http://elec3004.com

Digital Filters IIR
(& Their Corresponding Analog Filters)

ELEC 3004: **Systems**: Signals & Controls
Dr. Surya Singh

Lecture 11

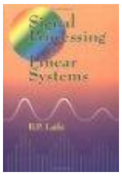
elec3004@itee.uq.edu.au
http://robotics.itee.uq.edu.au/~elec3004/

April 6, 2017

Lecture Schedule:

Week	Date	Lecture Title
1	28-Feb	Introduction
	2-Mar	Systems Overview
2	7-Mar	Systems as Maps & Signals as Vectors
	9-Mar	Systems: Linear Differential Systems
3	14-Mar	Sampling Theory & Data Acquisition
	16-Mar	Aliasing & Antialiasing
4	21-Mar	Discrete Time Analysis & Z-Transform
	23-Mar	Second Order LTID (& Convolution Review)
5	28-Mar	Frequency Response
	30-Mar	Filter Analysis
6	4-Apr	Digital Filters (IIR) & Filter Analysis
	6-Apr	Digital Filter (FIR)
7	11-Apr	Digital Windows
	13-Apr	FFT
	18-Apr	Holiday
	20-Apr	
	25-Apr	
8	27-Apr	Active Filters & Estimation
9	2-May	Introduction to Feedback Control
	4-May	Servoregulation/PID
10	9-May	Introduction to (Digital) Control
	11-May	Digitial Control
11	16-May	Digital Control Design
	18-May	Stability
12	23-May	Digital Control Systems: Shaping the Dynamic Response
	25-May	Applications in Industry
13	30-May	System Identification & Information Theory
	1-Jun	Summary and Course Review

Follow Along Reading:



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

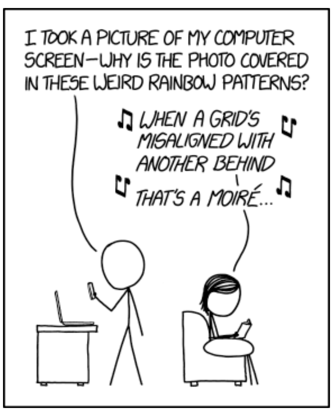
Today

- 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

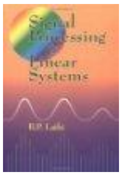
Next Time

Announcements



- Lab next week
  - Only on Thursday (April 14)
  - No lab sessions on the other days of the week
  - Thanks!

Follow Along Reading:



B. P. Lathi
Signal processing and linear systems
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

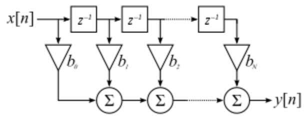
Next Time

FIR!

** FIR Filter Design **

- How to get all these coefficients?

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

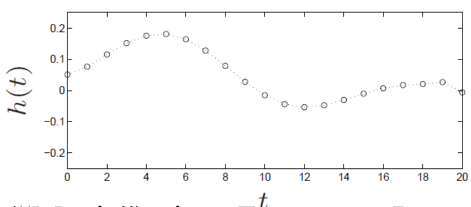


FIR Design Methods:

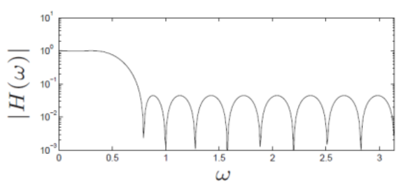
1. Impulse Response Truncation
   + Simplest
   – Undesirable frequency domain-characteristics, not very useful
2. Windowing Design Method
   + Simple
   – Not optimal (not minimum order for a given performance level)
3. Optimal filter design methods
   + "More optimal"
   – Less simple…

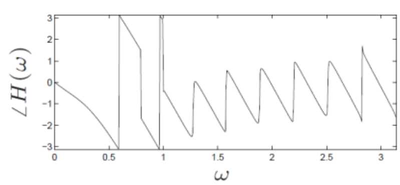
FIR Filter Design & Operation
Ex: Lowpass FIR filter

- 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

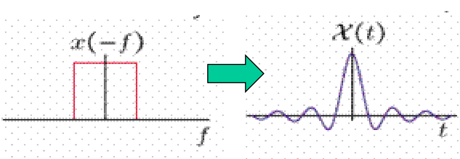
$$H_d(\omega) = \begin{cases} 1 \text{ if } |\omega| \le \omega_c \\ 0 \text{ if } \omega_c < |\omega| < \pi \end{cases}$$

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

Flashback: Fourier Series & Rectangular Functions

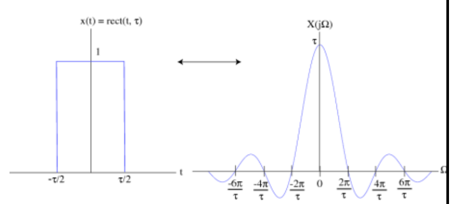
$\mathfrak{F}$: Fourier Tranform

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



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

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



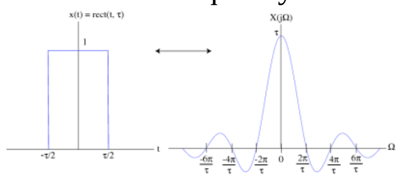
Ref: http://cnx.org/content/m32899/1.8/
http://www.thefouriertransform.com/pairs/box.php

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!

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

∴ FIR and Low Pass Filters…

∴

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

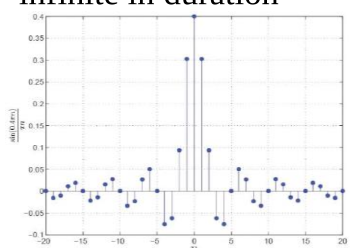
Has impulse response:

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

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

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

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



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

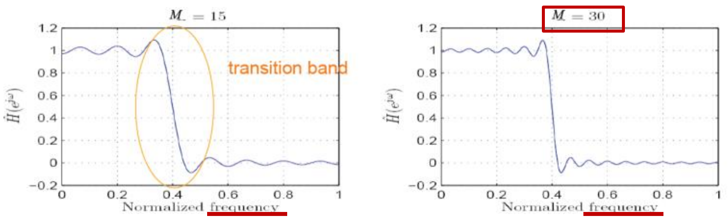
∵ we need to know all samples of the input, both in the **past** and in the **future**

Plan 0: Impulse Response Truncation

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)

→ Digital Filters Types

FIR

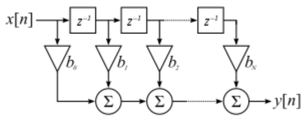
From H(z):

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

→ Filter becomes a “multiply, accumulate, and delay” system:

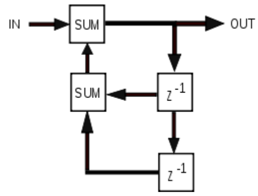
$$y(t) = \sum_{\tau=0}^{n-1} h_\tau u(t-\tau)$$

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



IIR

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



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):

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

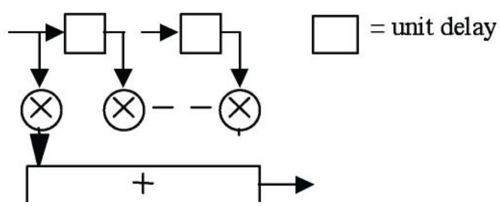
∵ H(ω) is periodic and conjugate

∴ Consider $\omega \in [0, \pi]$

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

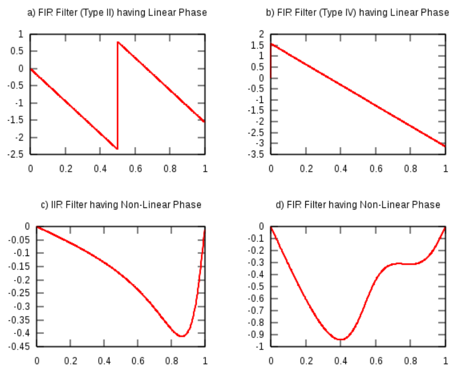
Obtain the impulse response immediately with $x(n)= \delta(n)$:

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

- 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

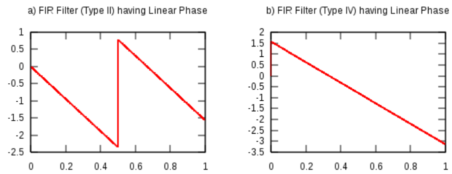


Ref: Wikipedia (Linear Phase)

- 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 $\omega=\pi$
  (not suitable as a high-pass)
- Type 3 and 4: introduce a $\pi/2$ phase shift, 0 at $\omega=0$
  (not suitable as a high-pass)

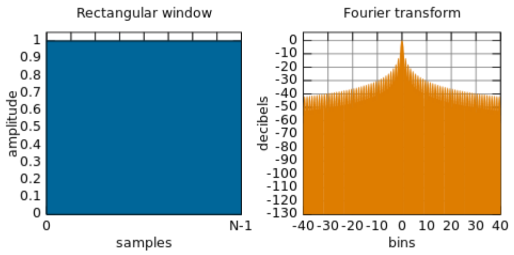
Digital Windows!

(Preview Edition)

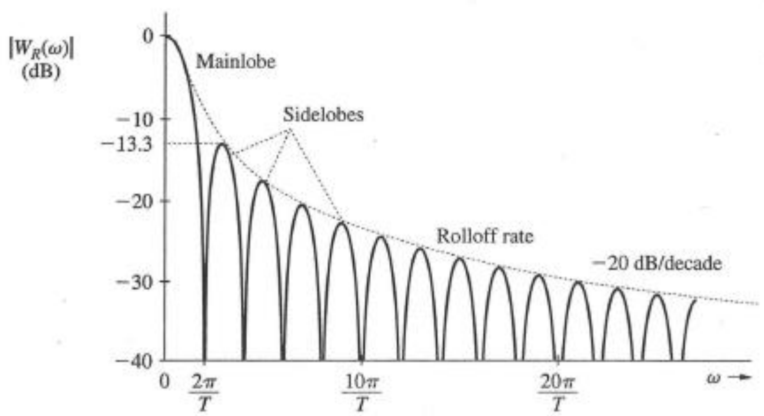
Some Window Functions [1]

1. Rectangular

$$w(n) = 1$$



Windowing and its effects/terminology

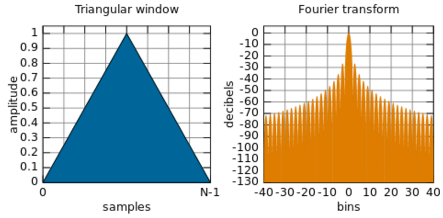


Lathi, Fig. 7.45

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:

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

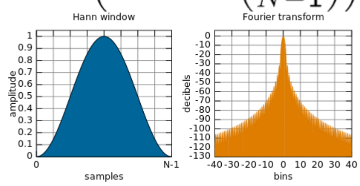
Some More Window Functions…

3. Generalized Hamming Windows

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

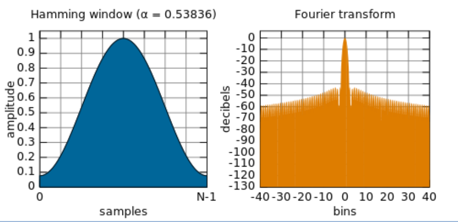
→ Hanning Window

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



→ Hamming's Window

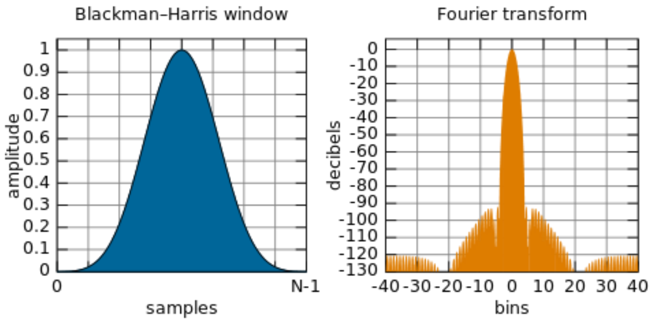
$$\rightarrow \alpha = 0.54,\ \beta = 1 - \alpha = 0.46$$



Some More Window Functions…

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

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

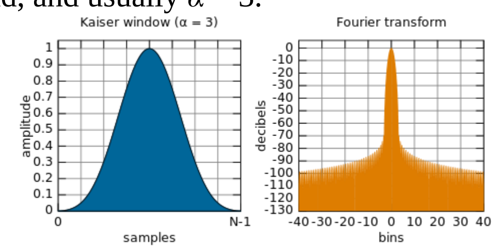


Some More Window Functions…

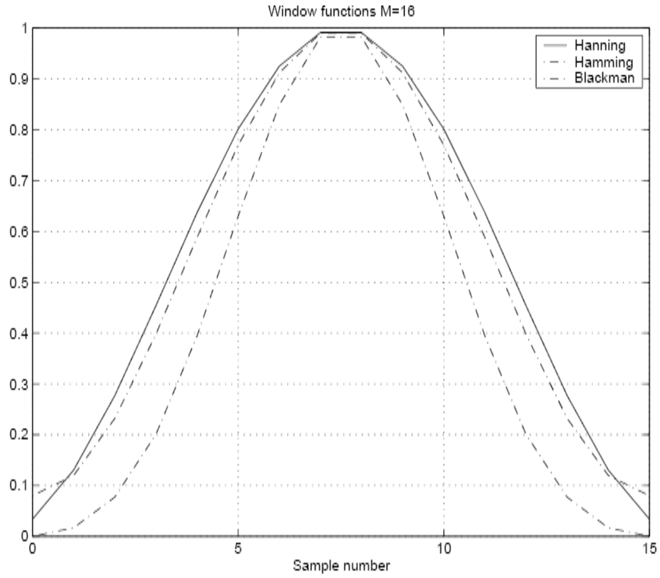
5. Kaiser window
   – A DPSS (discrete prolate spheroidal sequence)
   – Maximize the energy concentration in the main lobe

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

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

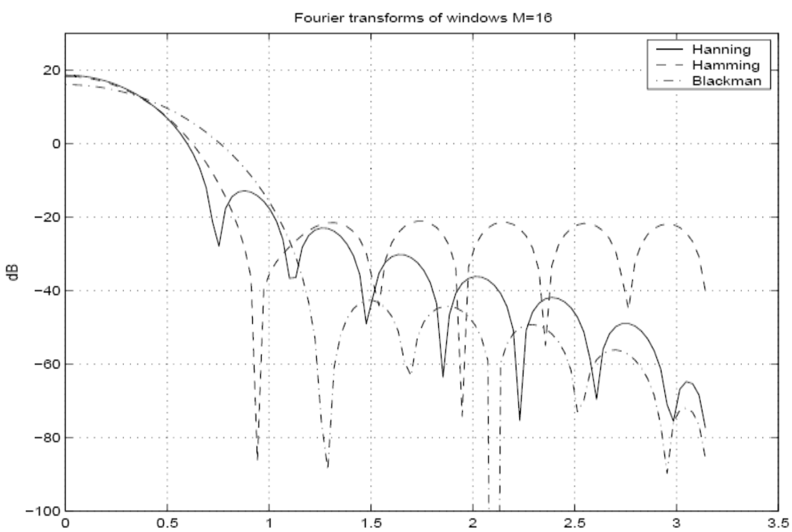


Comparison of Alternative Windows –Time Domain



Punskaya, Slide 90

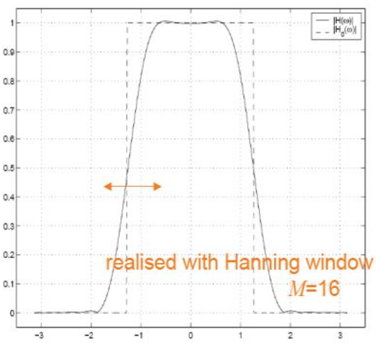
Comparison of Alternative Windows Frequency Domain

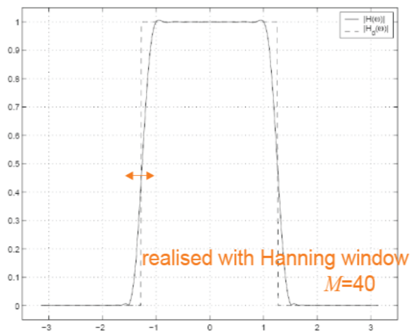


Punskaya, Slide 91

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: $\frac{I_0\left[\alpha\sqrt{1-4\left(\frac{t}{T}\right)^2}\right]}{I_0(\alpha)}$ $0\le\alpha\le10$	$\frac{11.2\pi}{T}$	−6	−59.9 ($\alpha = 8.168$)	

Lathi, Table 7.3
Punskaya, Slide 92

BREAK

Back to

FIR!

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

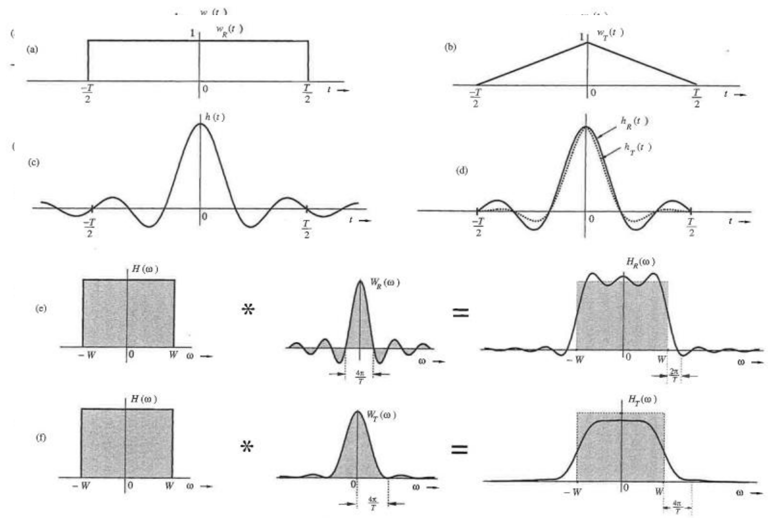
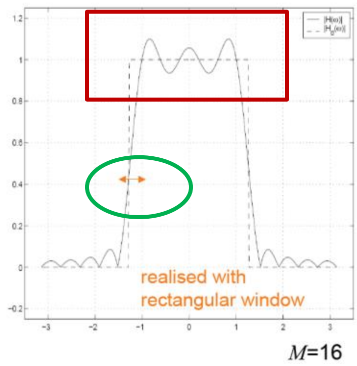


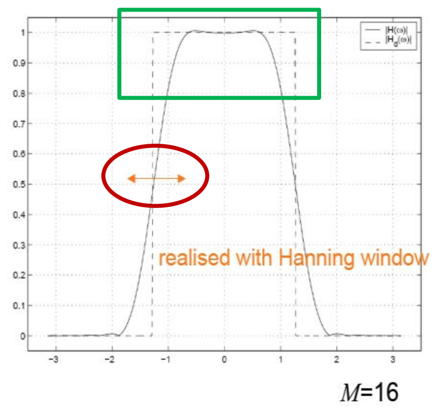
Fig. 4.48 Filter design using windows.

FIR: Rectangular & Hanning Windows

- Rectangular



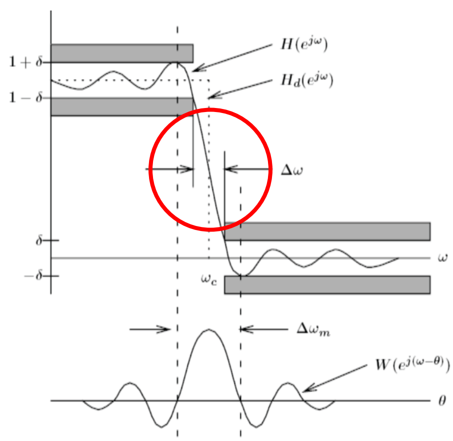
- Hanning



➔ Hanning: Less ripples, but wider transition band

Punskaya, Slide 93

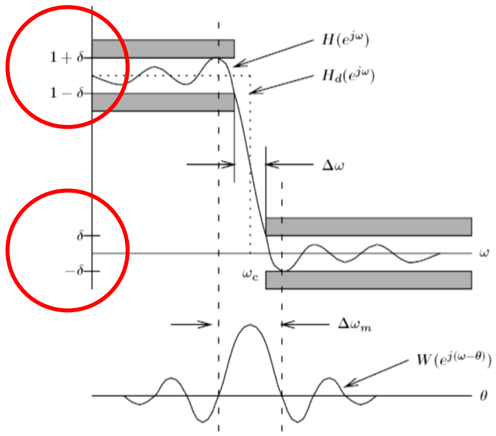
Windowed FIR Property 1: Equal transition bandwidth



Punskaya, Slide 96

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

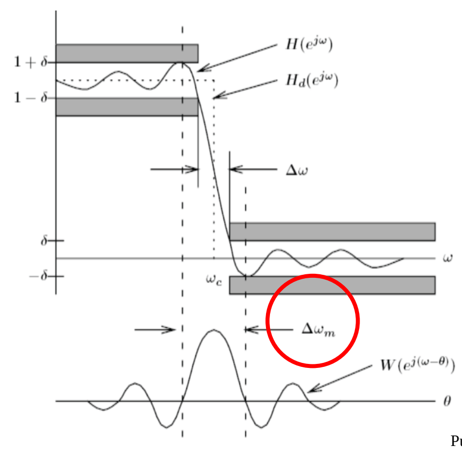
Windowed FIR Property 2:
Peak Errors same in Passband & Stopband



Punskaya, Slide 96

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

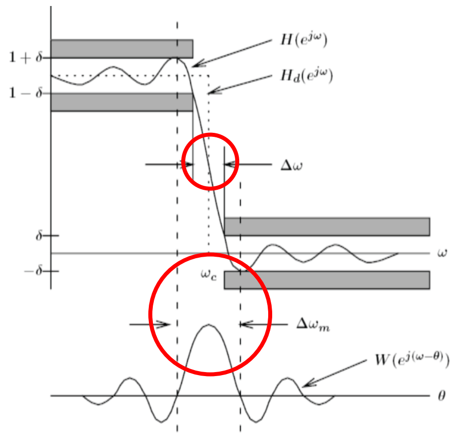
Windowed FIR Property 3:
Mainlobe Width



Punskaya, Slide 99

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

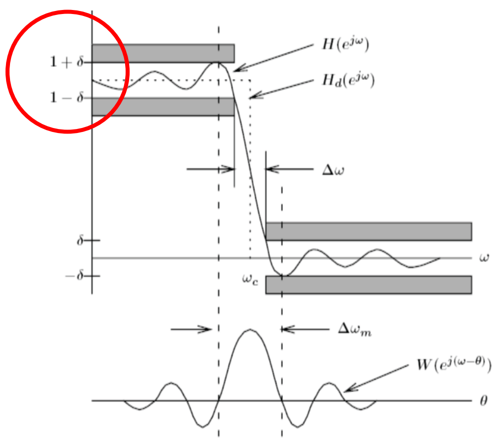
Windowed FIR Property 4: Mainlobe Width [2]



Punskaya, Slide 96

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

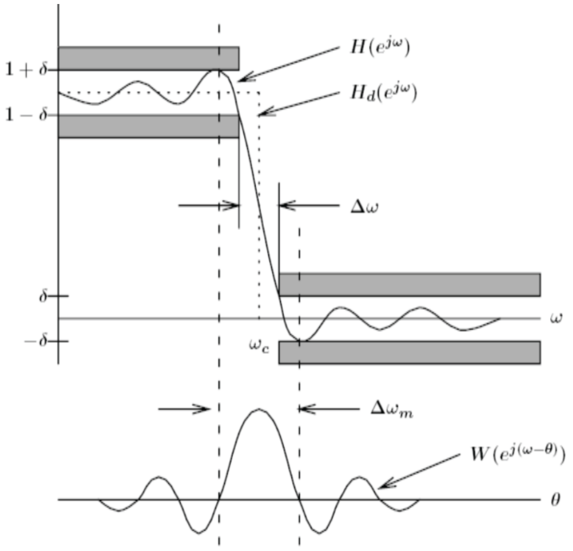
Windowed FIR Property 5: Peak Δδ is determined by the window shape



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

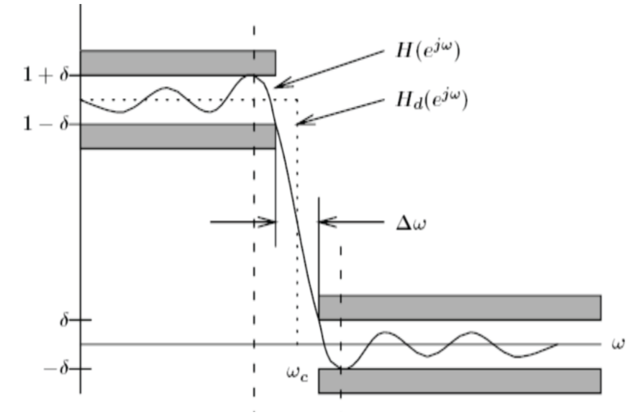
Summary of Design Procedure

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:
  - $\omega_p = \mathbf{0.2\pi}$
  - $\omega_s = \mathbf{0.3\pi}$
  - $\delta = \mathbf{0.01}$



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: $\omega_p = \mathbf{0.2\pi}$, $\omega_s = \mathbf{0.3\pi}$, $\delta = \mathbf{0.01}$
- $\delta = \mathbf{0.01}$**:** The required peak error spec: $-20\log10(\delta) = -40$ dB
- Main-lobe width:

} Hanning Window

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

→ Filter length $M \geq 80$ & Filter order $N \geq 79$

- 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 $H_d(\omega)$

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

+ Delayed impulse response (to make it causal)

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

- 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

Frequency Response of Discrete-Time Systems

For (asymptotically stable) continuous-time systems we showed that the system response to an input $e^{j\omega t}$ is $H(j\omega)e^{j\omega t}$, and that the response to an input $\cos \omega t$ is $|H(j\omega)| \cos[\omega t + \angle H(j\omega)]$. Similar results hold for discrete-time systems. We now show that for an (asymptotically stable) LTID system, the system response to an input $e^{j\Omega k}$ is $H[e^{j\Omega}]e^{j\Omega k}$ and the response to an input $\cos \Omega k$ is $|H[e^{j\Omega}]| \cos(\Omega k + \angle H[e^{j\Omega}])$.

The proof is similar to the one used in continuous-time systems. In Sec. 9.4-2 we showed that an LTID system response to an (everlasting) exponential z^k is also an (everlasting) exponential $H[z]z^k$. It is helpful to represent this relationship by a directed arrow notation as

$$z^k \Longrightarrow H[z]z^k \tag{12.1}$$

Setting $z = e^{\pm j\Omega}$ in this relationship yields

$$e^{j\Omega k} \Longrightarrow H[e^{j\Omega}]e^{j\Omega k} \tag{12.2a}$$
$$e^{-j\Omega k} \Longrightarrow H[e^{-j\Omega}]e^{-j\Omega k} \tag{12.2b}$$

Addition of these two equations yields

$$2\cos \Omega k \Longrightarrow H[e^{j\Omega}]e^{j\Omega k} + H[e^{-j\Omega}]e^{-j\Omega k} = 2\text{Re}\left(H[e^{j\Omega}]e^{j\Omega k}\right) \tag{12.3}$$

Expressing $H[e^{j\Omega}]$ in the polar form

$$H[e^{j\Omega}] = |H[e^{j\Omega}]|e^{j\angle H[e^{j\Omega}]} \tag{12.4}$$

Eq. (12.3) can be expressed as

Frequency Response of Discrete-Time Systems

$$\cos \Omega k \Longrightarrow |H[e^{j\Omega}]| \cos \left(\Omega k + \angle H[e^{j\Omega}]\right) \tag{12.5}$$

In other words, the system response $y[k]$ to a sinusoidal input $\cos \Omega k$ is given by

$$y[k] = |H[e^{j\Omega}]| \cos \left(\Omega k + \angle H[e^{j\Omega}]\right) \tag{12.6a}$$

Following the same argument, the system response to a sinusoid $\cos(\Omega k + \theta)$ is

$$y[k] = |H[e^{j\Omega}]| \cos \left(\Omega k + \theta + \angle H[e^{j\Omega}]\right) \tag{12.6b}$$

This result applies only to asymptotically stable systems because Eq. (12.1) is valid only for values of z lying in the region of convergence of $H[z]$. For $z = e^{j\Omega}$, z lies on the unit circle ($|z| = 1$). The region of convergence for unstable and marginally stable systems does not include the unit circle.

This important result shows that the response of an asymptotically stable LTID system to a discrete-time sinusoidal input of frequency Ω is also a discrete-time sinusoid of the same frequency. *The amplitude of the output sinusoid is $|H[e^{j\Omega}]|$ times the input amplitude, and the phase of the output sinusoid is shifted by $\angle H[e^{j\Omega}]$ with respect to the input phase.* Clearly $|H[e^{j\Omega}]|$ is the amplitude gain, and a plot of $|H[e^{j\Omega}]|$ versus Ω is the amplitude response of the discrete-time system. Similarly, $\angle H[e^{j\Omega}]$ is the phase response of the system, and a plot of $\angle H[e^{j\Omega}]$ vs Ω shows how the system modifies or shifts the phase of the input sinusoid. Note that $H[e^{j\Omega}]$ incorporates the information of both amplitude and phase response and therefore is called the **frequency response** of the system.

These results, although parallel to those for continuous-time systems, differ from them in one significant aspect. In the continuous-time case, the frequency response is $H(j\omega)$. A parallel result for the discrete-time case would lead to frequency response $H[j\Omega]$. Instead, we found the frequency response to be $H[e^{j\Omega}]$. This deviation causes some interesting differences between the behavior of continuous-time and discrete-time systems.

Frequency Response of Discrete-Time Systems

■ **Example 12.1**

For a system specified by the equation

$$y[k+1] - 0.8y[k] = f[k+1]$$

find the system response to the input **(a)** $1^k = 1$ **(b)** $\cos\left[\frac{\pi}{6}k - 0.2\right]$ **(c)** a sampled sinusoid $\cos 1500t$ with sampling interval $T = 0.001$.

The system equation can be expressed as

$$(E - 0.8)y[k] = Ef[k]$$

Therefore, the transfer function of the system is

$$H[z] = \frac{z}{z - 0.8} = \frac{1}{1 - 0.8z^{-1}}$$

The frequency response is

$$\begin{aligned} H[e^{j\Omega}] &= \frac{1}{1 - 0.8e^{-j\Omega}} \qquad (12.7) \\ &= \frac{1}{1 - 0.8(\cos\Omega - j\sin\Omega)} \\ &= \frac{1}{(1 - 0.8\cos\Omega) + j0.8\sin\Omega} \end{aligned}$$

Therefore

$$\begin{aligned} \left|H[e^{j\Omega}]\right| &= \frac{1}{\sqrt{(1 - 0.8\cos\Omega)^2 + (0.8\sin\Omega)^2}} \\ &= \frac{1}{\sqrt{1.64 - 1.6\cos\Omega}} \qquad (12.8a) \end{aligned}$$

and

$$\angle H[e^{j\Omega}] = -\tan^{-1}\left[\frac{0.8\sin\Omega}{1 - 0.8\cos\Omega}\right] \tag{12.8b}$$

The amplitude response $|H[e^{j\Omega}]|$ can also be obtained by observing that $|H|^2 = HH^*$. Therefore

$$\begin{aligned} \left|H[e^{j\Omega}]\right|^2 &= H[e^{j\Omega}]H^*[e^{j\Omega}] \\ &= H[e^{j\Omega}]H[e^{-j\Omega}] \qquad (12.9) \end{aligned}$$

From Eq. (12.7) it follows that

$$\begin{aligned} \left|H[e^{j\Omega}]\right|^2 &= \left(\frac{1}{1 - 0.8e^{-j\Omega}}\right)\left(\frac{1}{1 - 0.8e^{j\Omega}}\right) \\ &= \frac{1}{1.64 - 1.6\cos\Omega} \end{aligned}$$

which yields the result found earlier in Eq. (12.8a).

Frequency Response of Discrete-Time Systems

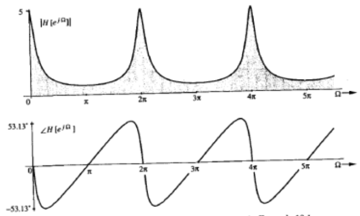


Fig. 12.1 Frequency response of an LTID system in Example 12.1.

Figure 12.1 shows plots of amplitude and phase response as functions of Ω. We now compute the amplitude and the phase response for the various inputs:

(a) $f[k] = 1^k = 1$

Since $1^k = (e^{j\Omega})^k$ with $\Omega = 0$, the amplitude response is $H[e^{j0}]$. From Eq. (12.8a) we obtain

$$H[e^{j0}] = \frac{1}{\sqrt{1.64 - 1.6\cos(0)}} = \frac{1}{\sqrt{0.04}} = 5 = 5\angle 0$$

Therefore

$$|H[e^{j0}]| = 5 \quad \text{and} \quad \angle H[e^{j0}] = 0$$

These values also can be read directly from Figs. 12.1a and 12.1b, respectively, corresponding to $\Omega = 0$. Therefore, the system response to input 1 is

$$y[k] = 5(1^k) = 5 \qquad (12.10)$$

(b) $f[k] = \cos[\frac{\pi}{6}k - 0.2]$

Here $\Omega = \frac{\pi}{6}$. According to Eqs. (12.8)

$$|H[e^{j\pi/6}]| = \frac{1}{\sqrt{1.64 - 1.6\cos\frac{\pi}{6}}} = 1.983$$

$$\angle H[e^{j\pi/6}] = -\tan^{-1}\left[\frac{0.8\sin\frac{\pi}{6}}{1 - 0.8\cos\frac{\pi}{6}}\right] = -0.916 \text{ rad.}$$

These values also can be read directly from Figs. 12.1a and 12.1b, respectively, corresponding to $\Omega = \frac{\pi}{6}$. Therefore

Next Time…

- **Digital Windows**
- Review:
  - Chapter 12 of Lathi
- A signal has many signals ☺
  [Unless it's bandlimited. Then there is the one ω]

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 …