Signal Processing

Physics 77

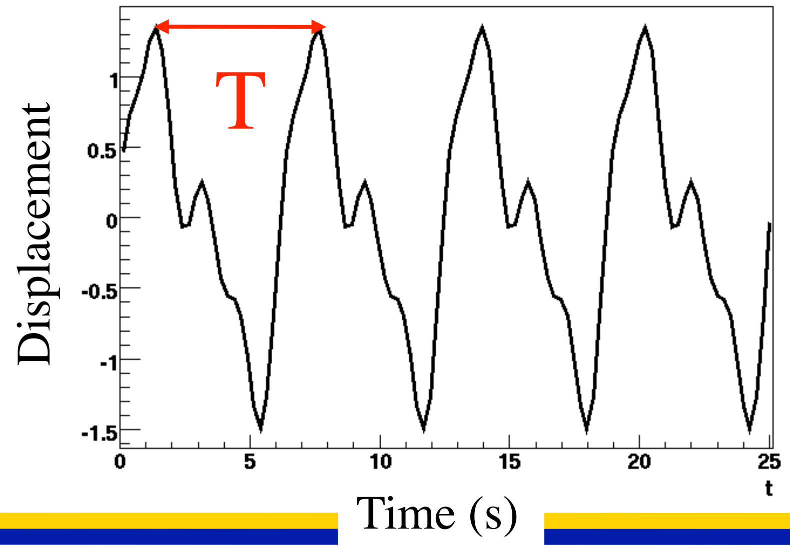
April 29, 2019

Outline

- Definitions
- Discrete Fourier Transforms (DFT and FFT)
- Finite-response filters
- Infinite-response filters

Definitions

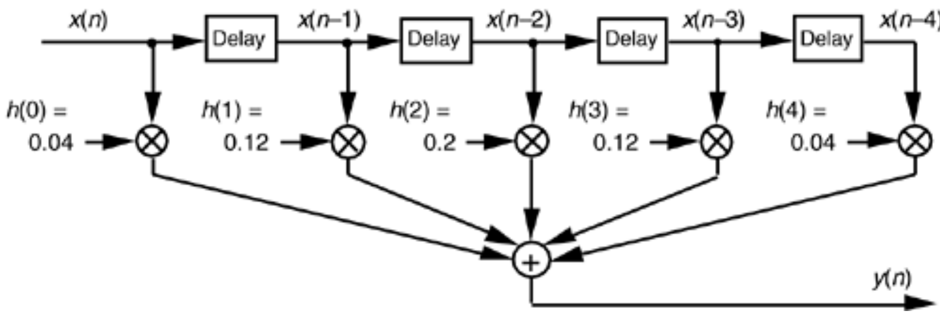
- Suppose $x(t)$ is some (periodic) function that we sample at some fixed frequency f_s
  - ☞ Measure $x(n)$ at times $t_n = t_0+nt_s$
- ❑ Goal is to analyze $x(n)$ and infer properties of $x(t)$
  - ☞ E.g. magnitude $|x(t)|$, power vs time $P(t)\sim|x(t)|^2$, spectral composition, etc.



Acronyms

- FFT = Fast Fourier Transform
  - ☞ DFT = Discrete Fourier Transform
- DSP = Digital Signal Processing
- SNR = Signal-to-noise ratio
  - ❑ Often expressed in dB, i.e. SNR = $20\log_{10}(S/N)$ where S and N are signal and noise *amplitudes*
    - ☞ For power, SNR = $10\log_{10}(P_S/P_N)$
- ASD = Amplitude Spectral Density
  - ☞ PSD = Power Spectral Density
  - ☞ NPS = Noise Power Spectrum
- FIR = Finite Impulse Response (filter)
- IIR = Infinite Impulse Response (filter)

Schematic Notation



- Represent addition, multiplication (mixing), delay operations
- Most linear systems can be represented this way
  - ☞ I.e. $c_1x_1(n)+c_2x_2(n) \rightarrow c_1y_1(n) + c_2y_2(n)$

Fourier Transform

- Most common way to analyze a periodic function is by means of a Fourier Transform
  - ☞ Represent as a superposition of harmonic oscillations

$$x(t) = C\int_{-\infty}^{+\infty} x(\omega)\exp[j\omega t]\,\mathrm{d}\omega$$

Here x(ω) is a Fourier coefficient, which represents the strength of the periodic signal at particular frequency. Also known as spectral density.

Inverse transform:

$$x(\omega) = C\int_{-\infty}^{+\infty} x(t)\exp[-j\omega t]dt$$

$$x(f) = C\int_{-\infty}^{+\infty} x(t)\exp[-j2\pi ft]dt$$

$$= C\int_{-\infty}^{+\infty} x(t)\left[\cos(2\pi ft) - j\sin(2\pi ft)\right]dt$$

Discrete Fourier Transform

- For a sampled waveform, replace the integral with a discrete sum:

$$x(m) = \sum_{n=0}^{N-1} x(n) \left[\cos \frac{2\pi mn}{N} - j \sin \frac{2\pi nm}{N} \right]$$

m=0..N-1

  - Define

$$x_{\text{mag}}(m) = |x(m)| = \sqrt{x_{\text{re}}(m)^2 + x_{\text{im}}(m)^2}$$

$$\Delta\phi(m) = \text{phase}(m) = \tan^{-1}[x_{\text{im}}(m)/x_{\text{re}}(m)]$$

$$P(m) = |x(m)|^2 = x^*(m)x(m) = x_{\text{re}}(m)^2 + x_{\text{im}}(m)^2$$

Aliasing

- For real signals *x(n)*, can show that
  - Exercise for the reader

$$x(m) = x^*(N - m)$$

  - Moreover (obvious)

$$x(m) = x(N + m)$$

  - This is called aliasing. Spectrum for $0<m<N/2$ is redundant with $N/2<m<N$, $N<m<3N/2$, etc
  - Nyquist theorem:
    - ☞ $x(t)$ can be reconstructed perfectly from $x(m)$ iff $x(t)$ is limited to the band $f<f_B$ and $fs>2f_B$
    - ☞ $f_N = f_s/2$ is often called the Nyquist frequency

Fast Fourier Transform

- Brute-force discrete Fourier transforms can be slow
  - Require N^2 calculations
  - However, trig functions are symmetric and obey trig relations, which is used in the Fast Fourier Transform algorithms (FFT)
    - ☞ Scale as N log(N)
    - ☞ Multiple FFT algorithms exist, and are interfaced in Python
    - ☞ E.g. FFTW from MIT

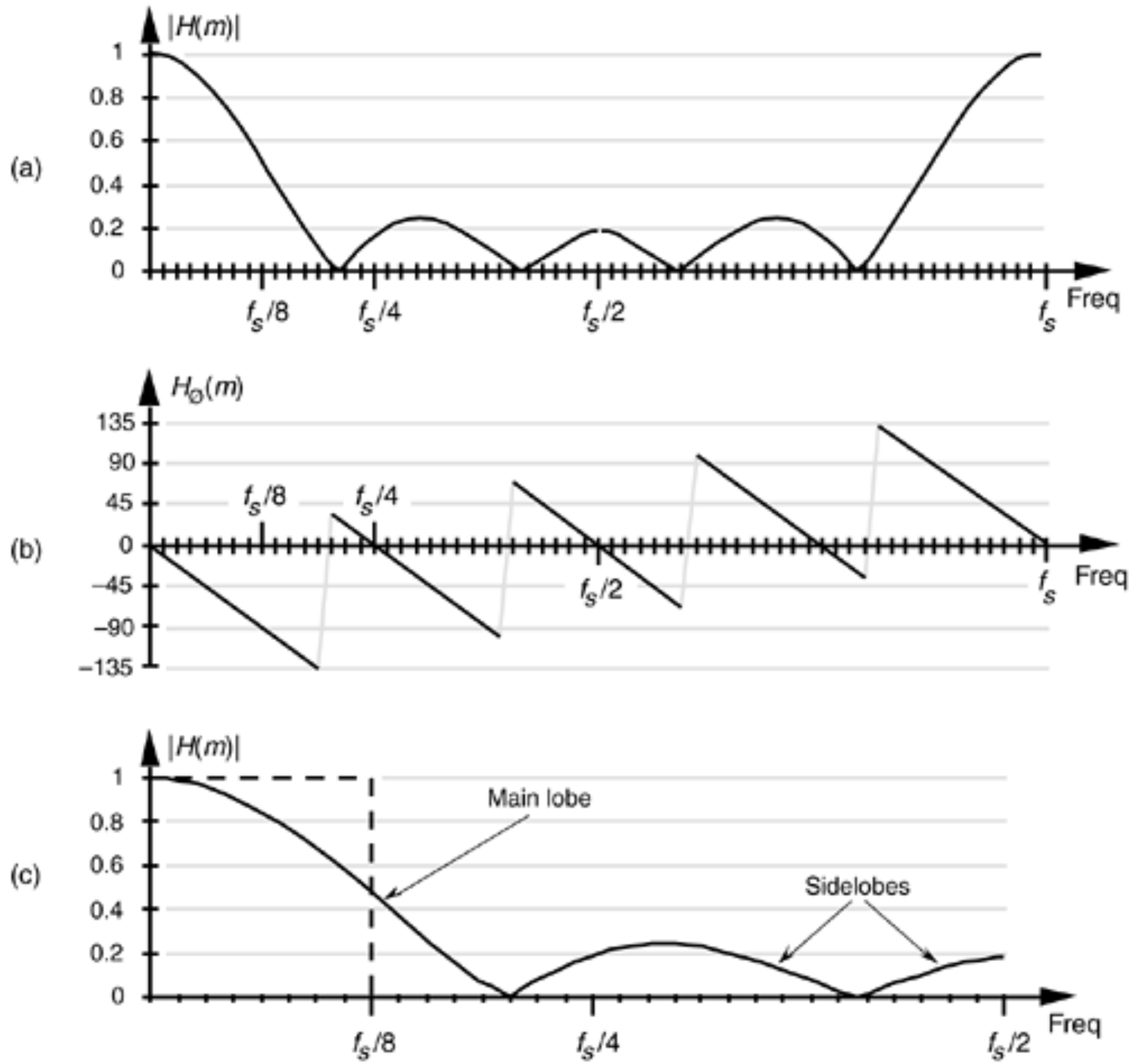
Applications of FFT

- FFT is useful to
  - Look at the composition of a periodic signal, e.g. harmonics, tone analysis, etc
  - Look at the composition of noise
    - ☞ Line noise
    - ☞ White noise (flat in frequency)
    - ☞ Pink (1/f) noise
  - Signal processing
    - ☞ Measure amplitude of the signal in “frequency domain”
    - ☞ Design filters to maximize SNR

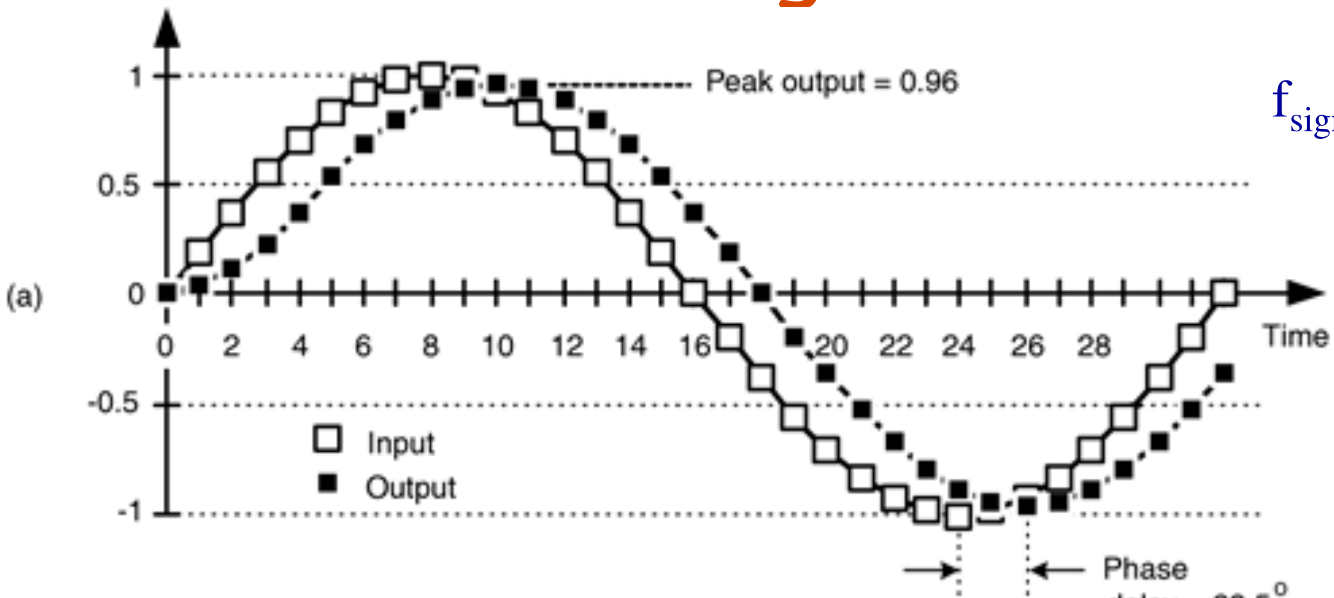
FIR filters

- Filtering (in the time domain) is a way to reduce the contributions of (random) noise to the characteristics of the signal
- Usually implemented by summing over samples with some (predetermined) weights
  - Equivalent to integration/averaging

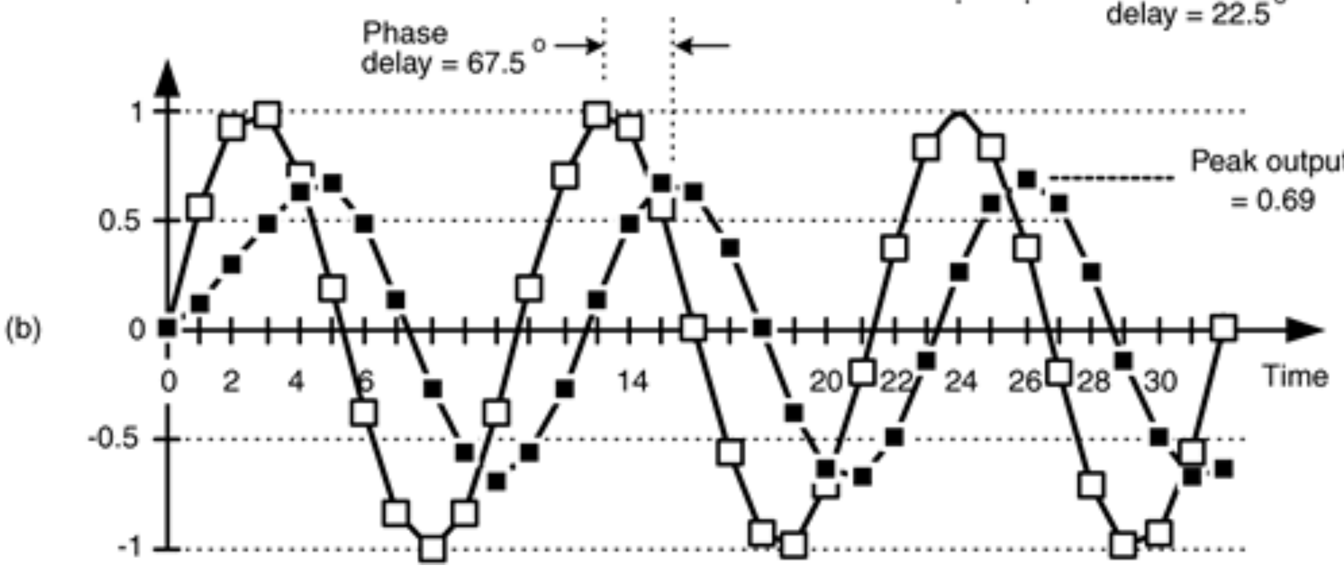
Box-car Filter Response



Time-Domain Signal After Filtering

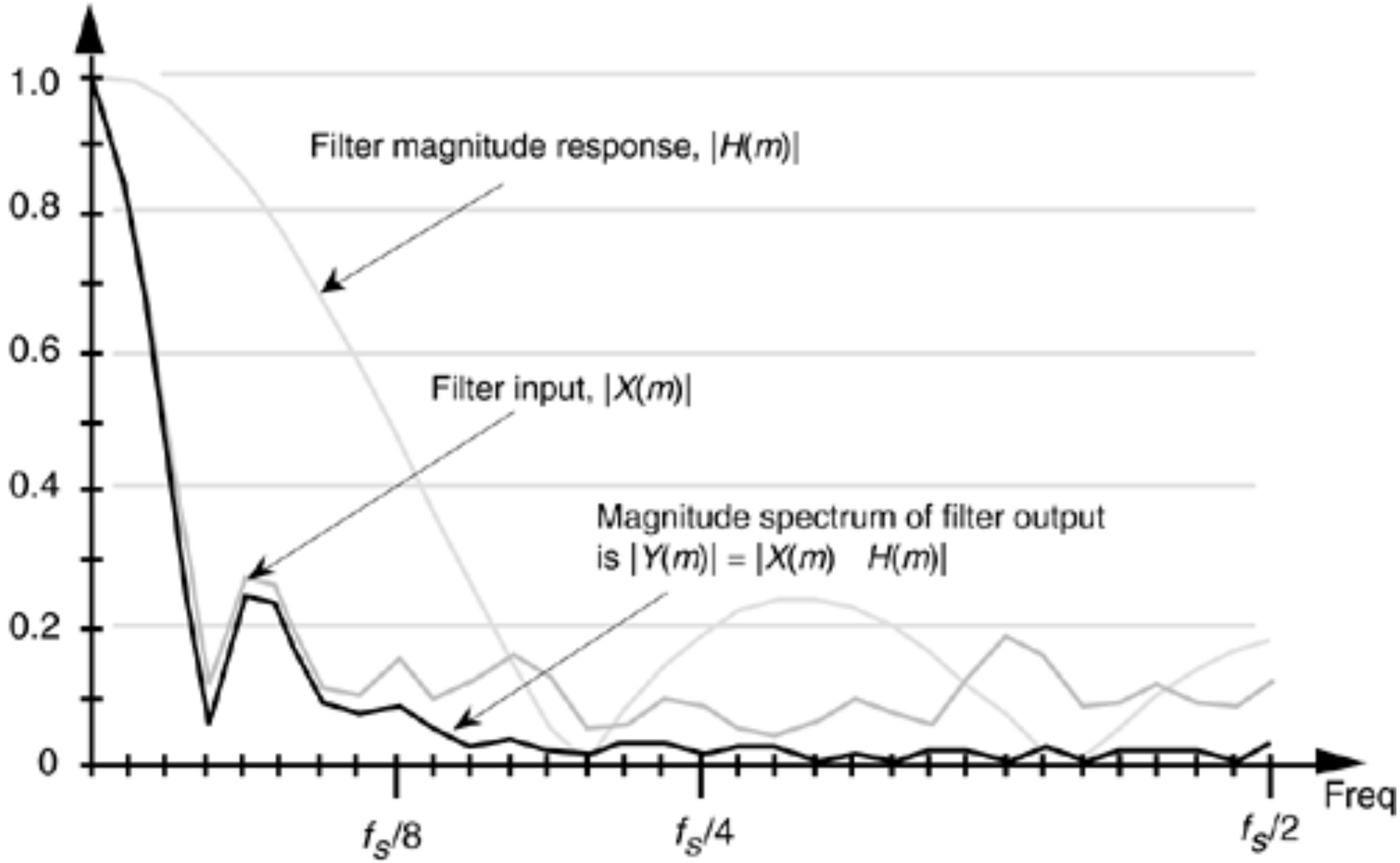


$f_{signal}=f_s/32$

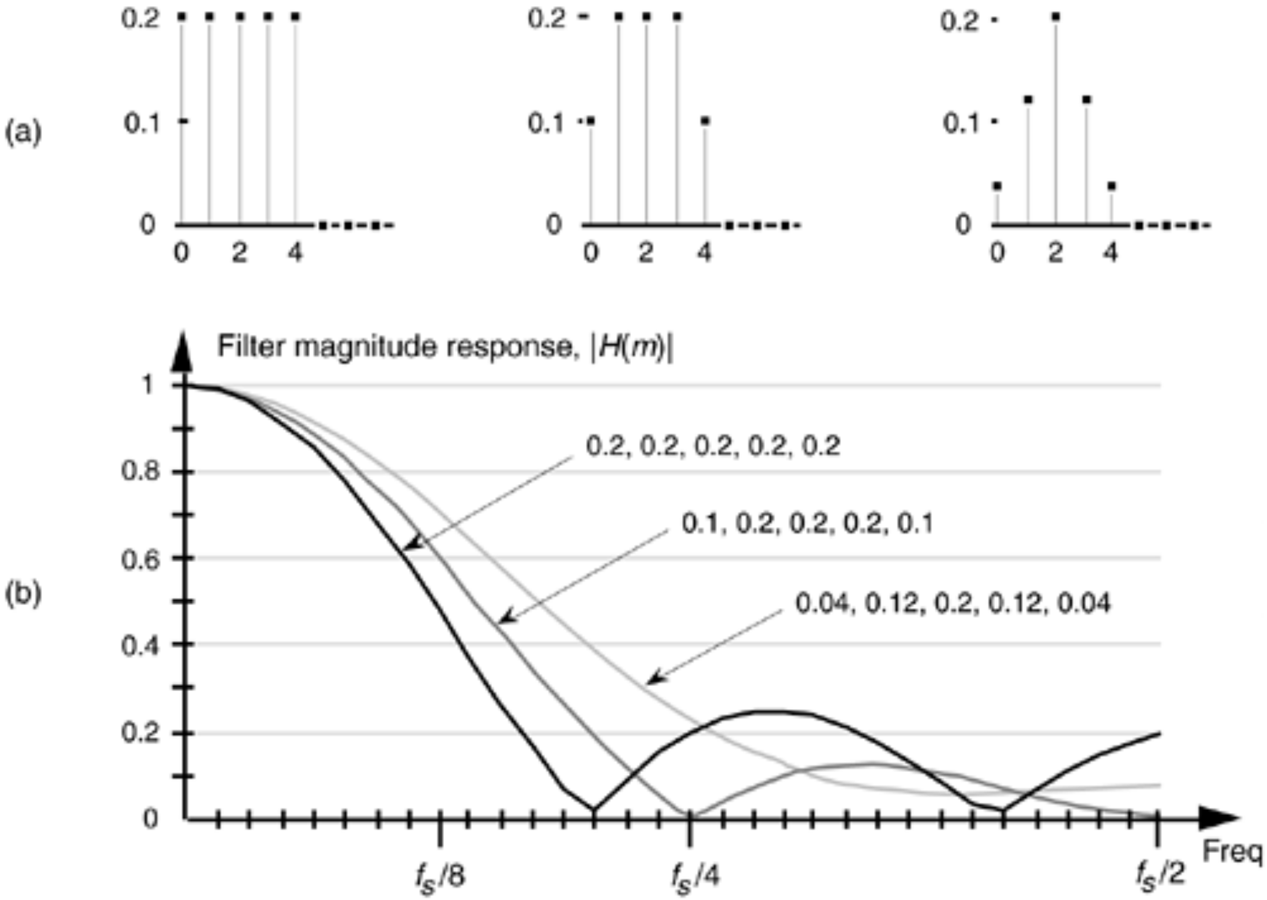


$f_{signal}=3f_s/32$

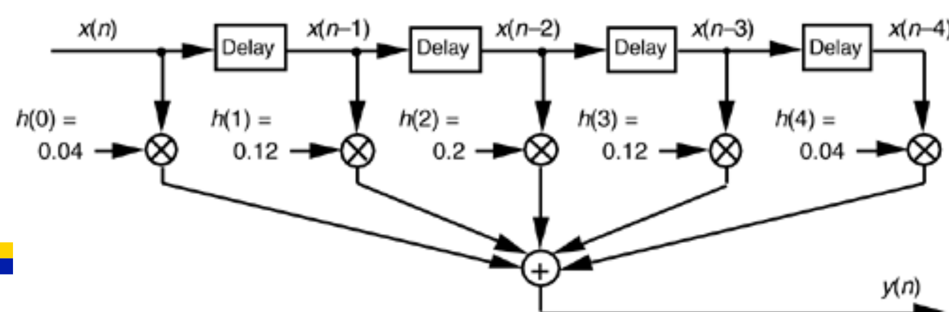
Frequency-Domain Response



More FIR Filters

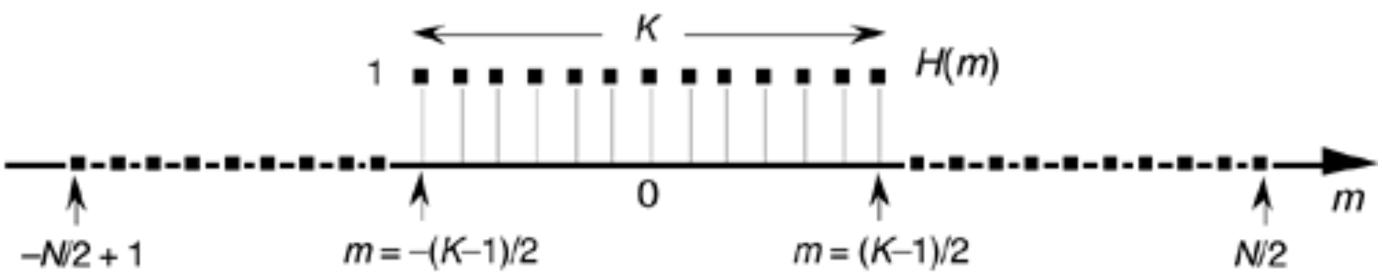


5-tap "Gaussian" FIR filter

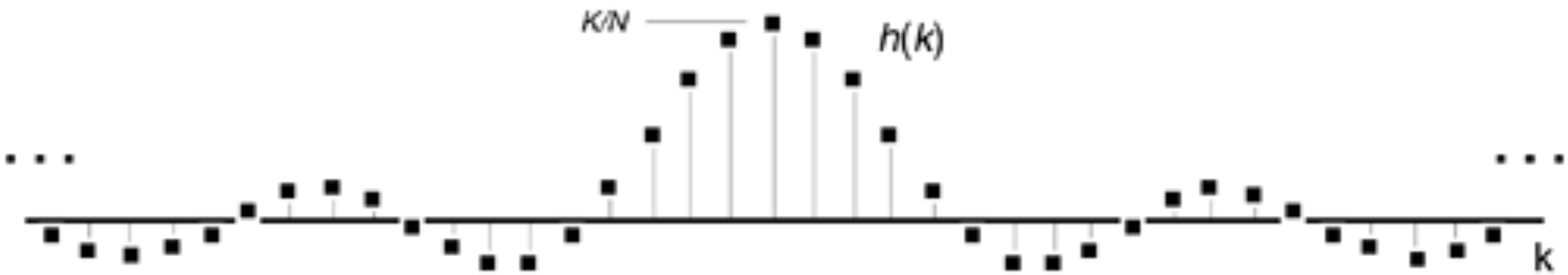


Ideal Low-Pass FIR Filter

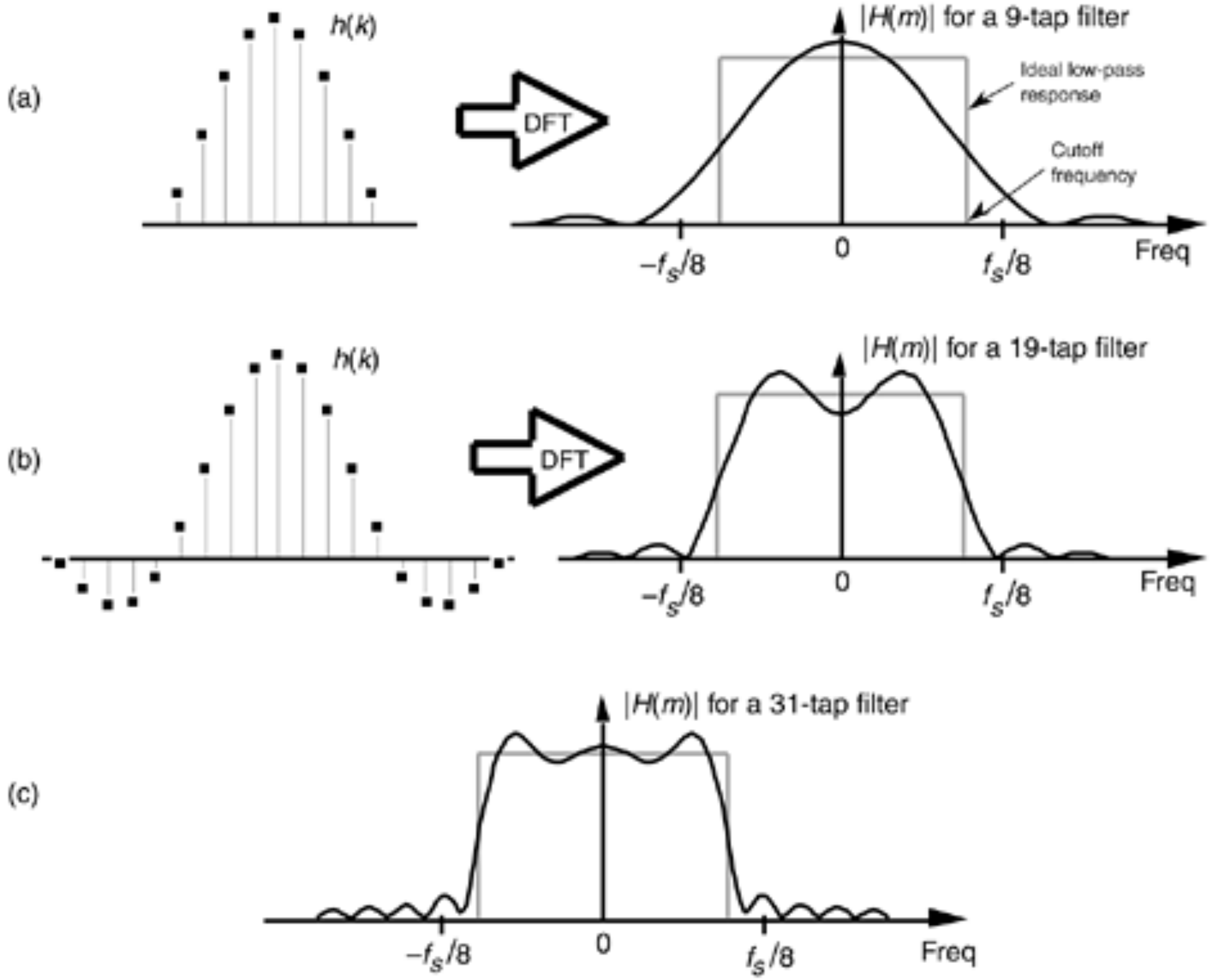
Frequency response



Time-domain coefficients

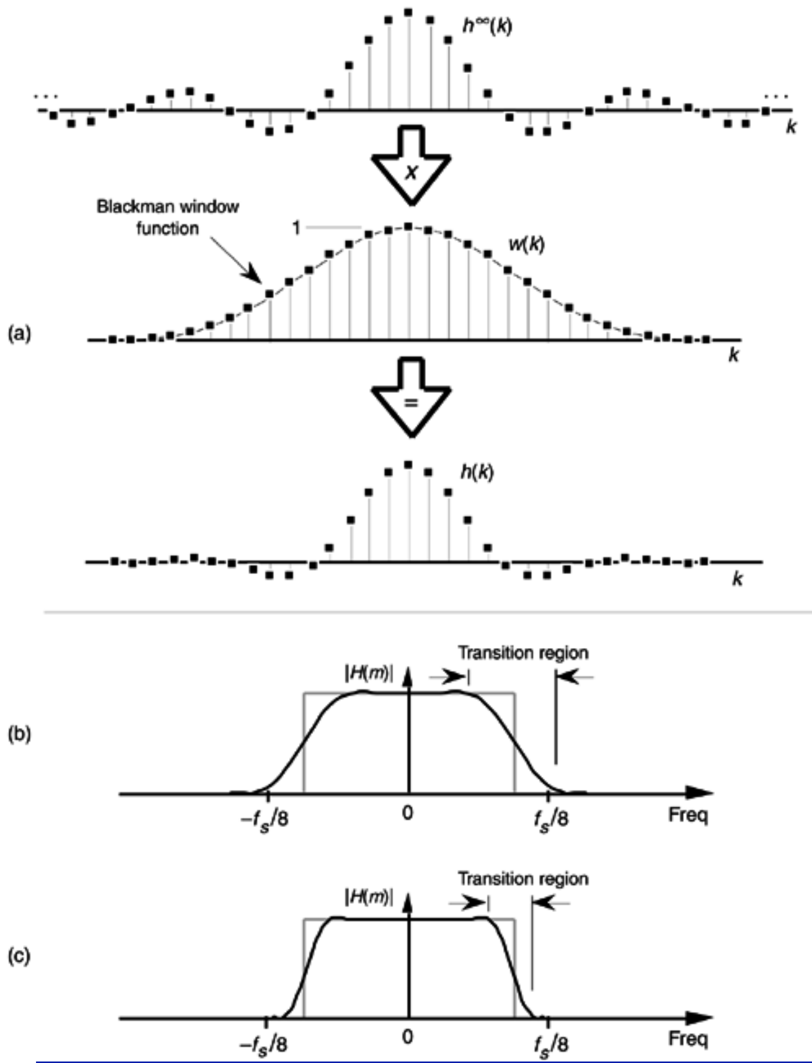


Convoluted Low-Pass Filter

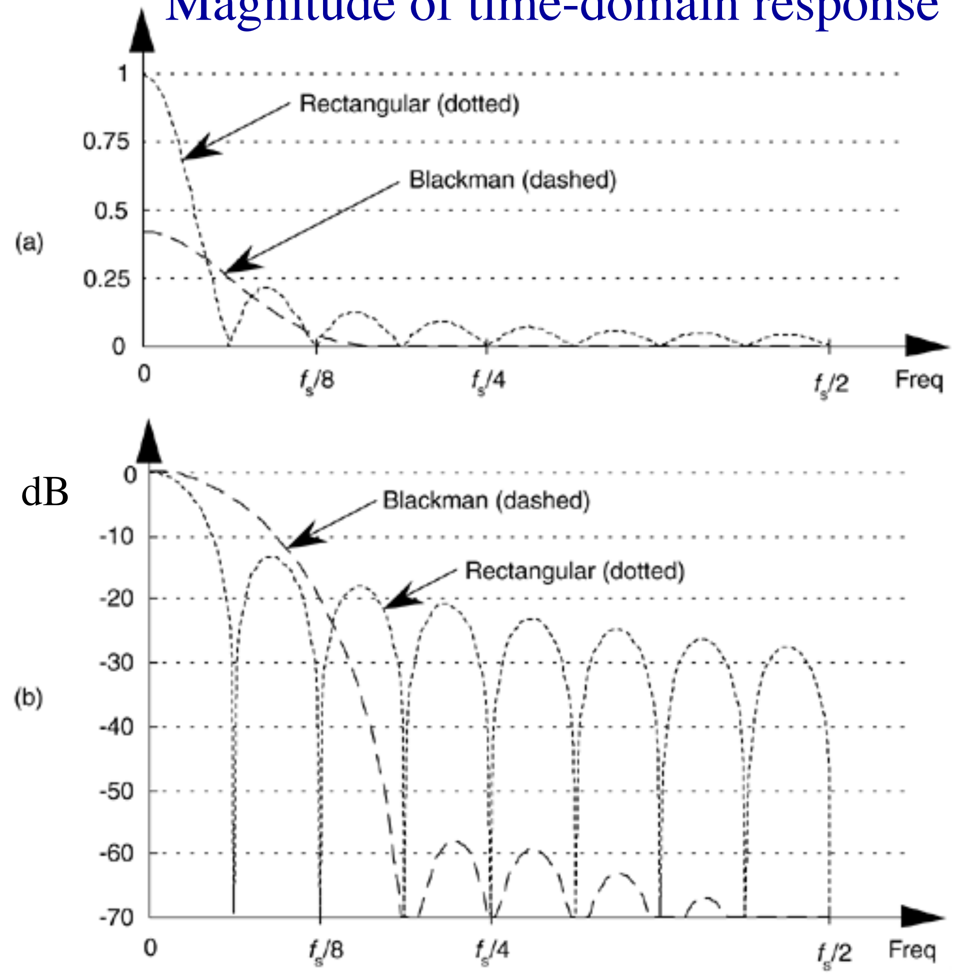


Example: Blackman Window

$$\omega(k) = 0.42 - 0.5\cos\left(\frac{2\pi k}{N}\right) + 0.08\cos\left(\frac{4\pi k}{N}\right), \text{ for } k = 0,1,2,\ldots,N-1 \ .$$ (approximates Gauss)



Magnitude of time-domain response



More (Tunable) Filters

Chebyshev window:→
(also called the Dolph–Chebyshev and the Tchebyschev window)

$w(k)$ = the N-point inverse DFT of

$$\frac{\cos\left[N\cdot\cos^{-1}\left[\alpha\cdot\cos\left(\pi\frac{m}{N}\right)\right]\right]}{\cosh[N\cdot\cosh^{-1}(\alpha)]},$$

where $\alpha = \cosh\left(\frac{1}{N}\cosh^{-1}(10^{\gamma})\right)$ and $m = 0, 1, 2, \ldots, N$

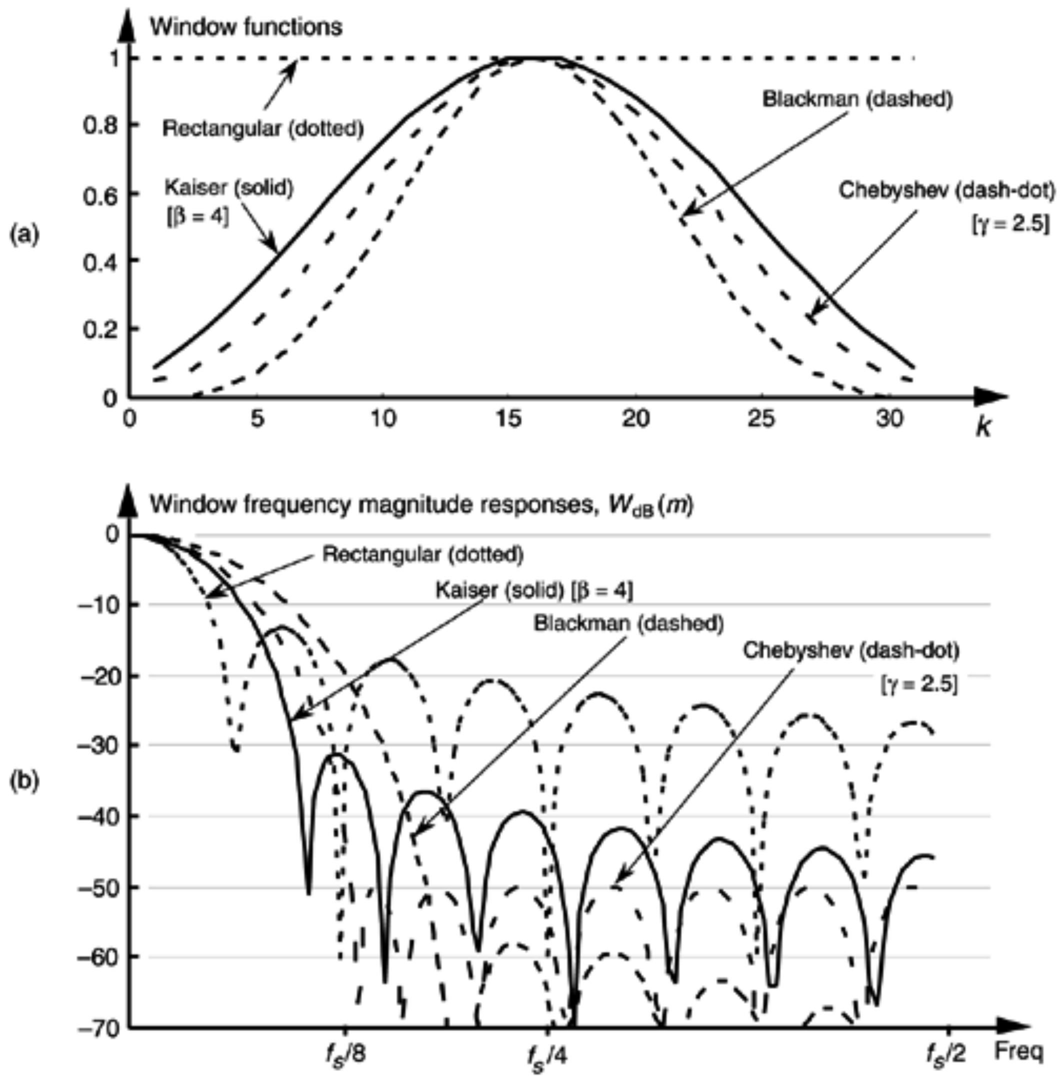
(c.f. hardware filters)

Kaiser window:→
(also called the Kaiser–Bessel window)

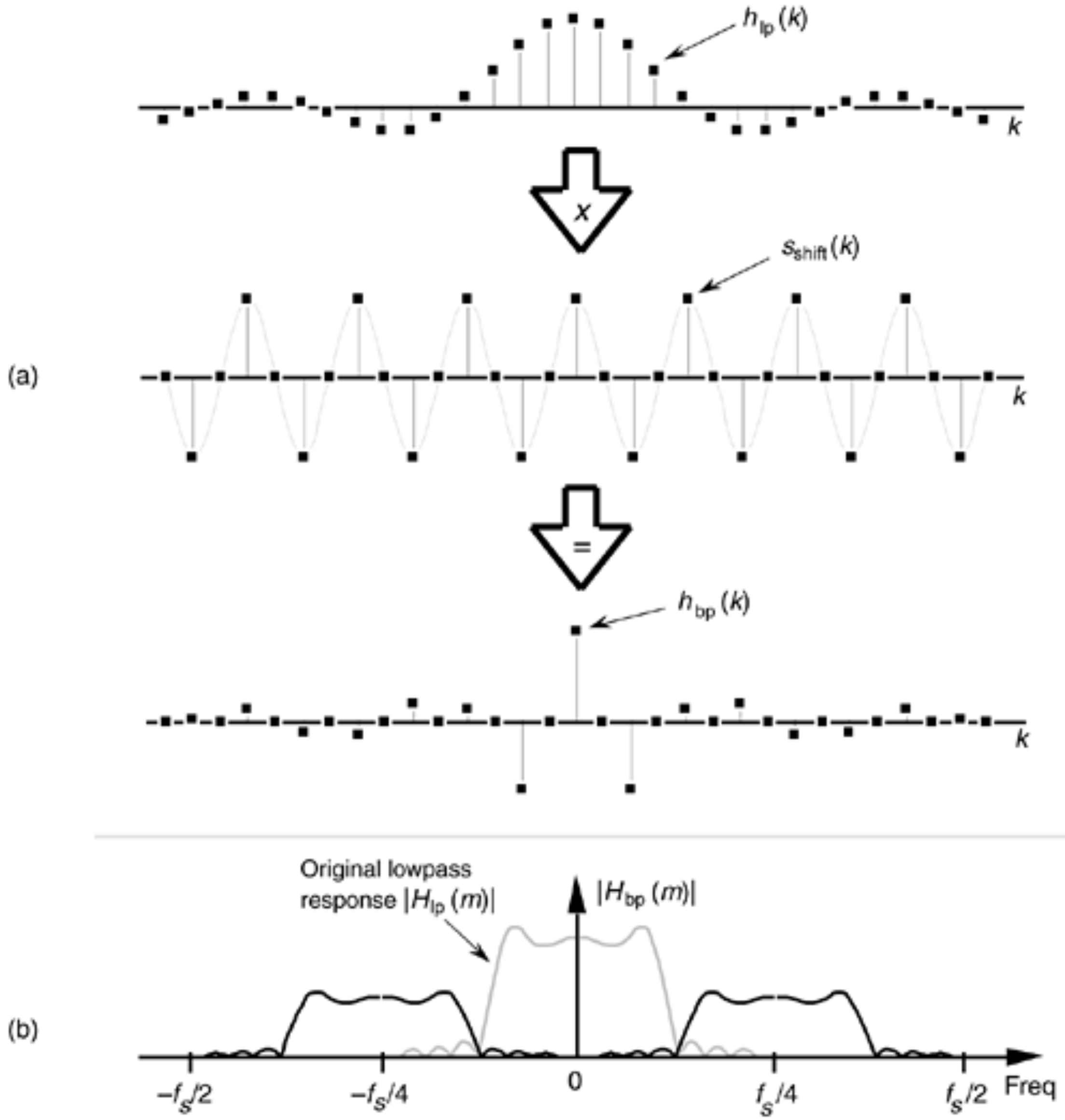
$$\omega(k) = \frac{I_o\left[\beta\sqrt{1-\left(\frac{k-p}{p}\right)^2}\right]}{I_o(\beta)},$$

for $k = 0, 1, 2, \ldots, N-1$, and $p = (N-1)/2$.

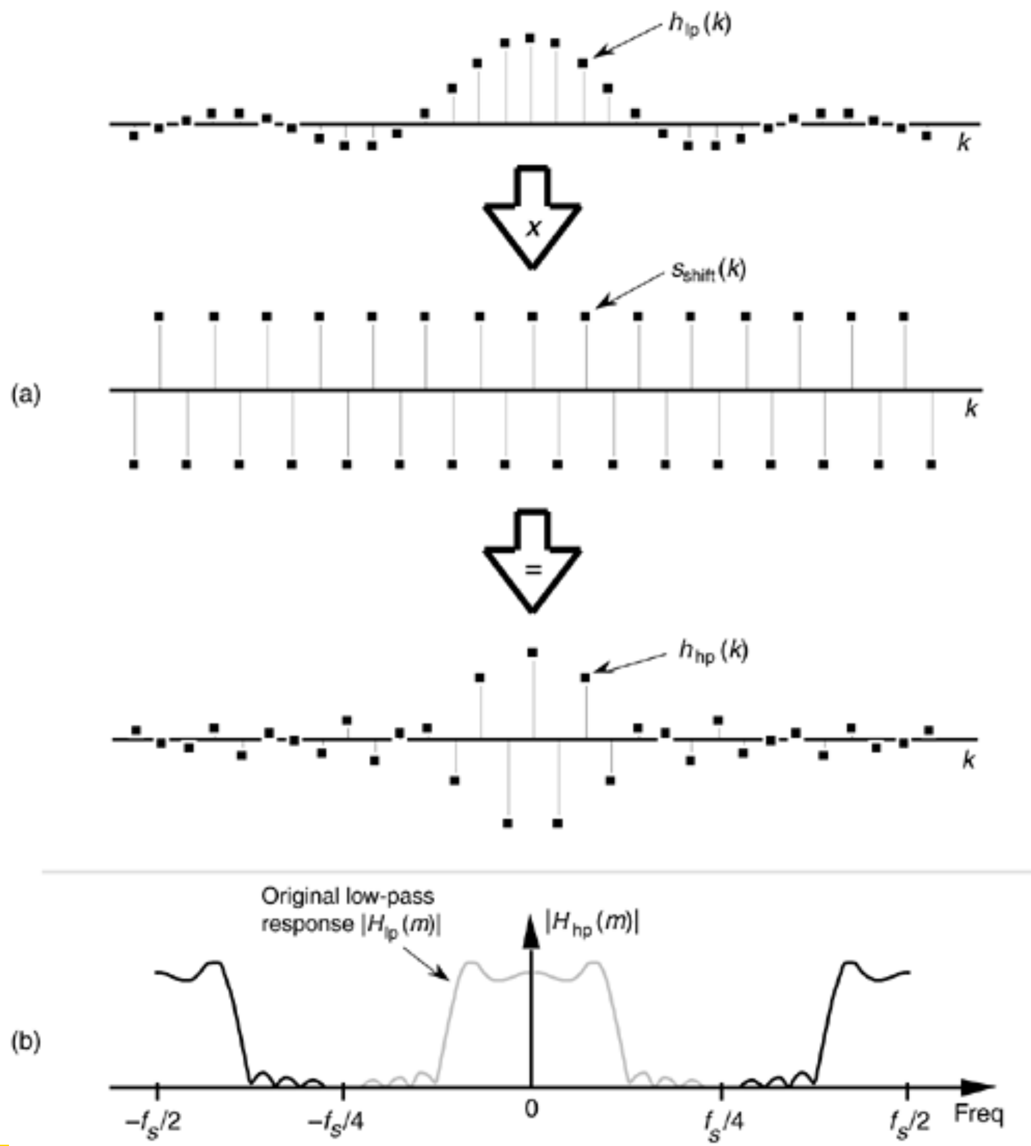
More Tunable Filters



Bandpass Filter

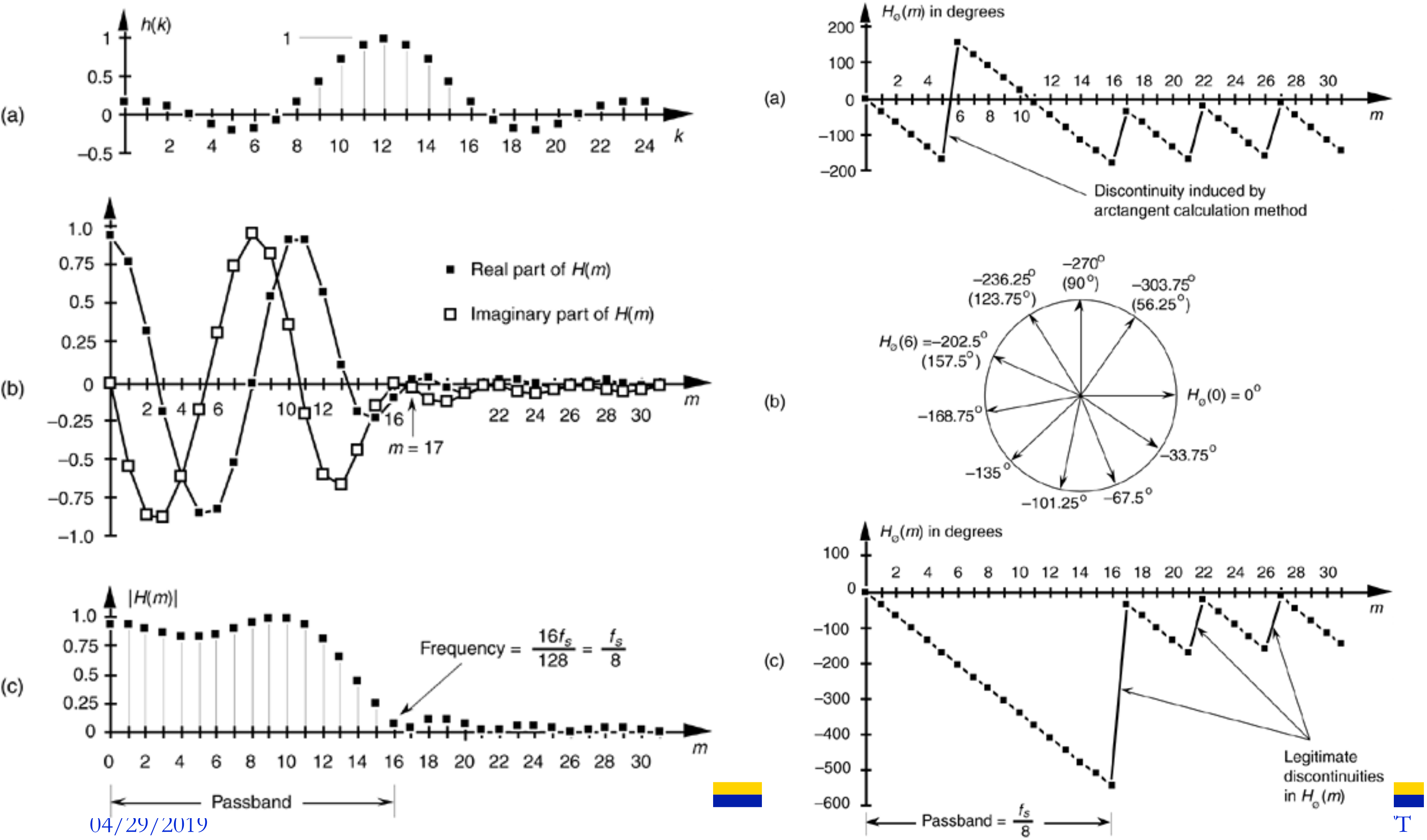


Highpass Filter

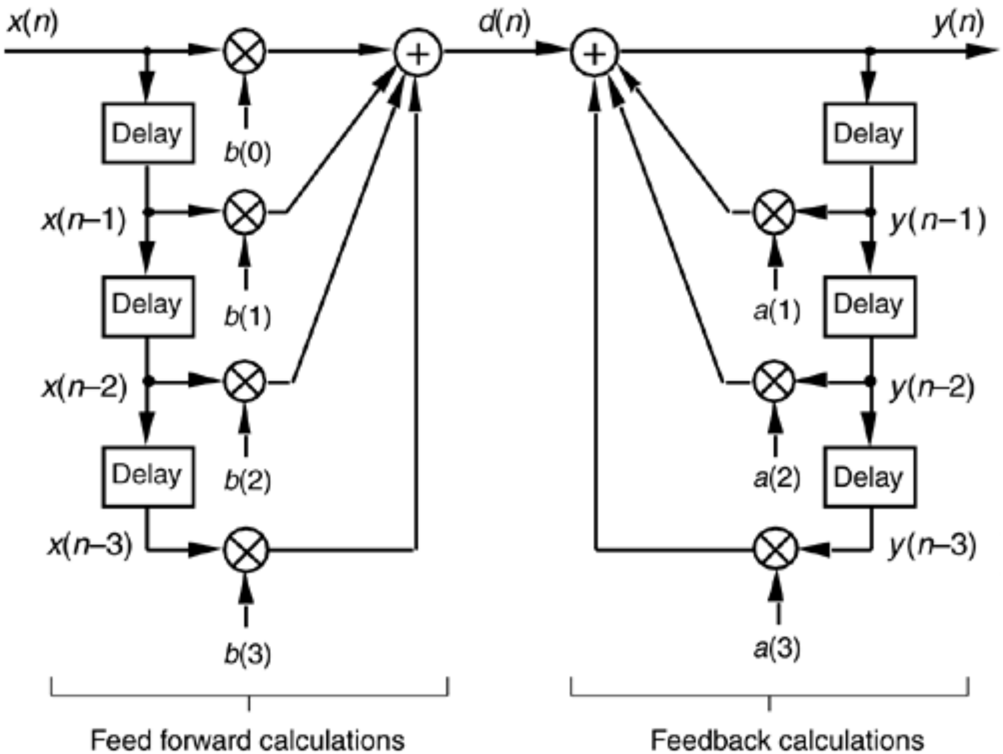


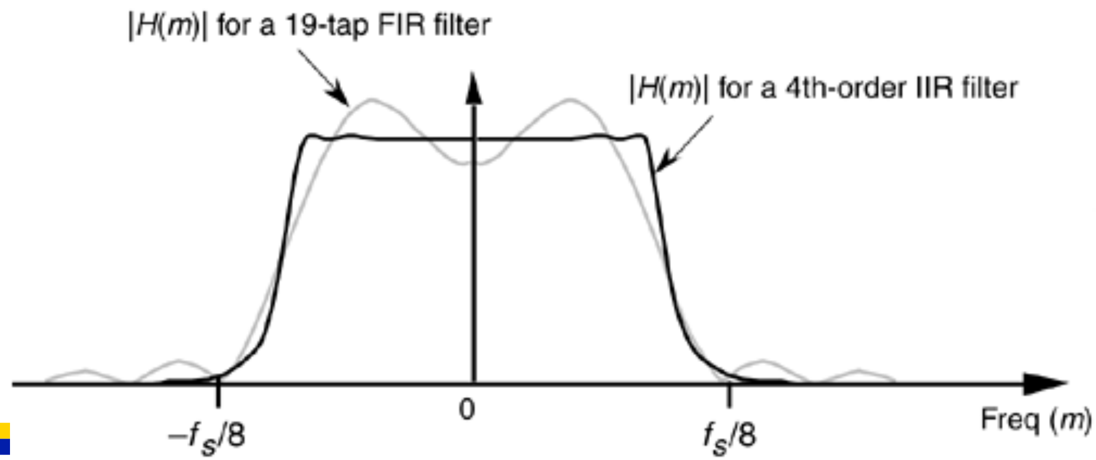
Phase Response in FIR Filters

Linear phase shift in passband: constant group delay (no spectral distortions): good !



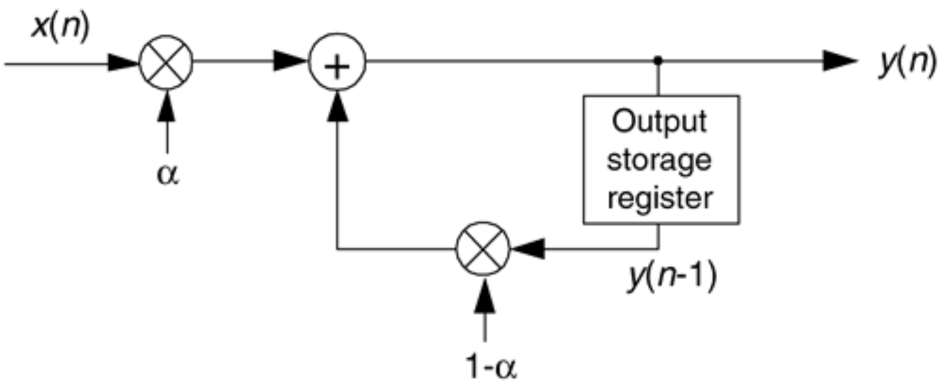
IIR Filters

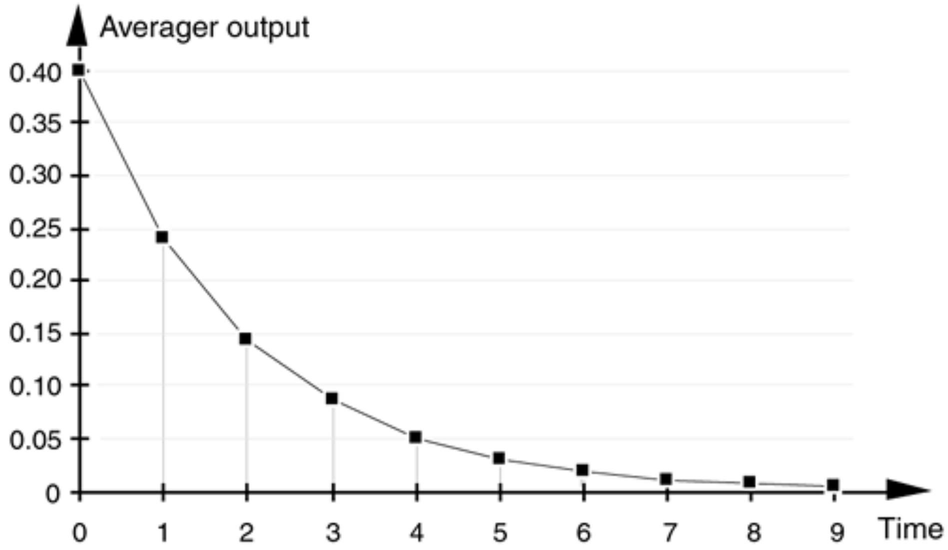

x(n)
d(n)
y(n)
Delay
Delay
Delay
Delay
Delay
Delay
b(0)
b(1)
b(2)
b(3)
x(n–1)
x(n–2)
x(n–3)
y(n–1)
y(n–2)
y(n–3)
a(1)
a(2)
a(3)
Feed forward calculations
Feedback calculations


|H(m)| for a 19-tap FIR filter
|H(m)| for a 4th-order IIR filter
–fs/8
0
fs/8
Freq (m)

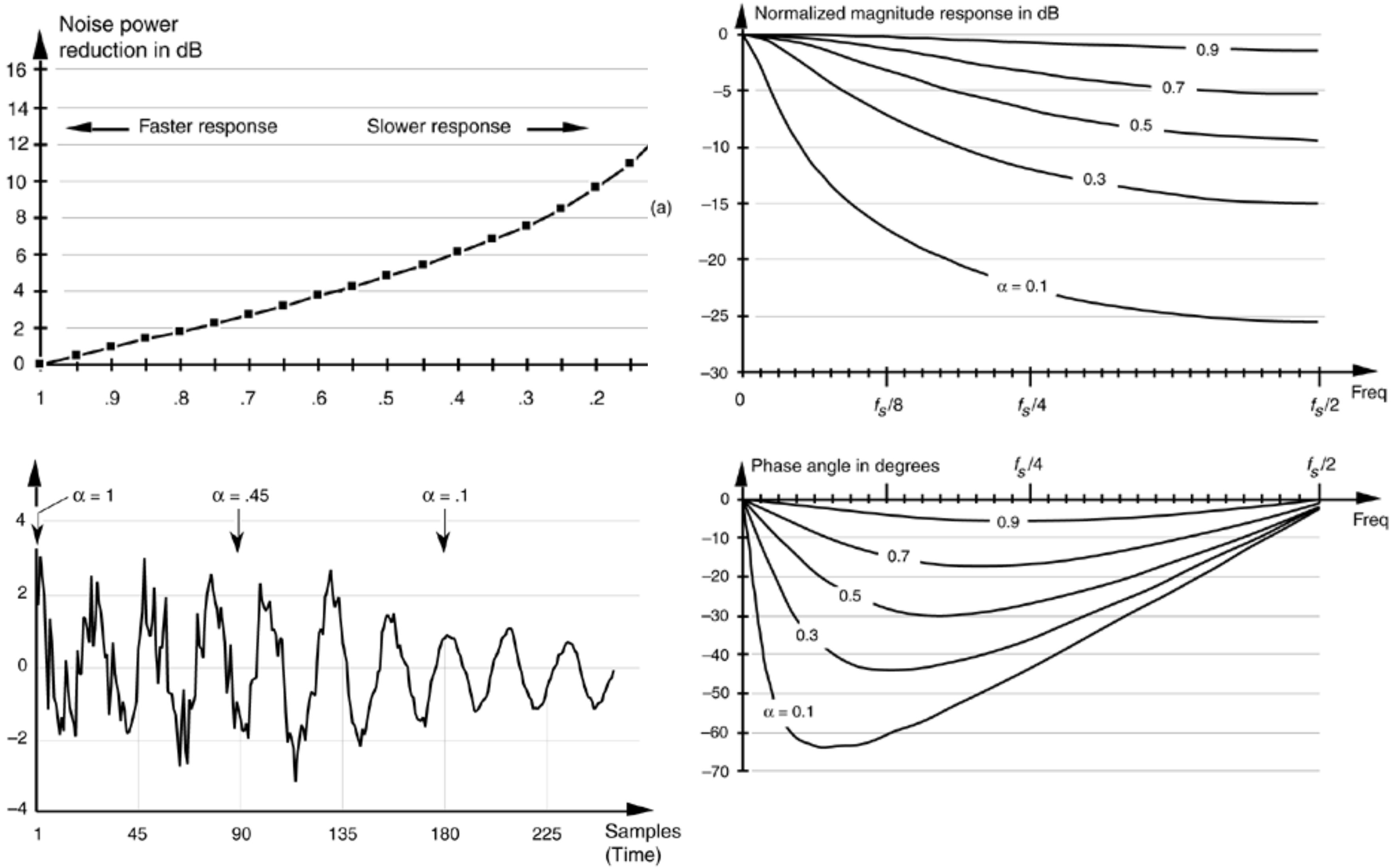
Example: Exponential Averaging Filter

$y(n)=\alpha x(n)+(1-\alpha)x(n-1)$





Exponential IIR Filter



What We Have Not Covered

- Many topics to cover, so I so far focused on most immediately relevant
  - ☞ There are courses dedicated to DSP
- Other possible topics of interest (at your leisure):
  - ❑ Digital Signal Processing
    - ☞ Digital mixing
    - ☞ Modulation/demodulation
  - ❑ Smoothing, windowing
    - ☞ Often useful for image processing
  - ❑ Down-sampling (decimating), re-sampling
  - ❑ "Optimal" filters