STANFORD UNIVERSITY
EE 102B Spring-2013

Lecture 06

LTI Systems and Discrete Convolution

April 12, 2013

ASSIGNMENTS

- Reading for this Lecture:
  - SPF: Chapter 5, Sections 6.1 and 6.2
  - S&S: Sections 2.1 and 3.2
- HW#02 is posted. It is due by 5pm on Weds., April 17 in Packard 263. Late papers accepted until Fri., 04-19 at 5pm. 10% off/day late.
- Lab #02 is posted. It is due by 5pm, Friday, April 19, in Packard 263. Late papers accepted until Mon., 04-22 at 11am. Penalty - 15%.

Office Hours for Course Staff – Come see us.

- Ron Schafer: Mon. 2:00-3:00 pm and Weds. 4:00-5:00 pm in Packard 211
- Dookun Park: Tues. 1:30 ~ 3:30 pm in Packard 107
- Ruo Yu Gu (Roy): Mon. 4:00 ~ 6:00 pm in Packard 106
- https://class2go.stanford.edu/EE102B/Spring2013/pages/staff

LECTURE OBJECTIVES

- Review finite-duration impulse response filters
- LTI discrete-time systems
- Discrete convolution
- Lab 02: Getting started

GENERAL CAUSAL FIR FILTER

- FILTER COEFFICIENTS $\{b_k\}$

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

- FILTER **ORDER** is M
- FILTER **"LENGTH"** is L = M+1
  - NUMBER of FILTER COEFFS is L

Causal Discrete-Time Systems

- A causal system is one whose output at time n depends only on only on present and past samples of the input.

$$y[n] \text{ depends only on } x[m] \text{ for } m \le n$$

- The causal FIR filter is defined by the difference equation

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

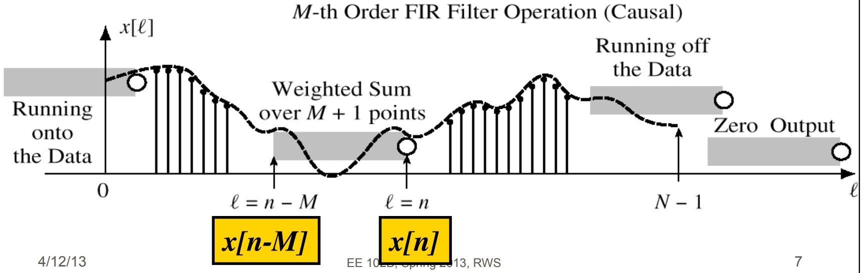
GENERAL CAUSAL FIR FILTER

- SLIDE a WINDOW across x[n]

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

Causal because y[n] depends only on x[n] and past samples

M-th Order FIR Filter Operation (Causal)



FIR IMPULSE RESPONSE

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

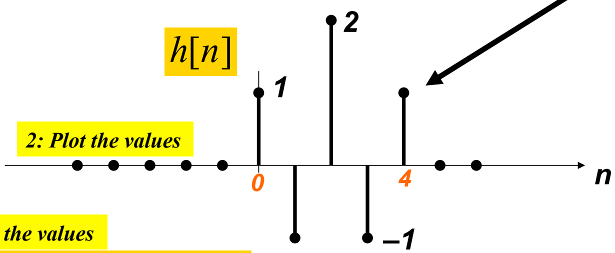
FIR means **F**inite-duration **I**mpulse **R**esponse

n	$n<0$	0	1	2	3	$\dots$	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	$\dots$	b_M	0	0

Three Representations for a FIR filter

1 Use SHIFTED IMPULSES to write $h[n]$

$$h[n] = \delta[n] - \delta[n-1] + 2\delta[n-2] - \delta[n-3] + \delta[n-4]$$



3: List the values

$$b_k = \{1, -1, 2, -1, 1\}$$

Also true for any signal, x[n]

FILTERING EXAMPLE

- *3-point AVERAGER*
  - Changes A slightly
  - Shifts by 1 sample

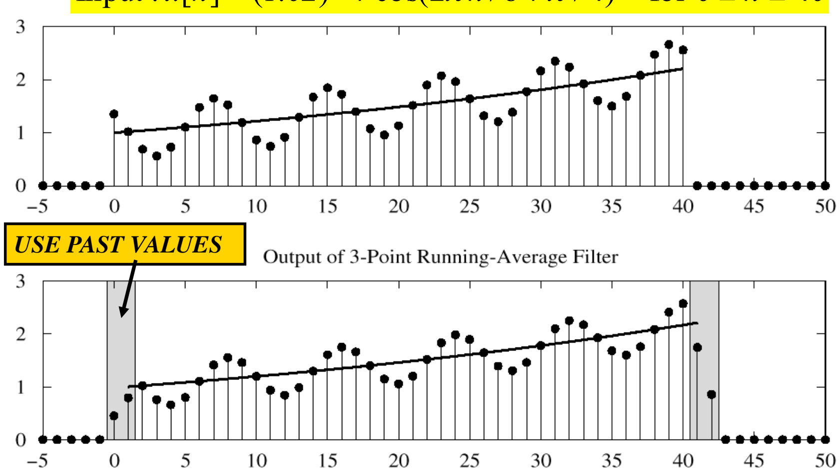
$$y_3[n] = \sum_{k=0}^{2} \left(\tfrac{1}{3}\right) x[n-k]$$

- 7-point AVERAGER
  - Removes cosine
    - By making its amplitude (A) smaller
  - Shifts by 3 samples

$$y_7[n] = \sum_{k=0}^{6} \left(\tfrac{1}{7}\right) x[n-k]$$

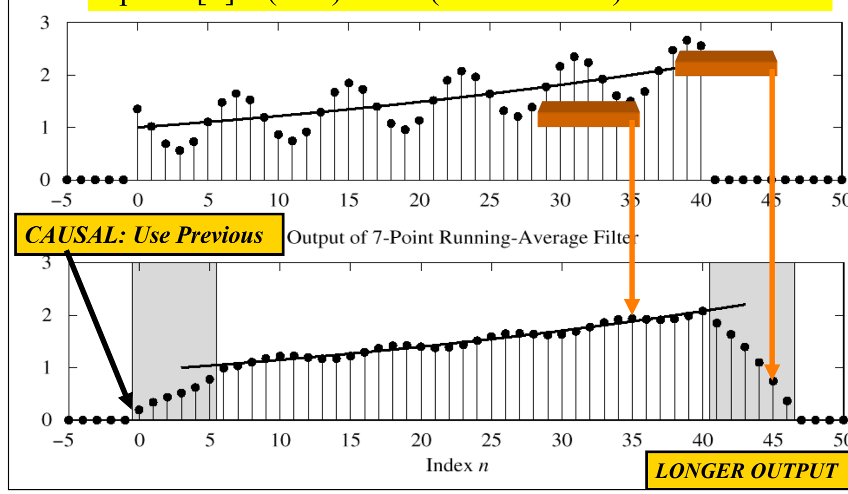
3-pt AVG EXAMPLE

Input : $x[n] = (1.02)^n + \cos(2\pi n/8 + \pi/4) \quad \text{for } 0 \le n \le 40$



7-pt FIR EXAMPLE (AVG)

Input : $x[n] = (1.02)^n + \cos(2\pi n/8 + \pi/4) \quad \text{for } 0 \le n \le 40$



POP QUIZ

- FIR Filter is “FIRST DIFFERENCE”
  - $y[n] = x[n] - x[n-1]$
- What are the filter coefficients?

$$b_k = \{1, -1\}$$

- Find ***h***[***n***]

$$h[n] = \delta[n] - \delta[n-1]$$

Discrete-Time Linear Systems

- A linear system obeys the principle of superposition

$$x[n] \mapsto y[n]$$

$$\alpha x[n] \mapsto \alpha y[n] \quad \text{(homogeneous)}$$

$$x_1[n] \mapsto y_1[n] \text{ and } x_2[n] \mapsto y_2[n]$$

$$x_1[n] + x_2[n] \mapsto y_1[n] + y_2[n] \quad \text{(additive)}$$

$$\alpha x_1[n] + \beta x_2[n] \mapsto \alpha y_1[n] + \beta y_2[n] \quad \text{(superposition)}$$

TESTING LINEARITY

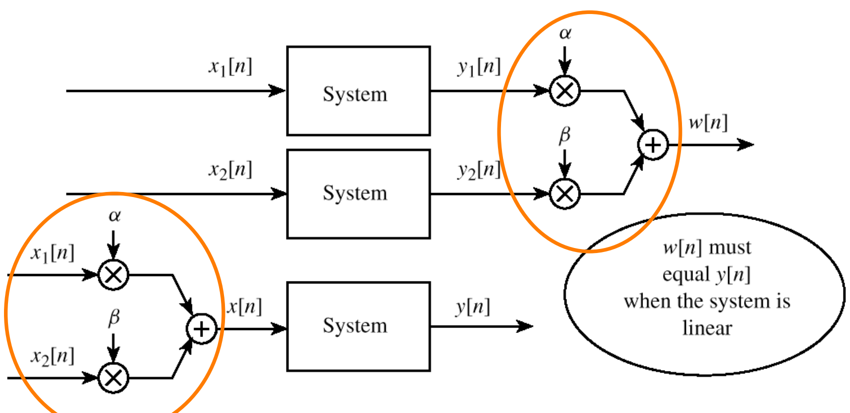


Figure 5.17 Testing linearity by checking the interchange of operations.

Time-Invariant Systems

- A time-invariant system is one whose output is the same for a given input no matter when the input occurs.

$$x[n] \mapsto y[n]$$

$$x[n - n_0] \mapsto y[n - n_0]$$

- An LTI discrete-time system is both linear and time-invariant.

TESTING Time-Invariance

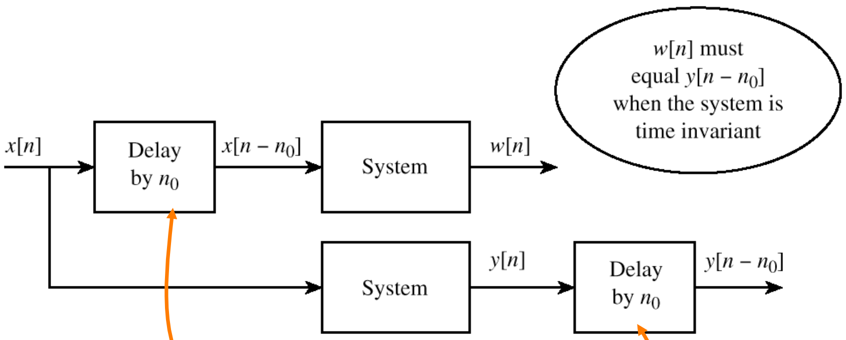


Figure 5.16 Testing time-invariance property by checking the interchange of operations.

Derivation of Discrete Convolution

- Represent x[n] in terms of impulses

$$x[n]=\sum_{k=-\infty}^{\infty} x[k]\delta[n-k]$$

- Assume LTI system

$$\delta[n]\mapsto h[n]$$

$$\delta[n-k]\mapsto h[n-k]$$

$$x[k]\delta[n-k]\mapsto x[k]h[n-k]$$

$$x[n]=\sum_{k=-\infty}^{\infty} x[k]\delta[n-k]\mapsto \sum_{k=-\infty}^{\infty} x[k]h[n-k]=y[n]$$

Discrete Convolution

- LTI systems defined by convolution. The limits can be infinite.

$$y[n]=\sum_{k=-\infty}^{\infty} x[k]h[n-k]=x[n]*h[n]=(x*h)[n]$$

- Discrete convolution is commutative

$$y[n]=\sum_{k=-\infty}^{\infty} h[k]x[n-k]=h[n]*x[n]=(h*x)[n]$$

Discrete Convolution and FIR Systems

- LTI systems defined by convolution

$$y[n]=\sum_{k=-\infty}^{\infty} h[k]x[n-k]=h[n]*x[n]=(h*x)[n]$$

- FIR system has finite-duration impulse response $h[n]=\begin{cases} b_n & n=0,1,\ldots,M \\ 0 & \text{otherwise}\end{cases}$

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

CONVOLUTION Example

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

$$h[n] = \delta[n] - \delta[n-1] + 2\delta[n-2] - \delta[n-3] + \delta[n-4]$$

$$x[n] = u[n]$$

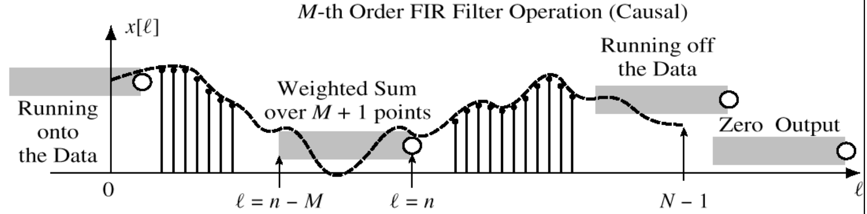
n	-1	0	1	2	3	4	5	6	7
$x[n]$	0	1	1	1	1	1	1	1	...
$h[n]$	0	1	-1	2	-1	1	0	0	0
$h[0]x[n]$	0	1	1	1	1	1	1	1	1
$h[1]x[n-1]$	0	0	-1	-1	-1	-1	-1	-1	-1
$h[2]x[n-2]$	0	0	0	2	2	2	2	2	2
$h[3]x[n-3]$	0	0	0	0	-1	-1	-1	-1	-1
$h[4]x[n-4]$	0	0	0	0	0	1	1	1	1
$y[n]$	0	1	0	2	1	2	2	2	...

GENERAL FIR FILTER

- SLIDE a Length-L WINDOW over x[n]

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

M-th Order FIR Filter Operation (Causal)



x[n-M] **x[n]**

MATLAB for FIR FILTER

- `yy = conv(bb,xx)`
  - VECTOR **bb** contains Filter Coefficients
  - SP-First: `yy = firfilt(bb,xx)`
- FILTER COEFFICIENTS $\{b_k\}$

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

conv2() *for images*

What system does this implement?

- `yy = conv([1,2,1],xx)`
- How about this one?

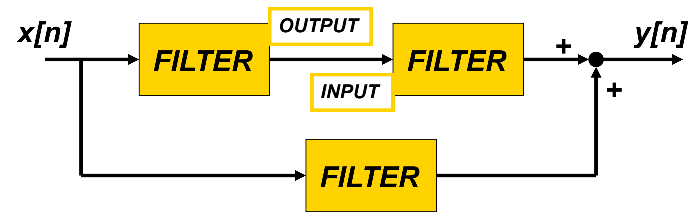
`yy =conv([1,2,1],conv([1,-1],xx))`

HARDWARE STRUCTURES

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

- INTERNAL STRUCTURE of “FILTER”
  - WHAT COMPONENTS ARE NEEDED?
  - HOW DO WE “HOOK” THEM TOGETHER?
- SIGNAL FLOW GRAPH NOTATION

BUILDING BLOCKS

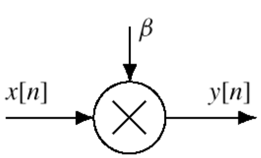


- BUILD UP COMPLICATED FILTERS
  - FROM SIMPLE MODULES
  - Ex: FILTER MODULE MIGHT BE 3-pt FIR

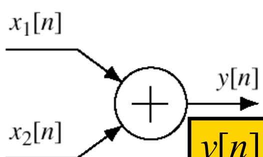
HARDWARE ATOMS

- Add, Multiply & Store

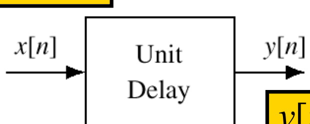
$$y[n] = \sum_{k=0}^{M} b_k x[n-k]$$



$$y[n] = \beta x[n]$$



$$y[n] = x_1[n] + x_2[n]$$



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

FIR STRUCTURE

- Direct Form

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

SIGNAL FLOW GRAPH

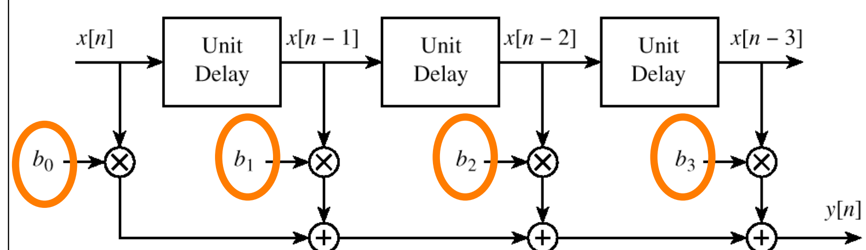


Figure 5.13 Block-diagram structure for the Mth order FIR filter.

Warmup for Lab #02

DCONVDEMO: MATLAB GUI