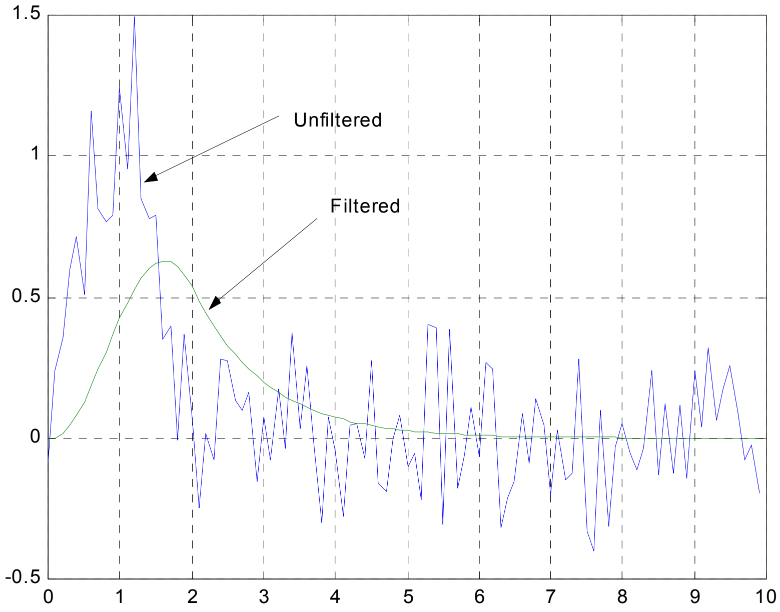
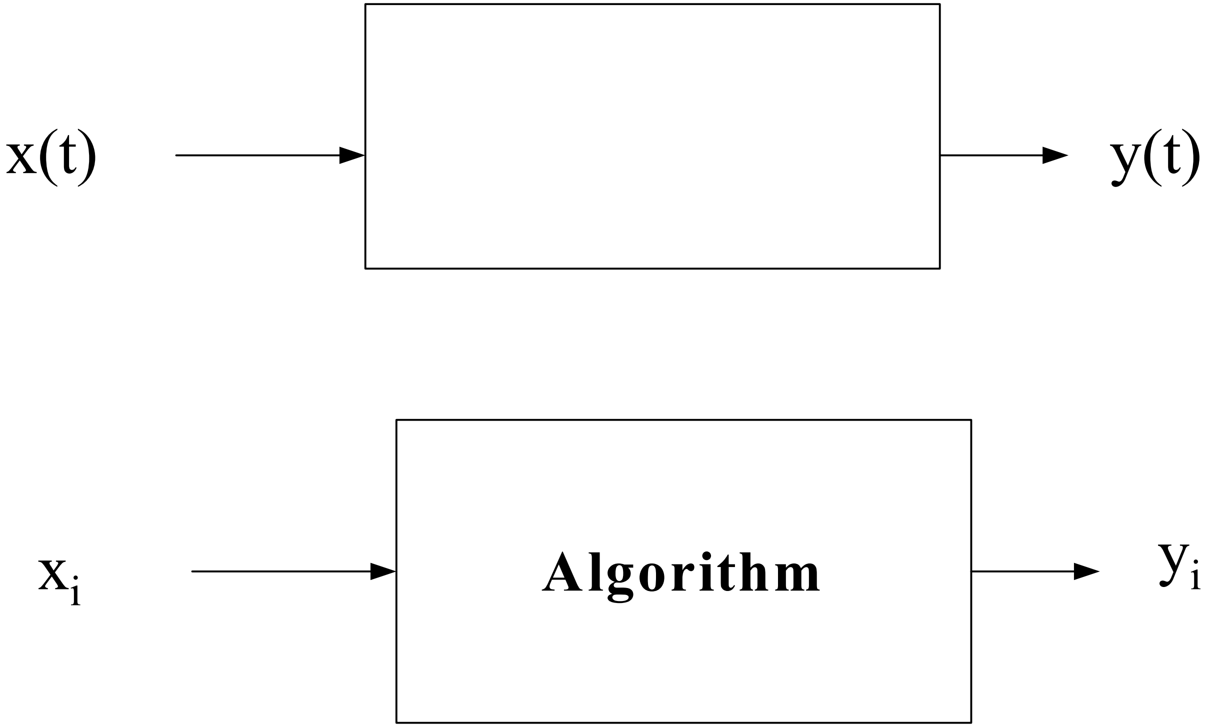
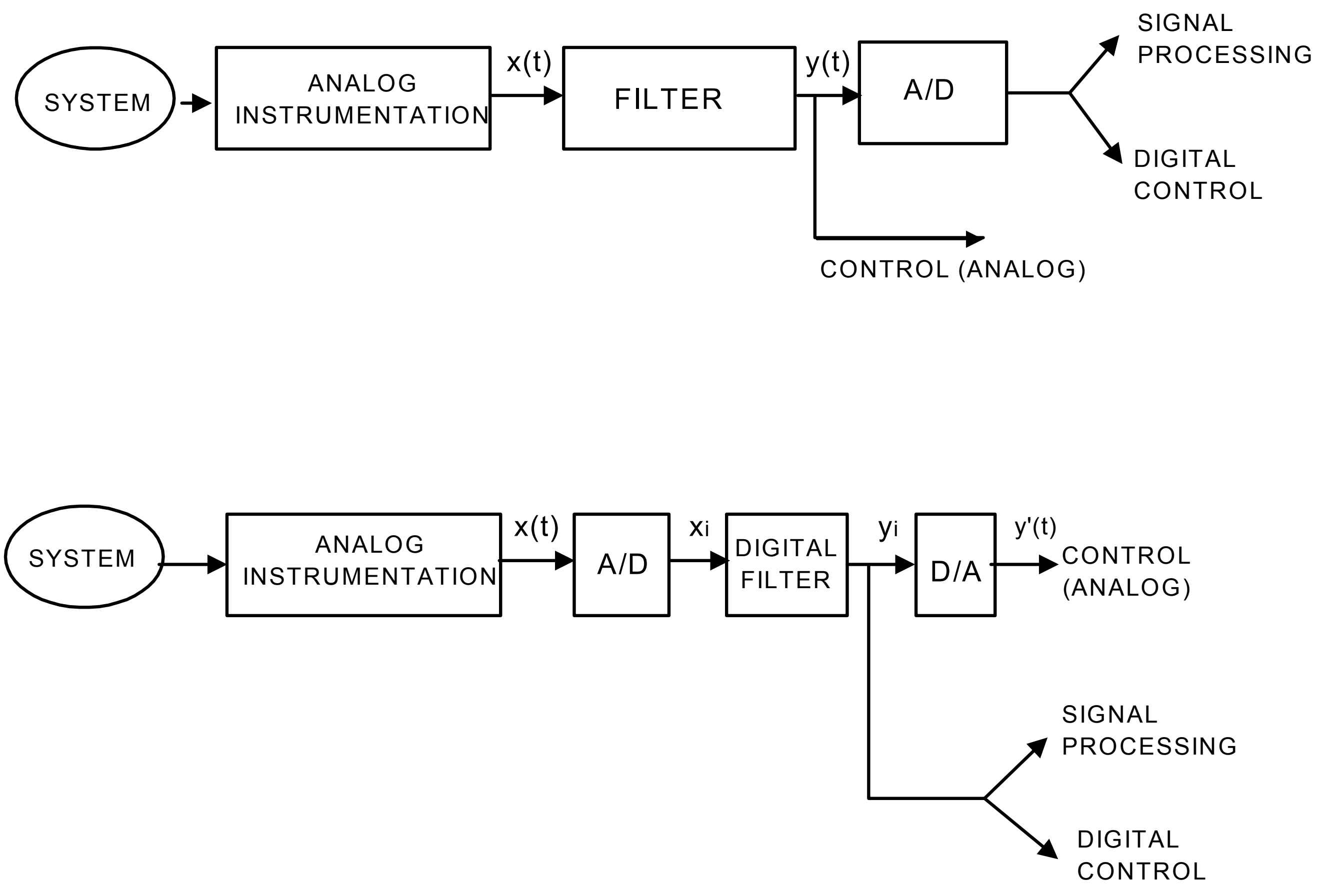
Filters

Reject/pass signal components according to frequency



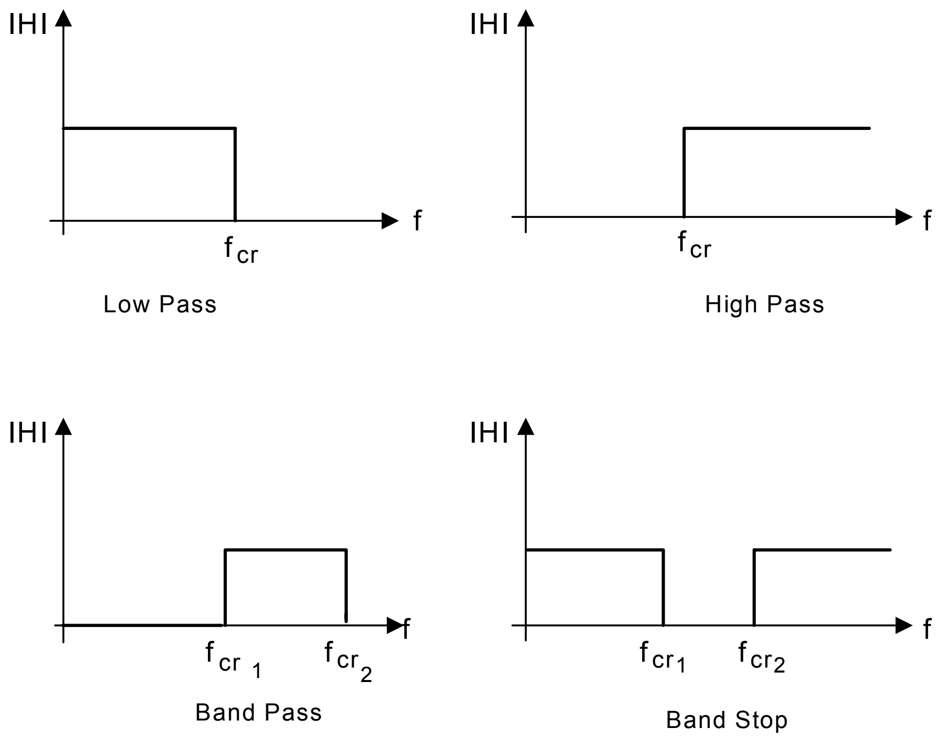
Algorithm as system





Signal Processing: Filters

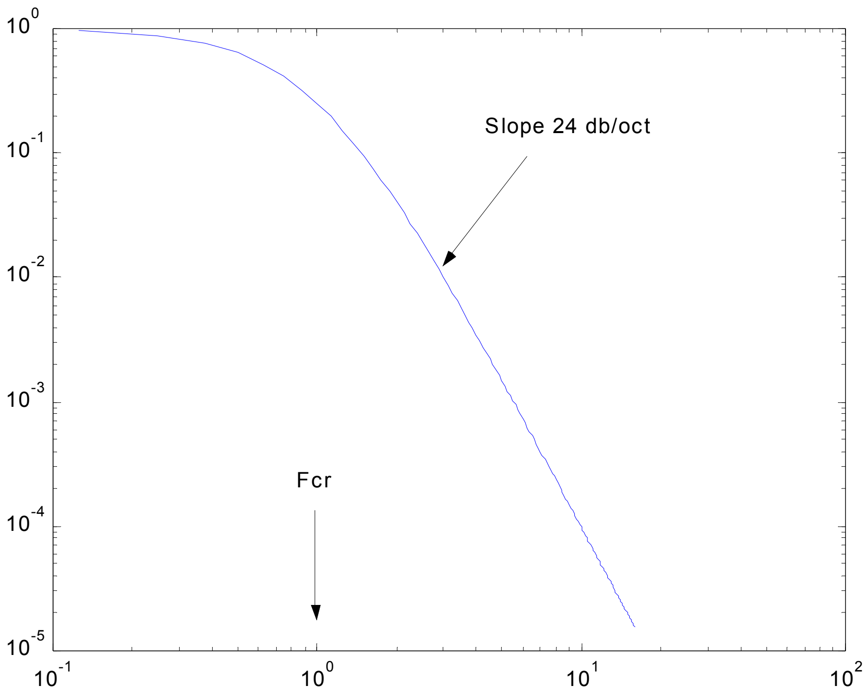
Filters: Classification

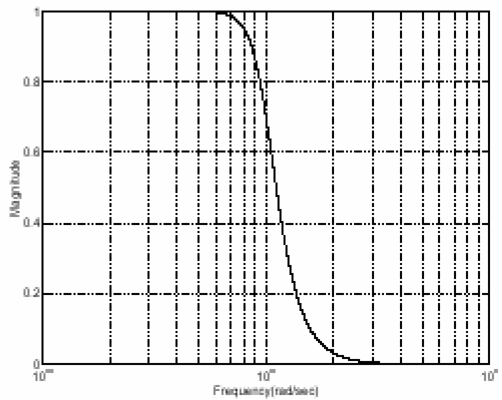


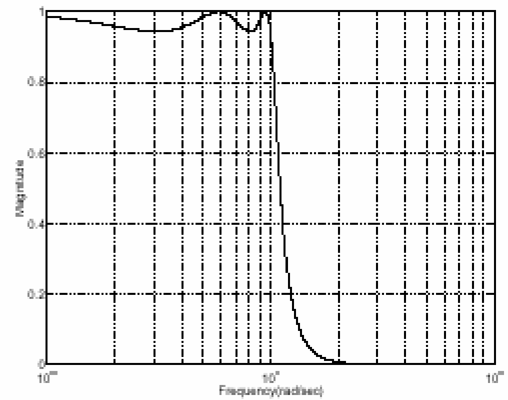
Parameters:

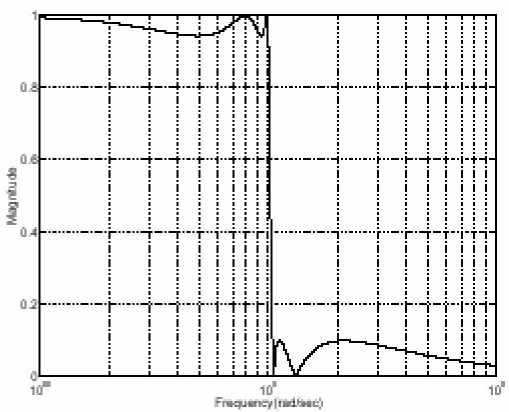
Critical frequency (ies)

Slope









1. **Slope**
2. **Attenuation**
3. **Ripple in pass (stop) band**

Description of Linear System

* Differential (difference) Equation

* Transfer Function

* Frequency Response Function $H(i\omega) = |H| \langle \Phi$

* Impulse Response h(t)

* Zero/pole locations

$$y_i = \sum_{n=1}^{N} -a_n y_{i-n} + \sum_{n=0}^{M} b_n x_{i-n}$$

$$Y(z) = [-a_1 z^{-1} - a_2 z^{-2} - \ldots a_N z^{-N}]\, Y(z) + [b_0 + b_1 z^{-1} + \ldots\ldots b_M z^{-M}]\, X(z)$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} \ldots. + b_M z^{-M}}{1 + a_1 z^{-1} + \ldots\ldots\ldots a_N z^{-N}}$$

$$H(z) = \frac{\prod (z - z_i)}{\prod (z - p_i)}$$

$$y_i = \sum_{n=1}^{N} -a_n y_{i-n} + \sum_{n=0}^{M} b_n x_{i-n}$$

$$\delta_i = \begin{cases} 1 & i = 0 \\ 0 & i \neq 0 \end{cases} \qquad x_i = \delta_i \rightarrow h_i = y_i$$

$$H(j\omega) = H(z)\Big|_{z = exp(j\omega\Delta t)}$$

IIR $$y_i = \sum_{n=1}^{N} - a_n y_{i-n} + \sum_{n=0}^{M} b_n x_{i-n}$$

FIR $$y_i = \sum_{n=0}^{M} b_n x_{i-n}$$

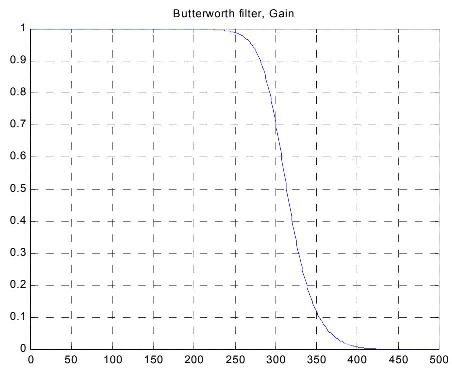
FIR:

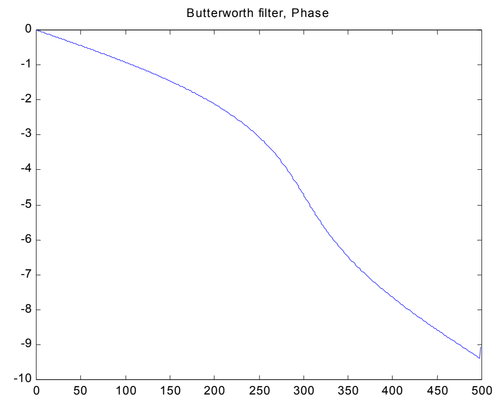
Always stable

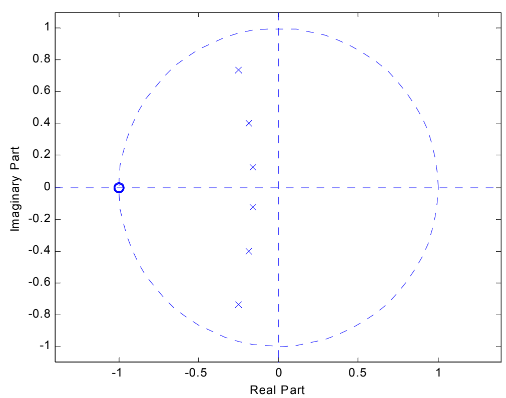
Can have linear phase

Much higher order needed

IIR Filter – Butterworth (order 6)





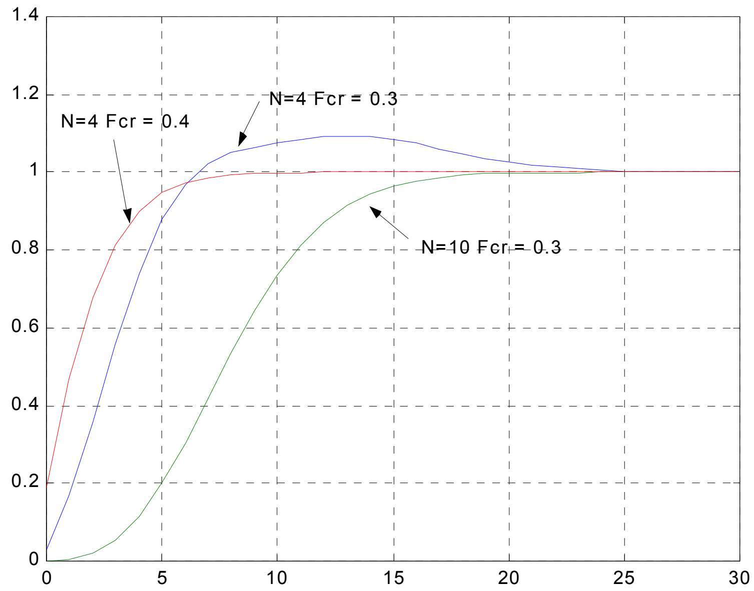


2 Step Procedure

1. Design
2. Apply

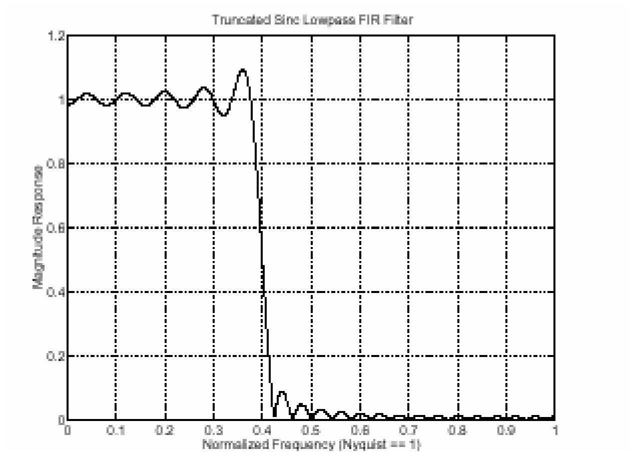
Example (Matlab)

```
[b,a]=butter(6,0.4);   %Design
Y=filter(b,a,x)        % Perform filtering
```



1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
5
10
15
20
25
30
N=4 Fcr = 0.4
N=4 Fcr = 0.3
N=10 Fcr = 0.3

FIR – window method



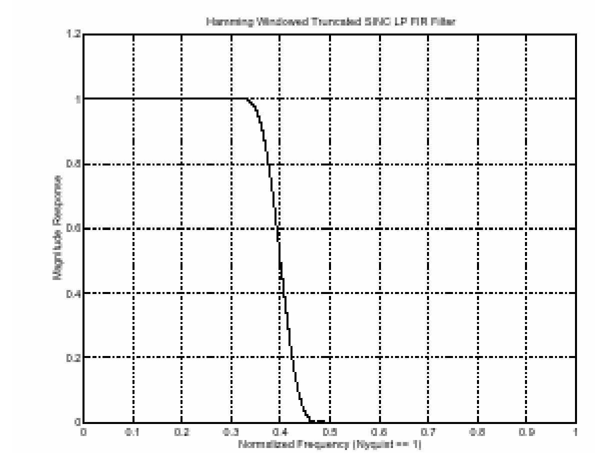
$$h(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H(\omega)e^{j\omega n}d\omega = \frac{1}{2\pi}\int_{-\omega_0}^{\omega_0} e^{j\omega n}d\omega = \frac{\omega_0}{\pi}\mathrm{sinc}(\frac{\omega_0}{\pi}n)$$

This filter is not implementable since its impulse response is infinite and noncausal. To create a finite-duration impulse response, truncate it by applying a window. By retaining the central section of impulse response in this truncation, you obtain a linear phase FIR filter. For example, a length 51 filter with a lowpass cutoff frequency ω_0 of 0.4π rad/sec is

```
b = 0.4*sinc(0.4*(-25:25));
```

The window applied here is a simple rectangular or "boxcar" window. By Parseval's theorem, this is the length 51 filter that best approximates the ideal lowpass filter, in the integrated least squares sense. To view its frequency response,

```
[H,w] = freqz(b,1,512,2);
plot(w,abs(H)), grid
```

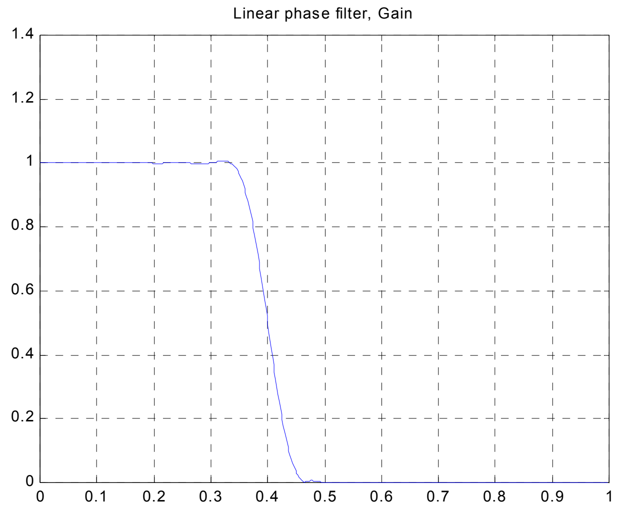


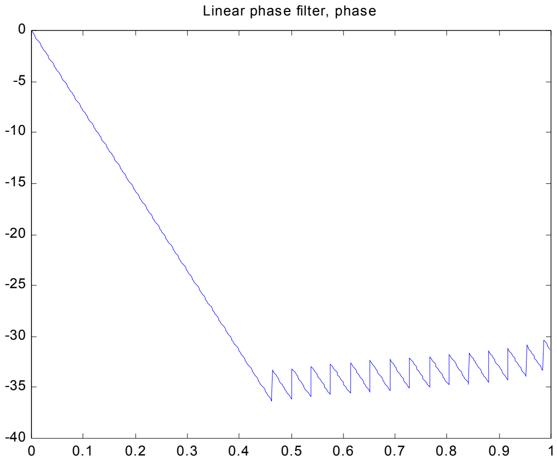
Apply a length 51 Hamming window to the filter:

```
b = b.*hamming(51)';
[H,w] = freqz(b,1,512,2);
plot(w,abs(H)), grid
```

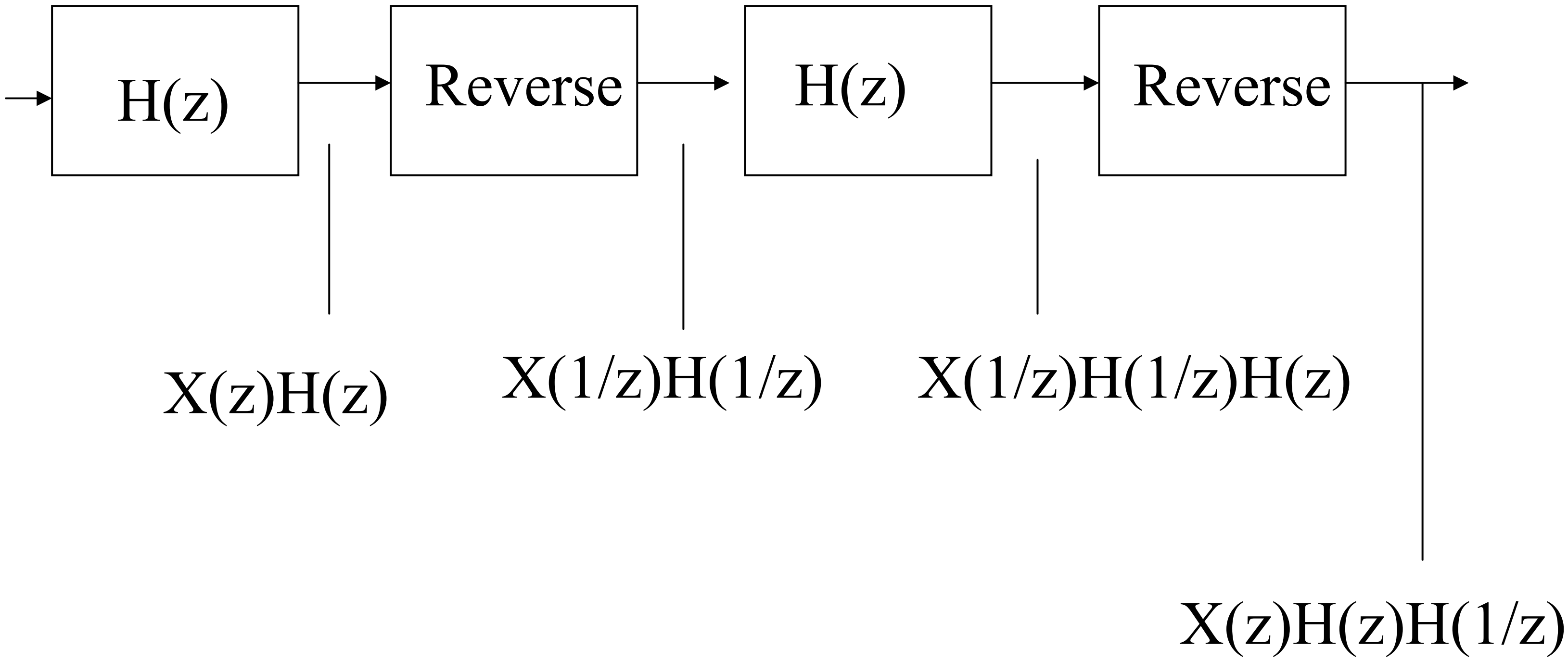
Linear phase filter

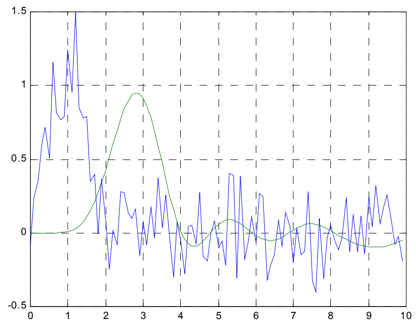
Window method -Hanning



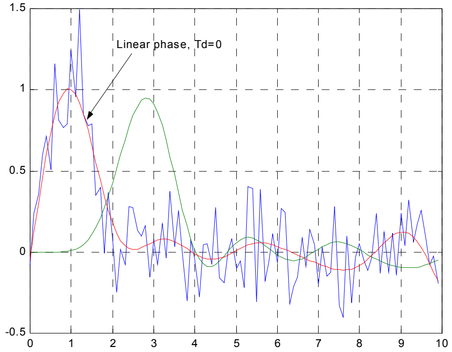


Linear phase with IIR





IIR low pass filter



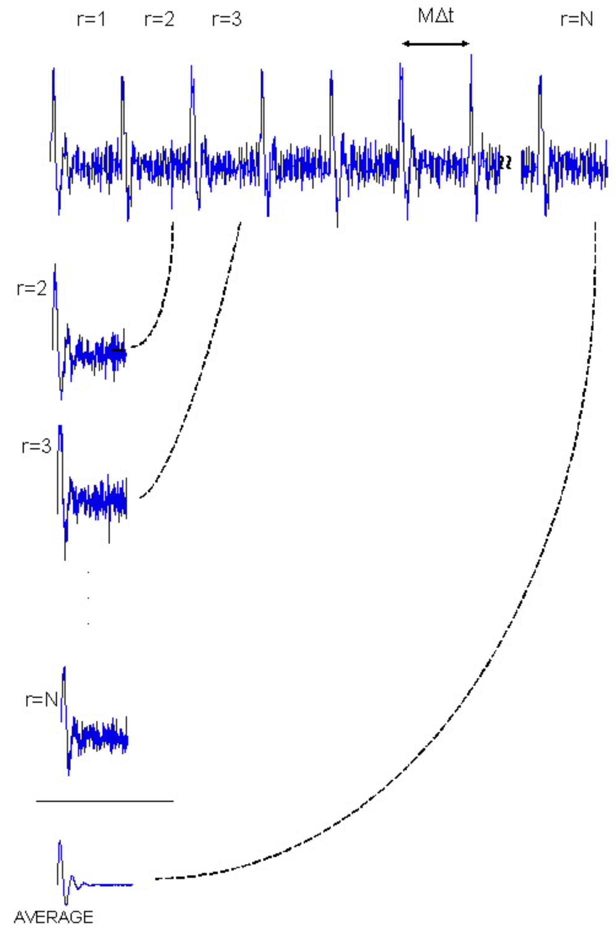
Linear phase filter (zero delay)

Time Domain Averaging

(Synchronous Averaging)

$$y(n\Delta t) = \frac{1}{N}\sum_{r=0}^{N-1} x(n\Delta t - rM\Delta t)$$

$$y_r(n\Delta t) = y_{r-1}(n\Delta t) + \frac{x_r(n\Delta t) - y_{r-1}(n\Delta t)}{r}$$

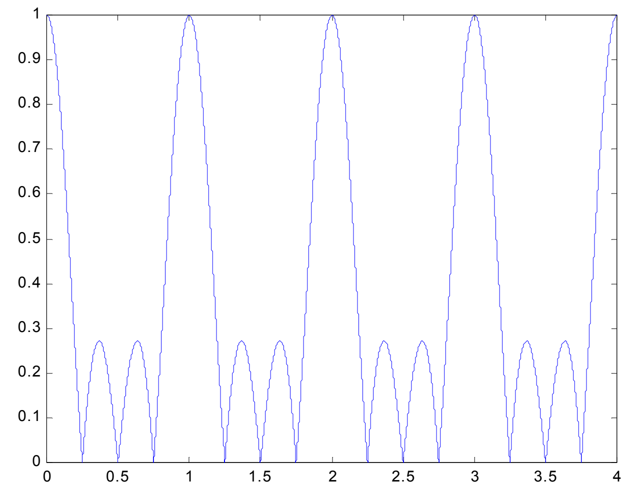


$$H(z) = \frac{1}{N}\frac{1 - z^{-MN}}{1 - z^{-N}}$$

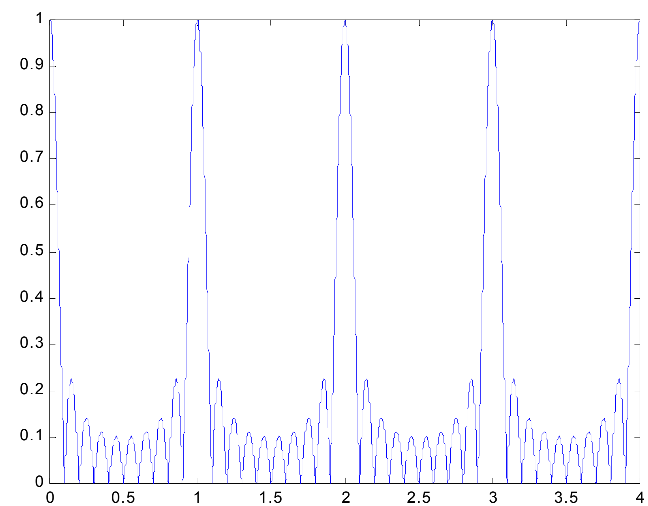
$$H(f/f_p) = \frac{1}{N}\frac{\mathrm{Sin}(\pi N f/f_p)}{\mathrm{Sin}(\pi f/f_p)}$$

$$\phi_H(f/f_p) = -\pi(N-1)f/f_p$$

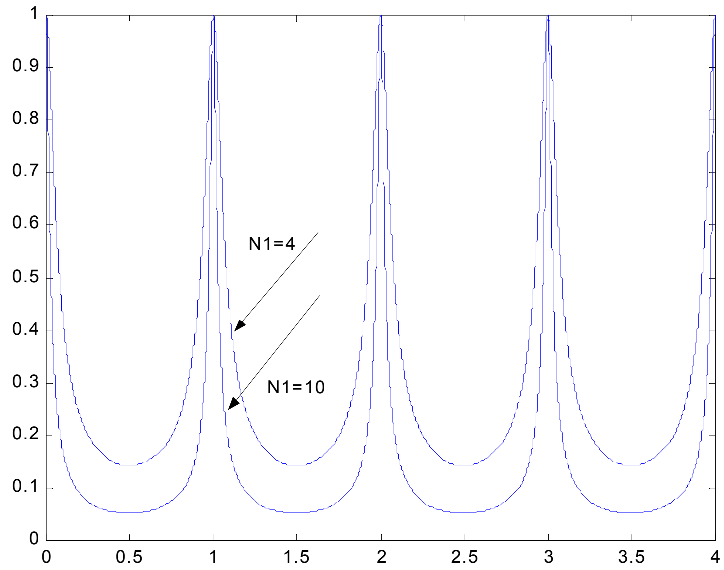
$$f_p = \frac{1}{MN\Delta t}$$



N=4



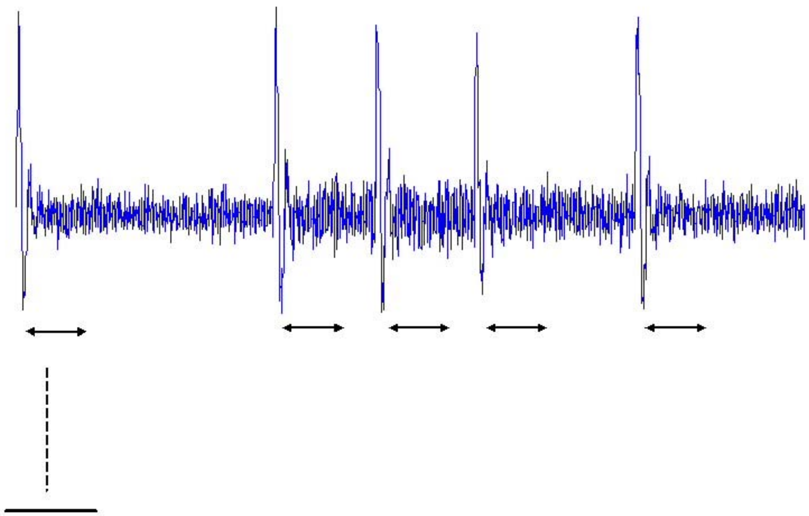
N=10

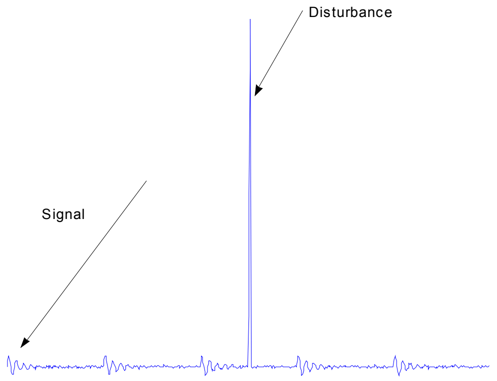


$$y_r(n\Delta t) = y_{r-1}(n\Delta t) + \frac{x_r(n\Delta t) - y_{r-1}(n\Delta t)}{N_1}$$

$$y_r(n\Delta t) = y_{r-1}(n\Delta t) + \frac{x_r(n\Delta t) - y_{r-1}(n\Delta t)}{r}$$

Triggered TDA





Signal Processing: Filters