is time-limited to less than the period N of $x_p[n]$.This is depicted in Fig. 1.2 below:

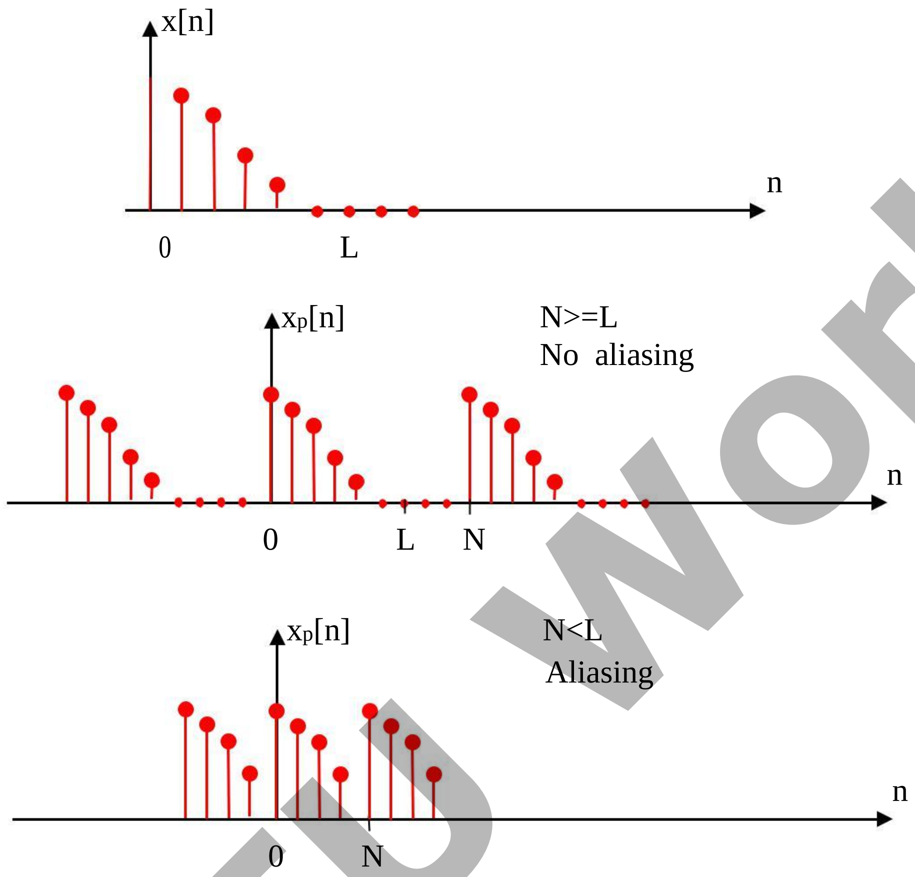


Fig. 1.2 Signal Reconstruction

Hence we conclude:
The spectrum of an aperiodic discrete-time signal with finite duration L can be exactly recovered from its samples at frequencies $w_k = \frac{2\pi k}{N}$ if N >= L.

We compute $x_p[n]$ for n=0, 1,....., N-1 using equation (1.6)
Then X[w] can be computed using equation (1.1).

1.3 Discrete Fourier Transform:

The DTFT representation for a finite duration sequence is

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

$$X(n) = 1/2\pi \int X(j\omega)\, e^{j\omega n} d\omega, \quad \text{Where } \omega = 2\pi k/n$$

2π

Where x(n) is a finite duration sequence, X(jω) is periodic with period 2π.It is convenient sample X(jω) with a sampling frequency equal an integer multiple of its period =m that is taking N uniformly spaced samples between 0 and 2π.

Let $\omega_k = 2\pi k/n$, $0 \le k \le N-1$

Therefore $X(j\omega) = \sum_{n=-\infty}^{\infty} x(n)\, e^{-j2\pi kn/N}$

Since X(jω) is sampled for one period and there are N samples X(jω) can be expressed as

$$X(k) = X(j\omega)\Big|_{\omega=2\pi kn/N} = \sum_{n=0}^{N-1} x(n)\, e^{-j2\pi kn/N} \qquad 0 \le k \le N-1$$

1.4 Matrix relation of DFT

The DFT expression can be expressed as

[X] = [x(n)] [WN]

Where $[X] = [X(0), X(1), \ldots\ldots]^T$

[x] is the transpose of the input sequence. WN is a N x N matrix

WN = 1 1 11 ………………1
1 wn1 wn2 wn3……………...wn n-1
1 wn2 wn4 wn6 ……………wn2(n-1)
…………………………………………….
…………………………………………….
1………………………………..wN (N-1)(N-1)

ex;
4 pt DFT of the sequence 0,1,2,3

X(0)		1	1	1	1
X(1)		1	-j	-1	j
X(2)	=	1	-1	1	-1
X(3)		1	j	-1	-j

Solving the matrix X(K) = 6 , -2+2j, -2 , -2-2j

1.5 Relationship of Fourier Transforms with other transforms

1.5.1 Relationship of Fourier transform with continuous time signal:

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

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

Where {ck} are the Fourier coefficients. If we sample $x_a(t)$ at a uniform rate $F_s = N/T_p = 1/T$, we obtain discrete time sequence

$$x(n) \equiv x_a(nT) = \sum_{k=-\infty}^{\infty} c_k e^{j2\pi k F_0 n T} = \sum_{k=-\infty}^{\infty} c_k e^{j2\pi kn/N}$$

$$= \sum_{k=0}^{N-1} \left[\sum_{l=-\infty}^{\infty} c_{k-lN} \right] e^{j2\pi kn/N}$$

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

Thus $\{c_{k'}\}$ is the aliasing version of $\{c_k\}$

1.5.2 Relationship of Fourier transform with z-transform

Let us consider a sequence x(n) having the z-transform

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

With ROC that includes unit circle. If X(z) is sampled at the N equally spaced points on the unit circle $Z_k = e^{j2\pi k/N}$ for K= 0,1,2,………..N-1 we obtain

$$X(k) \equiv X(z)|_{z=e^{j2\pi nk/N}} \qquad k = 0, 1, \ldots, N-1$$

$$= \sum_{n=-\infty}^{\infty} x(n) e^{-j2\pi nk/N}$$

The above expression is identical to Fourier transform X(ω) evaluated at N equally spaced frequencies $\omega_k = 2\pi k/N$ for K= 0,1,2,………..N-1.

If the sequence x(n) has a finite duration of length N or less. The sequence can be recovered from its N-point DFT. Consequently X(z) can be expressed as a function of DFT as

$$X(z) = \sum_{n=0}^{N-1} x(n) z^{-n}$$

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

$$X(z) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) \sum_{n=0}^{N-1} \left(e^{j2\pi k/N} z^{-1} \right)^n$$

$$X(z) = \frac{1 - z^{-N}}{N} \sum_{k=0}^{N-1} \frac{X(k)}{1 - e^{j2\pi k/N} z^{-1}}$$

Fourier transform of a continuous time signal can be obtained from DFT as

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

Recommended Questions with solutions

Question 1

The first five points of the 8-point DFT of a real valued sequence are {0.25, 0.125-j0.318, 0, 0.125-j0.0518, 0}. Determine the remaining three points

Ans: Since x(n) is real, the real part of the DFT is even, imaginary part odd. Thus the remaining points are {0.125+j0.0518,0,0, 0.125+j0.318}.

Question 2

Compute the eight-point DFT circular convolution for the following sequences.

$x_2(n) = \sin 3\pi n/8$

Ans:

(a)

$$\begin{aligned}
\tilde{x}_2(l) &= x_2(l), \quad 0 \le l \le N-1 \\
&= x_2(l+N), \quad -(N-1) \le l \le -1 \\
\tilde{x}_2(l) &= \sin(\frac{3\pi}{8}l), \quad 0 \le l \le 7 \\
&= \sin(\frac{3\pi}{8}(l+8)), \quad -7 \le l \le -1 \\
&= \sin(\frac{3\pi}{8}|l|), \quad |l| \le 7
\end{aligned}$$

$$\begin{aligned}
\text{Therefore, } x_1(n) \textcircled{8} x_2(n) &= \sum_{m=0}^{3} \tilde{x}^2(n-m) \\
&= \sin(\frac{3\pi}{8}|n|) + \sin(\frac{3\pi}{8}|n-1|) + \ldots + \sin(\frac{3\pi}{8}|n-3|) \\
&= \{1.25, 2.55, 2.55, 1.25, 0.25, -1.06, -1.06, 0.25\}
\end{aligned}$$

Question 3

Compute the eight-point DFT circular convolution for the following sequence $X_3(n) = \cos 3\pi n/8$

$$\begin{aligned}
\tilde{x}_2(n) &= \cos(\frac{3\pi}{8}n), \quad 0 \le l \le 7 \\
&= -\cos(\frac{3\pi}{8}n), \quad -7 \le l \le -1 \\
&= [2u(n)-1]\cos(\frac{3\pi}{8}n), \quad |n| \le 7
\end{aligned}$$

$$\begin{aligned}
\text{Therefore, } x_1(n) \textcircled{8} x_2(n) &= \sum_{m=0}^{3} \left(\frac{1}{4}\right)^m \tilde{x}^2(n-m) \\
&= \{0.96, 0.62, -0.55, -1.06, -0.26, -0.86, 0.92, -0.15\}
\end{aligned}$$

Question 4

Define DFT. Establish a relation between the Fourier series coefficients of a continuous time signal and DFT

Solution

The DTFT representation for a finite duration sequence is

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

$$X(n) = 1/2\pi \int X(j\omega)\, e^{j\omega n}\, d\omega \text{ , Where } \omega = 2\pi k/n \ 2\pi$$

Where x(n) is a finite duration sequence, X(jω) is periodic with period 2π.It is convenient sample X(jω) with a sampling frequency equal an integer multiple of its period =m that is taking N uniformly spaced samples between 0 and 2π.

Let $\omega_k = 2\pi k/n,\ 0 \le k \le N$

$$\text{Therefore } X(j\omega) = \sum_{n=-\infty}^{\infty} x(n)\, e^{-j2\pi kn/N}$$

Since X(jω) is sampled for one period and there are N samples X(jω) can be expressed as

$$X(k) = X(j\omega)\Big|_{\omega=2\pi kn/N} = \sum_{n=0}^{N-1} x(n)\, e^{-j2\pi kn/N} \qquad 0 \le k \le N-1$$

Question 5

5.7 If $X(k)$ is the DFT of the sequence $x(n)$, determine the N-point DFTs of the sequences

$$x_c(n) = x(n)\cos\frac{2\pi kn}{N} \qquad 0 \le n \le N-1$$

and

$$x_s(n) = x(n)\sin\frac{2\pi kn}{N} \qquad 0 \le n \le N-1$$

in terms of $X(k)$.

Solution:-

$$X_c(k) = \sum_{n=0}^{N-1} \frac{1}{2} x(n)\left(e^{j\frac{2\pi k_0 n}{N}} + e^{-j\frac{2\pi k_0 n}{N}}\right) e^{-\frac{2\pi kn}{N}}$$

$$= \frac{1}{2}\sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi(k-k_0)n}{N}} + \frac{1}{2}\sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi(k+k_0)n}{N}}$$

$$= \frac{1}{2}X(k-k_0)_{\text{mod}N} + \frac{1}{2}X(k+k_0)_{\text{mod}N}$$

$$\text{similarly, } X_s(k) = \frac{1}{2j}X(k-k_0)_{\text{mod}N} - \frac{1}{2j}X(k+k_0)_{\text{mod}N}$$

Question 6

Find the 4-point DFT of sequence x(n) = 6+ sin(2πn/N), n= 0,1,………N-1

Solution :-

$$\text{Here } x(n) = 6+\sin\left(\frac{2\pi n}{N}\right), \quad with\ N=4$$

$$x(n) = 6+\sin\left(\frac{2\pi n}{4}\right), n=0,1,2,3.$$

$$= 6+\sin\left(\frac{\pi}{2}n\right), n=0,1,2,3.$$

$$= \{6, 7, 6, 5\}.$$

The N-point DFT is given as,

$$X_N = [W_N]x_N$$

$$\therefore \quad X_4 = \begin{bmatrix}1 & 1 & 1 & 1\\ 1 & -j & -1 & j\\ 1 & -1 & 1 & -1\\ 1 & j & -1 & -j\end{bmatrix}\begin{bmatrix}6\\7\\6\\5\end{bmatrix}$$

$$= \begin{bmatrix}6+7+6+5\\ 6-j7-6+j5\\ 6-7+6-5\\ 6+j7-6-j5\end{bmatrix} = \begin{bmatrix}24\\-j2\\0\\+j2\end{bmatrix}$$

Question 7

Determine the eight-point DFT of the signal

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

and sketch its magnitude and phase.

Solution

$$X(k) = \sum_{n=0}^{7} x(n) e^{-j\frac{2\pi}{8}kn}$$
$$= \{6, -0.7071 - j1.7071, 1 - j, 0.7071 + j0.2929, 0, 0.7071 - j0.2929, 1 + j, -0.7071 + j1.7071\}$$
$$|X(k)| = \{6, 1.8478, 1.4142, 0.7654, 0, 0.7654, 1.4142, 1.8478\}$$
$$\angle X(k) = \left\{0, -1.9635, \frac{-\pi}{4}, 0.3927, 0, -0.3927, \frac{\pi}{4}, 1.9635\right\}$$

Question 8

Compute the N-point DFTs of the signal

$$x(n) = \cos\frac{2\pi}{N}k_0 n \quad 0 \le n \le N-1$$

Solution

$$X(k) = \sum_{n=0}^{N-1} e^{j\frac{2\pi}{N}nk_0} e^{-j\frac{2\pi}{N}kn}$$
$$= \sum_{n=0}^{N-1} e^{-j\frac{2\pi}{N}(k-k_0)n}$$
$$= N\delta(k-k_0)$$

$$x(n) = \frac{1}{2}e^{j\frac{2\pi}{N}nk_0} + \frac{1}{2}e^{-j\frac{2\pi}{N}nk_0}$$

From (e) we obtain $X(k) = \frac{N}{2}[\delta(k-k_0) + \delta(k-N+k_0)]$

Digital Signal Processing

Properties of DFT

2.1 Properties:-

The DFT and IDFT for an N-point sequence x(n) are given as

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

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

where W_N is defined as

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

In this section we discuss about the important properties of the DFT. These properties are helpful in the application of the DFT to practical problems.

The notation used below to denote the N-point DFT pair $x(n)$ and $X(k)$ is

$$x(n) \overset{\text{DFT}}{\underset{N}{\longleftrightarrow}} X(k)$$

Periodicity:-

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

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

$$X(k+N) = X(k) \quad \text{for all } k$$

2.1.2 Linearity: If

$$x_1(n) \overset{\text{DFT}}{\underset{N}{\longleftrightarrow}} X_1(k)$$

$$x_2(n) \overset{\text{DFT}}{\underset{N}{\longleftrightarrow}} X_2(k)$$

Then A x1 (n) + b x2 (n) $\longleftrightarrow$ a X1(k) + b X2(k)

2.1.3 Circular shift:

In linear shift, when a sequence is shifted the sequence gets extended. In circular shift the number of elements in a sequence remains the same. Given a sequence x (n) the shifted version x (n-m) indicates a shift of m. With DFTs the sequences are defined for 0 to N-1.

If x (n) = x (0), x (1), x (2), x (3)

X (n-1) = x (3), x (0), x (1).x (2)

X (n-2) = x (2), x (3), x (0), x (1)

2.1.4 Time shift:

If $x(n) \longleftrightarrow X(k)$

Then $x(n-m) \longleftrightarrow W_N^{mk} X(k)$

2.1.5 Frequency shift

If $x(n) \longleftrightarrow X(k)$

$W_n^{+nok}\ x(n) \longleftrightarrow X(k+no)$

Consider $x(k) = \sum_{n=0}^{N-1} x(n) W_n^{kn}$

$$X(k+no) = \sum_{n=0}^{N-1} x(n) W_N^{(k+no)n}$$

$$= \sum x(n) W_N^{kn} W_N^{non}$$

$\therefore X(k+no) \longleftrightarrow x(n) W_N^{non}$

2.1.6 Symmetry:

For a real sequence, if x(n)←→ X(k)

X(N-K) = X* (k)

For a complex sequence
DFT(x*(n)) = X*(N-K)

If x(n)	then X(k)
Real and even	real and even
Real and odd	imaginary and odd
Odd and imaginary	real odd
Even and imaginary	imaginary and even

2.2 Convolution theorem;

Circular convolution in time domain corresponds to multiplication of the DFTs

If y(n) = x(n) ⊗ h(n) then Y(k) = X(k) H(k)

Ex let x(n) = 1,2,2,1 and h(n) = 1,2,2,1
Then y (n) = x(n) ⊗ h(n)

Y(n) = 9,10,9,8

N pt DFTs of 2 real sequences can be found using a single

DFT If g(n) & h(n) are two sequences then let x(n) = g(n) +j

h(n) G(k) = ½ (X(k) + X*(k))

H(k) = 1/2j (X(K) +X*(k))

2N pt DFT of a real sequence using a single N pt DFT

Let x(n) be a real sequence of length 2N with y(n) and g(n) denoting its N pt DFT

Let y(n) = x(2n) and g(2n+1)

$X(k) = Y(k) + W_N^k G(k)$

Using DFT to find IDFT

The DFT expression can be used to find IDFT

X(n) = 1/N [DFT(X*(k)]*

Recommended Questions with solutions

Question 1

State and Prove the Time shifting Property of DFT

Solution

The DFT and IDFT for an N-point sequence x(n) are given as

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

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

where W_N is defined as

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

Time shift:

If x (n) ←→ X (k)

Then x (n-m) ←→ W_N^{mk} X (k)

Question 2

State and Prove the: (i) Circular convolution property of DFT; (ii) DFT of Real and even sequence.

Solution

(i) Convolution theorem

Circular convolution in time domain corresponds to multiplication of the DFTs
If y(n) = x(n) ⊗ h(n) then Y(k) = X(k) H(k)
Ex let x(n) = 1,2,2,1 and h(n) = 1,2,2,1 Then y (n) = x(n) ⊗ h(n)

Y(n) = 9,10,9,8
N pt DFTs of 2 real sequences can be found using a single DFT

If g(n) & h(n) are two sequences then let x(n) = g(n) +j
h(n) G(k) = ½ (X(k) + X*(k))
H(k) = 1/2j (X(K) +X*(k))
2N pt DFT of a real sequence using a single N pt DFT
Let x(n) be a real sequence of length 2N with y(n) and g(n) denoting its N pt DFT
Let y(n) = x(2n) and g(2n+1)

$X(k) = Y(k) + W_N^{K}\ G(k)$
Using DFT to find IDFT
The DFT expression can be used to find IDFT
X(n) = 1/N [DFT(X*(k)]*

(ii)DFT of Real and even sequence.
For a real sequence, if x(n)←→ X(k)
X (N-K) = X* (k)

For a complex sequence
DFT(x*(n)) = X*(N-K)

If x(n) then	X(k)
Real and even	real and even
Real and odd	imaginary and odd
Odd and imaginary	real odd
Even and imaginary	imaginary and even

Question 3

Distinguish between circular and linear convolution

Solution

1) Circular convolution is used for periodic and finite signals while linear convolution is used for aperiodic and infinite signals.
2) In linear convolution we convolved one signal with another signal where as in circular convolution the same convolution is done but in circular pattern depending upon the samples of the signal
3) Shifts are linear in linear in linear convolution, whereas it is circular in circular convolution.

Question 4

For the sequences

$$x_1(n) = \cos\frac{2\pi}{N}n \qquad x_2(n) = \sin\frac{2\pi}{N}n \qquad 0 \le n \le N-1$$

determine the N-point:

(a) Circular convolution $x_1(n) \circledN x_2(n)$
(b) Circular correlation of $x_1(n)$ and $x_2(n)$
(c) Circular autocorrelation of $x_1(n)$
(d) Circular autocorrelation of $x_2(n)$

Solution(a)

$$\begin{aligned}
x_1(n) &= \frac{1}{2}\left(e^{j\frac{2\pi}{N}n} + e^{-j\frac{2\pi}{N}n}\right) \\
X_1(k) &= \frac{N}{2}\left[\delta(k-1) + \delta(k+1)\right] \\
\text{also } X_2(k) &= \frac{N}{2j}\left[\delta(k-1) - \delta(k+1)\right] \\
\text{So } X_3(k) &= X_1(k)X_2(k) \\
&= \frac{N^2}{4j}\left[\delta(k-1) - \delta(k+1)\right] \\
\text{and } x_3(n) &= \frac{N}{2}sin(\frac{2\pi}{N}n)
\end{aligned}$$

Solution(b)

$$\begin{aligned}
\tilde{R}_{xy}(k) &= X_1(k)X_2^*(k) \\
&= \frac{N^2}{4j}\left[\delta(k-1) - \delta(k+1)\right] \\
\Rightarrow \quad \tilde{r}_{xy}(n) &= -\frac{N}{2}sin(\frac{2\pi}{N}n)
\end{aligned}$$

Solution(c)

$$\begin{aligned}
\tilde{R}_{xx}(k) &= X_1(k)X_1^*(k) \\
&= \frac{N^2}{4}\left[\delta(k-1) + \delta(k+1)\right] \\
\Rightarrow \quad \tilde{r}_{xx}(n) &= \frac{N}{2}cos(\frac{2\pi}{N}n)
\end{aligned}$$

Solution(d)

$$\begin{aligned} \hat{R}_{yy}(k) &= X_2(k)X_2^*(k) \\ &= \frac{N^2}{4}[\delta(k-1)+\delta(k+1)] \\ \Rightarrow \quad \hat{r}_{yy}(n) &= \frac{N}{2}cos(\frac{2\pi}{N}n) \end{aligned}$$

Question 5

Use the four-point DFT and IDFT to determine the sequence

$$x_3(n) = x_1(n) \, Ⓝ \, x_2(n)$$

where $x_1(n)$ and $x_2(n)$ are the sequence given

$$x_1(n) = \{\underset{\uparrow}{1}, 2, 3, 1\}$$

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

Solution

$$\begin{aligned} y(n) &= x_1(n)4x_2(n) \\ &= \sum_{m=0}^{3} x_1(m)_{\text{mod}4} x_2(n-m)_{\text{mod}4} \\ &= \{17, 19, 22, 19\} \end{aligned}$$

$$\begin{aligned} X_1(k) &= \{7, -2-j, 1, -2+j\} \\ X_2(k) &= \{11, 2-j, 1, 2+j\} \\ \Rightarrow X_3(k) &= X_1(k)X_2(k) \\ &= \{17, 19, 22, 19\} \end{aligned}$$

Question 6

A linear time-invariant system with frequency response $H(\omega)$ is excited with the periodic input

$$x(n) = \sum_{k=-\infty}^{\infty} \delta(n-kN)$$

Suppose that we compute the N-point DFT $Y(k)$ of the samples $y(n)$, $0 \le n \le N-1$ of the output sequence. How is $Y(k)$ related to $H(\omega)$?

Solution

$$\begin{aligned} x(n) &= \sum_{i=-\infty}^{\infty} \delta(n - iN) \\ y(n) &= \sum_{m} h(m)x(n-m) \\ &= \sum_{m} h(m)\left[\sum_{i} \delta(n-m-iN)\right] \\ &= \sum_{i} h(n-iN) \end{aligned}$$

Therefore, $y(.)$ is a periodic sequence with period N. So

$$\begin{aligned} Y(k) &= \sum_{n=0}^{N-1} y(n)W_N^{kn} \\ &= H(w)|_{w=\frac{2\pi}{N}k} \end{aligned}$$

$$Y(k) = H(\frac{2\pi k}{N}) \quad k = 0, 1, \ldots, N-1$$

FAST-FOURIER-TRANSFORM (FFT) ALGORITHMS

3.1 Digital filtering using DFT

In a LTI system the system response is got by convoluting the input with the impulse response. In the frequency domain their respective spectra are multiplied. These spectra are continuous and hence cannot be used for computations. The product of 2 DFT s is equivalent to the circular convolution of the corresponding time domain sequences. Circular convolution cannot be used to determine the output of a linear filter to a given input sequence. In this case a frequency domain methodology equivalent to linear convolution is required. Linear convolution can be implemented using circular convolution by taking the length of the convolution as N >= n1+n2-1 where n1 and n2 are the lengths of the 2 sequences.

3.1.1 Overlap and add

In order to convolve a short duration sequence with a long duration sequence x(n) ,x(n) is split into blocks of length N x(n) and h(n) are zero padded to length L+M-1 . circular convolution is performed to each block then the results are added. These data blocks may be represented as

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

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

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

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 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 method. This overlapping and adding yields the output sequences given below.

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

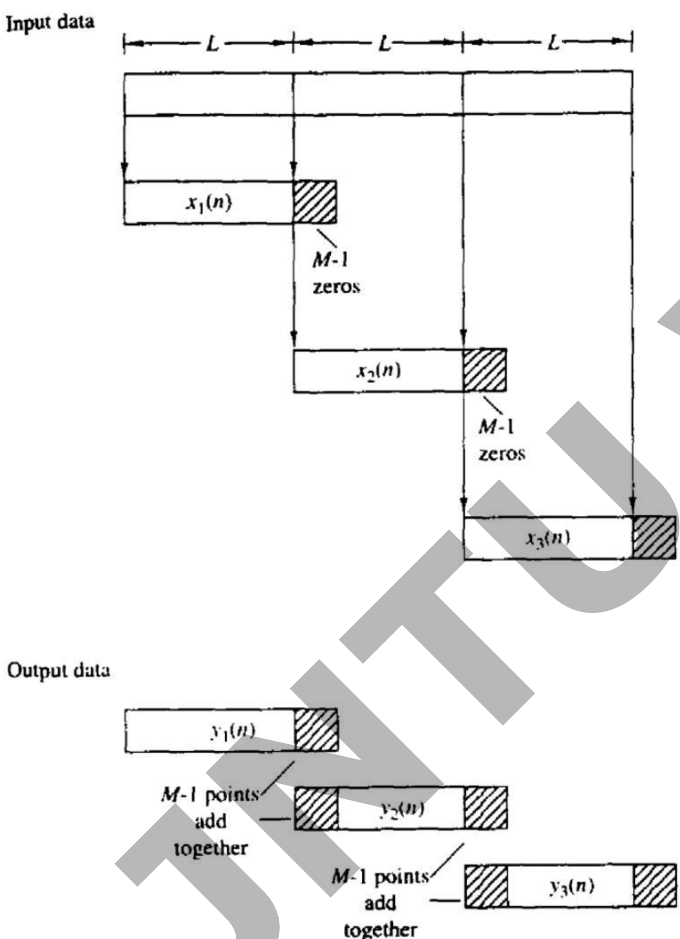


Figure 5.11 Linear FIR filtering by the overlap-add method.

2.1.2 Overlap and save method

In this method x (n) is divided into blocks of length N with an overlap of k-1 samples. The first block is zero padded with k-1 zeros at the beginning. H (n) is also zero padded to length N. Circular convolution of each block is performed using the N length DFT .The output signal is obtained after discarding the first k-1 samples the final result is obtained by adding the intermediate results.

In this method the size of the I/P data blocks is N= L+M-1 and the size of the DFts and IDFTs 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 from 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 two N-point DFTs {H(k)} and {Xm(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

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

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

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

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

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

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

and so forth. The resulting data sequences from the IDFT are given by (5.3.8), 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 are graphically illustrated in Fig. 5.10.

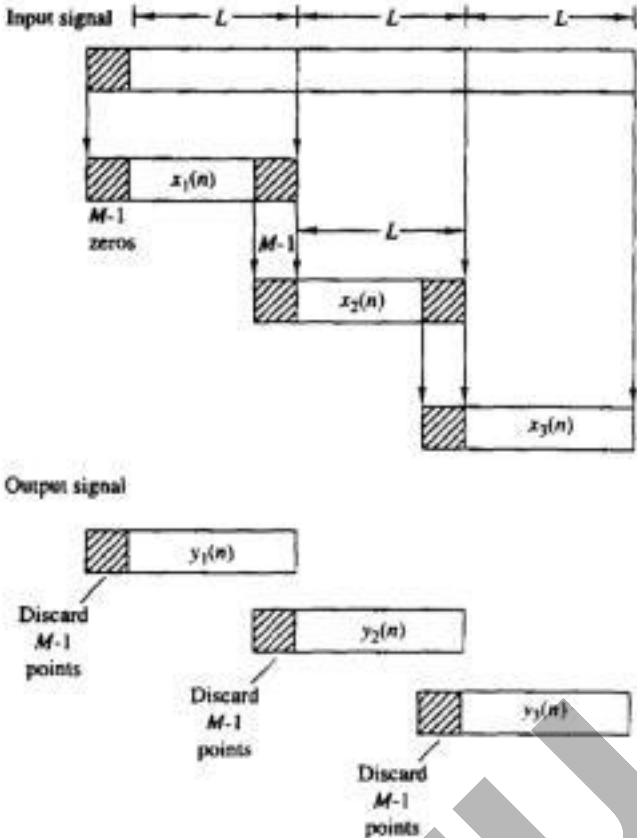


Figure 5.10 Linear FIR filtering by the overlap-save method.

3.2 Direct Computation of DFT

The problem:
Given signal samples: x[0], . . . , x[N - 1] (some of which may be zero), develop a procedure to compute

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

for k = 0, . . . , N - 1 where

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

We would like the procedure to be fast, simple, and accurate. Fast is the most important, so we will sacrifice simplicity for speed, hopefully with minimal loss of accuracy

3.3 Need for efficient computation of DFT (FFT Algorithms)

Let us start with the simple way. Assume that W_N^{kn} has been precompiled and stored in a table for the N of interest. How big should the table be? W_N^m is periodic in m with period N, so we just need to tabulate the N values:

$$W_N^m = \cos\left(\frac{2\pi}{N}m\right) - \jmath \sin\left(\frac{2\pi}{N}m\right)$$

(Possibly even less since Sin is just Cos shifted by a quarter periods, so we could save just Cos when N is a multiple of 4.)

Why tabulate? To avoid repeated function calls to Cos and sin when computing the DFT. Now we can compute each X[k] directly form the formula as follows

$$X[k] = \sum_{n=0}^{N-1} x[n]\, W_N^{kn} = x[0]\, W_N^0 + x[1]\, W_N^k + x[2]\, W_N^{2k} + \cdots + x[N-1]\, W_N^{(N-1)k}.$$

For each value of k, there are N complex multiplications, and (N-1) complex additions. There are N values of k, so the total number of complex operations is

$$N \cdot N + N(N-1) = 2N^2 - N = O(N^2).$$

Complex multiplies require 4 real multiplies and 2 real additions, whereas complex additions require just 2 real additions N^2 complex multiplies are the primary concern.

N^2 increases *rapidly* with N, so how can we reduce the amount of computation? By exploiting the following properties of W:

- Symmetry property: $W_N^{k+N/2} = -W_N^k = e^{j\pi}\, W_N^k$
- Periodicity property: $W_N^{k+N} = W_N^k$
- Recursion property: $W_N^2 = W_{N/2}$

The first and third properties hold for even N, *i.e.*, when 2 is one of the prime factors of N. There are related properties for other prime factors of N.

Divide and conquer approach

We have seen in the preceding sections that the DFT is a very computationally intensive operation. In 1965, Cooley and Tukey published an algorithm that could be used to compute the DFT much more efficiently. Various forms of their algorithm, which came to be known as the Fast Fourier Transform (FFT), had actually been developed much earlier by other mathematicians (even dating back to Gauss). It was their paper, however, which stimulated a revolution in the field of signal processing.

It is important to keep in mind at the outset that the FFT is *not* a new transform. It is simply a very efficient way to compute an existing transform, namely the DFT. As we saw, a straight forward implementation of the DFT can be computationally expensive because the number of multiplies grows as the square of the input length (i.e. N^2 for an N point DFT). The FFT reduces this computation using two simple but important concepts. The first concept, known as divide-and-conquer, splits the problem into two smaller problems. The second concept, known as recursion, applies this divide-and-conquer method repeatedly until the problem is solved.

Recommended Questions with solutions

Question1

A designer has available a number of eight-point FFT chips. Show explicitly how he should interconnect three such chips in order to compute a 24-point DFT.

Solution:-

Create three subsequences of 8-pts each

$$\begin{aligned} Y(k) &= \sum_{n=0,3,6,\ldots}^{21} y(n)W_N^{kn} + \sum_{n=1,4,7,\ldots}^{22} y(n)W_N^{kn} + \sum_{n=2,5,\ldots}^{23} y(n)W_N^{kn} \\ &= \sum_{i=0}^{7} y(3i)W_{\frac{N}{3}}^{ki} + \sum_{i=0}^{7} y(3i+1)W_{\frac{N}{3}}^{ki}W_N^{k} + \sum_{i=0}^{7} y(3i+2)W_{\frac{N}{3}}^{ki}W_N^{2k} \\ &\triangleq Y_1(k) + W_N^k Y_2(k) + W_N^{2k} Y_3(k) \end{aligned}$$

where Y_1, Y_2, Y_3 represent the 8-pt DFTs of the subsequences.

Question 2

Let $x(n)$ be a real-valued N-point ($N = 2^\nu$) sequence. Develop a method to compute an N-point DFT $X'(k)$, which contains only the odd harmonics [i.e., $X'(k) = 0$ if k is even] by using only a real $N/2$-spoint DFT.

Solution:-

$$\begin{aligned}
X(k) &= \sum_{n=0}^{N-1} x(n)W_N^{kn} \quad 0 \le k \le N-1 \\
&= \sum_{n=0}^{\frac{N}{2}-1} x(n)W_N^{kn} + \sum_{n=\frac{N}{2}}^{N-1} x(n)W_N^{kn} \\
&= \sum_{n=0}^{\frac{N}{2}-1} x(n)W_N^{kn} + \sum_{r=0}^{\frac{N}{2}-1} x(r+\frac{N}{2})W_N^{(r+\frac{N}{2})k} \\
\text{Let} X'(k') &= X(2k+1), \quad 0 \le k' \le \frac{N}{2}-1 \\
\text{Then, } X'(k') &= \sum_{n=0}^{\frac{N}{2}-1} \left[x(n)W_N^{(2k'+1)n} + x(n+\frac{N}{2})W_N^{(n+\frac{N}{2})(2k'+1)} \right] \\
\text{Using the fact that } W_N^{2k'n} &= W_{\frac{N}{2}}^{k'n}, \quad W_N^N = 1 \\
X'(k') &= \sum_{n=0}^{\frac{N}{2}-1} \left[x(n)W_N^n W_{\frac{N}{2}}^{k'n} + x(n+\frac{N}{2})W_{\frac{N}{2}}^{k'n} W_N^n W_N^{\frac{N}{2}} \right] \\
&= \sum_{n=0}^{\frac{N}{2}-1} \left[x(n) - x(n+\frac{N}{2}) \right] W_N^n W_{\frac{N}{2}}^{k'n}
\end{aligned}$$

Question 3

The z-transform of the sequence $x(n) = u(n) - u(n-7)$ is sampled at five points on the unit circle as follows

$$x(k) = X(z)|_{z} = e^{j2\pi k/5} \qquad k = 0, 1, 2, 3, 4$$

Solution:-

$$\begin{aligned}
X(z) &= 1 + z^{-1} + \ldots + z^{-6} \\
X(k) &= X(z)|_{z=e^{j\frac{2\pi}{5}}} \\
&= 1 + e^{-j\frac{2\pi}{5}} + e^{-j\frac{4\pi}{5}} + \ldots + e^{-j\frac{12\pi}{5}} \\
&= 2 + 2e^{-j\frac{2\pi}{5}} + e^{-j\frac{4\pi}{5}} + \ldots + e^{-j\frac{8\pi}{5}} \\
x'(n) &= \{2, 2, 1, 1, 1\} \\
x'(n) &= \sum_m x(n+7m), \qquad n = 0, 1, \ldots, 4
\end{aligned}$$

Temporal aliasing occurs in first two points of $x'(n)$ because $X(z)$ is not sampled at sufficiently small spacing on the unit circle.

Question 4

Consider a finite-duration sequence $x(n)$, $0 \le n \le 7$, with z-transform $X(z)$. We wish to compute $X(z)$ at the following set of values:

$$z_k = 0.8e^{j[(2\pi k/8)+(\pi/8)]} \qquad 0 \le k \le 7$$

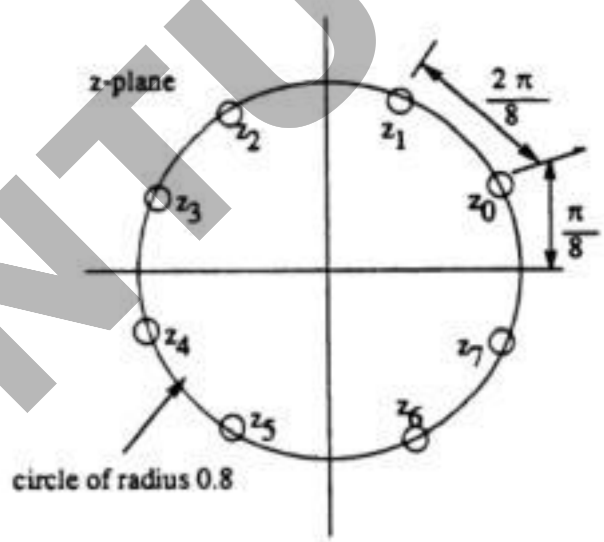
(a) Sketch the points $\{z_k\}$ in the complex plane.

(b) Determine a sequence $s(n)$ such that its DFT provides the desired samples of $X(z)$.

Solution:- (a)

$$Z_k = 0.8e^{j\left[\frac{2\pi k}{8}+\frac{\pi}{8}\right]}$$

(b)



$$\begin{aligned} X(k) &= X(z)|_{z=z_k} \\ &= \sum_{n=0}^{7} x(n)\left[0.8e^{j\left[\frac{2\pi k}{8}+\frac{\pi}{8}\right]}\right]^{-n} \\ s(n) &= x(n)\,0.8e^{-j\frac{\pi}{8}n} \end{aligned}$$

RADIX-2 FFT ALGORITHM FOR THE COMPUTATION OF DFT AND IDFT

4.1 Introduction:

Standard frequency analysis requires transforming time-domain signal to frequency domain and studying Spectrum of the signal. This is done through DFT computation. N-point DFT computation results in N frequency components. We know that DFT computation through FFT requires N/2 $\log_2 N$ complex multiplications and N $\log_2 N$ additions. In certain applications not all N frequency components need to be computed (an application will be discussed). If the desired number of values of the DFT is less than 2 $\log_2 N$ than direct computation of the desired values is more efficient that FFT based computation.

4.2 Radix-2 FFT

Useful when N is a power of 2: $N = r^v$ for integers r and v. 'r' is called the **radix**, which comes from the Latin word meaning .a root, and has the same origins as the word radish.

When N is a power of r = 2, this is called **radix-2**, and the natural .divide and conquer approach. is to split the sequence into two sequences of length N=2. This is a very clever trick that goes back many years.

4.2.1 Decimation in time

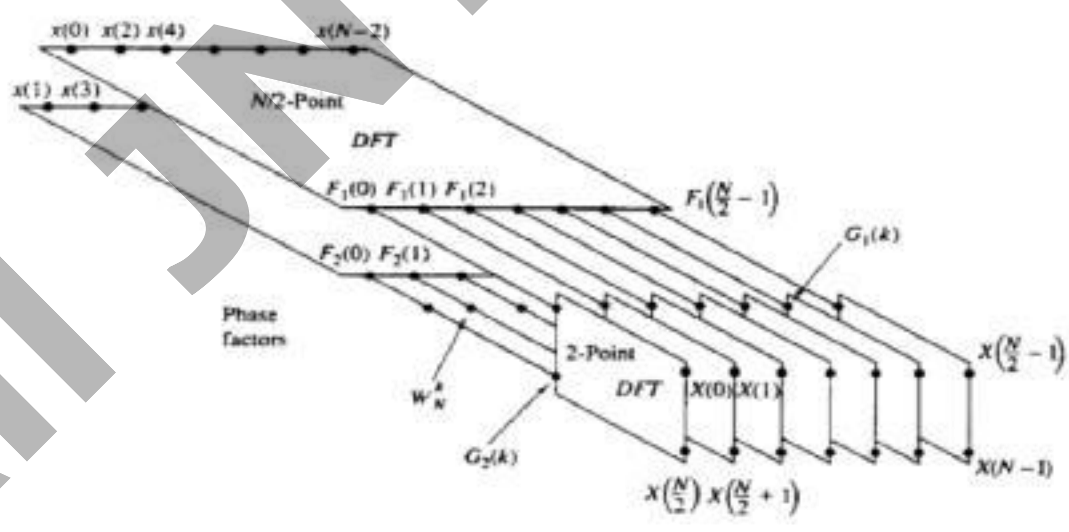
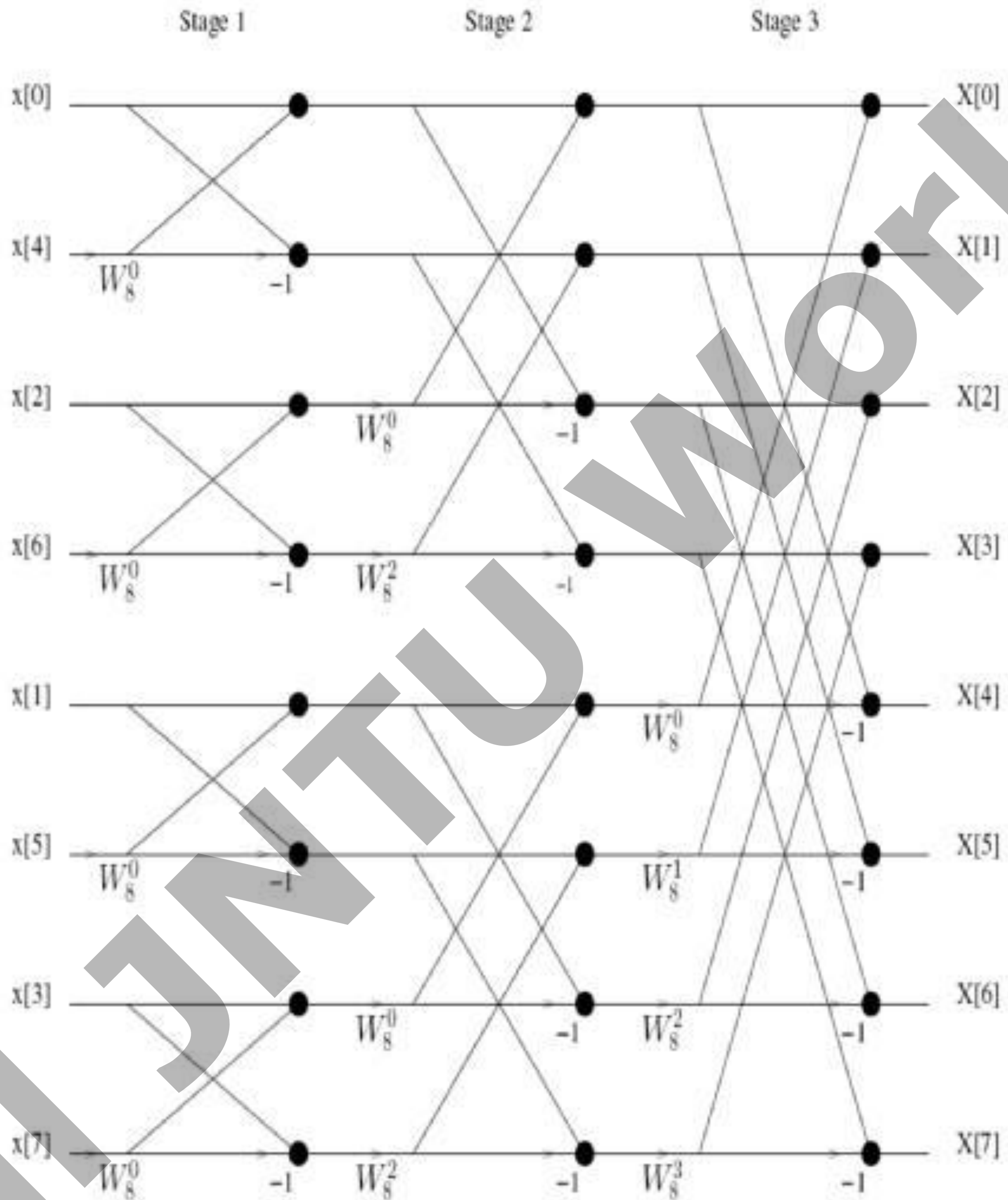


Fig 4.1 First step in Decimation-in-time domain Algorithm

N = 8-point decimation-in-time FFT algorithm



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

Each dot represents a complex addition.

Each arrow represents a complex multiplication.

4.2.2 Decimation-in-frequency Domain

Another important radix-2 FFT algorithm, called decimation-in-frequency algorithm is obtained by using divide-and-conquer approach with the choice of M=2 and L= N/2.This choice of data implies a column-wise storage of the input data sequence. 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

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

Since $W_N^{kN/2} = (-1)^k$, the expression (6.1.33) can be rewritten as

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

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

$$X(2k) = \sum_{n=0}^{(N/2)-1} \left[x(n) + x\left(n + \frac{N}{2}\right)\right] W_{N/2}^{kn} \qquad k = 0, 1, \ldots, \frac{N}{2} - 1 \qquad (6.1.35)$$

and

$$X(2k+1) = \sum_{n=0}^{(N/2)-1} \left\{\left[x(n) - x\left(n + \frac{N}{2}\right)\right] W_N^n\right\} W_{N/2}^{kn} \qquad k = 0, 1, \ldots, \frac{N}{2} - 1 \qquad (6.1.36)$$

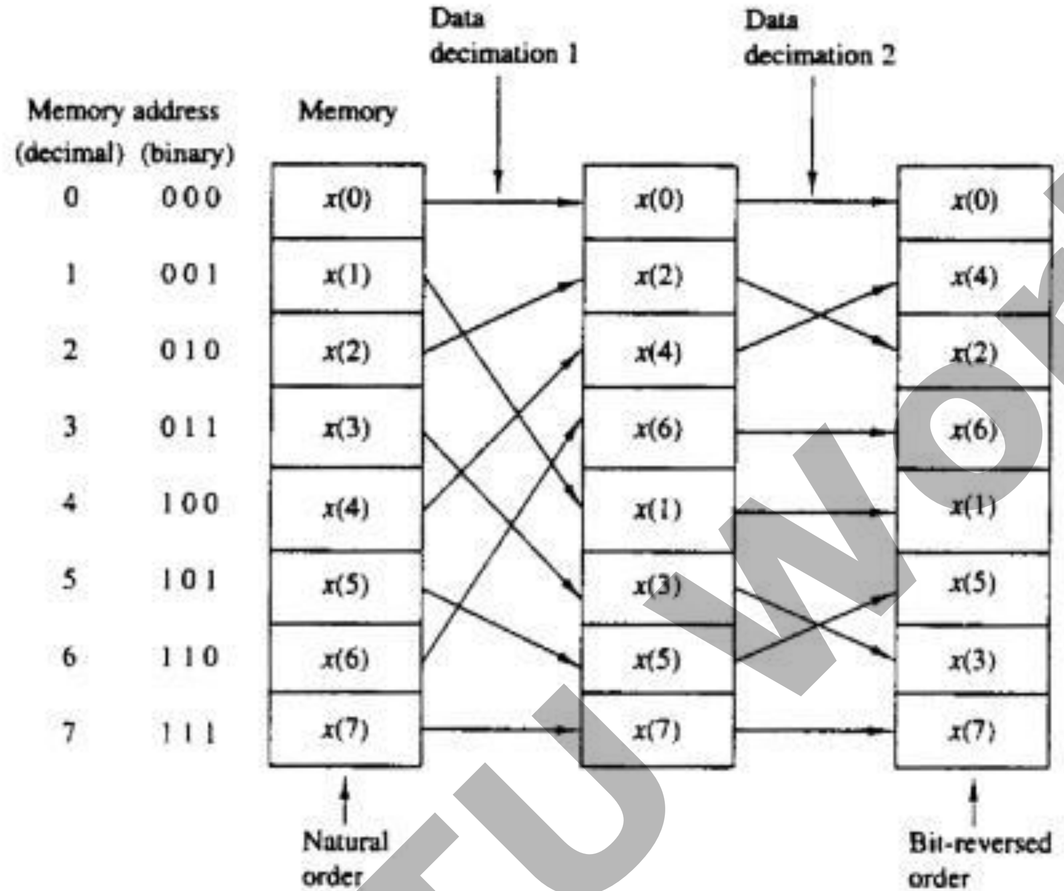
where we have used the fact that $W_N^2 = W_{N/2}$.

If we define the $N/2$-point sequences $g_1(n)$ and $g_2(n)$ as

$$g_1(n) = x(n) + x\left(n + \frac{N}{2}\right)$$
$$g_2(n) = \left[x(n) - x\left(n + \frac{N}{2}\right)\right] W_N^n \qquad n = 0, 1, 2, \ldots, \frac{N}{2} - 1$$

then

$$X(2k) = \sum_{n=0}^{(N/2)-1} g_1(n) W_{N/2}^{kn}$$
$$X(2k+1) = \sum_{n=0}^{(N/2)-1} g_2(n) W_{N/2}^{kn}$$



(a)

$(n_2n_1n_0)$	→	$(n_0n_2n_1)$	→	$(n_0n_1n_2)$
(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)

(b)

Fig 4.2 Shuffling of Data and Bit reversal

The computation of the sequences g1 (n) and g2 (n) and subsequent use of these sequences to compute the N/2-point DFTs depicted in fig we observe that the basic computation in this figure involves the butterfly operation.

The computation procedure can be repeated through decimation of the N/2-point DFTs, X(2k) and X(2k+1). The entire process involves $v = \log_2 N$ of decimation, where each stage involves N/2 butterflies of the type shown in figure 4.3.

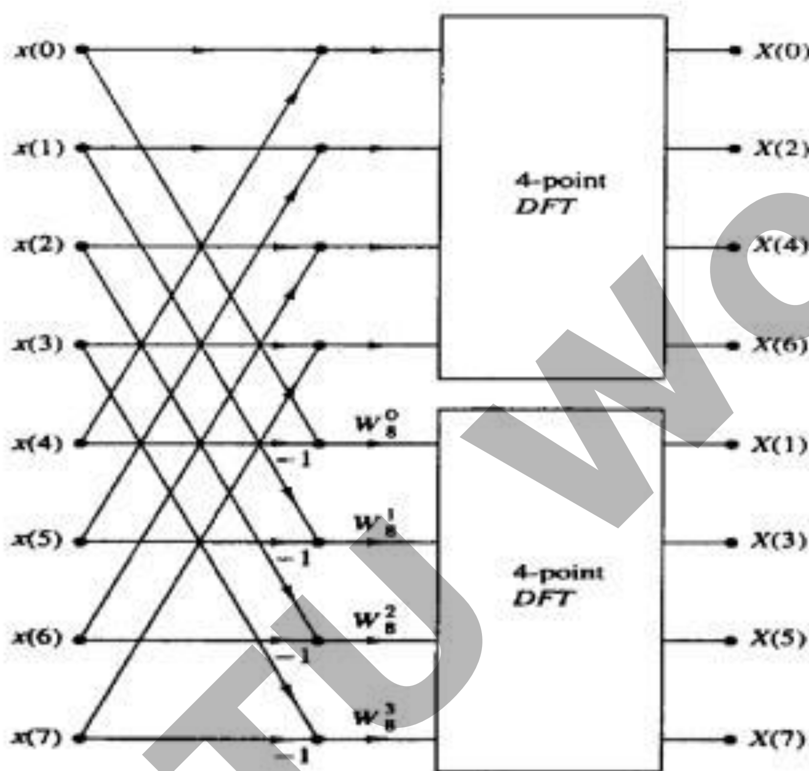


Fig 4.3 First step in Decimation-in-time domain Algorithm

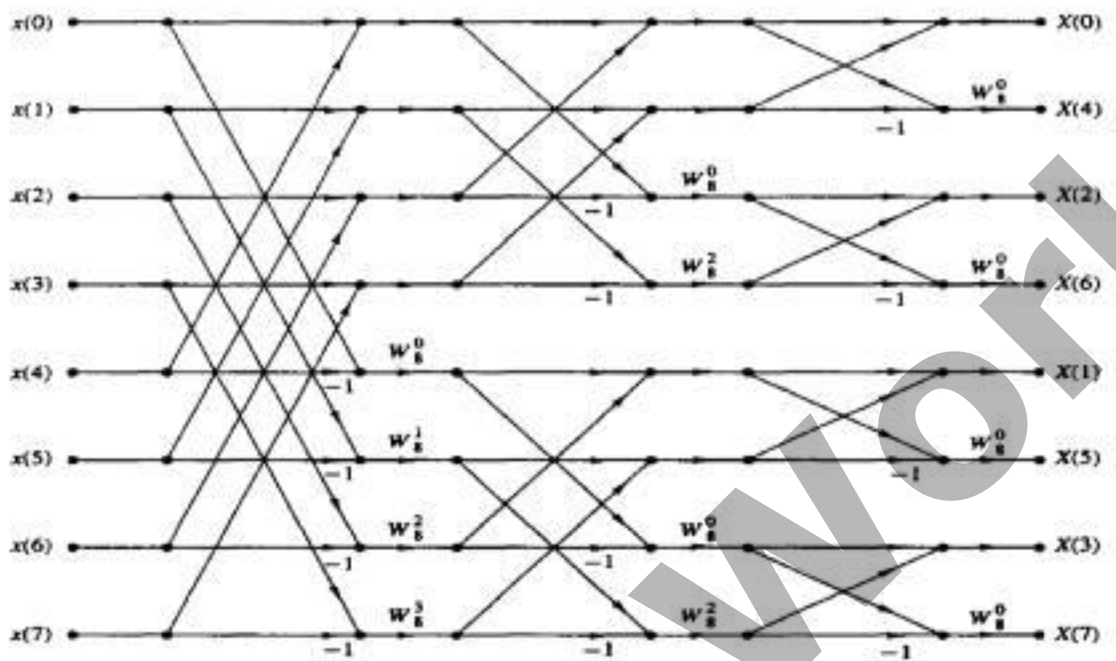
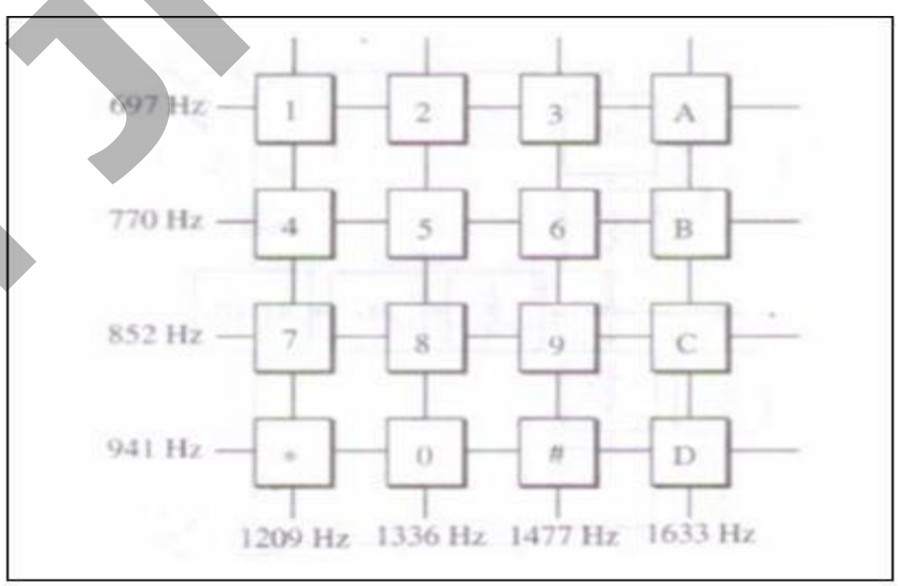


Fig 4.4 N=8 point Decimation-in-frequency domain Algorithm

4.2 Example: DTMF – Dual Tone Multi frequency

This is known as touch-tone/speed/electronic dialing, pressing of each button generates a unique set of two-tone signals, called DTMF signals. These signals are processed at exchange to identify the number pressed by determining the two associated tone frequencies. Seven frequencies are used to code the 10 decimal digits and two special characters (4x3 array)



In this application frequency analysis requires determination of possible seven (eight) DTMF fundamental tones and their respective second harmonics .For an 8 kHz sampling freq, the best value of the DFT length N to detect the eight fundamental DTMF tones has been found to be 205 .Not all 205 freq components are needed here, instead only those corresponding to key frequencies are required. FFT algorithm is not effective and efficient in this application. The direct computation of the DFT which is more effective in this application is formulated as a linear filtering operation on the input data sequence.

This algorithm is known as Goertzel Algorithm

This algorithm exploits periodicity property of the phase factor. Consider the DFT definition

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

Since W_N^{-kN} is equal to 1, multiplying both sides of the equation by this results in;

$$X(k) = W_N^{-kN} \sum_{m=0}^{N-1} x(m) W_N^{mk} = \sum_{m=0}^{N-1} x(m) W_N^{-k(N-m)} \qquad (2)$$

This is in the form of a convolution $y_k(n) = x(n) * h_k(n)$

$$y_k(n) = \sum_{m=0}^{N-1} x(m) W_N^{-k(n-m)} \qquad (3)$$

$$h_k(n) = W_N^{-kn} u(n) \qquad (4)$$

Where $y_k(n)$ is the out put of a filter which has impulse response of $h_k(n)$ and input x(n).

The output of the filter at n = N yields the value of the DFT at the freq $\omega_k = 2\pi k/N$

The filter has frequency response given by

$$H_k(z) = \frac{1}{1 - W_N^{-k} z^{-1}} \qquad (6)$$

The above form of filter response shows it has a pole on the unit circle at the frequency $\omega_k = 2\pi k/N$.

Entire DFT can be computed by passing the block of input data into a parallel bank of N single-pole filters (resonators)

The above form of filter response shows it has a pole on the unit circle at the frequency $\omega_k = 2\pi k/N$.

Entire DFT can be computed by passing the block of input data into a parallel bank of N single-pole filters (resonators)

1.3 Difference Equation implementation of filter:

From the frequency response of the filter (eq 6) we can write the following difference equation relating input and output;

$$H_k(z) = \frac{Y_k(z)}{X(z)} = \frac{1}{1 - W_N^{-k} z^{-1}}$$

$$y_k(n) = W_N^{-k} y_k(n-1) + x(n) \qquad y_k(-1) = 0 \qquad (7)$$

The desired output is $X(k) = y_k(n)$ for k = 0,1,…N-1. The phase factor appearing in the difference equation can be computed once and stored.

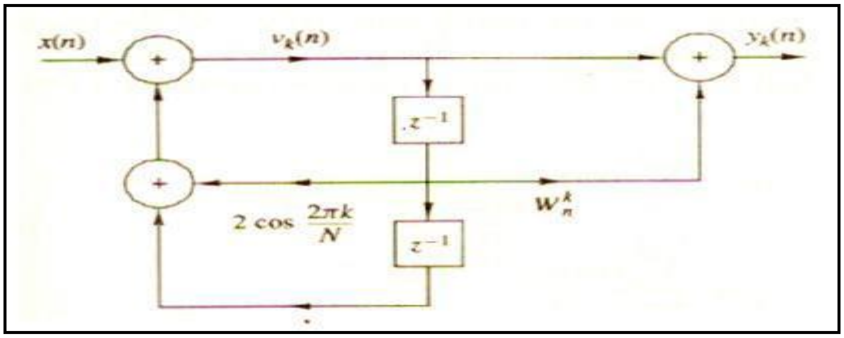
The form shown in eq (7) requires complex multiplications which can be avoided doing suitable modifications (divide and multiply by $1 - W_N^k z^{-1}$). Then frequency response of the filter can be alternatively expressed as

$$H_k(z) = \frac{1 - W_N^k z^{-1}}{1 - 2\cos(2\pi k/N) z^{-1} + z^{-2}} \qquad (8)$$

This is second –order realization of the filter (observe the denominator now is a second-order expression). The direct form realization of the above is given by

$$v_k(n) = 2\cos(2\pi k/N) v_k(n-1) - v_k(n-2) + x(n) \qquad (9)$$

$$y_k(n) = v_k(n) - W_N^k v_k(n-1) \qquad v_k(-1) = v_k(-2) = 0 \qquad (10)$$



The recursive relation in (9) is iterated for n = 0,1,……N, but the equation in (10) is computed only once at time n =N. Each iteration requires one real multiplication and two additions. Thus, for a real input sequence x(n) this algorithm requires (N+1) real multiplications to yield X(k) and X(N-k) (this is due to symmetry). Going through the Goertzel algorithm it is clear that this algorithm is useful only when M out of N DFT values need to be computed where $M \leq 2\log_2 N$, Otherwise, the FFT algorithm is more efficient method. The utility of the algorithm completely depends on the application and number of frequency components we are looking for.

4.2. Chirp z- Transform

4.2.1 Introduction:

Computation of DFT is equivalent to samples of the z-transform of a finite-length sequence at equally spaced points around the unit circle. The spacing between the samples is given by 2π/N. The efficient computation of DFT through FFT requires N to be a highly composite number which is a constraint. Many a times we may need samples of z-transform on contours other than unit circle or we my require dense set of frequency samples over a small region of unit circle. To understand these let us look in to the following situations:

1. Obtain samples of z-transform on a circle of radius 'a' which is concentric to unit circle The possible solution is to multiply the input sequence by a^{-n}
2. 128 samples needed between frequencies $\omega = -\pi/8$ to $+\pi/8$ from a 128 point sequence

From the given specifications we see that the spacing between the frequency samples is π/512 or 2π/1024. In order to achieve this freq resolution we take 1024- point FFT of the given 128-point seq by appending the sequence with 896 zeros. Since we need only 128 frequencies out of 1024 there will be big wastage of computations in this scheme.

For the above two problems Chirp z-transform is the alternative.

Chirp z- transform is defined as:

$$X(z_k) = \sum_{n=0}^{N-1} x(n) z_k^{-n} \qquad k = 0,1,......L-1 \qquad (11)$$

Where z_k is a generalized contour. z_k is the set of points in the z-plane falling on an arc which begins at some point z_0 and spirals either in toward the origin or out away from the origin such that the points $\{z_k\}$are defined as,

$$z_k = r_0 e^{j\theta_0} (R_0 e^{j\phi_0})^k \qquad k = 0,1,....L-1 \qquad (12)$$

Note that,

a. if $R_0 < 1$ the points fall on a contour that spirals toward the origin

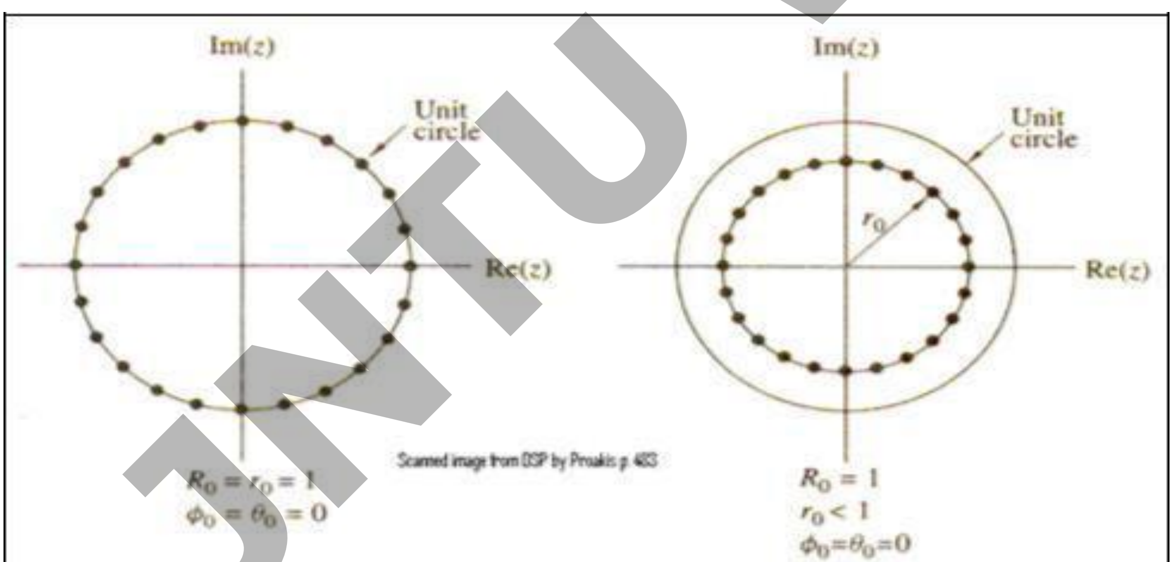
b. If $R_0 > 1$ the contour spirals away from the origin

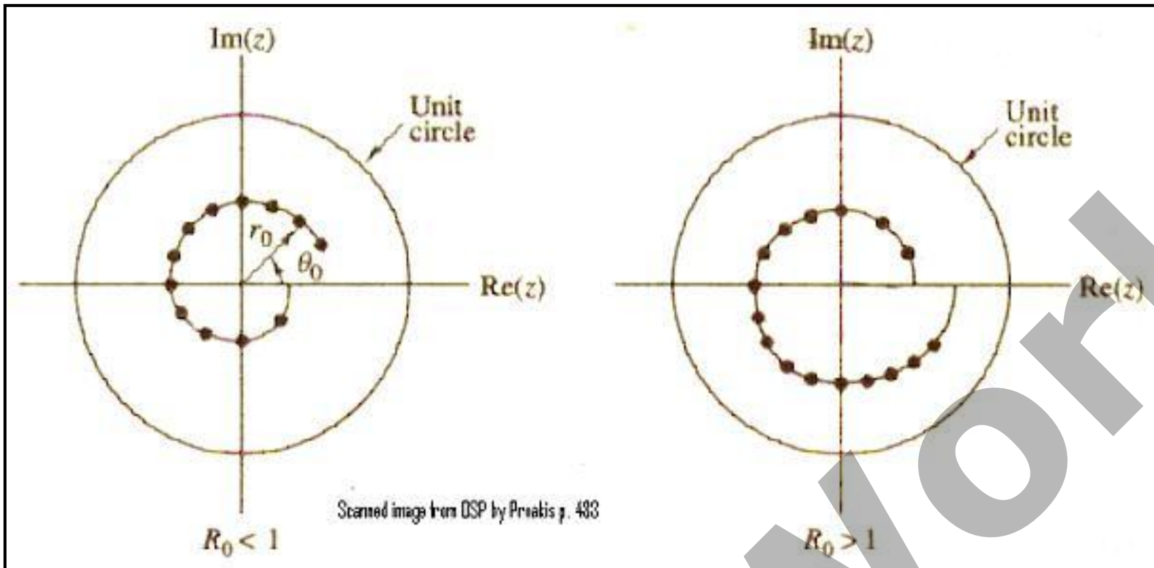
c. If $R_0 = 1$ the contour is a circular arc of radius

d.If $r_0=1$ and $R_0=1$ the contour is an arc of the unit circle.

(Additionally this contour allows one to compute the freq content of the sequence x(n) at dense set of L frequencies in the range covered by the arc without having to compute a large DFT (i.e., a DFT of the sequence x(n) padded with many zeros to obtain the desired resolution in freq.))

e. If $r_0 = R_0 = 1$ and $\theta_0 = 0$ $\Phi_0 = 2\pi/N$ and L = N the contour is the entire unit circle similar to the standard DFT. These conditions are shown in the following diagram.





Substituting the value of z_k in the expression of $X(z_k)$

$$X(z_k) = \sum_{n=0}^{N-1} x(n) z_k^{-n} = \sum_{n=0}^{N-1} x(n)(r_0 e^{j\theta_0})^{-n} \, {}^{-nk} \qquad (13)$$

where $\quad W = R_0 e^{j\phi_0} \qquad (14)$

4.2.2 Expressing computation of X(z_k) as linear filtering operation:

By substitution of

$$nk = \frac{1}{2}(n^2 + k^2 - (k-n)^2) \qquad (15)$$

we can express X(z_k) as

$$X(z_k) = W^{-k^2/2} y(k) = y(k)/h(k) \qquad k = 0,1,\ldots\ldots L-1 \qquad (16)$$

Where

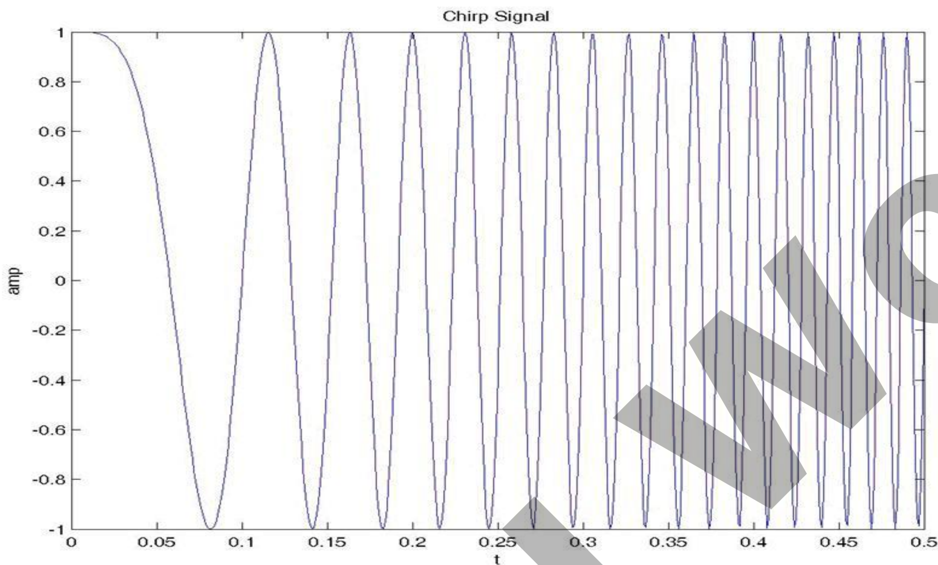
$$h(n) = W^{n^2/2} \qquad g(n) = x(n)(r_0 e^{j\theta_0})^{-n} W^{-n^2/2}$$

$$y(k) = \sum_{n=0}^{N-1} g(n) h(k-n) \qquad (17)$$

both g(n) and h(n) are complex valued sequences

4.2.3 Why it is called Chirp z-transform?

If $R_0 = 1$, then sequence h(n) has the form of complex exponential with argument $\omega n = n^2 \Phi_0/2 = (n\,\Phi_0/2)\,n$. The quantity $(n\,\Phi_0/2)$ represents the freq of the complex exponential signal, which increases linearly with time. Such signals are used in radar systems are called chirp signals. Hence the name chirp z-transform.



4.2.4 How to Evaluate linear convolution of eq (17)

1. Can be done efficiently with FFT
2. The two sequences involved are g(n) and h(n). g(n) is finite length seq of length N and h(n) is of infinite duration, but fortunately only a portion of h(n) is required to compute L values of X(z), hence FFT could be still be used.
3. Since convolution is via FFT, it is circular convolution of the N-point seq g(n) with an M- point section of h(n) where M > N
4. The concepts used in overlap –save method can be used
5. While circular convolution is used to compute linear convolution of two sequences we know the initial N-1 points contain aliasing and the remaining points are identical to the result that would be obtained from a linear convolution of h(n) and g(n), In view of this the DFT size selected is M = L+N-1 which would yield L valid points and N-1 points corrupted by aliasing. The section of h(n) considered is for $-(N-1) \leq n \leq (L-1)$ yielding total length M as defined
6. The portion of h(n) can be defined in many ways, one such way is,

$h_1(n) = h(n-N+1)$ n = 0,1,…..M-1 7.

Compute $H_1(k)$ and G(k) to obtain

$Y_1(k) = G(K)H_1(k)$

8. Application of IDFT will give $y_1(n)$, for

n =0,1,…M-1. The starting N-1 are discarded and desired values are $y_1(n)$ for

$N-1 \leq n \leq M-1$ which corresponds to the range $0 \leq n \leq L-1$ i.e.,

$y(n)= y_1(n+N-1)$ n=0,1,2,…..L-1

9. Alternatively $h_2(n)$ can be defined as

$$h_2(n) = h(n) \qquad 0 \leq n \leq L-1$$
$$= h(n-(N+L-1)) \qquad L \leq n \leq M-1$$

10. Compute $Y_2(k) = G(K)H_2(k)$, The desired values of $y_2(n)$ are in the range

$0 \leq n \leq L-1$ i.e.,

$y(n) = y_2(n)$ n=0,1,….L-1

11. Finally, the complex values $X(z_k)$ are computed by dividing y(k) by h(k) For k =0,1,……L-1

4.3 Computational complexity

In general the computational complexity of CZT is of the order of M $\log_2 M$ complex multiplications. This should be compared with N.L which is required for direct evaluation. If L is small direct evaluation is more efficient otherwise if L is large then CZT is more efficient.

4.3.1 Advantages of CZT

a. Not necessary to have N =L

b.Neither N or L need to be highly composite

c.The samples of Z transform are taken on a more general contour that includes the unit circle as a special case.

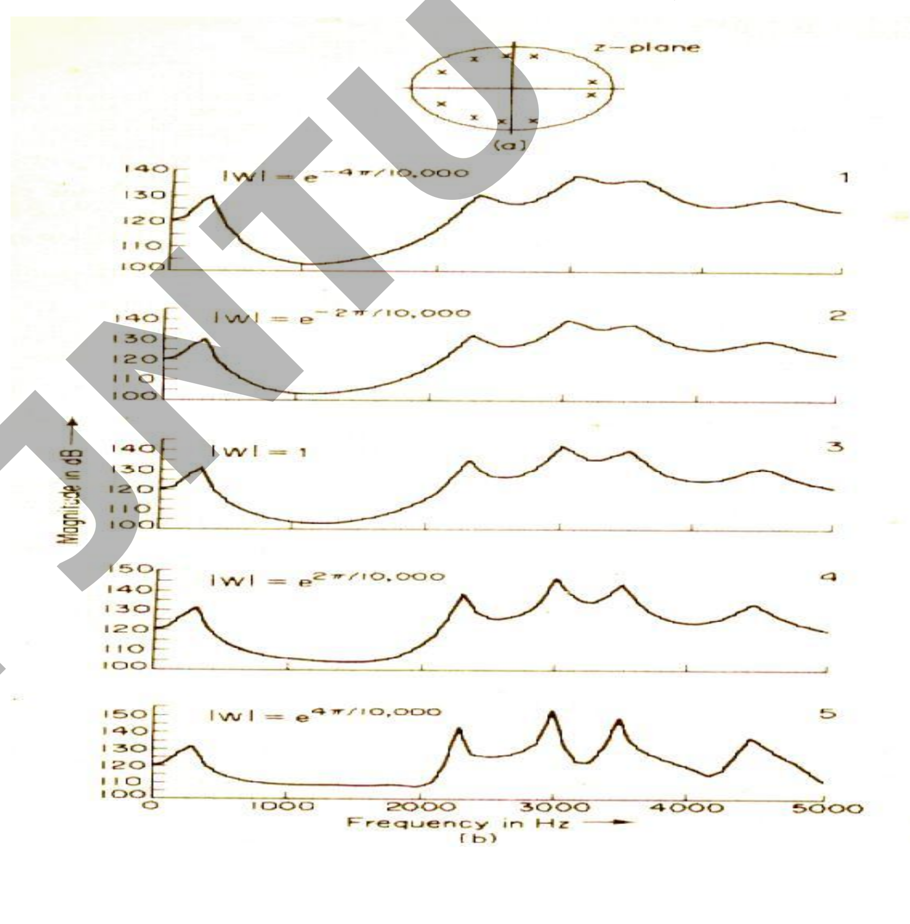
4.4 Example to understand utility of CZT algorithm in freq analysis

(ref: DSP by Oppenheim Schaffer)

CZT is used in this application to sharpen the resonances by evaluating the z-transform off the unit circle. Signal to be analyzed is a synthetic speech signal generated by exciting a five-pole system with a periodic impulse train. The system was simulated to correspond to a sampling freq. of 10 kHz. The poles are located at center freqs of 270,2290,3010,3500 & 4500 Hz with bandwidth of 30, 50, 60,87 & 140 Hz respectively.

Solution: Observe the pole-zero plots and corresponding magnitude frequency response for different choices of |w|. The following observations are in order:

- The first two spectra correspond to spiral contours outside the unit circle with a resulting broadening of the resonance peaks
- |w| = 1 corresponds to evaluating z-transform on the unit circle
- The last two choices correspond to spiral contours which spirals inside the unit circle and close to the pole locations resulting in a sharpening of resonance peaks.

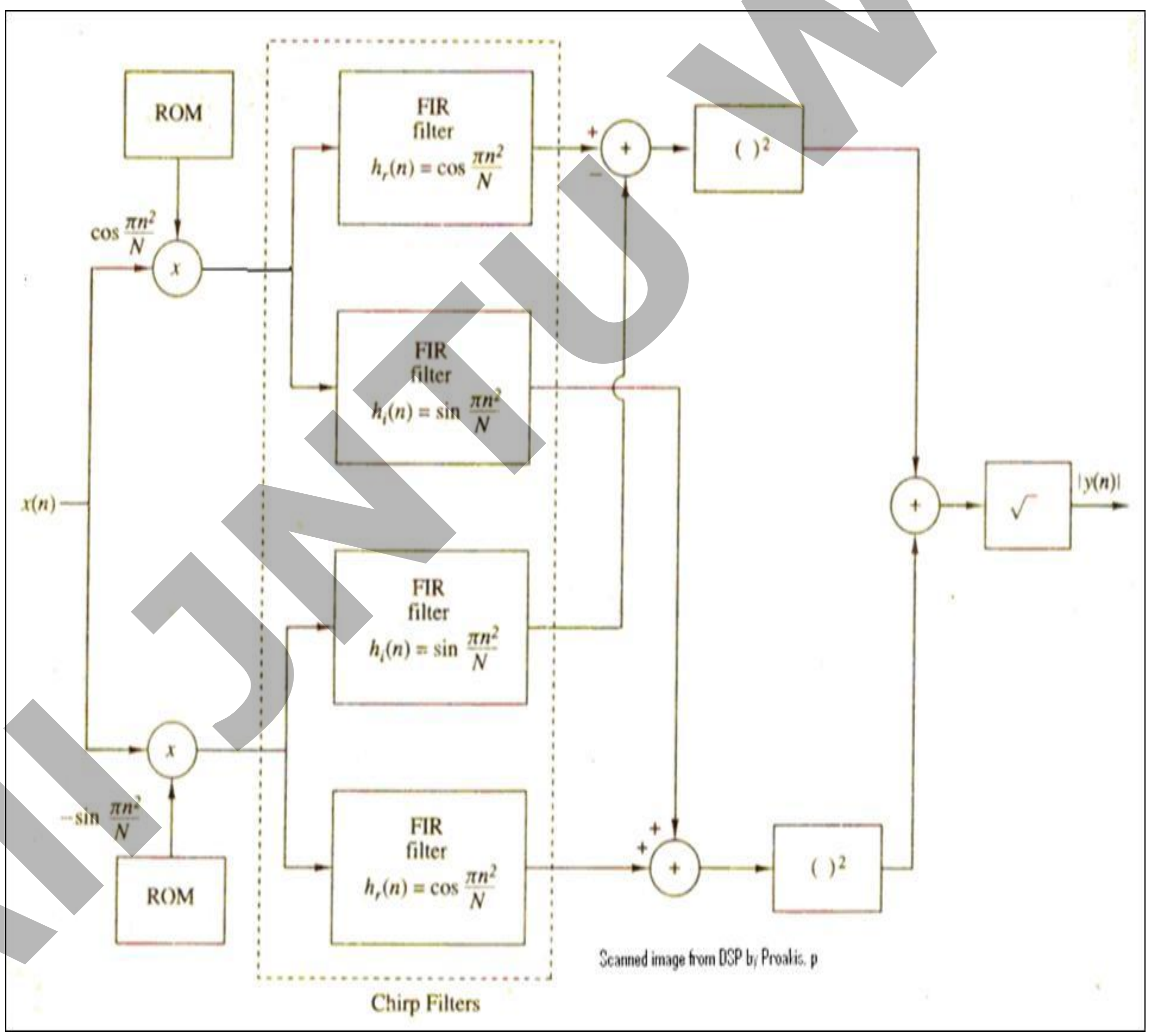


4.5 Implementation of CZT in hardware to compute the DFT signals

The block schematic of the CZT hardware is shown in down figure. DFT computation requires $r_0 = R_0 = 1$, $\theta_0 = 0$ $\Phi_0 = 2\pi/N$ and $L = N$.

The cosine and sine sequences in h(n) needed for pre multiplication and post multiplication are usually stored in a ROM. If only magnitude of DFT is desired, the post multiplications are unnecessary,

In this case $|X(z_k)| = |y(k)|$ k =0,1,….N-1



Recommended Questions with solutions

Question 1

Compute the 16-point DFT of the sequence

$$x(n) = \cos \frac{\pi}{2} n \qquad 0 \le n \le 15$$

using the radix-4 decimation-in-time algorithm.

Solution:-

$$\underline{A} \triangleq \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}$$

$$\underline{x}_1 \triangleq [\ x(0) \quad x(4) \quad x(8) \quad x(12)\ ]^T$$

$$\underline{x}_2 \triangleq [\ x(1) \quad x(5) \quad x(9) \quad x(13)\ ]^T$$

$$\underline{x}_3 \triangleq [\ x(2) \quad x(6) \quad x(10) \quad x(14)\ ]^T$$

$$\underline{x}_4 \triangleq [\ x(3) \quad x(7) \quad x(11) \quad x(15)\ ]^T$$

$$\begin{bmatrix} F(0) \\ F(4) \\ F(8) \\ F(12) \end{bmatrix} = \underline{A}\underline{x}_1 = \begin{bmatrix} 4 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} F(1) \\ F(5) \\ F(9) \\ F(13) \end{bmatrix} = \underline{A}\underline{x}_2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} F(2) \\ F(6) \\ F(10) \\ F(14) \end{bmatrix} = \underline{A}\underline{x}_3 = \begin{bmatrix} -4 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} F(3) \\ F(7) \\ F(11) \\ F(15) \end{bmatrix} = \underline{A}\underline{x}_4 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

As every $F(i) = 0$ except $F(0) = -F(2) = 4$,

$$\begin{bmatrix} z(0) \\ z(7) \\ z(8) \\ z(12) \end{bmatrix} = \Delta z_4 \begin{bmatrix} F(0) \\ F(1) \\ F(2) \\ F(3) \end{bmatrix} = \begin{bmatrix} 0 \\ 8 \\ 0 \\ 8 \end{bmatrix}$$

which means that $X(4) = X(12) = 8$. $X(k) = 0$ for other K.

Question 2

Draw the flow graph for the decimation-in-frequency (DIF) SRFFT algorithm for $N = 16$. What is the number of nontrivial multiplications?

Solution :- There are 20 real , non trial multiplications

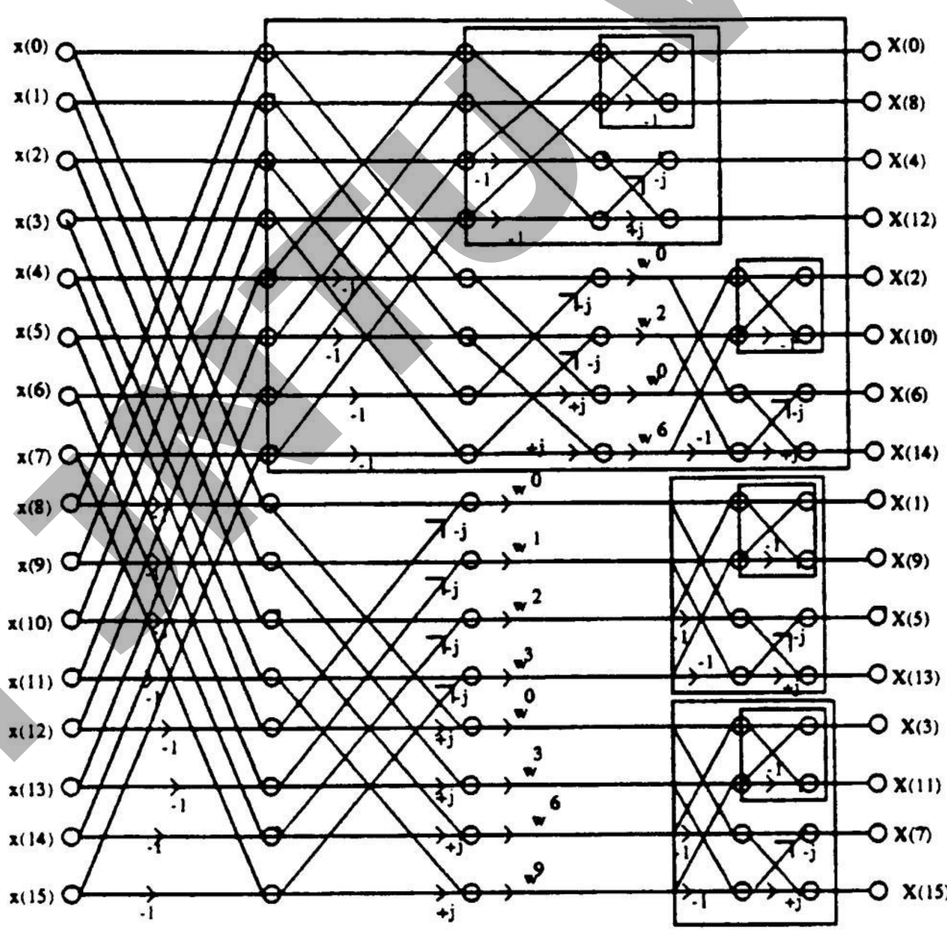


Figure 4.1 DIF Algorithm for N=16

Question 3

Explain how the DFT can be used to compute N equispaced samples of the z-transform, of an N-point sequence, on a circle of radius r.

Solution:-

$$X(z) = \sum_{n=0}^{N-1} x(n) z^{-n}$$

$$\text{Hence, } X(z_k) = \sum_{n=0}^{N-1} x(n) r^{-n} e^{-j\frac{2\pi}{N}kn}$$

where $z_k = re^{-j\frac{2\pi}{N}k}, k = 0, 1, \ldots, N-1$ are the N sample points. It is clear that $X(z_k), k = 0, 1, \ldots, N-1$ is equivalent to the DFT (N-pt) of the sequence $x(n)r^{-n}, n \in [0, N-1]$.

Question 4

Let $X(k)$ be the N-point DFT of the sequence $x(n)$, $0 \le n \le N-1$. What is the N-point DFT of the sequence $s(n) = X(n)$, $0 \le n \le N-1$?

Solution:-

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

Let $F(t)$, $\quad t = 0, 1, \ldots, N-1$ be the DFT of the sequence on k $X(k)$.

$$\begin{aligned} F(t) &= \sum_{k=0}^{N-1} X(k) W_N^{tk} \\ &= \sum_{k=0}^{N-1} \left[\sum_{n=0}^{N-1} x(n) W_N^{kn} \right] W_N^{tk} \\ &= \sum_{n=0}^{N-1} x(n) \left[\sum_{k=0}^{N-1} W_N^{k(n+t)} \right] \\ &= \sum_{n=0}^{N-1} x(n) \delta(n+t)_{\text{mod N}} \\ &= \sum_{n=0}^{N-1} x(n) \delta(N-1-n-t) \qquad t = 0, 1, \ldots, N-1 \\ &= \{x(N-1), x(N-2), \ldots, x(1), x(0)\} \end{aligned}$$

Question 5

Develop a radix-3 decimation-in-time FFT algorithm for $N = 3^v$ and draw the corresponding flow graph for $N = 9$. What is the number of required complex multiplications? Can the operations be performed in place?

Solution:-

$$\begin{aligned} Y(k) &= \sum_{n=0}^{8} y(n)W_9^{nk} \\ &= \sum_{n=0,3,6} y(n)W_9^{nk} + \sum_{n=1,4,7} y(n)W_9^{nk} + \sum_{n=2,5,8} y(n)W_9^{nk} \\ &= \sum_{m=0}^{2} y(3m)W_9^{3km} + \sum_{m=0}^{2} y(3m+1)W_9^{(3m+1)k} + \sum_{m=0}^{2} y(3m+2)W_9^{(3m+2)k} \\ &= \sum_{m=0}^{2} y(3m)W_3^{km} + \sum_{m=0}^{2} y(3m+1)W_3^{mk}W_9^{k} + \sum_{m=0}^{2} y(3m+2)W_3^{mk}W_9^{2k} \end{aligned}$$

Question 6

Determine the system function $H(z)$ and the difference equation for the system that uses the Goertzel algorithm to compute the DFT value $X(N-k)$.

Solution:-

$$\begin{aligned} X(k) &= \sum_{m=0}^{N-1} x(m)W_N^{km} \\ &= \sum_{m=0}^{N-1} x(m)W_N^{km}W_N^{-kN} \text{ since } W_N^{-kN} = 1 \\ &= \sum_{m=0}^{N-1} x(m)W_N^{-k(N-m)} \end{aligned}$$

This can be viewed as the convolution of the N-length sequence x(n) with implulse response of a linear filter

$$\begin{aligned} h_k(n) &\stackrel{\Delta}{=} W_N^{kn}u(n), \text{ evaluated at time N} \\ H_k(z) &= \sum_{n=0}^{\infty} W_N^{kn}z^{-n} \end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{1 - W_N^k z^{-1}} \\
&= \frac{Y_u(z)}{X(z)} \\
y_k(n) &= W_N^k y_k(n-1) + x(n), \qquad y_k(-1) = 0 \\
y_k(N) &= X(k)
\end{aligned}
$$

Unit 3
Design of IIR Filters

5.1 Introduction

A digital filter is a linear shift-invariant discrete-time system that is realized using finite precision arithmetic. The design of digital filters involves three basic steps:

- The specification of the desired properties of the system.
- The approximation of these specifications using a causal discrete-time system.
- The realization of these specifications using finite precision arithmetic.

These three steps are independent; here we focus our attention on the second step. The desired digital filter is to be used to filter a digital signal that is derived from an analog signal by means of periodic sampling. The specifications for both analog and digital filters are often given in the frequency domain, as for example in the design of low pass, high pass, band pass and band elimination filters.

Given the sampling rate, it is straight forward to convert from frequency specifications on an analog filter to frequency specifications on the corresponding digital filter, the analog frequencies being in terms of Hertz and digital frequencies being in terms of radian frequency or angle around the unit circle with the point Z=-1 corresponding to half the sampling frequency. The least confusing point of view toward digital filter design is to consider the filter as being specified in terms of angle around the unit circle rather than in terms of analog frequencies.

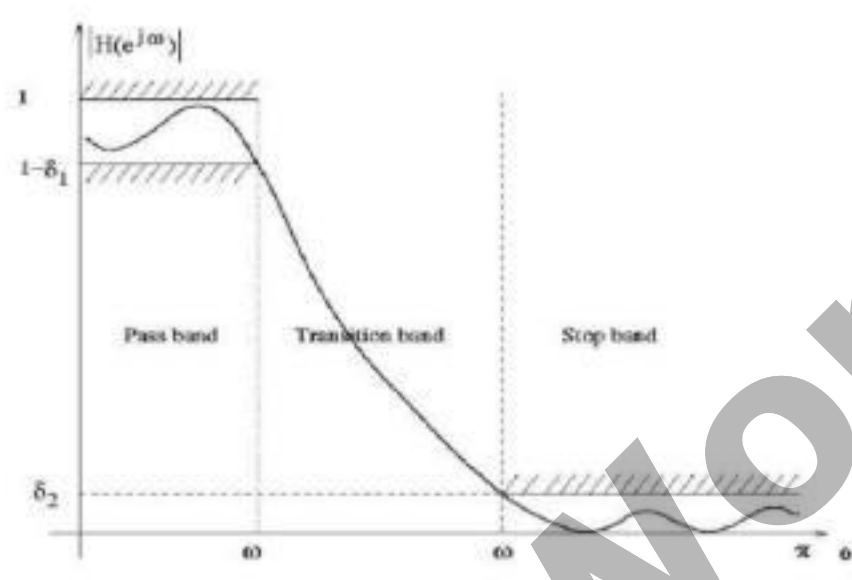


Figure 5.1: Tolerance limits for approximation of ideal low-pass filter

A separate problem is that of determining an appropriate set of specifications on the digital filter. In the case of a low pass filter, for example, the specifications often take the form of a tolerance scheme, as shown in Fig. 5.1.

$$1-\delta_1 \le | H(e^{j\omega}) | \le 1, \quad | \omega | \le \omega_p$$

$$| H(e^{j\omega}) | \le \delta_2, \quad \omega_s \le | \omega | \le \pi$$

Many of the filters used in practice are specified by such a tolerance scheme, with no constraints on the phase response other than those imposed by stability and causality requirements; i.e., the poles of the system function must lie inside the unit circle. Given a set of specifications in the form of Fig. 5.1, the next step is to and a discrete time linear system whose frequency response falls within the prescribed tolerances. At this point the filter design problem becomes a problem in approximation. In the case of infinite impulse response (IIR) filters, we must approximate the desired frequency response by a rational function, while in the finite impulse response (FIR) filters case we are concerned with polynomial approximation.

5.1 Design of IIR Filters from Analog Filters:

The traditional approach to the design of IIR digital filters involves the transformation of an analog filter into a digital filter meeting prescribed specifications. This is a reasonable approach because:

- The art of analog filter design is highly advanced and since useful results can be achieved, it is advantageous to utilize the design procedures already developed for analog filters.
- Many useful analog design methods have relatively simple closed-form design formulas.

Therefore, digital filter design methods based on analog design formulas are rather simple to implement. An analog system can be described by the differential equation

$$\sum_{k=0}^{N} c_k \frac{d^k y_a(t)}{dt^k} = \sum_{k=0}^{M} d_k \frac{d^k x_a(t)}{dt^k}$$

And the corresponding rational function is

$$H_a(s) = \frac{\sum_{k=0}^{M} d_k s^k}{\sum_{k=0}^{N} c_k s^k} = \frac{y_a(s)}{x_a(s)}$$

The corresponding description for digital filters has the form

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

and the rational function

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

In transforming an analog filter to a digital filter we must therefore obtain either H(z) or h(n) (inverse Z-transform of H(z) i.e., impulse response) from the analog filter design. In such transformations, we want the imaginary axis of the S-plane to map into the nit circle of the Z-plane, a stable analog filter should be transformed to a stable digital filter. That is, if the analog filter has poles only in the left-half of S-plane, then the digital filter must have poles only inside the unit circle. These constraints are basic to all the techniques discussed here.

5.2 Characteristics of Commonly Used Analog Filters:

From the previous discussion it is clear that, IIT digital filters can be obtained by beginning with an analog filter. Thus the design of a digital filter is reduced to designing an appropriate analog filter and then performing the conversion from Ha(s) to H (z). Analog filter design is a well - developed field, many approximation techniques, viz., Butterworth, Chebyshev, Elliptic, etc., have been developed for the design of analog low

pass filters. Our discussion is limited to low pass filters, since, frequency transformation can be applied to transform a designed low pass filter into a desired high pass, band pass and band stop filters.

5.2.1 Butterworth Filters:

Low pass Butterworth filters are all - pole filters with monotonic frequency response in both pass band and stop band, characterized by the magnitude - squared frequency response

$$|H_a(\Omega)|^2 = \frac{1}{1+(\Omega/\Omega_c)^{2N}} = \frac{1}{1+\epsilon^2(\Omega/\Omega_p)^{2N}}$$

Where, N is the order of the filter, Ώc is the -3dB frequency, i.e., cutoff frequency, Ώp is the pass band edge frequency and 1= (1 /1+ε^2) is the band edge value of │Ha(Ώ)│^{2}. Since the product Ha(s) Ha(-s) and evaluated at s = jΏ is simply equal to │Ha(Ώ)│^{2}, it follows that

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

The poles of Ha(s)Ha(-s) occur on a circle of radius Ώc at equally spaced points. From Eq. (5.29), we find the pole positions as the solution of

$$\frac{-s^2}{\Omega_c^2} = (-1)^{1/N} = e^{j(2k+1)\pi/N},\ \ k = 0, 1, , \cdots, N-1$$

And hence, the N poles in the left half of the s-plane are

$$\begin{aligned} s_k &= \Omega_c e^{j\pi/2} e^{(2k+1)\pi/2N},\ \ k = 0, 1, \cdots, N-1 \\ &= \sigma_k + j\Omega_k \end{aligned}$$

Note that, there are no poles on the imaginary axis of s-plane, and for N odd there will be a pole on real axis of s-plane, for N even there are no poles even on real axis of s-plane. Also note that all the poles are having conjugate symmetry. Thus the design methodology to design a Butterworth low pass filter with δ2 attenuation at a specified frequency Ώs is Find N,

$$N = \frac{\log[(1/\delta_2^2) - 1]}{2\log(\Omega_s/\Omega_c)} = \frac{\log(\delta/\epsilon)}{\log(\Omega_s/\Omega_p)}$$

Where by definition, δ2 = 1/√1+δ^2. Thus the Butterworth filter is completely characterized by the parameters N, δ2, ε and the ratio Ώs/Ωp or Ώc.Then, from Eq. (5.31) find the pole positions Sk; k = 0,1, 2,……..(N-1). Finally the analog filter is given by

$$H_a(s) = \prod_{k==1}^{N} \frac{1}{(s - s_k)}$$

5.2.2 Chebyshev Filters:

There are two types of Chebyshev filters. Type I Chebyshev filters are all-pole filters that exhibit equiripple behavior in the pass band and a monotonic characteristic in the stop band. On the other hand, type II Chebyshev filters contain both poles and zeros and exhibit a monotonic behavior in the pass band and an equiripple behavior in the stop band. The zeros of this class of filters lie on the imaginary axis in the s-plane. The magnitude squared of the frequency response characteristic of type I Chebyshev filter is given as

$$| H_a(\Omega) |^2 = \frac{1}{1 + \epsilon^2 T_N^2(\Omega/\Omega_p)}$$

Where ε is a parameter of the filter related to the ripple in the pass band as shown in Fig. (5.7), and T_N is the Nth order Chebyshev polynomial defined as

$$T_N(x) = \begin{cases} \cos(N\cos^{-1} x), & | x | \leq 1 \\ \cosh(N\cosh^{-1} x), & | x | > 1 \end{cases}$$

The Chebyshev polynomials can be generated by the recursive equation

$$T_{N+1}(x) == 2xT_N(x) - T_{N-1}(x), \quad N = 1, 2, \cdots$$

Where $T_0(x) = 1$ and $T_1(x) = x$.

At the band edge frequency $\Omega = \Omega_p$, we have

$$\frac{1}{\sqrt{1+\epsilon^2}} = 1 - \delta_1$$

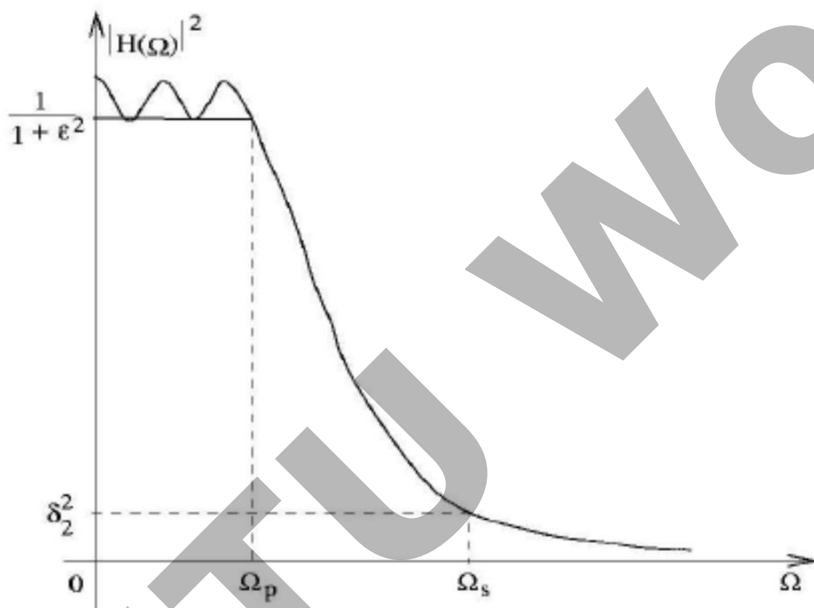


Figure 5.2: Type I Chebysehev filter characteristic

Or equivalently

$$\epsilon^2 = \frac{1}{(1-\delta_1)^2} - 1$$

Where δ_1 is the value of the pass band ripple.

The poles of Type I Chebyshev filter lie on an ellipse in the s-plane with major axis

$$r_1 = \Omega_p \frac{\beta^2 + 1}{2\beta}$$

And minor axis

$$r_2 = \Omega_p \frac{\beta^2 - 1}{2\beta}$$

Where β is related to ε according to the equation

$$\beta = [\frac{\sqrt{1+\epsilon^2}+1}{\epsilon}]^{1/N}$$

The angular positions of the left half s-plane poles are given by

$$\phi_k = \frac{\pi}{2} + \frac{(2k+1)\pi}{2N}, \ \ k = 0, 1, \cdots, N-1$$

Then the positions of the left half s-plane poles are given by

$$s_k = \sigma_k + j\Omega_k, \ \ k = 0, 1, \cdots, N-1$$

Where $\sigma_k = r_2 \text{ Cos } \varphi k$ and $\Omega_k = r_1 \text{ Sin}\varphi k$. The order of the filter is obtained from

$$\begin{aligned} N &= \frac{\log[(\sqrt{1-\delta_2^2}+\sqrt{1-\delta_2^2(1+\epsilon^2)})/\epsilon\delta_2]}{\log[\frac{\Omega_s}{\Omega_p}+\sqrt{(\frac{\Omega_s}{\Omega_p})^2-1}]} \\ &= \frac{\cosh^{-1}(\frac{\delta}{\epsilon})}{\cosh^{-1}(\frac{\Omega_s}{\Omega_p})} \end{aligned}$$

Where, by definition $\delta_2 = 1/\sqrt{1+\delta^2}$.

Finally, the Type I Chebyshev filter is given by

$$H_a(s) = \prod_{k=1}^{N} \frac{1}{(s-s_k)}$$

A Type II Chebyshev filter contains zero as well as poles. The magnitude squared response is given as

$$|\ H_a(-\Omega)\ |^2 = \frac{1}{1+\epsilon^2[T_N^2(\frac{\Omega_s}{\Omega_p})/T_N^2(\frac{\Omega_s}{\Omega})]}$$

Where $T_N(x)$ is the N-order Chebyshev polynomial. The zeros are located on the imaginary axis at the points

$$z_k = j\frac{\Omega_s}{\sin\phi_k}, \ k = 0, 1, \ldots, N-1$$

and the left-half s-plane poles are given

$$s_k = \sigma_k + j\Omega_k, \ k = 0, 1, \ldots, N-1$$

Where

$$\sigma_k = \frac{\Omega_s r_2 \cos\phi_k}{\sqrt{r_2^2 \cos^2\phi_k + r_1^2 \sin^2\phi_k}}$$

and

$$\Omega_k = \frac{\Omega_s r_1 \sin\phi_k}{\sqrt{r_2^2 \cos^2\phi_k + r_1^2 \sin^2\phi_k}}$$

Finally, the Type II Chebyshev filter is given by

$$H_a(s) = \prod_{k=1}^{N} \frac{s - z_k}{s - s_k}$$

The other approximation techniques are elliptic (equiripple in both passband and stopband) and Bessel (monotonic in both passband and stopband).

5.3 Analog to Analog Frequency Transforms

Frequency transforms are used to transform lowpass prototype filter to other filters like highpass or bandpass or bandstop filters. One possibility is to perform frequency transform in the analog domain and then convert the analog filter into a corresponding digital filter by a mapping of the s-plane into z-plane. An alternative approach is to convert the analog lowpass filter into a lowpass digital filter and then to transform the lowpass digital filter into the desired digital filter by a digital transformation.

Suppose we have a lowpass filter with pass edge Ω_P and if we want convert that into another lowpass filter with pass band edge Ω'_P then the transformation used is

$$s \longrightarrow \frac{\Omega_p}{\Omega'_p} s \qquad \text{(lowpass to lowpass)}$$

Thus we obtain a lowpass filter with system function $H_l(s) = H_p[(\Omega_p/\Omega'_p)s]$, where $H_p(s)$ is the system function of the prototype filter with passband edge frequency Ω_p.

To convert low pass filter into highpass filter the transformation used is

$$s \longrightarrow \frac{\Omega_p \Omega'_p}{s} \qquad \text{(lowpass to highpass)}$$

s

The system function of the highpass filter is $H_h(s) = H_p(\Omega_p \Omega'_p / s)$.

The transformation for converting a lowpass analog filter with passband edge frequency Ω_p into a band filter, having a lower band edge frequency Ω_l and an upper band edge frequency Ω_u, can be accomplished by first converting the lowpass

$$s \longrightarrow \frac{s^2 + \Omega_l \Omega_u}{s(\Omega_u - \Omega_l)} \qquad \text{(lowpass to bandpass)}$$

Thus we obtain

$$H_b(s) = H_p\left(\Omega_p \frac{s^2 + \Omega_l \Omega_u}{s(\Omega_u - \Omega_l)}\right)$$

Finally, if we wish to convert a lowpass analog filter with band edge frequency Ω_p into a bandstop filter, the transformation is simply the inverse of (8.4.3) with the additional factor Ω_p serving to normalize for the band edge frequency of the lowpass filter. Thus the transformation is

$$s \longrightarrow \Omega_p \frac{s(\Omega_u - \Omega_l)}{s^2 + \Omega_u \Omega_l} \qquad \text{(lowpass to bandstop)}$$

The filter function is

$$H_{bs}(s) = H_p\left(\Omega_p \frac{s(\Omega_u - \Omega_l)}{s^2 + \Omega_u \Omega_l}\right)$$

Recommended Questions with answers

Question 1

- I Design a digital filter to satisfy the following characteristics.
  -3dB cutoff frequency of 0:5_ rad.
- Magnitude down at least 15dB at 0:75_ rad.
- Monotonic stop band and pass band Using
- Impulse invariant technique
- Approximation of derivatives
- Bilinear transformation technique

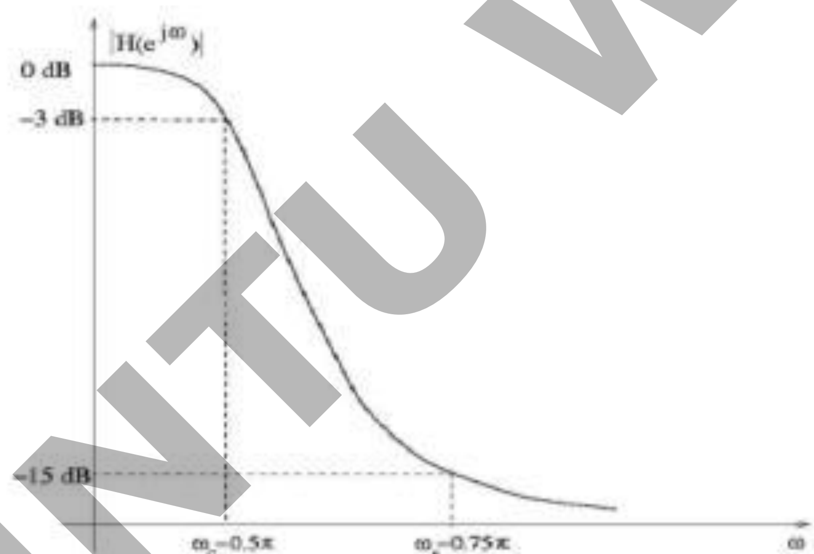


Figure 5.8: Frequency response plot of the example

Solution:-

a) Impulse Invariant Technique

From the given digital domain frequency, _nd the corresponding analog domain frequencies.

$$\Omega_c = \frac{\omega_c}{T} \text{ and } \Omega_s = \frac{\omega_s}{T}$$

Where T is the sampling period and 1/T is the sampling frequency and it always corresponds to 2Π radians in the digital domain. In this problem, let us assume T = 1sec.

Then Ώc = 0:5Π and Ώs = 0:75Π

Let us find the order of the desired filter using

$$N = \frac{(\frac{1}{\delta_2^2} - 1)}{2\log(\frac{\Omega_s}{\Omega_c})}$$

Where δ2 is the gain at the stop band edge frequency ωs.

$$-15 \text{ dB} = 20 \log \delta_2$$

$$\delta_2 = 20 \log \delta_2$$

$$\delta_2 = 10^{-\frac{15}{20}} = 0.1778$$

$$N = \frac{\log(\frac{1}{(0.1778)^2} - 1}{2\log(\frac{0.75\pi}{0.5\pi})} = 4.219 \simeq 5$$

Order of filter N =5.
Then the 5 poles on the Butterworth circle of radius Ώc = 0:5 Π are given by

$$\begin{aligned}
s_0 &= 0.5\pi e^{j(\frac{\pi}{2}+\frac{\pi}{10})} = -0.485 + j1.493 \\
s_1 &= 0.5\pi e^{j(\frac{\pi}{2}+\frac{3\pi}{10})} = -1.27 + j0.923 \\
s_2 &= 0.5\pi e^{j(\frac{\pi}{2}+\frac{5\pi}{10})} = -1.57 + j0.0 \\
s_3 &= 0.5\pi e^{j(\frac{\pi}{2}+\frac{7\pi}{10})} = -1.27 - j0.923 \\
s_5 &= 0.5\pi e^{j(\frac{\pi}{2}+\frac{9\pi}{10})} = -0.485 - j1.493
\end{aligned}$$

Then the filter transfer function in the analog domain is

$$\begin{aligned}
H_a(s) &= \frac{1}{(s+0.485-j1.493)(s+1.27-j0.923)(s+1.57)(s+1.27+j0.923)(s+0.485+j0.923)} \\
&= \sum_{k=1}^{5} \frac{A_k}{(s-s_k)}
\end{aligned}$$

where Ak's are partial fractions coefficients of Ha(s).

Finally, the transfer function of the digital filter is

$$H(z) = \sum_{k=1}^{5} \frac{A_k}{(1-e^{s_k}z^{-1})}, \text{ where } s_k\text{'s are the poles on the Butterworth circle}$$

b)

$$H(z) = H_a(s)\,|_{s=\frac{1-z^{-1}}{T}=1-z^{-1}}$$

$$H(z) = \sum_{k=1}^{5} \frac{1}{(1 - z^{-1} - s_k)}$$

c) For the bilinear transformation technique, we need to pre-warp the digital frequencies into corresponding analog frequencies.

i.e., $\Omega = \frac{2}{T}\tan(\frac{\omega}{2})$

$$\Omega_c = 2\tan(\frac{0.5\pi}{2}) = 2 \text{ rad.}$$

and

$$\Omega_s = 2\tan(\frac{0.75\pi}{2}) = 4.828 \text{ rad.}$$

Then the order of the filter

$$N = \frac{\log(\frac{1}{(0.1778)^2} - 1)}{2\log(\frac{4.828}{2})}$$

The pole locations on the Butterworth circle with radius Ώc = 2 are

$$s_0 = 2e^{j(\frac{\pi}{2}+\frac{\pi}{4})} = -1.414 + j1.414$$

$$s_1 = 2e^{j(\frac{\pi}{2}+\frac{3\pi}{4})} = -1.414 - j1.414$$

Then the filter transfer function in the analog domain is

$$H_a(s) = \frac{1}{(s + 1.414 - j1.414)(s + 1.414 + j1.414)}$$

Finally, the transfer function of the digital filter is

$$H(z) = H_a(s)\,|_{s=\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}=2\frac{1-z^{-1}}{1+z^{-1}}}$$

$$H(z) = \frac{1}{(2\frac{1-z^{-1}}{1+z^{-1}} + 1.414 - j1.414)(2\frac{1-z^{-1}}{1+z^{-1}} + 1.414 + j1.414)}$$

Question 2

Design a digital filter using impulse invariant technique to satisfy following characteristics

(i) Equiripple in pass band and monotonic in stop band

(ii) -3dB ripple with pass band edge frequency at 0:5Π radians.

(iii) Magnitude down at least 15dB at 0:75 Π radians.

Solution: Assuming T=1, Ώ= 0:5 Π and s = 0:75 Π

The order of desired filter is

$$N = \frac{\log[(\sqrt{1-\delta_2^2}) + \sqrt{1-\delta_2^2(1-\epsilon^2)})/\epsilon\delta_2]}{\log[\frac{\Omega_s}{\Omega_p} + \sqrt{(\frac{\Omega_s}{\Omega_p})^2 - 1}]}$$

when

$$20\log\frac{1}{\sqrt{1+\epsilon^2}} = -3|mboxdB$$

i.e.,

$$10\log(1+\epsilon^2) = 3dB$$

$$\epsilon^2 = 10^{0.3} - 1 = 0.9952$$

$$\epsilon = 0.9976$$

and

$$20\log\delta_2 = -15 \text{ dB}$$

$$\delta_2 = 10^{-0.75} = 0.1778$$

Hence

$$\begin{aligned} N &= \frac{[(\sqrt{1-(0.1778)^2} + \sqrt{1-(0.1778)^2(1+0.9952)})/0.9976 \times 0.1778]}{\log[\frac{0.75\pi}{0.5\pi} + \sqrt{(\frac{0.75\pi}{0.5\pi})^2 - 1}]} \\ &= 2.48 \\ &\simeq 3 \end{aligned}$$

The order of filter, N = 3.

The 3 poles on the ellipse are determined by

$$\beta = [\frac{\sqrt{1+\epsilon^2}+1}{\epsilon}]^{\frac{1}{N}} = [\frac{\sqrt{1+0.9976^2}+1}{0.9976}]^{\frac{1}{3}} = 1.342$$

$$\begin{aligned} r_1 &= \Omega_p \frac{\beta^2+1}{2\beta} \\ &= 0.5\pi \times \frac{(1.341)^2+1}{2 \times 1.341} \\ &= 1.639 \end{aligned}$$

$$\begin{aligned} r_2 &= \Omega_p \frac{\beta^2-1}{2\beta} \\ &= 0.5\pi \times \frac{(1.341)^2-1}{2 \times 1.341} \\ &= 0.469 \end{aligned}$$

The angles,

$$\phi_k = \frac{\pi}{2} + \frac{(2k+1)\pi}{2N}, \text{ k = 0,1 ,2}$$

The poles are at

$$s_k = r_2 \cos\phi_k + j r_1 \sin\phi_k$$

$$s_0 = 0.469\cos(\frac{4\pi}{6}) + j1.639\sin(\frac{4\pi}{6})$$
$$= -0.2345 + j1.419$$

$$s_1 = 0.469\cos(\pi) + j1.639\sin(\pi)$$
$$= -0.469 + j0.0$$

$$s_2 = 0.469\cos(\frac{8\pi}{6}) + j1.6939\sin(\frac{8\pi}{6})$$
$$= -0.2345 - j1.419$$

The analog filter transfer function is given by

$$H_a(s) = \frac{1}{s + 0.2345 - j1.419)(s + 0.469) + (s + 0.2345 + j1.419)}$$
$$= \sum_{k=1}^{3} \frac{A_k}{(s - s_k)}$$

where A_k's are the partial fraction coefficients.

Finally, the digital filter transfer function is given by

$$H(z) = \sum_{k=1}^{3} \frac{A_k}{(1 - e^{s_k} z^{-1})}$$

Question 3

An IIR digital low-pass filter is required to meet the following specifications:

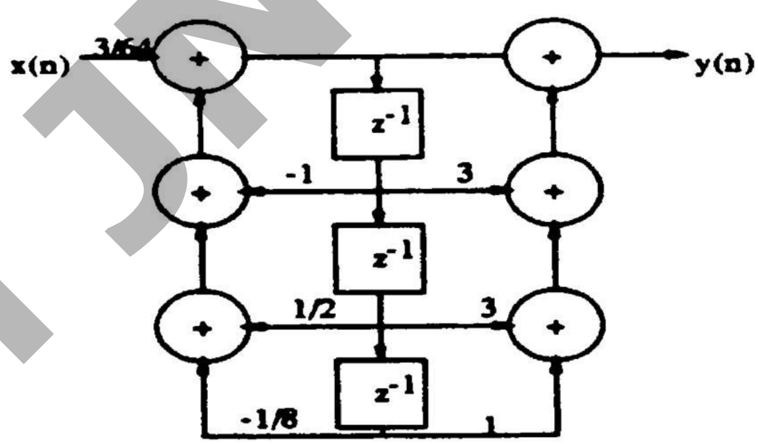
Passband ripple (or peak-to-peak ripple): ≤ 0.5 dB
Passband edge: 1.2 kHz
Stopband attenuation: ≥ 40 dB
Stopband edge: 2.0 kHz
Sample rate: 8.0 kHz

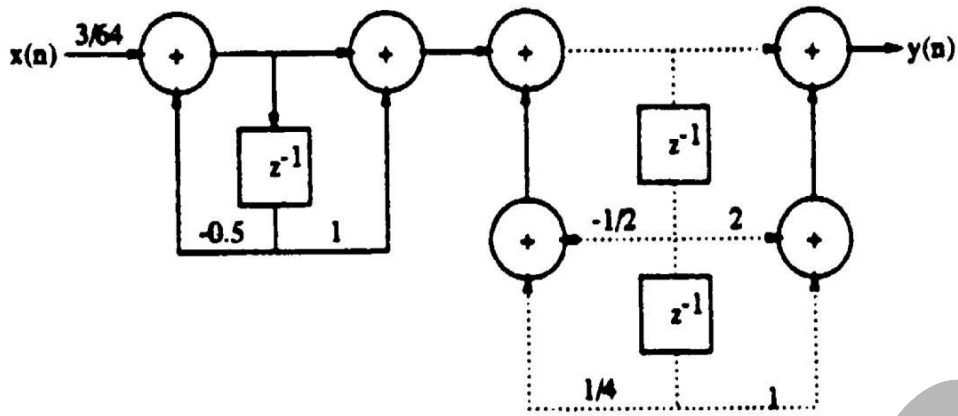
Use the design formulas in the book to determine the required filter order for

(a) A digital Butterworth filter
(b) A digital Chebyshev filter
(c) A digital elliptic filter

Solution:-
For the design specifications we have

$$\begin{aligned} \epsilon &= 0.349 \\ \delta &= 99.995 \\ f_p &= \frac{1.2}{8} = 0.15 \\ f_s &= \frac{2}{8} = 0.25 \\ \Omega_p &= 2tan\frac{w_p}{2} = 1.019 \\ \Omega_s &= 2tan\frac{w_s}{2} = 2 \\ \eta &= \frac{\delta}{\epsilon} = 286.5 \\ k &= \frac{\Omega_s}{\Omega_p} = 1.963 \end{aligned}$$





Butterworth filter: $N_{\min} \geq \frac{\log\eta}{\log k} = 8.393 \Rightarrow N = 9$

Chebyshev filter: $N_{\min} \geq \frac{\cosh^{-1}\eta}{\cosh^{-1}k} = 4.90 \Rightarrow N = 5$

Elliptic filter: $N_{\min} \geq \frac{k(\frac{1}{k})}{k(\sqrt{1-\frac{1}{k^2}})} \cdot \frac{k(\sqrt{1-\frac{1}{\eta^2}})}{k(\frac{1}{\eta})} \Rightarrow N = 4$

Question 4

Determine the system function $H(z)$ of the lowest-order Chebyshev digital filter that meets the following specifications:

(a) $\frac{1}{2}$-dB ripple in the passband $0 \leq |\omega| \leq 0.24\pi$.

(b) At least 50-dB attenuation in the stopband $0.35\pi \leq |\omega| \leq \pi$. Use the bilinear transformation.

Solution:-

Passband ripple = 0.5dB $\Rightarrow \epsilon = 0.349$
Stopband attenuation = 50dB

$$\begin{aligned} w_p &= 0.24\pi \\ w_s &= 0.35\pi \\ \Omega_p &= 2\tan\frac{w_p}{2} = 0.792 \\ \Omega_s &= 2\tan\frac{w_s}{2} = 1.226 \\ \eta &= \frac{\delta}{\epsilon} = 906.1 \\ k &= \frac{\Omega_s}{\Omega_p} = 1.547 \\ N_{\min} &\geq \frac{\cosh^{-1}\eta}{\cosh^{-1}k} = \frac{7.502}{1.003} = 7.48 \Rightarrow N = 8 \end{aligned}$$

Implementation of Discrete-Time Systems

6.1 Introduction

The two important forms of expressing system leading to different realizations of FIR & IIR filters are

a) Difference equation form

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

b) Ration of polynomials

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

The following factors influence choice of a specific realization,

- Computational complexity
- Memory requirements
- Finite-word-length
- Pipeline / parallel processing

6.1.1 Computation Complexity

This is do with number of arithmetic operations i.e. multiplication, addition & divisions. If the realization can have less of these then it will be less complex computationally.

In the recent processors the fetch time from memory & number of times a comparison between two numbers is performed per output sample is also considered and found to be important from the point of view of computational complexity.

6.1.2 Memory requirements

This is basically number of memory locations required to store the system parameters, past inputs, past outputs, and any intermediate computed values. Any realization requiring less of these is preferred.

6.1.3 Finite-word-length effects

These effects refer to the quantization effects that are inherent in any digital implementation of the system, either in hardware or in software. No computing system has infinite precision. With finite precision there is bound to be errors. These effects are basically to do with truncation & rounding-off of samples. The extent of this effect varies with type of arithmetic used(fixed or floating). The serious issue is that the effects have influence on system characteristics. A structure which is less sensitive to this effect need to be chosen.

6.1.4 Pipeline / Parallel Processing

This is to do with suitability of the structure for pipelining & parallel processing. The parallel processing can be in software or hardware. Longer pipelining make the system more efficient.

6.2 Structure for FIR Systems:

FIR system is described by,

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

Or equivalently, the system function

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

Where we can identify $h(n) = \begin{cases} b_n & 0 \le n \le n-1 \\ 0 & otherwise \end{cases}$

Different FIR Structures used in practice are,

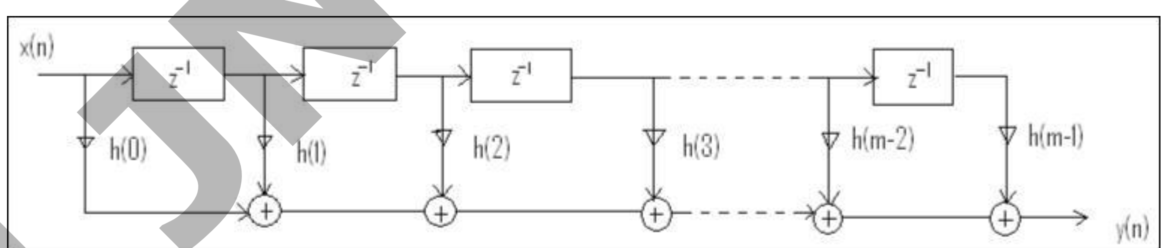
1. Direct form
2. Cascade form
3. Frequency-sampling realization
4. Lattice realization

6.2.1 Direct – Form Structure

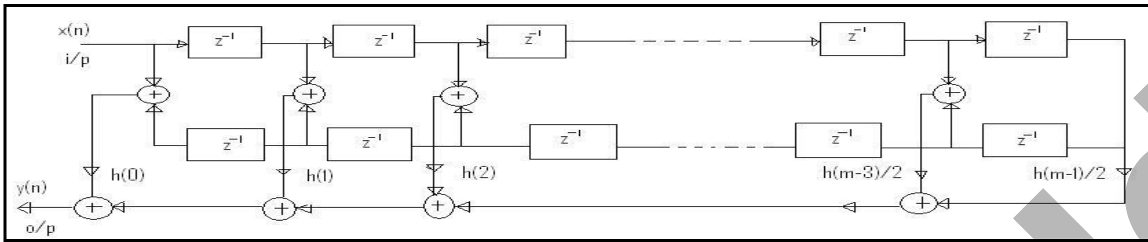
Convolution formula is used to express FIR system given by,

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

- It is Non recursive in structure



- As can be seen from the above implementation it requires M-1 memory locations for storing the M-1 previous inputs
- It requires computationally M multiplications and M-1 additions per output point
- It is more popularly referred to as tapped delay line or transversal system
- Efficient structure with linear phase characteristics are possible where $h(n) = \pm h(M-1-n)$



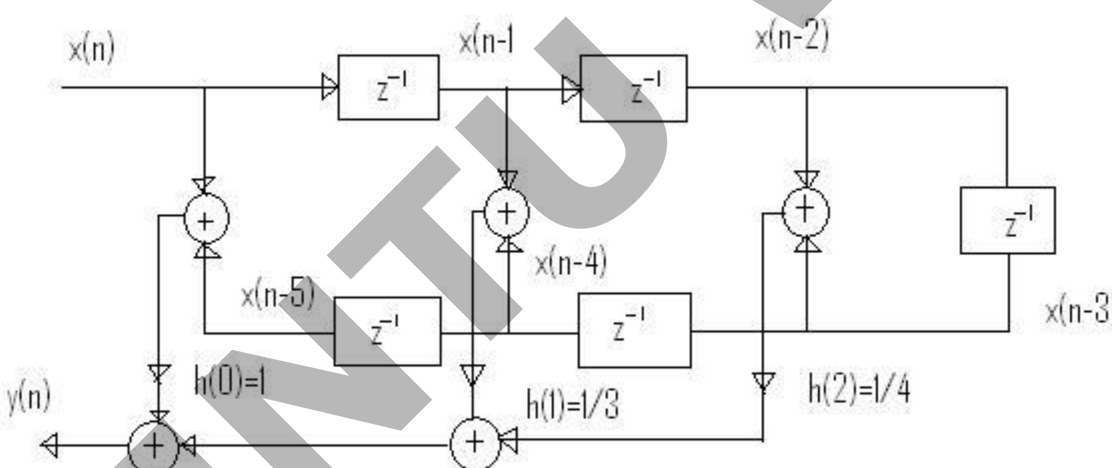
Prob:

Realize the following system function using minimum number of multiplication

(1) $H(Z) = 1 + \frac{1}{3}Z^{-1} + \frac{1}{4}Z^{-2} + \frac{1}{4}Z^{-3} + \frac{1}{3}Z^{-4} + Z^{-5}$

We recognize $h(n) = \left[1, \frac{1}{3}, \frac{1}{4}, \frac{1}{4}, \frac{1}{3}, 1\right]$

M is even = 6, and we observe h(n) = h(M-1-n) h(n) = h(5-n)
i.e h(0) = h(5) h(1) = h(4) h(2) = h(3) Direct form structure for Linear phase FIR can be realized



Exercise: Realize the following using system function using minimum number of multiplication.

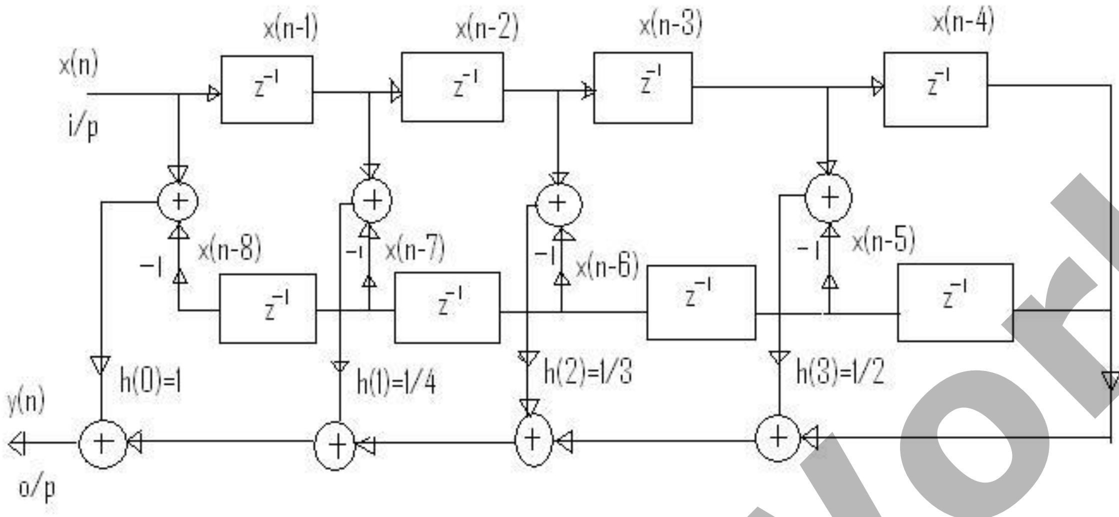
$H(Z) = 1 + \frac{1}{4}Z^{-1} + \frac{1}{3}Z^{-2} + \frac{1}{2}Z^{-3} - \frac{1}{2}Z^{-5} - \frac{1}{3}Z^{-6} - \frac{1}{4}Z^{-7} - Z^{-8}$

m=9 $h(n) = \left[1, \frac{1}{4}, \frac{1}{3}, \frac{1}{2}, -\frac{1}{2}, -\frac{1}{3}, -\frac{1}{4}, -1\right]$

odd symmetry

h(n) = -h(M-1-n); h(n) = -h(8-n); h(m-1/2) = h(4) = 0

h(0) = -h(8); h(1) = -h(7); h(2) = -h(6); h(3) = -h(5)



6.2.2 Cascade – Form Structure

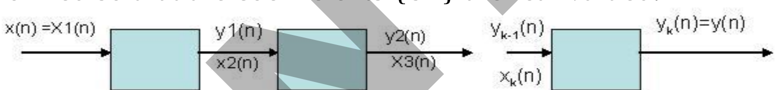
The system function H(Z) is factored into product of second – order FIR system

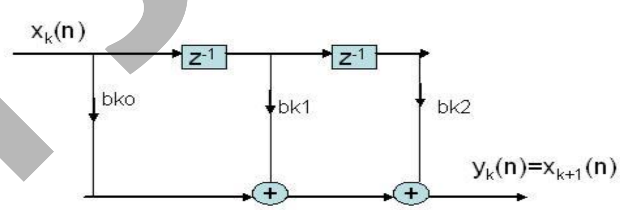
$$H(Z) = \prod_{k=1}^{K} H_k(Z)$$

Where $H_k(Z) = b_{k0} + b_{k1} Z^{-1} + b_{k2} Z^{-2}$ k = 1, 2, ….. K

and K = integer part of (M+1) / 2

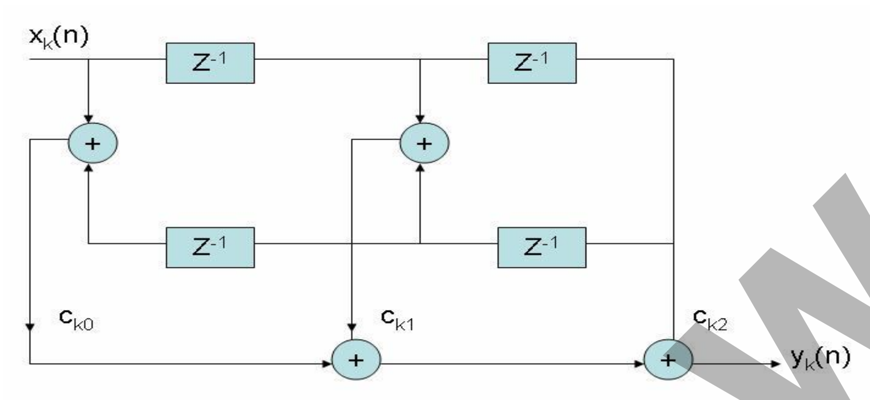
The filter parameter b_0 may be equally distributed among the K filter section, such that $b_0 = b_{10}\ b_{20}$ …. b_{k0} or it may be assigned to a single filter section. The zeros of H(z) are grouped in pairs to produce the second – order FIR system. Pairs of complex-conjugate roots are formed so that the coefficients $\{b_{ki}\}$ are real valued.





In case of linear –phase FIR filter, the symmetry in h(n) implies that the zeros of H(z) also exhibit a form of symmetry. If zk and zk* are pair of complex – conjugate zeros then 1/zk and 1/zk* are also a pair complex –conjugate zeros. Thus simplified fourth order sections are formed. This is shown below,

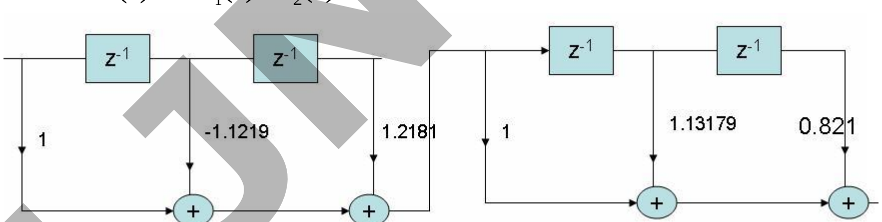
$$H_k(z) = C_{k0}(1 - z_k z^{-1})(1 - z_k{}^* z^{-1})(1 - z^{-1}/z_k)(1 - z^{-1}/z_{k^*})$$
$$= C_{k0} + C_{k1} z^{-1} + C_{k2} z^{-2} + C_{k1} z^{-3} + z^{-4}$$



Problem: Realize the difference equation
$y(n) = x(n) + 0.25x(n-1) + 0.5x(n-2) + 0.75x(n-3) + x(n-4)$
in cascade form.

$$Y(z) = X(z)\{1 + 0.25z^{-1} + 0.5z^{-2} + 0.75z^{-3} + z^{-4})$$

Soln:
$$H(z) = 1 + 0.25z^{-1} + 0.5z^{-2} + 0.75z^{-3} + z^{-4}$$
$$H(z) = (1 - 1.1219z^{-1} + 1.2181z^{-2})(1 + 1.3719z^{-1} + 0.821z^{-2})$$
$$H(z) = H_1(z)H_2(z)$$



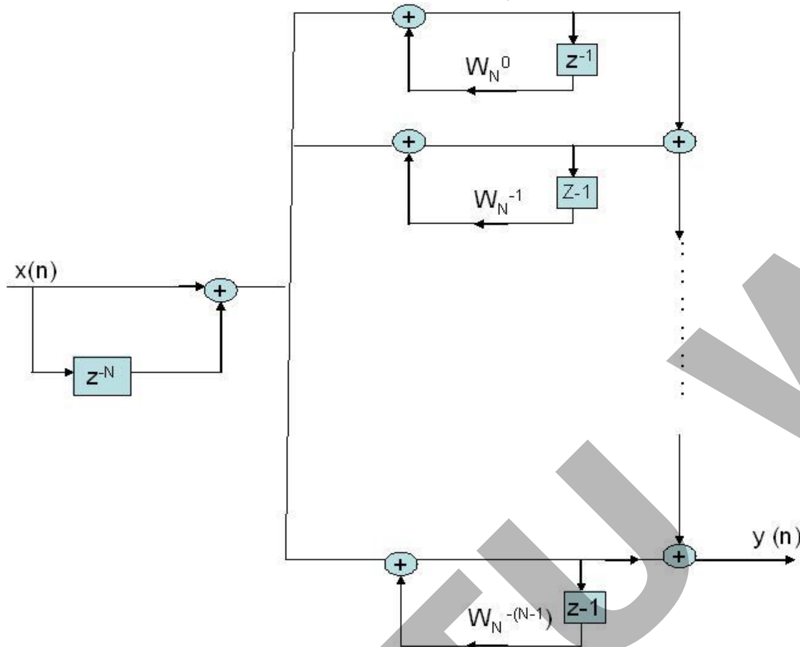
6.3 Frequency sampling realization:

We can express system function H(z) in terms of DFT samples H(k) which is given by

$$H(z) = (1 - z^{-N})\frac{1}{N}\sum_{k=0}^{N-1}\frac{H(k)}{1 - W_N^{-k} z^{-1}}$$

This form can be realized with cascade of FIR and IIR structures. The term $(1-z^{-N})$ is realized as FIR and the term $\frac{1}{N}\sum_{k=0}^{N-1}\frac{H(k)}{1-W_N^{-k}z^{-1}}$ as IIR structure.

The realization of the above freq sampling form shows necessity of complex arithmetic. Incorporating symmetry in h(n) and symmetry properties of DFT of real sequences the realization can be modified to have only real coefficients.



6.4 Lattice structures

Lattice structures offer many interesting features:

1. Upgrading filter orders is simple. Only additional stages need to be added instead of redesigning the whole filter and recalculating the filter coefficients.
2. These filters are computationally very efficient than other filter structures in a filter bank applications (eg. Wavelet Transform)
3. Lattice filters are less sensitive to finite word length effects.

Consider

$$H(z)=\frac{Y(z)}{X(z)}=1+\sum_{i=1}^{m}a_m(i)z^{-i}$$

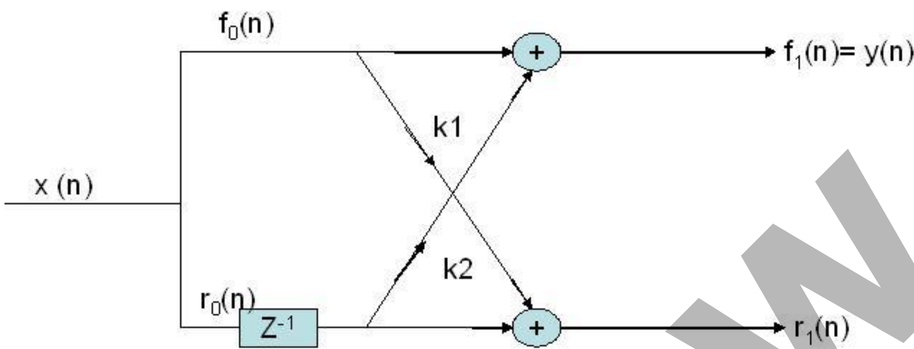
m is the order of the FIR filter and $a_m(0)=1$

when m = 1 $Y(z)/X(z) = 1+ a_1(1) z^{-1}$

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

$f1(n)$ is known as upper channel output and $r1(n)$as lower channel

output. $f_0(n)= r_0(n)=x(n)$



The outputs are

$$f_1(n) = f_0(n) + k_1 r_0(n-1) \qquad 1a$$
$$r_1(n) = k_1 f_0(n) + r_0(n-1) \qquad 1b$$
$$if \quad k_1 = a_1(1), then \quad f_1(n) = y(n)$$

If m=2

$$\frac{Y(z)}{X(z)} = 1 + a_2(1)z^{-1} + a_2(2)z^{-2}$$
$$y(n) = x(n) + a_2(1)x(n-1) + a_2(2)x(n-2)$$
$$y(n) = f_1(n) + k_2 r_1(n-1) \qquad (2)$$

Substituting 1a and 1b in (2)

$$y(n) = f_0(n) + k_1 r_0(n-1) + k_2 [k_1 f_0(n-1) + r_0(n-2)]$$
$$= f_0(n) + k_1 r_0(n-1) + k_2 k_1 f_0(n-1) + k_2 r_0(n-2)]$$
$$\sin ce \quad f_0(n) = r_0(n) = x(n)$$
$$y(n) = x(n) + k_1 x(n-1) + k_2 k_1 x(n-1) + k_2 x(n-2)]$$
$$= x(n) + (k_1 + k_1 k_2)x(n-1) + k_2 x(n-2)$$

We recognize

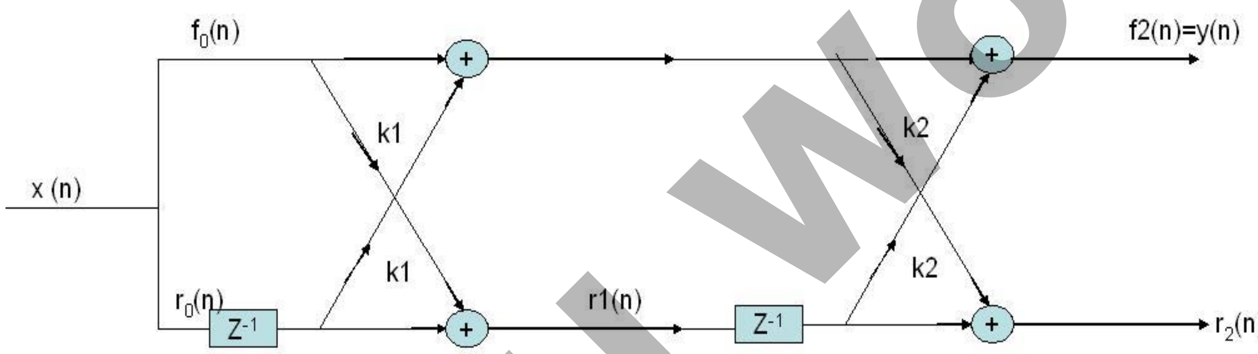
$a_2(1) = k_1 + k_1 k_2$

$a_2(1) = k_2$

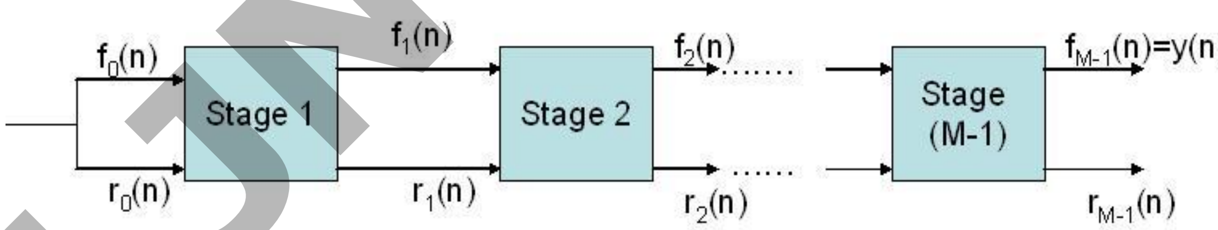
Solving the above equation we get

$$k_1 = \frac{a_2(1)}{1 + a_2(2)} \quad and \quad k_2 = a_2(2) \qquad (4)$$

Equation (3) means that, the lattice structure for a second-order filter is simply a cascade of two first-order filters with k1 and k2 as defined in eq (4)



Similar to above, an Mth order FIR filter can be implemented by lattice structures with M – stages



8.4.1 Direct Form –I to lattice structure

For m = M, M-1, ………..2, 1 do

$k_m = a_m(m)$

$$a_{m-1}(i) = \frac{a_m(i) - a_m(m) a_m(m-i)}{1 - k_m^2} \qquad 1 \le i \le m-1$$

- The above expression fails if k_m=1. This is an indication that there isa zero on the unit circle. If k_m=1, factor out this root from A(z) and the recursive formula can be applied for reduced order system.

$$\textit{for } m = 2 \textit{ and } m = 1$$
$$k_2 = a_2(2) \quad \& \quad k_1 = a_1(1)$$

$$\textit{for } m = 2 \ \& \ i = 1$$
$$a_1(1) = \frac{a_2(1) - a_2(2)a_2(1)}{1 - k_2^2} = \frac{a_2(1)[1 - a_2(2)]}{1 - a_2^2(2)} = \frac{a_2(1)}{1 + a_2(2)}$$
$$\textit{Thus} \quad k_1 = \frac{a_2(1)}{1 + a_2(2)}$$

8.4.2 Lattice to direct form –I

For m = 1,2,…….M-1

$$a_m(0) = 1$$
$$a_m(m) = k_m$$
$$a_m(i) = a_{m-1}(i) + a_m(m)a_{m-1}(m-i) \quad 1 \le i \le m-1$$

Problem:

Given FIR filter $H(Z) = 1 + 2Z^{-1} + \frac{1}{3}Z^{-2}$ obtain lattice structure for the same

Given $a_1(1) = 2$, $a_2(2) = 1/3$

Using the recursive equation for

m = M, M-1, ……, 2, 1

here M=2 therefore m = 2, 1

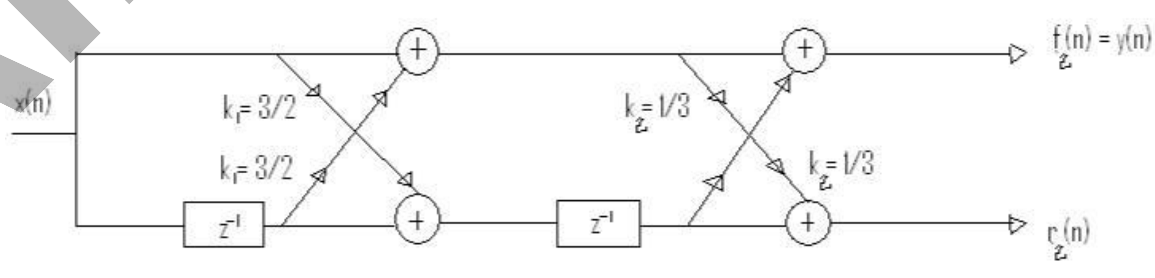
if m=2 $k_2 = a_2(2) = 1/3$

if m=1 $k_1 = a_1(1)$

also, when m=2 and i=1

$$a_1(1) = \frac{a_2(1)}{1 + a_2(2)} = \frac{2}{1 + 1/3} = \frac{3}{2}$$

Hence $k_1 = a_1(1) = 3/2$



Recommended questions with solution

Problem:1

Consider an FIR lattice filter with co-efficients $k_1 = \frac{1}{2}$, $k_2 = \frac{1}{3}$, $k_3 = \frac{1}{4}$. Determine the FIR filter co-efficient for the direct form structure ($H(Z) = a_3(0) + a_3(1)Z^{-1} + a_3(2)Z^{-2} + a_3(3)Z^{-3}$)

$a_3(0) = 1 \qquad a_3(3) = k_3 = 1/4$

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

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

for m=2, i=1

$$a_2(1) = a_1(1) + a_2(2)a_1(1)$$

$$= a_1(1)[1 + a_2(2)] = \frac{1}{2}\left[1 + \frac{1}{3}\right]$$

$$= \frac{4}{6} = \frac{2}{3}$$

for m=3, i=1

$$a_3(1) = a_2(1) + a_3(3)a_2(2)$$

$$= \frac{2}{3} + \frac{1}{4}.\frac{1}{3}$$

$$= \frac{2}{3} + \frac{1}{12} = \frac{8+1}{12}$$

$$= \frac{9}{12} = \frac{3}{4}$$

for m=3 & i=2

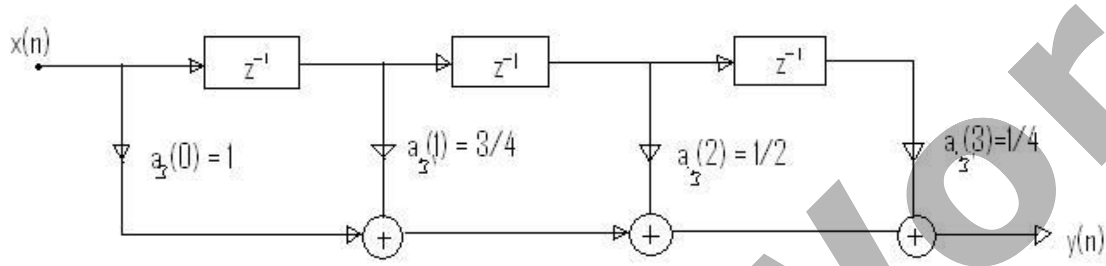
$$a_3(2) = a_2(2) + a_3(3)a_2(1)$$

$$= \frac{1}{3} + \frac{1}{4}.\frac{2}{3}$$

$$= \frac{1}{3} + \frac{1}{6} = \frac{2+1}{6}$$

$$= \frac{3}{6} = \frac{1}{2}$$

$a_3(0) = 1$, $a_3(1) = \frac{3}{4}$, $a_3(2) = \frac{1}{2}$, $a_3(3) = \frac{1}{4}$



6.5 Structures for IIR Filters

The IIR filters are represented by system function;

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

and corresponding difference equation given by,

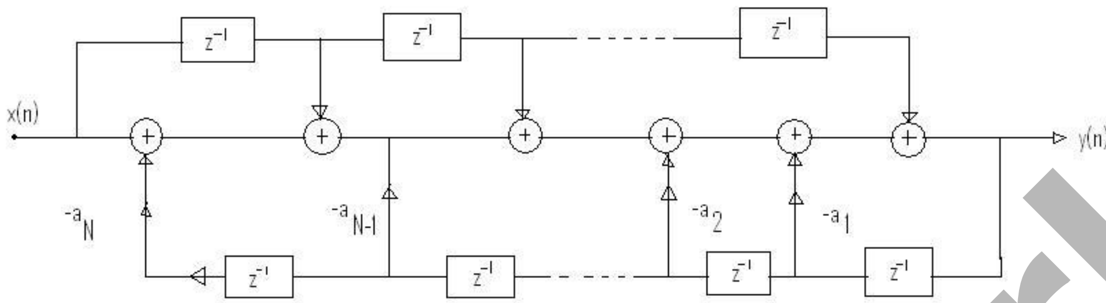
$$y(n) = -\sum_{k=1}^{N} a_k y(n-k) + \sum_{k=0}^{N} b_k x(n-k)$$

Different realizations for IIR filters are,

1. Direct form-I
2. Direct form-II
3. Cascade form
4. Parallel form
5. Lattice form

6.5.1 Direct form-I

This is a straight forward implementation of difference equation which is very simple. Typical Direct form – I realization is shown below . The upper branch is forward path and lower branch is feedback path. The number of delays depends on presence of most previous input and output samples in the difference equation.



6.5.2 Direct form-II

The given transfer function H(z) can be expressed as,

$$H(z) = \frac{Y(z)}{X(z)} = \frac{V(z)}{X(z)} \cdot \frac{Y(z)}{V(z)}$$

where V(z) is an intermediate term. We identify,

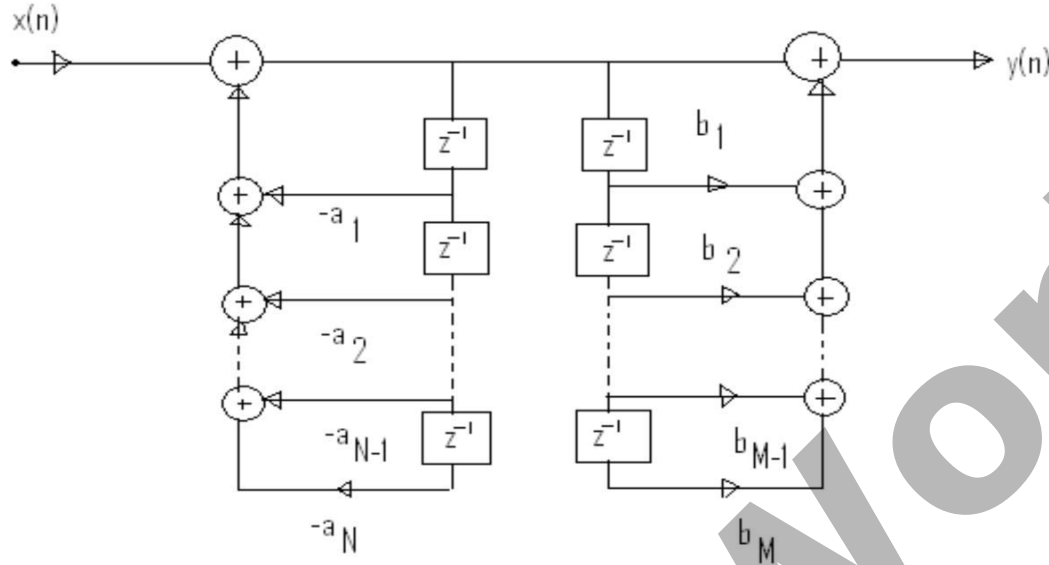
$$\frac{V(z)}{X(z)} = \frac{1}{1 + \sum_{k=1}^{N} a_k z^{-k}}$$ -------------------all poles

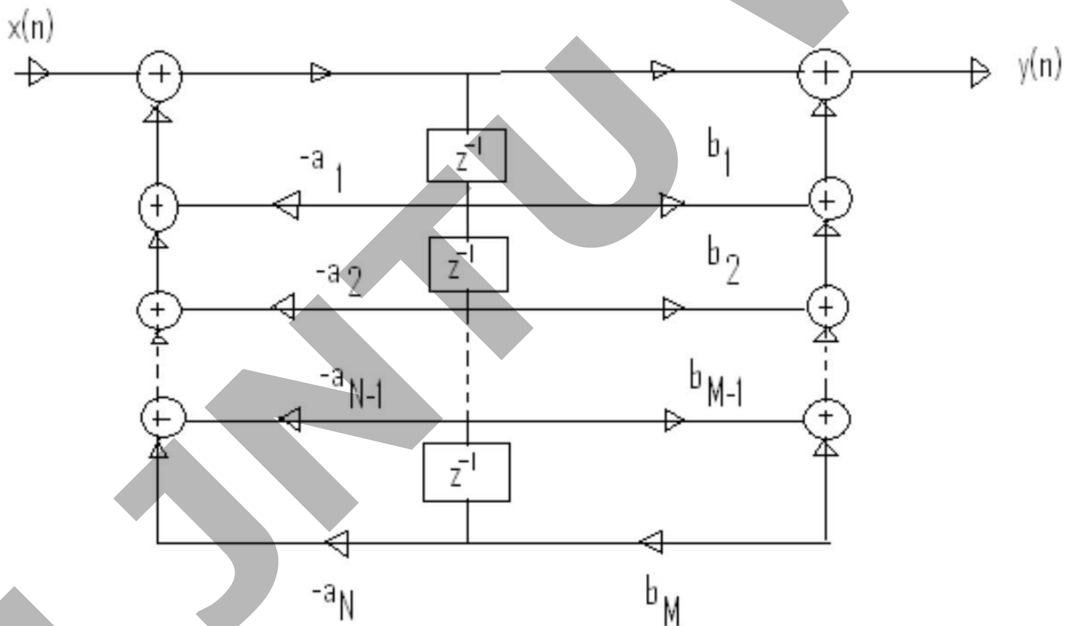
$$\frac{Y(z)}{V(z)} = \left(1 + \sum_{k=1}^{M} b_k z^{-k} \right)$$ -------------------all zeros

The corresponding difference equations are,

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

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





This realization requires M+N+! multiplications, M+N addition and the maximum of {M, N} memory location

6.5.3 Cascade Form

The transfer function of a system can be expressed as,

$$H(z) = H_1(z)\, H_2(z) \ldots . H_k(z)$$

Where $H_k(Z)$ could be first order or second order section realized in Direct form – II form i.e.,

$$H_k(Z) = \frac{b_{k0} + b_{k1} Z^{-1} + b_{k2} Z^{-2}}{1 + a_{k1} Z^{-1} + a_{k2} Z^{-2}}$$

where K is the integer part of (N+1)/2

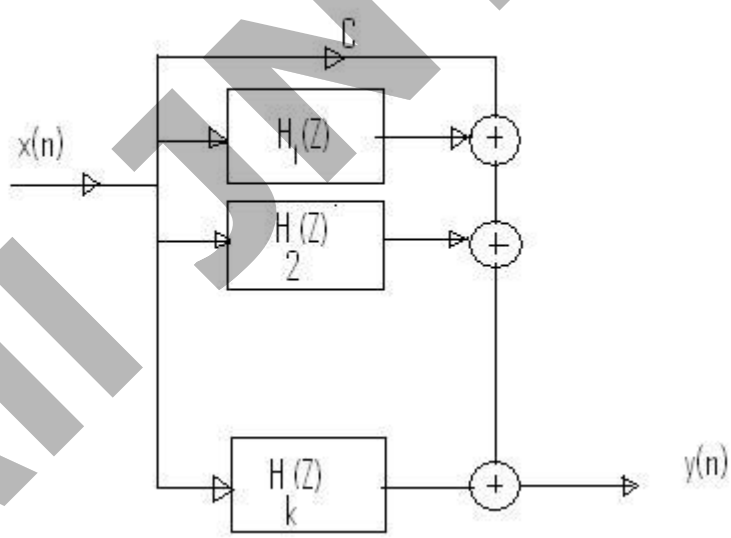
Similar to FIR cascade realization, the parameter b_0 can be distributed equally among the k filter section B_0 that $b_0 = b_{10} b_{20} \ldots . b_{k0}$. The second order sections are required to realize section which has complex-conjugate poles with real co-efficients. Pairing the two complex-conjugate poles with a pair of complex-conjugate zeros or real-valued zeros to form a subsystem of the type shown above is done arbitrarily. There is no specific rule used in the combination. Although all cascade realizations are equivalent for infinite precision arithmetic, the various realizations may differ significantly when implemented with finite precision arithmetic.

6.5.4 Parallel form structure

In the expression of transfer function, if $N \ge M$ we can express system function

$$H(Z) = C + \sum_{k=1}^{N} \frac{A_k}{1 - p_k Z^{-1}} = C + \sum_{k=1}^{N} H_k(Z)$$

Where $\{p_k\}$ are the poles, $\{A_k\}$ are the coefficients in the partial fraction expansion, and the constant C is defined as $C = b_N / a_N$, The system realization of above form is shown below.



Where $H_k(Z) = \dfrac{b_{k0} + b_{k1} Z^{-1}}{1 + a_{k1} Z^{-1} + a_{k2} Z^{-2}}$

Once again choice of using first- order or second-order sections depends on poles of the denominator polynomial. If there are complex set of poles which are conjugative in nature then a second order section is a must to have real coefficients.

Problem 2

Determine the
(i)Direct form-I (ii) Direct form-II (iii) Cascade &
(iv)Parallel form realization of the system function

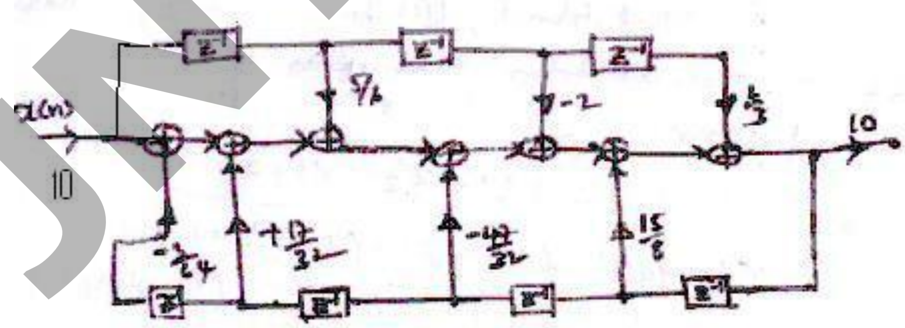
$$H(Z) = \frac{10\left(1-\frac{1}{2}Z^{-1}\right)\left(1-\frac{2}{3}Z^{-1}\right)\left(1+2Z^{-1}\right)}{\left(1-\frac{3}{4}Z^{-1}\right)\left(1-\frac{1}{8}Z^{-1}\right)\left(1-\left(\frac{1}{2}+j\frac{1}{2}\right)Z^{-1}\right)\left(1-\left(\frac{1}{2}-j\frac{1}{2}\right)Z^{-1}\right)}$$

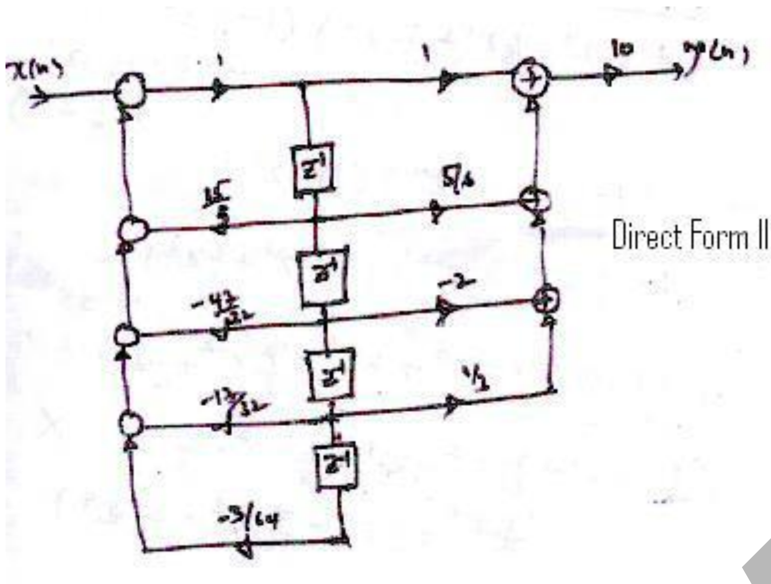
$$= \frac{10\left(1-\frac{7}{6}Z^{-1}+\frac{1}{3}Z^{-2}\right)\left(1+2Z^{-1}\right)}{\left(1+\frac{7}{8}Z^{-1}+\frac{3}{32}Z^{-2}\right)\left(1-Z^{-1}+\frac{1}{2}Z^{-2}\right)}$$

$$H(Z) = \frac{10\left(1+\frac{5}{6}Z^{-1}-2Z^{-2}+\frac{2}{3}Z^{-3}\right)}{\left(1-\frac{15}{8}Z^{-1}+\frac{47}{32}Z^{-2}-\frac{17}{32}Z^{-3}+\frac{3}{64}Z^{-4}\right)}$$

$$H(z) = \frac{(-14.75-12.90z^{-1})}{(1+\frac{7}{8}z^{-1}+\frac{3}{32}z^{-2})} + \frac{(24.50+26.82z^{-1})}{(1-z^{-1}+\frac{1}{2}z^{-2})}$$

Direct Form I





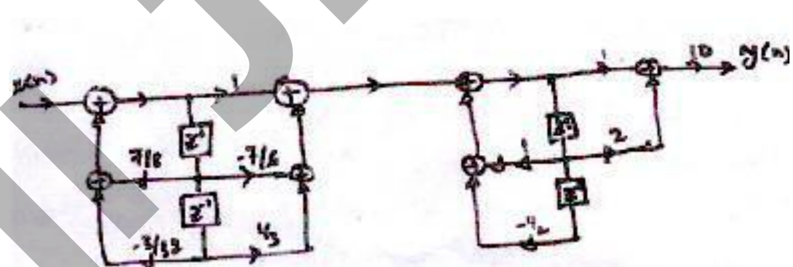
Cascade Form

$H(z) = H_1(z)\, H_2(z)$

Where

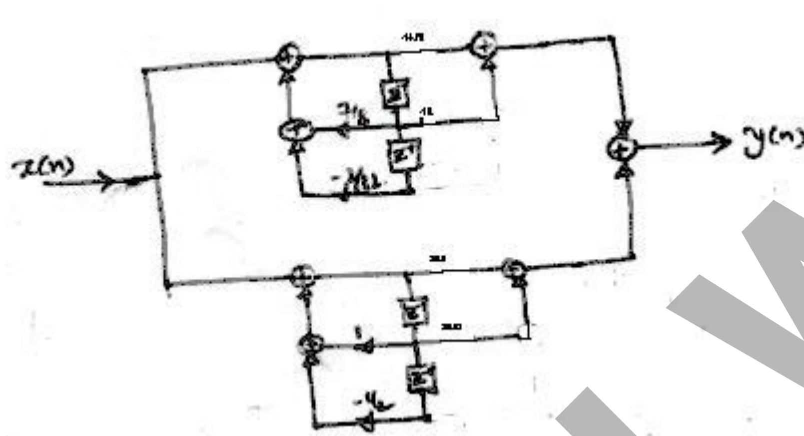
$$H_1(z) = \frac{1 - \frac{7}{6} z^{-1} + \frac{1}{3} z^{-2}}{1 - \frac{7}{8} z^{-1} + \frac{3}{32} z^{-2}}$$

$$H_1(z) = \frac{10(1 + 2z^{-1})}{1 - z^{-1} + \frac{1}{2} z^{-2}}$$



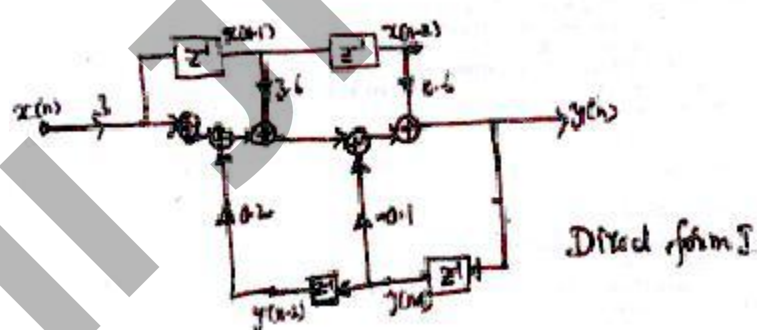
Parallel Form
$H(z) = H_1(z) + H_2(z)$

$$H(z) = \frac{(-14.75 - 12.90z^{-1})}{(1 + \frac{7}{8}z^{-1} + \frac{3}{32}z^{-2})} + \frac{(24.50 + 26.82z^{-1})}{(1 - z^{-1} + \frac{1}{2}z^{-2})}$$



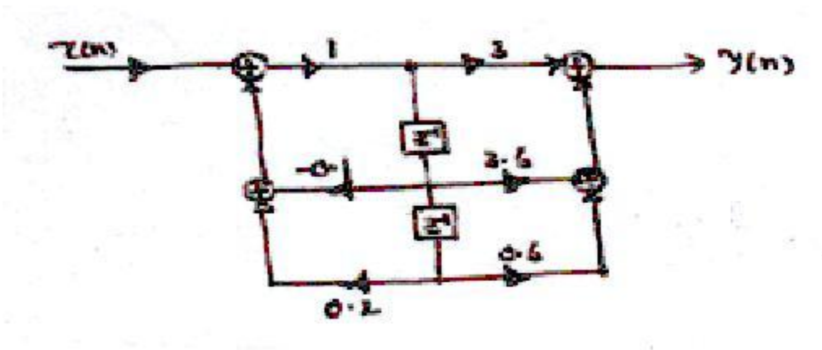
Problem: 3
Obtain the direct form – I, direct form-II
Cascade and parallel form realization for the following system,
y(n)= -0.1 y(n-1)+0.2y(n-2)+3x(n)+3.6 x(n-1)+0.6 x(n-2)

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)

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



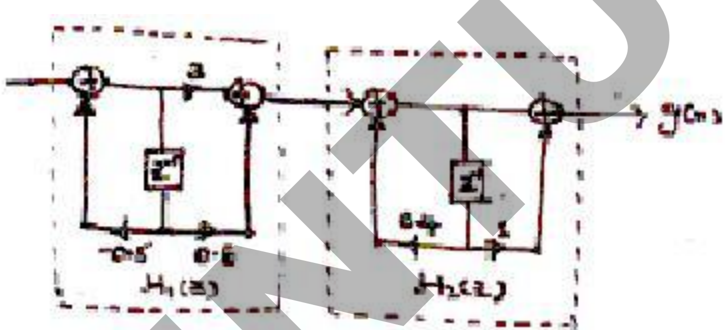
Cascade form realization

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

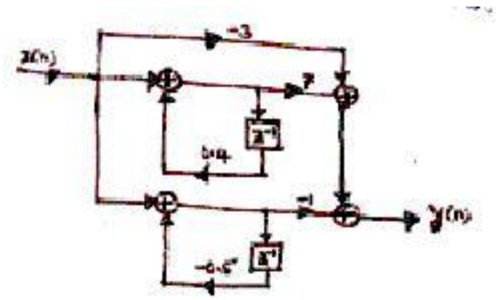


Parallel Form realization

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,

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



6.6 Lattice Structure for IIR System:

Consider an All-pole system with system function.

$$H(Z) = \frac{1}{1 + \sum_{k=1}^{N} a_N(k) Z^{-k}} = \frac{1}{A_N(Z)}$$

The corresponding difference equation for this IIR system is,

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

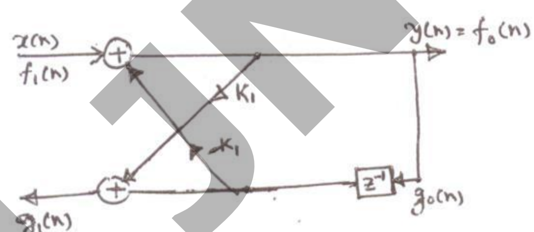
OR

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

For N=1

$$x(n) = y(n) + a_1(1)\, y(n-1)$$

Which can realized as,



We observe

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

$$y(n) = f_0(n) = f_1(n) - k_1 g_0(n-1)$$

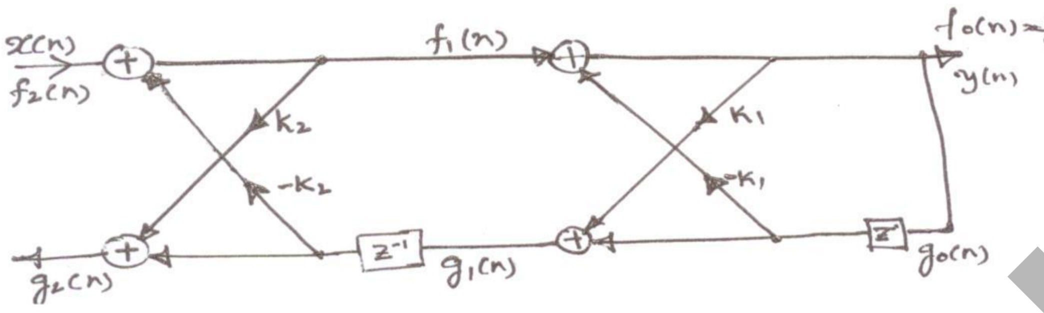
$$= x(n) - k_1 y(n-1) \qquad k_1 = a_1(1)$$

$$g_1(n) = k_1 f_0(n) + g_0(n-1) = k_1 y(n) + y(n-1)$$

For N=2, then

$$y(n) = x(n) - a_2(1)\, y(n-1) - a_2(2)\, y(n-2)$$

This output can be obtained from a two-stage lattice filter as shown in below fig



$$
\begin{aligned}
f_2(n) &= x(n) \\
f_1(n) &= f_2(n) - k_2 g_1(n-1) \\
g_2(n) &= k_2 f_1(n) + g_1(n-1) \\
f_0(n) &= f_1(n) - k_1 g_0(n-1) \\
g_1(n) &= k_1 f_0(n) + g_0(n-1)
\end{aligned}
$$

$$
\begin{aligned}
y(n) &= f_0(n) = g_0(n) = f_1(n) - k_1 g_0(n-1) \\
&= f_2(n) - k_2 g_1(n-1) - k_1 g_0(n-1) \\
&= f_2(n) - k_2\left[k_1 f_0(n-1) + g_0(n-2)\right] - k_1 g_0(n-1) \\
&= x(n) - k_2\left[k_1 y(n-1) + y(n-2)\right] - k_1 y(n-1) \\
&= x(n) - k_1(1 + k_2)\, y(n-1) - k_2\, y(n-2)
\end{aligned}
$$

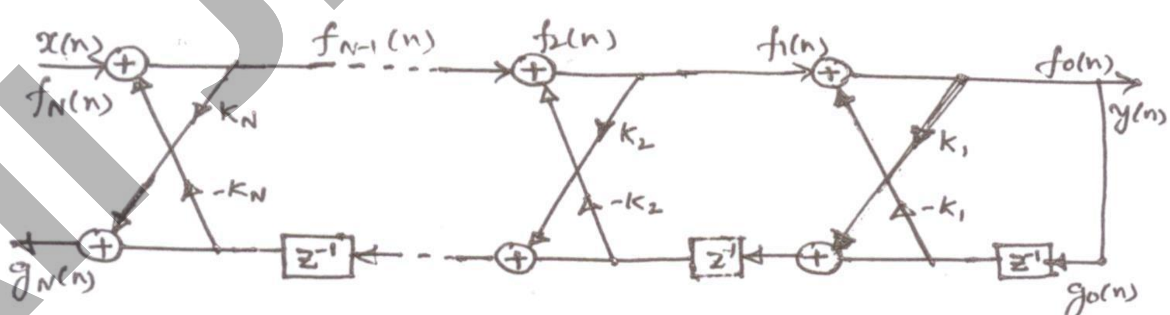
Similarly

$$g_2(n) = k_2 y(n) + k_1(1 + k_2)\, y(n-1) + y(n-2)$$

We observe

$$a_2(0) = 1;\ a_2(1) = k_1(1 + k_2);\ a_2(2) = k_2$$

N-stage IIR filter realized in lattice structure is,



$$f_N(n) = x(n)$$

$$f_{m-1}(n) = f_m(n) - k_m g_{m-1}(n-1) \qquad \text{m=N, N-1,---1}$$

$$g_m(n) = k_m f_{m-1}(n) + g_{m-1}(n-1) \qquad \text{m=N, N-1,---1}$$

$$y(n) = f_0(n) = g_0(n)$$

8.6.1 Conversion from lattice structure to direct form:

$$a_m(m) = k_m \quad ; \qquad a_m(0) = 1$$
$$a_m(k) = a_{m-1}(k) + a_m(m) a_{m-1}(m-k)$$

Conversion from direct form to lattice structure

$$a_{m-1}(0) = 1 \qquad\qquad k_m = a_m(m)$$

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

6.6.2 Lattice – Ladder Structure:

A general IIR filter containing both poles and zeros can be realized using an all pole lattice as the basic building block.

If,

$$H(Z) = \frac{B_M(Z)}{A_N(Z)} = \frac{\sum_{k=0}^{M} b_M(k) Z^{-k}}{1 + \sum_{k=1}^{N} a_N(k) Z^{-k}}$$

Where $N \geq M$

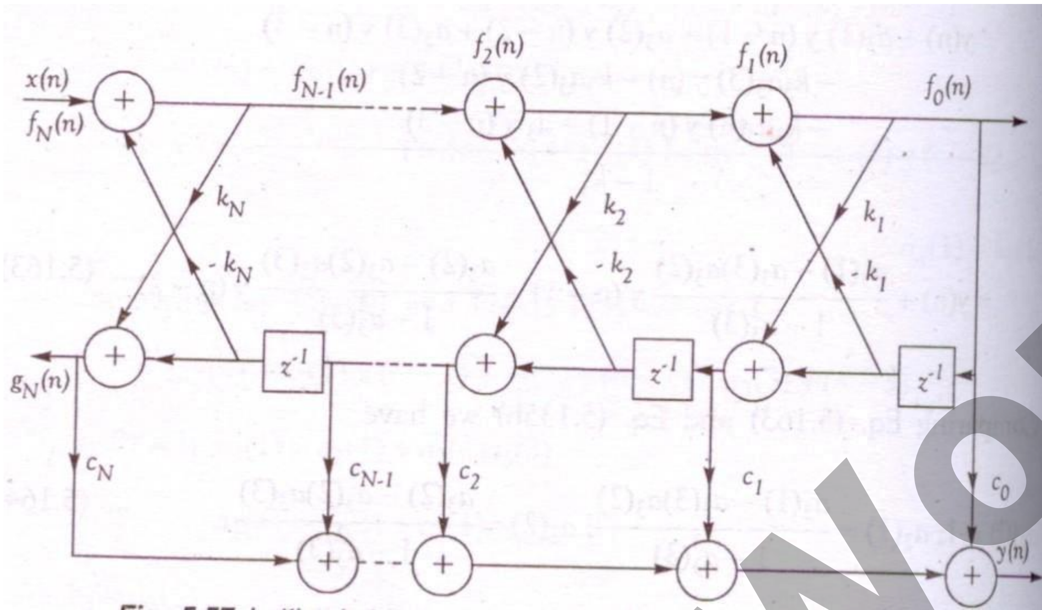
A lattice structure can be constructed by first realizing an all-pole lattice co-efficients k_m, $1 \leq m \leq N$ for the denominator $A_N(Z)$, and then adding a ladder part for M=N. The output of the ladder part can be expressed as a weighted linear combination of $\{g_m(n)\}$.

Now the output is given by

$$y(n) = \sum_{m=0}^{M} C_m\, g_m(n)$$

Where $\{C_m\}$ are called the ladder co-efficient and can be obtained using the recursive relation,

$$C_m = b_m - \sum_{i=m+1}^{M} C_i\, a_i(i-m); \qquad\qquad m=M, M-1, \ldots.0$$



Problem:4

Convert the following pole-zero IIR filter into a lattice ladder structure,

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

Solution:

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

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

for m=3, k=1

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

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

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

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

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

For ladder structure

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

M=3 $\quad C_3 = b_3 = 1; \quad C_2 = b_2 - C_3 a_3(1) = 2 - 1.(\frac{13}{24}) = 1.4583$

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

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

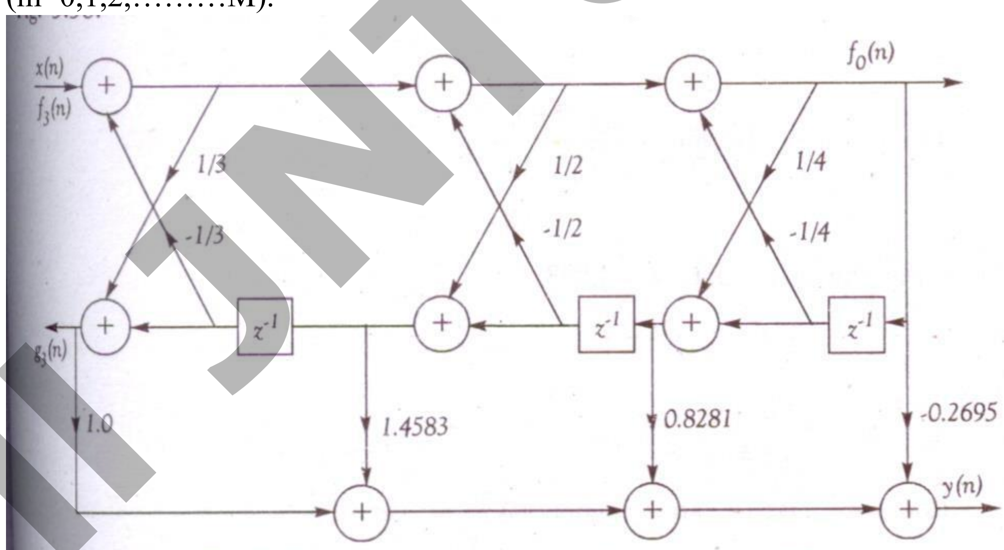
$$= 2 - [(1.4583)(\frac{3}{8}) + \frac{5}{8}] = 0.8281$$

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

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

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

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).



Problem 5

A z-plane pole-zero plot for a certain digital filter is shown in figure 7. The filter has unity gain at DC. Determine the system function in the form

$$H(z) = A\left[\frac{(1+a_1z^{-1})(1+b_1z^{-2}+b_2z^{-2})}{(1+c_1z^{-1})(1+d_1z^{-1}+d_2z^{-2})}\right]$$

giving the numerical values for parameters A, a_1, b_1, b_2, c_1, d_1 and d_2. Sketch the direct form-II and cascade realizations of the system. **[9]**

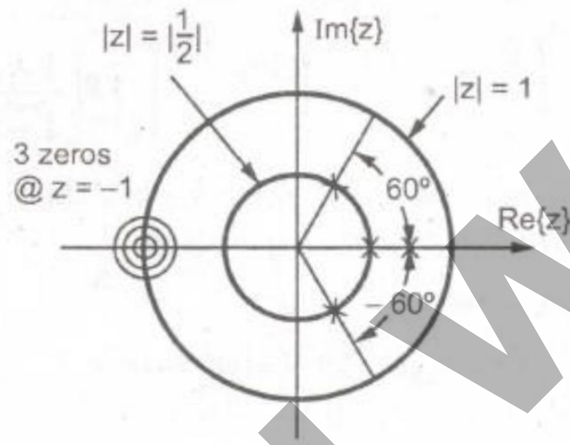


Fig. 7

Sol. :

$$H(z) = A\frac{(1+z^{-1})^3}{\left(1-\frac{1}{2}z^{-1}\right)\left(1-\frac{1}{2}z^{-1}+\frac{1}{4}z^{-1}\right)}$$

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

$$H(z)\big|_{z=1} = 1$$

$$\Rightarrow \quad A = \frac{3}{64}, a_1 = 1, b_1 = 2, b_2 = 1, c_1 = -\frac{1}{2}, d_1 = -\frac{1}{2} \text{ and } d_2 = \frac{1}{4}$$

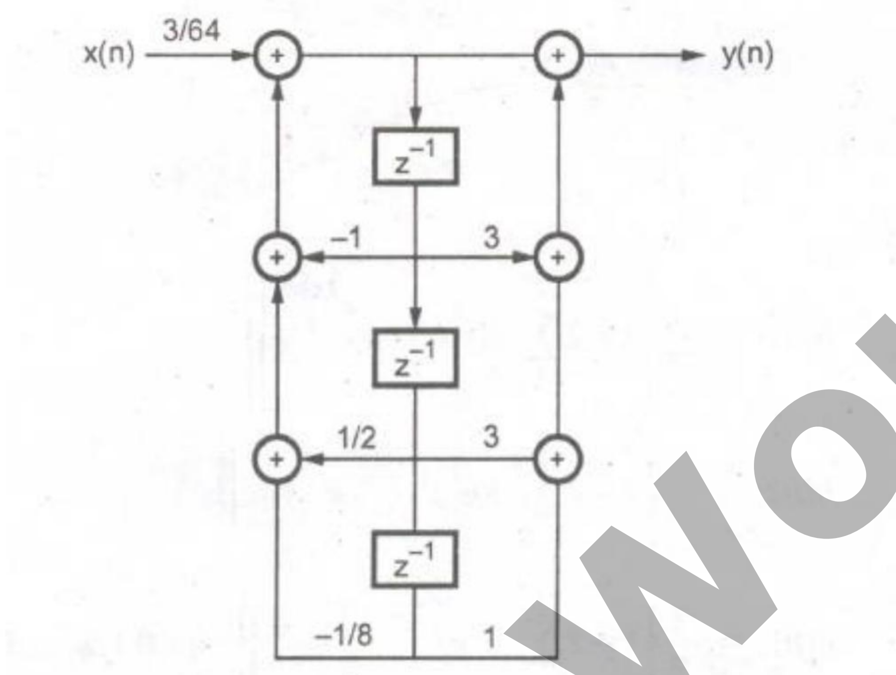


Fig. 8 (a)

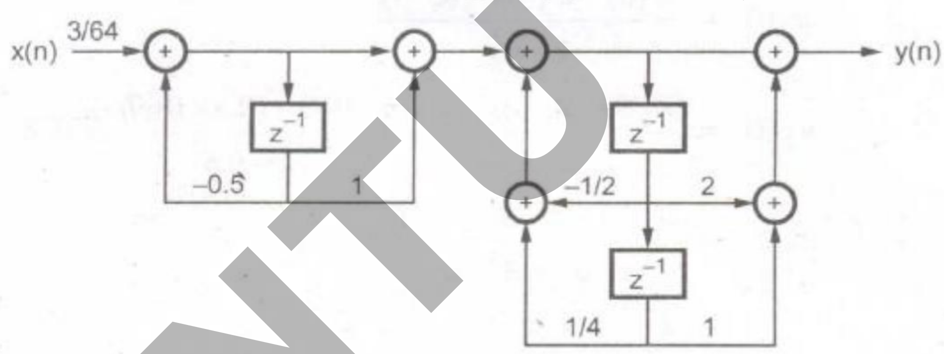


Fig. 8 (b)

Question 6

Consider a FIR filter with system function:

$H(z) = 1+2.82\, Z^{-1} +3.4048z^{-2} +1.74z^{-3}$. Sketch the direct form and lattice realizations of the filter.

Sol. : $A_3(z) = H(z) = 1 + 2.82z^{-1} + 3.4048z^{-2} + 1.74z^{-3}$

$$B_3(z) = 1.74 + 3.4048z^{-1} + 2.82z^{-2} + z^{-3}$$

Hence $k_3 = 1.74$

$$A_2(z) = \frac{A_3(z) - k_3 B_3(z)}{1 - k_4^2}$$

$$= \frac{1 + 2.82z^{-1} + 3.4048z^{-2} + 1.74z^{-3} - 3.0276 - 5.9243z^{-1} - 4.9068z^{-2} - 1.74z^{-3}}{(-2.0276)}$$

$$= \frac{-2.0276 - 3.1043z^{-1} - 1.502z^{-2}}{(-2.0276)} = 1 + 1.531z^{-1} + 0.7407z^{-2}$$

$$B_2(z) = 0.7407 + 1.531z^{-1} + z^{-2}$$

$$k_2 = 0.7407$$

$$A_1(z) = \frac{A_2(z) - k_2 B_2(z)}{1 - k_2^2}$$

$$= \frac{1 + 1.531z^{-1} + 0.7407z^{-2} - 0.5486 - 1.134z^{-1} - 0.7407z^{-2}}{0.4514}$$

$$= 1 + 0.8795z^{-1}$$

$$k_1 = 0.8795$$

Direct form realization :

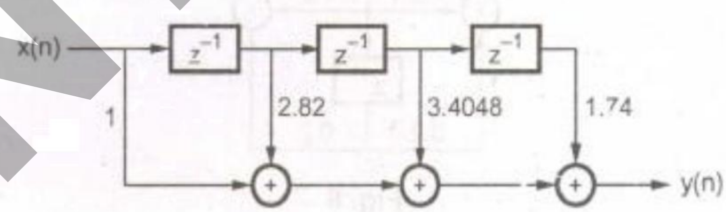


Fig. 5

Lattice realization :

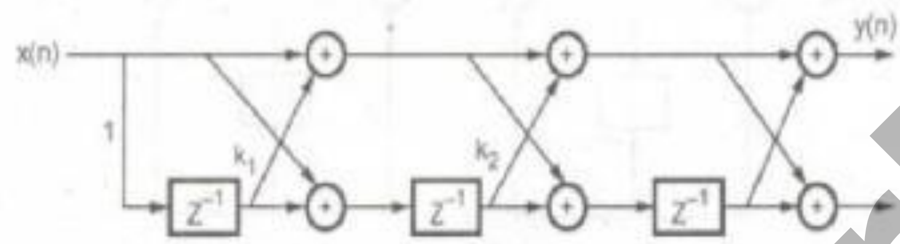


Fig. 6

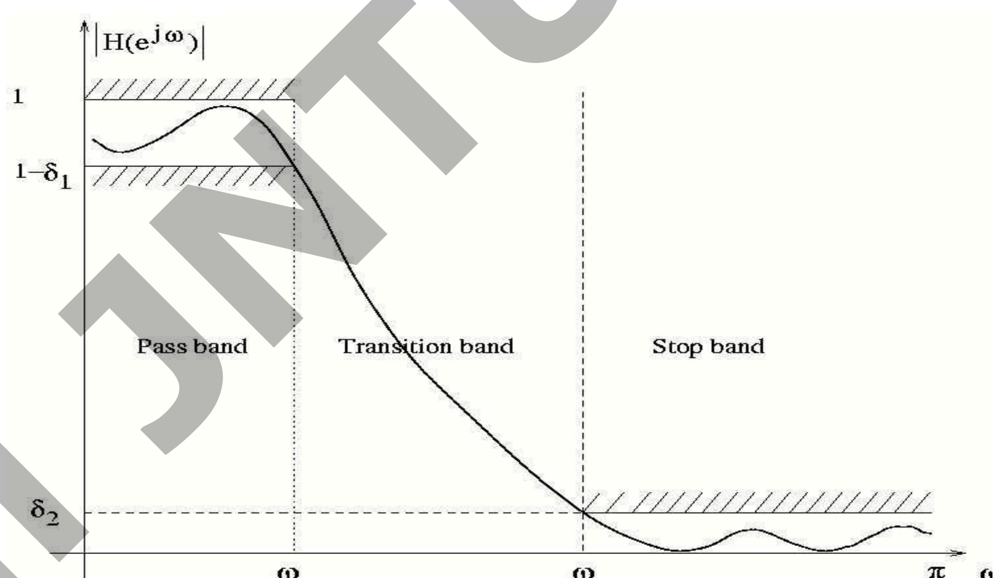
DESIGN OF IIR FILTERS FROM ANALOG FILTERS (BUTTERWORTH AND CHEBYSHEV)

8.1 Introduction

A digital filter is a linear shift-invariant discrete-time system that is realized using finite precision arithmetic. The design of digital filters involves three basic steps:

- The specification of the desired properties of the system.
- The approximation of these specifications using a causal discrete-time system.
- The realization of these specifications using _nite precision arithmetic.

These three steps are independent; here we focus our attention on the second step. The desired digital filter is to be used to filter a digital signal that is derived from an analog signal by means of periodic sampling. The speci_cations for both analog and digital filters are often given in the frequency domain, as for example in the design of low pass, high pass, band pass and band elimination filters. Given the sampling rate, it is straight



forward to convert from frequency specifications on an analog _lter to frequency speci_cations on the corresponding digital filter, the analog frequencies being in terms of Hertz and digital frequencies being in terms of radian frequency or angle around the unit circle with the point Z=-1 corresponding to half the sampling frequency. The least confusing point of view toward digital filter design is to consider the filter as being specified in terms of angle around the unit circle rather than in terms of analog frequencies.

Figure 7.1: Tolerance limits for approximation of ideal low-pass filter

A separate problem is that of determining an appropriate set of specifications on the digital filter. In the case of a low pass filter, for example, the specifications often take the form of a tolerance scheme, as shown in Fig. 4.1

$$1-\delta_1 \le |H(e^{j\omega})| \le 1, \quad |\omega| \le \omega_p$$

$$|H(e^{j\omega})| \le \delta_2, \quad \omega_s \le |\omega| \le \pi$$

Many of the filters used in practice are specified by such a tolerance scheme, with no constraints on the phase response other than those imposed by stability and causality requirements; i.e., the poles of the system function must lie inside the unit circle. Given a set of specifications in the form of Fig. 7.1, the next step is to and a discrete time linear system whose frequency response falls within the prescribed tolerances. At this point the filter design problem becomes a problem in approximation. In the case of infinite impulse response (IIR) filters, we must approximate the desired frequency response by a rational function, while in the finite impulse response (FIR) filters case we are concerned with polynomial approximation.

7.2 Design of IIR Filters from Analog Filters:

The traditional approach to the design of IIR digital filters involves the transformation of an analog filter into a digital filter meeting prescribed specifications. This is a reasonable approach because:

- The art of analog filter design is highly advanced and since useful results can be achieved, it is advantageous to utilize the design procedures already developed for analog filters.
- Many useful analog design methods have relatively simple closed-form design formulas.

Therefore, digital filter design methods based on analog design formulas are rather simple to implement.
An analog system can be described by the differential equation

$$\sum_{k=0}^{N} c_k \frac{d^k y_a(t)}{dt^k} = \sum_{k=0}^{M} d_k \frac{d^k x_a(t)}{dt^k}$$

---7.1

And the corresponding rational function is

$$H_a(s) = \frac{\sum_{k=0}^{M} d_k s^k}{\sum_{k=0}^{N} c_k s^k} = \frac{y_a(s)}{x_a(s)}$$

---7.2

The corresponding description for digital filters has the form

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

--7.3

and the rational function

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

--7.4

In transforming an analog filter to a digital filter we must therefore obtain either H(z)or h(n) (inverse Z-transform of H(z) i.e., impulse response) from the analog filter design. In such transformations, we want the imaginary axis of the S-plane to map into the finite circle of the Z-plane, a stable analog filter should be transformed to a stable digital filter. That is, if the analog filter has poles only in the left-half of S-plane, then the digital filter must have poles only inside the unit circle. These constraints are basic to all the techniques discussed

7.3 IIR Filter Design by Impulse Invariance:

This technique of transforming an analog filter design to a digital filter design corresponds to choosing the unit-sample response of the digital filter as equally spaced samples of the impulse response of the analog filter. That is,

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

7.5 Where T is the sampling period. Because of uniform sampling, we have

$$H(e^{j\omega T}) = \frac{1}{T}\sum_{k=-\infty}^{\infty} H_a(j\Omega + j\frac{2\pi}{T}k)$$

--7.6

Or

$$H(z)\,|_{z=e^{sT}} = \frac{1}{T}\sum_{k=-\infty}^{\infty} H_a(s + j\frac{2\pi}{T}k)$$

--7.7

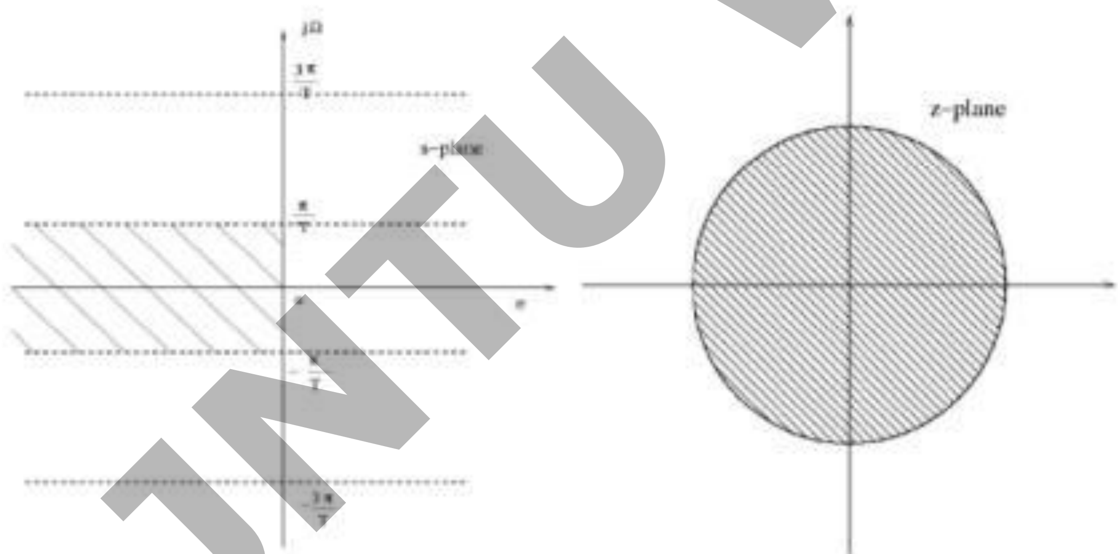


Figure 7.2: Mapping of s-plane into z-plane

Where s = jω and Ω=ω/T, is the frequency in analog domain and ω is the frequency in digital domain.

From the relationship $Z = e^{ST}$ it is seen that strips of width 2π/T in the S-plane map into the entire Z-plane as shown in Fig. 7.2. The left half of each S-plane strip maps into interior of the unit circle, the right half of each S-plane strip maps into the exterior of the unit circle, and the imaginary axis of length 2π/T of S-plane maps on to once round the unit circle of Z-plane. Each horizontal strip of the S-plane is overlaid onto the Z-plane to form the digital filter function from analog filter function. The frequency response of the digital filter is related to the frequency response of the

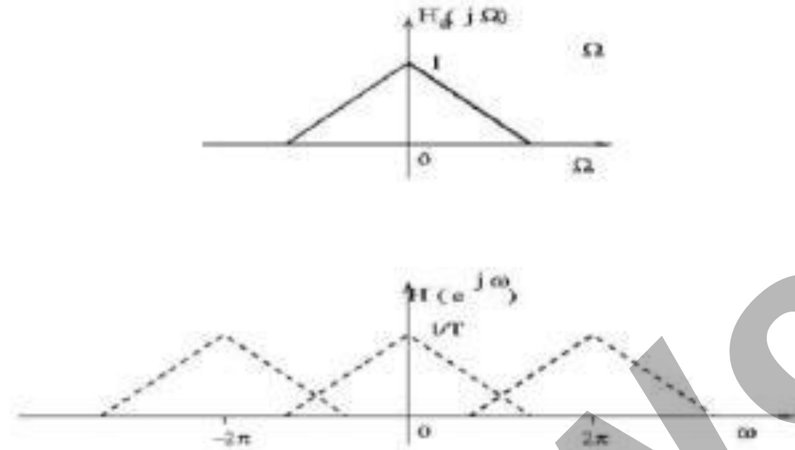


Figure 7.3: Illustration of the effects of aliasing in the impulse invariance technique

analog filter as

$$H(e^{j\omega}) = \frac{1}{T}\sum_{k=-\infty}^{\infty} H_a\left(j\frac{\omega}{T} + j\frac{2\pi}{T}k\right)$$ --7.8

From the discussion of the sampling theorem it is clear that if and only if

$$H_a(j\Omega) = 0, \quad |\,\Omega\,| \geq \frac{\pi}{T}$$

Then

$$H(e^{j\omega}) = \frac{1}{T}H_a\left(j\frac{\omega}{T}\right), \quad |\,\omega\,| \leq \pi$$

Unfortunately, any practical analog filter will not be band limited, and consequently there is interference between successive terms in Eq. (7.8) as illustrated in Fig. 7.3. Because of the aliasing that occurs in the sampling process, the frequency response of the resulting digital filter will not be identical to the original analog frequency response. To get the filter design procedure, let us consider the system function of the analog filter expressed in terms of a partial-fraction expansion

$$H_a(s) = \sum_{k=1}^{N}\frac{A_k}{s - s_k}$$ --- -----7.9

The corresponding impulse response is

$$h_a(t) = \sum_{k=1}^{N} A_k e^{s_k t} U(t)$$ --- 7.10

And the unit-sample response of the digital filter is then

$$h(n) = h_a(nT) = \sum_{k=1}^{N} A_k e^{s_k nT} u(n) = \sum_{k=1}^{N} A_k (e^{s_k T})^n U(n)$$ --------------7.11

The system function of the digital filter H(z) is given by

$$H(z) = \sum_{k=1}^{N} \frac{A_k}{(1 - \exp^{s_k T} z^{-1})}$$ --7.12

In comparing Eqs. (7.9) and (7.12) we observe that a pole at s=sk in the S-plane transforms to a pole at $\exp^{skT}$ in the Z-plane. It is important to recognize that the impulse invariant design procedure does not correspond to a mapping of the S-plane to the Z-plane.

8.4 IIR Filter Design By Approximation Of Derivatives:

A second approach to design of a digital filter is to approximate the derivatives in Eq. (4.1) by finite differences. If the samples are closer together, the approximation to the derivative would be increasingly accurate. For example, suppose that the first derivative is approximated by the first backward difference

$$\frac{dy_a(t)}{dt} \Big|_{t=nT} \longrightarrow \nabla^{(1)}[y(n)] = \frac{y(n) - y(n-1)}{T}$$ --------------------------7.13

Where y(n)=y(nT). Approximation to higher-order derivatives are obtained by repeated application of Eq. (7.13); i.e.,

$$\frac{d^k y_a(t)}{dt^k} \Big|_{t=nT} = \frac{d}{dt}\left(\frac{d^{k-1} y_a(t)}{dt^{k-1}}\right) \Big|_{t=nT} \longrightarrow \nabla^{(k)}[y(n)] = \nabla^{(1)}[\nabla^{(k-1)}[y(n)]]$$ --------------------------7.14

For convenience we define

$$\nabla^{(0)}[y(n)] = y(n)$$ ---7.15

Applying Eqs. (7.13), (7.14) and (7.15) to (7.1), we obtain

$$\sum_{k=0}^{N} c_k \nabla^{(k)}[y(n)] = \sum_{k=0}^{M} d_k \nabla^{(k)}[x(n)]$$ ---7.16

Where y(n) = ya(nT) and x(n) = xa(nT). We note that the operation $\Delta^{(1)}$[ ] is a linear shift-invariant operator and that $\Delta^{(k)}$[ ] can be viewed as a cascade of (k) operators $\Delta^{(1)}$[ ]. In particular

$$Z[\nabla^{(1)}[x(n)]] = [\frac{1 - z^{-1}}{T}]X(z)$$

And

$$Z[\nabla^{(1)}[x(n)]] = [\frac{1 - z^{-1}}{T}]^k X(z)$$

Thus taking the Z-transform of each side in Eq. (7.16), we obtain

$$H(z) = \frac{\sum_{k=0}^{M} d_k [\frac{1-z^{-1}}{T}]^k}{\sum_{k=0} N c_k [\frac{1-z^{-1}}{T}]^k}$$ ---7.17

Comparing Eq. (7.17) to (7.2), we observe that the digital transfer function can be obtained directly from the analog transfer function by means of a substitution of variables

$$s = \frac{1 - z^{-1}}{T}$$ ---7.18

So that, this technique does indeed truly correspond to a mapping of the S-plane to the Z-plane, according to Eq. (7.18). To investigate the properties of this mapping, we must express z as a function of s, obtaining

$$z = \frac{1}{1 - sT}$$

Substituting s = jΩ, i.e., imaginary axis in S-plane

$$
\begin{aligned}
z &= \frac{1}{1 - j\Omega T} \\
&= \frac{1}{1 - j\Omega T} + \frac{1}{2} - \frac{1}{2} \\
&= \frac{1}{2} + \frac{1}{2}[\frac{1 + j\Omega T}{1 - j\Omega T}] \\
&= \frac{1}{2}[1 + \frac{1 + j\Omega T}{1 - j\Omega T}] \\
&= \frac{1}{2}[1 + e^{j2\tan^{-1}(\Omega T)}]
\end{aligned}
$$

--7.19

Which corresponds to a circle whose center is at z =1/2 and radius is 1/2, as shown in Fig. 7.4. It is easily verified that the left half of the S-plane maps into the inside of the small circle and the right half of the S-plane maps onto the outside of the small circle. Therefore, although the requirement of mapping the jΩ-axis to the unit circle is not satisfied, this mapping does satisfy the stability condition.

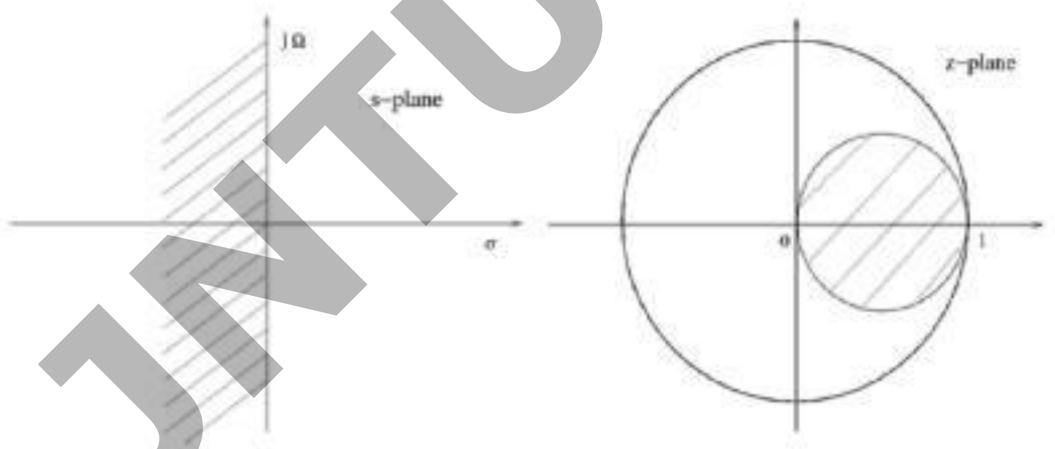


Figure 4.4: Mapping of s-plane to z-plane corresponding to first backward-difference approximation to the derivative

In contrast to the impulse invariance technique, decreasing the sampling period T, theoretically produces a better filter since the spectrum tends to be concentrated in a very small region of the unit circle. These two procedures are highly unsatisfactory for anything but low pass filters. An alternative approximation to the derivative is a forward difference and it provides a mapping into the unstable digital filters.

8.5 IIR Filter Design By The Bilinear Transformation:

In the previous section a digital filter was derived by approximating derivatives by differences. An alternative procedure is based on integrating the differential equation and then using a numerical approximation to the integral. Consider the first - order equation

$$c_1 y_a'(t) + c_0 y_a(t) = d_0 x_a(t)$$

---7.20

Where $y'_a(t)$ is the first derivative of ya(t). The corresponding analog system function is

$$H_a(s) = \frac{d_0}{c_0 + c_1 s}$$

We can write ya(t) as an integral of $y'_a(t)$, as in

$$y_a(t) = \int_{t_0}^{t} y_a'(t)dt + y_a(t_0)$$

In particular, if t = nT and t_0 = (n - 1)T,

$$y_a(nT) = \int_{(n-1)T}^{nT} y_a'(\tau)d\tau + y_a((n-1)T)$$

If this integral is approximated by a trapezoidal rule, we can write

$$y_a(nT) = y_a((n-1)T) + \frac{T}{2}[y_a'(nT) + y_a'((n-1)T)]$$

----------------------7.21

However, from Eq. (7.20),

$$y_a'(nT) = -\frac{c_0}{c_1}y_a(nT) + \frac{d_0}{c_1}x_a(nT)$$

Substituting into Eq. (4.21) we obtain

$$[y(n) - y(n-1)] = \frac{T}{2}[-\frac{c_0}{c_1}(y(n) + y(n-1)) + \frac{d_0}{c_1}(x(n) + x(n-1))]$$

Where y(n) = y(nT) and x(n) = x(nT). Taking the Z-transform and solving for H(z) gives

$$H(z) = \frac{Y(z)}{X(z)} = \frac{d_0}{c_0 + c_1 \frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}}$$

--7.22

From Eq. (7.22) it is clear that H(z) is obtained from Ha(s) by the substitution

$$s = \frac{2}{T}\frac{1 - z^{-1}}{1 + z^{-1}}$$

---7.23

That is,

$$H(z) = H_a(s)\,|_{s=\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}}$$

---7.24

This can be shown to hold in general since an N^{th} - order differential equation of the form of Eq. (7.1) can be written as a set of N first-order equations of the form of Eq. (7.20). Solving Eq. (7.23) for z gives

$$z = \frac{1 + \frac{T}{2}s}{1 - \frac{T}{2}s}$$

---7.25

The invertible transformation of Eq. (7.23) is recognized as a bilinear transformation. To see that this mapping has the property that the imaginary axis in the s-plane maps onto the unit circle in the z-plane, consider $z = e^{j\omega}$, then from Eq. (7.23), s is given by

$$\begin{aligned} s &= \frac{2}{T}\frac{1 - e^{-j\omega}}{1 + e^{-j\omega}} \\ &= \frac{2}{T}\frac{j\sin(\omega/2)}{\cos(\omega/2)} \\ &= \frac{2}{T}j\tan(\omega/2) \\ &= \sigma + j\Omega \end{aligned}$$

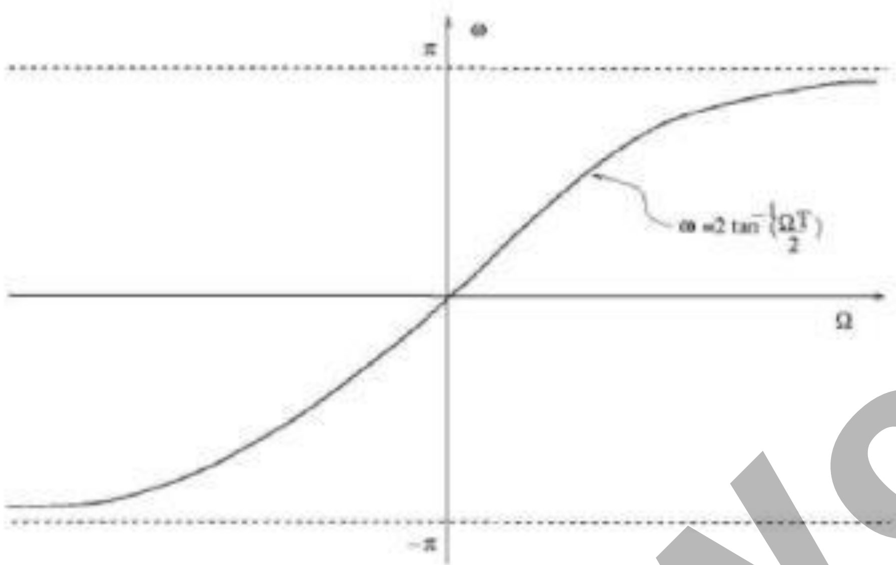


Figure 7.5: Mapping of analog frequency axis onto the unit circle using the bilinear Transformation

Thus for z on the unit circle, $\sigma = 0$ and Ω and ω are related by

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

This relationship is plotted in Fig. (7.5), and it is referred as frequency warping. From the _gure it is clear that the positive and negative imaginary axis of the s-plane are mapped, respectively, into the upper and lower halves of the unit circle in the z-plane. In addition to the fact that the imaginary axis in the s-plane maps into the unit circle in the z-plane, the left half of the s-plane maps to the inside of the unit circle and the right half of the s-plane maps to the outside of the unit circle, as shown in Fig. (7.6). Thus we see that the use of the bilinear transformation yields stable digital filter from analog filter. Also this transformation avoids the problem of aliasing encountered with the use of impulse invariance, because it maps the entire imaginary axis in the s-plane onto the unit circle in the z-plane. The price paid for this, however, is the introduction of a distortion in the frequency axis.

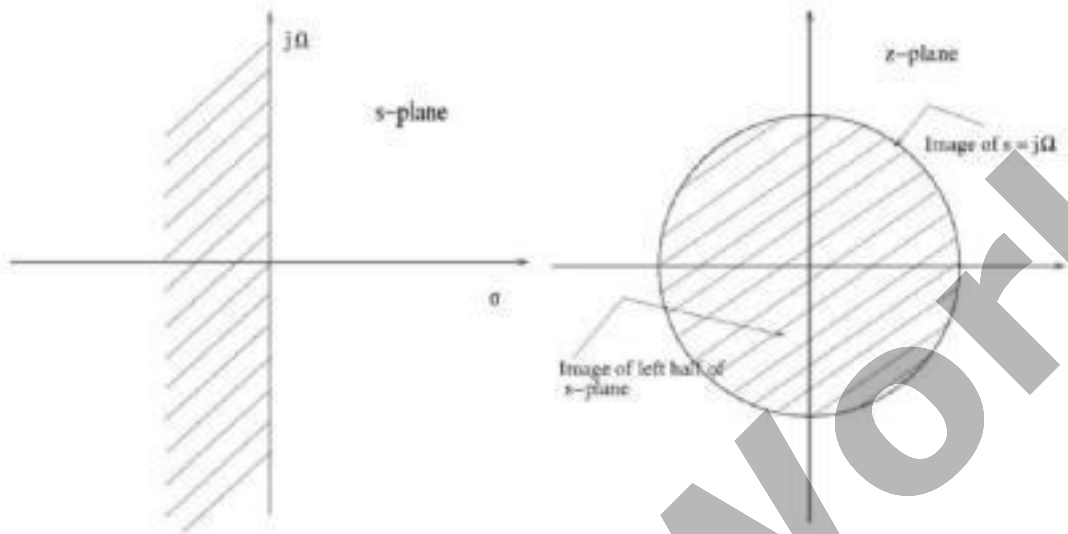


Figure 4.6: Mapping of the s-plane into the z-plane using the bilinear transformation

8.6 The Matched-Z Transform:

Another method for converting an analog filter into an equivalent digital filter is to map the poles and zeros of Ha(s) directly into poles and zeros in the z-plane. For analog filter

$$H_a(s) = \frac{\prod_{k=1}^{M}(s - s_k)}{\prod_{k=1}^{N}(s - p_k)}$$

---7.26

the corresponding digital filter is

$$H(z) = \frac{\prod_{k=1}^{M}(1 - e^{z_k T} z^{-1})}{\prod_{k=1}^{N}(1 - e^{p_k T} z^{-1})}$$

---7.27

Where T is the sampling interval. Thus each factor of the form (s-a) in Ha(s) is mapped into the factor (1- $e^{aT} z^{-1}$).

Recommended questions with solution

Question 1

Design a digital band pass filter from a 2^{nd} order analog low pass Butterworth prototype filter using bilinear transformation. The lower and upper cut-off frequencies for band pass filter are $5\pi/12$ and $7\pi/12$. Assume T = 2 sec. **[12]**

Sol. :

$$\omega_l = \frac{5\pi}{12}$$

$$\Omega_l = \frac{2}{T}\tan\frac{\omega_l}{2}$$

For

$$T = 2$$

$$\Omega_l = \tan\frac{\omega_l}{2}$$

$$\omega_u = \frac{7\pi}{12}$$

$$\therefore \quad \Omega_u = \tan\frac{\omega_u}{2}$$

Analog low pass to band pass

$$s \rightarrow \frac{s^2 + \Omega_l\Omega_u}{s(\Omega_u - \Omega_l)} \qquad \text{... (1)}$$

Analog prototype is,

$$H(s) = \frac{1}{s^2 + \sqrt{2}s + 1} \qquad \text{... (2)}$$

Putting equation (1) in equation (2) and then to get bilinear analog to digital

$$s \rightarrow \frac{z-1}{z+1} = \frac{1-z^{-1}}{1+z^{-1}}$$

Combining above two steps we get

$$s \rightarrow \frac{\left(\dfrac{1-z^{-1}}{1+z^{-1}}\right)^2 + \Omega_l\Omega_u}{\left(\dfrac{1-z^{-1}}{1+z^{-1}}\right)(\Omega_u - \Omega_l)} = \frac{(1-z^{-1})^2 + \Omega_l\,\Omega_u(1+z^{-1})^2}{(1-z^{-2})(\Omega_u - \Omega_l)}$$

$$\therefore \quad H(z) = \frac{1}{\left[\dfrac{(1-z^{-1})^2 + \Omega_l \Omega_u (1+z^{-1})^2}{(1-z^{-2})(\Omega_u - \Omega_l)}\right]^2 + \sqrt{2}\left[\dfrac{(1-z^{-1})^2 + \Omega_l \Omega_u (1+z^{-1})^2}{(1-z^{-2})(\Omega_u - \Omega_l)}\right] + 1}$$

Question 2

Show that the bilinear transformation maps.

i) The $j\Omega$ axis in s-plane onto the unit circle, $|z| = 1$.

ii) The left half s-plane, Re (s) < 0 inside the unit circle, $|z < 1|$.

Sol. :

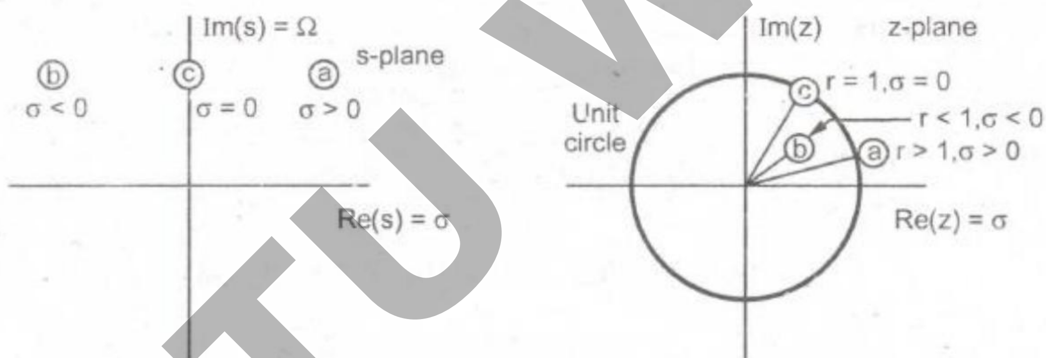


Fig. 3

Here $\quad s = \sigma + j\Omega \quad$ and $z = re^{j\omega}$

Question 3

Fig. 4 shows the frequency response of an infinite-length ideal multi-band real filter. Find. h(n), impulse response of this filter. Present the sketch of implementation of w(n) h(n) (Truncated impulse response of this filter) via block diagram. Where ω (n) is a finite length window sequence ? **[12]**

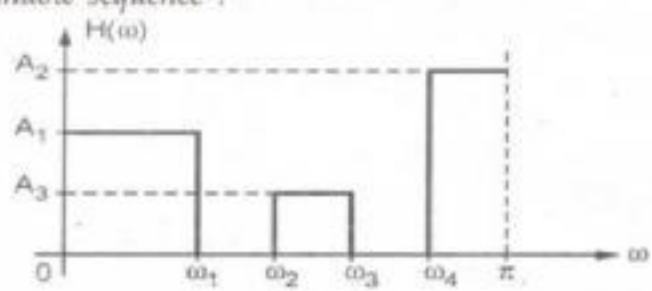


Fig. 4

Question 4

We are interested to design an FIR filter with a stopband attenuation of 64 dB and $\Delta\omega = 0.05\pi$ using windows. Provide the means to achieve precisely this attenuation using suitable window function. **[3]**

Sol. : Hamming window will satisfy the stopband attenuation requirement i.e. 64 dB. Because it has lower transition width.

Hamming window function is given by :

$$\omega(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right) \qquad 0 \le n \le N-1$$

Question 5

The transfer function of analog low pass filter is given by $H(s) = \dfrac{(s-1)}{(s^2-1)(s^2+s+1)}$.

Find H(z) using impulse invariance method. Take T = 1 sec. **[6]**

Sol. :

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

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

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

$$= \frac{1}{(s+1)(s+0.5-j\,0.866)(s+0.5+j\,0.866)}$$

$$= \frac{C_1}{s+1} + \frac{C_2}{s+0.5-j\,0.866} + \frac{C_2^*}{s+0.5+j\,0.866}$$

Using practical fraction expansion, we get

$$C_1 = 1, \quad C_2 = 0.577e^{-j2.62} \quad \text{and} \quad C_2^* = 0.577e^{j2.62}$$

$$\therefore \quad H(s) = \frac{1}{s+1} + \frac{0.577e^{-j2.62}}{s+0.5-j\,0.866} + \frac{0.577e^{j2.62}}{s+0.5+j\,0.866}$$

The three poles are :

$$s_1 = -1, \quad s_2 = -0.5 + j\,0.866 \quad \text{and} \quad s_3 = -0.5 - j\,0.866$$

We know that,

$$H(z) = \sum_{i=1}^{3} \frac{C_i}{1 - e^{S_i T} z^{-1}}$$

$$= \frac{C_1}{1 - e^{s_1 T} z^{-1}} + \frac{C_2}{1 - e^{s_2 T} z^{-1}} + \frac{C_3}{1 - e^{s_3 T} z^{-1}}$$

Here $C_3 = C_2^*$

$$\therefore \quad H(z) = \frac{1}{1 - e^{-T} z^{-1}} + \frac{0.577 e^{-j2.62}}{1 - e^{(-0.5 + j0.866)T} z^{-1}} + \frac{0.577 e^{j2.62}}{1 - e^{(0.5 - j0.866)T} z^{-1}}$$

$$= \frac{1}{1 - e^{-T} z^{-1}} + \frac{0.577 e^{-j2.62}}{1 - e^{-0.5T} e^{j0.866T} z^{-1}} + \frac{0.577 e^{j2.62}}{1 - e^{-0.5T} e^{-j0.866T} z^{-1}}$$

$$= \frac{1}{1 - e^{-T} z^{-1}} + \frac{2(0.577)\cos(-2.62) - 2(0.577) e^{-0.5T} z^{-1} \cos(-2.62 - 0.866T)}{1 - 2e^{-0.5T} \cos(0.866T) z^{-1} + e^{-T} z^{-2}}$$

Multiplying the numerator and denominator of first term on RHS by z and by τ^2 for second term on RHS, above equation becomes,

$$H(z) = \frac{z}{z - e^{-T}} + \frac{-z^2 - 1.154 e^{-0.5T} \cos\left(\frac{5\pi}{6} + 0.866T\right) z}{z^2 - 2e^{-0.5T} \cos(0.866T) z + e^{-T}}$$

In terms of sampling interval T = 1, transfer function is,

$$H(z) = \frac{b_0 z^{-1} + b_1 z^{-2}}{1 - a_1 z^{-1} - a_2 z^{-2} - a_3 z^{-3}}$$

Where $b_0 = -2e^{-0.5T} \cos(0.866T) + e^{-T} + 1.154 e^{-0.5T} \cos\left(\frac{5\pi}{6} + 0.866T\right) = 1.0773$

$b_1 = e^{-T} + 1.154 e^{-1.5T} \cos\left(\frac{5\pi}{6} + 0.866T\right) = 0.1254$

$a_1 = e^{-T} + 2e^{-0.5T} \cos(0.866T) = 1.1538$

$a_2 = -e^{-T} - 2e^{-1.5T} \cos(0.866T) = -0.657$

$a_3 = e^{-2T} = 0.1353$

Question 6

Design a linear phase high pass filter using the Hamming window for the following desired frequency response.

$$H_d(\omega) = \begin{cases} e^{-j3\omega} & \frac{\pi}{6} \le |\omega| \le \pi \\ 0 & |\omega| < \frac{\pi}{6} \end{cases}$$

$\omega(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right)$, *where N is the length of the Hamming window.*

Sol. :

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

$$= \frac{1}{2\pi}\int_{-\pi}^{-\pi/6} e^{-j3\omega}e^{j\omega n}d\omega + \frac{1}{2\pi}\int_{\pi/6}^{\pi} e^{-j3\omega}e^{j\omega n}d\omega$$

$$= \frac{1}{\pi(n-3)}\left[\sin[\pi(n-3)] - \sin\left[\frac{\pi}{6}(n-3)\right]\right] \quad n \neq 3$$

Also, $h_d(3) = \frac{1}{2\pi}\left(\frac{5\pi}{6} + \frac{5\pi}{6}\right) \frac{5}{6}$

Let $N = 7$

Impulse response of FIR filter is :

$$h(n) = h_d(n)\omega(n)$$

$$= \begin{cases} \left\{\dfrac{1}{\pi(n-3)}\left[\sin[\pi(n-3)] - \sin\left[\dfrac{\pi}{6}(n-3)\right]\right]\right\}\left\{0.54 - 0.46\cos\left(\dfrac{2\pi n}{6}\right)\right\} & n \neq 3 \\ \dfrac{5}{6}\left[0.54 - 0.46\cos\left(\dfrac{2\pi n}{6}\right)\right] & n = 3 \end{cases}$$

n	$h_d(n)$	$\omega(n)$	h(n)
0	– 0.1061	0.08	0.0085
1	– 0.1378	0.31	0.0427
2	– 0.1592	0.77	0.1226
3	0.8333	1	0.8333
4	0.1592	0.77	0.1226
5	0.1378	0.31	0.0427
6	0.1061	0.08	0.0085

Question 7

Design a digital lowpass Butterworth filter using bilinear transformation method to meet the following specifications. Take T = 2 sec.

Passband ripple ≤ 1.25 dB

Passband edge = 200 Hz

Stopband attenuation ≥ 15 dB

Stopband edge = 400 Hz

Sampling frequency = 2 kHz **[12]**

Sol. :

$$\Omega_P = 2\pi \times 200 = 400\pi\, rad/\text{sec}$$

$$\Omega_s = 2\pi \times 400 = 800\pi\, rad/\text{sec}$$

$$T_s = \frac{1}{f_s} = \frac{1}{2000}\text{ sec}$$

$$\omega_P = \Omega_P T_s = 400\pi \times \frac{1}{2000} = 0.2\pi \text{ rad}$$

$$\omega_s = \Omega_P T_s = 800\pi \times \frac{1}{2000} = 0.4\pi \text{ rad}$$

Given : T = 2 sec.

$$\Omega'_p = \frac{2}{T}\tan\left(\frac{\omega_p}{2}\right) = \tan\left(\frac{0.2\pi}{2}\right) = 0.3249$$

$$\Omega'_s = \frac{2}{T}\tan\left(\frac{\omega_s}{2}\right) = \tan\left(\frac{0.4\pi}{2}\right) = 0.7265$$

$$N = \frac{log\left[(10^{-k_P/10} - 1)/(10^{-k_s/10} - 1)\right]}{2log\,(\Omega'_P/\Omega'_s)}$$

$$= \frac{log(0.3335/30.6228)}{2\,log\,(0.3249/0.7265)} = 2.8083 \cong 3$$

$$\Omega_c = \frac{\Omega'_P}{(10^{-k_P/10} - 1)^{1/2N}} = 1.7688$$

Referring to normalized lowpass butterworth filter tables

$$H_3(s) = \frac{1}{(s^2 + s + 1)(s + 1)}$$

The required prewarped analog filter is obtained by applying lowpass to lowpass transformation.

$$H_a(s) = H_3(s)\big|_{s \to \frac{s}{2}}$$

$$= \frac{1}{(s^2 + s + 1)(s + 1)}\bigg|_{s \to \frac{s}{2}}$$

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

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

$$= \frac{1}{\left(\frac{2s^2 + 4s + 8}{8}\right)\left(\frac{s + 2}{2}\right)}$$

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

$$= \frac{8}{(s^2+2s+4)(s+2)} = \frac{8}{s^3+2s^2+2s^2+4s+8}$$

$$= \frac{8}{s^3+4s^2+8s+8}$$

Applying bilinear transformation to $H_a(s)$

$$H(z) = H_3(s)\Big|_{s \to \frac{2}{T}\left(\frac{1-z^{-1}}{1+z^{-1}}\right)}$$

$$\therefore \quad H(z) = \frac{8}{\left(\frac{1-z^{-1}}{1+z^{-1}}\right)^3 + 4\left(\frac{1-z^{-1}}{1+z^{-1}}\right)^2 + 8\left(\frac{1-z^{-1}}{1+z^{-1}}\right) + 8}$$

UNIT 4
Design of FIR Filters

7.1 Introduction:

Two important classes of digital filters based on impulse response type are

Finite Impulse Response (FIR)

Infinite Impulse Response (IIR)

The filter can be expressed in two important forms as:

1) System function representation;

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

2) Difference Equation representation;

$$\sum_{k=0}^{N} a_k y(n-k) = \sum_{k=0}^{M} b_k x(n-k) \qquad (2)$$

Each of this form allows various methods of implementation. The eq (2) can be viewed as a computational procedure (an algorithm) for determining the output sequence y(n) of the system from the input sequence x(n). Different realizations are possible with different arrangements of eq (2)

The major issues considered while designing a digital filters are :

- Realiability (causal or non causal)
- Stability (filter output will not saturate)
- Sharp Cutoff Characteristics
- Order of the filter need to be minimum (this leads to less delay)
- Generalized procedure (having single procedure for all kinds of filters)
- Linear phase characteristics

The factors considered with filter implementation are ,

a. It must be a simple design

b. There must be modularity in the implementation so that any order filter can be obtained with lower order modules.

c. Designs must be as general as possible. Having different design procedures for different types of filters(high pass, low pass,…) is cumbersome and complex.

d. Cost of implementation must be as low as possible

e. The choice of Software/Hardware realization

7.2 Features of IIR:

The important features of this class of filters can be listed as:

- Out put is a function of past o/p, present and past i/p's
- It is recursive in nature
- It has at least one Pole (in general poles and zeros)
- Sharp cutoff chas. is achievable with minimum order
- Difficult to have linear phase chas over full range of freq.
- Typical design procedure is analog design then conversion from analog to digital

7.3 Features of FIR : The main features of FIR filter are,

- They are inherently stable
- Filters with linear phase characteristics can be designed
- Simple implementation – both recursive and nonrecursive structures possible
- Free of limit cycle oscillations when implemented on a finite-word length digital system

7.3.1 Disadvantages:

- Sharp cutoff at the cost of higher order
- Higher order leading to more delay, more memory and higher cost of implementation

7.4 Importance of Linear Phase:

The group delay is defined as

$$\tau_g = -\frac{d\theta(\omega)}{d\omega}$$

which is negative differential of phase function.

Nonlinear phase results in different frequencies experiencing different delay and arriving at different time at the receiver. This creates problems with speech processing and data communication applications. Having linear phase ensures constant group delay for all frequencies.

The further discussions are focused on FIR filter.

6.5 Examples of simple FIR filtering operations: ***1.Unity Gain Filter***

y(n)=x(n)

2. Constant gain filter

y(n)=Kx(n)

3. Unit delay filter

y(n)=x(n-1)

4.Two - term Difference filter

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

5. Two-term average filter

y(n) = 0.5(x(n)+x(n-1))

6. Three-term average filter (3-point moving average

filter) y(n) = 1/3[x(n)+x(n-1)+x(n-2)]

7. Central Difference filter

y(n)= 1/2[x(n) – x(n-2)]

When we say Order of the filter it is the number of previous inputs used to compute the current output and Filter coefficients are the numbers associated with each of the terms x(n), x(n-1),.. etc

The table below shows order and filter coefficients of above simple filter types:

Ex.	order	a0	a1	a2
1	0	1	-	-
2	0	K	-	-
3	1	0	1	-
4(HP)	1	1	-1	-
5(LP)	1	1/2	1/2	-
6(LP)	2	1/3	1/3	1/3
7(HP)	2	1/2	0	-1/2

7.6 Design of FIR filters:

The section to follow will discuss on design of FIR filter. Since linear phase can be achieved with FIR filter we will discuss the conditions required to achieve this.

7.6.1 Symmetric and Antisymmetric FIR filters giving out Linear Phase characteristics:

Symmetry in filter impulse response will ensure linear phase

An FIR filter of length M with i/p x(n) & o/p y(n) is described by the difference equation:

$$y(n)= b_0\, x(n) + b_1\, x(n-1)+\ldots\ldots.+b_{M-1}\, x(n-(M-1)) = \sum_{k=0}^{M-1} b_k\, x(n-k) \qquad -(1)$$

Alternatively. it can be expressed in convolution form

$$y(n) = \sum_{k=0}^{M-1} h(k)x(n-k) \qquad - (2)$$

i.e b_k= h(k), k=0,1,…..M-1

Filter is also characterized by

$H(z)=\sum_{k=0}^{M-1} h(k)z^{-k}$ -(3) polynomial of degree M-1 in the variable z^{-1}. The roots of this polynomial constitute zeros of the filter.

An FIR filter has linear phase if its unit sample response satisfies the condition
h(n)= ± h(M-1-n) n=0,1,…….M-1 -(4)

Incorporating this symmetry & anti symmetry condition in eq 3 we can show linear phase chas of FIR filters

$$H(z)=h(0)+h(1)z^{-1}+h(2)z^{-2}+..........+h(M-2)z^{-(M-2)}+h(M-1)z^{-(M-1)}$$

If M is odd

$$H(z)=h(0)+h(1)z^{-1}+.........+h\left(\frac{M-1}{2}\right)z^{-\left(\frac{M-1}{2}\right)}+h\left(\frac{M+1}{2}\right)z^{-\left(\frac{M+1}{2}\right)}+h\left(\frac{M+3}{2}\right)z^{-\left(\frac{M+3}{2}\right)}+..........$$
$$+h(M-2)z^{-(M-2)}+h(M-1)z^{-(M-1)}$$

$$=z^{-\left(\frac{M-1}{2}\right)}\left[h(0)z^{\left(\frac{M-1}{2}\right)}+h(1)z^{\left(\frac{M-3}{2}\right)}+...........+h\left(\frac{M-1}{2}\right)+h\left(\frac{M+1}{2}\right)z^{-1}+h\left(\frac{M+3}{2}\right)z^{-2}+.....h(M-1)z^{-\left(\frac{M-1}{2}\right)}\right]$$

Applying symmetry conditions for M odd

$$h(0)=\pm h(M-1)$$
$$h(1)=\pm h(M-2)$$
.
.
$$h\left(\frac{M-1}{2}\right)=\pm h\left(\frac{M-1}{2}\right)$$
$$h\left(\frac{M+1}{2}\right)=\pm h\left(\frac{M-3}{2}\right)$$
.
.
$$h(M-1)=\pm h(0)$$

$$H(z) = z^{-(\frac{M-1}{2})}\left[h(\frac{M-1}{2}) + \sum_{n=0}^{\frac{M-3}{2}} h(n)\{z^{(M-1-2n)/2} \pm z^{-(M-1-2n)/2}\} \right]$$

similarly for M even

$$H(z) = z^{-(\frac{M-1}{2})}\left[\sum_{n=0}^{\frac{M}{2}-1} h(n)\{z^{(M-1-2n)/2} \pm z^{-(M-1-2n)/2}\} \right]$$

7.6.2 Frequency response:

If the system impulse response has symmetry property (i.e.,h(n)=h(M-1-n)) and M is odd $H(e^{j\omega}) = e^{j\theta(\omega)} \, |H_r(e^{j\omega})|$ where

$$H_r(e^{j\omega}) = \left[h(\frac{M-1}{2}) + 2\sum_{n=0}^{\frac{M-3}{2}} h(n) \cos \omega(\frac{M-1}{2} - n) \right]$$

$$\theta(\omega) = -(\frac{M-1}{2})\omega \quad if\ |H_r(e^{j\omega})| \geq 0$$

$$= -(\frac{M-1}{2})\omega + \pi \quad if\ |H_r(e^{j\omega})| \leq 0$$

In case of M even the phase response remains the same with magnitude response expressed as

$$H_r(e^{j\omega}) = \left[2\sum_{n=0}^{\frac{M}{2}-1} h(n) \cos \omega(\frac{M-1}{2} - n) \right]$$

If the impulse response satisfies anti symmetry property (i.e., h(n)=-h(M-1-n))then for M odd we will have

$$h(\frac{M-1}{2}) = -h(\frac{M-1}{2})\ i.e.,\ h(\frac{M-1}{2}) = 0$$

$$H_r(e^{j\omega}) = \left[2\sum_{n=0}^{\frac{M-3}{2}} h(n) \sin \omega(\frac{M-1}{2} - n) \right]$$

If M is even then,

$$H_r(e^{j\omega}) = \left[2\sum_{n=0}^{\frac{M-1}{2}} h(n) \sin \omega\left(\frac{M-1}{2} - n\right) \right]$$

In both cases the phase response is given by

$$\theta(\omega) = -\left(\frac{M-1}{2}\right)\omega + \pi/2 \quad if\,|H_r(e^{j\omega})| \geq 0$$
$$= -\left(\frac{M-1}{2}\right)\omega + 3\pi/2 \quad if\,|H_r(e^{j\omega})| \leq 0$$

Which clearly shows presence of Linear Phase characteristics.

7.6.3 Comments on filter coefficients:

- The number of filter coefficients that specify the frequency response is (M+1)/2 when is M odd and M/2 when M is even in case of symmetric conditions
- In case of impulse response antisymmetric we have h(M-1/2)=0 so that there are (M-1/2) filter coefficients when M is odd and M/2 coefficients when M is even

7.6.5 Choice of Symmetric and antisymmetric unit sample response

When we have a choice between different symmetric properties, the particular one is picked up based on application for which the filter is used. The following points give an insight to this issue.

- If h(n)=-h(M-1-n) and M is odd, $H_r(w)$ implies that Hr(0)=0 & $H_r(\pi)$=0, consequently not suited for lowpass and highpass filter. This condition is suited in Band Pass filter design.
- Similarly if M is even $H_r(0)$=0 hence not used for low pass filter
- Symmetry condition h(n)=h(M-1-n) yields a linear-phase FIR filter with non zero response at w = 0 if desired.

Looking at these points, antisymmetric properties are not generally preferred.

7.6.6 Zeros of Linear Phase FIR Filters:

Consider the filter system function

$$H(z) = \sum_{n=o}^{M-1} h(n) z^{-n}$$

Expanding this equation

$$H(z) = h(0) + h(1)z^{-1} + h(2)z^{-2} \quad ++ h(M-2)z^{-(M-2)} + h(M-1)z^{-(M-1)}$$

$\sin ce\ for\ Linear - phase\ we\ need$

$h(n) = h(M-1-n) \quad i.e.,$

$h(0) = h(M-1);\ h(1) = h(M-2); ......h(M-1) = h(0);$

$then$

$$H(z) = h(M-1) + h(M-2)z^{-1} + ... + h(1)z^{-(M-2)} + h(0)z^{-(M-1)}$$

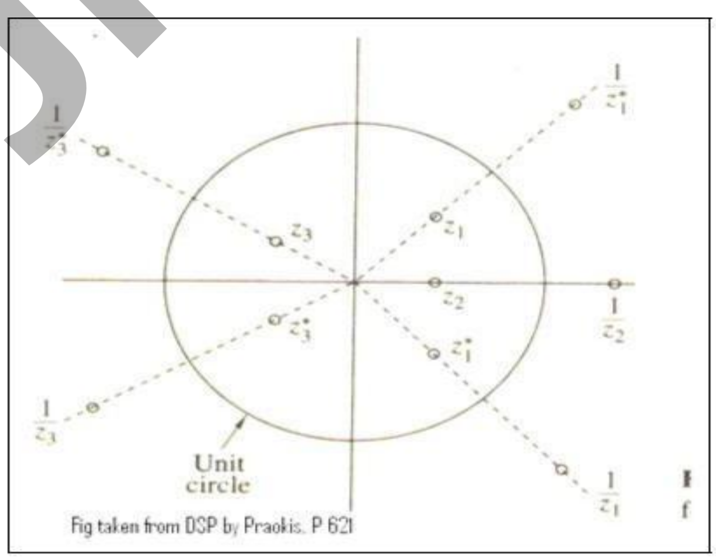
$$H(z) = z^{-(M-1)}[h(M-1)z^{(M-1)} + h(M-2)z^{(M-2)} + ... + h(1)z + h(0)]$$

$$H(z) = z^{-(M-1)}[\sum_{n=0}^{M-1} h(n)(z^{-1})^{-n}] = z^{-(M-1)} H(z^{-1})$$

This shows that if $z = z_1$ is a zero then $z = z_1^{-1}$ is also a zero

The different possibilities:

1. If $z_1 = 1$ then $z_1 = z_1^{-1} = 1$ is also a zero implying it is one zero
2. If the zero is real and |z|<1 then we have pair of zeros
3. If zero is complex and |z|=1then and we again have pair of complex zeros.
4. If zero is complex and |z|≠1 then and we have two pairs of complex zeros



The plot above shows distribution of zeros for a Linear – phase FIR filter. As it can be seen there is pattern in distribution of these zeros.

7.7 Methods of designing FIR filters:

The standard methods of designing FIR filter can be listed as:

1. Fourier series based method
2. Window based method
3. Frequency sampling method

7.7.1 Design of Linear Phase FIR filter based on Fourier Series method:

Motivation: Since the desired freq response $H_d(e^{j\omega})$ is a periodic function in ω with period 2π, it can be expressed as Fourier series expansion

$$H_d(e^{j\omega}) = \sum_{n=-\infty}^{\infty} h_d(n) e^{-j\omega n}$$

where $h_d(n)$ are fourier series coefficients

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

This expansion results in impulse response coefficients which are infinite in duration and non causal. It can be made finite duration by truncating the infinite length. The linear phase can be obtained by introducing symmetric property in the filter impulse response, i.e., h(n) = h(-n). It can be made causal by introducing sufficient delay (depends on filter length)

7.7.2 Stepwise procedure:

1. From the desired freq response using inverse FT relation obtain $h_d(n)$
2. Truncate the infinite length of the impulse response to finite length with (assuming M odd)

$$h(n) = h_d(n) \quad for -(M-1)/2 \le n \le (M-1)/2$$
$$= 0 \quad otherwise$$

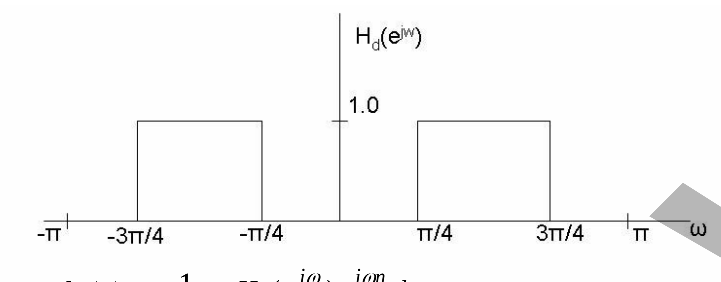
3. Introduce h(n) = h(-n) for linear phase characteristics
4. Write the expression for H(z); this is non-causal realization
5. To obtain causal realization $H'(z) = z^{-(M-1)/2} H(z)$

Exercise Problems

Problem 1 : Design an ideal bandpass filter with a frequency response:

$$H_d(e^{j\omega}) = 1 \qquad for\ \frac{\pi}{4} \le |\omega| \le \frac{3\pi}{4}$$
$$= 0 \qquad otherwise$$

Find the values of h(n) for M = 11 and plot the frequency response.



$$h_d(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d(e^{j\omega})e^{j\omega n}\,d\omega$$
$$= \frac{1}{2\pi}\left[\int_{-3\pi/4}^{-\pi/4} e^{j\omega n}\,d\omega + \int_{\pi/4}^{3\pi/4} e^{j\omega n}\,d\omega\right]$$
$$= \frac{1}{\pi n}\left[\sin\frac{3\pi}{4}n - \sin\frac{\pi}{4}n\right] \quad -\infty \le n \le \infty$$

truncating to 11 *samples we have* $h(n) = h_d(n)$ *for* $|n| \le$
5 = 0 *otherwise*

For n = 0 the value of h(n) is separately evaluated from the basic

integration h(0) = 0.5

Other values of h(n) are evaluated from h(n)

expression h(1)=h(-1)=0

h(2)=h(-2)=-

0.3183 h(3)=h(-

3)=0 h(4)=h(-4)=0

h(5)=h(-5)=0

The transfer function of the filter is

$$H(z) = h(0) + \sum_{n=1}^{(N-1)/2} [h(n)\{z^{n} + z^{-n}\}]$$
$$= 0.5 - 0.3183(z^{2} + z^{-2})$$

the transfer function of the realizable filter is

$$H'(z) = z^{-5}[0.5 - 0.3183(z^{2} + z^{-2})]$$
$$= -0.3183z^{-3} + 0.5z^{-5} - 0.3183z^{-7}$$

the filter coeff are

$$h'(0) = h'(10) = h'(1) = h'(9) = h'(2) = h'(8) = h'(4) = h'(6) = 0$$
$$h'(3) = h'(7) = -0.3183$$
$$h'(5) = 0.5$$

The magnitude response can be expressed as

$$|H(e^{j\omega})| = \sum_{n=1}^{(N-1)/2} a(n)\cos\omega n$$

comparing this exp *with*

$$|H(e^{j\omega})| = |z^{-5}[h(0) + 2\sum_{n=1}^{5} h(n)\cos\omega n]|$$

We have a(0)=h(0)

a(1)=2h(1)=0

a(2)=2h(2)=-0.6366

a(3)=2h(3)=0

a(4)=2h(4)=0

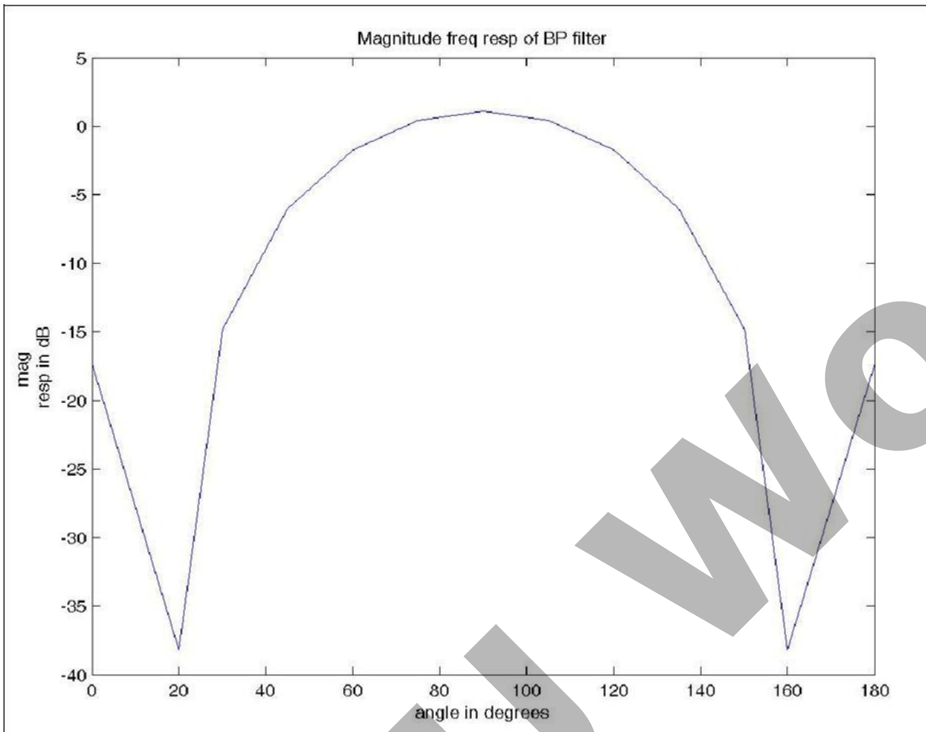
a(5)=2h(5)=0

The magnitude response function is

$|H(e^{j\omega})| = 0.5 - 0.6366 \cos 2\omega$ which can plotted for various values of ω

ω in degrees =[0 20 30 45 60 75 90 105 120 135 150 160 180];

$|H(e^{j\omega})|$ in dBs= [-17.3 -38.17 -14.8 -6.02 -1.74 0.4346 1.11 0.4346 -1.74 -6.02 -14.8 -38.17 -17.3];



Problem 2: Design an ideal lowpass filter with a freq response

$$H_d(e^{j\omega}) = 1 \quad for -\frac{\pi}{2} \le \omega \le \frac{\pi}{2}$$

$$= 0 \quad for \frac{\pi}{2} \le |\omega| \le \pi$$

Find the values of h(n) for N =11. Find H(z). Plot the magnitude response

From the freq response we can determine $h_d(n)$,

$$h_d(n) = \frac{1}{2\pi}\int_{-\pi/2}^{\pi/2} e^{j\omega n}\, d\omega = \frac{\sin\frac{\pi n}{2}}{\pi n} \quad -\infty \le n \le \infty \ \ and \quad n \ne 0$$

Truncating $h_d(n)$ to 11 samples

h(0) = 1/2 h(1)=h(-1)=0.3183 h(2)=h(-2)=0 h(3)=h(-3)=-0.106

h(4)=h(-4)=0
h(5)=h(-5)=0.06366

The realizable filter can be obtained by shifting h(n) by 5 samples to right h'(n)=h(n-5)

h'(n)= [0.06366, 0, -0.106, 0, 0.3183, 0.5, 0.3183, 0, -0.106, 0, 0.06366];

$$H'(z) = 0.06366 - 0.106z^{-2} + 0.3183z^{-4} + 0.5z^{-5} + 0.3183z^{-6} - 0.106z^{-8} + 0.06366z^{-10}$$

Using the result of magnitude response for M odd and symmetry

$$H_r(e^{j\omega}) = [h(\frac{M-1}{2}) + \sum_{n=0}^{\frac{M-3}{2}} h(n)\cos\omega(\frac{M-1}{2} - n)]$$

$$|H_r(e^{j\omega})| = |[0.5 + 0.6366\cos w - 0.212\cos 3w + 0.127\cos 5w]|$$

Problem 3 :

Design an ideal band reject filter with a frequency response:

$$H_d(e^{j\omega}) = 1 \qquad for\ |\omega| \le \frac{\pi}{3}\ and\ |\omega| \ge \frac{2\pi}{3}$$
$$= 0 \qquad otherwise$$

Find the values of h(n) for M = 11 and plot the frequency response

Ans:h(n)= [0 -0.1378 0 0.2757 0 0.667 0 0.2757 0 -0.1378 0];

7.8 Window based Linear Phase FIR filter design

The other important method of designing FIR filter is by making use of windows. The arbitrary truncation of impulse response obtained through inverse Fourier relation can lead to distortions in the final frequency response.The arbitrary truncation is equivalent to multiplying infinite length function with finite length rectangular window, i.e.,

h(n) = h_d(n) w(n) where w(n) = 1 for n = ±(M-1)/2

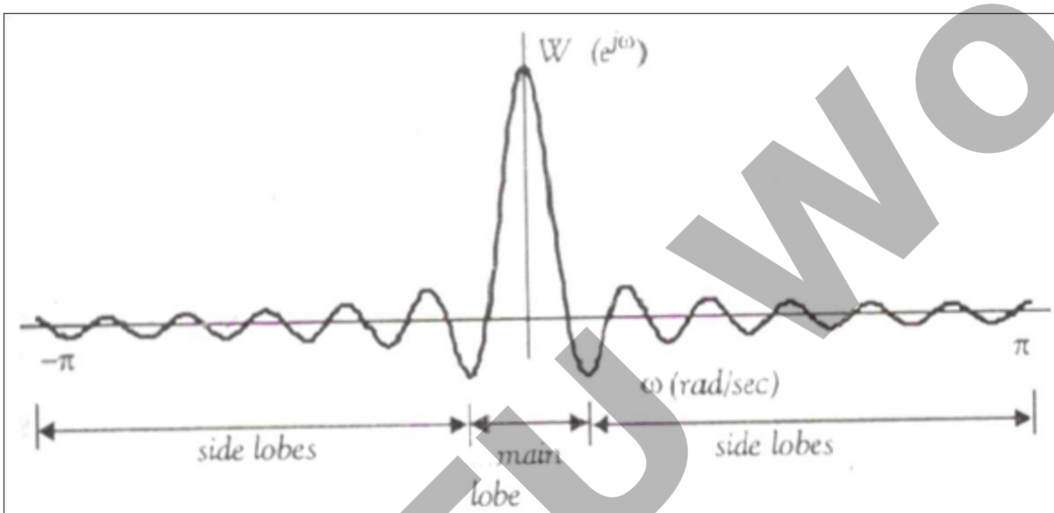
The above multiplication in time domain corresponds to convolution in freq domain, i.e.,

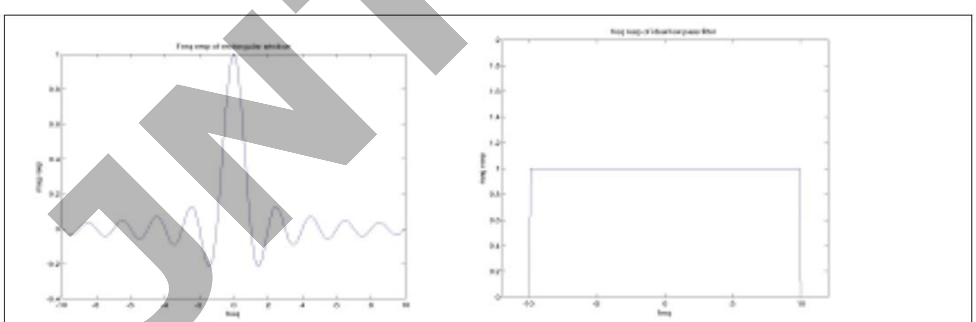
$H(e^{j\omega}) = H_d(e^{j\omega}) * W(e^{j\omega})$ where $W(e^{j\omega})$ is the FT of window function w(n).

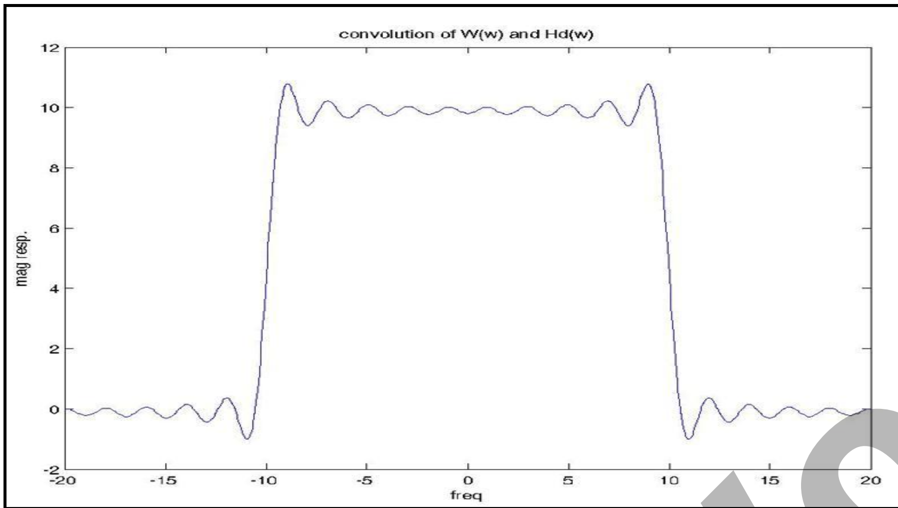
The FT of w(n) is given by

$$W(e^{j\omega}) = \frac{\sin(\omega M/2)}{\sin(\omega/2)}$$

The whole process of multiplying h(n) by a window function and its effect in freq domain are shown in below set of figures.







Suppose the filter to be designed is Low pass filter then the convolution of ideal filter freq response and window function freq response results in distortion in the resultant filter freq response. The ideal sharp cutoff chars are lost and presence of ringing effect is seen at the band edges which is referred to Gibbs Phenomena. This is due to main lobe width and side lobes of the window function freq response.The main lobe width introduces transition band and side lobes results in rippling characters in pass band and stop band. Smaller the main lobe width smaller will be the transition band. The ripples will be of low amplitude if the peak of the first side lobe is far below the main lobe peak.

7.8.1 How to reduce the distortions?

1. Increase length of the window

- as M increases the main lob width becomes narrower, hence the transition band width is decreased

-With increase in length the side lobe width is decreased but height of each side lobe increases in such a manner that the area under each sidelobe remains invariant to changes in M. Thus ripples and ringing effect in pass-band and stop-band are not changed.

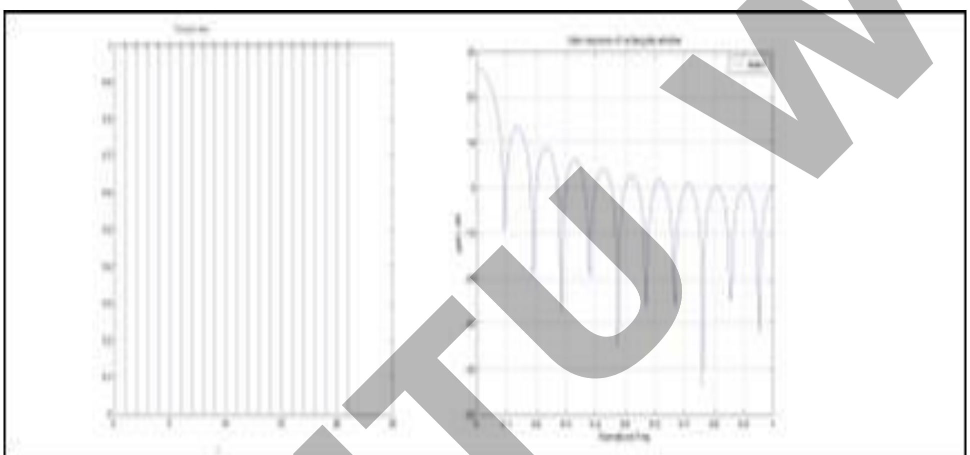
2. Choose windows which tapers off slowly rather than ending abruptly - Slow tapering reduces ringing and ripples but generally increases transition width since main lobe width of these kind of windows are larger.

7.8.2 What is ideal window characteristics?

Window having very small main lobe width with most of the energy contained with it (i.e.,ideal window freq response must be impulsive).Window design is a mathematical problem, more complex the window lesser are the distortions. Rectangular window is one of the simplest window in terms of computational complexity. Windows better than rectangular window are, Hamming, Hanning, Blackman, Bartlett, Traingular,Kaiser. The different window functions are discussed in the following sention.

7.8.3 Rectangular window: The mathematical description is given by,

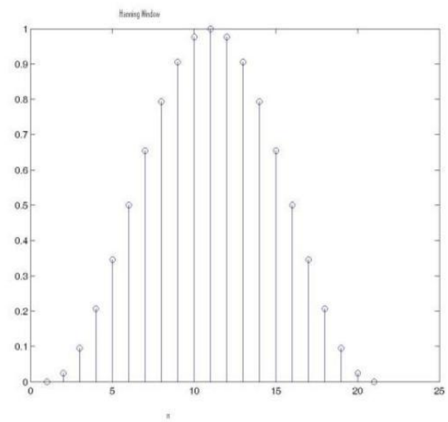
$$w_r(n) = 1 \quad for\ 0 \le n \le M-1$$

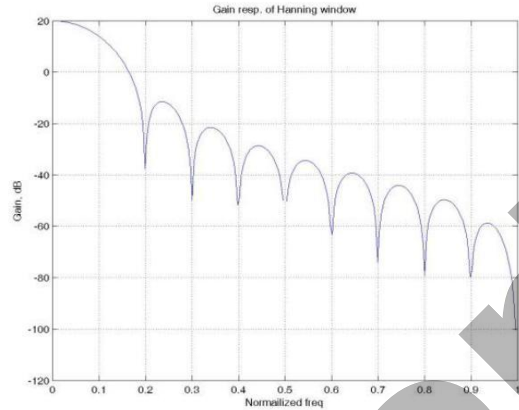


7.8.4 Hanning windows:

It is defined mathematically by,

$$w_{han}(n) = 0.5\left(1 - \cos\frac{2\pi n}{M-1}\right) \quad for\ 0 \le n \le M-1$$

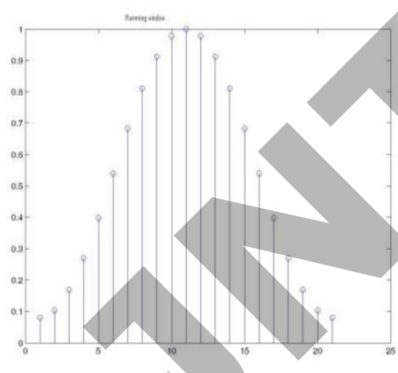


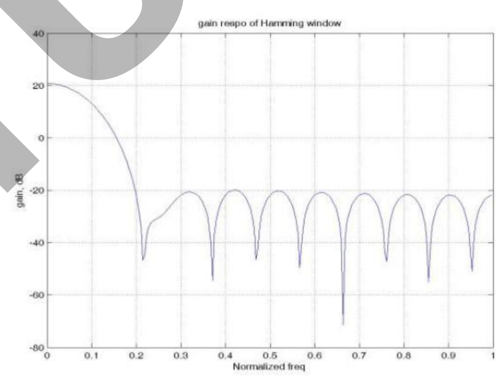


7.8.5 Hamming windows:

This window function is given by,

$$w_{ham}(n) = 0.54 - 0.46 \cos \frac{2\pi n}{M-1} \quad for\ 0 \le n \le M-1$$

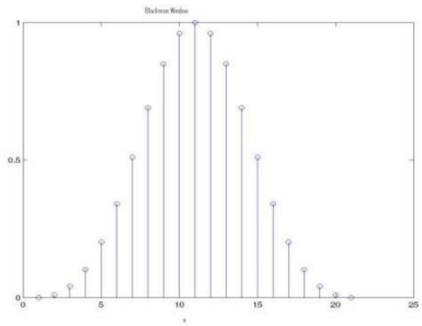


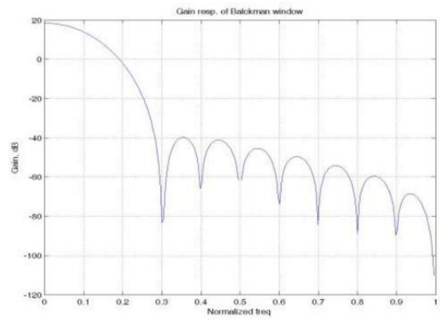


7.8.6 Blackman windows:

This window function is given by,

$$w_{blk}(n) = 0.42 - 0.5 \cos \frac{2\pi n}{M-1} + 0.08 \cos \frac{4\pi n}{M-1} \quad for\ 0 \le n \le M-1$$

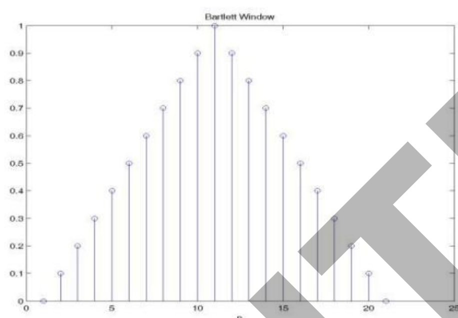


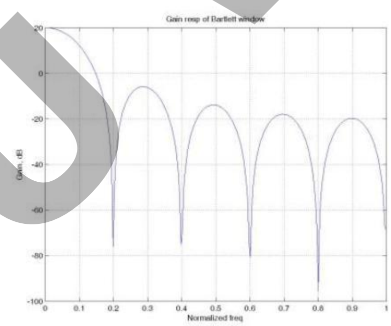


7.8.7 Bartlett (Triangular) windows:

The mathematical description is given by,

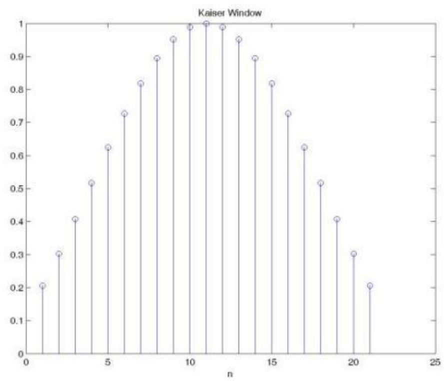
$$w_{bart}(n) = 1 - \frac{2\left|n - \frac{M-1}{2}\right|}{M-1} \quad for\ 0 \le n \le M-1$$

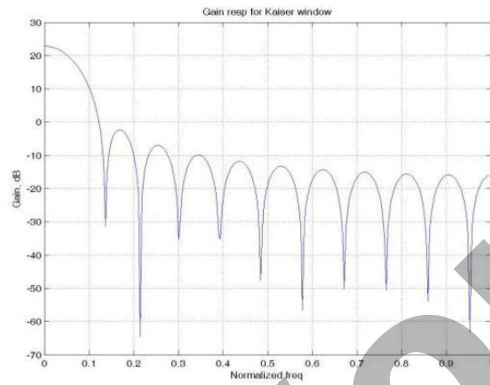




7.8.8 Kaiser windows: The mathematical description is given by,

$$w_k(n) = \frac{I_0\left[\alpha\sqrt{\left(\frac{M-1}{2}\right)^2 - \left(n - \frac{M-1}{2}\right)^2}\right]}{I_0\left[\alpha\left(\frac{M-1}{2}\right)\right]} \quad for\ 0 \le n \le M-1$$





Type of window	Appr. Transition width of the main lobe	Peak sidelobe (dB)
Rectangular	4π/M	-13
Bartlett	8π/M	-27
Hanning	8π/M	-32
Hamming	8π/M	-43
Blackman	12π/M	-58

Looking at the above table we observe filters which are mathematically simple do not offer best characteristics. Among the window functions discussed Kaiser is the most complex one in terms of functional description whereas it is the one which offers maximum flexibility in the design.

7.8.9 Procedure for designing linear-phase FIR filters using windows:

1. Obtain $h_d(n)$ from the desired freq response using inverse FT relation
2. Truncate the infinite length of the impulse response to finite length with

(assuming M odd) choosing proper window

$$h(n) = h_d(n)w(n) \quad where$$
$$w(n) \; is \; the \; window \; function \; defined \;\; for -(M-1)/2 \le n \le (M-1)/2$$

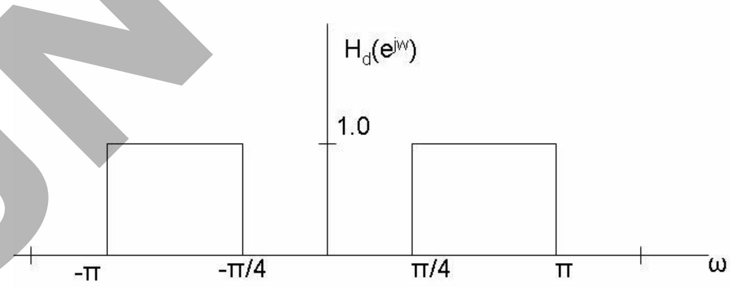
3. Introduce h(n) = h(-n) for linear phase characteristics
4. Write the expression for H(z); this is non-causal realization
5. To obtain causal realization H'(z) = z $^{-(M-1)/2}$ H(z)

Exercise Problems

Prob 1: Design an ideal highpass filter with a frequency response:

$$H_d(e^{j\omega}) = 1 \quad for \; \frac{\pi}{4} \le |\omega| \le \pi$$
$$= 0 \quad |\omega| < \frac{\pi}{4}$$

using a hanning window with M = 11 and plot the frequency response.



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

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

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

$h_d(1) = h_d(-1)=-0.225$
$h_d(2) = h_d(-2)= -0.159$
$h_d(3) = h_d(-3)= -0.075$
$h_d(4) = h_d(-4)= 0$ $h_d(5)$
$= h_d(-5) = 0.045$

The hamming window function is given by

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

$for \quad N = 11$

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

$w_{hn}(0) = 1$
$w_{hn}(1) = w_{hn}(-1)=0.9045$
$w_{hn}(2)= w_{hn}(-2)=0.655$
$w_{hn}(3)= w_{hn}(-3)= 0.345$
$w_{hn}(4)= w_{hn}(-4)=0.0945$
$w_{hn}(5)= w_{hn}(-5)=0$

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

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

$$h'(n) = h(n-5)$$
$$H'(z) = -0.026z^{-2} - 0.104z^{-3} - 0.204z^{-4} + 0.75z^{-5} - 0.204z^{-6} - 0.104z^{-7} - 0.026z^{-8}$$

Using the equation

$$H_r(e^{jw}) = [h(\frac{M-1}{2}) + 2\sum_{n=0}^{\frac{M-3}{2}} h(n)\cos\omega(\frac{M-1}{2} - n)$$

$$H_r(e^{jw}) = 0.75) + 2\sum_{n=0}^{4} h(n)\cos\omega(5-n)$$

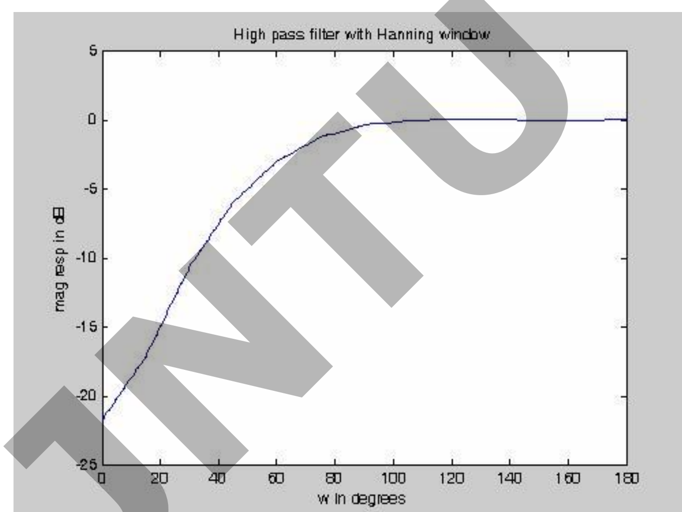
The magnitude response is given by,

$|Hr(e^{j\omega})| = |0.75 - 0.408\cos\omega - 0.208\cos 2\omega - 0.052\cos 3\omega|$

ω in degrees = [0 15 30 45 60 75 90 105 120 135 150 165 180]

$|H(e^{j\omega})|$ in dBs = [-21.72 -17.14 -10.67 -6.05 -3.07 -1.297 -0.3726

-0.0087 0.052 0.015 0 0 0.017]



Prob 2 : Design a filter with a frequency response:

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

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

using a Hanning window with M = 7

Soln:

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 gives $h_d(0) = h_d(6) = 0.075$

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

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

$h_d(3) = 0.25$

The Hanning window function values are given by

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

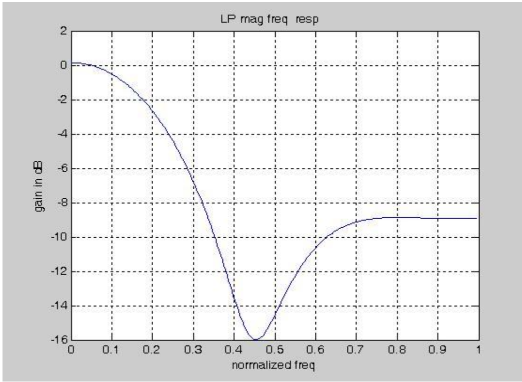
$w_{hn}(1) = w_{hn}(5) = 0.25$

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

$w_{hn}(3) = 1$

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

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



7.9 Design of Linear Phase FIR filters using Frequency Sampling method

7.9.1 Motivation: We know that DFT of a finite duration DT sequence is obtained by sampling FT of the sequence then DFT samples can be used in reconstructing original time domain samples if frequency domain sampling was done correctly. The samples of FT of h(n) i.e., H(k) are sufficient to recover h(n).

Since the designed filter has to be realizable then h(n) has to be real, hence even symmetry properties for mag response |H(k)| and odd symmetry properties for phase response can be applied. Also, symmetry for h(n) is applied to obtain linear phase chas.

Fro DFT relationship we have

$$h(n) = \frac{1}{N}\sum_{k=0}^{N-1} H(k) e^{j2\pi kn/N} \quad for \quad n = 0,1,......N-1$$

$$H(k) = \sum_{n=0}^{N-1} h(n) e^{-j2\pi kn/N} \quad for \quad k = 0,1,.........N-1$$

Also we know $H(k) = H(z)|_{z=e^{j2\pi kn/N}}$

The system function H(z) is given by

$$H(z) = \sum_{n=0}^{N-1} h(n) z^{-n}$$

Substituting for h(n) from IDFT relationship

$$H(z) = \frac{1 - z^{-N}}{N} \sum_{k=0}^{N-1} \frac{H(k)}{1 - e^{j2\pi kn/N} z^{-1}}$$

Since H(k) is obtained by sampling $H(e^{j\omega})$ hence the method is called Frequency Sampling Technique.

Since the impulse response samples or coefficients of the filter has to be real for filter to be realizable with simple arithmetic operations, properties of DFT of real sequence can be used. The following properties of DFT for real sequences are useful:

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

$|H(k)|=|H(N-k)|$ - magnitude response is even

$\theta(k) = -\theta(N-k)$ – Phase response is odd

$$h(n) = \frac{1}{N}\sum_{k=0}^{N-1} H(k) e^{j2\pi kn/N}$$ can be rewritten as (for N odd)

$$h(n) = \frac{1}{N}\left[H(0) + \sum_{k=1}^{N-1} H(k) e^{j2\pi kn/N} \right]$$

$$h(n) = \frac{1}{N}\left[H(0) + \sum_{k=1}^{N-1/2} H(k) e^{j2\pi kn/N} + \sum_{k=N-1/2}^{N-1} H(k) e^{j2\pi kn/N} \right]$$

Using substitution k = N – r or r = N- k in the second substitution with r going from now (N- 1)/2 to 1 as k goes from 1 to (N-1)/2

$$h(n) = \frac{1}{N}\left[H(0) + \sum_{k=1}^{(N-1)/2} H(k) e^{j2\pi kn/N} + \sum_{k=1}^{(N-1)/2} H(N-k) e^{-j2\pi kn/N} \right]$$

$$h(n) = \frac{1}{N}\left[H(0) + \sum_{k=1}^{(N-1)/2} H(k) e^{j2\pi kn/N} + \sum_{k=1}^{(N-1)/2} H^*(k) e^{-j2\pi kn/N} \right]$$

$$h(n) = \frac{1}{N}\left[H(0) + \sum_{k=1}^{(N-1)/2} H(k) e^{j2\pi kn/N} + \sum_{k=1}^{(N-1)/2} (H(k) e^{j2\pi kn/N})^* \right]$$

$$h(n) = \frac{1}{N}\left[H(0) + \sum_{k=1}^{(N-1)/2} (H(k) e^{j2\pi kn/N} + (H(k) e^{j2\pi kn/N})^* \right]$$

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

Similarly for N even we have

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

Using the symmetry property h(n)= h (N-1-n) we can obtain Linear phase FIR filters using the frequency sampling technique.

Exercise problems

Prob 1 : 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.

The desired response can be expressed as

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

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

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

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

$$H(k) = e^{-j\frac{2\pi k}{17}8} \quad for \quad 0 \le \frac{2\pi k}{17} \le \frac{\pi}{2}$$
$$= 0 \quad for \quad \pi/2 \le \frac{2\pi k}{17} \le \pi$$
$$H(k) = e^{-j\frac{16\pi k}{17}} \quad for \quad 0 \le k \le \frac{17}{4}$$
$$= 0 \quad for \quad \frac{17}{4} \le k \le \frac{17}{2}$$

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

$$0 \le k \le 4$$
$$and \quad 5 \le k \le 8$$

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

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

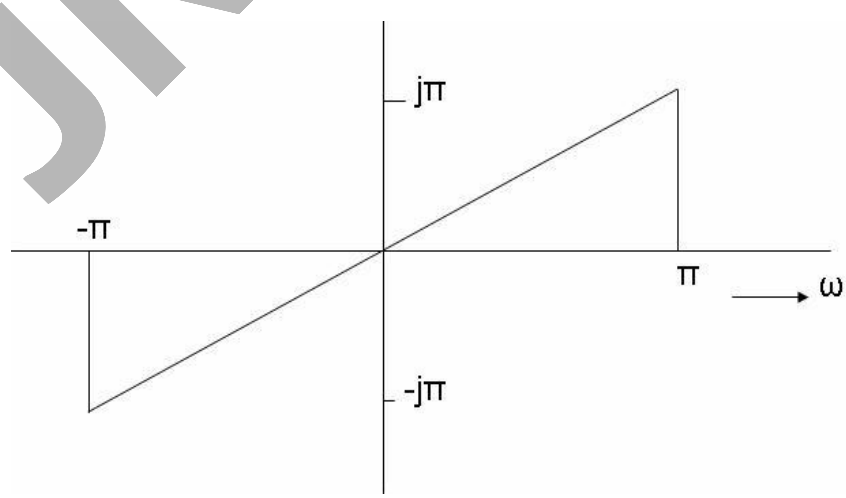
i.e., $$h(n) = \frac{1}{17}(1 + 2\sum_{k=1}^{4} \mathrm{Re}(e^{-j16\pi k/17} e^{j2\pi kn/17}))$$

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

- 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)

7.10 Design of FIR Differentiator

Differentiators are widely used in Digital and Analog systems whenever a derivative of the signal is needed. Ideal differentiator has pure linear magnitude response in the freq range –π to +π. The typical frequency response characteristics is as shown in the below figure.



Problem 2: Design an Ideal Differentiator using a) rectangular window and b)Hamming window with length of the system = 7.

Solution:
As seen from differentiator frequency chars. It is defined as

$H(e^{j\omega}) = j\omega$ between $-\pi$ to $+\pi$

$$h_d(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} j\omega\, e^{j\omega n}\, d\omega = \frac{\cos \pi n}{n} \qquad -\infty \le n \le \infty \quad and \quad n \neq 0$$

The $h_d(n)$ is an add function with $h_d(n)=-h_d(-n)$ and $h_d(0)=0$

a) rectangular window

$h(n)=h_d(n)w_r(n)$

h(1)=-h(-1)=hd(1)=-1
h(2)=-h(-2)=hd(2)=0.5
h(3)=-h(-3)=hd(3)=-0.33

h'(n)=h(n-3) for causal
system thus,

$$H'(z) = 0.33 - 0.5z^{-1} + z^{-2} - z^{-4} + 0.5z^{-5} - 0.33z^{-6}$$

Also from the equation

$$H_r(e^{j\omega}) = 2\sum_{n=0}^{(M-3)/2} h(n)\sin\omega\left(\frac{M-1}{2} - n\right)$$

For M=7 and h'(n) as found above we obtain this as

$$H_r(e^{j\omega}) = 0.66\sin 3\omega - \sin 2\omega + 2\sin\omega$$

$$H(e^{j\omega}) = jH_r(e^{j\omega}) = j(0.66\sin 3\omega - \sin 2\omega + 2\sin\omega)$$

b) Hamming window
$h(n)=h_d(n)w_h(n)$

where $w_h(n)$ is given by

$$w_h(n) = 0.54 + 0.46 \cos \frac{2\pi n}{(M-1)} \quad -(M-1)/2 \le n \le (M-1)/2$$
$$= 0 \quad otherwise$$

For the present problem

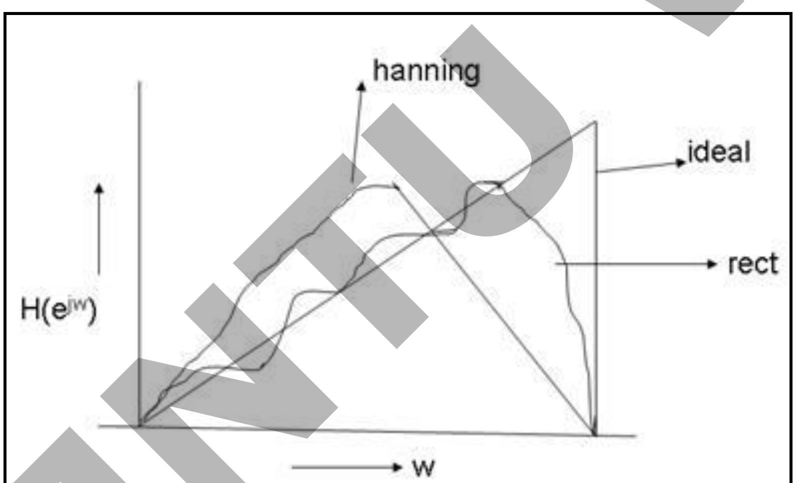
$$w_h(n) = 0.54 + 0.46 \cos \frac{\pi n}{3} \quad -3 \le n \le 3$$

The window function coefficients are given by for n=-3 to +3
Wh(n)= [0.08 0.31 0.77 1 0.77 0.31 0.08]

Thus h'(n) = h(n-5) = [0.0267, -0.155, 0.77, 0, -0.77, 0.155, -0.0267]

Similar to the earlier case of rectangular window we can write the freq response of differentiator as

$$H(e^{j\omega}) = jH_r(e^{j\omega}) = j(0.0534 \sin 3\omega - 0.31\sin 2\omega + 1.54 \sin \omega)$$



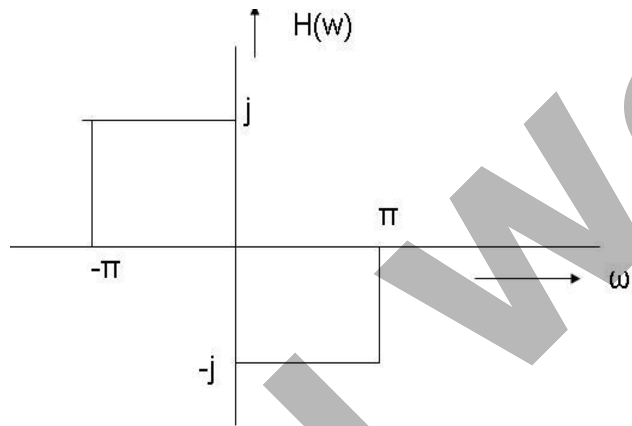
We observe

- With rectangular window, the effect of ripple is more and transition band width is small compared with hamming window
- With hamming window, effect of ripple is less whereas transition band is more

7.11 Design of FIR Hilbert transformer:

Hilbert transformers are used to obtain phase shift of 90 degree. They are also called j operators. They are typically required in quadrature signal processing. The Hilbert transformer is very useful when out of phase component (or imaginary part) need to be generated from available real component of the signal.

Problem 3: Design an ideal Hilbert transformer using a) rectangular window and b) Blackman Window with M = 11



Solution:

As seen from freq chars it is defined as

$$H_d(e^{j\omega}) = j \qquad -\pi \le \omega \le 0$$
$$= -j \qquad 0 \le \omega \le \pi$$

The impulse response is given by

$$h_d(n) = \frac{1}{2\pi}\left[\int_{-\pi}^{0} je^{j\omega n} d\omega + \int_{0}^{\pi} -je^{j\omega n} d\omega\right] = \frac{(1-\cos \pi n)}{\pi n} \quad -\infty \le n \le \infty \quad except \quad n = 0$$

At n = 0 it is hd(0) = 0 and hd(n) is an odd function

a) Rectangular window

$h(n) = h_d(n)\, w_r(n) = h_d(n)$ for $-5 \geq n \geq 5$

h’(n)=h(n-5)

h(n)= [-0.127, 0, -0.212, 0, -0.636, 0, 0.636, 0, 0.212, 0, 0.127]

$$H_r(e^{j\omega}) = 2\sum_{n=0}^{4} h(n)\sin\omega(5-n)$$

$$H(e^{j\omega}) = j\,|H_r(e^{j\omega})| = j\{0.254\sin 5\omega + 0.424\sin 3\omega + 1.272\sin\omega\}$$

b) Blackman Window

window function is defined as

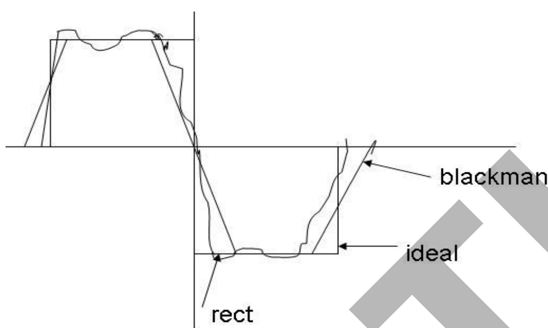
$$w_b(n) = 0.42 + 0.5\cos\frac{\pi n}{5} + 0.08\cos\frac{2\pi n}{5} \qquad -5 \le n \le 5$$

$$= 0 \qquad otherwise$$

$W_b(n)$ = [0, 0.04, 0.2, 0.509,0.849,1,0.849, 0.509, 0.2, 0.04,0] for -5≥n≥5

h’(n) = h(n-5) = [0, 0, -0.0424, 0, -0.5405, 0, 0.5405, 0, 0.0424, 0, 0]

$$H(e^{j\omega}) = -j[0.0848\sin 3\omega + 1.0810\sin\omega]$$



Recommended questions with solution

Question1

Design an FIR linear phase, digital filter approximating the ideal frequency response

$$H_d(\omega) = \begin{cases} 1, & \text{for } |\omega| \le \frac{\pi}{6} \\ 0, & \text{for } \frac{\pi}{6} < |\omega| \le \pi \end{cases}$$

(a) Determine the coefficients of a 25-tap filter based on the window method with a rectangular window.

(b) Determine and plot the magnitude and phase response of the filter.

(c) Repeat parts (a) and (b) using the Hamming window.

(d) Repeat parts (a) and (b) using a Bartlett window.

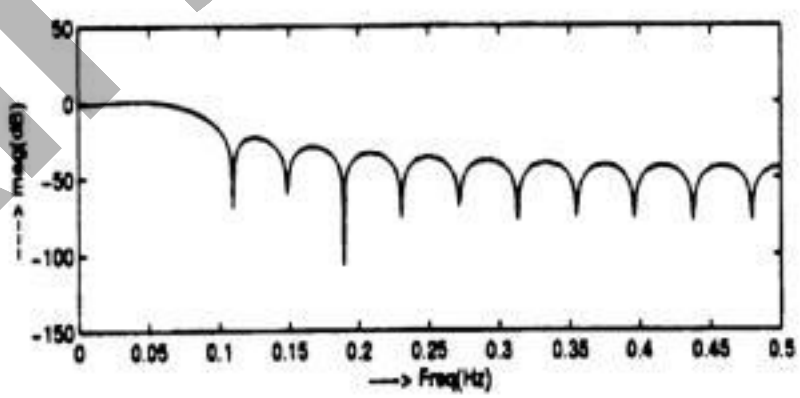
Solution:-

(a) To obtain the desired length of 25, a delay of $\frac{25-1}{2} = 12$ is incorporated into $H_d(w)$. Hence,

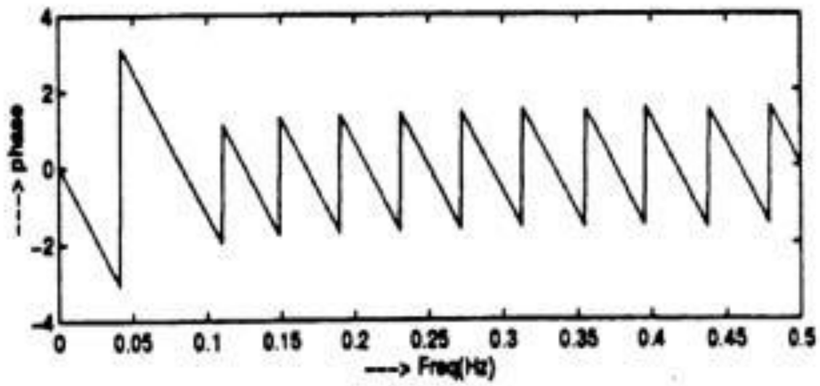
$$\begin{aligned} H_d(w) &= 1e^{-j12w}, \quad 0 \le |w| \le \frac{\pi}{6} \\ &= 0, \quad \text{otherwise} \\ h_d(n) &= \frac{1}{2\pi}\int_{-\frac{\pi}{6}}^{\frac{\pi}{6}} H_d(w)e^{-jwn}dw \\ &= \frac{\sin\frac{\pi}{6}(n-12)}{\pi(n-12)} \\ \text{Then, } h(n) &= h_d(n)w(n) \end{aligned}$$

where $w(n)$ is a rectangular window of length $N = 25$.

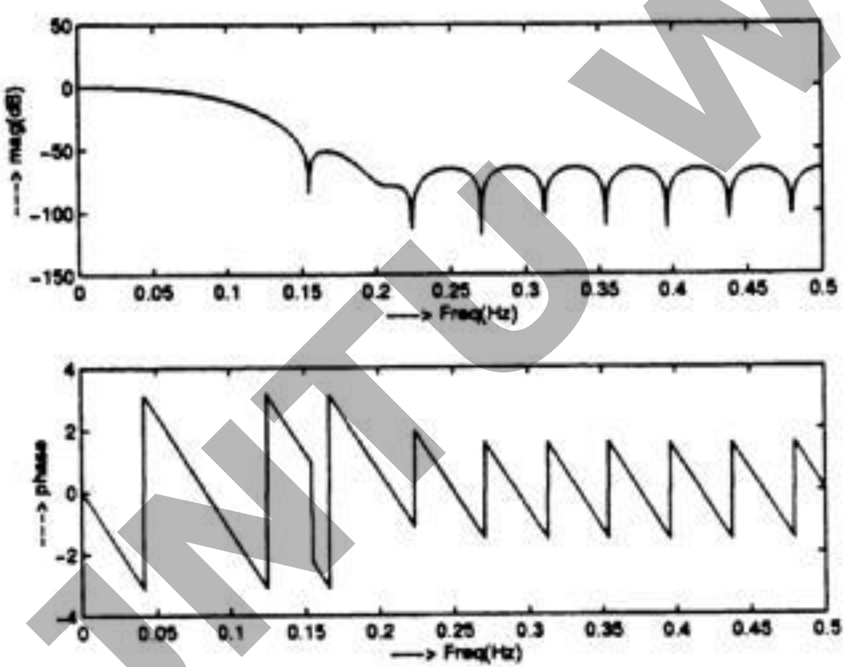
(b) Magnitude plot



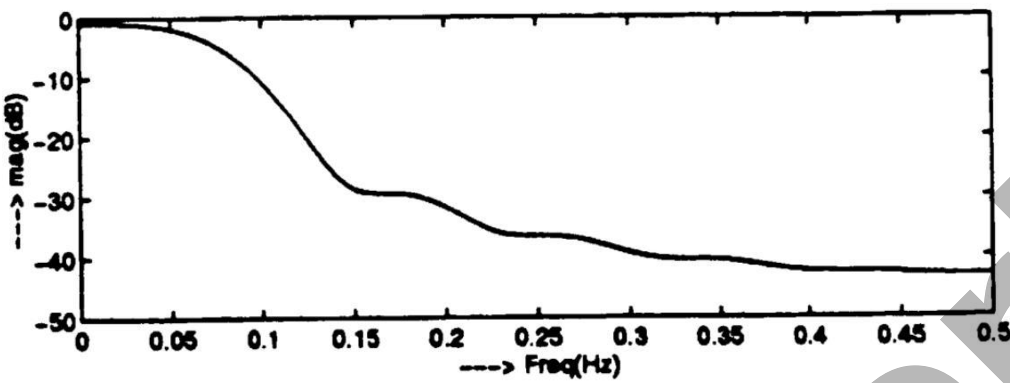
Phase plot

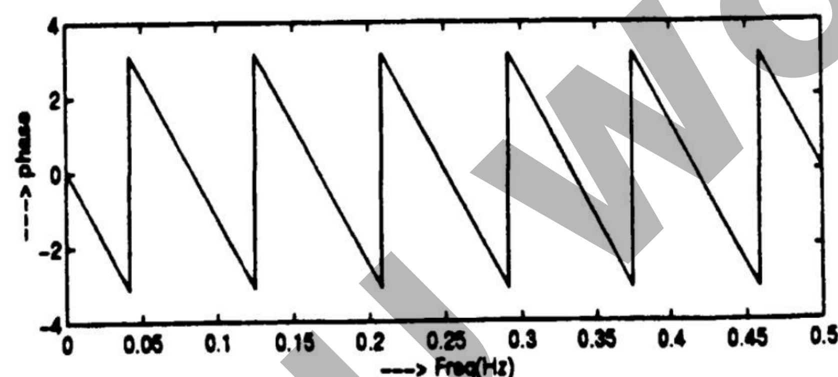


(c) Hamming window



(d) Bartlett window





Question 2

Determine the unit sample response $\{h(n)\}$ of a linear-phase FIR filter of length $M = 4$ for which the frequency response at $\omega = 0$ and $\omega = \pi/2$ is specified as

$$H_r(0) = 1 \qquad H_r\left(\frac{\pi}{2}\right) = \frac{1}{2}$$

Solution:-

$$M = 4, \quad H_r(0) = 1, \quad H_r\left(\frac{\pi}{2}\right) = \frac{1}{2}$$

$$H_r(w) = 2\sum_{n=0}^{\frac{M}{2}-1} h(n)cos[w(\frac{M-1}{2} - n)]$$

$$= 2\sum_{n=0}^{1} h(n)cos[w(\frac{3}{2} - n)]$$

$$\text{At } w = 0, H_r(0) = 1 = 2\sum_{n=0}^{1} h(n)cos[0]$$

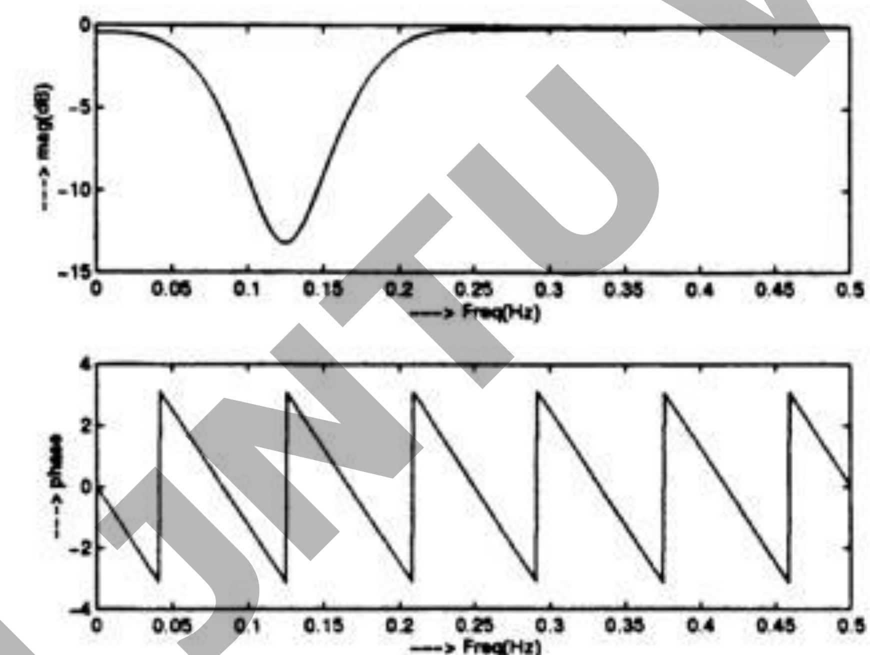
$$2[h(0) + h(1)] = 1 \qquad (1)$$

$$\text{At } w = \frac{\pi}{2}, H_r(\frac{\pi}{2}) = \frac{1}{2} \quad = \quad 2\sum_{n=0}^{1} h(n)cos[\frac{\pi}{2}(\frac{3}{2} - n)]$$

$$-h(0) + h(1) \quad = \quad 0.354 \qquad (2)$$

Solving (1) and (2), we get

$$h(0) = 0.073 \text{ and}$$
$$h(1) = 0.427$$
$$h(2) = h(1)$$
$$h(3) = h(0)$$
$$\text{Hence, } h(n) = \{0.073, 0.427, 0.427, 0.073\}$$



Question 3

Use the window method with a Hamming window to design a 21-tap differentiator as shown in Fig. P8.9. Compute and plot the magnitude and phase response of the resulting filter.

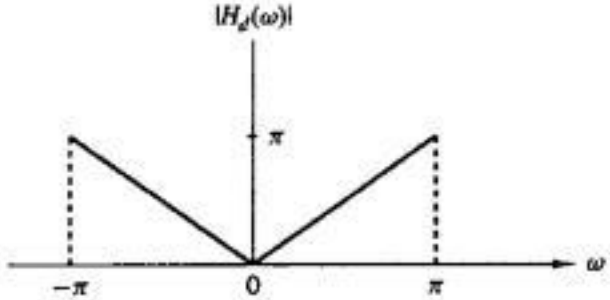
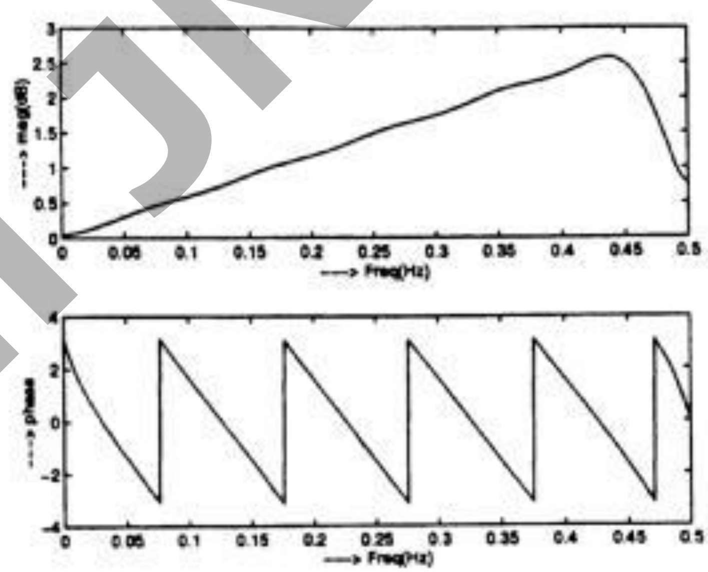


Figure P8.9

Solution:-

$$
\begin{aligned}
H_d(w) &= we^{-j10w}, \quad 0 \leq w \leq \pi \\
&= -we^{-j10w}, \quad -\pi \leq w \leq 0 \\
h_d(n) &= \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d(w)e^{-jwn}dw \\
&= \frac{\cos\pi(n-10)}{(n-10)}, \quad n \neq 10 \\
&= 0, \quad n = 10 \\
h_d(n) &= \frac{\cos\pi(n-10)}{(n-10)}, \quad 0 \leq n \leq 20, n \neq 10 \\
&= 0, \quad n = 10
\end{aligned}
$$

Magnitude and phase response



Question 4

A digital low-pass filter is required to meet the following specifications:

Passband ripple: ≤ 1 dB
Passband edge: 4 kHz
Stopband attenuation: ≥ 40 dB
Stopband edge: 6 kHz
Sample rate: 24 kHz

The filter is to be designed by performing a bilinear transformation on an analog system function. Determine what order Butterworth, Chebyshev, and elliptic analog designs must be used to meet the specifications in the digital implementation.

Solution:-

From the design specifications we obtain

$$\epsilon = 0.509$$
$$\delta = 99.995$$
$$f_p = \frac{4}{24} = \frac{1}{6}$$
$$f_s = \frac{6}{24} = \frac{1}{4}$$

Assume $t = 1$. Then,
$$\Omega_p = 2tan\frac{w_p}{2}$$
$$= 2tan\pi f_p = 1.155$$

and
$$\Omega_s = 2tan\frac{w_s}{2}$$
$$= 2tan\pi f_s = 2$$
$$\eta = \frac{\delta}{\epsilon} = 196.5$$

304

UNIT - 5

Multi-rate Digital Signal Processing

10.1 INTRODUCTION

Discrete-time systems may be single-rate systems or multi-rate systems. The discrete-time systems that use single sampling rate from A/D converter to D/A converter are known as single-rate systems and the discrete-time systems that process data at more than one sampling rate are known as multi-rate systems. In digital audio, the different sampling rates used are 32 kHz for broadcasting, 44.1 kHz for compact disc and 48 kHz for audio tape. In digital video, the sampling rates for composite video signals are 14.3181818 MHz and 17.734475 MHz for NTSC and PAL respectively. But the sampling rates for digital component of video signals are 13.5 MHz and 6.75 MHz for luminance and colour difference signal. Different sampling rates can be obtained using an up sampler and down sampler. The basic operations in multirate processing to achieve this are decimation and interpolation. Decimation is for reducing the sampling rate and interpolation is for increasing the sampling rate. There are many cases where multi-rate signal processing is used. Few of them are as follows:

1. In high quality data acquisition and storage systems
2. In audio signal processing
3. In video
4. In speech processing
5. In transmultiplexers
6. For narrow band filtering

The various advantages of multirate signal processing are as follows:

1. Computational requirements are less.
2. Storage space for filter coefficients is less.
3. Finite arithmetic effects are less.
4. Filter order required in multirate application is low.
5. Sensitivity to filter coefficient lengths is less.

While designing multi-rate systems, effects of aliasing for decimation and pseudoimages for interpolators should be avoided.

10.2 SAMPLING

A continuous-time signal $x(t)$ can be converted into a discrete-time signal $x(nT)$ by sampling it at regular intervals of time with sampling period T. The sampled signal $x(nT)$ is given by

$$x(nT) = x(t)\big|_{t=nT}, \quad -\infty < n < \infty$$

A sampling process can also be interpreted as a modulation or multiplication process.

Sampling Theorem

Sampling theorem states that a band limited signal $x(t)$ having finite energy, which has no spectral components higher than f_h hertz can be completely reconstructed from its samples taken at the rate of $2f_h$ or more samples per second.

The sampling rate of $2f_h$ samples per second is the Nyquist rate and its reciprocal $1/2f_h$ is the Nyquist period.

10.3 DOWN SAMPLING

Reducing the sampling rate of a discrete-time signal is called down sampling. The sampling rate of the discrete-time signal can be reduced by a factor D by taking every Dth value of the signal. Mathematically, down sampling is represented by

$$y(n) = x(Dn)$$

and the symbol for the down sampler is shown in Figure 10.1.

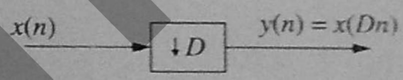


Figure 10.1 A down sampler.

If $\quad x(n) = \{1, 2, 3, 1, 2, 3, 1, 2, 3, ...\}$

Then, $\quad x(2n) = \{1, 3, 2, 1, 3, ...\}$

and $\quad x(3n) = \{1, 1, 1, 1, ...\}$

$x(2n)$ is obtained by keeping every second sample of $x(n)$ and $x(3n)$ is obtained by keeping every 3rd sample of $x(n)$ and removing other samples.

If the input signal $x(n)$ is not band limited, then there will be overlapping of spectra at the output of the down sampler. This overlapping of spectra is called aliasing which is undesirable. This aliasing problem can be eliminated by band limiting the input signal by inserting a low-pass filter called anti-aliasing filter before the down sampler. The anti-aliasing filter and the down sampler together is called decimator. The decimator is also known as sub sampler, down sampler or under sampler. Decimation (sampling rate compression) is the process of decreasing the sampling rate by an integer factor D by keeping every Dth sample and removing $D-1$ in between samples.

Figure 10.2 shows the signal $x(n)$ and its down sampled versions by a factor of 2 and 3.

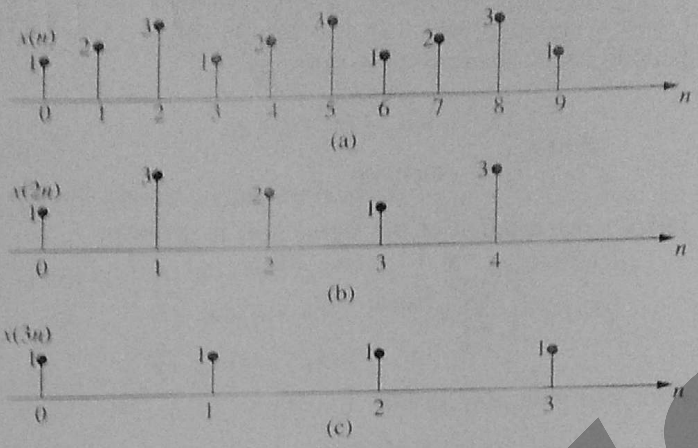


Figure 10.2 Plots of (a) $x(n)$, (b) $x(2n)$ and (c) $x(3n)$.

The block diagram of the decimator is shown in Figure 10.3. The decimator comprises two blocks such as anti-aliasing filter and down sampler. Here the anti-aliasing filter is a low-pass filter to band limit the input signal so that aliasing problem is eliminated and the down sampler is used to reduce the sampling rate by keeping every Dth sample and removing $D-1$ in between samples.

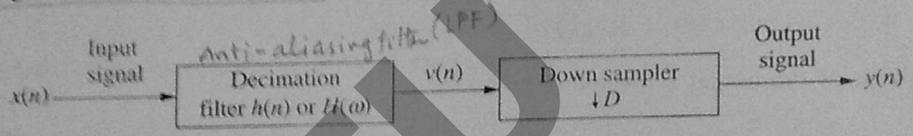


Figure 10.3 Block diagram of decimator.

Spectrum of down sampled signal

Let T be sampling period of input signal $x(n)$, and let F be its sampling rate or frequency. When the signal is down sampled by D, let T' be its new sampling period and F' be its sampling frequency, then

$$\frac{T'}{T} = D$$

$$F' = \frac{1}{T'} = \frac{1}{TD} = \frac{F}{D}$$

Let us derive the spectrum of a down sampled signal $x(Dn)$ and compare it with the spectrum of input signal $x(n)$. The Z-transform of the signal $x(n)$ is given by

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

The down sampled signal $y(n)$ is obtained by multiplying the sequence $x(n)$ with a periodic train of impulses $p(n)$ with a period D and then leaving out the $D-1$ zeros between each pair of samples. The periodic train of impulses is given by

$$p(n) = \begin{cases} 1, & n = 0, \pm D, \pm 2D, \dots \\ 0, & \text{otherwise} \end{cases}$$

The discrete Fourier series representation of the signal $p(n)$ is given by

$$p(n) = \frac{1}{D} \sum_{k=0}^{D-1} e^{j2\pi kn/D}, \quad -\infty < n < \infty$$

Multiplying the sequence $x(n)$ with $p(n)$ yields

$$x'(n) = x(n)p(n)$$

That is

$$x'(n) = \begin{cases} x(n), & n = 0, \pm D, \pm 2D, \dots \\ 0, & \text{otherwise} \end{cases}$$

If we leave $D-1$ zeros between each pair of samples, we get the output of down sampler

$$y(n) = x'(nD) = x(nD)\, p(nD)$$
$$= x(nD)$$

The Z-transform of the output sequence is given by

$$Y(z) = \sum_{n=-\infty}^{\infty} y(n) z^{-n}$$

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

$$= \sum_{n=-\infty}^{\infty} x'(n) z^{-n/D}$$

where $x'(n) = 0$ except at multiple of D. Since $x'(n) = x(n)\, p(n)$, we get

$$Y(z) = \sum_{n=-\infty}^{\infty} x(n)\, p(n) z^{-n/D}$$

$$= \sum_{n=-\infty}^{\infty} x(n) \left[\frac{1}{D} \sum_{k=0}^{D-1} e^{j2\pi kn/D} \right] z^{-n/D}$$

$$= \frac{1}{D} \sum_{k=0}^{D-1} \sum_{n=-\infty}^{\infty} x(n) \left(e^{-j2\pi k/D} z^{1/D}\right)^{-n}$$

$$= \frac{1}{D} \sum_{k=0}^{D-1} X[e^{-j2\pi k/D} z^{1/D}]$$

Substituting $z = e^{j\omega}$, we get the frequency response

$$Y(\omega) = \frac{1}{D} \sum_{k=0}^{D-1} X(e^{-j2\pi k/D} e^{j\omega/D}) = \frac{1}{D} \sum_{k=0}^{D-1} X(e^{j(\omega - 2\pi k)/D})$$

i.e.,
$$Y(\omega) = \frac{1}{D} \sum_{k=0}^{D-1} X\left[\frac{(\omega - 2\pi k)}{D}\right]$$

From the above relation we find that if the Fourier transform of the input signal $x(n)$ of a down sampler is $X(\omega)$, then the Fourier transform $Y(\omega)$ of the output signal $y(n)$ is a sum of D uniformly shifted and stretched versions of $X(\omega)$ scaled by a factor $1/D$.

If the spectrum of the original signal $X(\omega)$ is band limited to $\omega = \pi/D$, as shown in Figure 10.4(a), the spectrum being periodic with period 2π, the spectrum of the down sampled signal $Y(\omega)$ is the sum of all the uniformly shifted and stretched versions of $X(\omega)$ scaled by a factor $1/D$ as shown in Figure 10.4(b). In every interval of 2π in addition to the original spectrum we find $D-1$ equally spaced replica. In Figure 10.4(b), the frequency variable ω_x is related to the original sampling rate. In Figure 10.4(c), the frequency variable ω_y is normalized with respect to reduced sampling rate.

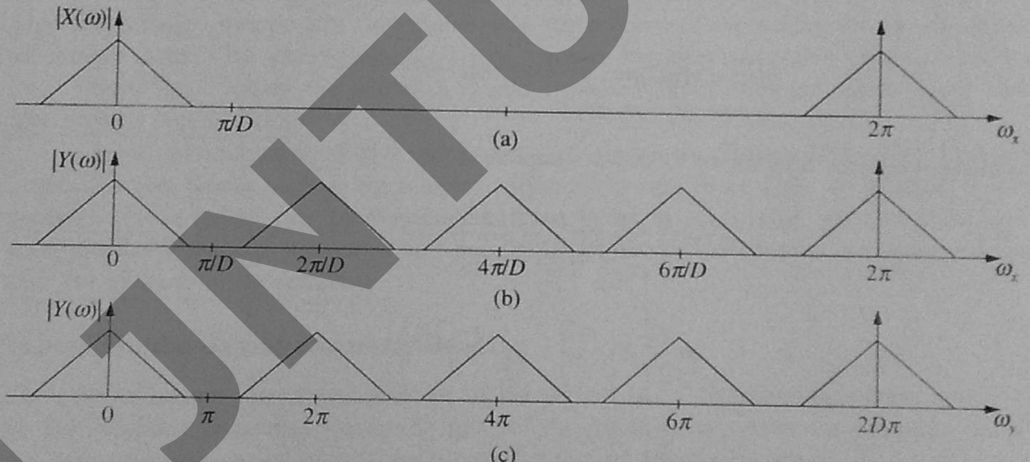


Figure 10.4 Spectrum of (a) input, (b) output, and (c) normalized output.

Aliasing effect and Anti-aliasing filter

From Figure 10.5, we can find that the spectrum obtained after down sampling will overlap if the original spectrum is not band limited to $\omega = \pi/D$. This overlapping of spectra is called aliasing. Therefore, aliasing due to down sampling a signal by a factor of D is absent if and only if the signal $x(n)$ is band limited to $\pm\pi/D$. If the signal $x(n)$ is not band limited to $\pm\pi/D$, then a low-pass filter with a cutoff frequency π/D is used prior to down sampling. This low-pass filter which is connected before the down sampler to prevent the effect of aliasing by band limiting the input signal is called the anti-aliasing filter.

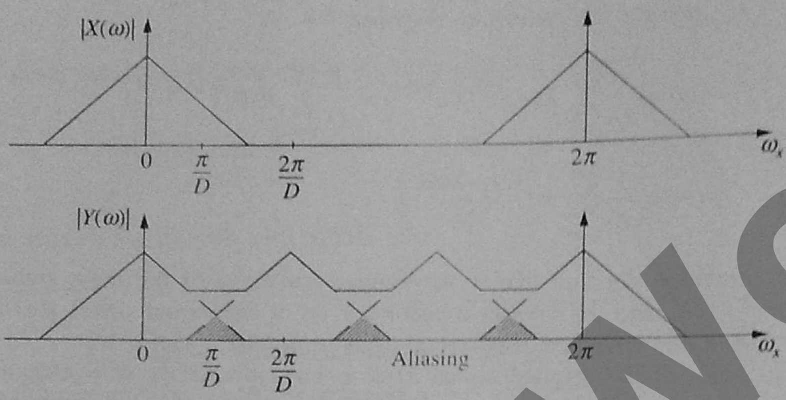


Figure 10.5 (a) Input spectrum, (b) aliased output spectrum.

The signal obtained after filtering is given by

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

and

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

For example, consider a factor of D down sampler, then (D = 2)

$$Y(\omega) = \frac{1}{2}\sum_{k=0}^{1} X\left(\frac{\omega - 2\pi k}{2}\right)$$

$$= \frac{1}{2}\left[X\left(\frac{\omega}{2}\right) + X\left(\frac{\omega - 2\pi}{2}\right)\right]$$

$$= \frac{1}{2}\left[X\left(\frac{\omega}{2}\right) + X\left(-\frac{\omega}{2}\right)\right]$$

The second term $X(-\omega/2)$ is simply obtained by shifting the first term $X(\omega)$ to the right by an amount of 2π.

10.4 UP SAMPLING

Increasing the sampling rate of a discrete-time signal is called up sampling. The sampling rate of a discrete-time signal can be increased by a factor I by placing $I-1$ equally spaced zeros between each pair of samples.

Mathematically, up sampling is represented by

$$y(n) = \begin{cases} x\left(\dfrac{n}{I}\right), & n = 0, \pm I, \pm 2I, \ldots \\ 0, & \text{otherwise} \end{cases}$$

and the symbol for up sampler is shown in Figure 10.6.

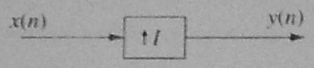


Figure 10.6 Up sampler.

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

Then, $y(n) = x\left(\dfrac{n}{2}\right)$ $\{1, 0, 2, 0, 3, 0, 1, 0, 2, 0, 3, 0, \ldots\}$ for an up-sampling factor of $I = 2$.

and $y(n) = x\left(\dfrac{n}{3}\right)$ $\{1, 0, 0, 2, 0, 0, 3, 0, 0, 1, \ldots\}$ for an up-sampling factor of $I = 3$.

Usually an anti-imaging filter is to be kept after the up sampler to remove the unwanted images developed due to up sampling. The anti-imaging filter and the up sampler together is called interpolator. Interpolation is the process of increasing the sampling rate by an integer factor I by interpolating $I-1$ new samples between successive values of the signal.

Figure 10.7 shows the signal $x(n)$ and its two-fold up-sampled signal $y_1(n)$ and the interpolated signal $y_2(n)$.

The block diagram of the interpolator is shown in Figure 10.8. The interpolator comprises two blocks such as up sampler and anti-imaging filter. Here up sampler is used to increase the sampling rate by introducing zeros between successive input samples and the interpolation filter, also known as anti-imaging filter, is used to remove the unwanted images that are yielded by up sampling.

Expression for output of interpolator

Let I be an integer interpolating factor of the signal. Let T be sampling period and $F = 1/T$ be the sampling frequency (sampling rate) of the input signal. After up sampling, let T' be the new sampling period and F' be the new sampling frequency, then

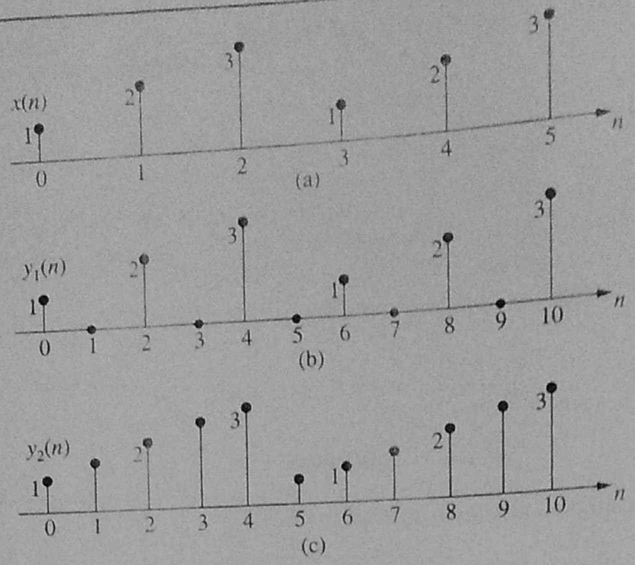


Figure 10.7 (a) Input signal $x(n)$, (b) Output of 2 fold up sampler $y_1(n) = x(n/2)$, (c) Output of interpolator $y_2(n) = x(n/2)$.

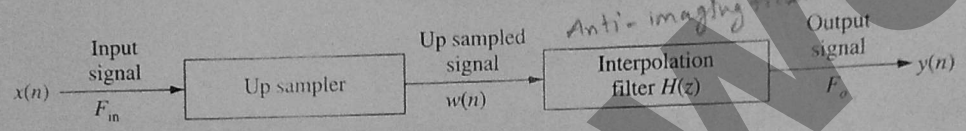


Figure 10.8 Block diagram of an interpolator.

$$\frac{T'}{T} = \frac{1}{I} \qquad T' = \frac{T}{I}$$

The sampling rate is given by

$$F' = \frac{1}{T'} = \frac{I}{T} = IF$$

Let $w(n)$ be the signal obtained by interpolating $I - 1$ samples between each pair of samples of $x(n)$.

$$w(n) = \begin{cases} x\left(\dfrac{n}{I}\right), & n = 0, \pm I, \pm 2I, \dots \\ 0, & \text{otherwise} \end{cases}$$

The Z-transform of the signal $w(n)$ is given by

$$W(z) = \sum_{n=-\infty}^{\infty} w(n) z^{-n} = \sum_{n=-\infty}^{\infty} x\left(\frac{n}{I}\right) z^{-n}$$

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

$$= X(z^I)$$

When considered over the unit circle $z = e^{j\omega'}$.

$$W(e^{j\omega'}) = X(e^{j\omega' I}), \text{ i.e. } W(\omega') = X(I\omega')$$

where $\omega' = 2\pi f T'$. The spectra of the signal $w(n)$ contains the images of base band placed at the harmonics of the sampling frequency $\pm 2\pi/I$, $\pm 4\pi/I$. To remove the images an anti-imaging filter is used. The ideal characteristics of low-pass filter is given by

$$H(e^{j\omega'}) = \begin{cases} G, & |\omega'| \le 2\pi f T'/2 = \pi/I \\ 0, & \text{otherwise} \end{cases}$$

where G is the gain of the filter and it should be I in the pass band. The frequency response of the output signal is given by

$$Y(e^{j\omega'}) = H(e^{j\omega'}) X(e^{j\omega' I})$$

$$= \begin{cases} GX(e^{j\omega' I}), & |\omega'| \le \pi/I \\ 0 \quad, & \text{otherwise} \end{cases}$$

The output signal $y(n)$ is given by

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

$$= \sum_{k=-\infty}^{\infty} h(n-k)\, x(k/I), \quad k/I \text{ is an integer}$$

Figure 10.9(a) shows the spectrum $X(\omega)$. The spectrum $X(I\omega)$ is sketched for $I = 3$ in Figure 10.9(b). Note that the frequency spectrum $X(3\omega)$ is three-fold repetition of $X(\omega)$. That is, inserting $I - 1$ zeros between successive values of $x(n)$ results in a signal whose spectrum $X(I\omega)$ is an I fold periodic repetition of the input spectrum $X(\omega)$. These additional spectra created are called image spectra and the phenomenon is known as imaging.

Anti-imaging Filter

The low-pass filter placed after the up sampler to remove the images created due to up sampling is called the anti-imaging filter.

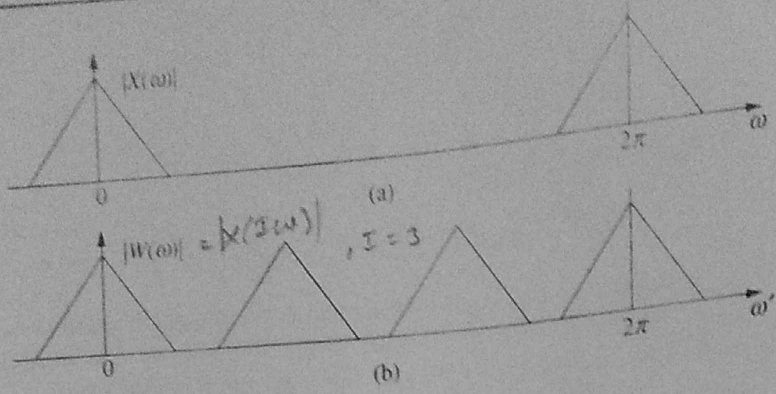


Figure 10.9 Spectrum of (a) $X(\omega)$ and (b) $X(3\omega)$.

EXAMPLE 10.1 Show that the up sampler and down sampler are time-variant systems.

Solution: Consider a factor of I up sampler defined by

$$y(n) = x\left(\frac{n}{I}\right)$$

The output due to delayed input is given by

$$y(n, k) = x\left(\frac{n}{I} - k\right)$$

The delayed output is given by

$$y(n-k) = x\left(\frac{n-k}{I}\right)$$

Therefore,

$$y(n, k) \neq y(n-k)$$

So the up sampler is a time-variant system.

Consider a factor of D down sampler defined by

$$y(n) = x(Dn)$$

The output due to delayed input is given by

$$y(n, k) = x(Dn - k)$$

The delayed output is given by

$$y(n-k) = x[D(n-k)]$$

Therefore,

$$y(n, k) \neq y(n-k)$$

So the down sampler is a time-variant system.

EXAMPLE 10.2 Consider a signal $x(n) = u(n)$.

(i) Obtain a signal with a decimation factor 3.
(ii) Obtain a signal with an interpolation factor 3.

Solution: Given that $x(n) = u(n)$ is the unit step sequence and is defined as

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

The graphical representation of unit step sequence is shown in Figure 10.10(a).

(i) Signal with a decimation factor 3.
The decimated signal is given by

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

It is obtained by considering only every third sample of $x(n)$. The output signal $y(n)$ is shown in Figure 10.10(b).

(ii) Signal with interpolation factor 3.
The interpolated signal is given by

$$y(n) = x\left(\frac{n}{I}\right) = x\left(\frac{n}{3}\right)$$

The output signal $y(n)$ is shown in Figure 10.10(c). It is obtained by inserting two zeros between two consecutive samples.

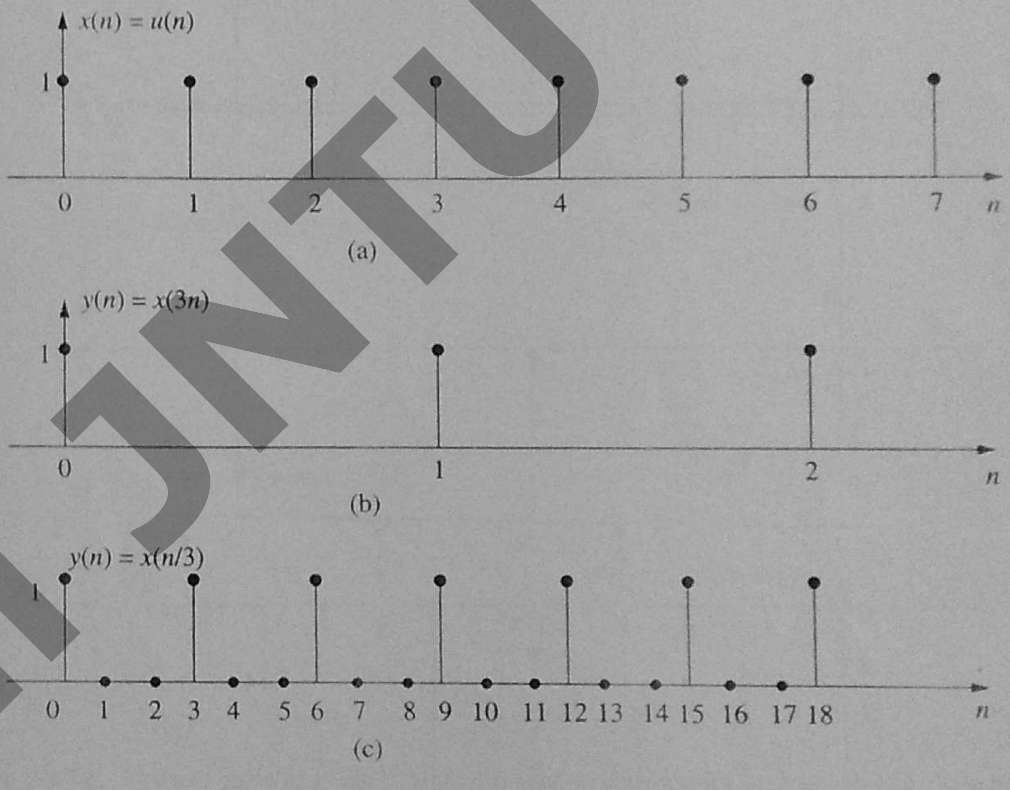


Figure 10.10 Plots of (a) $x(n) = u(n)$, (b) $x(3n)$ and (c) $x(n/3)$.

EXAMPLE 10.3 Consider a ramp sequence and sketch its interpolated and decimated versions with a factor of 3.

Solution: The ramp sequence is denoted as $r(n)$ and defined as

$$r(n) = \begin{cases} nu(n), & \text{for } n \geq 0 \\ 0, & \text{elsewhere} \end{cases}$$

The graphical representation of unit ramp signal is shown in Figure 10.11(a). The decimated signal is given by

$$y(n) = r(Dn) = r(3n)$$

The output signal $y(n) = r(3n)$ is shown in Figure 10.11(b). It is obtained by skipping 2 samples between every two successive sampling instants.

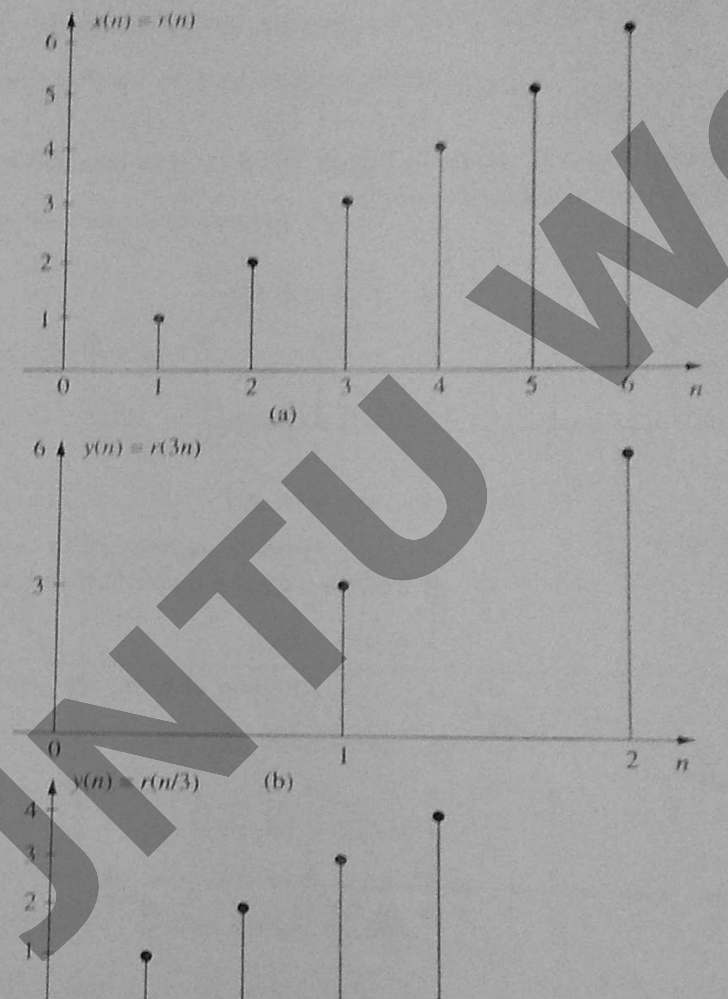


Figure 10.11 Plots of (a) $r(n) = nu(n)$, (b) $y(n) = r(3n)$ and (c) $y(n) = r(n/3)$.

The interpolated signal is given by

$$y(n) = r\left(\frac{n}{I}\right) = r\left(\frac{n}{3}\right)$$

The output signal $y(n) = r\left(\frac{n}{3}\right)$ is shown in Figure 10.11(c). It is obtained by inserting two zeros between every two successive sampling instants.

EXAMPLE 10.4 Consider a signal $x(n) = \sin \pi n u(n)$.

(i) Obtain a signal with a decimation factor 2.
(ii) Obtain a signal with an interpolation factor 2.

Solution: The given signal is $x(n) = \sin \pi n\, u(n)$. It is as shown in Figure 10.12(a).

(i) Signal with decimation factor 2.
The signal $x(n)$ with a decimation factor 2 is given by

$$y(n) = x(2n) = \sin 2\pi n\, u(n), \quad n = 0, 1, 2, \ldots$$

Figure 10.12(b) shows the plot of $x(n)$ decimated by a factor of 2, i.e., $x(2n)$ versus n.

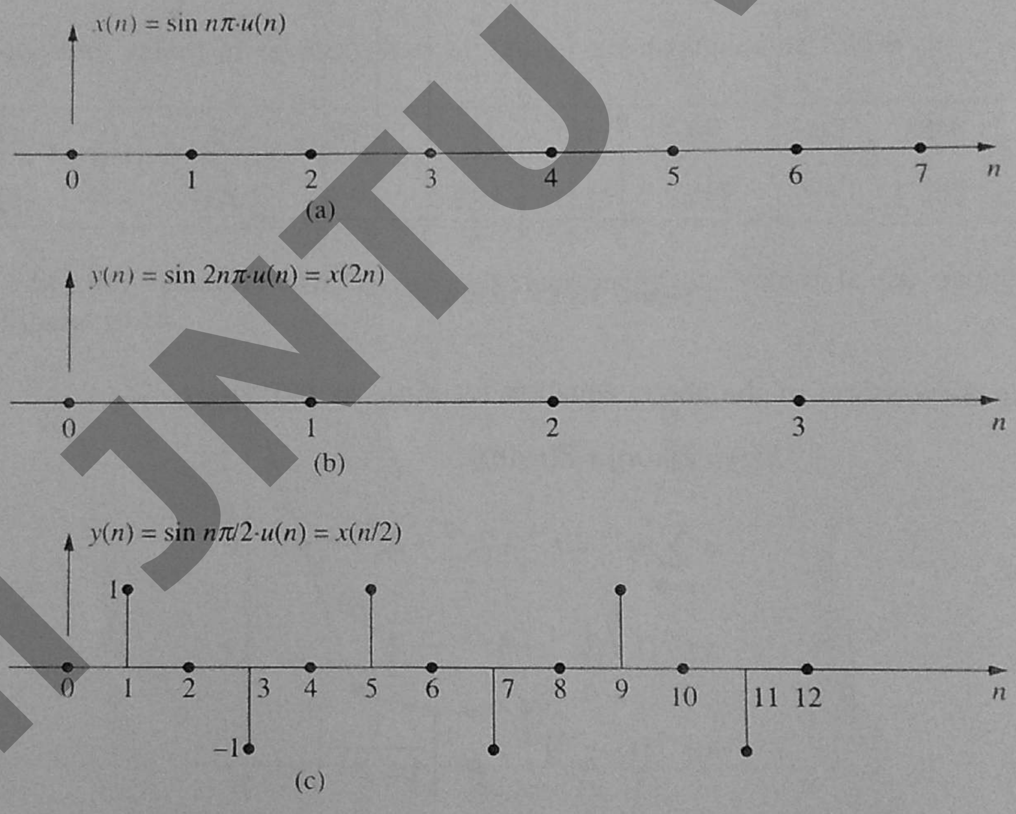


Figure 10.12 Plots of (a) $x(n) = \sin n\pi u(n)$, (b) $y(n) = \sin 2n\pi u(n)$ and (c) $y(n) = \sin (n\pi/2)u(n)$.

(ii) Signal with interpolation factor 2.
The signal $x(n)$ with an interpolation factor 2 is given by

$$x(n) = x\left(\frac{n}{2}\right) = \sin\frac{n\pi}{2}u(n), \quad n = 0, 1, 2, \ldots$$

The plot of interpolated signal $x(n/2)$ is shown in Figure 10.12(c).

EXAMPLE 10.5 Consider the signal $x(n) = nu(n)$.

(i) Determine the spectrum of the signal.

(ii) The signal is applied to a decimator that reduces the sampling rate by a factor 3. Determine the output spectrum.

(iii) Show that the spectrum in part (ii) is simply Fourier transform of $x(3n)$.

Solution: Given that $x(n) = nu(n) = r(n)$.
From the given data, the sequence $x(n)$ is a ramp sequence. Figure 10.13 shows the graphical representation of ramp signal.

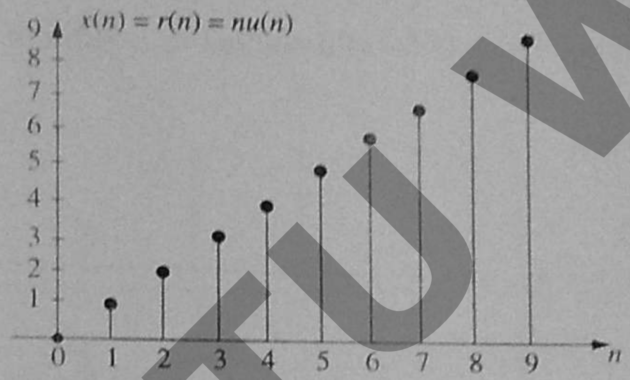


Figure 10.13 Ramp signal.

Taking Z-transform of the above equation for $x(n)$, we have

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

$$= \sum_{n=0}^{\infty} nz^{-n} = z^{-1} + 2z^{-2} + 3z^{-3} + \cdots$$

$$= z^{-1}[1 + 2z^{-1} + 3z^{-2} + \cdots]$$

$$= z^{-1}[1 - z^{-1}]^{-2} = z^{-1}\left[\frac{1}{1 - z^{-1}}\right]^2$$

$$= \frac{z}{(z-1)^2}$$

(i) The frequency spectrum of the signal $x(n)$, i.e., $X(\omega)$ is obtained by substituting $z = e^{j\omega}$ in $X(z)$. Therefore,

$$X(\omega) = \frac{z}{(z-1)^2}\bigg|_{z=e^{j\omega}} = \frac{e^{j\omega}}{(e^{j\omega}-1)^2}$$

$$= \frac{\cos\omega + j\sin\omega}{(\cos\omega + j\sin\omega - 1)^2}$$

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

$$\therefore \quad |X(\omega)| = \frac{1}{\sqrt{(\cos 2\omega - 2\cos\omega + 1)^2 + (\sin 2\omega - 2\sin\omega)^2}}$$

$$= \frac{1}{\sqrt{1+4+1-8\cos\omega + 2\cos 2\omega}}$$

$$= \frac{1}{\sqrt{6-8\cos\omega + 2\cos 2\omega}}$$

For different values of ω, the values of $|X(\omega)|$ are tabulated as follows:

ω	0	$\pi/4$	$\pi/2$	$3\pi/4$	π	$5\pi/4$	$3\pi/2$	$7\pi/4$	2π
$\lvert X(\omega)\rvert$	∞	1.707	0.5	0.293	0.25	0.293	0.5	1.707	∞

The frequency spectrum $X(\omega)$ of $x(n)$ obtained using the values in the above table is shown in Figure 10.14.

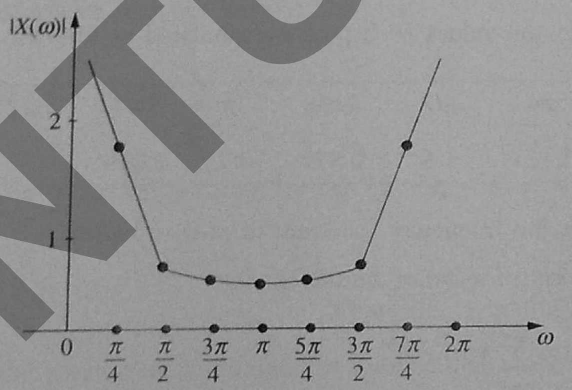


Figure 10.14 Magnitude spectrum of $x(n) = nu(n)$.

(ii) The sequence $x(n)$ is applied to a decimator that reduces sampling rate by a factor 3. So the respective decimated signal is given by $y(n) = x(3n) = 3nu(n)$. Then, the representation of the output sequence $y(n) = x(3n) = 3nu(n)$ is shown in Figure 10.15.

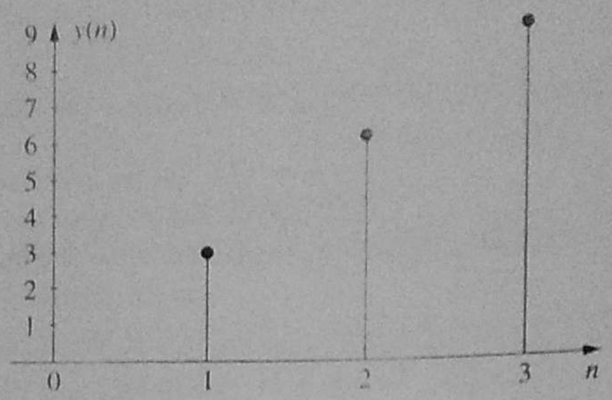


Figure 10.15 Plot of $y(n) = x(3n) = 3nu(n)$.

The Z-transform of $y(n) = 3nu(n)$ is:

$$Y(z) = 3\frac{z}{(z-1)^2}$$

Therefore, the frequency response is:

$$Y(\omega) = 3\frac{e^{j\omega}}{(e^{j\omega}-1)^2}$$

The magnitude spectrum $Y(\omega)$ of output sequence $y(n)$ is:

$$|Y(\omega)| = \frac{3}{\sqrt{6-8\cos\omega + 2\cos 2\omega}}$$

For different values of ω, the values of $|Y(\omega)|$ are tabulated as follows:

ω	0	$\pi/4$	$\pi/2$	$3\pi/4$	π	$5\pi/4$	$3\pi/2$	$7\pi/4$	2π
$\lvert Y(\omega)\rvert$	∞	5.121	1.5	0.879	0.75	0.879	1.5	5.121	∞

Figure 10.16 shows the frequency spectrum of $y(n) = 3nu(n)$.

(iii) Fourier transform of $x(3n) = 3nu(n)$ is:

$$X(\omega) = \sum_{n=0}^{\infty} x(n)\, e^{-j\omega n}$$

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

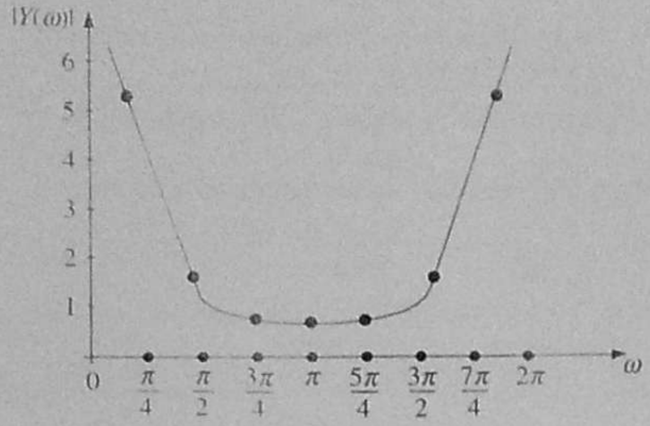


Figure 10.16 Frequency spectrum of $y(n) = 3nu(n)$.

$$= 0 + 3e^{-j\omega} + 6e^{-j2\omega} + 9e^{-j3\omega} + 12e^{-j4\omega}$$

$$= 3e^{-j\omega}[1 + 2e^{-j\omega} + 3e^{-j2\omega} + 4e^{-j3\omega} + \cdots]$$

$$= \frac{3e^{-j\omega}}{[1-e^{-j\omega}]^2} = \frac{3e^{j\omega}}{[e^{j\omega}-1]^2}$$

This is same as spectrum of signal in part (ii), that is, of $x(3n) = 3nu(n)$.

EXAMPLE 10.6 Consider the signal $x(n) = a^n u(n)$, $|a| < 1$.

(i) Determine the spectrum of the signal.

(ii) The signal is applied to an interpolator that increases sampling rate by a factor of 2. Determine its output spectrum.

(iii) Show that the spectrum in part (ii) is simply Fourier transform of $x(n/2)$.

Solution: The given signal is $x(n) = a^n u(n)$, $|a| < 1$.

(i) Taking Z-transform of $x(n)$, we have

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

$$= \sum_{n=0}^{\infty} (az^{-1})^n = 1 + az^{-1} + (az^{-1})^2 + (az^{-1})^3 + \cdots$$

$$= [1 - az^{-1}]^{-1} = \frac{1}{1-az^{-1}} = \frac{z}{z-a}, \ |z| > a$$

The spectrum $X(\omega)$ of the given $x(n)$ is obtained by substituting $z = e^{j\omega}$ in $X(z)$.

$$\therefore \qquad X(\omega) = X(z)\Big|_{z=e^{j\omega}} = \frac{e^{j\omega}}{e^{j\omega} - a} = \frac{\cos\omega + j\sin\omega}{(\cos\omega - a) + j\sin\omega}$$

$$\therefore \qquad |X(\omega)| = \frac{\sqrt{\cos^2\omega + \sin^2\omega}}{\sqrt{(\cos\omega - a)^2 + \sin^2\omega}} = \frac{1}{\sqrt{1 - 2a\cos\omega + a^2}}$$

The values of $|X(\omega)|$ for different values of ω are tabulated as follows:

ω	0	$\pi/4$	$\pi/2$	$3\pi/4$	π	$5\pi/4$	$3\pi/2$	$7\pi/4$	2π
$\|X(\omega)\|$	$\frac{1}{1-a}$	$\frac{1}{\sqrt{1-\sqrt{2}a+a^2}}$	$\frac{1}{\sqrt{1+a^2}}$	$\frac{1}{\sqrt{1+\sqrt{2}a+a^2}}$	$\frac{1}{1+a}$	$\frac{1}{\sqrt{1+\sqrt{2}a+a^2}}$	$\frac{1}{\sqrt{1+a^2}}$	$\frac{1}{\sqrt{1-\sqrt{2}a+a^2}}$	$\frac{1}{1-a}$

The frequency spectrum of the signal $x(n)$ is plotted as shown in Figure 10.17.

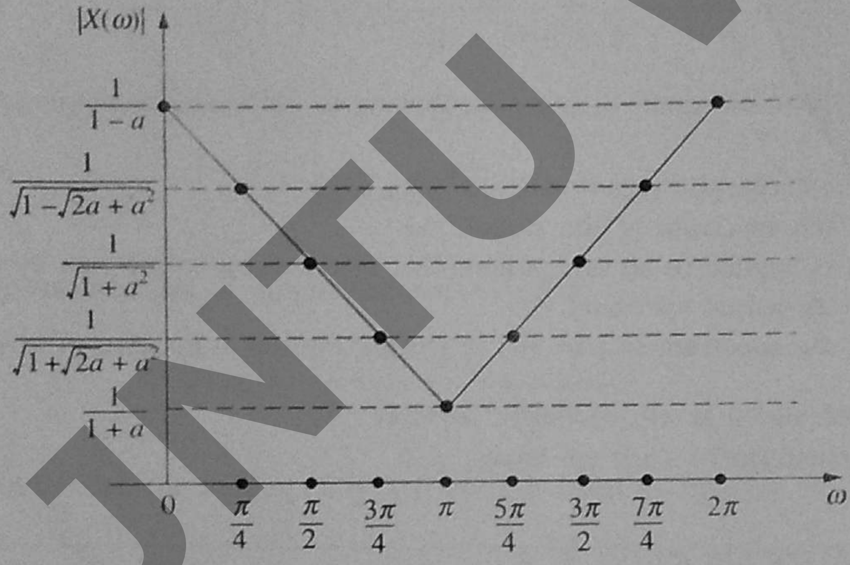


Figure 10.17 Magnitude spectrum of $X(\omega)$.

(ii) The interpolated signal $y(n)$ which is obtained by increasing the sampling rate by a factor of 2 for $a^n u(n)$ can be written as: $y(n) = x\left(\frac{n}{2}\right) = a^{n/2}u(n)$.

Taking Z-transform, we get

$$Z[y(n)] = Y(z) = \sum_{n=0}^{\infty} a^{n/2} z^{-n}$$

$$= 1 + a^{1/2}z^{-1} + az^{-2} + a^{3/2}z^{-3} + a^2 z^{-4} + \cdots$$

$$= 1 + \sqrt{a}z^{-1} + \left(\sqrt{a}z^{-1}\right)^2 + \left(\sqrt{a}z^{-1}\right)^3 + \cdots$$

$$= \left[1 - \sqrt{a}z^{-1}\right]^{-1} = \frac{1}{1 - \sqrt{a}z^{-1}}, \quad \sqrt{a}z^{-1} < 1$$

$$= \frac{z}{z - \sqrt{a}}, \ |z| > \sqrt{a}$$

The spectrum of signal $y(n)$ can be obtained by substituting $z = e^{j\omega}$ in the above equation.

$$\therefore \quad Y(\omega) = \frac{e^{j\omega}}{e^{j\omega} - \sqrt{a}} = \frac{\cos\omega + j\sin\omega}{(\cos\omega + j\sin\omega) - \sqrt{a}}$$

$$|Y(\omega)| = \frac{\sqrt{\cos^2\omega + \sin^2\omega}}{\sqrt{(\cos\omega - \sqrt{a})^2 + \sin^2\omega}} = \frac{1}{\sqrt{1 + a - 2\sqrt{a}\cos\omega}}$$

For different values of ω, the value of $|Y(\omega)|$ is tabulated below:

ω	0	$\pi/4$	$\pi/2$	$3\pi/4$	π	$5\pi/4$	$3\pi/2$	$7\pi/4$	2π
$\lvert Y(\omega)\rvert$	$\frac{1}{1-\sqrt{a}}$	$\frac{1}{\sqrt{1-\sqrt{2a}+a}}$	$\frac{1}{\sqrt{1+a}}$	$\frac{1}{\sqrt{1+\sqrt{2a}+a}}$	$\frac{1}{1+\sqrt{a}}$	$\frac{1}{\sqrt{1+\sqrt{2a}+a}}$	$\frac{1}{\sqrt{1+a}}$	$\frac{1}{\sqrt{1-\sqrt{2a}+a}}$	$\frac{1}{1-\sqrt{a}}$

The magnitude spectrum of $y(n) = x(n/2)$ is shown in Figure 10.18.

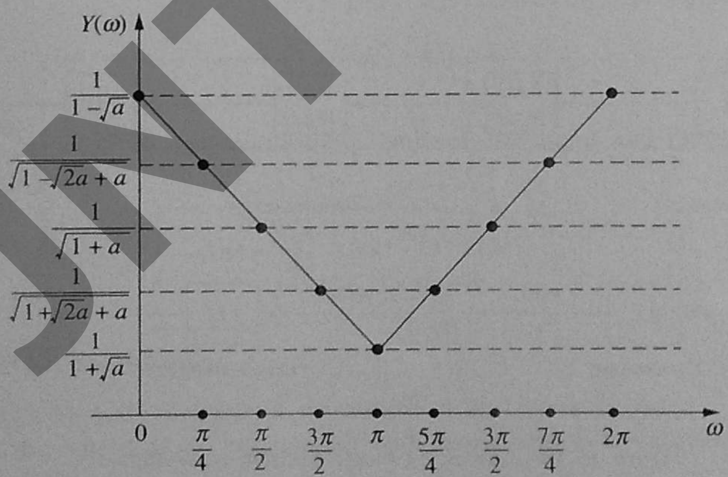


Figure 10.18 Magnitude spectrum of $y(n) = x(n/2)$.

(iii) The Fourier transform for $x(n/2)$ is obtained as:

$$F\left[x\left(\frac{n}{2}\right)\right] = F\left[a^{\frac{n}{2}}\, u(n)\right] = \sum_{n=0}^{\infty} a^{n/2} e^{-j\omega n}$$

$$= \sum_{n=0}^{\infty} [\sqrt{a} e^{-j\omega}]^n$$

$$= \frac{1}{1 - \sqrt{a} e^{-j\omega}}$$

$$= \frac{e^{j\omega}}{e^{j\omega} - \sqrt{a}}$$

This shows that spectrum in part (ii) is simply the Fourier transform of $x(n/2)$.

10.5 SAMPLING RATE CONVERSION

In some applications sampling rate conversion by a non-integer factor may be required. For example transferring data from a compact disc at a rate of 44.1 kHz to a digital audio tape at 48 kHz. Here the sampling rate conversion factor is 48/44.1, which is a non-integer.

A sampling rate conversion by a factor I/D can be achieved by first performing interpolation by factor I and then performing decimation by factor D. Figure 10.19(a) shows the cascade configuration of interpolator and decimator. The anti-imaging filter $H_u(z)$ and the anti-aliasing filter $H_d(z)$ are operated at the sampling rate, hence can be replaced by a simple low-pass filter with transfer function $H(z)$ as shown in Figure 10.19(b), where the low-pass filter has a cutoff frequency of $\omega_c = \min\left[\frac{\pi}{I}, \frac{\pi}{D}\right]$.

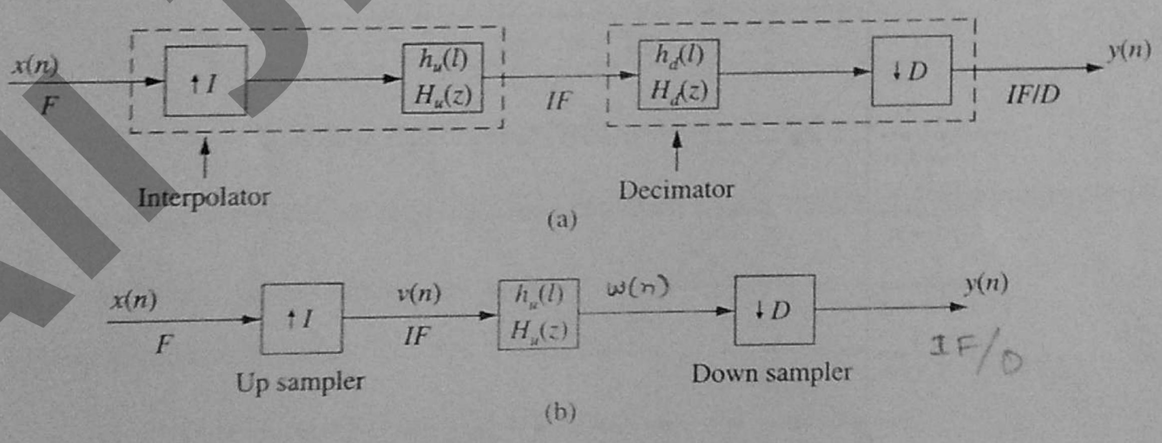


Figure 10.19 Cascading of sample rate converters.

Time domain and frequency domain relations of sampling rate converters

In Figure 10.19(a), $h_u(l)$ is Anti-imaging filter and $h_d(l)$ is Anti aliasing filter. The overall cutoff frequency of the two cascaded low-pass filters [i.e. $h_u(l)$ and $h_d(l)$] will be the minimum of the two cutoff frequencies.

The frequency response of $h_u(l)$ (anti-imaging filter) is given as:

$$H_u(\omega) = \begin{cases} I, & -\dfrac{\pi}{I} \le \omega \le \dfrac{\pi}{I} \\ 0, & \text{elsewhere} \end{cases}$$

The frequency response of $h_d(l)$ (anti-aliasing filter) is given as:

$$H_d(\omega) = \begin{cases} 1, & -\dfrac{\pi}{D} \le \omega \le \dfrac{\pi}{D} \\ 0, & \text{elsewhere} \end{cases}$$

Time domain relationship

From Figure 10.19(b), the output of the low-pass filter is given as:

$$w(l) = \sum_{k=-\infty}^{\infty} h(l-k)\, v(k)$$

$$= \sum_{k=-\infty}^{\infty} h(l-kI)\, x(k)$$

In Figure 10.19, $y(d)$ is the output of the down sampler and is given by

$$y(d) = w(dD)$$

$$= \sum_{k=-\infty}^{\infty} h(dD-kI)\, x(k)$$

Therefore, the time domain relationship between the input and output of a sampling rate converter is:

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

Frequency domain relationship

From Figure 10.19(b), $v(k)$ = output of up sampler with frequency ω_v.

Therefore, the output of the up sampler with frequency ω_v is expressed as:

$$V(\omega_v) = X(\omega_v I)$$

The output of the up sampler is passed through a LPF and hence we obtain $w(l)$ with frequency ω_v. Therefore, the output of the low-pass filter with frequency ω_v is given as

$$W(\omega_v) = H(\omega_v)X(\omega_v I)$$

$$\therefore \qquad W(\omega_v) = \begin{cases} IX(\omega_v I), & |\omega| \le \min\left(\dfrac{\pi}{D}, \dfrac{\pi}{I}\right) \\ 0, & \text{elsewhere} \end{cases}$$

The spectrum of the output sequence is given by

$$Y(\omega_y) = \frac{1}{D} W\left(\frac{\omega_v}{D}\right)$$

$$\therefore \qquad Y(\omega_y) = \frac{1}{D} W\left(\frac{\omega_y}{D}\right)$$

We know that, $\qquad \omega_v = \dfrac{\omega_y}{D}$

$$\therefore \qquad Y(\omega_y) = \frac{1}{D} W(\omega_v)$$

Substituting $W(\omega_v) = IX(\omega_v I)$, we get

$$Y(\omega_y) = \begin{cases} \dfrac{I}{D} X(\omega_v I), & |\omega_v| \le \min\left(\dfrac{\pi}{D}, \dfrac{\pi}{I}\right) \\ 0, & \text{elsewhere} \end{cases}$$

$$Y(\omega_y) = \begin{cases} \dfrac{I}{D} X\left(\dfrac{I\omega_y}{D}\right), & \left|\dfrac{\omega_y}{D}\right| \le \min\left(\dfrac{\pi}{D}, \dfrac{\pi}{I}\right) \\ 0, & \text{elsewhere} \end{cases}$$

So the frequency domain relationship between input and output of a sampling rate converter is:

$$Y(\omega_y) = \begin{cases} \dfrac{I}{D} X\left(\dfrac{I\omega_y}{D}\right), & |\omega_y| \le \min\left(\pi, \dfrac{\pi D}{I}\right) \\ 0, & \text{elsewhere} \end{cases}$$

Figure 10.20 shows the sampling rate conversion by a factor of 5/3. Figure 10.20(a) shows the actual signal $x(n)$. The sampling rate is increased by 5, by inserting 4 zero valued samples between successive values of $x(n)$ as shown in Figure 10.20(b). The output of anti-imaging filter is shown in Figure 10.20(c). The filtered data is then reduced for every three samples as shown in Figure 10.20(d).

A cascade of a factor of D down sampler and a factor of I up sampler is interchangeable with no change in the input and output relation if and only if I and D are co-prime.

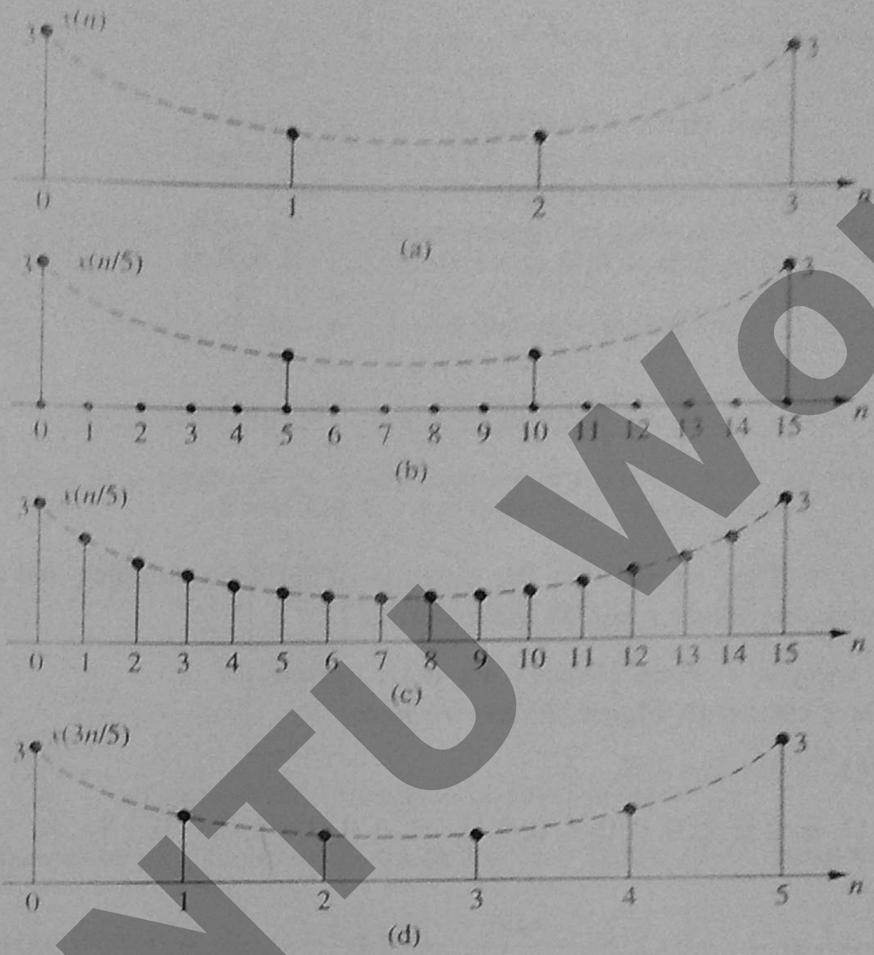


Figure 10.20 Sampling rate conversion by a factor of 5/3.

EXAMPLE 10.7 Considering an example

$$x(n) = \{1, 3, 2, 5, 4, -1, -2, 6, -3, 7, 8, 9, ...\}$$

show that a cascade of D down sampler and I up sampler is interchangeable only when D and I are co-prime.

Solution: Given $x(n) = \{1, 3, 2, 5, 4, -1, -2, 6, -3, 7, 8, 9, ...\}$

(i) Let $D = 2$ and $I = 3$. Here D and I are co-prime.

For the cascading shown in Figure 10.21, we have

$$x_d(n) = \{1, 2, 4, -2, -3, 8, \ldots\}$$

$$y_1(n) = \{1, 0, 0, 2, 0, 0, 4, 0, 0, -2, 0, 0, -3, 0, 0, 8, \ldots\}$$

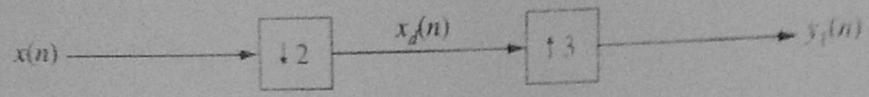


Figure 10.21 Cascading of $D = 2$ and $I = 3$.

Interchanging the cascading as shown in Figure 10.22, we have

$$x_u(n) = \{1, 0, 0, 3, 0, 0, 2, 0, 0, 5, 0, 0, 4, 0, 0, -1, 0, 0, -2, 0, 0, 6, 0, 0, -3, 0, 0, 7, 0, 0, 8, \ldots)$$

$$y_2(n) = \{1, 0, 0, 2, 0, 0, 4, 0, 0, -2, 0, 0, -3, 0, 0, 8, \ldots\}$$

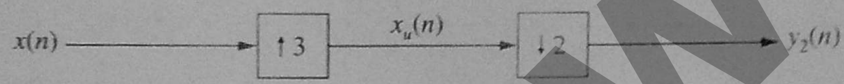


Figure 10.22 Cascading of $I = 3$ and $D = 2$.

Now $y_1(n) = y_2(n)$. This shows that the cascade of an I up sampler and a D down sampler are interchangeable when I and D are co-prime.

(ii) Let $D = 2$ and $I = 4$. Here D and I are not co-prime.
For the cascading shown in Figure 10.23, we have

$$x_d(n) = \{1, 2, 4, -2, -3, 8, \ldots\}$$

$$y_3(n) = \{1, 0, 0, 0, 2, 0, 0, 0, 4, 0, 0, 0, -2, 0, 0, 0, -3, 0, 0, 0, -8, \ldots\}$$

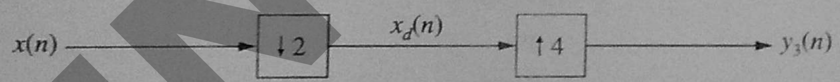


Figure 10.23 Cascading of $D = 2$ and $I = 4$.

Interchanging the cascading as shown in Figure 10.24, we have

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

$$y_4(n) = \{1, 0, 3, 0, 2, 0, 5, 0, 4, 0, -1, \ldots\}$$

Now, $y_3(n) \neq y_4(n)$.

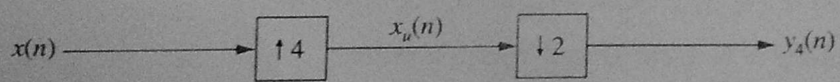


Figure 10.24 Cascading of $I = 4$ and $D = 2$.

This shows that the cascading of up sampler and down sampler is not interchangeable when D and I are not co-prime, i.e., when D and I have a common factor.

EXAMPLE 10.8 Show that the transpose of a factor of D decimator is a factor of D interpolator if the transpose of a factor of D down sampler is a factor of D up sampler.

Solution: The transpose of a digital filter is obtained by reversing all paths, interchanging the input and output nodes, replacing the pick off node with an adder and vice versa. The factor of D decimator is shown in Figure 10.25.

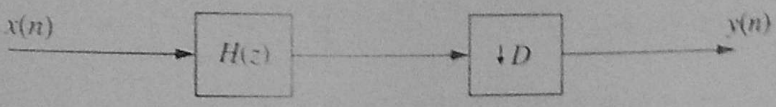


Figure 10.25 Factor of D decimator.

Interchanging the input and output nodes and reversing the paths results in Figure 10.26.

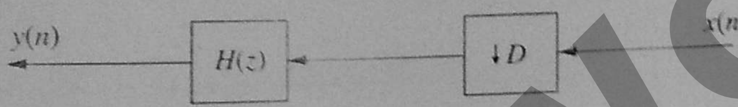


Figure 10.26 Transpose of decimator.

If the transpose of a factor of D down sampler is a factor of D up sampler, we have Figure 10.27.

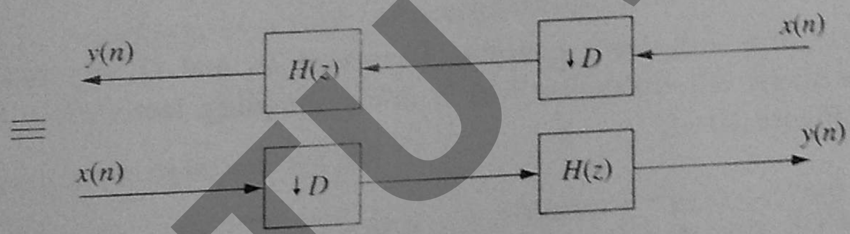


Figure 10.27 Transpose of down sampler.

Hence the transpose of a factor of D decimator is a factor of D interpolator.

10.6 IDENTITIES

1. The scaling of discrete-time signals and their addition at the nodes are independent of the sampling rate. It is illustrated in Figure 10.28.

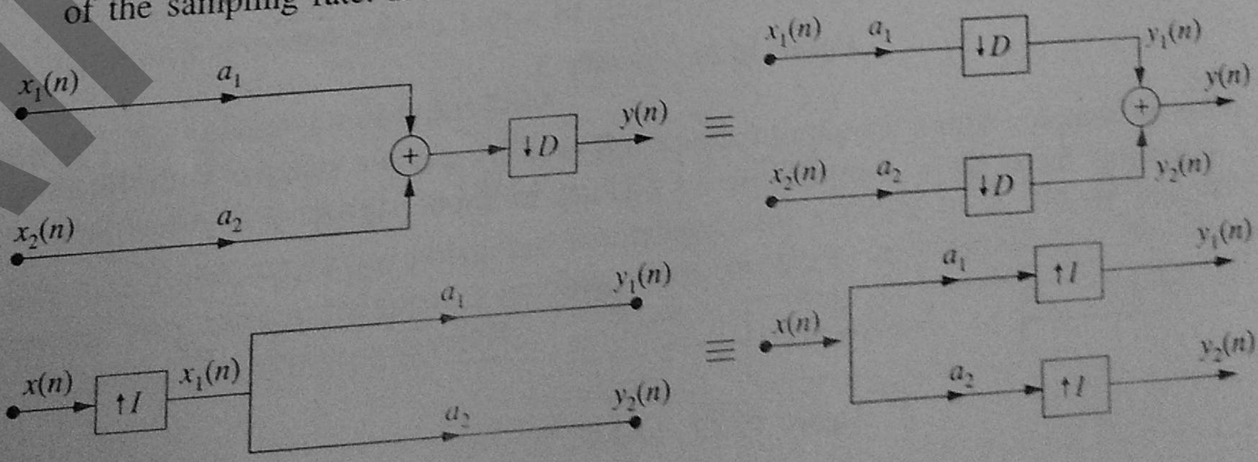


Figure 10.28 Identity 1.

2. A delay of D sample periods before a down sampler is the same as a delay of one sample period after the down sampler. It is illustrated in Figure 10.29.

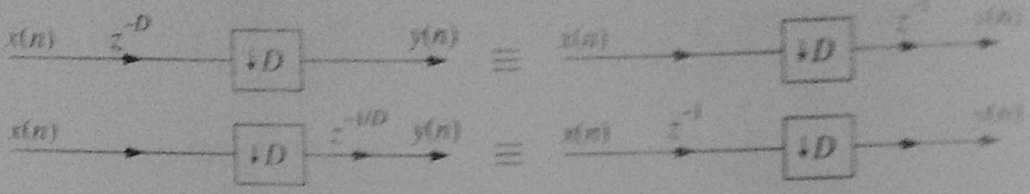


Figure 10.29 Identity 2.

3. A delay of one sample period before up sampling leads to a delay of I sample periods after the up sampling. It is illustrated in Figure 10.30.

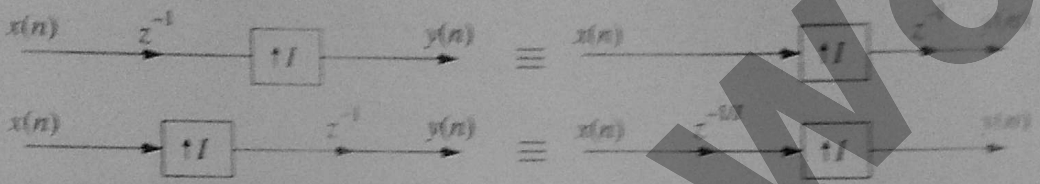


Figure 10.30 Identity 3.

4. Two down samplers with down sampling factors of D_1 and D_2 in cascade can be replaced by a single down sampler with a down sampling factor $D = D_1D_2$. It is illustrated in Figure 10.31.

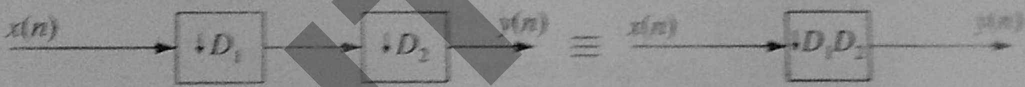


Figure 10.31 Identity 4.

5. Two up samplers with up-sampling factors of I_1 and I_2 in cascade can be replaced by a single up sampler with up-sampling factor $I = I_1I_2$. It is illustrated in Figure 10.32.

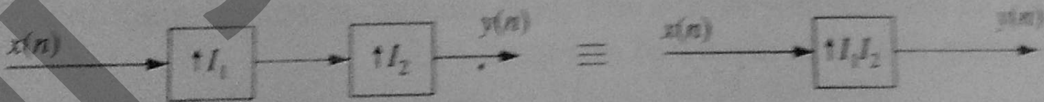


Figure 10.32 Identity 5.

6. An up sampler with a sampling factor I followed by a down sampler with the same sampling factor $D = I$ results in no change in input signal. It is illustrated in Figure 10.33.

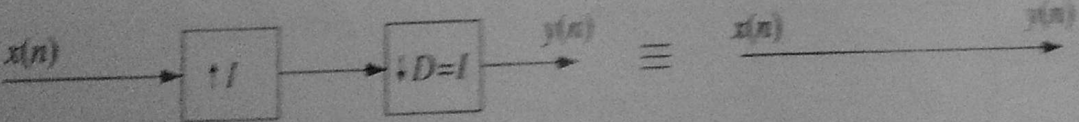


Figure 10.33 Identity 6.

7. A cascade of a down sampler with down sampling factor D followed by an up sampler with up sampling factor $I = D$ results in an output which is same as the input at the new sampling instants. It is illustrated in Figure 10.34.

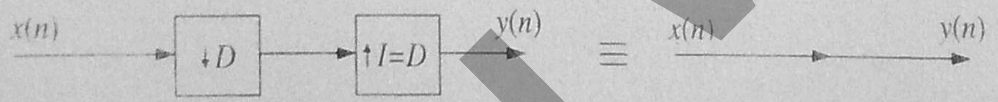


Figure 10.34 Identity 7.

8. and 9.

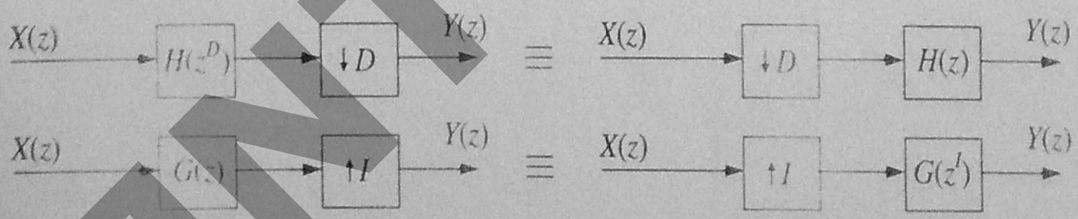


Figure 10.35 Identities 8 and 9.

8.1 FINITE WORD LENGTH EFFECTS

Finite Wordlength Effects

All the signals and systems are digital in DSP. The digital implementation has finite accuracy. When numbers are represented in digital form, errors are introduced due to their finite accuracy. These errors generate finite precision effects or finite wordlength effects.

Let us consider an example if the first order IIR filter to illustrate how errors are encountered in discretization. Such filter can be described as,

$$y(n)=\alpha\, y(n-1)+x(n) \qquad \text{.......(1)}$$

The z-transform of above equation gives

$$z[y(n)]=\alpha\, z[y(n-1)+x(n)]$$

$$X(z)=\alpha\, z^{-1}\,Y(z)+X(z)$$

Hence the transfer function

$$H(z)=\frac{1}{1-\alpha z^{-1}}=\frac{z}{z-\alpha}$$

Here observe that 'α' is the filter coefficient when this filter is implemented on some DSP processor or software, 'α' can have only discrete values. Let the discrete values of α be represented by α'.

Hence the actual transfer function which is implemented is given as,

$$\tilde{H}(z)=\frac{z}{z-\alpha'}$$

The transfer function given by above equation is slightly different from H(z). Hence the actual frequency response will be different from desired response.

The input x(n) is obtained by sampling the analog input signal. Since the quantizer takes only fixed(discrete) values of x(n) ,error is introduced. The actual input can be denoted by $x(n)$.

$$x(n)=x(n)+e(n)$$

Here e(n) is the error introduced during A/D conversion process due to finite wordlength of the quantizer. Similarly error is introduced in the multiplication of α and y(n-1) in equation(1). This is because the product α y(n-1) has to be quantized to one of the available discrete values. This introduces error. These errors generate finite wordlength effects.

Finite Wordlength Effects in IIR Digital Filters

When an IIR filter is implemented in a small system, such as an 8-bit microcomputer, errors arise in representing the filter coefficients and in performing the arithmetic operations indicated by the difference equation. These errors degrade the performance of the filter and in extreme cases lead to instability.

Before implementing an IIR filter, it is important to ~~ascertain~~ decide the extent to which its performance will be degraded by finite wordlength effects and to find a remedy if the degradation is not acceptable. The effects of these errors can be reduced to acceptable levels by using more bits but this may be at the expense of increased cost.

The main errors in digital IIR filters are:

i. ADC Quantization Noise:

This noise is caused by representing the samples of the input data ,by only a ~~small~~ fixed number of bits.

ii. Coefficient quantization errors:

These errors are caused by representing the IIR filter coefficients by a finite number of bits.

iii. Overflow errors

These errors are caused by the additions or accumulation of partial results in a limited register length.

iv. Product round-off errors

These errors are caused when the output ,and results of internal arithmetic operations are rounded to the permissible wordlength.

Finite Wordlength Effects in FFT Filters

As in most DSP algorithms, the main errors arising from implementing FFT algorithms using fixed point arithmetic are

i. Round off errors

These errors are produced when the product $W^k B$ is truncated or rounded to the system wordlength.

ii. Overflow errors

These errors result when the output of a butterfly exceeds the permissible wordlength.

iii. Coefficient quantization errors

These errors result from representing the twiddle factors using a limited number of bits.

8.2 LIMIT CYCLES

8.2.1 Overflow oscillations

Limit Cycle: The finite wordlength effects are analyzed using the linear model of the digital systems. But nonlinearities are introduced because of quantization of arithmetic operations. Because of these nonlinearities, the stable digital filter under infinite precision may become unstable under finite preision.Because of this instability, oscillating periodic output is generated.Such output is called limit cycle.The limit cycle occur in IIR filters due to feedback paths. Hence FIR filters cannot have limit cycle oscillations

Types of Limit Cycles

There are two types of limit cycles.

(1) Granular and

(2) Overflow.

1. Granular Limit Cycles

The granular limit cycles are of low amplitude. These cycles occur in digital filters when the input signal levels are very low. The granular limit cycles are of two types. They are

i. Inaccessible limit cycles

ii. Accessible limit cycles.

2. Overflow Limit Cycles

Overflow limit cycles occur because of overflow due to addition in digital filters implemented with finite precision. The amplitudes of overflow limit cycles are very large and it can cover complete dynamic range of the register. This further leads to overflow causing cumulative effect. Hence overflow limit cycles are more serious than granular limit cycles.

Transfer Characteristics and Example

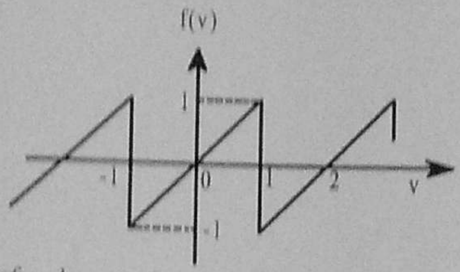


Figure: *Transfer characteristics of adder having overflow limit cycles*

Because of overflow limit cycle oscillations the output fluctuates between minimum and maximum values. The above figure shows the transfer characteristic of an adder that exihibit overflow limit cycle oscillations. Here f(v) indicates addition operation. Consider the addition of following two numbers in sign magnitude form.

$x_1 = 0.111$ i.e., $\frac{7}{8}$ $\qquad$ $x_2 = 0.101$ i.e., $\frac{5}{8}$

Then $\quad x_1 + x_2 = 1.010$ i.e., $-\frac{2}{8}$ $\qquad$ $x_1 + x_2 = 1.100 = -\frac{4}{8}$

Here overflow has occured in addition due to finite precision and the digit before decimal point makes the number negative.

Signal Scaling:

Need for Scaling: Limit cycle oscillations can be avoided by using the nonlinear transfer characteristic. But it introduces distortion in the output. Hence it is better to perform signal scaling such that overflow or underflow does not occur and hence limit cycle oscillations can be avoided.

Implementation of Signal Scaling

Figure shows the direct form-II structure of IIR filter. Let the input $x(n)$ be scaled by a factor s_0 before the summing node to prevent overflow. With the scaling, the transfer function will be,

$$H(z) = s_0 \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} = s_0 \frac{B(z)}{A(z)}$$

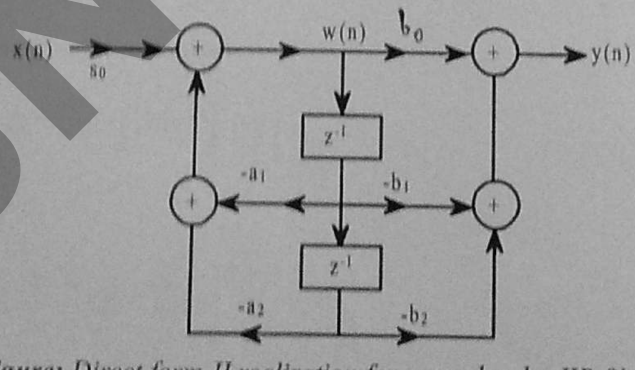


Figure: *Direct form-II realization for second order IIR filter*

Let us define the transfer function

$$H'(z) = \frac{W(z)}{X(z)}$$

From figure we can write above transfer function as,

$$H^{1}(z)=\frac{s_0}{1+a_1z^{-1}+a_2z^{-2}}$$

Since $A(z)=1+a_1z^{-1}+a_2z^{-2}$

$$H^{1}(z)=\frac{s_0}{A(z)}$$

Since, $H^{1}(z)=\frac{W(z)}{X(z)}$

$$\frac{W(z)}{X(z)}=\frac{s_0}{A(z)}$$

(or) $W(z)=\frac{s_0X(z)}{A(z)}$

Let $S(z)=\frac{1}{A(z)}$, then above equation becomes,

$$W(z)=s_0S(z)X(z)$$

Evaluating z-transform on unit circle, we put $z=e^{j\omega}$ in above equation,

$$W(e^{j\omega})=s_0S(e^{j\omega})X(e^{j\omega})$$

Taking inverse Fourier transform of above equation,

$$\omega(n)=\frac{1}{2\pi}\int W(e^{j\omega})e^{j\omega n}d\omega \qquad =\frac{1}{2\pi}\int s_0S(e^{j\omega})X(e^{j\omega})e^{j\omega n}d\omega$$

Hence $\omega^2(n)=\frac{s_0^2}{4\pi^2}\left|\int S(e^{j\omega})X(e^{j\omega})e^{j\omega n}d\omega\right|^2$

Schwartz inequality states that,

$$\left|\int x_1(t)x_2(t)dt\right|^2\leq\int|x_1(t)|^2dt\int|x_2(t)|^2dt$$

Using this relation we can write above equation as,

$$\omega^2(n)\leq\frac{s_0^2}{4\pi^2}\left[\int\left|S(e^{j\omega})\right|^2d\omega.\int\left|X(e^{j\omega})\right|^2d\omega\right],$$

since, $\left|e^{j\omega n}\right|=1$

$$\therefore\omega^2(n)\leq s_0^2\left[\frac{1}{2\pi}\int\left|S(e^{j\omega})\right|^2d\omega.\frac{1}{2\pi}\int\left|X(e^{j\omega})\right|^2d\omega\right]$$

Parseval's theorem states that $\frac{1}{2\pi}\int\left|X(e^{j\omega})\right|^2=\sum_{n=0}^{\infty}x^2(n)$. Then above equation can be written as,

$$\omega^2(n)\leq s_0^2\left[\frac{1}{2\pi}\int\left|S(e^{j\omega})\right|^2d\omega.\sum_{n=0}^{\infty}x^2(n)\right]$$

We have put $z = e^{j\omega}$. Hence $dz = je^{j\omega}d\omega$ or

$$d\omega = \frac{dz}{je^{j\omega}} = \frac{dz}{jz}, \text{ since } e^{j\omega} = z$$

Putting these values in equation

$$\omega^2(n) \leq s_0^2\left[\frac{1}{2\pi}\int\left|S(e^{j\omega})\right|^2.\frac{dz}{jz}.\sum_{n=0}^{\infty}x^2(n)\right]$$

$$\leq s_0^2\sum_{n=0}^{\infty}x^2(n).\frac{1}{2\pi j}\int\left|S(z)\right|^2.\frac{dz}{z}$$

Here $|S(z)|^2 = S(z).S(z^{-1})$. Then we have,

$$\omega^2(n) \leq s_0^2\sum_{n=0}^{\infty}x^2(n).\frac{1}{2\pi j}\int S(z).S(z^{-1}).z^{-1}dz$$

Here the integration is executed over a closed contour i.e. $\omega^2(n) \leq s_0^2\sum_{n=0}^{\infty}x^2(n).\frac{1}{2\pi j}\oint_C S(z).S(z^{-1}).z^{-1}dz$

(or) $$\omega^2(n) \leq \sum_{n=0}^{\infty}\frac{x^2(n)}{2\pi j}\left[s_0^2\oint_C S(z).S(z^{-1}).z^{-1}dz\right]$$

Here $\omega^2(n)$ represents instantaneous energy of signal after first summing node. And $x^2(n)$ represents instantaneous energy of input signal. Overflow will not occur if

$$\omega^2(n) \leq \sum_{n=0}^{\infty}x^2(n)$$

For this equation to be true we get following condition from equation

$$s_0^2\frac{1}{2\pi j}\oint_C S(z).S(z^{-1}).z^{-1}dz = 1$$

Earlier we have defined $S(z) = \frac{1}{A(z)}$. Hence above condition becomes,

$$s_0^2\frac{1}{2\pi j}\oint_C \frac{z^{-1}dz}{A(z).A(z^{-1})} = 1$$

(or) $$s_0^2 = \frac{1}{\frac{1}{2\pi j}\oint_C \frac{z^{-1}dz}{A(z).A(z^{-1})}}$$

Above equation gives the value of scaling factor s_0 to avoid overflow

8.3 ROUND OFF NOISE IN IIR DIGITAL FILTERS

Statistical Model for Analysis of Round-off Error Multiplication:

We perform arithmetic operations like addition and multiplication some errors will be occured. Those errors are called arithmetic errors. The results of arithmetic operations are required to be quantized so that they can occupy one of the finite set of digital levels. Such operation can be visualized as multiplier (or other arithmetic operation) with quantizer at its output.

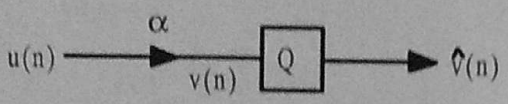


Figure: *Quantization of multiplication or product*

The above process can be represented by a statistical model for error analysis. The output $\hat{\upsilon}(n)$ and error $e_{\alpha}(n)$ in product quantization process.i.e.,

$$\hat{\upsilon}(n) = \upsilon(n) + e_{\alpha}(n)$$

The statistical model is shown below.

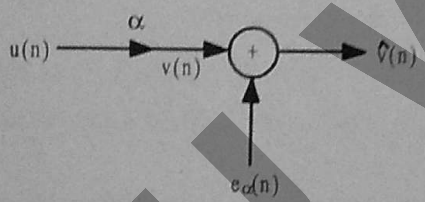


Figure: *Statistical model for analysis of round-off error multiplication*

For the analysis purpose following assumptions are made.

i) The error sequence $\{e_{\alpha(n)}\}$ is the sample sequence of a stationary white noise process.

ii $e_{\alpha}(n)$ is having uniform distribution over the range of quantization error.

iii) The sequence $\{e_{\alpha}(n)\}$ is uncorrelated with the sequence $\hat{\upsilon}(n)$ and input sequence ~~x(n)~~ v(n).

8.3.1Computational output round off noise

Product Round-off Errors and its Reduction:

The results of product or multiplication operations are quantized to fit into the finite wordlength,when the digital filters are implemented using fixed point arithmetic. Hence errors generated in such operation are called product round off errors.

The effect of product Round-off errors can be analyzed using the statistical model of the quantization process. The noise due to product round-off errors reduces the signal to noise ratio at the output of the filter. Some times this ratio may be reduces below acceptable levels. Hence it is necessary to reduce the effects of product round-off errors.

There are two solutions available to reduce product round-off errors.

a) Error feedback structures and

b)State space structure.

The error feedback structures use the difference between the unquantized and quantized signal to reduce the round-off noise. The difference between unquantized and quantized signal is fed back to the digital filter structure in such a way that output noise power due to round-off errors is reduced.

First Order Error-feedback Structure to reduce Round-off Error:

The results of product or multiplication operations are quantized to fit into the finite wordlength, when the digital filters are implemented using fixed point arithmetic. Hence errors generated in such operation are called product round off errors.

The effect of product round-off errors can be analyzed using the statistical model of the quantization process.

Let the quantization error signal be given as the difference between unquantized signal y(n) and quantized signal $\upsilon(n)$ .i.e.,

$$e(n) = y(n) - \upsilon(n)$$

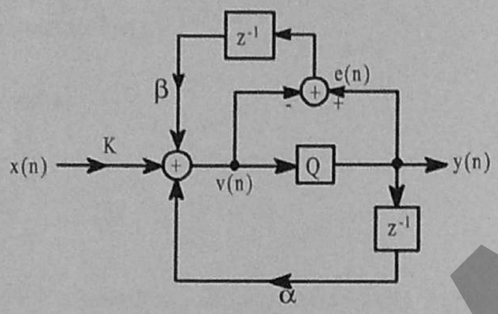


***Figure:** First order error feedback structure to reduce product round-off error*

This error signal is fed back in the structure such that round-off noise is reduced. Such structure for first order digital filter is shown in figure.

The incorporation of quantization error feedback as shown in figure helps in reducing the noise power at the output. This statement can be proved mathematically.

Round-off Errors in FFT Algorithms:

FFT is used in large number of applications. Hence it is necessary to analyze the effects due to finite wordlengths in FFT algorithms. The most critical error in FFT computation occurs due to arithmetic round-off errors.

The DFT is used in large number of applications such as filtering, correlation, spectrum analysis etc. In such applications DFT is computed with the help of FFT algorithms. Therefore it is important to study the quantization errors in FFT algorithms. These quantization effects mainly take place because of round-off errors. These errors are introduced when multiplications are performed in fixed point arithmetic.

FFT algorithms require less number of multiplications compared to direct computation of DFT. But it does not mean that quantization errors are also reduced in FFT algorithms.

Let $\sigma_x{}^2$ represents the variance of output DFT coefficients i.e., $|X(k)|$.

For N-point DFT σ_x is given as,

$$\sigma_x^2 = \frac{1}{3N} \qquad \text{......(1)}$$

For direct computation of DFT, the variance of quantization errors in multiplications is given as,

$$\sigma_q^2 = \frac{N}{3}.\Delta^2 \qquad \text{......(2)}$$

Here σ_q^2 is variance of quantization errors and Δ is step size which is given as,

$$\Delta = 2^{-b} \qquad \text{.....(3)}$$

And b is the number of bits to represent one level.

Hence equation (2) becomes,

$$\sigma_q^2 = \frac{N}{3}.2^{-2b} \qquad .....(4)$$

The signal to noise power ratio at the output (i.e., DFT coefficients) can be considered as the measure of quantization errors. This ratio is the ratio of variance of DFT coefficients $\left(\sigma_x^2\right)$ to the variance of quantization errors $\left(\sigma_q^2\right)$ i.e.,

Signal to noise ratio in direct computation of DFT $= \left(\frac{\sigma_x^2}{\sigma_q^2}\right)_{Direct\,DFT}$

From equation (1) and equation (2) we have,

$$\left(\frac{\sigma_x^2}{\sigma_q^2}\right)_{Direct\,DFT} = \frac{\frac{1}{3N}}{\frac{N}{3}.2^{-2b}} = \frac{2^{2b}}{N^2} \qquad ......(5)$$

When DFT is computed using FFT algorithms, the variance of the signal remains same i.e.,

$$\sigma_x^2 = \frac{1}{3N} \text{ from equation (1)} \qquad ......(6)$$

But with algorithms the variance of the quantization errors is given as,

$$\sigma_q^2 = \frac{2}{3}.2^{-2b} \qquad ......(7)$$

Hence signal to noise ration in FFT algorithms is,

$$\left(\frac{\sigma_x^2}{\sigma_q^2}\right)_{FFT} = \frac{\frac{1}{3N}}{\frac{2}{3}.2^{-2b}} = \frac{2^{2b}}{2N}$$

In the above expression, the signal to noise ration is inversely proportional to N. whereas in direct DFT computation the signal to noise ratio is inversely proportional to N^2 as given by equation (5). This means quantization errors increase fast with increase in 'N' in direct computation of DFT. But in FFT algorithms the quantization errors increase slowly with increase in 'N'.

Product of Round-off Errors in IIR Digital Filters:

The results of product or multiplication operations are quantized to fit into the finite wordlength, when the digital filters are implemented using fixed point arithmetic. Hence errors generated in such operation are called product round off errors.

Product round-off error analysis is an extensive topic.Our presentation here will be brief and aims to make you aware of the nature of the errors, their effects and how to reduce them if necessary.

The basic operations in IIR filtering are defined by the familiar second- order difference equation:

$$y(n) = \sum_{k=0}^{2} b_k x(n-k) - \sum_{k=1}^{2} a_k y(n-k)$$

Where x(n-k) and y(n-k) are the input and output data samples,and b_k and a_k are the filter coefficients. In practice these variables are often represented as fixed point numbers. Typically , each of the products b_k x(n-k) and a_k y(n-k) would require more bits to represent than any of the operands. For example, the product of a B-bit data and a B-bit coefficient is 2B bits long.

Truncation or rounding is used to quantize the products back to the permissible wordlength. Quantizing the products leads to errors,popularly known as round-off errors,in the output data and hence a reduction in the SNR. These errors can also lead to small-scale oscillations in the output of the digital filter,even when there is no input to the filter.

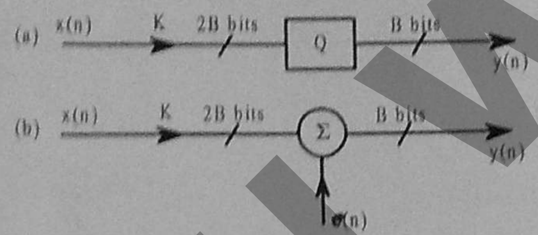


***Figure:** Representation of the product quantization error: (a) a block diagram representation of the quantization process; (b) a linear model of the quantization process*

The figure(a) represents a block diagram of the product quantization process,and figure (b) represents a linear model of the effect of product quantization. The model consists of an ideal multiplier,with infinite precision, in series with an adder fed by a noise sample, e(n), representing the error in the quantized product ,where we have assumed,for simplicity,that x(n),y(n), and K are each represented by B bits. Thus

y(n) = Kx(n) + e(n)

The noise power, due to each product quantization, is given by

$$\sigma_r^2 = \frac{q^2}{12}$$

Where r symbolizes the round-off error and q is the quantization step defined by the wordlength to which product is quantized. The round-off noise is assumed to be a random variable with zero mean and constant variance. Although this assumption may not always be valid, it is useful in assesing the performance of the filter.

8.4 METHODS TO PREVENT OVERFLOW

Prevent Overflow Limit Cycle Oscillations: The overflow limit cycles occur because of overflow due to addition in digital filters implemented with finite precision. The amplitudes of overflow limit cycles are very large and it can cover complete dynamic range of the register.

The specific design of filter coefficients do not assure prevention of overflow limit cycle oscillations. The transfer characteristic can be modified to avoid overflow limit cycle oscillations.

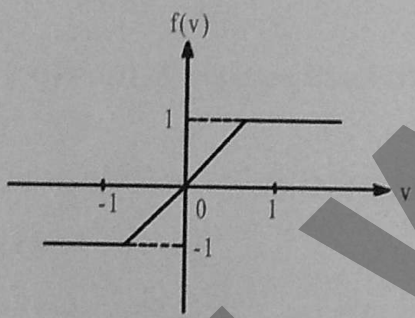


Figure: *Prevention of overflow limit cycle oscillations*

As shown in figure when an overflow or underflow is sensed, the output of the adder is set to its full scale value of ± 1. This prevents oscillatory output. This nonlinearity of the characteristic causes very small distortion in the output because overflow/underflow occurs rarely.

Scaling is also used to prevent overflow limit cycle oscillations. Limit cycle free structures are normally used to avoid the effects of limit cycles.

Characteristics of a Limit Cycle Oscillation with respect to the System by the following Difference equation $y(n) = 0.95y(n-1) + x(n)$.

Let $y_r(n)$ be the output of the system after the product term 0.95 $y(n-1)$ is quantized after rounding. i.e.,

$$y_r(n) = Q_r[0.95y(n-1)] + x(n)$$

Let $$x(n) = \begin{cases} 0.75 & \text{for } n = 0 \\ 0 & \text{for } n \neq 0 \end{cases}$$

Let $b = 4$ bits are used to represent the quantized product excluding sign bit.

With n=0

$y_r(n) = Q_r[0.95y_r(n-1)] + x(n)$

$\therefore\ y_r(0) = Q_r[0.95y_r(-1)] + x(0) = Q_r[0.95\times 0] + 0.75 = 0.75$

Since $y_r(-1) = 0 = 0.75$, $x(0) = 0.75$

$[0.75]_{10} = [0.11]_2$

$\therefore$ 4-bits rounded value of $[0.11]_2$ will be $[0.1100]_2$ i.e., 0.75 only.

$\therefore\ y_r(0) = 0.75$ after 4 bits rounding

With n = 1

$y_r(1) = Q_r[0.95y_r(0)] + x(1) = Q_r[0.95\times 0.75] + 0 = Q_r[0.7125]$

$[0.7125]_{10} = [0.1\ 0\ 1\ 1\ 0\ 1\ 1\ 0\ 0\ 1\ 1\ 0\ 0\ \cdots]_2$

Note that 0.7125 requires infinite binary digits for its representation. Let us round it to 4 bits.

$\therefore\ Q_r[0.7125]_{10} = [0.1\ 0\ 1\ 1]_2$ upto 4 bits

But decimal equivalent of $[0.1\ 0\ 1\ 1]_2$ is 0.6875.

$\therefore\ y_r(1) = 0.6875$

This means the actual value of $y_r(1) = 0.7125$ is changes to 0.6875 due to 4-bits quantization.

With n = 2

$y_r(2) = Q_r[0.95y_r(1)] + x(2) = Q_r[0.95\times 0.6875] + 0 = Q_r[0.653125]$

$[0.653125]_{10} = [0.1\ 0\ 1\ 0\ 0\ 1\ 1\ 1\ 0\ 0\ 1\ 1\ 0\ 0\ 1\ 1\ 0\ 0\ \cdots]$

$\therefore\ Q_r[0.653125]_{10} = [0.1\ 0\ 1\ 0]_2$ upto 4 bits $= [0.625]_{10}$

$\therefore\ y_r(2) = 0.625$

With n = 3

$y_r(3) = Q_r[0.95y_r(2)] + x(3) = Q_r[0.95\times 0.625] + 0 = Q_r[0.59375]$

$[0.59375]_{10} = [0.1\ 0\ 1\ 0\ 1]_2$

$\therefore\ Q_r[0.59375]_{10} = [0.1\ 0\ 1\ 1]_2$ upto 4 bits $= 0.625$

$\therefore\ y_r(3) = 0.625$

Thus $y_r(2) = y_r(3) = \ldots\ldots = 0.625$

Thus the system enters into limit oscillation when $n = 2$.

To calculate dead band

Consider equation

$$y(n-1) \le \frac{\delta/2}{1-|\alpha|}$$

Here $\delta = \frac{1}{2^b} = \frac{1}{2^4} = 0.0625$

$$y(n-1) \le \frac{0.0625/2}{1-0.95} \le 0.625$$

Dead band = [-0.625, +0.625]

Signal Scaling to Prevent Limit Cycle Oscillations: This is zero input condition. Following table lists the values of $y(n)$ before and after quantization. Here the values are rounded to nearest integer value.

n	$y(n)$ before quantization	$y(n)$ after quantization
-1	12	12
0	10.8	11
1	9.72	10
2	8.748	9
3	7.8732	8
4	7.08588	7
5	6.377292	6
6	5.7395628	6
7	5.1656065	5
8	4.6490459	5

***Table:** Values of $y(n)$ before and after quantization*

~~From table observe that if $|y(-1)| \le 5$, $y(n) = y(-1)$ for $n \ge 0$ for zero input. Hence the dead band will be $[-5,5]$.~~

Since the values are rounded to nearest integer after quantization, the step size will be $\delta = 1$. Hence dead band can also be calculated as follows:

$$y(n-1) = \frac{\delta/2}{1-\alpha}, \text{Here } \alpha = 0.9,\ y(n-1) = \frac{1/2}{1-0.9} = 5$$

Thus the dead band is $[-5,5]$.

Dynamic Range Scaling to Prevent the Effects of Overflow: The overflow can take place at some internal nodes when the digital filters are implemented by using fixed point arithmetic. Such nodes can be inputs/outputs of address or multipliers. This overflow can take place even if the inputs are scaled. Because of such overflow at intermediate points,produces totally undesired output or oscillations. The overflow can be avoided by scaling the internal signal levels with the help of scaling multipliers. These scaling multipliers are inserted at the apprppriate points in the filter structure to avoid possibilities of overflow. Sometimes these scaling multipliers are absorbed with the existing multipliers in the structure to reduce the total number and complexity.

At which node the overflow will take place is not known in advance. This is because the overflow depends upon type of input signal samples. Hence whenever overflow takes place at some node, the scaling should be done dynamically. Hence dynamic range scaling in the filter structure can avoid the effects of overflow.

Let $u_r(n)$ be the signal sample at r^{th} node in the structure. Then the scaling should ensure that,

$$|u_r(n)| \le 1 \qquad \text{for all r and n.}$$

8.5 TRADE OFF BETWEEN ROUND-OFF AND OVERFLOW NOISE, MEASUREMENT OF COEFFICIENT QUANTIZATION EFFECTS THROUGH POLE-ZERO MOVEMENT

Errors in Rounding and Truncation Operations : The computations like multiplication or addition are performed the result is truncated or rounded to nearest available digital level. This operation introduces an error. Hence the performance of the system is changed from expected value.

Truncation Error: This error is introduced whenever the number is represented by reduced number of bits.

Let $Q_t(x)$ be the value after truncation ,then truncation error will be,

$$\varepsilon_r = Q_t(x) - (x)$$

Here x is the original value of the number.

Rounding Error : This error is introduced whenever the number is rounded off to the nearest digital level. The number of bits used to represent the rounded number are generally less than the number of bits required for actual number.

Let $Q_r(x)$ be the value after rounding.Then rounding error will be,

$$\varepsilon_r = Q_r(x) - x$$

Here x is the original value of a number.

Tradeoff between roundoff and overflow noise:

Scaling operation

Scaling is a process of readjusting certain internal gain parameters in order to constrain internal signals to a range appropriate to the hardware with the constraint that the transfer function from input to output should not be changes.

The filter in figure with unscaled node x has the transfer function

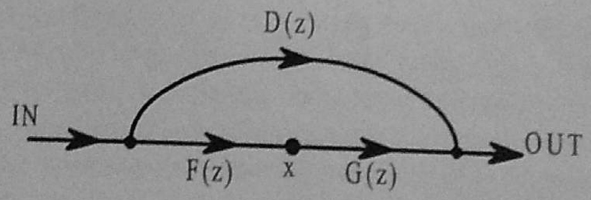
$$H(z) = D(z)+F(z)G(z) \qquad ......(1)$$

To scale the node x, we divide F(z) by some number β and multiply G(z) by the same number as in figure. Although the transfer function does not change by this operation, the signal level at node x has been changes. The scaling parameter β can be chosen to meet any specific scaling rule such as

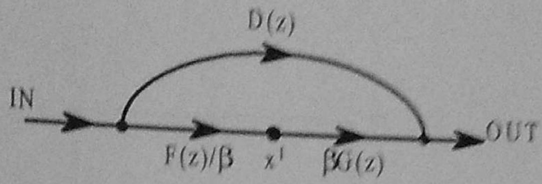
$$l_1 \text{ scaling}: \beta = \sum_{i=0}^{\infty} \lfloor f(i) \rfloor \qquad ......(2)$$

$$l_2 \text{ scaling}: \beta = \delta \sqrt{\sum_{i=0}^{\infty} \lfloor f^2(i) \rfloor} \qquad .....(3)$$

Where f(i) is the unit-sample response from input to the node x and the parameter δ can be interpreted to represent the number of standard deviations



(a)



(b)

Figure: *A filter with unscaled node x and (b) A filter with scaled node x'*

Representable in the register at node x if the input is unit-variance white noise. If the input is bound by $|u(n)| \le 1$, then,

$$|x(n)| = \left|\sum_{i=0}^{\infty} f(i)u(n-i)\right| \le \sum_{i=0}^{\infty} |f(i)| \qquad \text{......(4)}$$

Equation represents the true bound on the range of x and overflow is completely avoided by l_1 scaling in (2), which is the most stringent scaling policy.

In many cases, input can be assumed to be white noise. Although we cannot compute the variance at node x. for unit-variance white noise input,

$$E[x^2(n)] = \sum_{i=0}^{\infty} f^2(i) \qquad \text{.....(5)}$$

Since most input signals can be assumed to be white noise, l_2 scaling is commonly used. In addition, (5) can be easily computed. Since (5) is the variance (not a strict bound), there is a possibility of overflow, which can be reduced by increasing δ in (3). For large values of δ, the internal variables are scaled conservatively so that no overflow occurs. However, there is a trade-off between overflow and roundoff noise, since increasing δ deteriorates the output SNR (signal to noise ratio).

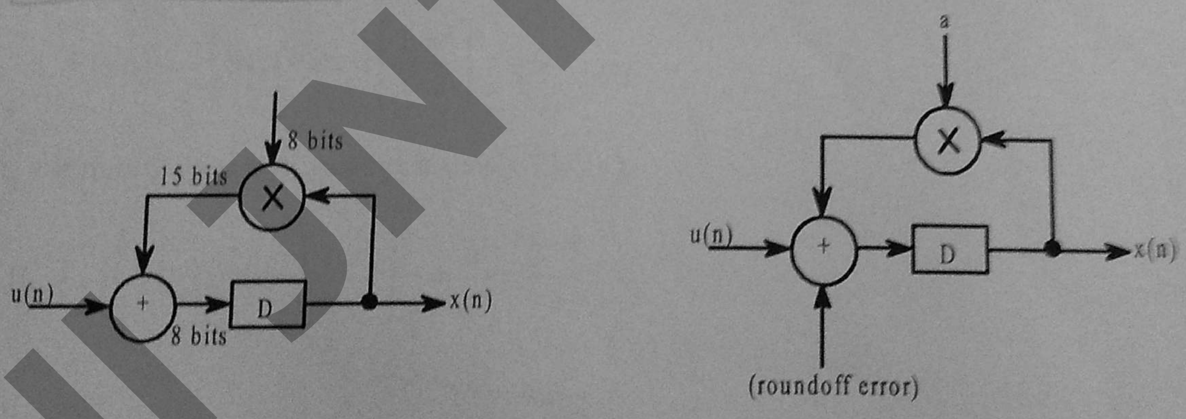


Figure: *Model of roundoff error*

Roundoff Noise: If two W-bit fixed point fraction numbers are multiplies together, the product is (2W-1) bit long. This product must eventually be quantized to W-bits by rounding or truncation. For example, consider the 1st-order IIR filter shown in figure. Assume that the input wordlength is W=8bits. If the multiplier coefficient wordlength is also the same, then to maintain full precision in the output we need to increase the output wordlength by 8 bits per iterations. This is clearly infeasible. The alternative is to roundoff (or truncate) the output to its nearest 8-bit representation.

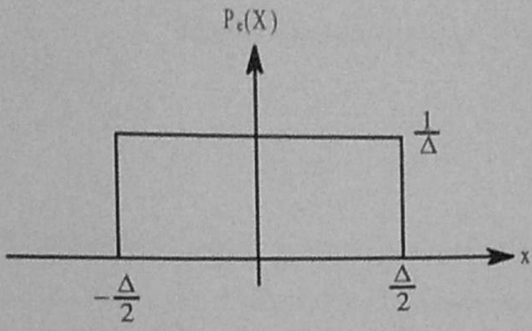


Figure: *Error probability distribution*

The result of such quantization introduces roundoff noise e(n). For mathematical ease a system with roundoff can be modeled as an infinite precision system with an external error input. For example in the previous case (shown in figure) we round off the output of the multiply add operation and an equivalent model is shown in figure.

Although rounding is not a linear operation, its effect at the output can be analyzed using linear system theory with the following assumptions about e(n):

1. E(n) is uniformly distributed white noise.
2. E(n) is a wide –sense stationary random process, i.e., mean and covariance of e(n) are independent of the time index n.
3. E(n) is uncorrelated to all other signals such as input and other noise signals.

Let the wordlength of the output be W-bits, then the roundoff error e(n) can be given by

$$\frac{-2^{-(w-1)}}{2} \le e(n) \le \frac{2^{-(w-1)}}{2} \qquad .....(6)$$

Since the error is assumed to be uniformly distributed over the interval given in (6), the corresponding probability distribution is shown in figure, where Δ is the length of the interval (i.e., $2^{-(w-1)}$).

Let us compute the mean E[e(n)] and variance $E\left[e^2(n)\right]$ of this error function.

$$E[e(n)] = \int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} x P_e(x)\,dx = \left[\frac{1}{\Delta}\frac{x^2}{2}\right]_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} = 0 \qquad .....(7)$$

Note that since mean is zero, variance is simply $E\left[e^2(n)\right]$

$$E\left[e^2(n)\right] = \int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} x^2 P_e(x)\,dx = \left[\frac{1}{\Delta}\frac{x^3}{3}\right]_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} = \frac{\Delta^2}{12} = \frac{2^{-2w}}{3} \qquad ......(8)$$

Figure: Signal flow graph

In other words (8) can be rewritten as

$$\sigma_e^2 = \frac{2^{-2w}}{3} \qquad \text{......(9)}$$

Where σ_e^2 is the variance of the roundoff error in a finite precision, W-bit wordlength system. Since the variance is proportional to 2^{-2w}, increase in wordlength by 1 bit decreases the error by a factor of 4.

The purpose of analyzing roundoff noise is to determine its effect at the output signal. If the noise variance at the output is not negligible in comparison to the output signal level, the worlength should be increase or some low noise structures should be used. Therefore, we need to compute SNR at the output, not just the noise gain to the output. In the noise analysis, we use a double length accumulator model, which means rounding is performed after two (2w-1)-bit products are added. Also, notice that multipliers are the sources for roundoff noise.

8.6DEADBAND EFFECTS

Deadband and Deadband of First Order Filter: Dead band is the range of output amplitudes over which limit cycle oscillations take place

Dead band of first order filter

Consider the first order filter,

$$y(n) = \alpha y(n-1) + x(n)$$

Here $\alpha \ y(n-1)$ is the product term. After rounding it to 'b' bits we get,

$$y(n) = Q_r[\alpha y(n-1)] + x(n)$$

When limit cycle oscillations take place,

$$Q_r[\alpha y(n-1)] = \pm y(n-1) \qquad .....(1)$$

The error due to rounding is less than $\frac{\delta}{2}$. Hence,

$$|Q[\alpha y(n-1)] - \alpha y(n-1)| \leq \frac{\delta}{2}$$

From equation (1) above equation can be written as,

$$|\pm y(n-1) - \alpha y(n-1)| \leq \frac{\delta}{2}$$

$$\therefore \ |y(n-1)|[1-|\alpha|] \leq \frac{\delta}{2}$$

$$\therefore \ |y(n-1)| \leq \frac{\delta/2}{1-|\alpha|}$$