Design of FIR Digital Filters;
LINEAR PHASE
Lecture 13-15

Andreas Spanias
spanias@asu.edu

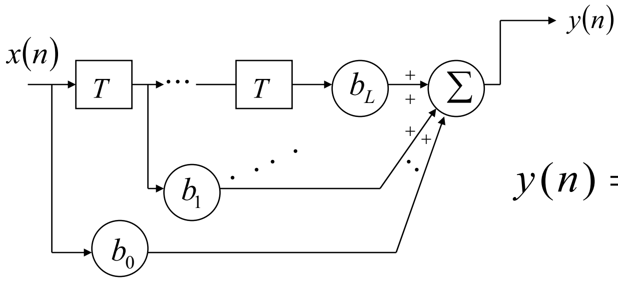
FIR Digital Filters

Advantages:
- Linear Phase Design
- Quite Efficient for designing notch filters
- Always Stable

Disadvantages:
- Requires High Order for Narrowband Design

Applications:
- Speech Processing, Telecommunications
- Data Processing, Noise Suppression, Radar
- Adaptive Signal Processing, Noise Cancellation, Echo Cancellation, Multipath channels

FIR Digital Filters



$$y(n) = \sum_{i=0}^{L} b_i x(n-i)$$

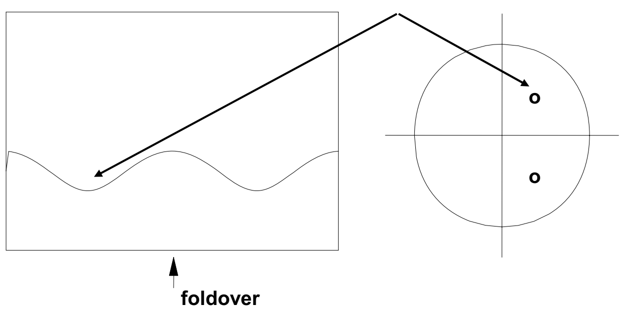
$$Y(z) = b_0 X(z) + b_1 X(z) z^{-1} + \ldots + b_L X(z) z^{-L}$$

$$H(z) = \frac{Y(z)}{X(z)} = b_0 + b_1 z^{-1} + \ldots + b_L z^{-L}$$

FIR Filter Frequency Response

$$H(e^{j\Omega}) = b_0 + b_1 e^{-j\Omega} + \ldots + b_L e^{-jL\Omega}$$

$$\Omega = 2\pi \frac{f}{f_s}$$



FIR Filter Design

1. LINEAR PHASE DESIGN
2. FOURIER SERIES DESIGN
3. ZERO PLACEMENT
4. FREQUENCY SAMPLING
5. LEAST SQUARES
6. IMPLEMENTATIONS

LINEAR PHASE DESIGN

Linear Phase (constant time delay) FIR filter design is important in pulse transmission applications where pulse dispersion must be avoided. The frequency response function of the FIR filter is written as:

$$H(e^{j\Omega}) = b_0 + b_1 e^{-j\Omega} + b_2 e^{-j2\Omega} + \ldots + b_L e^{-jL\Omega}$$

where

$$H(e^{j\Omega}) = M(\Omega) e^{j\Phi(\Omega)}$$

$$M(\Omega) = \left| H(e^{j\Omega}) \right|, \qquad \Phi(\Omega) = \arg(H(e^{j\Omega}))$$

GROUP DELAY

The time delay or group delay of a filter is defined as

$$\tau = -\frac{d\Phi(\Omega)}{d\Omega}$$

therefore if $\Phi(\Omega)$ is a linear function of Ω then τ is a constant.

τ is given in terms of samples

LINEAR PHASE AND IMPULSE RESPONSE SYMMETRIES

It can be shown that linear phase is achieved if

$$h(n) = h(L - n)$$

where *h(n)* is the impulse response of the filter. For $L = odd$

$$H(z) = \sum_{n=0}^{\frac{L-1}{2}} h(n)(z^{-n} + z^{-(L-n)})$$

$$H(e^{j\Omega}) = e^{-j\frac{\Omega L}{2}} \sum_{n=0}^{\frac{L-1}{2}} 2\ h(n)\ \cos\left(\Omega\left(\frac{L}{2} - n\right)\right)$$

LINEAR PHASE DESIGN

if we define the pseudomagnitude

$$H_1(e^{j\Omega}) = \sum_{n=0}^{\frac{L-1}{2}} 2\ h(n)\ \cos\left(\Omega\left(\frac{L}{2} - n\right)\right)$$

Then

$$\Phi(\Omega) = \begin{cases} -\dfrac{\Omega L}{2} & , \quad H_1(e^{j\Omega}) \geq 0 \\ -\dfrac{\Omega L}{2} + \Pi & , \quad H_1(e^{j\Omega}) < 0 \end{cases}$$

hence the phase response is piecewise linear.

SYMMETRIC AND ANTI-SYMMETRIC LINEAR PHASE FILTERS

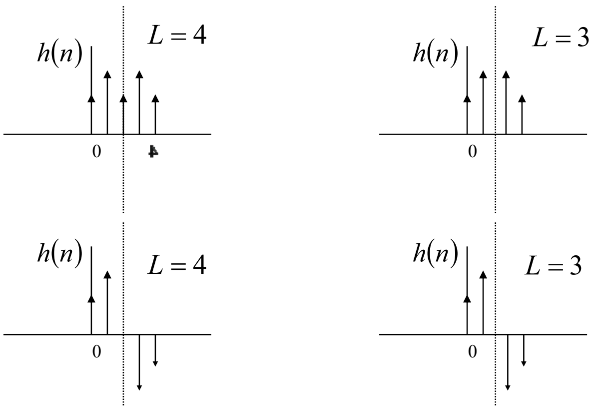
Two Anti-symmetries for *L=even* or *L=odd* for

$$h(n) = -h(L-n)$$

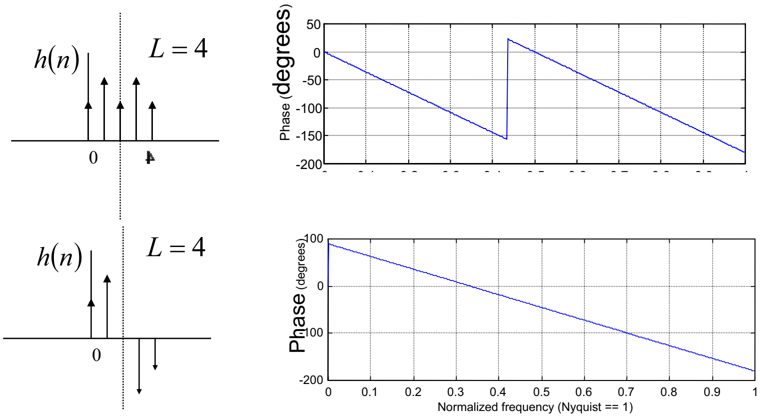
Two Symmetries for *L=even* or *L=odd* for

$$h(n) = h(L-n)$$

EXAMPLES OF SYMMETRIES



EXAMPLES OF PHASE AND SYMMETRY IN h(n)



HPF USING CERTAIN LINEAR PHASE FILTERS

Note that for $h(n) = h(L-n)$

then $$H(z) = z^{L} H(z^{-1})$$

Hence for L=*odd* and z = *-1* then

$$H(-1) = (-1)^{L} H(-1)$$

Thus the filter must have a zero at $\Omega = \pi$ and is therefore not adequate for high-pass filtering.

LPF USING CERTAIN LINEAR PHASE FILTERS

Note that for $h(n) = -h(L-n)$

then $$H(z) = -z^{L} H(z^{-1})$$

Hence any L and z = *1* then

$$H(1) = -H(1)$$

Thus the filter must have a zero at $\Omega = 0$ and is therefore not adequate for low-pass filtering.

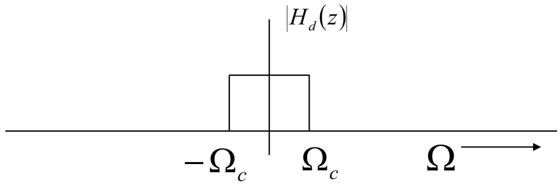
Design of FIR Digital Filters

Lecture 14 - FIR DESIGN USING THE FOURIER SERIES

Andreas Spanias
spanias@asu.edu

Design Using the Fourier Series

In filter design, there is an ideal transfer function $H_d(z)$ that is approximated by $H(z)$. For example for a low-pass filter, an ideal frequency response is given below:



We like to minimize the mean square error, i.e.,

$$\in = \frac{1}{2\Pi} \int_{-\Pi}^{\Pi} \left| H_d(e^{j\Omega}) - H(e^{j\Omega}) \right|^2 d\Omega$$

Design Using the Fourier Series (Cont.)

Recalling that in general

$$H(e^{j\Omega}) = \sum_{i=-n_1}^{n_2} h(i) e^{ji\Omega}$$

then

$$\in = \frac{1}{2\Pi} \int_{-\Pi}^{\Pi} \left| H_d(e^{j\Omega}) - \sum_{i=-n_1}^{n_2} h(i) e^{ji\Omega} \right|^2 d\Omega$$

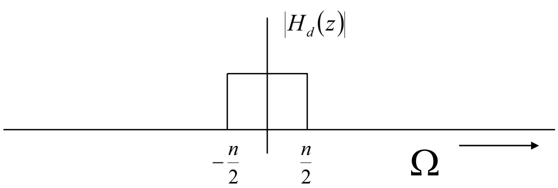
Minimization of the integral above leads to an *h(n)* sequence that is precisely equal to the sequence of Fourier series coefficients characterizing $H_d(z)$. The resultant impulse response sequence is a sampled truncated sinc function.

Fourier Series Design Example

For the ideal low pass filter the impulse response sequence is an infinite length sampled sinc function. Lets say the sampling frequency is 8KHz and we wish to have a cutoff frequency at 2KHz. This results in

$$\Omega_c = 2\Pi \frac{f_c}{f_s} = 2\pi \frac{2000}{8000} = \frac{\Pi}{2}$$

That is



Fourier Series Design Example (Cont.)

The ideal impulse response $h_d(n)$ is given by

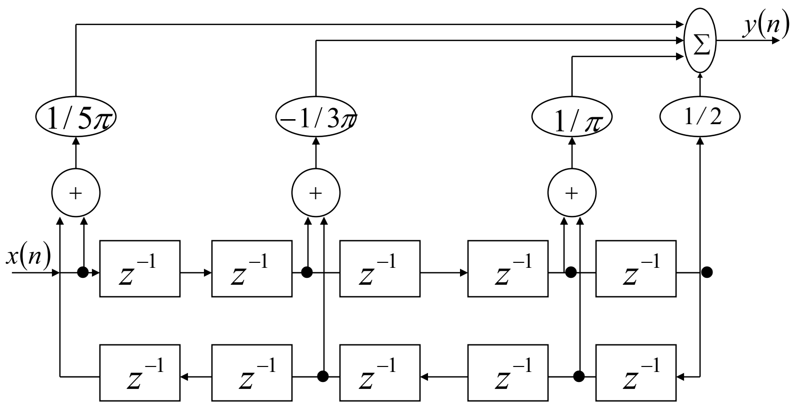
$$h_d(n) = \frac{1}{2}\,\text{sinc}\left(\frac{n\pi}{2}\right),\ n = 0, \pm 1, \pm 2, ...$$

For an FIR filter of 11 coefficients

$$h(n) = \left\{...0,0,\frac{1}{5\pi},0,-\frac{1}{3\pi},0,\frac{1}{\pi},\frac{1}{2},\frac{1}{\pi},0,-\frac{1}{3\pi},0,\frac{1}{5\pi},0,0,...\right\}$$

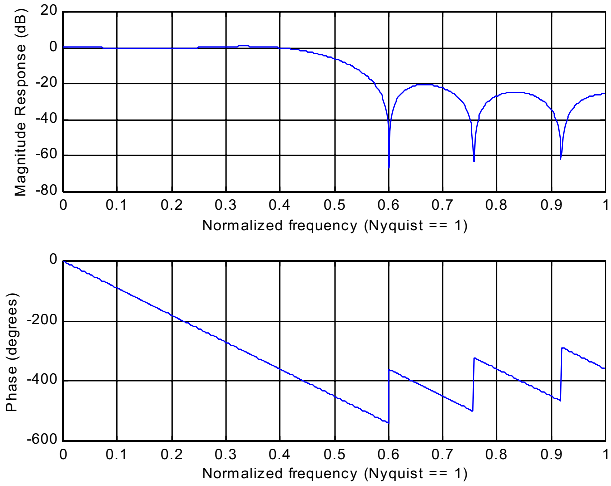
This impulse response is not causal, however a shift operator of 5 delays (z^{-5}) will convert it into a causal one.

REALIZATION



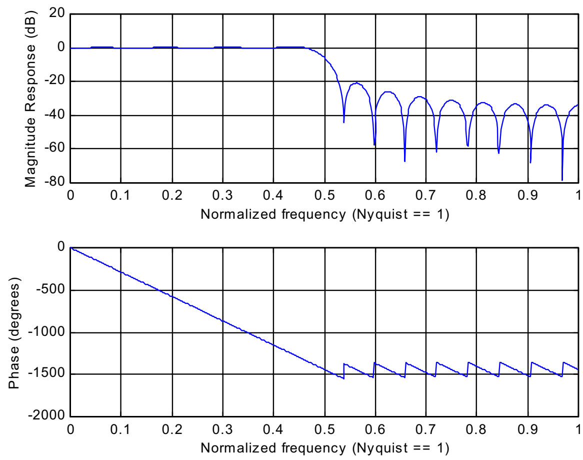
Fourier Series Design Example (Cont.)

$$b_0 = \frac{1}{5\pi}\,,\; b_2 = -\frac{1}{3\pi}, b_4 = \frac{1}{\pi}\,, b_5 = \frac{1}{2}, \; b_6 = \frac{1}{\pi}\,, b_8 = -\frac{1}{3\pi}\,,\; b_{10} = \frac{1}{5\pi}$$



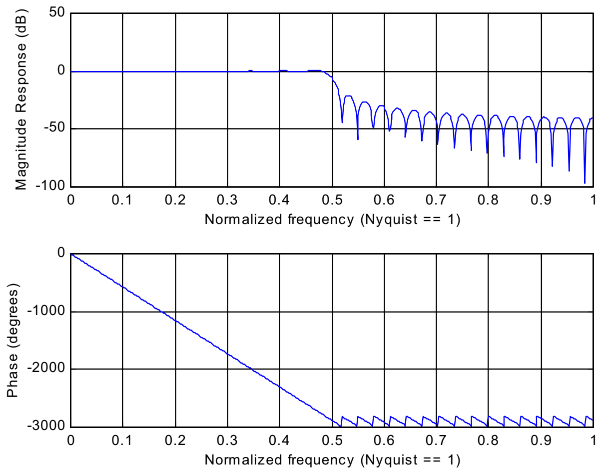
Fourier Series Design Example L=32

$$b_n = \frac{1}{2}\,\text{sinc}\left(\frac{(n-16)\pi}{2}\right), n = -16,-15,\ldots.,0,\ldots.\,15,16$$



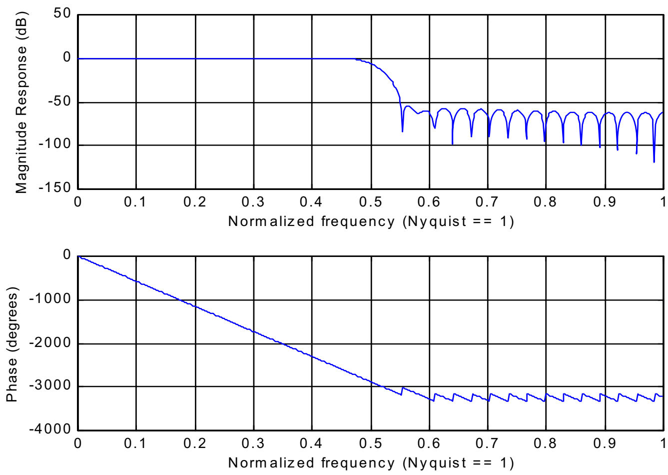
Fourier Series Design Example L=64

$$b_n = \frac{1}{2}\,\mathrm{sinc}\left(\frac{(n-32)\pi}{2}\right), n = -32,-31,....,0,....31,32$$

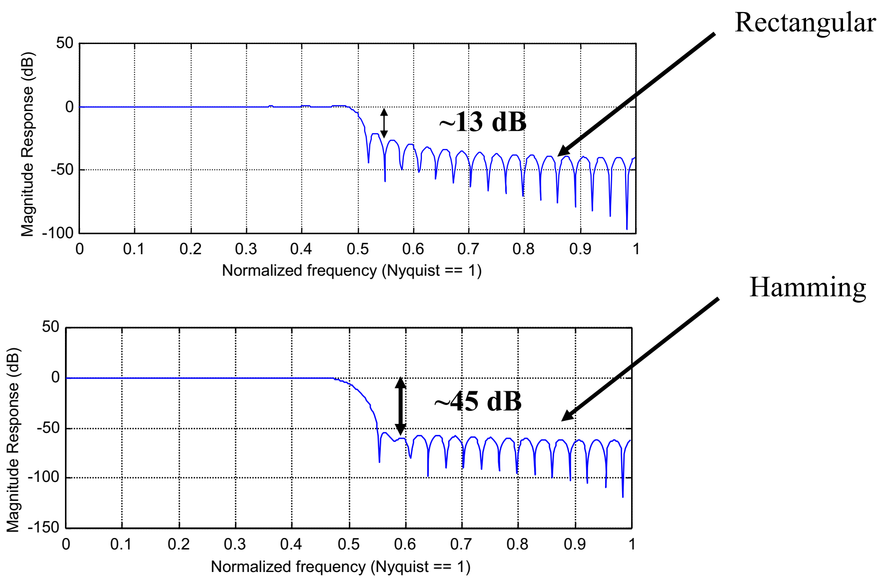


Truncating with Hamming Window L=64

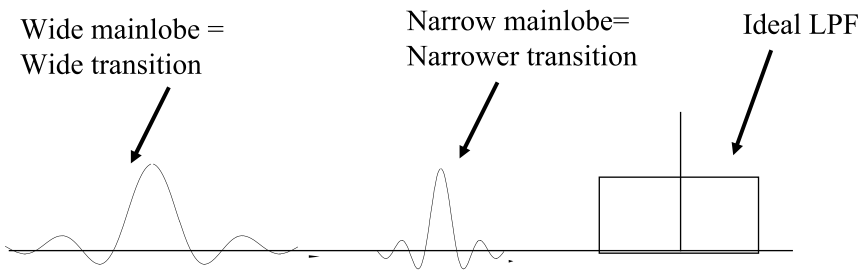
$$b_n = \frac{1}{2}\,\mathrm{sinc}\left(\frac{(n-32)\pi}{2}\right) \quad * \quad \text{Hamming(L)}$$



F.S. Design Rectangular vs Hamming Window - L=64



Truncating in time- frequency convolution and F.S. design



- The main-lobe width determines transition characteristics
- The sidelobe level determines rejection characteristics

Notes On Fourier Series Design

•The design performed in the previous example involved truncation of an ideal symmetric impulse response.A symmetric impulse response produces a linear phase response.

•Truncation involves the use of a window function which is multiplied with the impulse response. Multiplication in the time domain maps into frequency-domain convolution and the spectral characteristics of the window function affect the design.

•The main-lobe width determines transition characteristics

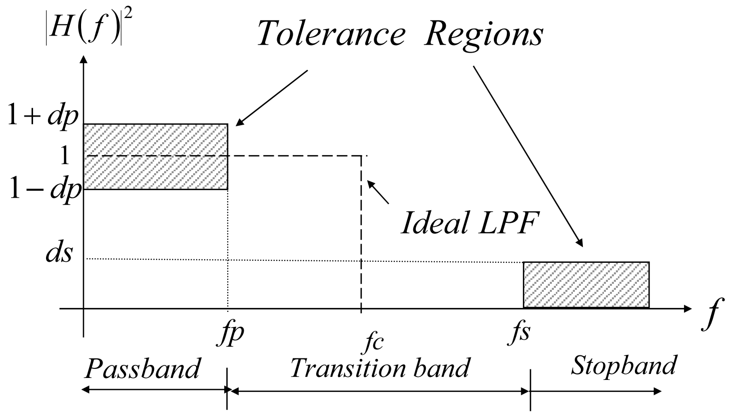
•The sidelobe level determines rejection characteristics

Design of FIR Digital Filters

Lecture 15 - FIR DESIGN USING THE KAISER WINDOW

Andreas Spanias
spanias@asu.edu

DEFINING DESIGN SPECIFICATIONS



DESIGN USING THE KAISER WINDOW

The Kaiser window is parametric and its bandwidth as well as its sidelobe energy can be designed. Mainlobe bandwidth controls the transition characteristics and sidelobe energy affects the ripple characteristics.

$$w(n) = \frac{I_0\left(\beta\left[1-\left(\frac{n-\alpha}{\alpha}\right)^2\right]^{1/2}\right)}{I_0(\beta)}, 0 \le n \le L-1$$

$\alpha =$ L/2 ; associated with the order of the filter

β is a design parameter that controls the shape of the window

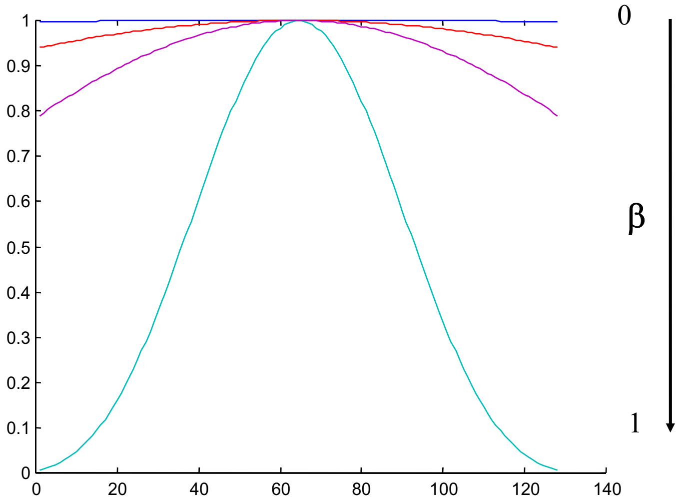
$I_0(.)$ is a zeroth Bessel function of the first kind

DESIGN USING THE KAISER WINDOW (Cont.)

$$I_0(x) = 1 + \sum_{k=1}^{\infty} \left[\frac{1}{k!} \left(\frac{x}{2} \right)^k \right]^2$$

25 terms from the Bessel function are sufficient

EXAMPLES OF KAISER WINDOW



KAISER WINDOW DESIGN EQUATIONS

Given f_p, f_s, T and dp, ds determine the FIR filter coefficients.

$$\delta = \min(ds , dp)$$
$$A = -20 \log_{10} \delta$$
$$\Delta\Omega = 2\pi (f_s - f_p) T$$

The filter order is $$L = \frac{A-8}{2.285\ \Delta\Omega} (\pm 2)$$

and the kaiser parameter β is given by

$$\beta = \begin{cases} 0.1102(A-8.7), A > 50 \\ 0.5842(A-21)^{0.4} + 0.07886(A-21), 21 \le A \le 50 \\ 0, A < 21 \end{cases}$$

DESIGN PROCEDURE

1. Determine the cutoff frequency for the ideal Fourier Series method.
$$f_c = \frac{f_s - f_p}{2}$$
2. Design the ideal LPF using the Fourier Series.
3. Design the Kaiser window
4. Shift and truncate the ideal impulse response

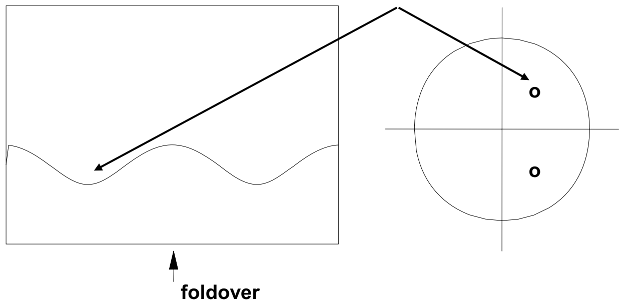
$$h_{LPF}(n) = w(n)\, h_d\left(n - \frac{L}{2}\right), \quad 0 \le n \le L$$

Note that this procedure can be generalized for the design of BPF, HPF, and BSF.

Design by Zero-Placement

As zeros are placed towards the unit circle the frequency response magnitude decreases at and in the vicinity of the frequency of the zeros.

.



Design by Zero-Placement

Example: Design a linear phase FIR filter for 60Hz interference cancellation, that will pass a 10Hz signal of interest without attenuation. The sampling frequency is 500Hz.

$$\Omega_{10} = 2\pi \frac{10}{500} = \frac{\pi}{25}$$

$$\Omega_{60} = 2\pi \frac{60}{500} = \frac{6\pi}{25}$$

A second order filter is sufficient, since only a zero pair on the unit circle is required for 60Hz cancellation. The 10Hz response is adjusted with a gain factor.

Design by Zero-Placement (Cont.)

A linear phase (steady state) design means symmetric impulse response:

$$H(e^{j\Omega_{10}}) = b_0 + b_1 e^{-j\Omega_{10}} + b_0 e^{-2j\Omega_{10}} = 1$$

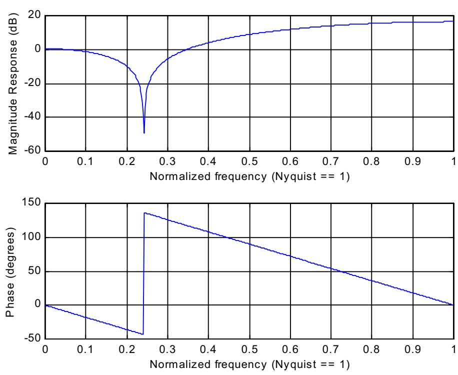
$$H(e^{j\Omega_{60}}) = b_0 + b_1 e^{-j\Omega_{60}} + b_0 e^{-2j\Omega_{60}} = 0$$

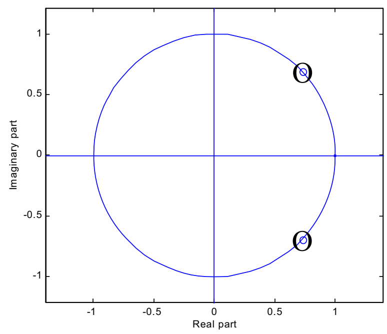
and

$$b_0 \cong 1.9, \qquad b_1 \cong -2.77$$

Design by Zero-Placement (Cont.)

$$H(e^{j\Omega}) = 1.9 - 2.77e^{-j\Omega} + 1.9e^{-2j\Omega}$$





Frequency Sampling Methods for FIR Filter design

The Frequency Sampling Method (FSM) involves: a) uniform sampling of a desired continuous frequency response function at N points, b) applying the N-point inverse Discrete Fourier Transform to obtain an N-point impulse response.

The FSM guarantees that the FIR frequency response matches that of the desired filter at the sampled points, however the response between the sampled points is different. If the desired filter function is due to an infinite impulse response then the FSM suffers also from time-domain aliasing.

Min-Max and Parks-McClellan Optimum FIR Design

The Parks-McClellan design is based on Min-Max

Equiripple and linear phase design is possible

This class of methods involve minimizing the maximum error between the designed FIR filter frequency response and a prototype

$$\min_{\{h(i),\, i=0,1,\ldots,L\}} \left\{ \max \left| E(e^{j\theta}) \right| \right\}$$

where

$$E(e^{j\theta}) = W(e^{j\theta})(H_d(e^{j\theta}) - H(e^{j\theta}))$$

FIR Filter Design Using MATLAB+

IN THE MATLAB SP TOOLBOX

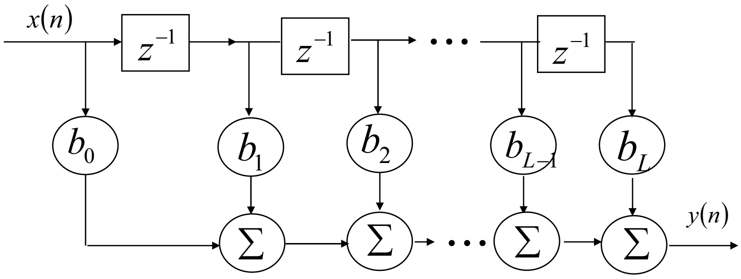
cremez - Complex and nonlinear phase equiripple FIR filter design.
fir1 - Window based FIR filter design - low, high, band, stop, multi.
fir2 - Window based FIR filter design - arbitrary response.
fircls - Constrained Least Squares filter design - arbitrary response.
fircls1 - Constrained Least Squares FIR filter design - low and highpass
firls - FIR filter design - arbitrary response with transition bands.
firrcos - Raised cosine FIR filter design.
intfilt - Interpolation FIR filter design.
kaiserord - Window based filter order selection using Kaiser window.
remez - Parks-McClellan optimal FIR filter design.
remezord - Parks-McClellan filter order selection.

+ MATLAB is registered trade mark of the MathWorks

FIR Filter Realizations

Direct Realizations

$$H(z) = \sum_{i=0}^{L} b_i z^{-i}$$

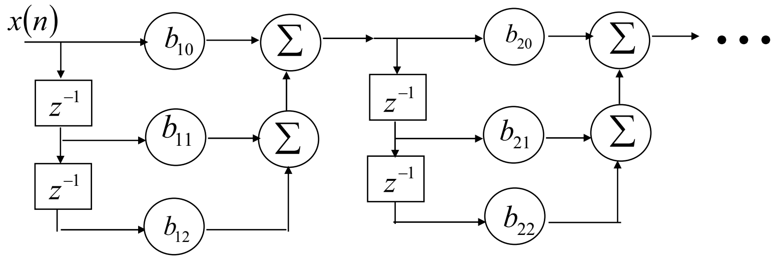


•Require multiply accumulate instructions

FIR Filter Cascade Realizations

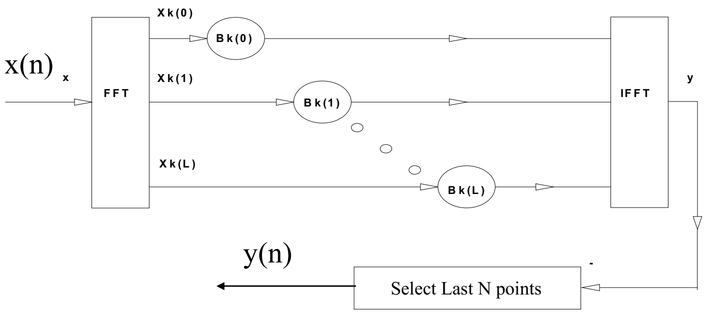
Cascade Realizations

$$H(z) = \prod_{i=1}^{q} (b_{i0} + b_{i1} z^{-1} + b_{i2} z^{-2})$$



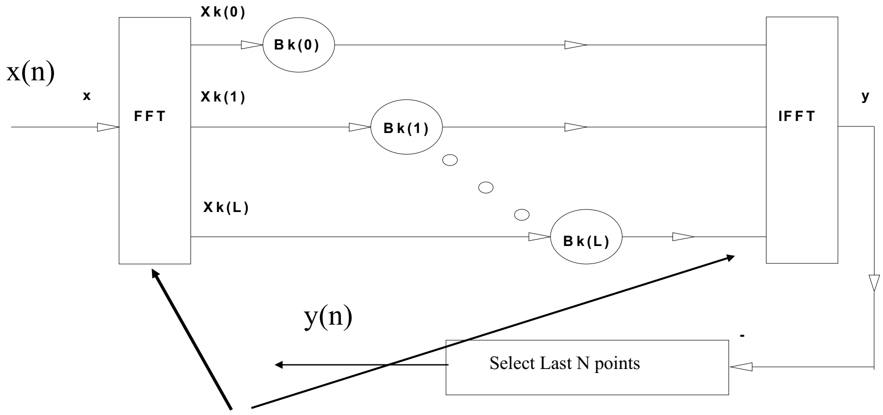
•Reduced Effects from Coefficient Quantization and round-off
•In Fixed-Point implementation signal scaling must be done carefully at each stage

Transform-Domain FIR Filter Realizations



A Transform domain realization is possible using the overlap and save and the FFT. This yields computational savings for high order implementations. Input data is organized in 2N-point blocks and blocks are shifted N points at a time. The data blocks and N zero-padded coefficients are transformed and multiplied and the results is inverse transformed. The last N-points are selected as the result. The blocks are updated and the process is repeated.

Transform-Domain FIR Filter Realizations (2)



Implements an N-th order filter with two 2-N point FFTs

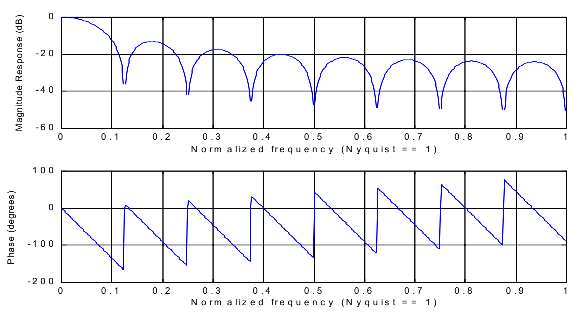
For processing N points complexity is reduced from **O**(N^2) to **O**($N\log_2 N$)

Simple and Smart FIR Filter Realizations

A simple L-tap filter can be built such that no multiplies are not used

$$H(z) = \frac{1}{L}\sum_{i=0}^{L} z^{-i}$$

This is a LPF with linear phase – if L is radix 2 the division can be implemented with shifts - below an example with L=16



Reducing Complexity by realizing FIR as IIR

An FIR filter with all its coefficients equal can be realized by looking at geometric series convergence and pole-zero cancellation

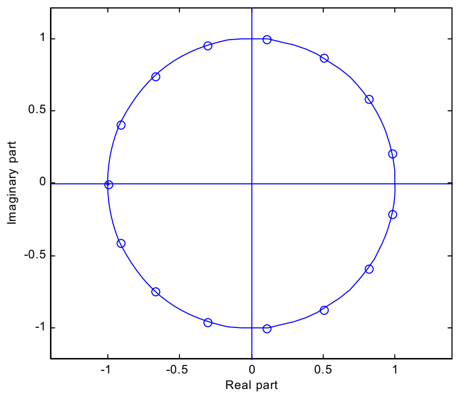
$$H(z) = \frac{1}{L}\sum_{i=0}^{L} z^{-i} = \frac{1 - z^{-(L+1)}}{L\ (1 - z^{-1})}$$

The IIR filter can be implemented using a simple difference equation with long delay. Precision may become a problem as the pole zero cancellation is on the unit circle.

$$y(n) = \frac{1}{L}(x(n) - x(n-L-1)) + y(n-1)$$

Pole-zero cancellation in FIR to IIR transformation

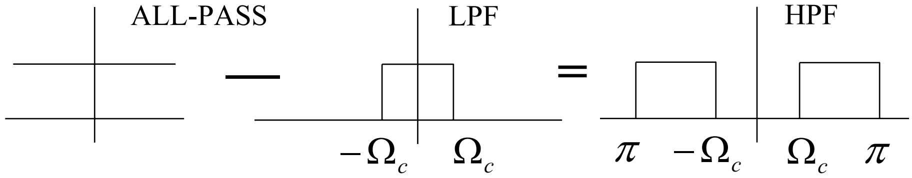
$$H(z) = \frac{1}{L}\sum_{i=0}^{L} z^{-i} = \frac{1 - z^{-(L+1)}}{L\ (1 - z^{-1})}$$



Pole zero cancellation at z=1

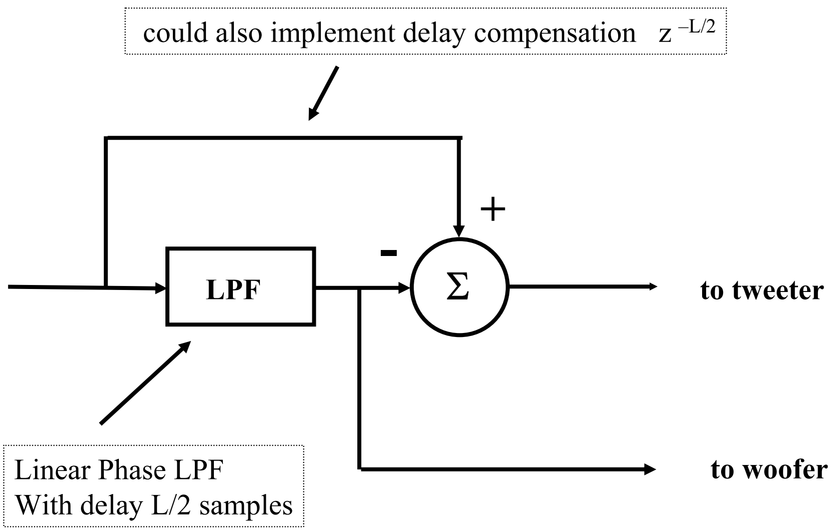
Modifying Filter Response by Subtractive Operations

$$1 \; - \; H_{LPF}(e^{j\Omega}) \; = \; H_{HPF}(e^{j\Omega})$$



Using simple operations like this we can transform prototype LPF to HPF or BPF, etc.

Implementing Efficiently Digital Cross-Over Using Subtractive Operations



Frequency Sampling

$$H_d(k) = H_d(e^{j\Omega_k}) = \sum_{n=0}^{N-1} h_d(n) e^{-j2\pi kn/N}$$

$$h(n) = \sum_{k=-\infty}^{\infty} h_d(n+kN) \qquad 0 \le n \le N-1 \qquad \text{; aliasing}$$