DIGITAL SIGNAL PROCESSING NOTES

AKSHANSH CHAUDHARY

Digital Signal Processing Notes, First Edition

Copyright © 2013 Akshansh

ALL RIGHTS RESERVED.

Presented by: Akshansh Chaudhary
Graduate of BITS Pilani, Dubai Campus
Batch of 2011

Course content by: Dr. R. Mary Lourde
Then Faculty, BITS Pilani, Dubai Campus

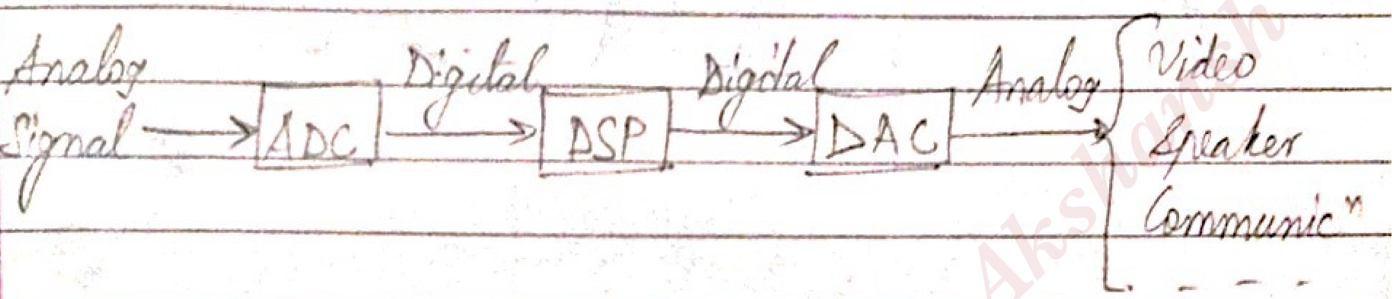
Layout design by: AC Creations © 2013

The course content was prepared during Fall, 2013.
More content available at: www.Akshansh.weebly.com

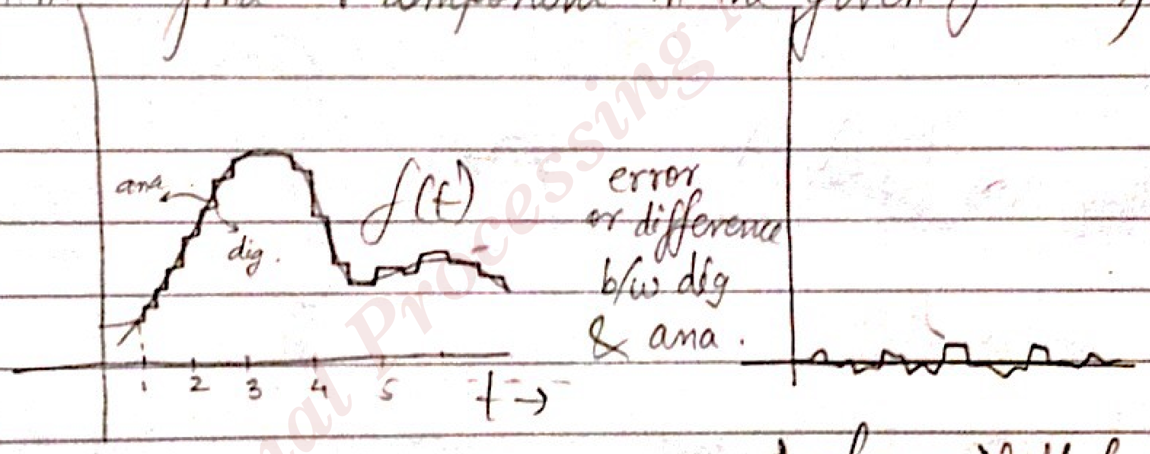
DISCLAIMER: While the document has attempted to make the information as accurate as possible, the information on this document is for personal and/or educational use only and is provided in good faith without any express or implied warranty. There is no guarantee given as to the accuracy or currency of any individual items. The document does not accept responsibility for any loss or damage occasioned by use of the information contained and acknowledges credit of author(s) where ever due. While the document makes every effort to ensure the availability and integrity of its resources, it cannot guarantee that these will always be available, and/or free of any defects, including viruses. Users should take this into account when accessing the resources. All access and use is at the risk of the user and owner reserves that right to control or deny access.
Information, notes, models, graph etc. provided about subjects, topics, units, courses and any other similar arrangements for course/paper, are an expression to facilitate ease of learning and dissemination of views/personal understanding and as such they are not to be taken as a firm offer or undertaking. The document reserves the right to discontinue or vary such subjects, topic, units, courses, or arrangements at any time without notice and to impose limitations on accessibility in any course.

Puffin
Date 9. 9. 2013
Page

Introduction



* Consider an analog signal & its digital signal.
Aim :- find ? component in the given fⁿ (f(t))

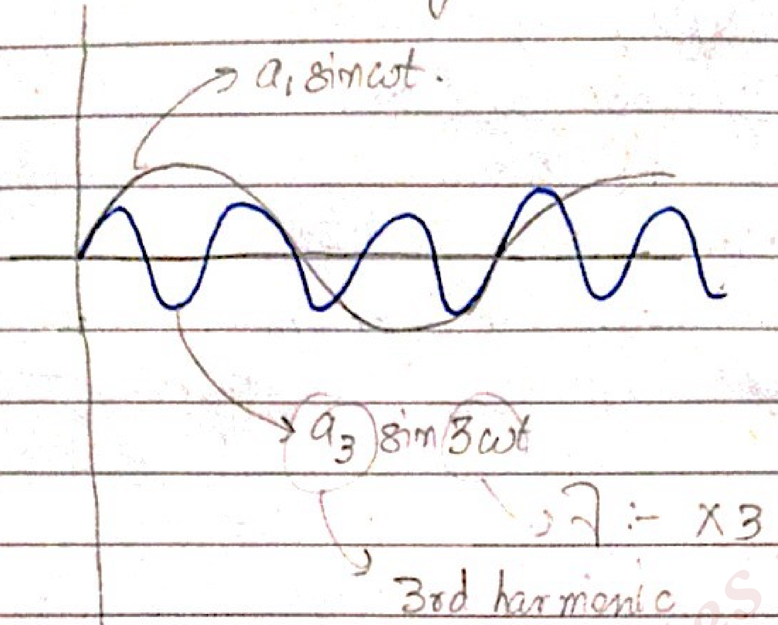


Analog = Digital + error

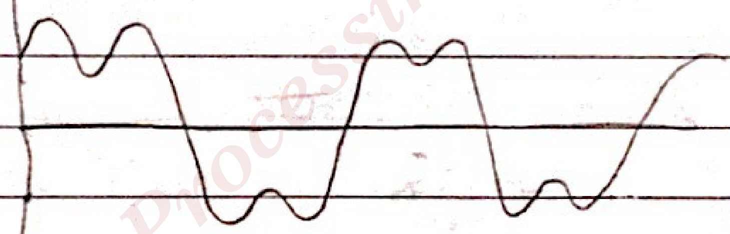
We have a time domain & want to use frequency
⇓
USE FOURIER SERIES.

$$f(t) = \underbrace{a_0}_{\text{DC component in the given electrical signal}} + \underbrace{\sum (a_n \cos n\omega t + b_n \sin n\omega t)}_{\text{harmonic components}}$$

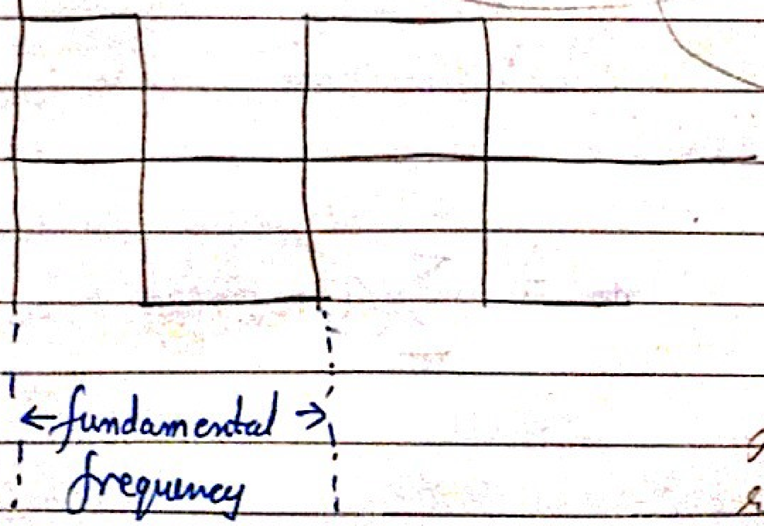
☆ Idea :- Consider 2 signals :-



Adding above 2 f^{ns}



If further harmonics are taken & added Going to ∞ frequencies, we get



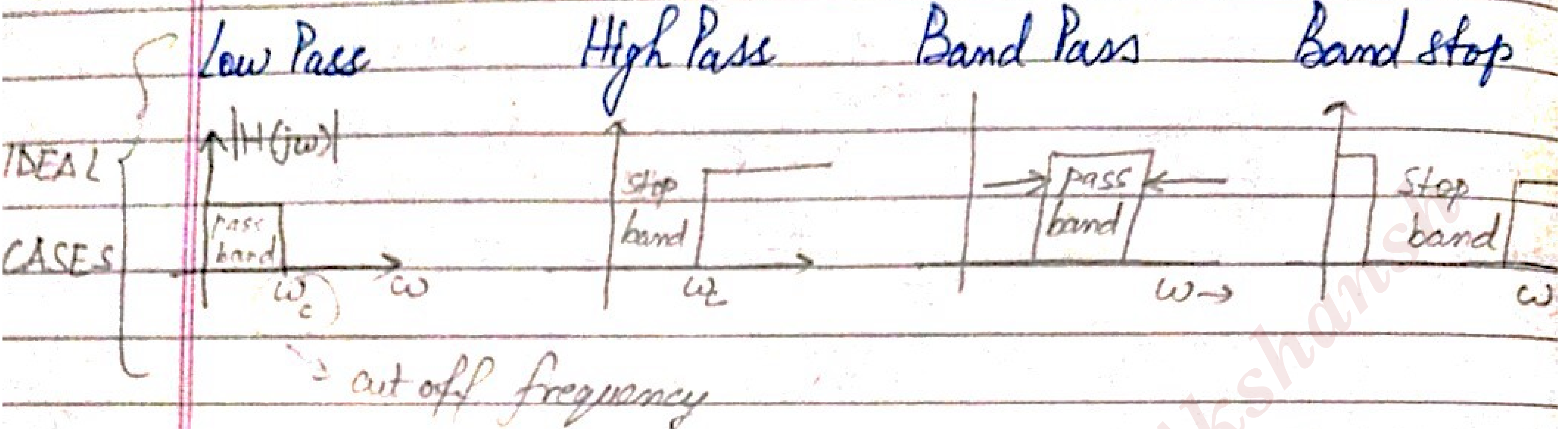
This pulse cannot be exact, because physically, ∞ freq. can't be realised.

* Discrete electrical components :- R, L, C.

* Note :-
A capacitor consists of 2 $\parallel$ plates. These 2 are the conducting mediums. Whenever ∃ conducting medium, ∃ resistance ; whenever ∃ resistance, ∃ delay.
So, capacitors have delays & ideal square pulses cant be physically realized.

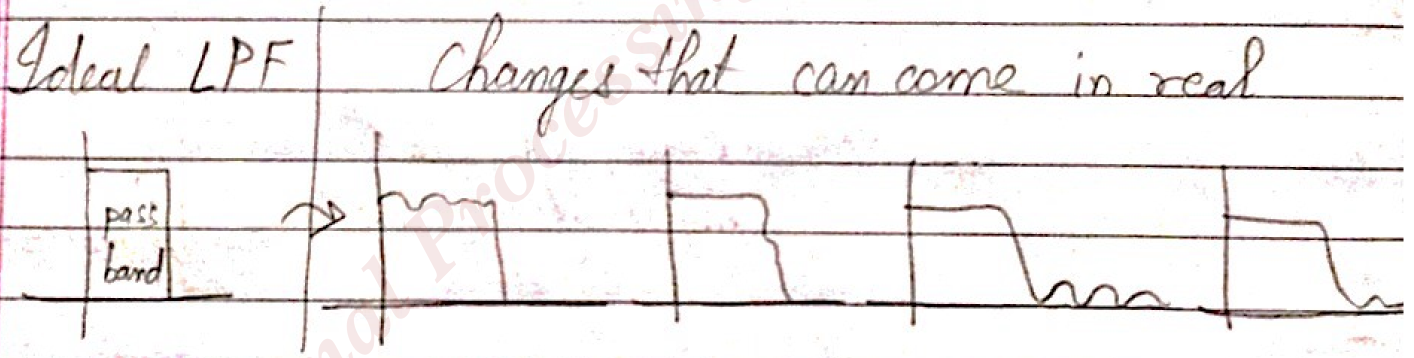
* Note :- We saw that analog signal is converted to digital by using many freq., in terms of harmonics.
So, manually, while converting analog to dig, we are intentionally adding some extra freq.
Same goes true when we change from dig. to analog. These extra freq$_s$. can be termed as errors in the signal.
So, in any physical applicⁿ, to retain the actual signal, these errors need to be removed. This removal is called FILTERING THE FREQUENCY.
Various filters such as Band pass, Band stop, High pass & low pass ∃.

✱ Analog & Digital filters :-



✱ Widely used low Pass Filters :-

(a) Butterworth filter
(b) Chebyshev filter.
(c) Elliptical



- Properties of ideal filter : • group delay
  • phase delay :-

Let $x(t)$ be a distortionless filter

o/p $y(t) = G\,x(t-\tau)$: change in amp. & phase shift

Taking Fourier transform, $H(j\omega) = \frac{Y(j\omega)}{R(j\omega)}$

$= G.e^{-j\omega\tau}$

wrt frequency

∴ $A(\omega) = G$ & $\phi(\omega) = -\omega\tau$

Amplitude response   Phase response

GROUP DELAY :- $\boxed{-\frac{d}{dw}\phi(\omega) = T_g(\omega)} = \tau$

↳ for a distortionless filter, gain & group delay are const over non zero range of i/p spectrum.

* let $x(t) = A\cos\omega t$ → a single spectral component

$y(t) = B\cos(\omega t + \theta) = B\cos\omega(t - t_0)$; $t_0 = \frac{-\theta}{\omega}$

(θ: phase shift, the delay in transforming signal from i/p to o/p)

PHASE DELAY : $\boxed{-\frac{\phi(\omega)}{\omega}} = t_0$

↳ $\phi(\omega)$ is phase shift of filter.

* IDEAL filters : • non causal -- physically unrealizable.
Hence, characteristics or TF approxim is reqd.

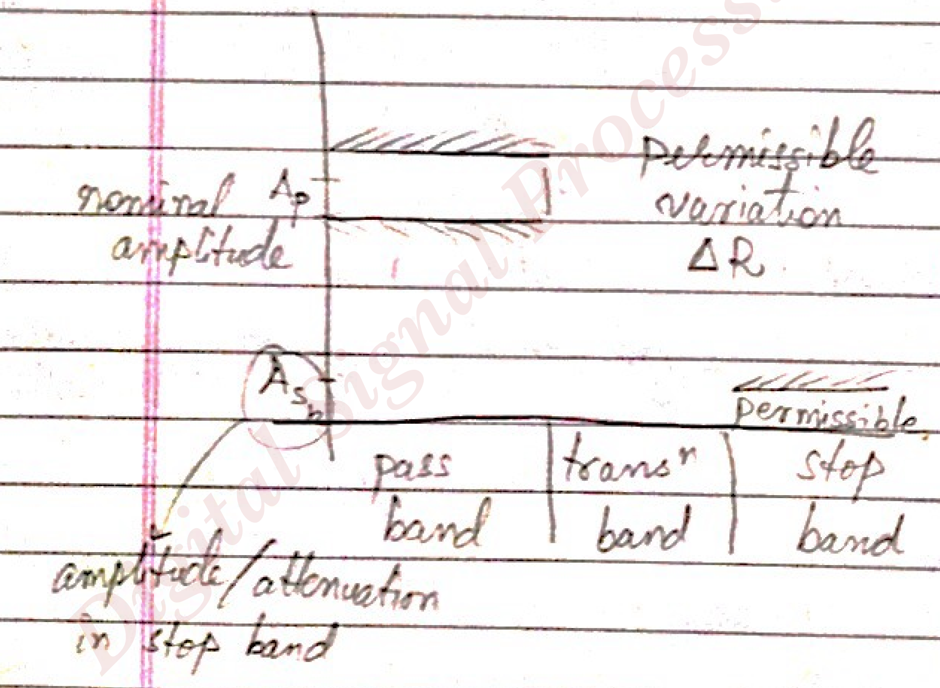
* Distortion from Ampl. char. is called **Amplitude distortion**
Distortion from Phase char. is called **Phase distortion**.

- For Communic^n applic^{ns} :-
  - Amp. distortion should be max
  - Phase distortion should be tolerable to some extent
- Image, video applic^{ns} :-
  - Linear phase char. is imp.
  - Amp. distortion can be tolerated to some extent

if not, then pixel distortion will happen

if not, quality/color can change; no overall change of image

Approxim^n of ideal filters :-



(a) BUTTERWORTH FILTER (BF)

↳ monotonically ↓ char. ; no oscill^{ns}

$$\left|H_{B_A}(j\omega)\right| = \frac{1}{\sqrt{\left[1+\left(\frac{\omega}{\omega_c}\right)^{2n}\right]}}$$

→ n : +ve integer

w_c : 3 dB cut off freq

3 dB : Also called $\frac{1}{2}$ power pt

✡ WHAT ARE WE DOING?

For any signal applicⁿ, a particular type of signal is reqᵈ. So, a particular type of freq is reqᵈ.

These filters have specific charac & we want to choose a specific filter for our reqᵈ charac

---

Normalising :-

$$|H_{Br}(j\omega)| = \frac{1}{\sqrt{\left[1+\left(\frac{\omega}{\cancel{\omega_c}}\right)^{2n}\right]}}$$

Note ✡ Increase order, increase ideality

✡ S-domain TF of BF is $|H_{Bn}(j\omega)|^2 = \frac{1}{1+\omega^{2n}}$.

$\boxed{j\omega = s}$

$$|H_{Bn}(s)|^2 = \frac{1}{1+(s/j)^{2n}} = \frac{1}{1+(-s^2)}$$

Considering LHS & RHS of s-plane $\left\{ \text{Or } H_{Bn}(s) \cdot H_{Bn}(-s) = \frac{1}{1+(-1)^n s^{2n}} \right.$

$D_n(s)$: factorise, consider roots in LHS for physically realisable filter fⁿ.

* Polynomials corresponding to each order

n = 1

$$H_{B1}(s)\,H_{B1}(-s) = \frac{1}{1-s^2} = \frac{1}{(1+s)(1-s)}$$

≡ Butterworth TF for n=1

$$\Rightarrow H_{B1}(s) = \frac{1}{1+s}\,.$$

n = 2

$$H_{B2}(s)\,H_{B2}(-s) = \frac{1}{1+s^4} = \frac{1}{(1+\sqrt{2}s+s^2)(1-\sqrt{2}s+s^2)}$$

$$\Rightarrow H_{B2}(s) = \frac{1}{(s^2+\sqrt{2}s+1)}$$

||ly , n = 3,

$$H_{B3}(s) = \frac{1}{(1+s)\,(1+s+s^2)}$$

— — — — & so on

Q Design a butterworth filter for a signal with amp. 1 dB & cut off freq 3 KHz

Stop band Attenuation should be < 1 dB at 1 KHz

Pass band Attenuation should be > 60 dB at 8 KHz

Idea :- We have 3 pts. We want to find the order of filter which will fit the box given by these 3 pts.

Corresponding to order, we can get the normalised polynomial.

→ has normalised freq.

So, we have to denormalise it to suit our freq.

Butterworth filters

Order	Polynomial
1	$s+1$
2	$s^2+1.414s+1$
3	$(s+1)(s^2+s+1)$
4	$(s^2+0.7654s+1)(s^2+1.8478s+1)$
5	$(s+1)(s^2+0.618s+1)(s^2+1.618s+1)$
6	$(s^2+.5176s+1)(s^2+1.4142s+1)(s^2+1.939s+1)$

✵ Finding pole loc^{ns}

loc^n of m^{th} pole, $S_m = e^{j(2m-1)\frac{\pi}{2n}} \cdot e^{j\frac{\pi}{2}}$ $\left(\because s^k = m, s = m^{1/k}\right)$

$$= j e^{j(2m-1)\frac{\pi}{2n}} \text{ or } e^{j\left((2m-1)\frac{\pi}{2n}+\frac{\pi}{2}\right)}$$

$$= \sigma_m + j\omega_m$$

Solving & getting : $1\left[\underbrace{(2m-1)\left(\frac{\pi}{2n}+\frac{\pi}{8}\right)}\right]$ $\rightarrow m = 1, 2, \ldots 2n$

by Euler's thm

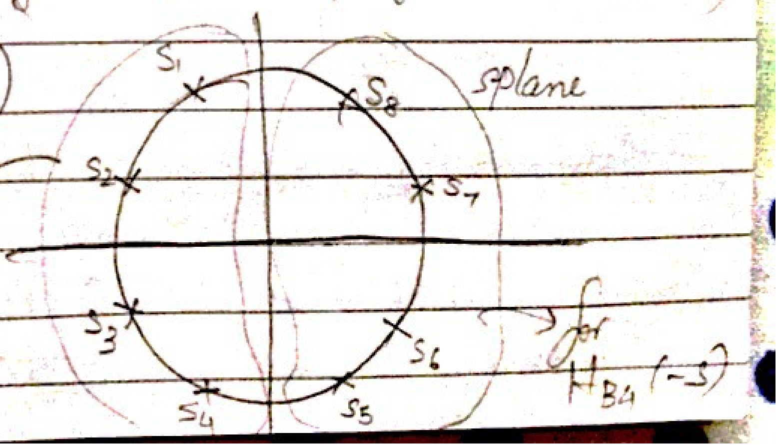
diff^t values of m gives diff^t loc^n of poles.

(generally, order of sys = no. of poles)

✵ Consider n (order = 4)

Loc^n of poles :

for $H_{B4}(S)$



* BF Design: Filter design means determining the coeffs of polynomial of filter f^n which will satisfy the specific^n.

Q Specific^n:- ① Attenuation atleast 10 dB at $2\omega_c$
② cut off freq, $f_c = 300$ kHz

Sol^n:- $-20\log_{10}|H_{Bn}(j\omega)| = -20\log_{10}(1+\omega^{2n})^{-1/2}$

$= 10\log(1+\omega^{2n})$

As per specs, $10\log(1+\omega^{2n}) \geq 10$ or $\log(1+\omega^{2n}) \geq 1$

For normalis^n, make $\omega = 1$
As per specs, $\omega \rightarrow 2\omega_c$ So, make $\omega = 2$.

$\therefore (1+\omega^{2n}) \geq 10$ at $2\omega_c$ (normalised, $f=2$)
$\therefore (1+2^{2n}) \geq 10 \Rightarrow n = 1.584$.
↳ should be Z.
So, $n = 2$ (assume)

So, order of filter $= 2$.
for order $= 2$, polynomial (normalised)

$$H_{B2}(s) = \frac{1}{s^2+\sqrt{2}s+1}$$

As per spec ②, $f_c = 300$ kHz,
$\omega_c = 1.89 \times 10^3$ rad/s

Changing $s \rightarrow s/\omega_c$ (denormalising it),

$$\Rightarrow H_{B2}(s/\omega_c) = \frac{1}{(s/\omega_c)^2+\sqrt{2}(s/\omega_c)+1} = \frac{1}{2.8\times10^{-7}s^2+.7482\times10^{-3}s+1}$$

So, coeff.s are $A = 2.8 \times 10^{-7}$
$B = 0.7482 \times 10^{-8}$

(b) CHEBYSHEV FILTERS :- (CF)

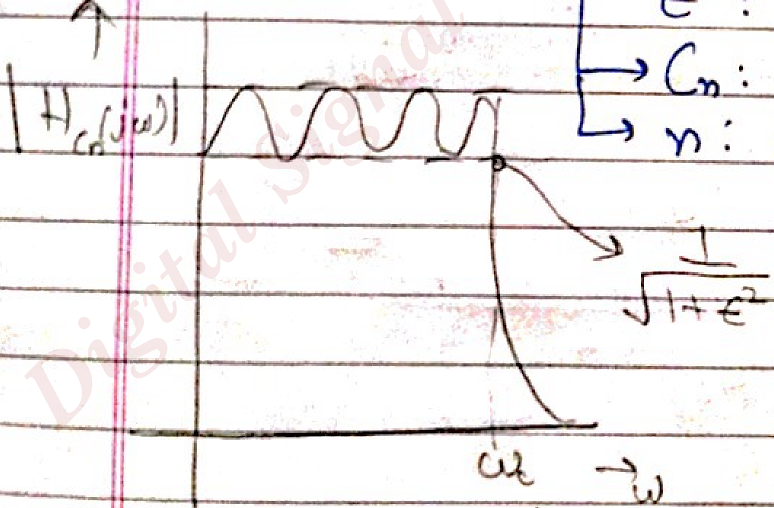
BF gives better amplitude response at nearly $\omega = 0$, but poor response at the cut off freq.
→ Pass band : Oscillatory responce ⇒ sin/cos
Stop band : Monotonic response ⇒ by linear f

On the other hand, CF gives better cut off freq. response but poor (oscillating) amplitude response below cut off freq.

CF is given by :-

$$\left| H_{cn}(j\omega) \right| = \frac{1}{\sqrt{1 + \epsilon^2 C_n^2(\omega)}}$$

→ ϵ : ripple factor $= 1$ (if not given)
→ C_n : n^{th} ord. Chebyshev Poly.
→ n : order of filter



$n \equiv$ total no. of maximas or minimas in graph

Amp. response of CF.

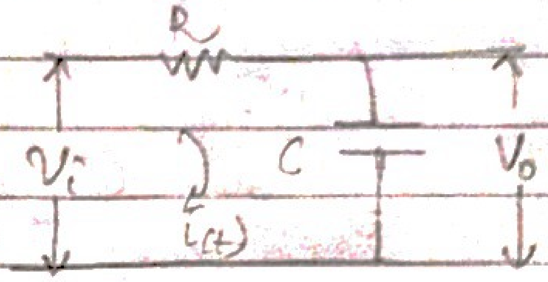
Polynomial : $C_n(\omega) = \cos(n \cos^{-1}\omega)$; $0 \leq \omega \leq 1$ below ω_c — for pass band
$= \cosh(n \cosh^{-1}\omega)$; $\omega \geq 1$ above ω_c — for stop band

normalised freq. w.r.t cut off freq.

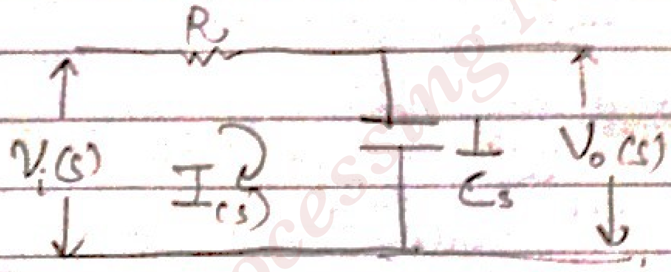
* For zero ord. filter :- ∃ only resistance

Idea: find coeff. of the polynomial in s (denominator)
↳ Make poly. of each order, make TF, relate with original TF, find coeff

* Extra :- Consider a ckt



↳ convert to s domain



So, $V_o = \frac{I(s)}{Cs}$

$V_i = \left(R + \frac{1}{Cs}\right) I(s)$

So, $TF = \frac{\frac{I(s)}{Cs}}{I(s)\left(R + \frac{1}{Cs}\right)} = \frac{1}{1+\underbrace{RCs}_{\tau}} = \frac{1}{1+\tau s}$

(time constt form)

* Polynomials corresponding to each order :-

(i) Let $\cos^{-1}\omega = \theta$; $\omega = \cos\theta$ below ω_c

$C_n(\omega) = \cos n\theta$ ∴ n can be zero also.

DEA

✱ Time constt form $\dfrac{(\ \ )}{(\ )s^n + (\ )^{n-1} + \ldots (\ )s + \frac{1}{\tau}}$

eg :- $TF = \dfrac{1}{1 + 0.2s}$ (time constt form)

$\Rightarrow T = 0.2 = RC$

So assume any value of R & C

So, what to take?

R = 1, C = 0.2 ? or anything else?

C = 0.2F is very big, so, not practical & ||ly, we deal with ckt of R = 1 kΩ or 1 MΩ

Here, R = 1 Ω, very less, not good

$\therefore C_0(\omega) = 1$; $C_1(\omega) = \cos\theta = \omega$.

$C_2(\omega) = \cos 2\theta = 2\cos^2\theta - 1 = 2\omega^2 - 1$

& so on, - - -

✱ Chebyshev's Poly.

1 → ω

Order (n)	$C_n(\omega)$ Poly.
3	$4\omega^3 - 3\omega$
4	$8\omega^4 - 8\omega^2 + 1$
5	$16\omega^6 - 20\omega^3 + 5\omega$
6	$32\omega^6 - 48\omega^4 + 18\omega^2 - 1$
7	$74\omega^7 - 112\omega^5 + 57\omega^3 - 7\omega$

Steps :- find order, see polynomial from above table, find TF from formula

Under Normalised

$\xrightarrow{}$ cut off freq.

* When $\omega > \omega_c$

$\cosh^{-1}\omega = x$; $\omega = \cosh x = \frac{e^x + e^{-x}}{2}$.

for $x >> 1$, $\omega = \frac{e^x}{2}$.

or $2\omega = e^x$

$\Rightarrow \boxed{x = \ln 2\omega}$; $C_n(\omega) = \cosh(n\cosh^{-1}\omega) = \cosh nx$

$= \cosh(n\ln 2\omega) = \cosh(\ln(2\omega)^n)$

for $\omega >> 1$; $C_n(\omega) = \frac{e^{\ln(2\omega)^n}}{2} = \frac{1}{2}(2\omega)^n$.

eg Design a filter s.t.

$$|H_{C_n}(j\omega)|^2 = \frac{1}{1+\epsilon^2 C_n^2(\omega)}$$

$$|H_{C_n}(s)|^2 = \frac{1}{1+\epsilon^2 C_n^2(s/j)} = H_{C_n}(s) \cdot H_{C_n}($$

$\to s = j\omega$

Chebyshev Poly. is $1+\epsilon^2 C_n^2(s/j) = 0$

or $C_n(s/j) = \pm j/\epsilon$

for pass band, $0 \leq \omega \leq 1$; $\cos\left[n\cos^{-1}\left(\frac{s}{j}\right)\right] = \pm \frac{j}{\epsilon}$

let $\cos^{-1}\left(\frac{S}{j}\right) = \alpha - j\beta$ ∴ $\cos(n\alpha - jn\beta) = \pm\frac{j}{\epsilon}$

//

Real $+ j$ Imaginary $= \pm\frac{j}{\epsilon}$

Now,

$\Rightarrow 0 + j\left(\pm\frac{1}{\epsilon}\right) = \pm\frac{j}{\epsilon}$

$\cos j\theta = \cosh\theta$ & $\sin j\theta = j\sinh\theta$.

Hence, $\cos n\alpha \cosh n\beta + j\sin n\alpha \sinh n\beta \pm \frac{j}{\epsilon}$

equating real & imaginary parts -

$\Rightarrow \cos n\alpha \cosh n\beta = 0$ & $\sin n\alpha \sinh n\beta = \pm\frac{j}{\epsilon}$

But $\cosh n\beta \neq 0$.

$\Rightarrow \cos n\alpha = 0 \Rightarrow 2\alpha = (2m-1)\frac{\pi}{2} \Rightarrow \alpha = (2m-1)\frac{\pi}{2n}$

& $\sin n\alpha = \sin(2m-1)\frac{\pi}{2}$; $m = Z^+$ $(1, 2, 3, \ldots)$

$\Rightarrow \sinh n\beta = \pm\frac{1}{\epsilon}$

∴ $\beta = \pm\frac{1}{n}\sinh^{-1}\left(\frac{1}{\epsilon}\right)$

Roots of Cheby. Poly is $\cos^{-1}\left(\frac{S}{j}\right) = \alpha - j\beta$.

or $S_m = j\cos(\alpha - j\beta)$

$= -\sin\alpha \sinh\beta + j\cos\alpha\cosh\beta$

Substituting for α & β .

$$s_m = -\left[\underbrace{\sin(2m-1)\frac{\pi}{2n}\cdot\sinh\left(\frac{1}{n}\sinh^{-1}\left(\frac{1}{\epsilon}\right)\right)}_{\sigma_m} + j\underbrace{\cos(2m-1)\frac{\pi}{2n}\cdot\cosh\left(\frac{1}{n}\sinh^{-1}\left(\frac{1}{\epsilon}\right)\right)}_{\omega_m}\right]$$

$$\Rightarrow S_{\boxed{m}} = \sigma_m + j\omega_m .$$

mth pole is given by this eqn

$$\text{Hence, TF, } H_{Cn}(s) = \frac{k}{(-1)^n \prod_{m=1}^{n}\left(\frac{s}{s_m}-1\right)}$$

→ in normalised ω or s .

$\therefore \omega_c = 1$

Prev. Knowledge:-

$$\underbrace{\frac{1}{(s-a)(s-b)\ldots}}_{\text{Pole zero form}} = \frac{1}{abc\left(\frac{s}{a}-1\right)\left(\frac{s}{b}-1\right)\ldots} = \overbrace{\frac{1}{abc\,(T_1s-1)(T_2s-1)\ldots}}^{\text{Time constt. form}}$$

abc → open loop gain

Denormalising,

If ω is other than $\omega_c = 1$

$$H_{C_n}(s) = \frac{k}{(-1)^n \prod_{m=1}^{n} \left(\frac{s}{S_m \omega_c} - 1 \right)} \quad \left(\text{Replacing } s \rightarrow s/\omega_c \text{ in prev. formula} \right)$$

Within pass band, $C_n(\omega)$ will have maximas & minimas values at +1 & −1 & beyond ω_c, the response monotonically decreases.

∴ Irrespective of order 'n',

$$C_n(\omega)\Big|_{max} = 1 \quad \therefore \text{Min. value of } \left| H_{C_n}(j\omega) \right| = \frac{1}{\sqrt{1+\epsilon^2}}$$

$$\therefore \text{Ripple}_{P-P} = \gamma = 1 - \frac{1}{\sqrt{1+\epsilon^2}} \rightarrow \text{in dB}(\gamma),\ 20\log(1+\epsilon^2)^{-1/2}$$

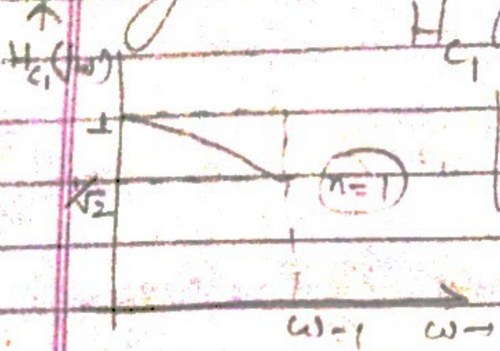
$$\Rightarrow \boxed{\text{Ripple}_{P-P} = -10\log(1+\epsilon^2)}$$

↳ For a change of y dB, ripple factor can be found as :- $y = -10\log(1+\epsilon^2)$

* No. of ripples : i.e. no. of maxima & minima is equal to order of the filter and independent of ϵ.

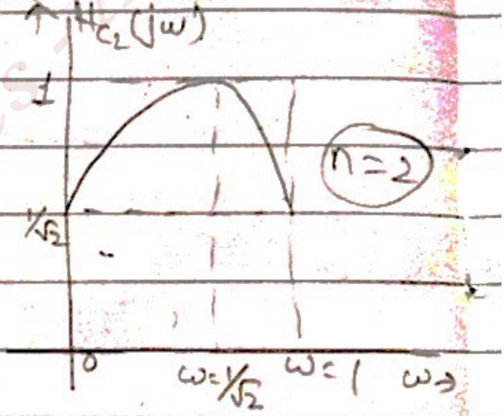
Let $\epsilon = 1$, So, $\left| H_{C_n}(j\omega) \right| = \frac{1}{\sqrt{1 + C_n^2(\omega)}}$

for $n=1$, $H_{C_1}(j\omega) = \frac{1}{(1+\omega^2)^{1/2}}$



→ At $\omega=0$, $|H_{C_1}| = 1$ (maxima)
$\omega = 1$, $= 1/\sqrt{2}$ (minima)
(cut off freq)

Inference :- The response monotonically attenuates from 1 to $1/\sqrt{2}$. ⇒ one ripple for $n = 1$.



for $n=2$, $C_2 = 2\omega^2 - 1$

$$|H_{C_2}(j\omega)| = \frac{1}{\sqrt{1+(2\omega^2-1)^2}}$$

→ $\omega=0$, $H_{C_2} = 1/\sqrt{2}$ (minima)
$\omega = 1$ $= 1/\sqrt{2}$ minima

Its max. value occurs at $\omega = 1/\sqrt{2}$ (∵ den. becomes 0 at that pt.)

⇒ one maxima & one minima at $n=2$

So, from the curve given, order of Chebyshev's poly. can be known.

Normalised freq (ω_n) = $\frac{\omega}{\omega_c}$

eg Specific^{ns} given :- (i) Ripple in pass band = 1 dB.
(ii) ω_c = 3 KHz
(iii) Amplitude attenuation atleast 20 dB at 6 kHz

To find : Design chebyshev Filter for specs given

Idea : Find ϵ, n.
Then, TF ✓

Spec. 1

$$10 \log (1+\epsilon^2) = 1 \Rightarrow \boxed{\epsilon = 0.51}$$

Spec 2

$$\omega_c = 3 kHz$$

$\therefore$ Normalised, $\omega = \frac{6}{3} = 2$.

Spec 3

$$|H_{Cn}(j\omega)| dB = 10 \log \left[1 + 0.51^2 C_n^2(2)\right] \geq 20 dB$$

$$\Rightarrow C_n^2(2) \geq 382.$$

→ $n=1$, $C_1 = \omega$.
$\Rightarrow \omega^2 = 2^2 = 4 \cancel{\geq} 382$

→ $n=2$.
$C_2 = 49 \cancel{\geq} 382$

→ $n=3$.
$$C_3 = (4\omega^3 - 3\omega)^2$$
$$= (4\times8 - 3\times2)^2 = 676$$
≥ 38 ✓

So, order = 3.

Now, TF $H_{c_3}(s)$ is to be found out from s_1, s_2 & s_3.

$$S_m = -\sin\frac{(2n-1)\pi}{6}\sinh\left(\left(\frac{1}{3}\right)\sinh^{-1}\left(\frac{1}{0.5}\right)\right)$$

$$+ j\cos\frac{(2m-1)\pi}{2}\cosh\left[\left(\frac{1}{3}\right)\cosh^{-1}\left(\frac{1}{0.5}\right)\right]$$

$$s_1 = -0.2471 + j(0.956) \leftarrow$$

$$S_2 = -0.4942$$

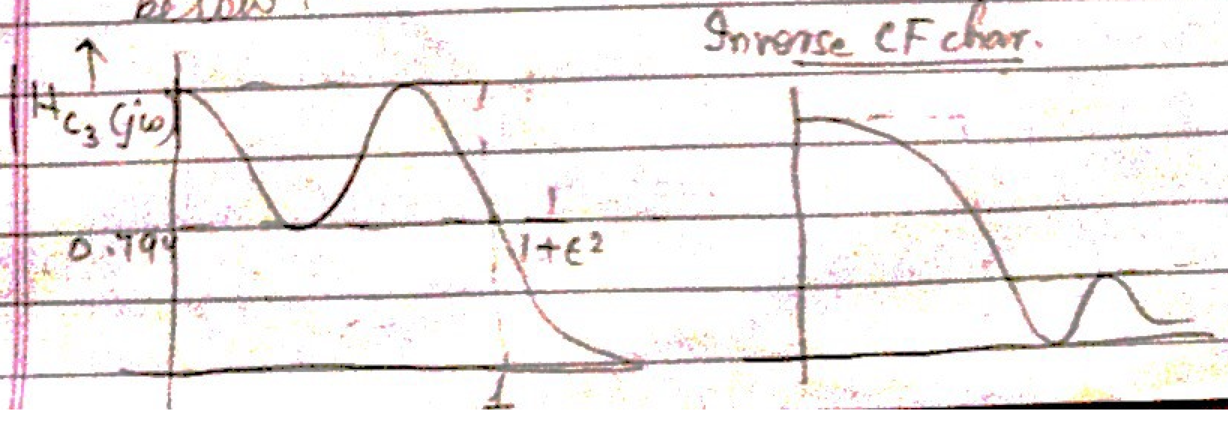
$$S_3 = -0.2471 - j0.966$$ & Complex conjugates (occur in pairs)

$$\therefore H_{c_n}\left(\frac{s}{\omega_c}\right) = \frac{k}{(-1)^n \prod_{m=1}^{n}\left(\frac{s}{S_m\omega_c} - 1\right)}$$

$\hookrightarrow$ for $k = 1$ & $\omega_c = 2\pi \times (3 \times 10^3)$ rad/s

$$H_{c_3}\left(\frac{s}{\omega_c}\right) = \left[\frac{0.4913}{\left(\frac{s}{6\pi\times10^3}\right) + 0.988\left(\frac{s}{6\pi\times10^3}\right)^2 + 1.23618\left(\frac{s}{6\pi\times10^3}\right) + 0.4913}\right]$$

So, the normalised magnitude for $H_{c_3}(s)$ will be shown below :-



* Change in freq. from pass band to stop band
  - ↳ easy in Chebyshev.
  - ↳ not that possible in Butterworth

✲ BESSEL FILTER

↳ which gives max. linear phase response.

↳ An all pole filter with

TF, $H_{BE_n}(s) = \frac{1}{BE_n(s)} = \frac{1}{\text{Bessel Poly.}}$

↳ $BE_n(s) = \sum_{k=0}^{n} \frac{(2n-k)!\, s^k}{2^{n-k}.\,k!\,(n-k)!}$

, where $\sum_{k=0}^{n} \frac{(2n-k)!}{2^{n-k}.\,k!\,(n-k)!} = a_k$

∴ $BE_n(s) = \sum_{k=0}^{n} a_k s^k$

↳ $BE_0(s) = 1$

↳ $BE_1(s) = s+1$

↳ $BE_2(s) = s^2 + 2s + 3$

↳ $BE_3(s) = s^3 + 6s^2 + 15s + 15$ - - - etc

* 1st ord. Bessels filter TF is $H_{BE_1}(s) = \frac{1}{1+s}$.

⇒ $H_{BE_1}(j\omega) = \frac{1}{1+j\omega} \times \frac{(1-j\omega)}{(1-j\omega)}$

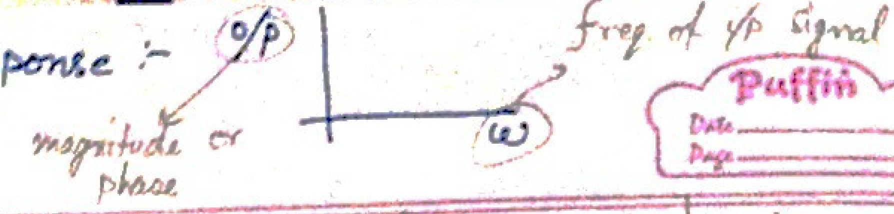
$= \frac{1}{1+\omega^2} - j\left(\frac{\omega}{1+\omega^2}\right)$

Phase angle = $\phi_1(\omega) = \tan^{-1}(-\omega) = -\tan^{-1}(\omega$

↳ $\frac{Im(\omega)}{Re(\omega)}$

Phase delay = $-\frac{(\text{Phase angle})}{\omega} = \frac{\tan^{-1}(\omega)}{\omega}$.

* Freq. response :-



Group delay or time delay $= -\frac{d}{d\omega}\phi(\omega)$

$= \frac{d}{d\omega}\left(\tan^{-1}(\omega)\right)$

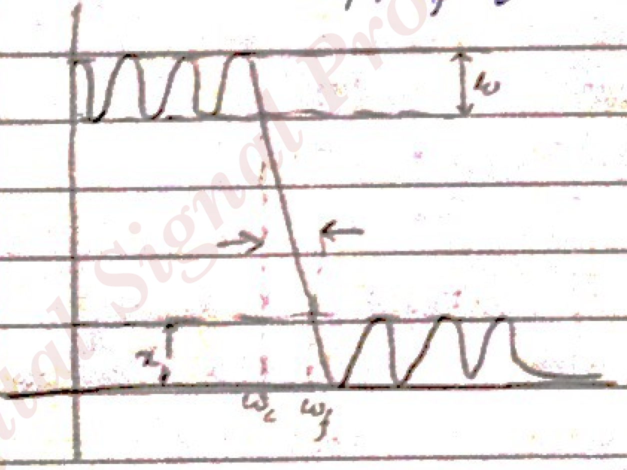
⇒ Group Delay $(T_p) = \frac{1}{1+\omega^2}$

↳ for a TF $= \frac{1}{1+s}$

||ly,
Phase delay & group delay can be known ∀ TF$_s$

continued →

3) ☆ ELLIPTICAL FILTER
↳ variation both in pass band & in stop band ⇒ trans^n band is less ⇒ ideal toward cut off freq.



Note
Order ↑ ⇒ no. of energy storage devices (L, C) ↑
⇒ Cost ↑

	* Butterworth	Chebyshev	Elliptical	Bessel
Order(n)	Char. b/w elliptical & Bessel (BF > CF)		lowest ⇒ (min cost)	Max.
Phy. Char.	linear over 3/4th freq range		Highly sluggish	Max.
Trans^n band	b/w elliptical & Bessel (BF > CF)		Min	Max

* Nyquist theorem :- Sampling freq. for any signal (with freq, f) . =
Sampling freq = 2f

Considering 2nd ord. Bessel's fn :-

$$H_{BE_2}(s) = \frac{1}{s^2+3s+3}$$

S1) $s \rightarrow j\omega$

S2) Divide into real & imaginary part

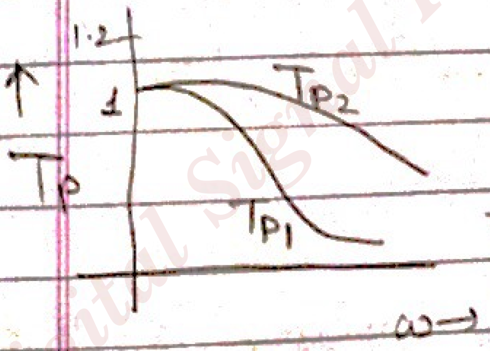
$$\Rightarrow H_{BE_2}(j\omega) = \frac{1}{(3-\omega^2)+j(3\omega)} = \frac{3-\omega^2 - j3\omega}{(3-\omega^2)^2+9\omega^2}$$

$$= \frac{(3-\omega^2)}{(3-\omega^2)^2+9\omega^2} - j\frac{(3\omega)}{(3-\omega^2)^2+9\omega^2}$$

$$-\phi_2(\omega) = \tan^{-1}\left(\frac{3\omega}{3-\omega^2}\right)$$

$$\therefore T_{P_2} = \text{group delay} = \frac{d}{d\omega}\phi_2(\omega)$$

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



$$\Rightarrow T_{P_2} = \frac{9+3\omega^2}{9+3\omega^2+\omega^4}$$

* FREQUENCY TRANSFORMATION

The normalised freq (s) used so far was the prototype.

Actual = S_T

$\overset{\text{prototype}}{}$

Case ① :- LP to LP

$s = \frac{S_T}{\omega_c}$; $\omega = \frac{\omega_T}{\omega_c}$

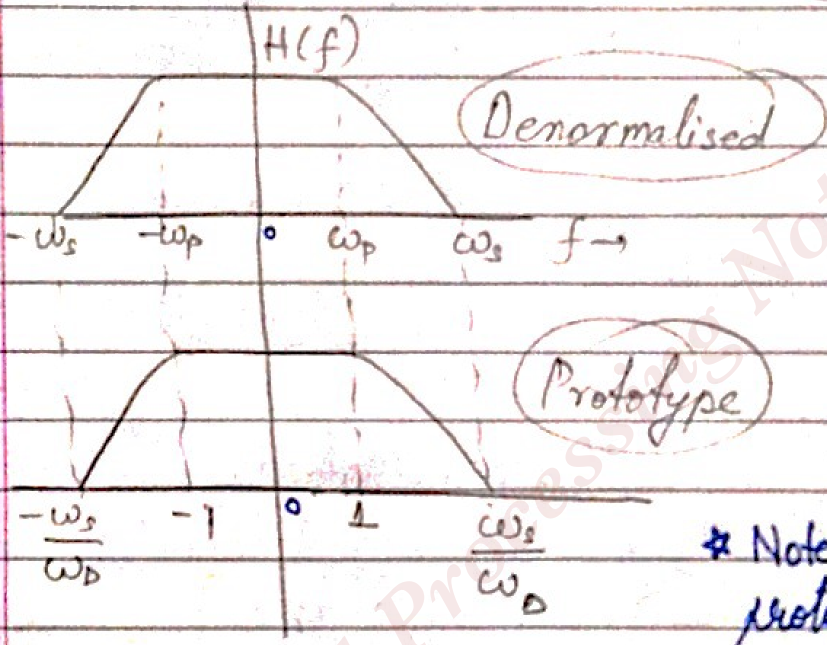
$\omega_{lp} = 0 \rightarrow \omega^{p} = 0$

lp → Low pass

$\omega_{lp} = \omega_p \rightarrow \omega^{p} = 1$

$\omega_{lp} = \omega_s \rightarrow \omega^{p} = \frac{\omega_s}{\omega_p}$

(where pass band & stop band edge freq. are :- ω_s & ω_p)



* Note: Critical freq. of prototype filters are 0, 1, $\frac{\omega_s}{\omega_p}$

Case ② : LP to HP

↳ can be obtained by replacing $s \rightarrow 1/s$ in the TF. ∴ $H(s) = G(s)\Big|_{s = \frac{1}{S_T}}$

transformed ʔ

* H.P filter freq ω_{hp}, prototype LP filter ω^{p}.

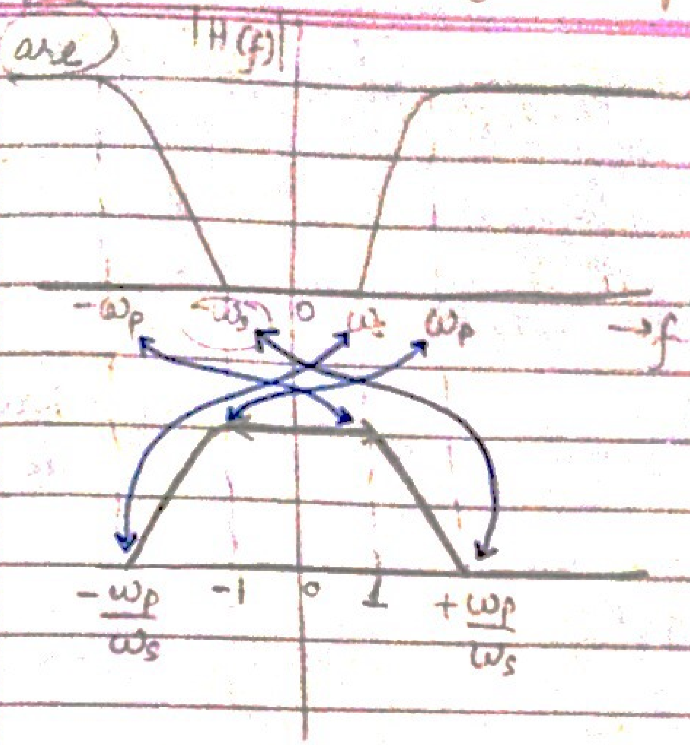
Relⁿ b/w LP prototype freq. & dimensional hp

* For passing low ʔ :- use X_L ($\propto \omega$)

For passing high ʔ : use X_C ($\propto \frac{1}{\omega}$)

$R = \omega L = X_L \rightarrow$ Series. → low pass

$= X_L \rightarrow$ in parallel



Prototype

1. $\omega_{hp} = 0 \rightarrow \omega^p \to \infty$

pass band freq

2. $\omega_{hp} = \omega_p \rightarrow \omega^p = -1$

3. $\omega_{hp} = \omega_s \rightarrow \omega^p = -\frac{\omega_p}{\omega_s}$

4. $\omega_{hp} = -\omega_p \rightarrow \omega^p = 1$

5. $\omega_{hp} = -\omega_s , \omega^p = \frac{\omega_p}{\omega}$

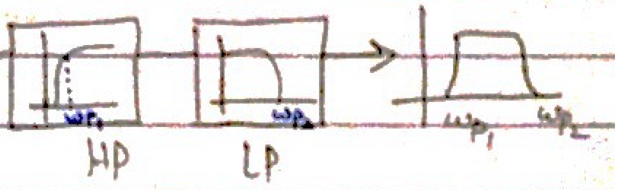
★ Note $\frac{1}{s} = \frac{1}{j\omega} = \frac{-j}{\omega}$

So, +ve pass band edge ⁊ correspond to −ve prototype ⁊ & vice versa

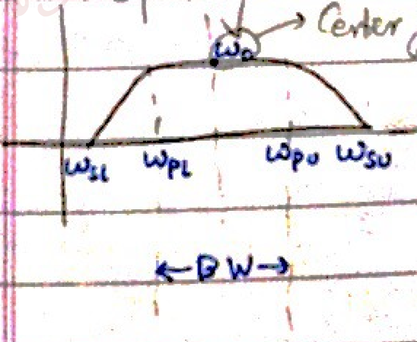
Case ③ : LP to Band Pass filters

→ combin^n of HP & LP

★ Designing Band pass filter :-



Defining 4 ⁊s :-



→ Center freq $(\omega_{SL}, \omega_{SU}, \omega_{PL}, \omega_{PU})$

→ Passband, upper freq

→ Stopband, lower freq.

$$S = \frac{\omega_0}{\omega_b}\left(\frac{S_T}{\omega_0} + \frac{\omega_0}{S_T}\right)$$

$$= \frac{S_T^2 + \omega_0^2}{\omega_b S_T}$$

- ω_{PL} : lower cut off freq. of pass band
  ω_0 : center freq.
  S_T : transformed S.

& $\omega_0^2 = \omega_{PL} \cdot \omega_{PU}$. $\because j^2 = -1$

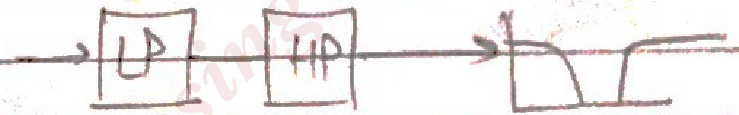
i.e, $\omega = \frac{\omega_T^2 - \omega_0^2}{\omega_T \cdot \omega_b}$

comes from $S_T = j\omega_T$

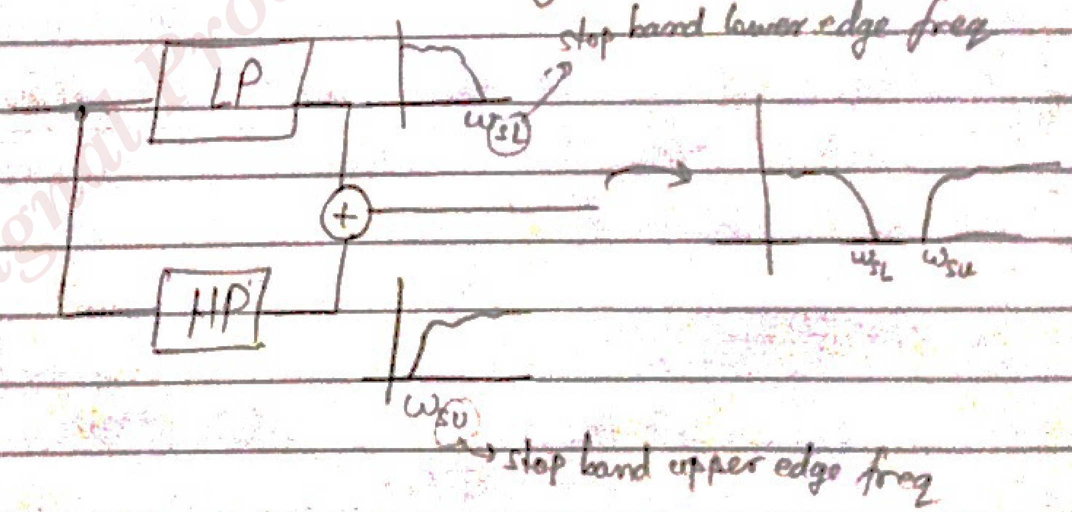
$\boxed{\omega_b = BW = \omega_{PU} - \omega_{PL}}$

Case ④ :- Band Stop filter from (HP & LP).

Idea :-



Connect LP & HP filter in PARALLEL

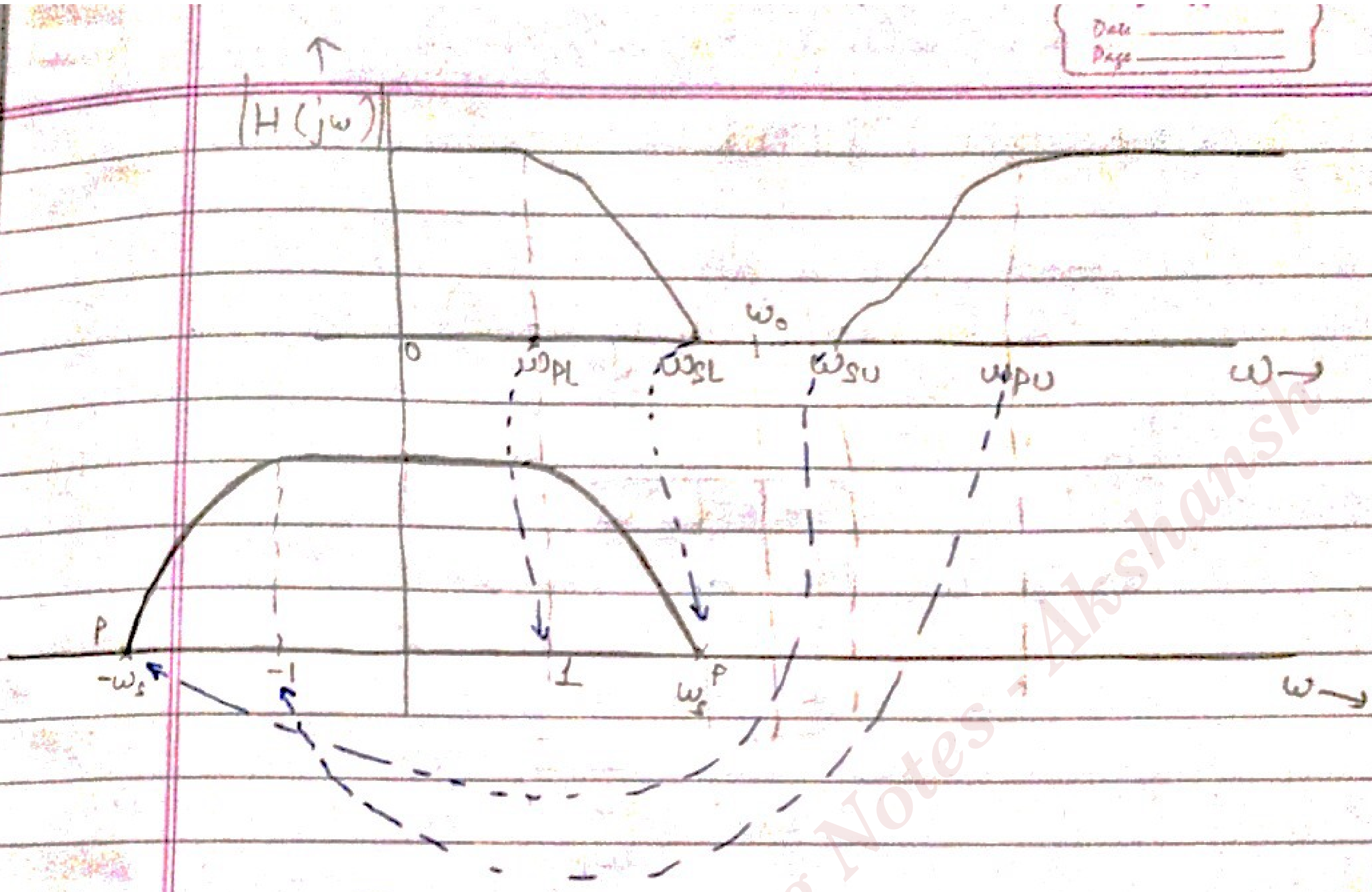


★ LP to Band stop transformⁿ

$$S \rightarrow \frac{S_T \omega_b}{S_T^2 + \omega_0^2} \quad ; \quad j\omega^P = \frac{j\omega_T \omega_b}{(j\omega_T)^2 + \omega_0^2}$$

$$\omega^P = \frac{\omega_b \cdot \omega_{bs}}{\omega_0^2 - \omega_{bs}^2}$$

band stop



★ SUMMARY

1) BUTTERWORTH FILTER

$$|H(j\omega)|^2 = \frac{1}{1+\omega^{2n}} \; ; \quad N \geq \frac{\log\left(\frac{10^{\frac{A_s}{10}}-1}{10^{\frac{A_p}{10}}-1}\right)}{2\log\left(\frac{\omega_s}{\omega_p}\right)}$$

For normalised filter:

$$S_k = e^{j\left[\frac{\pi(2k-1+N)}{2N}\right]} = \cos\theta + j\sin\theta$$

↳ $k = 1, 2, \ldots N$

↳ $\theta = \frac{\pi(2K-1+N)}{2N}$

Note ✻ If asked to design a filter, design a LP filter first (choosing from Butterworth, Cheby...) then denormalise it → finally, convert to reqd filter

2) CHEBYSHEV'S FILTER

Pass band ripple in dB

$$= -20\log_{10}(1-\delta_p) = 10\log_{10}(1+\epsilon^2)$$

$$|H(j\omega)| = \frac{1}{\sqrt{1+\epsilon^2 C_n^2 \omega}} ;$$

$$N \geq \frac{\cosh^{-1}\left[\dfrac{10^{\frac{A_s}{10}}-1}{10^{A_P/10}-1}\right]}{\cosh^{-1}\left(\dfrac{\omega_s}{\omega_p}\right)}$$

For normalized filter :-

$$S_K = \sinh(\alpha)\cos(\beta_k) + j\cosh(\alpha)\sin(\beta_k)$$

$$\hookrightarrow \alpha = \frac{1}{N}\sinh^{-1}\left(\frac{1}{\epsilon}\right)$$

$$\hookrightarrow \beta_K = \frac{\pi(2K+N-1)}{2N} \quad \hookrightarrow k = 1, 2, \ldots N$$

Note ★ If asked to design a filter, design a LP filter first (choosing from Butterworth, Cheby...) then denormalise it → finally, convert to reqd filter

2) CHEBYSHEV'S FILTER

Pass band ripple in dB

$$= -20\log_{10}(1-\delta_p) = 10\log_{10}(1+\epsilon^2)$$

$$|H(j\omega)| = \frac{1}{\sqrt{1+\epsilon^2 C_n^2 \omega}} ;$$

$$N \geqslant \frac{\cosh^{-1}\left[\dfrac{10^{\frac{As}{10}}-1}{10^{Ap/10}-1}\right]}{\cosh^{-1}\left(\dfrac{\omega_s}{\omega_p}\right)}$$

For normalized filter :-

$$S_k = \sinh(\alpha)\cos(\beta_k) + j\cosh(\alpha)\sin(\beta_k)$$

$$\rightarrow \alpha = \frac{1}{N}\sinh^{-1}\left(\frac{1}{\epsilon}\right)$$

$$\rightarrow \beta_k = \frac{\pi(2k+N-1)}{2N} \quad \hookrightarrow k = 1, 2, --N$$

* * *

eg Design a BP filter to meet the following specs:

(a) 3 dB attenuation at 10 k rad/s & 15 k rad/s.

(b) Attenuation more than 25 dB for freq. less than 5 k rad/s & more than 20 k rad/s.

Idea :- look for pass band & stop band edge freq.

For 3 dB attenuation, its pass band freq

So, $\omega_{pl} = 10\ k\ rad/s$

$\omega_{pu} = 15\ k\ rad/s$.

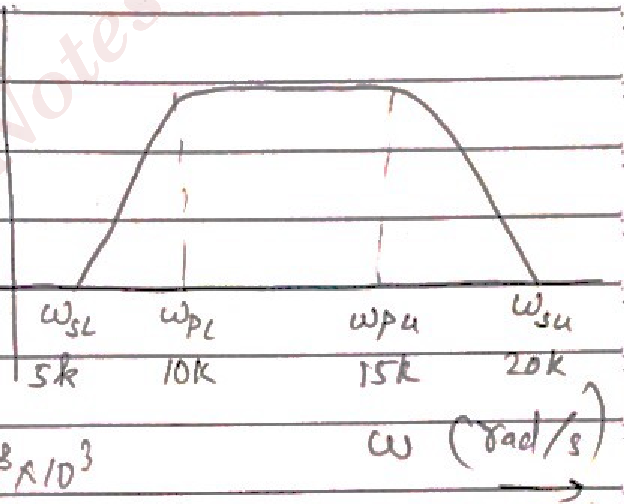
25 dB freq tells stop band freq

=> $\omega_{sl} = 5\ k\ rad/s$.

$\omega_{su} = 20\ k\ rad/s$.

Now,

freq. response :-



$$\omega_0^2 = \omega_{pl} \times \omega_{pu} = 10 \times 15 \times 10^3 \times 10^3$$
$$= 150 \times 10^6$$

$\omega_b = \omega_{sl} = 5\ k\ rad/s$

$$\Omega = \frac{\omega_T^2 - \omega_0^2}{\omega_T \cdot \omega_b} = \frac{\omega_T^2 - (150 \times 10^6)}{\omega_T\,(5 \times 10^3)}$$

2nd spec. says :- for $\omega_T > \omega_{su}$ & $\omega_T < \omega_{sl}$, attenuation is more than 25 dB

freq. (Ω) is assumed to be min. of the given stop band freqs. (∵, for higher freq, attenuation will be more than that of lesser freq.)

So,

normalised $\omega_s^p = \min\left\{\frac{(20\times10^3)^2-150\times10^6}{(20\times10^3)(5\times10^3)}, \frac{(5\times10^3)^2-150\times10^6}{(5\times10^3)(5\times10^3)}\right\}$

$\Rightarrow \omega_s^p = 5$ rad/s.

Assuming a Butterworth filter (for simplicity), the order has to be determined by TRIAL & ERROR

i.e, taking $n=1$.

$$HB_1(s) = \frac{1}{1+s} \Rightarrow HB_1(j\omega) = \frac{1}{1+j\omega}$$

$$= \frac{1}{1+j(5)}$$

Taking in dB

$$\Rightarrow \left|HB_1(j5)\right| = 20\log_{10}\left(\frac{1-}{\#}\right.$$

taking $n=2$.

$$HB_2(s) = \frac{1}{s^2+\sqrt{2}s+1} ; H_{B_2}(j\omega) = \frac{1}{(1-\omega^2)+j\sqrt{2}\omega}.$$

for $\omega=5$

$$H_{B_2}(j5) = \frac{1}{(1-25)+j\sqrt{2}5+1}$$

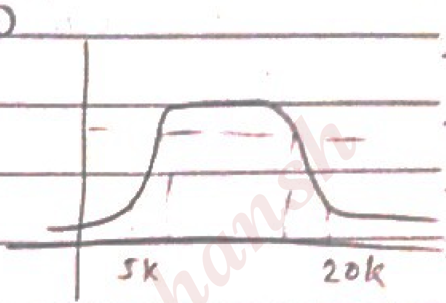
in (dB) :-

$$\left|H_{B_2}(j5)\right| = 20\log_{10}\left(\frac{1}{25}\right) = 30\,dB$$

∴ Its satisfying 25 dB requirement

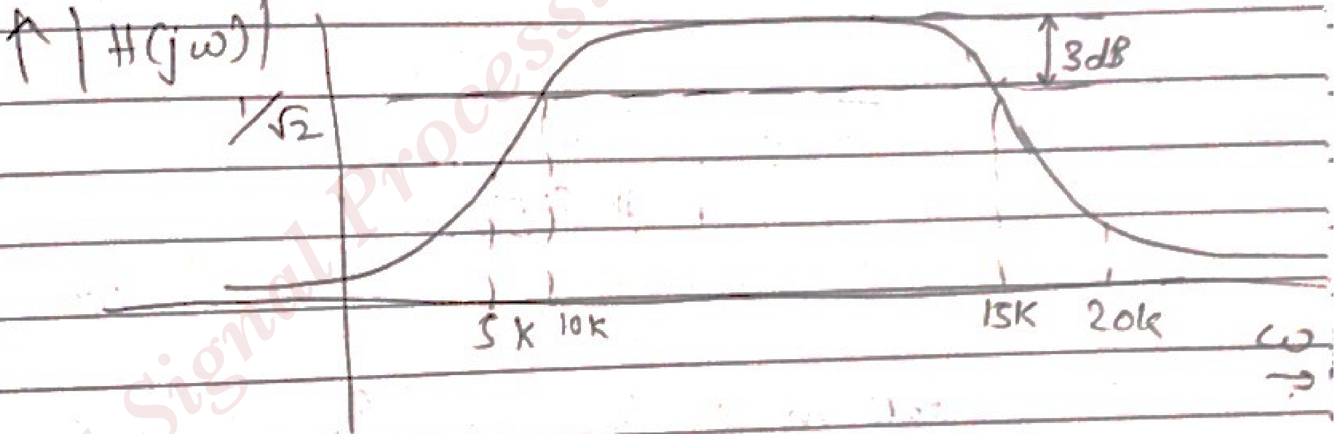
Now,

$$H_{B_{2T}}(j\omega_T) = \frac{1}{(1-\omega^2) + j\sqrt{2}\omega}$$



$$= \frac{1}{1-\left(\frac{\omega_T^2-150\times10^6}{5\times10^3\times\omega_T}\right)^2 + j\sqrt{2}\left(\frac{\omega_T^2-150\times10^6}{5\times10^3\times\omega_T}\right)}$$

$$= \frac{(5\times10^3)^2\times\omega_T^2}{\omega_T^4 + 275\times10^6\omega_T^2 - (150\times10^6)^2 + j\sqrt{2}\times5\times10^3(\omega_T^2-150\times10^6)}$$



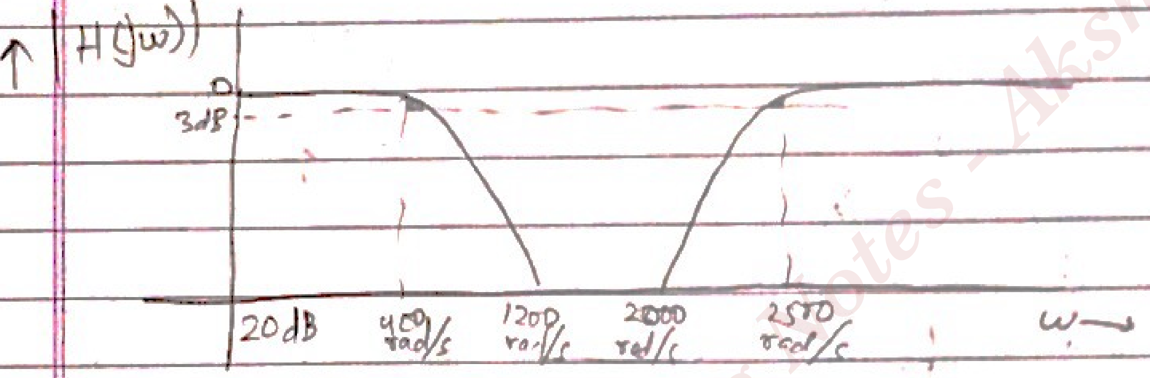
Observ^n :-

We took BF of ord = 2.
When transformed to BP filter, we get ord = 4.
(High Pass + low pass)
So, order is doubled.

eg Design a Band stop filter for following specs:

Stop (a) Attenuation between : 1200 rad/s & 2000 rad/s must be atleast 20 dB

Pass (b) Attenuation for less than 400 rad/s & higher than 2500 rad/s must be less than 3 dB



Transformⁿ is done for the specs:-

$$\omega = \frac{\omega_T (2000-1200)\omega_p}{-\omega_T^2 + (2000\times 1200)}$$

$$\Rightarrow \omega = \frac{800\,\omega_T \cdot \omega_p}{-\omega_T^2 + (2.4\times 10^6)}$$

ω_p: pass band edge freq. of BP filter (critical freq) w.r.t prototype LP filter. To determine the order of filter satisfying attenuation condⁿ.

Transforming 2500 & 400 to prototype freq :-

$\omega(2500) = -0.5195\,\omega_p$

$\omega(400) = 0.1429\,\omega_p$.

Attenuation has to be less than 3dB for $\omega \geq 0.5195\,\omega_p$ & $\omega < 0.1429\,\omega_p$.

$\therefore$ Choose ω_c of LP filter,

$$\omega_c = 0.5195\,\omega_p$$

prototype, $\omega_c = 1$ $\therefore$ ω_p of filter (BP)

$$= \frac{1}{0.5195} = 1.925 \text{ rad/s}$$

Also, prototype of filter should have an attenuation of more than 20 dB for $\omega \geq 1.925$ rad/s.
Based on this, determine the order of LP Butterworth filter i.e $n=4$ (after checking for given attenuation)

From table $\left\{ \therefore H_{B4}(s) = \dfrac{1}{(s^2 + 0.7654s+1)(s^2+1.8478s+1)} \right.$

Substitute :-

$$\omega = \frac{800 \times 1.925\,\omega_T}{2.4 \times 10^6 - \omega_T^2}$$

$$\Rightarrow H_{B_4T}(s) = \frac{1}{5.62 \times 10^{12} s_T^4 + 9.55 \times 10^7 s_T^3 + 3.36 \times 10^9 s_T^2 + 1464 s_T + 1}$$

Date 23/9/13
Page

* For converting analog ⟶ digital signal (approx.)

Sampling freq ⟶ ∞.

DESIGNING A DIGITAL FILTER

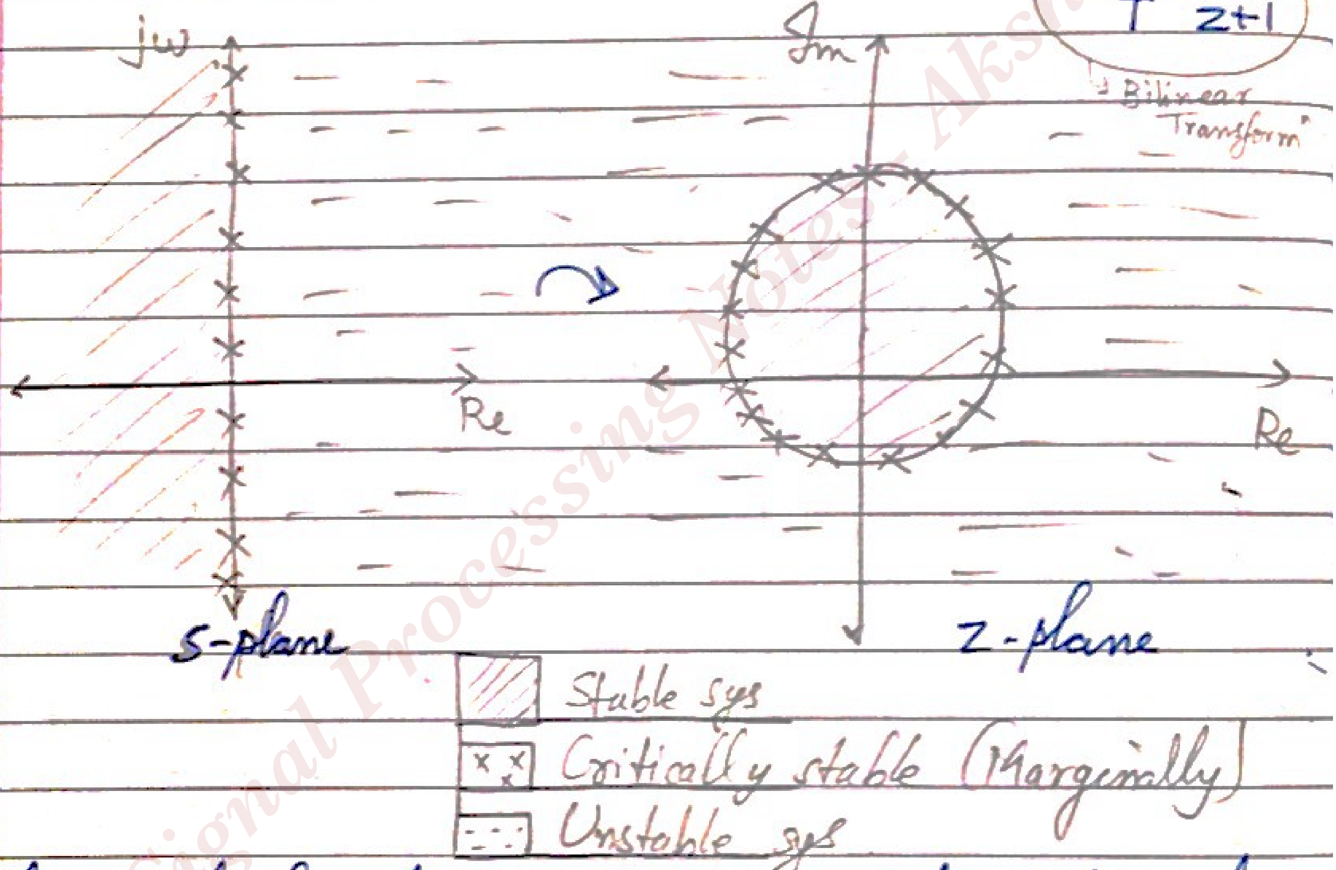
BILINEAR Z- TRANSFORM (BZT) METHOD:

Convert $H(s) \to H(z)$ by $H(z) = H(s)\Big|_{s = \frac{2}{T}\cdot\frac{z-1}{z+1}}$

Imp

$$\boxed{s = \frac{2}{T}\cdot\frac{z-1}{z+1}}$$

↳ Bilinear Transform



* In a digital domain, TF is written in z-domain (by z-transform)

* s-plane to z-plane mapping by Bilinear z-transform

$Z = e^{j\omega t}$; $s = j\omega'$; ω' : analog freq.

ωT = digital freq : represented as ω_d.

TP corresponding to sampling freq

$= \frac{1}{f_s}$

Substituting in $s = \frac{2}{T}\left(\frac{z-1}{z+1}\right) \Rightarrow \boxed{\omega' = \frac{2}{T}\tan\left(\frac{\omega T}{2}\right)}$

Whole of the left half of s-plane freq. is mapped to inside of unit circle i.e, digital freq. will be cramped up. This effect is compensated by Prewarping the analog filter freq. before the Bilinear Transform"

eg for a LP filter, either the cut off freq. or band edge freq. as follows:-

ω_p'

★ Steps to BZT & Methods :-

S1. Use digital filter specs and design a normalised prototype analog low pass filter, H(s).

S2. Determine the prewarp edge freq. of the desired filter. ω_c for LPF or HPF & for BPF & BSF, ω_s & ω_p.

as $\omega_c' = \tan\left(\frac{\omega_c T}{2}\right)$ & $\omega_p' = \tan\left(\frac{\omega_p T}{2}\right)$

$\omega_s' = \tan\left(\frac{\omega_s T}{2}\right)$

(not using $\times \frac{2}{T}$)

* Z-transform : & $Z[x(n)] = X(z)$

$$Z[x(n-1)] = z^{-1} X(z)$$

S3) Denormalise the analog prototype filter by replacing s in H(s) with one of the transformⁿ depending on the type of desired filter.

LPF to LPF $s = \frac{s}{\omega_p'}$

LPF to HPF $s = \frac{\omega_p'}{s}$

LPF to BPF $s = \frac{s^2 + \omega_0^2}{B(bw) \cdot s}$

LPF to BSF $s = \frac{(bw)s}{s^2 + \omega_0^2}$

S4) Apply BZT to obtain H(z) by replacing s, in the denormalised TF H'(s) as :-

$$H(z) = H'(s)\Big|_{s = \frac{z-1}{z+1}}$$

* * *

eg Given a 2nd order LPF TF (analog)

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

Design a digital 2nd LPF TF

Using BZT method, obtain TF ÷ H(z) of digital filter, assuming 3 dB cut off freq. of 150 Hz & a sampling freq of 1.28 kHz

Time domain	z-domain
$x(n)$ is one instant.	$X(z)$ is one instant
& $x(n-1)$ is prev. instant	$\Rightarrow z^{1}X(z)$ gives prev instant
$x(n+1)$ is next instant	$\Rightarrow z^{+1}X(z)$ gives next instant

Solⁿ :- Critical freq ; $\omega_p = 2\pi \times 150$ rad/s.

$$F_s = \frac{1}{T} = 1.28 \text{ kHz}.$$

$\Rightarrow$ Prewarped critical freq :-

$$\omega_p' = \tan\left(\frac{\omega_p T}{2}\right) = 0.3857$$

The freq - scaled analog filter is given by

$$H'(s) = H(s)\Big|_{s=\frac{s}{\omega_p'}} = \frac{1}{\left(\frac{s}{\omega_p'}\right)^2 + \sqrt{2}\left(\frac{s}{\omega_p'}\right) + 1}$$

$$= \frac{(\omega_p')^2}{s^2 + \sqrt{2}\,\omega_p' s + (\omega_p')^2}$$

$$= \frac{0.1488}{s^2 + 0.5455 s + 0.1488}$$

Applying BZT, gives

$$H(z) = H'(s)\Big|_{s=\frac{z-1}{z+1}} = \frac{0.0878 z^2 + 0.1756 z + 0.0878}{z^2 - 1.0048 z + 0.3561}$$

$$\Rightarrow H(z) = \frac{0.0878\left(1 - 2z^{-1} + z^{-2}\right)}{1 - 1.0048 z^{-1} + 0.3561 z^{-2}}$$

✯ $z^{(+ve\ no.)}$ = future prediction

$z^{(-ve\ no.)}$ = past instant ⇒ Physically realisable

So, its necessary to convert to $z^{(-ve)}$

→ samples

Drop T. So 1, 2, 3 are samples } discrete nos.

★ Note :- $Z^{-1}(Y(z)) = y(k)$ or $y(n)$ } any notation → discrete nos

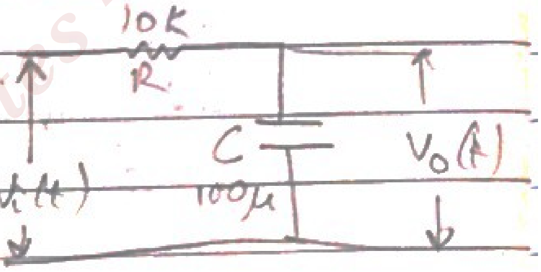
eg An analog LP RC filter's normalised TF

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

Starting from s-plane eqⁿ, determine (BZT method) TF of equivalent discrete-time highpass filter. Assume sampling freq of 150 Hz & cutoff freq of 30 Hz.

Critical freq, $\omega_p = 2\pi \times 30$.

$$\omega_p' = \tan\left(\frac{\omega_p T}{2}\right)$$



$$T = \frac{1}{150\ Hz}$$

$$\omega_p' = \tan\left(\frac{\pi}{5}\right)$$

$$H'(s) = H(s)\Big|_{s=\frac{\omega_p'}{s}} = \frac{1}{\left(\frac{\omega_p'}{s}\right)+1} = \frac{s}{s+0.7265}$$

LP to HP transformⁿ

Applying BZT

$$H(z) = H'(s)\Big|_{s=\frac{z-1}{z+1}} = \frac{(z-1)/(z+1)}{\left(\frac{z-1}{z+1}\right)+0.7265}$$

Simplifying, $H(z) = 0.5792\left(\frac{1-z^{-1}}{1+0.1584z^{-1}}\right)$

Coeff. of discrete time filter are

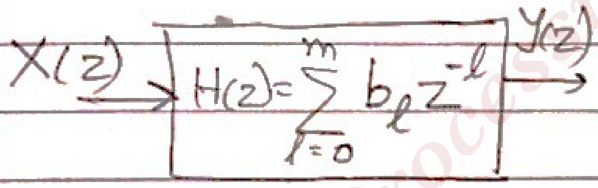
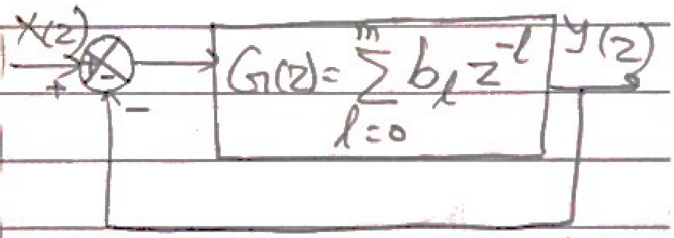
$b_0 = 0.5792$, $a_1 = 0.1584$

$b_1 = -0.5792$

If $x(n)$ = present i/p, say
$x(n-1)$ = one step back i/p
$x(n-2)$ = two step back i/p.

DIGITAL FILTERS

Devices that transform a set of input sequence ($x(n)$) into a set of o/p sequence without loss of info.

Finite Impulse Response ~~FL~~ FIR (finite duration)	Infinite Impulse Response IIR (Infinite dur^n)
An open loop filter whose o/p depends only on present & past i/ps	Its closed loop filter whose o/p depends not only on present & past i/ps, but also on PAST OUTPUTS
	
$TF = H(z) = \frac{Y(z)}{X(z)}$ $= \sum_{l=0}^{m} b_l z^{-l}$	$TF = H(z) = \frac{Y(z)}{X(z)} = \frac{G(z)}{1+G(z)}$ $H(z) = \left[\frac{\sum_{l=0}^{m} b_l z^{-l}}{1+\sum_{l=1}^{m} a_l z^{-l}} \right]$
non feedback	~~non~~ feedback
non recursive	~~non~~ recursive
Needs more ~~cof~~ coeff. for same set of specific^n.	Merit: It needs fewer coeff.
Less chances of instability.	Demerit: ∃ chances that sys. may go in unstable mode (∵ f/b sys).

* $z^{-1}(b_1 z^{-1}) = b_1 x(n-1)$: i.e, considering one instant before present i/p & sampling it at that time; fraction of that sample is taken as b_1

* Concepts of Digital Filter Design.

FIR IIR

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

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

Reason:-

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + \dots b_n z^{-n}}{1 + a_1 z^{-1} + \dots + a_M z^{-M}}$$

$$\Rightarrow z^{-1}\left(Y(z)\left[1 + a_1 z^{-1} + a_2 z^{-2} + \dots a_M z^{-M}\right]\right)$$

$$= z^{-1}\left(\left[b_0 + b_1 z^{-1} + \dots b_n z^{-N}\right] X(z)\right)$$

$$\Rightarrow y(n) + a_1 y(n-1) + \dots + a_M y(0)$$

$$= b_0 x(n) + b_1 x(n-1) + \dots + b_N x(0)$$

$$\Rightarrow y(n) = -\left[a_1 y(n-1) + \dots + a_M y(0)\right] + \left[b_0 x(n) + b_1 x(n-1) + \dots + b_N x(0)\right]$$

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

⇒ *Designing any filter ⇒ find coeff's a_k & b_k of num. & den.

☆ For a stable sys,
All poles must lie inside unit circle or coincident with zero's on unit circle.
No restriction on zero loc^n

☆ 5 Main STAGES of IIR filter Design

1. Filter specific^{ns}
2. Determin^n of a_k, b_k of $H(z)$
3. Realiz^n (parallel / cascade) using first order or second order filter section.
   - ✓ Any higher ord. sys can be implemented in series/parallel connection using different first order or 2nd order different sections.
     eg :- $\underbrace{H(z)}_{\text{ord }2} = \underbrace{H_1(z)}_{\text{ord }1} \cdot \underbrace{H_2(z)}_{\text{ord }1}$
   - ✓ why only 1st & 2nd ord. division is possible? → self
4. Analysis of errors based on represent^n of coeff$_s$ & arithmetic using limited no. of bits.
5. Implementation : Building hardware or software.

Suppose coeff$_s$ are 0.93579. This, in binary has to be represented by some fixed no. of bits
If, say we have a coeff = 0.27 and 0.28
Then, using 2 bits, the representation by '00' will give '0 - 0.25' range. Hence, the represent^n for coeff. 0.27 & 0.28 will be same i.e '00' In such case, error will occur.
If coeff are >1, write it as $0.4 \times e^{1}$ (say).

If we make 2 bit represent^n for $\frac{1}{4} = 0.25$:

0 – 0.25	00
0.25 – 0.5	01
0.5 – 0.75	10
0.75 – 1	11

*pole : freq. at which TF is max.

* Stage 2 : finding coeffs (a_k & b_k) for IIR filter
Methods

M1) Pole-zero placement : simple filter where filter specs need not be specified precisely
eg : notch filters

M2) Converting an apt analog filter TF to that of digital filter

(a) Impulse invariant
(b) Matched z-transform
(c) Bilinear z-transform

(M1) Pole-zero placement

Corresponding to a zero :- freq response is zero.
pole :- freq response will give a peak.

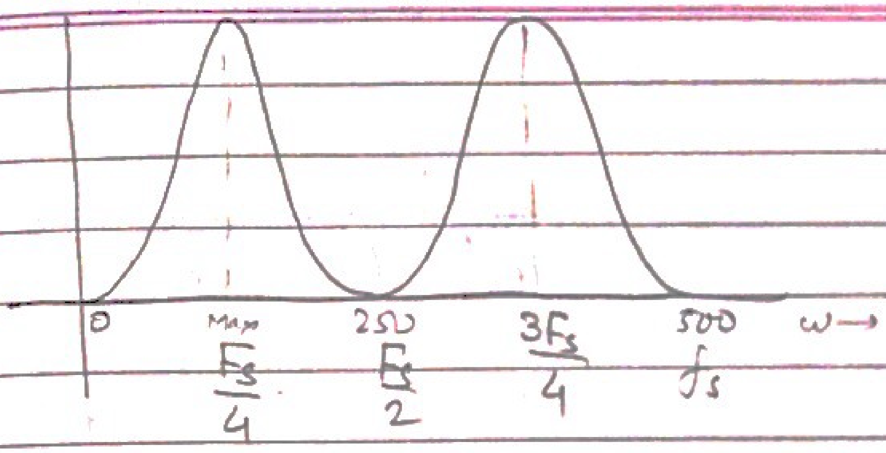
Points closer to unit circle will give rise to large peaks whereas zeros near unit circle will lead to a minima.

Assuming coeff. to be real, poles & zeros ~~are~~ must either be real or complex conjugate pairs.

Q Design a band pass filter with

Specs
(a) Complete signal rejection at 0 & 250 Hz
(b) a narrow pass band centered at 125 Hz
(c) a 3 dB BW of 10 Hz

Assume sampling freq, $f_s = 500$ Hz.



Keeping a zero anywhere ≡ min response
pole " ≡ max response.

Now, placing a pole/zero anywhere on unit circle is seen by the freq. of the poles/zeros.

So, Zeros at 0 Hz. $\longrightarrow \frac{0}{500\,Hz} \times 360^\circ = 0^\circ$

250 Hz $\longrightarrow \frac{250}{500} \times 360^\circ = 180^\circ$

Poles at $\pm \frac{125}{500} \times 360^\circ = \pm 90^\circ$.

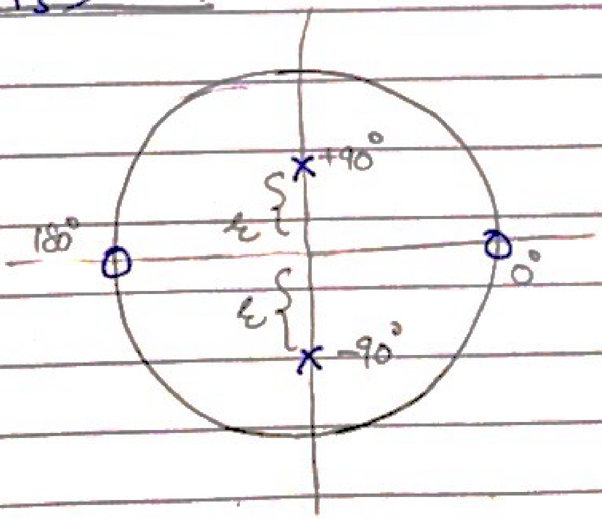
Determining radius of pole :-

$$\boxed{r \simeq 1 - \left(\frac{BW}{F_s}\right)\pi}$$

given $BW = 10\,Hz$

$F_s = 500\,Hz$

$\Rightarrow r \simeq 0.937$



Writing TF

$$H(z) = \frac{(z-1)(z+1)}{(z-re^{j\pi/2})(z-re^{-j\pi/2})}$$

$$\Rightarrow H(z) = \frac{z^2-1}{z^2+0.8779}$$

$\downarrow$ Convert to −ve power of z

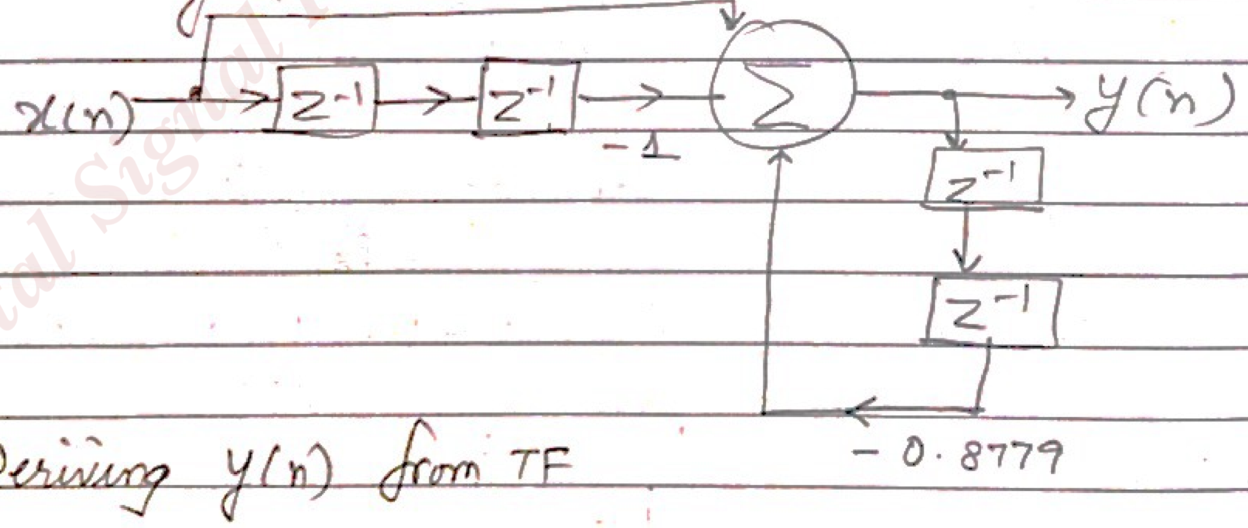
$$\Rightarrow H(z) = \frac{1-z^{-2}}{1+0.8779z^{-2}}$$

$$\left(\equiv \frac{b_0 z^0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}\right)$$

$\Rightarrow$ $b_0 = 1$, $b_1 = 0$, $b_2 = -1$

$a_1 = 0$, $a_2 = 0.8779$

Block diagram :-



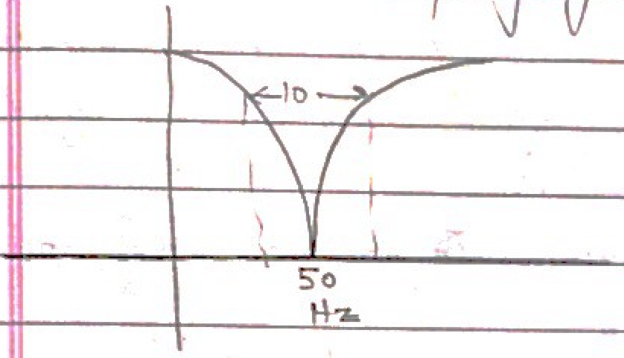
Deriving y(n) from TF

$$y(n) = x(n) - x(n-2) - 0.8779\,y(n-2)$$

☆ Narrow Band stop filter : NOTCH filter

eg : Design a Notch filter.

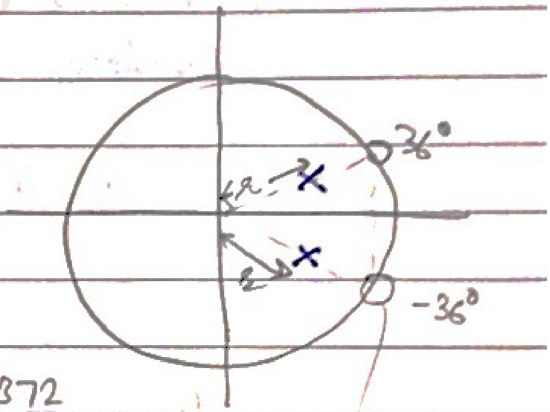
specs → (a) Notch freq : 50 HZ .
3 dB width of notch : ± 5 Hz .
sampling freq :- 500 Hz



Placing zero :- at 50 Hz

$$\frac{50}{500} \times 360° = \underline{\underline{36°}}$$

To achieve a sharp notch, a pair of complex conj. poles at radius of r<1 has to be placed.



complex conjugate (for physically realising)

Finding $r \simeq 1 - \left(\frac{bw}{F_s}\right)\pi \rightarrow 0.9372$

Making TF :-

$$H(z) = \frac{(z - e^{-j36°})(z - e^{j36°})}{(z - 0.937e^{-j36°})(z - 0.937e^{j36°})}$$

$$= \frac{z^2 - 1.618z + 1}{z^2 - 1.5164z + 0.8783} \xrightarrow{z^{(-ve)}} \frac{1 - 1.618z^{-1} + z^{-2}}{1 - 1.5164z^{-1} + 0.8783z^{-2}}$$

Comparing gain, we see the coeff :-

$b_0 = 1$ $a_1 = 1.5164$
$b_1 = -1.618$ $a_2 = 0.8783$.
$b_2 = 1$

M2) Translating Analog to Digital

(a) Impulse Invariant Method

⇒ Impulse response remains same for analog TF & digital TF

Steps

S1) Obtain an apt. analog TF in s-domain.

S2) Determine the impulse response by applying LT & obtain $h(t)$

S3) Sample $h(t)$ so as to produce $h(\underset{\text{discrete no.}}{n}T)$

S4) Derive $H(z)$ by z-transforming $h(nT)$ where T is sampling period.

eg :- $H(s) = \frac{c}{s-p}$ (given TF)

$$\Rightarrow L^{-1}(H(s)) = h(t) = Ce^{+pt}$$

Now, converting $h(nT) \rightarrow$ discrete impulse response (of digital filter)

Replace $t \curvearrowright nT$

$$\Rightarrow h(nT) = Ce^{P_{nt}};$$

Doing z-transform

$$H(z) = \underbrace{\sum_{n=0}^{\infty}}_{\text{its discrete (for cts, use integration)}} h_{(nT)} z^{-n}$$

$$= \underbrace{\sum_{n=0}^{\infty} Ce^{P_{nT}} z^{-n}}_{\text{sum of } \infty \text{ terms} = \frac{1}{1-z^{-1}}}$$

* Complex qty cannot be physically realised. Coeff. have to be real. So, if they are imaginary → have to be in pair to represent real.

So, $h_{(nT)} = c\left(\frac{1}{1-e^{PT}z^{-1}}\right)$

For higher ord. filters, H(s) has to be factorized using partial fraction as simple pole sums:-

$$\therefore H(s) = \frac{C_1}{s-P_1} + \frac{C_2}{s-P_2} + \cdots + \frac{C_m}{s-P_m} = \sum_{K=1}^{m} \frac{C_K}{s-P_K}$$

$$\therefore H(z) = \sum_{K=1}^{m} \frac{C_K}{1-e^{P_K T}z^{-1}}$$

* If poles, P_1 & P_2 are complex conj, C_1 & C_2 will also be complex conj.

So,

Imp. formula (learn)

$$\frac{C_1}{1-e^{P_1 T}z^{-1}} + \frac{C_1^*}{1-e^{P_1^* T}z^{-1}}$$

$$= \frac{2C_r - [C_r \cos(P_i T) + C_i \sin(P_i T)] \cdot 2e^{P_r T}z^{-1}}{1-2e^{P_r T}\cos(P_i T)z^{-1} + e^{2P_r T}z^{-2}}$$

→ C_r & C_i are real & imaginary parts of C_1

→ P_r & P_i are that of P_1

eg $H(s) = \frac{1}{s^2+\sqrt{2}s+1}$, assuming 3dB cut off freq of 150 Hz & $f_s = 1.28$ kHz

Checking if poles are complex & then using steps

① $\omega_c = 2\pi \times 150 = 942.4778$ rad

② denormalised TF = $H'(s) = H(s)\Big|_{s=\frac{s}{\omega_c}}$

$$= \frac{\omega_c^2}{s^2+\sqrt{2}\,\omega_c s+\omega_c^2}$$

$$= \frac{C_1}{s-P_1} + \frac{C_2}{s-P_2}$$

Doing partial fraction

⇒ $P_1 = -\frac{\sqrt{2}\,\omega_c(1-j)}{2} = -666.4324(1-j)$

$P_2 = \overline{P_1} = -\frac{\sqrt{2}\,\omega_c(1+j)}{\sqrt{2}} = -666.4324(1+j)$

& $C_1 = \frac{-\omega_c^2 j}{\sqrt{2}} = -666.4324j$

$C_2 = \overline{C_1}$ or $C_1^* = +\frac{\omega_c^2 j}{\sqrt{2}} = 666.4324j$

Now, see :-

P_r, P_i, C_r, C_i & substitute in eqn

$$H(z) = \frac{\overbrace{(393.9264)}^{b_1}z^{-1}}{1-\underbrace{1.0308}_{a_1}z^{-1}+\underbrace{0.3530}_{a_2}z^{-2}}$$

$\hookrightarrow b_0 = 0$

Now, these coeff. have to be represented using an 8 bit processor, say

So, representⁿ has to be done from values 0.35 to 393($\simeq$400)

$\frac{1}{2^8} = \frac{1}{256} \simeq 0$.

So, value of 0.3530 $\simeq$ 0 } their values won't

1.0306 $\simeq$ 0 } come correctly.

So, take TF separately & normalise it

So, instead of using ω_c (as 400) we use $\frac{\omega_c}{\omega_s}$ ($\simeq$ 0.3078)

So, range goes from 0.3078 – 0.35

Now, easy.

To avoid high gain to prevent overflow, the TF gain can be normalised by sampling freq.

$\omega_s = 2\pi f_s$

$= 2\pi \times 1.2\times10^3$

$\Rightarrow H(z) = \frac{0.3078\, z^{-1}}{1 - 1.0308z^{-1} + 0.3530z^{-2}}$

Or, ω_c can be normalised with ω_s and H(z) resulting will be same.

* Comments :-

1. Filter responce is same at discrete intervals & hence, the name impulse invariant.
2. f_s affects the response for a similar freq. responce of analog filter, fs has to be very high.
3. At multiple of f_s, H(\$) is repeated & hence, aliasing will result ∴ Anti aliasing filter has to be used along with the filter.
4. Can be used for analog filter whose freq. is stop or band limited before applying Impulse Invariant method.
5. Suitable for very sharp cut off low pass filter with little aliasing & reasonably high sampling freq. not suitable for high pass or band stop filters.

ex. Given :- Bandpass filter with Butterworth char.
specs :- pass band 200-300 Hz
f_s = sampling freq = 2 KHz
filter ord = N = 2
→ instead of order, sometimes, we are given stop band attenuation & pass band ripple.
Obtain coeff of filter using BZT method

Note :- $\text{ord}(LP) = \frac{1}{5}\,\text{ord}(BP)$

$\Rightarrow$ Low pass order = $\frac{1}{2}(2) = 1$

$$\Rightarrow \quad TF\Big)_{LPF} = H(s) = \frac{1}{s+1}$$

Doing prewarping of critical freq$_s$

$$\omega'_{P_1} = \tan\left(\frac{\omega_{P_1}T}{2}\right) = \tan\left(\frac{2\pi\, 250}{2\times 2000}\right) = 0.3249$$

$$\omega'_{P_2} = \tan\left(\frac{\omega_{P_2}T}{2}\right) = \tan\left(\frac{2\pi\times 300}{2\times 2000}\right) = 0.5095$$

$$\omega_o^2 = \omega'_{P_1}\,\omega'_{P_2} = 0.1655$$

$$W = \omega'_{P_2} - \omega'_{P_1} = 0.1846.$$

LP ⟶ BP transformⁿ:-

$$H'(s) = H(s)\Big|_{s = \frac{s^2+\omega_o^2}{\omega_s}} = \frac{1}{\frac{s^2+\omega_o^2}{\omega_s} + 1}$$

$$= \frac{\omega_s}{s^2+\omega_s+\omega_o^2}$$

substituting values & finding value of P_1, P_2

$$\hookrightarrow = \underbrace{-0.0923 \pm j(0.3962)}_{\text{analog domain}}$$

Now $z \longrightarrow \frac{z-1}{z+1}$ (BZT)

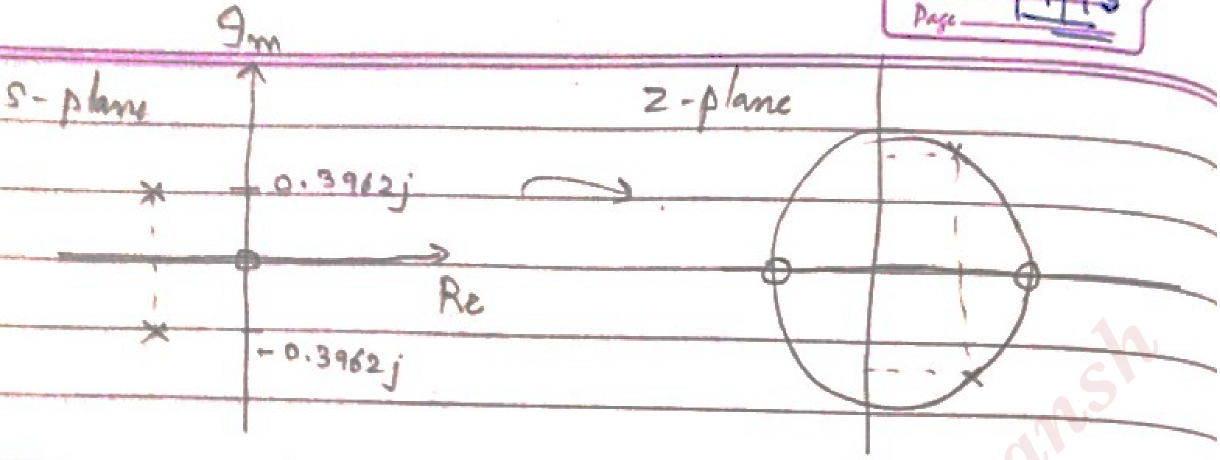
$$TF = H(z) = \frac{0.1367(1-z^{-1})}{1-1.2362z^{-1}+0.7265z^{-2}}$$

$$\Rightarrow P_1, P_2 = \underbrace{0.6040 \pm j(0.6015)}_{\text{digital domain}}$$

fig 2.11

↳ Pole zero diagrams

Puffin
Date 30/9/13
Page



DIGITAL FILTER STRUCTURE

→ Imp., ∵ of computational complexities.
looking for min. no. of multipliers, delay units or registers
time delay memory

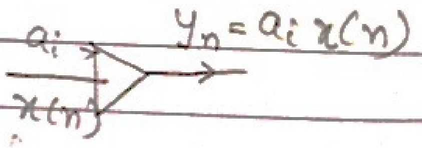
→ No. of delay units ≡ give order of sys

→ registers :- for storing past outputs ; A's & B's (coeff of filter)

→ Another imp. pt. : Word length (register size)

* Most commonly used structures:
1. Direct form
2. Cascade form
3. Parallel form

Notations



☆ Realization of IIR filter.

Assume :- $H(z) = \frac{\sum_{l=0}^{m} a_l z^{-l}}{l + \sum_{l=1}^{m} b_l z^{-l}}$

(i) Digital form I realisation :-

$$H(z) = \frac{N(z)}{D(z)} = \frac{Y(z)}{X(z)} = \left[\frac{a_0 + a_1 z^{-1} + \dots a_n z^{-m}}{1 + b_1 z^{-1} + \dots b_m z^{-m}}\right]$$

Cross multiplying

$$\Rightarrow Y(z)\left[1 + b_1 z^{-1} + \dots + b_m z^{-m}\right] = X(z)\left[a_0 + a_1 z^{-1} + \dots a_n z^{-m}\right]$$

Idea: Trying to find $y(n) \left[= z^{-1}(y(z))\right]$

Taking z^{-1}

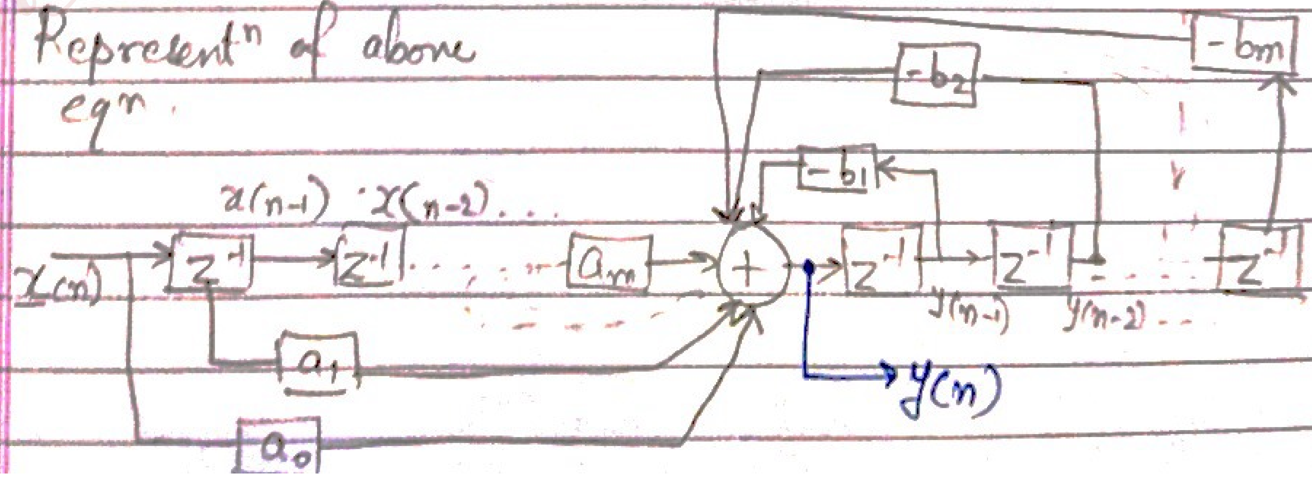
$$\Rightarrow y(n) + b_1 y(n-1) + \dots b_m y(n-m) = \left[a_0 X(n) + a_1 X(n-1) + \dots a_n X(n-m)\right]$$

$$\Rightarrow y(n) = \left[a_0 x(n) + a_1 x(n-1) + \dots + a_n x(n-m)\right]$$

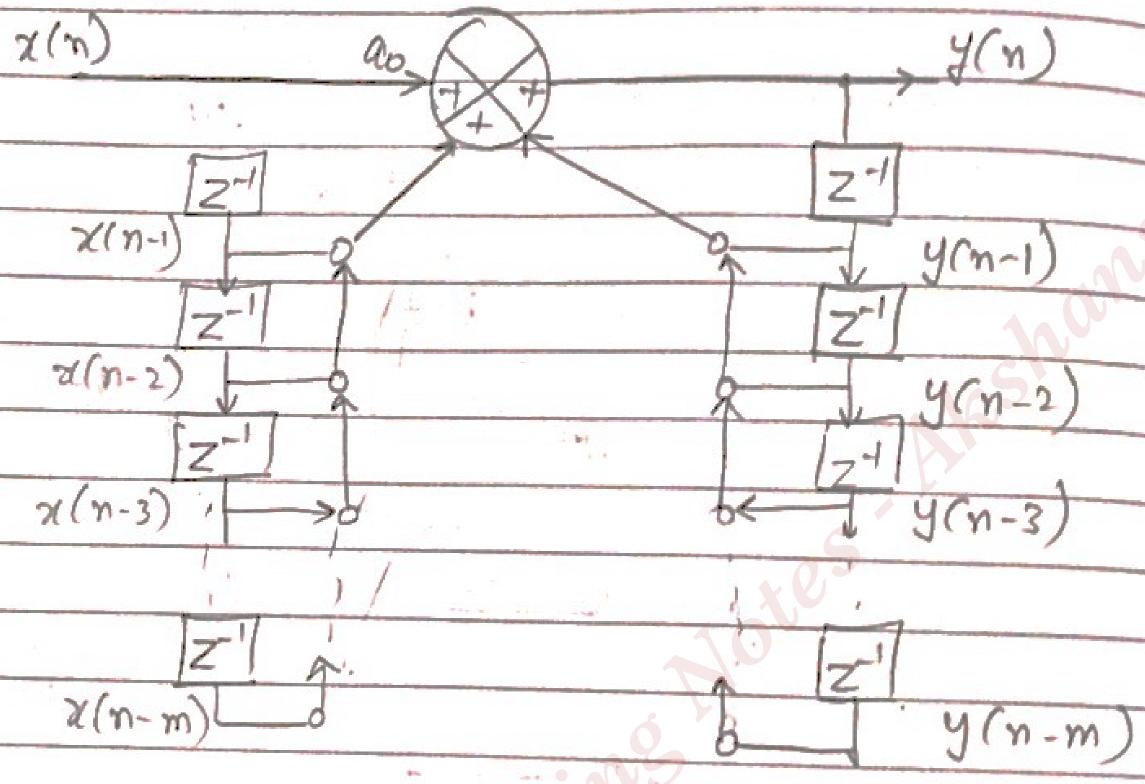
$$- \left[b_1 y(n-1) + \dots + b_m y(n-m)\right]$$

☆ Represent^n of above eq^n.

Direct form I IIR



Another form



☆ No. of additions = 2(m-1)
multiplic^{ns} = 2n

☆ (ii) Direct form (II) realization :-
(also called Canonical form)

$$H(z) = \frac{N(z)}{D(z)} = \frac{Y(z)}{X(z)} \; ; \; Y(z) = \frac{N(z)}{D(z)} \cdot X(z)$$

$$= N(z) \cdot W(z)$$

Assuming $W(z) = \frac{X(z)}{D(z)}$; $X(z) = W(z) \cdot D(z)$

Now,

$$N(z) = a_0 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_m z^{-m}$$

$$D(z) = 1 + b_1 z^{-1} + b_2 z^{-2} + \ldots + b_m z^{-m}.$$

So,

$$X(z) = W(z)\left[1 + b_1 z^{-1} + \dots + b_m z^{-m}\right]$$

$$\Rightarrow x(n) = w(n) + b_1 w(n-1) + \dots + b_m w(n-m)$$

$$\Rightarrow w(n) = x(n) - \left[b_1 w(n-1) + b_2 w(n-2) + \dots + b_m w(n-m)\right]$$

$$\& \quad Y(z) = W(z)\left[a_0 + a_1 z^{-1} + a_2 z^{-2} + \dots a_m z^{-m}\right]$$

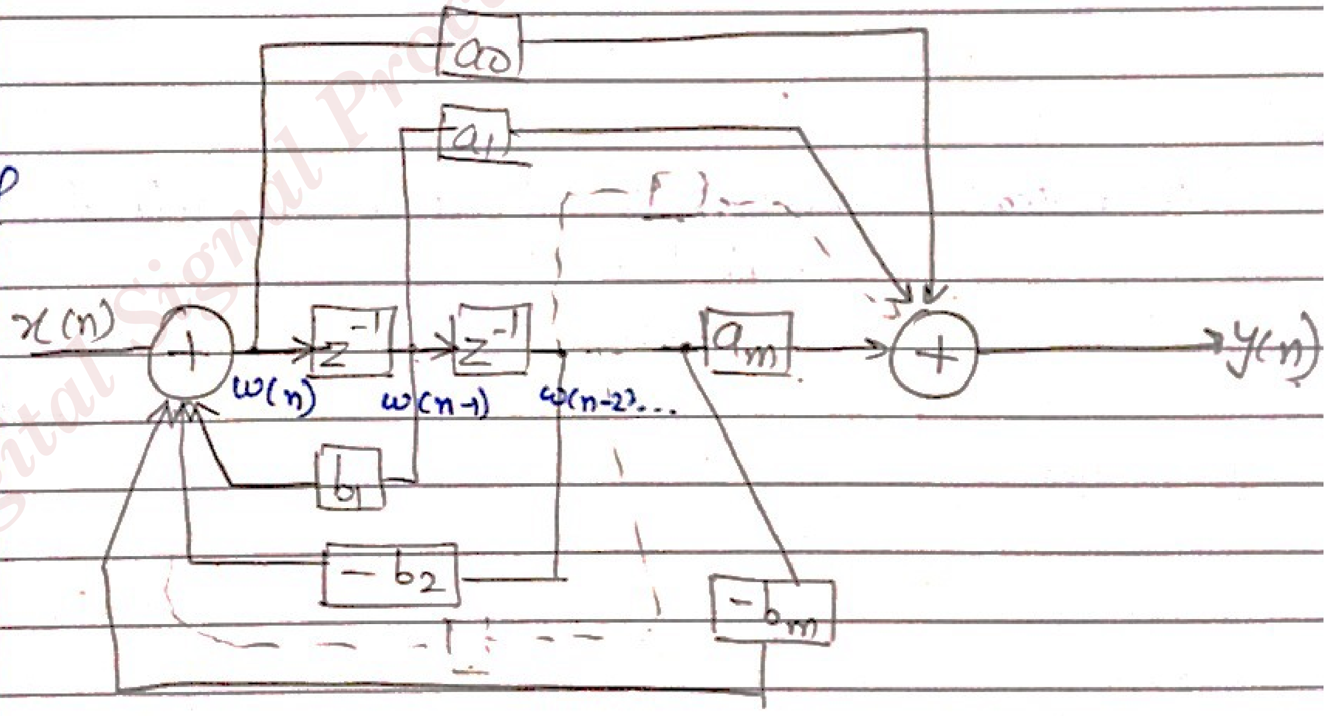
$$\downarrow$$

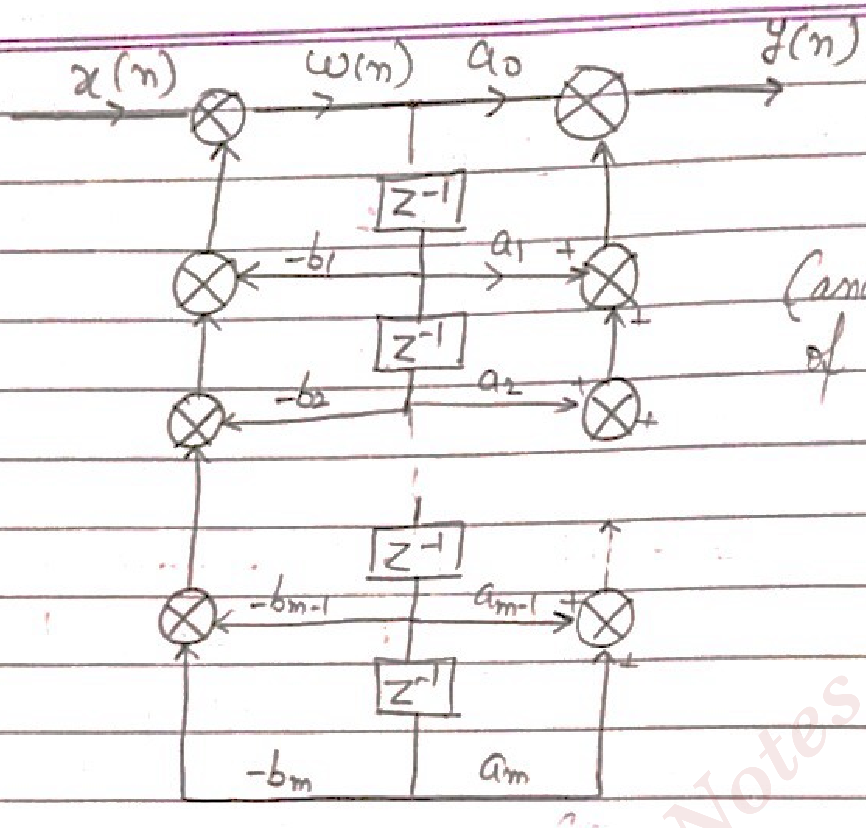
$$\Rightarrow y(n) = a_0 w(n) + a_1 w(n-1) + \dots + a_m w(n-m)$$

Reason for doing this? → both x(n) & y(n) are now in terms of one variable (w). So, no need to store them separately.

Represent^n :-

Direct form II realis^n of IIR filters





(another form/way of realiz^n)

(iii) **Cascade form**: H(z) TF is to be factored first so that $H(z) = k \cdot H_1(z) H_2(z) \ldots H_m(z)$; m is a +ve integer & each $H_m(z)$ can be either a first order or second order TF

$$\therefore H(z) = \frac{k \cdot \prod_{i=1}^{p} (1-a_i z^{-1}) \cdot \prod_{i=1}^{q} (1-b_i z^{-1})(1-b_i^* z^{-1})}{\prod_{i=1}^{r} (1-c_i z^{-1}) \prod_{i=1}^{l} (1-d_i z^{-1})(1-d_i^* z^{-1})}$$

→ Total no. of zeroes :- $p+2q$
poles :- $r+2l$.

eg:- TF :

$$H(z) = \frac{0.7(z^2-0.36)}{z^2+0.1z-0.72}$$

factorizing :-

$$= \frac{(0.7)(z-0.6)(z+0.6)}{(z-0.8)(z+0.9)}$$

$$= (0.7)\underbrace{\left[\frac{z-0.6}{z-0.8}\right]}_{H_1(z)}\underbrace{\left[\frac{z+0.6}{z+0.9}\right]}_{H_2(z)} \text{, say.}$$

So, overall TF, H(z) can look like :-

$$0.7 \rightarrow \boxed{H_1(z)} \rightarrow \boxed{H_2(z)} \xrightarrow{H(z)}$$

How to choose $H_1(z)$ & $H_2(z)$ from given factors of $H(z)$?

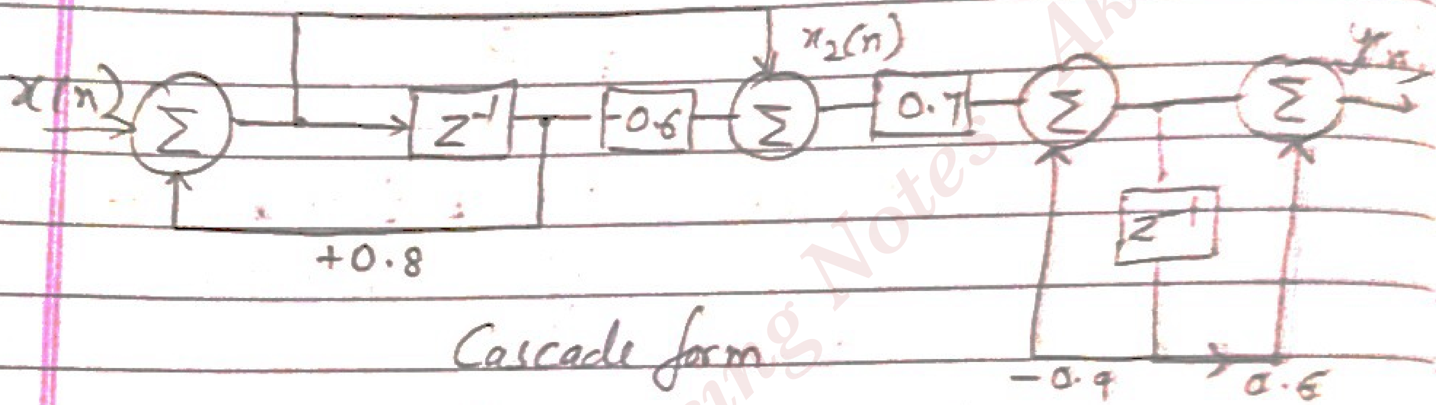
↳ z-( ) terms : Take together
z+( ) terms : Take together

↳ These coeff have to be implemented in binary. Now, for a -ve coeff., one extra bit for sign will be used. So, if represent^n is done in 4 bits, one bit used in sign & 3 bits left to represent the coeff in digital. So, -ve coeff. together makes similar things together

If all coeffs are +ve, then take those pairs s.t difference in their values is min.

$k = 0.7$, $H_1(z) = \frac{z+0.6}{z+0.9}$, $H_2(z) = \frac{z-0.6}{z-0.8}$

$= \frac{1+0.6z^{-1}}{1+0.9z^{-1}}$. $= \frac{1-0.6z^{-1}}{1-0.8z^{-1}}$



Cascade form

$$Y(z) = k \cdot H_1(z) \cdot \underbrace{H_2(z) \cdot X(z)}_{X_2(z)}$$

$$= k \cdot H_1(z) X_2(z)$$

$H_2(z) = \frac{X_2(z)}{X(z)}$, $H_1(z) = \frac{Y(z)}{k \cdot X_2(z)}$

$\therefore x_2(n) = x_n - 0.6x(n-1) + 0.8\, x_2(n-1)$

$y(n) = k\, z^{-1} \, [H_1(z) X_2(z)]$

$= k[\, x_2(n) + 0.6x_2(n-1) - 0.9\, y(n-1)]$

Parallel form of realizn :-
Doing partial fraction to given TF

5 28
8
•1.96
77

$$\Rightarrow H(z) = \frac{(0.7)(z-0.6)(z+0.6)}{(z-0.8)(z+0.9)}$$

$$= \frac{A}{z-0.8} + \frac{B}{z+0.9}$$

$$\Rightarrow (z+0.9)A + (z-0.8)B = (0.7)(z-0.6)(z+0.6)$$

$$\Rightarrow (1.7)A = (0.7)(0.2)(1.4)$$

$$\Rightarrow A = 0.144$$

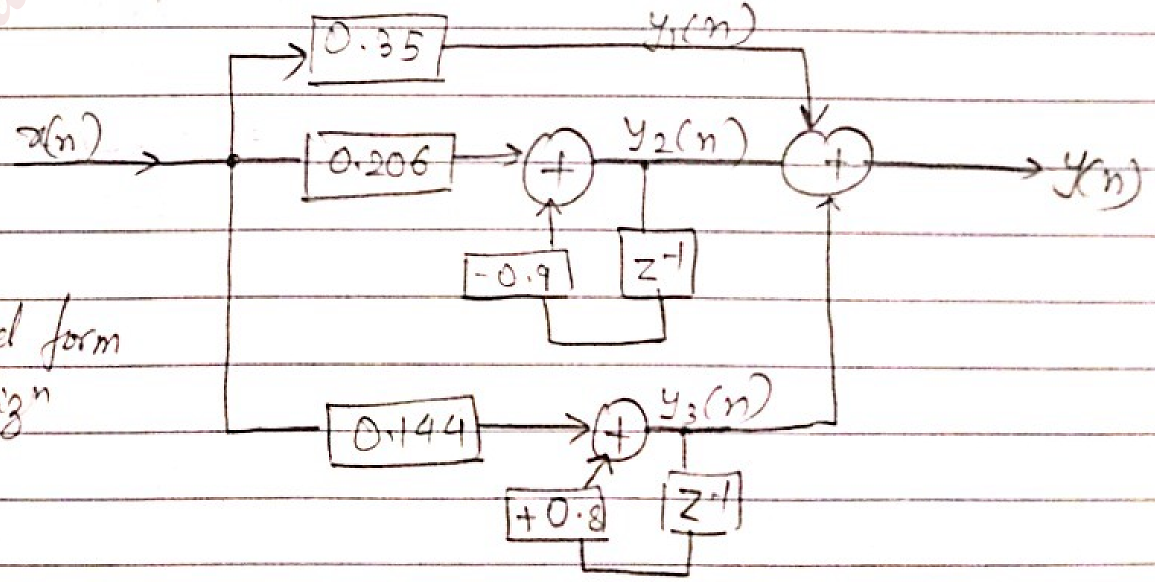
$$\text{IIly, } B = 0.206$$

$$\text{So, } H(z) = 0.35 + \frac{0.206}{1+0.9z^{-1}} + \frac{0.144}{1-0.8z^{-1}}$$

So,

$$Y(z) = \underbrace{0.35\,X(z)}_{Y_1(z)} + \underbrace{\frac{0.206}{1+0.9z^{-1}}X(z)}_{Y_2(z)} + \underbrace{\frac{0.144}{1-0.8z^{-1}}X(z)}_{Y_3(z)}$$

Implementing :



Parallel form
realizn

FIR FILTER DESIGN

* Key features of FIR filters

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

TF = $H(z) = \sum_{k=0}^{n-1} h(k) z^{-k}$

→ always stable (no o/p dependant)

→ less coeff. have to be implemented (∵ ∃ no feedback)

2. Gives linear phase response (exact)

→ advantageous over IIR.

3. Simple to implement, suffers less from word length is more.

Linear phase response & its implications

phase delay, $T_p = -\frac{\theta(\omega)}{\omega}$

group delay, $T_g = -\frac{d\theta(\omega)}{d\omega}$

result of non linear phase char. in phase distortion which is undesirable in music, video and biomedicine etc & can be avoided by filters with linear phase characteristics

+ve symmetry : symm about y axis
−ve symmetry : symm about line $x = y$

Puffin
Date 6/10/13
Page

phase DELAY

For linear phase response,

$$\theta(\omega) = -\alpha.\omega \quad \text{or} \quad \beta - \alpha.\omega.$$

$$(\equiv y = mx + c)$$

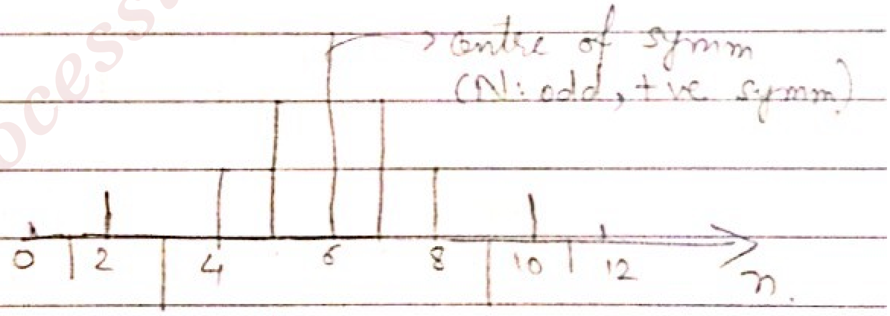
If $\theta(\omega) = -\alpha.\omega$: filter will have constt T_p & T_g
Also, for this impulse response must have positive symmetry
→ $h(n)$

$$\therefore h(n) = h(N-n-1) \begin{cases} n = 0, 1, \dots (N-1)/2 ; & N = \text{odd} \\ n = 0, 1, \dots \frac{N}{2} - 1 ; & N = \text{even} \end{cases}$$

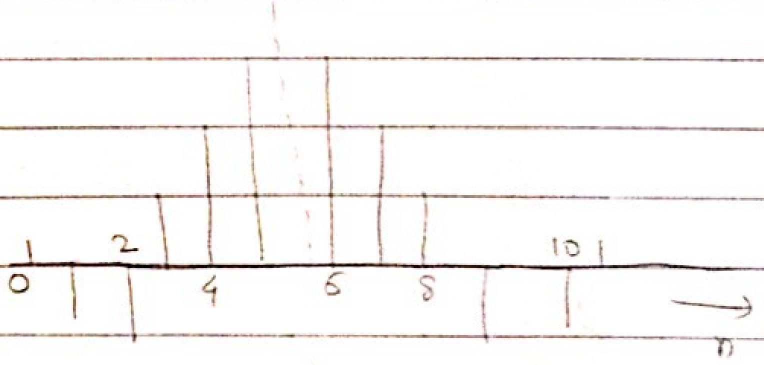
$$\& \quad \alpha = \frac{N-1}{2} \Big\} \; T_p \text{ is a } f^n \text{ of filter length}$$

* represent^n of signals :

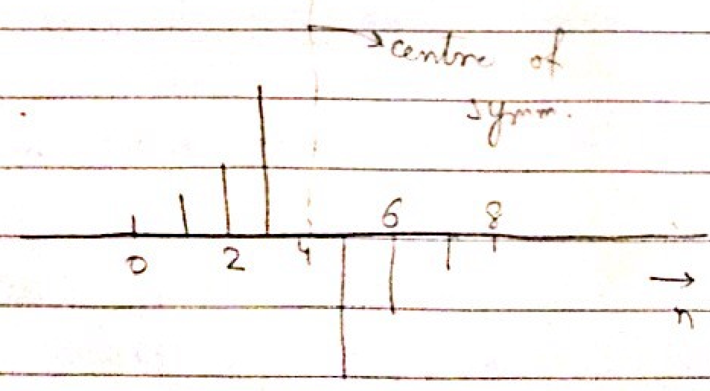
N = 13 (odd)



N = 12 (even)

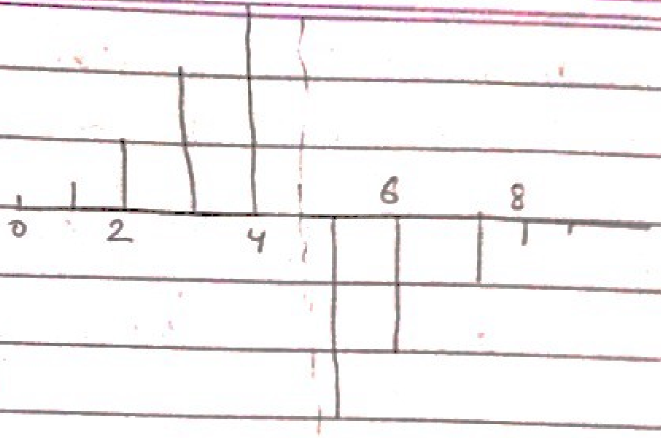


N = 9 (odd)



N = 10 (even)

-ve symm.



✲ For an FIR sys, we can have a group of 4 (combin^{ns})

Length : odd/even

Symmetry : +ve/-ve

Using this, type - 1, 2, 3 & 4 are made

eg Design a digital FIR filter defined over interval $0 \leq n \leq N-1$

If N=7 & $h(\omega)$ satisfies symm. cond^n :
$\underline{h(n) = h(N-n-1)}$, show that filter has linear phase char.

+ve symm.

Also repeat for N = 8

(If $h(n) = (-)h(N-n-1)$)

-ve symm

For linear phase response :- necessary & sufficient cond^n :- impulse response is symm.

∴ if N = 7 ⇒ $h(0) = h(6)$

$h(1) = h(5)$

$h(2) = h(4)$

$h(3)$: center sample.

Idea :- find $\theta(\omega)$ & prove $\theta(\omega) \propto \omega$ done ($y \propto x$ or $y = mx$: linear)

$$H(\omega) = H(e^{j\omega t})$$

$$= \sum_{k=0}^{6} h(k)e^{-j\omega T} = h(0) + h(1)e^{-j\omega t} + \dots + h(6)e^{-j6\omega T}$$

$$= e^{-j3\omega t}[h(0)e^{j3\omega T} + h(1)e^{j2\omega T} + h(2)e^{j\omega T} + h(3) + h(6)e^{-j3\omega T} + h(5)e^{-j2\omega T} + h(4)e^{-j\omega T}]$$

$\because \cos\theta = \frac{e^{j\theta} + e^{-j\theta}}{2}$

$$= e^{-j3\omega T}[h(0)e^{j3\omega T} + h(6)e^{-j3\omega T} + h(1)e^{j2\omega T} + h(5)e^{-j2\omega T} + h(2)e^{j\omega T} + h(4)e^{-j\omega T} + h(3)]$$

$a_0 = 2h(\frac{N-1}{2} - n)$

or

$a_0 = h(\frac{N-1}{2})$

$$= e^{-j3\omega T}[h(0)[2\cos 3\omega T] + h(1)[2\cos 2\omega T] + h(2)[\cos \omega T] + h(3)]$$

if $a_0 = h(3)$, $a_k = 2h(3-k)$

↳ $k = 1, 2, 3,$

then,

★ $$H(\omega) = \sum_{k=0}^{3} a_k \cos k\omega T \, e^{-j3\omega T}$$

$$= \pm|H(\omega)| \cdot e^{j\theta(\omega)}$$

↳ where, $\pm|H(\omega)| = \sum_{k=0}^{3} a_k \cos n\omega T$ &

$\theta(\omega) = -3\omega T.$

$\because \theta(\omega) \propto \omega \; . \; (= -3\omega T)$

So, linear phase response ✓

If $N = 8$ (even)

$\Rightarrow h(0) = h(7)$

$h(1) = h(6)$

$h(2) = h(5)$

$h(3) = h(4)$

&

$$H(\omega) = \sum_{k=0}^{7} h(k) e^{-jk\omega T}$$

$$= h(0) + h(1)e^{-j\omega T} + h(2)e^{-j2\omega T} + \cdots + h(6)e^{-j6\omega T} + h(7)e^{-j7\omega T}$$

$$= e^{-j\frac{7\omega T}{2}}\left[h(0)\left(e^{j\frac{7\omega T}{2}} + e^{-j\frac{7\omega T}{2}}\right) + h(1)\left(e^{j\frac{5\omega T}{2}} + e^{-j\frac{5\omega T}{2}}\right) + h(2)\left(e^{j\frac{3\omega T}{2}} + e^{-j\frac{3\omega T}{2}}\right) + h(3)\left(e^{j\frac{\omega T}{2}} + e^{-j\frac{\omega T}{2}}\right)\right]$$

$$= e^{-j\frac{7\omega T}{2}}\left[2h(0)\cos\frac{7\omega T}{2} + 2h(1)\cos\frac{5\omega T}{2} + 2h(2)\cos\left(\frac{3\omega T}{2}\right) + 2h(3)\cos\left(\frac{\omega T}{2}\right)\right]$$

Polar form

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

linear

$$\because \pm|H(\omega)| = \sum_{k=1}^{4} b(k)\cos\left[\omega T\left(k - \frac{1}{2}\right)\right] \; ; \; \theta(\omega) = \frac{-7}{2}\omega T$$

$\hookrightarrow b(k) = 2h\left(\frac{N}{2} - k\right) \; ; \; k = 1, 2 \ldots N/2$

★ cos : +ve symm.

★ sin : -ve symm.

★ Summery of key points about 4 types of linear phase FIR filt

	Impulse response Symm	No. of coeff. (N)	Freq response $H(\omega)$	Type of linea phase
T_p, T_g linear	• +ve symm. $h(n)=h(N-1-n)$	odd	$e^{-j\omega(\frac{N-1}{2})}\sum_{n=0}^{(\frac{N-1}{2})} a_n\cos(\omega n)$	① → most versatile of the four
	• "	even	$e^{-j\omega(\frac{N-1}{2})}\sum_{n=1}^{N/2} b_n\cos[(\omega)(n-\frac{1}{2})]$	2
T_g linear	• -ve symm. $h(n)=-h(N-1-n)$	odd	$e^{-j[\omega(\frac{N-1}{2})-\frac{\pi}{2}]}\sum_{n=1}^{(\frac{N-1}{2})} a_n\sin(\omega n)$	3.
	• "	even	$e^{-j[\omega(\frac{N-1}{2})-\frac{\pi}{2}]}\sum_{n=1}^{N/2} b_n\sin[\omega(n-\frac{1}{2})]$	4.

→ $a(0) = h(\frac{N-1}{2})$; $a_n = 2h(\frac{N-1}{2} - n)$

→ $b(n) = 2h(\frac{N}{2} - n)$

→ for type 2 & 3:-
freq resp. at $f = 0.5$ is always 0. & unsuitable for high pass.

→ for type 3:-
at $\omega = 0$, freq $= 0$. So, cannot be used for low pass filter
phase shift : $\frac{\pi}{2}$

✻ FIR Coefficient Calculation .

- Impulse response coeff. of FIR filter

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

$$H(z) = \sum_{n=0}^{N-1} h(n) z^{-n}$$: z tranform of h(n)

✻ Designing FIR filter ⇒ finding coeff ⇒ finding h(n) values

- Methods to find coeff:-

M1) ↳ Window method
M2) ↳ Optimal method
M3) ↳ Freq. sampling method.

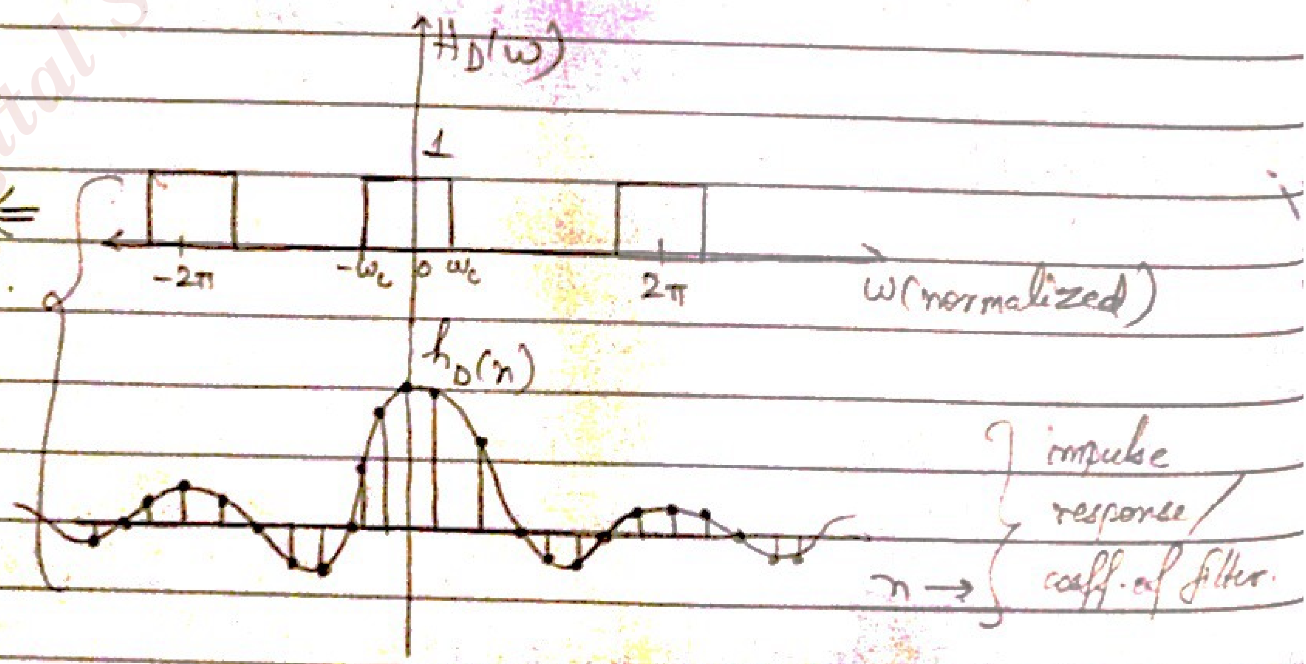
comes from graph below

$$h_D(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} 1\times e^{j\omega n}\, d\omega = \frac{1}{2\pi}\int_{-\omega_c}^{\omega_c} e^{j\omega n}\, d\omega$$

desired or ideal sinc fn.

$$= \begin{cases} 2f_c \frac{\sin(n\omega_c)}{(n\omega_c)} & ; n \neq 0,\ -\infty \leq n \leq \infty . \\ f_c & ; n = 0 \ (L' \text{ Hospital's}) \end{cases}$$

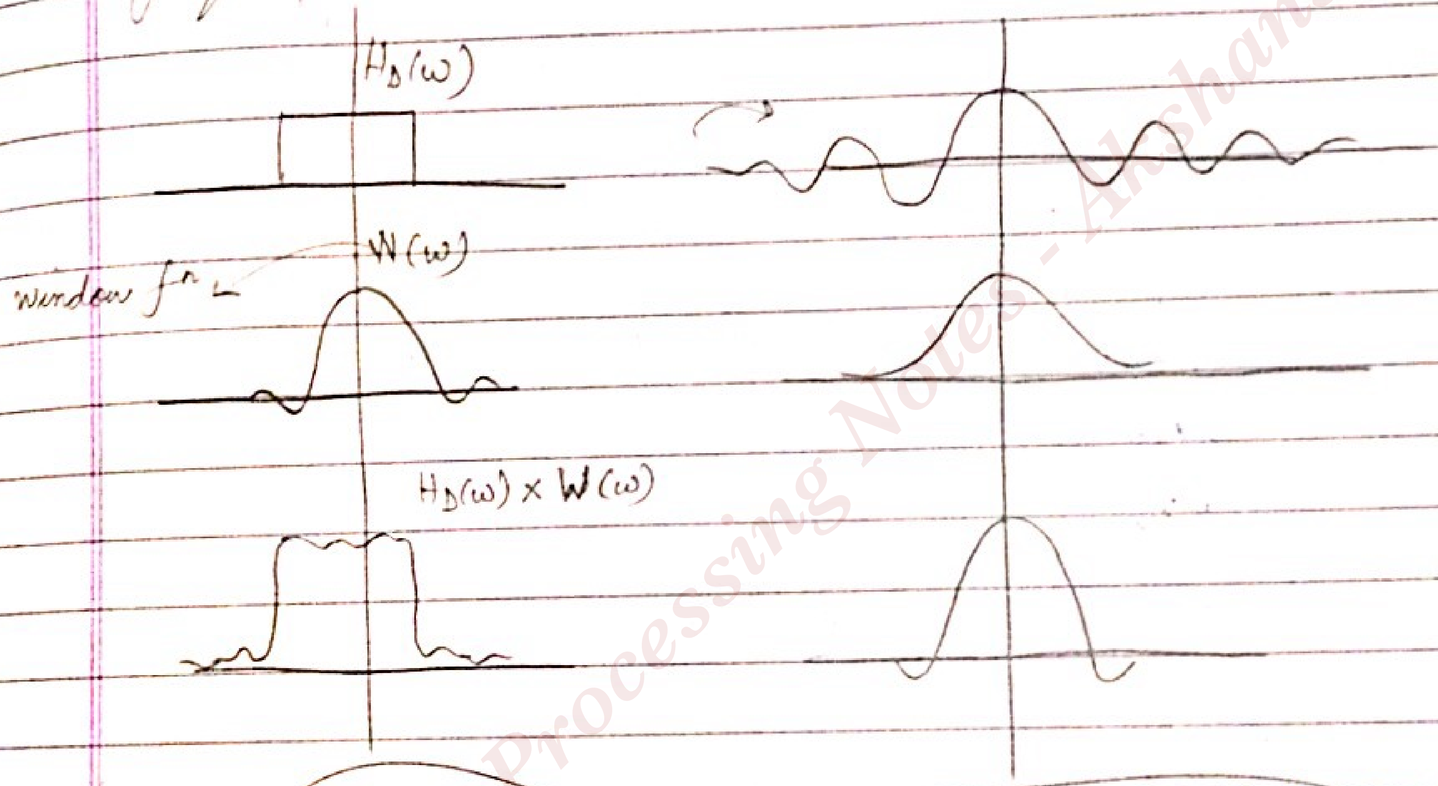
∃ ∞ ⇐ no. of coeff. for an ideal LPF



impulse response / coeff. of filter.

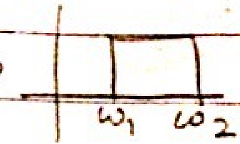
$$h_D(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_D(\omega) e^{j\omega n}\, d\omega \longrightarrow IFT$$

Now,
If no. of coeff. are limited (given), how does the frequency response becomes?

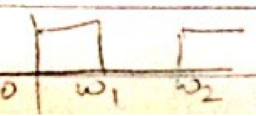


* Summary of ideal impulse responses for std. f selective filters

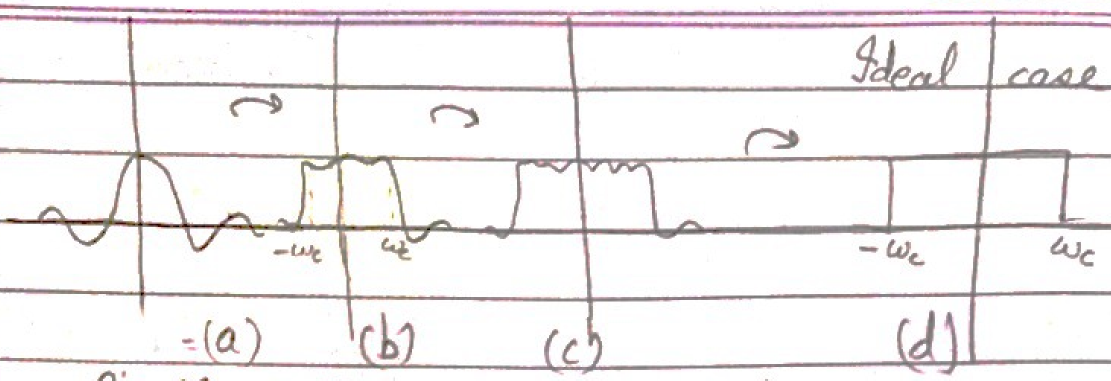
	$h_D(n),\ n \neq 0$	$h_D(n);\ n = 0$
Low Pass filter	$2f_c \frac{\sin(n\omega_c)}{n\omega_c}$	$2f_c$
High Pass	$-2f_c \frac{\sin(n\omega_c)}{n\omega_c}$	$1-2f_c$
Bandpass	$2f_2 \frac{\sin(n\omega_2)}{n\omega_2} - 2f_1 \frac{\sin(n\omega_1)}{n\omega_1}$	$2(f_2-f_1)$
Bandstop	$2f_1 \frac{\sin(n\omega_1)}{n\omega_1} - 2f_2 \frac{\sin(n\omega_2)}{n\omega_2}$	$1-2(f_2-f_1)$



$\rightarrow \frac{1}{2\pi}\int_{\omega_1}^{\omega_2} 1 . e^{j\omega n} d\omega$



*



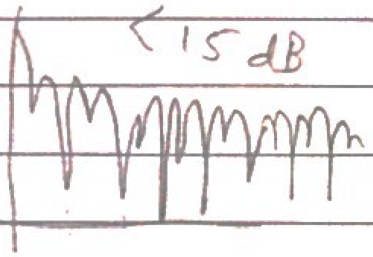
no. of coeff. (d) > (c) > (b) > (a)
$\to \infty$

* how to reduce no. of $coeff_s$?

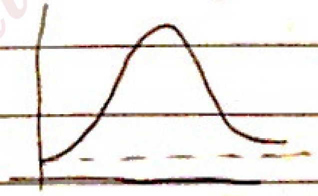
↳ M1) Create a window f^n (basically a box of unit = 1 which gets multiplied by my f^n) for desired no. of coeff.

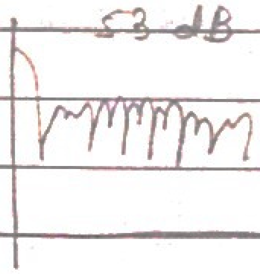
☆ Window f^n — Attenuation (dB)

- Rectangular

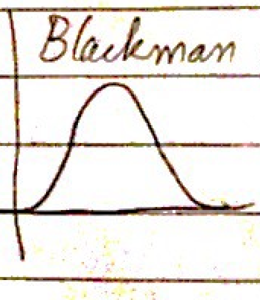


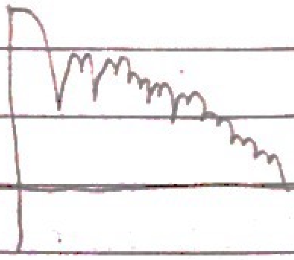
- Hamming window





- Blackman window





More ideal : Blackman > Hamming > Hanning > Rectangular

Window Difficult implement^n : " < " < " < "

$\therefore^{c}$ of complex window f^n

☆	Window	Trans^n width (Hz) (normalized)	Pass band ripple (dB)	Main lobe relative to side lobe (dB)	Stop band attenuation (Max) (dB)	Window f^n $w(n)$, $\lvert n \rvert \leq \frac{(N-1)}{2}$
1)	Rectangular	$\frac{0.9}{N}$	0.7416	13	21	1
2)	Hanning	$\frac{3.1}{N}$	0.0546	31	44	$0.5+0.5\cos\left(\frac{2\pi n}{N}\right)$
3)	Hamming	$\frac{3.3}{N}$	0.0194	41	53	$0.54+0.46\cos\left(\frac{2\pi n}{N}\right)$
4)	Blackman	$\frac{5.5}{(N)}$	0.0017	57	75	$0.42+0.5\cos\left(\frac{2\pi n}{N-1}\right)$

$\rightarrow$ Length of filter : mainly responsible for change in freq. response.

☆ Idea :- find N from above table (using trans^n width cond^n)
Use that to find window f^n $w(n)$.
Then, $h_D(n)$ is known.
So, the f^n $h_D(n) \times w(n)$ can be obtained.
(which window to choose ? → see the cond^{ns} given)

Continued

	Window	Trans^n width	Pass band ripple	Main lobe relative to side lobe	Stop band attenuation	Window f^n
5)	Kaiser	$2.93/N$ ($\beta = 4.54$)	0.0274		50	$\frac{I_0(\beta[1-[\frac{2n}{(N-1)}]^2]^{1/2})}{I_0(\beta)}$
		$\frac{4.32}{N}$ ($\beta = 6.76$)	0.00275		70	"
		$\frac{5.71}{N}$ ($\beta = 8.96$)	0.000275		90	"

☆ Always choose the window which has the simplest complexi

From table,

$$\text{Kaiser } f^n,\ \omega(n) \begin{cases} = \dfrac{I_0\left\{\beta\left[1-\left(\frac{2n}{N-1}\right)^2\right]^{1/2}\right\}}{I_{0(\beta)}} & \begin{array}{l} \text{if } N=\text{odd} \rightarrow\ ;\ -\frac{(N-1)}{2} \leqslant n \leqslant \frac{(N-1)}{2} \\ \text{if } N=\text{even} \rightarrow\ ;\ -\left(\frac{N}{2}-1\right) \leqslant n \leqslant \left(\frac{N}{2}-1\right) \end{array} \\ = 0 & ;\ \text{elsewhere} \end{cases}$$

↳ $I_0(x)$ is a modified Bessel f^n of 1st kind.
β : control factor b/w trans^n band & pass band ripple.

$I_0(x)$ is evaluated using power series

$$I_0(x) = 1 + \sum_{k=1}^{L}\left[\frac{\left(\frac{x}{2}\right)^k}{k!}\right]^2 \quad ;\ \text{typically, } L<25.$$

- → when $\beta = 0$
  Kaiser window corresponds to rectangular window
- → when $\beta = 5.44$
  Kaiser window is very similar to Hamming window
- → β: determined based on stop band attenuation requirements.

* Emperical formula to estimate β
  - ↳ $\beta = 0$ if $A \leqslant 21$ (corresponding to rectangular window)
  - ↳ $\beta = 0.5842(A-21)^{0.4} + 0.07886(A-21)$ if $21 < A < 50\,dB$ (for Hamming window)

⋆ linear phase char. got only when fn is symm.
So, only half. the length of filter coeff will be needed to compute

↳ $\beta = 0.1102(A-8.7)$ if $A \geq 50dB$

* where, $A = -20 \log_{10}(\delta)$

↳ $\delta = \min(\delta_s, \delta_p)$

stopband ripple   passband ripple

- computing coeff. for filter ~~coeff.~~ (Kaiser window)

↳ $N \geq \frac{A-7.95}{14.36\Delta f}$

↳ Δf : normalised trans^n width

- Note:

values of β & N are to be used to compute Kaiser window coeff $w(n)$

SOLVING PROBLEMS

☆ Idea: for any given question & we have to use window f^n,

① See stopband attenuation using data given in question. Compare using data given in table. Decide the apt. window.

② Note the trans^n width from question. Use that in table to find N

③ Using respective formulas, find
$h_D(n)$ from 0 to N-1
$w(n)$ from 0 to N-1

④ Now, $h(n) = h_D(n) \times w(n)$ ✓

* Steps to design FIR filter : by calculating FIR coeff.

Method 1 Using window method

s1) Specify the desired freq. responce $H_D(\omega)$

s2) Obtain the impulse response $h_D(n)$ by IFT.

s3) Select window fn that specifies attenuation specs.
~~Detel~~ Determine N using apt. reln b/w N & Δf (Transn width)

s4) Obtain values of $\omega(n)$ & hence $h(n)$ by

$$h(n) = h_D(n) \cdot \omega(n)$$

↳ method is straight forward but not optimal

Q. Obtain low pass FIR filter coeff to meet the following specifications using window method.

① passband edge freq = 1.5 kHz

② $\Delta f = 0.5$ kHz

⇒ Transn width = 0.5 kHz

③ Stop band attenuation ≥ 50 dB.

④ $f_s = 8$ kHz

We choose Hamming ② (∵ of condn ③)

Now,

$$\frac{3.1}{N} = \frac{0.5 \text{ kHz}}{(8 \text{ kHz})} = 0.625 \Rightarrow N = 52.8$$

→ normalising

From table

$$h_D(n) \text{ for LPF} = \begin{cases} 2f_c \dfrac{\sin(n\omega_c)}{(n\omega_c)} & n \neq 0 \\ 2f_c & n = 0 \end{cases}$$

from cond^n ③,

Hamming ②, Blackman or Kaiser satisfy.

Use hamming ② for simplicity.

Now, $N = 52.8$

$\Rightarrow$ n varies from -26 to 26.

Now,

$$w(n) = 0.54 + 0.46 \cos\left(\frac{2\pi n}{53}\right) ; \ -26 \leq n \leq 26.$$

Because of smearing effect of window, cut off freq of resultant filter will be diff^t from specs. To compensate this,

$$f_c' = f_c + \frac{\Delta f}{2} = 1.75 \text{ kHz} = \frac{1.75}{8} = 0.21875$$

Since $h(n)$ is symmetrical, we need to compute values for $h(0)$ to $h(26)$ only.

$\therefore$ $h_D(0) = 2f_c = 0.4375$; $w(0) = 1$; $h(0) = (0.4375)(1) = 0.4375.$

$$h_D(1) = 2f_c \frac{\sin(n\omega_c)}{(n\omega_c)} = 0.31219$$

$$w(1) = 0.54 + 0.46 \cos\left(\frac{2\pi(1)}{53}\right)$$

$$w(1) = 0.99677.$$

$$h(1) = 0.31118 = h(-1)$$

$$h_D(2) = 2f_c \frac{\sin(2\omega_c)}{(2\omega_c)} = 0.06013$$

$$w(2) = 0.98713$$

$$h(2) = 0.06012 = h(-2)$$

||ly,

$h_D(26) = h_D(-26) = -0.000914$

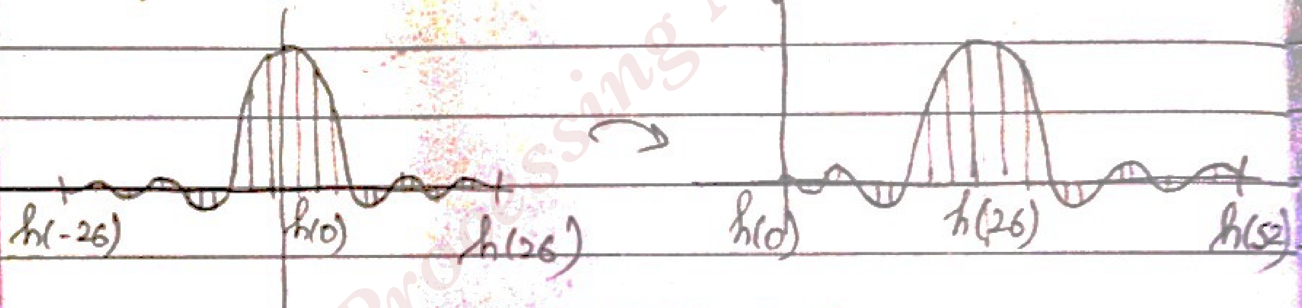
we did for -26 to 26.

Now, we need to implement for 0 to 52

So, we have to shift it

hence, for 53 coeff. of filter to be implemented, index has to be modified by adding 26 to each index.

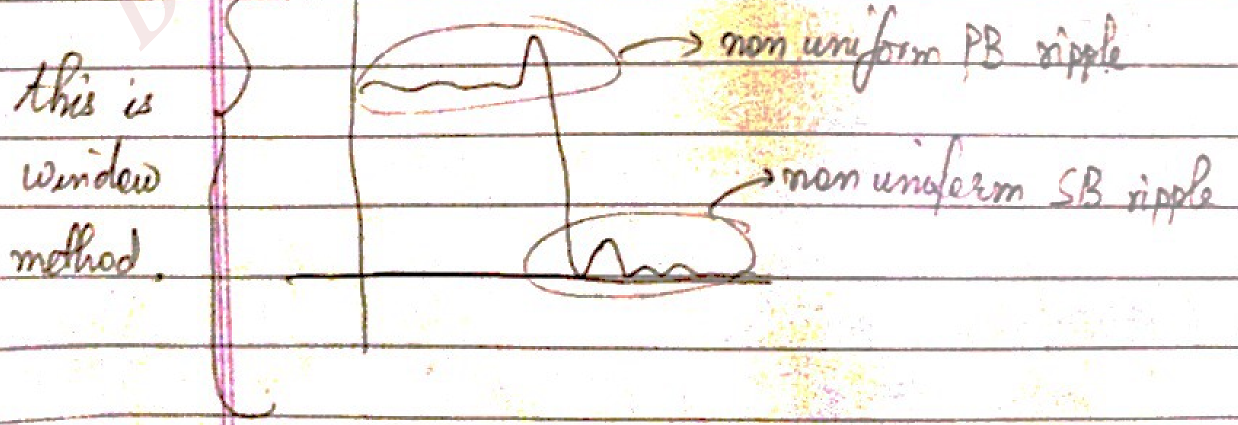
Graphically (what are we doing)



(we have this) (we want this)

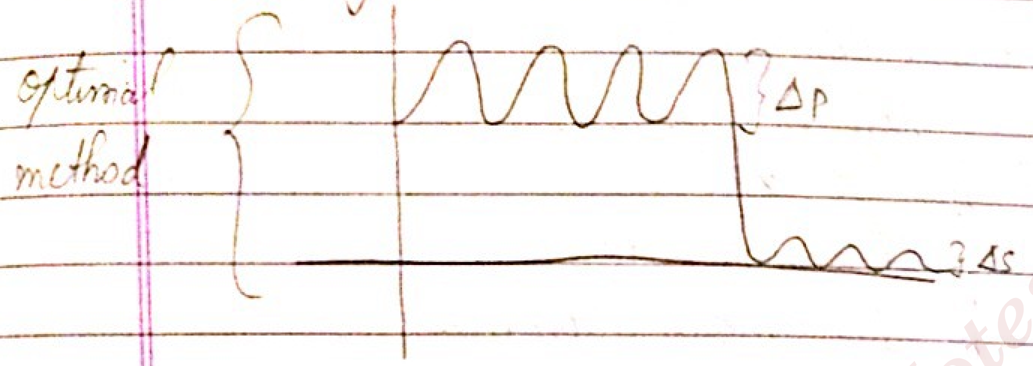
* **Method 2**: OPTIMAL METHOD for filter design (FIR)

computer based method

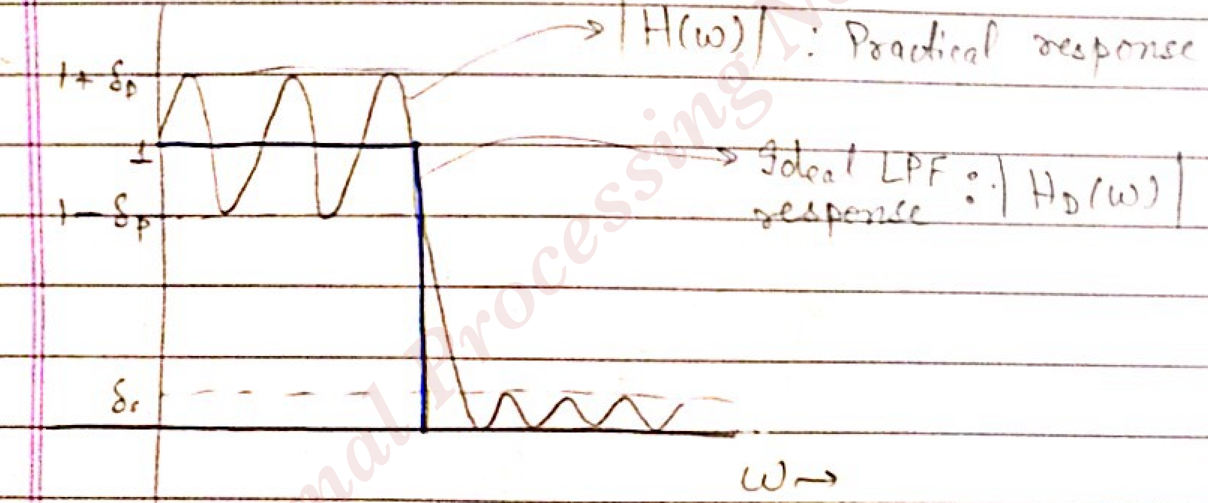


In optimal method, we want to find FIR coeffs. s.t. PB & SB ripple is UNIFORM.

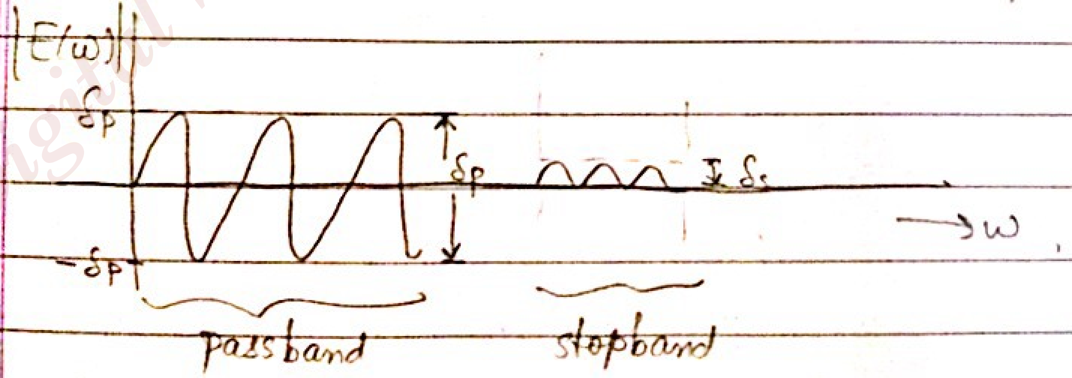
(we are trying to equalise the variations) something like:-



(ΔP need not be equal to ΔS)



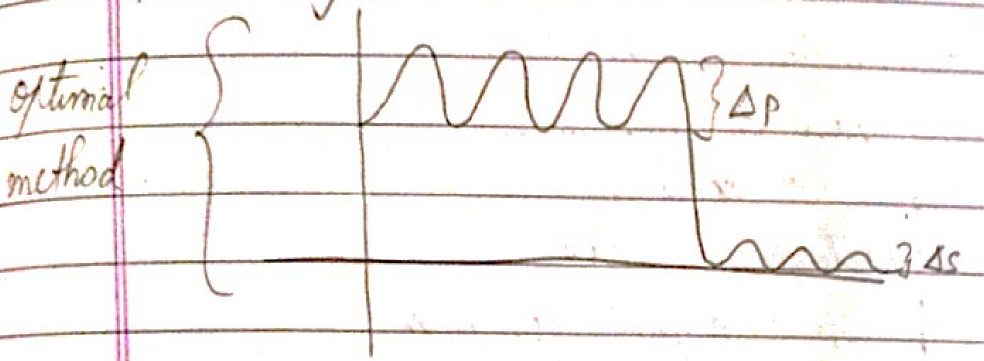
We need to reduce error s.t ∃ equal ripples



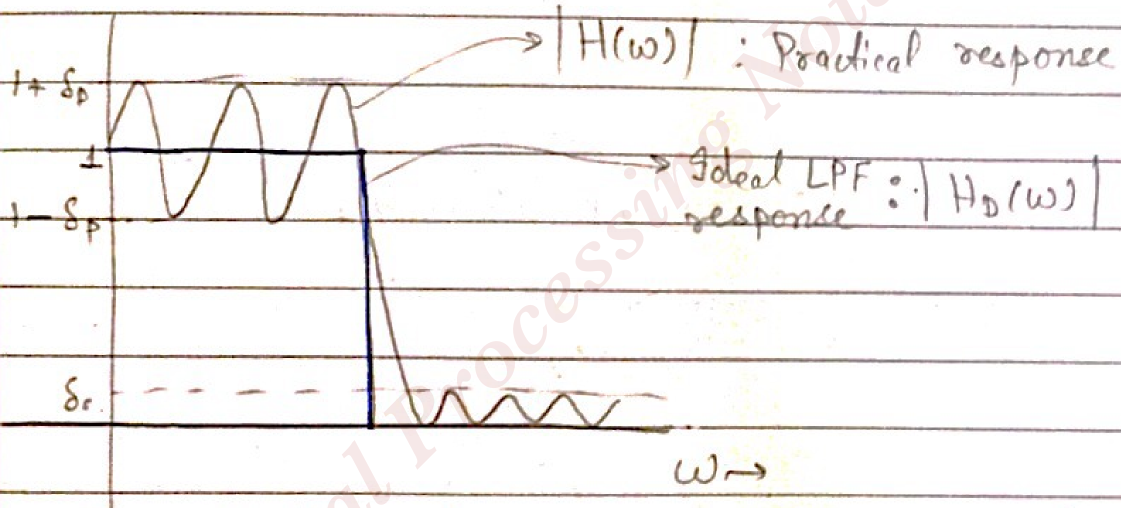
In optimal method, we want to find FIR coeffs, s.t. PB & SB ripple is UNIFORM.

(We are trying to equalise the variations)

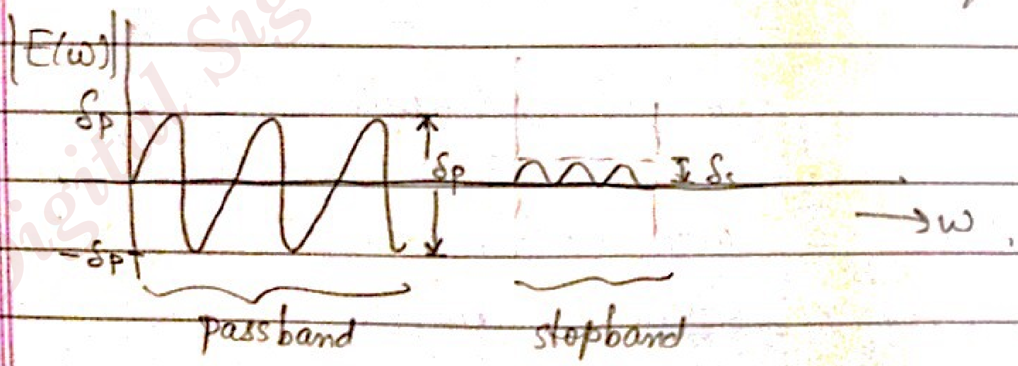
something like :-



(Δp need not be equal to Δs)

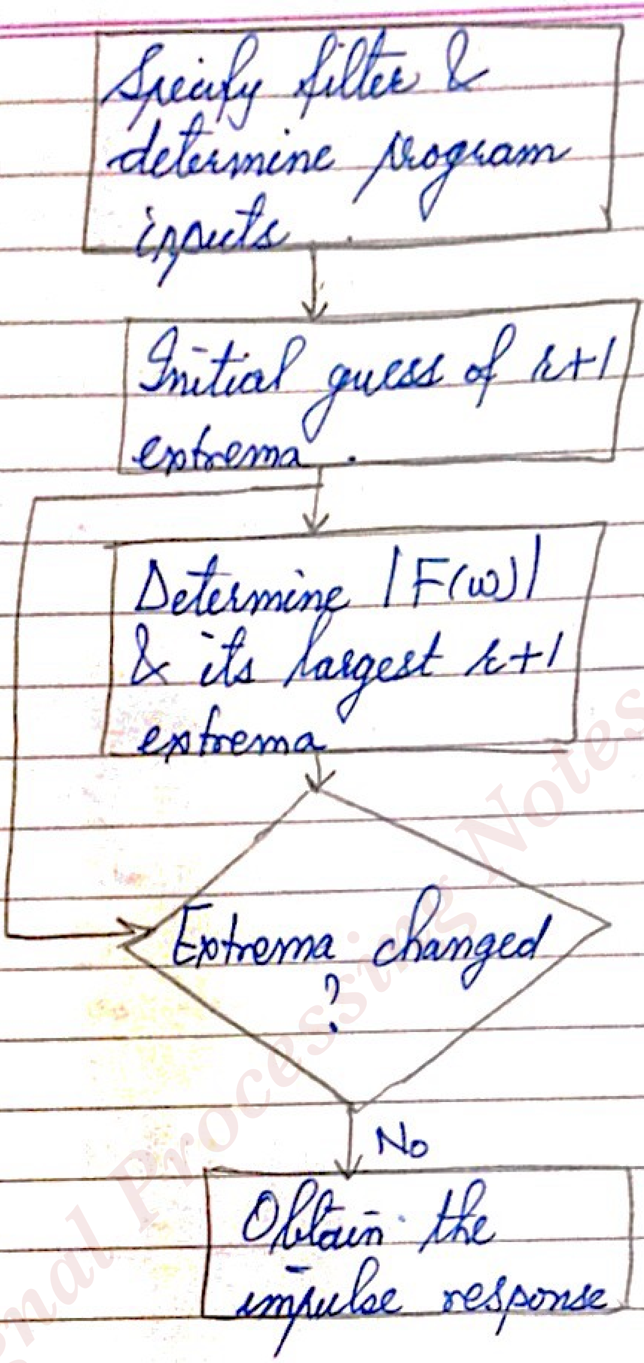


We need to reduce error s.t ∃ equal ripples



Computer Program Steps

A computational method



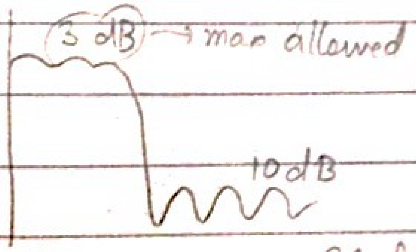
- Parameters req^d to use optimal program:

N : no. of filter coeff., i.e, filter length

J type: specifies type of filter (J=1, 2, 3)

W(ω) : Weighting f^n

We see, how weight ratio is given.



So, more weight has to applied to PB. So, ratio = 10:3

Puffin
Date
Page

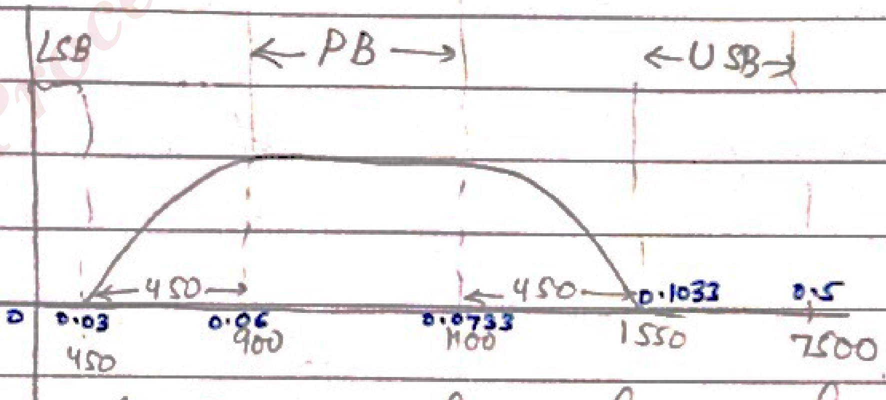
N grid : no. of pts. we are using to plot freq response

Edge : the edge freq.

eg A linear phase BP filter is to be designed
$\Downarrow$ FIR

given :- PB 900-1100 Hz
PB ripple < 0.87 dB
SB attenuation > 30 dB
sampling freq. 15 kHz
transn freq. 450 Hz.

So, we have



Values given to computer in normalised form, always

So, 450 ⟶ 450/15000 = 0.03
900 ⟶ 900/15000 = 0.06
1100 ⟶ 1100/15000 ⟶ 0.0733
1550 ⟶ 1550/15000 = 0.1033
7500 ⟶ 7500/1500 = 0.5

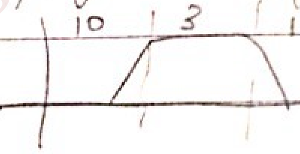
Now, choose weights : depending on PB & SB deviations
Weight : found by PB ripple & SB attenuation

0.87 dB ripple : $20 \log(1+\delta_p) \Rightarrow \delta_p = 0.10535$
30 dB attenuation : $-20 \log(\delta_s) \Rightarrow \delta_s = 0.031623$

$\delta_p \times 100 \simeq 10$
$\delta_s \times 100 \simeq 3$.

Idea So, ripple in PB is more (10), so, apply less weight(3) & attenuation in SB is less (3), so, apply more weight(10)

So, ratio $\delta_p : \delta_s = 10:3$



Using these values & computing :

Filter length, found as 40; for making odd,
let $N = 41$
J type = 1
weights $W(\omega) = 10, 3, 10$
N grid 32.
edge freq. :- 0, 0.03, 0.06, 0.0733, 0.1033, 0.5

⇒ +ve symm.
So, compute only
20 samples
+
center sample

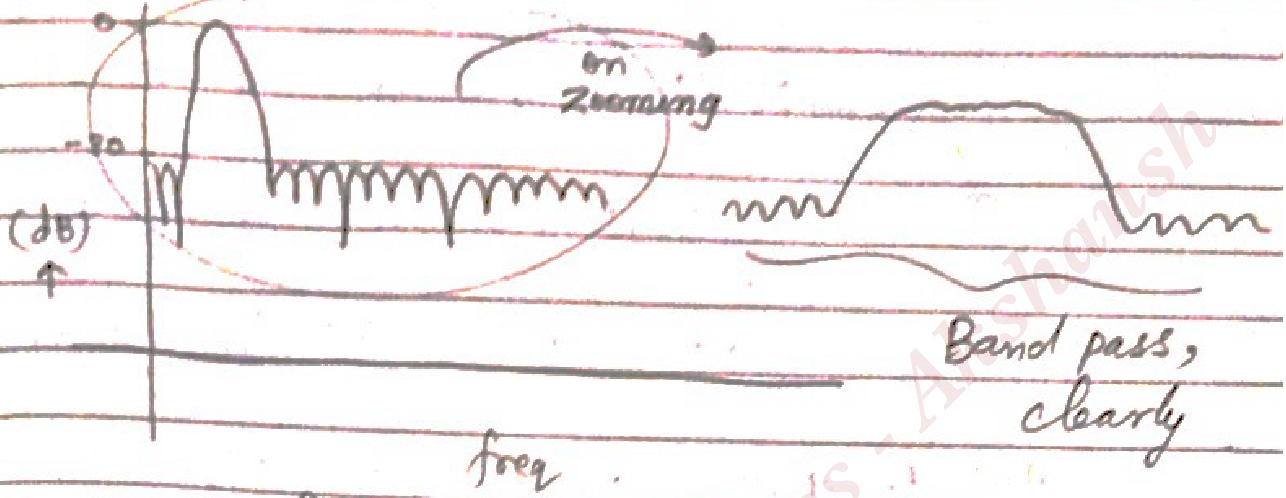
What do we get ?

Impulse response coeff.

$h(1) = \ldots = h(41)$ & corresponding
$h(2) \ldots = h(40)$ extreme frequencies.
$\vdots$
$h(21) = h(21)$

✱ N grid : can be seen on processor's speed (16, or 32 or 64)

Corresponding to impulse response coeff, we find freq. response (normalised)



Now, we have 2 SB & 1 PB

let SB$_1$ = Band 1
PB = Band 2
SB$_2$ = Band 3

Now, we calculated deviation values

$\delta_p = 0.1033$
$\delta_s = 0.031$

After computing & getting result, we get o/p showing summary of all the values (& deviations in graph got above

§ OPTIMAL METHOD FOR DESIGNING FIR FILTER

① ✱ for a LPF (low pass filter), emperical formula for finding N, using ~~optimiz~~ optimal method:

$$\underbrace{N}_{\text{no. of coeffs}} \simeq \frac{D_\infty(\delta_p, \delta_s)}{\Delta F} - f(\delta_p, \delta_s)\,\Delta F + 1$$

ripple in PB
ripple in stopband (SB)
continued

ΔF : width of transⁿ band normalised to sampling freq.

$$D_\infty(\delta_p, \delta_s) = \log_{10} \delta_s [a_1(\log_{10}\delta_p)^2 + a_2 \log_{10}\delta_p + a_3] + [a_4(\log_{10}\delta_p)^2 + a_5 \log_{10}\delta_p + a_6]$$

$$f(\delta_p, \delta_s) = 11.01217 + 0.51244[\log_{10}\delta_p - \log_{10}\delta_s]$$

XED
lues

$a_1 = 5.309 \times 10^{-3}$ $\qquad$ $a_4 = -2.66 \times 10^{-3}$

$a_2 = 7.114 \times 10^{-2}$ $\qquad$ $a_5 = -5.941 \times 10^{-1}$

$a_3 = -4.761 \times 10^{-1}$ $\qquad$ $a_6 = -4.278 \times 10^{-1}$

② ✻ for BPF, emperical formula :

$$N \simeq \frac{C_\infty(\delta_p, \delta_s)}{\Delta F} + g(\delta_p, \delta_s)\Delta F + 1$$

where

$$C_\infty(\delta_p, \delta_s) = \log_{10} \delta_s [b_1(\log_{10}\delta_p)^2 + b_2 \log_{10}\delta_p + b_3] + [b_4(\log_{10}\delta_p)^2 + b_5 \log_{10}\delta_p + b_6]$$

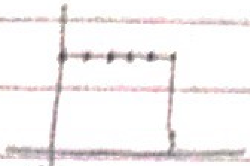
$$g(\delta_p, \delta_s) = -14.6 \log\left(\frac{\delta_p}{\delta_s}\right) - 16.9$$

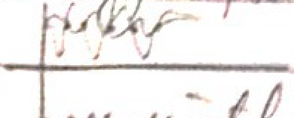
$b_1 = 0.01201$

$b_2 = 0.09664$

Method ③ : Frequency sampling method.

consider an ideal LPF

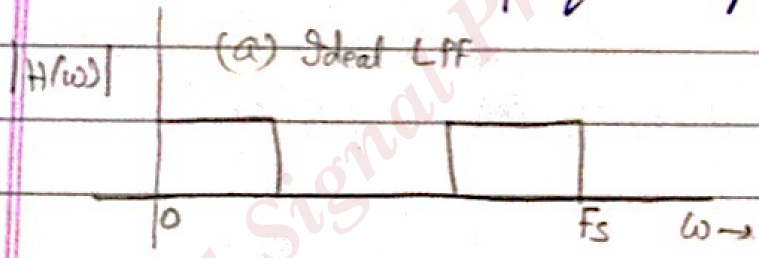


we are taking diff^t samples of freq. So, we have freq. values at diff^t pts. as shown above Now, the points in between  can vary in any way. So, deciding coeff. appropriately, we can design the req^d filter & limit the ripples in PB if req^d.

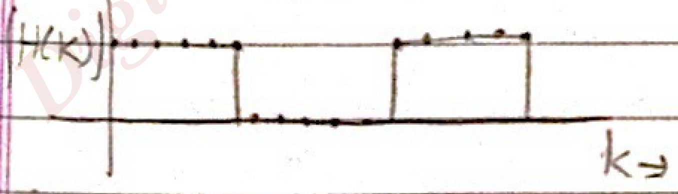
This method allows non-recursive FIR design both freq. selective and arbitrary freq. response filters.

Unique attraction is that this method allows recursive implementation leading to computationally efficient filters.

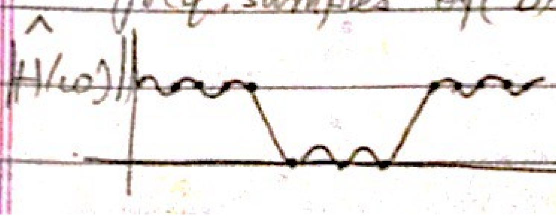
- Non recursive freq. sampling filters :



(b) Samples of ideal LPF



(c) freq. response derived from freq. samples of (b).



We have freq. at diff^t intervals ⇒ discrete intervals ⇒ H(Z) values (whose FT is H(ω))

Now, take F^{-1} to get h(n)

freq. response

$$h(n) = \frac{1}{N}\sum_{K=0}^{N-1} H(k)\, e^{j\left(\frac{2\pi}{N}\right)nk}$$

$$= \frac{1}{N}\sum_{k=0}^{N-1} \underbrace{|H(k)|}_{\text{mag.}}\, \underbrace{e^{-j\left(\frac{2\pi\alpha k}{N}\right)}}_{\text{phase}} \cdot e^{j\left(\frac{2\pi}{N}\right)n}$$

$$= \frac{1}{N}\sum_{K=0}^{N-1} |H(k)\, e^{j\frac{2\pi k(n-\alpha)}{N}}$$

$$= \frac{1}{N}\sum_{K=0}^{N-1} |H(K)| \cos\left[\frac{2\pi k(n-\alpha)}{N}\right]$$

Since h(n) is real, for linear phase, h(n) will be symmetrical. Assuming +ve symmetry & N is even,

$$\therefore h(n) = \frac{1}{N}\left[\sum_{K=1}^{\frac{N}{2}-1} 2|H(K)| \cos\left[\frac{2\pi k(n-\alpha)}{N}\right] + H(0)\right]$$

N is odd, upper limit changes to $\frac{N-1}{2}$

$\alpha = \frac{N-1}{2}$

Samp intervals

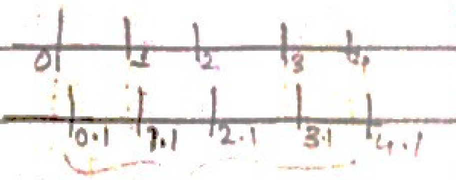
$\frac{kF_s}{N}$

$$\therefore h(n) = \frac{1}{N}\left[\sum_{K=1}^{\frac{N-1}{2}} 2|H(K)| \cos\left[\frac{2\pi k(n-\alpha)}{N}\right] + H(0)\right]$$

∴ freq. response is exactly same as sampling instant. Other than that instant, response is significantly diff^t. ∴ we must take a sufficient no. of freq. samples.

A diff^t freq sampling filter - type 2 will result if samples are taken at intervals of $f_K = \frac{(k+\frac{1}{2})}{N}$

eg: Suppose one sampling is next, shift it a bit & make type-2

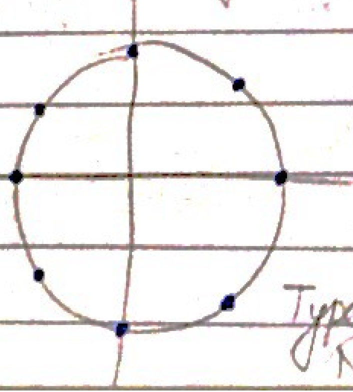


Starts at 0

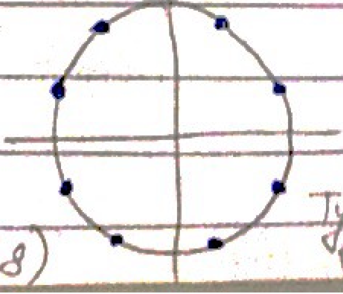
starts at 0.1

So, same type of sampling used to get diff^t samples

In terms of angle :

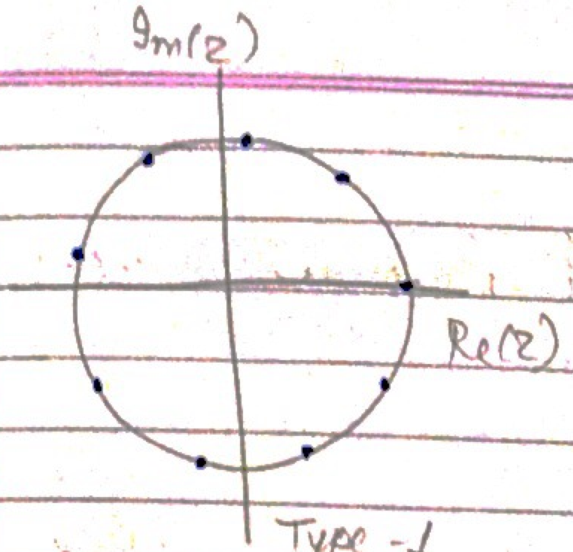


Type - 1
(N - even (8))

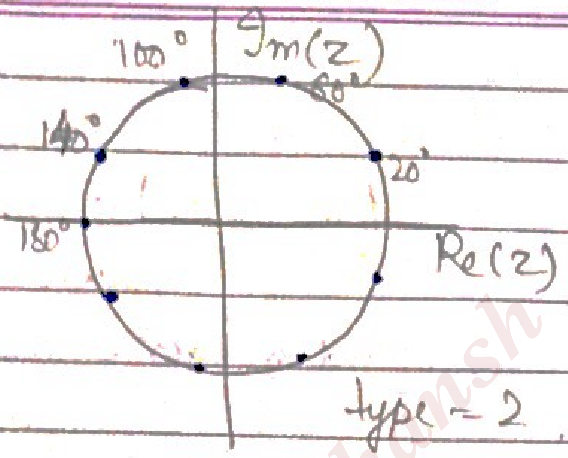


Type - 2
(N - even (8))

N : no. of samples



Type -1
N = 9 (odd)
(Start from 0° for type 1, always)



type - 2
Total pts = 9 = N

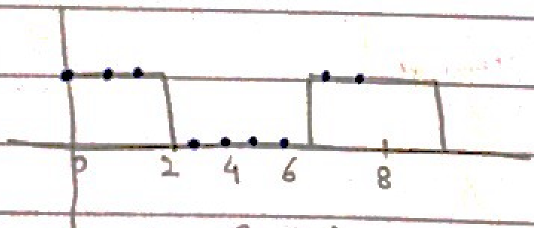
$\frac{360°}{9} = 40°$

for type 2, take $\frac{1}{2}$ of angle = 20°. So, start from 20°.

eg :- Requirement : PB = 0 – 5 kHz.
Fs = 18 kHz
N = 9

find FIR coeff (h(0)) using freq. sampling method.

Given : Ideal LPF :-



we have 9 pts. & f_s = 18 kHz.
So, sampling interval = $\frac{18}{9}$ = 2 kHz.
PB = 0 – 5 kHz. So, ∃ 3 pts at 0 kHz, 2 kHz & 4 kHz in the LPF

So, from above fig, seeing in terms of H(K) as :-

$$|H(k)| = \begin{cases} 1 , & k = 0, 1, 2 \\ 0 , & k = 3, 4 \end{cases}$$

Imp Note : It's symmetrical. So, only take till h(4).

Now, finding $h(n)$

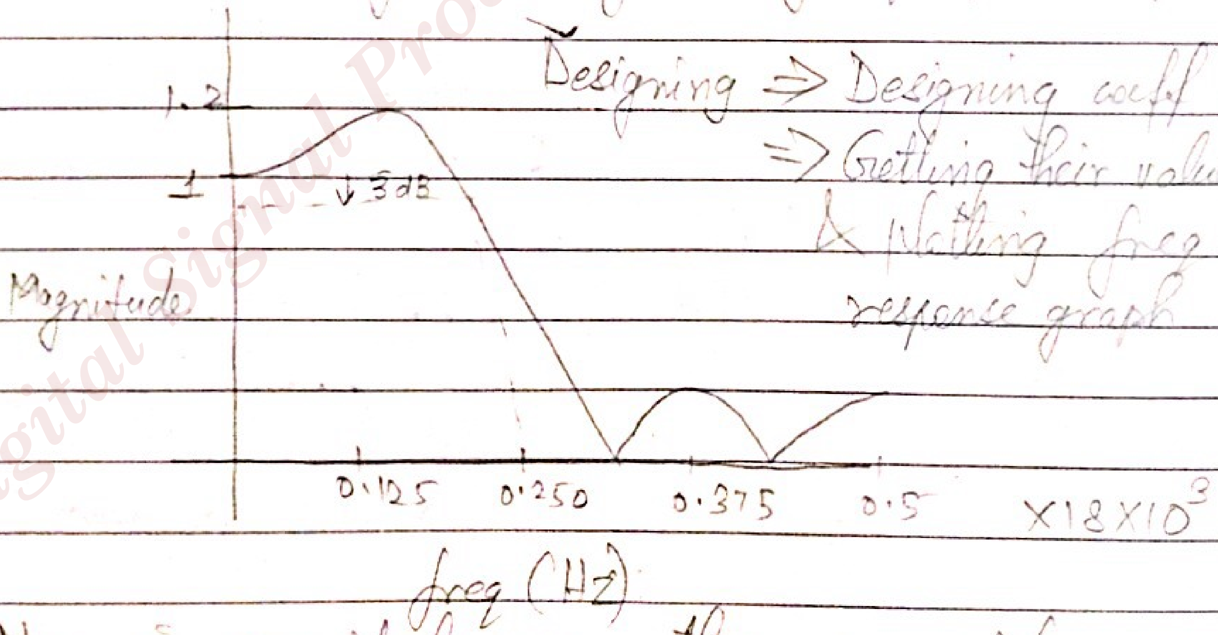
$$h(n) = \frac{1}{N}\left[\sum_{k=1}^{\frac{N-1}{2}} 2\left|H(k)\right| \cos\left[\frac{2\pi k(n-\alpha)}{N}\right] + H(0)\right]$$

Using values, we get

$h(0)$	7.2522627e-02	$h[8]$
$h(1)$	-1.1111111 e-01	$h[7]$
$h[2]$	-5.9120987e-02	$h[6]$
$h[3]$	3.1993169e-01	$h(5)$
$h[4]$	5.5555556e-01	$h[4]$

Now, seeing if these coeff.s are correct ?
Checking done by seeing freq. response.

Designing ⇒ Designing coeff.
⇒ Getting their values
& Noting freq. response graph



Now, in non ideal case, there may exist some samples in transn band.

☆ Optimising the amplitude response in freq. sampling method:-

In window method, wider trans^n width gives improved amplitude response.

Illy, in freq. sampling method, we can allow more samples in trans^n band to have more attenuation.

For a LPF, stopband attenuation varies approx. 20 dB for each trans^n band freq. sample.

approx. stopband attenuation : ~~20M~~ $(25+20\,M)$ dB.

" trans^n width : $(M+1)\frac{F_s}{N}$.

M : no. of freq. sample in trans^n band

N : filter length

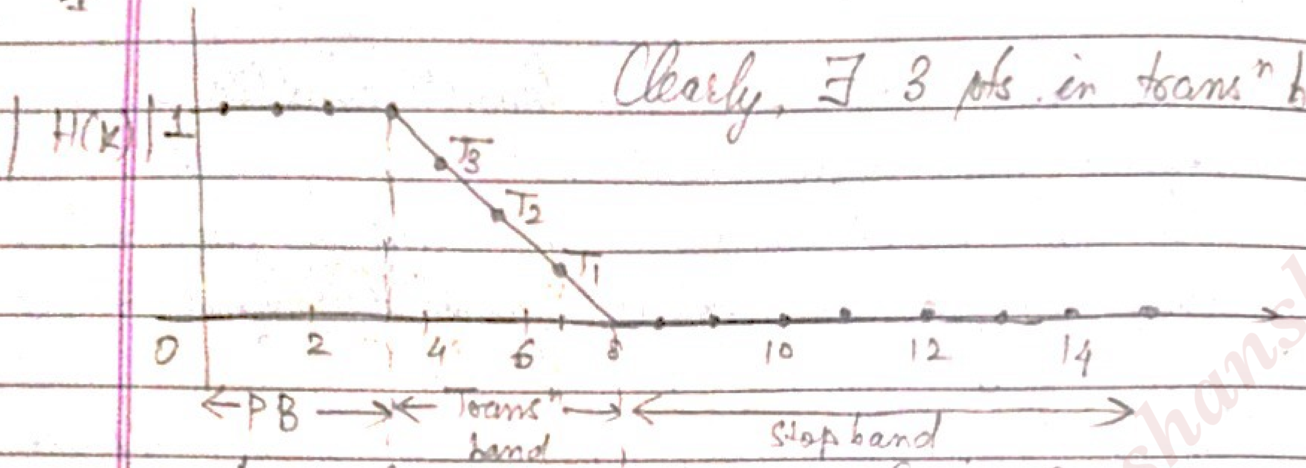
The value of trans^n band freq. samples that will give optimum stopband attenuation are determined by Optimiz^n process. The desired property can be mathematically represented as :-

$$\underset{(T_1, T_2, \ldots T_M)}{\text{minimise}} \left[\max \left| \omega \left[H_0(\omega) - H(\omega) \right] \right| \right]$$

☆ Depending on N & no. of samples in trans^n band, optimised value will differ.

eg Consider LPF : It's a 15 pt FIR filter

Clearly, ∃ 3 pts. in trans^n band.



Any value in trans^n band has value b/w 0 to 1

↳ Observed values :

- If ∃ one value in trans^n band, its value lies b/w $0.25 < T_1 < 0.45$
- 2 values in trans^n band
  $0.04 < T_1 < 0.15$
  $0.45 < T_2 < 0.65$
- 3 values in trans^n band
  $0.003 < T_1 < 0.035$
  $0.100 < T_2 < 0.300$
  $0.55 < T_3 < 0.75$

eg Consider a 15 pt. FIR filter.

Given freq responces:-

$$|H(k)| = \begin{cases} 1 ; & k = 0, 1, 2, 3 \\ 0 ; & k = 4, 5, 6, 7 \end{cases}$$

→ taking only half $\left(\frac{15}{2}\right)$

assume $f_s = 2$ kHz.

① Obtain its freq responce.

Idea:- find h(n) & plot graph.

$$3 = 20 \log_{10}(1/\sqrt{2})$$

② Compare freq. response of filter if
(a) One sample lies in transn band
(b) 2 " " "
(c) 3 " " "

Basically, see how the design varies.

② (a) |H(k)| changes as

$$|H(k)| = \begin{cases} 1 & k = 0,1,2,3 \\ \boxed{0.40406} & k = 4 \\ 0 & k = 5,6,7 \end{cases}$$

(from table 7.11)
See the no. (k) in B/W column
Note down its SB attenuation & its value (T_1)

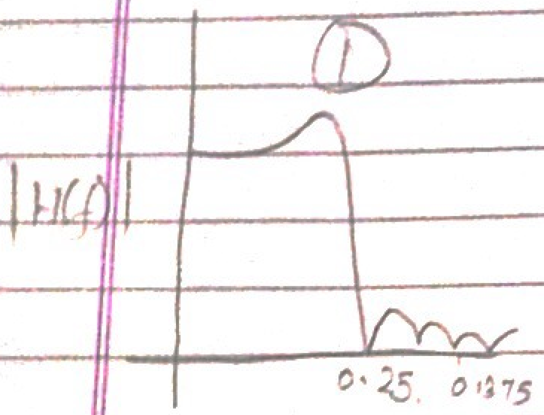
having known these values, put in eqn & find h(n) & plot.

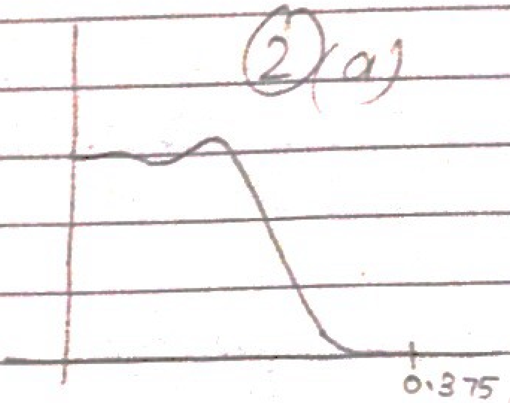
(b) ||ly for 2 samples

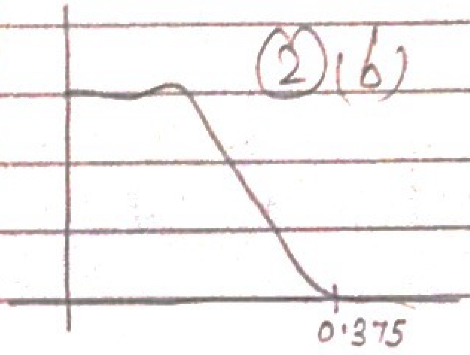
$$|H(k)| = \begin{cases} 1 & k = 0,1,2,3 \\ 0.084 & k = 4 \\ 0.557 & k = 5 \\ 0 & k = 6,7 \end{cases}$$

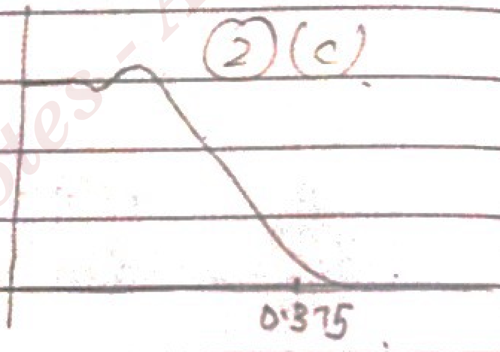
(c) ||ly for 3 samples.
k = 4, 5, 6 in transn band.
Note values from table & make freq. response.

Results:


①


②(a)


②(b)


②(c)

Inference :- Trans^n band is increasing as no. of samples in it increase.

PHYSICALLY

Say, 3 dB variation in PB. So, if max. value is 100% or 1, what is value at low 3 dB i.e, $-3\text{dB} \rightarrow$

$$* \ (-)3\text{ dB} = 20 \log_{10}(\text{value}) = 0.707.$$

Now, similarly, if SB attenuation = 40 dB, what is value at SB?

$$* \ (-)40\text{ dB} = 20 \log_{10}(\text{value}) = 0.01.$$

So, value = 0.01 at SB (goes down from 1 to 0.01)

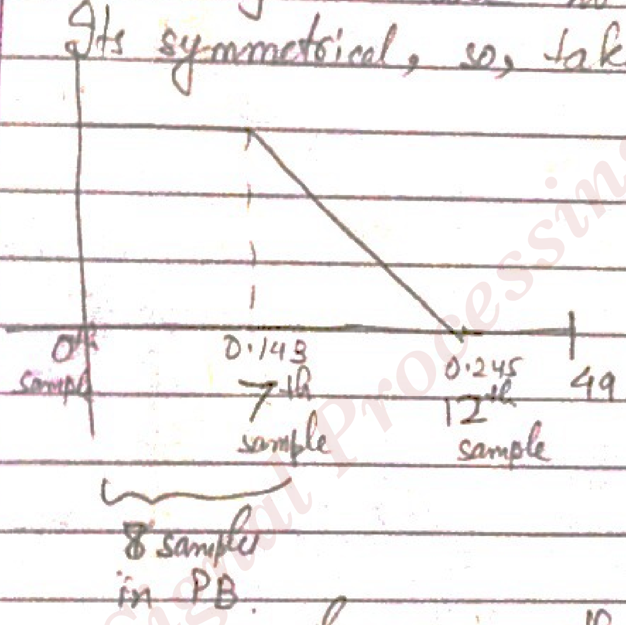
Q. Find optimum freq. samples of transⁿ band freq given :-

PB edge freq = 0.143 (normalised)
SB edge freq = 0.245 (normalised)
no. of filter coeff = 49

⇒ fs not reqᵈ

Idea : Find the coeff no. corresponding to PB & SB, i.e what no. coeff is corresponding to 0.143 & 0.245 (say, x & y). So, y - x are no. of samples in transⁿ band

Its symmetrical, so, take 25 samples



for 1 ⟶ 49
0.143 = 49 × 0.143 = 7.005
= 7th sample at PB edge
0.245 = 49 × 0.245
= 12th sample at SB edge

So, 8th, 9th, 10th & 11th sample in transⁿ band.

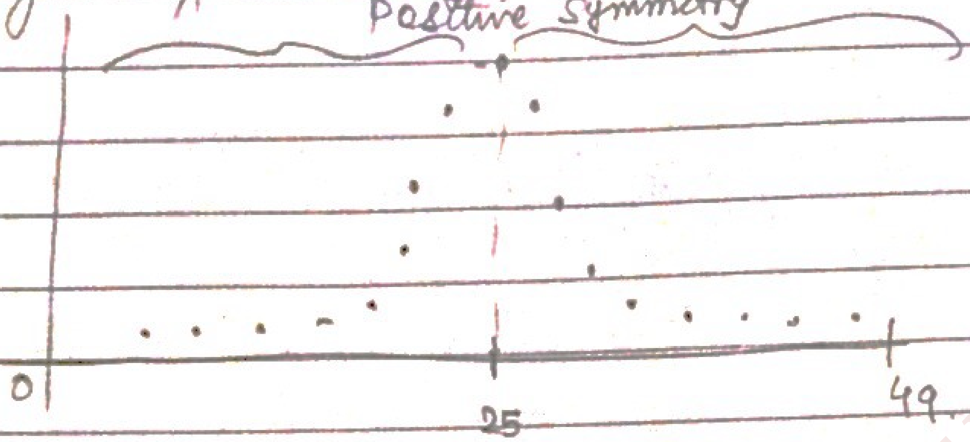
So, total 4 samples in transⁿ band.

finding value corresponding to the sample

So, 8th ⟶ $\frac{8}{49}$, - - -

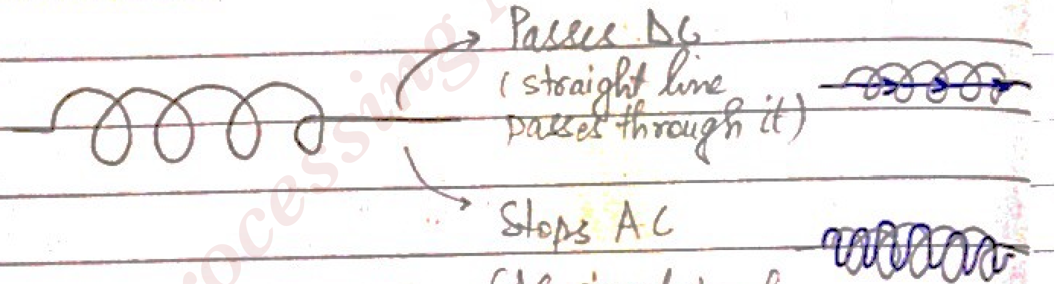
Using these values,
check if the given freq response (in any problem) can be got

Note : In case of a value 7.005, choose 7th or 8th → see condⁿˢ final o/p something like



★ Way to remember :-

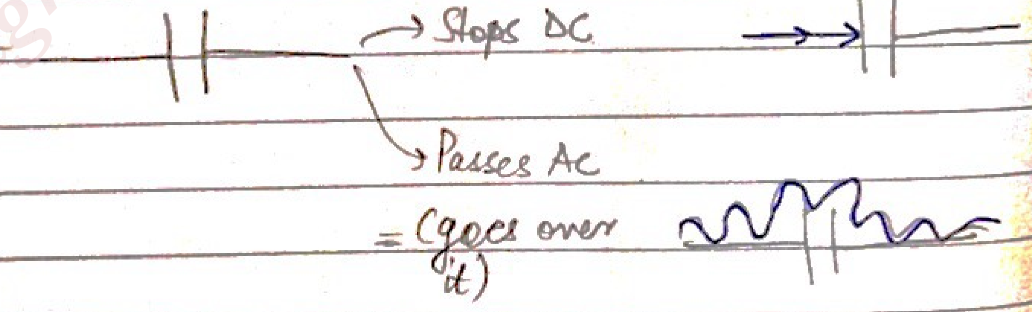
Inductor :



(straight line passes through it)

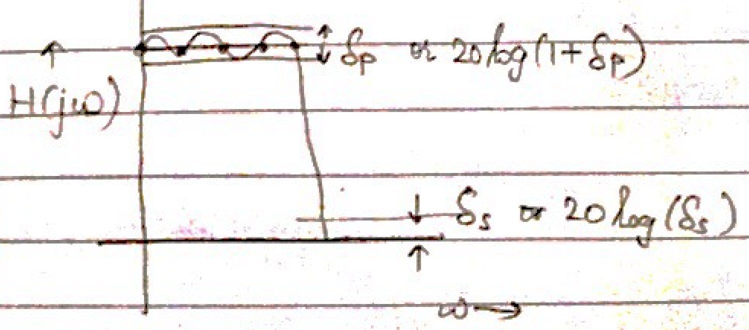
(AC signal signal gets sort of jumbled up)

Capacitor :-



= (goes over it)

★



✭ Recursive freq. sampling method.

⟹ freq. sampling with FEEDBACK

$$h(n) = \frac{1}{N}\sum_{k=0}^{N-1} H(k)\, r^n e^{j(2\pi nk/N)} \; ; \; k = 0, 1, \ldots N-1$$

based on no. of coeff

$\boxed{r \leq 1}$ → radius of unit circle; for $r \leq 1$, sys. stays stable

↳ Discrete inverse FT.

Now,

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

$H(z)$ → Discrete freq. domain

$h(n)$ → Discrete time domain.

Interchanging limits

$$H(z) = \frac{1}{N}\sum_{k=0}^{N-1} H(k)\left\{\underbrace{\sum_{n=0}^{N-1}\overbrace{\left(r e^{j(2\pi k/N)} z^{-1}\right)^n}^{(\delta)^n}}\right\}.$$

Now, for finite GP, we can say,

$$\text{Sum} = S_N = \sum_{n=0}^{N-1} \delta^n = \frac{1-\delta^N}{1-\delta} \; , \; \delta \neq 1$$

(δ : common ratio of GP)
$= r e^{j2\pi k/N} z^{-1}$

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

$$\left(= \frac{\text{Numerator}}{\text{Denominator}}\right)$$

$$\left(= \frac{N \text{ zeros on/inside unit circle } (r=1)\ (r<1)}{1 \text{ pole on/inside unit circle}}\right)$$

So, ∵ $e^{j2\pi k} = 1$ $\left(\cos(2\pi k) + j(0)\right)$

$$\sum_{n=0}^{N-1} \delta^n = \frac{1 - r^N z^{-N}}{1 - re^{j2\pi k/N} z^{-1}}$$

Now,

$$H(z) = \frac{1}{N} \sum_{k=0}^{N-1} H(k) \times \left(\frac{1 - r^N z^{-N}}{1 - re^{j2\pi k/N} z^{-1}} \right)$$

$$= \left(\frac{1 - r^N z^{-N}}{N} \right) \sum_{k=0}^{N-1} \frac{H(K)}{1 - re^{j2\pi K/N} z^{-1}}$$

$$= \left(\text{Poly. in } -\text{ve power of } z\right) \times \left(\frac{1}{\text{Poly. in } -\text{ve power of } z} \right)$$

$$\left\{ = \frac{\text{Numerator}}{\text{Denominator}} \quad \text{✱ So, feedback sort of seen} \right\}$$

$$\Rightarrow H(z) = H_1(z) \cdot H_2(z)$$

↳ $H_1(z) = \dfrac{1 - r^N z^{-N}}{N}$ → Radius of zero locⁿ

$$H_2(z) = \sum_{k=0}^{N-1} \frac{H(k)}{1 - re^{j2\pi k/N} z^{-1}}$$

→ Radius of pole locⁿ ($r < 1$, mostly, for stability)

↳ $k \in Z$, N: no. of samples

$\Rightarrow \dfrac{k}{N} \notin Z$

$$= \frac{H(0)}{1 - rz^{-1}} + \frac{H(1)}{1 - re^{j2\pi/N} z^{-1}} + ---- + \frac{H(N-1)}{1 - re^{j2\pi(N-1)/N} z^{-1}}$$

* N zeroes ⇒ sth like 

So, looks like comb

if $r = 1$

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

$$= H_1(z) \cdot H_2(z)$$

(cascaded)

So, ∃ N zeroes around unit circle (Comb filter)
& ∃ N sections of single poles filter with $z_k = e^{j2\pi k/N}$

∴ Poles exactly cancel the zeroes
(each change in zeroes is compensated by pole loc^n)

Wordlength effects:

↳ doesn't cancel the zeroes exactly thus, making FIR filter, an IIR and thus, unstable.
Stability issue is solved by sampling H(z) at a radius slightly less than unity.

Consider $H_2(z)$

If poles are complex, then residues H(k) will also be complex. But, H(k) (freq. response) can't be imaginary. So, we take double poles (as, complex poles are in complex conjugate).

∴ So, we take 2nd order ($\frac{N}{2}$ double pole) sections instead of N sections of single poles

So, in this case, $H_2(z)$ changes to

$$H_2(z) = \frac{H(0)}{1 - rz^{-1}} + \underbrace{\frac{H(1)}{1 - re^{j2\pi/N} z^{-1}} + \frac{H^*(1)}{1 - re^{-j2\pi/N} z^{-1}}}_{\text{one Double pole section}} + \overbrace{\frac{H(2)}{1 - re^{j2\pi 2/N} z^{-1}} + \cdots}$$

* If integer or powers of 2 coeff. are used, computational efficiency is improved. Its only possible if ∃ restriction on pole loc^{ns} i.e at specific freq.

Thus, poles are seen in conjugate pairs

For linear phase filters of even length, $H(N/2) = 0$

Combining k^{th} pole & its conjugate

$$\Rightarrow \frac{H(k)}{1 - re^{j2\pi k/N} z^{-1}} + \frac{H^*(k)}{1 - re^{-j2\pi k/N} z^{-1}}$$

Taking LCM & solving,

simplifying denominator :-

$$(\quad)(\quad) = 1 - 2r\cos\left(\frac{2\pi k}{N}\right) z^{-1} + r^2 z^{-2}$$

For linear phase filter, $H(k) = |H(k)| e^{-j2\pi k\alpha/N}$

$$\hookrightarrow \alpha = \frac{N-1}{2}$$

& simplifying numerator :

we get

$$|H(k)| \left\{ 2\cos\left(\frac{2\pi k\alpha}{N}\right) 2r\cos\left(\frac{2\pi k(1+\alpha)}{N}\right) z^{-1} \right\}$$

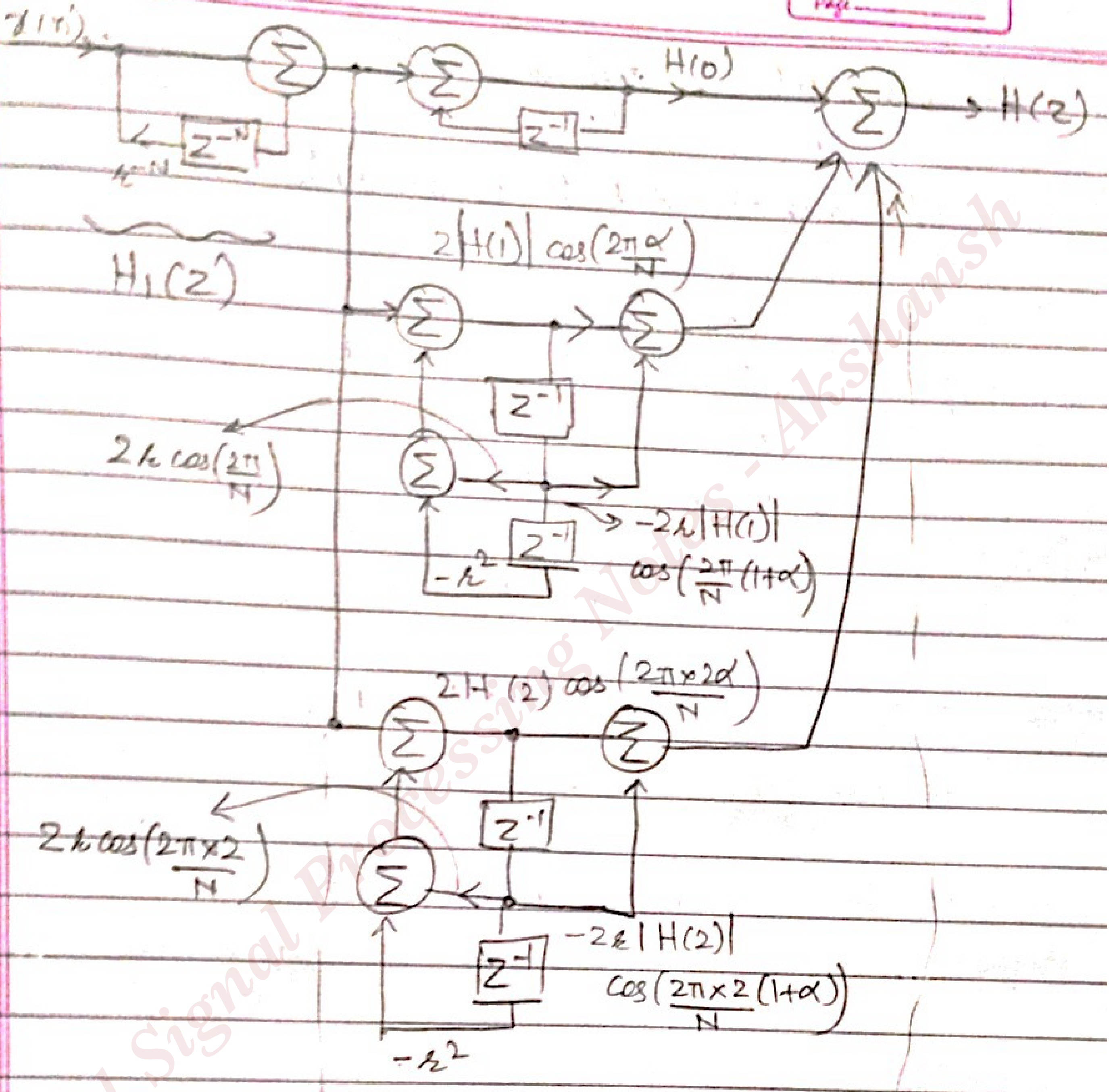
Combining numerator & denominator,

$$☆\; H(z) = \underbrace{\frac{1 - r^N z^{-N}}{N}}_{H_1(z)} \left\{ \sum_{k=1}^{M} \frac{|H(k)| \left\{ 2\cos\left(\frac{2\pi k\alpha}{N}\right) 2r\cos\left(\frac{2\pi k(1+\alpha)}{N}\right) z^{-1} \right\}}{1 - 2r\cos\left(\frac{2\pi k}{N}\right) z^{-1} + r^2 z^{-2}} + \frac{H(0)}{1 - rz^{-1}} \right\}$$

assuming M: double poles

$$\hookrightarrow \text{for } N = \begin{cases} \text{odd}, & M = \frac{N-1}{2} \\ \text{even}, & M = \frac{N}{2} \end{cases}$$

Implementing



||ly, for M filters.

$H_2(z)$

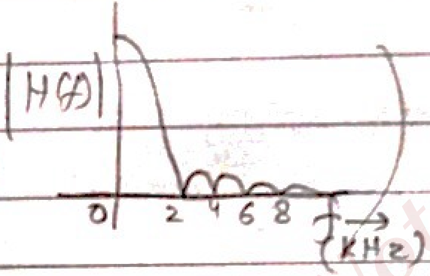
Q. Obtain TF & difference eq^n.

① a recursive FIR Lowpass filter with simple integer coeff. meeting following specs :-

centre freq :- 0 Hz

sampling freq :- 18 kHz

(i.e, we are asking something like

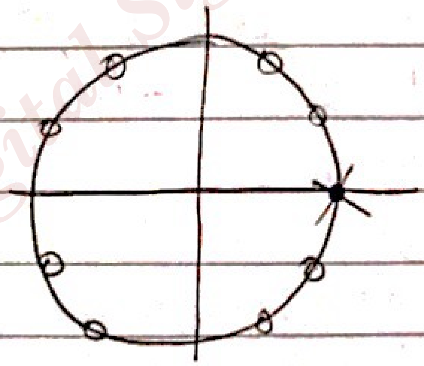


② a recursive FIR bandpass filter with simple integer coeff. meeting following specs :-

centre freq :- 3 kHz

sampling freq :- 12 kHz

① Assume we take 9 samples. So, placing them on unit circle, we get



$$H(z) = \underbrace{\left(\frac{1-z^{-9}}{9}\right)}_{\substack{\text{for } k=1 \\ N=9 \\ \text{for } H_1(z)}} \underbrace{\left(\frac{1}{1-ke^{0}z^{-1}}\right)}_{\text{for } H_2(z)}$$

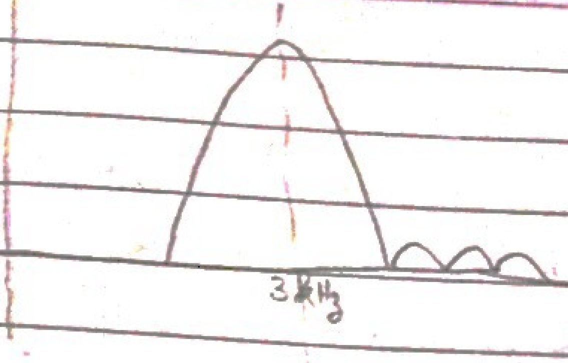
for $H_2(z)$ keeping k = 0, i.e, pole at 0°

∵ we want centre freq. at 0 Hz

Diff. eq^n :-

$$y(n) = y(n-1) + \frac{1}{9}\left(x(n)-x(n-9)\right)$$

②



Its reqd to keep a pole at 3 kHz

And there should be a pole corresponding to 3kHz on unit circle.

Assuming 8 samples So, $\frac{12}{8} = 1.5$ kHz.

Hence, each sample at 1.5 kHz (So, on 2nd sample, 3 kHz comes, as wanted).

Now, for z plane, for 360° & 8 samples, each sample is at $\frac{360}{8} = 45°$.

Now, we want pole at 2nd sample. &, So, at angle = 90°, we get 3 kHz sample.

(0, 1.5 kHz, 3 kHz
0°, 45°, 90°)

So, 3 kHz sample comes on Imaginary axis.

Hence, its conjugate will be there at 270° also.

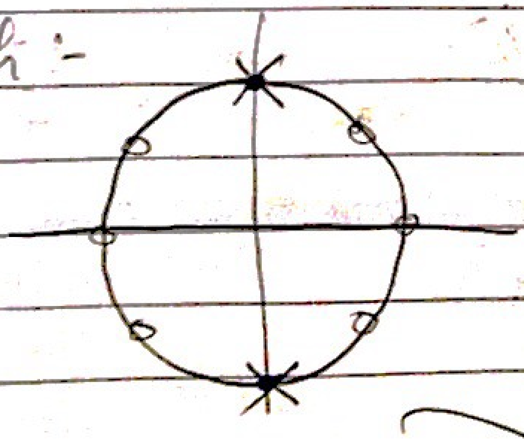
Hence, we have

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

Diff eqn:-

$$y(n) = -y(n-2) + \frac{1}{8}\left(x(n) - x(n-8)\right)$$

& graph :-



Q. A LPF with spec :-

PB : 0 - 4 kHz.

$f_s = 18$ kHz

length of filter = 9

Find :- TF of filter in recursive form using freq. sampling method.

(Assume $r = 1$.)

Draw realizⁿ diagram & compare computⁿᵃˡ complexities with direct form FIR.

Solⁿ :-

$$\text{Sampling interval} = \frac{18\text{ kHz}}{9} = 2\text{ kHz}.$$

So, samples are at 0 kHz, 0+2 kHz, 0+2+2 kHz, ------

$$\text{So, } H(k) = \begin{cases} 1 & ; k = 0, 1, 2 \\ 0 & ; k = 3, 4 \end{cases}$$

∵ LP Band is 0-4 kHz, given

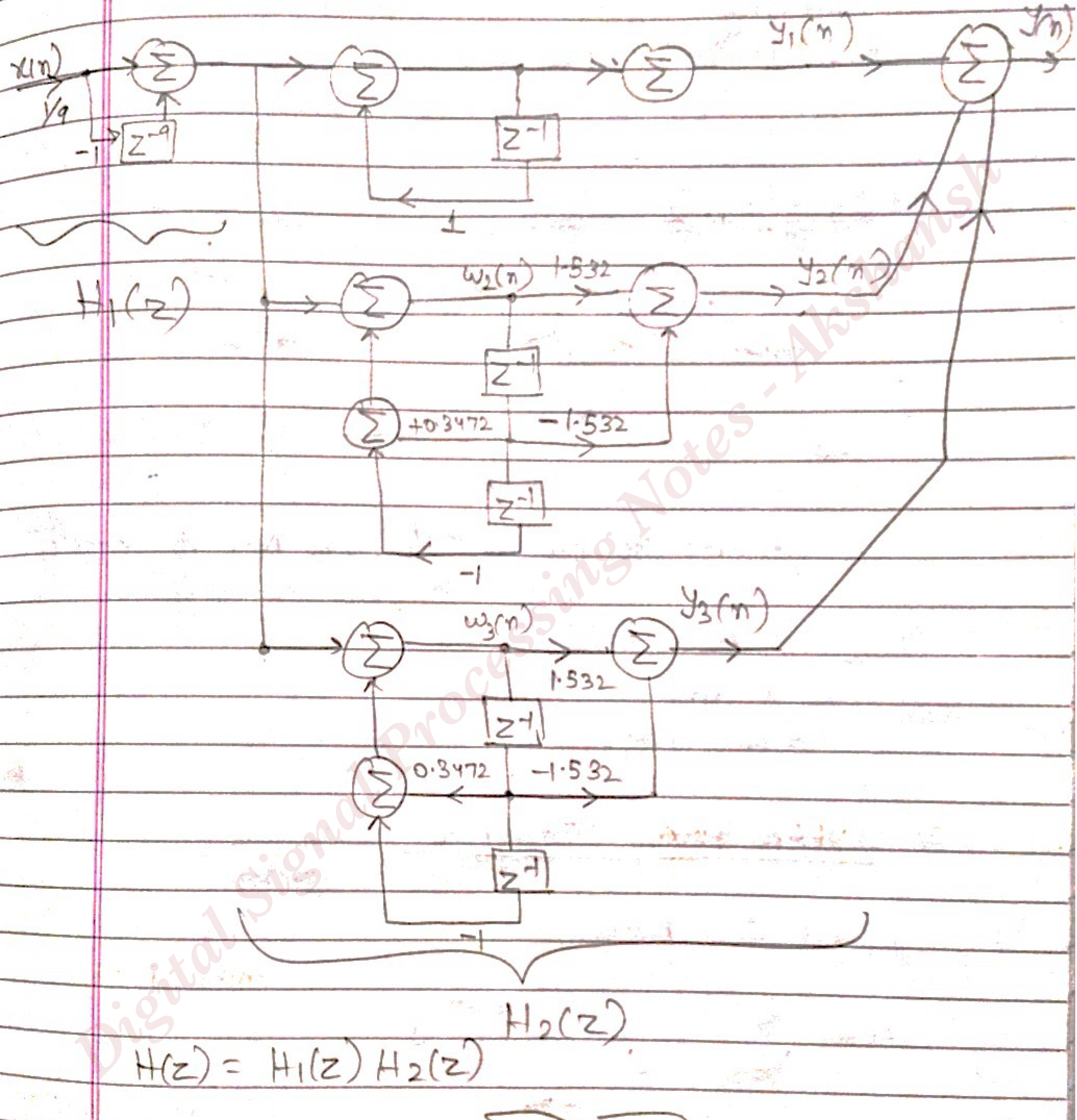
Now,

$$H(z) = \frac{1 - z^{-9}}{9}\left\{ 2|H(1)|\left(- \; - \; - \right) + 2|H(2)|\left(- \right) + \underbrace{\frac{1}{1 - z^{-1}}}_{H(0)} \right\}$$

↳ using the expression derived before.

$$\Rightarrow H(z) = \underbrace{\frac{1 - z^{-9}}{9}}_{H_1(z)} \underbrace{\left[\frac{-1.8794(1 - z^{-1})}{1 - 1.532z^{-1} + z^{-2}} + \frac{1.532(1 - z^{-1})}{1 - 0.3472z^{-1} + z^{-2}} + \frac{1}{1 - z^{-1}} \right]}_{H_2(z)}$$

Canonical form represent^n :

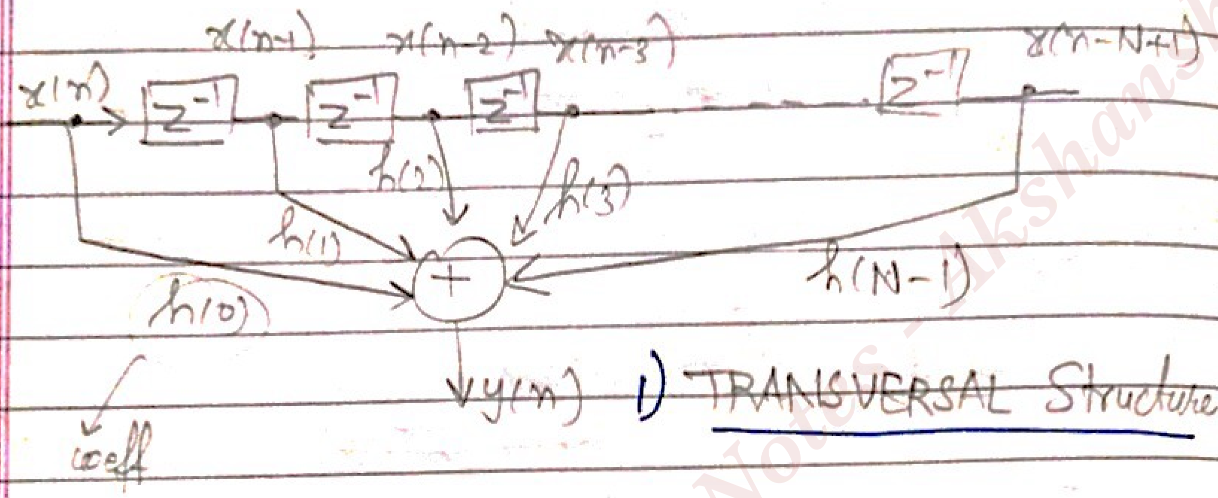


$H(z) = H_1(z)\,H_2(z)$

* Impulse response coeff. of FIR sys. are symmetrical.

✯ Realisation structures of FIR filters

$$H(z) = \sum_{n=0}^{N-1} h(n) z^{-n} \; ; \; y(n) = \sum_{m=0}^{N-1} h(m) x(n-m)$$



1) TRANSVERSAL Structure.

Uses :
- N-1 memory loc^{ns} to store N-1 i/p samples
- N memory loc^{n} to store N coeff.
- N multipliers
- N-1 addition.

hardware device requirements to implement N point FIR filter.

2) LINEAR PHASE STRUCTURE (take advantage of symm.)

⇒ Suppose N = 13

$h(0) = h(13)$

$h(1) = h(12)$

$$h(n) = \pm h(N-n-1)$$

$$H(z) = \sum_{n=0}^{\left(\frac{N-1}{2}\right)-1} h(n)\left[z^{-n} + z^{-(N-1-n)}\right] + h\left(\frac{N-1}{2}\right) z^{-\left(\frac{N-1}{2}\right)}$$

↳ for N = odd

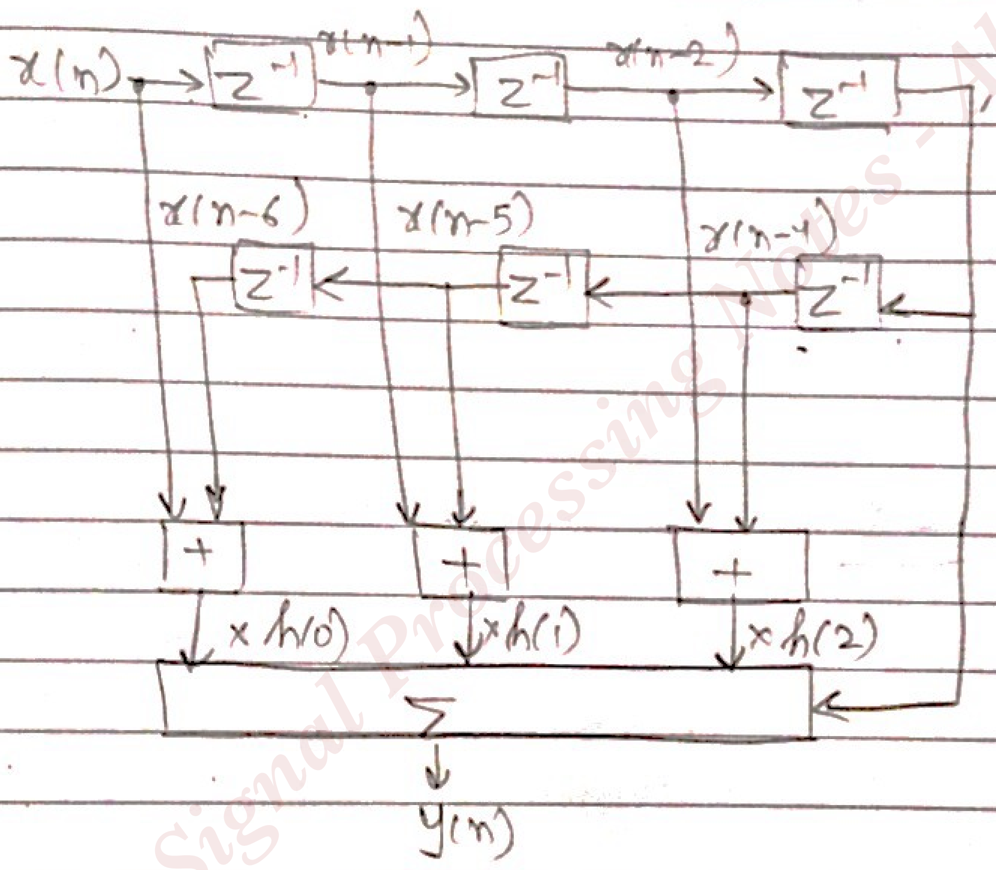
So, we are multiplying $x(n)\, h(n-n)$ we needed N multipliers above. Now, we'll need $\frac{N}{2}$ or $\frac{N-1}{2}$

or

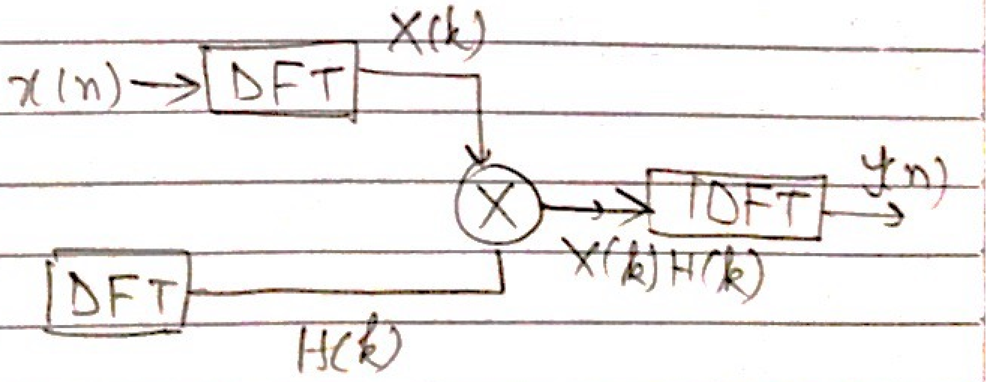
$$H(z) = \sum_{k=0}^{\frac{N}{2}-1} h(k)\left[x(n-k) + x[n-(N-1-k)]\right]$$

→ for n = even

eg : N=7 : 7 pt FIR



3) DFT METHOD



FINITE WORD LENGTH EFFET ON IIR FILTERS

Direct form I : Considering 2nd order section

o/p, $y(n) = \sum_{i=0}^{2} b_i x(n-1) - \sum_{i=1}^{2} a_i y(n-i)$

$\hookrightarrow$ (difference eq^n)

&

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} = \frac{num}{den}$$

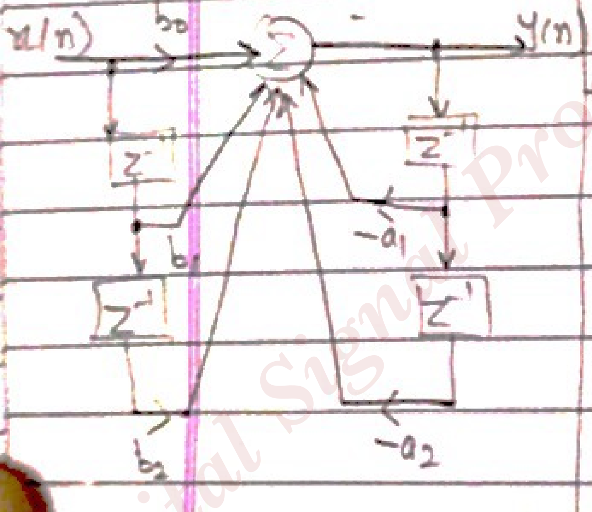
- → 2nd ord. std. TF.
- → ∃ 5 filter coeff.
  - ⇒ 5 memory loc^ns are kept to store them
- → $H(z) = \frac{Y(z)}{X(z)} = \frac{num}{den}$

⇒ $Y(z)\,den = X(z)\,num$

Taking Z^{-1}

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

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

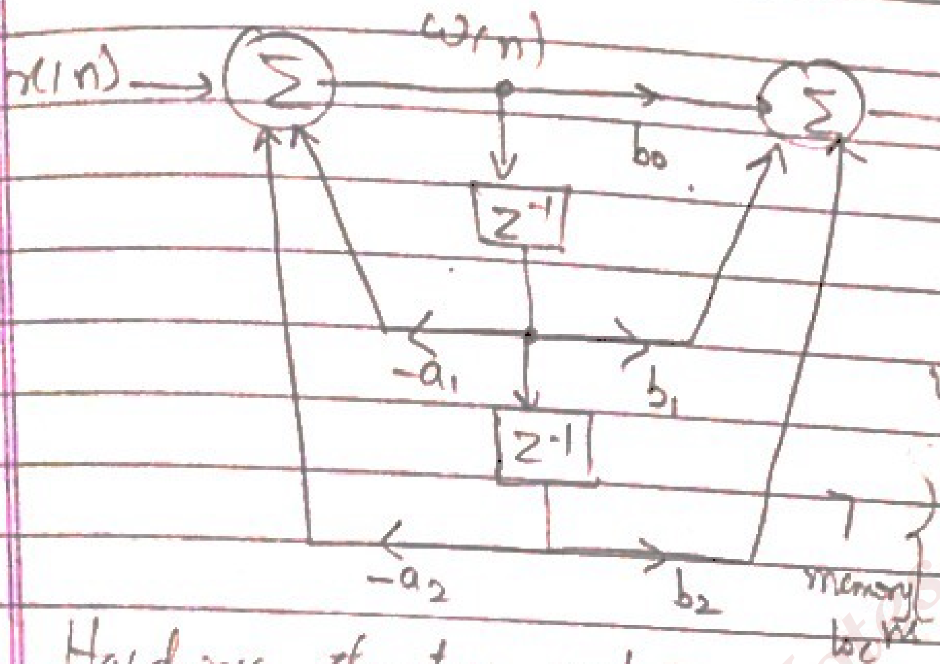


↳ 9 memory loc^ns to store data

9 $\{b_0, b_1, b_2, a_1, a_2, x(n-1), x(n-2), y(n-1), y(n-2)\}$

- → delay elements (z^{-ve} terms)
  - 2 in num & 2 in den ⇒ Total 4
- → 1 adder (4 additions)
- → 1 quantiz^n pt. for sum of product.
- → 1 multiplier (5 multiplic^ns)

Direct form II :-



We need to store w(n-1), w(n-2), b₀, b₁, b₂, a₁, a₂

Hardware structure reqd :-

- ↳ 5 filter coeff
- ↳ 2 delay elements
- ↳ 2 adders (4 additions)
- ↳ 2 quantiz^n pt. for SOP.
- ↳ 7 memory loc^ns

• Quantiz^n effect on IIR filters :

given, a BPF (digital) with $f_s = 153.6\ kHz$
(where

$$H(z) = \frac{1}{1 + a_1 z^{-1} + a_2 z^{-2}}$$ (single 2nd order section)

where $a_1 = -1.957558$, $a_2 = 0.995913$.

Now, quantize the coeff. to 8 bits & see effect on pole loc^ns & hence, on centre freq

$\hookrightarrow r = \sqrt{a_2}$, $\theta = \cos^{-1}\left(-\frac{a_1}{2r}\right)$

(finding r & θ from given coeff. of TF)

M2 :- find poles & use calculator to convert to polar

Here, we get

$r = \sqrt{0.995913} = 0.99795$

$\theta = \cos^{-1}\left(\frac{1.957558}{2\times 0.99795}\right) = 11.25^\circ$

Corresponding centre freq.

$= \left(\frac{11.25}{360}\right)\times 153.6\times 10^3 = 4.799\ kHz$.

Now, representing in 8 bits

↳ 1 bit : signed bit

1 bit : to represent integer part

6 bits : to represent fractional part

↳ sensitivity = 2^6.

So, $a_1 = -1.957558\times 2^6 = -125.23$

Truncated coeff = −125.

$a_2 = 0.995913\times 10^6 = 63.8$.

Truncated coeff = 63

(we are truncating, not rounding off)

So, $a_1 = -\frac{125}{2^6} = -1.953125$

$a_2 = \frac{63}{2^6} = 0.984375$

} Change of coeff. when 8 bits representⁿ is used.

So, new r & θ values are

$r = 0.992156$ ($\sqrt{a_2}$)

$\theta = \cos^{-1}\left(\frac{-a_1}{2r}\right) = 10.17^\circ$.

& centre freq : $f_0 = \left(\frac{10.17}{360}\right)\times(153.6\times 10^3) = 4.34$ kHz.

Conclusion :- (freq response gets shifted if exact coeff are not implemented)

Seeing change in pole locn :-

M1) Use calculator for new r, θ

M2) $x + jy = r\cos\theta + j\sin\theta$ ✓

ex.

Now, consider a higher order section :-

$H(z) = H_1(z) \cdot H_2(z) \cdot H_3(z) \cdot H_4(z)$

(error for each $H_i(z)$ will get accumulated. We are seeing that :)

Given : PB : 20.5 - 23.5 kHz

SB : 0-19 kHz, 25-50 kHz.

PB ripple : $\leq$ 0.25 dB

SB attenuation : > 45 dB

f_s = 100 kHz

We got 4 sections of 2nd order (using program)

$$H_1(z) = \frac{1+0.0339z^{-1}+z^{-2}}{1-\underbrace{0.1743}_{a_1}z^{-1}+\underbrace{0.9662}_{a_2}z^{-1}}$$

$H_2(z) = (\qquad\qquad) \rightarrow$ order 2

$H_3(z) = (\qquad\qquad) \rightarrow$ order 2

$H_4(z) = (\qquad\qquad) \rightarrow$ order 2

got using computer program. Total : 8 order sys.

Now, Determine suitable coeff. wordlength.

(a) to maintain stability

(b) to satisfy freq. response specs.

Suppose we use 8 bits (wordlength), we see response (for all 4 sections)

$$\text{For } H_1(z) = a_1 = -(0.1743 \times 2^7 + \overset{\text{for rounding off effect}}{0.5}) = -22.8104 = -22$$

$$a_2 = (0.9662 \times \underset{\text{1 bit for sign, rest 7 for fraction}}{2^7} + 0.5) = 124.1736 = 124$$

corresponding modified coeffs become :-

$a_1 = -22/128 = -0.171875$

$a_2 = 124/2^7 = 0.96875$

& $(r, \theta)_{mod}$:

$r = 0.9843$

$\theta = 84.99$.

Illy do ∀ 4 sections :

Then do $H(z) = H_1(z)\, H_2(z)\, H_3(z)\, H_4(z)$

we get $\frac{\text{num}}{\text{den}}$ (order 8).

Then, find the values of coeff after multiplicⁿ

Now, choose no. of bits to represent these coeff.

Then see what change comes in coeff.

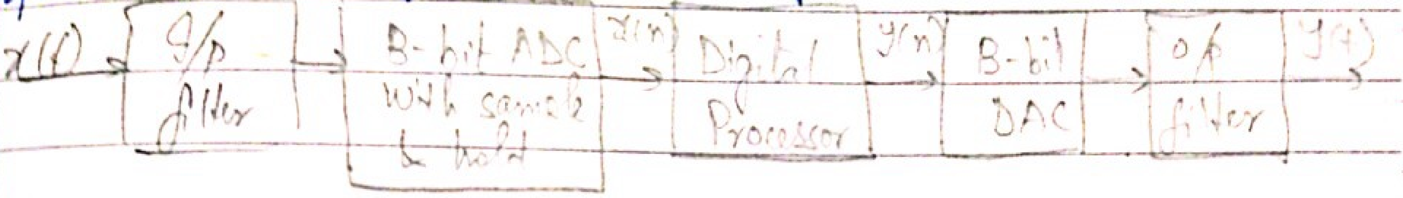
After that, see if these coeff. are giving/ satisfying the reqd specs.

(for this problem, representⁿ in 5 & 16 bits showed to satisfy given specs)

↳ seen by finding centre freq & pole locⁿˢ.

or, plot freq. response & see if specs are meeting.

* Block diagram of real time digital filter with analog i/p or o/p signals :-



(how a sys works in DSP)

This o/p. y(t) will / can act as reference i/p for any electrical sys. (motor etc.)

* DAC & ADC are introducing errors

* Word length effects :-

1) ADC noise
2) Coeff. quantiz^n.
3) Round off errors from arithmetic oper^{ns}.
4) Arithmetic overflow.

eg Consider 2 bit represent^n (binary (digital)) variation of 9.4 V.

We have 2 bits. So, sensitivity $= \frac{1}{2^2} = 0.25$

Considering V from 0 to 10 V, So, Sensitivity = 2.5×10

So, 0 - 2.5 → 00
2.5 - 5 → 01
5 - 7.5 → 10
7.5 - 10. → 11

Idea:- if I represent 0.1 V, its 00, 2.4 V → 00. If its 2.55, its 01. So, error, max is of 2.5 (for max 10 V) For 1 V, its $\boxed{0.25}$ → 1/4 → $\frac{1}{2^2}$

, So, ∀ values from 7.5 V - 10 V ⟶ 11

(errors will come.)

4 bit, suppose.

⇒ Sensitivity $= \frac{1}{2^4} = 0.06$

0 - 0.06 → 0000

0.94 - 1.00 ⟶ 1111

* So, clearly, max. variation b/w actual value & quantized one is $\frac{1}{2^B}$, for B bits

Illy, 7.5 - 8.1 → 10000, say ⟶ So, 7.5 - 8.1 has one represent^n.
So, error is reduced

- ✱ Increasing no. of bits, ⇒ cost ↑.
  So we want to reduce no. of bits (with some adjustable errors)
  If Errors are involved & we are able to get req^d response in that case also, that design of B bits is req^d.

- ✱ Suppose we have to do 10 bit represent^n & we have 8 bit 8086 μP Then, to implement, truncation is req^d.

(3) ROUNDIN OFF ERROR

eg Given decimal no → 1898
Represent it in 3 bits

Idea: We have diff^t forms of represent^n.
eg: Real no, Integer, Exponential form, double exponential forms etc.

Here, use

Exponential form = I . F E(_ _)

I → Integer part
F → Fractional part
E(_ _) → Exponential part; → sign, → no.

So, here 1898 = 1.8 E(3)

eg:
8375.348 → represent in 10 bits
st I → 2 bits
F → 3 bits
E → 5 bits.

I → 83, F → 753, E → +0002

$$= \underbrace{83}_{2}.\underbrace{753}_{3}\ E(\underbrace{+0002}_{5})$$

(4) Arithmetic Overflow.

Suppose we get x(0) that is represented in 8 bits.

Now, we do x(0) . h(2). Then, again represent-ing in ~~bu~~ 8 bits (say) might overflow. So, truncating is req^d.

FINITE WORD LENGTH EFFECTS ON FIR FILTERS

Coeff. quantiz^n :-

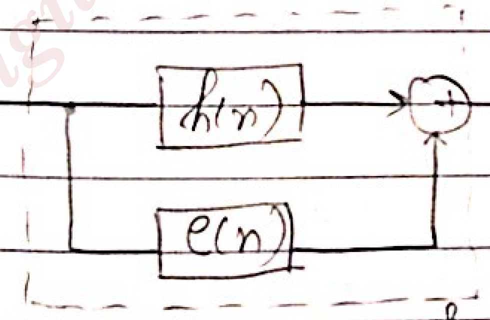
$$h_q(n) = h(n) + e(n) \; ; \; n = 0, 1, - - -, N-1$$

$$\Rightarrow H_q(\omega) = H(\omega) + E(\omega)$$

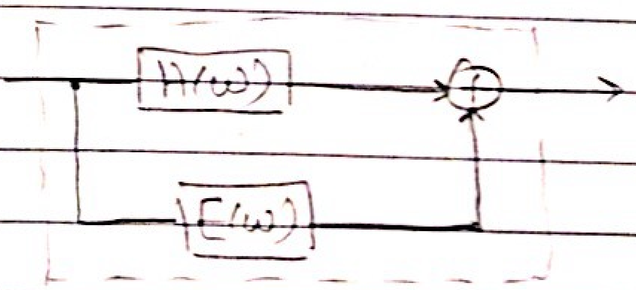
where

$$E(\omega) = \sum_{m=0}^{N-1} e(m)\, e^{-j\omega m}$$

↳ Objective is to limit E(ω)



$h_q(n)$



$H_q(\omega)$

✯ In digital represent^n of B bits, worst case error (max. error) that can occur in ~~f~~ implementing quantized coeff. is $\frac{1}{2^B} = 2^{-B}$.

For N coeff., error = $N \times 2^{-B}$.

So, $|E(\omega)| = N2^{-B}$

→ N : filter length

→ B : no. of bits.

Based on statistical data, we assume only $\frac{N}{3}$ coeff will introduce error$)_{max}$.

$$\Rightarrow |E(\omega)| = \left(\frac{N}{3}\right)^{1/2} 2^{-B}$$

Based on some other approxim^n

$$|E(\omega)| = 2^{-B}\left[\left(\frac{N\log_e N}{3}\right)\right]^{1/2}$$

✯ Coeff. quantiz^n error

eg: Determine effects of quantizing by rounding off coeff to 8 bits.

SB attenuation	> 90 dB
PB ripple	< 0.002 dB
PB edge freq	3.375 kHz
SB edge freq	5.625 KHz
Sampling freq.	20 kHz
No. of coeff	45

Idea: See how design changes when quantized coeff are used than actual

$N = 45.$

bandedge freq: $\left(\underset{\text{PB freq}}{0}, \frac{3.375}{(20/2)}, \frac{5.625}{20/2}, \frac{20}{2} \right)$

Nyquist PB edge freq

SB edge freq

Nyquist freq

weights : 1, 7.28

From table 7.15 (Ifeachor), we find symmetrical coeff.

$h(0) = h(44) = -1.05023e-04$

$h(1) = h(43) = -1.25856e-04$

Now, these values have to be converted to 8 bit representⁿ (hexadecimal, binary ... depends

Seeing the values, we see some are +ve, some −ve. So, 1 bit exclusively for sign. So, now represent with remaining 2^7 bits.

Now, the intervals are

1st bit : 0 to $\frac{1}{2^7}$ i.e. $0 - 7.8125\times10^{-3}$

$7.8125\times10^{-3} - 2(7.8125\times10^{-3})$

2nd bit : $\frac{1}{2^7}$ to $\frac{1}{2^7}+\frac{1}{2^7}$

Say the interval is

Suppose

$0 - 7.81\times10^{-3} \equiv 0 - a$ 00

$7.81\times10^{-3} - 15.6\times10^{-3}$ $a - 2a$ 01

⋮ $2a - 3a$ 10

⋮ $3a - 4a$ 11

Now,

Seeing Rounding

For the bit 00, if ∃ any value b/w

$0 - a/2$ 00

$a/2 - a \xrightarrow[\text{off}]{\text{rounded}} 01$

Illy, for bit 01, if ∃ any value b/w

$a \to a+\frac{a}{2} \longrightarrow 01$

$a+\frac{a}{2} \to 2a \xrightarrow[\text{off}]{\text{rounded}} 10$

On similar lines,

$\frac{7.81\times10^{-3}}{2} \simeq 3.9\times10^{-3}$.

Now, if any filter has value of coeff as 3.7×10^{-3} (3.7654e-03)

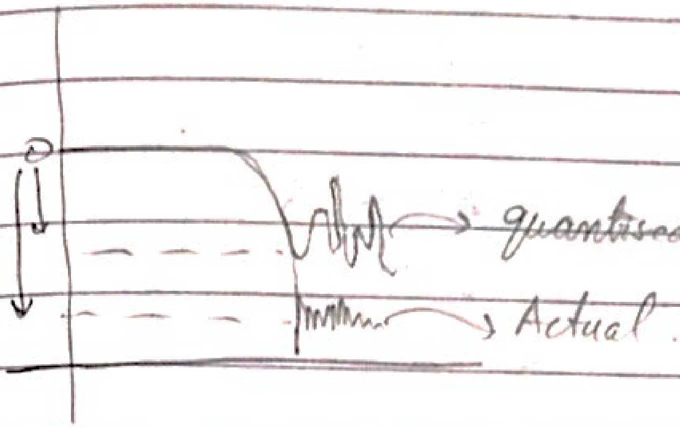
This value is b/w $0 - a/2$

hence, its bit will be $\overset{2^7}{0}000000\overset{2^1}{0}\overset{2^0}{0}$

if any coeff has value $= 4.5\times10^{-3}$, say,

its bit will be 00000001 (rounded off)

Graphically, seeing it



} attenuation got reduced with quantizⁿ.

eg. Show that max SB attenuation possible (A_{max}) for a direct form LP FIR filter with coeff. rounded by

$$\boxed{A_{max} \leq 20 \log_{10} (2^{-B} N)} \rightarrow eq^n\ 7.45$$

Given, LPF FIR filter with specs:-

PB deviation	0.05 dB
sampling freq	10 kHz
PB edge	1.8 kHz
TW	500 Hz
N	65

(a) Estimate no. of bits reqᵈ to represent each coeff. for filter to have attenuation of atleast 60 dB in SB

(b) If coeff. wordlength in (a) is used, estimate increase in PB ripple & redⁿ in SB attenuation (in dB)

(c) Compare actual SB attenuation & PB ripple of filter using coeff. wordlength in(a)

(a) For 60 dB attenuation,
no. of bits can be got as

$$A_{max} \leq 20 \log_{10}(2^{-B} N)$$

$\hookrightarrow B \simeq 15.982$ bits $\simeq 16$ bits

With no. of bits got, analysis of freq response will give (b) & (c)

(b) After quantizⁿ, let the worst case peak ripple in passband, R_{max} & SB attenuation A_{max}.
So, change in PB:

$$R_{max} = 20\log(1 + \delta_p + \underbrace{|E(\omega)|}_{\text{error} = N2^{-B}})$$

$$\Rightarrow R_{max} = 20\log(1 + \underbrace{\boxed{\delta_p}}_{\text{got as } 20\log(1+\delta_p) = 0.05 \text{ given}} + (65 \times 2^{-16}))$$

$$\Rightarrow R_{max} = 20\log(1 + 0.005773 + 0.001)$$

$$R_{max} = 0.0586 \text{ dB}$$

(So, by quantizⁿ, 0.05 → 0.0586)

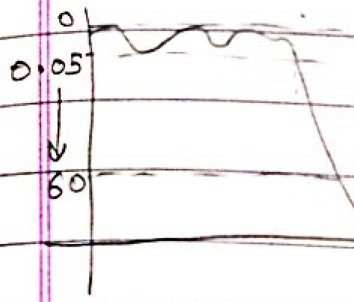
Now,
Change in SB:

$$A_{max} = -20\log(\overbrace{\delta_s}^{\text{got as } 20\log(\delta_s) = 60 \text{ given}} + \underbrace{|E(\omega)|}_{N2^{-B}}) = -20\log(0.001 + 0.001) = 54 \text{ dB}$$

(due to quantizⁿ, 60dB → 54dB)

So, basically, if

I had : $\xrightarrow{\text{on quantiz}^n}$ I got :



(PB ripple increases
SB attenuation decreases)

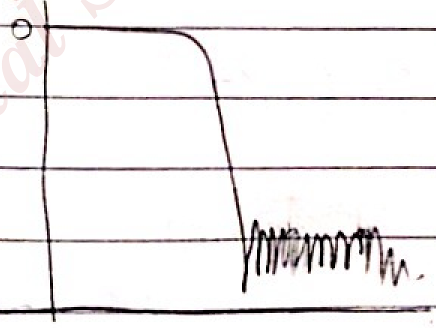
(c) Frequency response got by (Optimal filter design)

N = 65

$$\text{edge freq} = \left(\frac{0}{f_s/2}, \frac{1.8}{f_s}, \frac{1.8+\overset{\to TW}{\boxed{0.5}}}{f_s}, \frac{f_s/2}{f_s}\right)$$

$$= (0, 0.18, 0.23, 0.5)$$

weights: 1, 5.733



: Graph that we get

(*) Round off error :

represent^n using double length registers and rounding at the final sum after y(n)

overflows overcome by normalising

→ seen in adder circuits when sum of values exceeds the size of register

So, value never exceeds 1 ← how to prevent overflow? → scale down (normalise) → this decreases value.

Then, denormalise it at o/p. ∴

Ⓜ1 Suppose sum(coeff.) $a_1 + a_2 + a_3 = x$.

So, normalise all values, $\frac{a_1}{x}, \frac{a_2}{x}, \frac{a_3}{x}$

$$h(m) = \frac{h(m)}{\sum_{k=0}^{N-1} |h(k)|}$$

Ⓜ2 Normalising using RMS values.

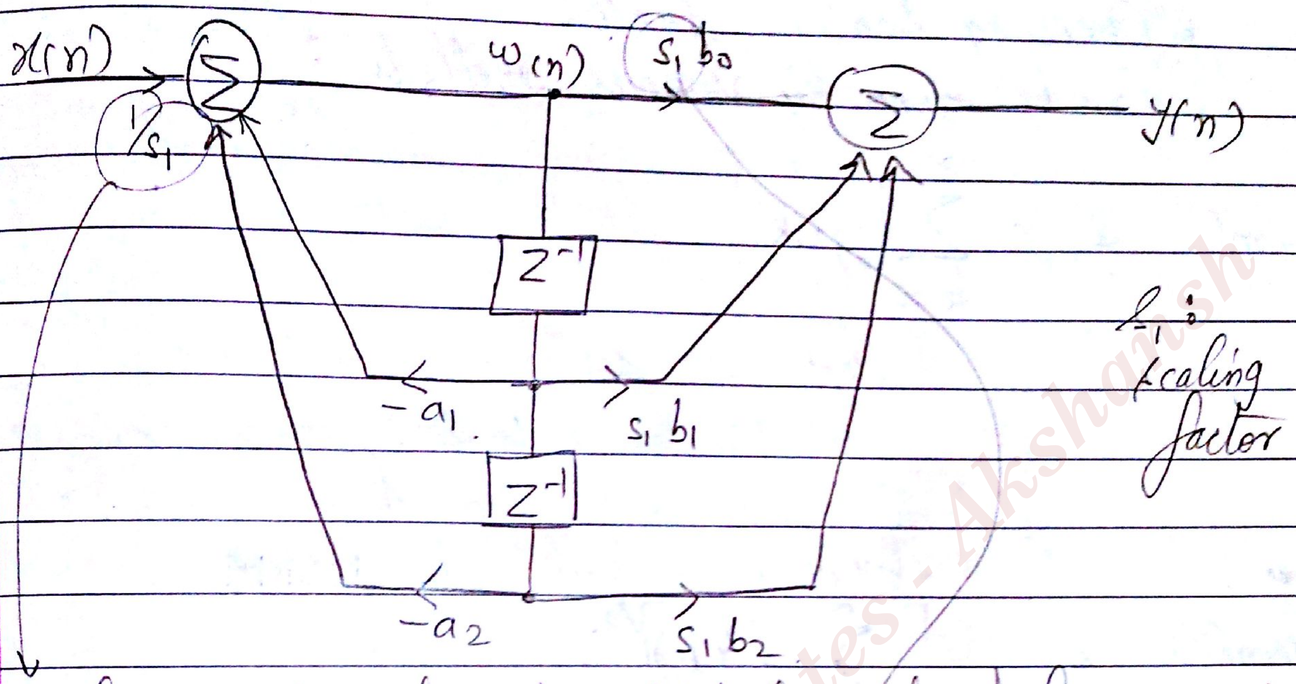
$$h(m) = \frac{h(m)}{\left[\sum_{k=0}^{N-1} h^2(k)\right]^{1/2}}$$

- Better signal to noise ratio (SNR) is got when this done.
- Idea: Normalise using powers of 2.
  (∵ ÷2 is like shifting right by one bit.
  So, calculation becomes easy.)

* Seeing Scaling in block diagram

Canonic form of implementⁿ.

Direct form 2

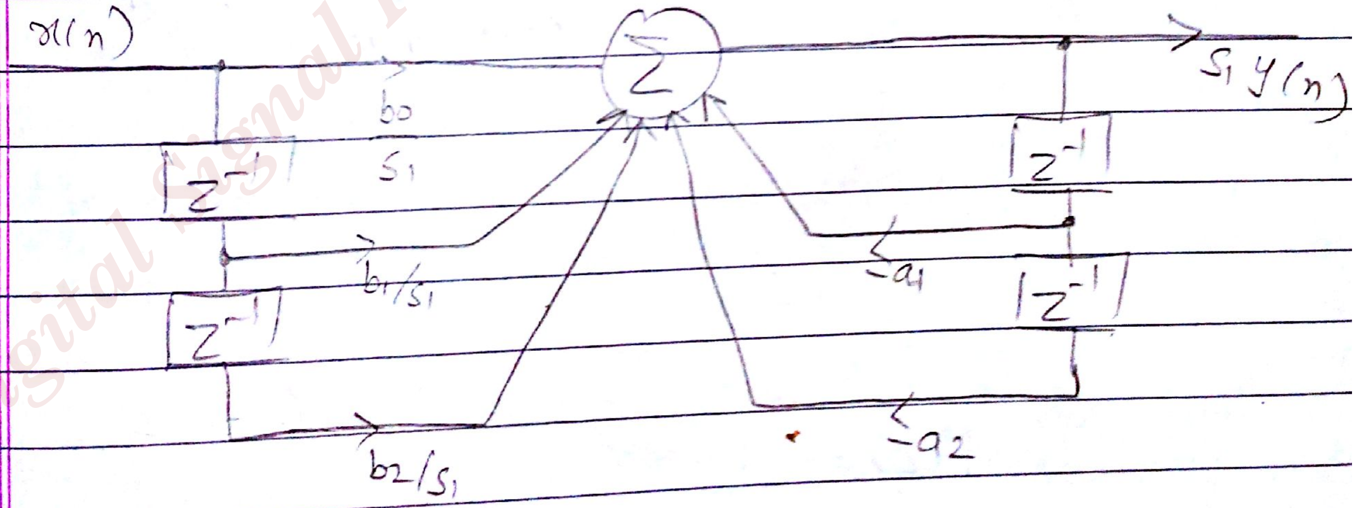


s_1: scaling factor

Scaling x(n) values by $(1/s_1)$ & to balance out, numerator of TF is $\times (s_1)$

$$\frac{Y(Z)}{X(Z)} = \frac{N(Z)}{D(Z)} \Rightarrow \quad Y(Z) = \frac{(S) \times N(Z)}{D(Z)} \times X(Z)$$

Direct form 1



* Choosing Scaling factor

Can be done by various methods :

Discrete impulse response coeff

Form ① $s_1 = \sum_{k=0}^{\infty} |f(k)|$

- Includes ∞ series summation.
- Computation of infinite series can be done by evaluating freq. response.
- This factor is called L_1 NORM

Form ② $s_1 = \left[\sum_{k=0}^{\infty} f^2(k)\right]^{1/2}$

- This scaling factor is called as L_2 NORM.
- finding L_2 norm :

$$\boxed{\sum_{k=0}^{\infty} f^2(k) = \frac{1}{2\pi j}\oint F(z)F(z^{-1})\frac{dz}{z}}$$

- Closed CONTOUR integral (i.e, evaluating / considering values lying inside unit circle only)

* Consider a 2nd order section, with

$$F(z) = \frac{1}{1+a_1z^{-1}+a_2z^{-2}}$$

So, $F(z^{-1}) = \frac{1}{1+a_1z+a_2z^2}$

So, $s_1^2 = \sum_{k=0}^{\infty} f^2(k) = \frac{1}{2\pi j}\oint\left(\frac{1}{1+a_1z^{-1}+a_2z^{-2}}\right)\left(\frac{1}{1+a_1z+a_2z^{2}}\right)$

$$s_1^2 = \frac{1}{1-a_2^2-a_1^2(1-a_2)/(1+a_2)}$$

Date 19/11/13
Page

$$\Rightarrow \boxed{s_1^2 = \frac{1}{1-a_2^2 - a_1^2\frac{(1-a_2)}{(1+a_2)}}} \quad ✫$$

↳ If $F(z) = \dfrac{1}{1+a_1z^{-1}+a_2z^{-2}}$

Form ③ $s_1 = \max|F(\omega)|$

- → $F(\omega)$: peak amplitude of freq. respo
  b/w i/p & ω(n)
- → called as L_∞ NORM
- → ensures no overflow for SINUSOIDAL INPUT

Practically, on evaluation, we get :

$$L_2 < L_\infty < L_1$$

Q. Determine a suitable scale factor to prevent or redu the possibility of overflow in an IIR lowpass filt characterised by following TF :-

$$H(z) = \frac{1+2z^{-1}+z^{-2}}{1-1.0581359z^{-1}+0.338541z^{-2}}$$

* Given TF, L_1 Norm, L_2 norm & L_∞ norm can be computed using software (program)

Found values :-

	L_1	L_2	L_3
S_1	3.7112	$\boxed{1.7352}$	3.5663

↳ PTO

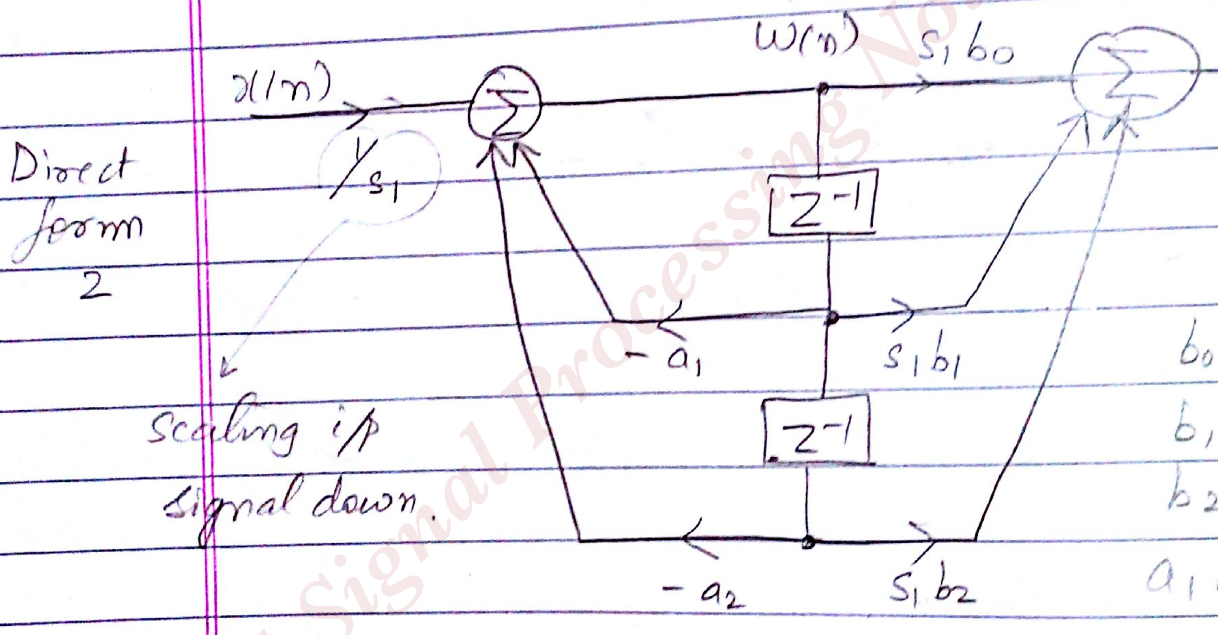
L_2 norm is found as :-

$$S_1^2 = \frac{1}{1 - a_2^2 - a_1^2 \frac{(1-a_2)}{1+a_2}}$$

$$\Rightarrow S_1^2 = \frac{1}{1 - (0.3385)^2 - (1.058)^2 \left(\frac{1 - 0.3385}{1 + 0.3385}\right)}$$

$$\Rightarrow S_1 = 1.7350.$$

Structure :-



Coeff :-

$b_0 = 1$

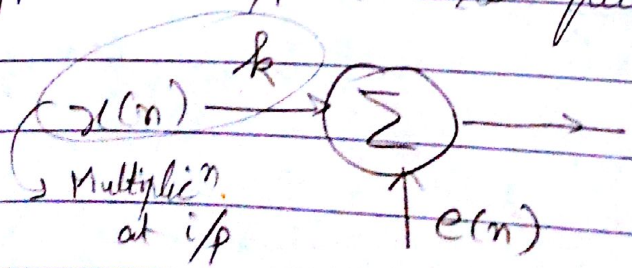
$b_1 = 2$

$b_2 = 1$

$a_1 = -1.0581359$

$a_2 = 0.338544$

Effect on o/p due to quantizⁿ error



(Scaling) Multiplicⁿ seen

$a_k * y(n-k)$: at feedback part

$b_k \; x(n-k)$: at feedforward part

wherever ∃ multiplicⁿ of say

$x(n) \times k$

or $a_k \times y(n-k)$

or $\underbrace{b_k}_{B \text{ bits}} \cdot \underbrace{x(n-k)}_{B \text{ bits}} = 2B \text{ bits}.$

So, I need 2B bit storage to store result.
But, I want to truncate/round off/scale appropriately/quantize it to fit in B bits
Hence, ∃ an error.

Let $y(n) = k.x(n) + e(n) \rightarrow \text{error}$

Noise power ; $\sigma_e^2 = \dfrac{q^2}{12}$

rounding

→ q : quantizⁿ error

→ $q = \dfrac{1}{2^{B-1}}$; B : no. of bits

In rounding off, max. of 1/2 bit error can come So, $q = \dfrac{1}{2^B/2} = \dfrac{1}{2^{B-1}}$

i.e., if rounding off is the only error considered:- $q = 2^{-B+1}$

Note:- If both sign bit & round off is taken, $q = \dfrac{1}{2^{B-2}}$

If truncation, $q = \dfrac{1}{2^B}$

So, for rounding off,

$$\sigma_e^2 = \frac{\left[2^{-(B-1)}\right]^2}{12} = \frac{2^{-2B}}{3}.$$

Total noise power = ADC noise power + Round of
noise po

$$\Rightarrow \boxed{\sigma_0^2 = \sigma_A^2 + \sigma_\varepsilon^2}$$

Let i/p signal variance $= \sigma_x^2$.

↳ Signal to noise ratio (SNR) due to roun

$$(SNR)_{\text{rounding}} = \frac{\sigma_x^2}{q^2/12} = 12 \times 2^{2B} \cdot \sigma_x^2$$

↳ in dB $= 20 \log \left(\frac{\sigma_x^2}{\sigma_e^2} \right)$ → error due to rounding $= \sigma_\varepsilon^2$.

∴ its power, basically a square term

$$= 10 \log \left(\frac{\sigma_x^2}{q^2/12} \right)$$

$$= 10 \log \left(12 \times 2^{2B} \sigma_x^2 \right)$$

$$= 6.02B + 10.79 + 10 \log \sigma_x^2$$

→ variable → const

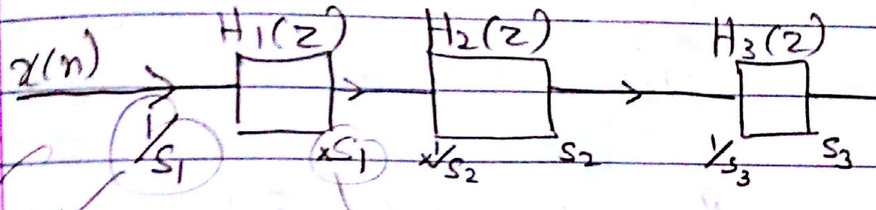
⇓

6 dB increase is seen in SNR with incr of every bit.

=> series

* Scaling factors for $\mathrm{cascade}$ & parallel structures:

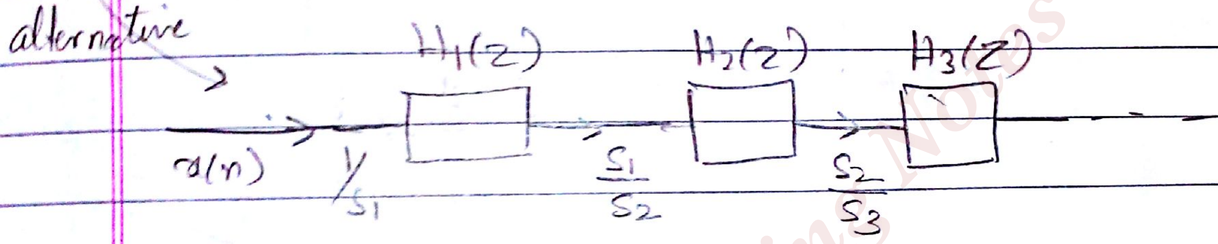
CASCADE STRUCTURE



applying scaling

multiplying by s_1, ∴ we can get original signal

Given: Overall TF, $H(z) = H_1(z).H_2(z) ---- H_n(z)$



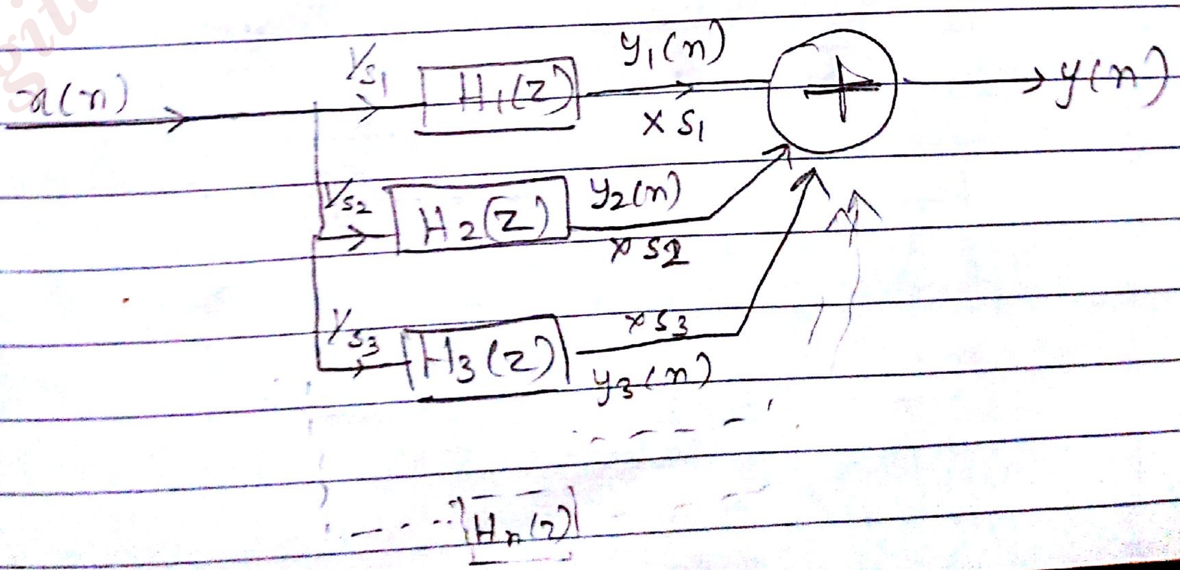
Without multiplier circuit, do calcul^{ns}.

Note: We need to consider only $1/S_1$. Rest all is internal

PARALLEL STRUCTURE

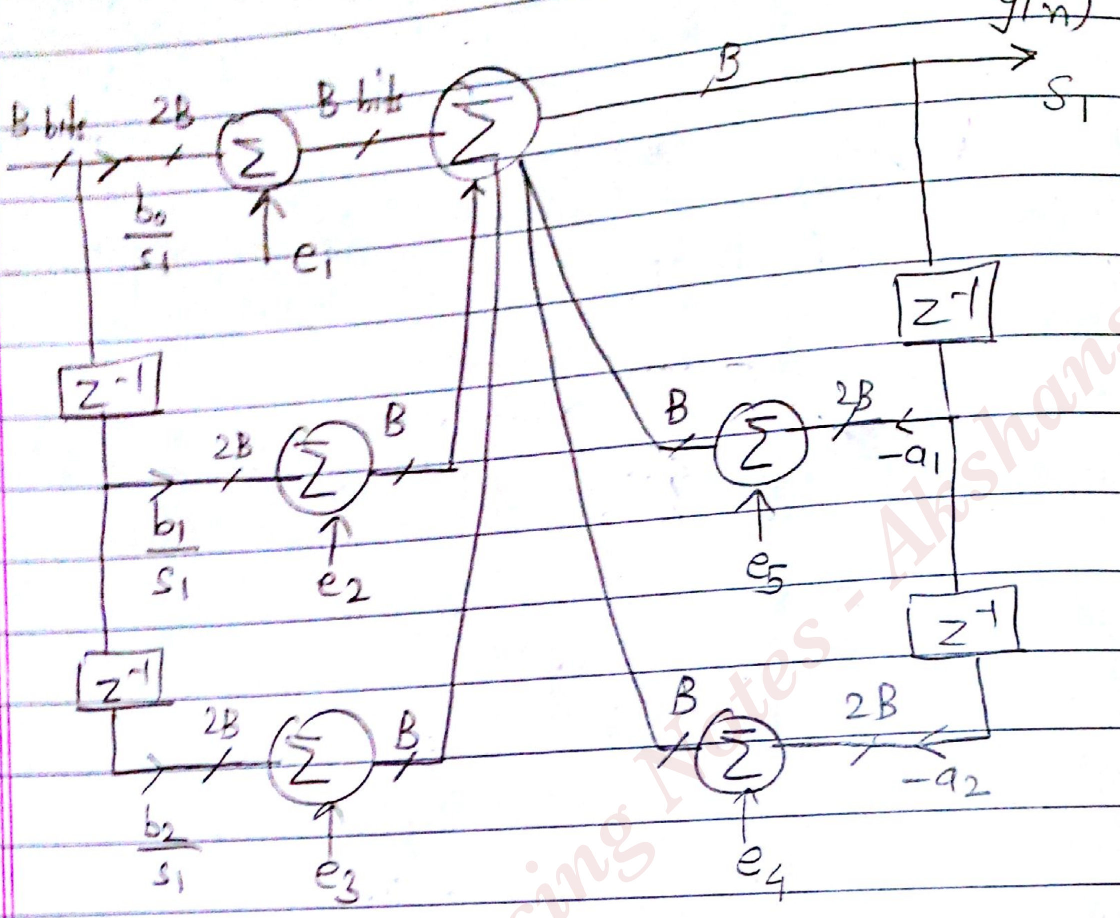
Overall TF, $H(z) = H_1(z) + H_2(z) + - - -$

Structure:-

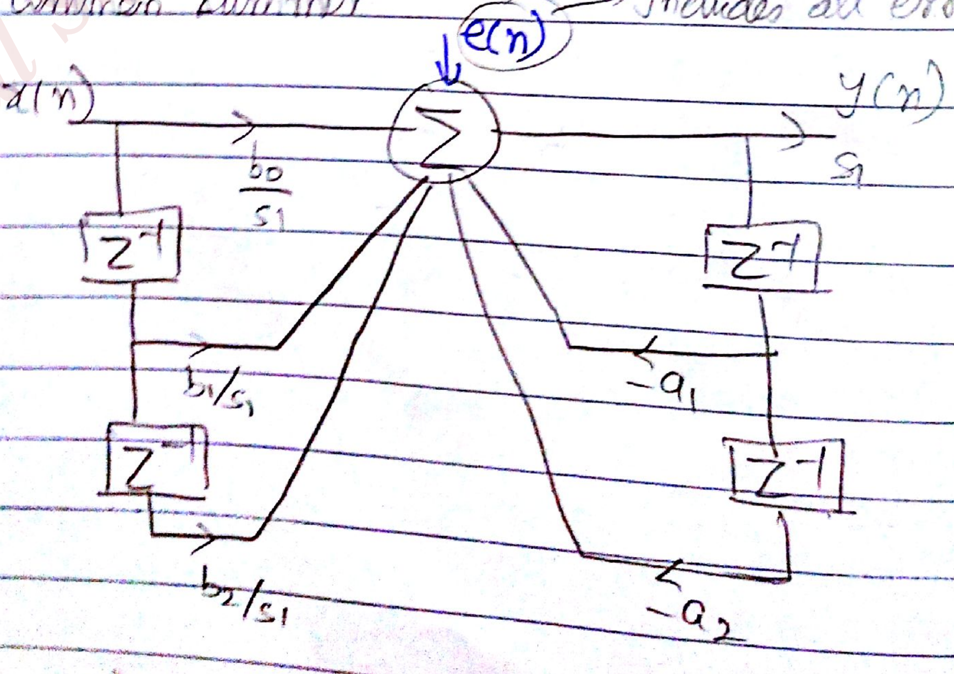


• Product roundoff errors in IIR filter

Direct form 'I



2B bits are got on multiplic^{n}.
Now, each of them is converted into B bit so, ∃ errors in each segment. This can be also made by taking 5 times error at the common summer → Includes all errors : e_1, e_2, e_3, e_4, e_5



Finding error :—

Idea : Find TF b/w $e(n)$ & $y(n)$. Take other i/ps = 0. So, only f/b is left

$$\sigma_{or}^2 = 5\,\frac{q^2}{12}\left[\frac{1}{2\pi j}\oint_c F(z)\,F(z^{-1})\,\frac{dz}{z}\right]\boxed{s_1^2}$$

We found o/p is being $\times s_1$. So, corresponding power : $\times s_1^2$

$$= 5\,\frac{q^2}{12}\left[\sum_{k=0}^{\infty} f^2(k)\right]s_1^2$$

σ_{or}^2 Total noise quantizn error at o/p (roundoff noise)

$$= \frac{5q^2}{12}\,\|F(z)\|_{\boxed{2}}^{\boxed{2}}\; s_1^2$$

square

$\Rightarrow L_2$ norm

$\hookrightarrow F(z) = \dfrac{1}{1 + a_1 z^{-1} + a_2 z^{-2}}$; 2nd ord TF

$\hookrightarrow f(k) = Z^{-1}\left[F(z)\right]$

Now, we know

$$\sigma_o^2 = \boxed{\sigma_{oA}^2} + \sigma_{or}^2$$

ADC error

$$= \frac{q^2}{12}\left[\sum_{k=0}^{\infty} h^2(k) + 5s_1^2 \sum_{k=0}^{\infty} f^2(k)\right]$$

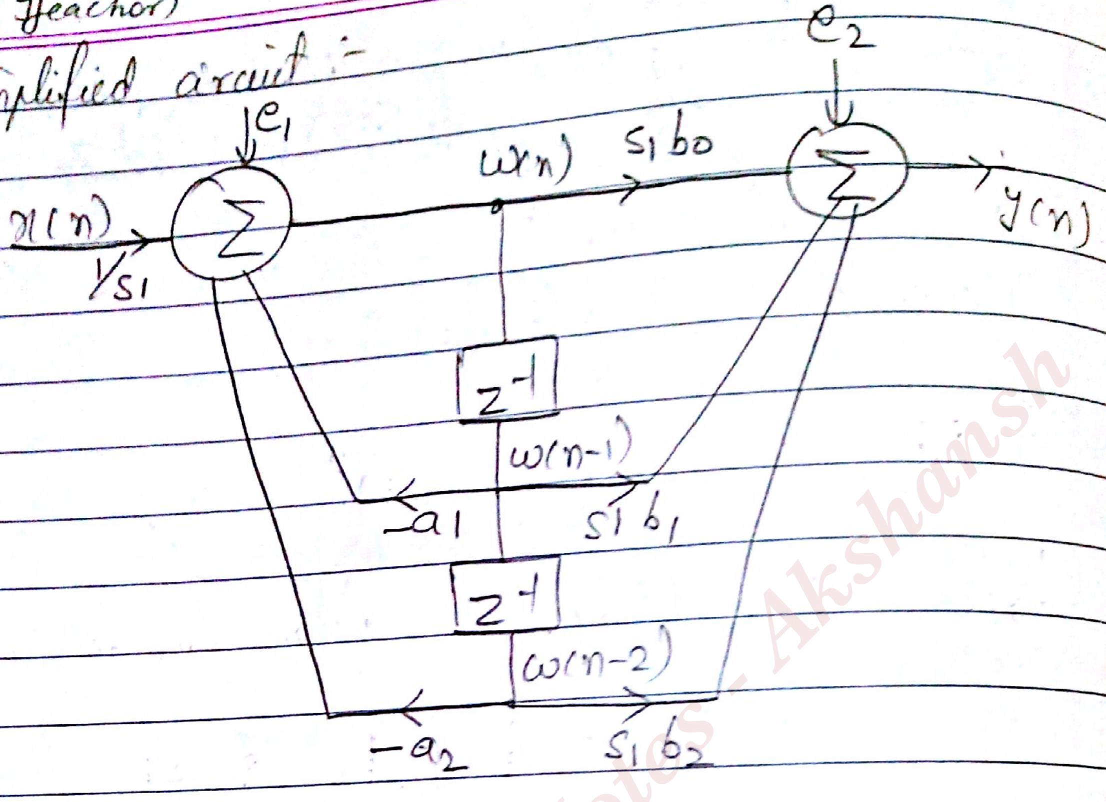
Direct form I o/p noise power :

$$\boxed{\sigma_o^2} = \frac{q^2}{12}\left[\|H(z)\|_2^2 + 5s_1^2\,\|F(z)\|_2^2\right]$$

Pg - 839 (Ifeachor)

Simplified circuit :-

Direct form II.



Quantiz^n errors have been distributed on the above simplified model having 2 adder circuits

So,

$\exists$ 2 TFs :-

① b/w e_2 & $y(n)$

② b/w e_1 & $y(n)$

↳ assuming all other signals=0

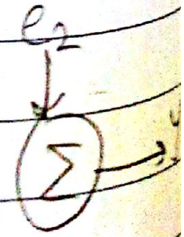
- TF b/w e_1 & $y(n)$:

  f/b & feedforward f^n remains the same

  So, its $H(z)$

- TF b/w e_2 & $y(n)$:-

  All inputs = 0. So, we have →



So, TF = 1

$$\sigma_{0e}^2 = ③ \frac{q^2}{12} \sum_{k=0}^{\infty} f^2(k) + ③ \frac{q^2}{12}(1)$$

∴ 3 multiplicⁿ & truncⁿ error for e_1 & 3 for e_2

for e_1, TF is same

for e_2, TF = 1

$$\Rightarrow \sigma_{0e}^2 = 3\frac{q^2}{12}\left(\|F(z)\|_2^2 + 1\right)$$

$f(k)$ is impulse response from e_1 to y

&

$$F(z) = s_1 \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} = s_1 H(z)$$

· Total noise = ADC + roundoff noise,

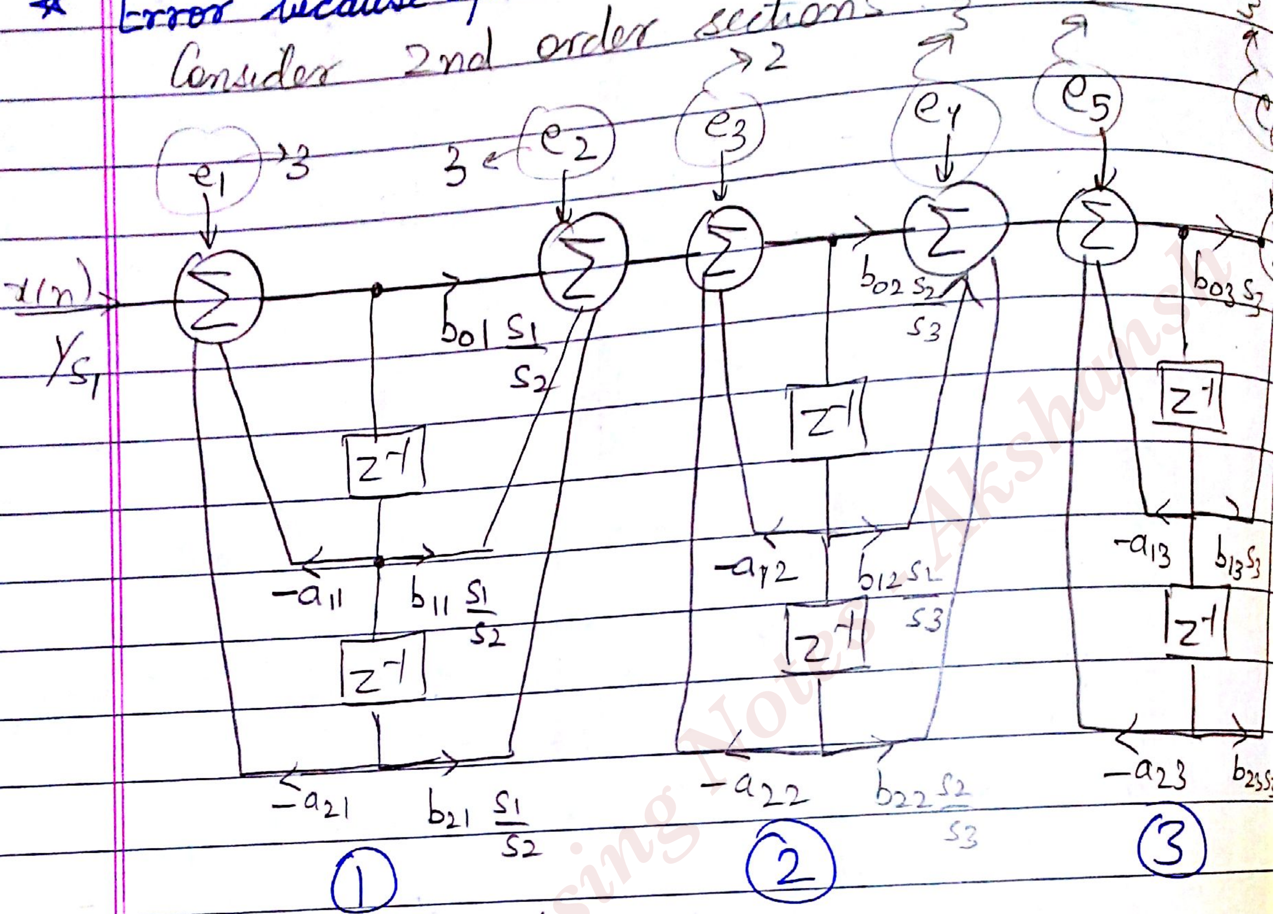
$$\sigma_0^2 = \sigma_{0e}^2 + \sigma_{0A}^2$$

$$= \frac{q^2}{12}\left[3\left[1 + s_1^2 \sum_{k=0}^{\infty} h^2(k)\right] + \sum_{k=0}^{\infty} h^2(k)\right]$$

& so, $$\sigma_0^2 = \frac{q^2}{12}\left[3\left(1 + s_1^2 \|H(z)\|_2^2\right) + \|H(z)\|_2^2\right]$$

★ Error because of cascade :-

Consider 2nd order sections



e_1 has 3 quantizn errors

$e_2 \rightarrow 3$

$e_3 \rightarrow 2$

$e_4 \rightarrow 3$

$e_5 \rightarrow 2$

$e_6 \rightarrow 3$

Total o/p noise due to roundoff error :-

$$\sigma_q^2 = 3\frac{q^2}{12}\sum_{k=0}^{\infty} f_1^2(k) \quad : e_1$$

$$+ \; 3\frac{q^2}{12}\sum_{k=0}^{\infty} f_2^2(k) \quad : e_2$$

$$+ \; 2\frac{q^2}{12}\sum_{k=0}^{\infty} f_3^2(k) \quad : e_3$$

$$+ \; 3\frac{q^2}{12}\sum_{k=0}^{\infty} f_4^2(k) \quad e_4$$

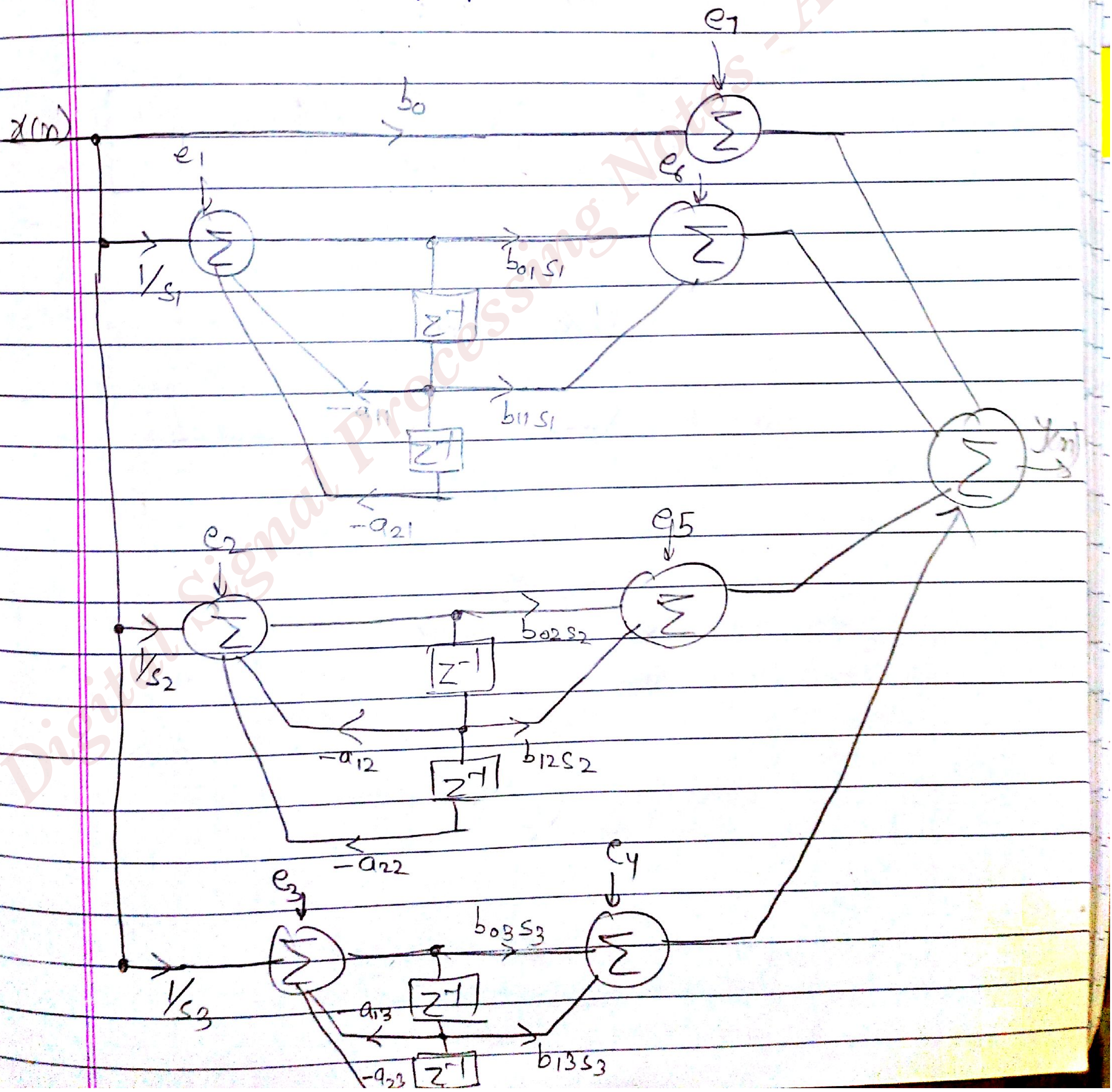
$$+ \; 2\frac{q^2}{12}\sum_{k=0}^{\infty} f_5^2(k) \quad : e_5$$

$$+ \; 3 \; q^2/ \quad : e_6$$

$$\Rightarrow \sigma_{0\epsilon}^2 = \frac{q^2}{12}\left[3\sum_{k=0}^{\infty} f_1^2(k) + 5\sum_{k=0}^{\infty} f_3^2(k) + 5\sum_{k=0}^{\infty} f_5^2(k) + 3\right]$$

$$\Rightarrow \sigma_{0\epsilon}^2 = \frac{q^2}{12}\left[3\|F_1(z)\|_2^2 + 5\|F_3(z)\|_2^2 + 5\|F_5(z)\| + 3\right]$$

* error because of parallel



$$\sigma_{e,i}^2 = \frac{3q^2}{12} \sum_{k=0}^{\infty} f_i^2(k) = \frac{3q^2}{12} \| F_1(z)\|_2^2 \quad ; i=1,2,3$$

$$= 3\frac{q^2}{12} s_i^2 \sum_{k=0}^{\infty} h_i^2(k) = \frac{3q^2}{12} s_i^2 \|H_i(z)\|_2^2$$

$\hookrightarrow i = 1, 2, 3$

$$\sigma_{oh}^2 = \frac{q^2}{12}\left\{7 + 3\sum_{i=1}^{3}\left[s_i^2 \sum_{k=0}^{\infty} h_i^2(k)\right]\right\}$$

$$= \frac{q^2}{12}\left[7 + 3\sum_{i=1}^{3} s_i^2 \|H_i(z)\|_2^2\right]$$

$\hookrightarrow H(z) = H_0(z) + H_1(z) + H_2(z) + H_3($

TMS : Texas Instrument Processors
320 : series for DSP,
C : CMOS Tech

✩ Designing filters :-

Texas Instruments

ARCHITECTURE OF C5X series processors:

TMS 320....

- ✩ Fixed point : Characteristic & mantessa has fixed no. of bits
- ✩ Floating point : has exponential terms. So, much more accurate

- ✩ C1x, C2x, C5x – 16 bit fixed point processors
- ✩ C3x, C4x : 32 bit floating point processor
- ✩ C6x : VLIW architecture – 1600 MIPS
  ↳ || processing ↳ very large Instruction Word processors
- ✩ C8x : Multiple ADSPs and a RISC master processsor
  ↳ Advanced DSP,

Applicns :-

- C1x, C2x, C5x : toys, hard disk drives, modems, cellular phones
- C3x : filters, analysis, hi-fi sys, voice mail, imaging, bar-code readers, motor control, 3D graphics, scientific processing
- C4x : parallel processing, image recognition etc.
- C6x : Wireless base station, communicn applicns, multichannel communicn.
- C8x : Video telephony, 3D comp. graphics etc.

C : CMOS tech.

E : EPROM (no chip non ~~volt~~ volatile memory) — If E is written instead of C.

TMS 320 5x ⇒ NMOS Tech.
↳ no alphabet

- C50, 51, 5x : have same instruction sets. but diff in capacity of on chip ROM & RAM

- C5x has 4 buses (has program & data memory separately)

1) Program Bus (PB)
2) Program Address Bus (PAB)
3) Data Bus (DB)
4) Data Address Bus (DAB)

Char. of some TMS320 family DSP Chips

	'C15	'C25	'C30	'C50	'C5
Cycle time	200	100	60	50	25
On chip RAM	4K	4K	4K	2K	5k
Total memory	4K	128K	16M	128K	128k
Parallel ports	8	16	16M	64K	64k

Total memory → RAM+ROM
Parallel ports → no. of addresses used for i/o port

Self : Basics of addressing :
✓ memory mapped i/o.
✓ i/o mapped i/o.

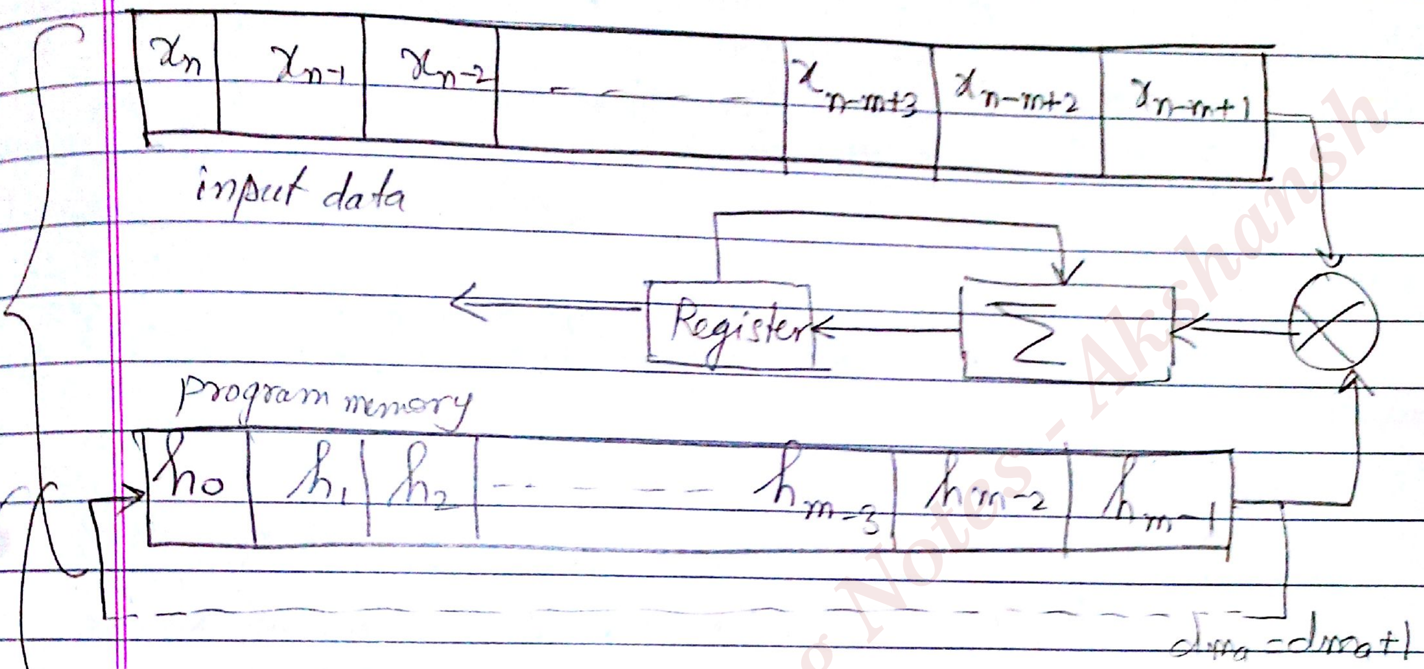
* Programmable DSPs.

1. MAC
   can be → hardware / software
   ↳ has Motorola DSP 5600(x) series — hardware singl

TI 3205X : multiplier o/p stored into product regi
ACC is in ALU (CPU)

✓ instructions executed in single clock pulse.

data memory



Multiplicⁿ operation & then, addition with prev. result (executed in 1 clk)

Implementation of convolver with single multiplier/adder.

$y_n = x_n \times h_n$ (vector mult.)

2. **MACD**: Multiply, Acc & Data Shift.

Bus structures & Mem. Access Schemes.

The 4 memory access/clock for MACD in count proc.

1) Fetch instruction from program mem.
2) " one operand from " "
3) " second " " data mem.
4) Write content of dma to address dma+1.

Bus : Group of conducto
of data takes place.

Q Advancement in architecture

1) Von Neumann Architecture
Single address bus & single data bus (4 clk for MAC)

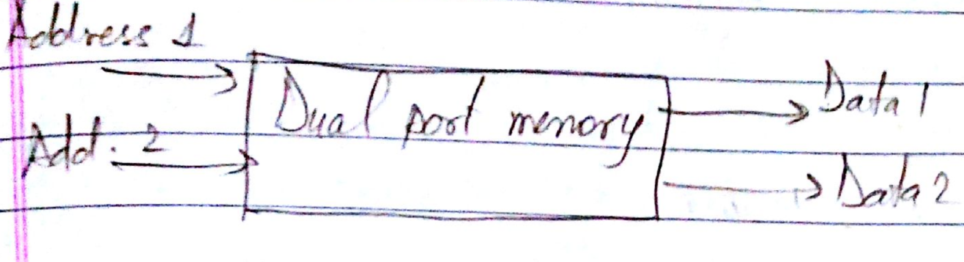
2) Harvard architecture
→ has separate buses for program & data m
So, data can be fetched simultaneously
Speed ↑.
→ consists of MAC units, multipliers, ALU, shifters

3) Modified Harvard architecture
- One set of buses for program & data
- Another set exclusively for data
- Texas Instruments use this

Q Why dont we keep shifting to advanced archite
✓ Cost of IC is high. → on no. of pins.
how to cater cost?
multiple buses used for internal data transfer (on chip). & , off chip data transfer using single bus.

✯ Multiple Access Memory :
no. of memory access/clock – high speed mem
eg: DARAM
2 DARAMs used as program & data memory will allow 4 data access/cl

☆ Multiported Memory : has 2 diff^t data & memory bus.



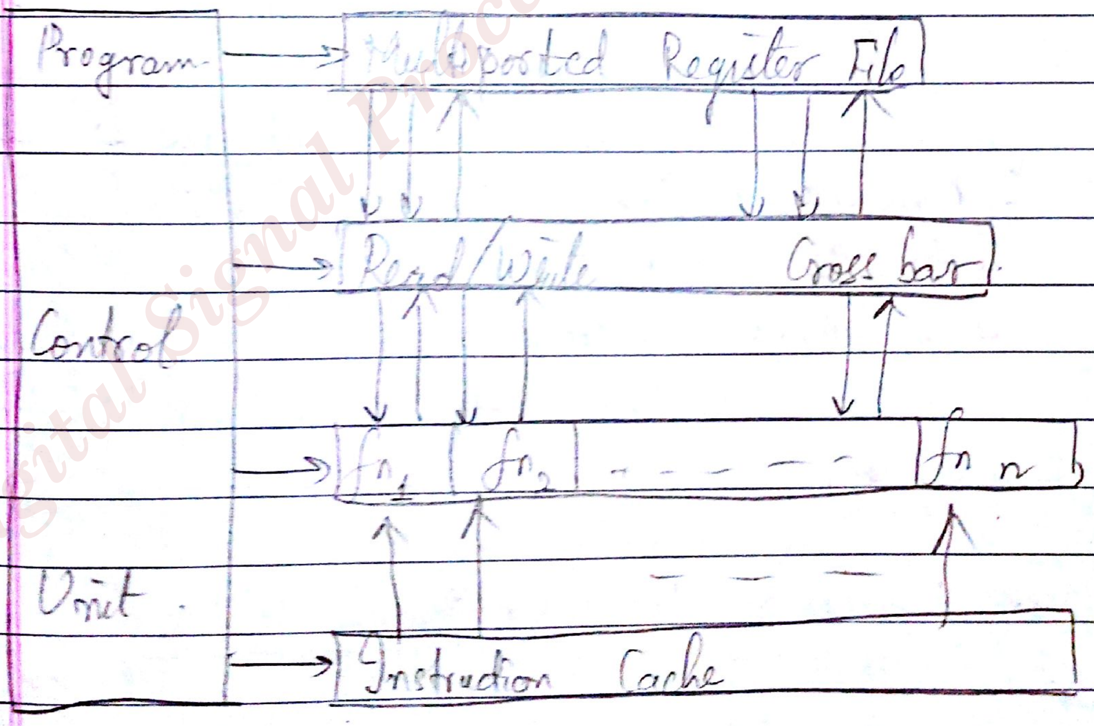
✓ It's costly

eg Motorola 56 series has single ported Program memory & Dual ported data memory.

—o—

☆ VLIW : Very Large Instruction Word Architecture.

eg :- TMS 320 C 6 X



★ Now, we have to see diff. ways of addressing
↳ seen through mapping

eg: Suppose total data of 2k. size (2K memory)
So, 11 bits reqd to address it.
Suppose we take 4 ICs of 512 size.

⇒ 0 - 511	IC_1
512 - 1023	IC_2
1024 - 15..	IC_3
15... - 2023	IC_4

So, in hexadecimal, addresses vary from
$000_H - 7FF_H$

Starting & ending addresses of

IC_1	000 - 1FF
IC_2	1FF - 3FF
IC_3	3FF - 5FF
IC_4	5FF - 7FF

00 00 0_{11} 0_4
00 01 F F

$$\begin{array}{r} = 000 \\ 1FF \\ \hline + 200 \\ \hline 3FF \\ + 200 \\ \hline 5FF \end{array}$$

Suppose bits are

00 → IC_1
01 → IC_2
10 → IC_3
11 → IC_4

$\underbrace{A_0\ A_1\ A_2\ A_3\ A_4\ A_5\ A_6\ A_7\ A_8}_{\text{Store Address.}}$ $\underbrace{A_9\ A}_{}$

Choose
IC
IC_1 -

- memory mapped addressing

★ Choosing i/o or memory using PIN ; { masking addressing
$\underset{1}{I/O} / \underset{0}{\overline{M}}$;

• Special addressing modes in P-DSP.

1) Short immediate addressing

↳ Length of instruction data depends on processor.

↳ feeding data, alongwith instruction

eg: TI TMS 320C5X: allows an 8 bit data to be specified as one of the operand alongwith single word instruction for add: sub, AND, OR, XOR etc.

2) Short direct addressing:

↳ only current page memory can be used.
lower order n bits can be specified along with the instruction.
higher order address will be loaded in
* DPP (Data page pointer)
TMS 320 → n = 7
Motorola 5600 → n = 6.

3) Memory mapped addressing:-

↳ i/o & memory are continiously addressed
⇒ unique address for i/o & memory.

eg: TMS 320C 5X : Page 0 : CPU registers & i/o registers
1 : 128 words

data is stored as pages.

Motorola 5600X :- last page : 64 loc^n is memory mapped for register & i/o.

4) Indirect addressing :-

↳ done through registers called Auxiliary registers (AR : AR0 - AR7)

Many options : TI offset registers - INDX registers

Analog : modifier registers are there.

- ✓ The pointer to AR will be of 3 bits
- ✓ These registers can be used with/without auto increment /decrement
- ✓ These registers are used to fetch the ADDRESSES of the memory, which has some data. Using AR with auto inc/dec will keep fetching data stored in memory
- ✓ Incrementing /decrementing can be done before/after execution of instruction ≡ POST/PRE PROCESSING.

5) BIT-Reversal Addressing Mode :-

- Bit reversal technique is used in Fast Fourier Transform (FFT).

6) Circular Addressing

- Cyclic f^{ns} req^{d} to implement this type of addressing
  ↳ registers .

★ On Chip Peripherals :

1. On Chip timer.
2. Serial port : with i/p & o/p buffers
   Communic^n with A/D (external), ---
3. TDM serial port.
4. Parallel port
5. Bit i/o ports – single bit wide.
6. Host port – 8 bit or 16 bit wide
7. Communic^n ports (inter processor communic^n bit P-DSP$_s$)
8. On chip A/D – D/A converters.

★ Instruction Cycle :-

PIPE LINING :

- ① Fetch from Program memory
- ② Decode
- ③ Memory read from Data Memory
- ④ Execution

→ Every instruction goes through 4 T-states as written above.

4 T-states ⇓ 4 clk pulses req^d.

Value of T	Fetch	Decode	Read	Execute
1	I1			
2		I1		
3			I1	
4				I1

Pipe line depth = 4

In these T-states, processor is idle

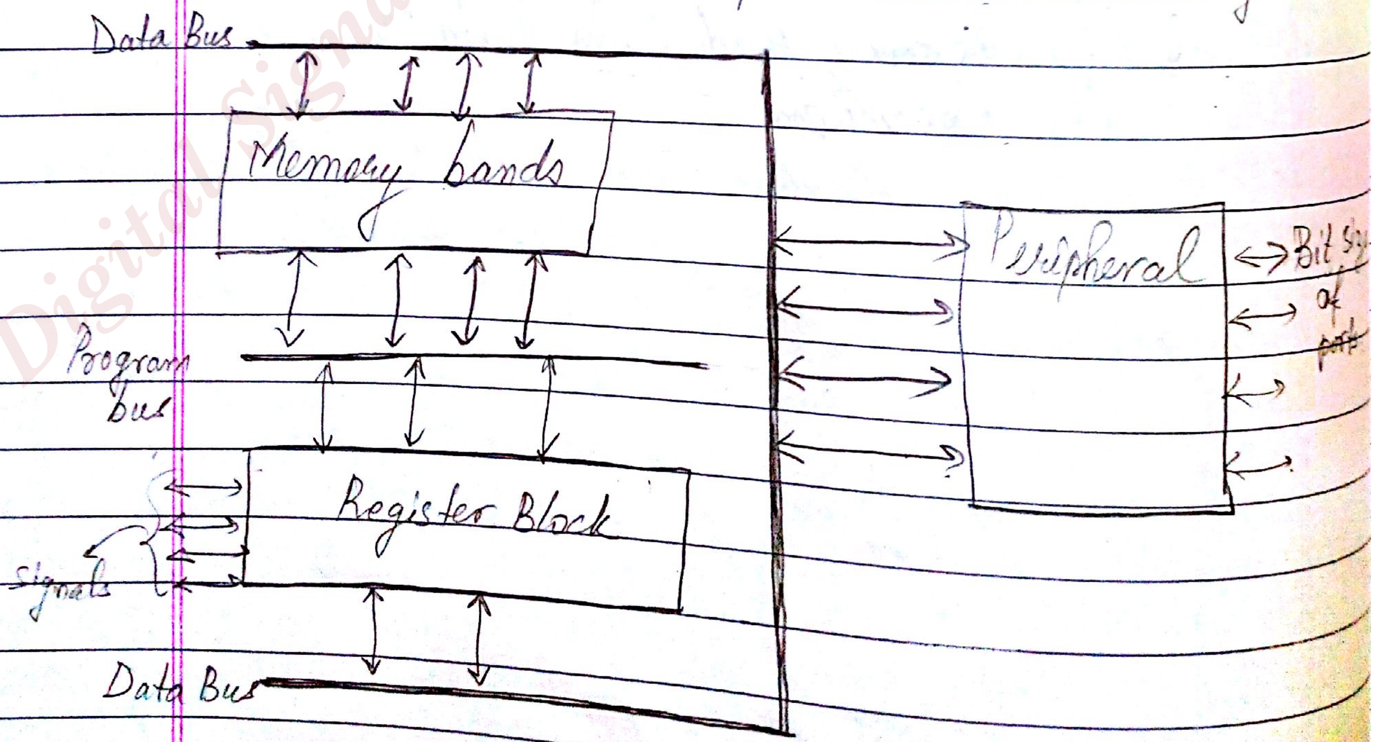
PLU: Parallel Logic Unit.

Now, we do pipelining to prevent processor being idle :-

Value of T	Fetch	Decode	Read	Exe
1	I1			
2	I2	I1		
3	I3	I2	I1	
4	I_4	I3	I2	I1
5		I_4	I3	I2
			I_4	

- ✫ Total time in execution of program = $(N+4)T_c$
- ✫ How many instructions have been queued up → tells depth of Pipelining
  (depth = diff^t for Analog, TMS, Motorol

✫ Internal Architecture of C5X (detailed diagram 'se



Register block contains

CALU :- It has

- ✓ 16 × 16 bit parallel multiplier
- ✓ ALU, ACC, ACCB, PREG (each 32 bit)
  - ↳ Product Register
- ✓ 0-16 bit barrel shifter (left & right)
  - ↳ Max. no. we can have = $2^{16}-1$
- ✓ TREG0 (Temporary reg.) (multiplicand can be held)
- ✓ ACC & PREG can also be shifted, but in 2 cycles

5 bit TREG-1 : Specifies no. of bits by which scaling shifter should shift

ARAU – Auxiliary Register ALU

has :

- ✓ 8 : 16 bit ARs (AR0-AR7)
- 1 : 3 bit ARP
- 1 : unsigned 16-bit ALU

f : ✓ ARAU calculates indirect addresses by using inputs ARs, 16 bit INDX, ARCR.

✓ ARAU can auto index current AR while data memory locn is being addressed and can index either by ±1 or by contents of INDX.

eg. consider instruction

MOV A, M : Its indirect addressing.

Data stored in HL register pair is located by M & stored in A.

eg ② INX H : Incrementing HL.

H	L
20	00.

INXH : 20 01

INRH : 21 00

We can execute these 2 instructions with 1 inst.

MOV A, M
INX H } ⇒ LACC *+, 0

LACC → Load Accumulator

* → Auxiliary register content

\+ → Increment

0 → No shifting

(; 1 → shifting once ⇒ ×2

; 2 → ×4 ...)

✡ DECIMATION BY INTEGER FACTORS.

✓ suppose we have CD & CD reader.
Both have different f_s → sampling freq
Now, how will we synchronise that?

* Also, large f_s requires large size. So, f_s needs to be compressed at times (reducing f_s).
(∵ storage requirement needs to be reduced)
Idea: If we are able to get the signal from lower f_s, large isn't req^d. (Knowing its oversampled)

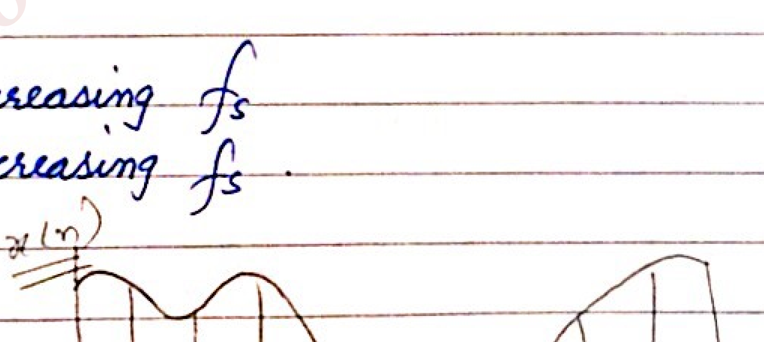
✓ for Audio-Video signal
↳ f_s more : Clarity more
f_s less : Clarity less
So, in such signals its req^d to ↑ f_s.

✓ Now, reduction in f_s should be done by keeping in mind Nyquist Criteria

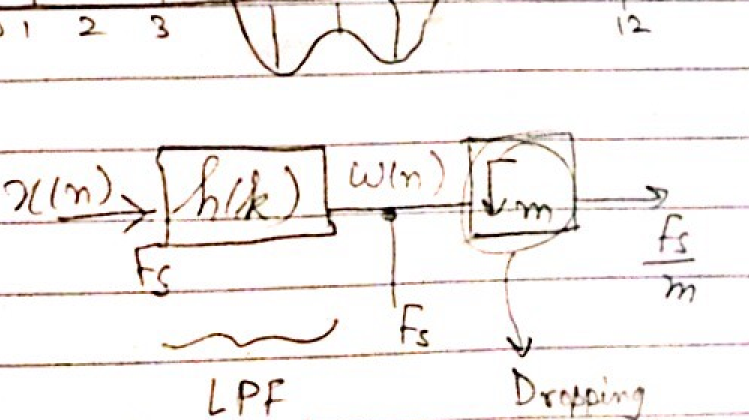
* Decimation : Decreasing f_s
* Interpolation : Increasing f_s.

Case ⓐ ✡ DECIMATION

Consider i/p signal:



Pass signal through LPF & remove high γ components first

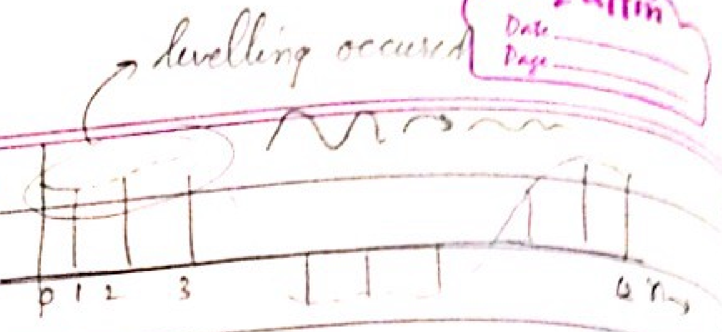


LPF
(remove drastic changes)

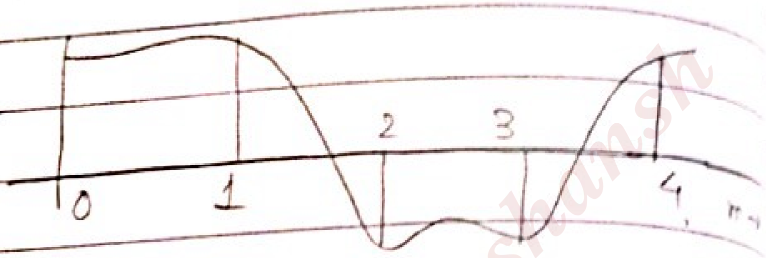
Dropping out samples by a factor of m

Time Domain

After passing through LPF w(n)



After decimation y(m) to $\frac{1}{3}$ fs.

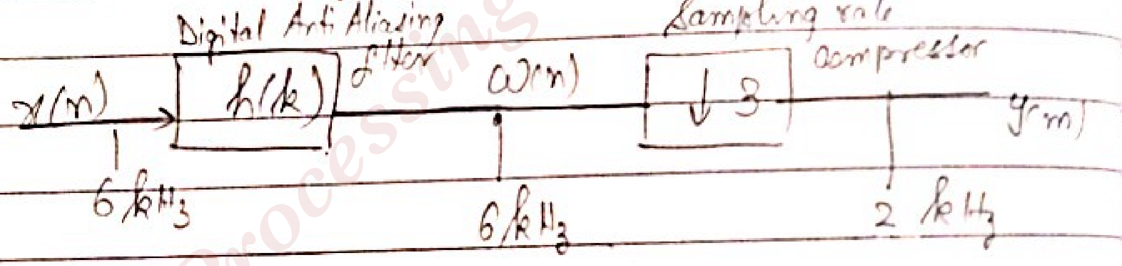


i.e 4 samples instead of 12.

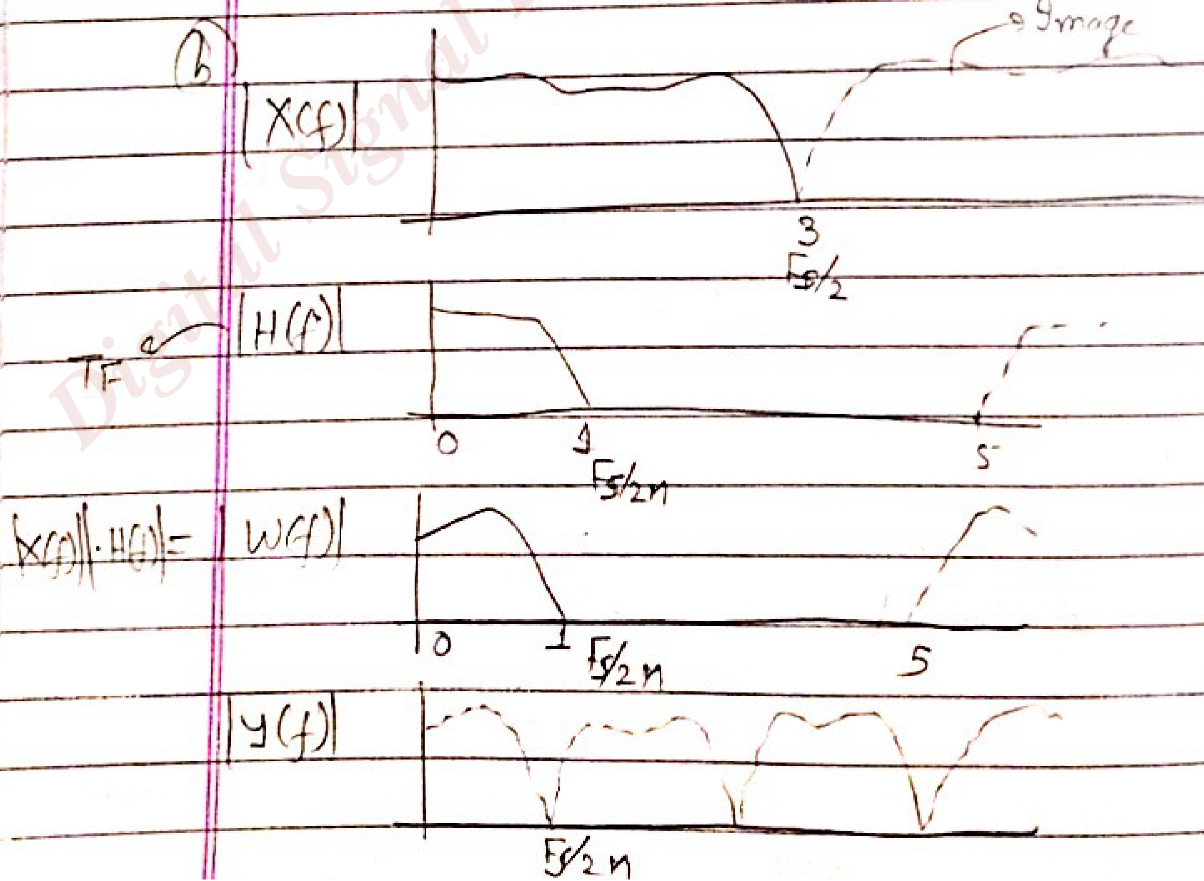
Freq Domain

Consider x(n) freq = 6kHz as i/p signal.

(a)



(b)



Case b ★ INTERPOLATION

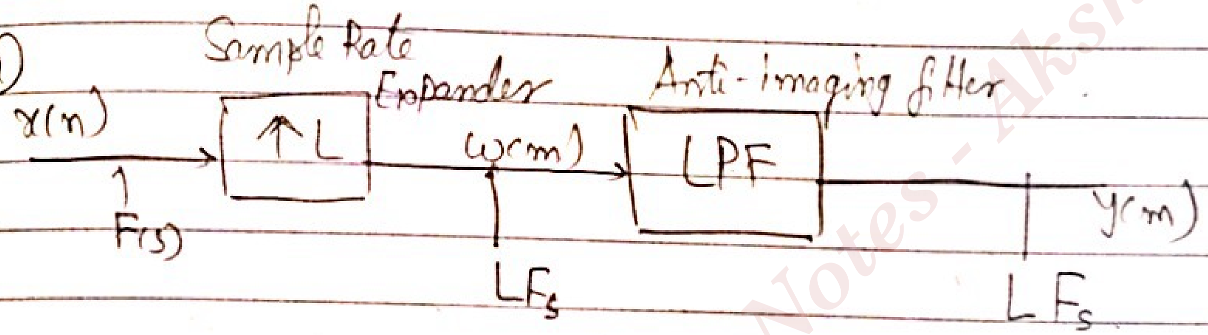
Interpolation by integer factor

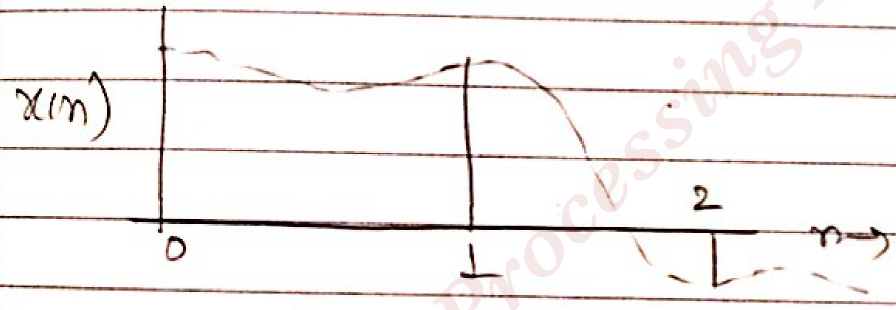
$$y_m = \sum_{k=-\infty}^{\infty} h(k)\, w(m-k)$$

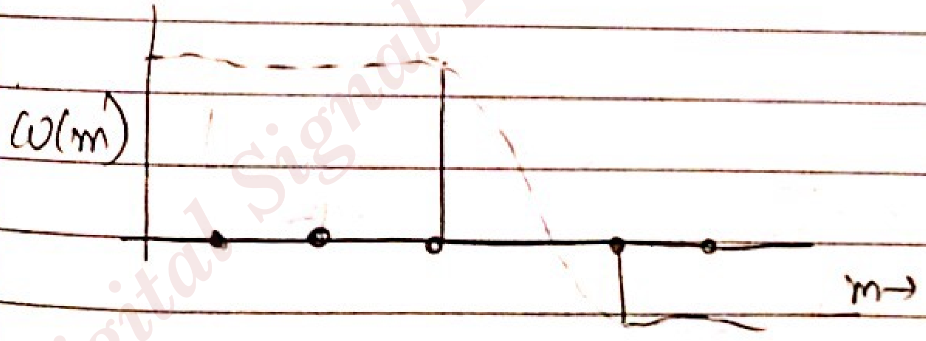
Time Domain

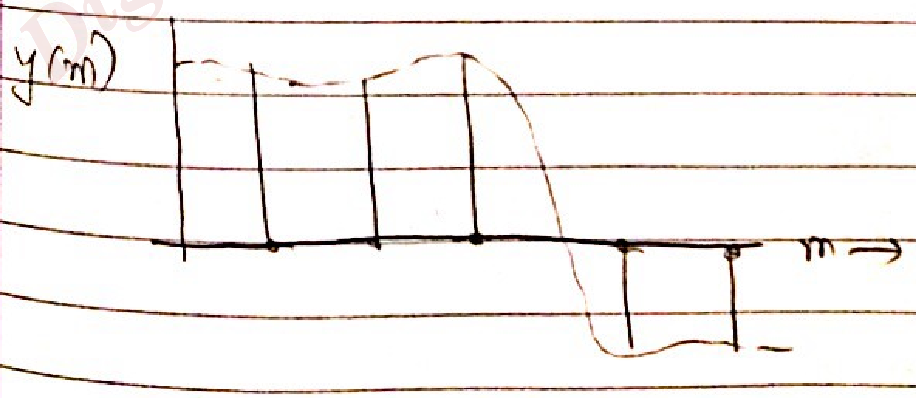
$$w(m) = \begin{cases} x\left(\frac{m}{L}\right) & ; m = 0, \pm L, \pm 2L \} - - - \\ 0 \end{cases}$$

(a)

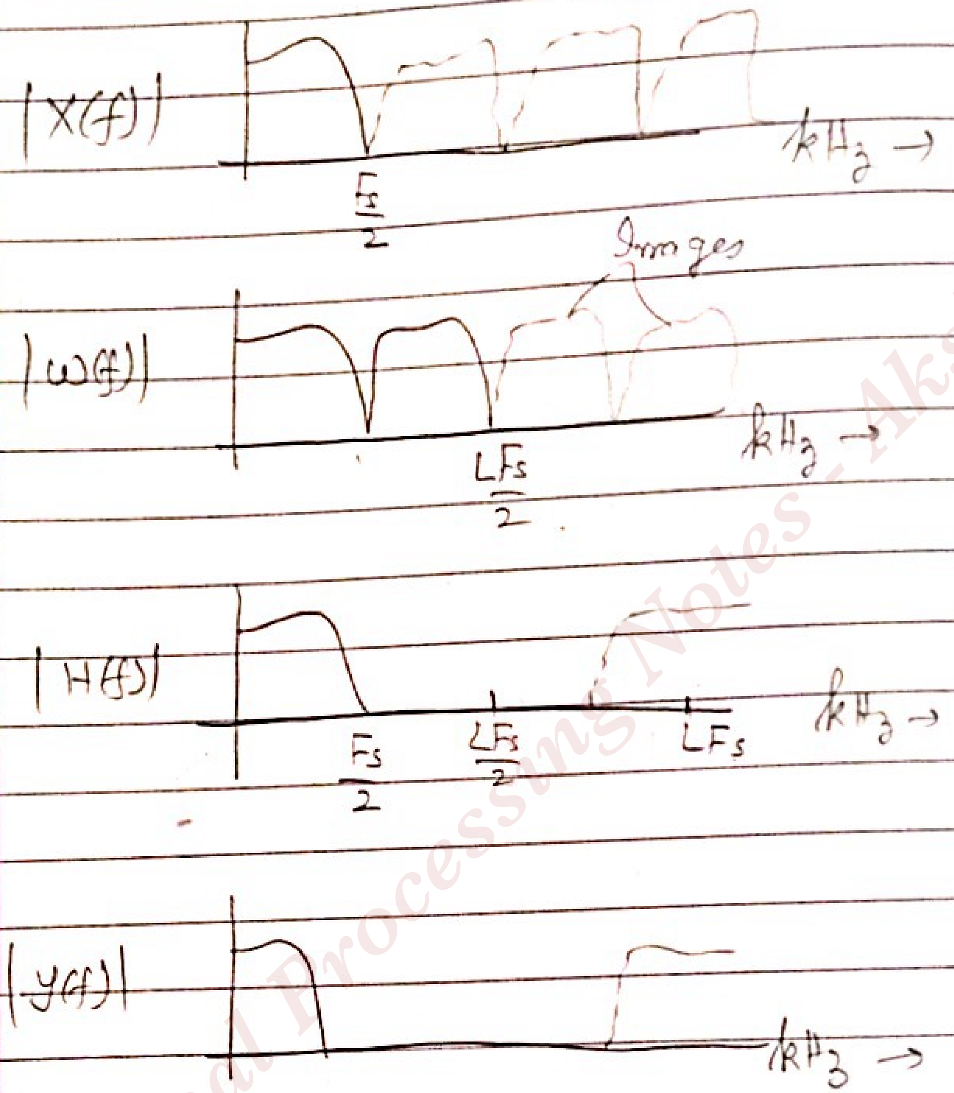








Freq. domain



§ MULTIRATE PROCESSING

✱ Till now, we did Integer decimation & Interpol^n
Now,
for Non-Integer value :
Use Interpol^n & Decim^n together
Interpolation — followed by — Decim^n.

(∵ If Decim^n comes first, ∃ loss of info)

* Time domain & freq. domain diagrams of Interpolⁿ followed by Decimⁿ (in Book)

* Design of practical sampling rate converters :

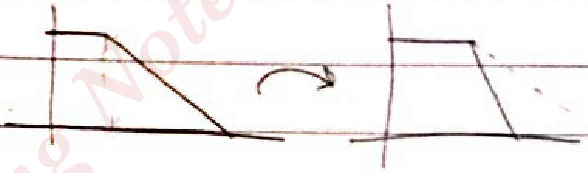
1. Down Sampling

remains same ← (we don't change PB)

Pass Band f : $0 \leq f \leq f_p$; δ_p

Stop Band f : $\frac{F_s}{2M} \leq f \leq \frac{F_s}{2}$; δ_s

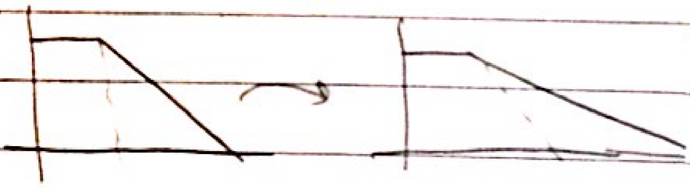
reducing SB freq & its ripple content



2. Up sampling :

$0 \leq f \leq f_p$; δ_p

$\frac{F_s}{2} \leq f \leq L\frac{F_s}{2}$; δ_s



* Individual Stages design.

✓ PB $0 \leq f \leq f_p$

✓ SB, $f_{s_i} = F_i - \frac{F_s}{2m} < f < F_{\frac{i-1}{2}}$.

$F_i \rightarrow$ o/p δ_s for ith stage

✓ PB ripple $= \frac{\delta_p}{I}$ $\left((\delta_p)_{new} < \frac{\delta_p}{\boxed{I}}\right)$ → no. of stages

✓ SB ripple $= \delta_s$ $\left((\delta_s)_{new} > \delta_s\right)$

✓ o/p sampling rate, $F_i = \frac{F_{i-1}}{m_i}$

✓ N_i, Δf_i → for i^{th} stage.

✓ $N_i \simeq \frac{D_\infty(\delta_p, \delta_s)}{\Delta f_i} - f(\delta_p, \delta_s)\Delta f_i + 1$

✓ Multiplic^n per second, MPS = $\sum_{i=1}^{I} N_i F_i$

✓ Total storage requirement, TSR = $\sum_{i=1}^{I} N_i$

✓ Initial sampling rate is $F_0 = F_s$ & final is $F_I = \frac{F_s}{M}$.

✩ Usually, if decim^n factor is more, we get more no. of coeff. So, its broken into different stages.

eg: If decim^n has to be done by $\frac{1}{50}$.

$50 = 2 \times 5 \times 5$.

I can divide it into 3 stages

$\frac{1}{2} \rightarrow \frac{1}{5} \rightarrow \frac{1}{5}$

Just like this, we could have had diff^t combin^{ns}. How to choose the optimum combin^n — seen through MPS, TSR (min. Value)

✩ Optimum M & I is for <u>min</u> MPS & TSR.

Q Given a signal $x(n)$, $f_s = 2.048$ kHz
It has to be decimated by a factor of 32 to yield a signal at a sampling freq. of 64 Hz.
Signal band of interest extends from 0 – 30 Hz
The anti-aliasing digital filter should satisfy :-

PB deviation : 0.01 dB
SB deviation : 80 dB
PB : 0 - 30 Hz
SB : 32 - 64 Hz

The signal components b/w 30 - 30 Hz should be protected from aliasing. Design suitable one-stage decimator :

Soln. $\Delta f = \frac{(32-30)}{2048} = 9.766 \times 10^{-4}$.

$\delta_p = 0.00115$ $\left(20 \log(1+\delta_p) = 0.01 \text{ dB}\right)$
$\delta_s = 0.0001$ $\left(-20 \log(\delta_s) = 80 \text{ dB}\right)$

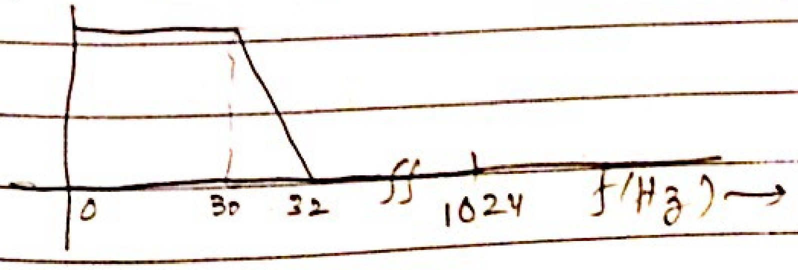
Now, estimating no. of coeff. to meet these specs:-

$$x(n) \rightarrow \boxed{\text{LPF}} \rightarrow \boxed{\downarrow 32} \rightarrow y(n)$$

Band limitation (making 0-64 Hz)

down rating f_s.

freq. response of this is till $\frac{f_s}{2}$ i.e 1024 Hz.



Taking Optimum filter design (assuming)

$$N \approx \underbrace{\frac{D_\infty(\delta_p,\delta_s)}{\Delta f}}_{} - f(\delta_p,\delta_s)\Delta f + 1$$

can be found using fixed formulas discussed before.

After putting values, we get

$N \simeq 3947$ coeffs

(obviously too large → no filter can implement it)

So, for implementing, down rate it to multiple stages.

We have to do decimation of 32, So, for

2 stages $\downarrow 8 \rightarrow \downarrow 4$

or $\downarrow 16 \rightarrow \downarrow 2$

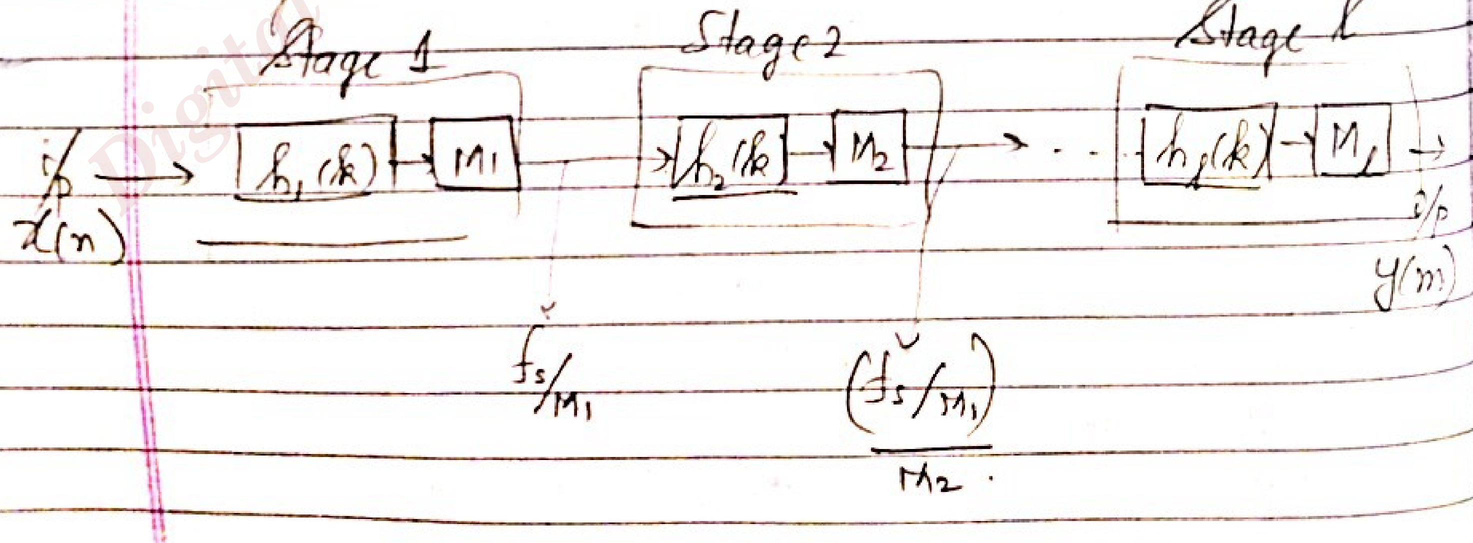
or - - - -

3 stages $\downarrow 4 \rightarrow \downarrow 4 \rightarrow \downarrow 2$

or - - -

Now, optimise decimation factor & no. of stages.

In general



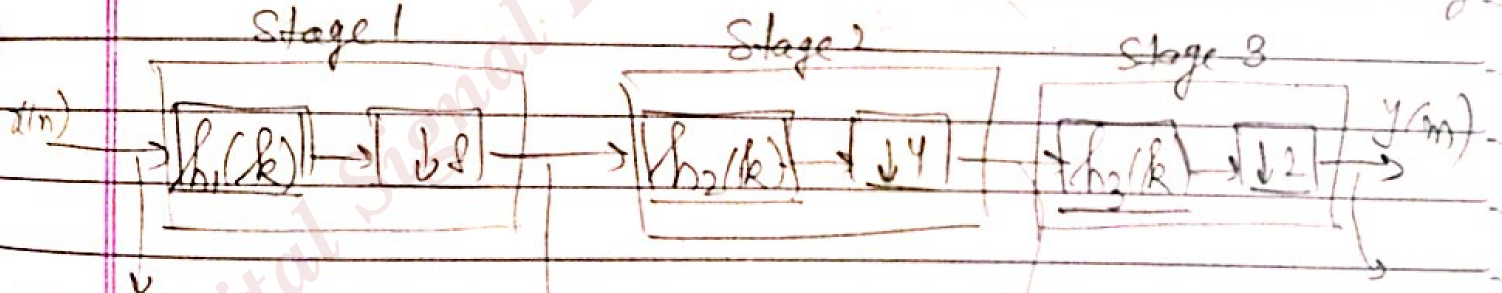
eg 9.2 (||ly, eg. 9.3, 9.4, 9.5)

Q Design 3 stage decimator used to reduce f_s from 3072 kHz to 48 kHz.
Assuming decimation factors of 8, 4 & 2 indicate sampling rate & o/p of each stage

Given specs: i/p f_s, F_s = 3072 kHz
decimⁿ factor, M = 64
PB ripple = 0.01dB
SB ripple = 60 dB
freq. band of interest = 0 - 20 kHz

So, $f_s|_{min}$ should be $\geq 2 \times 20$ kHz (Nyquist) = 40.
So, we have taken 48 kHz

Structure :



f_s 3072 kHz

$f_{s_1} = \frac{3072}{8}$ kHz $= 384$ kHz

$f_{s_2} = \frac{384}{4}$ kHz $= 96$ kHz

$f_{s_3} = \frac{96}{2} = 48$ kHz

So, f_{s1} = 384 kHz
f_{s2} = 96 kHz
f_{s3} = 48 kHz.

Now computing MPS & TSR to decide the optimum decimⁿ.

Now, designing filter :

eqⁿ :- $$\star \quad \boxed{f_{si} = F_i - \frac{F_s}{2M}} \quad ; i = 1, 2, 3$$

f_{si} → SB edge freq of ith stage.

F_i → o/p sampling rate for stage i

i = 1

$$f_{s_1} = 384 - \frac{3072}{2 \times 64} = 360 \text{ kHz}$$

edge freqs :- PB : 0, 20 kHz
SB : 360 kHz

IIly, **i = 2**

$$f_{s_2} = 96 - \underset{24}{\frac{3072}{2 \times 64}} = 72 \text{ kHz}$$

edge freq :- $\underbrace{0, 20}_{PB}, \underbrace{72, 192}_{SB}$ kHz

i = 3

$$f_{s_3} = 48 - \frac{3072}{2 \times 64} = 24 \text{ kHz}$$

edge freq :- $\underbrace{0, 20}_{PB}, \underbrace{24, 48}_{SB}$ kHz

(i) Now, overall decim^n factor = 64 (3072/48)

(ii) Seeing possible combin^{ns} for 2 stages :

32 × 2
16 × 4
8 × 8

(iii) Taking 3 stages :

16 × 2 × 2
8 × 4 × 2

(iv) 4 stages

8 × 2 × 2 × 2
4 × 4 × 2 × 2

} Seeing all possibilities.

∀ these possibilities find MPS & TSR ∀ of them, see the min. value & choose these factors.

Q. Design 2 stage decimator that downsamples by factor of 30. Specify fs at i/p & o/p of each stage of decim^n.

Given Specs :

i/p F_s : 240 kHz

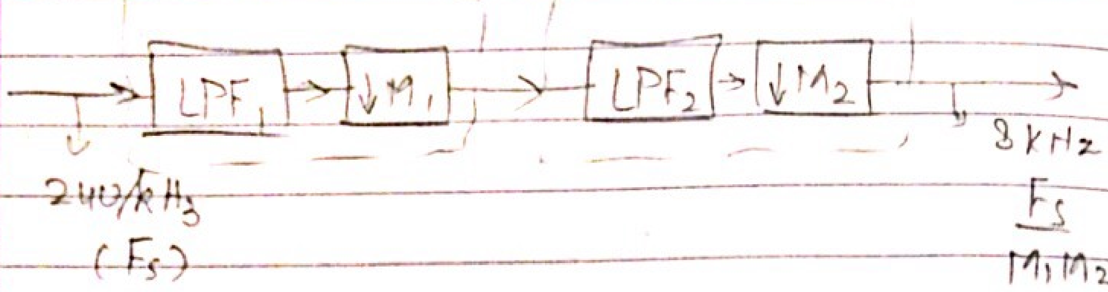
highest freq of interest in data : 3.4 kHz

PB ripple, δ_p : 0.05 (already value is given, not in dB)

SB ripple, δ_s : 0.01

filter length, $N = \frac{-10 \log(\delta_p \delta_s) - 13}{14.6 \Delta f} + 1$

↳ Δf : Normalised trans^n width



The combin^{ns} of factors can be

(i) 15 × 2
(ii) or 10 × 3
(iii) or 6 × 5 .

(i) $M_1 = 15, M_2 = 2.$

$\frac{240}{15} \to \frac{16}{2} \to 8$ kHz

1st stage : bandedge freq$_s$: 3.4 kHz & 12 kHz $\left(16 - \frac{240}{2\times30}\right)$

$$\Delta f = \frac{12 - 3.4}{240} = 0.0358$$

$$\delta p = \frac{0.05}{2} = 0.025, \ \delta_s = 0.01, \ N_1 = 45$$

2nd stage bandedge freq$_s$: 3.4 kHz & 4 kHz

$$\Delta f = 0.0375 \left(\frac{4 - 3.4}{16}\right)$$

$$\delta p = \frac{0.05}{2} = 0.025 ; \ \delta_s = 0.01 , \ N_2 = 43$$

||ly do for 10×3 & 6×5 cases & tabulate MPS & TCR.

	MPS	TSR
$M_1 = 15, M_2 = 2$	1064×10^3	88
$M_1 = 10, M_2 = 3$	1088×10^3	88
$M_1 = 6, M_2 = 5$	1368×10^3	119

Seeing min. value

So, optimum → $M_1 = 15, M_2 = 2$
decimⁿ factors.

Q. Sampling rate of $x(n)$ has to be reduced from
- ✓ 96 kHz to 1 kHz
- ✓ highest freq. of interest after decimⁿ = 450 Hz → tells about PB edge freq.
- ✓ Use optimal FIR filter
- ✓ PB ripple = $\delta_p = 0.01$
- ✓ SB deviation, $\delta_s = 0.001$

Design efficient decimator.

Here overall decimⁿ factor = 96.
Now, see optimum decimⁿ factor for one or two or 3 or 4 - stages
Using computer, seeing values for diffᵗ stages

no. of coeff at each stage

	N_1	N_2	N_3	N_4	M_1	M_2	M_3	M_4	MPS	TSR
①	4881				96	–	–	–	48881000	4886
②	131	167			32	3	–	–	540000	298
③	25	34	117		8	6	2	–	485000	176
④	11	13	17	120	4	4	3	2	496000	161

Choosing min. values of MPS & TSR.

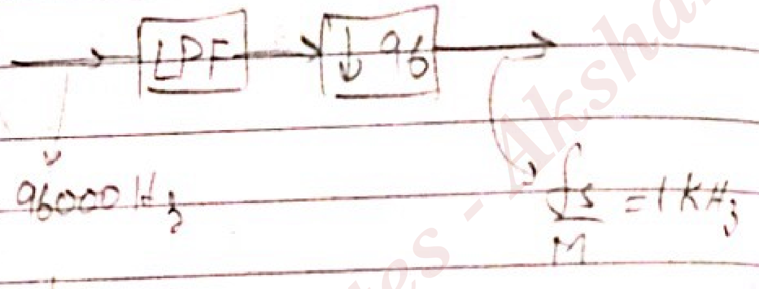
Clearly ① & ② are ruled out

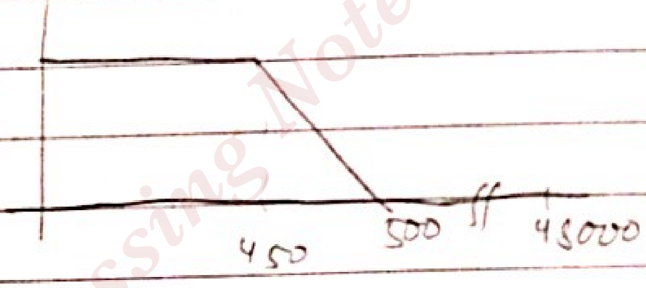
We have to choose b/w ③ & ④

Based on our requirement we choose b/w ③ & ④

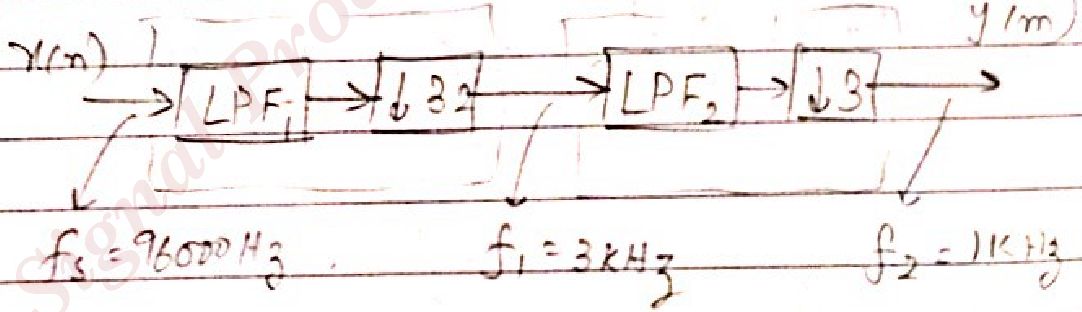
Graph :

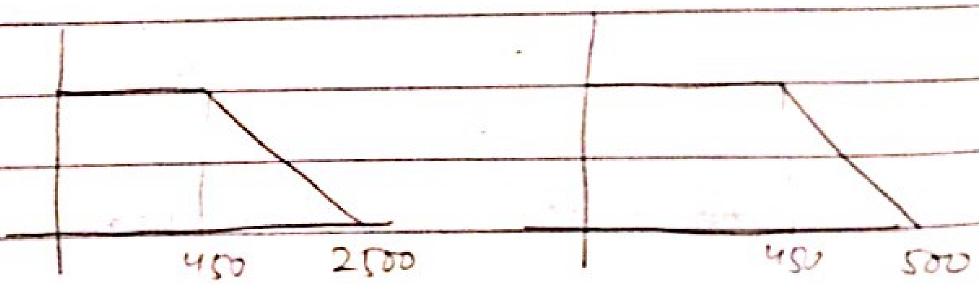
for single stage :





2 stages :





$$f_{s1} = 3000 - \frac{96000}{2\times 96} = 2500$$

$$\Delta f_1 = \frac{2500-450}{96000}$$

Illy for ③ & ④ stages (helped to choose)

Q A digital audio sys. exploits OVERSAMPLING techniques to relax requirements of analog anti-imaging filter.

Specs:

Baseband : 0 - 20 kHz.

Fs : i/p fs 44.1 kHz.

o/p fs 176.4 kHz.

SB attenuation 50 dB

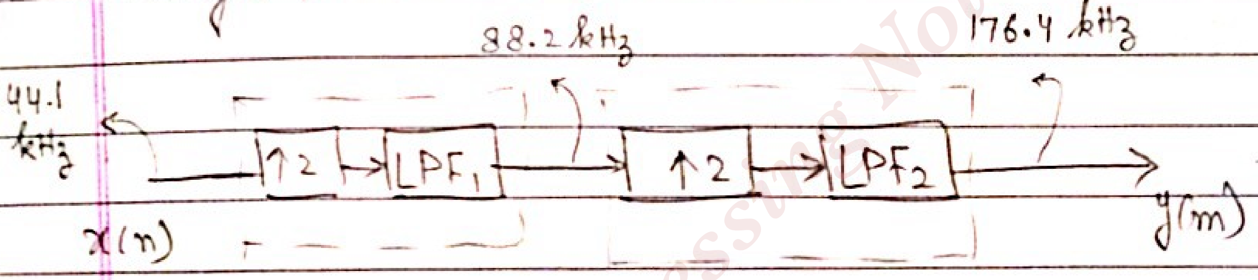
PB ripple 0.5 dB

Trans^n width 2 kHz.

SB edge freq. 22.05 kHz

Design suitable INTERPOLATOR

done through interpol^n



Idea: find no. of coeff. So, value of filter design coeff. can be evaluated.

→ find SB edge freq.

$$\omega_c \text{ i.e } f_{SB} = \frac{F_s}{2} + \frac{F_i}{2L_i}$$

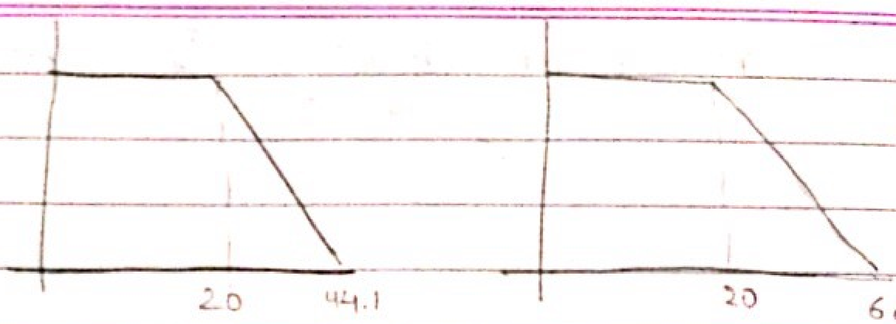
i/p fs

overall or o/p fs

overall L

✓ having known SB edge freq, we get trans^n band (Δf)

✓ Normalise it wrt APPROPRIATE Sampling freq.



got using fsb formula (prev. page)

$\Delta f_1 = \frac{44.1 - 20}{88.2}$ $\Delta f_2 = \frac{\boxed{66.15} - 20}{176.4}$

$0.5 = -20 \log(1+\delta_{p_1})$ $\delta_{p_2} = 0.0296, \ \delta_{s_2} = 0.00316$

$50 = -20 \log(\delta_{s_1})$ ✓ $\boxed{N_2 = 6}$

$\delta_{p_1} = 0.0296$

$\delta_{s_1} = 0.00316$

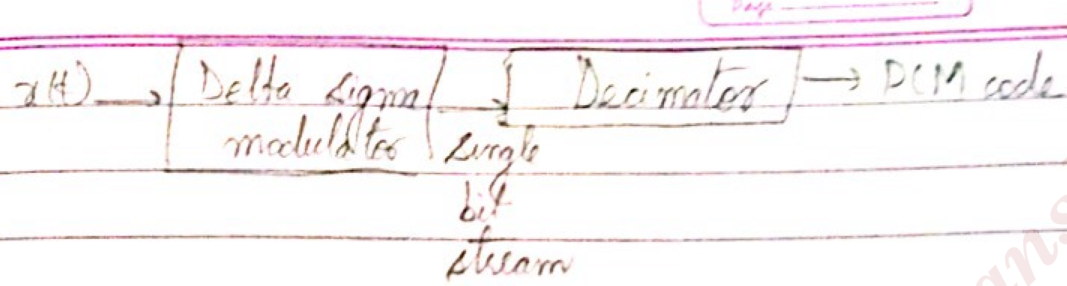
✓ $\boxed{N_1 = 83}$

no. of coeff. got in every stage.

Tabulated result for interpolator :

No. of stages	Interpolⁿ factor, L	filter length N_i	Normalised transⁿ width, Δf_i	PB ripple δ_p	SB ripple δ_s
1	4	146	0.04535	0.05925	0.00316
2	2	83	0.26162	0.0296	0.00316
	2	6	0.27324	0.0296	0.00316

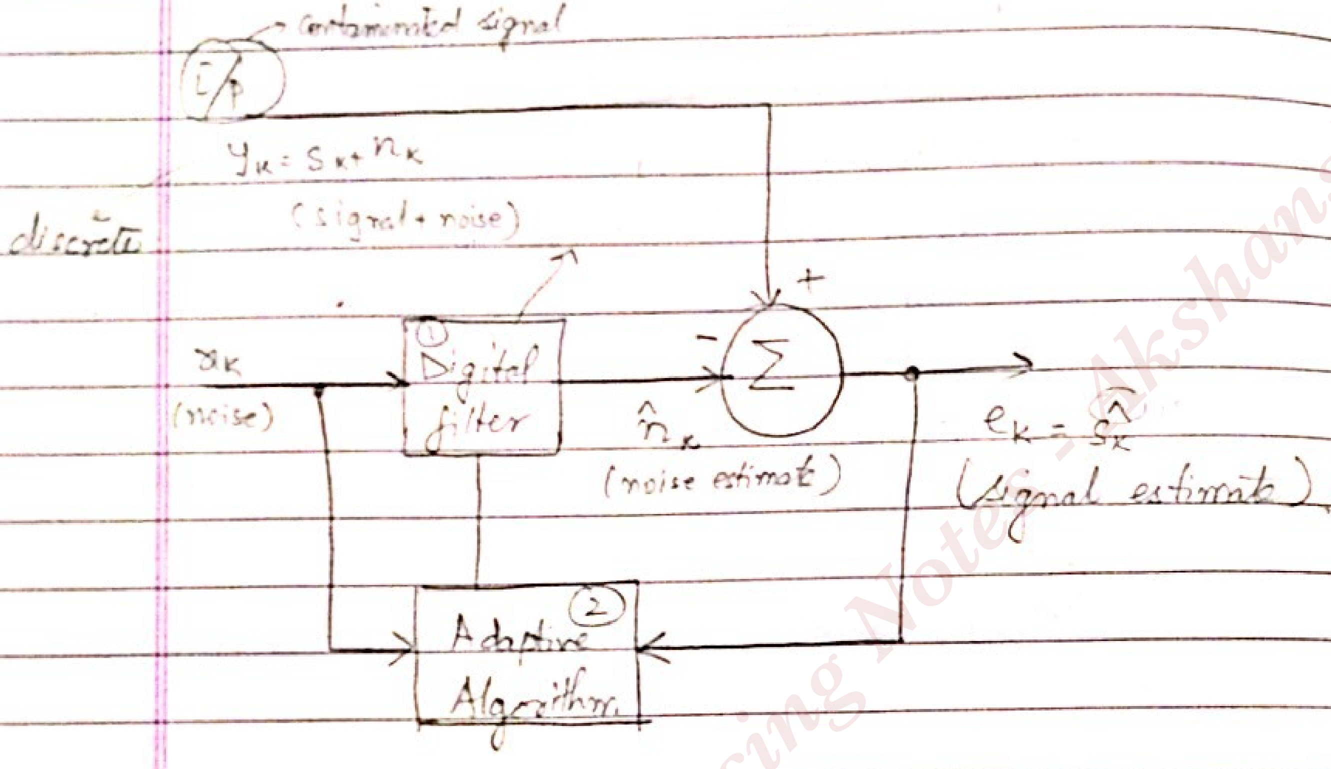
- Multirate signal processing applications.
  - → Data storage (excess data can be there in case of ADC. So, decimator can be useful)
  - → In communicⁿ, when Δ-σ modulⁿ is being done.



→ ✓ D → A in compact hi-fi systems.
✓ high quality A → D conversion of digital audio
↳ Interpolⁿ is done for quality improvement
(Block diagram)

→ Sub band coding of speech signals.
i.e, in the processes where multiple messages are multiplexed & sent through transmitter (radio stations).
Decimation is done before multiplexing & Interpolⁿ is done after demultiplexing

ADAPTIVE DIGITAL FILTERS



Sys. parameters change wrt environmental ~~con~~ cond^n changes (eg. temperature)

If the parameters don't change, we get o/p which is unaffected, such a sys. is called Adaptive Control System.

here, sys. is filter, which should be adaptive.

- Sys. parameters get changed ≡ Noise
  ↳ eg. variation in pixel values in Photography
  ↳ ≡ Noise
- Original signal + Noise = Contaminated signal.

Idea: Subtract noise signal from contaminated signal to get wanted signal

$(\hat{\ })$ → making estimate

In the block diagram (↙),

Digital filter makes Noise estimⁿ

↳ Idea : Make any signal of noise (x_k) on your own. Get its coeff. values & then do $y_k - x_k$. If we get our wanted signal back, the signal of noise we made was correct. If not, keep changing it

Mathematically,

$$* \quad \hat{s}_k = y_k - \hat{n}_k$$

$$\hat{s}_k = s_k + n_k - \hat{n}_k$$

↳ for $s_k = \hat{s}_k \Rightarrow n_k = \hat{n}_k$

i.e, noise signal should be exactly matched to get signal back

* APPLICATION of Adaptive Filters

①/ EEG, ECG Signals.

↳ Micro sensors in body. That is amplified to Mili level of voltage. That level is detected through sensors.

Noise comes if ∃ any movement of hand/ anything in the body.

Now, this movement signals & reqᵈ signals are in same freq & amplitude, then, filtering them out using Adaptive filters.

② ✓ Digital Communicⁿ – jamming signals
narrow band high power
(wideband spectrum) signals within the band.

③ ✓ Data communicⁿ through telephone lines

§ Summary

✓ When its necessary to have variable filter characteristic – adapting to condⁿ

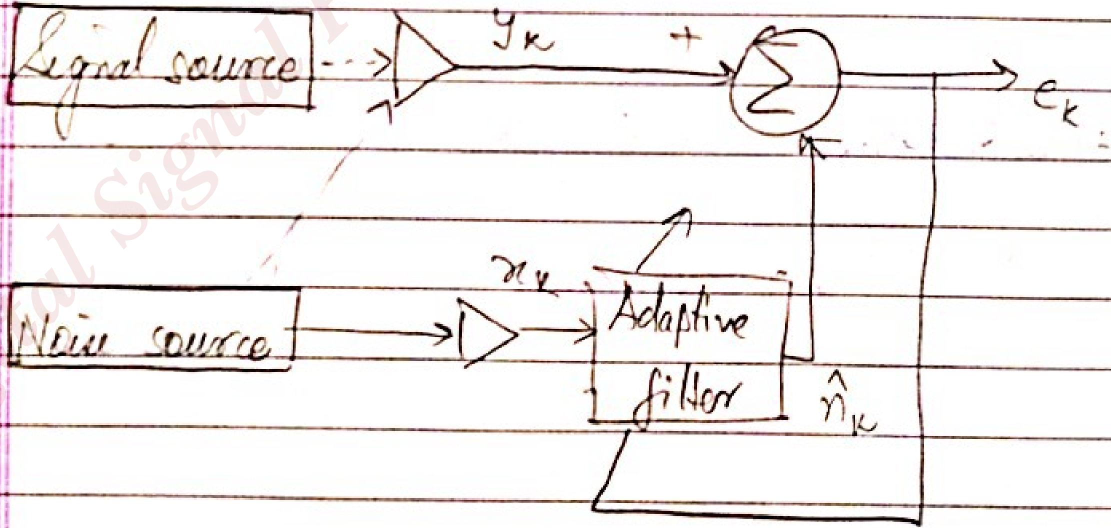
✓ Noise band is unknown or varies with time.

✓ When there is spectral overlap

* Block diagram representⁿ:

→ Adaptive noise canceller

fig 10.3 (complete)



+ other block diagrams for other applicⁿˢ (line enhancer, System modelling . .)

To design any Adaptive DSP : Main components:

1) FIR filter (tuning o/p) $\hat{n}_K = \sum_{K=0}^{N-1} w_K(i) x_K$
2) Algorithm (for tuning filter)

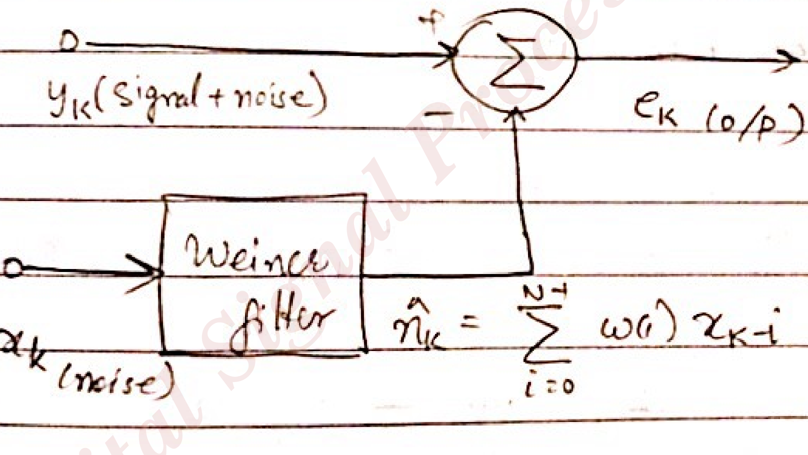
Ⓐ • The Weiner filter

↳ used for designing adaptive filter
↳ aim : error = diff. b/w actual & estimated should be min.
i/p.

mathematically,

$$e_K = y_K - \hat{n}_k = y_k - W^T X_k$$

$$= y_k - \sum_{i=0}^{N-1} \omega(i)\, x_{k-i}$$



2 signals x_K & y_K are applied simultaneously.

y_K has noise signal components ;

>noise signal can be :
1) Correlated with x_k
2) Uncorrelated with x_k

Weiner filter produces the optimum estimate of 1'st part and is subtracted to get e_K

→ correlated with x_K

* Relⁿ on one signal onto itself : Auto correlⁿ
others : Cross

Tuning filter / extracting info :
We should see error = 0.
Seeing signal ⇒ seeing power. So, seeing square of error signal

$$X_k = \begin{bmatrix} x_k \\ x_{k-1} \\ x_{k-2} \\ \vdots \\ x_{k-(N-1)} \end{bmatrix} , \quad W = \begin{bmatrix} W(0) \\ w(1) \\ \vdots \\ w(N-1) \end{bmatrix}$$

The square of error, $e_k^2 = y_k^2 - 2y_k X_k^T W + W^T X_k X_k^T W$.

Mean square error (MSE), J is obtained by taking E (expectation) on both sides assuming X_k & y_k are jointly stationary

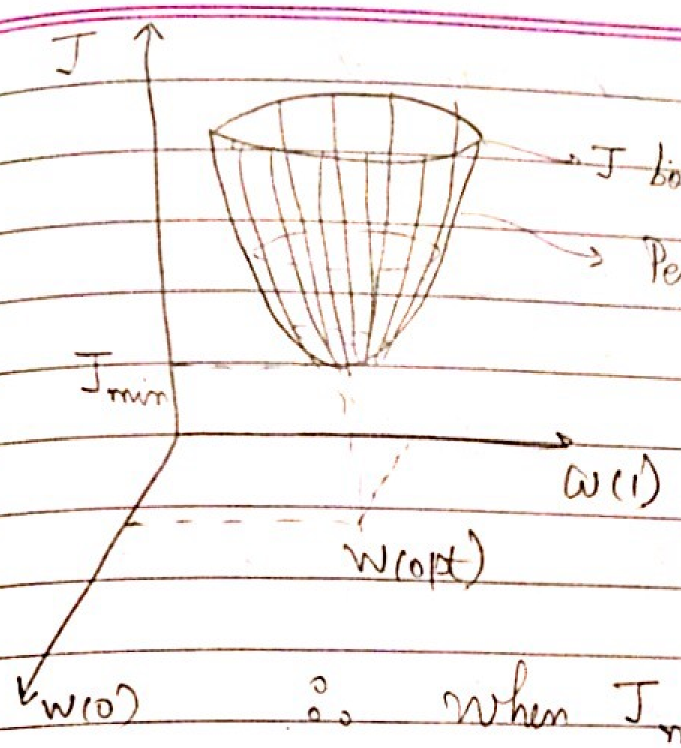
$$J = E(e_k)^2 = E(y_k)^2 - 2E(y_k X_k^T W) + E(W^T X_k X_k^T W$$

$$= \sigma^2 - 2P^T W + W^T R W.$$

↳ estimating error power & trying to make it min.

Taking min. value ⇒ do $\frac{dJ}{dW} = 0$ & see max & min. values.

$$\nabla = \frac{dJ}{dw} = -2P + 2RW$$



→ J bowl (can be of any size)

→ Performance surface & is non-ve

∴ when $J_{min} \Rightarrow \nabla = 0 \Rightarrow W \longrightarrow W_{opt}$.

Hence, $\boxed{W_{opt} = R^{-1}P}$

- → Weiner Hoffman eq^n.
- → R: autocorrelation of ~~x_k~~ X_k
- → P: Prior info of cross correl^n.
- → maybe computationally complex. So, we see other methods.

~~Est~~ Task: Adjust filter weights: W(0)...W(N−1) using suitable algorithm to find optimum pt. on performance surface.

→ Limitations:-

- → R, P needed (not known prior)
- → Requires matrix inversion

⋮

Q. Estimate desired signal at o/p of adaptive noise canceller, given by :

$$\hat{s}_k = y_k - \hat{n}_k = s_k + n_k - \hat{n}_k$$

Now, error can be seen as,

$$\text{min. of } E(\hat{s}_k^{\,2}) = E(s_k^2) + \text{min. } E(n_k - \hat{n}_k)$$

↳ error min. ⇒ SNR is better

(⇒ Noise is min.

⇒ Signal is max)

§ Other Algorithms (overcoming Weiner filter)

Ⓑ **Least Mean Square method** (LMS)

(sys is stable, but ∃ more MPS, TSR)

Ⓒ **Recursive Least Square method** (RLS)

(superior convergence quality)

Ⓓ **Kalman's filtering.**

Ⓑ Basic LMS Adaptive Algorithm

Mean Sq. error (MSE) gives W_{opt} instead in LMS, coeffs are adjusted from sample to sample to minimize MSE.

This leads to descending along surface of J bowl

LMS is based on Steepest descent algorithm.

When W_k value is updated from sample to sample,

$$\therefore W_{k+1} = W_k - \mu \nabla_k$$

gradient

correction/weightage factor $(0 < \mu < 1)$

$$\nabla = \overset{(cross)}{-2P} + \overset{(auto)}{2RW}$$

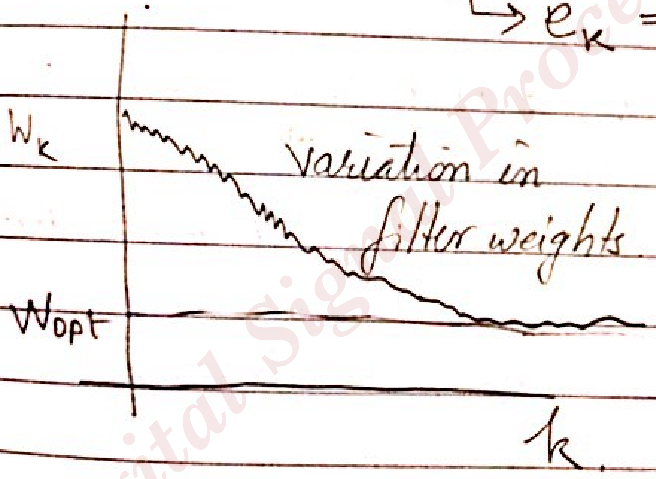
$$\therefore \nabla_k = -2P_k + 2R_kW_k = -2X_kY_k + 2X_kX_k^TW_k$$

$$= -2X_k(\underbrace{Y_k - X_k^TW_k}_{e_k}) = -2e_k x_k$$

$\therefore$

$$W_{k+1} = \cancel{W} W_k + 2\mu e_k X_k$$

$$\hookrightarrow e_k = y_k - w_k^T X_k$$



- We need instt. values of R & P
- W_{k+1} is only estimates but slowly improves & converges to W_{opt}.

→ Limitations :

① Effects of non stationarity :

In on stationary environment, the min. point keeps changing ; its orientation & curvature may also be changing

② Effect of signal component on interference i/p channel

- - - ** see

③ Computer word length requirements

digital filter :- $\hat{n}_k = \sum_{i=0}^{N-1} w_k(i)\, x_{k-i}$

Adaptive algorithm : $W_{k+1} = W_k + 2\mu e_k X_k$

$e_k = y_k - w_k^T x_k$

↳ finite wordlength effect ∴ may grow without limit ⇒ error grows.

→ Problems on implementing adaptive alogorithms

✓ possible non convergence of adaptive filter to optimum solⁿ

✓ filter o/p may contain some noise which causes random fluctuations.

✓ premature terminⁿ of algorithm.

→ **Most of adaptive sys. in literature shows :-**

x_{k-i} & y_k : fixed pt. no. b/w 8 & 16 bits

Quantizⁿ of coeff, multiplier & accumulator.

16 - 24 bits   8×8 to 24×16 bit   16 ↔ 40 bits

④ Coefficient drift

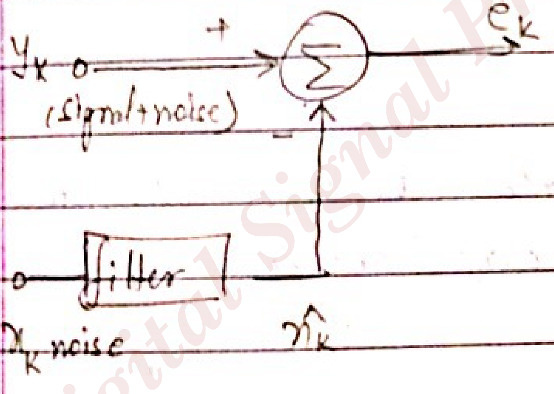
eg : narrow band signal coeff. drift from optimum & grow slow by demanding more word length.

Can be compensated by introducing a leakage factor 'δ' which will correct drift

- Sophisticated LMS Algo:-
  ① Handle complex data
  ② Computationally advantageous.

(Fig 10.14) Simplified block diagram of freq. domain LMS filter.
→ FFT: Fast Fourier Transform
→ IFFT: Inverse Fast Fourier Transform.

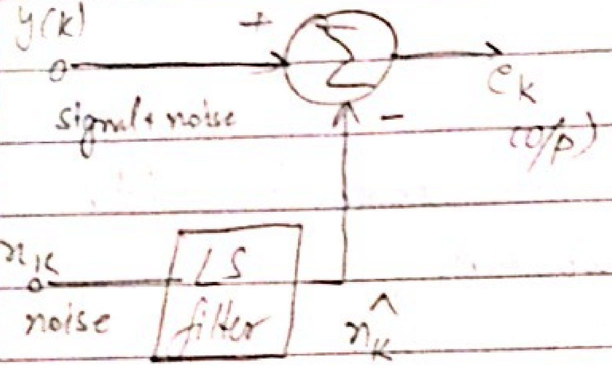
*LMS Algorithm requires 2N+1 multiplic^{ns} & 2N+1 additions for each new set of i/p & o/p samples.



Most signal processors are suited for implementing LMS algorithm (Multiply accumulate)

Hardware implement^n of real time adaptive filtering
↳ Block diagram #.

☆ Recursive Least squares algorithm (RLS algorithm)



$$y_k = \sum_{i=0}^{n-1} w_i x_{k-i} + e_k$$

$$\underset{(opt)}{W_m} = R^{-1} P$$
$$= [X_m^T X_m]^{-1} X_m^T y_m$$

; where y_m, X_m, W_m are all column vectors.

$$\underbrace{\hat{x_k}}_{\text{estimate noise signal}} = \sum_{i=0}^{n-1} \underbrace{\hat{w}_{(i)}}_{\text{filter response coeff for each sample}} \underbrace{x_{k-i}}_{\text{i/p signal}} \quad ; \; k = 1, 2, -- m.$$

For each set of new data, ω_m is updated using previous values & i/p$_s$ avoiding time consuming inversion process.

$$\omega_k = \underbrace{\omega_{k-1}}_{\text{prev. coeff}} + \underbrace{\overset{\text{how fast it comes to present value}}{G_k} e_k}_{\text{updation}} \; ; \; P_k = \frac{1}{\gamma}\left[P_{k-1} - G_k X^T_{(k)} P_{k-1}\right]$$

$$G_k = \frac{P_{k-1} \, x(k)}{\alpha_k} \; ; \; e_k = y_k - X^T_{(k)} w_{k-1}$$

Cross correl^n of prev. sample

a variable

gamma $(0 < \gamma < .98)$

$$\alpha_k = \gamma + x^T(k) P_{k-1} x(k)$$

P_K is measured or computed ~~sep~~ recursively instead of $[X_K^T X_K]^{-1}$.

λ : forgetting factor : Smaller values for recent data which leads to wildly fluctuating estimates

The no. of prev. samples which affects or contributes to the value of ω_K at each sample point is called **Asymptotic Sample Length (ASL)** is given by :-

$$\sum_{K=0}^{\infty} \lambda^K = \frac{1}{1-\lambda}$$: This defines the memory of RLS filter.

When $\lambda = 1$; LS → filter has finite memory

- **Limitation of RLS algorithm**

① If $x(k)$ is zero for long time, P_K will grow exponentially $\lim_{k\to\infty} P_K = \lim_{k\to\infty}\left(\frac{P_{K-1}}{\lambda_{k-1}}\right)$

② Sensitivity to computer roundoff errors which results in -ve definite P_{matrix} & hence to instability.

③ For successful estimⁿ of ω_K, its necessary that P must be PSD (+ve semi definite) equivalent of X^TX is invertible (∵ of differentiating term in P_K)

④ Worse in multiparameter system where the parameter are linearly dependend & when algo. is implemented on a small sys with finite word length.

⑤ Problem of numerical instability may be solved by suitably factorising matrix P so that differencing is avoided.

2 such algo are Square Root Algo & UD factoriz^n Algo.

↳ UD factoriz^n algorithm :

$$P_K = U_K D_K U_K^T$$

→ Diagonal matrix

unit upper triangular matrix

* Instead of P_K, U & D are updated here.

ARCHITECTURE

Ch - 3

✡ BLOCK REPEAT REGISTERS

Registers for block oper^{ns}

RPTC, BRCR	PASR	PAER
	Program Address Start Register.	Program Address End Register

- Parallel Logic Unit (PLU)
  Performs boolen or bit manipul^{ns} req^d for high speed controllers

- Memory mapped registers
  ✓C5X :- 96 registers into page 0 of DM
  28 CPU registers & 16 i/o port registers

- Program controller :-
  has logic circuitry which decodes instructions, manage CPU pipeline.
  – entire module consists of → 16 bit program counter (PC)
  → 16 bit shift registers
  ST0, ST1, PMST,
  CBCR (Circular Buffer control Register)
  → 16 × 16 bit hardware stack register
  → Address Gener^n logic
  → Instruction register
  → INF TR - -

DP : Data Page Pointer

sign extended => keeping sign same (arithmetic right shift)

- Result analysis happens in Flag Registers or Status Registers (ST0, ST1, ST2)

ST0 bit config :

10th bit

15-13	12	11	10	9	8 – 0
ARP	OV	OVM	1	INTM	DP

ARP: Auxiliary Register Pointer (3 MSB's of ST0 stored in AR pointer)

suppose the bits are 111 (=7)
So, content is taken from AR_7

OVM: Overflow masking

1: not used for C5X So, = 1.

INTM: Interrup Mask bit (Set/Reset)

ST1 Config :

15 – 13	12	11	10	9	8-7	6	5	4	3-2	1-0
ARB	CNF	TC	SXM	C	11	HM	1	XF	11	PM

ARB: buffer

CNF: config. bit (specifies RAM posn)
= 0 : DARAM
B0 is in DM space
= 1 : DARAM
B0 is in PM space

TC: Checking data (0/1) Test / Control Flag bit

SXM: Sign extended mode

C: Carry bit

HM: Hold mode (holding processor execution)

XF: level of external flag

PM: Processor Mode (defines data shift)

PM bits	fn
00	No shift
01	left shift by 1 bit, shifted data filled with 0 ⇒ x2
10	left shift by 4 bits ⇒ x16
11	Right shift by 6 bits & sign extended

* C5X- On Chip Memory :

Program ROM
Data/Program DARAM
Data/Program SARAM

} Total address rang of 224 K words x16 bits
divided into 4 memory segments.

1) 64 K word PM
2) 64K word DM
3) 64K word I/O ports
4) 64 K word global DM

* Seeing how instructions are being executed :

Instruction sets :

1) Data transfer
2) Logic instructions
3) Control instruction
4) Arithmetic -ADD/SUB
5) Multiply

* Data Page Memory (0-8) : 9 bits
Data Memory (0-6) : 7 bits.
ARP : 3 bits
Short immediate data (0-255) : 8 bit
Long immediate data (0-65535) : 16 bit

Syntax of operands :

indirect addressing

$\circledast$

* +

* − → autodecrement AR

increment AR after fetching data

* 0+ → Increment, after fetching data : Incrementⁿ amount is specified in Index register

* 0− → Increment, by amount in index register.

*BR0+

*BR0−

* Implementⁿ of above instructions.

eg LACC(*,0) :- Load accumulator data AR_2 with 2345 ∵ ARP = 2

→ no. of bits shifted

refer index addressing mode

ie, go to value of [1250] ie 2345 store in AR_2

Assuming content in the following

ARP 2

AR2 1250h

INDX 10 h

[1240] → [1260]

2345

content from 1240 → 1260

SXM = 0.

LAAC *1, 1

increment by 1

→ shift left i.e × 2.

	ACC	AR2
LACC *, 0	$(2345)_H$	$(1250)_H$
LACC *+, 1	$(468A)_H$	$(1251)_H$
LACC *−, 2	$(8D14)_H$	$(124E)_H$

2 3 4 5

0010 0011 0100 0101 2345 (Shift by 2)

discard ← ↓ ↓ ↓ ↓ ← 0

← 0100 0110 1000 1010

4 6 8 A

↓ ↓ ↓ ↓ ← 0

1000 1101 0001 0100 2345 (shifted by 4)

8 D 1 4

AR2 = 1250

AR2+1 = 1251

AR2 −1 = 124F

✱ TMS 320C5X.pdf

page.

160 168 Instruction Set Summary

Remember f^n of each Mnemonic

✱ Table 6.4: Accumulator memory reference instructions

Mnemonic	Description
ABS	Absolute value of ACC Zero carry bit
ADCB	Add ACCB & carry bit to Accumulator (ACC)
ADD	

Pg-186

address of DAT1

eg: ADD DAT1, 1 ; DP = 6.

DAT1 value shift by 1 bit

⇒ See memory locn 301

→ means data page pointer tells me 6. So, its value is $(300)_H$ → starting address

Before Instruction			After Instruction		
Data Memory			Data Memory		
301h		1h	301h		1h
ACC	X	2h	ACC	0	04h
	C			C	

X: don't care

C: Carry

1 bit shift

$1h \times 2^1 + 2h$

after execution, AR pointer points to ARO for NEXT instruction

eg ②
ADD *+, 0, ARO

	Before Instruction			After Instruction	
ARP		4	ARP		0
AR4		0302h	AR4		0303h
Data Memory					
302h		2h	302h		2h
ACC	X	2h	ACC	0	04h
	C			C	

eg ③
Indicates Immediate addressing
ADD #1h ; Add short Immediate

	Before		After	
ACC	X	2h	0	03h
	C		C	

eg ④
Add #1111h, 1

	Before		After	
Acc	X	2h	0	2224h
	C		C	

$1111h \times 2^{(1)}$ → 1 bit shift
$+ 2h$.

- ☆ ACCH : Accumulator High Data
- ☆ ACCL : Accumulator Low Data
- ☆ BSAR : Where data is being shifted → CALU or PALU ?
- ☆ ACCB : Accumulator Buffer ; Stores prev. data
- ☆ TREG : Temporary Register 1
- ☆ ROLB : Rotate ACC left by 1 bit through ACC buffer

★ PC : Program counter
★ SX : Sign extension bit

ex SAMM PRD; (DP = 6)

Store ACC to memory mapped register (PRD) ← Product Register, decided by DP (= 6)

	Before	After
ACC	80h	80h
PRD	05h	80h
Data Memory 325h	0Fh	0Fh

ex SAMM *, AR2); BMAR = 1Fh

	Before	After
ARP	7	2
AR7	31Fh	31Fh
ACC	080h	080h
BMAR	0h	080h
Data Memory 31Fh	11h	11h

$0011\ 0\underbrace{001\ 1111}$

Seeing LSB 7 bits = 1F

CMPR : Compares b/w AR & ARCR (Accumulator Register Count Register).
No. of bits compared = no. of CM bits

XPL : Ex-OR operⁿ

★ LT : Load data memory value to T-Register

V.Imp & useful ★ MAC : Multiply & accumulator

★ Further tables which are used:
Tables 6.5, 6.6, 6.7
↳ diff^t instruction for diff applic^ns

★ What will come in exam?
Description for any mnemonic will be given (from tables from pages 160 onwards in TMS C5X manual)
We have to tell the final value present in each register

★ For multiplic^n oper^n, data is available in PREG & TREG (not ACC)

Pg-320 Multiplic^n : ways :-

① Direct	MPY	dma
② Indirect	MPY	{ind}[, AR$_n$] → optional
③ Short Immediate	MPY	#k
④ long Immediate.	MPY	# lk

① → (PC) + 1 → PC (→ program counter)
(TREG0) × (dma) → PREG

③ → (PC) + 1 → PC
(TREG0) × k → PREG

④ → (PC) + 2 → PC
(TREG0) × lk → PREG

★ TRM bit : T-Register Mode bit.

Pg - 322 Direct mode

eg MPY DAT13 ; (DP = 18)

offset is 13

Data Page pointer = 8
⇓
address = 400

Address of data = $(400+13)_H$ = 40D

	Before	After
Data Memory		
40Dh	7h	7h
TREG0	6h	6h
PREG	36h	2Ah

7h x 6h
= 2Ah
(rest content remains same)

Indirect mode

MPY *, AR2

ARP → modified to AR2.

	Before	After
ARP	1	2
AR1	40Dh	40Dh
Data memory		
40Dh	7h	7h
TREG0	6h	6h
PREG	36h	2Ah

from here, take address of operand

Short Immediate

MPY #031h

	Before	After
TREG0	2h	2h
PREG	36h	62h

→ 2h × 31h

Long Immediate

MPY #01234h.

	Before	After
TREG0	2h	2h
PREG	36h	2468h.

eg. See pg. 324 examples.

Pg-329

NEG , (OVM = X) → (overflow mask bit) is none

2's complement of ACC.

		Before	After
ACC	$\boxed{X}$ C	FFFF F228h	00000 DD8h
	$\boxed{X}$ OV		X C X OV

FFFF FFFF

− FFFF F228

0000 0DD7

+ 1

0000 0DD8

Pg-333 Normalising content.

Syntax NORM {ind}

Execution :

```
(PC)+1 → PC
If (ACC) = 0;
    TC → 1
Else :
    if (ACC(31)) XOR (ACC(30)) = 0:
        TC → 0.
        (ACC) × 2 → ACC
        Modify current AR as specified.
    Else :
        TC → 1
```

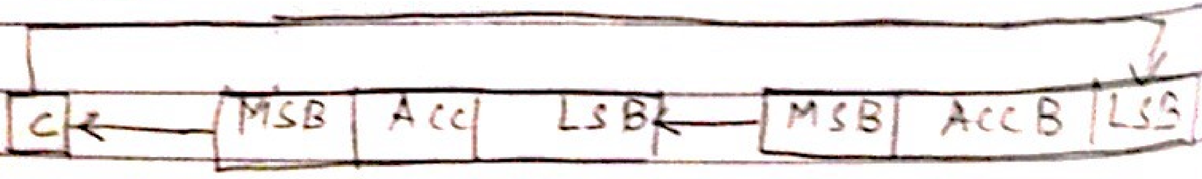
Here, TC bit is getting affected

Pg-334
eg

NORM *+ → Increment AR content

ARP		2		2
AR2		00		01h
ACC	X	FFFF F001h	0	0FFF E002h
	TC	×2	TC	
		Before ↑		After ↑

Pg-363 ROLB → Rotate left through Buffer



		Before		After
ACC	① C	0808 0808h	0 C	1010 1011h
ACCB		FFFF FFFEh		FFFF FFFDh

0 8 0 8 0 8 0 8

0000 1000 0000 1000 0000 1000 0000 1000 ← ①

copies from ACCB

1 0 1 0 1 0 1 1

1010 1011 h

end of course.