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

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-phase FIR transfer function $G(z)$ with a non-negative amplitude response:

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

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 $G(z)$

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

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

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

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

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

Spectral Factorization

and

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

- The matrix **A** can be expressed as

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

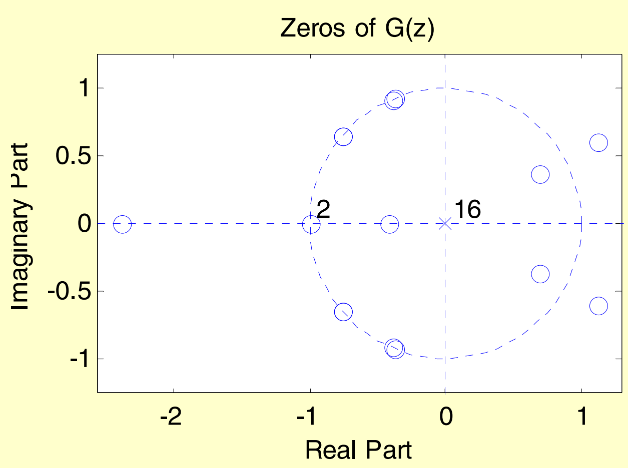
Spectral Factorization

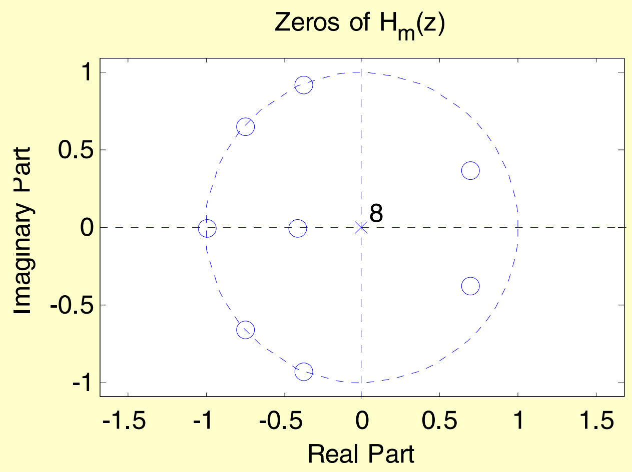
- The iteration convergence is checked at each step by evaluating the error term $\sum_{i=0}^{3} e_i^2$
- The error term first decreases monotonically and the iteration is stopped when the error starts increasing
- The M-file `minphase.m` implements the above spectral factorization method

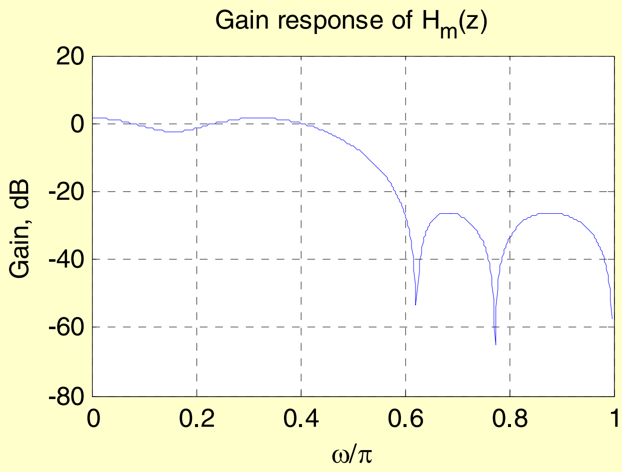
Minimum-Phase FIR Filter Design Using MATLAB

- **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

Minimum-Phase FIR Filter Design Using MATLAB







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)
```

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

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

Design of Computationally Efficient FIR Digital Filters

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

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

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

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

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

The Periodic Filter Section

- Note: The number of multiplers and adders 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 $2\pi/L$

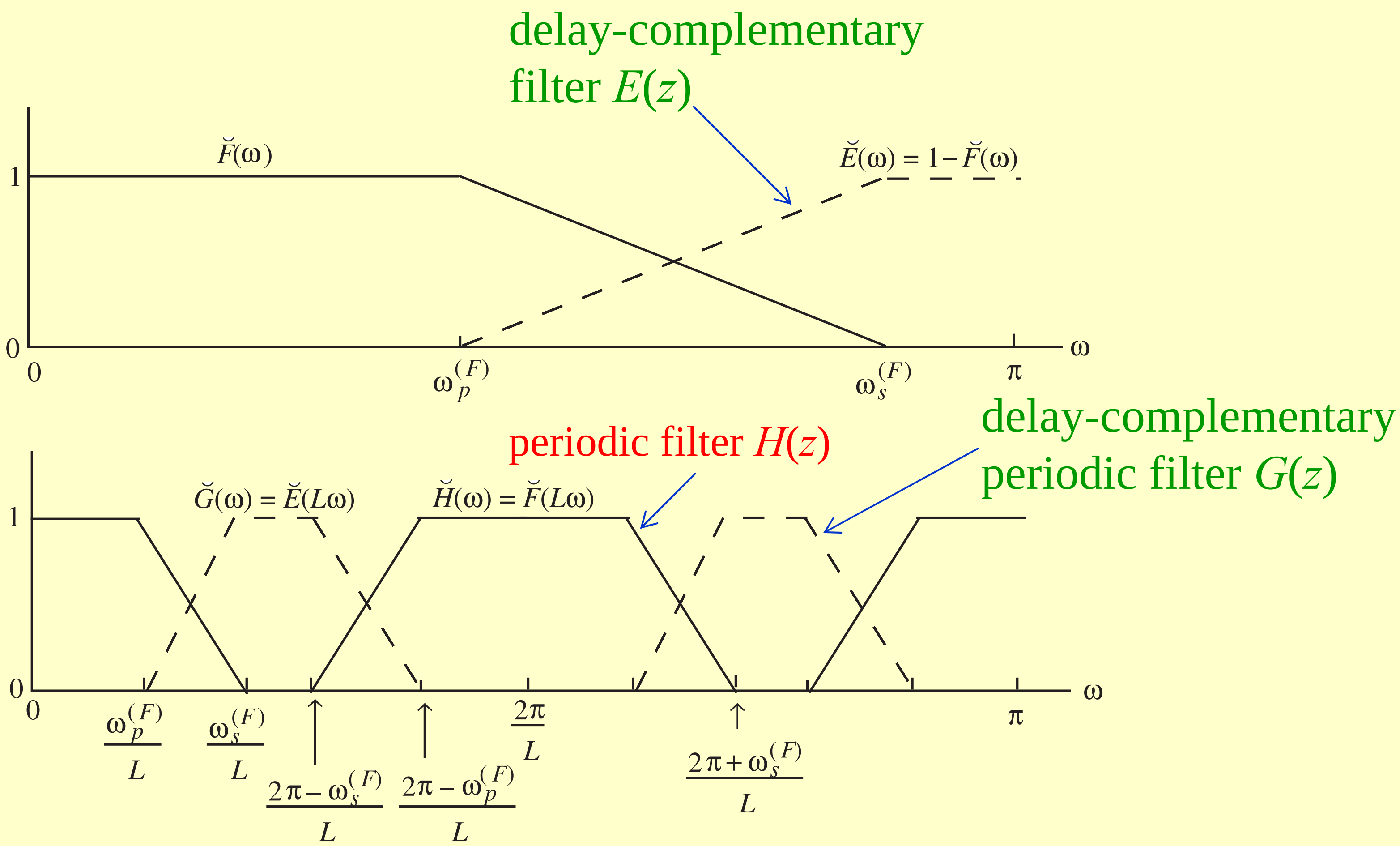
The Periodic Filter Section

- 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 $2\pi/L$ 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 $\omega_s^{(F)} / L$, respectively
- The passband and stopband edges of the second band of $H(z)$ are at $(2\pi \pm \omega_p^{(F)}) / L$ and $(2\pi \pm \omega_s^{(F)}) / L$, 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

$$G(z) = E(z^L) = z^{-NL/2} - F(z^L)$$

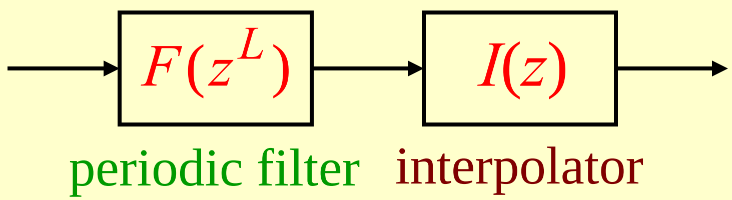
$$= z^{-NL/2} - \sum_{n=0}^{N} f[n] z^{-nL}$$

- 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 suppreses 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 $\delta_s^{(F)} = \delta_s$

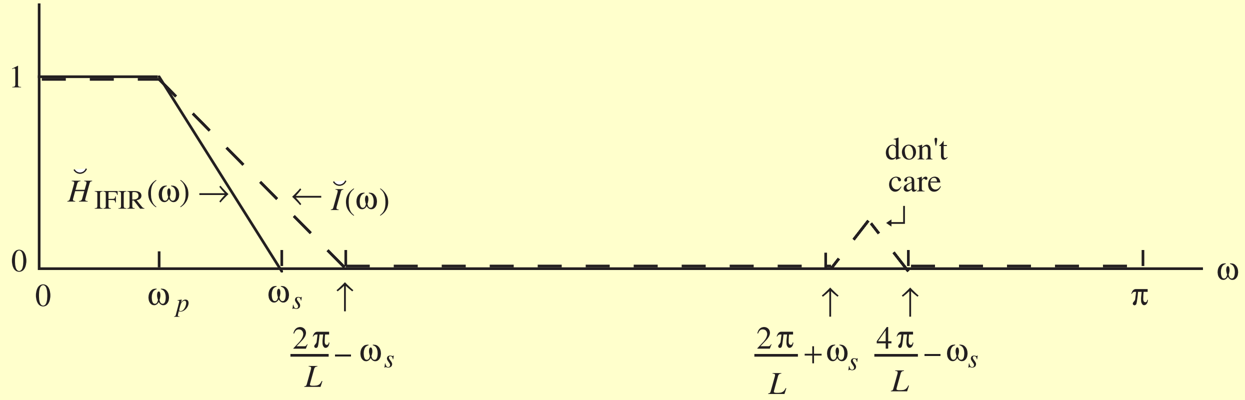
Interpolated FIR Filter

- 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:

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

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

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

$$-\delta_s^{(I)} \le \breve{I}(\omega) \le \delta_s^{(I)} \quad \text{for} \quad \omega \in R_\omega$$

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

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 22 that to ensure no overlaps between adjacent passbands of $F(z^L)$, we should choose L to satisfy the condition

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

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

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 `firpmord` are:

  Order of $F(z)$ = 32

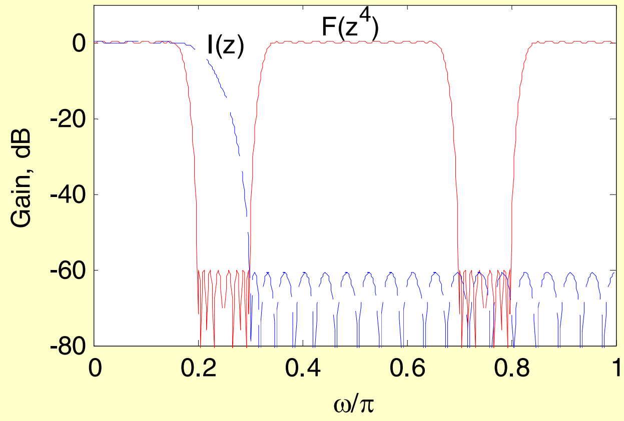
  Order of $I(z)$ = 43

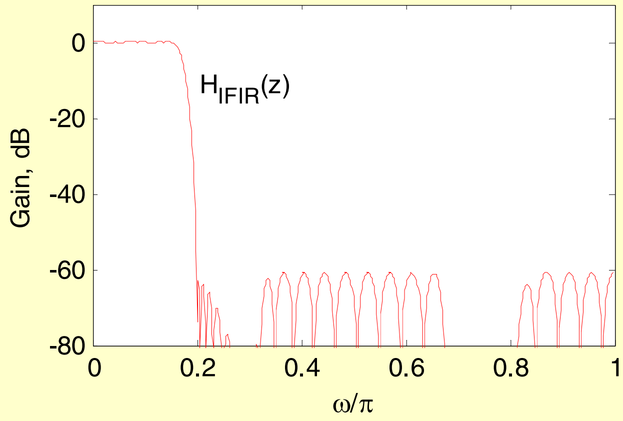
Interpolated FIR Filter

- 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

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

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

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

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

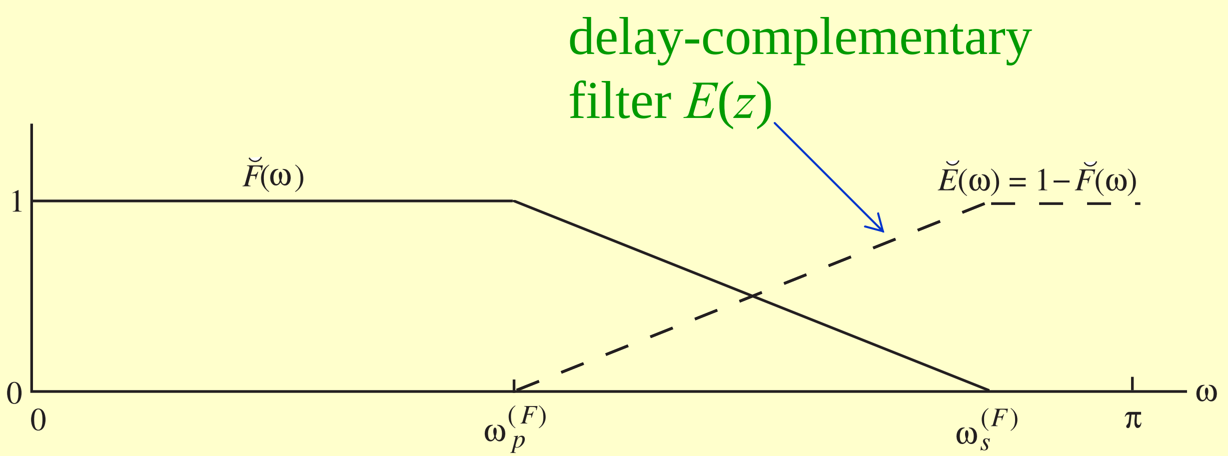
Frequency-Response Masking Approach

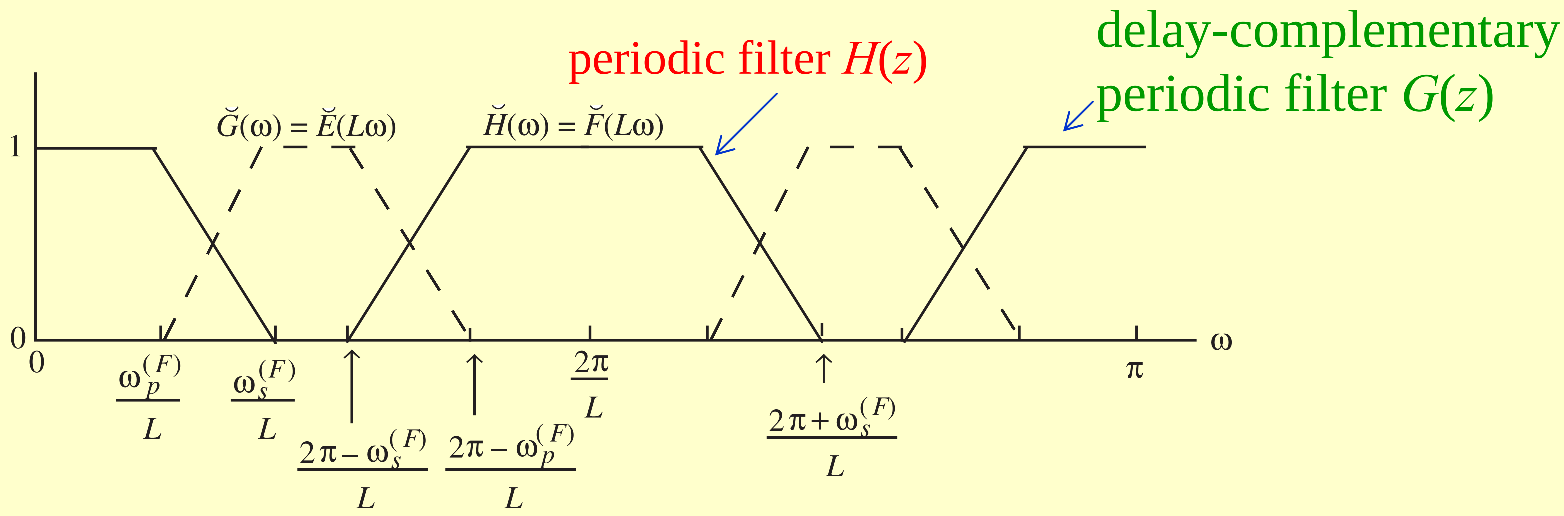
- This approach makes use of the relation between a periodic filter $H(z) = F(z^L)$ generated from a Type 1 linear-phase FIR filter of even degree N and its delay-complementary filter $G(z)$ given by

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

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

Frequency-Response Masking Approach

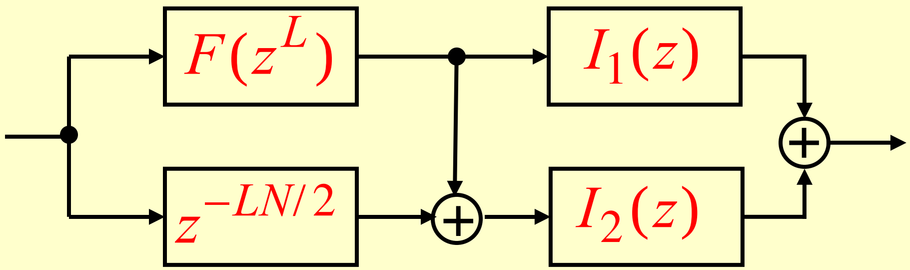




Frequency-Response Masking Approach

- 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

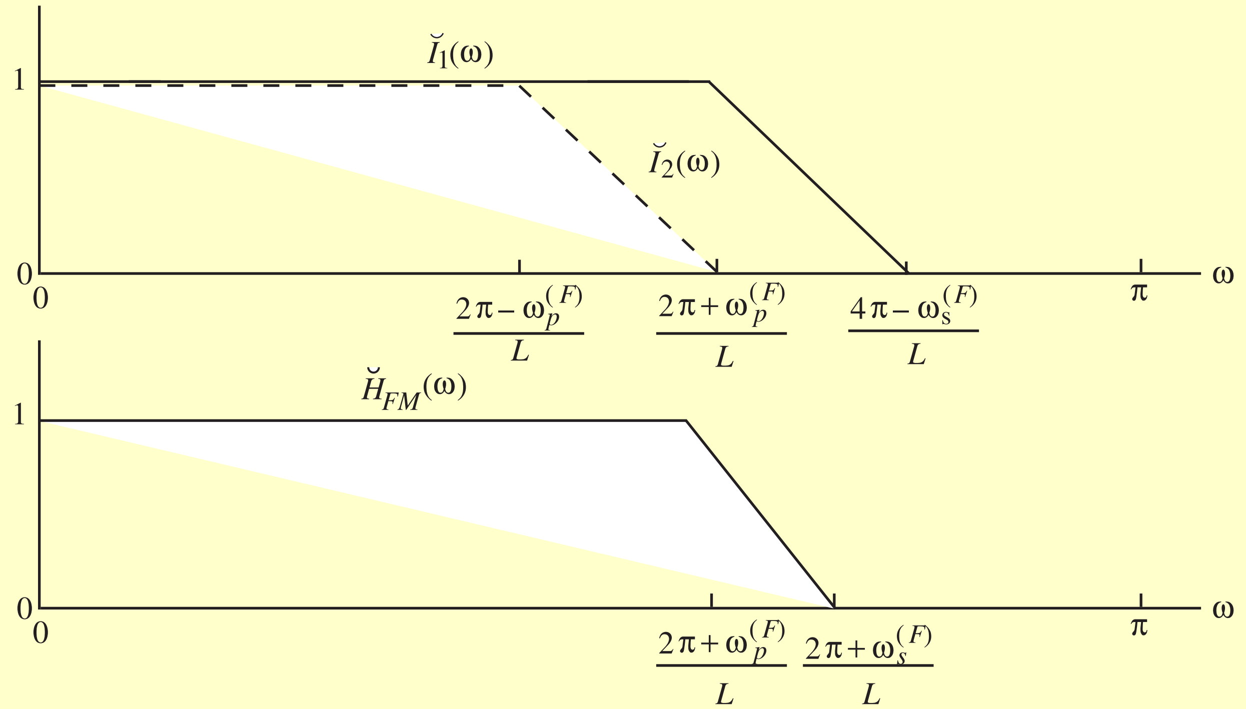
- The overall computational complexity is given by the complexities of $F(z)$, $I_1(z)$ and $I_2(z)$
- All these three filters have wide transition bands and, in general, require considerably fewer multipliers and adders than that required in a direct design of the desired sharp cutoff filter

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

Frequency-Response Masking Approach

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



Frequency-Response Masking Approach

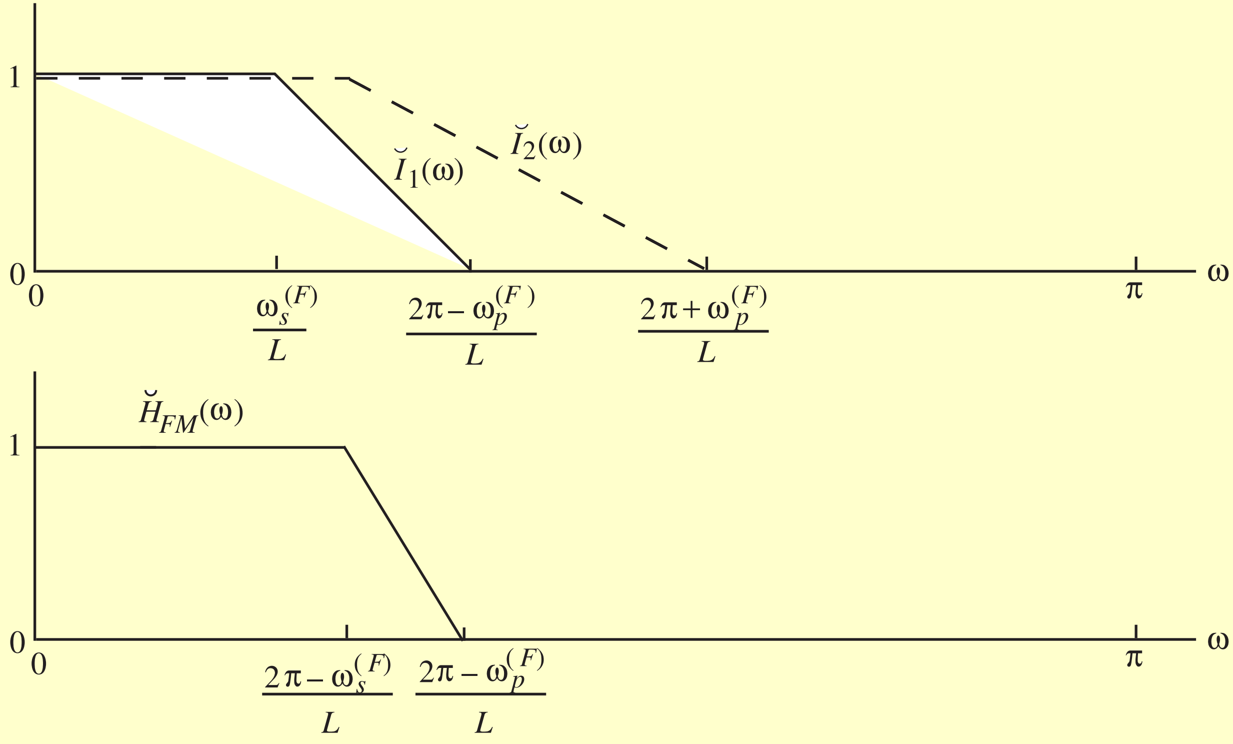
- Bandedges of $H_{FM}(z)$ are related to the bandedges of $F(z)$ as follows:

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

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

Frequency-Response Masking Approach

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



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 $\delta_s = 0.0001$

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

Frequency-Response Masking Approach

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