Design of Minimum-Phase FIR Filters

1. 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.

2. 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.

Design of Minimum-Phase FIR Filters

3. A very simple method of minimum-phase FIR filter is described next.

4. Consider an arbitrary FIR transfer function of degree $N$:

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

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

6. 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

7. Thus, unit circle zeros of $G(z)$ occur in pairs.

8. On the unit circle we have

$$G(e^{j\omega}) = H(e^{j\omega})^2 = G(\omega)$$

$G(\omega) \geq 0$

9. Moreover, the amplitude response $\tilde{G}(\omega)$ has double zeros in the frequency range $[0, \pi]$.
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
    \[ 1 + \delta_s^{(F)} - \delta_p^{(F)} \leq \tilde{G}(\omega) \leq 1 + \delta_s^{(F)} + \delta_p^{(F)} \]
    for \( \omega \in [0, \omega_p] \)
    \[ 0 \leq G(\omega) \leq 2\delta_s^{(F)} \]
    for \( \omega \in [\omega_s, \pi] \)

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 \( \delta_p \) and the stopband ripple \( \delta_s \) of \( H_m(z) \) are satisfied

FIR Digital Filter Design Using MATLAB

• **Order Estimation**
  • Kaiser’s Formula:
    \[ N \approx -20\log_{10}(\sqrt[2]{\delta_p \delta_s}) \]
    \[ = \frac{14.6(\omega_s - \omega_p)}{2\pi} \]
  • **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

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

• It can be shown
  \[ \delta_p^{(F)} = \sqrt{\frac{1 + \delta_p}{1 - \delta_s}} - 1, \quad \delta_s^{(F)} = \sqrt{\frac{2\delta_s}{1 - \delta_s}} \]
  • 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

• Hermann-Rabiner-Chan’s Formula:
  \[ N \approx D_s(\delta_p, \delta_s) - F(\delta_p, \delta_s) \left[ (\omega_s - \omega_p) / 2\pi \right]^2 \]
  where
  \[ D_s(\delta_p, \delta_s) = [a_1 \log_{10} \delta_p] \]
  \[ + [a_2 \log_{10} \delta_p] \]
  \[ + [a_3 \log_{10} \delta_p] \]
  \[ + [a_4 (\log_{10} \delta_p)^2] \]
  \[ + [a_5 \log_{10} \delta_s] \]
  \[ + [a_6 (\log_{10} \delta_s)^2] \]
  \[ + [a_7 \log_{10} \delta_s] \]
  \[ F(\delta_p, \delta_s) = b_1 + b_2 (\log_{10} \delta_p - \log_{10} \delta_s) \]
  with
  \[ a_1 = 0.06536, \quad a_2 = 0.07114, \quad a_3 = -0.4761 \]
  \[ a_4 = 0.000166, \quad a_5 = 0.001244, \quad a_6 = 0.4278 \]
  \[ b_1 = 11.01217, \quad b_2 = 0.51244 \]
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 $\delta_p$ and $\delta_s$
• 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

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 `firpmord` 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

MATLAB code fragments used are

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

Equiripple FIR Digital Filter Design Using MATLAB

• The M-file `firpm` 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 $\delta_s = 0.01$

MATLAB code fragments used are

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

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 $N = 30$
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:
  \[ N = 26 \]
  \[ fpts = [0 0.25 0.3 0.5 0.55 1] \]
  \[ mag = [0 0 1 1 0 0] \]
  \[ wt = [1 1 1] \]

Computed gain response shown below where \[ \alpha_p = 1 \text{ dB}, \ \alpha_s = 18.7 \text{ dB} \]

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

\[ \alpha_s \] can be increased at the expense of a larger \[ \alpha_p \] 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 \[ \alpha_p = 0.076 \text{ dB}, \ \alpha_s = 60.86 \text{ dB} \]

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.

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.
**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:
    
    \[
    \begin{align*}
    N &= 60 \\
    \text{fpts} &= [0 \ 0.25 \ 0.3 \ 0.5 \ 0.6 \ 1] \\
    \text{mag} &= [0 \ 0 \ 1 \ 1 \ 0 \ 0] \\
    \text{wt} &= [1 \ 1 \ 0.3]
    \end{align*}
    \]

- Plots of gain response and absolute error shown below

- 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

**Equiripple FIR Differentiator Design Using MATLAB**

- A lowpass differentiator has a bandlimited frequency response

\[
H_{DIF}(e^{j\omega}) = \begin{cases} 
  j\omega, & 0 \leq |\omega| \leq \omega_p \\
  0, & \omega_p < |\omega| \leq \pi
\end{cases}
\]

where \( 0 \leq |\omega| \leq \omega_p \) represents the passband and \( \omega_p \leq |\omega| \leq \pi \) represents the stopband
- For the design phase we choose

\[
W(\omega) = 1/\omega, \quad D(\omega) = 1, \quad 0 \leq |\omega| \leq \omega_p
\]

- The M-file `firpmord` 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

\[
b = \text{firpm}(N, \text{fpts}, \text{mag}, \text{’differentiator’});
\]

where

\[
N = 11 \\
\text{fpts} = [0 \ 1] \\
\text{mag} = [0 \ \pi]
\]
Equiripple FIR Differentiator Design Using MATLAB

- Plots of magnitude response and absolute error

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

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
  
  ```matlab
  b = firpm(N,fpts,mag,'differentiator');
  ```
  
  where
  
  ```matlab
  N = 50
  fpts = [0 0.4 0.45 1]
  mag = [0 0.4*pi 0 0]
  ```

Equiripple FIR Hilbert Transformer Design Using MATLAB

- Plot of the magnitude response of the lowpass differentiator

Equiripple FIR Hilbert Transformer Design Using MATLAB

- Desired amplitude response of a bandpass Hilbert transformer is $D(\omega) = 1$, $\omega_L \leq |\omega| \leq \omega_H$
- with weighting function $W(\omega) = 1$, $\omega_L \leq |\omega| \leq \omega_H$
- Impulse response of an ideal Hilbert transformer satisfies the condition $h_{HT}[n] = 0$, for $n$ even
- which can be met by a Type 3 FIR filter

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
  
  ```matlab
  b = firpm(N,fpts,mag,'Hilbert');
  ```
  
  where
  
  ```matlab
  N = 20
  fpts = [0.1 0.9]
  mag = [1 1]
  ```
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

- **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);
    w = kaiser(N+1, beta);
    b = fir1(N, Wn, kaiser(N+1, beta));
    ```

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

- **Plot of gain response**

  ![Gain Response of Kaiser Window](image1)

  ![Lowpass Filter Designed Using Kaiser Window](image2)
Example - Design using a Kaiser window a highpass FIR filter with the specifications:
\[ \omega_p = 0.55\pi, \quad \omega_s = 0.4\pi, \quad \delta_s = 0.02 \]

Code fragments to use

- \( [N, Wn, beta, ftype] = \text{kaiserord}(fpts, mag, dev); \)
- \( b = \text{fir1}(N, Wn, \text{ftype}, \text{kaiser}(N+1, beta)); \)

where
- \( fpts = [0.4 \quad 0.55] \)
- \( \text{mag} = [0 \quad 1] \)
- \( \text{dev} = [0.02 \quad 0.02] \)

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 \([0.52, 1.0]\)

Code fragment to use

- \( b = \text{fir2}(100, fpts, mval); \)
- where \( fpts = [0 \quad 0.28 \quad 0.3 \quad 0.5 \quad 0.52 \quad 1]; \)
- \( mval = [0.3 \quad 0.3 \quad 1.0 \quad 1.0 \quad 0.7 \quad 0.7]; \)