Signals, Instruments, and Systems – W5

Introduction to Signal Processing – Digital Filters, Order and Type of Filters

Outline

- Digital filters in time and frequency domains
- FIR and IIR filters
- Filter order and type

Digital Filters

Z-Transform

$$X(z) = Z\{x[n]\} = \sum_{n=-\infty}^{\infty} x[n]z^{-n}$$

$$z = Ae^{i\phi} \text{ or } z = A(\cos\phi + i\sin\phi)$$

Transform in the complex z-plane, see s. 16, W4

Translation (time-shifting) property:

$$Z\{x[n-n_0]\} = z^{-n_0}\mathrm{X(z)}$$

See also tables Lab 3

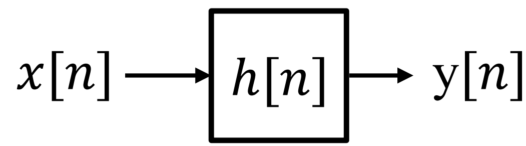
Example:

$$Z\{x[n-1]\} = z^{-1}\mathrm{X(z)}$$

General Representation (Causal Filters)

Difference equation:

$$y[n] = -\sum_{k=1}^{N} a_k y[n-k] + \sum_{k=0}^{M} b_k x[n-k]$$



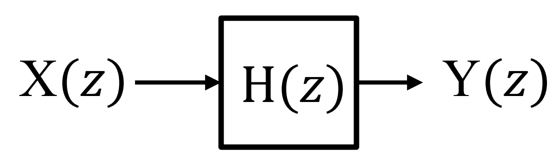
Z-transform:

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

Z-transform properties (s. 4)

Transfer function:

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



An Example

Difference equation:

$$y[n] = x[n] + 2x[n-1] + x[n-2] - \frac{1}{4}y[n-1] + \frac{3}{8}y[n-2]$$

Z-transform:

$$Y(z) + \frac{1}{4}Y(z)z^{-1} - \frac{3}{8}Y(z)z^{-2} = X(z) + 2X(z)z^{-1} + X(z)z^{-2}$$

Transfer function:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1 + 2z^{-1} + z^{-2}}{1 + \frac{1}{4}z^{-1} - \frac{3}{8}z^{-2}} = \frac{z^2 + 2z + 1}{z^2 + \frac{1}{4}z - \frac{3}{8}} = \frac{(z+1)^2}{\left(z - \frac{1}{2}\right)\left(z + \frac{3}{4}\right)}$$

Note: poles and zeros of H(z) can be represented in the zero-pole plot (complex z-plane) for further analysis (e.g., stability).

FIR Filters

Nonrecursive Digital Filters

- $a_k = 0$ for all k
- $y[n] = \sum_{k=0}^{M} b_k \, x[n-k]$
- Finite Impulse Response (FIR)
- The FIR filter above is a causal system: the output depends only on past values of inputs
- The filter coefficients b_k define the FIR filter
- The filter order is M, the number of coefficients b_k is M+1 (filter "length")

Examples of FIR Filters

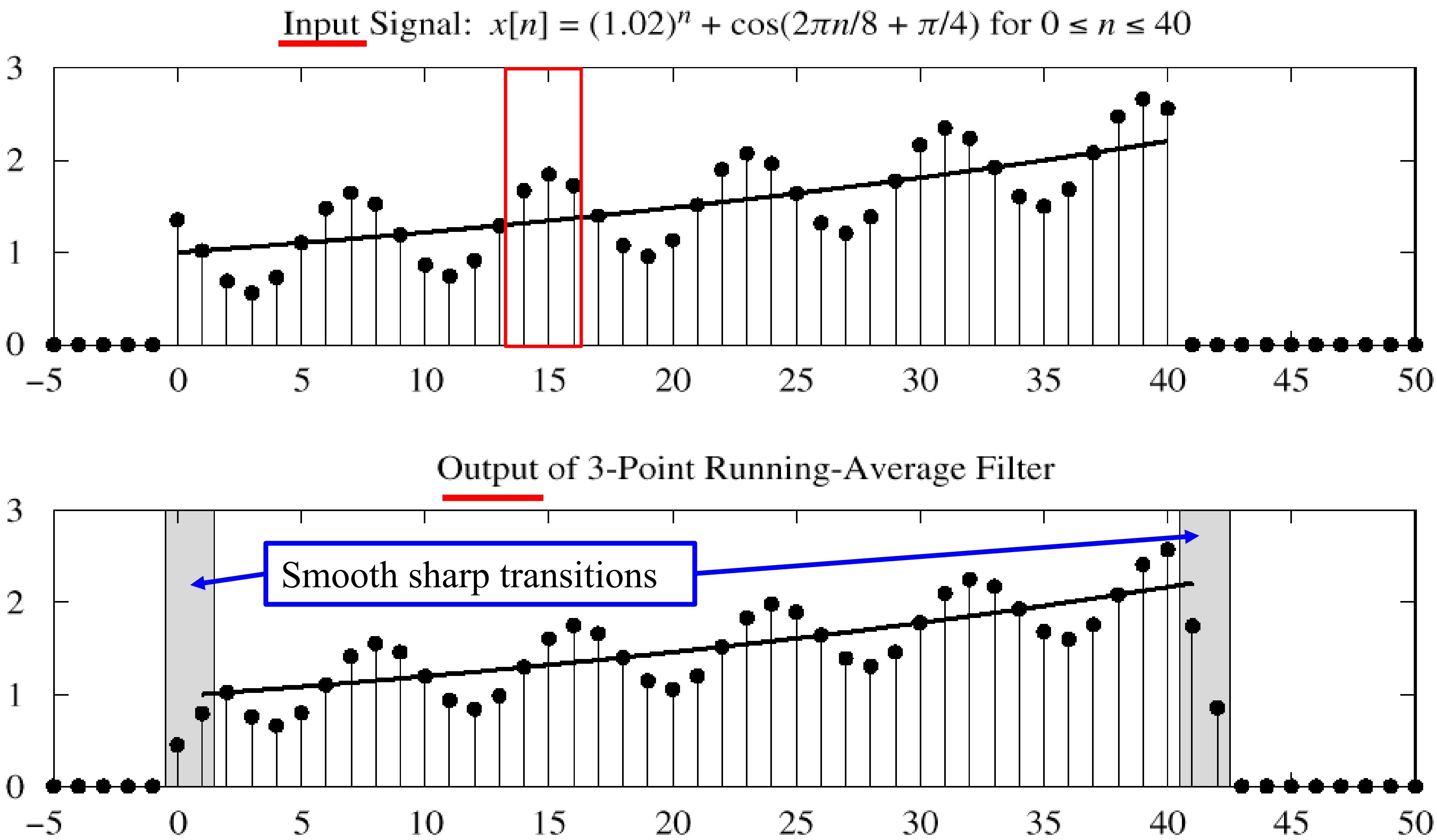
- 3-point moving average:

$$y_3[n] = \sum_{k=0}^{2} \frac{1}{3} x[n-k] = \frac{1}{3} x[n] + \frac{1}{3} x[n-1] + \frac{1}{3} x[n-2]$$

- 7-point moving average:

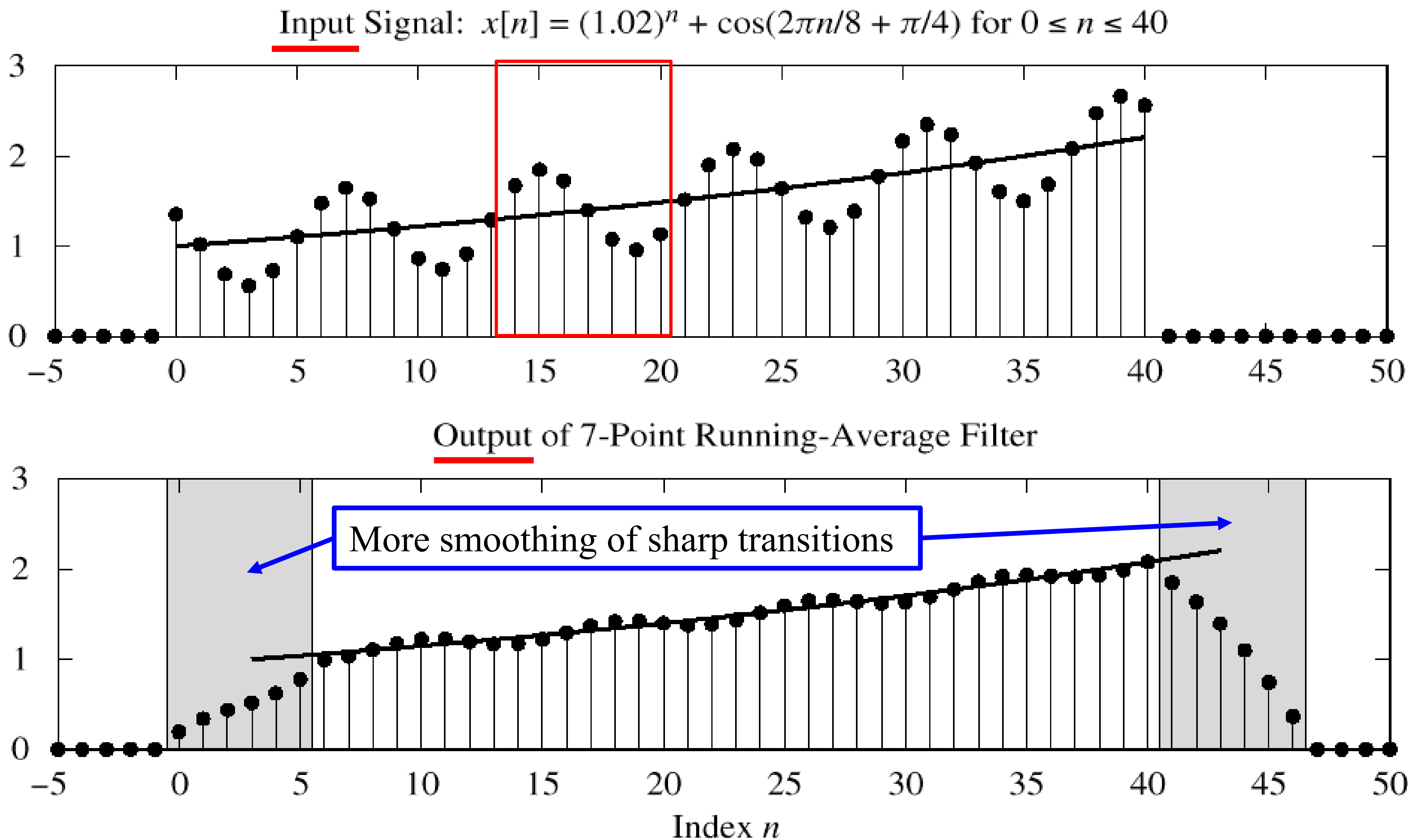
$$y_7[n] = \sum_{k=0}^{6} \frac{1}{7} x[n-k] = \frac{1}{7} x[n] + \frac{1}{7} x[n-1] + \cdots + \frac{1}{7} x[n-6]$$

3-pt Moving Average Example



[Picture from McClellan, Schafer, and Yoder, "DSP First: A Multimedia Approach"]

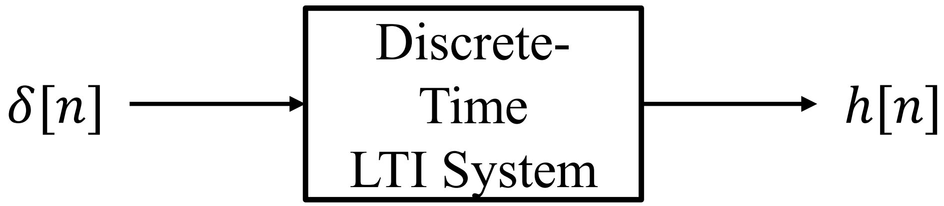
7-pt Moving Average Example

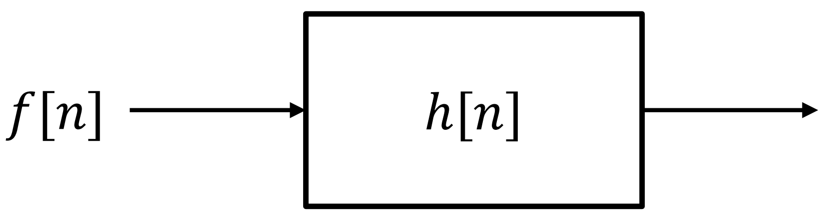


[Picture from McClellan, Schafer, and Yoder, "DSP First: A Multimedia Approach"]

Impulse Response of a Discrete-Time LTI System

From W3, s. 14-15



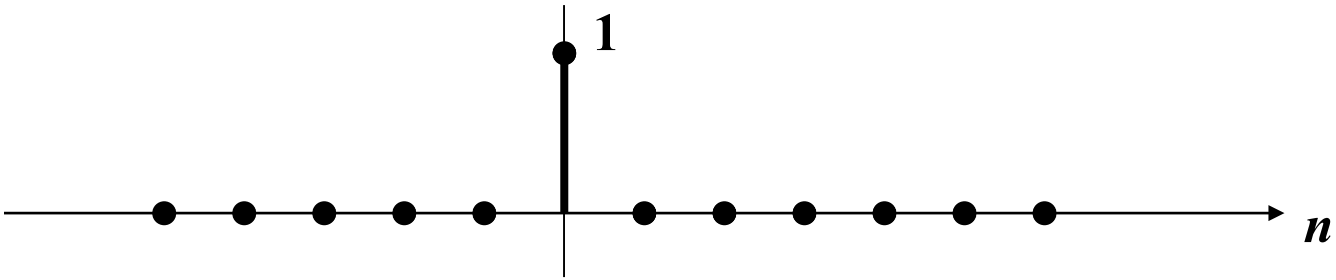


$$(f * h)[n] = \sum_{m=-\infty}^{\infty} f[m]h[n-m]$$

The Unit Impulse

$$\delta[n] = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases}$$

Notation:
Kronecker delta function

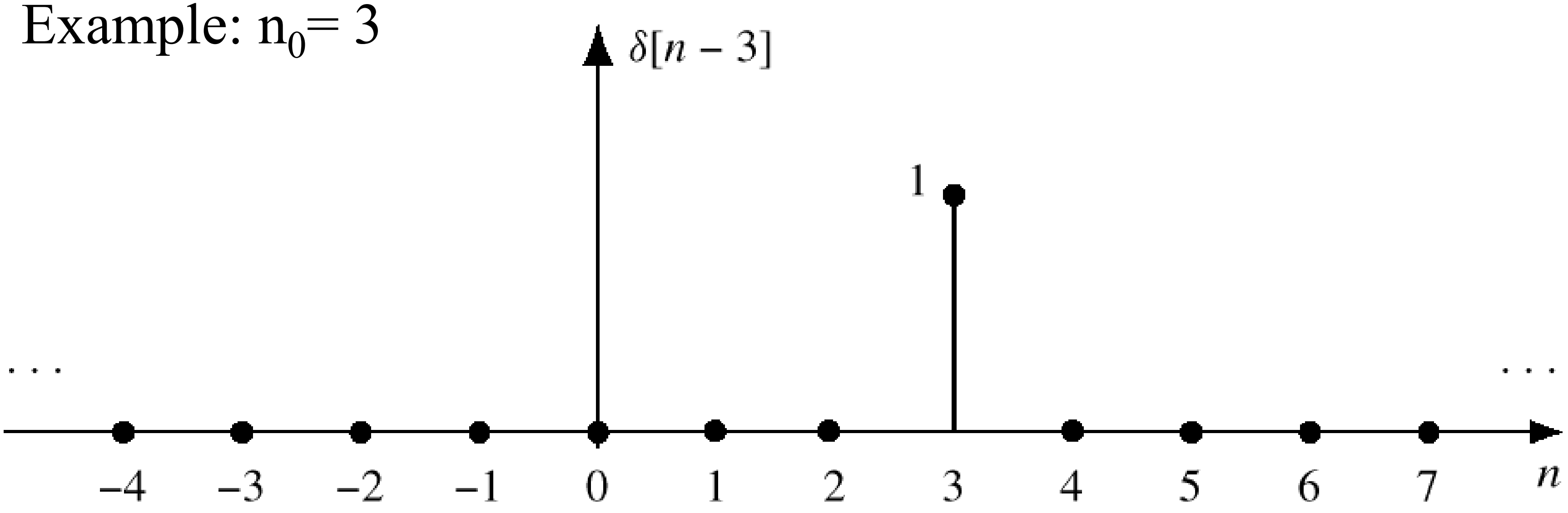


[Picture from McClellan, Schafer, and Yoder, "DSP First: A Multimedia Approach"]

Time-Shifted Unit Impulse

$$\delta[n - n_0] = \begin{cases} 1 & n = n_0 \\ 0 & n \neq n_0 \end{cases}$$

Example: $n_0 = 3$



[Picture from McClellan, Schafer, and Yoder, "DSP First: A Multimedia Approach"]

Impulse Response of a FIR Filter

$$y[n] = \sum_{k=0}^{M} b_k x[n-k]$$

A FIR filter usually described in terms of coefficients b_k

Alternatively, as any other LTI system, we can describe a FIR filter using its impulse response.

If $x[n] = \delta[n]$ then $y[n] = h[n]$, i.e. the impulse response

$$\Longrightarrow \quad \boxed{h[n] = \sum_{k=0}^{M} b_k \delta[n-k]}$$

Note: you can check $(x * h)[n] = y[n]$

Impulse Response of a FIR Filter

$$h[n] = \sum_{k=0}^{M} b_k \delta[n-k]$$

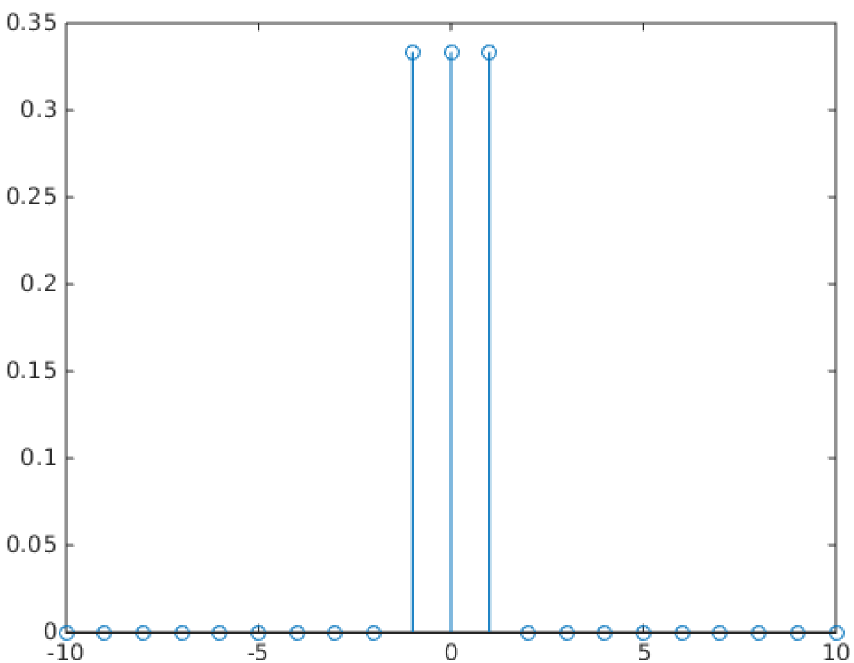
n	$n<0$	0	1	2	3	$\ldots$	M	$M+1$	$n>M+1$
$x[n]=\delta[n]$	0	1	0	0	0	0	0	0	0
$y[n]=h[n]$	0	b_0	b_1	b_2	b_3	$\ldots$	b_M	0	0

Impulse response is finite!

[Table from McClellan, Schafer, and Yoder, "DSP First: A Multimedia Approach"]

Example of FIR Filter: Noncausal 3-Point Moving Average

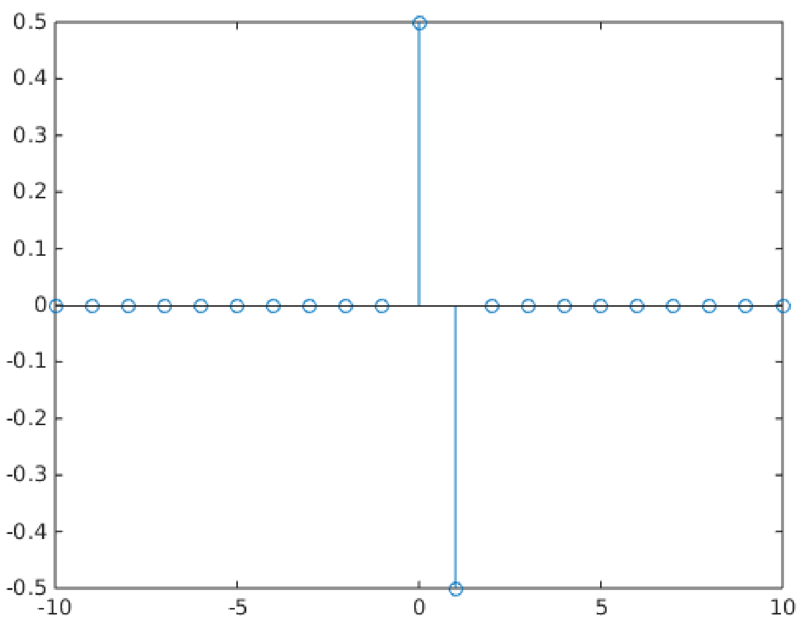
- $b_k = \{\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\}$ for k= -1,0,1 **Filter coefficients**
- $y[n] = \frac{1}{3}(x[n-1] + x[n] + x[n+1])$ **Difference equation**
- $h[n] = \frac{1}{3}(\delta[n-1] + \delta[n] + \delta[n+1])$ **Impulse response**



[Picture from Prof. A. S. Willsky, Signals and Systems course]

Example of FIR Filter: Causal High-Pass Filter

- $b_k = \{\frac{1}{2}, -\frac{1}{2}\}$ for k= 0,1 **Filter coefficients**
- $y[n] = \frac{1}{2}(x[n] - x[n-1])$ **Difference equation**
- $h[n] = \frac{1}{2}(\delta[n] - \delta\,[n-1])$ **Impulse response**



[Picture from Prof. A. S. Willsky, Signals and Systems course]

IIR Filters

Motivation

Original Image

Blurred (Motion)

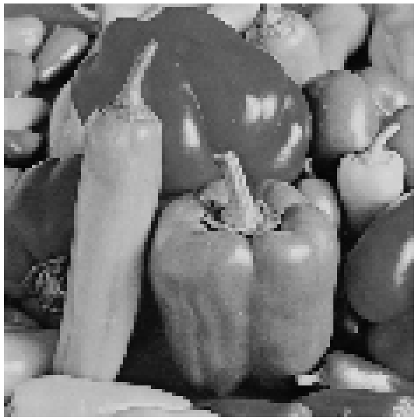
Restored w/ Inverse Filter

Can we remove the blur in postprocessing?
Yes! With a deconvolution filter!

[Picture from McClellan, Schafer, and Yoder, "DSP First: A Multimedia Approach"]

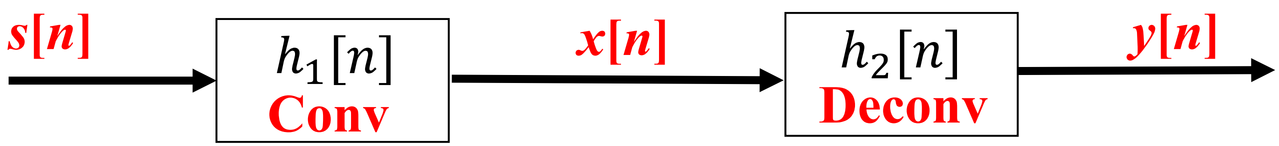
Motivation

Original Image



Blurred (Motion)

Restored w/ Inverse Filter



Given $h_1[n]$, can we find $h_2[n]$ so that $y[n] = s[n]$?

$$x[n] = s[n] * h_1[n]$$
$$y[n] = x[n] * h_2[n] = s[n] * h_1[n] * h_2[n]$$
$$\Rightarrow h_1[n] * h_2[n] = \delta[n]$$ (convolution with unit impulse $\leftrightarrow$ identity)

[Adapted from McClellan, Schafer, and Yoder, “DSP First: A Multimedia Approach”]

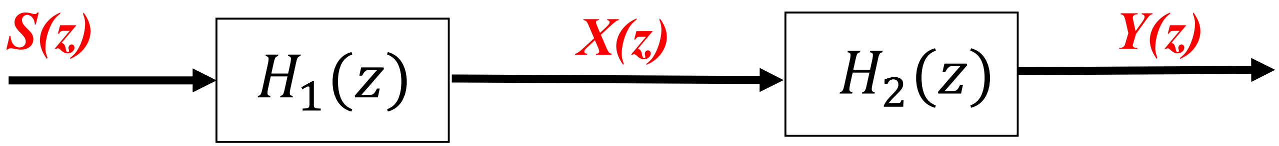
Motivation

Blurring filter with parameter a:

$$x[n] = s[n] - as[n-1] \Rightarrow h_1[n] = \delta[n] - a\delta[n-1]$$

Difficult to solve in time domain within convolution sum.

Z-domain:



$$Y(z) = H_2(z)H_1(z)S(z) = H(z)S(z)$$
$$H(z) = 1 = H_2(z)H_1(z)$$
$$\Rightarrow H_2(z) = 1/H_1(z)$$

Blurring filter in z-domain (from DE above): $H_1(z) = 1 - az^{-1}$

$$\Rightarrow H_2(z) = \frac{1}{1 - az^{-1}}$$

[Adapted from McClellan, Schafer, and Yoder, "DSP First: A Multimedia Approach"]

Motivation

$$H_2(z) = \frac{1}{1 - az^{-1}}$$

What type of filter?

$$H_2(z) = \frac{Y(z)}{X(z)} = \frac{1}{1 - az^{-1}}$$

$$Y(z) = aY(z)z^{-1} + X(z)$$

$$y[n] = ay[n-1] + x[n]$$

If a $\neq 0$ not a FIR (see s. 8)!

In fact H_2 is a first order IIR filter!

Recursive Digital Filters

- $y[n] = -\sum_{k=1}^{N} a_k\, y[n-k] + \sum_{k=0}^{M} b_k\, x[n-k]$
- $a_k \neq 0$ and $b_k \neq 0$ at least for one k
- a_k: feedback coefficients
- b_k: feed-forward coefficients
- Infinite Impulse Response (IIR)
- The IIR filter above is a causal system: the output depends only on past values of output and input
- The filter order is typically N and the total number of coefficients is N+M+1

Impulse Response of a IIR Filter

Example: generalized first order IIR filter:

$$y[n] = a_1 y[n-1] + b_0 x[n] \Rightarrow h[n] = a_1 h[n-1] + b_0 \delta[n]$$

n	$n<0$	0	1	2	3	4	$\cdots$
$\delta[n]$	0	1	0	0	0	0	$\cdots$
$h[n-1]$	0	0	b_0	$b_0(a_1)$	$b_0(a_1)^2$	$b_0(a_1)^3$	$\cdots$
$h[n]$	0	b_0	$b_0(a_1)$	$b_0(a_1)^2$	$b_0(a_1)^3$	$b_0(a_1)^4$	$\cdots$

Impulse response is infinite!

Frequency Response of Digital Filters

Motivation

- Typical configuration when digital filtering is applied to analog signals:

- Frequency response useful when input x[n] = sum of sinusoids, then output y[n] also sum of sinusoids
- Focus on the spectrum

Frequency Response

- Bode plots have been defined for continuous-time systems (e.g., analog filters)
- With z $= e^{i\omega}$ (see W4, s. 20), the Z-transform degrades to a DTFT (as the Laplace transform was degrading to the FT with s $= i\omega$)
- We can therefore calculate the frequency response of the corresponding discrete-time system (e.g., a digital filter) with the transfer function $H(e^{i\omega})$
- A FFT is then typically used to calculate numerically the frequency response on computers
- Matlab has a dedicated function for this: freqz

  Note: response in normalized angular frequency

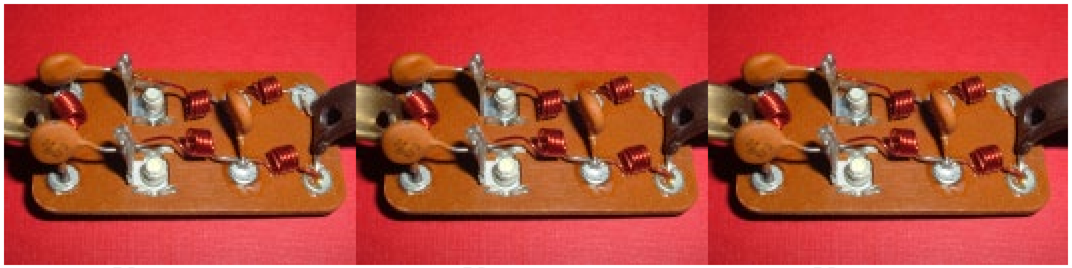
Order and Types of Filters

Filter Order and Type

- Several filters exist in both analog and digital form and are defined by the polynomials at the numerator/denominator (Bessel, Butterworth, Tschebishev, etc.)
- 1st order is equivalent to 20dB per decade
- Each successive order adds 20dB per decade
- Filter with a high order are closer to the ideal filter (rectangular function)

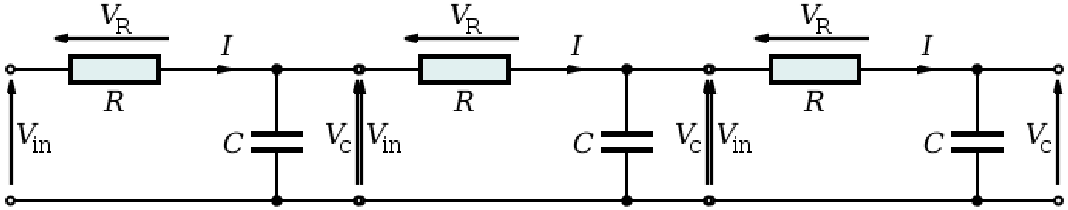
Filter Order

Analog



…

Filter order: 3
++ faster cutoff
-- more components
-- higher power
consumption



…

Digital

$$y[n] = b_0 x[n]$$

$$y[n] = b_0 x[n] + b_1 x[n-1]$$

$$y[n] = b_0 x[n] + b_1 x[n-1] + b_2 x[n-2]$$

…

Filter order: 0
Filter order: 1
Filter order: 2

++ faster cutoff
-- more computation
-- higher power
consumption

Conclusion

Take Home Messages

- Both analog and digital filters are:
  - characterized by different order and coefficient distributions
  - they can be expressed in time and (complex) frequency domains
- Programmable digital components (e.g., microcontrollers, DSPs) allow for easy encoding of digital filters
- Several equivalent forms to define digital filters are possible:
  - Coefficient set (time domain)
  - Difference equation (time domain)
  - Impulse response (time domain)
  - Transfer function (frequency domain)
- An important differentiation for digital filters can be based on the impulse response which can be finite (filter non recursive) or infinite (filter recursive); FIR and IIR filters are correspondingly defined

Additional Literature – Week 5

Books

- J. H. McClellan, R. W. Schafer, M. A. Yoder “DSP First: A Multimedia Approach”, Prentice Hall, 1999.
- A. Oppenheim and A. S. Willsky with S. Nawab, “Signals and Systems”, Prentice Hall, 1997.