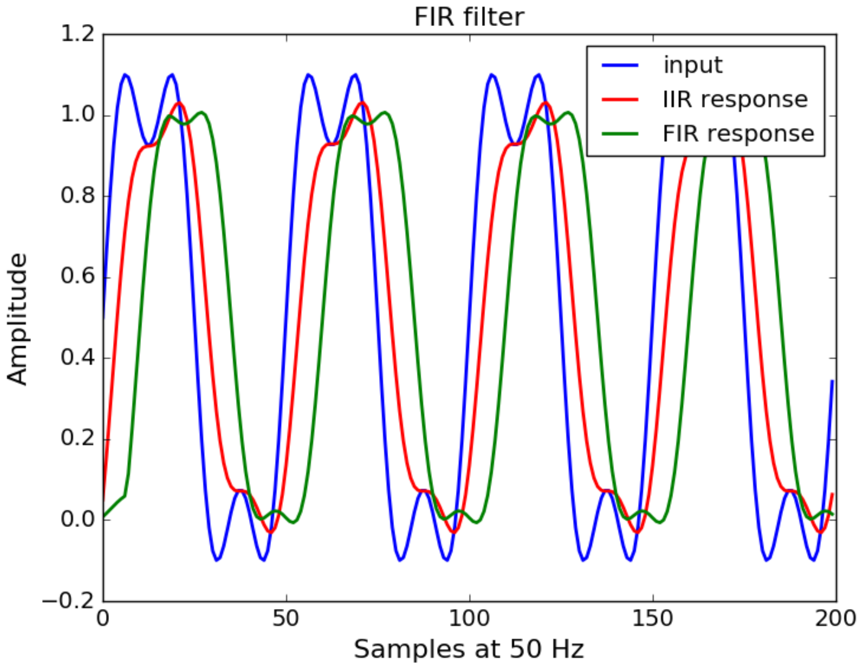
ENEL420 Advanced Signals



Digital Filters 2

Willkommen!

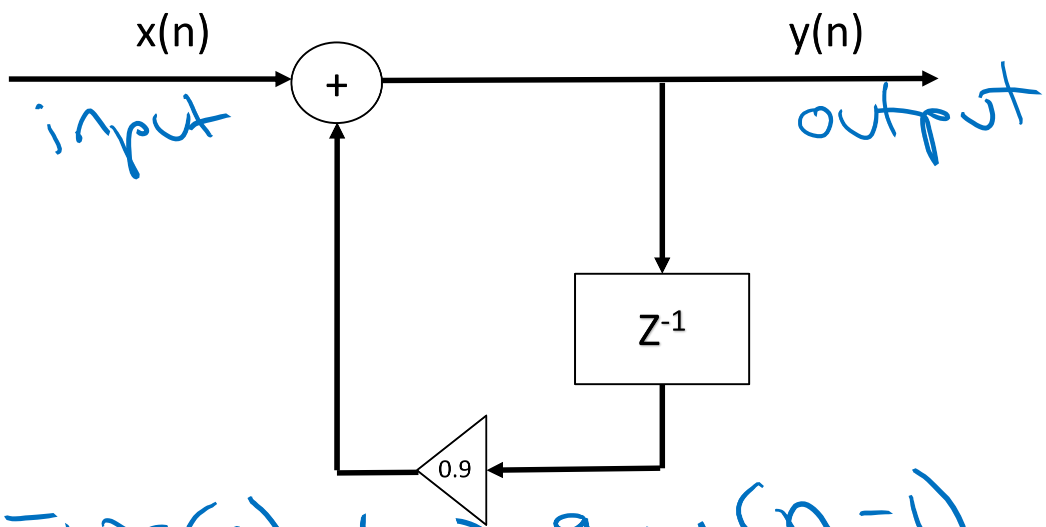
Richard Clare

2021

$$Z\{\alpha^n u(n)\} = \frac{z}{z-\alpha}$$

Infinite Impulse Response (IIR) filters

- Consider the impulse response of this filter



$$y(n) = x(n) + 0.9\,y(n-1)$$

Difference equation

$$Y(z) = X(z) + 0.9\,Y(z)z^{-1}$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1}{1-0.9z^{-1}} = \frac{z}{z-0.9}$$

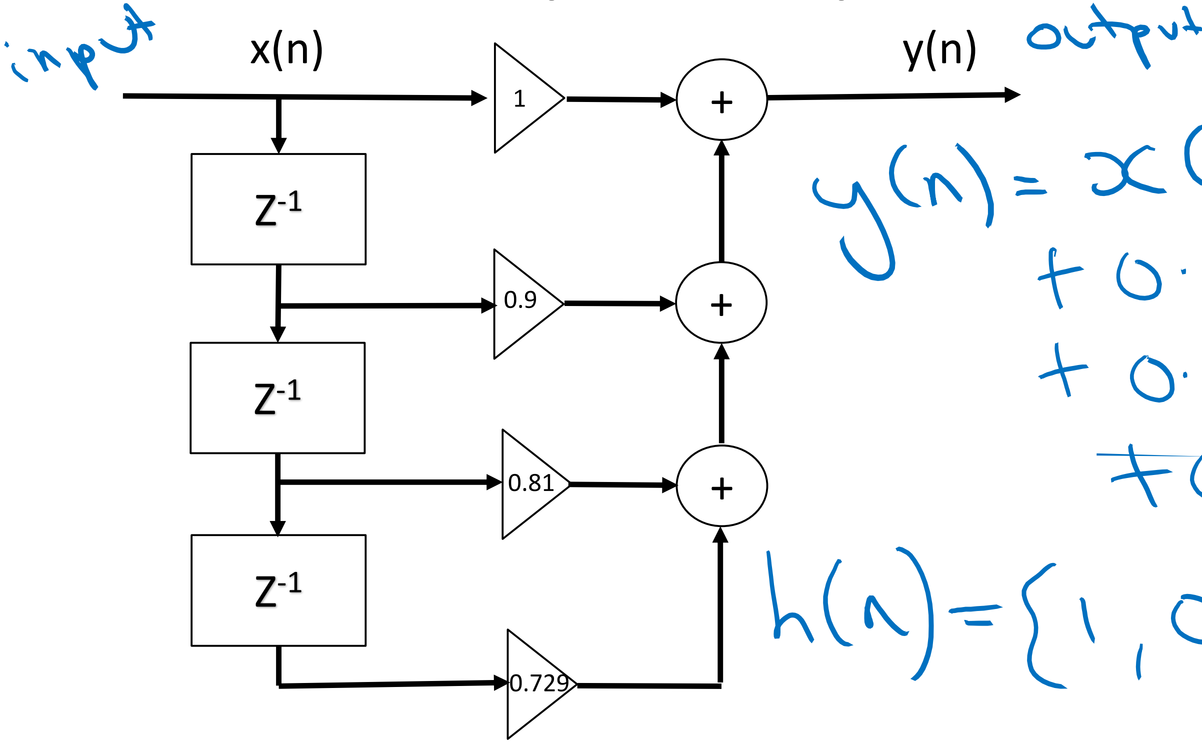
infinite

- Only requires 1 multiply and 1 addition.

$$h(n) = \{1, 0.9, 0.81, 0.729, \ldots\}$$

Finite Impulse Response (FIR) filters

- Consider the impulse response of this filter:



$$y(n) = x(n) + 0.9\,x(n-1) + 0.81\,x(n-2) + 0.729\,(n-3)$$

$$h(n) = \{1, 0.9, 0.81\ 0.729\}$$

finite

- Requires 4 multiplies and 3 additions so 7 storage units, compared to 2 for the IIR filter.

Direct Realization

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{K} b_k z^{-k}}{1 + \sum_{m=1}^{M} a_m z^{-m}}$$

- Rearranging the transfer function in Z space gives

$$\left(1 + \sum_{m=1}^{M} a_m z^{-m}\right) Y(z) = \left(\sum_{k=0}^{K} b_k z^{-k}\right) X(z)$$

- Taking the inverse Z transform and rearranging yields

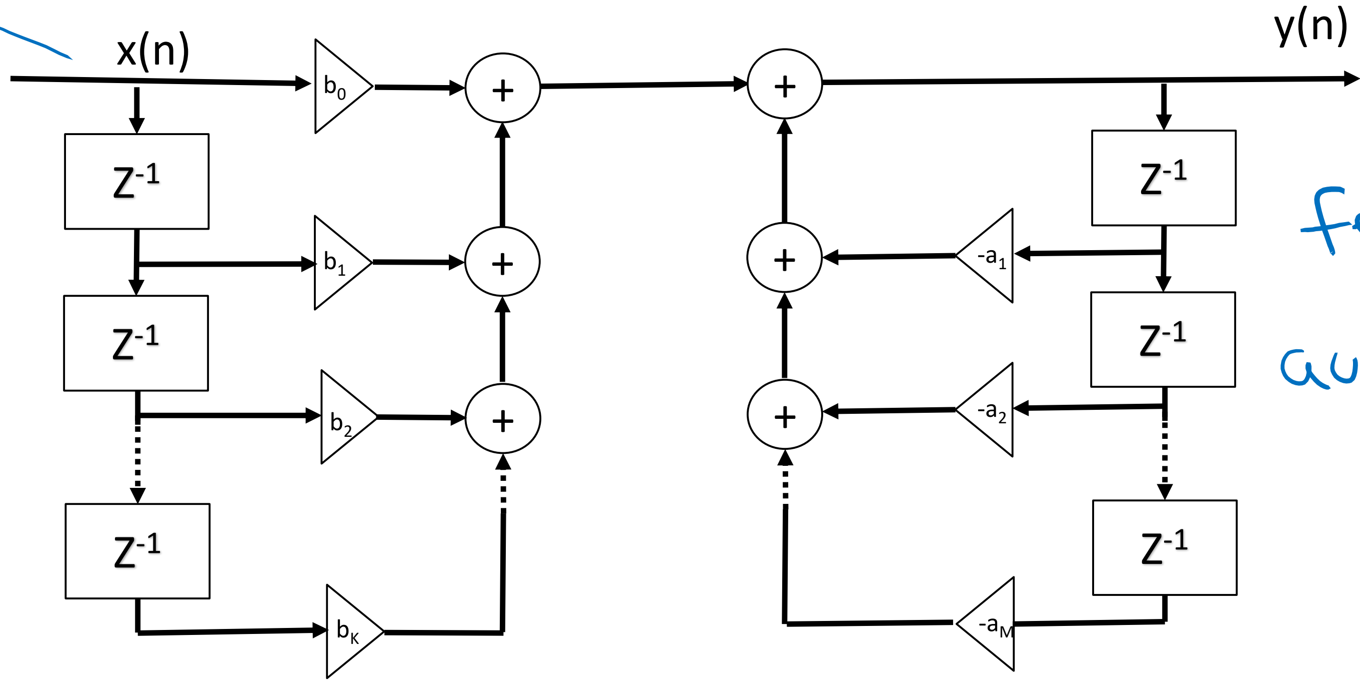
$$y(n) = \sum_{k=0}^{K} b_k x(n-k) - \sum_{m=1}^{M} a_m y(n-m)$$

input

output

Feed Forward

Moving average (MA)



feedback

autoregressive (AR)

Direct Realization Example

- Consider the following transfer function

$$H(z) = \frac{1 - z^{-1}}{1 + 0.5z^{-1}} = \frac{Y(z)}{X(z)}$$

$$Y(z)(1 + 0.5z^{-1}) = X(z)(1 - z^{-1})$$

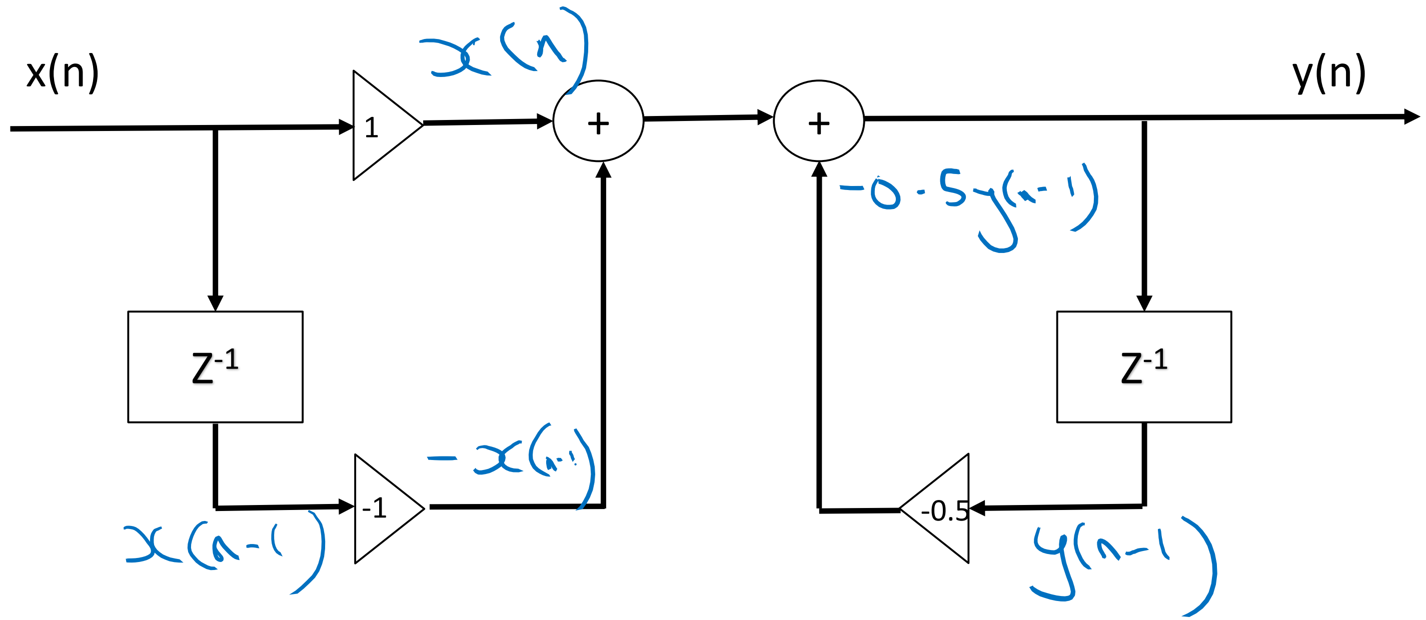
inverse z transform

$$y(n) = x(n) - x(n-1) - 0.5y(n-1)$$

difference equation

Direct Realization Example 2

- For transfer function on previous slide, we can directly realize the filter as:



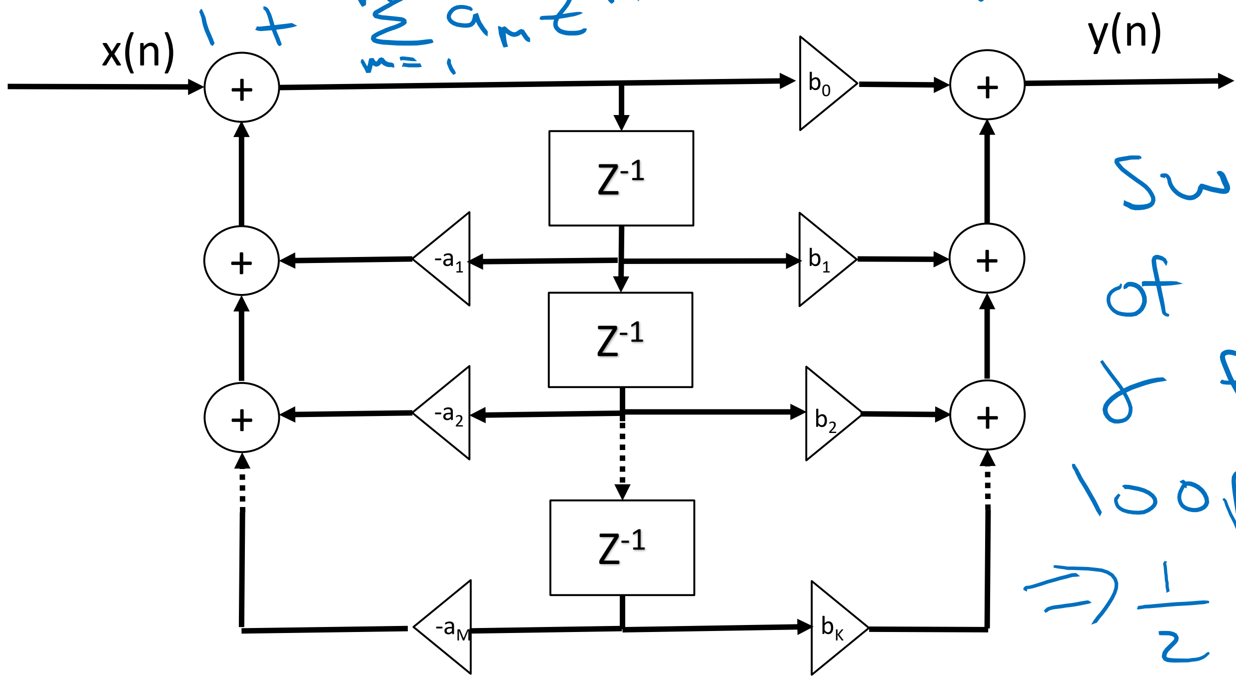
$$y(n) = x(n) - x(n-1) - 0.5\,y(n-1)$$

Alternative form

- The direct realization of a digital filter can be made more compact. Consider the general difference equation form:

$$y(n) = \sum_{k=0}^{K} b_k x(n-k) - \sum_{m=1}^{M} a_m y(n-m)$$

$$Y(z) = \frac{\sum_{k=0}^{K} b_k z^{-k}}{1 + \sum_{m=1}^{M} a_m z^{-m}} X(z) = \sum_{k=0}^{K} b_k z^{-k} \left(\frac{1}{1 + \sum_{m=1}^{M} a_m z^{-m}} \right) X(z)$$



Swap order of feedforward & feedback loops

$\Rightarrow \frac{1}{2}$ as many delay blocks

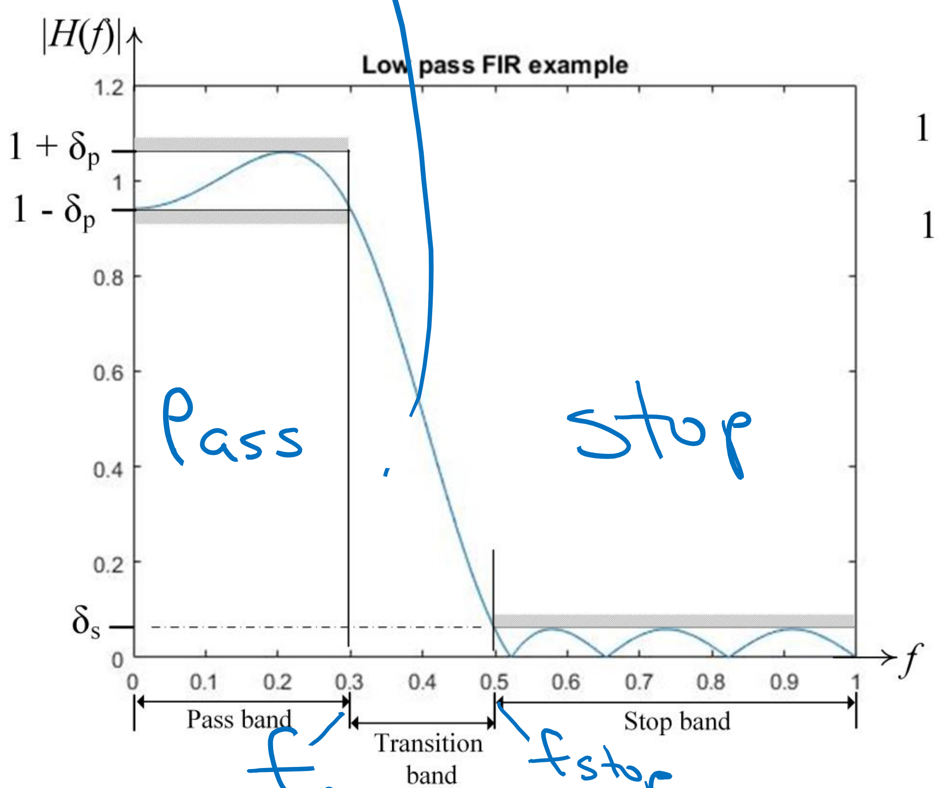
Filter Design

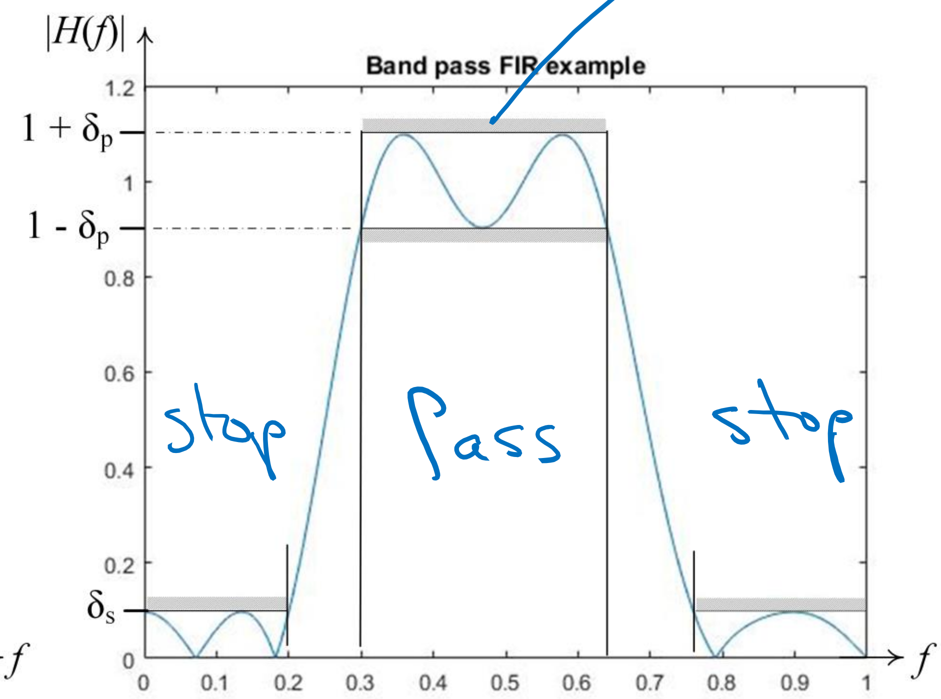
- A filter is used to selectively change the amplitude or phase of a signal in a desired manner.
- Designing a digital filter involves the following six steps:

1. Specification of the filter requirements (passband, stopband etc)
2. Choice of IIR or FIR filter
3. Calculation of filter coefficients
4. Realization of filter structure
   - direct realization
   - alternative form
5. Analysis of finite wordlength on performance (quantization noise & roundoff noise in computation)
6. Implementation in software or hardware

- These steps are not necessarily independent, nor do they have to be performed in that order. The design process is usually iterative.
- Filter Design is covered in Chapter 6 of Ifeachor & Jervis.

Filter specifications





f_p passband frequency

δ_p passband deviation

A_p passband ripple $= 20\log_{10}(1+\delta_p)$

f_{stop} = stopband frequency

δ_s = stopband deviation

A_s = stopband ripple

$A_s = -20\log_{10}\delta_s$

Methods for Coefficient calculation

- Window (FIR)
- Optimal (FIR)
- Frequency Sampling (FIR)
- Pole-zero placement (IIR)
- Impulse invariant (IIR)
- Bilinear transformation (IIR)

We will study these 4 in ENEL420

Linear Phase

- Shift theorem of Fourier transform

$$\mathcal{F}\{f(x-a)\} = e^{-j2\pi ua} F(u)$$

- So a linear phase filter acts as a pure time delay.
- Consider a speech signal. Ideally we want all frequencies to be delayed by the same amount in time to avoid distortion.
- A filter has linear phase if it can be expressed in the form $H(\omega) = |H(\omega)| e^{j\theta(\omega)}$

where $\theta(\omega) = -(\alpha\omega + \beta)$

- FIR filters can have exactly linear phase response, unlike IIR filters.

Even symmetry $f(x) = f(-x)$ eg $\cos(x$

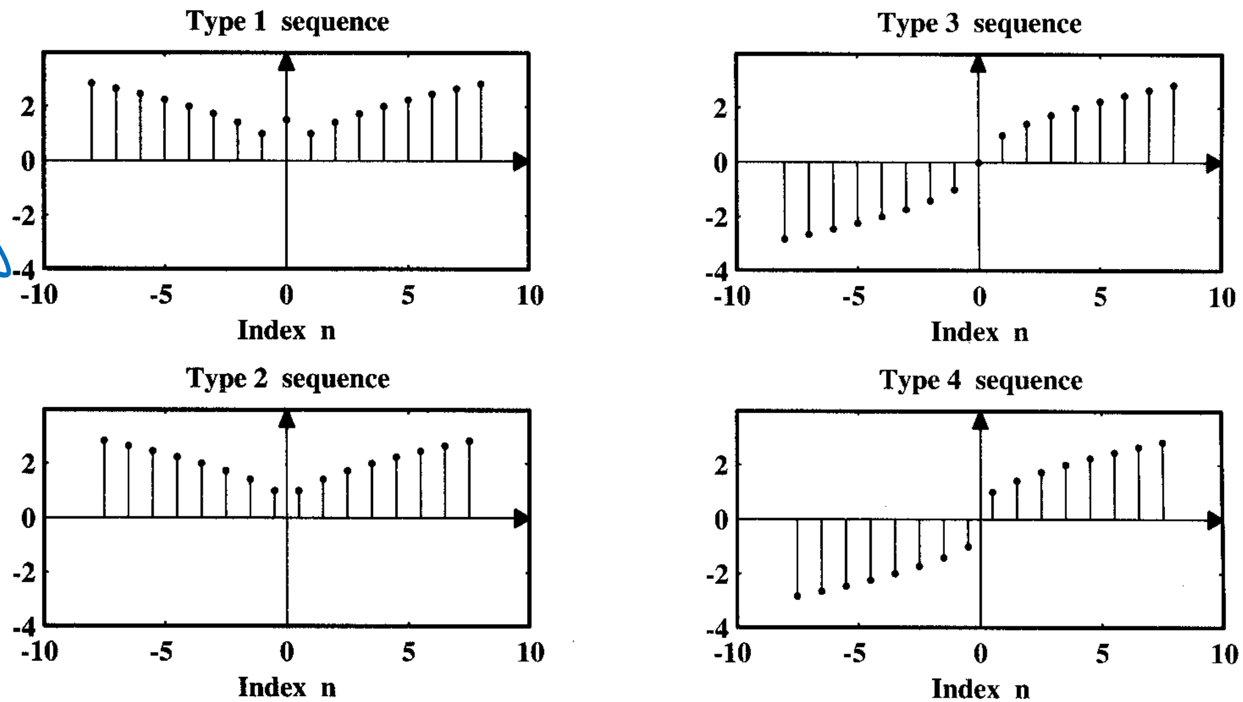
Symmetric filters

- A FIR filter with symmetric impulse response h(n) must have linear phase. There are 4 types of symmetric impulse responses.

Odd symmetry $-f(x) = f(-x)$ eg $\sin(x)$

Odd no. of elements
Even symmetry

Even no. of elements
Even symmetry



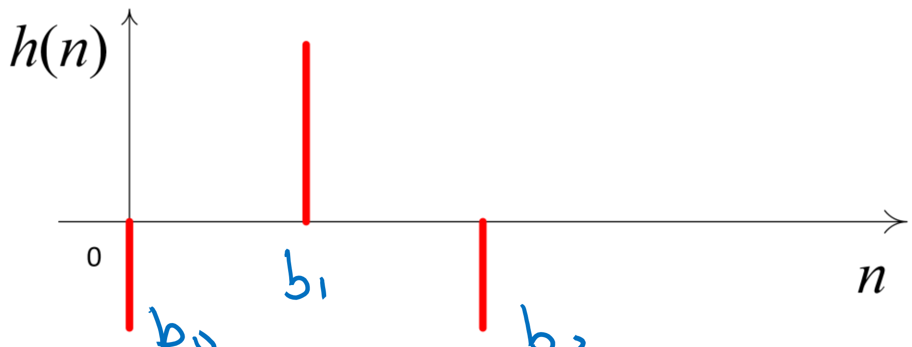
Odd no. of elements
Odd symmetry

Even no. of elements
Odd Symmetry

Phase delay Type 1 Symmetric Filters

- Consider this impulse response h(n)

$$\cos x = \frac{e^{jx} + e^{-jx}}{2}$$



$$b_0 = b_2$$

- Type 1 = Even symmetry, odd no. of coefficients

$$y(n) = b_0 x(n) + b_1 x(n-1) + b_2 x(n-2)$$

$$\frac{Y(z)}{X(z)} = H(z) = b_0 + b_1 z^{-1} + b_0 z^{-2}$$

$$H(z) = z^{-1}\left(b_0 z + b_1 + b_0 z^{-1}\right)$$

Let $z = e^{j\omega T}$

$$H(\omega) = e^{-j\omega T}\left(b_1 + b_0\left(e^{j\omega T} + e^{-j\omega T}\right)\right)$$

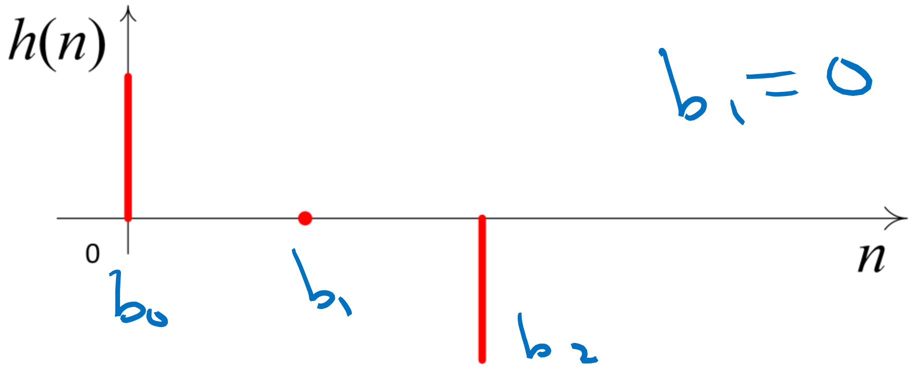
$$H(\omega) = e^{-j\omega T}\underbrace{\left(b_1 + 2b_0 \cos\omega T\right)}_{\text{real}}$$

$\theta(\omega) = -\omega T \Rightarrow$ liner phase

Phase delay Type 3 Symmetric filters

- Consider this impulse response h(n)

$b_0 = -b_2$



$b_1 = 0$

$\sin x = \frac{e^{jx} - e^{-jx}}{2j}$

- Type 3 = Odd symmetry, odd no. of coefficients

$y(n) = b_0 x(n) + b_1 x(n-1) + b_2 x(n-2)$

$H(z) = b_0 z^{-1}(z - z^{-1})$

Let $z = e^{j\omega T}$

$H(\omega) = b_0 e^{-j\omega T}(e^{j\omega T} - e^{-j\omega T})$

$H(\omega) = e^{-j\omega T} 2j\, b_0 \sin \omega T$

$\theta(\omega) = \pi/2 - \omega T \Rightarrow$ linear phase

D = 'design'

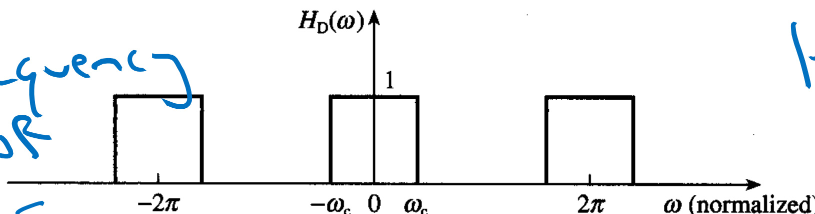
FIR filter design – Window method

- (Ifeachor & Jervis chapters 7.1 and 7.5)
- The frequency response of a filter and its impulse response h(n) are related by the inverse Fourier transform (here with frequency normalized so T=1):

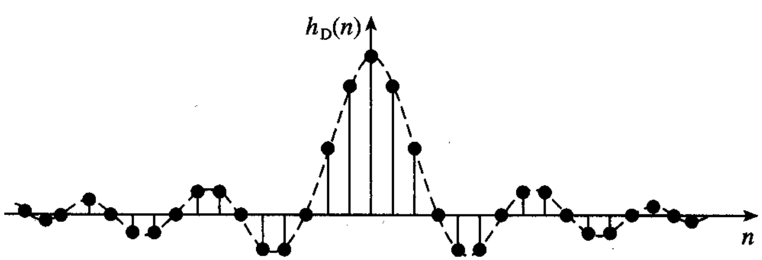
$$h_D(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_D(\omega)e^{j\omega n}\,d\omega$$

Ideal frequency response of a low pass filter

$$H_D(\omega) = \begin{cases} 1 & -\omega_c \le \omega \le \omega_c \\ 0 & \text{otherwise} \end{cases}$$



(a)



(b)

Impulse response of ideal LPF

$\omega_c = 2\pi f_c$

Window method 2

$$h_D(n) = \frac{1}{2\pi}\int_{-\omega_c}^{\omega_c} 1 \cdot e^{j\omega n}\, d\omega$$

$$= \frac{1}{2\pi}\left[\frac{e^{j\omega n}}{jn}\right]_{-\omega_c}^{\omega_c}$$

$$= \frac{1}{2\pi}\left[\frac{e^{j\omega_c n} - e^{-j\omega_c n}}{jn}\right]$$

$$= \frac{1}{2\pi}\,\frac{2\sin(\omega_c n)}{n} \times \frac{\omega_c}{\omega_c}$$

$$= \frac{2\pi f_c \, 2\operatorname{sinc}(\omega_c n)}{2\pi}$$

$$h_D(n) = 2 f_c \operatorname{sinc}(\omega_c n) \qquad n = 0, \pm 1, \pm 2$$

Window method 3

- Our expression for $h_D(n)$ is a sampled sinc function centred on n=0.
- This is not a FIR filter since the sinc function extends to infinity.
- The impulse response is Type 1 (even symmetry, odd number of coefficients).
- We could make this filter into a FIR filter by truncating $h_D(n)$ to a finite number of coefficients.

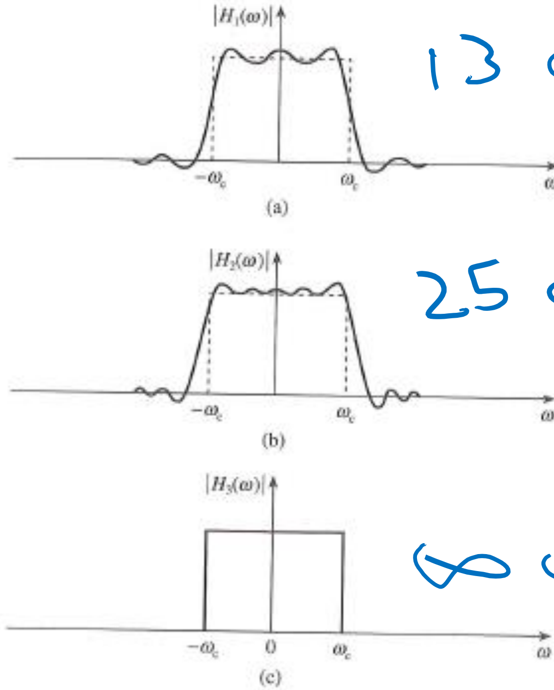
truncate:

$h_D(n) \times \text{rect}(n)$

$\downarrow$ F.T.

$H_D(\omega) * \text{sinc}(\omega)$

$\Rightarrow$ ringings & overshoots



13 coefficients

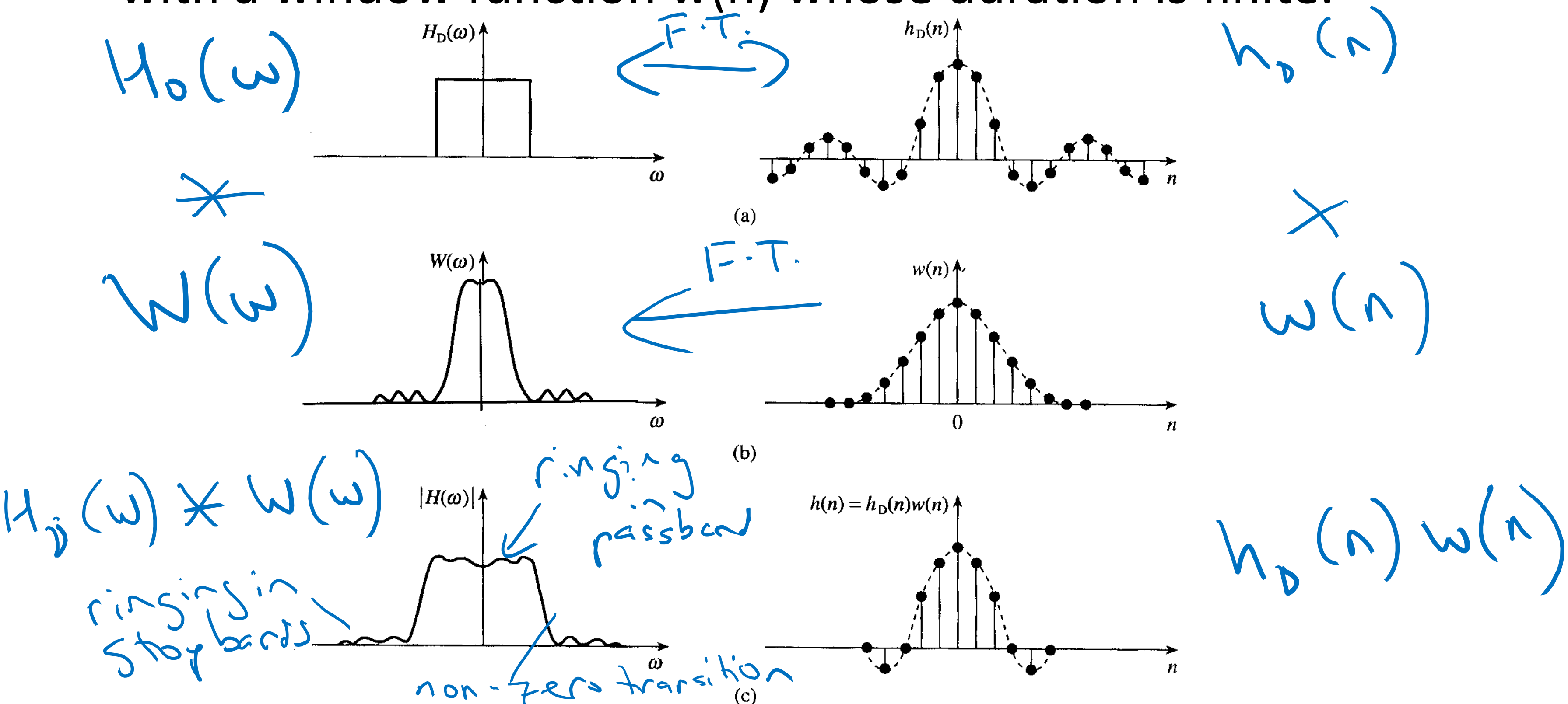
25 coefficients

∞ coefficients

Frequency response　　　　　　impulse response

Window method 4

- A practical approach to reduce this ringing is to multiply $h_D(n)$ with a window function w(n) whose duration is finite.



- In general, the more coefficients n the narrower the transition band.

Other filter types

- We can design the other types of filter besides Low Pass Filter in the same manner.

Table 7.2 Summary of ideal impulse responses for standard frequency selective filters.

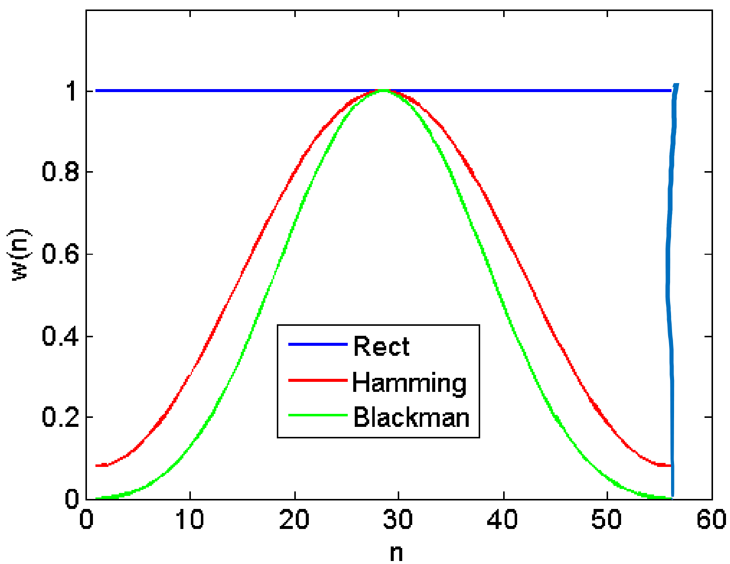
Filter type		*Ideal impulse response, $h_D(n)$* $h_D(n), n \neq 0$		*$h_D(0)$*
Lowpass	Sinc	$2f_c \dfrac{\sin(n\omega_c)}{n\omega_c}$	derived	$2f_c$
Highpass		$-2f_c \dfrac{\sin(n\omega_c)}{n\omega_c}$		$1-2f_c$
Bandpass	difference of	$2f_2 \dfrac{\sin(n\omega_2)}{n\omega_2} - 2f_1 \dfrac{\sin(n\omega_1)}{n\omega_1}$		$2(f_2-f_1)$
Bandstop	sincs	$2f_1 \dfrac{\sin(n\omega_1)}{n\omega_1} - 2f_2 \dfrac{\sin(n\omega_2)}{n\omega_2}$		$1-2(f_2-f_1)$

f_c, f_1 and f_2 are the normalized passband or stopband edge frequencies; N is the length of filter.

wider $\longleftrightarrow$ narrower

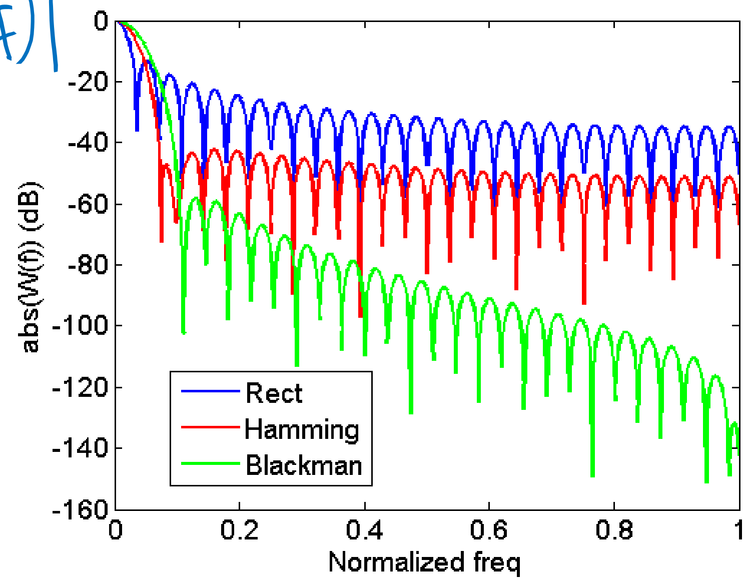
Window Functions

- Here the Hamming & Blackman windows have wider main lobes than the rect (no window) which leads to a wider transition width, but higher stopband attenuation due to the smaller sidelobes. $\mathcal{F}\{f(ax)\} = \frac{1}{|a|} F\left(\frac{u}{a}\right)$



F.T. $\longrightarrow$

$|W(f)|$



N = number of coefficients

Window function summary (I&J)

Not a function of N

Δf

Table 7.3 Summary of important features of common window functions.

Name of window function	*Transition width* (**Hz**) (*normalized*)	*Passband ripple* (**dB**)	*Main lobe relative to side lobe* (**dB**)	*Stopband attenuation* (**dB**) (*maximum*)	*Window function* $w(n), \lvert n\rvert \leq (N-1)/2$
Rectangular	$0.9/N$	0.7416	13	21	1
Hanning	$3.1/N$	0.0546	31	44	$0.5 + 0.5\cos\left(\frac{2\pi n}{N}\right)$
Hamming	$3.3/N$	0.0194	41	53	$0.54 + 0.46\cos\left(\frac{2\pi n}{N}\right)$
Blackman	$5.5/N$	0.0017	57	75	$0.42 + 0.5\cos\left(\frac{2\pi n}{N-1}\right) + 0.08\cos\left(\frac{4\pi n}{N-1}\right.$
	$2.93/N\ (\beta = 4.54)$	0.0274		50	$\frac{I_0(\beta\{1 - [2n/(N-1)]^2\}^{1/2})}{I_0(\beta)}$
Kaiser	$4.32/N\ (\beta = 6.76)$	0.002 75		70	
	$5.71/N\ (\beta = 8.96)$	0.000 275		90	

has a tunable parameter β

Window method – Example (7.3 in I&J)

Obtain the FIR coefficients for a lowpass filter to meet these specifications:

1. Passband edge frequency = 1.5 kHz f_c
2. Transition width = 0.5 kHz Δf
3. Stopband attenuation > 50 dB $A_{stop} \Rightarrow$ use Hamming
4. Sampling frequency 8 kHz f_s

Normalize frequencies with respect to f_s

transition width $\Delta f = \frac{0.5k}{8k} = 0.0625$

$f_c = \frac{1.5k}{8k} = 0.2188$

$n = -26:1:26$

$\Delta f = \frac{3.3}{N}$ $N = \frac{3.3}{0.0625} = 52.8 \Rightarrow 53$ coefficients actual coefficients

design coefficients

window

matlab : fir1

python : firwin

Window Method Example 2

$$h = h_0(n) \cdot w(n)$$

Table 7.2

$$h_0(n) = \frac{2f_c \sin(n\omega_c)}{n\omega_c} \quad n \neq 0 \qquad 2f_c \quad n = 0$$

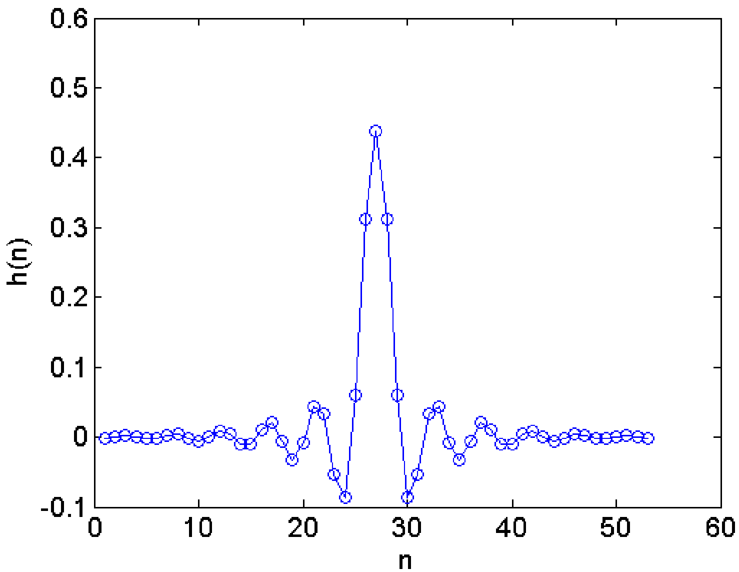
$$N = 53$$

Table 7.3

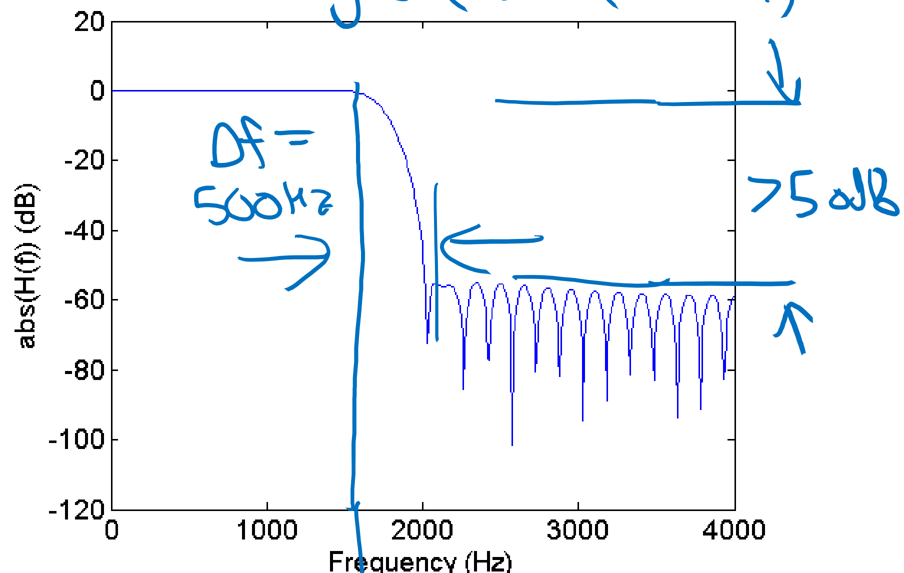
$$w(n) = 0.54 + 0.46\cos\left(\frac{2\pi n}{N}\right)$$

$\Rightarrow$ use matlab/python

$$20\log_{10}(\text{fft}(h(n)))$$



$h(n)$



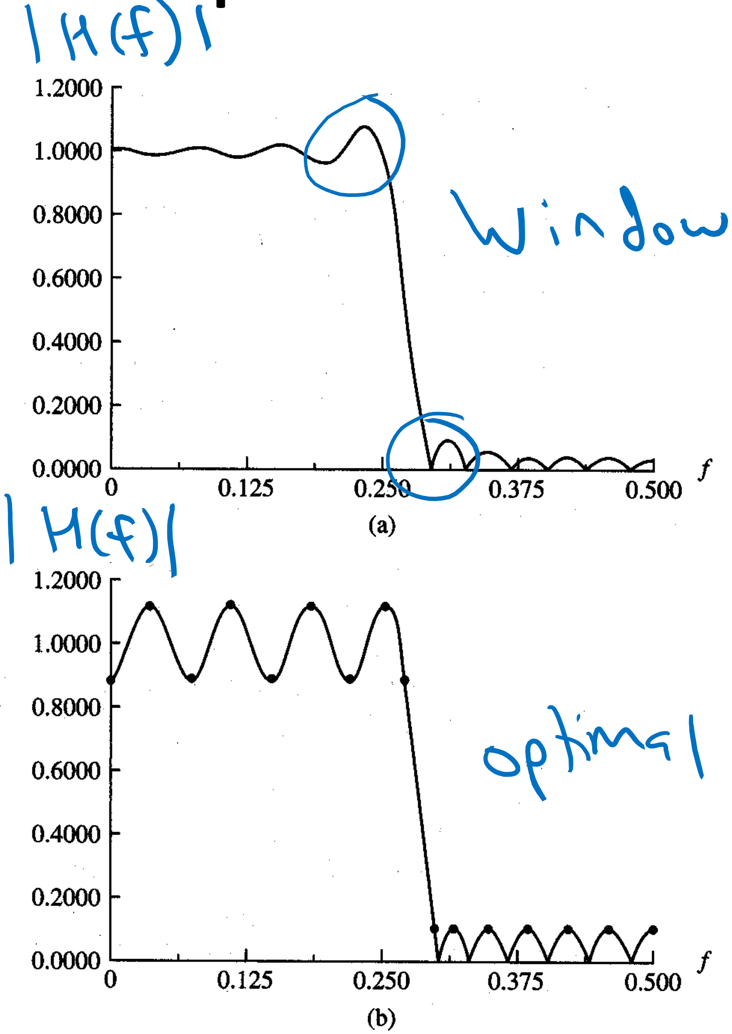
$f_c = 1500$ Hz

Pros/Cons of Window method

- Simple to apply and “to understand”. Minimal computational effort.
- Not flexible. Passband and stopband ripples are approximately equal.
- Because of effect of convolution, the passband and stopband frequencies cannot be precisely specified.
- The stopband attenuation for a given window is fixed. The filter designer must find a suitable window.
- We won’t always be able to calculate an analytic form for $h_D(n)$.

FIR filter design – Optimal method (Parks-McClellan) I&J Chapter 7.6

- In the Window method of FIR design, the ripple is largest at the transitions (ie the edge of the passband).
- If instead the ripples are distributed more evenly throughout the pass- and stopbands of the filter, we can better approximate the required frequency response.
- The optimal method of FIR design is based on the principle of equiripple passbands and stopbands.



Optimal method 2

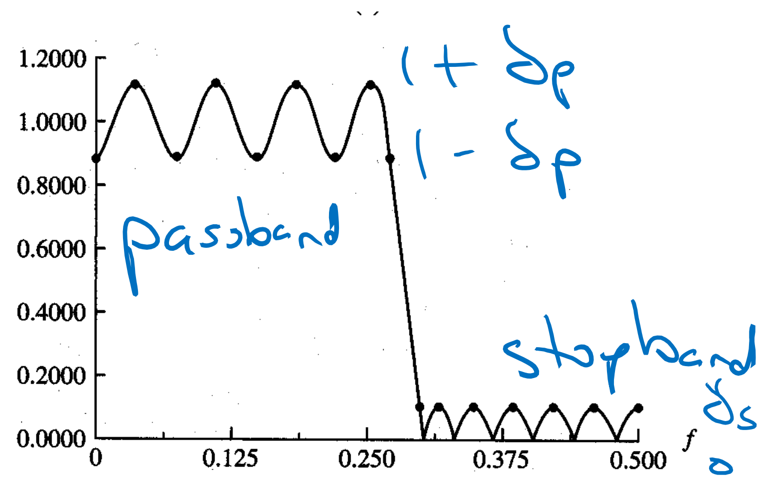
- We can write an error function E(ω) as the difference between the ideal (design) filter ($H_D(\omega)$) and the actual response H(ω):

$$E(\omega) = W(\omega)\left[H_D(\omega) - H(\omega)\right]$$

- where W(ω) is a piecewise weighting function to provide separate control over the stop- and passbands.

- In the optimal method, we determine the filter coefficients h(n) to minimize the maximum weighted error.

$$\min\left\{\max\left\{E(\omega)\right\}\right\}$$



N=11 = no. of coefficients

Optimal method in matlab

pass pass stop stop

```
h=firpm(N,[0, 0.45,0.55,1.0],[1,1,0,0])
```

normalized frequency wrt $f_s/2$

amplitude

h =

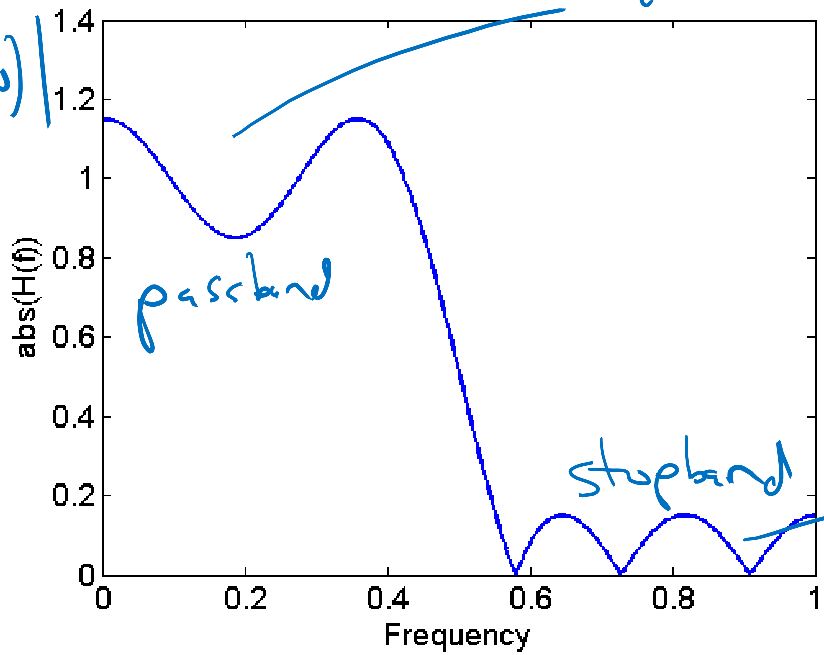
0.1110	-0.0000	-0.1032	0.0000	0.3171	0.5001	0.3171	0.0000	-0.1032	-0.0000	0.1110

filter coefficients

F.T. of h(n) our filter coefficients

$|H(\omega)|$

equiripple



equiripple

FIR

ratio $\frac{\delta_p}{\delta_s} = 3.33 = \frac{10}{3}$

Optimal method example

A linear phase bandpass filter is required to meet the following specifications:

1. Passband 900-1100 Hz
2. Passband ripple <0.87 dB A_p
3. Stopband attenuation >30dB A_s
4. Sampling frequency 15 kHz f_s
5. Transition frequency 450 Hz Δf

Normalize by $f_s/2$ (folding frequency) for Matlab's firpm

$A_p = 20\log_{10}(1+\delta_p) = 0.87$ $\quad \delta_p = 10^{0.87/20} - 1$

$\delta_p = 0.1054$

$A_s = -20\log_{10}\delta_s = 30$ $\quad \delta_s = 10^{-30/20} = 0.0316$

I & J use $N_{order} = 40 \Rightarrow N{+}1 = 41$ coefficients

Optimal method example 2

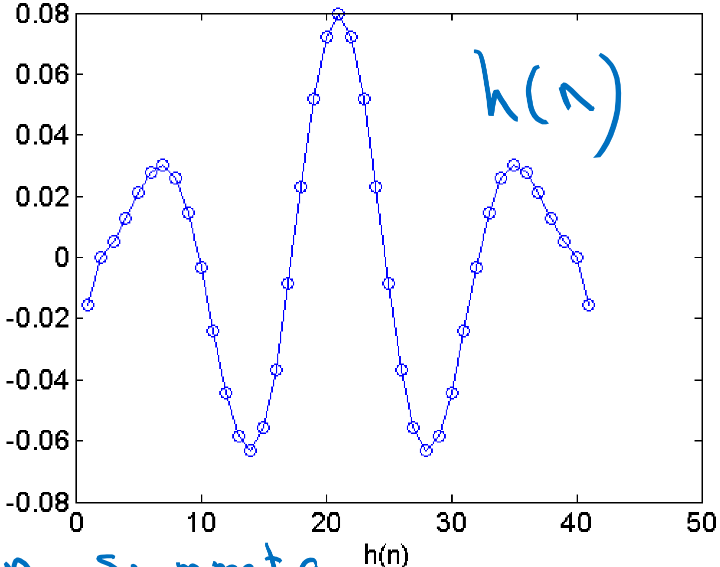
$f = (0, 0.06, 0.12, 0.1467, 0.2067, 1)$

$A = (0, 0, 1, 1, 0, 0)$

$w = [3.33, 1, 3.33] = [\text{ratio}, 1, \text{ratio}]$

weighting per band $\Rightarrow$ larger weight = smaller ripple

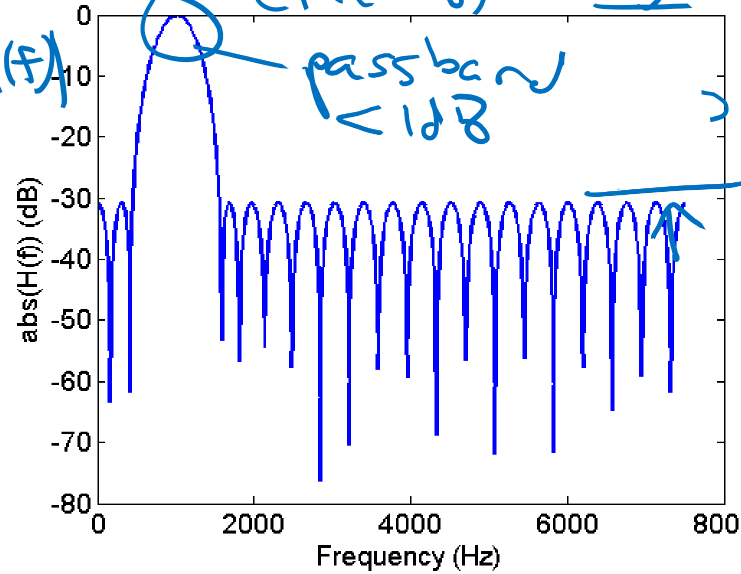
$h = \text{firpm}(N, f, A, w)$ (matlab)



h(n)

Even symmetry $\Rightarrow$ linear phase

|H(f)|



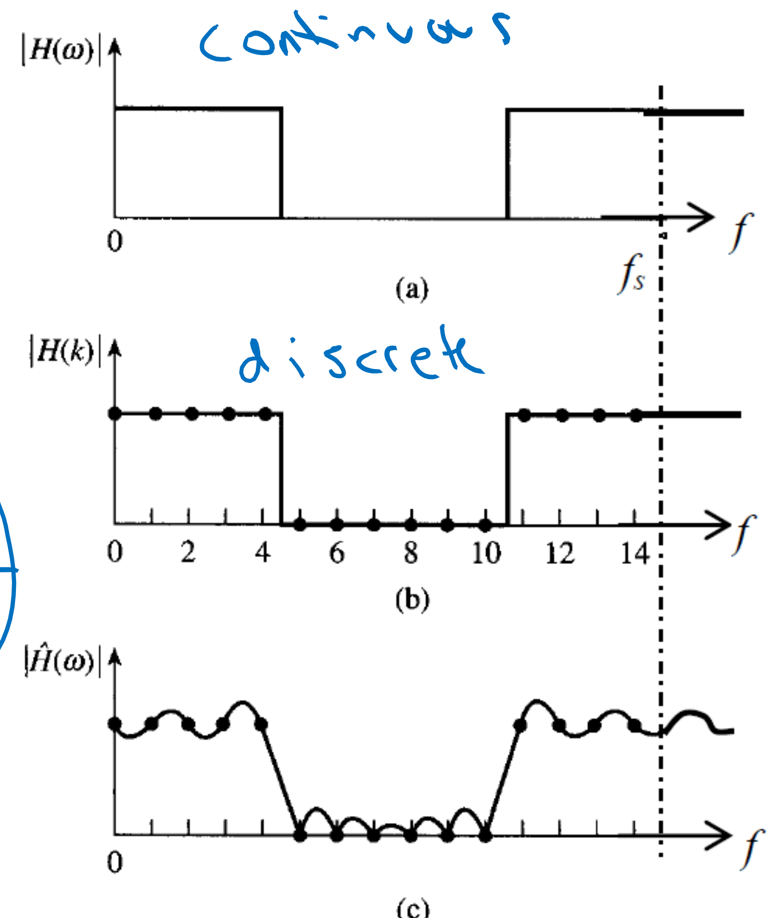
passband < 1dB

> 30 dB

Frequency response

FIR filter design – Frequency sampling Non-recursive (I&J Chapter 7.7)

- For a desired frequency response $H(\omega)$, we can take N samples to get $H(k)$ our sampled frequency response.
- The filter coefficients $h(n)$ are then the inverse discrete Fourier transform of $H(k)$



$$h(n) = \frac{1}{N}\sum_{k=0}^{N-1} H(k)\, e^{j\left(\frac{2\pi k n}{N}\right)}$$

Sampling $H(k)$ at $\frac{kf_s}{N}$

frequency response from N samples

0 Hz or DC $\omega = 0$ $z = e^{j\omega T} = e^{j0} = 1$

Frequency Sampling Schemes

- Four sampling schemes are possible. The number of samples N can be even or odd.
- One of the samples can be at 0 Hz, or two can be placed symmetrically either side of zero.

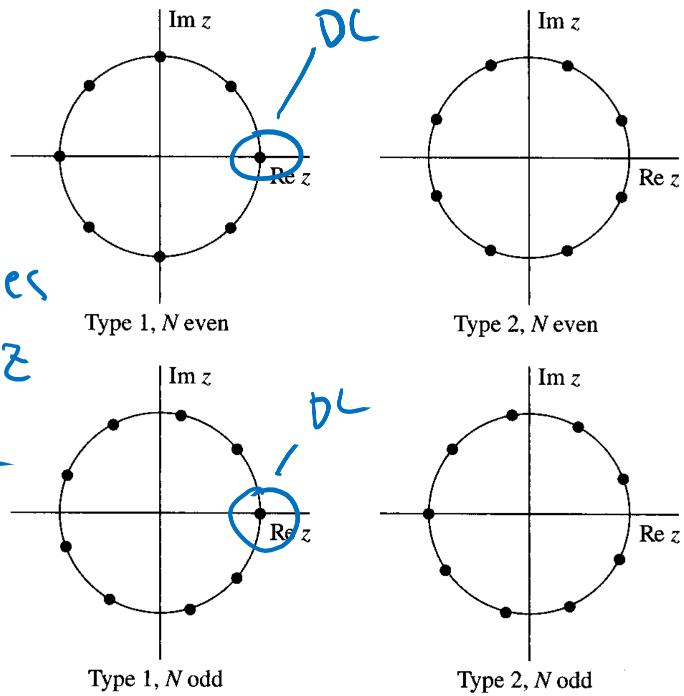
Even number of samples

$\frac{kf_s}{N}$

$k = 0, 1, 2, \ldots N-1$

Samples at 0Hz

Odd number of samples



Even number of samples

Samples Equi-spaced about 0HZ

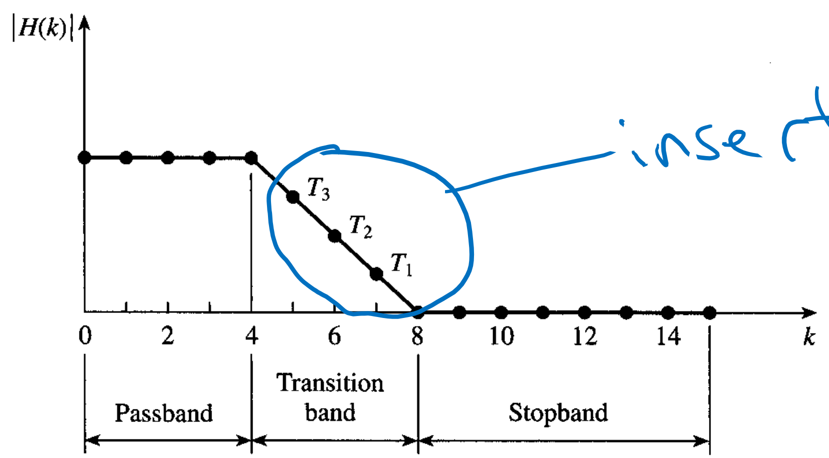
$\left(\frac{2k+1}{2N}\right)f_s$

$k = 0, 1, 2 \ldots N-1$

Odd number of samples

Frequency Sampling - Optimization

- We can improve the response in the stop- and passbands by including one or more transition samples.
- This comes at the expense of widening the transition band (similarly to the window method).
- Table 7.11 of I&J gives the optimal transition band values.



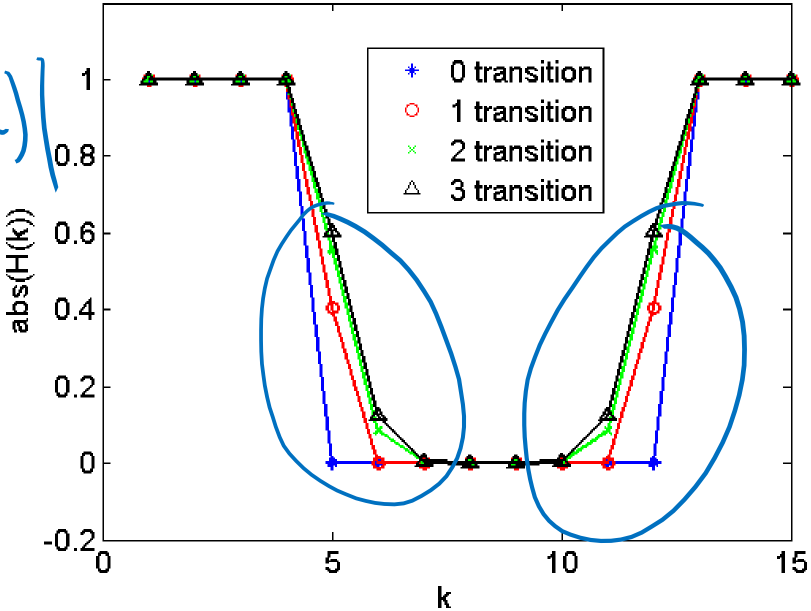
insert transition samples

Figure 7.19 Lowpass filter frequency samples including three transition band samples. Note: because of the symmetry in the amplitude response only one half of the filter response is shown.

Frequency Sampling Example

- A linear phase 15-point FIR filter is characterized by the following frequency sample:
- |H(k)| =1 for k=0,1,2,3
- 0 for k =4,5,6,7.
- Assume a sampling frequency of 2 kHz. Compare the frequency response of 0, 1, 2 and 3 transition band frequency samples.

|H(k)|



Values from Table 7.11 in I&J

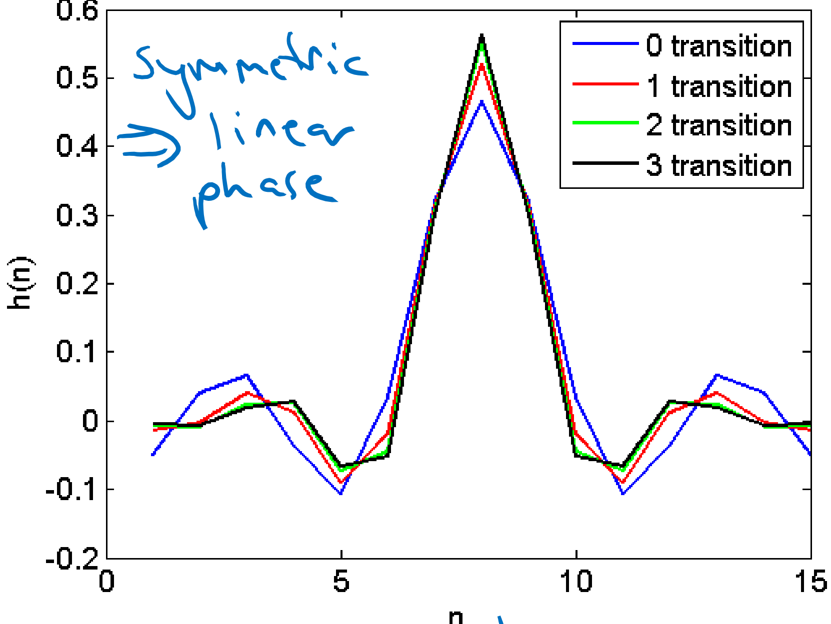
transition samples

Frequency Sampling Example 2

Calculate FIR filter coefficients

h(n) = real (ifft (H(k)))

narrower with more transition samples



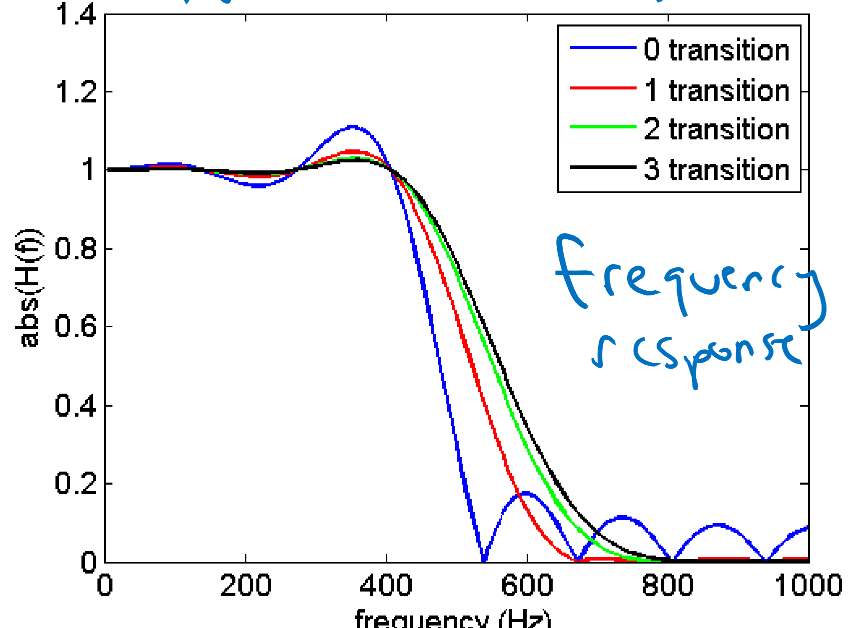
coefficients or impulse response

Calculate frequency response

h'(n) = zeropad(h(n))

abs(fft (h'(n)))

wider with more transition samples



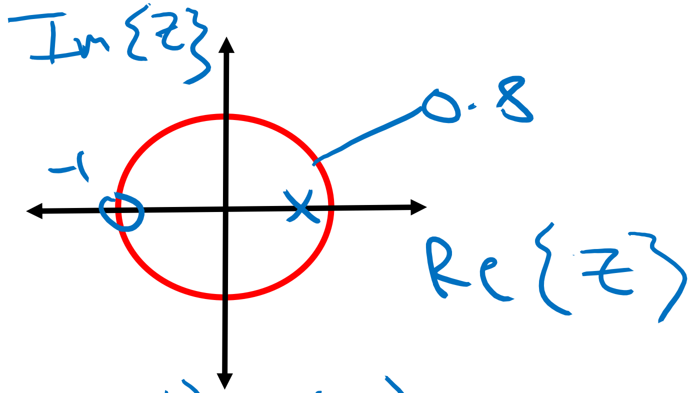
Less ringing in pass & stopbands with more transition samples

A tale of 2 filters

- Let's compare FIR and IIR realizations of a simple low pass filter.

1. IIR – A single pole filter with transfer function

$$H(z) = 0.1\frac{(z+1)}{(z-0.8)} = \frac{Y(z)}{X(z)}$$



$$\frac{Y(z)}{X(z)} = 0.1\left(\frac{1+z^{-1}}{1-0.8z^{-1}}\right)$$

$$(1-0.8z^{-1})Y(z) = 0.1(1+z^{-1})X(z)$$

inverse z transform

$$y(n) = 0.1x(n) + 0.1x(n-1) + 0.8y(n-1)$$

Difference equation

$A = (1, -0.8, 0)$ $\quad B = (0.1, 0.1, 0)$

matlab

python

IIR implementation

```
A=[1,-0.8,0]
B=[0.1,0.1,0]
fs=50
N=19
```

sampling frequency

number of points

```
[H,F]=freqz(B,A,N,fs);
```

frequency vector (absolute)

```
figure(1)
set(gca,'FontSize',16)
plot(F,abs(H))
xlabel('Frequency (Hz)')
ylabel('Magnitude')
title('IIR filter')

figure(3)
set(gca,'FontSize',16)
plot(F,angle(H))
xlabel('Frequency (Hz)')
ylabel('Phase (radians)')
title('IIR filter')
```

```
import numpy as np
import matplotlib.pyplot as plt
from scipy import signal

plt.rc('xtick', labelsize=14)
plt.rc('ytick', labelsize=14)

#IIR filter first up

A=np.array([1,-0.8,0])
B=np.array([0.1,0.1,0])
fs=50
N=19

w, H =signal.freqz(B,A,worN=N);
f = fs * w / (2*np.pi)

plt.figure(1)
plt.plot(f,abs(H), linewidth=2)
plt.xlabel('Frequency (Hz)', fontsize=16)
plt.ylabel('Magnitude', fontsize=16)
plt.title('IIR filter', fontsize=16)
plt.savefig('IIR_magnitude.png')

plt.figure(3)
plt.plot(f,np.angle(H), linewidth=2)
plt.xlabel('Frequency (Hz)', fontsize=16)
plt.ylabel('Phase (radians)', fontsize=16)
plt.title('IIR filter', fontsize=16 )
plt.savefig('IIR_phase.png')
```

w not f relative to f_s

FIR design

$$H(z) = \frac{0.1(z+1)}{z-0.8}$$

- Let's use frequency sampling method

let $z = e^{jw}$

$$H(w) = \frac{0.1 e^{jw} + 1}{e^{jw} - 0.8}$$

Sample at points $k = 0, 1, \ldots N-1$

$$w_k = \frac{2\pi k}{N}$$

$$h(n) = \mathcal{F}^{-1}\left\{ H(w_k) \right\}$$

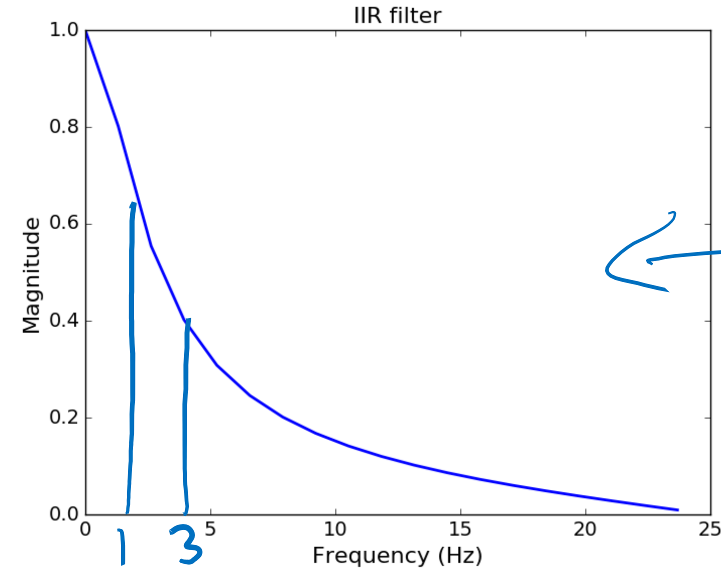
impulse response

Sampled frequency response

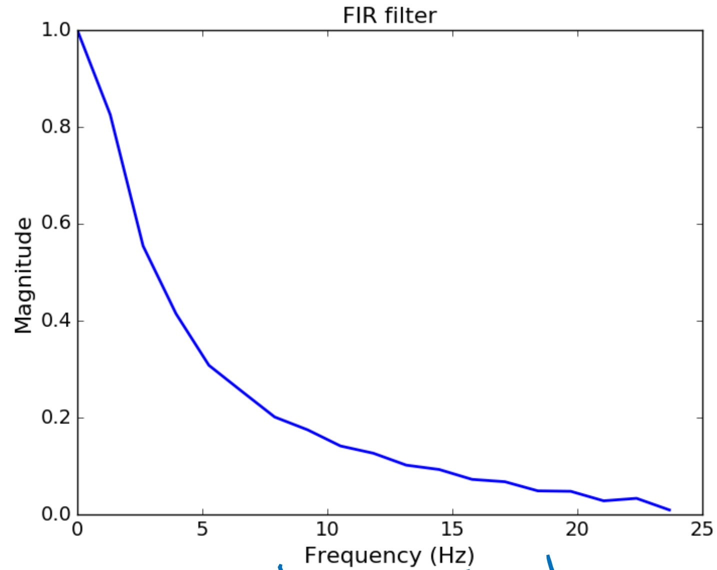
IIR

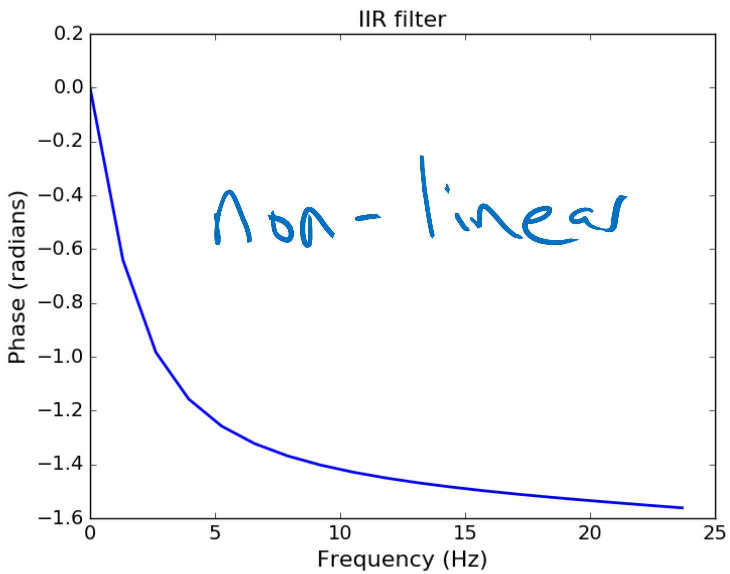
Magnitude & Phase

FIR

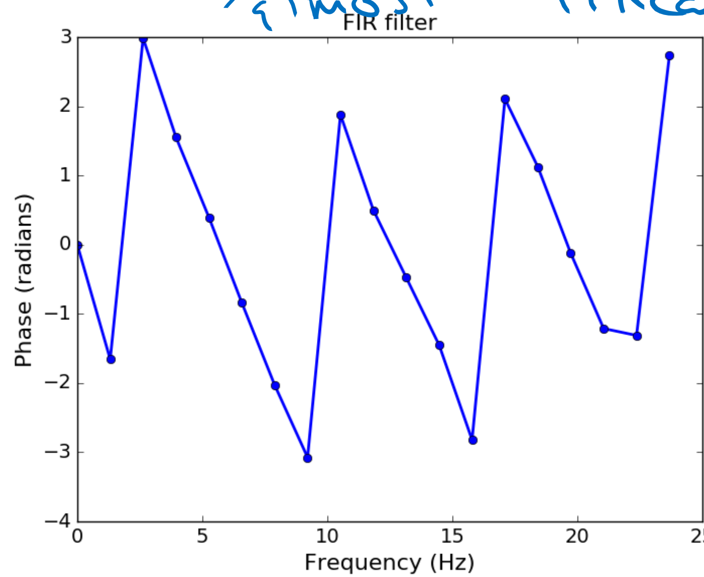


'Same'





'almost' linear

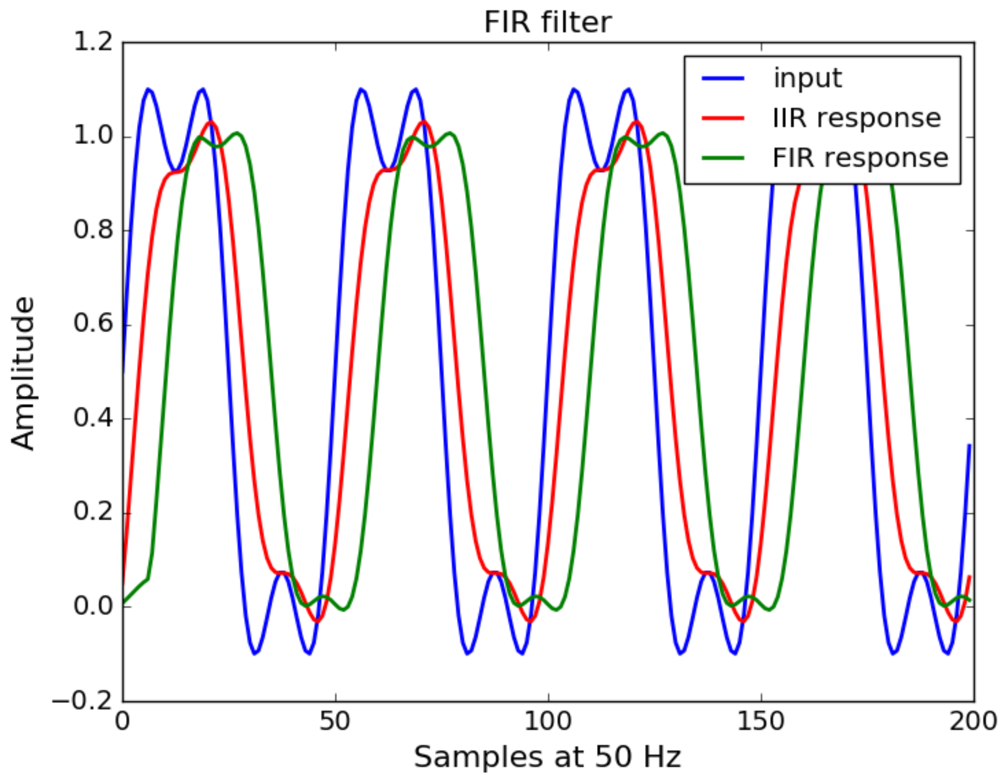


Fourier series first 3 terms for 1 Hz square wave

Effect of phase response

- Consider our FIR and IIR filters' response to the signal

$x(t) = 0.5 + \frac{2}{\pi}\sin(2\pi t) + \frac{2}{3\pi}\sin(6\pi t)$



1) FIR delayed by more than IIR
2) 3 Hz is attenuated by more than 1 Hz as expected.
3) IIR response is assymmetric, 3 Hz is delayed by more
4) FIR response is symmetric (1 & 3 Hz are equally delayed)