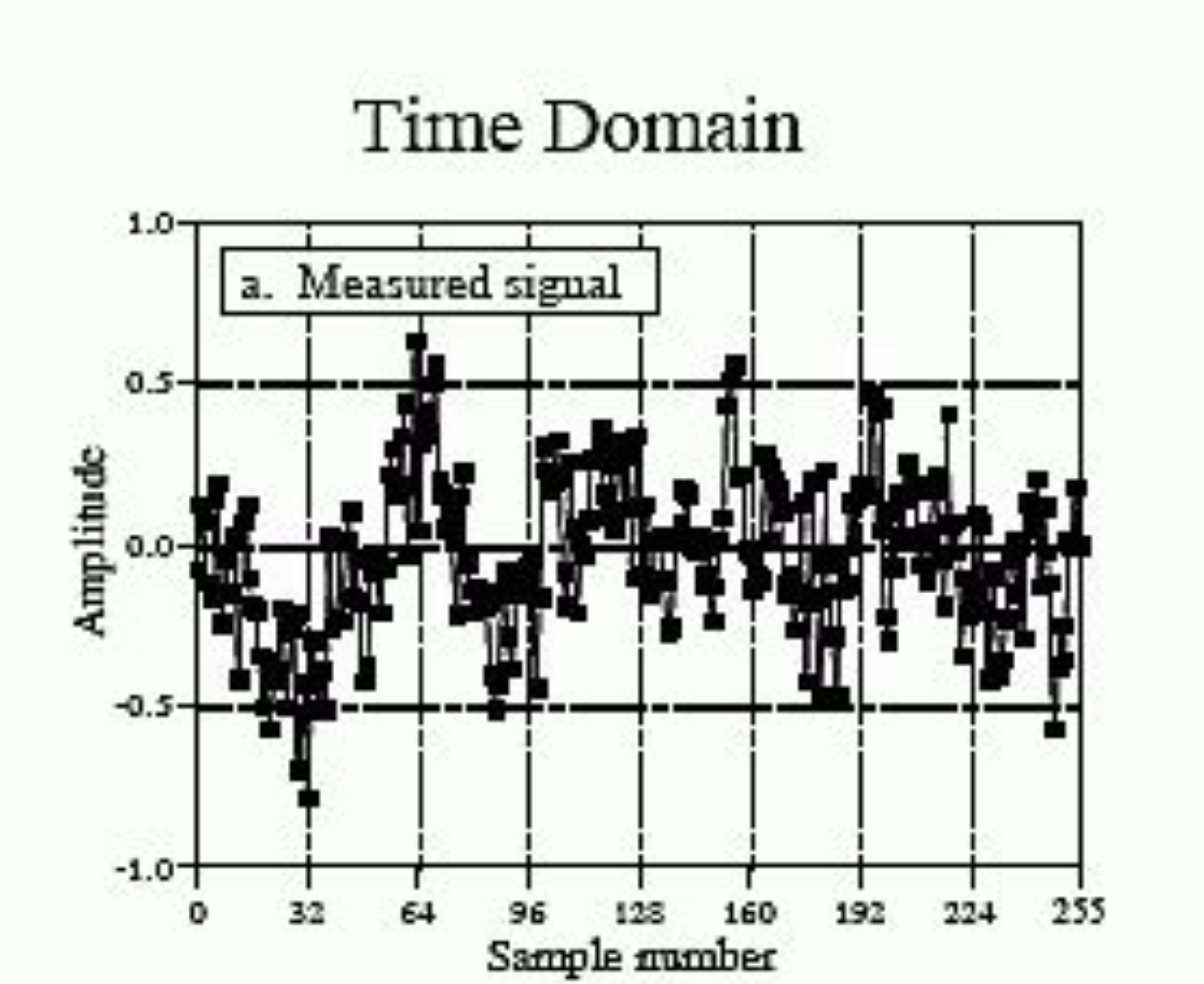
Digital Signal Processing Fundamentals II Digital Filter Design

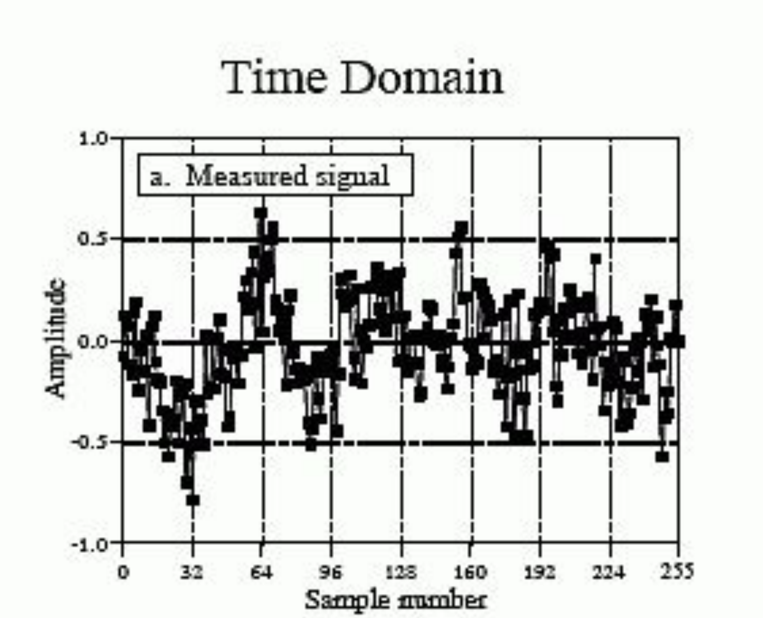
Overview

- Spectral analysis of signals
- Spectral leakage
- Digital filter basics
- FIR(Lowpass) & IIR filters
- Correlation (auto vs. cross)

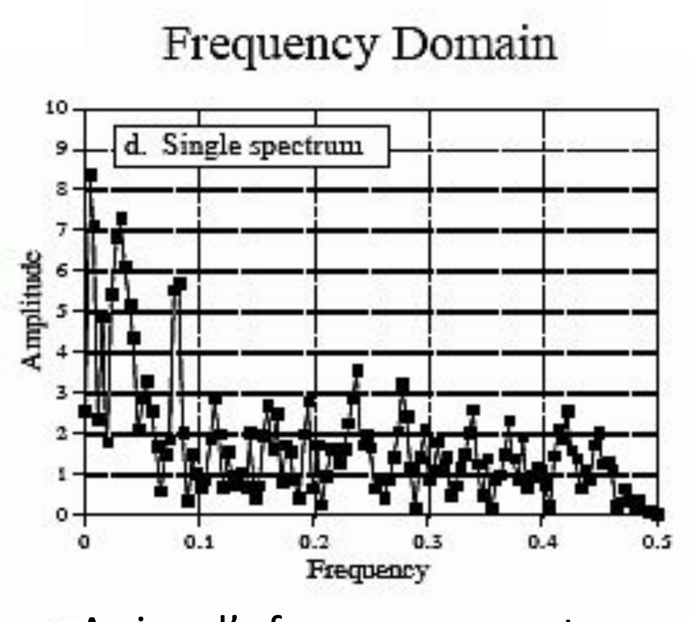
Spectral Analysis



Spectral Analysis

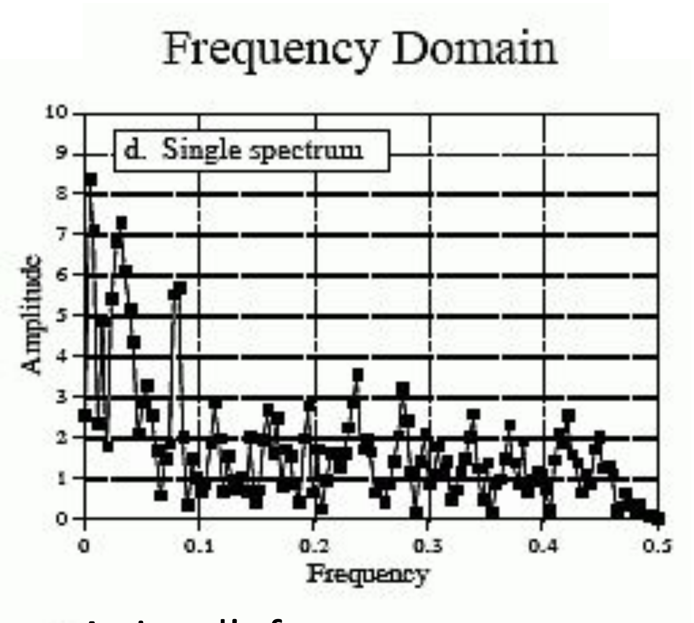


Time domain signal

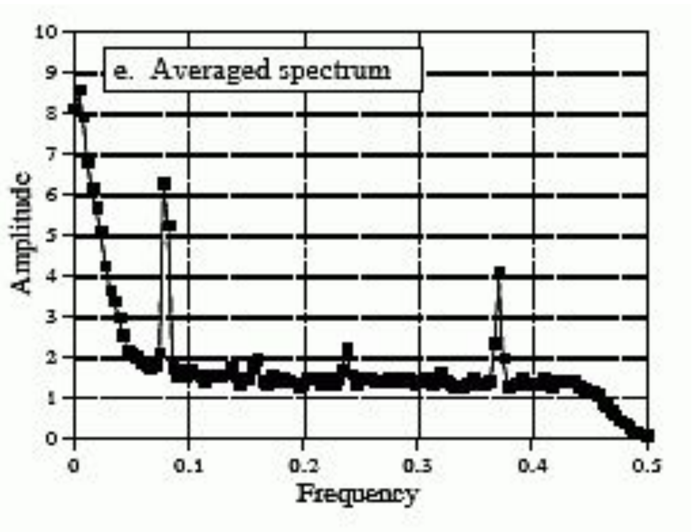


A signal's frequency spectrum

Spectral Analysis



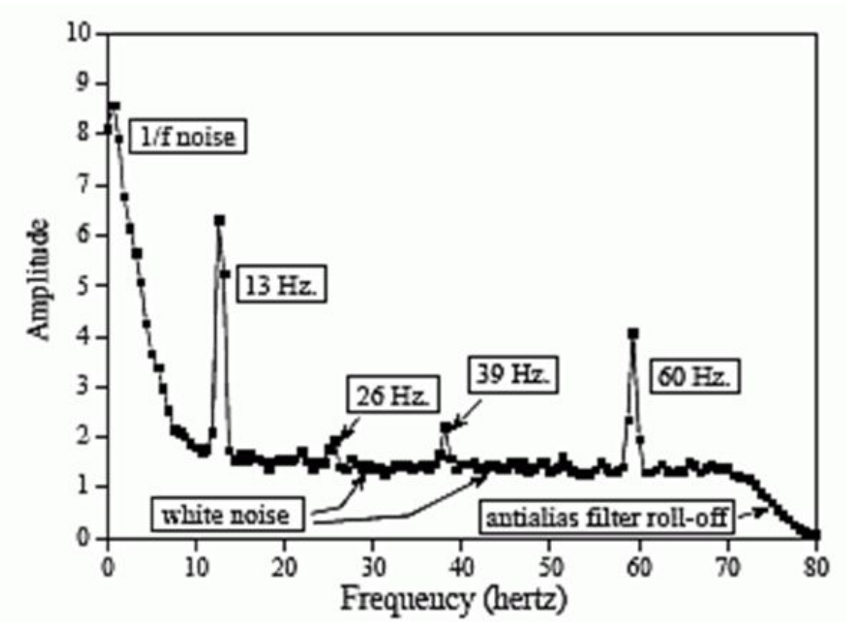
A signal's frequency spectrum



Averaged spectrum: averaging 100 of these spectra **reduces the random noise**

Spectral Analysis

Interpreting signal's frequency spectrum



- **White noise** (10-70 Hz): no information
- Below 10 Hz: **"curiosity" 1/f noise**
    - Mystery noise usually happens in between about 1 and 100 Hz
- **Real signal: strong peak** at 13 Hz + **weaker peaks** at 26 & 39 Hz
    - 13 Hz: fundamental frequency
    - 26 & 39 Hz: second & third harmonic
    - E.g., 13 Hz signal might be generated by a submarine's three bladed propeller turning at 4.33 revolutions per second
- **60 Hz: AC noise** (electromagnetic) (there could be smaller peaks: 120, 180, 240 Hz, etc.)
- **Antialias filter roll-off**: ideal filter would filter above 80 Hz

Why Do Harmonics Occur?

- Non-linearity

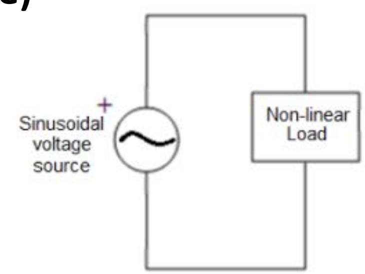
$$x \rightarrow non\text{-}linear\ \ device \rightarrow y = a + bx + cx^2 + dx^3 + \ldots$$

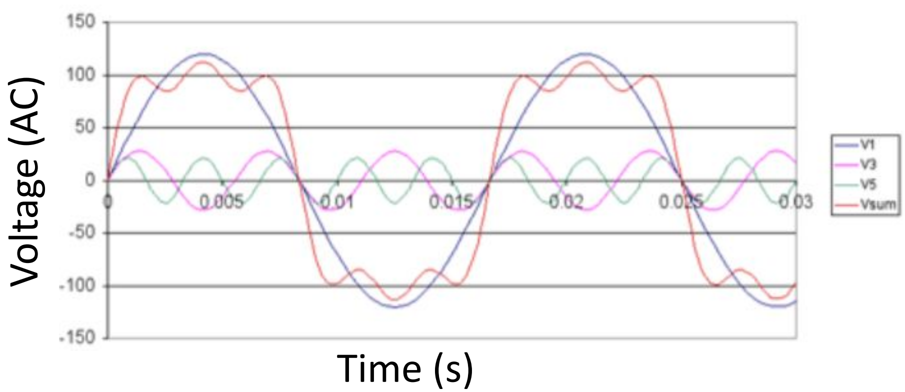
$$x = Acos(wt)$$

$$y = a + bAcos(wt) + \boxed{cA^2cos^2(wt)} + cA^3cos^3(wt) + \ldots$$

$$cA^2cos^2(wt) = cA^2\left(\frac{1+\ cos(2wt)}{2}\right) = \frac{cA^2}{2} + \boxed{\frac{cA^2}{2}cos(2wt)}$$

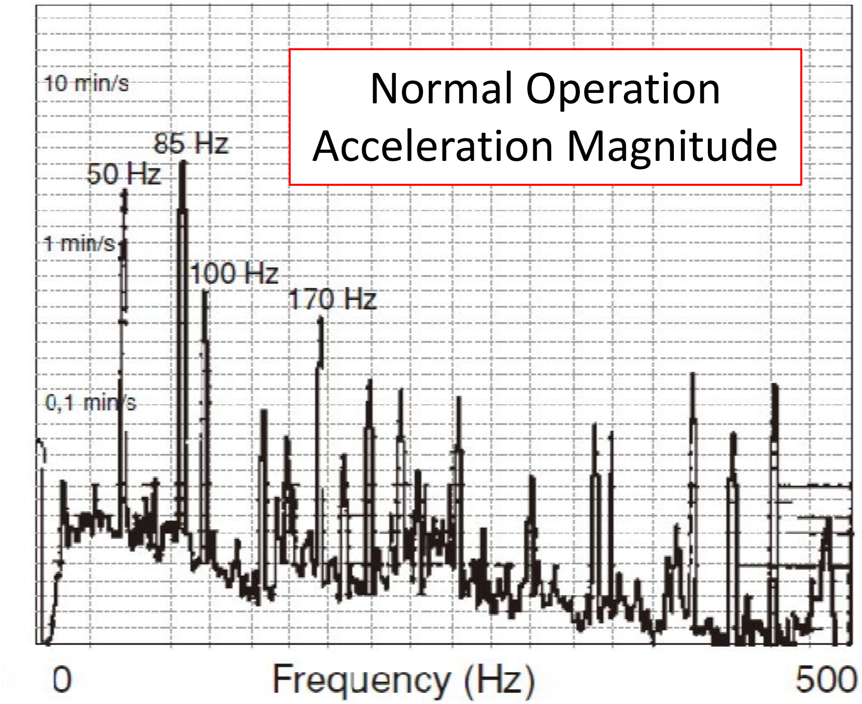
Example)

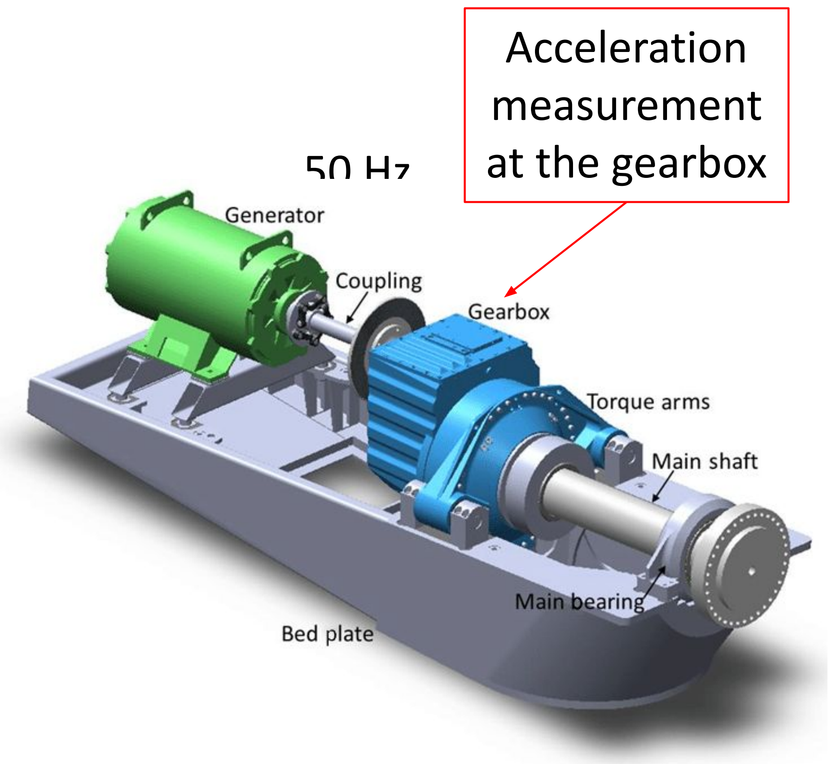




https://guslabview.tistory.com/139
https://www.ecmweb.com/archive/harmonics-made-simple#close-olyticsmodal
https://www.hawaiielectriclight.com/documents/for_business/power_quality/harmonic_primer.pdf

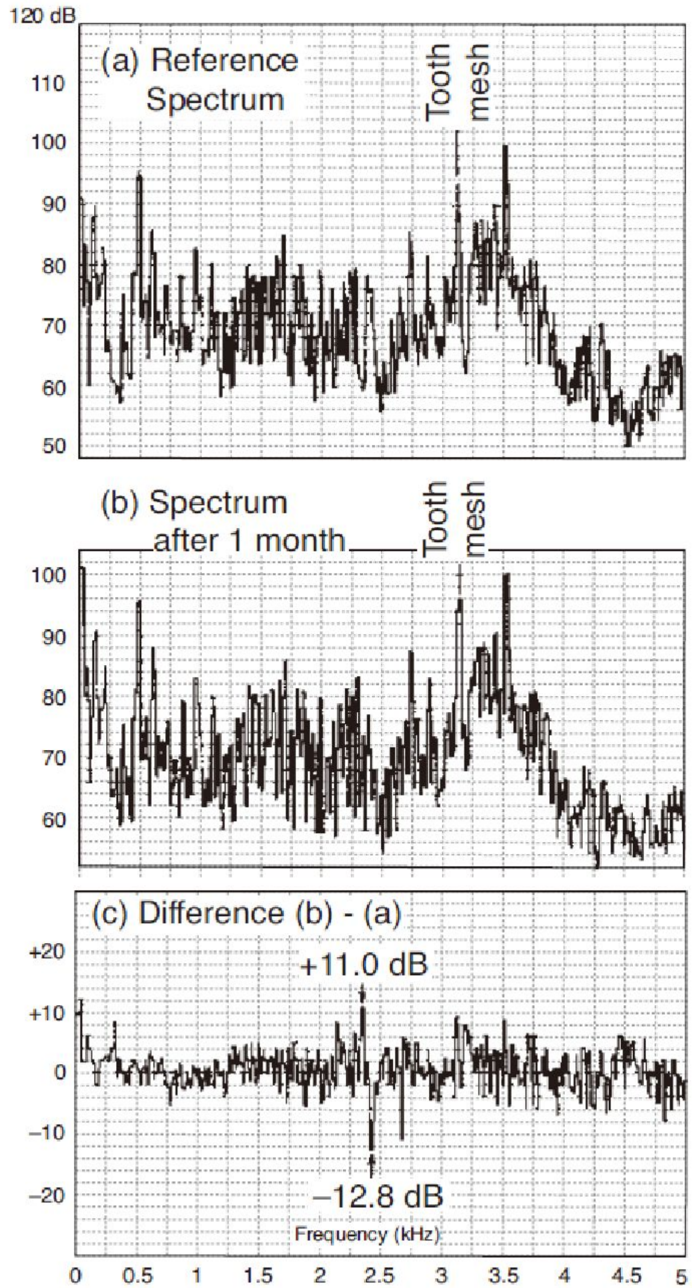
Spectral Analysis



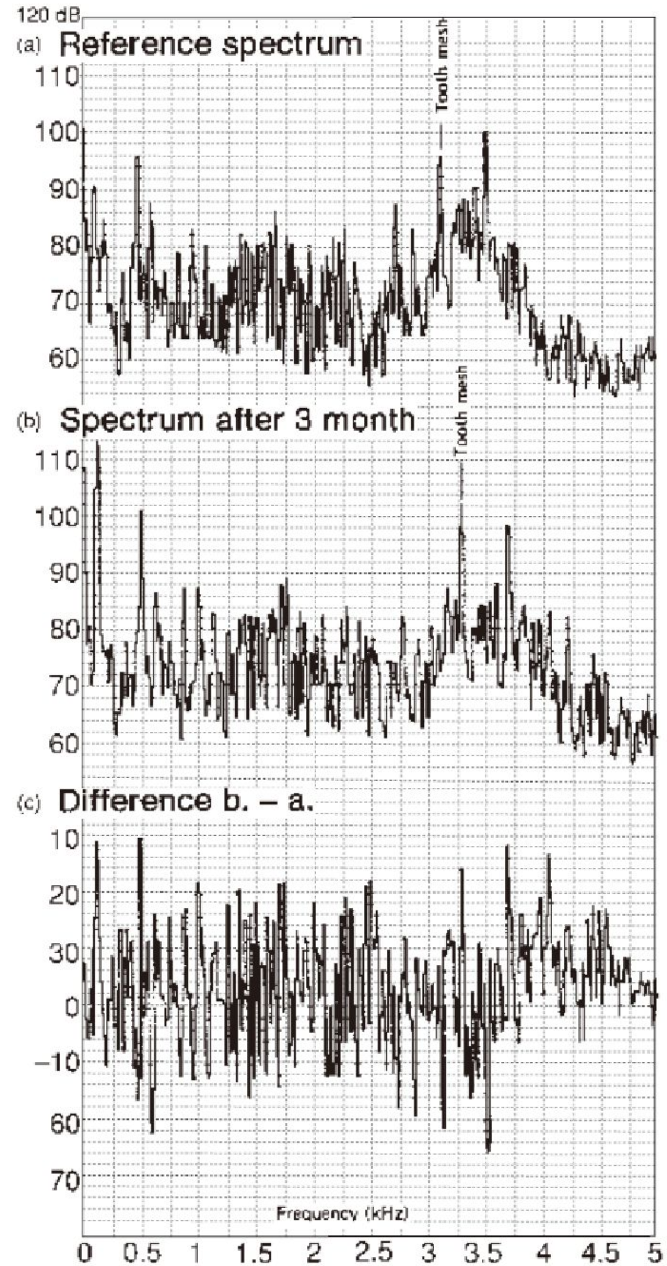


Harmonic signals: 50 & 100 Hz + 85 & 170 Hz

Vibration-based Condition Monitoring, Robert Bond Randall



Direct digital comparison of two spectra with **no change** in machine condition

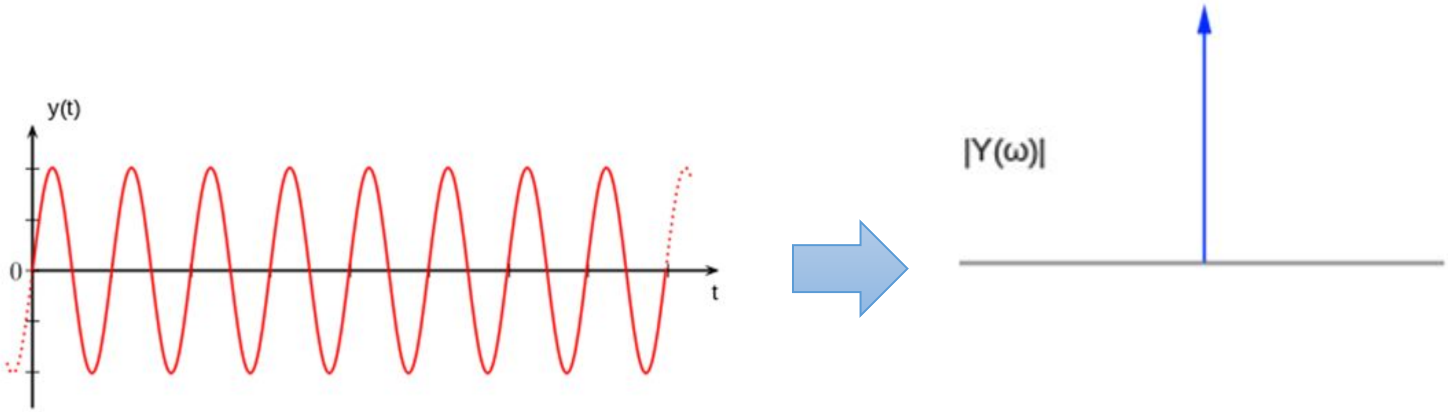


Direct digital comparison of two spectra with a **small speed change** in machine condition

Spectral Analysis

Issue: Spectral Leakage

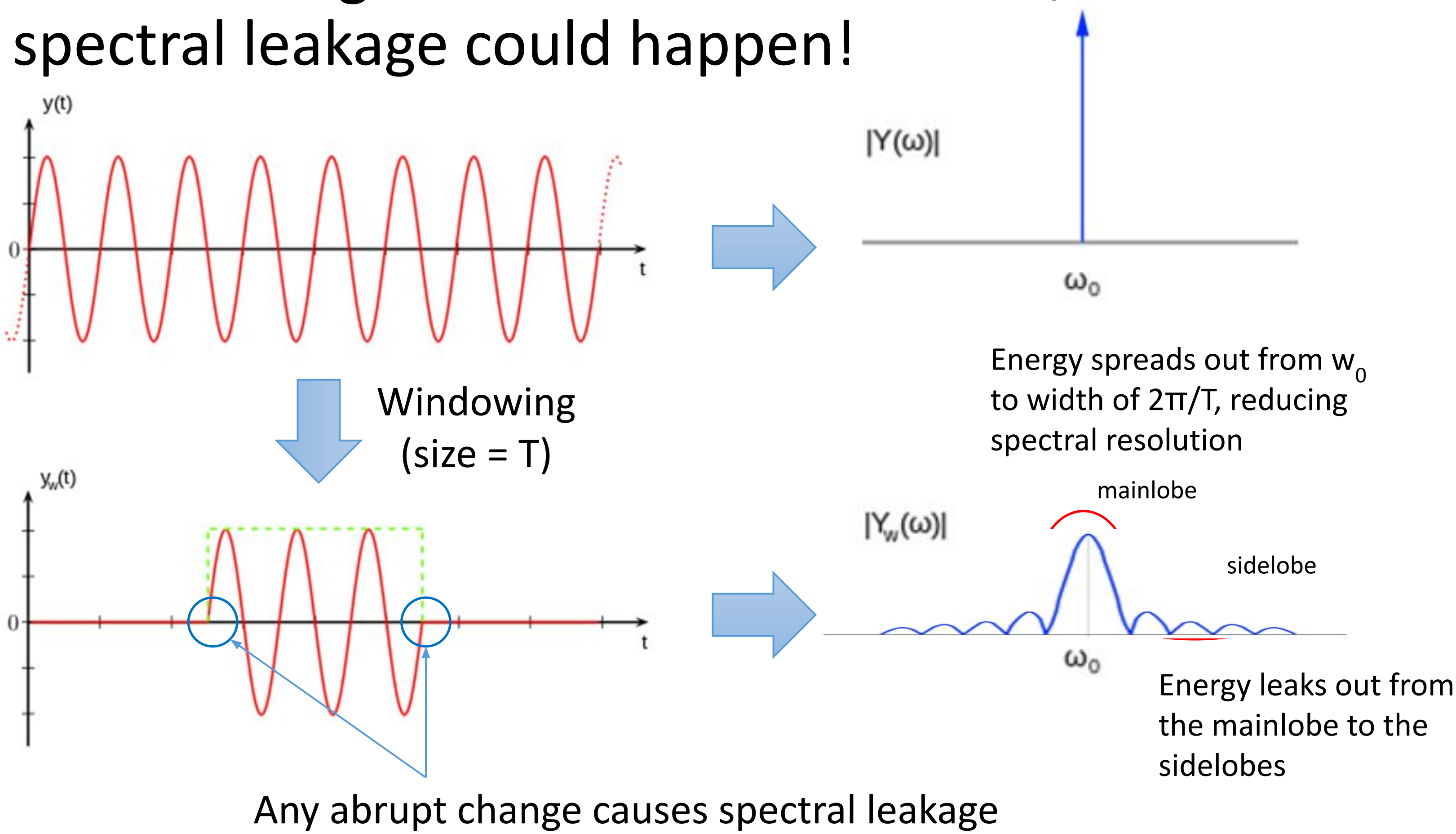
- When having a window of sensor data, spectral leakage could happen!



Spectral Analysis

Issue: Spectral Leakage

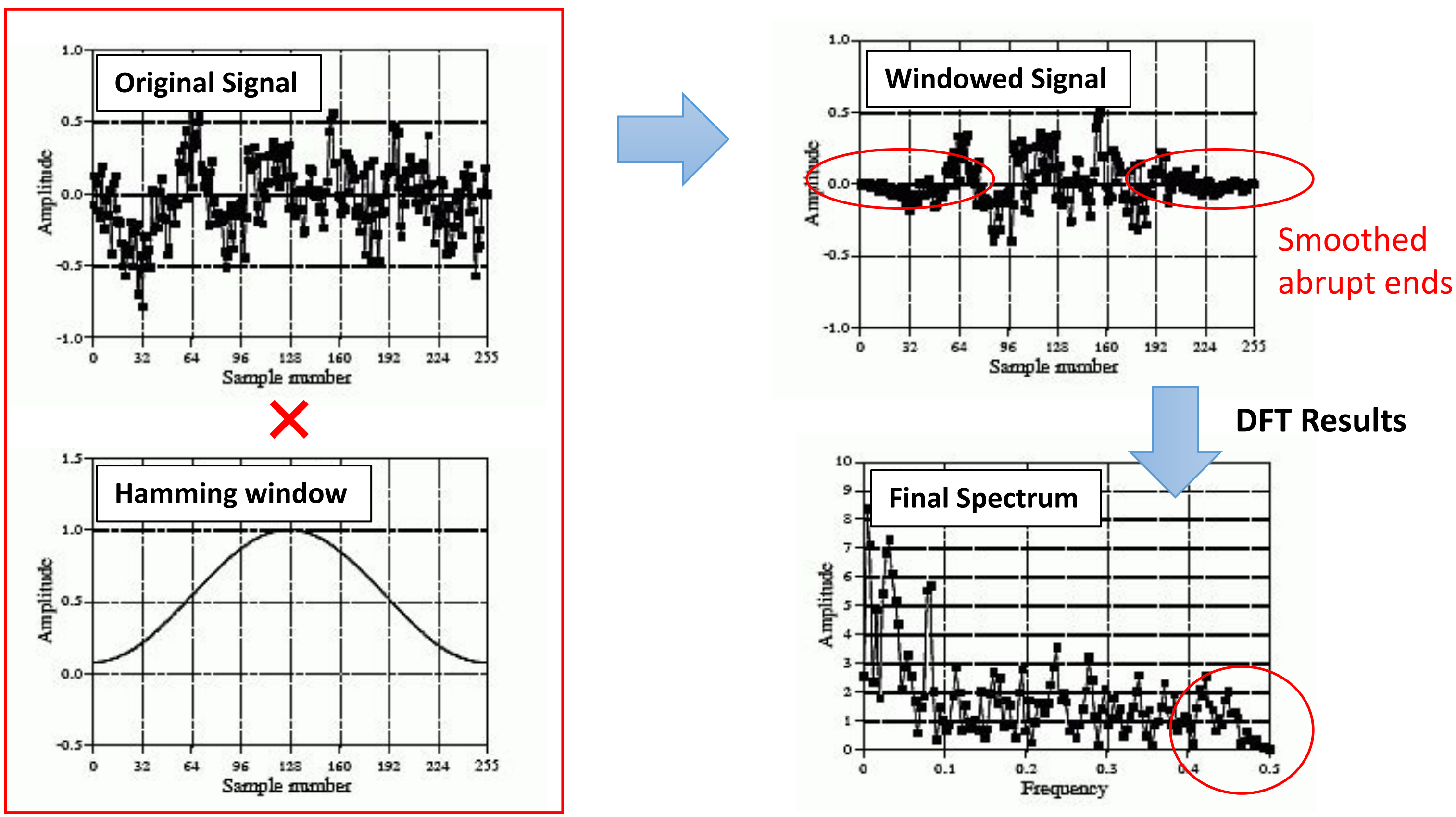
- When having a window of sensor data, spectral leakage could happen!



Spectral Analysis

Issue: Removing Spectral Leakage

- Removing the side effects of truncation (abrupt ends) by smoothing (e.g., Hamming window)



Overview

- Spectral analysis of signals
- Spectral leakage
- Digital filter basics
- FIR(Lowpass) & IIR filters
- Correlation (auto vs. cross)

Digital Filter

What is a Digital Filter?

Digital Filter: numerical procedure or algorithm that transforms a given sequence of numbers into a second sequence that has some more desirable properties.

https://www.slideshare.net/op205/basics-of-digital-filters-presentation

Digital Filter

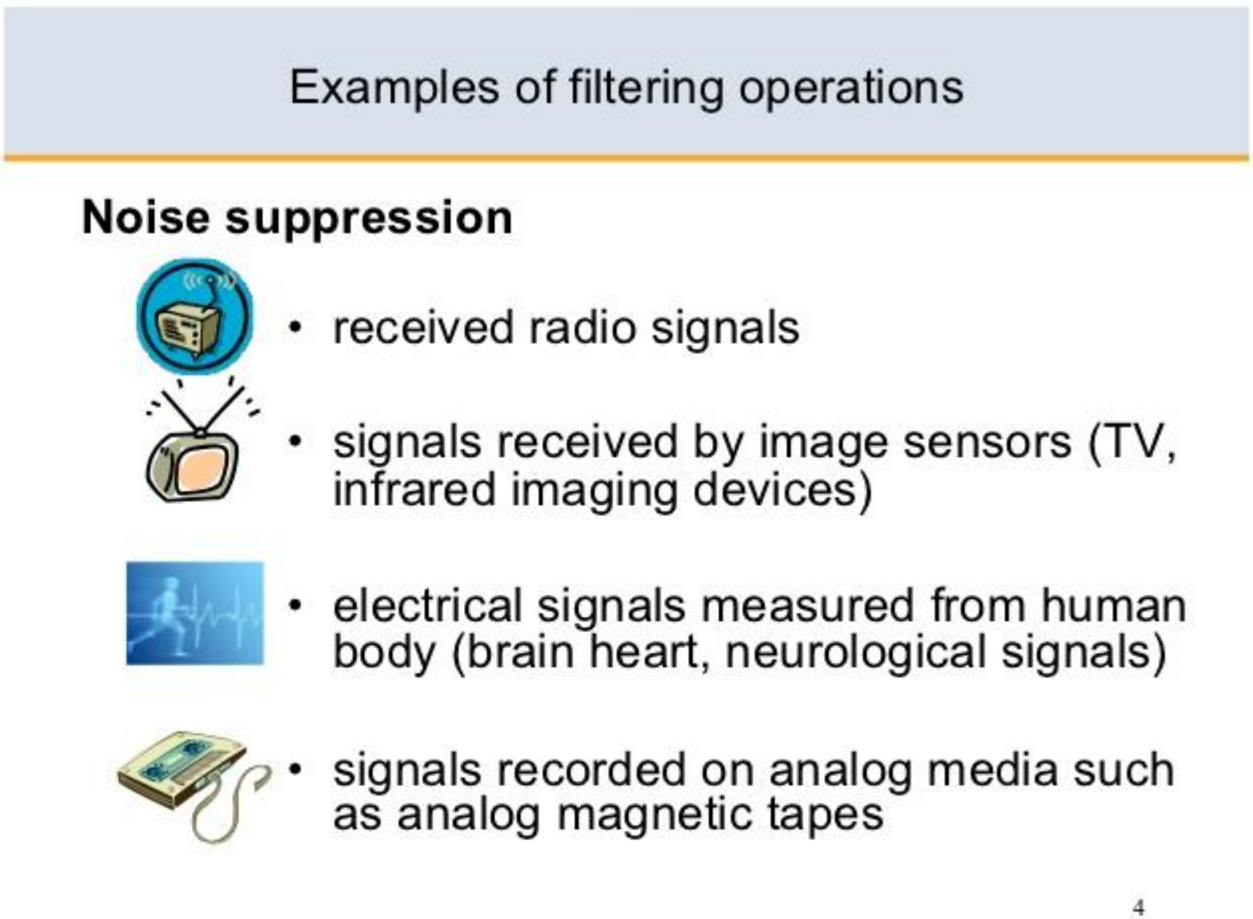
Desired features

Desired features depend on the application, for example

Input Signal		Output
generated by sensing device (microphone)	→	having less noise or interferences
speech	→	with reduced redundancy for more efficient transmission

https://www.slideshare.net/op205/basics-of-digital-filters-presentation

Digital Filter

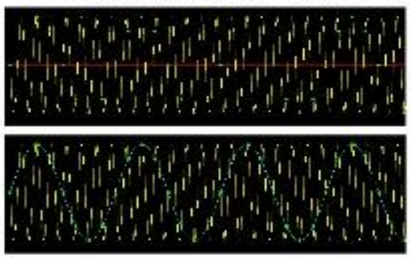


https://www.slideshare.net/op205/basics-of-digital-filters-presentation

Digital Filter

Examples of filtering operations

Bandwidth limiting



- means of aliasing prevention in sampling
- communication

  radio or TV signal transmitted over specific channel has to have a limited bandwidth to prevent interference with neighbouring channels

  frequency components outside the permitted band are attenuated below a specific power level

6

https://www.slideshare.net/op205/basics-of-digital-filters-presentation

Digital Filter

First method in time domain: Linear difference equations

The linear time-invariant digital filter can then be described by the **linear difference equation**:

$$\begin{aligned} y_n &= -a_1 y_{n-1} - a_2 y_{n-2} - \ldots - a_N y_{n-N} + b_0 x_n + \ldots + b_M x_{n-M} \\ &= -\sum_{k=1}^{N} a_k y_{n-k} + \sum_{k=0}^{M} b_k x_{n-k} \end{aligned}$$

where $\{a_k\}$ and $\{b_k\}$ real

The *order* of the filter is the larger of M or N

https://www.slideshare.net/op205/basics-of-digital-filters-presentation

Digital Filter

Second method in time domain: unit-sample response

- Input signal is resolved into a weighted sum of elementary signal components, i.e. sum of unit samples or impulses

$$\boxed{x_n = \sum_{k=-\infty}^{\infty} x_k \, \delta_{n-k}}$$

- The response of the system to the unit sample sequence is determined

$$\delta_n \longrightarrow \boxed{\textbf{LTI}} \longrightarrow h_n$$

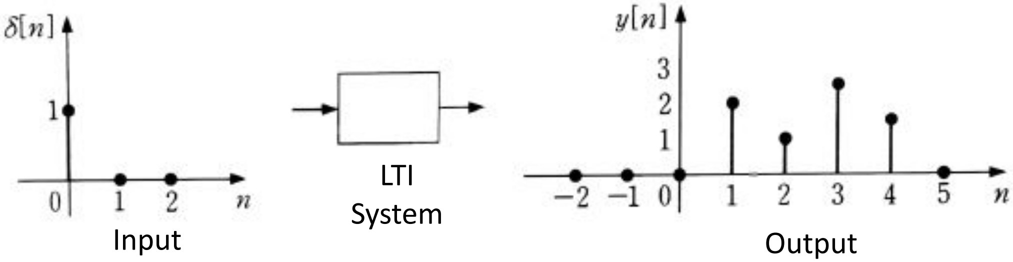
- Taking into account properties of the LTI system, the response of the system to x_n is the corresponding sum of weighted outputs

$$x_n = \sum_{k=-\infty}^{\infty} x_k \, \delta_{n-k} \longrightarrow \boxed{\textbf{LTI}} \longrightarrow y_n = \sum_{k=-\infty}^{\infty} x_k \, h_{n-k}$$

17

https://www.slideshare.net/op205/basics-of-digital-filters-presentation

Digital Filter



Linear convolution

Linear convolution

$$y_n = \sum_{k=-\infty}^{\infty} x_k h_{n-k}$$

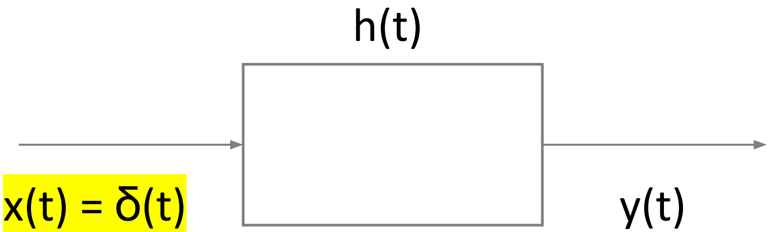
gives the response of the LTI system as a function of the input signal and the unit sample (impulse) response

LTI is completely characterized by h_n

https://www.slideshare.net/op205/basics-of-digital-filters-presentation

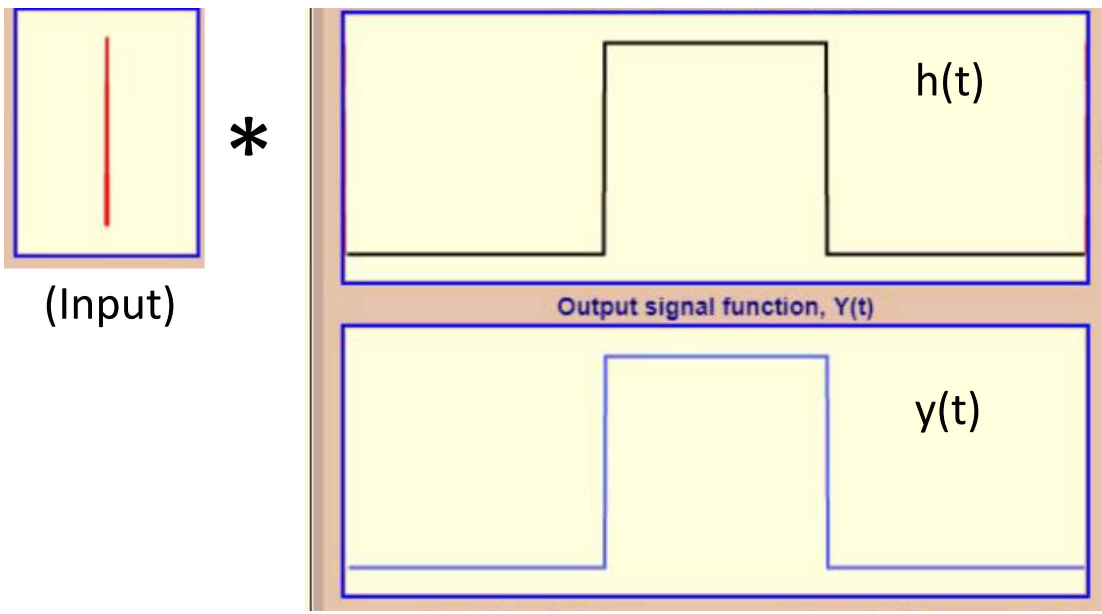
Digital Filter

Impulse Response Function



$y(t) = x(t) * h(t)$

x(t) = δ(t)



(Input)

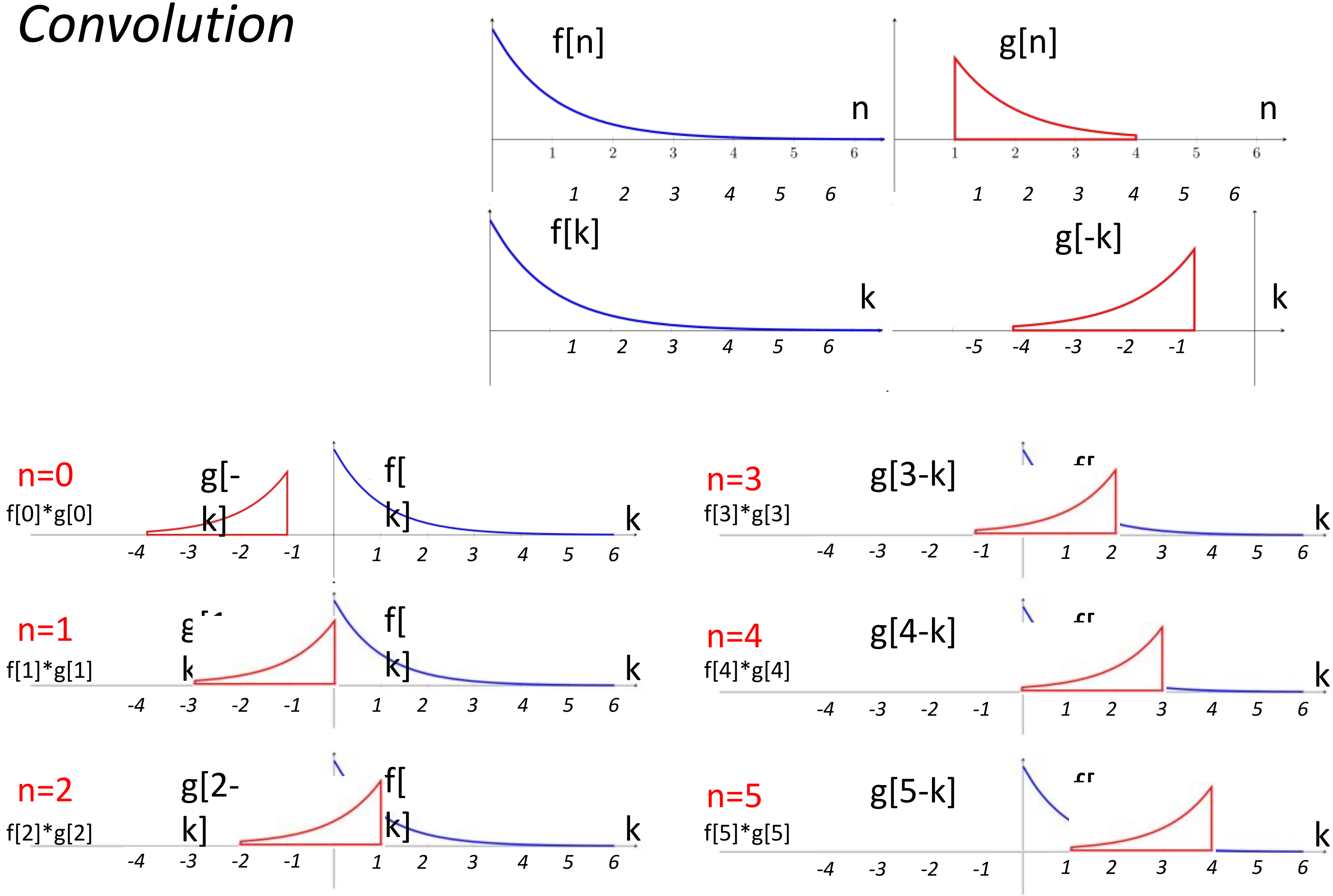
h(t)
impulse response
function

http://195.134.76.37/applets/AppletConvol/Appl_Convol2.html

Digital Filter

Convolution

$$f[n] * g[n] = \sum_{k=-\infty}^{\infty} f[k]g[n-k]$$



Digital Filter

Convolution

$$f[n] * g[n] = \sum_{k=-\infty}^{\infty} f[k]g[n-k]$$

Convolute two sequences
x[n] = {a,b,c} & h[n] = [e,f,g]

	a	b	c
e	ea	eb	ec
f	fa	fb	fc
g	ga	gb	gc

Convoluted output =
[ea, eb+fa, ec+fb+ga, fc+gb, gc]

Convolute two sequences
x[n] = {1,2,3} & h[n] = {-1,2,2}

×	1	2	3
-1	-1	-2	-3
2	2	4	6
2	2	4	6

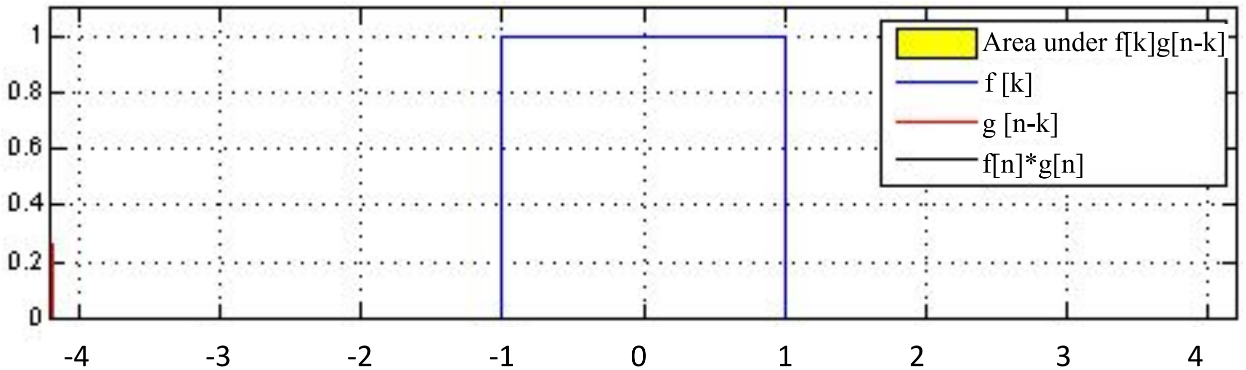
Convoluted output =
[-1, -2+2, -3+4+2, 6+4, 6]

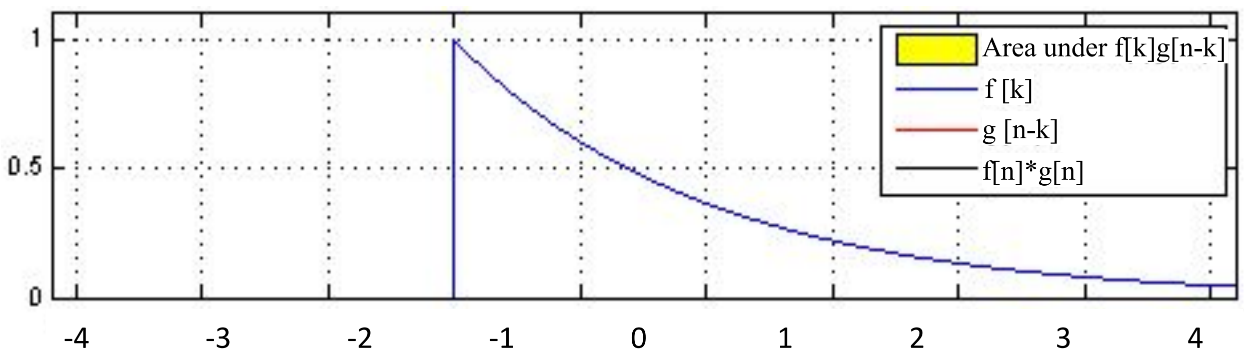
https://www.tutorialspoint.com/signals_and_systems/convolution_and_correlation.htm

Digital Filter
Convolution

$$f[n] * g[n] = \sum_{k=-\infty}^{\infty} f[k]g[n-k]$$

- Convolution

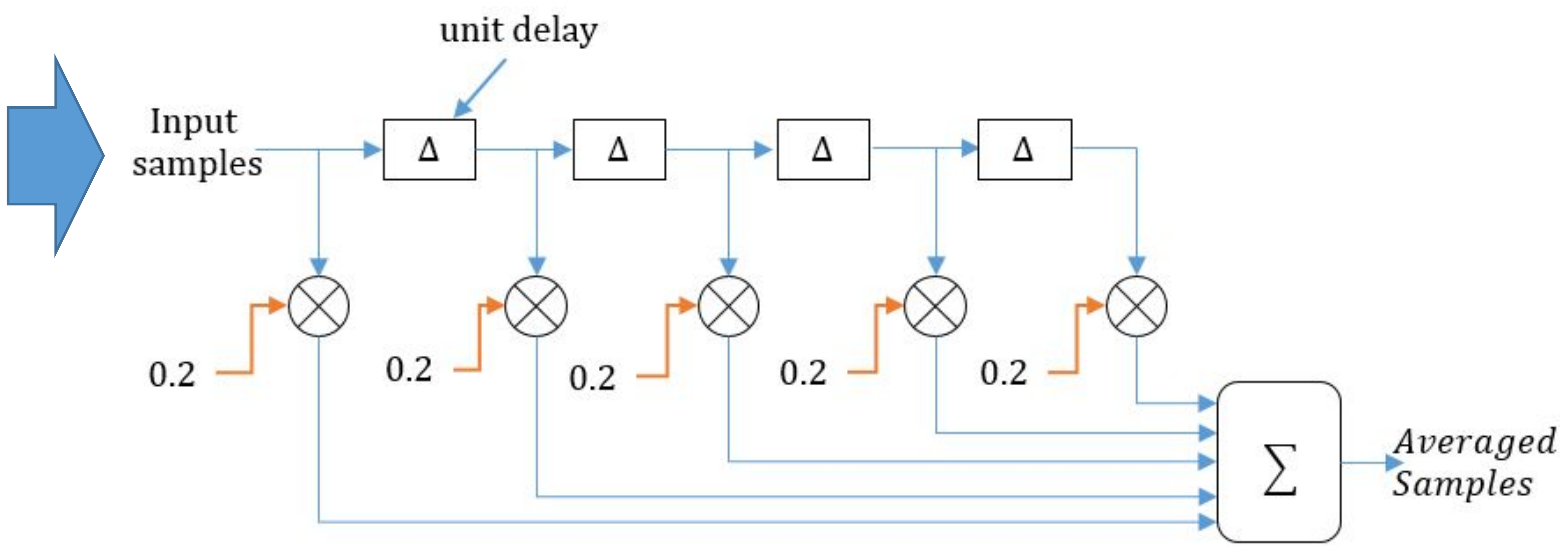




https://en.wikipedia.org/wiki/Convolution

Digital Filter

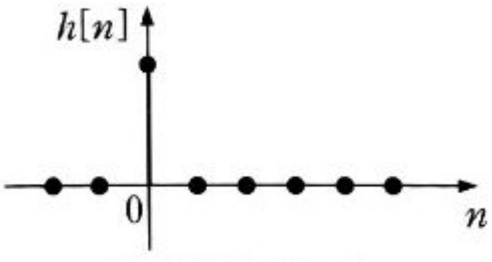
Convolution – Moving Average Filter



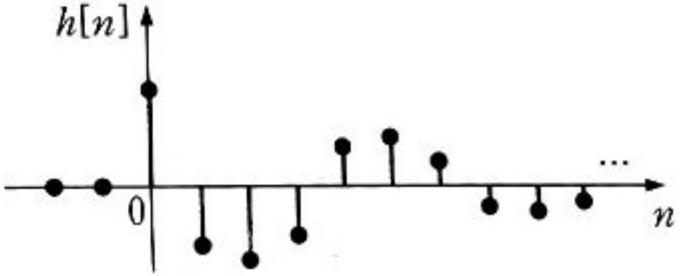
https://www.gaussianwaves.com/2010/11/moving-average-filter-ma-filter-2

Digital Filter

- Finite vs. Infinite Impulse Response



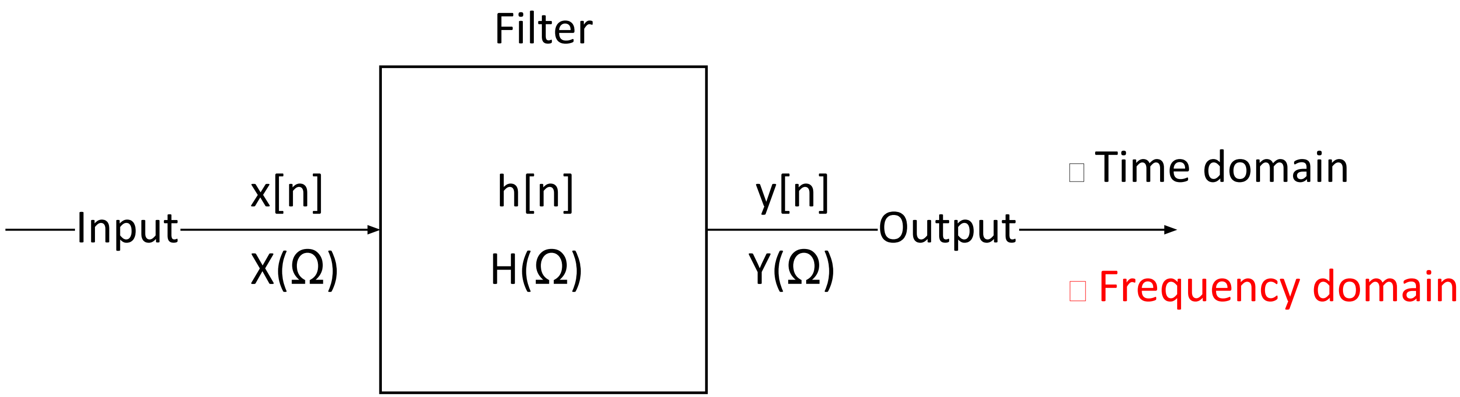
Vs.



FIR: Finite Number of Samples

IIR: Infinite Number of Samples

Digital Filter



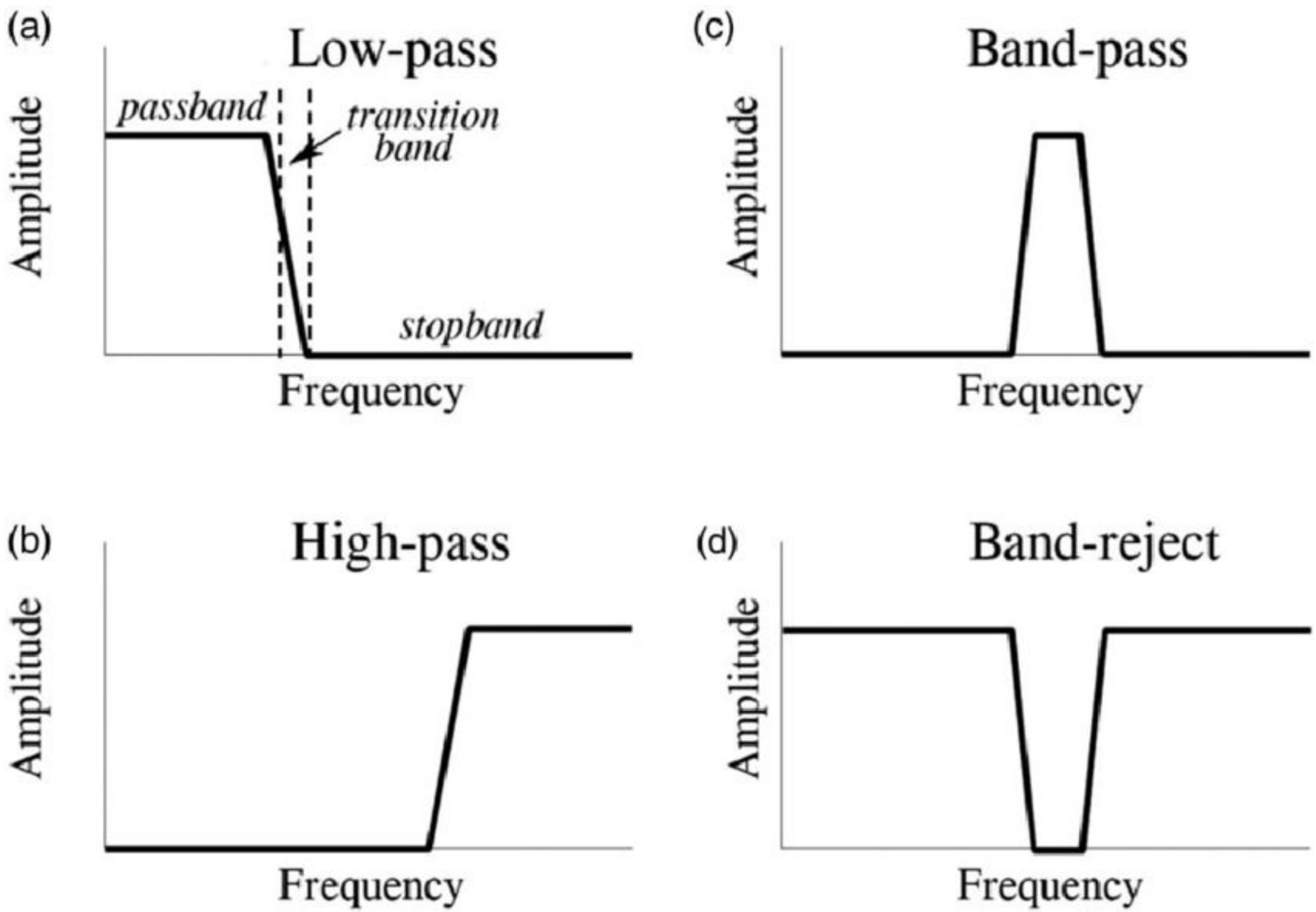
Filter only “wanted signals”

H(Ω) : filter’s transfer function

$y[n] = x[n] \otimes h[n]$ (convolution)

$Y(\Omega) = X(\Omega)H(\Omega)$ (multiplication) => $H(\Omega) = Y(\Omega) / X(\Omega)$

Digital Filter

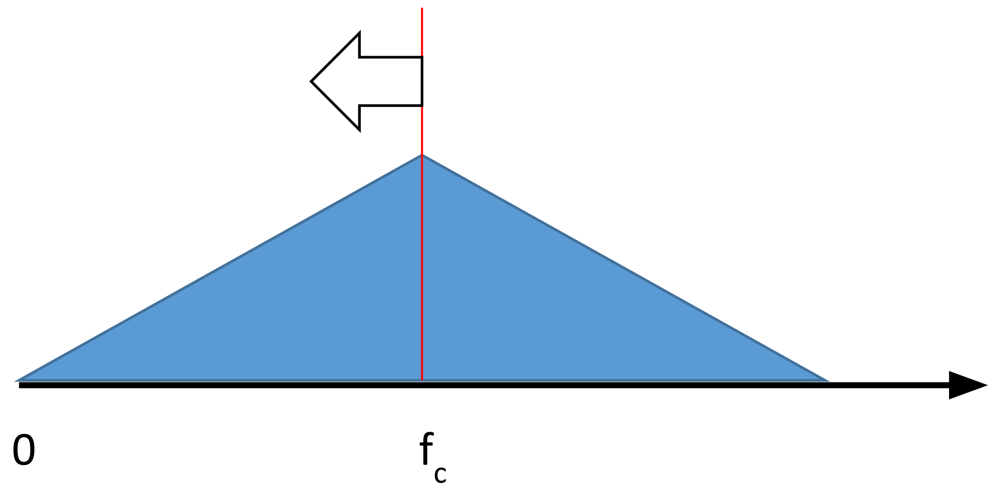


Evaluation of arterial stiffness by finger-toe pulse wave velocity: optimization of signal processing and clinical validation, 2017

Overview

- Spectral analysis of signals
- Spectral leakage
- Digital filter basics
- FIR(Lowpass) & IIR filters
- Correlation (auto vs. cross)

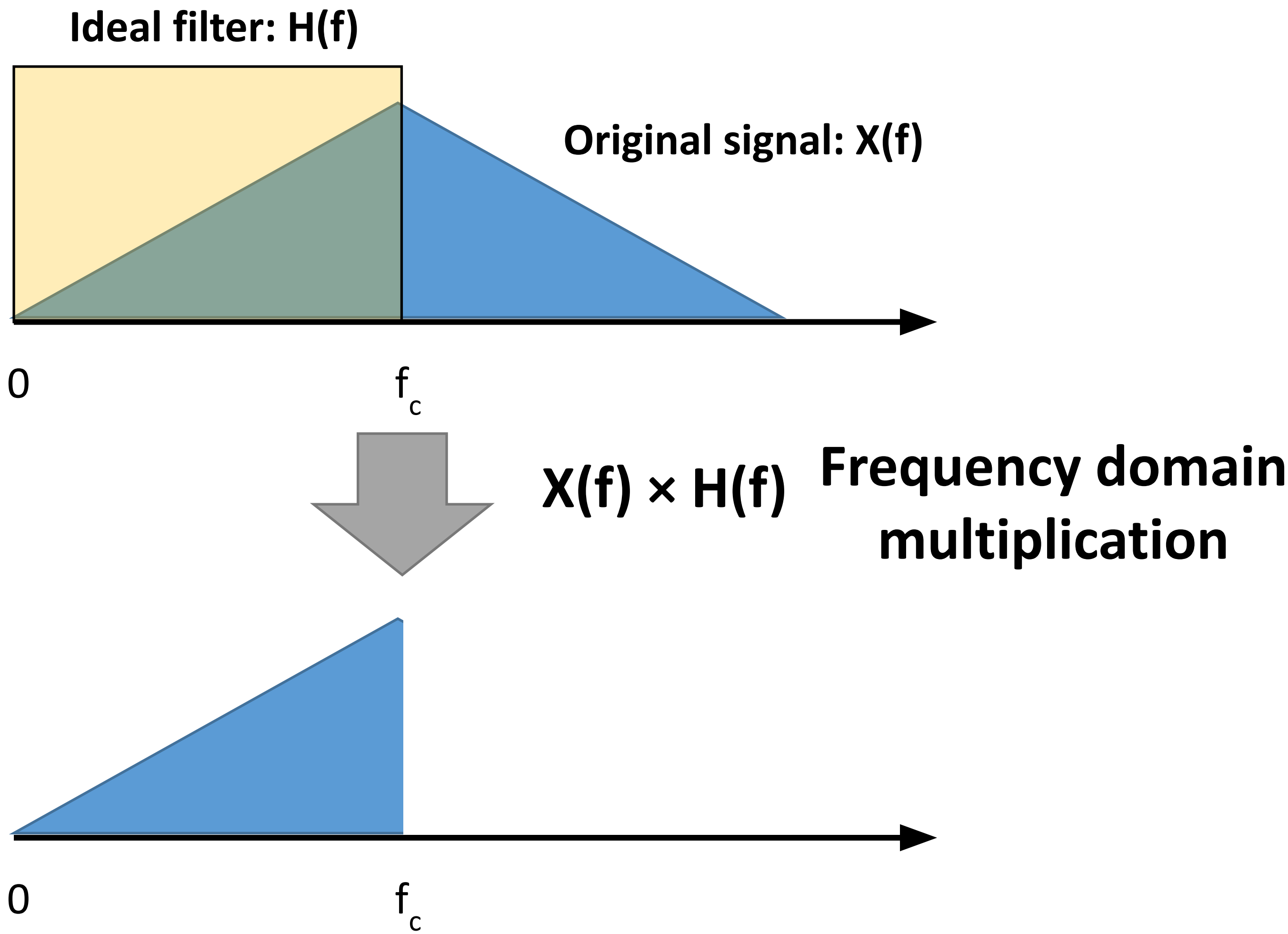
Low pass filter



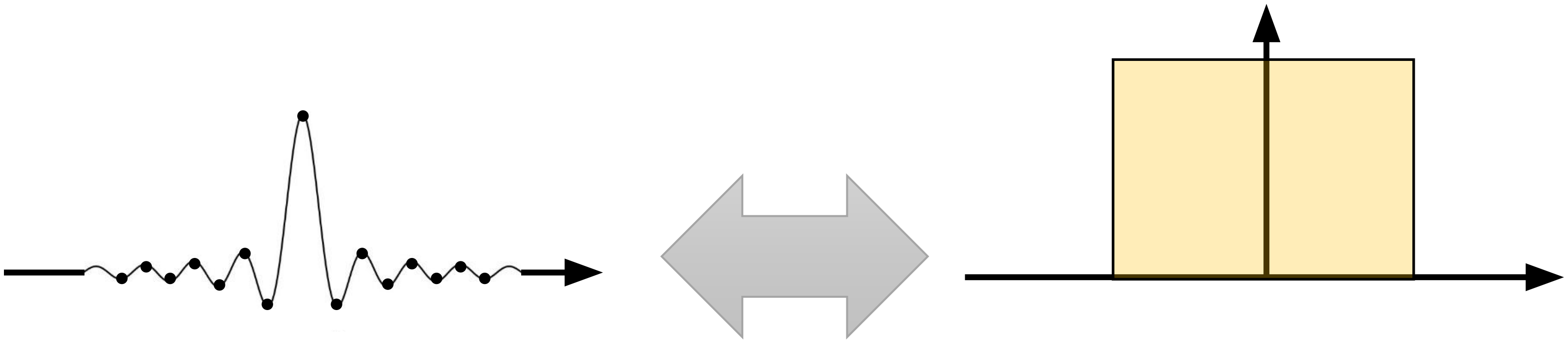
How can we keep only low frequency components (i.e., below f_c)?

Low pass filter – Ideal Filter

Frequency domain



Low pass filter – Ideal Filter
Frequency domain



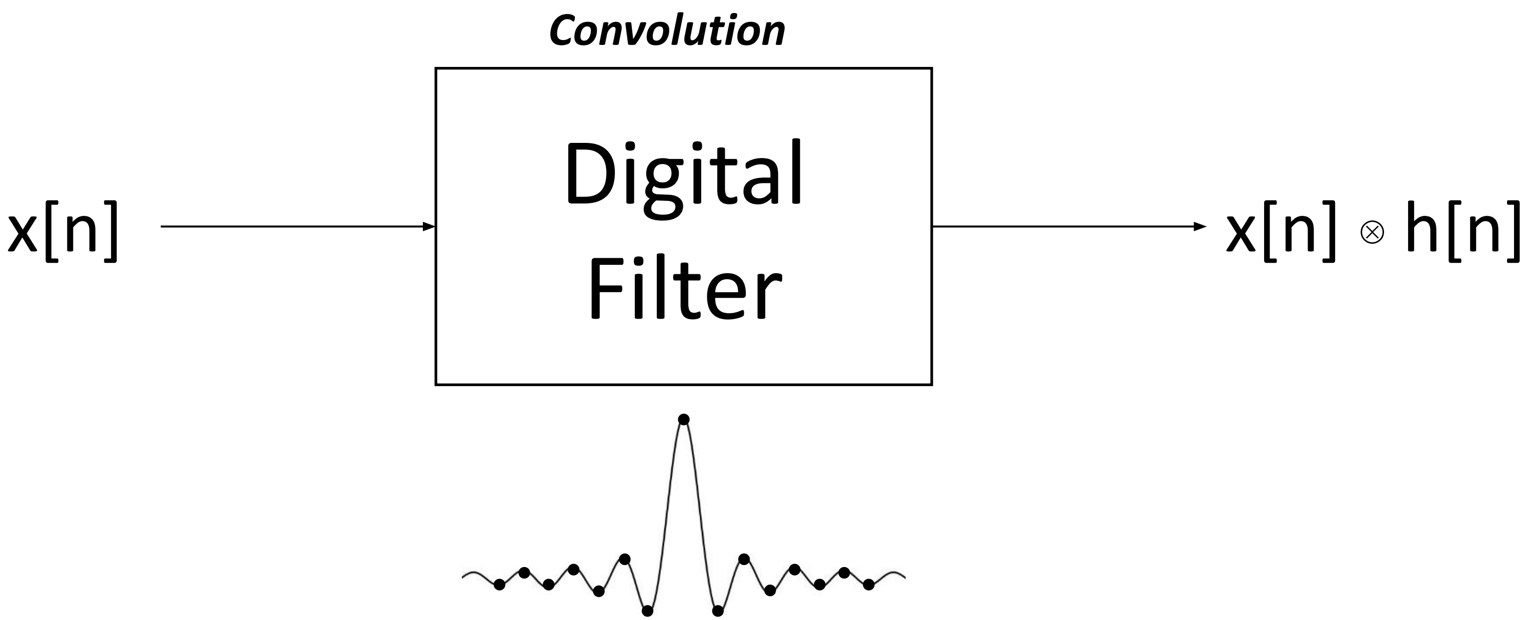
$h[n] = \sin n / n$

$H(f) = \mathrm{rect}(f)$

Time domain

Frequency domain

Low pass filter – Ideal Filter
Time domain

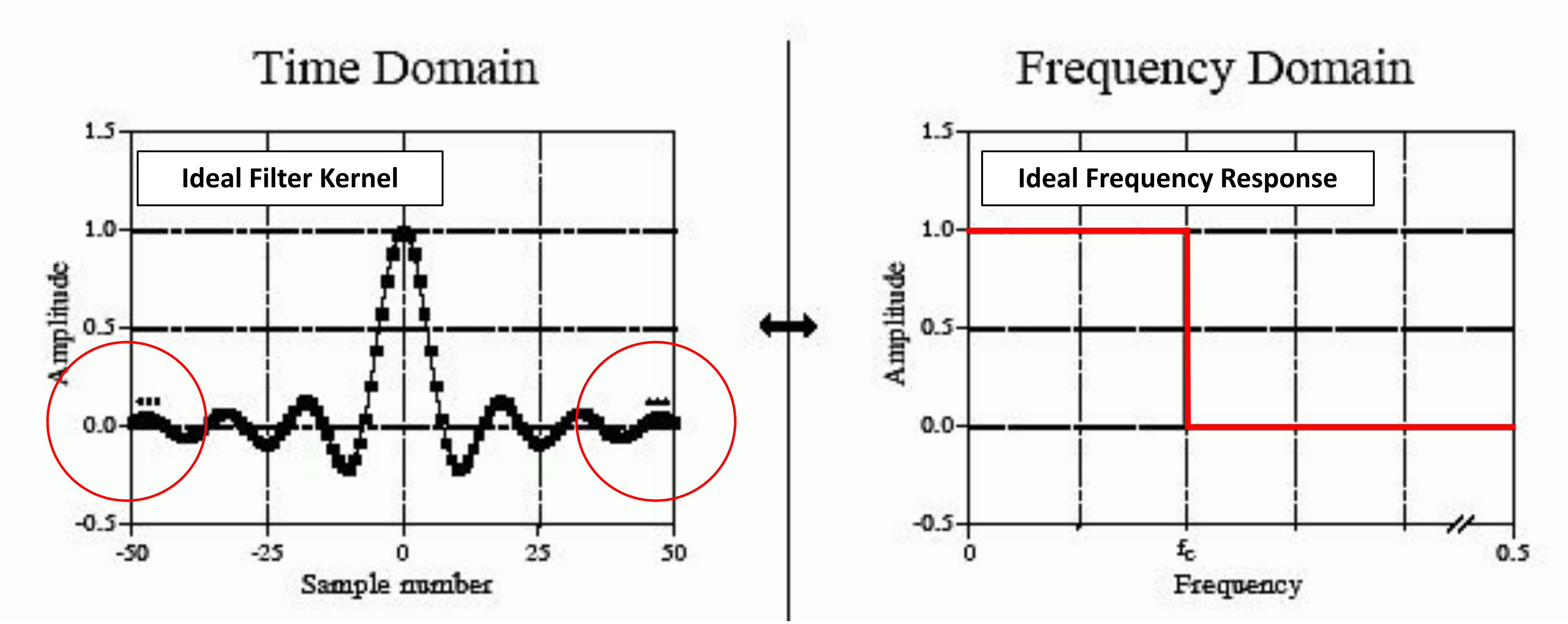


$h[n] = \sin n / n$

Frequency domain multiplication => Time domain convolution

Low pass filter – Ideal Filter

Issue: Infinite tails (infinite impulse response!)



$$h[i] = \frac{\sin(2\pi f_c i)}{i\pi}$$

Problem: sinc function (= sin x / x) has infinite tails (= infinite impulse responses)

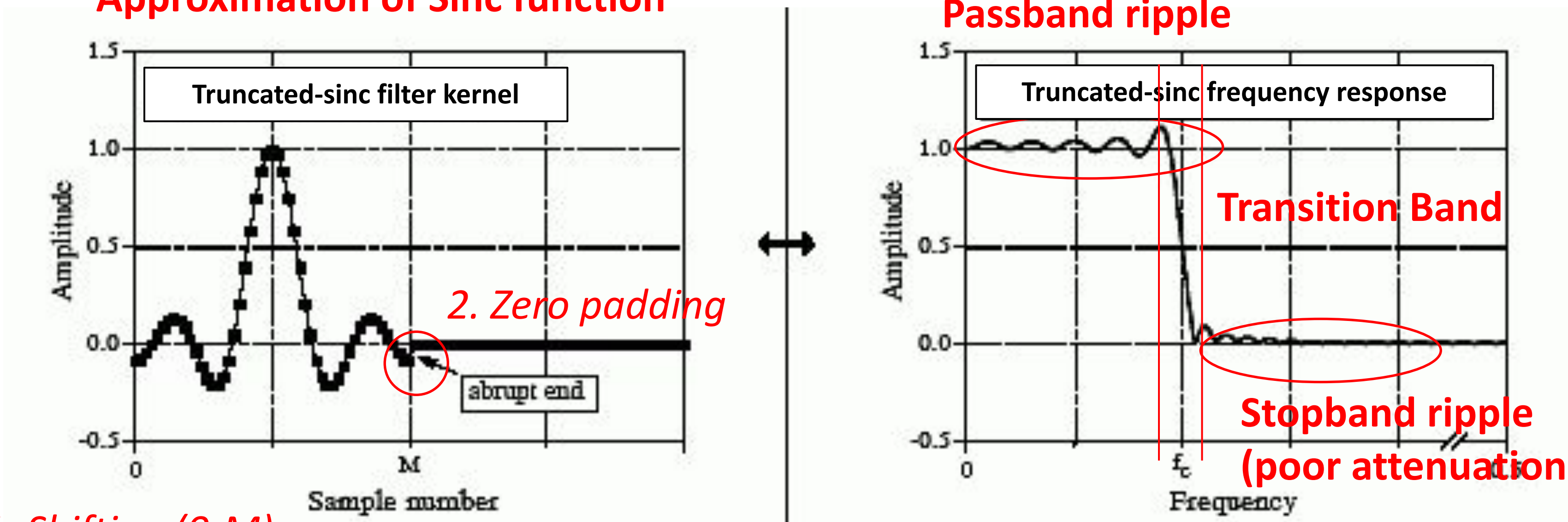
=> Finite length DFT cannot capture these infinite tails, missing valuable information (transform back to frequency domain will not be the same as before!)

=> Also, time domain signals are non-causal (meaning that we have negative time indices)

Low pass filter – Ideal Filter
Issue: Truncating infinite tails

Time Domain **Frequency Domain**

Approximation of Sinc function



1. Shifting (0-M)
(making it as a causal system)

Low pass filter – Ideal Filter

Issue: Truncating infinite tails + windowing

EQUATION 16-1
The Hamming window. These windows run from $i = 0$ to M, for a total of $M+1$ points.

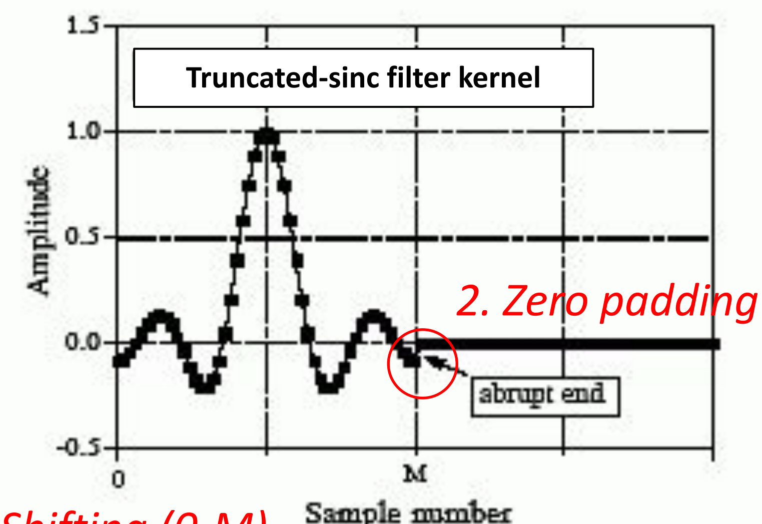
$$w[i] = 0.54 - 0.46\cos(2\pi i/M)$$

EQUATION 16-2
The Blackman window.

$$w[i] = 0.42 - 0.5\cos(2\pi i/M) + 0.08\cos(4\pi i/M)$$

Time Domain

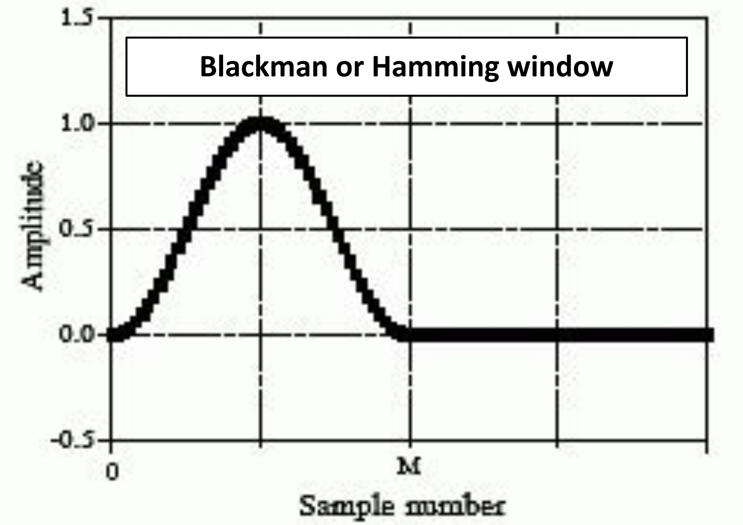
Approximation of Sinc function



1. Shifting (0-M)

×

Frequency Domain

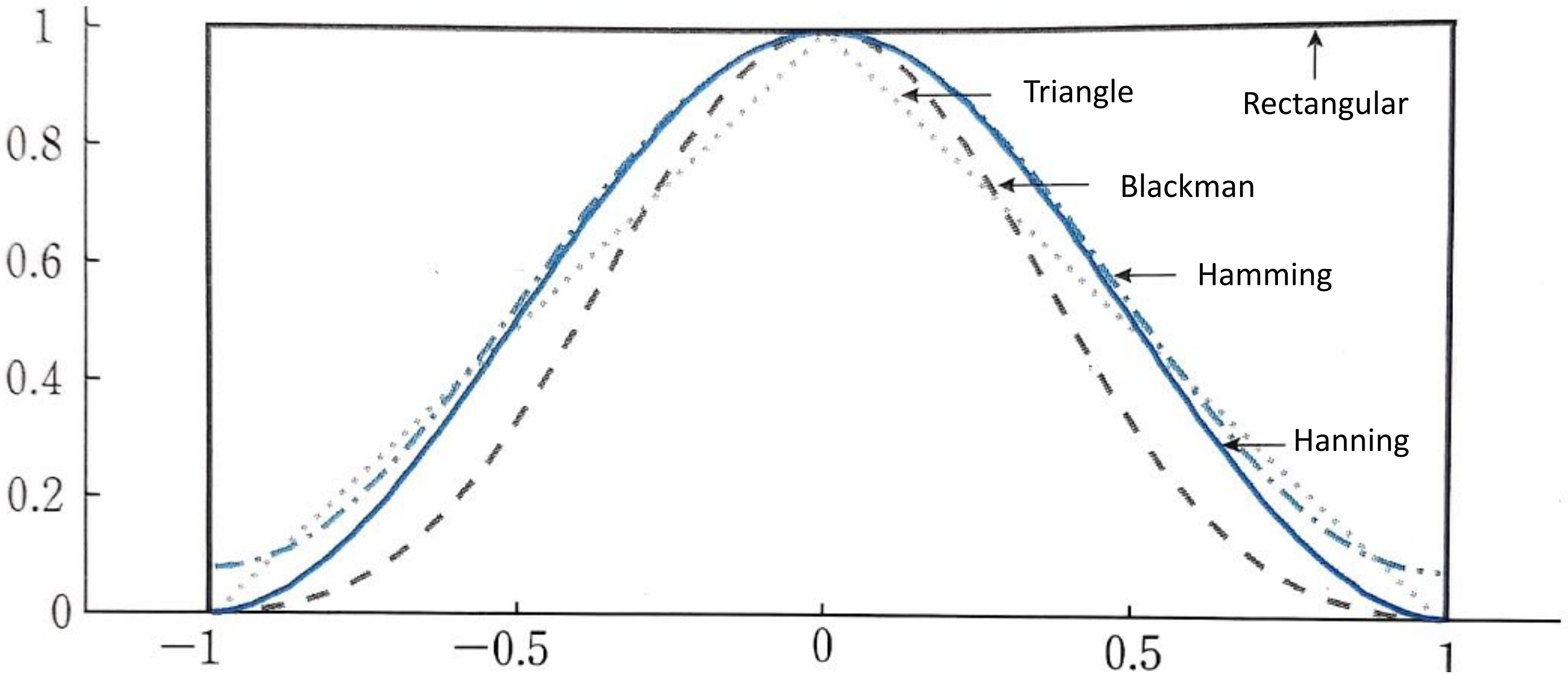


Removing abrupt ends w/ windowing (removing "spectral leakage")

Window Functions

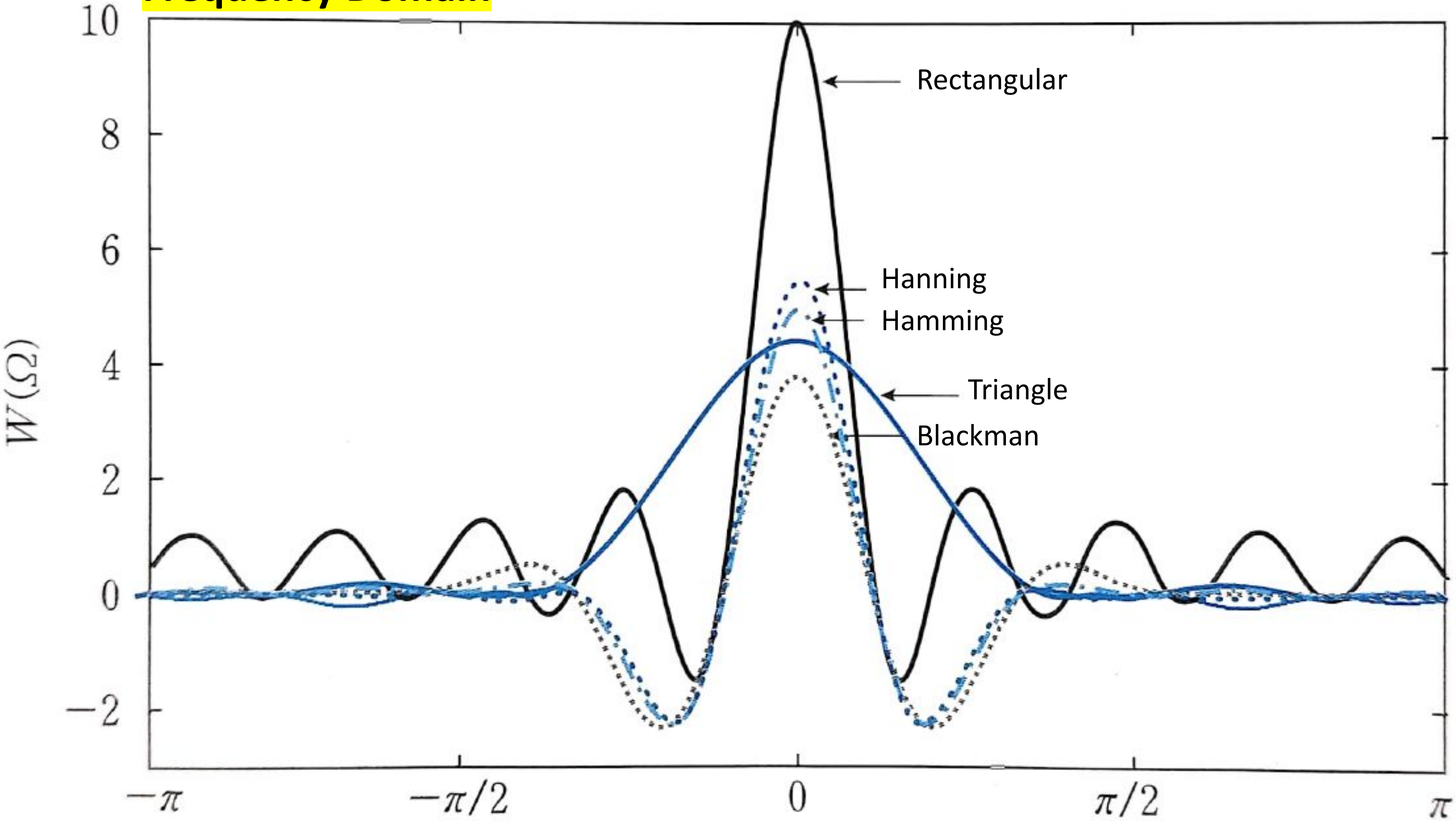
Name of Window function	*Window function* $w(n)$, $0 \le n \le N$
Rectangular	1
Hanning	$0.5 - 0.5\cos(\frac{2\pi n}{N})$
Hamming	$0.5 - 0.46\cos(\frac{2\pi n}{N})$
Blackman	$0.42 - 0.5\cos(\frac{2\pi n}{N}) + 0.08\cos(\frac{4\pi n}{N})$

Time Domain



Window Functions

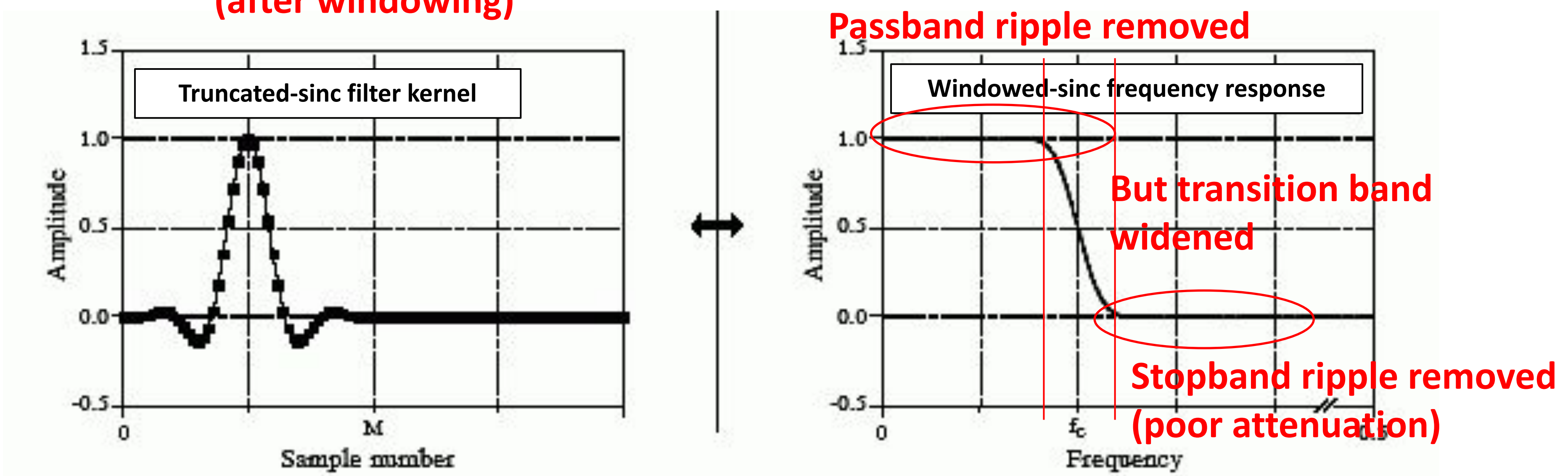
Frequency Domain



Low pass filter – Ideal Filter
Issue: Truncating infinite tails + windowing

Time Domain **Frequency Domain**

Approximation of Sinc function
(after windowing)



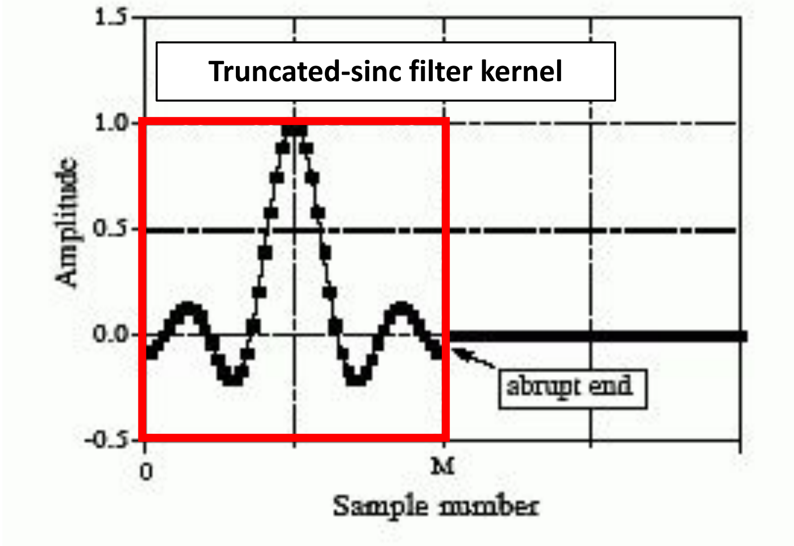
But this approach is a bit slow when there are many points

Low pass filter – Ideal Filter

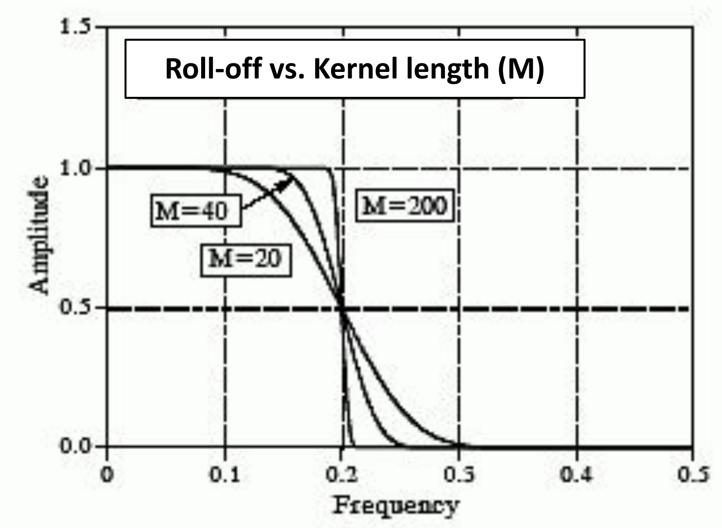
Issue: Truncating infinite tails + windowing

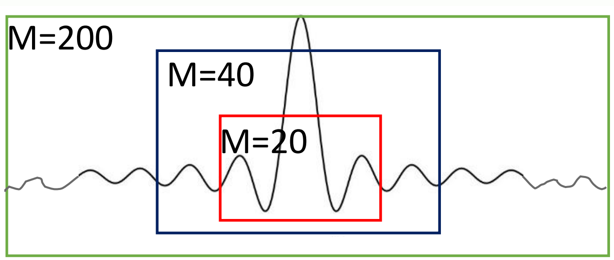
Approximation: How many samples should we retain? (=M)

Time Domain



Frequency Domain





The more the accurate!
(but takes more time!)

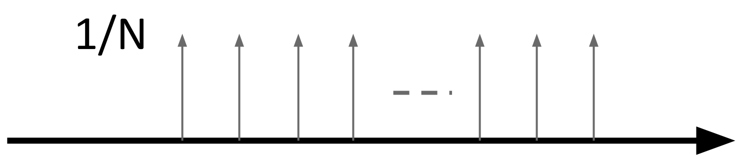
Low pass filter – Moving Average

$$y[n] = \frac{1}{N}\sum_{k=0}^{N-1} x[n-k]$$

Impulse response of moving average

$$h[n] = \frac{1}{N}\sum_{k=0}^{N-1} \delta[n-k]$$

y[n] = x[n] □ h[n]



Low pass filter – Moving Average

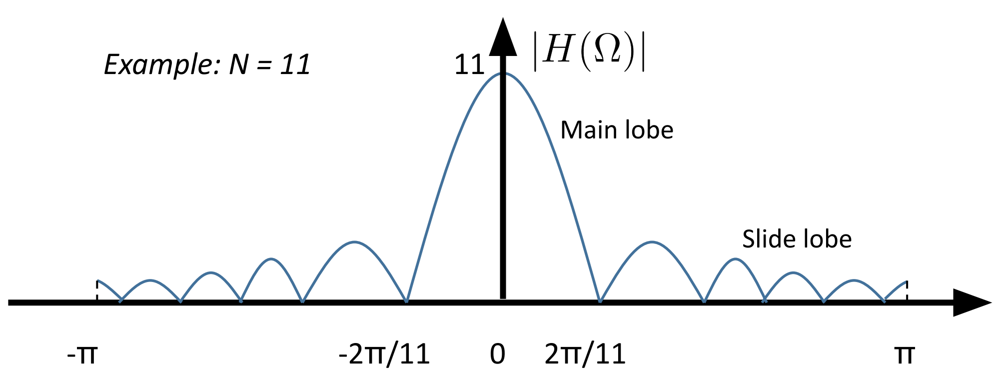
$$y[n] = \frac{1}{N}\sum_{k=0}^{N-1} x[n-k]$$

Impulse response of moving average

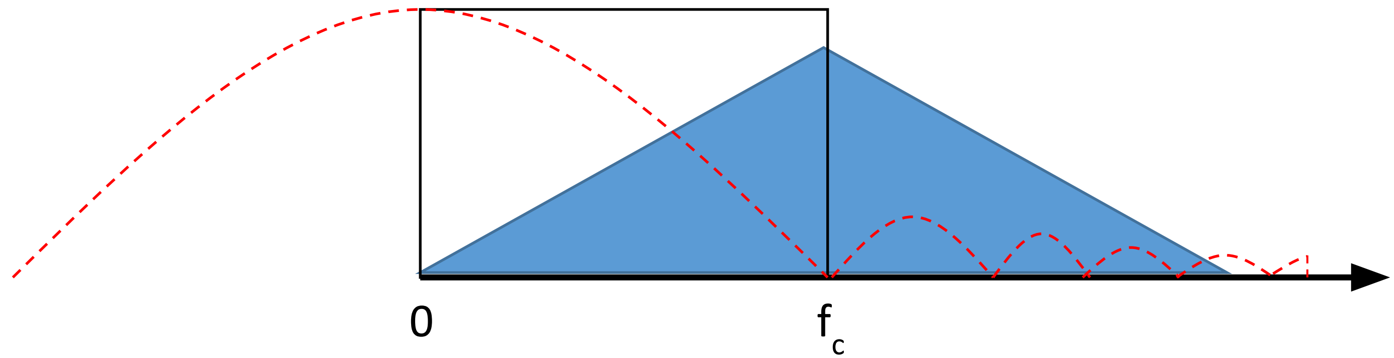
$$h[n] = \frac{1}{N}\sum_{k=0}^{N-1} \delta[n-k]$$

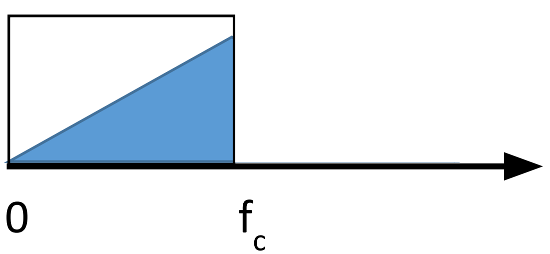
y[n] = x[n] □ h[n]

$$\Leftrightarrow \qquad H(\Omega) = \frac{1}{N}\frac{\sin(\frac{N\Omega}{2})}{\sin\frac{\Omega}{2}} e^{-j(N-1)\frac{\Omega}{2}}$$

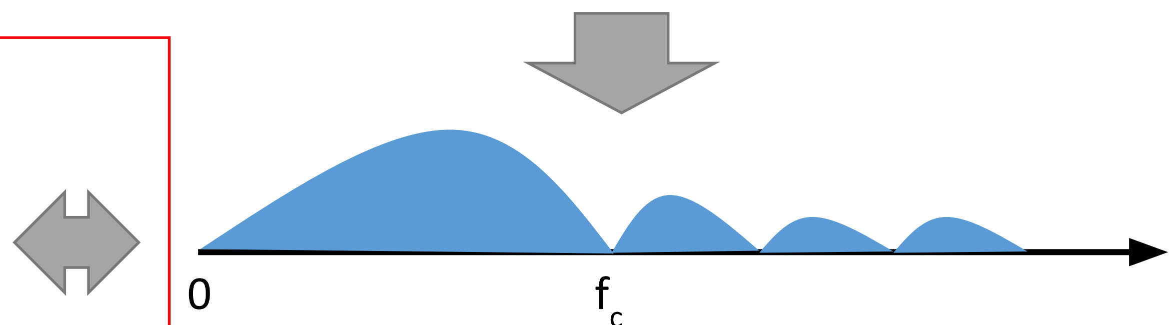


Low pass filter – Moving Average



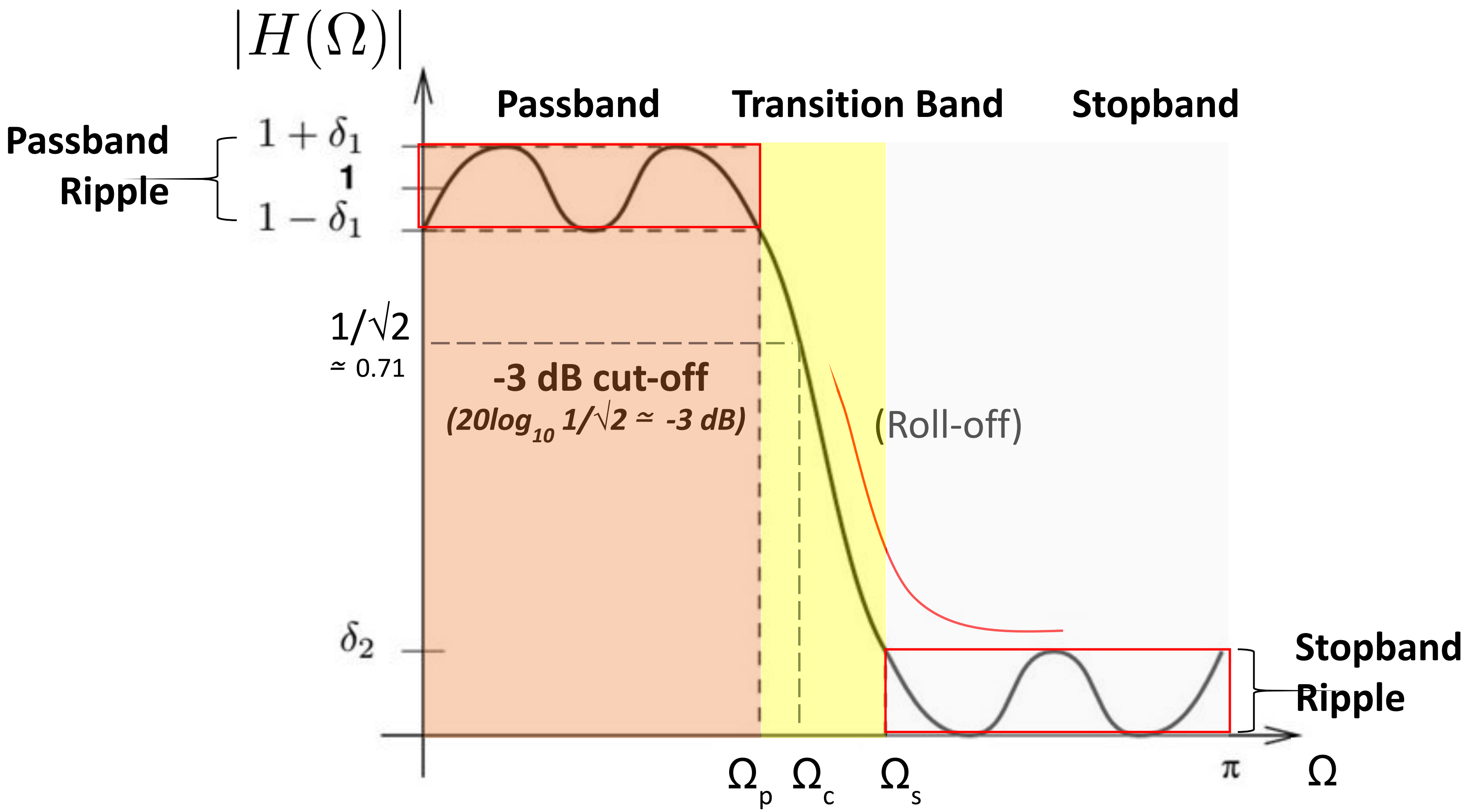


After Passing Low Pass Filter



After Moving Average Low Pass Filter

Filter Design



Ω_c is 3-dB cut-off frequency

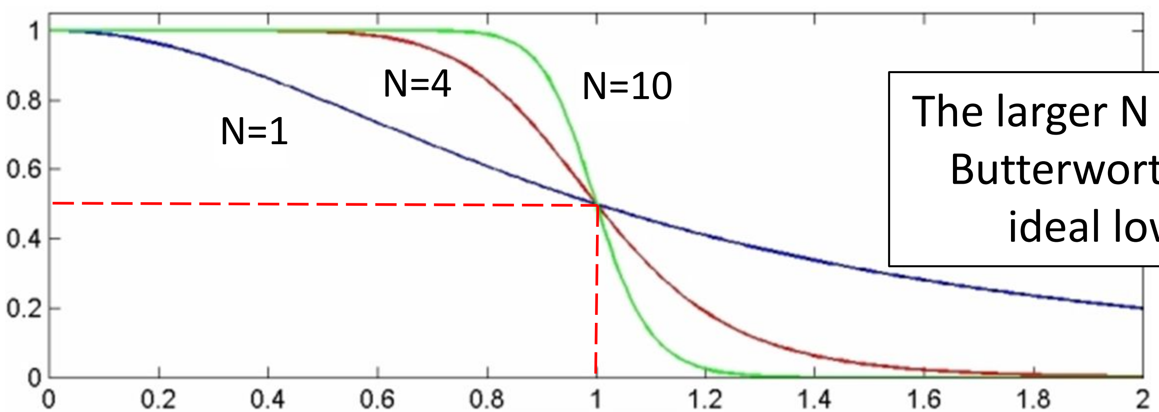
IIR Filter: Butterworth Filter

- Passband is designed to be maximally flat
- Butterworth filter of order **N**:

$$|H(\Omega)|^2 = \frac{1}{1+\left(\frac{\Omega}{\Omega_c}\right)^{2N}}$$

where Ω_c is 3 dB cut-off frequency

$|H(\Omega)|^2$



The larger N is, the closer the Butterworth filter is to an ideal low-pass filter

Ω_c Ω

IIR Filter: Chebyshev Filter

- Chebyshev filter w/ a ripple parameter of **ε** and the **N**-th order Chebyshev polynomial (C_N)

$$|H(\Omega)|^2 = \frac{1}{1+\epsilon^2 C_N^2\left(\frac{\Omega}{\Omega_c}\right)} \quad \textit{where} \quad C_N = \begin{array}{l}\cos(N\cos^{-1}x), |x| \le 1\\ \cosh(N\cosh^{-1}x), |x| > 1\end{array}$$

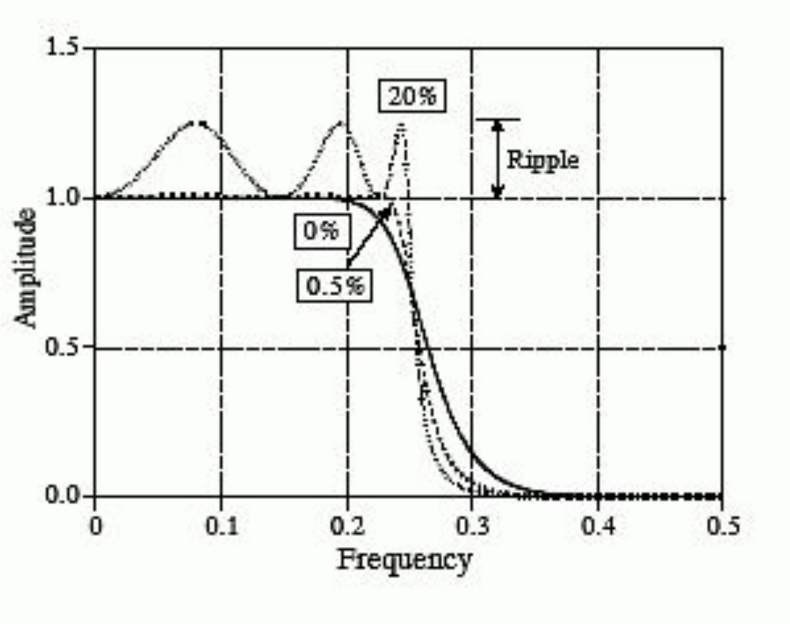


Fig 20.1 Chebyshev filters achieve a faster roll-off by allowing ripple in the passband. When the ripple is set to 0%, it's called a maximally flat or Butterworth filter. Consider using a ripple of 0.5%; this passband unflatness is so small that it cannot be seen in this graph, but the roll-off is much faster than the Butterworth

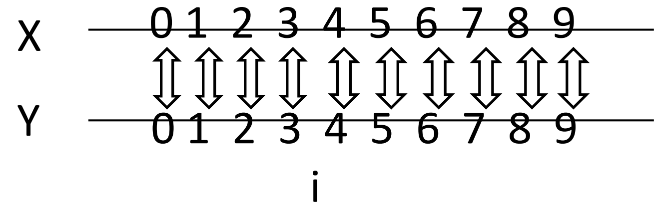
Overview

- Spectral analysis of signals
- Spectral leakage
- Digital filter basics
- FIR(Lowpass) & IIR filters
- Correlation (auto vs. cross)

Correlation

- The ***correlation*** r_{xy} between signals $x[n]$ and $y[n]$, both of duration N, is given by

$$r_{xy} = \sum_{i=0}^{N-1} x[i]y[i]$$



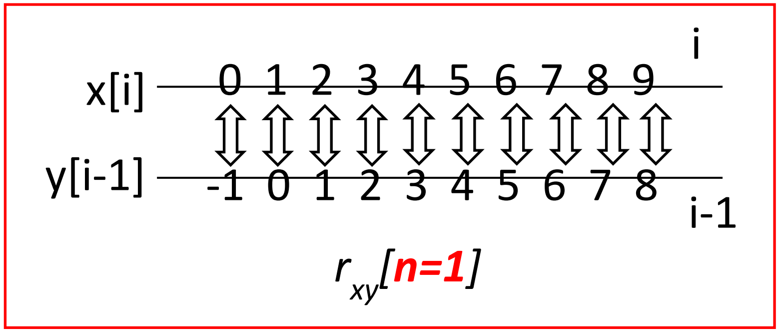
- Correlation is the inner product of two vectors: X and Y: $\mathbf{X^T Y = |X||Y| \cos \theta}$
- **cos θ** is also called as the **correlation coefficient** of x[n] and y[n], and represents an energy-normalized version of correlation
- If the two signals are initially not of the same duration, then ***zero-padding*** should be applied to make their lengths the same

Cross-correlation

- The ***cross-correlations*** $r_{xy}[n]$ and $r_{yx}[n]$ of signals $x[n]$ and $y[n]$ are defined as

$$r_{xy}[n] = \underbrace{x[n] * y[-n]}_{\text{convolution}} = \sum_{i=-\infty}^{\infty} x[i]\, y[i-n]$$

$$r_{yx}[n] = y[n] * x[-n] = \sum_{i=-\infty}^{\infty} y[i]\, x[i-n]$$



- The ***lag*** n of $r_{xy}[n]$ and $r_{yx}[n]$ represents the delay of one signal relative to the other

$$r_{xy}[0] = \ldots x[-1]\, y[-1] + x[0]\, y[0] + x[1]\, y[1] + \cdots$$

$$r_{xy}[1] = \ldots x[0]\, y[-1] + x[1]\, y[0] + x[2]\, y[1] + \cdots$$

$$r_{xy}[2] = \ldots x[0]\, y[-2] + x[1]\, y[-1] + x[2]\, y[0] + \cdots$$

Cross-correlation

- Compute the cross-correlation of $x[n] = \{3, 1, 4\}$ and $y[n] = \{2, 7, 1\}$.

Solution: The duration of r_{xy} is $N_x + N_y - 1 = 2+2-1 = 5$. The elements of $r_{xy}[n]$ are

$$
\begin{aligned}
r_{xy}[-2] &= (3)(1) = 3, \\
r_{xy}[-1] &= (3)(7) + (1)(1) = 22, \\
r_{xy}[0] &= (3)(2) + (1)(7) + (4)(1) = 17, \\
r_{xy}[1] &= (1)(2) + (4)(7) = 30, \\
r_{xy}[2] &= (4)(2) = 8.
\end{aligned}
$$

Hence,

$$r_{xy}[n] = \{3, 22, \underline{17}, 30, 8\}.$$

Cross-correlation

Time-delay estimation

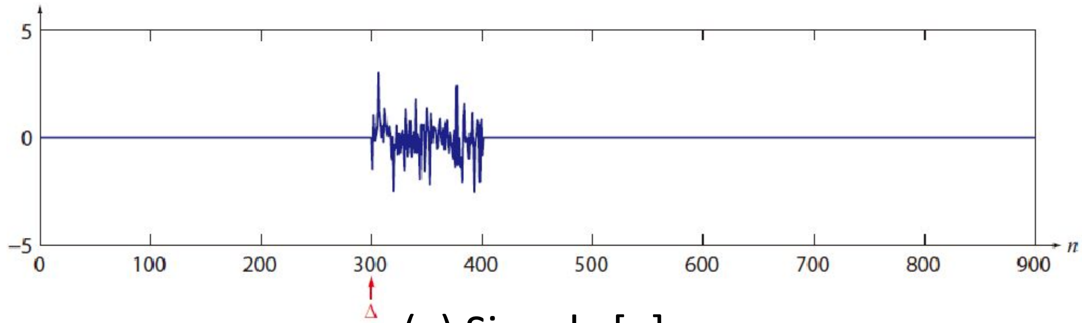
- Let $x[n]$ be a known signal with unknown time delay (assume the energy of $x[n]$ be one)
- Goal is to estimate time-delay Δ from the noisy observations $y[n] = x[n - \Delta] + w[n]$, where $w[n]$ is white Gaussian noise

$$\begin{aligned} r_{yx}[n] &= (x[n-\Delta] + w[n]) * x[-n] \\ &= \underbrace{x[n-\Delta] * x[-n]}_{r_x[n-\Delta]} + \underbrace{w[n] * x[-n]}_{r_{wx}[n]\approx 0} \\ &\approx r_x[n-\Delta] \approx \delta[n-\Delta]. \end{aligned} \tag{9.180}$$

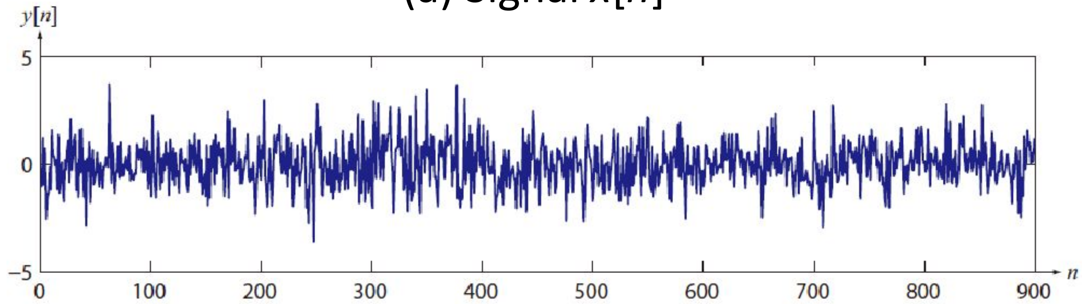
- Cross-correlation of the observations with the known signal is expected to have a peak at $n = \Delta$ (i.e., $\delta[n - \Delta]$)

Cross-correlation

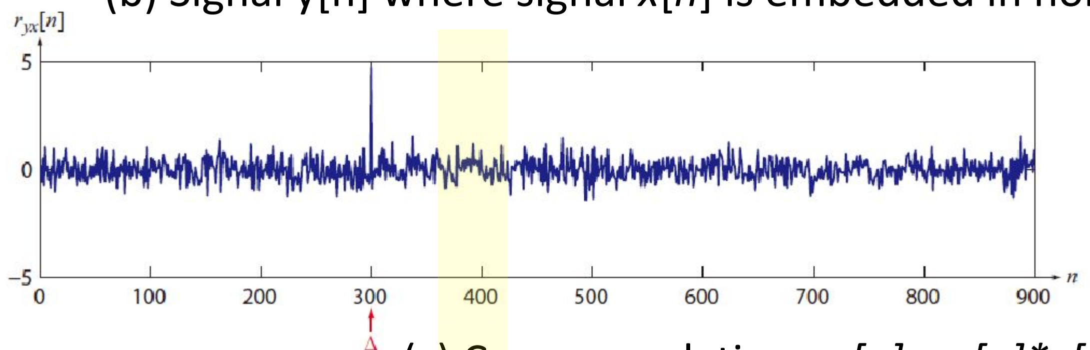
Time-delay estimation



(a) Signal *x*[*n*]



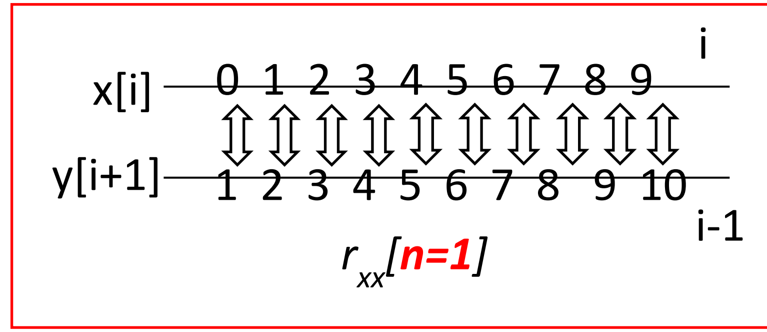
(b) Signal y[n] where signal *x*[*n*] is embedded in noise



(c) Cross-correlation $r_{yx}[n] = y[n]*x[-n]$

- Computing the cross-correlation by multiplying y[n] with many shifted versions of x[n] (i.e., n=0~900)
- The peak in the record of $r_{yx}[n]$ corresponds to the shift that generates the highest correlation, from which we deduce that the unknown delay is Δ = 300.

Autocorrelation



- Autocorrelation $r_x[n]$ of $x[n]$ is given by

$$r_x[n] = \underbrace{x[n] * x[-n]}_{\text{convolution}} = \sum_{i=-\infty}^{\infty} x[i]x[i+n]$$

- Examples:

$$r_x[0] = \cdots + (x[-1])^2 + (x[0])^2 + (x[1])^2 + \cdots$$
$$r_x[\pm 1] = \cdots + x[-1]\,x[0] + x[0]\,x[1] + x[1]\,x[2] + \cdots$$
$$r_x[\pm 2] = \cdots + x[-1]\,x[1] + x[0]\,x[2] + x[1]\,x[3] + \cdots\ .$$

Autocorrelation

- Autocorrelation $r_x[n]$ of $x[n]$ is given by

$$r_x[n] = x[n] * x[-n] = \sum_{i=-\infty}^{\infty} x[i]x[i+n]$$

- Example: Autocorrelation of x[n] = {3, 1, 4}
  - Compute $r_x[0]$, $r_x[1]$, $r_x[2]$ of x[n]

Autocorrelation

- Autocorrelation $r_x[n]$ of $x[n]$ is given by

$$r_x[n] = x[n] * x[-n] = \sum_{i=-\infty}^{\infty} x[i]x[i+n]$$

- Example: Autocorrelation of x[n] = {3, 1, 4}
  - Compute $r_x[0]$, $r_x[1]$, $r_x[2]$ of x[n]

Solution:

$$r_x[0] = 3^2 + 1^2 + 4^2 = 26,$$

$$r_x[\pm 1] = (3)(1) + (1)(4) = 7,$$

$$r_x[\pm 2] = (3)(4) = 12.$$

Autocorrelation

Using Autocorrelation to Compute Period of Noisy Signal

Suppose $x[n]$ is a signal with unknown period N. The goal is to estimate N from the noisy observations

$$y[n] = x[n] + w[n], \tag{9.173}$$

where $w[n]$ is white Gaussian noise (Section 6-10.1) with zero mean and variance σ^2.

The autocorrelation $r_y[n]$ of $y[n]$ is

$$\begin{aligned} r_y[n] &= (x[n] + w[n]) * (x[-n] + w[-n]) \\ &= x[n] * x[-n] + w[n] * w[-n] \\ &\quad + \underbrace{x[n] * w[-n]}_{r_{xw}[n] \approx 0} + \underbrace{x[-n] * w[n]}_{r_{xw}[-n] \approx 0} \\ &\approx r_x[n] + r_w[n] \approx r_x[n] + \sigma^2 \delta[n]. \end{aligned} \tag{9.174}$$

Autocorrelation

Using Autocorrelation to Compute Period of Noisy Signal

The final expression of Eq. (9.174) ignores the terms involving correlations between $x[n]$ and $w[n]$ because $w[n]$ is zero-mean uncorrelated noise.

Since the signal period is N,

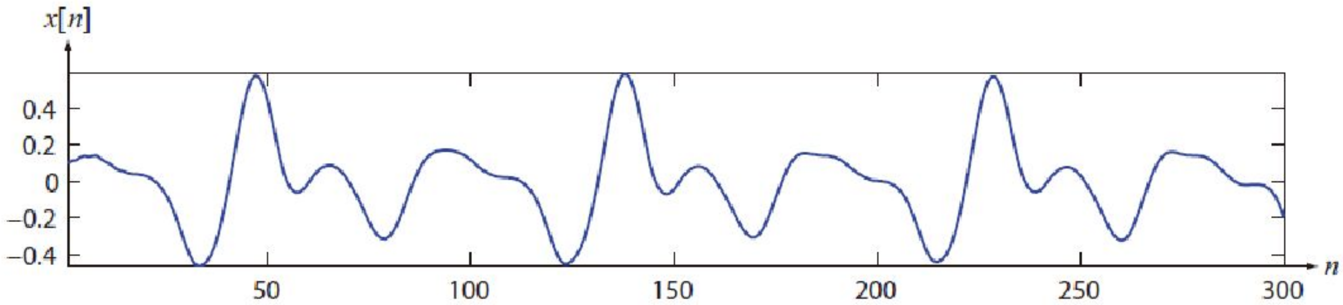
$$x[n] = x[n + N],$$

and the autocorrelations at lags N, $2N$, etc., are

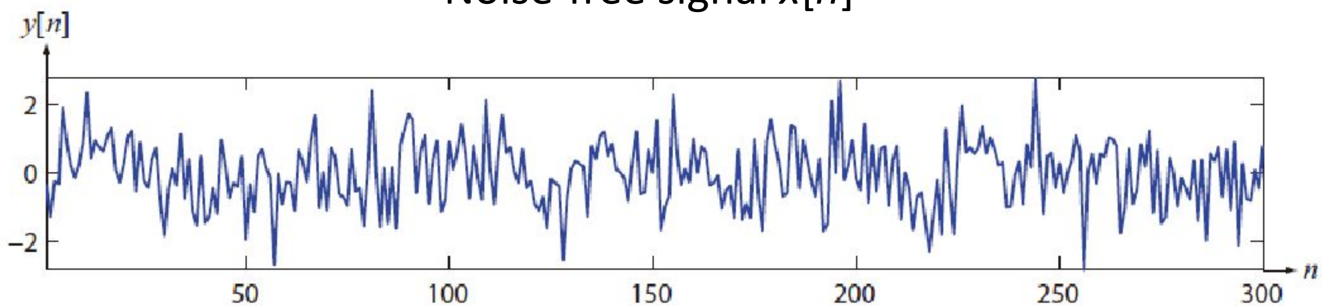
$$\begin{aligned}
r_x[N] &= \sum x[i]\, x[i + N] = \sum x[i]\, x[i] = r_x[0], \\
r_x[2N] &= \sum x[i]\, x[i + 2N] = \sum x[i]\, x[i] = r_x[0], \\
&\vdots \quad \vdots
\end{aligned} \tag{9.175}$$

which leads to the conclusion that $r_x[n]$ is periodic with period N. So, to a good approximation,

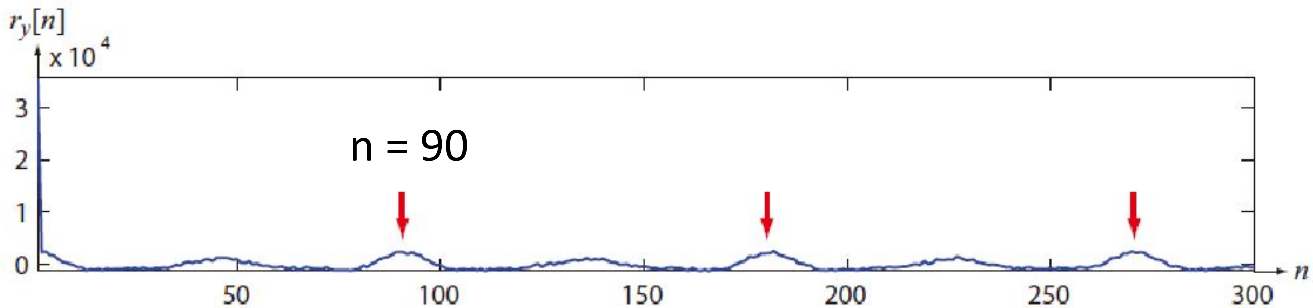
$$r_y[n] \approx \begin{cases} \sigma^2 + \sum x^2[i] & \text{for } n = 0, \\ \sum x^2[i] & \text{for } n = N, \\ \sum x^2[i] & \text{for } n = 2N, \\ \vdots & \quad \vdots \\ 0 & \text{otherwise.} \end{cases} \tag{9.176}$$



Noise-free signal $x[n]$

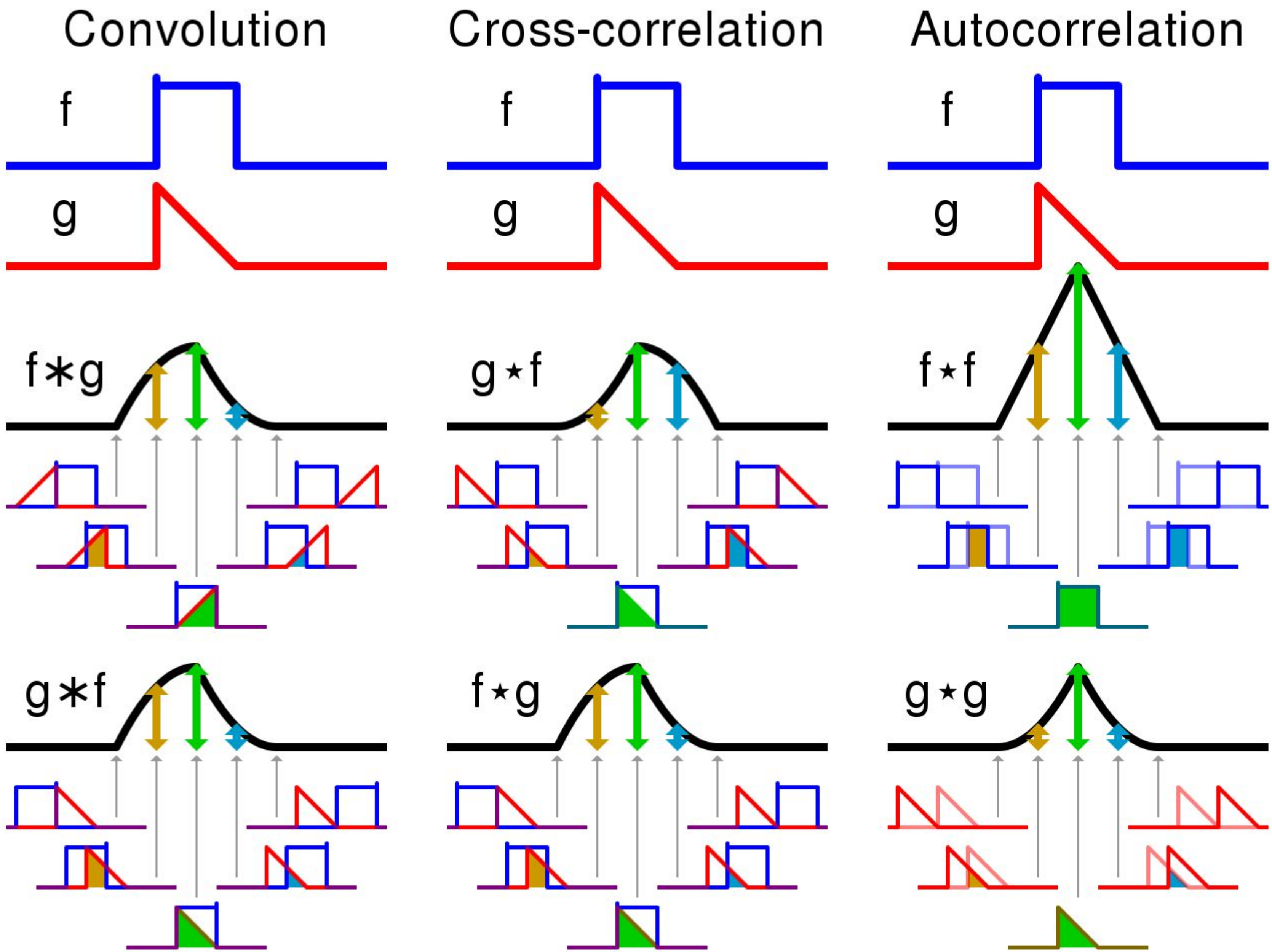


Noisy signal $y[n]$



Autocorrelation $r_y[n]$

Estimating the period of a trumpet signal Ignoring the peak at n = 0, the next highest peak is at n = 90 and multiples thereof. The original signal had been sampled at 44100 samples/s. Hence, the frequency of the signal corresponding to the peak at n = 90 is given by f = fs/90 = 44100/90 = 490 Hz

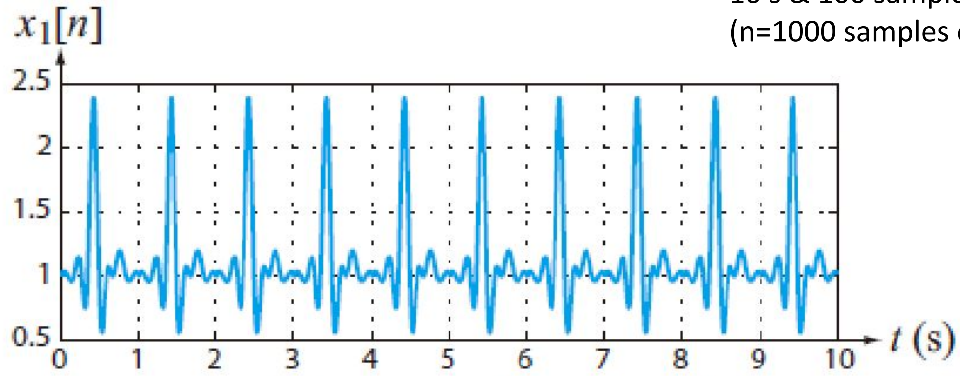


Convolution
Cross-correlation
Autocorrelation
f
g
f∗g
g∗f
g⋆f
f⋆g
f⋆f
g⋆g

EKG Example: Measuring Heart Rate

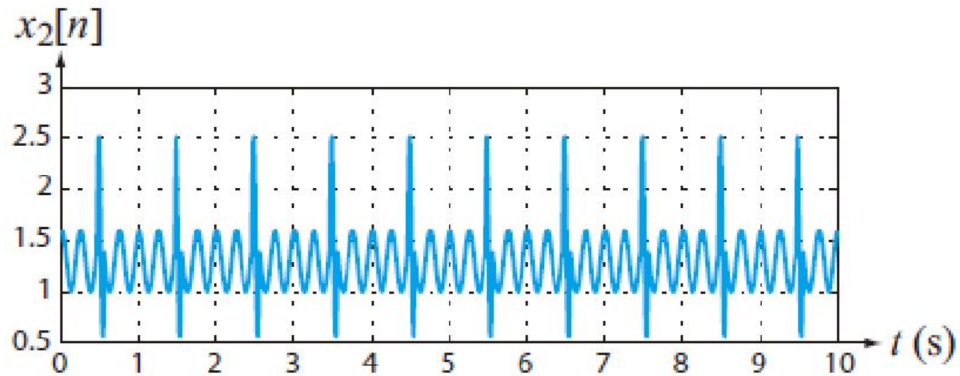
10 s & 100 samples/s
(n=1000 samples each)

Normal heart waveform consists of regular pulses at 1 pulse per second, separated by minor fluctuations



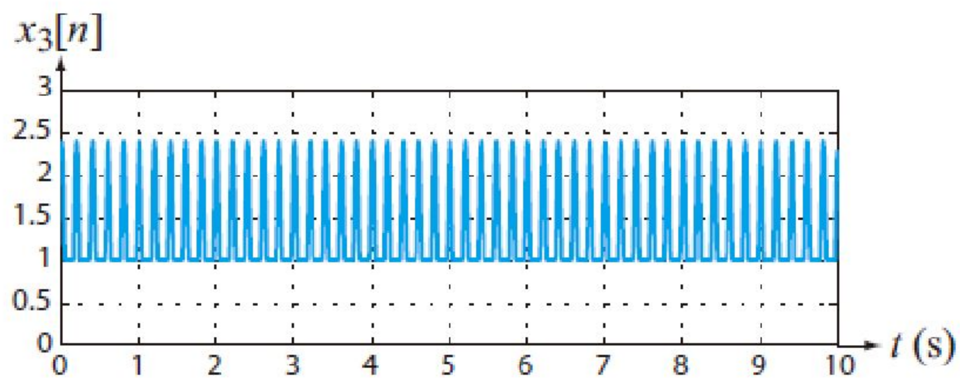
(a) EKG of normal cardiac activity

The flutter waveform contains three additional bumps (pulses) in the waveform. The spectrum of the flutter waveform will be dominated by a component at 240 beats per minute or, equivalently, 4 Hz



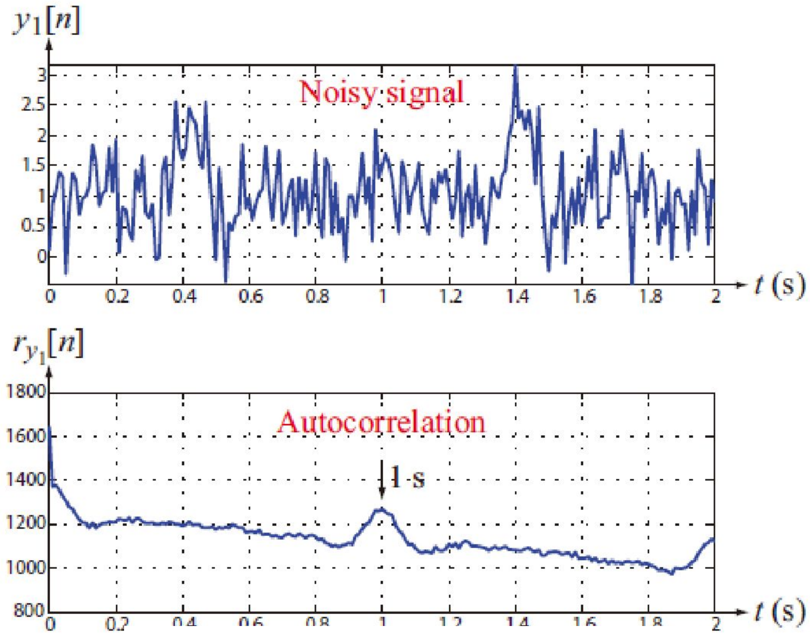
(b) EKG of heart with atrial flutter

The fibrillation waveform is Fig. 9-38(c) consists entirely of equal-amplitude bumps at 300 beats per minute, corresponding to a spectral line at 5 Hz. Atrial fibrillation is a serious heart condition requiring a pacemaker to correct its behavior into normal cardiac action.

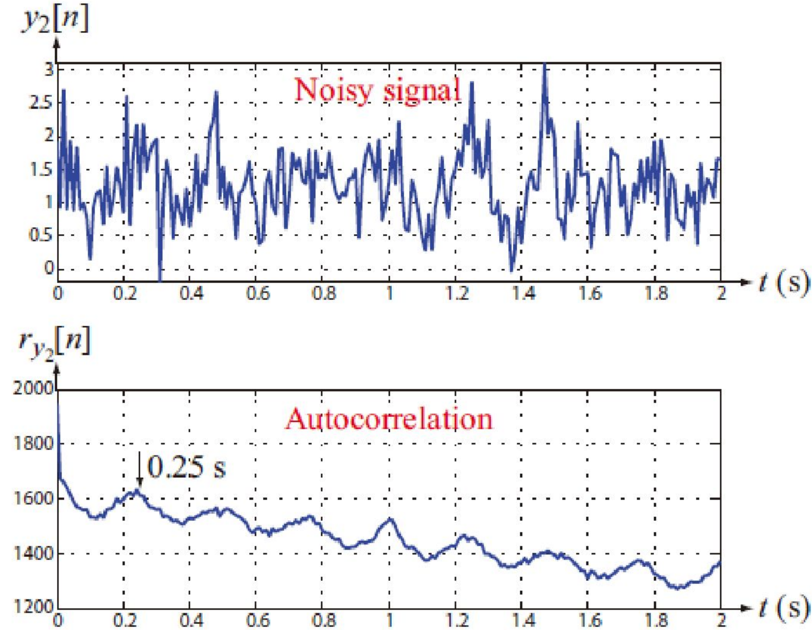


(c) EKG of heart with atrial fibrillation

EKG Example: Measuring Heart Rate

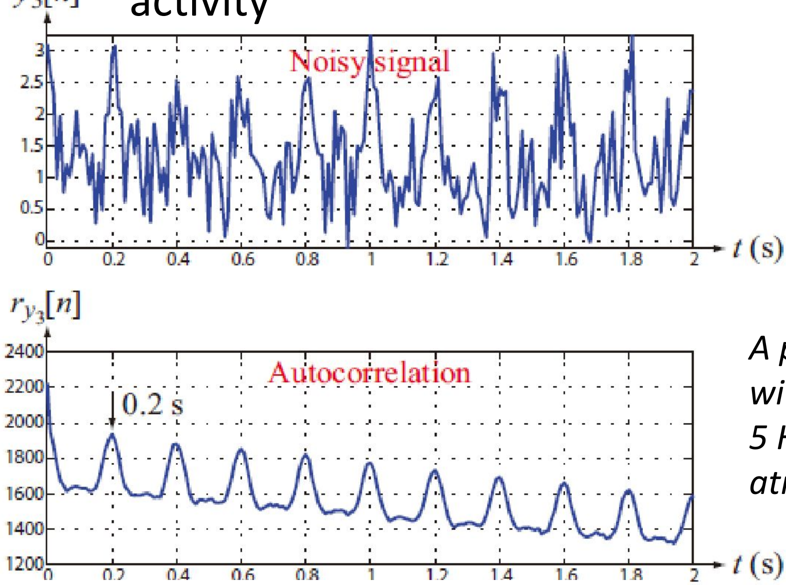


EKG of normal cardiac activity



EKG of heart with atrial flutter

A peak at 0.25 s (and its multiples), with a corresponding frequency of 4 Hz, for the heart with atrial flutter.

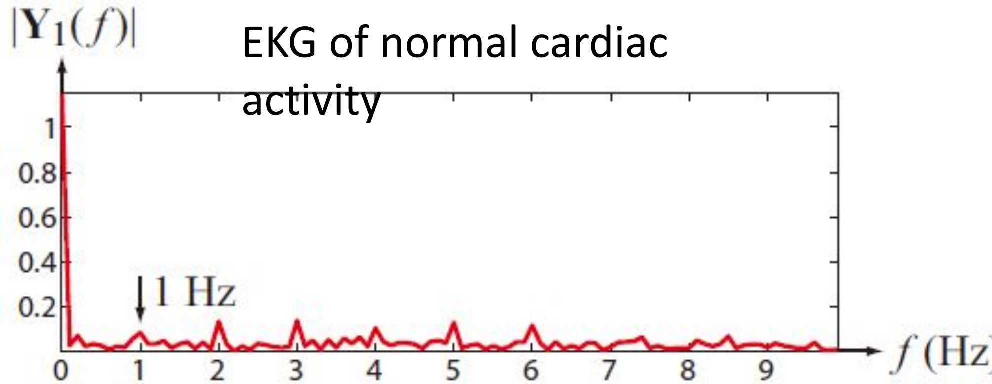


A peak at 0.2 s (and its multiples) with a corresponding frequency of 5 Hz, for the heart afflicted with atrial fibrillation.

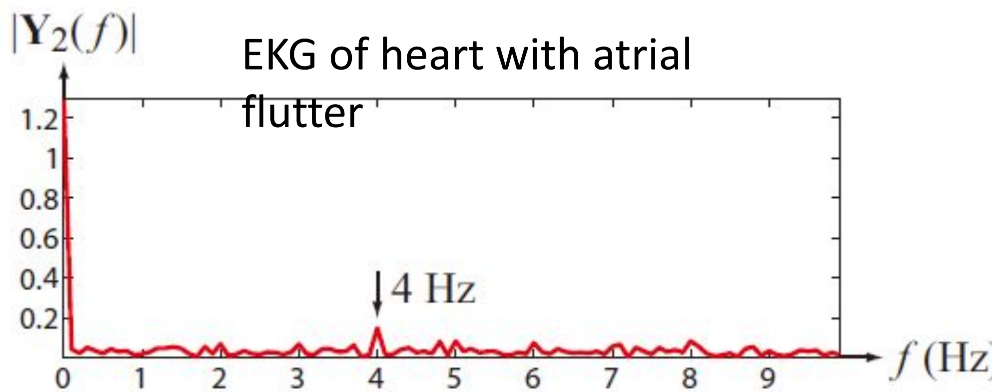
EKG of heart with atrial fibrillation

Spectra of noisy EKG waveforms

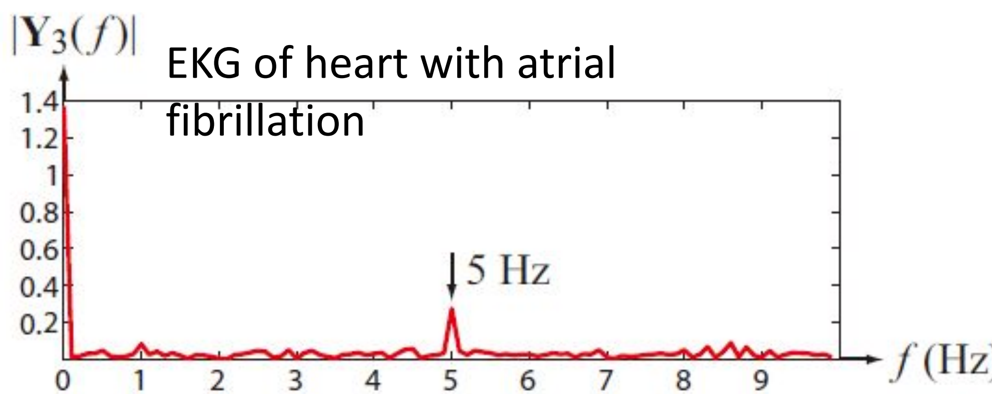
EKG of normal cardiac activity



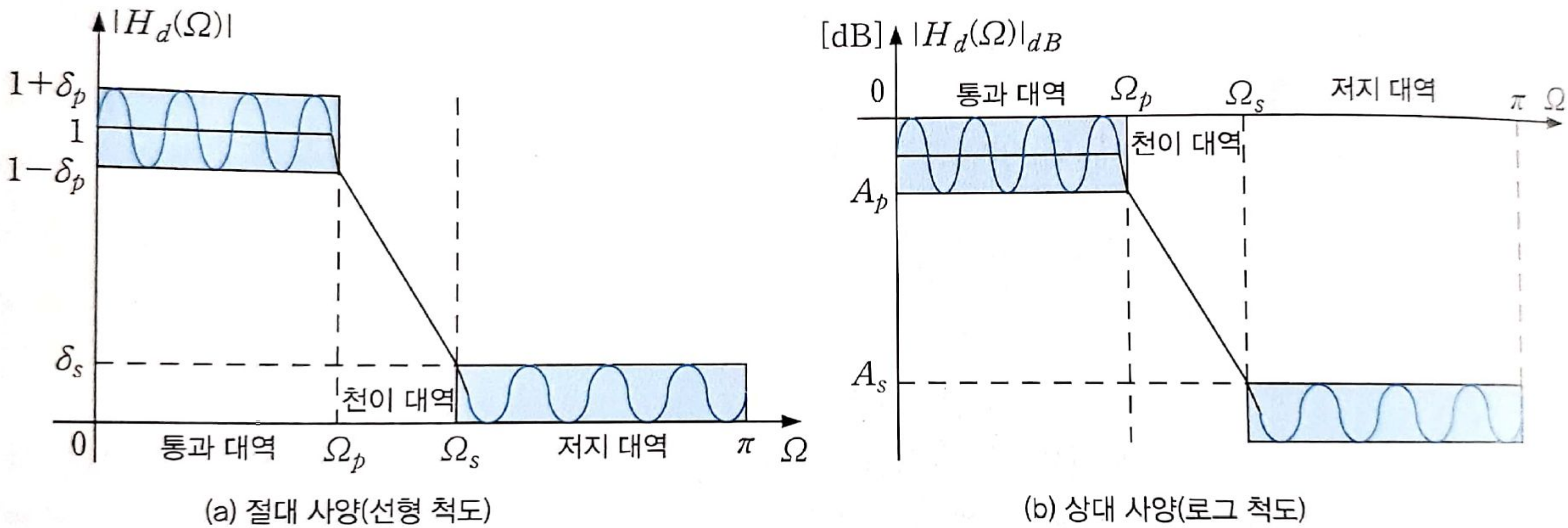
EKG of heart with atrial flutter



EKG of heart with atrial fibrillation



$$|H(\Omega)|_{dB} = 20\log_{10}|H(\Omega)|\,[\mathrm{dB}]$$



[그림 9-15] 저역 통과 필터의 진폭 응답 사양

$$A_p = -20\log_{10}\frac{1-\delta_p}{1+\delta_p}$$

$$A_s = -20\log_{10}\frac{\delta_s}{1+\delta_p}$$