http://elec3004.com

Digital Filters (Part II)

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

Dr. Surya Singh

Lecture 12
(with material from Lathi and Cannon (Discrete Systems))

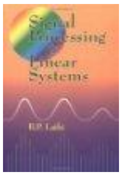
elec3004@itee.uq.edu.au
http://robotics.itee.uq.edu.au/~elec3004/

April 14, 2016

Lecture Schedule:

Week	Date	Lecture Title
1	29-Feb	Introduction
	3-Mar	Systems Overview
2	7-Mar	Systems as Maps & Signals as Vectors
	10-Mar	Data Acquisition & Sampling
3	14-Mar	Sampling Theory
	17-Mar	Antialiasing Filters
4	21-Mar	Discrete System Analysis
	24-Mar	Convolution Review
	28-Mar	Holiday
	31-Mar	
5	4-Apr	Frequency Response & Filter Analysis
	7-Apr	Filters
6	11-Apr	Digital Filters
	14-Apr	Digital Filters
7	18-Apr	Digital Windows
	21-Apr	FFT
8	25-Apr	Holiday
	28-Apr	Feedback
9	3-May	Introduction to Feedback Control
	5-May	Servoregulation/PID
10	9-May	Introduction to (Digital) Control
	12-May	Digitial Control
11	16-May	Digital Control Design
	19-May	Stability
12	23-May	Digital Control Systems: Shaping the Dynamic Response & Estimation
	26-May	Applications in Industry
13	30-May	System Identification & Information Theory
	2-Jun	Summary and Course Review

Follow Along Reading:



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

Today

- 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

- 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

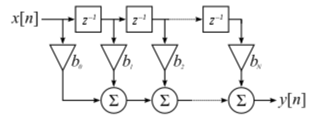
Next Time

FIR!

** FIR Filter Design **

- How to get all these coefficients?

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

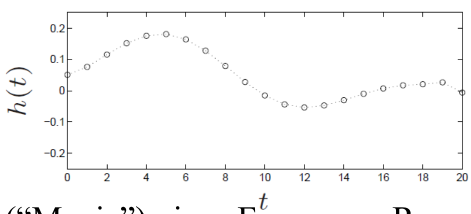


FIR Design Methods:

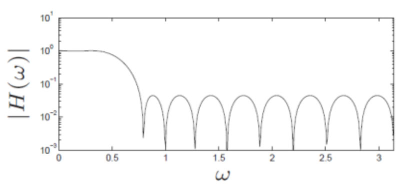
1. Impulse Response Truncation
   + Simplest
   – Undesirable frequency domain-characteristics, not very useful
2. Windowing Design Method
   + Simple
   – Not optimal (not minimum order for a given performance level)
3. Optimal filter design methods
   + "More optimal"
   – Less simple…

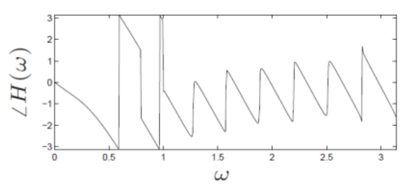
FIR Filter Design & Operation
Ex: Lowpass FIR filter

- 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

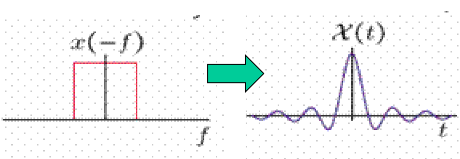
$$H_d(\omega) = \begin{cases} 1 \text{ if } |\omega| \leq \omega_c \\ 0 \text{ if } \omega_c < |\omega| < \pi \end{cases}$$

- 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
  - Let's consider what that means…
    - It basically suggests we need an **Inverse FFT** of a **"rectangle function"**

Flashback: Fourier Series & Rectangular Functions

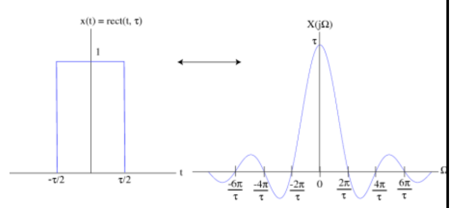
$\mathfrak{F}$: Fourier Tranform

$$\mathfrak{F}^{-1}\left\{rect\left(\frac{\omega}{2}\right)\right\} = \frac{sinc(t)}{\pi}$$



Ref: http://cnx.org/content/m26719/1.1/
http://www.wolframalpha.com/input/?i=IFFT%28sinc%28f%29%29

$$\mathfrak{F}\{rect(t)\} = sinc\left(\frac{\omega}{2}\right)$$



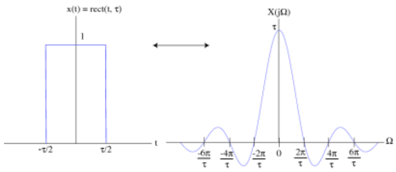
Ref: http://cnx.org/content/m32899/1.8/
http://www.thefouriertransform.com/pairs/box.php

See:

- Table 7.1 (p. 702) Entry 17 & Table 9.1 (p. 852) Entry 7

Flashback: Fourier Series & Rectangular Functions [2]

- The sinc function might look familiar
  - This is the frequency content of a square wave (box)



Ref: http://www.wolframalpha.com/input/?i=FFT%28rect%28t%29%29
http://cnx.org/content/m32899/1.8/

- This also applies to **signal reconstruction!**

➔ Whittaker–Shannon interpolation formula
  - This says that the "better way" to go from Discrete to Continuous (i.e. D to A) is not ZOH, but rather via the sinc!

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

∴ FIR and Low Pass Filters…

∴

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

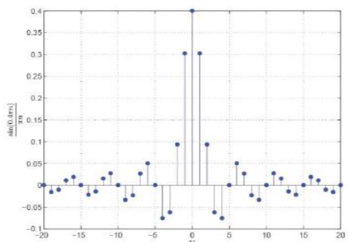
Has impulse response:

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

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

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

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



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

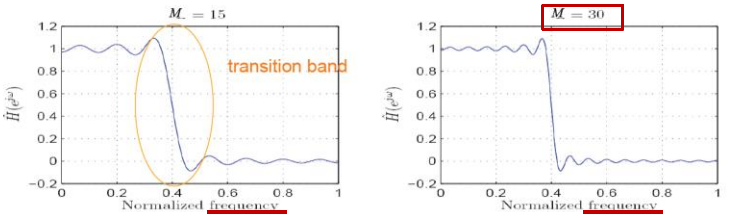
∵ we need to know all samples of the input, both in the **past** and in the **future**

Plan 0: Impulse Response Truncation

Maybe we saw this coming…

∴ Clip off the sinc at some large *n*

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



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

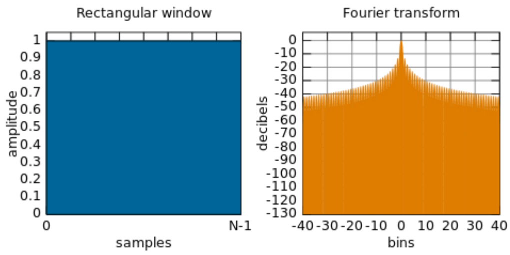
→ FIR Filters: Window Function Design Method

- 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)

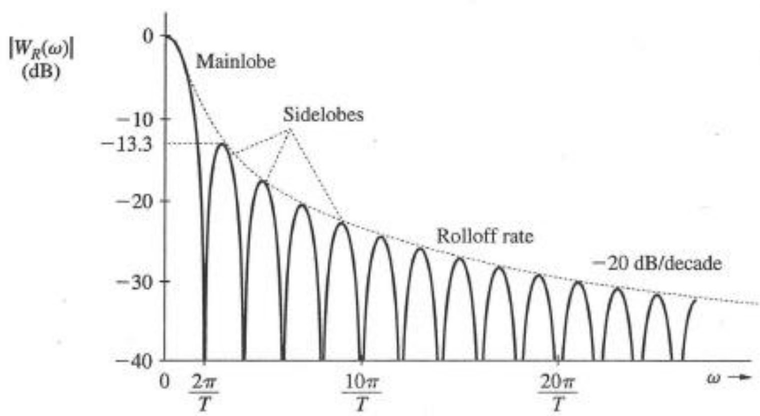
Some Window Functions [1]

1. Rectangular

$$w(n) = 1$$



Windowing and its effects/terminology

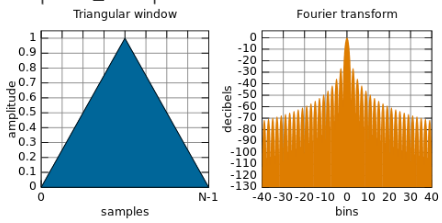


Lathi, Fig. 7.45

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:

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

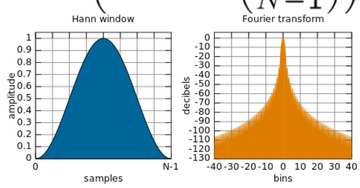
Some More Window Functions…

3. Generalized Hamming Windows

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

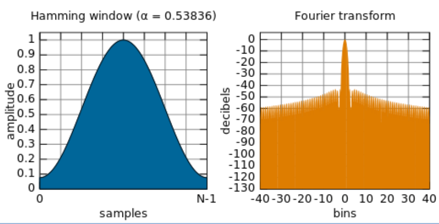
→ Hanning Window

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



→ Hamming's Window

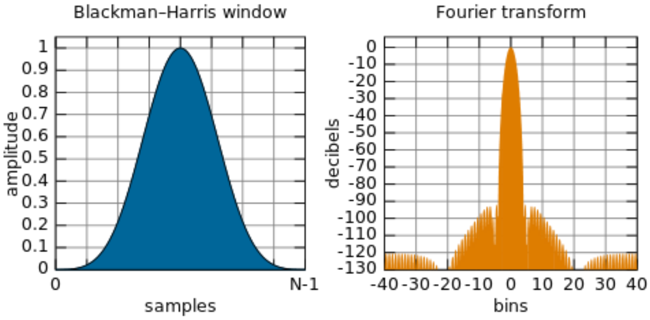
$$\rightarrow \alpha = 0.54, \ \beta = 1 - \alpha = 0.46$$



Some More Window Functions…

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

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

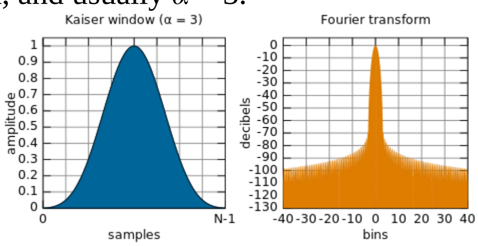


Some More Window Functions…

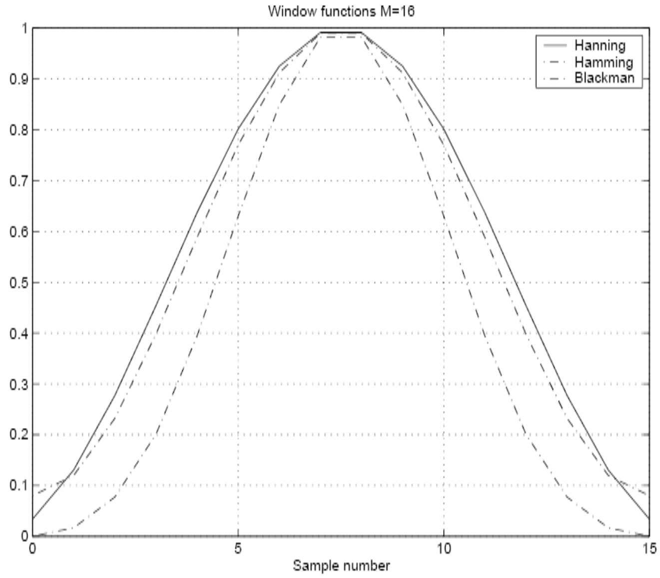
5. Kaiser window
   – A DPSS (discrete prolate spheroidal sequence)
   – Maximize the energy concentration in the main lobe

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

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



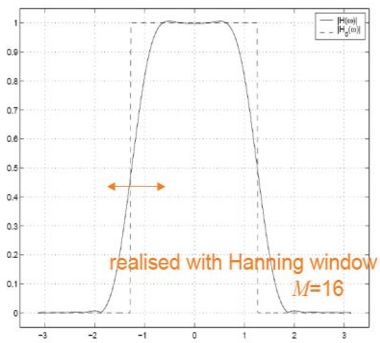
Comparison of Alternative Windows –Time Domain

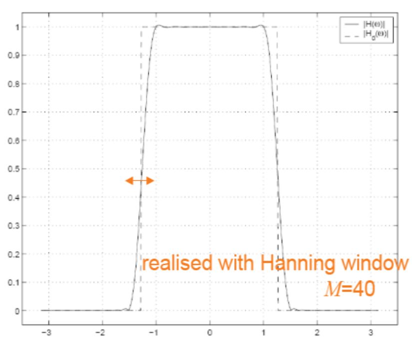


Punskaya, Slide 90

Adding Order

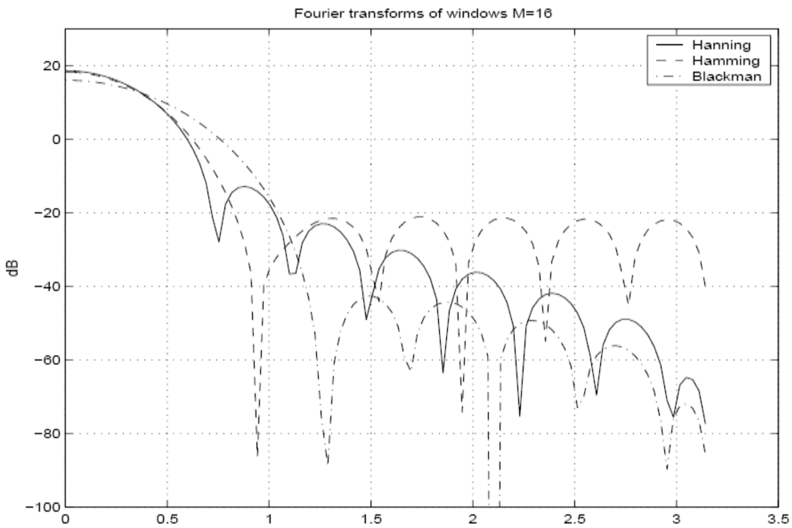
+ Transition and Smoothness
– Increased Size





Punskaya, Slide 94

Comparison of Alternative Windows
Frequency Domain



Punskaya, Slide 91

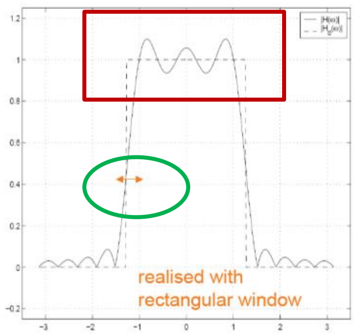
Summary Characteristics of Common Window Functions

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 ($\alpha = 8.168$)	

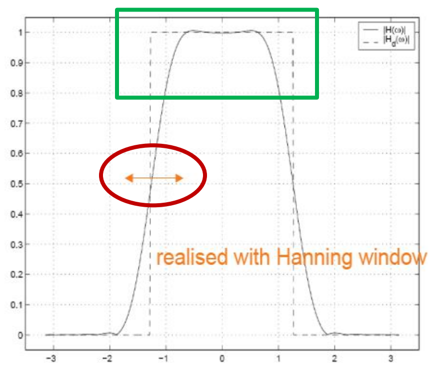
Lathi, Table 7.3
Punskaya, Slide 92

FIR: Rectangular & Hanning Windows

- Rectangular



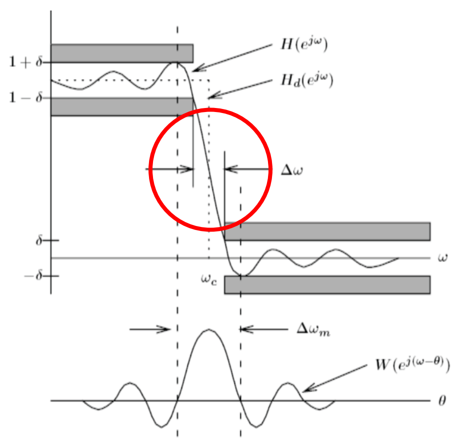
- Hanning



➔ Hanning: Less ripples, but wider transition band

Punskaya, Slide 93

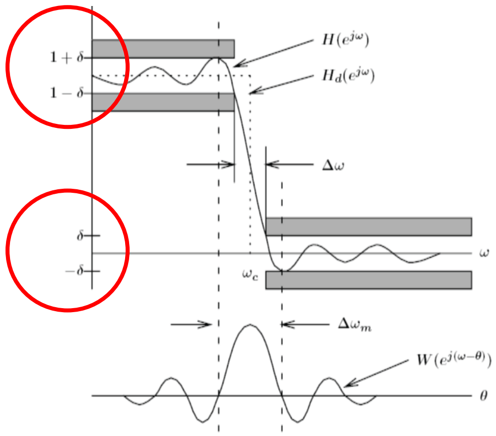
Windowed FIR Property 1: Equal transition bandwidth



Punskaya, Slide 96

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

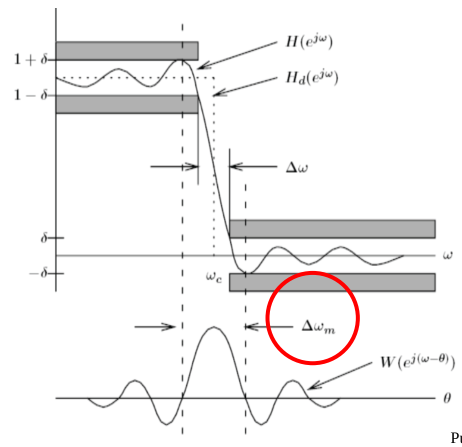
Windowed FIR Property 2:
Peak Errors same in Passband & Stopband



Punskaya, Slide 96

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

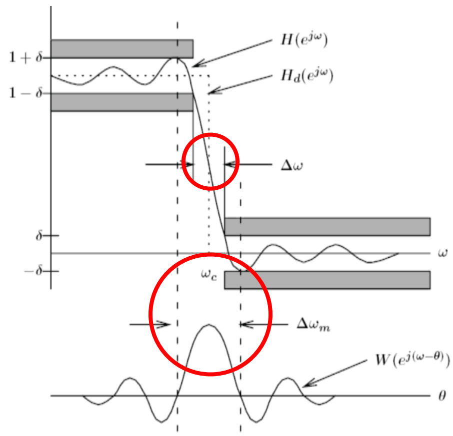
Windowed FIR Property 3:
Mainlobe Width



Punskaya, Slide 99

- The distance between approximation error peaks is approximately equal to the width of the mainlobe Δw_m

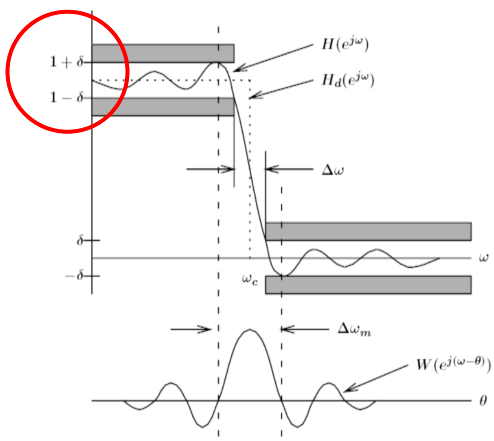
Windowed FIR Property 4: Mainlobe Width [2]



Punskaya, Slide 96

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

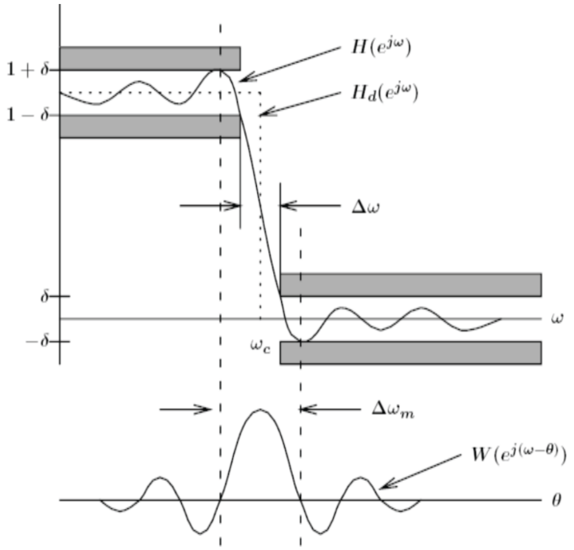
Windowed FIR Property 5: Peak Δδ is determined by the window shape



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:

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

Passband / stopband ripples

ω_s and ω_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} (δ_s) dB

Passband / stopband ripples

ω_s and ω_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 $\cong$ ~~$20\ log_{10}\ (1+2\delta_p)$ dB~~
  $\cong 20\ log_{10}\ (2\delta_p)$ dB
- minimum stopband attenuation = ~~$-20\ log_{10}\ (\delta_s)$ dB~~
  $=20\ log_{10}\ (\delta_s)$ dB

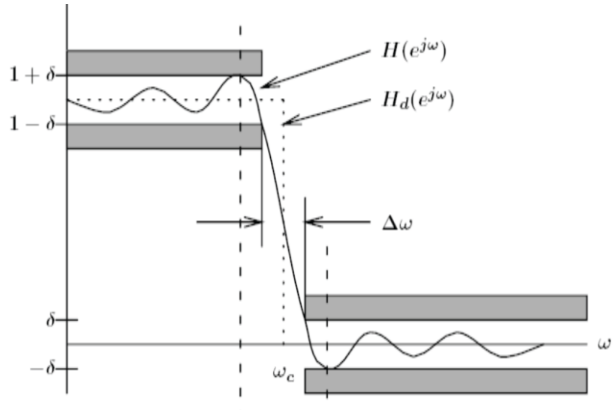
Summary of Design Procedure

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:
  - ω_p =**0.2π**
  - ω_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: ω_p =**0.2π**, ω_s =**0.3π**, δ =**0.01**
- **δ =0.01:** The required peak error spec: -20log10 (δ) = –40 dB } Hanning Window
- Main-lobe width:

ω_s-ω_p=0.3π-0.2π =0.1π → 0.1π = 8π / *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:

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

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 $H_d(\omega)$

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

\+ 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=FIR2(N,F,M):`** Designs a Nth order FIR digital filter with

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 Fs/2, with Fs 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

In Conclusion

- 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 …

Next Time…

- **Digital Filters**

- Review:
  - Chapter 12 of Lathi

- 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,

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