Chapter 10: Design of Digital Filters

With an understanding of sampling, rudimentary filter operations in the z domain, the Fourier series and transform the topic of filter design can be addresses.

Filters to discuss

- finite impulse response
- infinite impulse response

Important characteristics: order/length/completeness, magnitude response, phase response, group delay, filter region or type, passband/transition band/stopband,

**Finite Impulse Response Filters**

- all zeros
- linear phase if filter is symmetric or antisymmetric
- relatively high order
- simple implementation with delay elements, coefficient multiplier and sum
- the FIR coefficients in sample time are the same as the z-power coefficients, therefore, the impulse response reveals the filter coefficients
- special characteristics for symmetric and anti-symmetric coefficients and even versus odd lengths
- The “impulse invariance” method allows FIR filters to match the impulse response of analog filters

\[
H(z) = \sum_{k=0}^{M} b_k \cdot z^{-k}
\]

\[
H(z) = B(z) = b_0 + b_1 \cdot z^{-1} + \cdots + b_{M-1} \cdot z^{-(M-1)} + b_M \cdot z^{-M}
\]

\[
|H(w)| = |b_0| \cdot \prod_{k=1}^{M} \left| e^{jw} - z_k \right|
\]

\[
\angle H(w) = \Theta(w) = \angle b_0 + w \cdot (-M) + \sum_{k=1}^{M} \angle \left( e^{jw} - z_k \right)
\]

\[
\tau_g(w) = -\frac{d\theta(w)}{dw} = -\text{Re} \left[ z \cdot \frac{d[\ln(H(z))]}{dz} \right]_{z=e^{jw}}
\]
**Infinite Impulse Response Filters**

- zeros and poles
- significantly lower order than FIR
- feedback implementation
- methods to translate from analog Laplace-domain to digital z-domain filters can be used

\[
H(z) = \frac{\sum_{k=0}^{M} b_k \cdot z^{-k}}{\sum_{k=0}^{N} a_k \cdot z^{-k}}
\]

\[
H(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1 \cdot z^{-1} + \cdots + b_{M-1} \cdot z^{-(M-1)} + b_M \cdot z^{-M}}{a_0 + a_1 \cdot z^{-1} + \cdots + a_{N-1} \cdot z^{-(N-1)} + a_N \cdot z^{-N}}
\]

Typically,
\[
a_0 = 1
\]

\[
|H(z)|_{z=e^{j\omega}} = \left| \frac{p_0}{d_0} \right| \cdot |z^{(N-M)}| \cdot \prod_{k=1}^{M} \left| \frac{z - \xi_k}{z - \sigma_k} \right| = \left| \frac{p_0}{d_0} \right| \prod_{k=1}^{M} \left| \frac{e^{j\omega} - \xi_k}{e^{j\omega} - \sigma_k} \right|
\]

\[
\angle H(z)_{z=e^{j\omega}} = \angle \left( \frac{p_0}{d_0} \right) + \omega \cdot (N - M) + \sum_{k=1}^{M} \angle (e^{j\omega} - \xi_k) - \sum_{k=1}^{N} \angle (z - \lambda_k)
\]

\[
\tau_g(w) = -\frac{d\theta(w)}{dw} = -\text{Re} \left[ z \cdot \frac{d[\ln(H(z))]}{dz} \right]_{z=e^{j\omega}}
\]
**Textbook General Considerations**

**Causality and Its Implications**

The necessary and sufficient conditions that a frequency response characteristic must satisfy in order to be causal are given by the Paley-Wiener theorem.

Paley-Wiener Theorem (simplified): If \( h(n) \) has finite energy and \( h(n) = 0 \) for \( n < 0 \), then

\[
\int_{-\pi}^{\pi} \ln(|H(w)|) \cdot dw < \infty
\]

Conversely, if \( |H(w)| \) is square integrable and if the above integral is finite, then we can associate with \( |H(w)| \) a phase response \( \Theta(w) \), so that the resulting filter with frequency response

\[
H(w) = |H(w)| \cdot \exp(j \cdot \Theta(w))
\]

is causal.

*As a consequence, any ideal filter is non-causal!*

Interesting properties of the causal filter:

\( h(n) = 0, \quad \text{for all } n < 0 \)

Therefore we can decompose the filter into even and odd parts over all time as

\[
h_e(n) = \frac{1}{2} [h(n) + h(-n)], \quad \text{for } -\infty < n < \infty
\]

\[
h_o(n) = \frac{1}{2} [h(n) - h(-n)], \quad \text{for } -\infty < n < \infty
\]

Note:

\[
h_e(n) = h_o(n) = \frac{1}{2} [h(n)], \quad \text{for } 1 < n < \infty
\]

We can also describe filter in terms of the even and odd parts as

\[
h(n) = 2 \cdot h_e(n) \cdot u(n) - h_e(0) \cdot \delta(n), \quad \text{for } 0 \leq n
\]

\[
h(n) = 2 \cdot h_o(n) \cdot u(n) + h(0) \cdot \delta(n), \quad \text{for } 0 \leq n
\]
The Fourier transform exists

\[ H(w) = H_r(w) + j \cdot H_i(w) \]

Based on the properties of even and odd sequences,

\[ h_e(n) \overset{F}{\leftrightarrow} H_R(w) \]

\[ h_o(n) \overset{F}{\leftrightarrow} H_I(w) \]

But since \( h(n) \) is completely described by the even (or odd) function, the real and imaginary spectra are “inter-dependent” and cannot be defined independently!

This also relates to the inter-dependence of the magnitude and phase!

To develop the inter-dependence take the Fourier transform of

\[ h(n) = 2 \cdot h_e(n) \cdot u(n) - h_e(0) \cdot \delta(n) \]

\[ H(w) = H_R(w) + j \cdot H_I(w) = \frac{1}{2 \cdot \pi} \cdot \int_{-\pi}^{\pi} \left[ 2 \cdot h_e(n) \cdot u(n) - h_e(0) \cdot \delta(n) \right] \cdot \exp(j \cdot w \cdot n) \cdot dw \]

\[ H(w) = H_R(w) + j \cdot H_I(w) = \frac{1}{\pi} \cdot \int_{-\pi}^{\pi} [h_e(n) \cdot u(n)] \cdot \exp(j \cdot w \cdot n) \cdot dw - h_e(0) \]

\[ H(w) = H_R(w) + j \cdot H_I(w) = \frac{1}{\pi} \cdot \int_{-\pi}^{\pi} [H_R(\lambda) \cdot U(w - \lambda)] \cdot d\lambda - h_e(0) \]

But

\[ U(w) = \pi \cdot \delta(w) + \frac{1}{1 - \exp(-j \cdot w)} = \pi \cdot \delta(w) + \frac{1 - \exp(j \cdot w)}{2 \cdot (1 - \cos(w))} \]

\[ U(w) = \pi \cdot \delta(w) + \frac{1}{1 - \exp(-j \cdot w)} = \pi \cdot \delta(w) + \frac{1 - \cos(w) - j \cdot \sin(w)}{1 - \cos(w)} \]

\[ U(w) = \pi \cdot \delta(w) + \frac{1 - j \cdot \frac{1}{2} \left( \frac{\sin(w)}{1 - \cos(w)} \right)}{2} = \pi \cdot \delta(w) + \frac{1}{2} - j \cdot \frac{1}{2} \cdot \cot \left( \frac{w}{2} \right) \]

Therefore

\[ H(w) = H_R(w) + j \cdot H_I(w) = \frac{1}{\pi} \cdot \int_{-\pi}^{\pi} [H_R(\lambda) \cdot \left( \pi \cdot \delta(w - \lambda) + \frac{1}{2} - j \cdot \frac{1}{2} \cdot \cot \left( \frac{w - \lambda}{2} \right) \right)] \cdot d\lambda - h_e(0) \]
$$H_R(w) + j \cdot H_I(w) = H_R(w) + \frac{1}{2 \cdot \pi} \int_{-\pi}^{\pi} H_R(\lambda) \cdot d\lambda - h_e(0) - \frac{j}{2 \cdot \pi} \int_{-\pi}^{\pi} H_R(\lambda) \cdot \cot \left( \frac{w-\lambda}{2} \right) \cdot d\lambda$$

Removing similar terms and recognizing that

$$h_e(0) = \frac{1}{2 \cdot \pi} \int_{-\pi}^{\pi} H_R(\lambda) \cdot d\lambda$$

What remains is

$$H_I(w) = -\frac{1}{2 \cdot \pi} \int_{-\pi}^{\pi} H_R(\lambda) \cdot \cot \left( \frac{w-\lambda}{2} \right) \cdot d\lambda$$

This relationship defines the imaginary part based on the real part alone. This is also known as the “Hilbert Transform”!

The Hilbert transform is usually thought to allow us to describe the power spectrum as the positive only portion of the frequency spectrum.

Note that the above process can be repeated for $h(n) = 2 \cdot h_o(n) \cdot u(n) + h(0) \cdot \delta(n)$

$$H(w) = H_R(w) + j \cdot H_I(w) = \frac{1}{2 \cdot \pi} \int_{-\pi}^{\pi} \left[ 2 \cdot h_o(n) \cdot u(n) + h(0) \cdot \delta(n) \right] \cdot \exp(j \cdot w \cdot n) \cdot dw$$

$$H(w) = H_R(w) + j \cdot H_I(w) = h(0) + \frac{2}{2 \cdot \pi} \int_{-\pi}^{\pi} \left[ j \cdot H_I(\lambda) \cdot U(w-\lambda) \right] \cdot \exp(j \cdot w \cdot n) \cdot dw$$

$$H_R(w) + j \cdot H_I(w) = h(0) + \frac{2}{2 \cdot \pi} \int_{-\pi}^{\pi} j \cdot H_I(\lambda) \cdot \left[ \pi \cdot \delta(w-\lambda) + \frac{1}{2} - \frac{1}{2} \cdot \cot \left( \frac{w-\lambda}{2} \right) \right] \cdot d\lambda$$

$$H_R(w) + j \cdot H_I(w) = h(0) + \frac{j}{2 \cdot \pi} \int_{-\pi}^{\pi} H_I(\lambda) \cdot d\lambda + \frac{1}{2 \cdot \pi} \int_{-\pi}^{\pi} H_I(\lambda) \cdot \cot \left( \frac{w-\lambda}{2} \right) \cdot d\lambda$$

The integral of the anti-symmetric spectrum becomes zero and

$$H_R(w) = h(0) + \frac{1}{2 \cdot \pi} \int_{-\pi}^{\pi} H_I(\lambda) \cdot \cot \left( \frac{w-\lambda}{2} \right) \cdot d\lambda$$

(Please check my math.)
Characteristics of Practical Frequency-Selective Filters

Properties that must be defined in order to design a filter:

- Passband band edge
- Passband Ripple
- Transition band
- Stopband band edge
- Stopband Ripple

**Figure 10.1.2** Magnitude characteristics of physically realizable filters.
10.2 Design of FIR – Symmetric and Anti-symmetric Filters.

10.2.1 Zero-Phase Transfer Function, FIR Implementation (Ch07(1) p. 34-40)

For a zero phase, we want
\[ F(e^{j\omega}) = H(e^{j\omega}) \cdot H(e^{-j\omega}) = H(z) \cdot H(z^{-1}) \]

The resulting filter is a function in magnitude, but by employing conjugate phase sections, the phase is cancelled. The Matlab program `filtfilt` will perform this operation for any base filter \( H(z) \), but what are the coefficients?

Let
\[ H(z) = h_0 + h_1 \cdot z^{-1} + \cdots + h_{M-1} \cdot z^{-(M-1)} + h_M \cdot z^{-M} \]

then
\[ H(z^{-1}) = h_0 + h_1 \cdot z + \cdots + h_{M-1} \cdot z^{(M-1)} + h_M \cdot z^M \]

and
\[
F(z) = H(z) \cdot H(z^{-1}) = \begin{bmatrix}
    z^M \cdot (h_0 \cdot h_M) \\
    z^{(M-1)} \cdot (h_0 \cdot h_{M-1} + h_1 \cdot h_1) \\
    \vdots \\
    z^{-(M-1)} \cdot (h_0 \cdot h_{M-1} + h_1 \cdot h_M) \\
    z^{-M} \cdot (h_0 \cdot h_M)
\end{bmatrix}
\begin{bmatrix}
    f_M \cdot z^M \\
    f_{M-1} \cdot z^{(M-1)} \\
    \vdots \\
    f_1 \cdot z \cdot z^0 \\
    f_0 \cdot z^0
\end{bmatrix}
\]

Notice the symmetry of the coefficients. As a result, the filter can be expressed as
\[ F(z) = \sum_{k=0}^{M} f_k \cdot (z^k + z^{-k}) \]

\[ F(w) = \sum_{k=0}^{M} f_k \cdot (e^{jkw} + e^{-jkw}) = 2 \cdot f_0 + 2 \cdot \sum_{k=1}^{M} f_k \cdot \cos(k \cdot w) \]

Properties:
(1) Symmetric coefficients (note that this holds for even or odd length filters, odd derived above)
(2) It is inherently non-causal.
(3) This derivation is for FIR filters, an all zeros filter without any poles!
Textbook even length derivation

\[ H(z) = \sum_{k=0}^{M-1} h(k) \cdot z^{-k} \]

Let \[ h(n) = \pm h(M - 1 - n), \] for \( n = 0, 1, \ldots, M - 1 \)

Then for even \( M \)

\[ H(z) = \sum_{k=0}^{M-1} h(k) \cdot z^{-k} + \sum_{k=\frac{M}{2}}^{M-1} h(k) \cdot z^{-k} \]

\[ H(z) = \sum_{k=0}^{\frac{M}{2}-1} h(k) \cdot z^{-k} + \sum_{k=0}^{\frac{M}{2}-1} h(M - 1 - k) \cdot z^{-(M-1-k)} \]

But from symmetry

\[ H(z) = \sum_{k=0}^{\frac{M}{2}-1} h(k) \cdot z^{-k} - \sum_{k=0}^{\frac{M}{2}-1} h(k) \cdot z^{-(M-1-k)} = \sum_{k=0}^{\frac{M}{2}-1} h(k) \cdot (z^{-k} \pm z^{-(M-1-k)}) \]

\[ H(z) = z^{\left(\frac{M-1}{2}\right)} \cdot \sum_{k=0}^{\frac{M}{2}-1} h(k) \cdot z^{\left(\frac{M-1}{2}\right) - k} \pm z^{\left(\frac{M-1}{2}\right) + k} \]

Notice that the addition of \( z \) will be a sin or cos based on the anti-symmetry or symmetry of the coefficients.

Also note that

\[ H(z) = z^{-(M-1)} \cdot H\left(z^{-1}\right) \]

Then for odd-length \( M \) the “center coefficient is unique/unpaired” and therefore

\[ H(z) = z^{\left(\frac{M-1}{2}\right)} \cdot \left\{ h\left(\frac{M-1}{2}\right) + \sum_{k=0}^{\frac{M}{2}-1} h(k) \cdot z^{\left(\frac{M-1}{2}\right) - k} \pm z^{\left(\frac{M-1}{2}\right) + k} \right\} \]
See MATLAB Filter Design and Implementation for FIR filters.

<table>
<thead>
<tr>
<th>Filter Method</th>
<th>Description</th>
<th>Filter Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windowing</td>
<td>Apply window to truncated inverse Fourier transform of desired “brick wall” filter</td>
<td>fir1, fir2, kaiserord</td>
</tr>
<tr>
<td>Multiband with Transition Bands</td>
<td>Equiripple or least squares approach over sub-bands of the frequency range</td>
<td>firls, firpm, firpmord</td>
</tr>
<tr>
<td>Constrained Least Squares</td>
<td>Minimize squared integral error over entire frequency range subject to maximum error constraints</td>
<td>fircls, fircls1</td>
</tr>
<tr>
<td>Arbitrary Response</td>
<td>Arbitrary responses, including nonlinear phase and complex filters</td>
<td>cfirpm</td>
</tr>
<tr>
<td>Raised Cosine</td>
<td>Lowpass response with smooth, sinusoidal transition</td>
<td>firrcos</td>
</tr>
</tbody>
</table>

See the Mitra Notes “Ch10_FIR_Notes” for a complete discussion of symmetric/anti-symmetric properties. Complimentary, translated and complimentary-translated FIR filters are also presented.
10.2.2 Design of Linear-Phase FIR Filters Using Windows

Discrete-time, continuous frequency Fourier transform

\[ X(w) = \sum_{n=-\infty}^{\infty} x(n) \cdot \exp(- j \cdot w \cdot n) \]

\[ x(n) = \frac{1}{2\pi} \cdot \int_{-\pi}^{\pi} X(w) \cdot \exp(j \cdot w \cdot n) \cdot dw \]

(1) Define the desired ideal filter bandwidth. Example symmetric low pass filter.

\[ H_d(w) = \begin{cases} 1 \cdot \exp\left(- j \cdot w \cdot \left(\frac{M - 1}{2}\right)\right), & 0 \leq |w| \leq w_c \\ 0, & w_c < |w| \leq \pi \end{cases} \]

A unit magnitude filter with a linear phase delay for the time offset due to M, the length.

\[ h_d(n) = \frac{w_c}{\pi} \cdot \frac{\sin\left(w_c \cdot \left(n - \frac{M - 1}{2}\right)\right)}{w_c \cdot \left(n - \frac{M - 1}{2}\right)} \]

\[ h_d(n) = \frac{w_c}{\pi} \cdot \text{sinc}\left(w_c \cdot \left(n - \frac{M - 1}{2}\right)\right) \]

(2) Select a window, w(n) to apply to the ideal filter

\[ h(n) = h_d(n) \cdot w(n), \quad \text{for } 0 \leq n < M \]

Key value to estimate, how big should M be? The following tradeoffs or design iterations can tell

- The width of the transition band.
- How much passband you can afford to lose.
- The amount of acceptable ripple.

For the window selected, these values are related to the relative window peak width and the shape of the “window spectrum” that will be convolved with the ideal window.

See WindowTest2.m

And review Harris: Use of windows for harmonic analysis.
For bandpass filters, consider the problem from exam 1.

\[ X(w) = \begin{cases} 
1, & \frac{w_0 - \delta w}{2} \leq |w| \leq \frac{w_0 + \delta w}{2} \\
0, & \text{elsewhere}
\end{cases} \]

\[ x(n) = \frac{1}{2\pi} \cdot \int_{-w_0-\frac{\delta w}{2}}^{w_0+\frac{\delta w}{2}} \exp(j \cdot w \cdot n) \cdot dw + \frac{1}{2\pi} \cdot \int_{-w_0-\frac{\delta w}{2}}^{w_0+\frac{\delta w}{2}} \exp(j \cdot w \cdot n) \cdot dw \]

\[ x(n) = \frac{1}{2\pi} \cdot \frac{\exp\left(-j \cdot \left(w_0 - \frac{\delta w}{2}\right) \cdot n\right)}{j \cdot n} - \frac{\exp\left(-j \cdot \left(w_0 + \frac{\delta w}{2}\right) \cdot n\right)}{j \cdot n} \]

\[ + \frac{1}{2\pi} \cdot \left( \frac{\exp\left(j \cdot \left(w_0 + \frac{\delta w}{2}\right) \cdot n\right)}{j \cdot n} - \frac{\exp\left(j \cdot \left(w_0 - \frac{\delta w}{2}\right) \cdot n\right)}{j \cdot n} \right) \]

\[ x(n) = \frac{1}{2\pi} \cdot \left( \frac{\exp\left(j \cdot \frac{\delta w}{2} \cdot n\right)}{j \cdot n} - \frac{\exp\left(-j \cdot \frac{\delta w}{2} \cdot n\right)}{j \cdot n} \right) \cdot \exp(-j \cdot w_0 \cdot n) + \exp(j \cdot w_0 \cdot n) \]

\[ x(n) = \frac{1}{\pi \cdot n} \cdot \sin\left(\frac{\delta w}{2} \cdot n\right) \cdot 2 \cdot \cos(w_0 \cdot n) = \frac{\delta w}{\pi} \cdot \frac{\sin\left(\frac{\delta w}{2} \cdot n\right)}{\delta w \cdot n} \cdot \cos(w_0 \cdot n) \]

Adjusting for the non-causal time delay,

\[ x(n) = \frac{\delta w}{\pi} \cdot \text{sinc}\left(\frac{\delta w}{2} \cdot \left(n - \frac{M - 1}{2}\right)\right) \cdot \cos\left(w_0 \cdot \left(n - \frac{M - 1}{2}\right)\right) \]

Then apply a window:

\[ h(n) = h_d(n) \cdot w(n), \quad \text{for } 0 \leq n < M \]

See WindowBPFir.m
Matlab FIR with windows:

fir1
>> help fir1
fir1  FIR filter design using the window method.
    B = fir1(N,Wn) designs an N'th order lowpass FIR digital filter
    and returns the filter coefficients in length N+1 vector B.
    The cut-off frequency Wn must be between 0 < Wn < 1.0, with 1.0
    corresponding to half the sample rate. The filter B is real and
    has linear phase. The normalized gain of the filter at Wn is -6 dB.

If Wn is a two-element vector, Wn = [W1 W2], fir1 returns an
order N bandpass filter with passband W1 < W < W2. You can
also specify B = fir1(N,Wn,'bandpass'). If Wn = [W1 W2],
B = fir1(N,Wn,'stop') will design a bandstop filter.

B = fir1(N,Wn,WIN) designs an N-th order FIR filter using
the N+1 length vector WIN to window the impulse response.
If empty or omitted, fir1 uses a Hamming window of length N+1.
For a complete list of available windows, see the help for the
WINDOW function. KAISER and CHEBWIN can be specified with an
optional trailing argument. For example, B = fir1(N,Wn,kaiser(N+1,4))
uses a Kaiser window with beta=4. B = fir1(N,Wn,'high',chebwin(N+1,R))
uses a Chebyshev window with R decibels of relative sidelobe
attenuation.

How do you estimate the required filter length?

kaiserord FIR order estimator (lowpass, highpass, bandpass, multiband).
[N,Wn,BTA,FILTYPE] = kaiserord(F,A,DEV,Fs) is the approximate order N,
normalized frequency band edges Wn, Kaiser window beta parameter BTA
and filter type FILTYPE to be used by the FIR1 function:
    B = FIR1(N, Wn, FILTYPE, kaiser( N+1,BTA ), 'noscale' )

fir2
10.2.3 Design of Linear-Phase FIR Filters by the Frequency-Sampling Method

In this method, the desired frequency response at equally spaced frequencies is defined and then the inverse transform is performed.

\[ w_k = \frac{2\pi}{M} \cdot (k + \alpha) \]

where \( \alpha = 0 \) or \( \alpha = \frac{1}{2} \)

and for \( M \) even

\( k = 0, 1, \ldots, \frac{M}{2} - 1 \)

or for \( M \) odd

\( k = 0, 1, \ldots, \frac{M - 1}{2} \)

Note that at the specific frequency points,

\[ H(w_k) = \sum_{n=0}^{M} h(n) \cdot \exp\left(-j \cdot \frac{2\pi}{M} \cdot (k + \alpha) \cdot n\right) \]

or for the points specified

\[ H(k + \alpha) = H\left(\frac{2\pi}{M} \cdot (k + \alpha)\right) = H(w_k) \]

The filter is then based on an “inverse discrete-time, discrete-frequency Fourier transform”

\[ h(n) = \frac{1}{M} \cdot \sum_{k=0}^{M-1} H(k + \alpha) \cdot \exp\left(j \cdot \frac{2\pi}{M} \cdot (k + \alpha) \cdot n\right) \]

Since the filter is real, we expect that

\[ H(k + \alpha) = H\left(\frac{2\pi}{M} \cdot \frac{(k + \alpha)}{M}\right) = H^*(M - k - \alpha) = H^*\left(\frac{2\pi}{M} \cdot \frac{(M - k - \alpha)}{M}\right) \]

See Table 10.3 for the algorithm implementation.
Matlab FIR with windows:

see MATLAB example.

Example10_2_1.m and Example10_2_2.m

>> help fir2
fir2  FIR arbitrary shape filter design using the frequency sampling method.
B = fir2(N,F,A) designs an Nth order linear phase FIR digital filter
with the frequency response specified by vectors F and A and returns
the filter coefficients in length N+1 vector B.

The vectors F and A specify the frequency and magnitude breakpoints for
the desired frequency response. The frequencies in F must be given in
increasing order with 0.0 < F < 1.0 and 1.0 corresponding to half the
sample rate. The first and last elements of F must equal 0 and 1
respectively.

B = fir2(N,F,A,NPT) specifies the number of points, NPT, for the grid
onto which fir2 linearly interpolates the frequency response. NPT must
be greater than 1/2 the filter order (NPT > N/2). If desired, you can
interpolate F and A before passing them to fir2.

By default fir2 windows the impulse response with a Hamming window.
Other available windows, including Boxcar, Hann, Bartlett, Blackman,
Kaiser and Chebwin can be specified with an optional trailing argument.

>> help firls
firls Linear-phase FIR filter design using least-squares error minimization.
B=firls(N,F,A) returns a length N+1 linear phase (real, symmetric
coefficients) FIR filter which has the best approximation to the
desired frequency response described by F and A in the least squares
sense. F is a vector of frequency band edges in pairs, in ascending
order between 0 and 1. 1 corresponds to the Nyquist frequency or half
the sampling frequency. A is a real vector the same size as F
which specifies the desired amplitude of the frequency response of the
resultant filter B. The desired response is the line connecting the
points (F(k),A(k)) and (F(k+1),A(k+1)) for odd k; firls treats the
bands between F(k+1) and F(k+2) for odd k as "transition bands" or
"don't care" regions. Thus the desired amplitude is piecewise linear
with transition bands. The integrated squared error is minimized.
10.2.4 Design of Optimum Equiripple Linear-Phase FIR filters

p. 678-691

Note: A discussion of Type 1, 2, 3, and 4 FIR filters as previously described is presented here!

Case 1 Filter: M Odd, Symmetric It can do all filters. (the best option)
Case 2 Filter: M Even, Symmetric It can not be a highpass filter.
Case 3 Filter: M Odd, Anti-Symmetric It can only be a bandpass filter. (worthless?!)
Case 4 Filter: M Even, Anti-Symmetric It can not be a lowpass filter.

Seeking the solution of Equ 10.2.70:… take a look it is “wonderful” – p. 684.

The solution has lead to the Parks-McClellan filter implementation based on the remez exchange algorithm.

In MATLAB firpm.

See Example10_2_3.m and Example10_2_5.m

>> help firpm

firpm Parks-McClellan optimal equiripple FIR filter design.
B=firpm(N,F,A) returns a length N+1 linear phase (real, symmetric coefficients) FIR filter which has the best approximation to the desired frequency response described by F and A in the minimax sense. F is a vector of frequency band edges in pairs, in ascending order between 0 and 1. 1 corresponds to the Nyquist frequency or half the sampling frequency. At least one frequency band must have a non-zero width. A is a real vector the same size as F which specifies the desired amplitude of the frequency response of the resultant filter B.

B=firpm(N,F,A,W) uses the weights in W to weight the error. W has one entry per band (so it is half the length of F and A) which tells firpm how much emphasis to put on minimizing the error in each band relative to the other bands.
Summary of advanced FIR filter design using tools

Determine an approximate filter length: kaiserord or firpmord

Windowing:

- A Kaiser window should meet specification with minimal iterations if kaiserord is used.
- fir1
- wvtool or wintool in MATLAB can help visualize the different windows that are available.
- Iterate on specification and filter length as required to get the design required.

Frequency Domain Design (same as multiband):

- A firpmord can be used for general multiband designs or frequency domain.
- fir2 or firrls or fircls
- Iterate on specification and filter length as required to get the design required.

Frequency Domain Design – Parks-McClellan:

- A firpmord can be used for general multiband designs or frequency domain.
- firpmord
- Iterate on specification and filter length as required to get the design required.


<table>
<thead>
<tr>
<th>Filter Design Method</th>
<th>Description</th>
<th>Filter Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windowing</td>
<td>Apply window to truncated inverse Fourier transform of desired &quot;brick wall&quot; filter</td>
<td>fir1, fir2, kaiserord</td>
</tr>
<tr>
<td>Multiband with Transition Bands</td>
<td>Equiripple or least squares approach over sub-bands of the frequency range</td>
<td>fircls, firpm, firpmord</td>
</tr>
<tr>
<td>Constrained Least Squares</td>
<td>Minimize squared integral error over entire frequency range subject to maximum error constraints</td>
<td>fircls, fircls1</td>
</tr>
<tr>
<td>Arbitrary Response</td>
<td>Arbitrary responses, including nonlinear phase and complex filters</td>
<td>cfirpm</td>
</tr>
<tr>
<td>Raised Cosine</td>
<td>Lowpass response with smooth, sinusoidal transition</td>
<td>rcosdesign</td>
</tr>
</tbody>
</table>