Window Method for FIR Filter Design: Phase

Dr John O' Toole

[1] Irish Centre for Fetal and Neonatal Translational Research (INFANT)
Department of Paediatrics and Child Health,
University College Cork.
jotoole@ucc.ie
Tel: 021-4205599

February 22, 2016

lecture notes and example questions:
https://github.com/otoolej/DSP_notes_FIR

Review: FIR filters and phase

- for frequency response $H(\omega) = |H(\omega)|e^{j\theta}$
  - (generalised) linear phase when $\theta(\omega) = -\alpha\omega + \beta$
  - group delay (GD) :

$$G(\omega) = -\frac{d\theta(\omega)}{d\theta}$$

  - group delay is equal to time delay of filtered signal (in time-domain)
- if $h[n]$ is symmetric, either $h[n] = h[M-1-n]$ or $h[n] = -h[M-1-n]$, then filter has (generalised) linear phase

Review: 4 FIR filter types

For filter with generalised linear phase (i.e, $\theta(\omega) = -\alpha w + \beta$),

type	M	symmetry	delay	phase
Type I	odd	positive	integer	$\beta = 0, \pi$
Type II	even	positive	fractional	$\beta = 0, \pi$
Type III	odd	negative	integer	$\beta = \pi/2, 3\pi/2$
Type IV	even	negative	fractional	$\beta = \pi/2, 3\pi/2$

type	low-pass	high-pass	band-pass	band-stop
Type I	✓	✓	✓	✓
Type II	✓		✓	
Type III			✓	
Type IV		✓	✓	

Design Methods for FIR Filters

1. window method
   - ideal frequency response → Fourier transform → shift and window
2. frequency sampling method
3. optimum (minimax) filter design method

Frequency Sampling Method for FIR filter design

- for the window method, we define frequency response in the **continuous** frequency domain; but we can also define in the **discrete** frequency domain.
- recall that inverse Fourier transforms are:

$$h[n] = \frac{1}{2\pi}\int_{-\pi}^{\pi} H(\omega)e^{j\omega n}d\omega, \qquad \text{continuous frequency}$$

$$h[n] = \frac{1}{N}\sum_{k=0}^{N-1} H[\tfrac{k}{N}]e^{j2\pi kn/N}, \qquad \text{discrete frequency}$$

(see p.551[1]) assuming that sampling frequency $f_s = 1/T = 1$ and $h[n]$ is length-N (i.e. periodic in N)

- where

$$H[\tfrac{k}{N}] = H(2\pi f)|_{f=k/N}$$

[1] A. Oppenheim and R. Schafer, *Discrete-time signal processing*, Prentice Hall

define frequency response in $H[k/M]$

1. define frequency response:

$$A[\tfrac{k}{M}] = \begin{cases} 1 & \text{for passband} \\ 0 & \text{for stopband} \end{cases}$$

2. include linear phase term ;

$$H[\tfrac{k}{M}] = A[\tfrac{k}{M}]e^{-j\frac{2\pi}{M}k\frac{(M-1)}{2}}$$

3. inverse DFT (discrete Fourier transform) of $H[k/M]$ to obtain $h[n]$ (length-M)

$$h[n] = \frac{1}{N}\sum_{k=0}^{N-1} H[\tfrac{k}{N}]e^{j2\pi kn/N}$$

Example

using the frequency sampling method, design a low-pass filter of length $M = 35$ with a cut off frequency of $f_c = 200$ Hz and a sampling rate of $f_s = 800$ Hz.

- design $A[k]$ for $0 \le k \le (M+1)/2$

$$A[k] = \begin{cases} 1 & 0 \le k \le \lfloor Mf_c/f_s \rfloor \\ 0 & \text{otherwise} \end{cases}$$

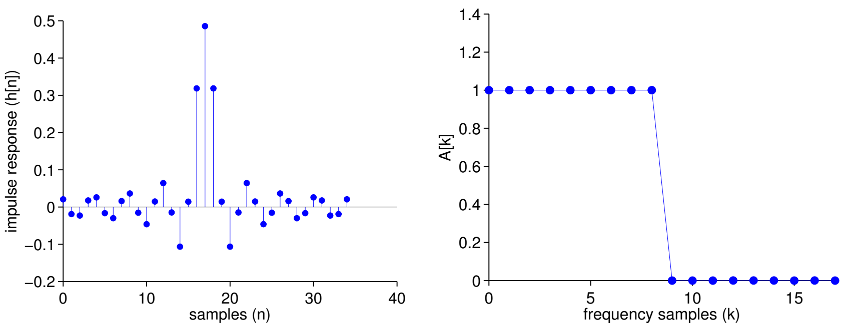
and let $A[M-k] = A[k]$ for $1 \le k \le M-1$

- include linear phase:

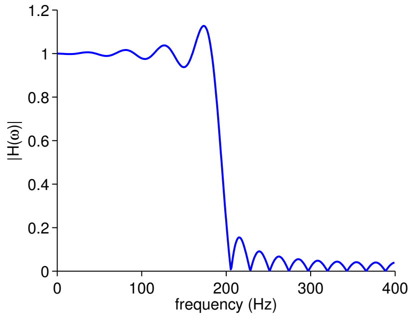
$$H[\tfrac{k}{M}] = A[\tfrac{k}{M}]e^{-j\frac{2\pi}{M}k\frac{(M-1)}{2}}$$

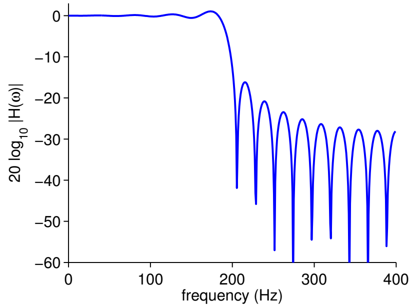
- DFT^{-1} of $H[k/M]$ to obtain $h[n]$
  - $h[n]$ is causal
  - and has linear phase and group delay $=(M-1)/2$

impulse response $h[n]$ generated from filter design $A[k]$

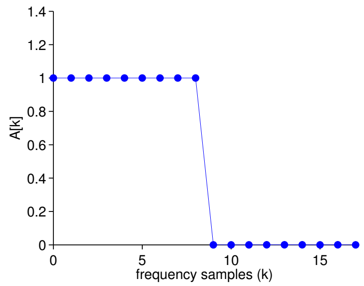


with frequency response (in continuous frequency domain):

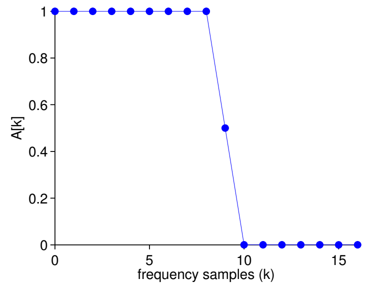




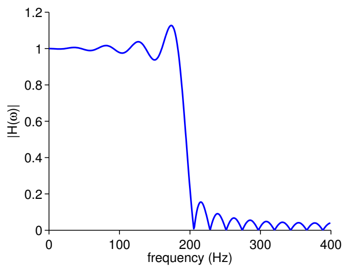
to improve on original design, include transition sample at $A[k] = 0.5$ between the points $A[k] = 1$ and $A[k] = 0$, i.e.



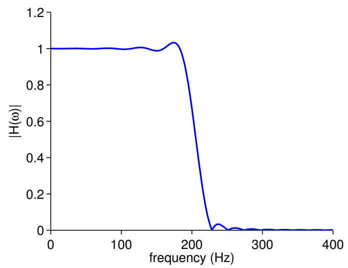
$\Rightarrow$

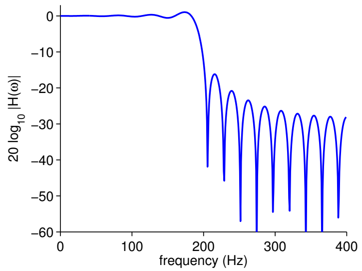


which improves the frequency response:

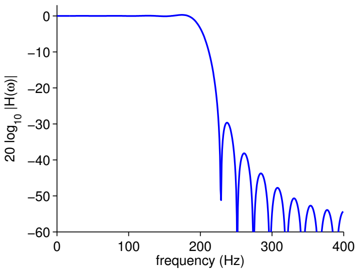


$\Rightarrow$





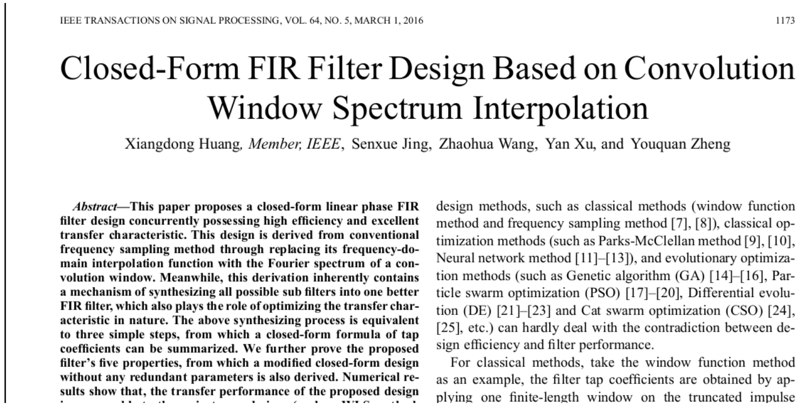
$\Rightarrow$



Pros and Cons for Frequency Sampling Method

- Pro: simplicity
- Pro: impulse response does not require inverse Fourier transform of frequency response (which may not exist for window method)
- Pro: can optimise selection of transition values to obtain maximum attenuation in the stop-band or minimum ripple in the pass-band (con: but not a simple procedure)
- Con: may be difficult to position the transition band (dependent on N)
- Con: little-to-no improvement over window method for stop-band attenuation or pass-band ripple

Could improve frequency sampling method?


IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 64, NO. 5, MARCH 1, 2016 1173

Closed-Form FIR Filter Design Based on Convolution Window Spectrum Interpolation

Xiangdong Huang, *Member, IEEE*, Senxue Jing, Zhaohua Wang, Yan Xu, and Youquan Zheng

***Abstract*—This paper proposes a closed-form linear phase FIR filter design concurrently possessing high efficiency and excellent transfer characteristic. This design is derived from conventional frequency sampling method through replacing its frequency-domain interpolation function with the Fourier spectrum of a convolution window. Meanwhile, this derivation inherently contains a mechanism of synthesizing all possible sub filters into one better FIR filter, which also plays the role of optimizing the transfer characteristic in nature. The above synthesizing process is equivalent to three simple steps, from which a closed-form formula of tap coefficients can be summarized. We further prove the proposed filter's five properties, from which a modified closed-form design without any redundant parameters is also derived. Numerical results show that, the transfer performance of the proposed design**

design methods, such as classical methods (window function method and frequency sampling method [7], [8]), classical optimization methods (such as Parks-McClellan method [9], [10], Neural network method [11]–[13]), and evolutionary optimization methods (such as Genetic algorithm (GA) [14]–[16], Particle swarm optimization (PSO) [17]–[20], Differential evolution (DE) [21]–[23] and Cat swarm optimization (CSO) [24], [25], etc.) can hardly deal with the contradiction between design efficiency and filter performance.

For classical methods, take the window function method as an example, the filter tap coefficients are obtained by applying one finite-length window on the truncated impulse

(IEEE Transactions on Signal Processing, March 2016)

Optimum FIR Filter Design

Given a set of specifications (f_c, f_p, δ_p, δ_s, and M), minimax procedure:

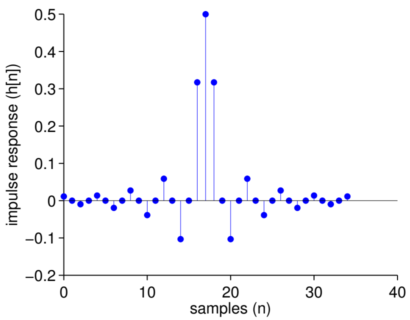
- maximise stop-band attenuation and pass-band ripple for a minimum filter length M
- optimise a series of Chebyshev polynomials using *Remez multiple exchange* or *Parks–McClellan* algorithm
- feature of minimax procedure:
  - pass-band and stop-band ripple is flat (equiripple)

Example

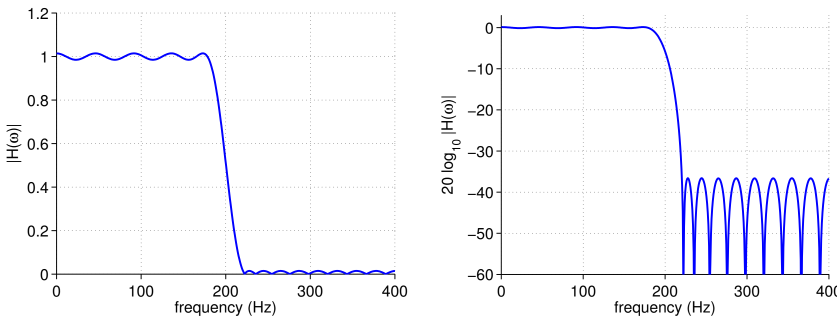
Design low-pass filter with $M = 35$ with transition band or 0.45π–0.55π

in Matlab:

```
>> M=35;
>> h=firpm(M-1,[0 0.4 0.6 1],[1 1 0 0]);
```



equiripple frequency response:



Matlab Example

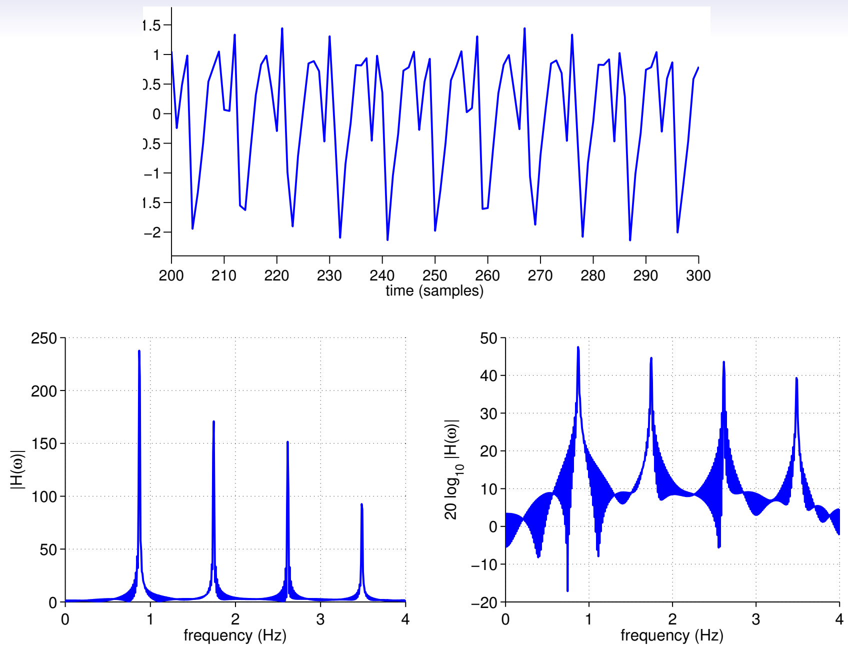
generate the harmonic test signal

$$x[n] = \sum_{p=1}^{4} a_p \sin\left(2\pi(pf_0/f_s)n + \phi_p\right)$$

for $0 \leq n \leq N-1$, fundamental frequency $f_0 = 0.87$ Hz, amplitude $a = \{1, 0.7, 0.6, 0.4\}$, phase shifts $\phi = \{\pi, \pi/3, \pi/5, \pi/7\}$, and sampling frequency $f_s = 8$ Hz.

```
N=500; f0=0.87; fs=8;
n=0:(N-1);
amp_scale=[1 0.7 0.6 0.4];
phase_shift=pi./[1 3 5 7];

x=zeros(1,N);
for p=1:length(amp_scale)
    x=x+sin( 2*pi*(p*f0/fs).*n + phase_shift(p) ).*amp_scale(p);
end
```



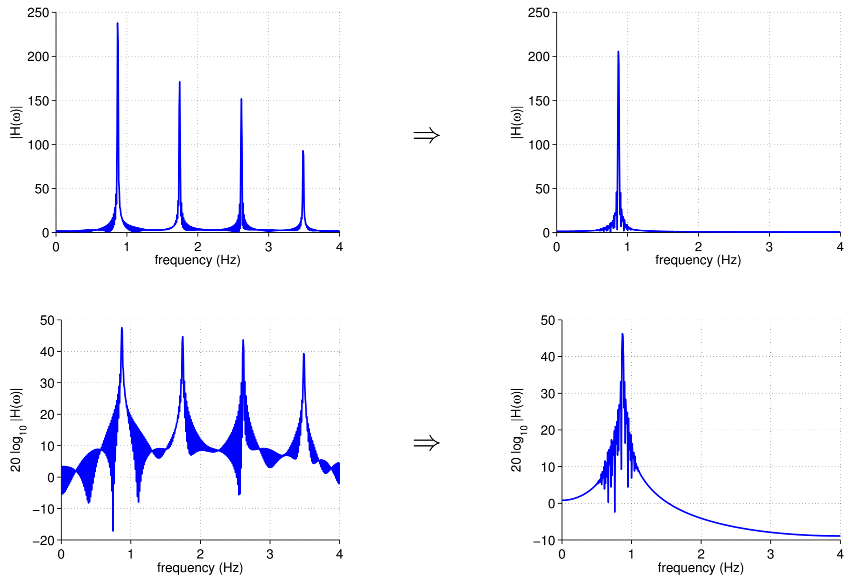
want to separate fundamental component (i.e. $a_1 \sin(2\pi(f_0/f_s)n + \phi_1)$ from signal $x[n]$.

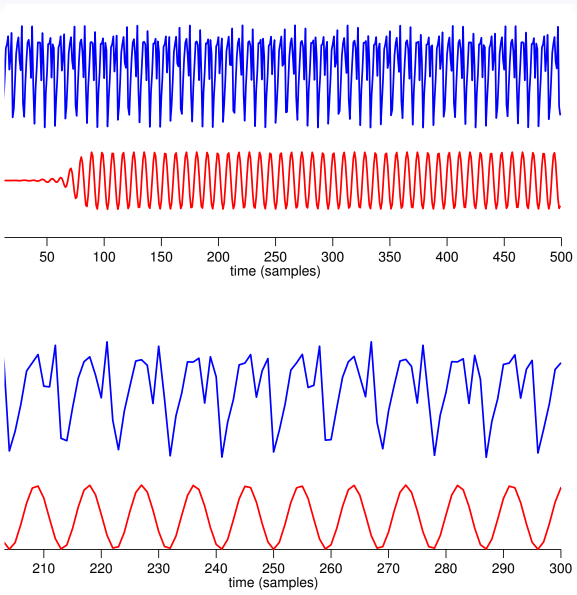
- use band-pass filter with pass-band 0.6–1 Hz;
- want > 60 dB attenuation in stop-band $\Rightarrow$ use Blackman window;
- transition band 0.3 Hz $\Rightarrow M = \lceil 5.5/(0.1/8) \rceil = 147$;
- M is odd $\Rightarrow$ integer delay ($(M-1)/2 = 73$ samples);

```
pass_band=[0.6 1];
h=fir1(M-1,pass_band./(fs/2),'band',blackman(M));
% to analyse frequency response:
freqz(h,1,1024,fs);

% filter signal:
y=filter(h,1,x);
```

filtered signal:





Summary

- different types of method to design FIR filters:
  1. windows method
  2. frequency sampling method
  3. optimum filter design method (e.g. minimax)

- frequency sampling method
  - simple to design but no improvement over windows method for standard filters (e.g. low-pass)

- can produce flat (equiripple) passband and stopband response with minimax method