Computer-Aided Design of Digital Filters

- The FIR filter design techniques discussed so far can be easily implemented on a computer
- In addition, there are a number of FIR filter design algorithms that rely on some type of optimization techniques that are used to minimize the error between the desired frequency response and that of the computer-generated filter

1

Computer-Aided Design of Digital Filters

- Basic idea behind the computer-based iterative technique
- Let $H(e^{j\omega})$ denote the frequency response of the digital filter $H(z)$ to be designed approximating the desired frequency response $D(e^{j\omega})$, given as a piecewise linear function of ω, in some sense

2

Computer-Aided Design of Digital Filters

- Objective - Determine iteratively the coefficients of $H(z)$ so that the difference between between $H(e^{j\omega})$ and $D(e^{j\omega})$ over closed subintervals of $0 \le \omega \le \pi$ is minimized
- This difference usually specified as a weighted error function

$$\mathcal{E}(\omega) = W(e^{j\omega})[H(e^{j\omega}) - D(e^{j\omega})]$$

where $W(e^{j\omega})$ is some user-specified weighting function

3

Computer-Aided Design of Digital Filters

- **Chebyshev** or **minimax criterion** - Minimizes the peak absolute value of the weighted error:

$$\varepsilon = \max_{\omega \in R} |\mathcal{E}(\omega)|$$

where R is the set of disjoint frequency bands in the range $0 \le \omega \le \pi$, on which $D(e^{j\omega})$ is defined

4

Design of Equiripple Linear-Phase FIR Filters

- The linear-phase FIR filter obtained by minimizing the peak absolute value of

$$\varepsilon = \max_{\omega \in R} |\mathcal{E}(\omega)|$$

is usually called the **equiripple FIR filter**

- After ε is minimized, the weighted error function $\mathcal{E}(\omega)$ exhibits an equiripple behavior in the frequency range R

5

Design of Equiripple Linear-Phase FIR Filters

- The general form of frequency response of a causal linear-phase FIR filter of length $2M+1$:

$$H(e^{j\omega}) = e^{-jM\omega} e^{j\beta} \breve{H}(\omega)$$

where the amplitude response $\breve{H}(\omega)$ is a real function of ω

- Weighted error function is given by

$$\mathcal{E}(\omega) = W(\omega)[\breve{H}(\omega) - D(\omega)]$$

where $D(\omega)$ is the desired amplitude response and $W(\omega)$ is a positive weighting function

6

Design of Equiripple Linear-Phase FIR Filters

- **Parks-McClellan Algorithm** - Based on iteratively adjusting the coefficients of $\breve{H}(\omega)$ until the peak absolute value of $\mathcal{E}(\omega)$ is minimized
- If peak absolute value of $\mathcal{E}(\omega)$ in a band $\omega_a \le \omega \le \omega_b$ is ε_o, then the absolute error satisfies

$$\left|\breve{H}(\omega) - D(\omega)\right| \le \frac{\varepsilon_o}{|W(\omega)|}, \quad \omega_a \le \omega \le \omega_b$$

7

Design of Equiripple Linear-Phase FIR Filters

- For filter design,

$$D(\omega) = \begin{cases} 1, & \text{in the passband} \\ 0, & \text{in the stopband} \end{cases}$$

- $\breve{H}(\omega)$ is required to satisfy the above desired response with a ripple of $\pm\delta_p$ in the passband and a ripple of δ_s in the stopband

8

Design of Equiripple Linear-Phase FIR Filters

- Thus, weighting function can be chosen either as

$$W(\omega) = \begin{cases} 1, & \text{in the passband} \\ \delta_p/\delta_s, & \text{in the stopband} \end{cases}$$

or

$$W(\omega) = \begin{cases} \delta_s/\delta_p, & \text{in the passband} \\ 1, & \text{in the stopband} \end{cases}$$

9

Design of Equiripple Linear-Phase FIR Filters

- **Type 1 FIR Filter** - $\breve{H}(\omega) = \sum_{k=0}^{M} a[k]\cos(\omega k)$

  where

  $a[0] = h[M],\ a[k] = 2h[M-k],\ 1 \le k \le M$
- **Type 2 FIR filter** -

$$\breve{H}(\omega) = \sum_{k=1}^{(2M+1)/2} b[k]\cos\left(\omega(k - \tfrac{1}{2})\right)$$

  where

  $b[k] = 2h[\tfrac{2M+1}{2} - k],\ 1 \le k \le \tfrac{2M+1}{2}$

10

Design of Equiripple Linear-Phase FIR Filters

- **Type 3 FIR Filter** - $\breve{H}(\omega) = \sum_{k=1}^{M} c[k]\sin(\omega k)$

  where

  $c[k] = 2h[M-k],\ 1 \le k \le M$
- **Type 4 FIR Filter** -

$$\breve{H}(\omega) = \sum_{k=1}^{(2M+1)/2} d[k]\sin\left(\omega(k - \tfrac{1}{2})\right)$$

  where

$$d[k] = 2h[\tfrac{2M+1}{2} - k], \quad 1 \le k \le \tfrac{2M+1}{2}$$

11

Design of Equiripple Linear-Phase FIR Filters

- Amplitude response for all 4 types of linear-phase FIR filters can be expressed as

$$\breve{H}(\omega) = Q(\omega)A(\omega)$$

where

$$Q(\omega) = \begin{cases} 1, & \text{for Type 1} \\ \cos(\omega/2), & \text{for Type 2} \\ \sin(\omega), & \text{for Type 3} \\ \sin(\omega/2), & \text{for Type 4} \end{cases}$$

12

Design of Equiripple Linear-Phase FIR Filters

and

$$A(\omega) = \sum_{k=0}^{L} \tilde{a}[k]\cos(\omega k)$$

where

$$\tilde{a}[k] = \begin{cases} a[k], & \text{for Type 1} \\ \tilde{b}[k], & \text{for Type 2} \\ \tilde{c}[k], & \text{for Type 3} \\ \tilde{d}[k], & \text{for Type 4} \end{cases}$$

13

Design of Equiripple Linear-Phase FIR Filters

with

$$L = \begin{cases} M, & \text{for Type 1} \\ \frac{2M-1}{2}, & \text{for Type 2} \\ M-1, & \text{for Type 3} \\ \frac{2M-1}{2}, & \text{for Type 4} \end{cases}$$

$\tilde{b}[k]$, $\tilde{c}[k]$, and $\tilde{d}[k]$, are related to $b[k]$, $c[k]$, and $d[k]$, respectively

14

Design of Equiripple Linear-Phase FIR Filters

- Modified form of weighted error function

$$\mathcal{E}(\omega) = W(\omega)[Q(\omega)A(\omega) - D(\omega)]$$
$$= W(\omega)Q(\omega)[A(\omega) - \tfrac{D(\omega)}{Q(\omega)}]$$
$$= \tilde{W}(\omega)[A(\omega) - \tilde{D}(\omega)]$$

where we have used the notation

$$\tilde{W}(\omega) = W(\omega)Q(\omega)$$
$$\tilde{D}(\omega) = D(\omega)/Q(\omega)$$

15

Design of Equiripple Linear-Phase FIR Filters

- Optimization Problem - Determine $\tilde{a}[k]$ which minimize the peak absolute value ε of

$$\mathcal{E}(\omega) = \tilde{W}(\omega)[\sum_{k=0}^{L} \tilde{a}[k]\cos(\omega k) - \tilde{D}(\omega)]$$

over the specified frequency bands $\omega \in R$

- After $\tilde{a}[k]$ has been determined, corresponding coefficients of the original $A(\omega)$ are computed from which $h[n]$ are determined

16

Design of Equiripple Linear-Phase FIR Filters

- Alternation Theorem - $A(\omega)$ is the best unique approximation of $\tilde{D}(\omega)$ obtained by minimizing peak absolute value ε of

$$\mathcal{E}(\omega) = W(\omega)[Q(\omega)A(\omega) - D(\omega)]$$

if and only if there exist at least L+2 extremal frequencies, $\{\omega_i\}$, $0 \le i \le L+1$, in a closed subset R of the frequency range $0 \le \omega \le \pi$ such that $\omega_0 < \omega_1 < \cdots < \omega_L < \omega_{L+1}$ and $\mathcal{E}(\omega_i) = -\mathcal{E}(\omega_{i+1})$, $|\mathcal{E}(\omega_i)| = \varepsilon$ for all i

17

Design of Equiripple Linear-Phase FIR Filters

- Consider a Type 1 FIR filter with an amplitude response $A(\omega)$ whose approximation error $\mathcal{E}(\omega)$ satisfies the Alternation Theorem
- Peaks of $\mathcal{E}(\omega)$ are at $\omega = \omega_i$, $0 \le i \le L+1$ where $d\mathcal{E}(\omega)/d\omega = 0$
- Since in the passband and stopband, $\tilde{W}(\omega)$ and $\tilde{D}(\omega)$ are piecewise constant,

$$\frac{d\mathcal{E}(\omega)}{d\omega} = \frac{dA(\omega)}{d\omega} = 0 \text{ at } \omega = \omega_i$$

18

Design of Equiripple Linear-Phase FIR Filters

- Using $\cos(\omega k) = T_k(\cos\omega)$, where $T_k(x)$ is the k-th order Chebyshev polynomial
$$T_k(x) = \cos(k \cos^{-1} x)$$
- $A(\omega)$ can be expressed as
$$A(\omega) = \sum_{k=0}^{L} \alpha[k](\cos\omega)^k$$
which is an Lth-order polynomial in $\cos\omega$
- Hence, $A(\omega)$ can have at most $L-1$ local minima and maxima inside specified passband and stopband

19

Design of Equiripple Linear-Phase FIR Filters

- At bandedges, $\omega = \omega_p$ and $\omega = \omega_s$, $|\mathcal{E}(\omega)|$ is a maximum, and hence $A(\omega)$ has extrema at these points
- $A(\omega)$ can have extrema at $\omega = 0$ and $\omega = \pi$
- Therefore, there are at most L+3 extremal frequencies of $\mathcal{E}(\omega)$
- For linear-phase FIR filters with K specified bandedges, there can be at most L+K+1 extremal frequencies

20

Design of Equiripple Linear-Phase FIR Filters

- The set of equations
$$\tilde{W}(\omega_i)[A(\omega_i) - \tilde{D}(\omega_i)] = (-1)^i \varepsilon, \quad 0 \le i \le L+1$$
is written in a matrix form
$$\begin{bmatrix} 1 & \cos(\omega_0) & \cdots & \cos(L\omega_0) & -1/\tilde{W}(\omega_0) \\ 1 & \cos(\omega_1) & \cdots & \cos(L\omega_1) & 1/\tilde{W}(\omega_1) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & \cos(\omega_L) & \cdots & \cos(L\omega_L) & (-1)^{L-1}/\tilde{W}(\omega_L) \\ 1 & \cos(\omega_{L+1}) & \cdots & \cos(L\omega_{L+1}) & (-1)^{L}/\tilde{W}(\omega_{L+1}) \end{bmatrix} \begin{bmatrix} \tilde{a}[0] \\ \tilde{a}[1] \\ \vdots \\ \tilde{a}[L] \\ \varepsilon \end{bmatrix} = \begin{bmatrix} \tilde{D}(\omega_0) \\ \tilde{D}(\omega_1) \\ \vdots \\ \tilde{D}(\omega_L) \\ \tilde{D}(\omega_{L+1}) \end{bmatrix}$$

21

Design of Equiripple Linear-Phase FIR Filters

- The matrix equation can be solved for the unknowns $\tilde{a}[i]$ and ε if the locations of the L+2 extremal frequencies are known a priori
- The **Remez exchange algorithm** is used to determine the locations of the extremal frequencies

22

Remez Exchange Algorithm

- Step 1: A set of initial values of extremal frequencies are either chosen or are available from completion of previous stage
- Step 2: Value of ε is computed using
$$\varepsilon = \frac{c_0\tilde{D}(\omega_0) + c_1\tilde{D}(\omega_1) + \cdots + c_{L+1}\tilde{D}(\omega_{L+1})}{\frac{c_0}{\tilde{W}(\omega_0)} - \frac{c_1}{\tilde{W}(\omega_1)} + \cdots + \frac{(-1)^{L+1}c_{L+1}}{\tilde{W}(\omega_{L+1})}}$$
where $\displaystyle c_n = \prod_{\substack{i=0 \\ i \ne n}}^{L+1} \frac{1}{\cos(\omega_n) - \cos(\omega_i)}$

23

Remez Exchange Algorithm

- Step 3: Values of $A(\omega)$ at $\omega = \omega_i$ are then computed using
$$A(\omega_i) = \frac{(-1)^i \varepsilon}{\tilde{W}(\omega_i)} + \tilde{D}(\omega_i), \quad 0 \le i \le L+1$$
- Step 4: The polynomial $A(\omega)$ is determined by interpolating the above values at the L+2 extremal frequencies using the Lagrange interpolation formula

24

Remez Exchange Algorithm

- Step 4: The new error function

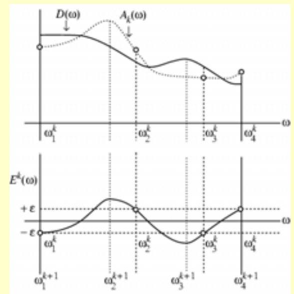
$$\mathcal{E}(\omega) = \tilde{W}(\omega)[A(\omega) - \tilde{D}(\omega)]$$

is computed at a dense set S ($S \geq L$) of frequencies. In practice $S = 16L$ is adequate. Determine the L+2 new extremal frequencies from the values of $\mathcal{E}(\omega)$ evaluated at the dense set of frequencies.

- Step 5: If the peak values ε of $\mathcal{E}(\omega)$ are equal in magnitude, algorithm has converged. Otherwise, go back to Step 2.

Remez Exchange Algorithm

- Illustration of algorithm



Iteration process is stopped if the difference between the values of the peak absolute errors between two consecutive stages is less than a preset value, e.g., 10^{-6}

Remez Exchange Algorithm

- Example - Approximate the desired function $D(x) = 1.1x^2 - 0.1$ defined for the range $0 \leq x \leq 2$ by a linear function $a_1 x + a_0$ by minimizing the peak value of the absolute error

$$\max_{x \in [0,2]} \left| 1.1x^2 - 0.1 - a_0 - a_1 x \right|$$

- Stage 1:

Choose arbitrarily the initial extremal points

$$x_1 = 0,\ x_2 = 0.5,\ x_3 = 1.5$$

Remez Exchange Algorithm

- Solve the three linear equations

$$a_0 + a_1 x_\ell - (-1)^\ell \varepsilon = D(x_\ell), \quad \ell = 1, 2, 3$$

i.e.,

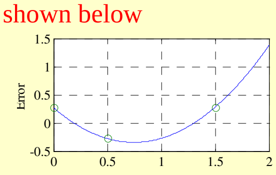
$$\begin{bmatrix} 1 & 0 & 1 \\ 1 & 0.5 & -1 \\ 1 & 1.5 & 1 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \varepsilon \end{bmatrix} = \begin{bmatrix} -0.1 \\ 0.175 \\ 2.375 \end{bmatrix}$$

for the given extremal points yielding

$$a_0 = -0.375,\ a_1 = 1.65,\ \varepsilon = 0.275$$

Remez Exchange Algorithm

- Plot of $\mathcal{E}_1(x) = 1.1x^2 - 1.65x + 0.275$ along with values of error at chosen extremal points shown below



- Note: Errors are equal in magnitude and alternate in sign

Remez Exchange Algorithm

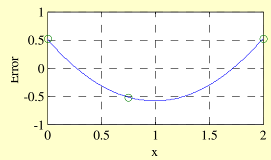
- Stage 2:
- Choose extremal points where $\mathcal{E}_1(x)$ assumes its maximum absolute values
- These are $x_1 = 0,\ x_2 = 0.75,\ x_3 = 2$
- New values of unknowns are obtained by solving

$$\begin{bmatrix} 1 & 0 & 1 \\ 1 & 0.75 & -1 \\ 1 & 2 & 1 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \varepsilon \end{bmatrix} = \begin{bmatrix} -0.1 \\ 0.5188 \\ 4.3 \end{bmatrix}$$

yielding $a_0 = -0.6156,\ a_1 = 2.2,\ \varepsilon = 0.5156$

Remez Exchange Algorithm

- Plot of $\mathcal{E}_2(x) = 1.1x^2 - 2.2x + 0.5156$ along with values of error at chosen extremal points shown below



31

Remez Exchange Algorithm

- Stage 3:
- Choose extremal points where $\mathcal{E}_2(x)$ assumes its maximum absolute values
- These are $x_1 = 0,\ x_2 = 1,\ x_3 = 2$
- New values of unknowns are obtained by solving

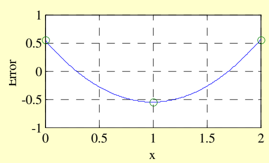
$$\begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & -1 \\ 1 & 2 & 1 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \varepsilon \end{bmatrix} = \begin{bmatrix} -0.1 \\ 1.0 \\ 4.3 \end{bmatrix}$$

yielding $a_0 = -0.65,\ a_1 = 2.2,\ \varepsilon = 0.55$

32

Remez Exchange Algorithm

- Plot of $\mathcal{E}_3(x) = 1.1x^2 - 2.2x + 0.55$ along with values of error at chosen extremal points shown below



- Algorithm has converged as ε is also the maximum value of the absolute error

33

Design of Minimum-Phase FIR Filters

- Linear-phase FIR filters with narrow transition bands are of very high order, and as a result have a very long group delay that is about half the filter order
- By relaxing the linear-phase requirement, it is possible to design an FIR filter of lower order thus reducing the overall group delay and the computational cost

34

Design of Minimum-Phase FIR Filters

- A very simple method of minimum-phase FIR filter is described next
- Consider an arbitrary FIR transfer function of degree N:

$$H(z) = \sum_{n=0}^{N} h[n] z^{-n} = h[0] \prod_{k=1}^{N} (1 - \xi_k z^{-1})$$

35

Design of Minimum-Phase FIR Filters

- The mirror-image polynomial to $H(z)$ is given by

$$\hat{H}(z) = z^{-N} H(z^{-1})$$
$$= \sum_{n=0}^{N} h[N-n] z^{-n} = h[N] \prod_{k=1}^{N} (1 - z^{-1}/\xi_k)$$

- The zeros of $\hat{H}(z)$ are thus at $z = 1/\xi_k$, i.e., are reciprocal to the zeros of $H(z)$ at $z = \xi_k$

36

Design of Minimum-Phase FIR Filters

- As a result,

$$G(z) = H(z)\hat{H}(z) = z^{-N} H(z) H(z^{-1})$$

has zeros exhibiting mirror-image symmetry in the z-plane and is thus a Type 1 linear-phase transfer function of order $2N$

- Moreover, if $H(z)$ has a zero on the unit circle, $\hat{H}(z)$ will also have a zero on the unit circle at the conjugate reciprocal position

Design of Minimum-Phase FIR Filters

- Thus, unit circle zeros of $G(z)$ occur in pairs
- On the unit circle we have

$$\left|H(e^{j\omega})\right|^2 = \breve{G}(\omega)$$

$$\Rightarrow \quad \breve{G}(\omega) \geq 0$$

- Moreover, the amplitude response $\breve{G}(\omega)$ has double zeros in the frequency range $[0, \pi]$

Design of Minimum-Phase FIR Filters

- **Design Procedure –**
- **Step 1**: Design a Type 1 linear-phase transfer function $F(z)$ of degree $2N$ satisfying the specifications:

$$1 - \delta_p^{(F)} \leq \breve{F}(\omega) \leq 1 + \delta_p^{(F)} \quad \text{for} \quad \omega \in [0, \omega_p]$$

$$-\delta_s^{(F)} \leq \breve{F}(\omega) \leq \delta_p^{(F)} \quad \text{for} \quad \omega \in [\omega_s, \pi]$$

- Note that $F(z)$ has single unit circle zeros

Design of Minimum-Phase FIR Filters

- **Step 2**: Determine the linear-phase transfer function

$$G(z) = \delta_s^{(F)} z^{-N} + F(z)$$

- Its amplitude response satisfies

$$1 + \delta_s^{(F)} - \delta_p^{(F)} \leq \breve{G}(\omega) \leq 1 + \delta_s^{(F)} + \delta_p^{(F)} \quad \text{for} \quad \omega \in [0, \omega_p]$$

$$0 \leq \breve{G}(\omega) \leq 2\delta_s^{(F)} \quad \text{for} \quad \omega \in [\omega_s, \pi]$$

Design of Minimum-Phase FIR Filters

- Note that $G(z)$ has double zeros on the unit circle and all other zeros are situated with a mirror-image symmetry
- Hence, it can be expressed in the form

$$G(z) = z^{-N} H_m(z) H_m(z^{-1})$$

where $H_m(z)$ is a minimum-phase transfer function containing all zeros of $G(z)$ that are inside the unit circle and one each of the unit circle double zeros

Design of Minimum-Phase FIR Filters

- **Step 3**: Determine $H_m(z)$ from $G(z)$ by applying a spectral factorization
- The passband ripple $\delta_p^{(F)}$ and the stopband ripple $\delta_s^{(F)}$ of $F(z)$ must be chosen to ensure that the specified passband ripple δ_p and the stopband ripple δ_s of $H_m(z)$ are satisfied

Design of Minimum-Phase FIR Filters

- It can be shown

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

- An estimate of the order N of $H_m(z)$ can be found by first estimating the order of $F(z)$ and then dividing it by 2
- If the estimated order of $F(z)$ is an odd integer, it should be increased by 1

FIR Digital Filter Design Using MATLAB

- Order Estimation -
- Kaiser's Formula:

$$N \cong \frac{-20\log_{10}(\sqrt{\delta_p \delta_s})}{14.6(\omega_s - \omega_p)/2\pi}$$

- Note: Filter order N is inversely proportional to transition band width $(\omega_s - \omega_p)$ and does not depend on actual location of transition band

FIR Digital Filter Design Using MATLAB

- Hermann-Rabiner-Chan's Formula:

$$N \cong \frac{D_\infty(\delta_p,\delta_s) - F(\delta_p,\delta_s)[(\omega_s-\omega_p)/2\pi]^2}{(\omega_s-\omega_p)/2\pi}$$

where

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

$$F(\delta_p,\delta_s) = b_1 + b_2[\log_{10}\delta_p - \log_{10}\delta_s]$$

with $a_1 = 0.005309,\ a_2 = 0.07114,\ a_3 = -0.4761$

$a_4 = 0.00266,\ a_5 = 0.5941,\ a_6 = 0.4278$

$b_1 = 11.01217,\quad b_2 = 0.51244$

FIR Digital Filter Design Using MATLAB

- Formula valid for $\delta_p \geq \delta_s$
- For $\delta_p < \delta_s$, formula to be used is obtained by interchanging δ_p and δ_s
- Both formulas provide only an estimate of the required filter order N
- Frequency response of FIR filter designed using this estimated order may or may not meet the given specifications
- If specifications are not met, increase filter order until they are met

FIR Digital Filter Design Using MATLAB

- MATLAB code fragments for estimating filter order using Kaiser's formula

```
num = - 20*log10(sqrt(dp*ds)) - 13;
den = 14.6*(Fs - Fp)/FT;
N = ceil(num/den);
```

- M-file remezord implements Hermann-Rabiner-Chan's order estimation formula

FIR Digital Filter Design Using MATLAB

- For FIR filter design using the Kaiser window, window order is estimated using the M-file kaiserord
- The M-file kaiserord can in some cases generate a value of N which is either greater or smaller than the required minimum order
- If filter designed using the estimated order N does not meet the specifications, N should either be gradually increased or decreased until the specifications are met

Equiripple FIR Digital Filter Design Using MATLAB

- The M-file remez can be used to design an equiripple FIR filter using the Parks-McClellan algorithm
- Example - Design an equiripple FIR filter with the specifications: $F_p = 0.8$ kHz, $F_s = 1$ kHz, $F_T = 4$ kHz, $\alpha_p = 0.5$dB, $\alpha_s = 40$ dB
- Here, $\delta_p = 0.0559$ and $\delta_s = 0.01$

49

Equiripple FIR Digital Filter Design Using MATLAB

- MATLAB code fragments used are

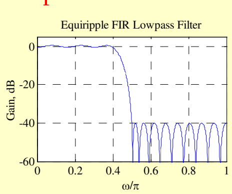
```
[N, fpts, mag, wt] =
    remezord(fedge, mval, dev, FT);
b = remez(N, fpts, mag, wt);
```

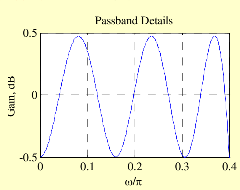
where fedge = [800 1000],
mval = [1 0], dev = [0.0559 0.01], and
FT = 4000

50

Equiripple FIR Digital Filter Design Using MATLAB

- The computed gain response with the filter order obtained (N = 28) does not meet the specifications ($\alpha_p = 0.6$dB, $\alpha_s = 38.7$dB)
- Specifications are met with N = 30





51

Equiripple FIR Digital Filter Design Using MATLAB

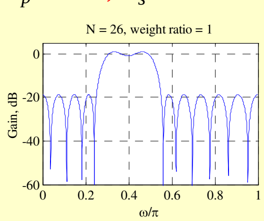
- Example - Design a linear-phase FIR bandpass filter of order 26 with a passband from 0.3 to 0.5, and stopbands from 0 to 0.25 and from 0.55 to 1
- The pertinent input data here are

```
N = 26
fpts = [0  0.25  0.3  0.5  0.55 1]
mag = [0  0  1  1  0  0]
wt = [1  1  1]
```

52

Equiripple FIR Digital Filter Design Using MATLAB

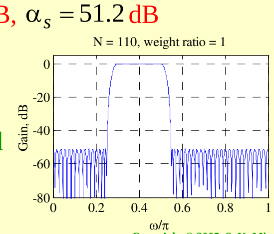
- Computed gain response shown below where $\alpha_p = 1$ dB, $\alpha_s = 18.7$ dB



53

Equiripple FIR Digital Filter Design Using MATLAB

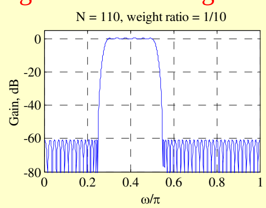
- We redesign the filter with order increased to 110
- Computed gain response shown below where $\alpha_p = 0.024$ dB, $\alpha_s = 51.2$ dB
- Note: Increase in order improves gain response at the expense of increased computational complexity



54

Equiripple FIR Digital Filter Design Using MATLAB

- α_s can be increased at the expenses of a larger α_p by decreasing the relative weight ratio $W(\omega) = \delta_p / \delta_s$
- Gain response of bandpass filter of order 110 obtained with a weight vector [1 0.1 1]

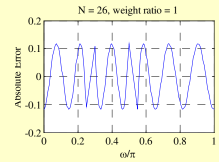


- Now $\alpha_p = 0.076$ dB, $\alpha_s = 60.86$ dB

55

Equiripple FIR Digital Filter Design Using MATLAB

- Plots of absolute error for 1st design
- Absolute error has same peak value in all bands

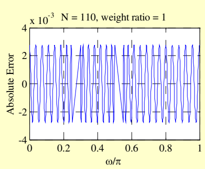


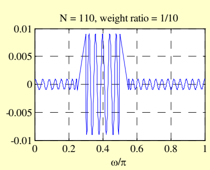
- As L = 13, and there are 4 band edges, there can be at most $L - 1 + 6 = 18$ extrema
- Error plot exhibits 17 extrema

56

Equiripple FIR Digital Filter Design Using MATLAB

- Absolute error has same peak value in all bands for the 2nd design
- Absolute error in passband of 3rd design is 10 times the error in the stopbands





57

Equiripple FIR Digital Filter Design Using MATLAB

- Example - Design a linear-phase FIR bandpass filter of order 60 with a passband from 0.3 to 0.5, and stopbands from 0 to 0.25 and from 0.6 to 1 with unequal weights
- The pertinent input data here are

  N = 60

  fpts = [0 0.25 0.3 0.5 0.6 1]

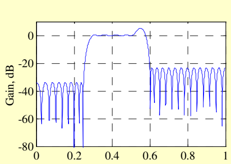
  mag = [0 0 1 1 0 0]

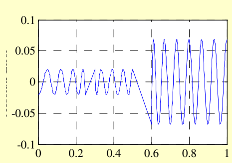
  wt = [1 1 0.3]

58

Equiripple FIR Digital Filter Design Using MATLAB

- Plots of gain response and absolute error shown below





59

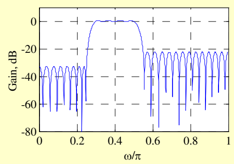
Equiripple FIR Digital Filter Design Using MATLAB

- Response in the second transition band shows a peak with a value higher than that in passband
- Result does not contradict alternation theorem
- As N = 60, M = 30, and hence, there must be at least M + 2 = 32 extremal frequencies
- Plot of absolute error shows the presence of 32 extremal frequencies

60

Equiripple FIR Digital Filter Design Using MATLAB

- If gain response of filter designed exhibits a nonmonotonic behavior, it is recommended that either the filter order or the bandedges or the weighting function be adjusted until a satisfactory gain response has been obtained
- Gain plot obtained by moving the second stopband edge to 0.55



61

Equiripple FIR Differentiator Design Using MATLAB

- A lowpass differentiator has a bandlimited frequency response

$$H_{DIF}(e^{j\omega}) = \begin{cases} j\omega, & 0 \le |\omega| \le \omega_p \\ 0, & \omega_s \le |\omega| \le \pi \end{cases}$$

where $0 \le |\omega| \le \omega_p$ represents the passband and $\omega_s \le |\omega| \le \pi$ represents the stopband

- For the design phase we choose

$$W(\omega) = 1/\omega, \quad D(\omega) = 1, \quad 0 \le |\omega| \le \omega_p$$

62

Equiripple FIR Differentiator Design Using MATLAB

- The M-file remezord cannot be used to estimate the order of an FIR differentiator
- Example - Design a full-band ($\omega_p = \pi$) differentiator of order 11
- Code fragment to use

b = remez(N, fpts, mag, 'differentiator');

where N = 11

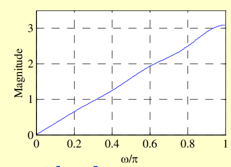
fpts = [0 1]

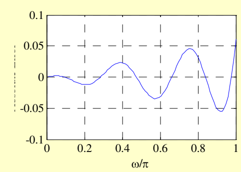
mag = [0 pi]

63

Equiripple FIR Differentiator Design Using MATLAB

- Plots of magnitude response and absolute error





- Absolute error increases with ω as the algorithm results in an equiripple error of the function $[\frac{A(\omega)}{\omega} - 1]$

64

Equiripple FIR Differentiator Design Using MATLAB

- Example - Design a lowpass differentiator of order 50 with $\omega_p = 0.4\pi$ and $\omega_s = 0.45\pi$
- Code fragment to use

b = remez(N, fpts, mag, 'differentiator');

where

N = 50

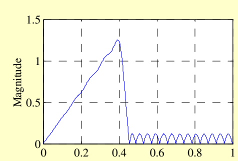
fpts = [0 0.4 0.45 1]

mag = [0 0.4*pi 0 0]

65

Equiripple FIR Differentiator Design Using MATLAB

- Plot of the magnitude response of the lowpass differentiator



66

Equiripple FIR Hilbert Transformer Design Using MATLAB

- Desired amplitude response of a bandpass Hilbert transformer is

$$D(\omega) = 1, \quad \omega_L \le |\omega| \le \omega_H$$

with weighting function

$$W(\omega) = 1, \quad \omega_L \le |\omega| \le \omega_H$$

- Impulse response of an ideal Hilbert transformer satisfies the condition

$$h_{HT}[n] = 0, \quad \text{for } n \text{ even}$$

which can be met by a Type 3 FIR filter

67

Equiripple FIR Hilbert Transformer Design Using MATLAB

- Example - Design a linear-phase bandpass FIR Hilbert transformer of order 20 with $\omega_L = 0.1\pi$, $\omega_H = 0.9\pi$
- Code fragment to use

```
b = remez(N, fpts, mag, 'Hilbert');
```

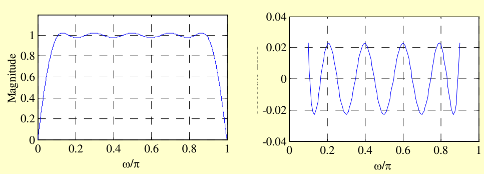
where

```
N = 20
fpts = [0.1   0.9]
mag = [1   1]
```

68

Equiripple FIR Hilbert Transformer Design Using MATLAB

- Plots of magnitude response and absolute error



69

Window-Based FIR Filter Design Using MATLAB

- Window Generation - Code fragments to use

```
w = blackman(L);
w = hamming(L);
w = hanning(L);
w = chebwin(L, Rs);
w = kaiser(L, beta);
```

where window length L is odd

70

Window-Based FIR Filter Design Using MATLAB

- Example - Kaiser window design for use in a lowpass FIR filter design
- Specifications of lowpass filter: $\omega_p = 0.3\pi$, $\omega_s = 0.4\pi$, $\alpha_s = 50$ dB $\Rightarrow \delta_s = 0.003162$
- Code fragments to use

```
[N, Wn, beta, ftype] = kaiserord(fpts, mag,dev);
w = kaiser(N+1, beta);
```

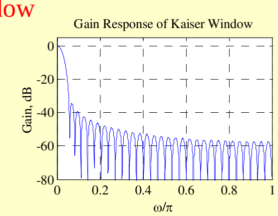
where

```
fpts = [0.3   0.4]
mag = [1   0]
dev = [0.003162   0.003162]
```

71

Window-Based FIR Filter Design Using MATLAB

- Plot of the gain response of the Kaiser window



72

Window-Based FIR Filter Design Using MATLAB

- M-files available are fir1 and fir2
- fir1 is used to design conventional lowpass, highpass, bandpass, bandstop and multiband FIR filters
- fir2 is used to design FIR filters with arbitrarily shaped magnitude response
- In fir1, Hamming window is used as a default if no window is specified

73

Window-Based FIR Filter Design Using MATLAB

- Example - Design using a Kaiser window a lowpass FIR filter with the specifications: $\omega_p = 0.3\pi$, $\omega_s = 0.4\pi$, $\delta_s = 0.003162$
- Code fragments to use

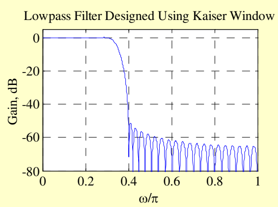
```
[N, Wn, beta, ftype] = kaiserord(fpts, mag, dev);
b = fir1(N, Wn, kaiser(N+1, beta));
```

where fpts = [0.3 0.4]
mag = [1 0]
dev = [0.003162 0.003162]

74

Window-Based FIR Filter Design Using MATLAB

- Plot of gain response



75

Window-Based FIR Filter Design Using MATLAB

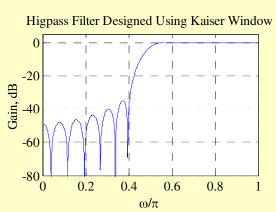
- Example - Design using a Kaiser window a highpass FIR filter with the specifications: $\omega_p = 0.55\pi$, $\omega_s = 0.4\pi$, $\delta_s = 0.02$
- Code fragments to use
- [N, Wn, beta, ftype] = kaiserord(fpts, mag, dev);
b = fir1(N, Wn, 'ftype', kaiser(N+1, beta));

where fpts = [0.4 0.55]
mag = [0 1]
dev = [0.02 0.02]

76

Window-Based FIR Filter Design Using MATLAB

- Plot of gain response



77

Window-Based FIR Filter Design Using MATLAB

- Example - Design using a Hamming window an FIR filter of order 100 with three different constant magnitude levels: 0.3 in the frequency range [0, 0.28], 1.0 in the frequency range [0.3, 0.5], and 0.7 in the frequency range [0.52, 1.0]

78

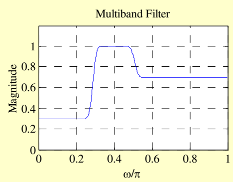
Window-Based FIR Filter Design Using MATLAB

- Code fragment to use

  b = fir2(100, fpts, mval);

  where fpts = [0 0.28 0.3 0.5 0.52 1];

  mval = [0.3 0.3 1.0 1.0 0.7 0.7];



79

Minimum-Phase FIR Filter Design Using MATLAB

- The minimum-phase FIR filter design method outlined earlier involves the spectral factorization of a Type 1 linear-phase FIR transfer function $G(z)$ with a non-negative amplitude response in the form

$$G(z) = z^{-N} H_m(z) H_m(z^{-1})$$

where $H_m(z)$ contains all zeros of $G(z)$ that are inside the unit circle and one each of the unit circle double zeros

80

Spectral Factorization

- We next outline the basic idea behind a simple spectral factorization method
- Without any loss of generality we consider the spectral factorization of a 6-th order linear-linear phase FIR transfer function $G(z)$ with a non-negative amplitude response:

$$G(z) = g_3 + g_2 z^{-1} + g_1 z^{-2} + g_0 z^{-3} + g_1 z^{-4} + g_2 z^{-5} + g_3 z^{-6}$$

81

Spectral Factorization

- Our objective is to express the above $G(z)$ in the form

$$G(z) = z^{-3} H_m(z) H_m(z^{-1})$$

where

$$H_m(z) = a_0 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3}$$

is the minimum-phase factor of $G(z)$

82

Spectral Factorization

- Expressing $G(z)$ in terms of the coefficients of $H_m(z)$ we get

$$G(z) = (a_0 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3}) \times (a_3 + a_2 z^{-1} + a_1 z^{-2} + a_0 z^{-3})$$

- Forming the product of the two polynomials given above and comparing the coefficients of like powers of z^{-1} the product with that of $G(z)$ given on the previous slide we arrive at 4 equations given in the next slide

83

Spectral Factorization

$$g_0 = a_0^2 + a_1^2 + a_2^2 + a_3^2$$
$$g_1 = a_0 a_1 + a_1 a_2 + a_2 a_3$$
$$g_2 = a_0 a_2 + a_1 a_3$$
$$g_3 = a_0 a_3$$

- The above set of equations is then solved iteratively using the Newton-Raphson method

84

Spectral Factorization

- First, the initial values of a_i are chosen to ensure that $H_m(z)$ has all zeros strictly inside the unit circle
- Then, the coefficients a_i are changed by adding the corrections e_i so that the modified values $a_i + e_i$ satisfy better the set of 4 equalities given in the previous slide
- The process is repeated until the iteration converges

85

Spectral Factorization

- Substituting $a_i + e_i$ in the 4 equations given earlier and expanding the products, a set of linear equations are obtained by eliminating all quadratic terms in e_i from the expansion
- In matrix form, these equations can be written as $\mathbf{Ae} = \mathbf{b}$ where

$$\mathbf{A} = \begin{bmatrix} 2a_0 & 2a_1 & 2a_2 & 2a_3 \\ a_1 & a_0 + a_2 & a_3 + a_1 & a_2 \\ a_2 & a_3 & a_0 & a_1 \\ a_3 & 0 & 0 & a_0 \end{bmatrix}$$

86

Spectral Factorization

and

$$\mathbf{e} = \begin{bmatrix} e_0 \\ e_1 \\ e_2 \\ e_3 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} g_0 - a_0^2 - a_1^2 - a_2^2 - a_3^2 \\ g_1 - a_0 a_1 - a_1 a_2 - a_2 a_3 \\ g_2 - a_0 a_2 - a_1 a_3 \\ g_2 - a_0 a_3 \end{bmatrix}$$

- The matrix $\mathbf{A}$ can be expressed as

$$\mathbf{A} = \begin{bmatrix} a_0 & a_1 & a_2 & a_3 \\ a_1 & a_2 & a_3 & 0 \\ a_2 & a_3 & 0 & 0 \\ a_3 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} a_0 & a_1 & a_2 & a_3 \\ 0 & a_0 & a_1 & a_2 \\ 0 & 0 & a_0 & a_1 \\ 0 & 0 & 0 & a_0 \end{bmatrix}$$

87

Spectral Factorization

- The iteration convergence is checked at each step by evaluating the error term $\Sigma_{i=0}^{3} e_i^2$
- The error term first decreases monotonically and the iteration is stopped when the error starts increasing
- The M-file `minphase.m` implements the above spectral factorization method

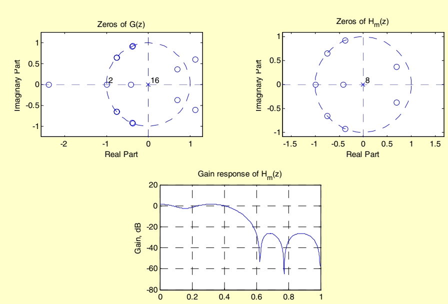
88

Minimum-Phase FIR Filter Design Using MATLAB

- **Example –** Design a minimum-phase lowpass FIR filter with the following specifications: $\omega_p = 0.45\pi$, $\omega_s = 0.6\pi$, $R_p = 2$ dB and $R_s = 26$ dB
- Using Program 10_3.m we arrive at the desired filter
- Plots of zeros of $G(z)$, zeros of $H_m(z)$, and the gain response of $H_m(z)$ are shown in the next slide

89

Minimum-Phase FIR Filter Design Using MATLAB



90

Maximum-Phase FIR Filter Design Using MATLAB

- A maximum-phase spectral factor of a linear-phase FIR filter with an impulse response `b` of even order with a non-negative zero-phase frequency response can be designed by first computing its minimum-phase spectral factor `h` and the using the statement

```
G = fliplr(h)
```

91

Design of Computationally Efficient FIR Digital Filters

- As indicated earlier, the order N of a linear-phase FIR filter is inversely proportional to the width $\Delta\omega$ of the transition band
- Hence, in the case of an FIR filter with a very sharp transition, the order of the filter is very high
- This is particularly critical in designing very narrow-band or very wide-band FIR filters

92

Design of Computationally Efficient FIR Digital Filters

- The computational complexity of a digital filter is basically determined by the total number of multipliers and adders needed to implement the filter
- The direct form implementation of a linear-phase FIR filter of order N requires, in general, $\left\lfloor \frac{N+1}{2} \right\rfloor$ multipliers and N two-input adders

93

Design of Computationally Efficient FIR Digital Filters

- We now outline two methods of realizing computationally efficient linear-phase FIR filters
- The basic building block in both methods is an FIR subfilter structure with a periodic impulse response

94

The Periodic Filter Section

- Consider a Type 1 linear-phase FIR filter $F(z)$ of even degree N:

$$F(z) = \sum_{n=0}^{N} f[n] z^{-n}$$

- Its delay-complementary filter $E(z)$ is given by

$$E(z) = z^{-N/2} - F(z) = z^{-N/2} - \sum_{n=0}^{N} f[n] z^{-n}$$

$$= (1 - f[N/2]) z^{-N/2} - \sum_{\substack{n=0 \\ n \neq N/2}}^{N} f[n] z^{-n}$$

95

The Periodic Filter Section

- The transfer function $H(z)$ obtained by replacing z^{-1} in $F(z)$ with z^{-L}, with L being a positive integer, is given by

$$H(z) = F(z^L) = \sum_{n=0}^{N} f[n] z^{-nL}$$

- The order of $H(z)$ is thus NL
- A direct realization of $H(z)$ is obtained by simply replacing each unit delay in the realization of $F(z)$ with L unit delays

96

The Periodic Filter Section

- Note: The number of multipliers and delays in the realization of $H(z)$ is the same as those in the realization of $F(z)$
- The transfer function $H(z)$ has a sparse impulse response of length $NL+1$, with $L-1$ zero-valued samples inserted between every consecutive pair of impulse response samples of $F(z)$

The Periodic Filter Section

- The parameter L is called the sparsity factor
- The relations between the amplitude responses of these two filters is given by

$$\breve{H}(\omega) = \breve{F}(L\omega)$$

- It follows from the above that the amplitude response $\breve{H}(\omega)$ is a period function of ω with a period $2\pi/L$

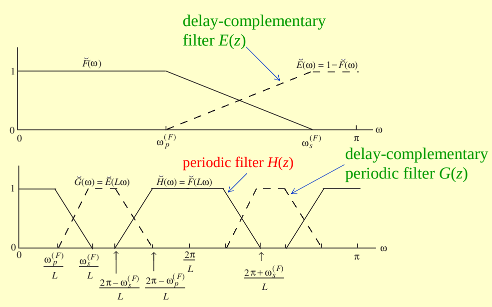
The Periodic Filter Section

- One period of $\breve{H}(\omega)$ is obtained by compressing the amplitude response $\breve{F}(\omega)$ in the interval $[0, 2\pi]$ to the interval $[0, 2\pi/L]$
- A transfer function $H(z)$ with a frequency response that is a periodic function of ω with a period 2π/L is called a periodic filter

The Periodic Filter Section

- If $F(z)$ is a lowpass filter with a single pasband and a single stopband, $H(z)$ will be a multiband filter with $\lfloor L/2 \rfloor + 1$ pasbands and $\lceil L/2 \rceil$ stopbands as shown in the next slide for $L = 4$

The Periodic Filter Section



The Periodic Filter Section

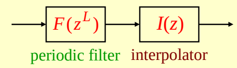
- Let $F(z)$ be a lowpass filter with passband edge at $\omega_p^{(F)}$ and and stopband edge at $\omega_s^{(F)}$, where $\omega_s^{(F)} < \pi$
- Then, the passband and stopband edges of the first band of $H(z)$ are at $\omega_p^{(F)}/L$ and $\omega_p^{(F)}/L$, respectively
- The passband and stopband edges of the second band of $H(z)$ are at $(2\pi \pm \omega_p^{(F)})/L$ and $(2\pi \pm \omega_p^{(F)})/L$, respectively, and so on as shown on the previous slide

The Periodic Filter Section

- The width of the transition bands of $H(z)$ are $(\omega_s^{(F)} - \omega_p^{(F)})/L$, which is $\frac{1}{L}$-th of that of $F(z)$
- Likewise, the transfer function $G(z)$ by replacing z^{-1} in $E(z)$ with z^{-L}, is given by
$$G(z) = E(z^L) = z^{-NL/2} - F(z^L)$$
$$= z^{-NL/2} - \sum_{n=0}^{N} f[n] z^{-nL}$$
- The amplitude response of $G(z)$ is given by
$$\breve{G}(\omega) = 1 - \breve{H}(\omega) = 1 - \breve{F}(L\omega)$$

Interpolated FIR Filter

- The overall filter $H_{IFIR}(z)$ is designed as a cascade of a linear-phase FIR filter $F(z^L)$ and another filter $I(z)$ that suppresses the undesired passbands of the periodic filter section as shown below



- The widths of the transition band and the passband of the cascade are $\frac{1}{L}$–th of those of $F(z)$

Interpolated FIR Filter

- The cascaded structure is called the interpolated finite impulse response (IFIR) filter, as the missing impulse response samples of the periodic filter section are being interpolated by the filter section $I(z)$, called the interpolator
- As the filter $F(z)$ determines approximately the shape of the amplitude response of the IFIR filter, it is called a shaping filter

Interpolated FIR Filter

- **Design Steps –**
- IFIR specifications: passband edge ω_p, stopband edge ω_s, passband ripple δ_p, stopband ripple δ_s
- Shaping filter specifications:

passband edge $\omega_p^{(F)} = L\omega_p$

stopband edge $\omega_s^{(F)} = L\omega_s$

passband ripple $\delta_p^{(F)} = \delta_p/2$

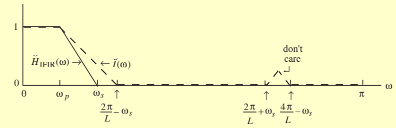
stopband ripple $\delta_s^{(F)} = \delta_s$

Interpolated FIR Filter

- The interpolator $I(z)$ has to be designed to preserve the passband of $F(z^L)$ in the frequency range $[0, \omega_p]$ and mask the amplitude response of $F(z^L)$ in the frequency range $[\omega_s, \pi]$, where the periodic subfilter has unwanted passbands and transition bands
- This latter region is defined by
$$R_\omega = \bigcup_{k=1}^{\lfloor L/2 \rfloor} \left[\frac{2\pi k}{L} - \omega_s, \min\left(\frac{2\pi k}{L} + \omega_s, \pi \right) \right]$$

Interpolated FIR Filter

- The transition band of the interpolator is the frequency range $\left[\omega_p, \frac{2\pi}{L} - \omega_s \right]$
- Figure below shows the amplitude responses of $H_{IFIR}(z)$ and $I(z)$



Interpolated FIR Filter

- Summarinzing, the design specifications for $F(z)$ and $I(z)$ are as follows:

$$1-\delta_p^{(F)} \le \breve{F}(\omega) \le 1+\delta_p^{(F)} \text{ for } \omega \in [0, L\omega_p]$$
$$-\delta_s^{(F)} \le \breve{F}(\omega) \le \delta_s^{(F)} \text{ for } \omega \in [L\omega_s, \pi]$$
$$1-\delta_p^{(I)} \le \breve{I}(\omega) \le 1+\delta_p^{(I)} \text{ for } \omega \in [0, \omega_p]$$
$$-\delta_s^{(I)} \le \breve{I}(\omega) \le \delta_s^{(I)} \text{ for } \omega \in R_\omega$$

The two linear-phase FIR filters $F(z)$ and $I(z)$ can be designed using the Parks-McClellan method

109

Interpolated FIR Filter

- **Example –** Filter specifications are as follows: $\omega_p = 0.15\pi$, $\omega_s = 0.2\pi$, $\delta_p = 0.002$, $\delta_s = 0.001$
- It follows from the figure in Slide 101 that to ensure no overlaps between adjacent passbands of $F(z^L)$, we should choose L to satisfy the condition

$$\frac{\omega_s^{(F)}}{L} < \frac{2\pi - \omega_s^{(F)}}{L}$$

110

Interpolated FIR Filter

- For our example, this reduces to

$$0.2\pi < \frac{2\pi}{L} - 0.2\pi$$

implying $L < 5$

- Hence, the largest value of L that can be used is $L = 4$, yielding an IFIR structure requiring the least number of multipliers
- As a result, the specifications for $F(z)$ and $I(z)$ are as given in the next slide

111

Interpolated FIR Filter

- $F(z)$: $\omega_p^{(F)} = 0.6\pi$, $\omega_s^{(F)} = 0.8\pi$
  $\delta_p^{(F)} = 0.001$, $\delta_s^{(F)} = 0.001$
- $I(z)$: $\omega_p^{(I)} = 0.15\pi$, $\omega_s^{(I)} = 0.3\pi$
  $\delta_p^{(I)} = 0.001$, $\delta_s^{(I)} = 0.001$
- The filter orders of $F(z)$ and $I(z)$ obtained using `remezord` are:

Order of $F(z)$ = 32

Order of $I(z)$ = 43

112

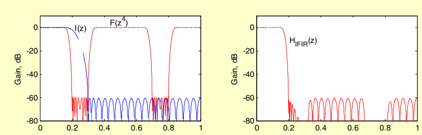
Interpolated FIR Filter

- It can be shown that the filters $F(z)$ and $I(z)$ designed using remez with the above orders do not lead to an IFIR design meeting the minimum stopband attenuation of 60 dB
- To meet the stopband specifications, the orders of $F(z)$ and $I(z)$ need to be increased to 33 and 46, respectively

113

Interpolated FIR Filter

- The pertinent gain responses of the redesigned IFIR filter are shown below:



- The number of multipliers needed to implement $F(z)$ and hence, $F(z^4)$ is

$$\mathcal{R}_F = \lceil (33+1)/2 \rceil = 17$$

114

Interpolated FIR Filter

- The number of multipliers needed to implement $I(z)$ is:

$$\mathcal{R}_I = \lceil (46+1)/2 \rceil = 24$$

- As a result, the total number of multipliers needed to implement $H_{IFIR}(z)$ is

$$\mathcal{R}_{IFIR} = 17 + 24 = 41$$

- The number of multipliers needed to implement the direct single-stage implementation of the FIR filter is

$$\lfloor (122+1)/2 \rfloor = 62$$

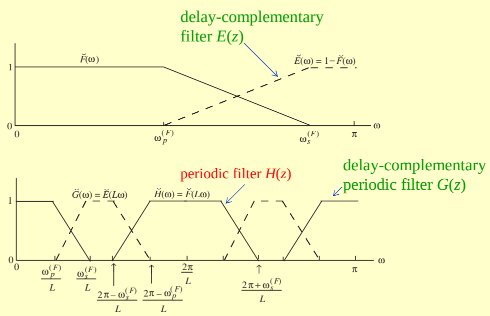
Frequency-Response Masking Approach

- This approach makes use of the relation between a periodic filter $H(z) = F(z^L)$ generated from a Type 1 linear-phase FIR filter of even degree N and its delay-complementary filter $G(z)$ given by

$$G(z) = z^{-N/2} - H(z) = z^{-N/2} - F(z^L)$$

- The amplitude responses of $F(z)$, its delay-complimentary filter $E(z)$, the periodic filter $H(z)$ and its delay-complimentary filter $G(z)$ are shown in the next slide

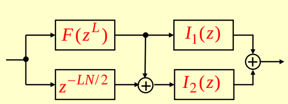
Frequency-Response Masking Approach



Frequency-Response Masking Approach

- By selectively masking out the unwanted pasbands of both $H(z)$ and $G(z)$ by cascading each with appropriate masking filters $I_1(z)$ and $I_2(z)$, respectively, and connecting the resulting cascades in prallel, we can design a large class of FIR filters with sharper transition bands
- The overall structure is then realized as indicated in the next slide

Frequency-Response Masking Approach



- Note: The delay block $z^{-NL/2}$ can be realized by tapping the FIR structure implementing $F(z^L)$
- Also, $I_1(z)$ and $I_2(z)$ can share the same delay-chain if they are realized using the transposed direct form structure

Frequency-Response Masking Approach

- The transfer function of the overall structure is given by

$$H_{FM}(z) = H(z)I_1(z) + G(z)I_2(z)$$
$$= F(z^L)I_1(z) + [z^{-NL/2} - F(z^L)]I_2(z)$$

- The corresponding amplitude response is

$$\breve{H}_{FM}(\omega) = \breve{F}(L\omega)\breve{I}_1(\omega) + [1 - \breve{F}(L\omega)]\breve{I}_2(\omega)$$

Frequency-Response Masking Approach

- The overall computational complexity is given by the complexities of $F(z)$, $I_1(z)$ and $I_2(z)$
- All these three filters have wide transition bands and, in general, require considerably fewer multipliers and adders than that required in a direct design of the desired sharp cutoff filter

121

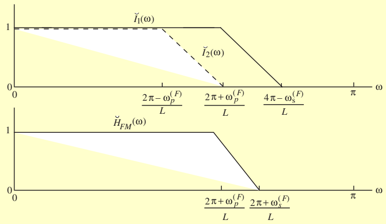
Frequency-Response Masking Approach

- Design Objective – Given the specifications of $H_{FM}(z)$, determine the specifications of $F(z)$, $I_1(z)$ and $I_2(z)$ design these 3 filters
- Design method – Illustrated for lowpass filter design
- Two different situations may arise depending on how the transition band of $H_{FM}(z)$ is created

122

Frequency-Response Masking Approach

- Case A – Transition band of $H_{FM}(z)$ is from one of the transition bands of $H(z)$



123

Frequency-Response Masking Approach

- Bandedges of $H_{FM}(z)$ are related to the bandedges of $F(z)$ as follows:

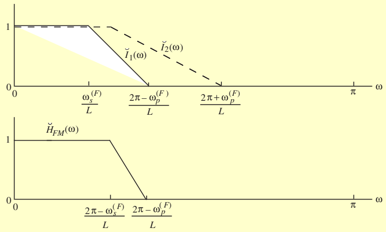
$$\omega_p = \frac{2\ell\pi + \omega_s^{(F)}}{L}, \quad \omega_s = \frac{2\ell\pi + \omega_p^{(F)}}{L},$$

$$0 \le \ell \le L-1$$

124

Frequency-Response Masking Approach

- Case B – Transition band of $H_{FM}(z)$ is from one of the transition bands of $G(z)$



125

Frequency-Response Masking Approach

- Bandedges of $H_{FM}(z)$ are related to the bandedges of $F(z)$ as follows:

$$\omega_p = \frac{2\ell\pi - \omega_p^{(F)}}{L}, \quad \omega_s = \frac{2\ell\pi - \omega_p^{(F)}}{L},$$

- Example – Specifications for a lowpass filter: $\omega_p = 0.4\pi$, $\omega_s = 0.402\pi$, $\delta_p = 0.01$, and $\delta_s = 0.0001$

126

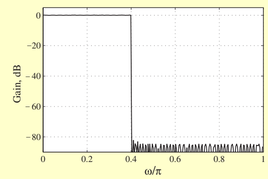
Frequency-Response Masking Approach

- For designing $H_{FM}(z)$ the optimum value of L is in the range
- By calculating the total number of multipliers needed to realize $F(z)$, $I_1(z)$, and $I_2(z)$ for all possible values of L, we arrive at the realization requiring the least number of multipliers obtained for $L = 16$ is 229 which is about 15% of that required in a direct single-stage realization

127

Frequency-Response Masking Approach

- The gain response of the designed filter is shown below:



128