25/3

Chapter 5

$$\longrightarrow \boxed{g_{Tx}} \longrightarrow \boxed{g_c(t)} \longrightarrow \oplus \longrightarrow \boxed{g_{Rx}(t)}$$

$$\uparrow$$
$$n(t)$$

Channel impulse response is not flat

$\Rightarrow \quad g_{Tx} * g_c * g_{Rx}'$ is not Nyquist.

- Ideally, $g_c' = \delta[k]$

eg Consider BPSK :-

$$g_c \equiv \delta[k] + 0.3\,\delta[k-1]$$

$$y(t) = \sum_k \Big(b[k]\, g_{Tx}(t - kT) \Big) * g_c(t) + n(t)$$

$$y[k] = b[k] + 0.3\,b[k-1] + n[k] \quad \ldots\ldots \text{ ISI.}$$

~~b[ ] =~~	$b[-1]$	$b[0]$	$b[1]$	$b[2]$	$b[3]$
	0	1	-1	1	1
$y[k]$ (ignoring noise)		1	-0.7	0.7	1.3

26/3

$$p(t) = g_{Tx}(t) * g_c(t)$$

$$s_b(t) = \sum b[k]\, p(t - kT)$$

ML : Maximize $\quad \mathrm{Re}(\langle y, s_b \rangle) - \dfrac{\|s_b\|^2}{2}$

$$y(t) = s_b(t) + n(t)$$

$$z[n] = (y * p_{MF})(nT)$$

$$p_{MF}(t) = p^*(-t)$$

$$\langle y, s_b \rangle = \langle y, \sum b[n]\, p(t-nT) \rangle$$

$$= \sum b^*[n] \langle y, p(t-nT) \rangle$$

$$= \sum b^*[n]\, z[n].$$

$\therefore$ ~~Minimizing~~

Maximizing that shit is equivalent to taking inner product of $z[n]$ with $b[n]$.

$$\rightarrow \|s_b\|^2 = \left\langle \sum_k b[k]\, p(t-kT), \sum_l b[l]\, p(t-lT) \right\rangle$$

$$= \sum_k \sum_l b[k]\, b^*[l] \underbrace{\langle p(t-kT), p(t-lT) \rangle}_{= h[l-k]}$$

where $h[m] = \int_{-\infty}^{\infty} p(t)\, p^*(t-mT)\, dt$

Note :- $h^*[-m] = \int_{-\infty}^{\infty} p^*(t)\, p(t+mT)\, dt = h[m]$

$$= \sum_k \sum_l b[k]\, b^*[l]\, h[l-k]$$

$$= \sum_m \sum_n b[m]\, b^*[n]\, h[m-n]$$

$$= h[0] \sum_m \|b[m]\|^2 + \sum_m \sum_{n<m} b[m]\, b^*[n]\, h[n-m]$$

$$+ \sum_m \sum_{n>m} b[m]\, b^*[n]\, h[n-m]$$

Page No.:
Date:

$m \leftrightarrow n$ in 3rd term

$$= h[0] \sum_m |b[m]|^2 + \sum_n \sum_{n<m} \Big(b[n]\, b^*[m]\, h[m-n] + b[m]\, b^*[n]\, \underbrace{h[n-m]}_{= h^*[m-n]} \Big)$$

$$= h[0] \sum_m |b[m]|^2 + \sum_n \sum_{n<m} 2\, \mathrm{Re}\big(b[n]\, b^*[m]\, h[m-n] \big)$$

→ Metric = $\Lambda(b) = \|y - s_b\|^2$

$$= \sum_n \Big\{ \mathrm{Re}\big(b^*[n]\, z[n]\big) - \frac{h[0]}{2} |b[n]|^2 - \mathrm{Re}\Big(b^*[n] \sum_{m<n} b[m]\, h[n-m] \Big) \Big\}.$$

$m = n - L \rightarrow n-1$

28/3

$\Lambda(b)$ can be updated in real time by simply adding the next term in summation

$$s[n] = \{ b[n-1] \ldots\ldots b[n-L] \}$$

Increment :- ~~from n-1 → n~~ after receiving $b[n]$ $\quad \lambda_n(b[n], s[n])$

$$\boxed{= \mathrm{Re}\{ b^*[n]\, z[n] \} - \frac{h[0]}{2} |b[n]|^2 - \mathrm{Re}\Big(b^*[n] \sum_{m=n-L}^{n-1} b[m]\, h[n-m] \Big)}$$

constant & useless for PSK

eg ~~$g_c = \delta[n] - \frac{1}{2}\delta[n-1]$~~

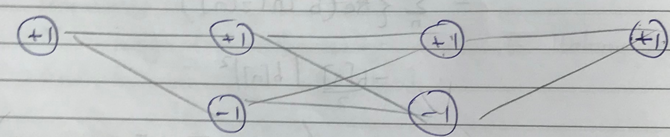
$b[n] \in \{+1, -1\}$

$h[0] = \frac{3}{2}$, $h[1] = h[-1] = \frac{-1}{2}$

Receiver assumes $b[0] = +1$ by default for reference.

$b[1]$ can be $+1$ or -1.

predecoded $b[0]$ $\quad$ $b[1]$ $\quad$ $b[2]$ $\quad$ predecode $b[3]$



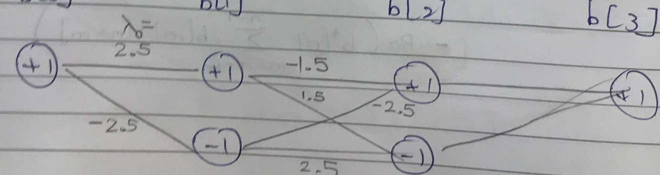
~~Consider~~ Here $L = 1$ (because $h[n] = 0$ for $n \geq 1$)

→ $\lambda_n(b[n], s[n]) = b[n]\left(Re(z[n]) + \frac{1}{2} b[n-1]\right)$

→ Consider $Re(z[n]) \rightarrow$

$Re(z[n])$	-1	2	-2	1.5
	0	1	2	3

$b[0]$ $\quad$ $b[1]$ $\quad$ $b[2]$ $\quad$ $b[3]$



1, 1, 1 → 1 ✓
1, −1, 1 → −5 ✗ (reject)

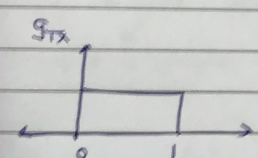
1, 1, −1 → 4 ✓
1, −1, −1 → 0 ✗ (reject)

} Only 2 paths will survive.

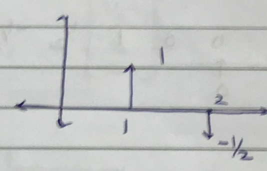
- If constellation has M symbols → M^L paths will survive
- Viterbi is influenced a lot by channel properties

---

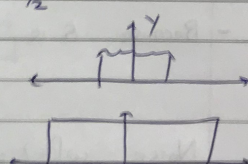
eg



and



$y(t) = \Sigma\, b[n]\, p(t - nT)$



Convolve y with $r[k] = (y * g_{Rx})(kT_s + \delta)$

$T_s \le T$
'Up-sampling'

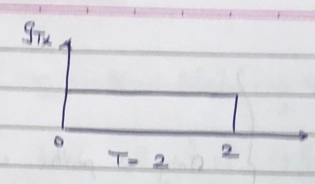
Because time frames won't match.

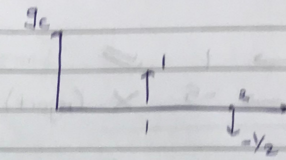
This convolution is after addition of noise to signal.

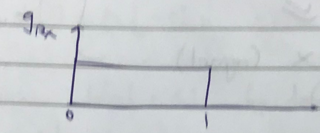
Define $f[k] = (p * g_{Rx})(kT_s + \delta)$

P.T.O.

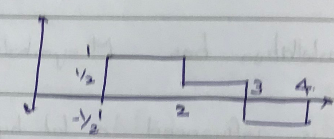
eg







$p(t) = (g_{Tx} * g_c)(t)$



- $(p * g_{Rx})(kT_s)$ .... $\delta = 0$, $T_s = 1$

0 0 1 1/2 -1/2
↑
0

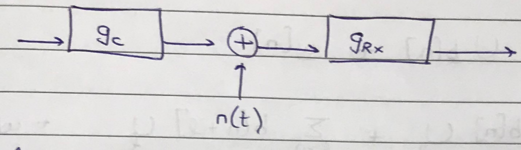
- Because δ is generally non-zero, $\Rightarrow$ ISI

Noise $w[k] = (n(t) * g_{Rx})(kT_s + \delta)$

$$C_w(\tau) = 2\sigma^2 \int g_{Rx}(t)\, g_{Rx}^*(t-\tau)\, dt$$

$$C_w[k] = 2\sigma^2 \int g_{Rx}(t)\, g_{Rx}^*(t-kT_s)\, dt$$

(Autocovariance = Autocorrelation)



$$y(t) = \sum_k b[n]p(t-nT) + n(t)$$

$$r[k] = \int_{k-1}^{k} y(t)\,dt$$

$\therefore$ $\omega[k]$ is iid, white with variance $2\sigma^2$

☺

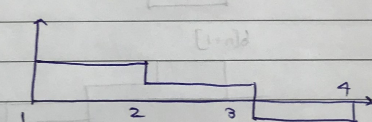
If g_{Rx} were different, $\omega[k]$ would not have been iid or white

Now, $\underline{r}[n] = b[n-1]\begin{bmatrix} 1/2 \\ -1/2 \\ 0 \\ 0 \\ 0 \end{bmatrix} + b[n]\begin{bmatrix} 0 \\ 1 \\ 1/2 \\ -1/2 \\ 0 \end{bmatrix} + b[n+1]\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1/2 \end{bmatrix} + \underline{\omega}_n$

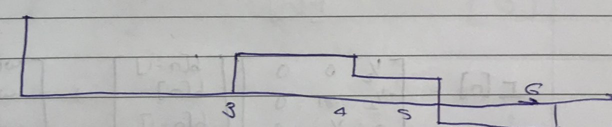
Group of 5 samples

b[0]

$b[0]p(t-0T)$



$b[1]p(t-1T)$



ie $\underline{r}[n] = \underbrace{\begin{bmatrix} 1/2 & 0 & 0 \\ -1/2 & 1 & 0 \\ 0 & 1/2 & 0 \\ 0 & -1/2 & 1 \\ 0 & 0 & 1/2 \end{bmatrix}}_{U_{5\times 3}} \underline{b}[n] + \underline{\omega}[n]$

This is an overdetermined system
(5 eq^n 3 variable)

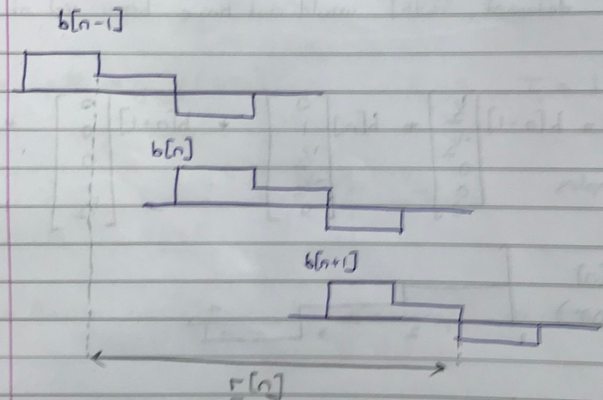
$$\underline{r}[n] = U\,\underline{b}[n] + \underline{w}[n]$$

$$= \underbrace{b[n]\,U_0}_{0^{th}\text{ column}} + \sum_{l \neq 0} b[n+l]\,U_l + w[n]$$

Linear equalizer $\underline{c}$

Use $z[n] = C^H \underline{r}[n]$

2/4



$$\underline{r}[n] = \underbrace{\begin{bmatrix} 1/2 & 0 & 0 \\ -1/2 & 1 & 0 \\ 0 & 1/2 & 0 \\ 0 & -1/2 & 1 \\ 0 & 0 & 1/2 \end{bmatrix}}_{U} \begin{bmatrix} b[n-1] \\ b[n] \\ b[n+1] \end{bmatrix} + \begin{bmatrix} w[n-1] \\ w[n] \\ w[n+1] \end{bmatrix}$$

If U were square, we could directly invert it

Else we will find pseudo-inverse

- Zero forcing equalizer :-

Make everything except $b[n]$ zero.

Use $Z[n] = \underline{C}^H \underline{r}[n]$ $\quad$ C is 5×1

As before, $\underline{r}[n] = b[n]\, \underline{u}_0[n] + \left(\sum_{l\neq 0} b[n+l]\, \underline{u}_l\right) + \omega[n]$

$\rightarrow Z[n] = \underline{C}^H \underline{r}[n]$

$= b[n]\, \underline{C}^H \underline{u}_0 + \underline{C}^H \sum b[n+l]\, u_l + \underline{C}^H \omega[n]$

Note :- ~~C^H~~ Find a C that makes $C^H u_0 = 1$

and $C^H u_i = 0 \quad i \neq 0$

If there are many options, choose the lowest because it is getting multiplied to ω (noise) :(

$$\text{If} \quad U = \begin{array}{c} \begin{array}{ccc} u_{-1} & u_0 & u_1 \end{array} \\ \begin{bmatrix} 1/2 & 0 & 0 \\ -1/2 & 1 & 0 \\ 0 & 1/2 & 0 \\ 0 & -1/2 & 1 \\ 0 & 0 & 1/2 \end{bmatrix} \end{array}, \quad \text{choose } C^H U = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}^T$$

$= e^T$

'One-hot vector'.

$\therefore$ ~~$CU^H = e$~~ $\therefore U^H C = e$

'C is a linear combination of U_i'

Proof ahead.

We want to minimize $\|C\|^2$ subject to above conditions (to reduce noise)

P.T.O.

Say $\quad c = U\underline{g}$ $\quad$ ... linear combination of U's

$$U^H c = U^H U \underline{g} = \underline{e}$$

$$\underline{g} = (U^H U)^{-1} \underline{e}$$

$$\underline{c} = \underbrace{\boxed{U(U^H U)^{-1}}}_{\text{(Left) "Pseudo inverse"}} \underline{e}$$ ... Zero forcing C

- Only linear combination of U's minimizes $\|C\|^2$

Finite Impulse Response Filter

- Impulse response is of finite duration ⇒ Is absolutely summable
  ⇒ Coefficient perturbation cannot cause instability.

- Can give linear phase response :-

$$\underline{H_{FIR}(e^{j\omega}) = |H_{FIR}(e^{j\omega})| \, e^{-j\omega D}}$$

<u>D is an integer or half-integer</u>
(Any other D would make filter unrealizable)

We also generally accept

$$H_{FIR}(e^{j\omega}) = e^{-j\omega D} H_R(e^{j\omega})$$

where $H_R(e^{j\omega})$ is any real function.

- Different from above expression because H_R can be negative and hence produce a phase shift of π.

* Pseudo linear H_{FIR} can be acceptable.
  - Group delay is constant
    Phase not.

- $H_R(e^{j\omega}) = e^{j\omega D} H_{FIR}(e^{j\omega})$ ...... ~ Shifting h_{FIR} in time by D.

What if D is a half integer?

If D is an integer, we are shifting h_{FIR} by D samples.

- Since $h_R[n]$ is real, $H_R(e^{-j\omega}) = H_R^*(e^{j\omega})$
  $= H_R(e^{j\omega})$

∴ <u>H_R is even-symmetric.</u>

* <u>Real and even sequences have real and even DTFTs.</u>

- Consider $H_{FIR}(e^{j\omega}) = j\, e^{-j\omega D} H_R(e^{j\omega})$

  ☆ $\therefore\ e^{j\omega D} H_{FIR}(e^{j\omega}) = \underbrace{j H_R(e^{j\omega})}_{\text{real}}$

☆ $\therefore\ j h_R[n]$ is real

$\therefore\ j H_R(e^{-j\omega}) = \left(j H_R(e^{j\omega})\right)^*$

☆ $= -j\, H_R(e^{j\omega})$

$\therefore\ H_R(e^{-j\omega}) = -H_R(e^{j\omega})$

H_R is antisymmetric.

* Real and antisymmetric sequences have pure ~~and~~ imaginary and odd DTFTs

eg $\quad 1 \quad \underset{0}{\overset{0}{\uparrow}} \quad 1 \qquad$ DTFT = $2\cos\omega$.

$\quad 1 \quad \underset{0}{\overset{0}{\uparrow}} \quad -1. \qquad$ DTFT = $2j\sin\omega$.

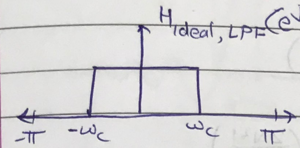
Both are accepted as pseudo linear phase responses.

Theorem: Pseudo linear phase requires symmetry or anti-symmetry in impulse responses

- IIR filters can never have symmetric/anti symmetric response.

Page No.:
Date:

I Lowpass Filter



$$h_{ideal}[n] = \begin{cases} \dfrac{\omega_c}{\pi} & \text{for } n = 0 \\ \dfrac{\sin \omega_c n}{\pi n} & \text{for } n \neq 0 \end{cases}$$

$$h[-n] = h[n]$$

- Infinitely, non-causal, unstable, irrational

- Analog filters can never give us ~~a~~ finite length impulse response
  - Cannot have linear phase
  - No transformations possible to design FIR filters.

- We approximate ideal LPF by truncating $h_{ideal}[n]$ to obtain finite length impulse response.
  - For linear phase response, we maintain symmetry in $h_{FIR}[n]$

eg – Choose $n \in \{-10, -9 \, ... \, 10\}$ to make $h_{FIR}[n]$ of ~~a~~ length 21.

15/3 – Truncation ~ Multiplication of $h[n]$ by a window sequence '$v[n]$'

$$= \begin{cases} 1 & \text{for } -N \leq n \leq N \\ 0 & \text{otherwise} \end{cases}$$

'Rectangular window sequence'.

$$h_{FIR}[n] = h_{ideal}[n] \, v_R[n]$$

$\hookrightarrow$ R for real

$$\therefore \; H_{FIR}(e^{j\omega}) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_{ideal}(e^{j\lambda}) \, V_R(e^{j(\omega-\lambda)}) \, d\lambda$$

where $V_R(e^{j\omega}) =$ DTFT of $v_R[n]$

$$= \sum_{n=-N}^{N} v_R[n] \, e^{-j\omega n}$$

P.T.O.

$$= \sum_{-N}^{N} e^{-j\omega n}$$

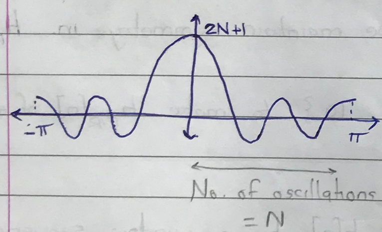
$$= \frac{e^{+j\omega N}\left(\left(e^{-j\omega}\right)^{2N+1} - 1\right)}{e^{-j\omega} - 1}$$

$$= \frac{e^{j\omega N}\, e^{-j\omega\left(\frac{2N+1}{2}\right)}\, 2j \sin\omega\left(\frac{2N+1}{2}\right)}{e^{-j\omega/2}\, 2j \sin\frac{\omega}{2}}$$

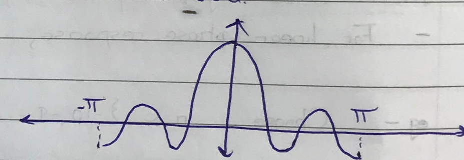
$$= e^{j\omega\left(N - N + \frac{1}{2} - \frac{1}{2}\right)} \frac{\sin\omega\left(\frac{2N+1}{2}\right)}{\sin\left(\frac{\omega}{2}\right)}$$

$$\boxed{V_R = \frac{\sin\omega\left(N + \frac{1}{2}\right)}{\sin\left(\frac{\omega}{2}\right)}}$$ .... even, because $V_R[n]$ was real and even

For N = even

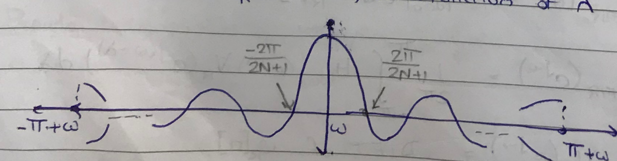


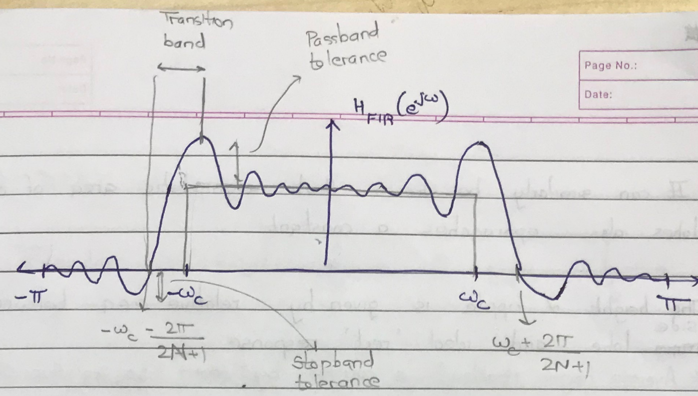
For N = odd



$$\therefore \quad H_{FIR}(e^{j\omega}) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_{ideal}(e^{j\lambda})\, V_R(e^{j(\omega-\lambda)})\, d\lambda$$

$V_R(e^{j(\omega-\lambda)})$ as a function of λ





- At $\omega = \pm\omega_c$, $H_{FIR}(e^{j\omega}) \approx \frac{1}{2}$ of maximum value
- $H_{FIR}(e^{j\omega})$ is maximum at $\pm\left(\omega_c - \frac{2\pi}{2N+1}\right)$
- Passband and stopband ~~transition bands~~ tolerances are caused by side-lobes of $V_R(e^{j\omega})$
- The transition width $= \frac{4\pi}{2N+1}$
- Asymptotic $(N \to \infty)$ area under the primary side-lobe of $V_R(e^{j\omega})$

$$= \int_{\frac{2\pi}{2N+1}}^{\frac{4\pi}{2N+1}} \frac{\sin \omega\left(\frac{2N+1}{2}\right)}{\sin \frac{\omega}{2}} \, d\omega$$

$\sin \frac{\omega}{2} \to \frac{\omega}{2}$

$$\approx \frac{2}{2N+1} \int_{\pi}^{2\pi} \frac{\sin \theta}{\frac{\theta}{2N+1}} \, d\theta$$

$$\approx 2 \int_{\pi}^{2\pi} \frac{\sin\theta}{\theta} \, d\theta$$

---- independent of N

Gibb's phenomenon :- Irrespective of value of N, there is always an asymptotic deviation near the point of discontinuity even though the transition band width decreases for higher N.

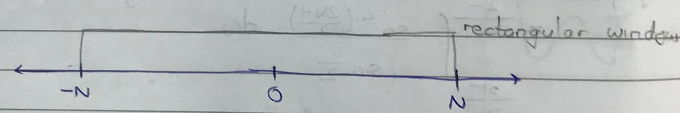
- ∴ We have no control over passband and stopband tolerance.

18/3

* It can similarly be shown that asymptotic area of other side lobes also approaches a constant.

- The height of ripple is given by relative area between side ~~main~~ lobe and ideal 'rect' response.
* Average height reached is caused by main lobe relative area.
- The stopband and passband tolerances are equal. (see figure) as they are both caused by side ~~main~~ lobe area.
  - They also do not depend on N for large enough N
  - If we want to control passband and stopband tolerances we will have to change the window sequence used.

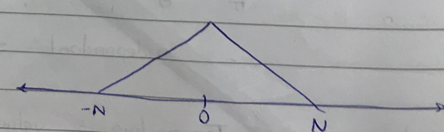
→ Other window sequences



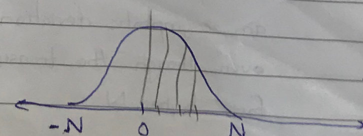
- The passband and stopband tolerances in rectangular window case were caused because of discontinuity in the rectangular window.

∴ - To reduce tolerances, we try to make the window smoother

eg Triangular

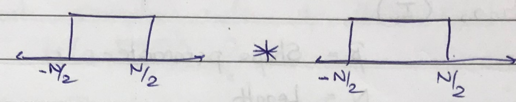


Better eg Raised Cosine (Hann window)



$$\frac{1}{2}\left(1+\cos\left(\frac{t}{N}\pi\right)\right)$$

Q Find DTFT of above window sequences

① Triangular = 

② Raised cosine :- Write cos as exponentials

$$\text{Generalized raised cosine} = \alpha + (1-\alpha)\cos\left(\frac{t}{N}\pi\right)$$

for $0.5 < \alpha < 1$

'Generalized Hamming Window'

- Smoothness of window is characterized by continuity of derivative(s). ~~Smoother is better~~

* Blackman window

$$\alpha + \beta\cos\left(\frac{t}{N}\pi\right) + \gamma\cos\left(\frac{2t}{N}\pi\right)$$

$\alpha, \beta, \gamma > 0$ and $\alpha + \beta + \gamma = 1$.

☹ • For the same window size 'N', there is a conflict between $\underbrace{\text{transition width}}_{\text{characterized by mainlobe width}}$ and $\underbrace{\text{passband/stopband tolerance}}_{\text{characterized by relative sidelobe area}}$.

- These window types have constant tolerances that cannot be tuned. ☹

P.T.O.

→ Kaiser proposed a window based on modified Bessel functions (I_0)

β = Shape parameter
N = Length

Window = $I_0(\beta N)$

- Kaiser window, through β and N, allows for changes in main lobe width and relative side lobe area

$\beta = 0$ ⇒ Rectangular Window
(High side lobe area :(
Low main lobe area :))

- Choose shape parameter based on tolerances required
Then choose length to satisfy transition width

→ Park McClellan algorithm   Handout
?

I ESTIMATING DTFT

Substitute $z = e^{j\omega}$ in $H(z)$ on a fine uniform grid of ω

$$\omega = \frac{2\pi k}{N_0}$$

$$k = 0 \ldots N_0 - 1$$

How to choose N_0 ? - Duality.

$$h[n] \xrightarrow{DTFT} H(e^{j\omega})$$

What is a sufficient sampling rate so that $h[n]$ can be reconstructed from $H(e^{j\omega_k n})$?

A sufficient sampling rate exists if $h[n]$ is time-limited (Length of $h[n]$ = M)

$$H(e^{j\omega_k}) = \sum_{n=0}^{M-1} h[n] e^{-j\frac{2\pi k}{N_0} n}$$

$$= \langle h[n], e^{j\frac{2\pi k}{N_0} n} \rangle$$

..... both sequences of length M $(n = 0 \ldots M-1)$

- Consider $M \leq N_0$

Consider $e^{j\frac{2\pi k}{N_0} n}$ for $k = 0 \ldots \cancel{M} N_0 - 1$

Since $M \leq N_0$, we can pad $h[n]$ with zeros in the end

$$\langle e^{j\frac{2\pi k_1}{N_0} n}, e^{j\frac{2\pi k_2}{N_0} n} \rangle$$

$$= \sum_{n=0}^{N_0-1} (\quad)(\quad)^* \quad \ldots \text{GP}$$

$$= \begin{cases} 0 & \text{if } k_1 \neq k_2 \\ \cancel{\boxed{\dfrac{1 - e^{j\frac{2\pi}{N_0}(k_1-k_2)N_0}}{1 - e^{j\frac{2\pi}{N_0}(k_1-k_2)}}}} \; N_0 & \text{if } k_1 = k_2 \end{cases}$$

'Orthogonal'.

- Define unit vector in k^{th} dimension as $\dfrac{1}{\sqrt{N_0}} e^{j\frac{2\pi}{N_0} kn}$

$$H_{new} \rightarrow \frac{1}{\sqrt{N_0}} H_{old}$$

Reconstructed $h[n] = \sum_{k=0}^{N_0-1} \left(\frac{1}{\sqrt{N_0}} H(e^{j\omega_k}) \right) \left(\frac{1}{\sqrt{N_0}} e^{j\omega_k n} \right)$

N_0-point Inverse DFT $\rightarrow$ $$\boxed{h[n] = \frac{1}{N_0} \sum_{k=0}^{N_0-1} H(e^{j\omega_k}) e^{j\frac{2\pi}{N_0}kn}}$$

As before $$\boxed{H(e^{j\omega_k}) = \sum_{n=0}^{N_0-1} h[n] e^{-j\frac{2\pi}{N_0}kn}}$$

'N_0-point "Discrete Fourier Transform" (DFT)' of finite length sequence $h[n]$.

- If $\text{len}(h[n]) > N_0$:- N_0-point DFT is invalid
  $<$:- Pad $h[n]$ with zeros.
- Any $N_0 \geq \text{len}(h[n])$ can give perfect reconstruction

→ Convolution :-

Let $h[n]$ & $g[n]$ be, WLOG, of equal length M.

Take inverse N_0-point DFT of $\underbrace{H(e^{j\omega_k})}\underbrace{G(e^{j\omega_k})}$

H & G are both N_0-point DFTs.

$$\frac{1}{N_0} \sum_{k=0}^{N_0-1} H(e^{j\omega_k}) G(e^{j\omega_k}) e^{j\omega_k n}$$

- $(h * g)[n]$ will be of length $2M-1$.
  $\therefore$ We need $N_0 \geq 2M-1$

$$* \quad h[n] = \frac{1}{N_0} \sum_{k=0}^{N_0-1} H(e^{j\omega_k}) e^{j\frac{2\pi}{N_0}kn}$$

Thus, the inverse N_0-point DFT is periodic with period N_0

$\therefore$ ~~Fourier~~

$$\frac{1}{N_0} \sum_{k=0}^{N_0-1} H(e^{j\omega_k}) e^{j\frac{2\pi}{N_0}kn} = \sum_{r=-\infty}^{\infty} h[n + rN_0]$$

CP Prove this ab initio

- $\therefore$ Inverse N_0-point DFT gives periodic extension of original sequence.

If $N_0 < M$, inverse N_0-point DFT results in time-domain aliasing.

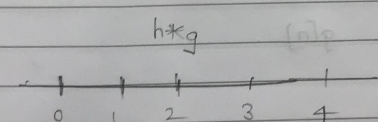
Q → Consider $h[n]$ and $g[n]$ with support $n = 0, 1, 2$

Take $N_0 = 3 < 2(3) - 1$

Find $q[n]$ = 3-point inverse DFT of $H(e^{j\frac{2\pi}{3}}) G(e^{j\frac{2\pi}{3}})$

$$H\left(e^{j\frac{2\pi}{3}k}\right) = \sum_{n=0}^{2} h[n] e^{-j\frac{2\pi k}{N_0} n}$$

$$G\left(e^{j\frac{2\pi k}{3}}\right) = \sum_{n=0}^{2} g[n] e^{-j\frac{2\pi}{3}k}$$



$q[0] = h*g[0] + h*g[3]$

$q[1] = h*g[1] + h*g[4]$

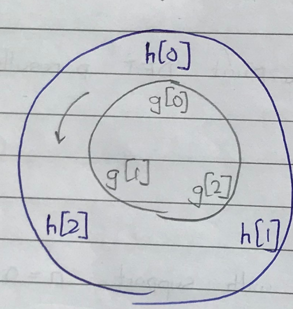
$q[2] = h*g[2]$

h[0] h[1] h[2]   g[2] g[1] g[0]

$$h*g[n] = \sum_{k=-\infty}^{\infty} h[k]\, g[n-k]$$

$h*g[0] = h[0]g[0]$
$h*g[1] = h[0]g[1] + h[1]g[0]$
$h*g[2] = h[0]g[2] + h[1]g[1] + h[2]g[0]$
$h*g[3] = h[1]g[2] + h[2]g[1]$
$h*g[4] = h[2]g[2]$

$\therefore$ $q[0] = h[0]g[0] + h[1]g[2] + h[2]g[1]$
$q[1] = h[0]g[1] + h[1]g[0] + h[2]g[2]$
$q[2] = h[0]g[2] + h[1]g[1] + h[2]g[0]$



'Circular Convolution' / 'Periodic Convolution'

Can be considered as an LTI system
Input = N_0-periodic extension of $h[n]$
~~Output =~~
Impulse response = $g[n]$

$$\sum_{r=-\infty}^{\infty} h[n+rN_0] \longrightarrow \boxed{g[n]} \longrightarrow q[n]$$

* No. of multiplications required for linear convolution of an L-length sequence with an M-length sequence $= LN$

(Every point of every sequence is multiplied by every sequence of the other sequence once)

28/3

Fast Fourier Transforms

- Efficient algorithms to compute DFT.

Notation:- DFT → $X(e^{j\omega_k}) \equiv X[k]$

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

- Consider a composite number $N = N_1 N_2$

To do:- Write N point DFT as combination of N_2 N_1 point DFTs and N_1 N_2 point DFTs.

- Simplest class of composite numbers : $N = 2^P$

- Decimation of $x[n]$ in time:-

$\left.\begin{array}{l} x[2n] \\ x[2n+1] \end{array}\right\}$ for $n = 0 \ldots\ldots \frac{N}{2} - 1$

$$\therefore X[k] = \sum_{n=0}^{\frac{N}{2}-1} x[2n]\, e^{-j\frac{2\pi}{N}2nk} \rightarrow \sum_{n=0}^{\frac{N}{2}-1} x[2n]\, e^{-j\frac{2\pi}{N/2}nk}$$

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

$$e^{-j\frac{2\pi}{N}k} \sum_{n=0}^{\frac{N}{2}-1} x[2n+1]\, e^{-j\frac{2\pi}{N/2}nk}$$

This is like $\frac{N}{2}$ point DFT, but k still goes from 0 to $N-1$

* $X[k+N] = X[k]$

... just as DTFT is periodic with period 2π.

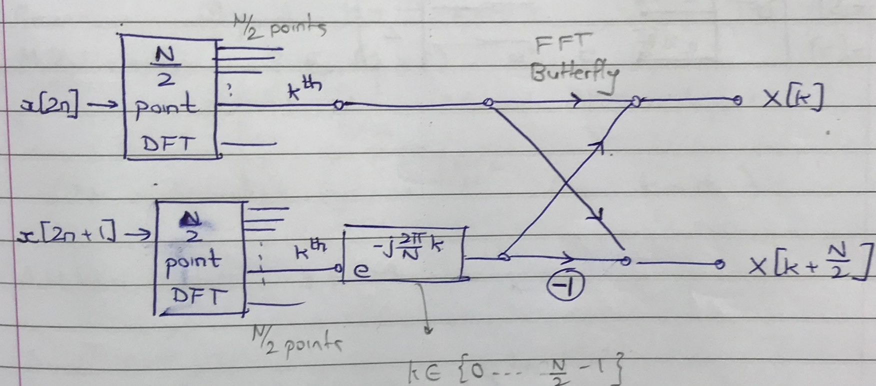
- First term and second term are both periodic with period $\frac{N}{2}$

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

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

$$= \sum_{n=0}^{N/2-1} x[2n]\, e^{-j\frac{2\pi}{N/2} nk}$$

$$- e^{-j\frac{2\pi}{N}k} \sum_{n=0}^{N/2-1} x[2n+1]\, e^{-j\frac{2\pi}{N/2} nk}$$

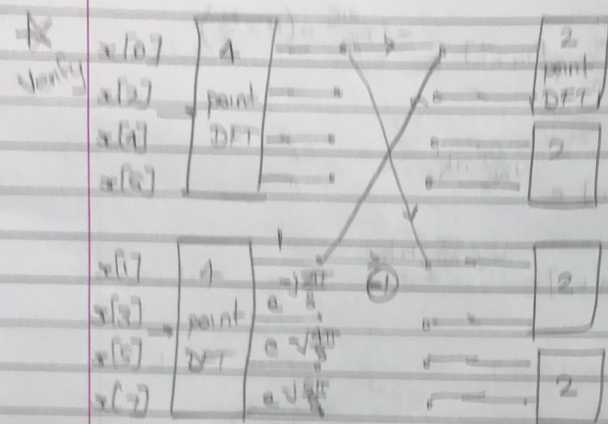


Further $\frac{N}{2}$ can be broken into $\frac{N}{4}$ and $\frac{N}{4}$ and so on

till we need 1-point DFT (which is the point itself)

~~eg 8-point DFT~~

Q) 8-point DFT


verify
x[0]
x[2]
x[4]
x[6]
4 point DFT
2 point DFT
2
x[1]
x[3]
x[5]
x[7]
4 point DFT
2
2

Page No.:
Date:

→ Precise computational complexity expansion

eg 8-point FFT (Decimation in time) (DIT)

- Broad comparison –

	N-point DFT	DIT FFT
# of multiplication	$O(N^2)$	$O(\cancel{N\log_2 N}\ N\log_2 N)$
# of addition	$O(N^2)$	

- Precise :- 8 point DIT FFT

Multiplications :- $\underbrace{3}_{\text{stage 1}} + (\underbrace{2}_{\text{two DFTs in stage 2}} \times \underbrace{1}_{\text{for each DFT in stage 2}}) = 5$

(No multiplication by $e^{j\frac{2\pi}{N}k}$ for $k=0$)

Additions :- $3\times 8 = 24$

- Direct DFT :-

$$X[k] = \sum_{n=0}^{7} x[n]e^{-j\frac{2\pi}{8}kn}$$

k \ n	0	1	2	3	4	5	6	7
0								
1		✓	✓	✓				
2		✓	×	✓			×	
3		✓	✓	✓				
4								
5								
6			×				×	
7								

Multiplications
$8\times 4 = 32$

Additions :- $8\times 7 = 56$

B] Decimation in Frequency (DIF)

$N = 2^P$

Consider $X[2k]$ and $X[2k+1]$

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

Partition in time:

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

$$= \sum_{n=0}^{\frac{N}{2}-1} x\left[n+\frac{N}{2}\right] \underbrace{e^{-j\frac{2\pi}{N}\left(n+\frac{N}{2}\right)(2k)}}_{(1)\left(e^{-j\frac{2\pi}{N/2}nk}\right)}$$

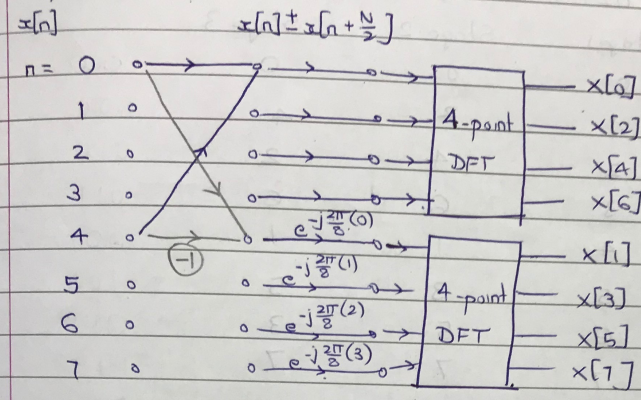
$$\therefore \; X[2k] = \sum_{n=0}^{\frac{N}{2}-1} \left(x[n] + x\left[n+\frac{N}{2}\right]\right) e^{-j\frac{2\pi}{N/2}nk}$$

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

$$\sum_{n=0}^{\frac{N}{2}-1} x\left[n+\frac{N}{2}\right] \underbrace{e^{-j\frac{2\pi}{N}\left(n+\frac{N}{2}\right)(2k+1)}}_{(-1)\, e^{-j\frac{2\pi}{N}n(2k+1)}}$$

$$X[2k+1] = \sum_{n=0}^{\frac{N}{2}-1} \left(x[n] - x\left[n+\frac{N}{2}\right]\right) \underbrace{e^{-j\frac{2\pi}{N}n(2k+1)}}_{e^{-j\frac{2\pi}{N/2}nk} \; e^{-j\frac{2\pi}{N}n}}$$

eg 8-point



- DIT SFG $\xrightarrow{\text{Transposition}}$ DIF SFG

  Transposition → Reversing all arrows

- Computational complexity is exactly same for DIT and DIF.

– Complete 8-point DIF FFT.

→ Reordering :- (DIF or DIT)

	Stage 1	Stage 2	Stage 3	
000	0	0	0	000
001	1	2	4	100
010	2	4	2	010
011	3	6	6	011
100	4	1	1	001
101	5	3	5	101
110	6	5	3	011
111	7	7	7	111

Bit reversal

2/4

I FFT FOR GENERAL COMPOSITE N

$N = LM$

N-point DFT:- $X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}nk}$

Let $n = Mn_1 + n_2$

$n_1 \in \{0 \dots L-1\}$, $n_2 \in \{0 \dots M-1\}$

and Let $k = Lk_1 + k_2$

$k_1 \in \{0 \dots M-1\}$, $k_2 \in \{0 \dots L-1\}$

Page No.:
Date:

$$X[k_1, k_2] = \sum_{n_1=0}^{L-1} \sum_{n_2=0}^{M-1} x[n_1, n_2] e^{-j\frac{2\pi}{N} nk}$$

$$e^{-j\frac{2\pi}{N} nk} = e^{-j\frac{2\pi}{LM}(Mn_1 + n_2)(Lk_1 + k_2)}$$

$$= \underbrace{e^{-j 2\pi n_1 k_1}}_{=1} e^{-j\frac{2\pi}{LM} n_1 k_2} e^{-j\frac{2\pi}{M} n_2 k_1} e^{-j\frac{2\pi}{LM} n_2 k_2}$$

$$= \sum_{n_2=0}^{M-1} e^{-j\frac{2\pi}{M} n_2 k_1} e^{-j\frac{2\pi}{LM} n_2 k_2} \left(\sum_{n_1=0}^{L-1} x[n_1, n_2] e^{-j\frac{2\pi}{L} n_1 k_2} \right)$$

For each n_2, this is an L-point DFT with index $k_2 \in \{0 \dots L-1\}$

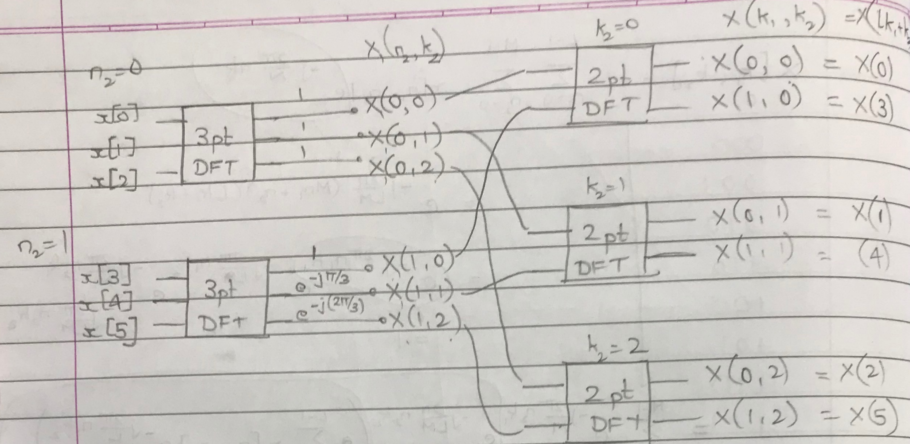
$$\triangleq X_1[n_2, k_2]$$

$$= \underbrace{\sum_{n_2=0}^{M-1} e^{-j\frac{2\pi}{M} n_2 k_1} X_1[n_2, k_2]}$$

M-point DFT with index $k_1 \in \{0 \dots M-1\}$

eg. $L = 3$, $M = 2$, $N = 6$

P.T.O.



II INVERSE ~~Direct~~ DFT

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

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

$\therefore$ Use the same DFT block, just complex conjugate all multipliers and finally divide by N.

- This works for any FFT algorithm too.

- $\therefore$ IDFT has same computational complexity as that of DFT.

→ Convolution of two 8-point sequences

Direct convolution would have required many multiplications (64)

- Better!

~~Pad~~ Answer will be of length 15.

Pad 8 zeros to both sequences

Take 16-point DFTs of both sequences.

Multiply 16-point DFTs point-by-point.

Take 16-point IDFT.