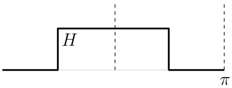
UNIVERSITY OF CAMBRIDGE

3F1, Signals and Systems

PART II.2: Design of FIR filters

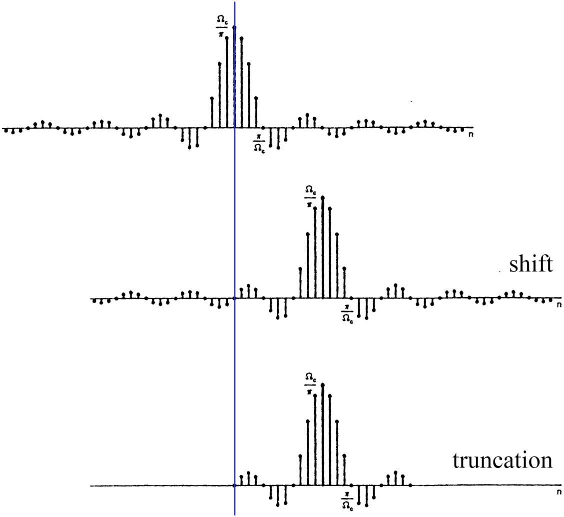
Fulvio Forni (f.forni@eng.cam.ac.uk)

October 29, 2018



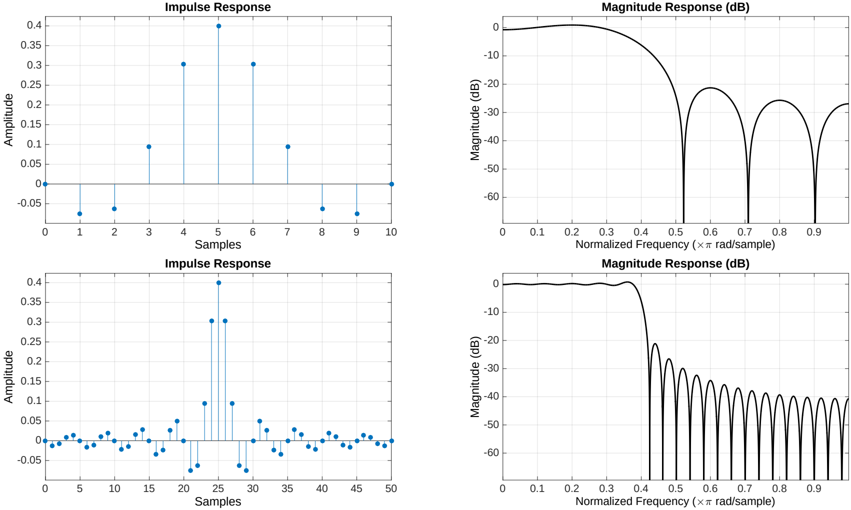
FIR filter derived by shift and truncation of the ideal response

Causality is recovered. Truncation of "small" samples has modest impact...



$$G(z) = \sum_{k=0}^{N} g_k z^{-k}$$

Truncation at N=10 and N=50 (11 and 51 samples)



Why frequency distortion?

Module A

Frequency distortion of truncation/windowing

Shifted desired impulse response

$$h_k$$

Truncated unit sample response

$$g_k = h_k \qquad 0 \le k \le N$$

Truncation = multiplication by a window

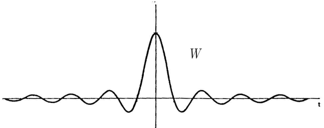
$$g_k = h_k w_k \qquad\qquad w_k = \begin{cases} 1 & 0 \le k \le N \\ 0 & \text{otherwise} \end{cases}$$

Using duality of multiplication/convolution

$$G(e^{j\omega}) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H(e^{j\theta}) W(e^{j(\omega-\theta)}) d\theta$$

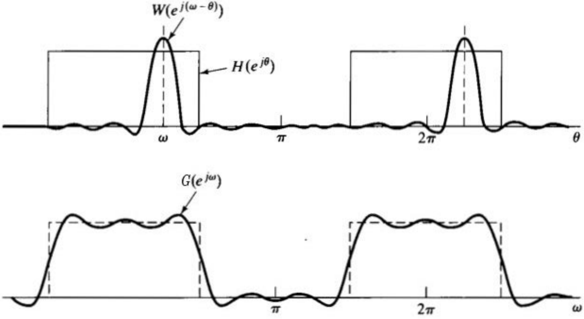
$$W(e^{j\theta}) = \sum_{k=-\infty}^{\infty} w_k e^{-j\theta k} = \sum_{k=0}^{N} e^{-j\theta k} = \frac{1-e^{-j\theta(N+1)}}{1-e^{-j\theta}} = e^{\frac{-j\theta N}{2}} \frac{\sin(\frac{\theta(N+1)}{2})}{\sin(\frac{\theta}{2})}$$

$H(e^{j\theta}) =$ 

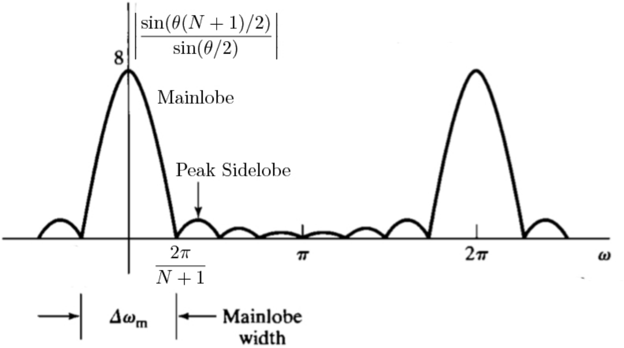


towards impulse as $N \to \infty$

$G = H * W$



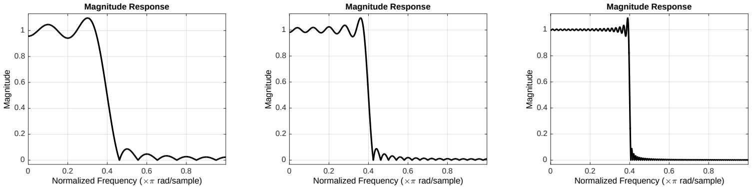
$|W(e^{j\theta})|$



- Transition band is related to mainlobe. It reduces as $N \to \infty$.
- Ripples of G are related to the area under sidelobes, which remains constant as N increases.

How to improve?

$|G(e^{j\theta})|$

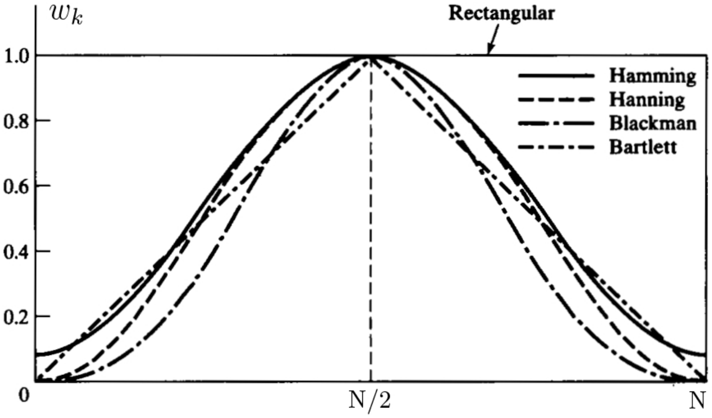


- Transition band is related to mainlobe. It reduces as $N \to \infty$.
- Ripples of G are related to the area under sidelobes, which remains constant as N increases.

How to improve?

Module B

Design by window method



$$g_k = h_k w_k$$

Reduce frequency distortion by adopting different windows

Rectangular

$$w_k = \begin{cases} 1 & 0 \leq k \leq N \\ 0 & \text{otherwise} \end{cases}$$

Bartlett (triangular)

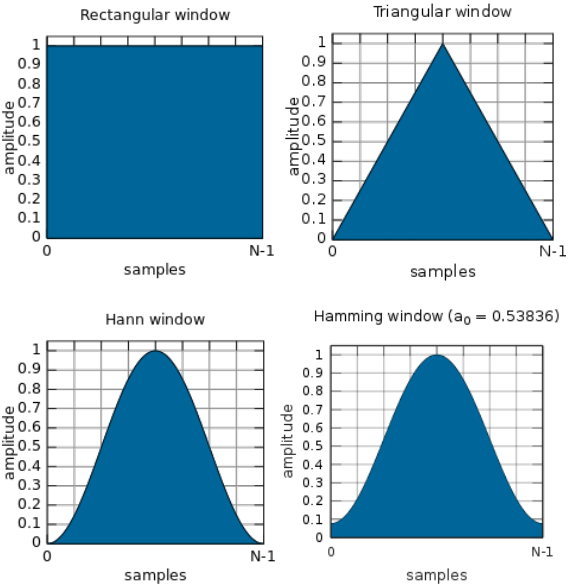
$$w_k = \begin{cases} 2k/N & 0 \leq k \leq N/2 \\ 2 - 2k/N & M/2 < k \leq N \\ 0 & \text{otherwise} \end{cases}$$

Hanning

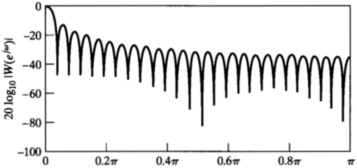
$$w_k = \begin{cases} 0.5 - 0.5\cos(2\pi k/N) & 0 \leq k \leq N \\ 0 & \text{otherwise} \end{cases}$$

Hamming

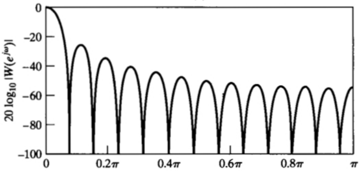
$$w_k = \begin{cases} 0.54 - 0.46\cos(2\pi k/N) & 0 \leq k \leq N \\ 0 & \text{otherwise} \end{cases}$$



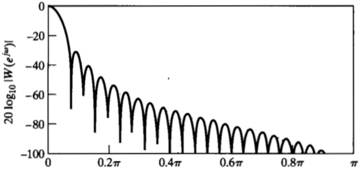
From `https://en.wikipedia.org/wiki/Window_function`



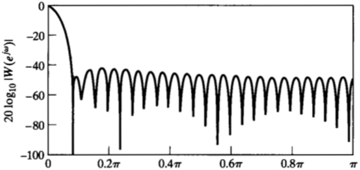
Rectangular



Triangular



Hanning

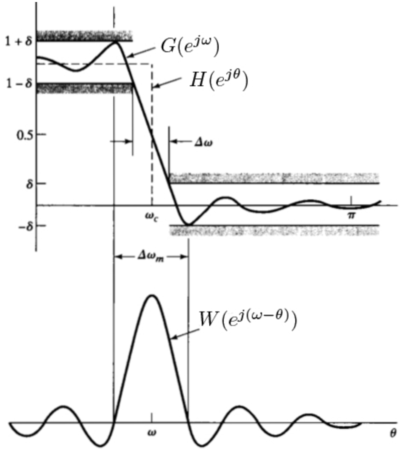


Hamming

$$G(e^{j\omega}) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H(e^{j\theta})W(e^{j(\omega-\theta)})d\theta$$

COMPARISON OF COMMONLY USED WINDOWS

Type of Window	Peak Side-Lobe Amplitude (Relative)	Approximate Width of Main Lobe	Peak Approximation Error, $20\log_{10}\delta$ (dB)
Rectangular	−13	$4\pi/(N+1)$	−21
Bartlett	−25	$8\pi/N$	−25
Hanning	−31	$8\pi/N$	−44
Hamming	−41	$8\pi/N$	−53
Blackman	−57	$12\pi/N$	−74



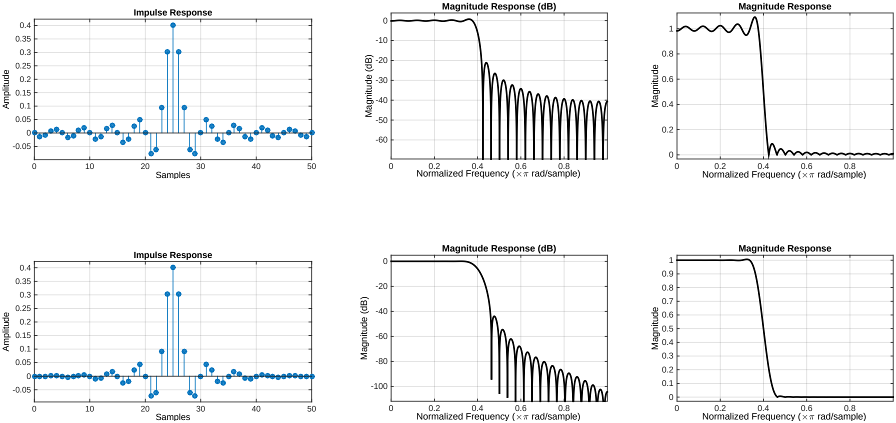
Shape and duration of the window to control the resulting filter properties.

- Smaller side lobes yield better approximations of the ideal response.
- Narrower transition bandwidth for increasing N. Symmetric at cutoff ω_c.
- Same δ for passband error and stopband approximation

The window method is conceptually simple and can quickly design filters to approximate a given target response. It does not explicitly impose amplitude response constraints, such as passband ripple or stopband attenuation, so it has to be used iteratively to produce designs which meet such specifications. Steps:

1. Select a suitable window function w_k.
2. Specify an ideal frequency response H.
3. Compute the coefficients of the ideal filter h_k.
4. Multiply the ideal coefficients by the window function to give the filter coefficients and delay to make causal.
5. Evaluate the frequency response of the resulting filter and iterate 1-5 if necessary.

Example: Rectangular vs Hanning, N=50

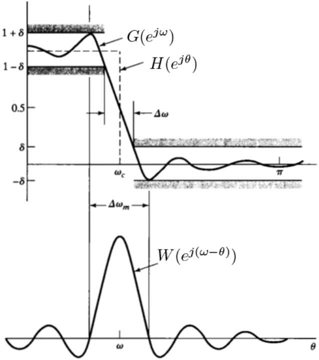


Example: Design of a low pass filter (Steps 1 and 2)

Passband $\omega_p = 0.2\pi$
Stopband $\omega_s = 0.3\pi$
Approximation error $\delta = 0.01$

COMPARISON OF COMMONLY USED WINDOWS

Type of Window	Peak Side-Lobe Amplitude (Relative)	Approximate Width of Main Lobe	Peak Approximation Error, $20\log_{10}\delta$ (dB)
Rectangular	−13	$4\pi/(N+1)$	−21
Bartlett	−25	$8\pi/N$	−25
Hanning	−31	$8\pi/N$	−44
Hamming	−41	$8\pi/N$	−53
Blackman	−57	$12\pi/N$	−74



Approximation error: -40 dB, Hanning window.
Cutoff frequency: $\omega_c = \frac{\omega_s+\omega_p}{2}\pi = 0.25\pi$.
Mainlobe width: $\omega_s - \omega_p = 0.1\pi \rightarrow N = 80$.

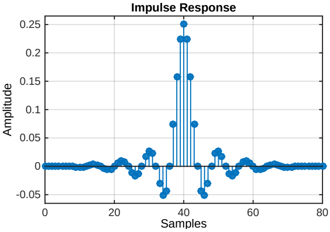
Example: Design of a low pass filter (Steps 3 and 4)

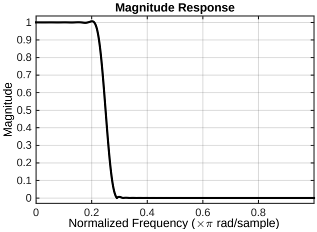
Ideal filter by inverse Fourier transform $\mathcal{F}^{-1}$

$$h_k = \frac{1}{2\pi}\int_{-\pi}^{\pi} H(e^{j\theta})e^{j\theta k}d\theta = \frac{1}{2\pi}\int_{-\omega_c}^{\omega_c} e^{j\theta k}d\theta = \frac{\sin(k\omega_c)}{\pi k} = \frac{\omega_c}{\pi}\mathrm{sinc}(k\omega_c)$$

Delay and product by window

$$g_k = h_{k-N/2}w_k$$





Note: h_k are computed by inverse Fourier transform of H. This is hard in general. Use of numerical algorithms (*lectures on Discrete Fourier Transform, Fast Fourier Transform...*).

Module C

Multi-band design

Design highpass, bandpass,…? **Composition of lowpass filters**

The ideal bandpass filter

$$
\begin{aligned}
H(e^{j\theta}) &= \begin{cases} 1 & \omega_1 \le \theta \le \omega_2 \\ 0 & \text{otherwise} \end{cases} \\
&= \mathrm{Lowpass}_{[0,\omega_2]}(e^{j\theta}) - \mathrm{Lowpass}_{[0,\omega_1]}(e^{j\theta})
\end{aligned}
$$

where

$$
\mathrm{Lowpass}_{[0,\omega_c]}(e^{j\theta}) = \begin{cases} 1 & 0 \le \theta \le \omega_c \\ 0 & \text{otherwise} \end{cases}
$$

Ideal impulse response by linearity and antitrasform:

$$
\begin{aligned}
h_k &= \mathcal{F}^{-1}(H) \\
&= \mathcal{F}^{-1}(\mathrm{Lowpass}_{[0,\omega_2]}) - \mathcal{F}^{-1}(\mathrm{Lowpass}_{[0,\omega_1]}) \\
&= \frac{\sin(k\omega_2)}{\pi k} - \frac{\sin(k\omega_1)}{\pi k}
\end{aligned}
$$

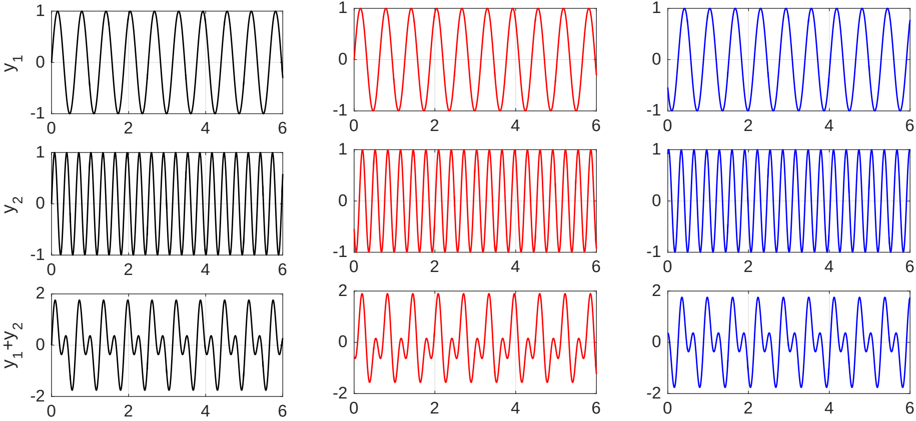
Final filter by delay and multiplication with desired window w_k:

$$
g_k = h_{k-N/2} w_k
$$

Module D

Linear Phase

Linear phase $G(e^{j\theta}) = |G(e^{j\theta})|e^{-j\theta\frac{N}{2}}$



No filter

Nonlinear phase filter

Linear phase filter

Linear phase $G(e^{j\theta}) = |G(e^{j\theta})|e^{-j\theta\frac{N}{2}}$ is achieved if $\mathbf{g_k} = \mathbf{g_{N-k}}$.

$$
\begin{aligned}
G(e^{j\theta}) &= \sum_{k=0}^{N} g_k e^{-j\theta k} \\
&= g_0 e^{-j\theta 0} + g_N e^{-j\theta N} + g_1 e^{-j\theta 1} + g_{N-1} e^{-j\theta(N-1)} + \dots \\
&= e^{-j\theta\frac{N}{2}}\Big(\underbrace{g_0 e^{j\theta\frac{N}{2}} + g_N e^{-j\theta\frac{N}{2}}}_{2g_0\cos(\theta\frac{N}{2})\text{ is real}} + \underbrace{g_1 e^{j\theta(\frac{N}{2}-1)} + g_{N-1} e^{-j\theta(\frac{N}{2}-1)}}_{2g_1\cos(\theta(\frac{N}{2}-1))\text{ is real}} + \dots\Big)
\end{aligned}
$$

Note: *window method gives linear phase filters*. The coefficients w_k of all window functions satisfies $w_k = w_{N-k}$. If the desired impulse response h_k is also symmetric $h_k = h_{N-k}$ then $g_k = h_k w_k$ is symmetric, that is, the resulting filter has linear phase.

Module E

Design by optimization

Given the ideal filter H and the weighting function W, find the optimal filter G of length N such that, given

$$E(\theta) = W(\theta)[H(\theta) - G(\theta)] \ ,$$

- G minimizes the least-squares error

$$\int_{-\pi}^{\pi} E^2(\theta) d\theta$$

- G minimizes the max error

$$\sup_{-\pi \leq \theta \leq \pi} |E(\theta)|$$

$\Rightarrow$ "Equiripple filters".

Module F

Matlab code

b = fir1(n,Wn)

b contains the coefficients of the order n Hamming-windowed filter.
This is a lowpass, linear phase FIR filter with cutoff frequency Wn.

b = fir1(n,Wn,window)

Uses the window specified in column vector window for the design.

b = fir2(n,f,m)

b contains the coefficients of the order n FIR filter whose frequency-magnitude characteristics match f and m.
f is a vector of frequency points ranging from 0 to 1.
m is a vector containing the desired magnitude response.

firls

The firls function minimizes the integral of the square of the error between the desired frequency response and the actual frequency response.

firpm

The firpm function returns filters that are optimal in the sense that the maximum error between the desired frequency response and the actual frequency response is minimized.

filterbuilder

GUI-based filter design

fvtool

Open Filter Visualization Tool

More info: Matlab: filter design Matlab: filter analysis