Administrative Stuff

- MATLAB1 assignment posted on Trunk
  - Under 'Assignments'
  - **Start early**, especially if your coding is rusty
  - I'll write a Python version of the DFS matlab file provided
- Today's homework is under Resources/Lecture 2
  - I'll provide scanned-in problems for first two weeks
  - Trying to locate an extra book copy to place on hold..
- **Due next Monday (more assigned Wed)**
  - Prob. 2.1; Prob 2.6, a & b; Prob. 2.10;
  - Problem 2.16, part a) and part b) # 1,2,4, 6
- Office hours are listed on Trunk – let me know what feedback you have! They can be adjusted

Tufts

Upcoming topics (full schedule on Trunk)

Unit	Topic
review	Course overview; LTI systems
review	Convolution, start Z transform
review	Z, Fourier transform
review	Sampling & Reconstruction
LTI systems	LTI system analysis using Z-transform; Rational systems
LTI systems	**15 min quiz, lectures 1-4**. LTI systems analysis using the Fourier transform
LTI systems	Phase and group delay, geometric interpretation
LTI systems	Filter design by pole-zero placement, common simple filters

Tufts

EE-125:
Digital Signal Processing

Review Lecture:
DT Systems and Convolution

Professor Tracey

Where we are

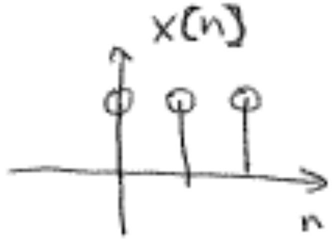
- Last week: we reviewed elementary "building block" functions
  - impulse, unit step, exponential
  - how to handle phase: wrapped vs unwrapped
- Today: quick review of convolution, etc
  - discrete-time systems properties (Linearity, time-invariance, causality, stability)
  - impulse response
  - Convolution
  - constant coefficient difference equations (CCDEs); FIR vs IIR systems
- Then, start Z transform

DT system properties and concepts

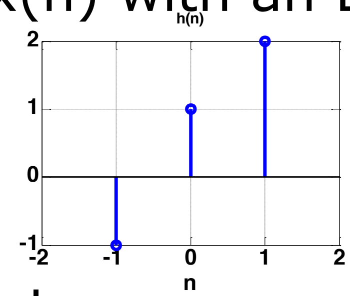
- The big two: “LTI” systems are **L**inear and **T**ime-**I**nvariant (also called *shift-invariant*)
- Causal: current output depends on past and current inputs
- BIBO stable (bounded input gives bound ed output)
- Concept: the impulse response h(n) is the response of the system when the input is an impulse, d(n)

Graphical "derivation" of convolution

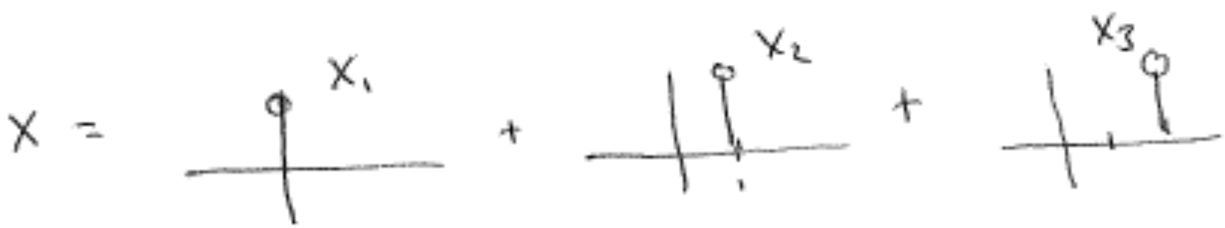
- Problem statement: convolve x(n) with an LTI system h(n)



*



- Break x(n) into a series of impulses



- Then, response to x1(n) is the impulse response scaled by x(0)
- Response to x2(n) is h(n) delayed by 1 and scaled by x(1)
- i.e. we use **linearity** and **time-shifting** (**L** and **TI**)

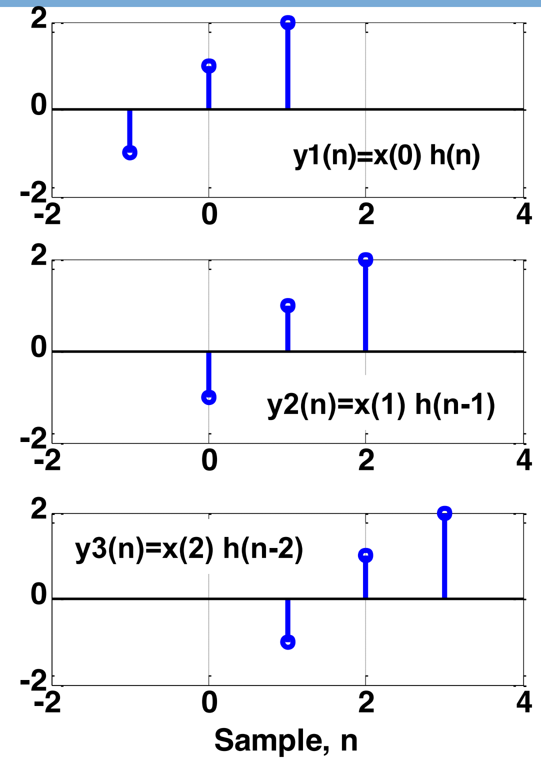
Graphical "derivation" cont'd

- Graphically, we have:

x_1 ->

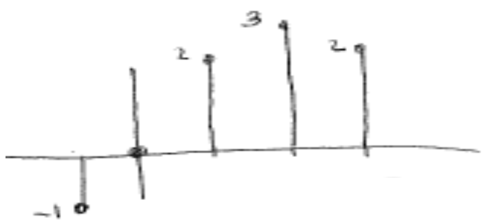
x_2 ->

x_3 ->



- Adding them up gives y:

$$y = \sum_{k=0}^{2} x[k]\, h[n-k]$$



More convolution….

Approach 2: more mathematical derivation

- A 1-page, more mathematical derivation is posted on Trunk under Lecture 2
- Similar discussion is in the book… please review

Approach 3: working w/ convolution formula

- DT convolution demo can be downloaded http://users.ece.gatech.edu/mcclella/matlabGUIs/, uploaded to Trunk

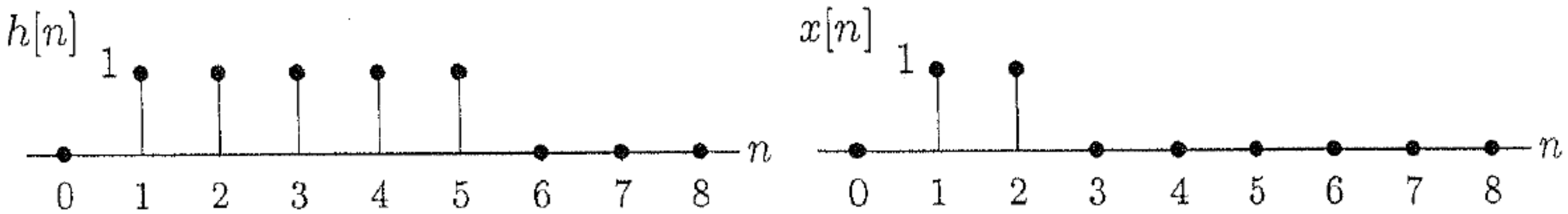
Just for fun – an example of audio convolution

https://www.youtube.com/watch?v=cGBn7sU6m3k

Tufts

A question…

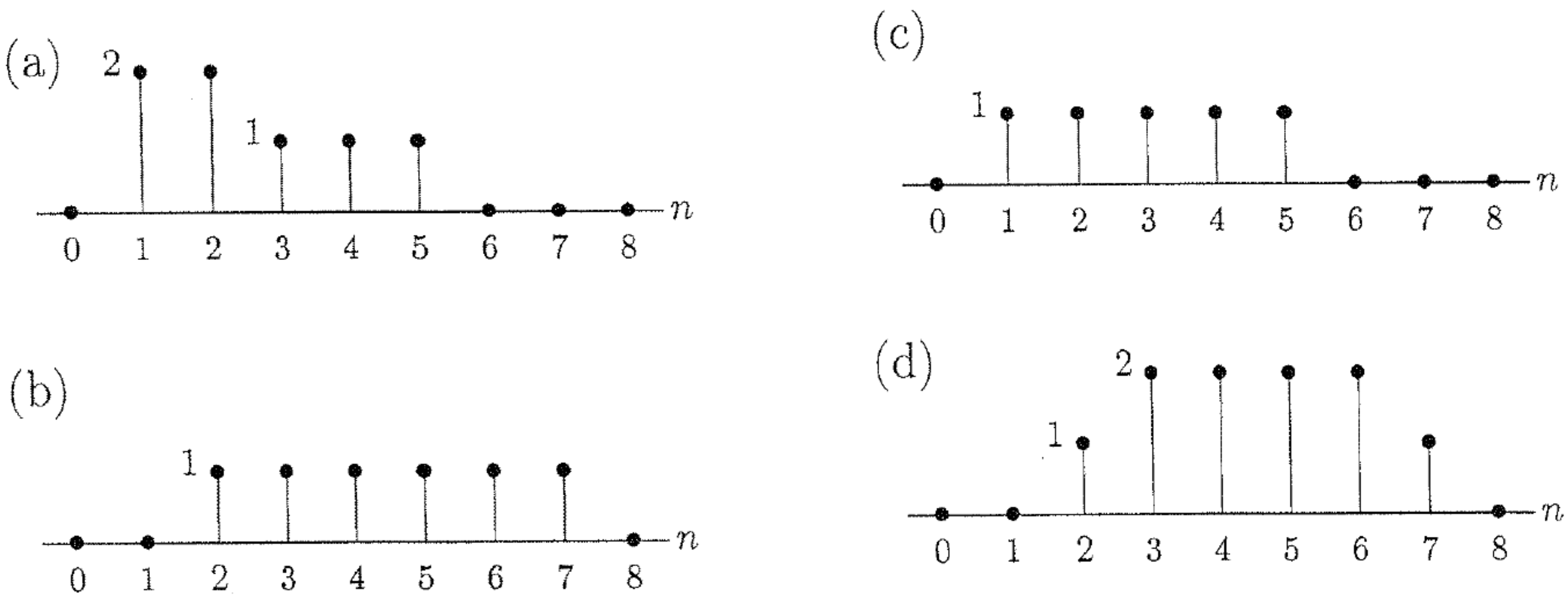
The plots below show the impulse response $h[n]$ of an LTI system and the input $x[n]$ to that system.



What does y = h*x look like - roughly?

Choices on next slide

Possible answers



Convolution properties: block diagrams

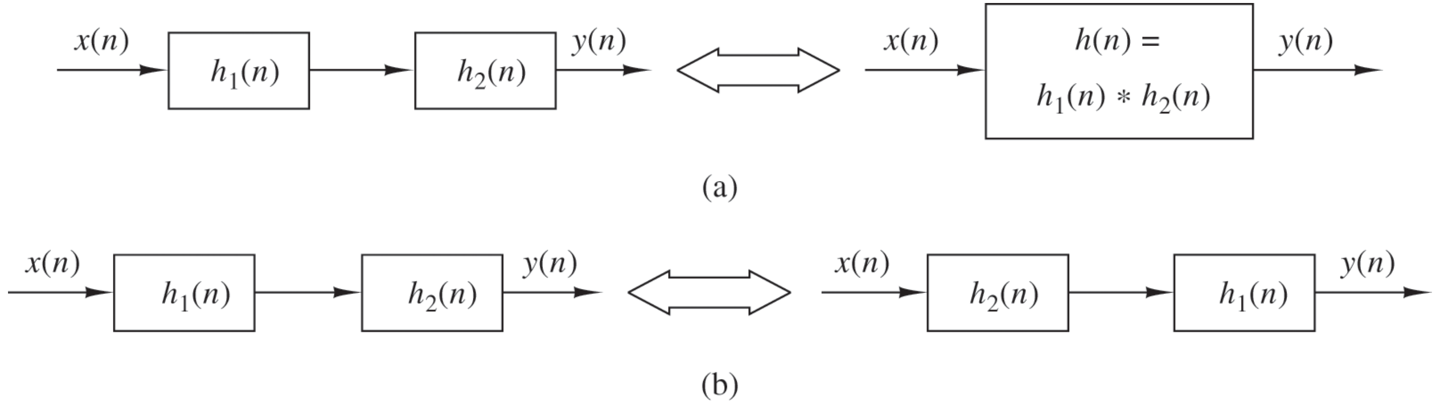


Figure 2.3.5 Implications of the associative (a) and the associative and commutative (b) properties of convolution.

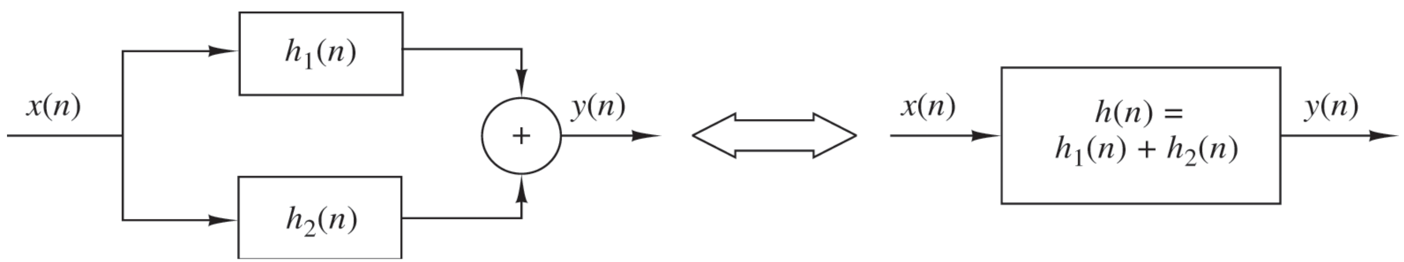


Figure 2.3.6 Interpretation of the distributive property of convolution: two LTI systems connected in parallel can be replaced by a single system with $h(n) = h_1(n) + h_2(n)$.

Constant-coefficient difference equations

- See board notes for definition
- Convolution can be considered as a special case of CCDE
  - no feedback
    - FIR filters or systems are sometimes called “convolutional”

Constant Coefficient Difference Equations (CCDE's): most general (

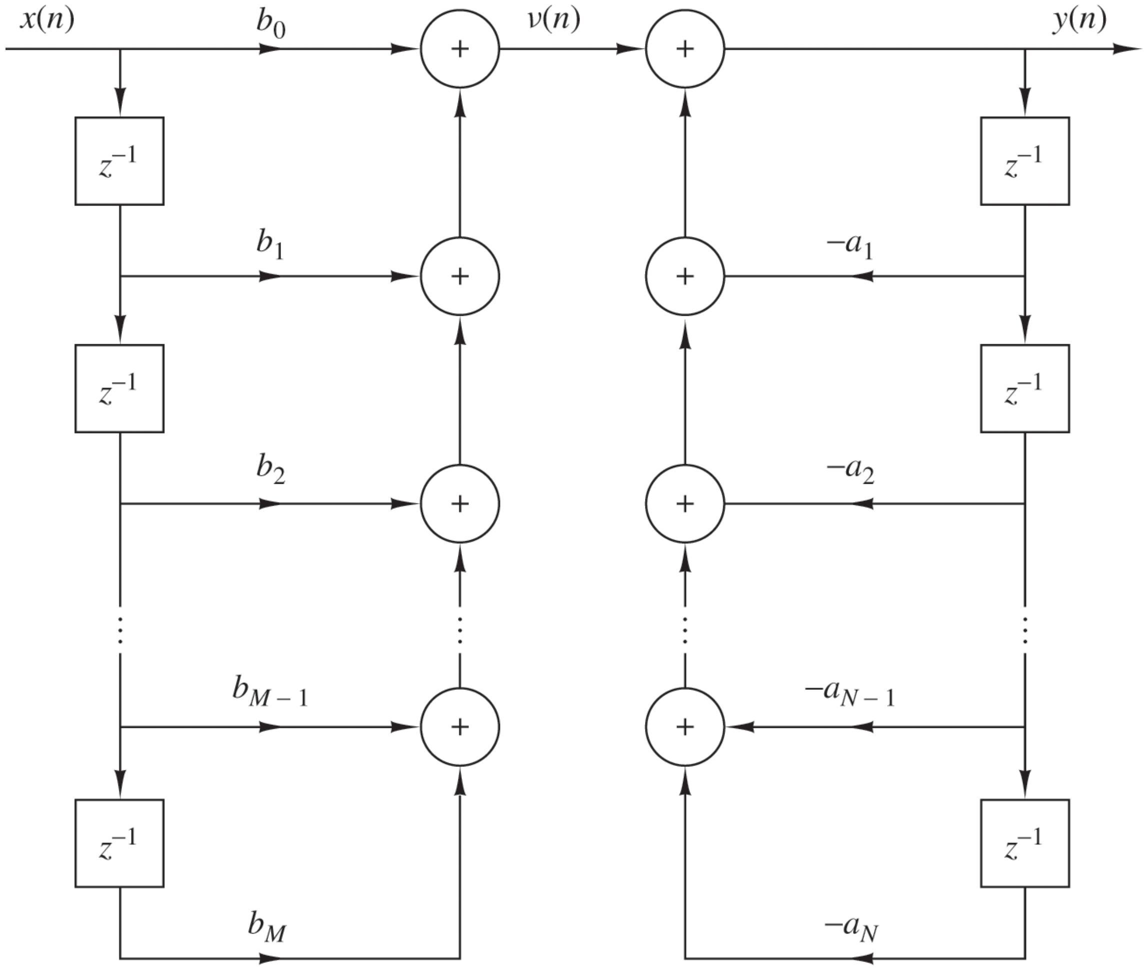


Figure 2.5.2 Direct form I structure of the system described by (2.5.6).

Direct Solution of CCDE's

- Section 2.4.3 – Direct solution. We won't cover in detail but the general idea is:

$$y[n] = yh[n] + yp[n]$$

- yh[n] is the **homogenous** solution, or 'zero-input' solution. Describes how *initial conditions* decay over time, assuming there are no other inputs (hence 'zero-input')
  - If system is initially relaxed (at rest), yh=0
  - If system is stable, yh decays away over time
- yp[n] is the **particular** solution, or 'zero-state' solution. Describes system response to the input x[n], assuming system started at rest
- In this class we will usually **ignore yh (assuming steady-state conditions, or at rest)** but sometimes it is very important

A BETTER ANALYSIS TOOL: Z-TRANSFORM