http://elec3004.com

# Digital Filters: Windows & FIR

ELEC 3004: **Systems**: Signals & Controls

Dr. Surya Singh

Lecture 13

elec3004@itee.uq.edu.au

http://robotics.itee.uq.edu.au/~elec3004/

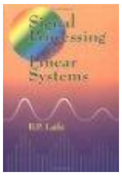
April 10, 2019



## Lecture Schedule:

| Week | Date | Lecture Title |
|---|---|---|
| 1 | 27-Feb | Introduction |
| | 1-Mar | Systems Overview |
| 2 | 6-Mar | Systems as Maps & Signals as Vectors |
| | 8-Mar | Systems: Linear Differential Systems |
| 3 | 13-Mar | Sampling Theory & Data Acquisition |
| | 15-Mar | Aliasing & Antialiasing |
| 4 | 20-Mar | Discrete Time Analysis & Z-Transform |
| | 22-Mar | Second Order LTID (& Convolution Review) |
| 5 | 27-Mar | Frequency Response |
| | 29-Mar | Filter Analysis |
| 6 | 3-Apr | Digital Filters (IIR) & Filter Analysis |
| | 5-Apr | PS 1: Q & A |
| **7** | **10-Apr** | **Digital Filter (FIR) & Digital Windows** |
| | 12-Apr | FFT |
| 8 | 17-Apr | Active Filters & Estimation |
| | 19-Apr | Holiday |
| | 24-Apr | |
| | 26-Apr | |
| 9 | 1-May | Introduction to Feedback Control |
| | 3-May | Servoregulation/PID |
| 10 | 8-May | PID & State-Space |
| | 10-May | State-Space Control |
| 11 | 15-May | Digital Control Design |
| | 17-May | Stability |
| 12 | 22-May | State Space Control System Design |
| | 24-May | Shaping the Dynamic Response |
| 13 | 29-May | System Identification & Information Theory |
| | 31-May | Summary and Course Review |

# Follow Along Reading:



**B. P. Lathi**
***Signal processing and linear systems***
**1998**
**TK5102.9.L38 1998**

**Today**

- Chapter 4
  - § 4.9 Data Truncation: Window Functions
- Chapter 12
  (**Frequency Response and Digital Filters**)
  - § 12.1 Frequency Response of Discrete-Time Systems
  - § 12.3 Digital Filters
  - § 12.4 Filter Design Criteria
  - § 12.7 Nonrecursive Filters

- Chapter 10
  (**Discrete-Time System Analysis Using the *z*-Transform**)
  - § 10.3 Properties of DTFT
  - § 10.5 Discrete-Time Linear System analysis by DTFT
  - § 10.7 Generalization of DTFT to the $\mathcal{Z}$–Transform
  - One of the days! ☺

***Next Time***



# Announcement: LaTeX Copy/Paste

Question 5

(a) *deriving the equation of the circuit would give us*

$\omega^2 < \alpha^2$ *then by deriving it again and dividing by L, we have our 2nd order ODE*

$\omega^2 < \alpha^2$ *it is linear and causal because it's values will only change with respect to time and does not look into the future for other values.*

(b) *the oscillating frequency is a standard formula given as*

$\omega^2 < \alpha^2$

(c) *by simply deriving the first derivative of the equation and not dividing it by L we get:*

$\omega^2 < \alpha^2$ *substituting q and getting the auxiliary equation*

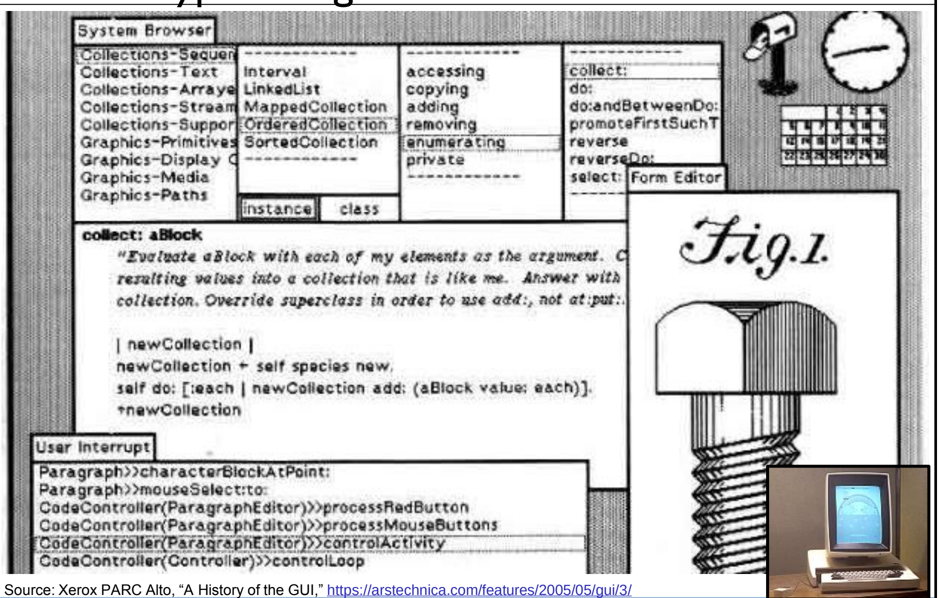
$\omega^2 < \alpha^2$ *getting the roots by quadratic formula* $\omega^2 < \alpha^2$

- When using external tools, be sure to copy the LaTeX not the image (because it might change)
- In this case, the “image” is a web-link which has expired!
  - https://www.latex4technics.com/l4ttemp/ysio4z.png?1458878525541

# Digital Windows!



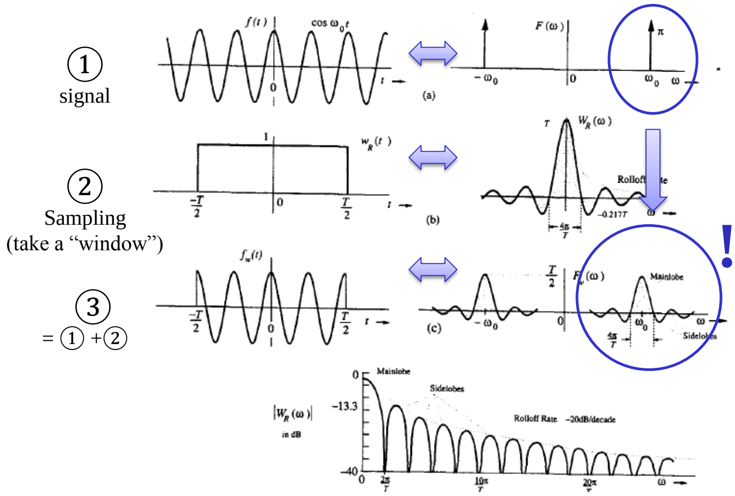
# Not this type of Digital Windows




Source: Xerox PARC Alto, "A History of the GUI," https://arstechnica.com/features/2005/05/gui/3/

# **Recall:** Windowing for the DFT

$$f_w(t) = f(t)w(t) \quad \text{and} \quad F_w(\omega) = \frac{1}{2\pi}F(\omega) * W(\omega)$$



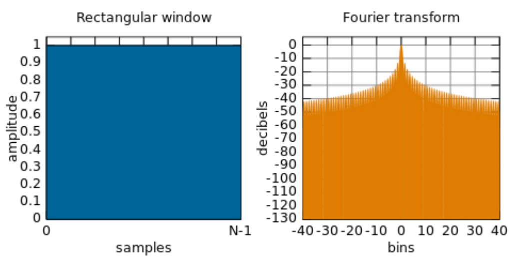

Source: Lathi, p.303



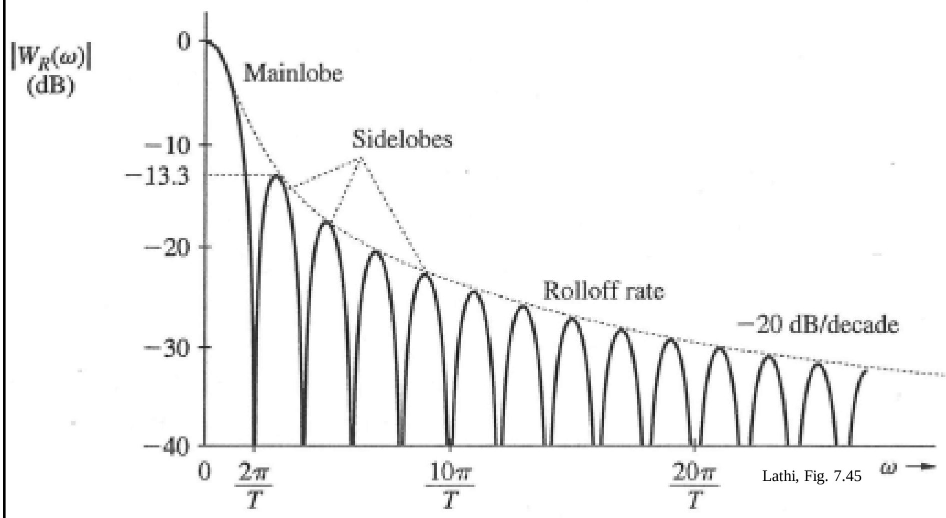
# Think Inside The Box: The "Rect" **Window** Functions

Rectangular

$$w(n) = 1$$



## Windowing and its effects/terminology




Lathi, Fig. 7.45



## Truncation ➔ Window Functions (Lathi 4.9)

- We often need to **truncate data**
  - Ex: Fourier transform of some signal, say $e^{-t}u(t)$
  - Truncate beyond a sufficiently large value of t (typically five time constants and above).
  - ∵ in numerical computations: we have data of finite duration.
  - For example: the impulse response *h(t)* of an ideal lowpass filter is noncausal, and approaches zero asymptotically as $|t| \to \infty$

- Data truncation can occur in **both** <u>time</u> and <u>frequency domain</u>
  - In signal sampling, to eliminate aliasing, we need to truncate the Signal spectrum beyond the half sampling frequency $\frac{\omega_s}{2}$, using an anti-aliasing filter

## Truncation ➔ Window Functions (Lathi 4.9) [2]

Truncation operation may be regarded as multiplying a signal of a large width by a window function of a smaller (finite) width. Simple truncation amounts to using a **rectangular window** $w_R(t)$ (Fig. 4.48a) in which we assign unit weight to all the data within the window width ($|t| < \frac{T}{2}$), and assign zero weight to all the data lying outside the window ($|t| > \frac{T}{2}$). It is also possible to use a window in which the weight assigned to the data within the window may not be constant. In a **triangular window** $w_T(t)$, for example, the weight assigned to data decreases linearly over the window width (Fig. 4.48b).

Consider a signal $f(t)$ and a window function $w(t)$. If $f(t) \iff F(\omega)$ and $w(t) \iff W(\omega)$, and if the windowed function $f_w(t) \iff F_w(\omega)$, then

$$f_w(t) = f(t)w(t) \qquad \text{and} \qquad F_w(\omega) = \frac{1}{2\pi}F(\omega) * W(\omega)$$



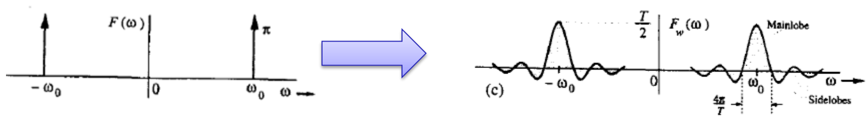
## Window Functions [1]

$$f_w(t) = f(t)w(t) \qquad \text{and} \qquad F_w(\omega) = \frac{1}{2\pi}F(\omega) * W(\omega)$$

According to the width property of convolution, it follows that the width of $F_w(\omega)$ equals the sum of the widths of $F(\omega)$ and $W(\omega)$. Thus, truncation of a signal increases its bandwidth by the amount of bandwidth of $w(t)$. Clearly, the truncation of a signal causes its spectrum to spread (or smear) by the amount of the bandwidth of $w(t)$. Recall that the signal bandwidth is inversely proportional to the signal duration (width). Hence, the wider the window, the smaller is its bandwidth, and the smaller is the **spectral spreading**. This result is predictable because a wider window means we are accepting more data (closer approximation), which should cause smaller distortion (smaller spectral spreading). Smaller window width (poorer approximation) causes more spectral spreading (more distortion). There are also other effects produced by the fact that $W(\omega)$ is really not strictly bandlimited, and its spectrum $\to 0$ only asymptotically. This causes the spectrum of $F_w(\omega) \to 0$ asymptotically also at the same rate as that of $W(\omega)$, even though the $F(\omega)$ may be strictly bandlimited. Thus, windowing causes the spectrum of $F(\omega)$ to leak in the band where it is supposed to be zero. This effect is called **leakage**. These twin effects, the spectral spreading and the leakage, will now be clarified by an example.

# Window Functions [2]

For an example, let us take $f(t) = \cos \omega_0 t$ and a rectangular window $w_R(t) = \text{rect}(\frac{t}{T})$, illustrated in Fig. 4.46b. The reason for selecting a sinusoid for $f(t)$ is that its spectrum consists of spectral lines of zero width (Fig. 4.46a). This choice will make the effect of spectral spreading and leakage clearly visible. The spectrum of the truncated signal $f_w(t)$ is the convolution of the two impulses of $F(\omega)$ with the sinc spectrum of the window function. Because the convolution of any function with an impulse is the function itself (shifted at the location of the impulse), the resulting spectrum of the truncated signal is ($1/2\pi$ times) the two sinc pulses at $\pm\omega_0$, as depicted in Fig. 4.46c. Comparison of spectra $F(\omega)$ and $F_w(\omega)$ reveals the effects of truncation. These are:






# Window Functions [3]

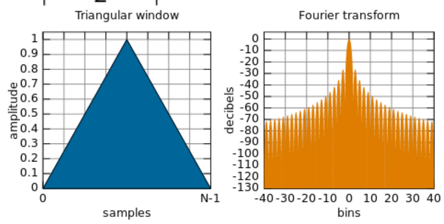
1 The spectral lines of $F(\omega)$ have zero width. But the truncated signal is spread out by $4\pi/T$ about each spectral line. The amount of spread is equal to the width of the mainlobe of the window spectrum. One effect of this **spectral spreading** (or smearing) is that if $f(t)$ has two spectral components of frequencies differing by less than $4\pi/T$ rad/s ($2/T$ Hz), they will be indistinguishable in the truncated signal. The result is loss of spectral resolution. We would like the spectral spreading (mainlobe width) to be as small as possible.

2 In addition to the mainlobe spreading, the truncated signal also has sidelobes, which decay slowly with frequency. The spectrum of $f(t)$ is zero everywhere except at $\pm\omega_0$. On the other hand, the truncated signal spectrum $F_w(\omega)$ is zero nowhere because of sidelobes. These sidelobes decay asymptotically as $1/\omega$. Thus, the truncation causes spectral **leakage** in the band where the spectrum of the signal $f(t)$ is zero. The peak **sidelobe** magnitude is 0.217 times the mainlobe magnitude (13.3 dB below the peak mainlobe magnitude). Also, the sidelobes decay at a rate $1/\omega$, which is $-6$ dB/octave (or $-20$ dB/decade). This is the **rolloff rate** of sidelobes. We want smaller sidelobes with a faster rate of decay (high rolloff rate). Figure 4.46d shows $|W_R(\omega)|$ (in dB) as a function of $\omega$. This plot clearly shows the mainlobe and sidelobe features, with the first sidelobe amplitude $-13.3$ dB below the mainlobe amplitude, and the sidelobes decaying at a rate of $-6$ dB/octave (or $-20$ dB per decade).

## **Other than Rect:** Some More Window Functions …

2. Triangular window

$$w(n) = 1 - \left| \frac{n - \frac{N-1}{2}}{\frac{N+1}{2}} \right|$$




- And Bartlett Windows
  - A slightly narrower variant with zero weight at both ends:

$$w(n) = 1 - \left| \frac{n - \frac{N-1}{2}}{\frac{N-1}{2}} \right|$$



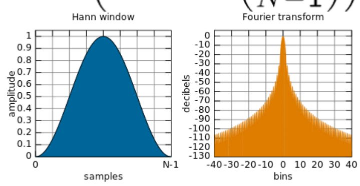
## Some More Window Functions…

3. Generalized Hamming Windows

$$w(n) = \alpha - \beta \cos\left(\frac{2\pi n}{N-1}\right)$$

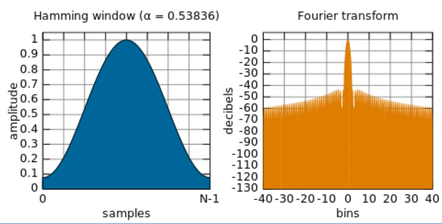
→Hanning Window

$$\rightarrow w(n) = 0.5\left(1 - \cos\left(\frac{2\pi n}{N-1}\right)\right)$$




→Hamming's Window

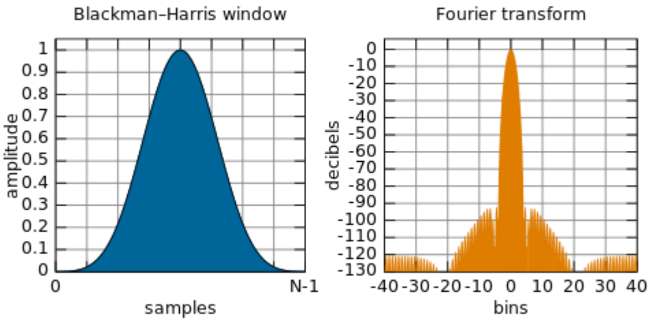
$$\rightarrow \alpha = 0.54,\ \beta = 1 - \alpha = 0.46$$



## Some More Window Functions…

4. Blackman–Harris Windows
   - A generalization of the Hamming family,
   - Adds more shifted sinc functions for less side-lobe levels

$$w(n) = a_0 - a_1 \cos\left(\frac{2\pi n}{N-1}\right) + a_2 \cos\left(\frac{4\pi n}{N-1}\right) - a_3 \cos\left(\frac{6\pi n}{N-1}\right)$$



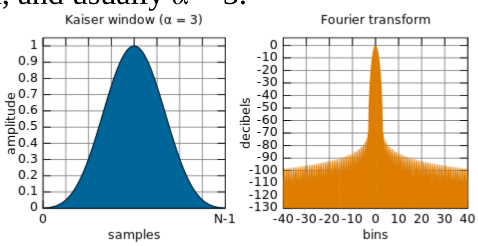



## Some More Window Functions…

5. Kaiser window
   - A DPSS (discrete prolate spheroidal sequence)
   - Maximize the energy concentration in the main lobe

$$\rightarrow w(n) = \frac{I_0\left(\pi\alpha\sqrt{1-\left(\frac{2n}{N-1}-1\right)^2}\right)}{I_0(\pi\alpha)}$$

   - Where: $I_0$ is the zero-th order modified Bessel function of the first kind, and usually $\alpha = 3$.



## Together: Remedies for Side Effects of Truncation

For better results, we must try to minimize the truncation's twin side effects, the spectral spreading (mainlobe width) and leakage (sidelobe). Let us consider each of these ills.

1. The spectral spread (mainlobe width) of the truncated signal is equal to the bandwidth of the window function $w(t)$. We know that the signal bandwidth is inversely proportional to the signal width (duration). Hence, to reduce the spectral spread (mainlobe width), we need to increase the window width.
2. To improve the leakage behavior, we must search for the cause of the slow decay of sidelobes. In Chapter 3, we saw that the Fourier spectrum decays as $1/\omega$ for a signal with jump discontinuity, and decays as $1/\omega^2$ for a continuous signal whose first derivative is discontinuous, and so on.† Smoothness of a signal is measured by the number of continuous derivatives it possesses. The smoother the signal, the faster the decay of its spectrum. Thus, we can achieve a given leakage behavior by selecting a suitably smooth window.
3. For a given window width, the remedies for the two effects are incompatible. If we try to improve one, the other deteriorates. For instance, among all the windows of a given width, the rectangular window has the smallest spectral spread (mainlobe width), but has high level sidelobes, which decay slowly. A tapered (smooth) window of the same width has smaller and faster decaying sidelobes, but it has a wider mainlobe.‡ But we can compensate for the increased mainlobe width by widening the window. Thus, we can remedy both the side effects of truncation by selecting a suitably smooth window of sufficient width.



## Remedies for Side Effects of Truncation

There are several well-known tapered-window functions, such as Bartlett (triangular), Hanning (von Hann), Hamming, Blackman, and Kaiser, which truncate the data gradually. These windows offer different tradeoffs with respect to spectral spread (mainlobe width), the peak sidelobe magnitude, and the leakage rolloff rate as indicated in Table 4.3.[8,9] Observe that all windows are symmetrical about the origin (even functions of $t$). Because of this feature, $W(\omega)$ is a real function of $\omega$; that is, $\angle W(\omega)$ is either 0 or $\pi$. Hence, the phase function of the truncated signal has a minimal amount of distortion.

Figure 4.47 shows two well-known tapered-window functions, the von Hann (or Hanning) window $w_{\text{HAN}}(x)$ and the Hamming window $w_{\text{HAM}}(x)$. We have intentionally used the independent variable $x$ because windowing can be performed in time domain as well as in frequency domain; so $x$ could be $t$ or $\omega$, depending on the application.

# Remedies for Side Effects of Truncation

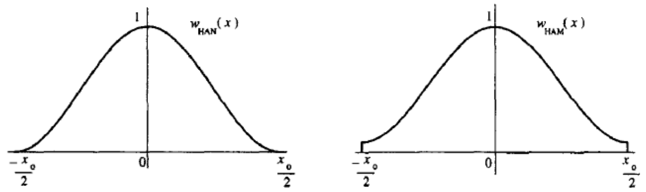
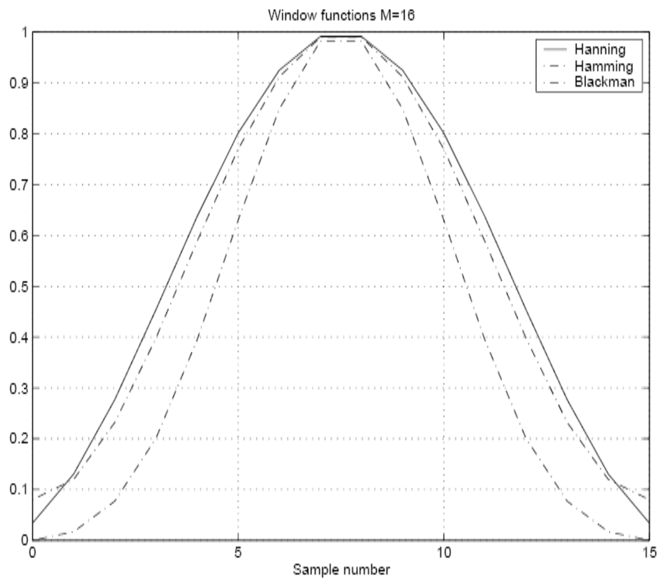



**Fig. 4.47** Hanning and Hamming windows.

There are hundreds of windows, each with differing characteristics. But the choice depends on a particular application. The rectangular window has the narrowest mainlobe. The Bartlett (triangle) window (also called the Fejer or Cesaro) is inferior in all respects to the Hanning window. For this reason it is rarely used in practice. Hanning is preferred over Hamming in spectral analysis because it has faster sidelobe decay. For filtering applications, on the other hand, the Hamming window is the choice because it has the smallest sidelobe magnitude for a given mainlobe width. The Hamming window is the most widely used, general purpose window. The Kaiser window, which uses $I_0(\alpha)$, the Bessel function of the order 0, is more versatile and adjustable. Selecting a proper value of $\alpha$ $(0 \leq \alpha \leq 10)$ allows the designer to tailor the window to suit a particular application. The parameter $\alpha$ controls the mainlobe and sidelobe trade-off. When $\alpha = 0$, the Kaiser window is the rectangular window. For $\alpha = 5.4414$, it is the Hamming window, and when $\alpha = 8.885$, it is the Blackman window. As $\alpha$ increases, the mainlobe width increases and the sidelobe level decreases.

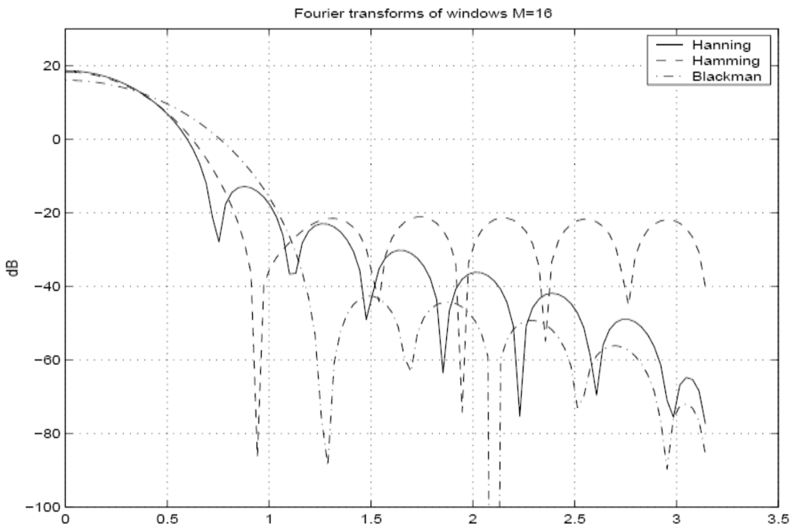


# Comparison of Alternative Windows –Time Domain




Punskaya, Slide 90

# Comparison of Alternative Windows
## Frequency Domain



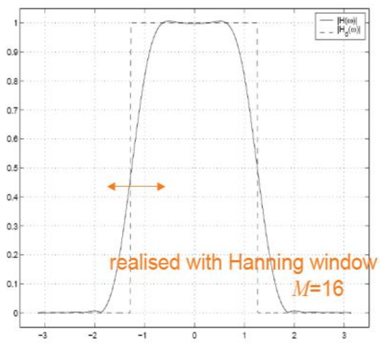

Punskaya, Slide 91

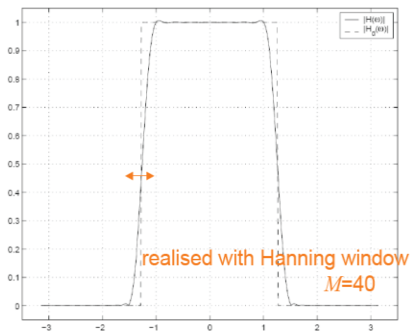


# Adding Order

\+ Transition and Smoothness

– Increased Size




Punskaya, Slide 94

## Summary Characteristics of Common Window Functions

| No. | Window $w(t)$ | Mainlobe Width | Rolloff Rate (dB/oct) | Peak Sidelobe level (dB) | Peak $20\log_{10}\delta$ |
|---|---|---|---|---|---|
| 1 | Rectangular: $\text{rect}\left(\frac{t}{T}\right)$ | $\frac{4\pi}{T}$ | $-6$ | $-13.3$ | -21dB |
| 2 | Bartlett: $\Delta\left(\frac{t}{2T}\right)$ | $\frac{8\pi}{T}$ | $-12$ | $-26.5$ | |
| 3 | Hanning: $0.5\left[1+\cos\left(\frac{2\pi t}{T}\right)\right]$ | $\frac{8\pi}{T}$ | $-18$ | $-31.5$ | -44dB |
| 4 | Hamming: $0.54+0.46\cos\left(\frac{2\pi t}{T}\right)$ | $\frac{8\pi}{T}$ | $-6$ | $-42.7$ | -53dB |
| 5 | Blackman: $0.42+0.5\cos\left(\frac{2\pi t}{T}\right)+0.08\cos\left(\frac{4\pi t}{T}\right)$ | $\frac{12\pi}{T}$ | $-18$ | $-58.1$ | -74dB |
| 6 | Kaiser: $\frac{I_0\left[\alpha\sqrt{1-4\left(\frac{t}{T}\right)^2}\right]}{I_0(\alpha)}$ $\quad 0 \le \alpha \le 10$ | $\frac{11.2\pi}{T}$ | $-6$ | $-59.9$ ($\alpha = 8.168$) | |

Lathi, Table 7.3
Punskaya, Slide 92



# FIR: Filter in Rect++ Window ☺

## *Flashback:* FIR and Low Pass Filters…

∴

$$H_d(\omega) = \begin{cases} 1 \text{ if } |\omega| \le \omega_c \\ 0 \text{ if } \omega_c < |\omega| < \pi \end{cases}$$

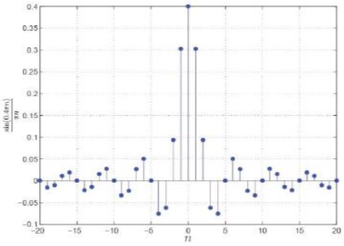
Has impulse response:

$$h_d(n) = \frac{\omega_c}{\pi} \frac{\sin \omega_c n}{\omega_c n}$$

Thus, to filter an impulse train with an ideal low-pass filter use:

$$x(t) = \left(\sum_{n=-\infty}^{\infty} x[n] \cdot \delta(t - nT)\right) * \text{sinc}\left(\frac{t}{T}\right)$$

- **However**!!
  a sinc is non-causal and infinite in duration



And, this **cannot** be implemented **in practice** ☹

∵ we need to know all samples of the input, both in the **past** and in the **future**

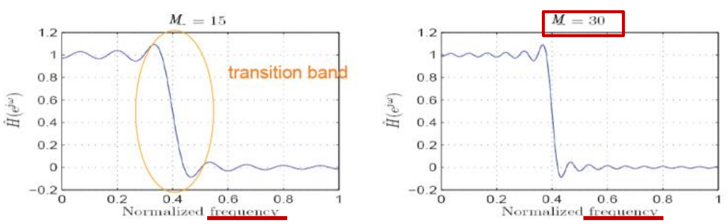


## Plan 0: Impulse Response Truncation

After "Windows", maybe we saw this coming…

∴ Clip off the sinc at some large *n*

$$\hat{h}(n) = \frac{\sin(n\omega_c)}{\pi n} \text{ for } |n| \le M \text{ and } 0 \text{ otherwise}$$

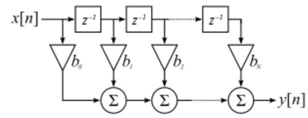



- *Ripples* in both passband/stopband and the transition not abrupt (i.e., a *transition band*).
- As M→∞, transition band→ 0 (as expected!)

## ** FIR Filter Design **

- How to get all these coefficients?

$$H(\omega) = h_0 + h_1 e^{-i\omega} + \cdots + h_{n-1} e^{-i(n-1)\omega}$$



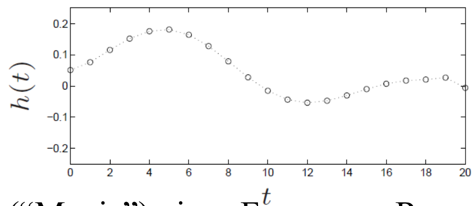

**FIR Design Methods:**

1. Impulse Response Truncation
   - + Simplest
   - – Undesirable frequency domain-characteristics, not very useful
2. Windowing Design Method
   - + Simple
   - – Not optimal (not minimum order for a given performance level)
3. Optimal filter design methods
   - + "More optimal" (treat the whole thing as a system to solve ☺)
   - – Less simple…

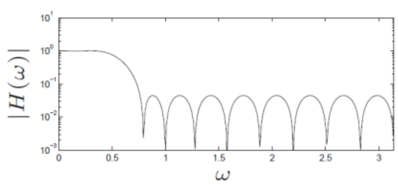


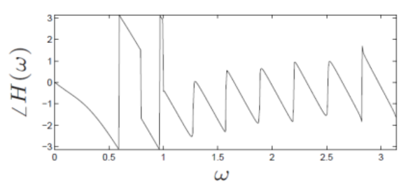
## FIR Filter Design & Operation
## Ex: Lowpass FIR filter

- Set Impulse response (order n = 21)
- "Determine" h(t)
  - – h(t) is a 20 element vector that we'll use to as a weighted sum




- FFT ("Magic") gives Frequency Response & Phase



## Why is this "hard"? Looking at the Low-Pass Example

$$H_d(\omega) = \begin{cases} 1 \text{ if } |\omega| \le \omega_c \\ 0 \text{ if } \omega_c < |\omega| < \pi \end{cases}$$

- Why is this hard?
  - Shouldn't it be "easy" ??
    … just hit it with some FFT "magic" and then keep the bands we want and then hit it with some Inverse-FFT "supermagic"???
  - Remember we need a "system" that does this "rectangle function" in frequency
    - It basically suggests we need an **Inverse FFT** of a **"rectangle function"**
  - As noted in the Window Truncation Section: The *sinc* is of infinite duration and noncausal



## ➔ Plan 2:
## FIR Filters: Window Function Design Method

- Windowing: a generalization of the truncation idea

- There many, many "window" functions:
  - Rectangular
  - Triangular
  - Hanning
  - Hamming
  - Blackman
  - Kaiser
  - Lanczos
  - Many More … (see: http://en.wikipedia.org/wiki/Window_function)

## ➔ Digital Filters Types

**FIR**

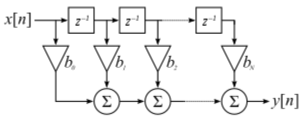
From H(z):

→ $$\begin{aligned} H(\omega) &= h_0 + h_1 e^{-\mathrm{i}\omega} + \cdots + h_{n-1} e^{-\mathrm{i}(n-1)\omega} \\ &= \sum_{t=0}^{n-1} h_t \cos t\omega - \mathrm{i} \sum_{t=0}^{n-1} h_t \sin t\omega \end{aligned}$$

→ Filter becomes a "multiply, accumulate, and delay" system:

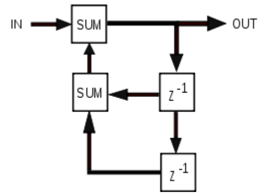
$$y(t) = \sum_{\tau=0}^{n-1} h_\tau u(t-\tau)$$

$$y[n] = b_0 x[n] + b_1 x[n-1] + \cdots + b_N x[n-N]$$




**IIR**

- Impulse response function that is non-zero over an infinite length of time.






## Filter Design Using Windows

We shall design an ideal lowpass filter of bandwidth $W$ rad/s. For this filter, the impulse response $h(t) = \frac{W}{\pi}\text{sinc}(Wt)$ (Fig. 4.48c) is noncausal and, therefore, unrealizable. Truncation of $h(t)$ by a suitable window (Fig. 4.48a) makes it realizable, although the resulting filter is now an approximation to the desired ideal filter.† We shall use a rectangular window $w_R(t)$ and a triangular (Bartlett) window $w_T(t)$ to truncate $h(t)$, and then examine the resulting filters. The truncated impulse responses $h_R(t)$ and $h_T(t)$ for the two cases are depicted in Fig. (4.48d).

$$h_R(t) = h(t)w_R(t) \qquad \text{and} \qquad h_T(t) = h(t)w_T(t)$$

Hence, the windowed filter transfer function is the convolution of $H(\omega)$ with the Fourier transform of the window, as illustrated in Fig. 4.48e and f. We make the following observations.

1. The windowed filter spectra show **spectral spreading** at the edges, and instead of a sudden switch there is a gradual transition from the passband to the stopband of the filter. The transition band is smaller ($2\pi/T$ rad/s) for the rectangular case compared to the triangular case ($4\pi/T$ rad/s).
2. Although $H(\omega)$ is bandlimited, the windowed filters are not. But the stopband behavior of the triangular case is superior to that of the rectangular case. For the rectangular window, the leakage in the stopband decreases slowly (as $1/\omega$) compared to that of the triangular window (as $1/\omega^2$). Moreover, the rectangular case has a higher peak sidelobe amplitude compared to that of the triangular window.

## Filter Design Using Windows

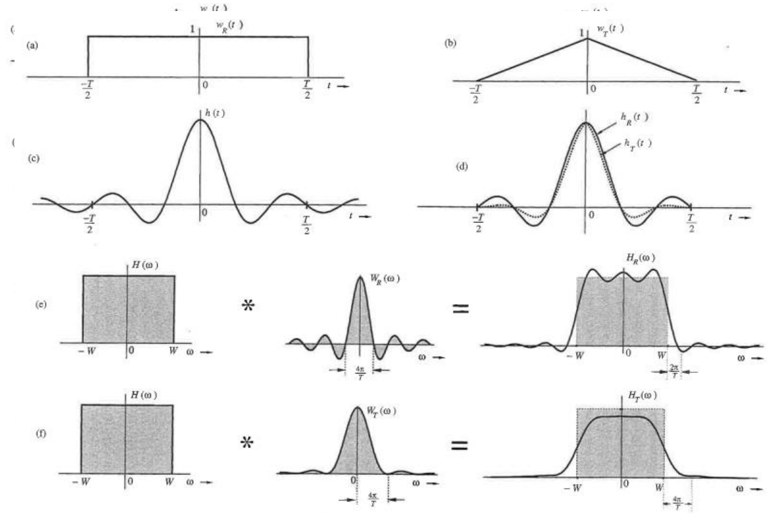
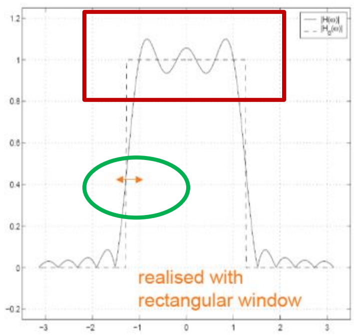


Fig. 4.48 Filter design using windows.

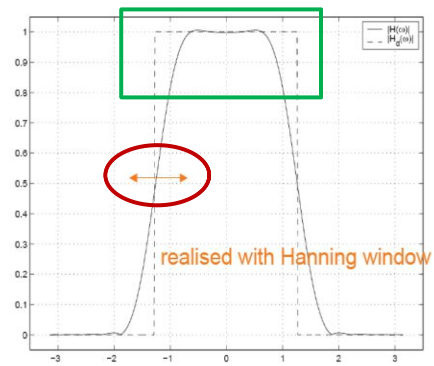


## FIR: Rectangular & Hanning Windows

- Rectangular



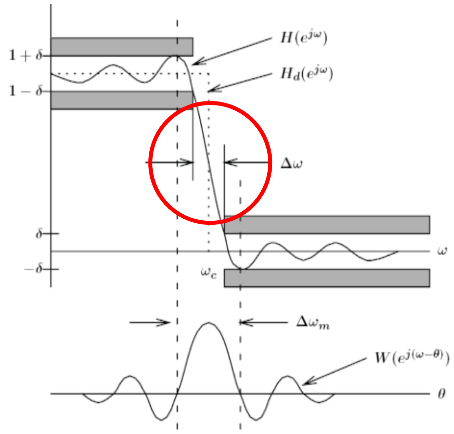

- Hanning




➔ Hanning: Less ripples, but wider transition band

Punskaya, Slide 93

## Windowed FIR Property 1: Equal transition bandwidth

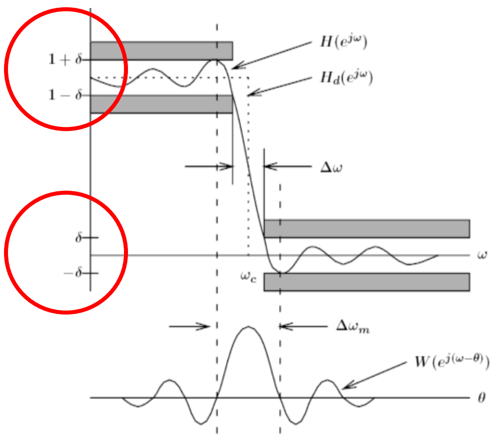



Punskaya, Slide 96

- Equal transition bandwidth on both sides of the ideal cutoff frequency



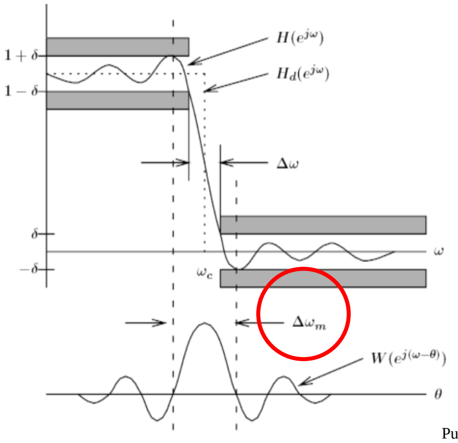
## Windowed FIR Property 2: Peak Errors same in Passband & Stopband




Punskaya, Slide 96

- Peak approximation error in the passband (1+δ → 1-δ) is equal to that in the stopband (δ → -δ)

## Windowed FIR Property 3: Mainlobe Width

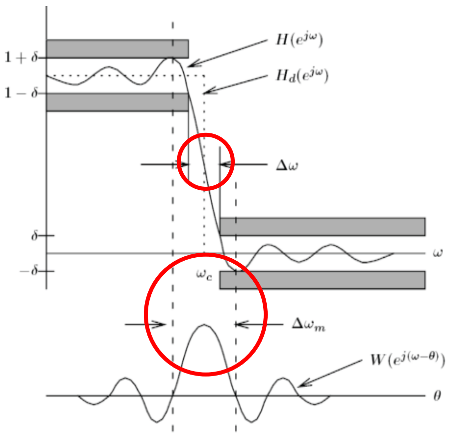



Punskaya, Slide 99

- The distance between approximation error peaks is approximately equal to the width of the mainlobe $\Delta w_m$



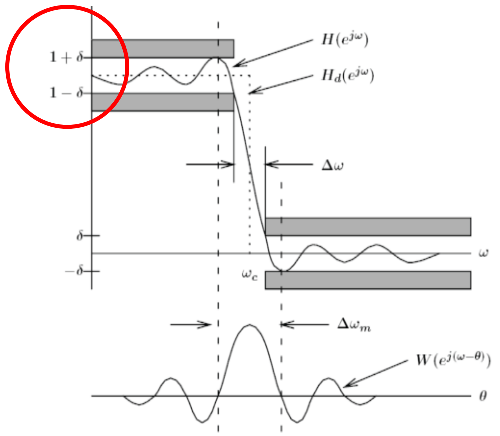
## Windowed FIR Property 4: Mainlobe Width [2]




Punskaya, Slide 96

- The width of the mainlobe is wider than the transition bandwidth

## Windowed FIR Property 5:
## Peak Δδ is determined by the window shape

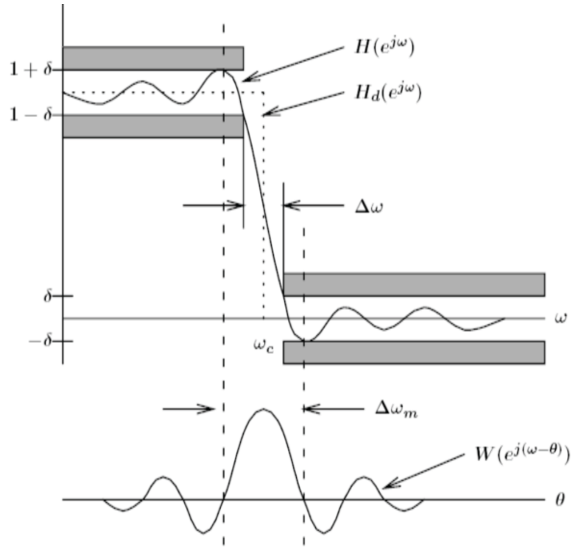



Punskaya, Slide 96

- peak approximation error is determined by the window shape, <u>independent</u> of the filter order



## Window Design Method Design Terminology




Punskaya, Slide 96

Where:

- $\omega_c$: cutoff frequency
- δ: maximum passband ripple
- Δω: transition bandwidth
- $\Delta\omega_m$: width of the window mainlobe

## Passband / stopband ripples

$\omega_s$ and $\omega_p$: Corner Frequencies

Passband / stopband ripples are often expressed in dB:

- passband ripple = 20 $log_{10}$ $(1+\delta_p)$ dB
- peak-to-peak passband ripple ≅ 20 $log_{10}$ $(1+2\delta_p)$ dB
- minimum stopband attenuation = -20 $log_{10}$ $(\delta_s)$ dB



## Passband / stopband ripples

$\omega_s$ and $\omega_p$: Corner Frequencies

Passband / stopband ripples are often expressed in dB:

- passband ripple = ~~20 $log_{10}$ $(1+\delta_p)$ dB~~ = 20 $log_{10}$ $(\delta_p)$ dB
- peak-to-peak passband ripple ≅ ~~20 $log_{10}$ $(1+2\delta_p)$ dB~~
  ≅ 20 $log_{10}$ $(2\delta_p)$ dB
- minimum stopband attenuation = ~~-20 $log_{10}$ $(\delta_s)$ dB~~
  =20 $log_{10}$ $(\delta_s)$ dB

## Summary of Design Procedure

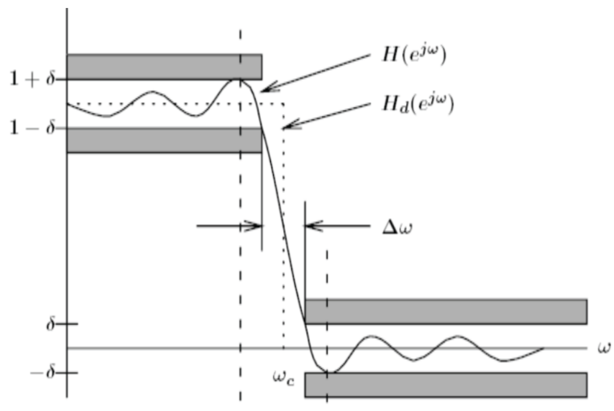
1. Select a suitable window function
2. Specify an ideal response $H_d(\omega)$
3. Compute the coefficients of the ideal filter $h_d(n)$
4. Multiply the ideal coefficients by the window function to give the filter coefficients
5. Evaluate the frequency response of the resulting filter and iterate if necessary (e.g. by increasing $M$ if the specified constraints have not been satisfied).

Punskaya, Slide 105



## Windowed Filter Design Example

- Design a type I low-pass filter with:
  - $\omega_p$ **=0.2π**
  - $\omega_s$ **=0.3π**
  - δ **=0.01**



## Windowed Filter Design Example: Step 1: Select a suitable Window Function

| No. | Window $w(t)$ | Mainlobe Width | Rolloff Rate (dB/oct) | Peak Sidelobe level (dB) | Peak $20\log_{10}\delta$ |
|---|---|---|---|---|---|
| 1 | Rectangular: $\text{rect}\left(\frac{t}{T}\right)$ | $\frac{4\pi}{T}$ | −6 | −13.3 | -21dB |
| 2 | Bartlett: $\Delta\left(\frac{t}{2T}\right)$ | $\frac{8\pi}{T}$ | −12 | −26.5 | |
| 3 | Hanning: $0.5\left[1+\cos\left(\frac{2\pi t}{T}\right)\right]$ | $\frac{8\pi}{T}$ | −18 | −31.5 | -44dB |
| 4 | Hamming: $0.54+0.46\cos\left(\frac{2\pi t}{T}\right)$ | $\frac{8\pi}{T}$ | −6 | −42.7 | -53dB |
| 5 | Blackman: $0.42+0.5\cos\left(\frac{2\pi t}{T}\right)+0.08\cos\left(\frac{4\pi t}{T}\right)$ | $\frac{12\pi}{T}$ | −18 | −58.1 | -74dB |
| 6 | Kaiser: $\frac{I_0\left[\alpha\sqrt{1-4\left(\frac{t}{T}\right)^2}\right]}{I_0(\alpha)}$ $0 \le \alpha \le 10$ | $\frac{11.2\pi}{T}$ | −6 | −59.9 ($\alpha = 8.168$) | |

- LP with: $\omega_p$ **=0.2π**, $\omega_s$ **=0.3π**, δ **=0.01**
- δ **=0.01:** The required peak error spec: =20log10 (δ) = –40 dB

} Hanning Window

- Main-lobe width:

$\omega_s - \omega_p = 0.3\pi - 0.2\pi = 0.1\pi$ → $0.1\pi = 8\pi / M$

→ Filter length M ≥ 80 & Filter order N ≥ 79

- BUT, Type-I filters have even order so N = 80



## Windowed Filter Design Example: Step 2: Specify the Ideal Response

- From Property 1 (Midpoint rule)

→ $\omega_c = (\omega_s + \omega_p)/2 = (0.2\pi+0.3\pi)/2 = 0.25\pi$

∴ An ideal response will be:

$$H_d(\omega) = \begin{cases} 1 & \text{if} \quad |\omega| \le 0.25\pi \\ 0 & \text{if} \quad 0.25\pi < |\omega| < \pi \end{cases}$$

## Windowed Filter Design Example:
## Step 3: Compute the coefficients of the ideal filter

- The ideal filter coefficients $h_d$ are given by the Inverse **Discrete time** Fourier transform of $H_d(\omega)$

$$x(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(\omega)\, e^{j\omega n} d\omega = \frac{1}{2\pi}\int_{-\omega_c}^{\omega_c} e^{j\omega n} d\omega$$
$$= \frac{\omega_c}{\pi}\frac{\sin \omega_c n}{\omega_c n}.$$

\+ Delayed impulse response (to make it causal)

$$\widetilde{h}(n) = \widehat{h}\left(n - \frac{N-1}{2}\right)$$

- Coefficients of the ideal filter (via equation or IFFT):

$$h(n) = \frac{\sin(0.25\pi(n-40))}{\pi(n-40)}$$



## Windowed Filter Design Example:
## Step 4: Multiply to obtain the filter coefficients

$$h(n) = \frac{\sin(0.25\pi(n-40))}{\pi(n-40)}$$

- Multiply by a Hamming window function for the passband:

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

## Windowed Filter Design Example: Step 5: Evaluate the Frequency Response and Iterate

- The frequency response is computed as the DFT of the filter coefficient vector
- **If** the resulting filter does not meet the specifications, **then**:
  - Adjust the ideal filter frequency response (for example, move the band edge) and repeat (step 2)
  - Adjust the filter length and repeat (step 4)
  - change the window (& filter length) (step 4)
- And/Or consult with **`Matlab`**:
  - **`FIR1`** and **`FIR2`**
  - **`B=FIR1(N,Wn):`** Designs a $N^{th}$ order FIR Window-Based FIR filter with passband given by
  - **`B=FIR2(N,F,M):`** Designs a $N^{th}$ order FIR digital filter with arbitrary frequency response specified by vectors ***F*** and ***M*** .

➔All elements of `Wn` must be [0 1):
➔where **1 corresponds to the Nyquist frequency**: 0 < Wn < 1. The Nyquist frequency is half the sample rate or **π rad/sample**.



## Windowed Filter Design Example: Consulting Matlab:

- **`FIR1`** and **`FIR2`**
  - **`B=FIR2(N,F,M):`** Designs a Nth order FIR digital filter
  - **`F`** and **`M`** specify frequency and magnitude breakpoints for the filter such that **plot(N,F,M)** shows a plot of desired frequency
  - Frequencies **`F`** must be in increasing order between 0 and $\frac{F_S}{2}$, with $F_S$ corresponding to the sample rate.
  - **B** is the vector of length N+1, it is real, has linear phase and symmetric coefficients
  - Default window is Hamming – others can be specified

## FIR Properties

- Require no feedback.
- Are inherently stable.
- They can easily be designed to be linear phase by making the coefficient sequence symmetric
- Flexibility in shaping their magnitude response
- Very Fast Implementation (based around FFTs)

- The main disadvantage of FIR filters is that considerably more computation power in a general purpose processor is required compared to an IIR filter with similar sharpness or selectivity, especially when low frequency (relative to the sample rate) cutoffs are needed.



## FIR as a class of LTI Filters

- Transfer function of the filter is

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

- Finite Impulse Response (FIR) Filters: (N = 0, no feedback)

➔From H(z):

$$\begin{aligned} H(\omega) &= h_0 + h_1 e^{-i\omega} + \cdots + h_{n-1} e^{-i(n-1)\omega} \\ &= \sum_{t=0}^{n-1} h_t \cos t\omega - i \sum_{t=0}^{n-1} h_t \sin t\omega \end{aligned}$$

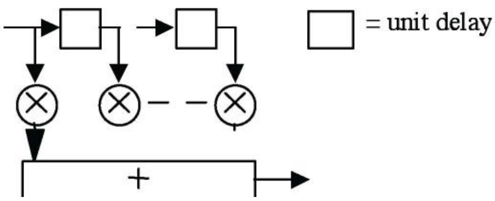
∵ H(ω) is periodic and conjugate

∴ Consider ω ∈ [0, π]

## FIR Filters

- Let us consider an FIR filter of length *M*
- Order *N*=*M*-1 **(watch out!)**
- Order → number of delays

$$y(n) = \sum_{k=0}^{M-1} b_k x(n-k) = \sum_{k=0}^{M-1} h(k)\, x(n-k)$$






## FIR Impulse Response

Obtain the impulse response immediately with $x(n) = \delta(n)$:

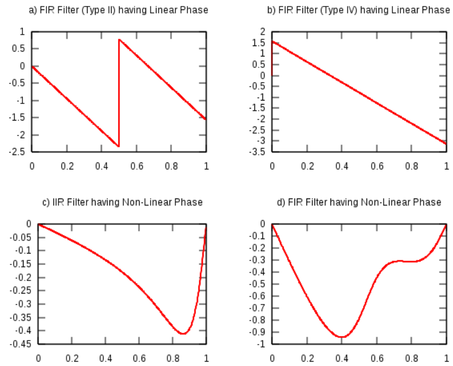
$$h(n) = y(n) = \sum_{k=0}^{M-1} b_k \delta(n-k) = b_n$$

- The impulse response is of finite length *M* (good!)
- FIR filters have only zeros (no poles) (as they must, N=0 !!)
  - Hence known also as **all-zero** filters
- FIR filters also known as **feedforward** or **non-recursive,** or **transversal** filters

# FIR & Linear Phase

- The phase response of the filter is a linear function of frequency

- Linear phase has constant group delay, all frequency components have equal delay times. ∴ No distortion due to different time delays of different frequencies



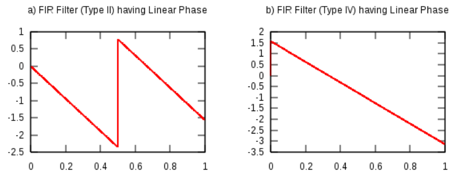

Ref: Wikipedia (Linear Phase)

- FIR Filters with:

$$\sum_{n=-\infty}^{\infty} h[n] \cdot \sin(\omega \cdot (n-\alpha)+\beta)=0$$



# FIR & Linear Phase → Four Types




Ref: Wikipedia (Linear Phase)

| Impulse response | # coefs | $H(\omega)$ | Type |
|---|---|---|---|
| $h(n)=h(M-1-n)$ | Odd | $e^{-j\omega(M-1)/2}\left(h\left(\frac{M-1}{2}\right)+2\sum_{k=1}^{(M-3)/2} h\left(\frac{M-1}{2}-k\right)\cos(\omega k)\right)$ | 1 |
| $h(n)=h(M-1-n)$ | Even | $e^{-j\omega(M-1)/2}2\sum_{k=1}^{(M-3)/2} h\left(\frac{M}{2}-k\right)\cos\left(\omega\left(k-\frac{1}{2}\right)\right)$ | 2 |
| $h(n)=-h(M-1-n)$ | Odd | $e^{-j[\omega(M-1)/2-\pi/2]}\left(2\sum_{k=1}^{(M-1)/2} h\left(\frac{M-1}{2}-k\right)\sin(\omega k)\right)$ | 3 |
| $h(n)=-h(M-1-n)$ | Even | $e^{-j[\omega(M-1)/2-\pi/2]}2\sum_{k=1}^{(M-1)/2} h\left(\frac{M}{2}-k\right)\sin\left(\omega\left(k-\frac{1}{2}\right)\right)$ | 4 |

- Type 1: most versatile
- Type 2: frequency response is always 0 at ω=π
  **(not suitable as a high-pass)**
- Type 3 and 4: introduce a π/2 phase shift, 0 at ω=0
  **(not suitable as a high-pass)**

## In Summary

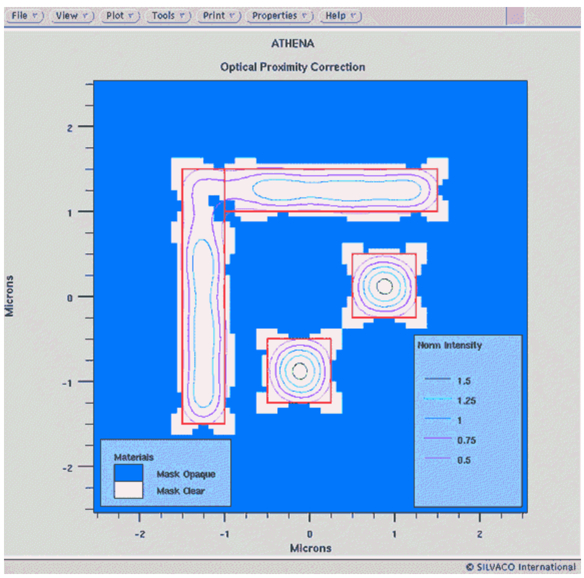
- FIR Filters are digital (can not be implemented in analog) and exploit the difference and delay operators

- A window based design builds on the notion of a truncation of the "ideal" box-car or rectangular low-pass filter in the Frequency domain (which is a sinc function in the time domain)

- Other Design Methods exist:
  – Least-Square Design
  – Equiripple Design
  – Remez method
  – The Parks-McClellan Remez algorithm
  – Optimisation routines …



## Advanced Application: Optical Proximity Correction



## Next Time…

- **FFTs**

- Review:
  - Chapter 12 of Lathi
  - § 10. 3 of Strang on FFTs (cached on Course Website)

- Ponder?

$$y[k] = f[k] * h[k] \qquad Y(\Omega) = F(\Omega)H(\Omega)$$

where $F(\Omega)$, $Y(\Omega)$, and $H(\Omega)$ are DTFTs of $f[k]$, $y[k]$, and $h[k]$, respectively; that is,

$$f[k] \Longleftrightarrow F(\Omega), \quad y[k] \Longleftrightarrow Y(\Omega), \quad \text{and} \quad h[k] \Longleftrightarrow H(\Omega)$$