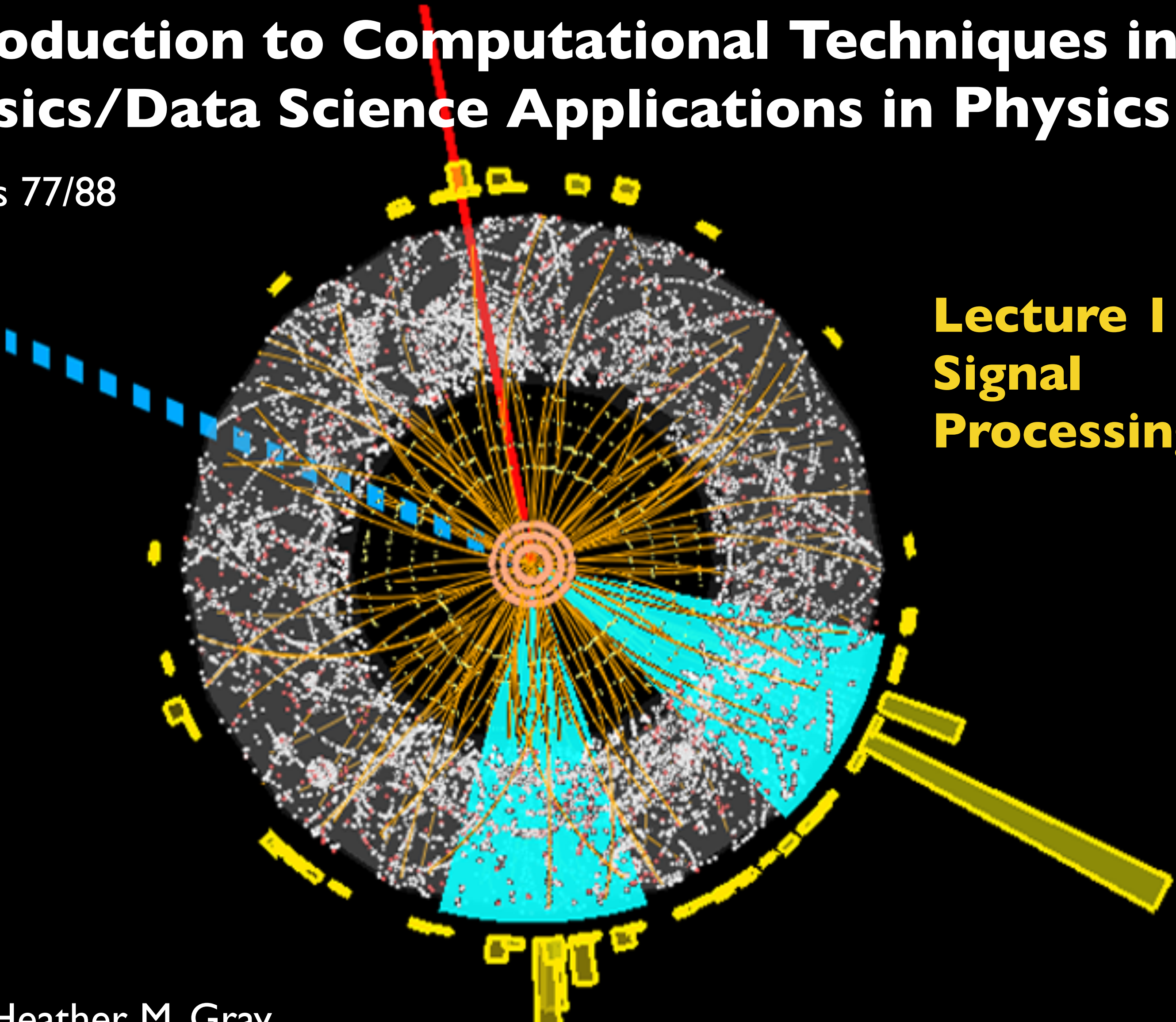
Introduction to Computational Techniques in Physics/Data Science Applications in Physics

Physics 77/88



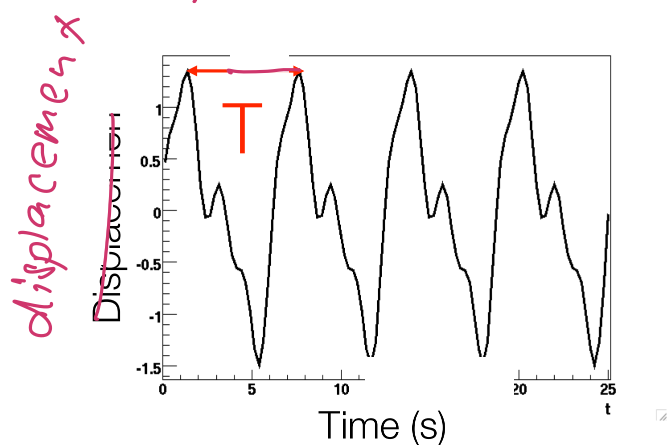
Lecture 11: Signal Processing

Prof. Heather M. Gray

Many thanks to Yury Kolomensky for material used in this course

Definitions

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



Acronyms

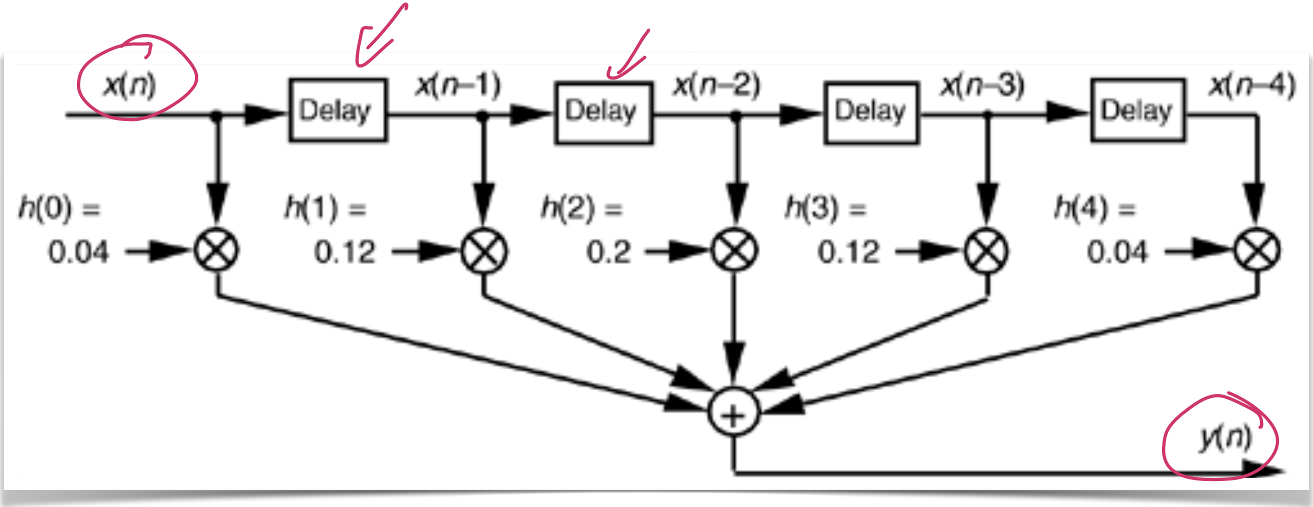
- FFT = Fast Fourier Transform
  - DFT = Discrete Fourier Transform
- DSP = Digital Signal Processing
- SNR = Signal-to-noise ratio
  - Often expressed in dB,
    - i.e. SNR = $20\log_{10}(S/N)$ where S and N are signal and noise amplitudes
    - For power, SNR = $10\log_{10}(P_S/P_N)$
- ASD = Amplitude spectral density
  - PSD = Power spectral density
  - NPS = Noise power spectrum

Acronyms

- FIR = Finite Impulse Response
- IIR = Infinite Impulse Response

filter

Schematic Notation



- Represent addition, multiplication (mixing), delay operations
- Most linear systems can be represented this way
  - i.e. $c_1 x_1(n) + c_2 x_2(n) \Rightarrow c_1 y(n) + c_2 y(n)$

Fourier Transform

- Most common way to analyze a periodic function is by means of a Fourier transform

- Represent as a superposition of harmonic oscillations

- $x(t) = c\int_{-\infty}^{\infty} \underline{x(\omega)}\, e^{j\omega t}\, d\omega$

    - Here $x(\omega)$ is a Fourier coefficient : represents the strength of the periodic signal at a particular frequency
        - Also known as the spectral density

Fourier Transform

- Inverse transform
  - $x(\omega) = C\int_{-\infty}^{\infty} x(t)\, e^{-j\omega t}\, dt$
  - $x(f) = C\int_{-\infty}^{\infty} x(t)\, e^{-j2\pi f t}\, dt$
  - $= C\int_{-\infty}^{\infty} x(t)\left[\cos(2\pi f t) - j\sin(2\pi f t)\right] dt$

Discrete Fourier Transform

- For a sampled waveform, replace the integral with a discrete sum
  - $x(m) = \sum_{n=0}^{N-1} x(n) \left[\cos \frac{2\pi m n}{N} - j \sin \frac{2\pi n m}{N} \right]$
- Define
  - $x_{mag}(m) = |x(m)| = \sqrt{x_{re}(m)^2 + x_{im}(m)^2}$
  - $\Delta\phi(m) =$ phase $= \tan^{-1}\left[x_{im}(m) / x_{re}(m) \right]$
  - $P(m) = |x(m)|^2 = x^*(m)\, x(m)$

    $= x_{re}(m)^2 + x_{im}(m)^2$

Aliasing

- For real signals $x(m)$, 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 $0 \leq m < N/2$ is redundant with $\frac{N}{2} < m < N$, $N < m < \frac{3N}{2}$
  - Nyquist theorem:
    - $x(t)$ can be reconstructed perfectly from $x(m)$
      - iff $x(t)$ is limited to the band $\pm f_B$ and $f_s > 2 f_B$
    - $f_N = f_s/2$ 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 rel., which is used in the Fast Fourier Transform (FFT)
    - Scale as $N \log N$
    - Multiple FFT algorithms exist, and are interfaced in Python
    - E.g. FFTW from MIT

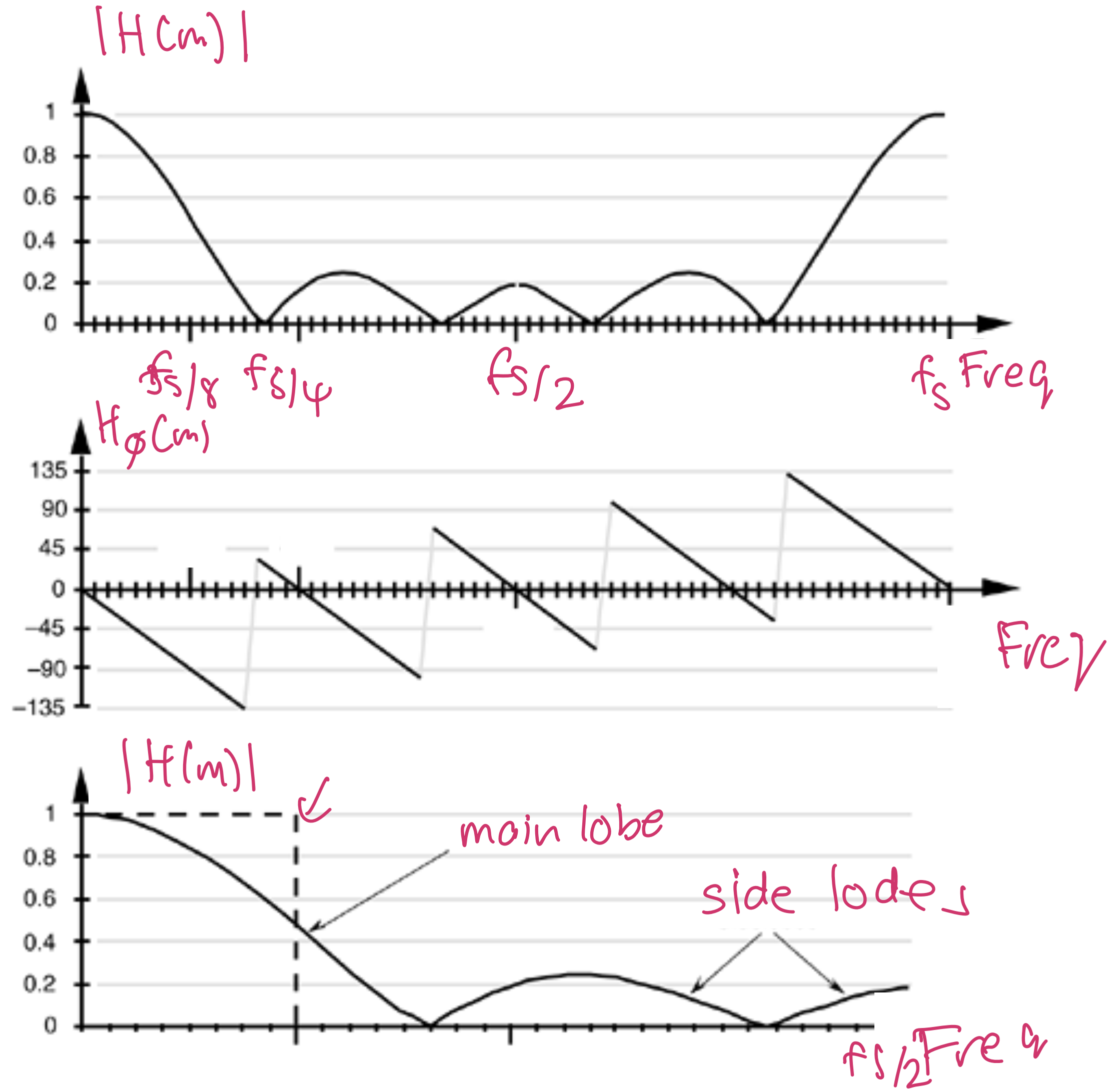
Applications of FFT

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

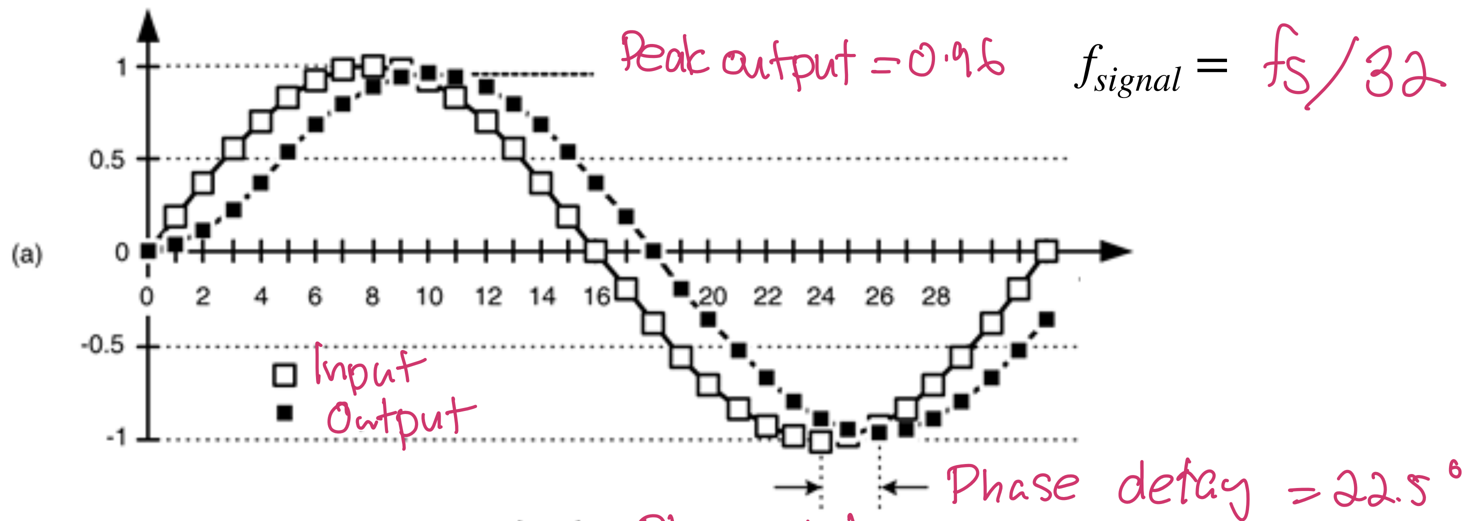
FIR filters

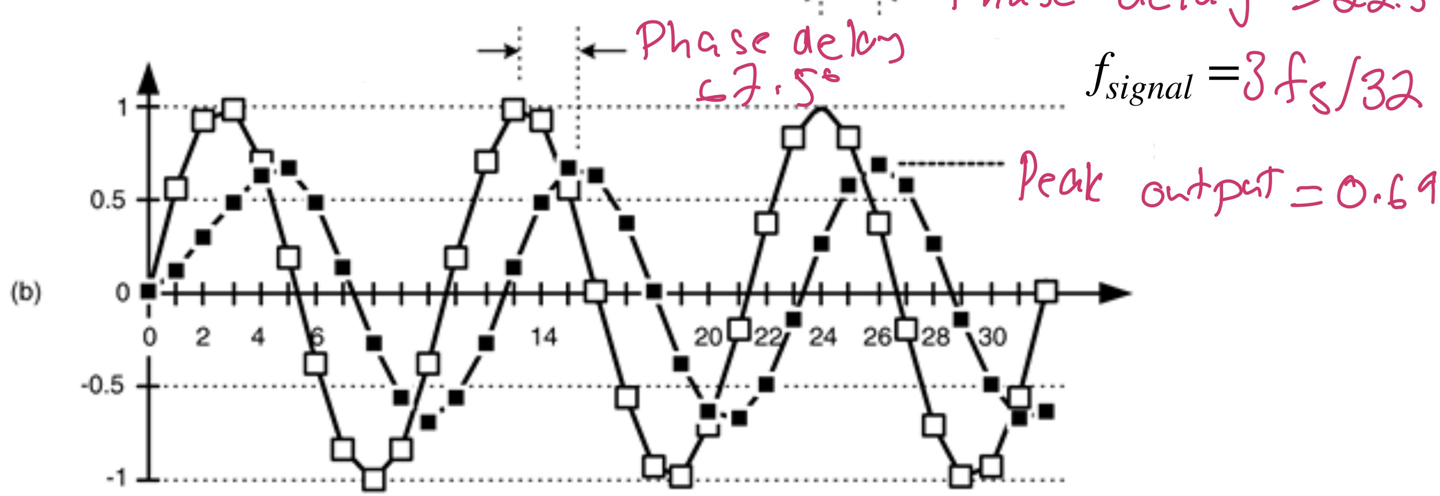
- Filtering (in the time domain) is a way to reduce the contributions of (random) noise to the characteristics of the signal
- Usually implemented by summing over samples with some (predetermined) weight
    - Equivalent to integration or averaging

Box-car Filter Response

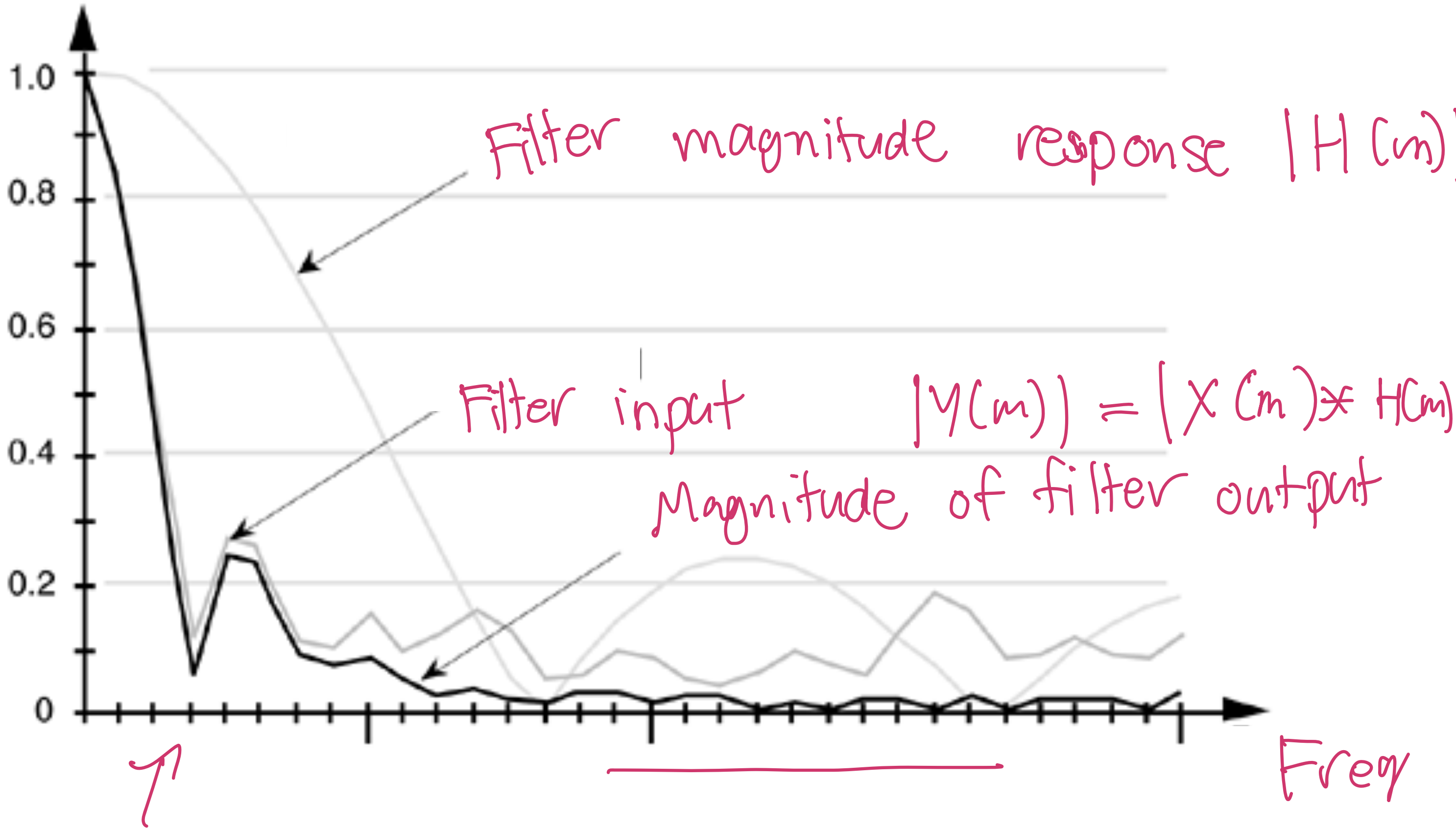


Time-Domain Signal After Filtering



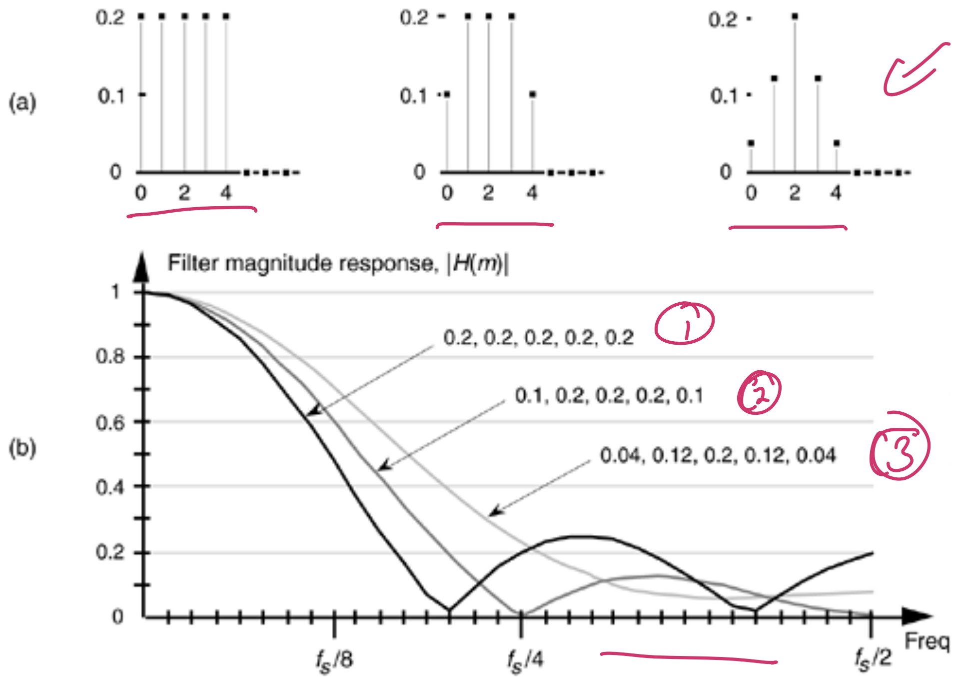


Frequency-Domain Response

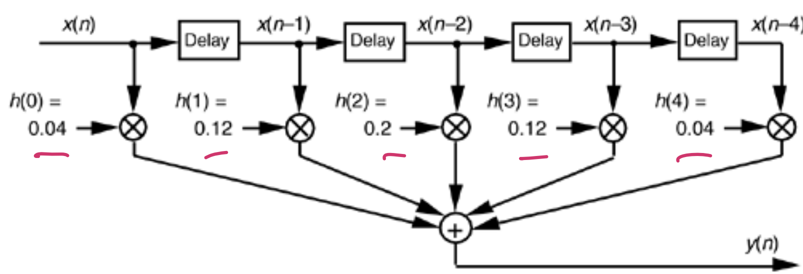


More FIR Filters

taps

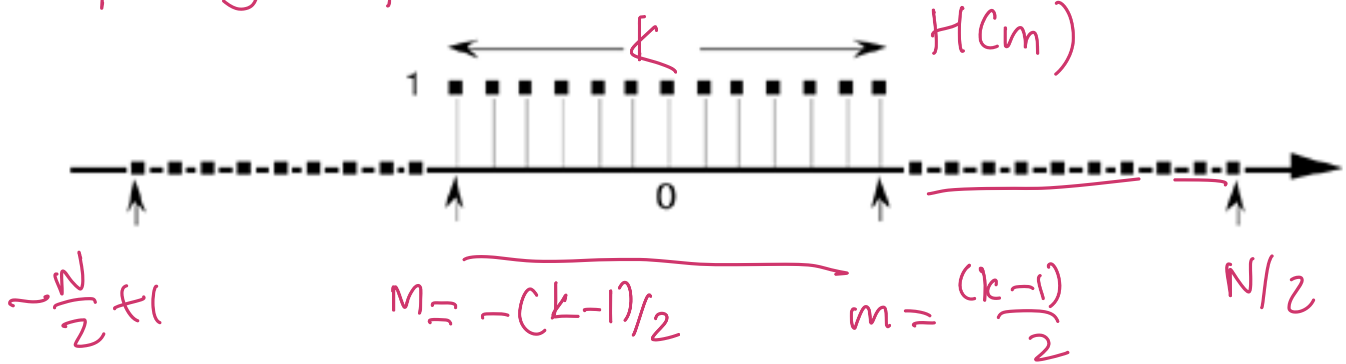


5-tap "Gaussian" FIR filter



Ideal Low-Pass FIR Filter

Frequency response



Time domain coefficients

Convoluted Low-Pass Filter

9-tap

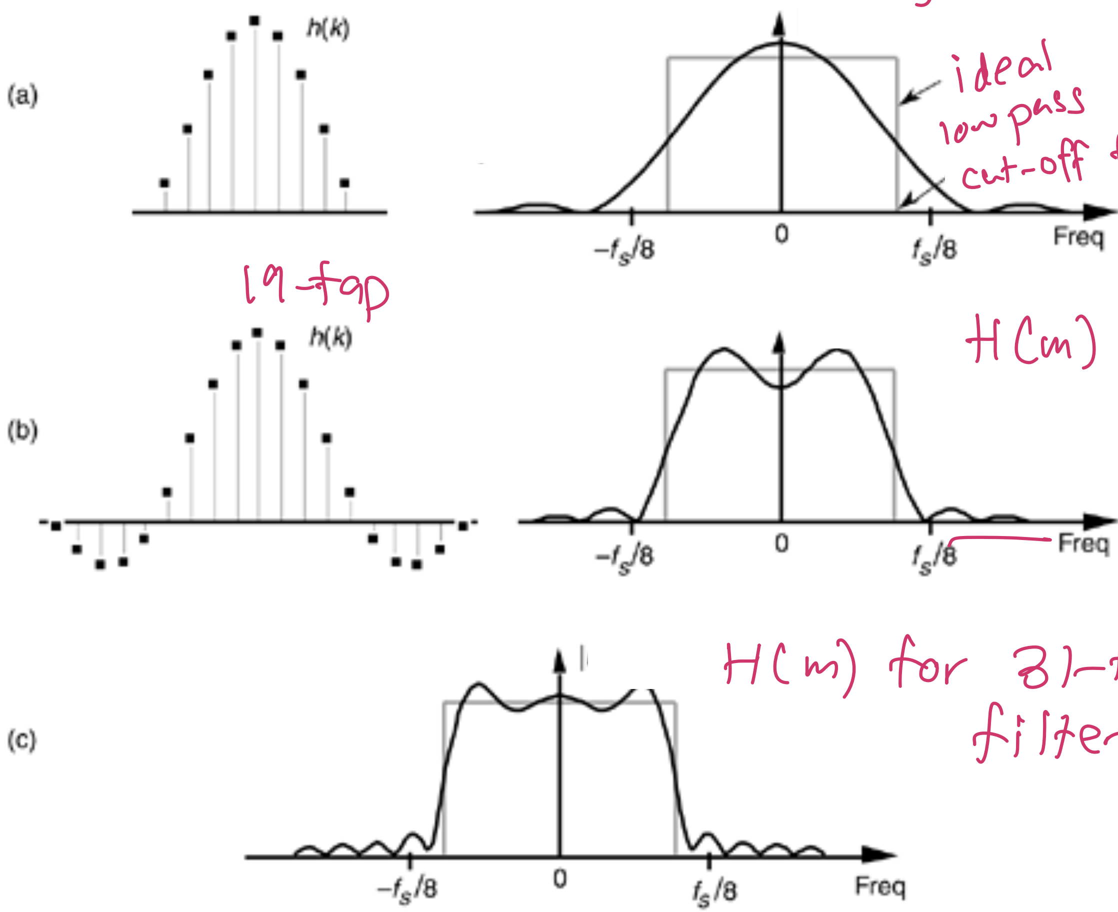
H(m) for 9-tap filter

ideal low pass cut-off freq.

19-tap

H(m)

H(m) for 31-tap filter

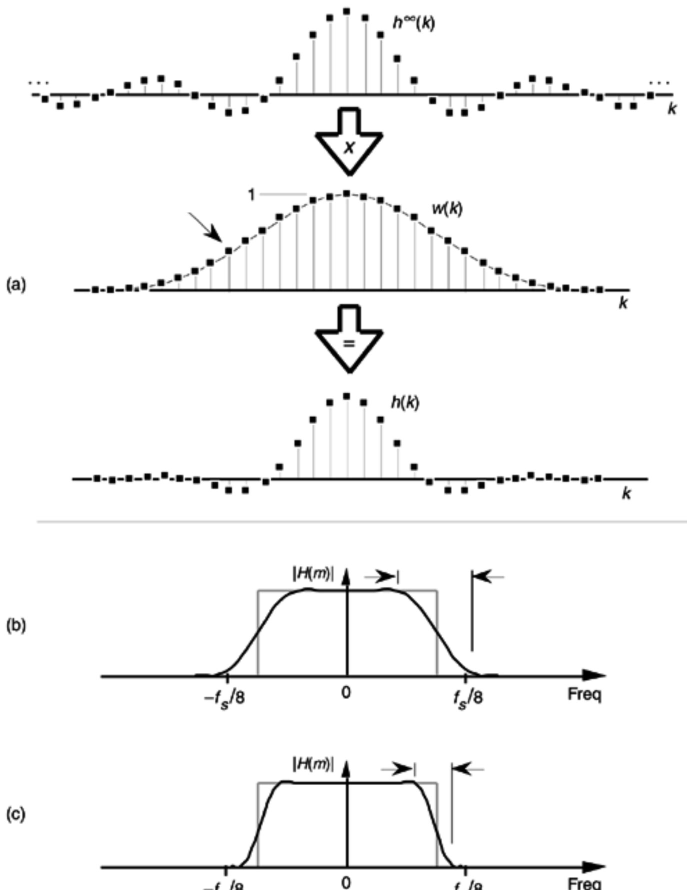


Example: Blackman Window

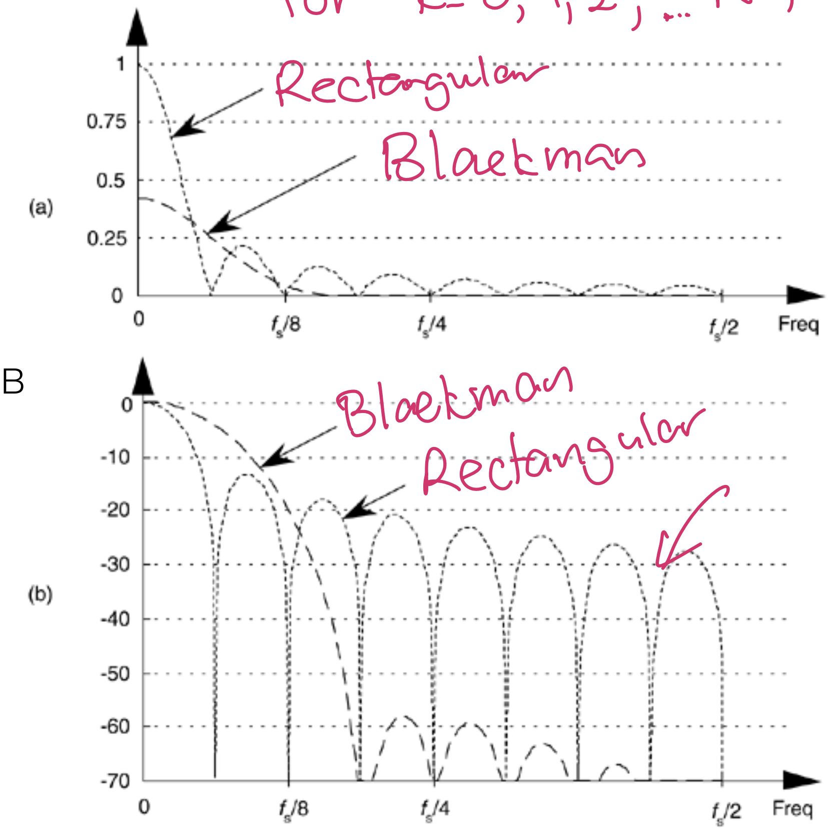
Gaussian filter

$$w(k) = 0.42 - 0.5\cos\left(\frac{2\pi k}{N}\right) + 0.08\cos\left(\frac{4\pi k}{N}\right)$$

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



dB



More (Tunable) Filters

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

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

Chebyshev window

- where $\alpha = \cosh\left[\frac{1}{N}\cosh^{-1}\left(10^{\gamma}\right)\right]$

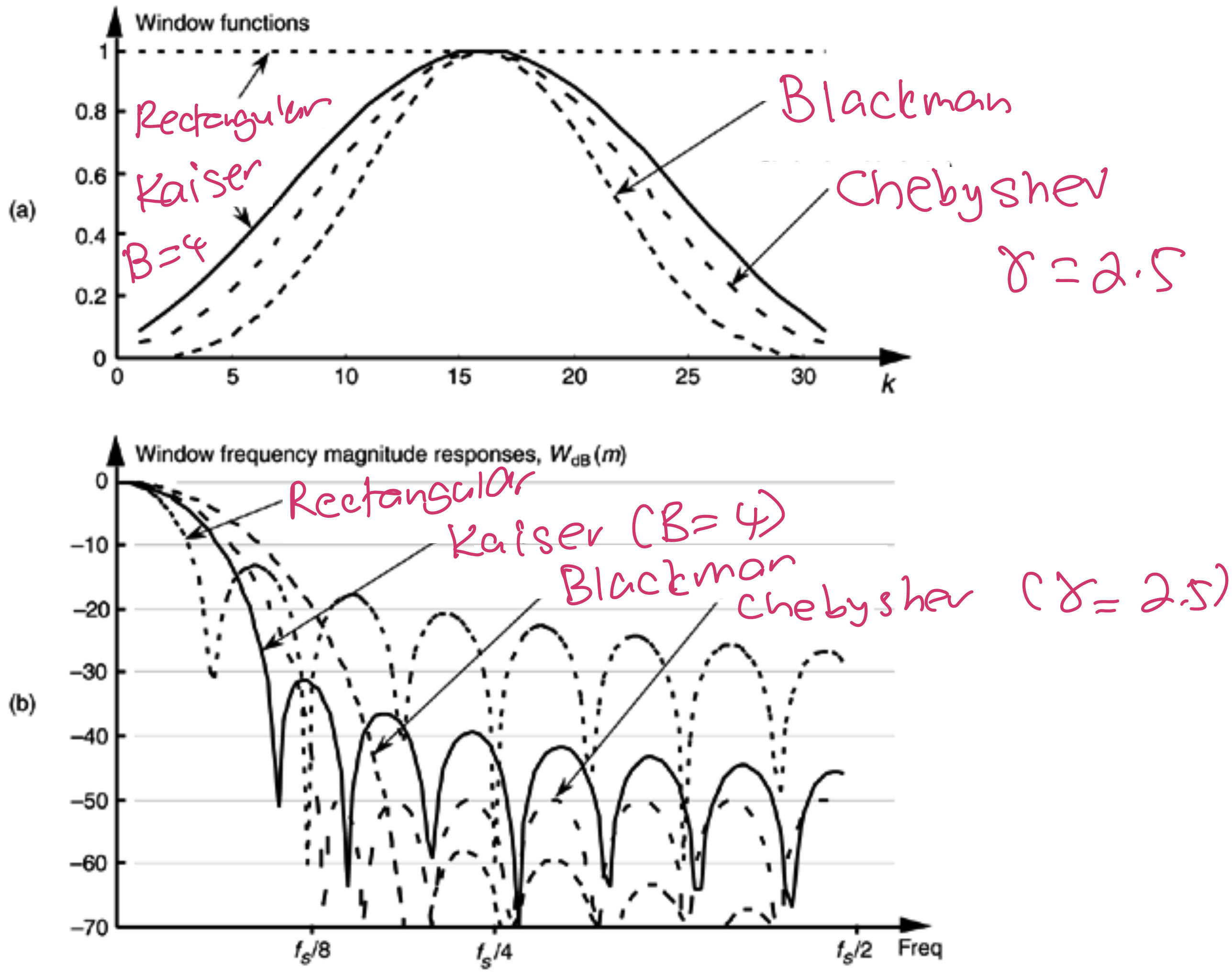
$m = 0, 1, 2 \ldots N$

- $\omega(k) = \dfrac{I_0\left[\beta\sqrt{1-\left(\frac{k-p}{p}\right)^2}\right]}{I_0(\beta)}$

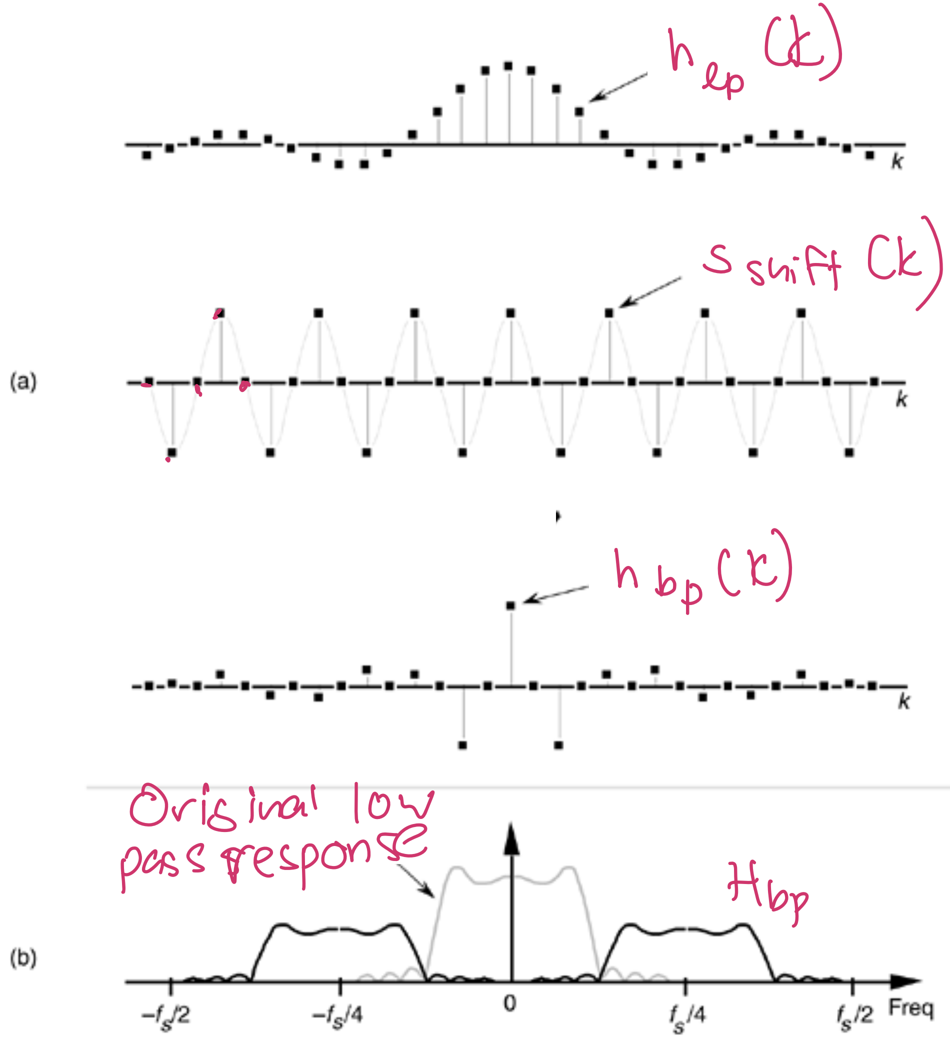
Kaiser window

$k = 0, 1, 2 \ldots N-1$ $\quad p = (N-1)/2$

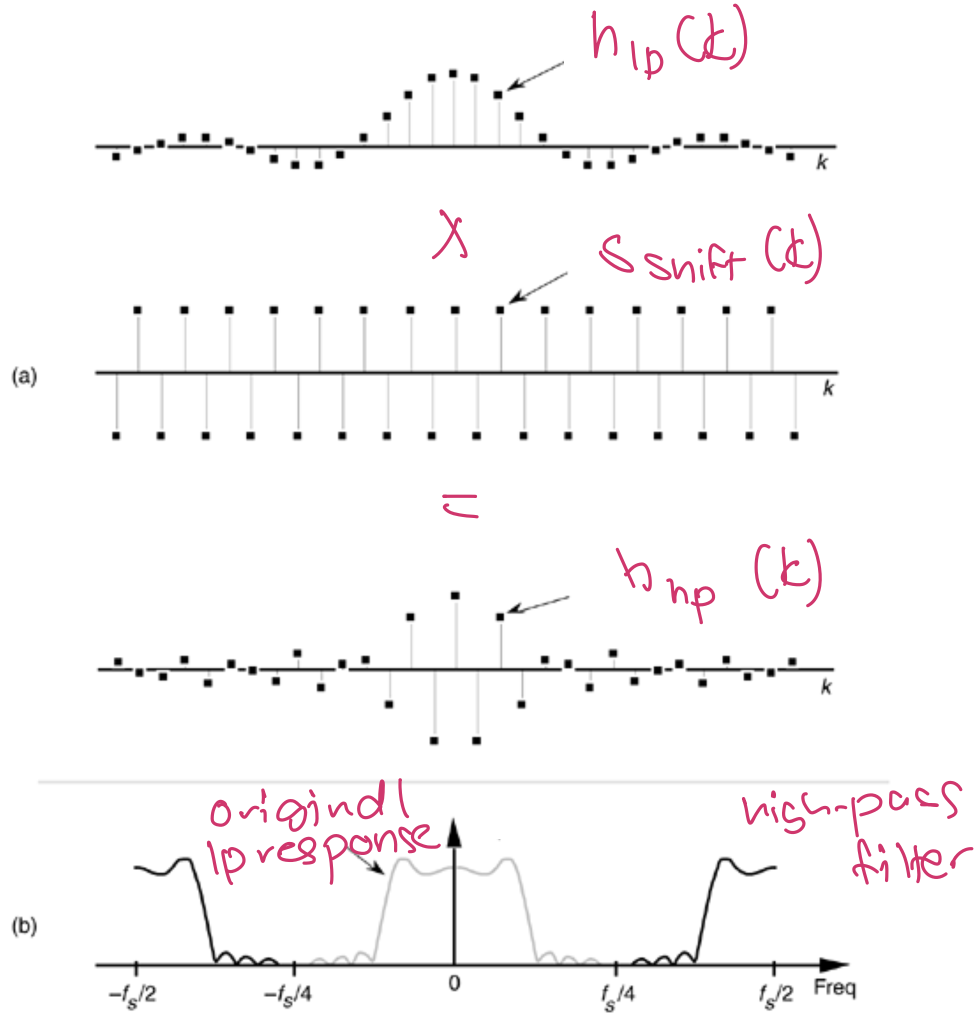
More Tunable Filters



Bandpass Filter



Highpass Filter

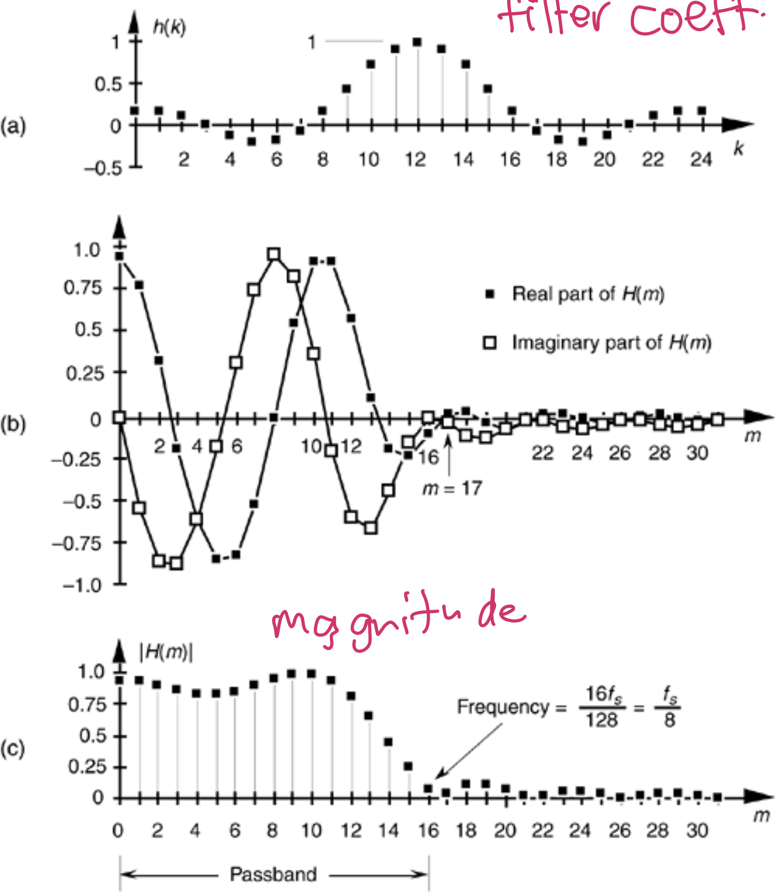


Phase Response in FIR Filters

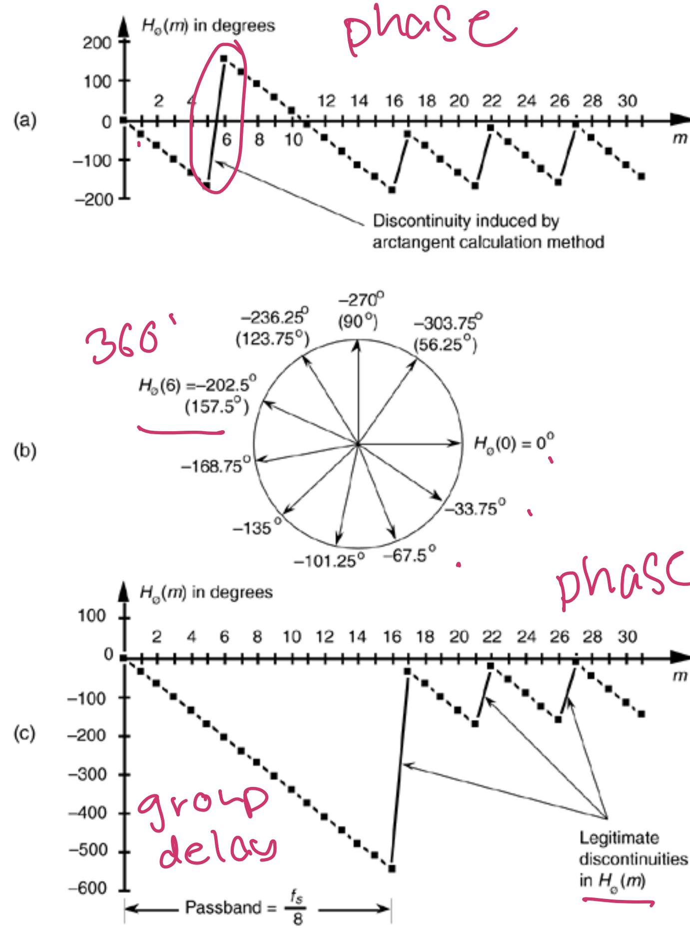
Linear phase shift in passband:
constant group delay (no spectral distortions):

filter coeff.

phase



magnitude

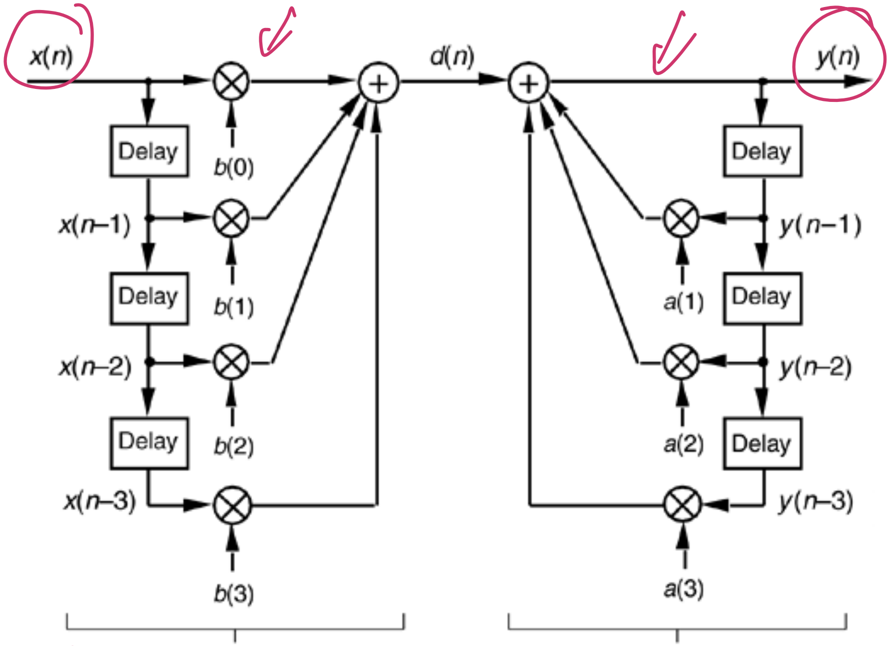


360'

phase

group delay

IIR Filters

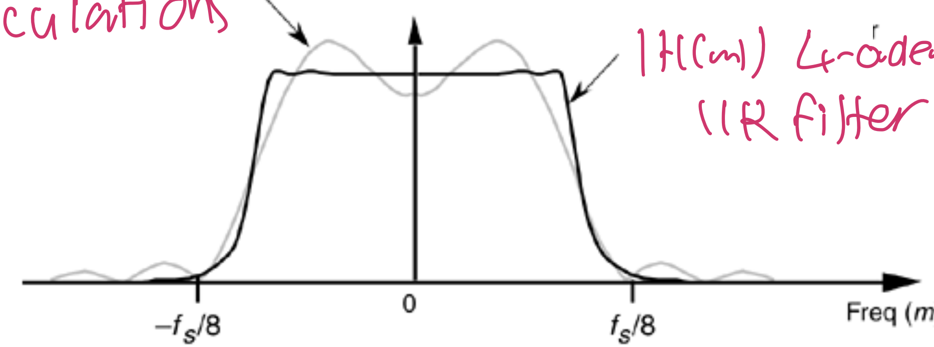


Feed forward calculations

Feedback calculations

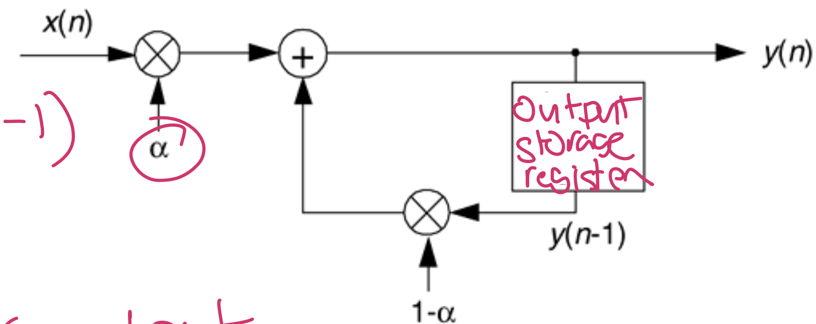
|H(m)| for 19-tap FIR filter

|H(m)| 4-order IIR filter

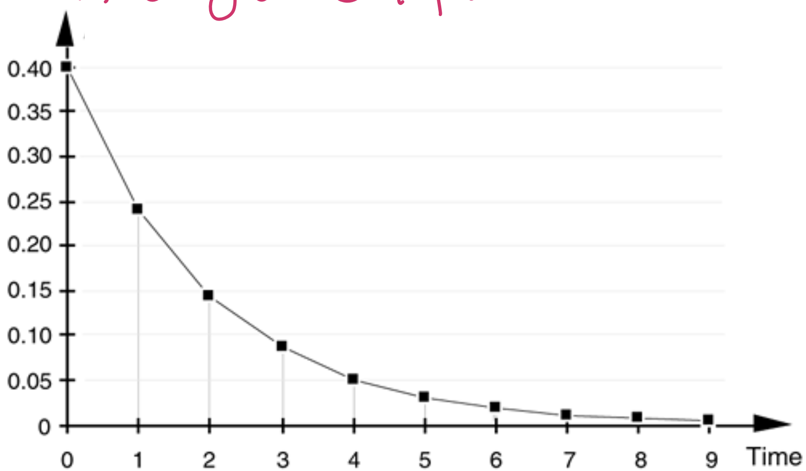


Example: Exponential Averaging Filter

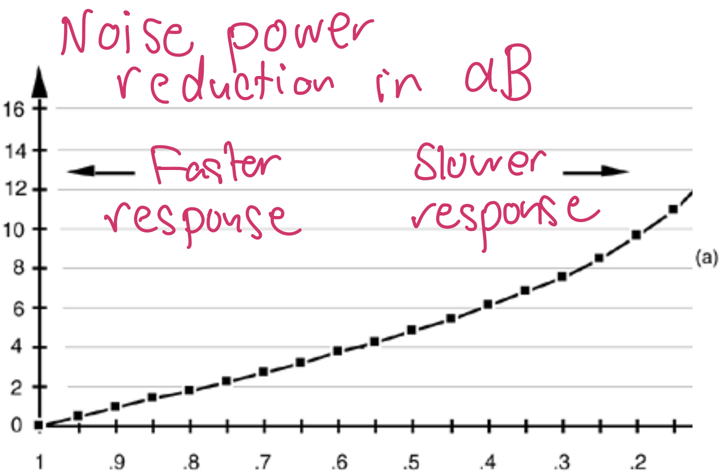
$$y(n) = \alpha x(n) + (1-\alpha) x(n-1)$$

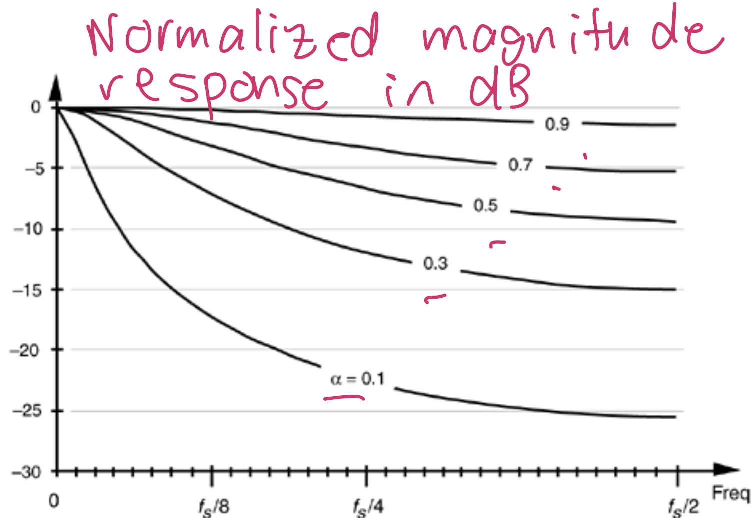


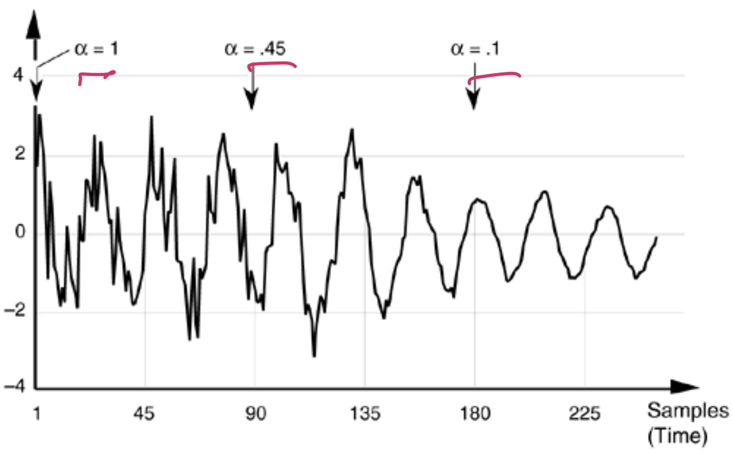
Averager output

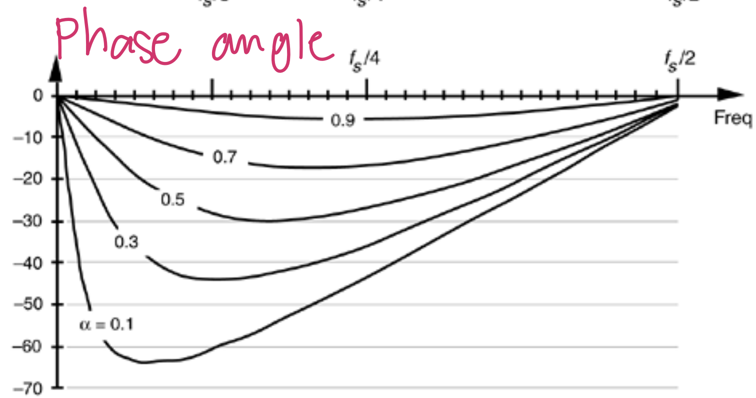


Exponential IIR Filter









What We Have Not Covered

- Many topics to cover, so I so far focused on most relevant immediately
    - 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