EE-2025 Fall-99

Lecture 9

Linearity & Time-Invariance

27-Sept-99

Info: Web-CT, Lab, HW

- RWS office hrs: Monday 1:30–3pm
- Survey this Week
  - Lab Quiz next week
- Lab #5: Image Processing
  - FM signals
  - Also Sampling and Reconstruction
- Prob Set #5 due FRIDAY
  - On-Line HW #4 also

READING ASSIGNMENTS

- This Lecture:
  - Chapter 5, pp. 133–152
- Other Reading:
  - Recitation: Ch. 5, pp. 127–133, 142–146
    - CONVOLUTION
  - Next Lecture: Chapter 6, start

DEBUGGING

- "Any Fool" can write code
- Debugging is the interesting part
  - It takes talent !!!
- HOWEVER,
  - Assume the stupid mistake is the problem

DOMAINS: Time & Frequency

- Time–Domain
  - x[n]
  - x(t)
- Frequency Domain
  - Spectrum vs. f (Hz)
  - Spectrum vs. omega–hat
- Move back and forth QUICKLY

LECTURE OBJECTIVES

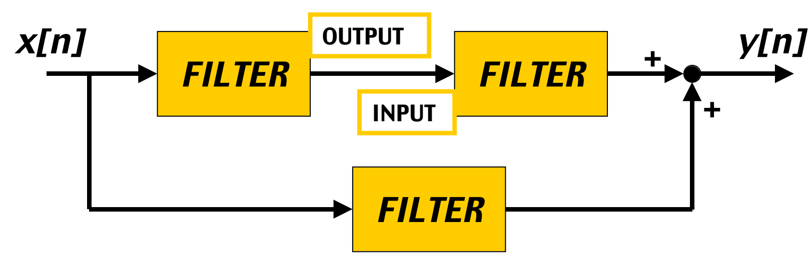
- BLOCK DIAGRAM REPRESENTATION
  - Components for Hardware
  - Connect Simple Filters Together to Build More Complicated Systems
- GENERAL PROPERTIES of FILTERS
  - LINEARITY
  - TIME–INVARIANCE
  - ==> CONVOLUTION

LTI SYSTEMS

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 $\{b_k\}$

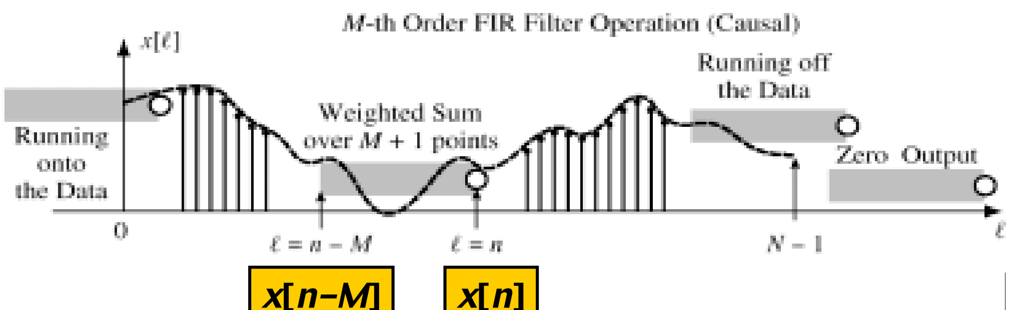
$$y[n] = \sum_{k=0}^{M} b_k\, x[n-k]$$

Conv2( ) *for images*

GENERAL FIR FILTER

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

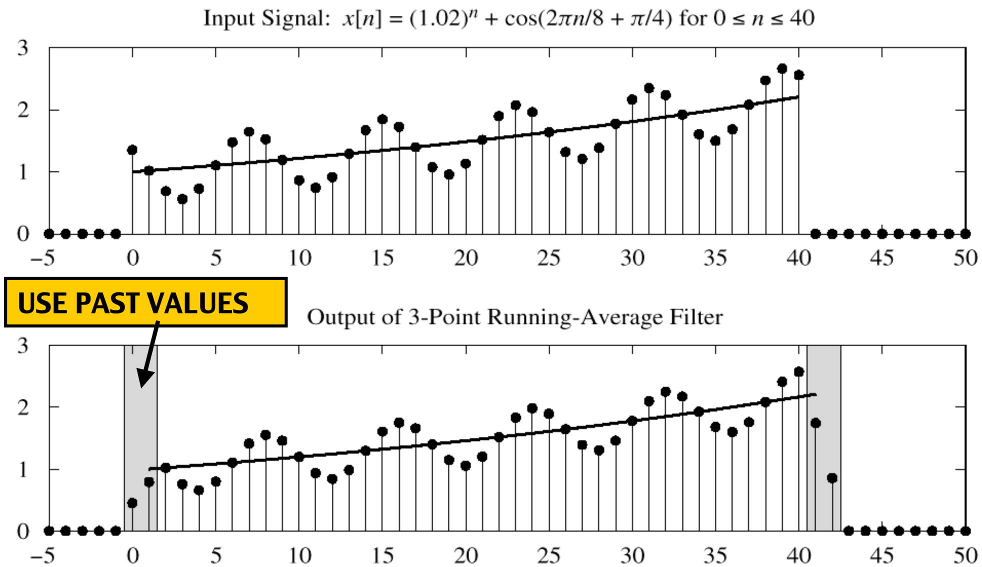
$$y[n] = \sum_{k=0}^{M} b_k\, x[n-k]$$



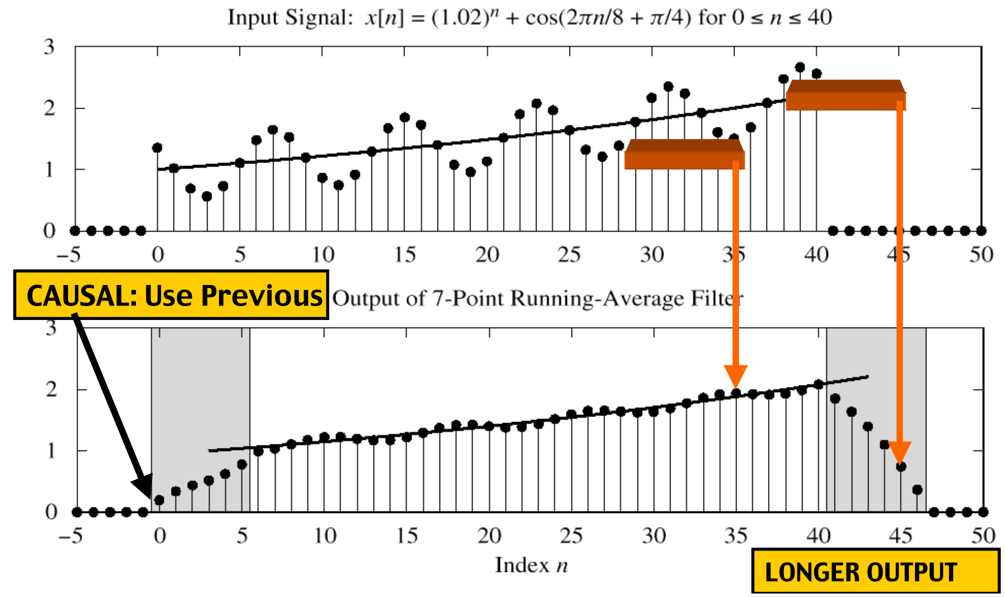
FILTERING EXAMPLE

- 7-point AVERAGER $\quad y_7[n] = \frac{1}{7}\left(\sum_{k=0}^{6} x[n-k]\right)$
  - Removes cosine
    - By making its amplitdue (A) smaller
- 3-point AVERAGER $\quad y_3[n] = \frac{1}{3}\left(\sum_{k=0}^{2} x[n-k]\right)$
  - Changes A slightly

3-pt AVG EXAMPLE



7-pt FIR EXAMPLE (AVG)

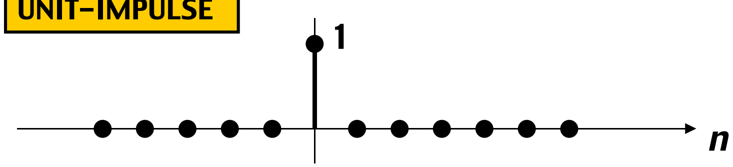


SPECIAL INPUT SIGNALS

- 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 SIGNAL δ[n]

n	...	−2	−1	0	1	2	3	4	5	6	...
$\delta[n]$	0	0	0	1	0	0	0	0	0	0	0
$\delta[n-3]$	0	0	0	0	0	0	1	0	0	0	0

n=3

δ[n] is NON–ZERO When its argument is equal to ZERO

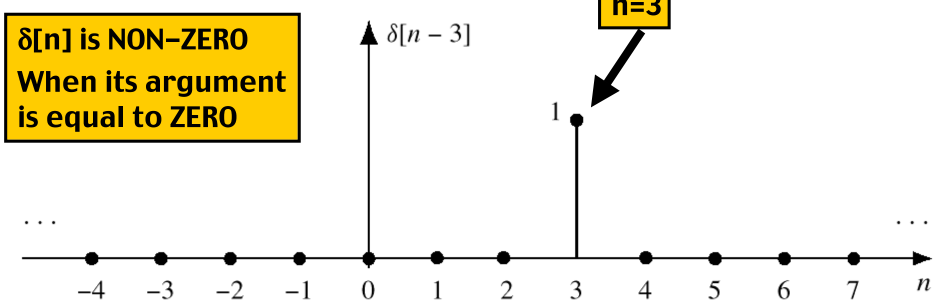
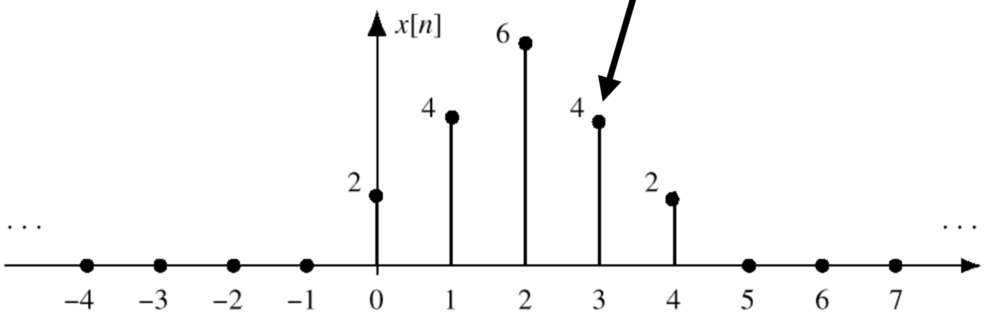


Figure 5.7 Shifted impulse sequence, $\delta[n-3]$.

MATH FORMULA for x[n]

- Use SHIFTED IMPULSES to write x[n]

$$x[n] = 2\delta[n] + 4\delta[n-1] + 6\delta[n-2] + 4\delta[n-3] + 2\delta[n-4]$$



SUM of SHIFTED IMPULSES

n	...	−2	−1	0	1	2	3	4	5	6	...
$2\delta[n]$	0	0	0	2	0	0	0	0	0	0	0
$4\delta[n-1]$	0	0	0	0	4	0	0	0	0	0	0
$6\delta[n-2]$	0	0	0	0	0	6	0	0	0	0	0
$4\delta[n-3]$	0	0	0	0	0	0	4	0	0	0	0
$2\delta[n-4]$	0	0	0	0	0	0	0	2	0	0	0
$x[n]$	0	0	0	2	4	6	4	2	0	0	0

$$x[n] = \sum_k x[k]\delta[n-k]$$ ← **This formula ALWAYS works**

$$= \ldots + x[-1]\delta[n+1] + x[0]\delta[n] + x[1]\delta[n-1] + \ldots \quad (5.3.6)$$

4-pt AVERAGER

- CAUSAL SYSTEM: USE PAST VALUES
  - y[n] = (x[n]+x[n−1]+x[n−2]+x[n−3])/4
- INPUT = UNIT IMPULSE SIGNAL = δ[n]
  - x[n] = δ[n]
  - y[n] = 0.25 δ[n] + 0.25δ[n−1] + 0.25 δ[n−2] +0.25δ[n−3]
- OUTPUT is called "IMPULSE RESPONSE"
  - h[n] = {...,0,0,0.25,0.25,0.25,0.25,0,0,...}

FIR IMPULSE RESPONSE

- Convolution = Filter Definition
  - Filter Coeffs = Impulse Response

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

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

HARDWARE STRUCTURES

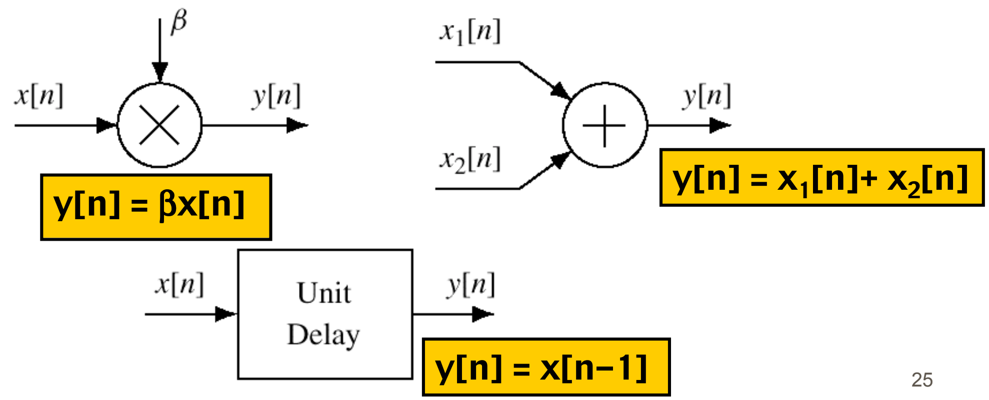
$$y[n] = \sum_{k=0}^{M} b_k\, x[n-k]$$

- **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

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

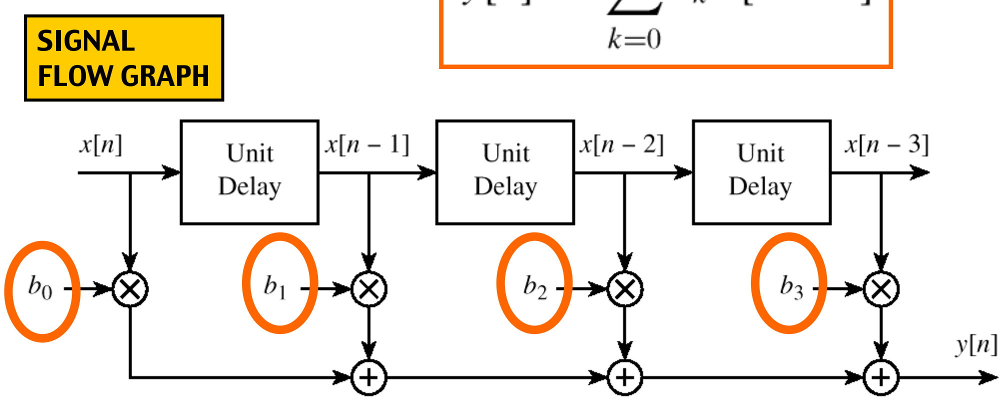
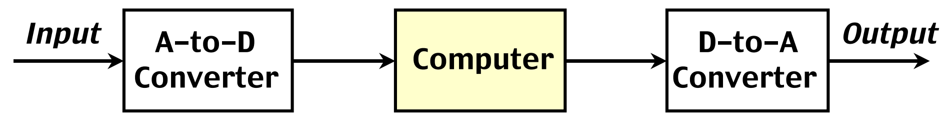
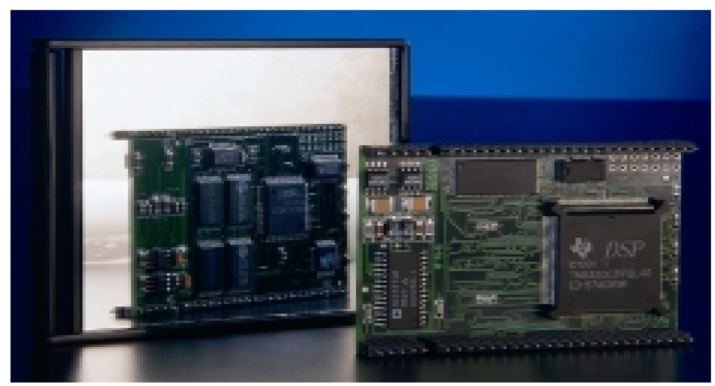


Figure 5.13 Block-diagram structure for the Mth order FIR filter.

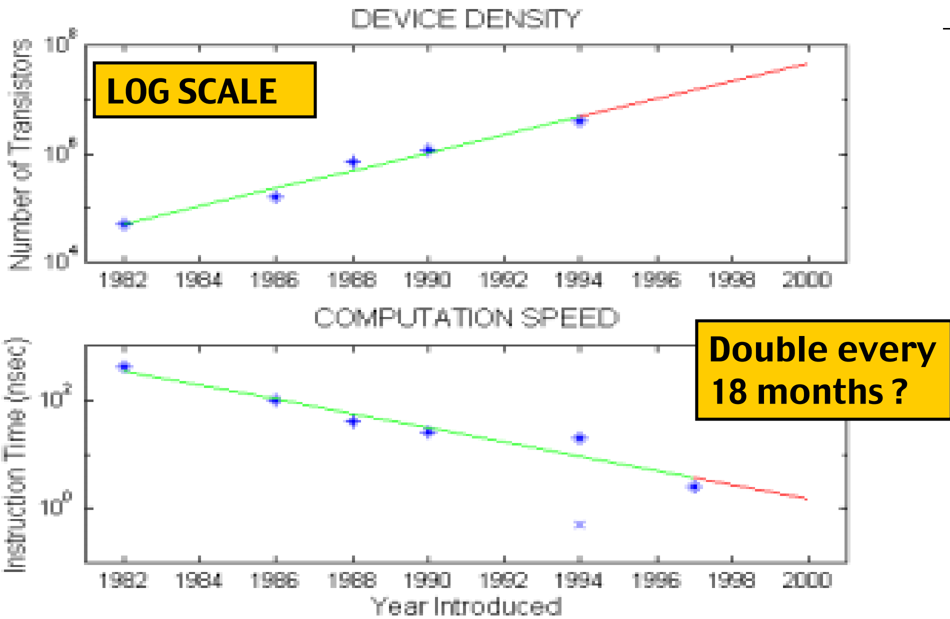
The Essence of DSP





MPEG audio encoder/decoder (from ASPI)

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 arbitrary

TESTING Time-Invariance

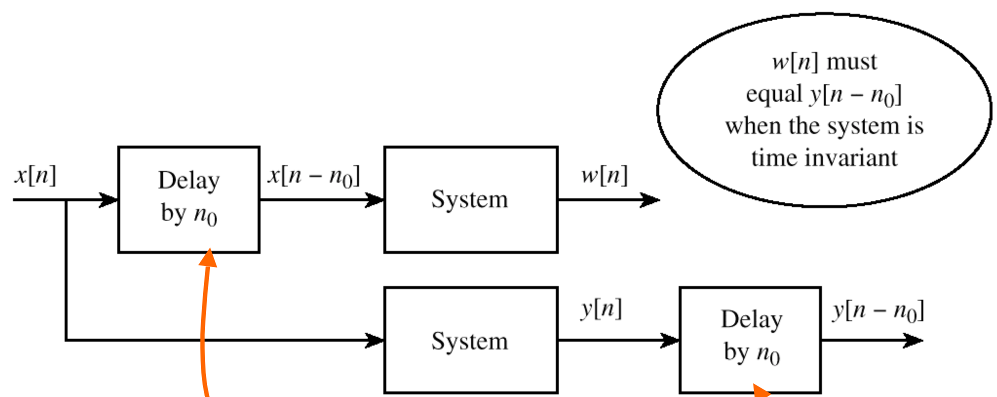


Figure 5.16 Testing time-invariance property by checking the interchange of operations.

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

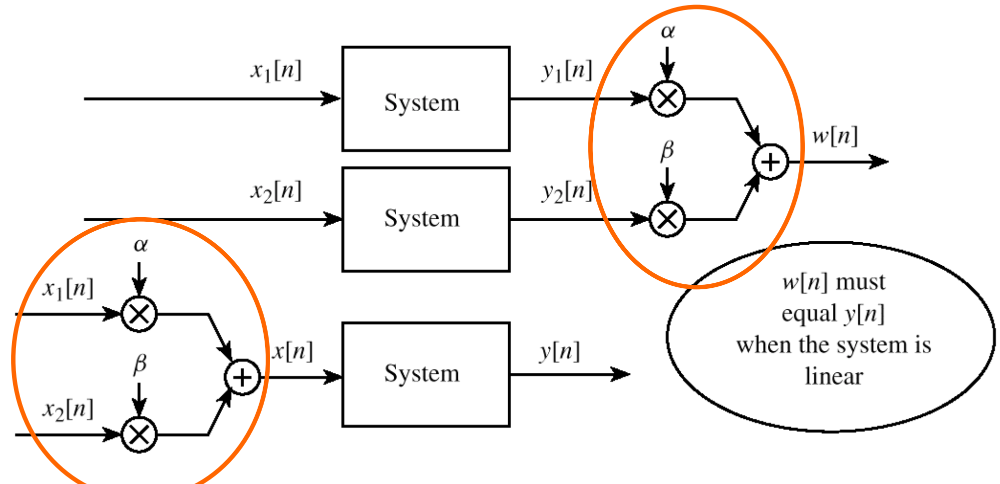


Figure 5.17 Testing linearity by checking the interchange of operations.

LTI SYSTEMS

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

LTI: Convolution

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

FINITE LIMITS

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

Same as b_k

FINITE LIMITS

CONVOLUTION Example

n	0	1	2	3	4	5	6	7	8
x[n]	2	4	6	4	2				
h[n]	3	-1	2	1					
h[0]x[n-0]	6	12	18	12	6				
h[1]x[n-1]		-2	-4	-6	-4	-2			
h[2]x[n-2]			4	8	12	8	4		
h[3]x[n-3]				2	4	6	4	2	
y[n]	6	10	18	16	18	12	8	2	