ECE-2025 Fall-2009

Lecture 12
Linearity, Time-Invariance
and Convolution
2-Oct-2009

Lab & HW Info

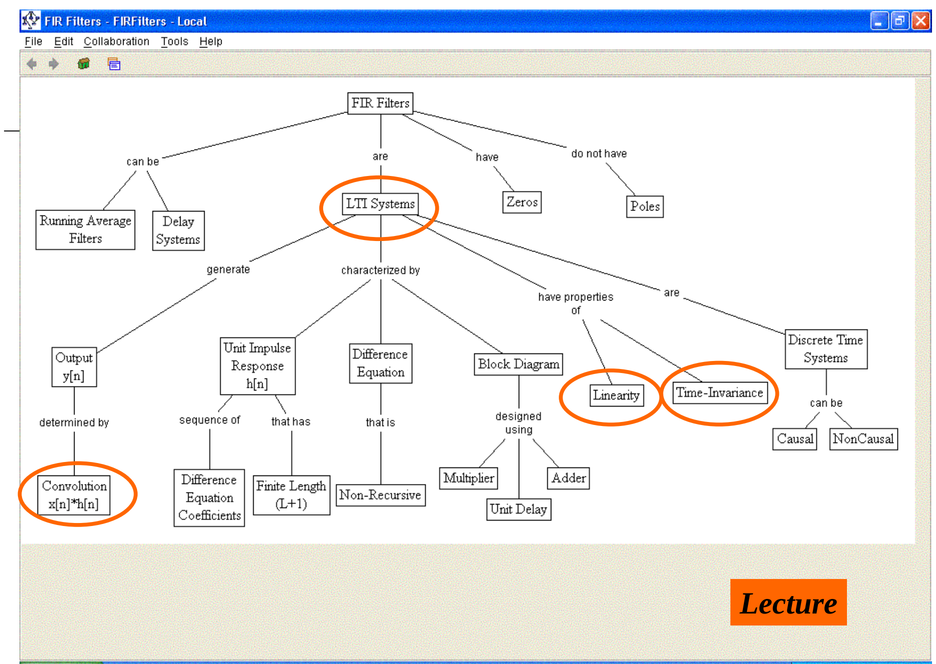
- Lab #5
  - **Formal** Lab Report (150 pts)
    - Due starting on 13-Oct (Tues)
    - Listening Test when report is turned in
- Recitations during week of 5-Oct
  - L01, L03 will have Recitation in Klaus-2440 on Wednesday
  - L05, L07, L09, L11 will have Rec in K-2440 on Thursday
- HW #6 due next week in Rec on 7 or 8-Oct
  - HW #7 due week of 12-Oct
- Quiz #2 will be 23-Oct

Lecture

Switching Lab Partners

- Choose a new partner
  - Must **change** partners
  - Starts with Lab #7 (week of 13-Oct)
- Fill out on-line form no later than Friday, 9-Oct
  - Go to **Quizzes/Tests** on t-square
  - Both partners must fill out the form
  - Both partners must be in the **same** lab section
- No chosen partner?
  - Then assignments will be made at random.

Lecture



Lecture

READING ASSIGNMENTS

- This Lecture:
  - Chapter 5, Sections 5-5 and 5-6
    - Section 5-4 will be covered, but not “in depth”
- Other Reading:
  - Recitation: Ch. 5, Sects 5-6, 5-7 & 5-8
    - CONVOLUTION
  - Next Lecture: start Chapter 6

LECTURE OBJECTIVES

- GENERAL PROPERTIES of FILTERS
  - LINEARITY
  - TIME-INVARIANCE
  - ==> **CONVOLUTION**

LTI SYSTEMS

- BLOCK DIAGRAM REPRESENTATION
  - Components for Hardware
  - Connect Simple Filters Together to Build More Complicated Systems

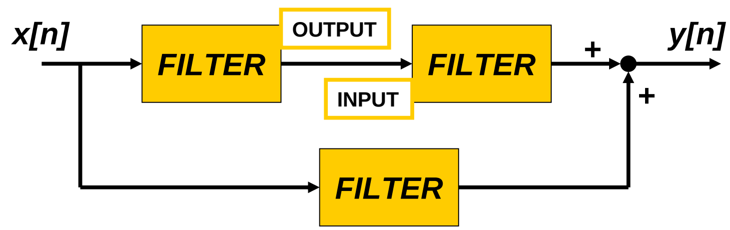
OVERVIEW

- IMPULSE RESPONSE, $h[n]$
  - FIR case: same as $\{b_k\}$
- CONVOLUTION
  - GENERAL: $y[n] = h[n] * x[n]$
  - GENERAL CLASS of SYSTEMS
  - LINEAR and TIME-INVARIANT
- ALL **LTI** systems have $h[n]$ & use convolution

DIGITAL FILTERING

- CONCENTRATE on the FILTER (DSP)
- DISCRETE-TIME SIGNALS
  - FUNCTIONS of n, the “time index”
  - INPUT $x[n]$
  - OUTPUT $y[n]$

BUILDING BLOCKS



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

GENERAL FIR FILTER

- FILTER COEFFICIENTS $\{b_k\}$
  - DEFINE THE FILTER

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

  - For example, $b_k = \{3, -1, 2, 1\}$

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

MATLAB for FIR FILTER

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

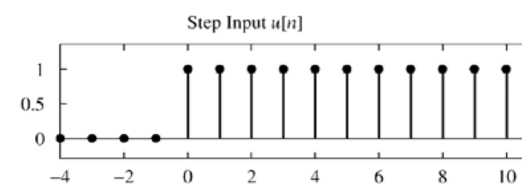
conv2()
for images

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

POP QUIZ

- FIR Filter is "FIRST DIFFERENCE"
  - $y[n] = x[n] - x[n-1]$
- INPUT is "UNIT STEP"

$$u[n] = \begin{cases} 1 & n \geq 0 \\ 0 & n < 0 \end{cases}$$



- Find $y[n]$

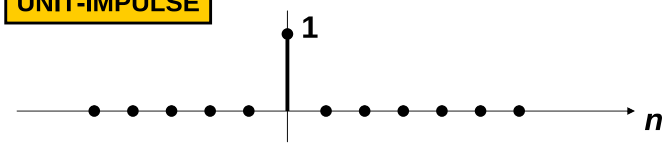
$$y[n] = u[n] - u[n-1] = \delta[n]$$

SPECIAL INPUT SIGNALS

- x[n] = SINUSOID **FREQUENCY RESPONSE**
- x[n] has only one NON-ZERO VALUE

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

UNIT-IMPULSE



UNIT IMPULSE RESPONSE

- FIR filter DIFFERENCE EQUATION is specified by the filter coefficients b_k

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

- EQUIVALENCE: can we describe the filter using a SIGNAL instead?

FIR IMPULSE RESPONSE

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

n	$n<0$	0	1	2	3	...	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	...	b_M	0	0

- Impulse response $h[k]=b_k$ is just a SIGNAL description of filter coefficients
- Allows us to write CONVOLUTION sum

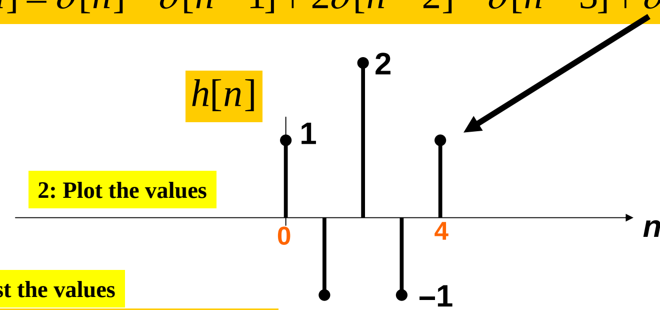
3 Representations for h[n]

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

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

$h[n]$

2: Plot the values



3: List the values

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

True for any signal, x[n]

LTI: Convolution Sum

- **Output = Convolution of x[n] & h[n]**
  - NOTATION: $y[n] = h[n] * x[n]$
  - FIR case:

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

FINITE LIMITS

FINITE LIMITS

Same as $\mathbf{b_k}$

CONVOLUTION Exa

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

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

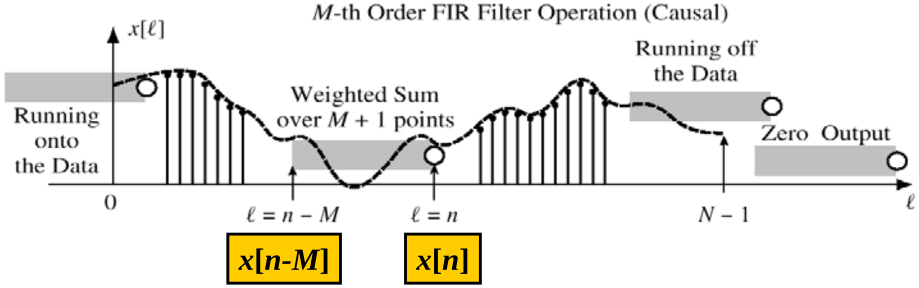
n	−1	0	1	2	3	4	5	6	7
$x[n]$	0	1	1	1	1	1	1	1	...
$h[n]$	0	1	−1	2	−1	1	0	0	0
$h[0]x[n]$	0	1	1	1	1	1	1	1	1
$h[1]x[n-1]$	0	0	−1	−1	−1	−1	−1	−1	−1
$h[2]x[n-2]$	0	0	0	2	2	2	2	2	2
$h[3]x[n-3]$	0	0	0	0	−1	−1	−1	−1	−1
$h[4]x[n-4]$	0	0	0	0	0	1	1	1	1
$y[n]$	0	1	0	2	1	2	2	2	...

GENERAL FIR FILTER

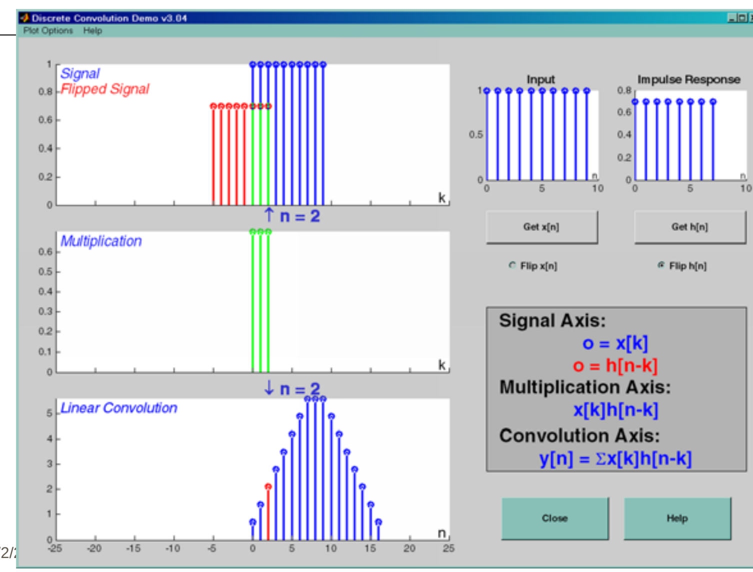
- SLIDE a Length-L WINDOW over x[n]

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

M-th Order FIR Filter Operation (Causal)



DCONVDEMO: MATLAB GUI



HARDWARE STRUCTURES

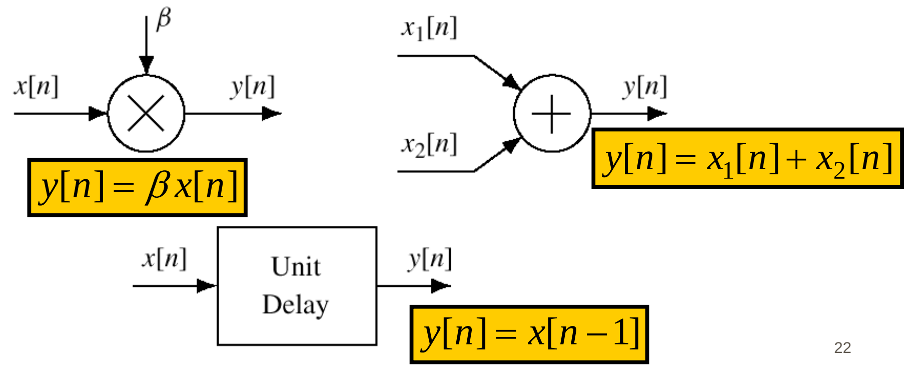
$$y[n] = \sum_{k=0}^{M} b_k x[n-k]$$

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

HARDWARE ATOMS

- Add, Multiply & Store

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



FIR STRUCTURE

- Direct Form

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

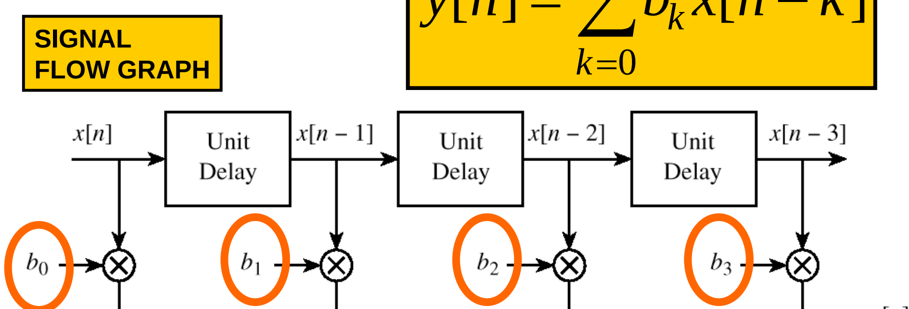
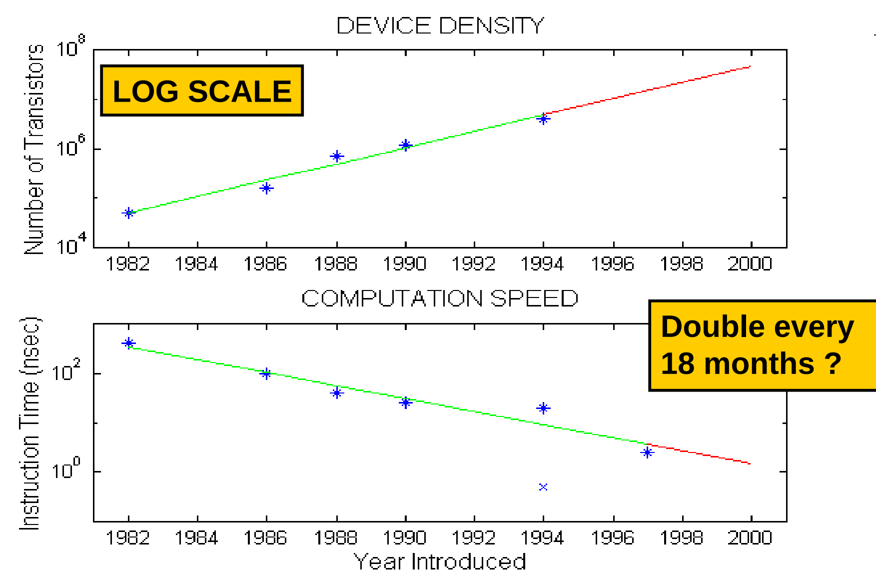


Figure 5.13 Block-diagram structure for the Mth order FIR filter.

Moore's Law for TI DSPs



SYSTEM PROPERTIES

- MATHEMATICAL DESCRIPTION
- TIME-INVARIANCE
- LINEARITY
- CAUSALITY
  - "No output prior to input"

TIME-INVARIANCE

- IDEA:
  - "Time-Shifting the input will cause the same time-shift in the output"
- EQUIVALENTLY,
  - We can prove that
    - The time origin (n=0) is picked arbitrary

TESTING Time-Invariance

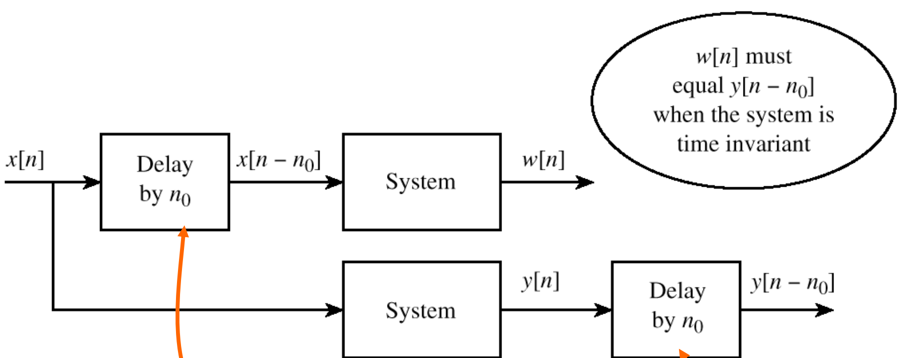


Figure 5.16 Testing time-invariance property by checking the interchange of operations.

LINEAR SYSTEM

- LINEARITY = Two Properties
- SCALING
  - "Doubling x[n] will double y[n]"
- SUPERPOSITION:
  - "Adding two inputs gives an output that is the sum of the individual outputs"

TESTING LINEARITY

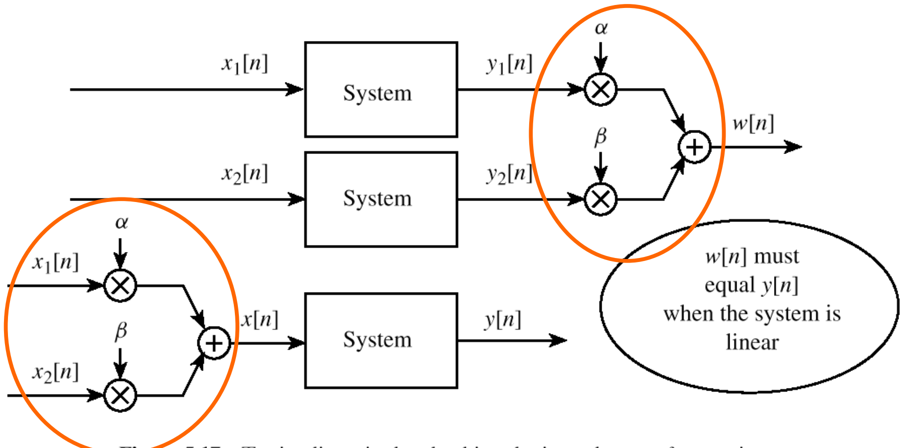


Figure 5.17 Testing linearity by checking the interchange of operations.

LTI SYSTEMS

- LTI: Linear & Time-Invariant
- COMPLETELY CHARACTERIZED by:
  - **IMPULSE RESPONSE** h[n]
  - **CONVOLUTION**: y[n] = x[n]*h[n]
    - The “rule”defining the system can ALWAYS be re-written as convolution
- FIR Example: h[n] is same as b_k

POP QUIZ

- FIR Filter is “FIRST DIFFERENCE”
  - ***y*[*n*] = *x*[*n*] - *x*[*n* -1]**
- Write output as a convolution
  - Need impulse response

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

  - Then, an equivalent way to compute the output:

$$y[n] = h[n] * x[n] = \left(\delta[n] - \delta[n-1]\right) * x[n]$$
$$= \left(\delta[n] * x[n]\right) - \left(\delta[n-1] * x[n]\right)$$
$$= x[n] - x[n-1]$$

CASCADE EQUIVALENT

- Find “overall” h[n] for a cascade ?

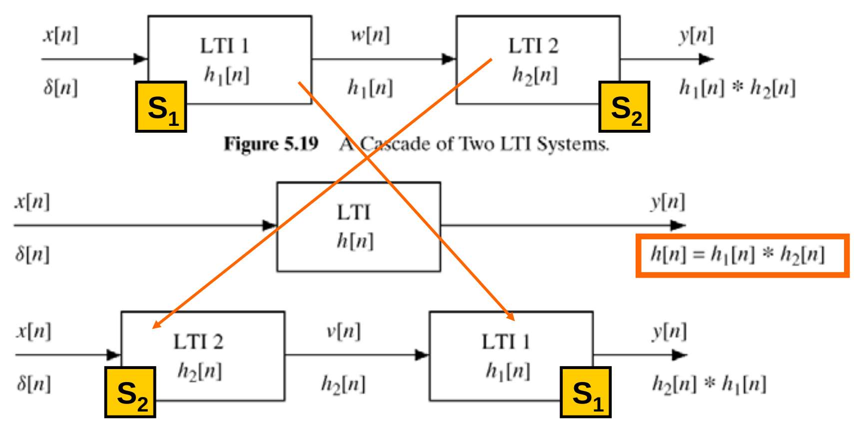


Figure 5.20 Switching the order of cascaded LTI systems.