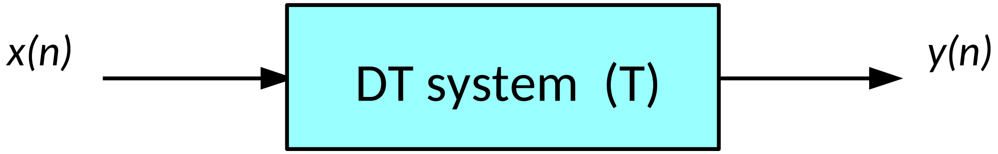
Discrete-time signals and systems, II

- Discrete-time systems
- Representation of discrete-time systems
- Classification of discrete-time systems
- Discrete Linear Time-Invariant (LTI) systems
- Convolution
- Convolution, summary
- Convolution properties
- Examples of LTI systems
- Impulse response and stability
- (Additional materials)

Discrete-time systems

- A *discrete-time (DT) system* is a device or algorithm that operates on a discrete-time signal, $x(n)$, called the *input* or *excitation*, according to some well-defined rule, to produce another discrete-time signal, $y(n)$, called the *output* or *response* of the system



$y(n) = T[\,x(n)\,]$

- The *input-output description* of a discrete-time system consists of a *mathematical expression* or a rule, which explicitly defines the relation between the input and output signals (T – *input-output relationship*)

  → *Difference equation (DE)*

- Examples of discrete-time systems:
  - Delaying the input $y(n) = x(n-5)$
  - Moving average filter $y(n) = 1/3\,[\,x(n+1) + x(n) + x(n-1)\,]$
  - Squaring the input $y(n) = x(n)\,x(n)$
  - Minimum $y(n) = \min\{\,x(n+1), x(n), x(n-1)\,\}$

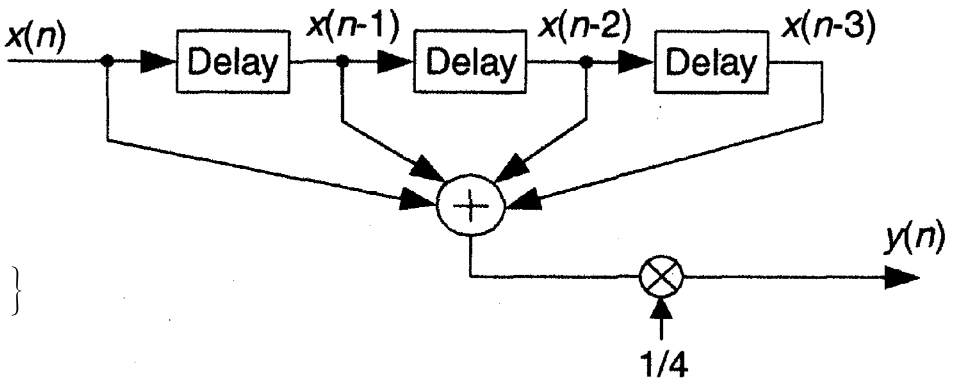
Representation of discrete-time systems

- Difference equation (DE) of a system, *y(n)* = T [*x(n)*, *x(n – 1)*, *x(n – 2)*, …]

  - Example

$$y(n) = \frac{1}{4}[x(n)+x(n-1)+x(n-2)+x(n-3)] = \frac{1}{4}\sum_{k=n-3}^{n} x(k)$$

- Block diagram realization of the system



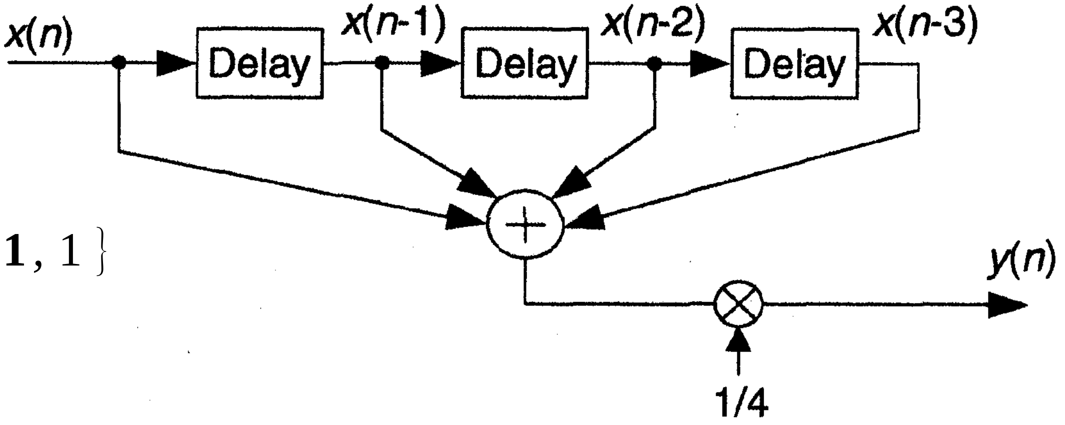
$$x(n) = \{\mathbf{1}, 1\}$$

- What is the output *y(n)*, if the input is rectangular signal, *x(n)* = {**1**, 1}?

[Lyons]

Representation of discrete-time systems

- Example $$y(n) = \frac{1}{4}[x(n)+x(n-1)+x(n-2)+x(n-3)] = \frac{1}{4}\sum_{k=n-3}^{n} x(k)$$



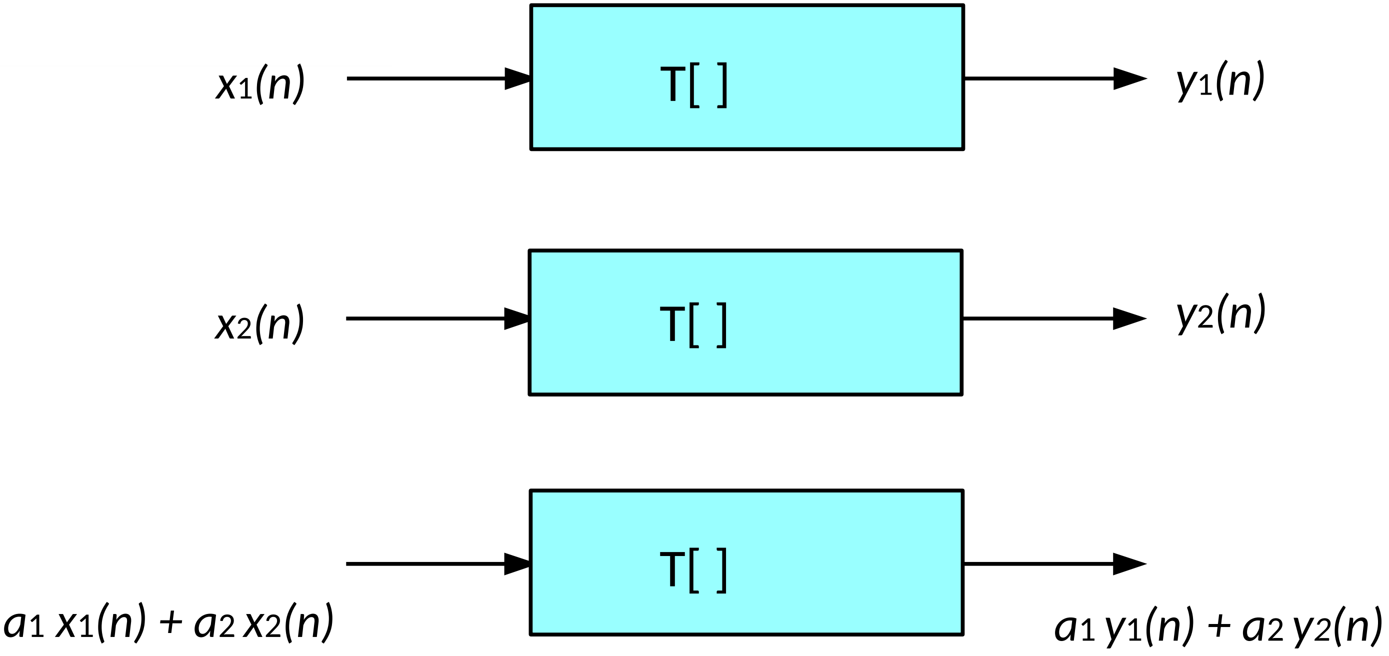
$$x(n) = \{\mathbf{1}, 1\}$$

$$y(n) = \{\mathbf{1/4}, 1/2, 1/2, 1/2, 1/4\}$$

[Lyons]

Classification of discrete-time systems

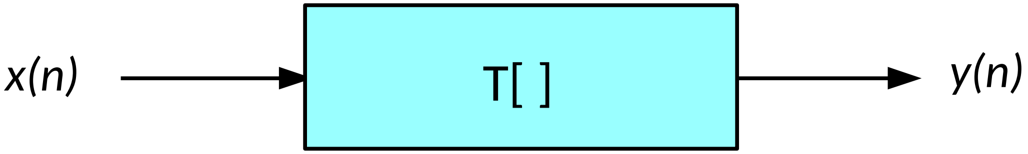
- Linear versus nonlinear systems
  - Linear systems obey *superposition*, i.e., *additivity and homogeneity (scaling)*



Classification of discrete-time systems

- Linear versus nonlinear systems

  - A system T



is linear if and only if

$$T\,[\,a_1 x_1(n) + a_2 x_2(n)\,] \;=\; a_1 T\,[\,x_1(n)\,] + a_2 T\,[\,x_2(n)\,]$$

for *all* $x_1(n)$, $x_2(n)$, and *all* constants a_1 and a_2 (additivity and scaling)

- How to determine if the system is linear?

1. $y_1(n) = T\,[\,x_1(n)\,]$
2. $y_2(n) = T\,[\,x_2(n)\,]$
3. $T\,[\,a_1 x_1(n) + a_2 x_2(n)\,] \overset{?}{=} a_1 y_1(n) + a_2 y_2(n)$

Classification of discrete-time systems

- Linearity, example, accumulator → $$y(n)=\sum_{l=-\infty}^{n} x(l)$$

1. $$y_1(n)=\sum_{l=-\infty}^{n} x_1(l)$$

2. $$y_2(n)=\sum_{l=-\infty}^{n} x_2(l)$$

3. $$x(n)=a_1 x_1(n)+a_2 x_2(n)$$ → Linear combination of inputs

T[*x*(*n*)] $$\longrightarrow \quad y(n)=\sum_{l=-\infty}^{n} (a_1 x_1(l)+a_2 x_2(l))$$

$$y(n)=\sum_{l=-\infty}^{n} (a_1 x_1(l))+\sum_{l=-\infty}^{n} (a_2 x_2(l))$$

$$y(n)=a_1 \sum_{l=-\infty}^{n} x_1(l)+a_2 \sum_{l=-\infty}^{n} x_2(l)$$

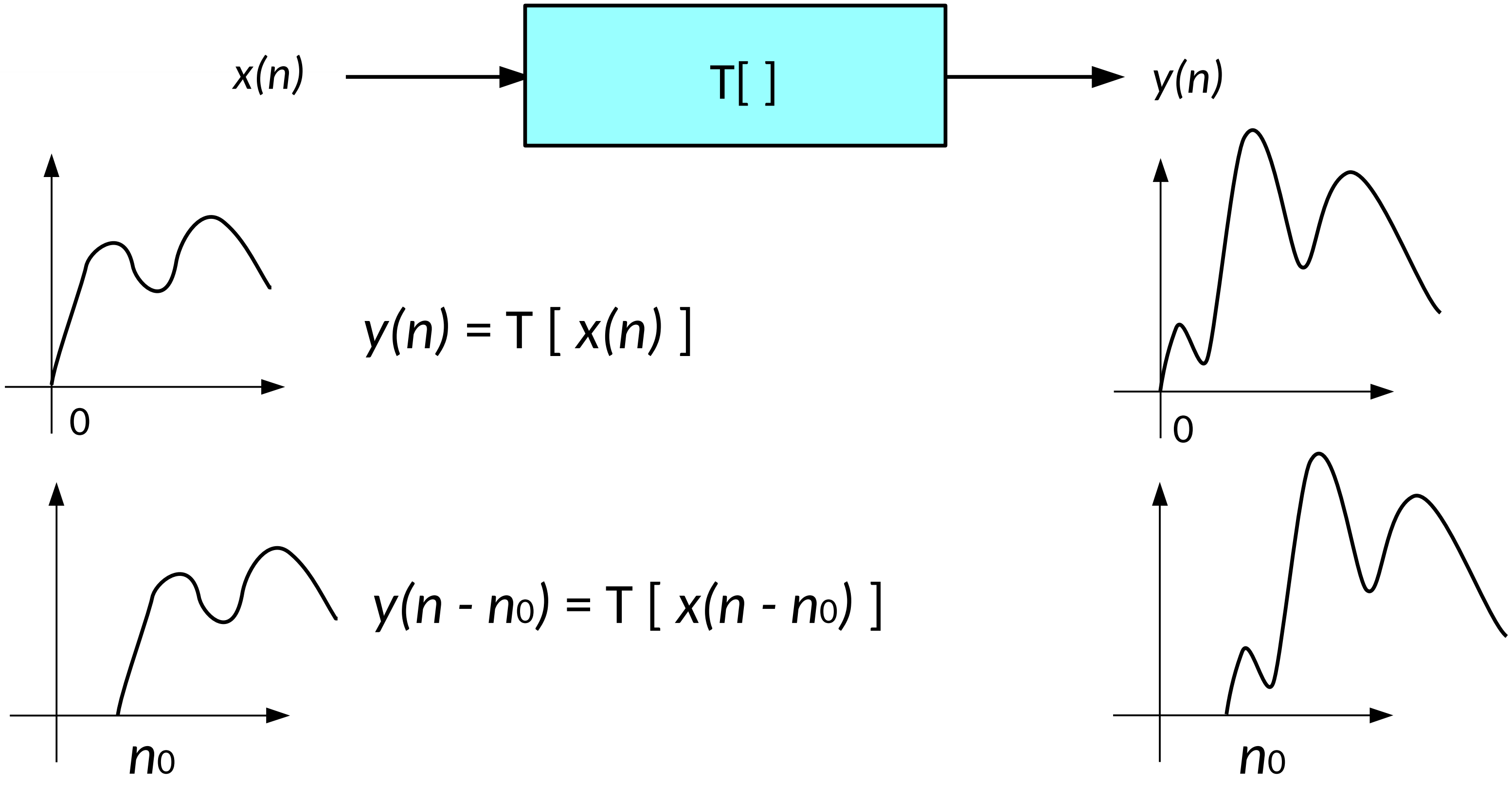
$$y_1(n)=\sum_{l=-\infty}^{n} x_1(l)$$

$$y_2(n)=\sum_{l=-\infty}^{n} x_2(l)$$

Linear $$\longleftarrow \quad y(n) = a_1 y_1(n) + a_2 y_2(n)$$

Classification of discrete-time systems

- Time-invariant systems versus time-variant systems
  - Time shift of input causes the same shift at output



$$y(n) = \mathrm{T}\,[\,x(n)\,]$$

$$y(n - n_0) = \mathrm{T}\,[\,x(n - n_0)\,]$$

Classification of discrete-time systems

- Time-invariant systems versus time-variant systems

  - A system T is time-invariant (or shift-invariant) if and only if

$$y(n) = T[\,x(n)\,]$$

implies that

$$y(n - n_0) = T[\,x(n - n_0)\,]$$

for *every* input signal x(n) and *every* time shift n_0)

  - How to determine if the system is time-invariant?

1. $y(n) = T[\,x(n)\,]$
2. $y_1(n, n_0) = T[\,x(n - n_0)\,]$ (delay the input, $x(n)$)
3. $y(n - n_0) = T[\,x(n - n_0)\,]$ (delay the output, n)
4. $y_1(n, n_0) \overset{?}{=} y(n - n_0)$

Classification of discrete-time systems

- Time-invariant, example, → (1.) $y(n) = (x(n))^2$

Delay the input 2. $y_1(n, n_0) = (x(n-n_0))^2$

Delay the output 3. $y(n-n_0) = (x(n-n_0))^2$

4. $y_1(n, n_0) = y(n-n_0)$

- Time-variant, example, → (1.) $y(n) = n \, . \, x(n)$

Delay the input 2. $y_1(n, n_0) = n \, . \, x(n-n_0)$

Delay the output 3. $y(n-n_0) = (n-n_0) \, . \, x(n-n_0)$

4. $y_1(n, n_0) \neq y(n-n_0)$

Classification of discrete-time systems

- Example of exam task
  - Is the following system linear? Is it time-invariant?

$$y(n)=x^2(n)-x(n-1).x(n+1)$$

Classification of discrete-time systems

- Causal versus non-causal systems
  - A system is called causal if the output of the system at any time n depends only on present and past inputs, but does not depend on future inputs

$$y(n) = \mathrm{T}\,[\, x(n), x(n-1), x(n-2), \ldots \,]$$

- A causal system is physically realizable system (not for off-line processing)
- (A signal is called causal if and only if $x(n) = 0$ for $n < 0$)
  - Backward difference, $y(n) = x(n) - x(n-1)$ → causal
  - Forward difference, $y(n) = x(n+1) - x(n)$ → non-causal, looks forward in time
- Causality → consequence happens after the cause

Classification of discrete-time systems

- Check for causality
- Moving average

$$y(n) = \frac{1}{M}\sum_{k=0}^{M-1} x(n-k)$$

→ causal, *y(n)* depends on *x(n - k)*, *k* ≥ 0,

- Centered moving average (counter example)

$$y(n) = \frac{1}{M}\sum_{k=-(M-1)/2}^{(M-1)/2} x(n-k)$$

→ non-causal, looks forward in time, can be made causal by delaying

Classification of discrete-time systems

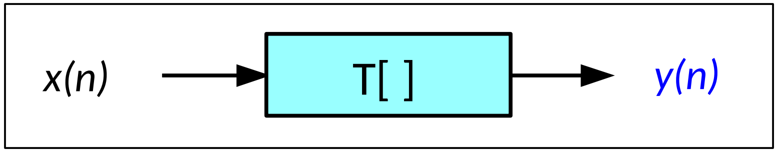
- Stable versus unstable systems
  - A system is said to be Bounded Input Bounded Output (BIBO) stable if and only if every bounded input produces a bounded output
  - There exist some finite numbers B_x and B_y such that

$$\exists B_x, B_y: \quad |x(n)| \leqslant B_x < \infty, \quad |y(n)| \leqslant B_y < \infty$$

Discrete Linear Time-Invariant (LTI) systems

- An important subset of discrete systems – discrete Linear Time-Invariant Systems (LTI)
  - Properties
    * Linearity (allows analysis using elementary signals)
    * Time-invariability (operating does not change over time)
  - Consequences
    * Mathematical tractability
    * Simple realization
    * → Convolution

Convolution

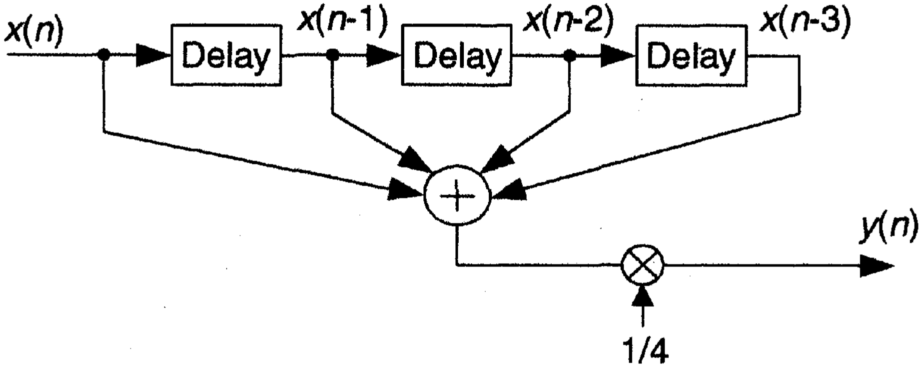


- Why convolution?

So far, we were able to determine

- the output, *y*(*n*), of
  an LTI system to the input signal, *x*(*n*),
- if knowing
  - * the difference equation
  - * or, block diagram of the LTI system

$$y(n) = \frac{1}{4}[x(n)+x(n-1)+x(n-2)+x(n-3)] = \frac{1}{4}\sum_{k=n-3}^{n} x(k)$$



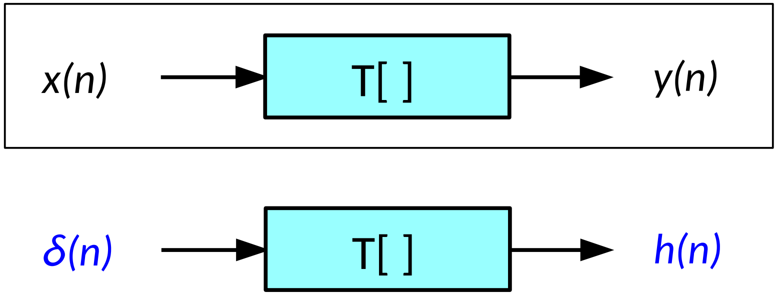
x(*n*) → T[ ] → *y*(*n*)

Convolution

- Why convolution?

Using convolution, we can determine

- the output, $y(n)$, of an LTI system to the input signal, x(n),
- if knowing
  - * the response of the LTI system (denoted $h(n)$) to one particular input signal, the unit sample signal, $\delta(n)$



If $x(n) = \delta(n)$ then $y(n) = h(n)$

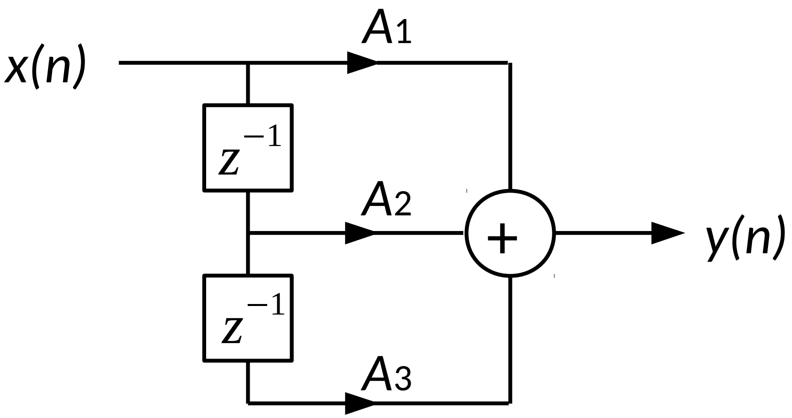
$\delta(n)$ - the unit sample $h(n)$ - impulse response

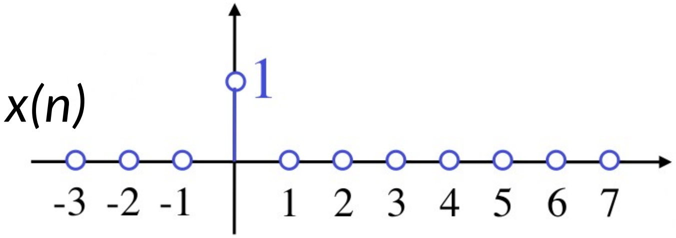
$x(n), h(n)$ → convolution → $y(n)$

Convolution

- Impulse response

- Difference equation, $y(n) =$?





$x(n) = \delta(n)$ impulse - unit sample

↓

$y(n) = h(n)$ impulse response = ?

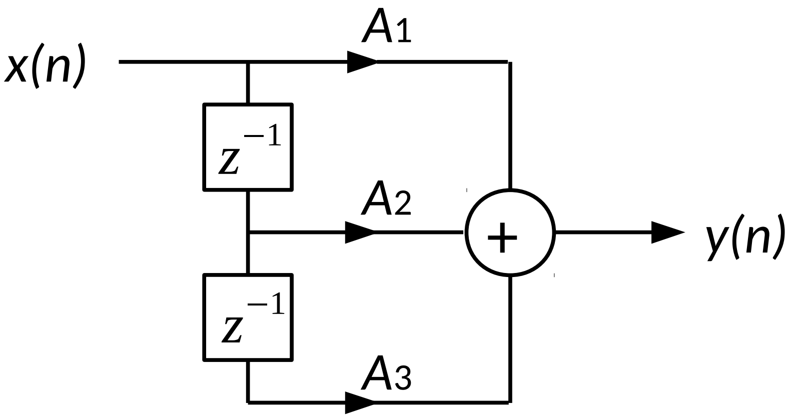
- Output values, $y(n) =$?

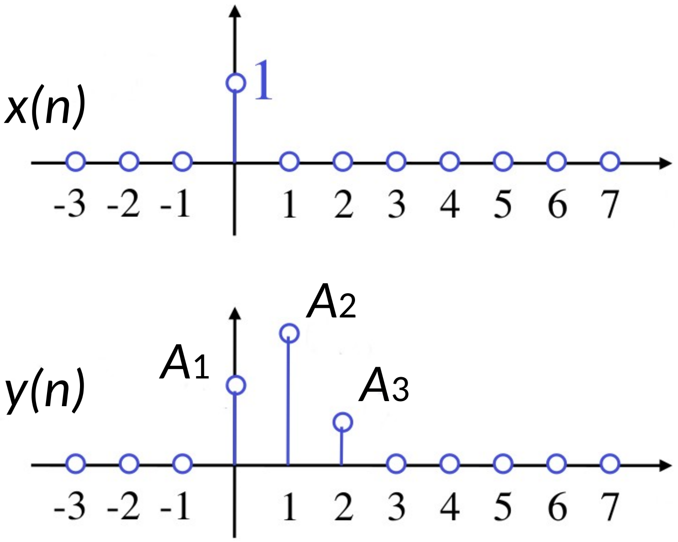
$y(n) = h(n) =$?

Convolution

- Impulse response

$$y(n) = A_1.x(n) + A_2.x(n-1) + A_3.x(n-2)$$





$x(n) = \delta(n)$ impulse - unit sample

$\downarrow$

$y(n) = h(n)$ impulse response

$h(n) = A_1.\delta(n) + A_2.\delta(n-1) + A_3.\delta(n-2)$

$y(n) = h(n) = \{A_1, A_2, A_3\}$

$\uparrow n = 0$

→ LTI system is completely specified by *h(n)*

Convolution

- Decomposition of discrete signal *x(n)*

- Discrete signal *x(n)* can be expressed by a sum of unit samples

$$x(n)=\sum_{k=-\infty}^{\infty} x(k)\,\delta(n-k)$$

where

$$\delta(n-k)=\begin{cases}1, & n=k\\ 0, & n\neq k\end{cases}$$

- Example, *causal signal*: *x(n)* = 0, *n* < 0

$$x(n) \;=\; x(0)\,\delta(n) \;+\; x(1)\,\delta(n-1) \;+\; x(2)\,\delta(n-2) \;+\; \ldots$$

Convolution

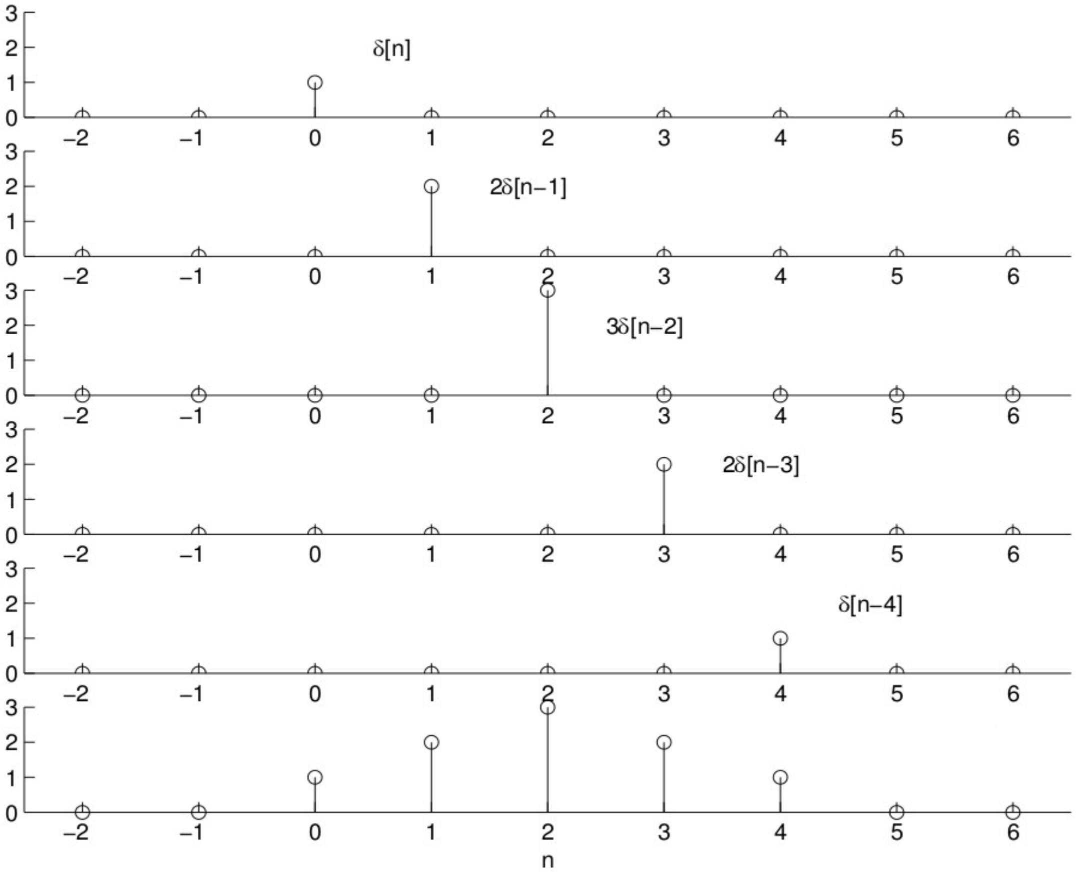
- The decomposition of a triangular signal, *x(n)*, into a sum of unit samples

$$x(n) = \{1, 2, 3, 2, 1\}$$

$n = 0$

$$x(n) = \sum_{k=-\infty}^{\infty} x(k)\,\delta(n-k)$$

$$x(n) \longrightarrow$$



$$x(n) \ = \ 1.\delta(n) \ + \ 2.\delta(n-1) \ + \ 3.\delta(n-2) \ + \ 2.\delta(n-3) \ + \ 1.\delta(n-4)$$

$$x(n) \ = \ x(0)\delta(n) + x(1)\delta(n-1) + x(2)\delta(n-2) + x(3)\delta(n-3) + x(4)\delta(n-4)$$

Convolution

- Response of an LTI system

  - Suppose that *h(n)* is impulse response to unit sample $\delta(n)$, $\delta(n) \rightarrow T[\,] \rightarrow h(n)$

  - Since the system is

    time-invariant $\longrightarrow$ $T[\delta(n)] = h(n) \rightarrow T[\delta(n-k)] = h(n-k)$

    and linear $\longrightarrow$ $T[a.\delta(n-k)+b.\delta(n-l)] = a.h(n-k)+b.h(n-l)$

  - The output is as follows:

$$y(n) = T[x(n)]$$

$$= T\left[\sum_{k=-\infty}^{\infty} x(k)\delta(n-k)\right]$$

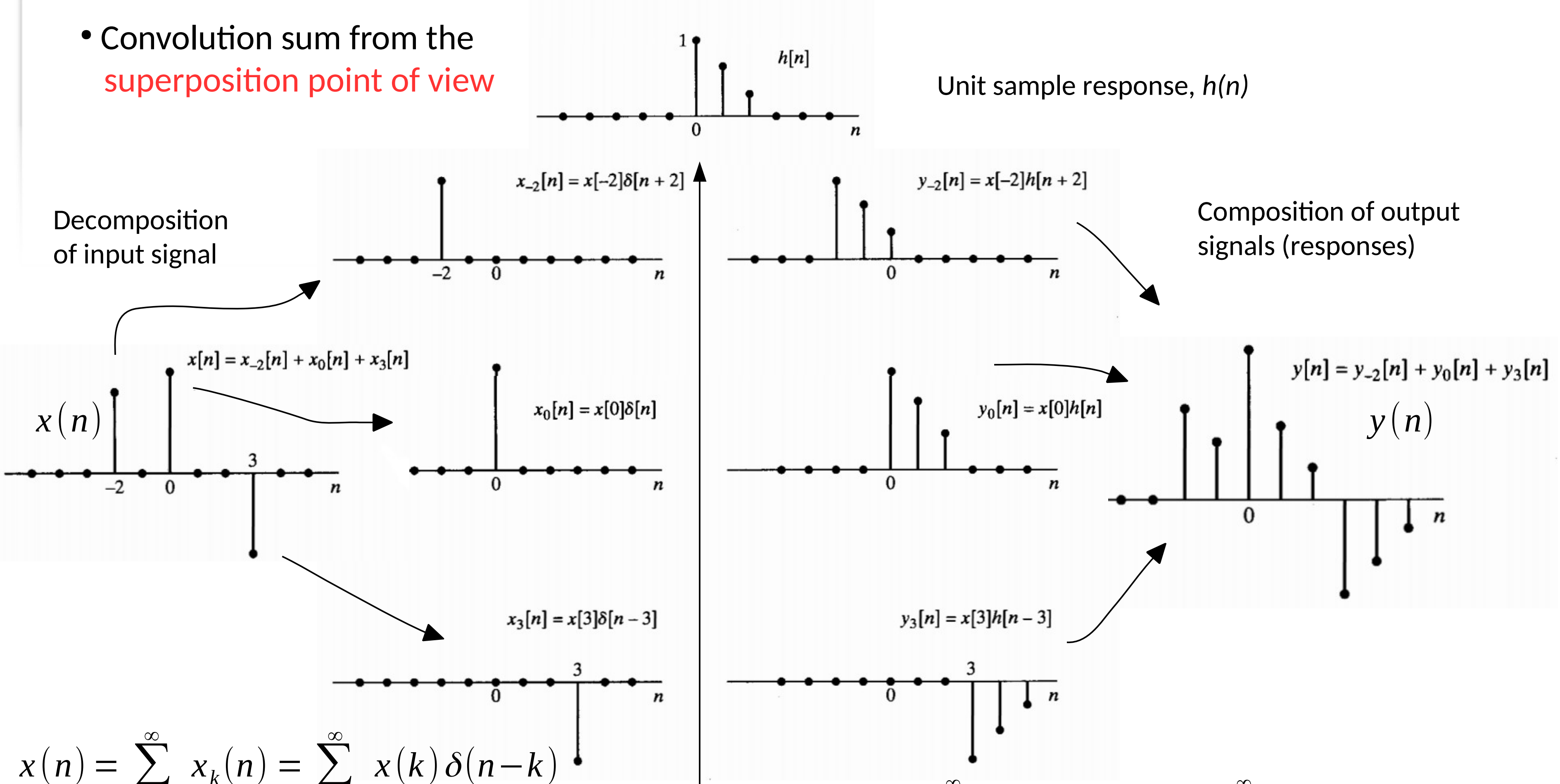
$$= \sum_{k=-\infty}^{\infty} x(k)T[\delta(n-k)] \longrightarrow \text{Due to linearity}$$

$$= \sum_{k=-\infty}^{\infty} x(k)h(n-k) \longrightarrow \text{Due to definition of } h(n) \text{ and time-invariance}$$

$$y(n) = \sum_{k=-\infty}^{\infty} x(k)\, h(n-k) = x(n) * h(n) \longrightarrow \text{Convolution sum}$$

Convolution

- Convolution sum from the superposition point of view



Unit sample response, *h(n)*

Decomposition of input signal

Composition of output signals (responses)

$$x(n) = \sum_{k=-\infty}^{\infty} x_k(n) = \sum_{k=-\infty}^{\infty} x(k)\,\delta(n-k)$$

$$\sum_{k=-\infty}^{\infty} x(k)\,h(n-k) = \sum_{k=-\infty}^{\infty} y_k(n) = y(n)$$

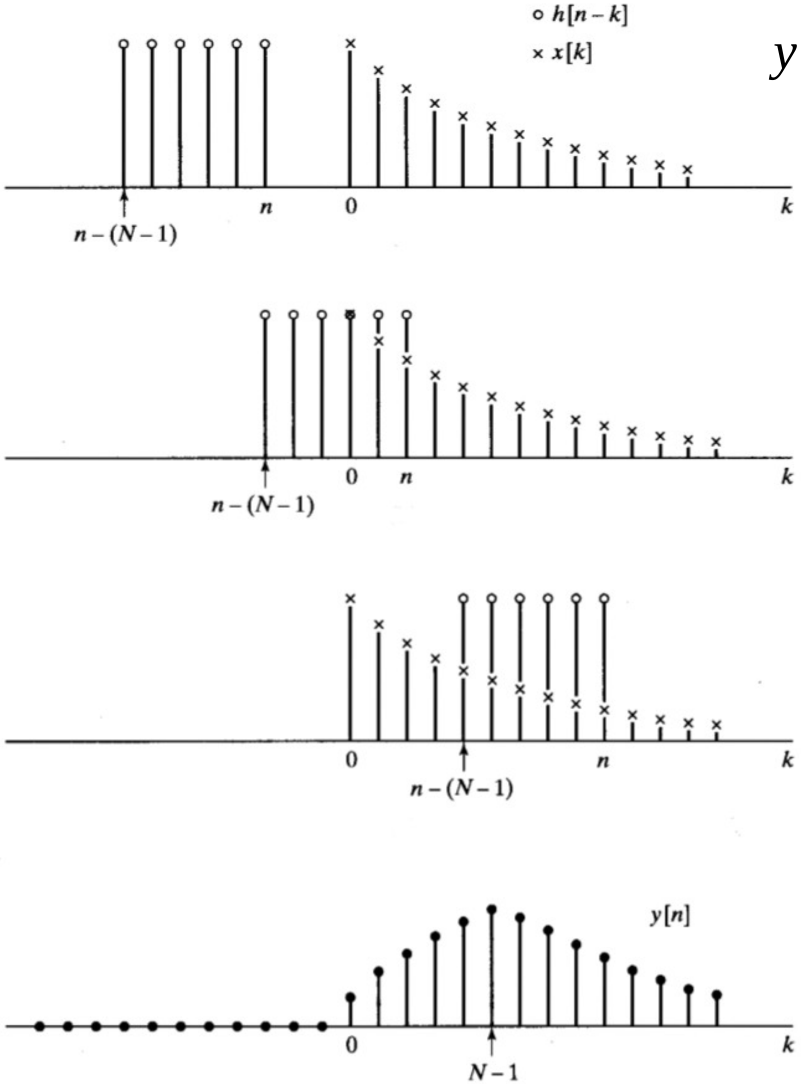
Demo: https://engineering.purdue.edu/VISE/ee438/demos/

[Oppenheim, Schafer]

Convolution

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

- More useful computational interpretation (consider *h(n-k)* as reversing it in time at *k* = 0 and then delaying i by *n* samples) !



- Convolution interpretation (the system is moving over the signal)

Off-line analysis,
flipped *h(n)* is
moving over
the signal *x(n)*

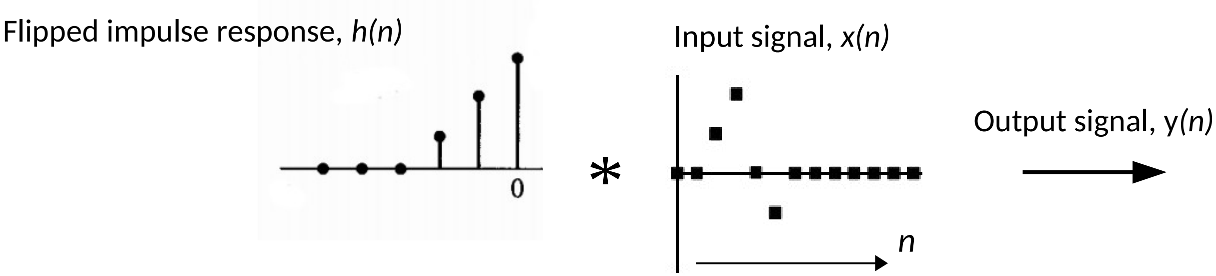
[Oppenheim, Schafer]

Convolution

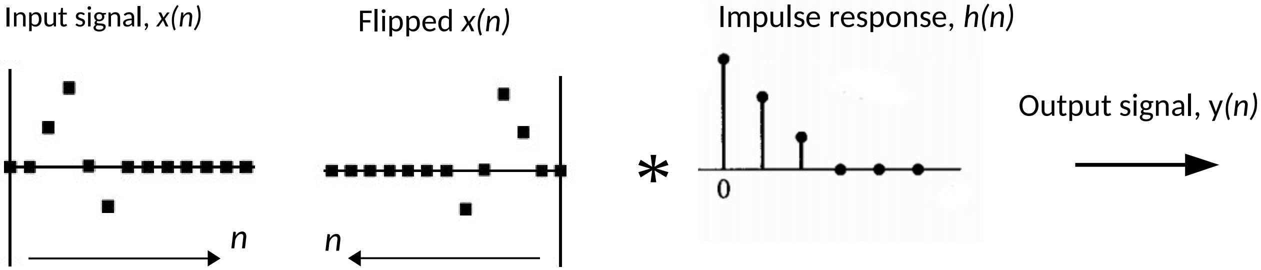
- Interpretation / implementation of convolution
  - Convolving a signal with an impulse response of an LTI system is equivalent to passing the signal through the system (summation is symmetric in *x* and *h*)

$$y(n) = \boxed{\sum_{k=-\infty}^{\infty} x(k)\,h(n-k) \quad 1.} = \boxed{\sum_{k=-\infty}^{\infty} h(k)\,x(n-k) \quad 2.}$$

1. The system is moving over the signal
   (“Off-line” analysis)


Flipped impulse response, *h*(*n*)
Input signal, *x*(*n*)
Output signal, y(*n*)

2. Input signal is passing the system
   (“Real-time” analysis)

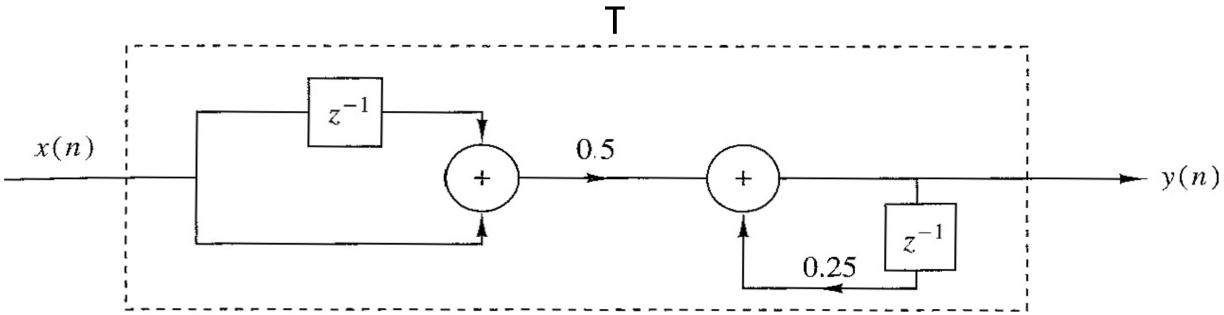

Input signal, *x*(*n*)
Flipped *x*(*n*)
Impulse response, *h*(*n*)
Output signal, y(*n*)

Demo: https://engineering.purdue.edu/VISE/ee438/demos/

Demo: http://www.fourier-series.com/

Convolution

- Example
- A block diagram of an LTI system, $y(n) = T[\,x(n), y(n-1)\,]$



- Difference equation: $y(n) = 0.25\, y(n-1) + 0.5\, x(n) + 0.5\, x(n-1)$
- *What is the length of impulse response?*
- *Convolution ?*

→ Convolution is applicable only for those LTI systems for which
$y(n) = T[x(n), x(n-1), \ldots, x(n-M)]$
i.e., impulse response, *h(n)*, is of finite duration

Convolution

- Example of exam task

  - An LTI system is defined by the following impulse response

$h(n) = 2 \cos(n\pi / 2) . [u(n) - u(n-3)]$

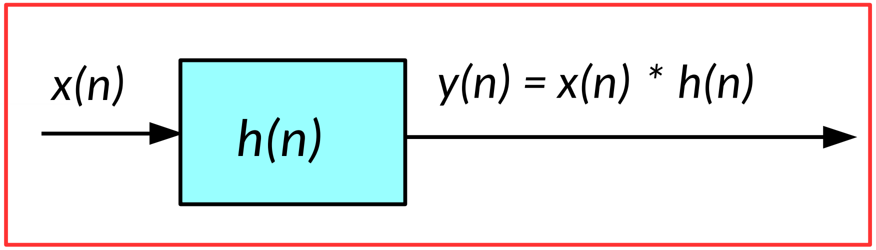
Determine the response of the system using convolution for the following input signal

$x(n) = n . [u(n) - u(n-4)]$

Hint: Consider the difference of two unit step signals, u(.)

Convolution, summary

$$y(n) = \sum_{k=-\infty}^{\infty} x(k)\, h(n-k) = x(n) * h(n) \longrightarrow$$ Convolution sum



- The impulse response *h(n)* can determine the characteristics of an LTI system completely
- For any input *x(n)*, the output *y(n)* is the convolution sum of *x(n)* and *h(n)*
- Convolution can be used to implement only those LTI systems for which the impulse response *h(n)* is of finite duration

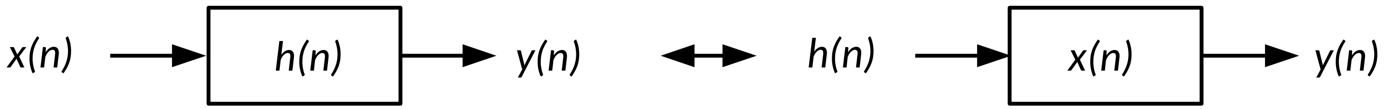
$$h(n) = \begin{cases} \neq 0, & 0 \leqslant n < N \\ 0, & \text{elsewhere} \end{cases}$$

Convolution properties

- Convolution properties define connections of LTI systems

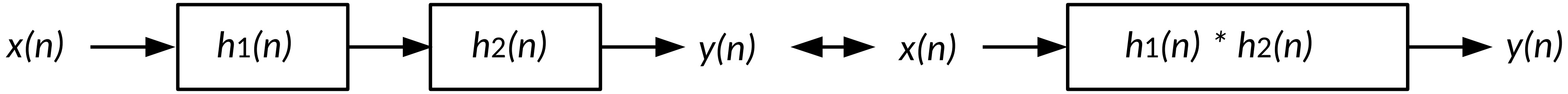
  - Convolution is a commutative operation

$$x(n) * h(n) = h(n) * x(n)$$



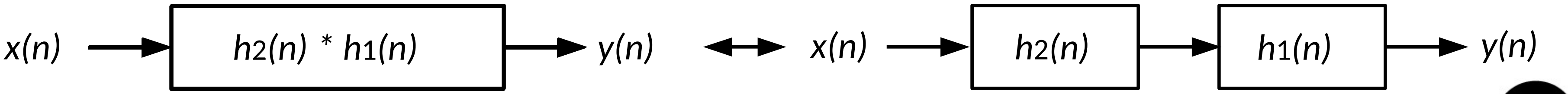
  - Convolution is an associative operation → *Cascade connection*

$$[\, x(n) \ast h1(n) \,] \ast h2(n) = x(n) \ast [\, h1(n) \ast h2(n) \,]$$



again commutativity and again associativity → *Order of systems exchanged*

$$= x(n) \ast [\, h2(n) \ast h1(n) \,] = [\, x(n) \ast h2(n) \,] \ast h1(n)$$

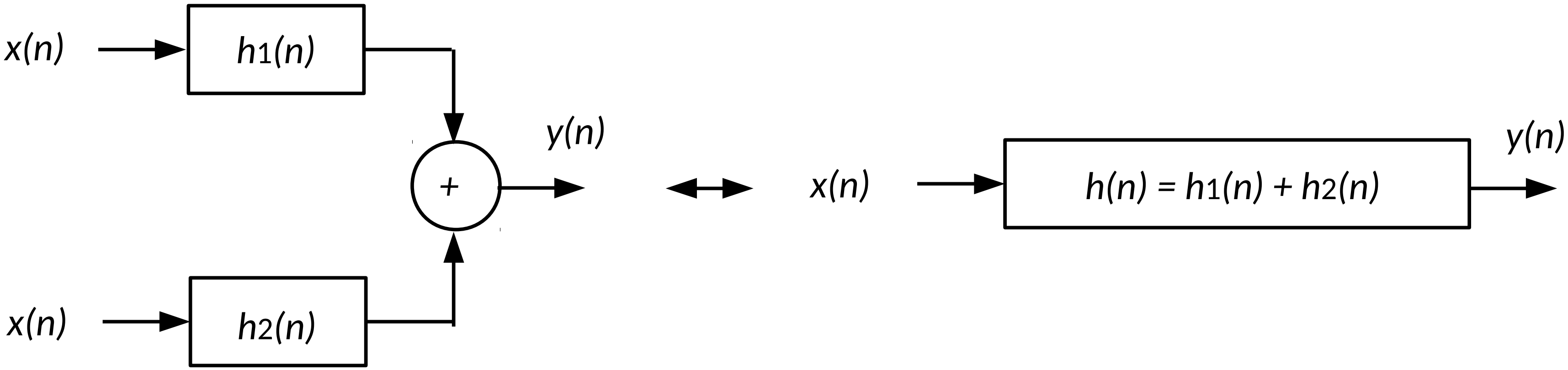


Convolution properties

- Convolution properties define connections of LTI systems

  - Convolution is distributive over addition → *Parallel connection*

$$x(n) * h_1(n) + x(n) * h_2(n) = x(n) * [h_1(n) + h_2(n)]$$



Convolution properties

- Example of exam task
  - $y_1 = x_1(n) - x_1(n\text{-}1)$, $y_2 = x_2(n) + x_2(n\text{-}1)$; Derive $h_1(n)$, $h_2(n)$, and $h_c(n)$ for cascade and $h_p(n)$ for parallel connection. Check commutativity for $h_1(n)$ and $h_2(n)$
- Example of exam task
  - Check the properties of output signal for cascade and parallel connection in the following case: F_s = 6000 smp/sec, h_1 passes frequencies from 0 – 250 Hz, h_2 passes from 1700 – 2400 Hz

Examples of LTI systems

- Example of finite impulse response LTI system, simple lowpass filter

- $y_L(n) = 1/2\, x(n) + 1/2\, x(n - 1)$

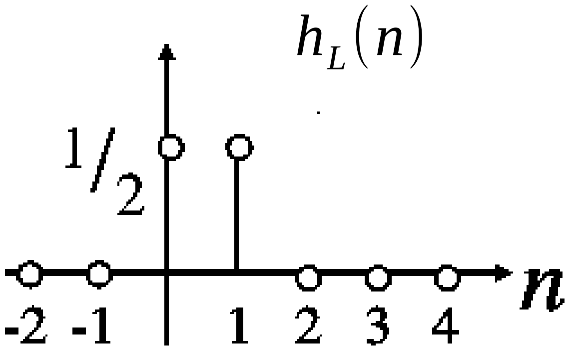
→ * Moving average (first order),

* Equalizes values of neighboring samples,

* Lowpass filtering effect,

* Keeps slowly varying trend of values of distant samples

* Rejects rapid changes of values of neighboring samples



→ *Smoothing effect*

$h_L(n) = 1/2\, \delta(n) + 1/2\, \delta(n - 1)$

$h_L(n) = \{1/2, 1/2\}$

Examples of LTI systems

- Example of finite impulse response LTI system, simple lowpass filter
- Repetitive use of *simple (first-order) moving average lowpass filter*, $h_L(n)$, will increase the *smoothing effect*

Since convolution is an associative operation

$$[\ x(n) \ast h_L(n)\ ] \ast h_L(n) \ = \ x(n) \ast [\ h_L(n) \ast h_L(n)\ ]$$

$$x(n) \longrightarrow \boxed{h_L(n)} \longrightarrow \boxed{h_L(n)} \longrightarrow y(n) \quad \longleftrightarrow \quad x(n) \longrightarrow \boxed{h_L(n) * h_L(n)} \longrightarrow y(n)$$

follows that the impulse response of the (second-order) moving average filter, $h_{L2}(n)$, which will increase the lowpass filtering effect, could be obtained by convolving $h_H(n)$ by itself

$$h_{L2}(n) = h_L(n) * h_L(n) = \{1/2,\ 1/2\} * \{1/2,\ 1/2\} = \{1/4,\ 1/2,\ 1/4\}$$

Verify the lowpass filtering effect of $h_{L2}(n)$ at http://www.fourier-series.com/

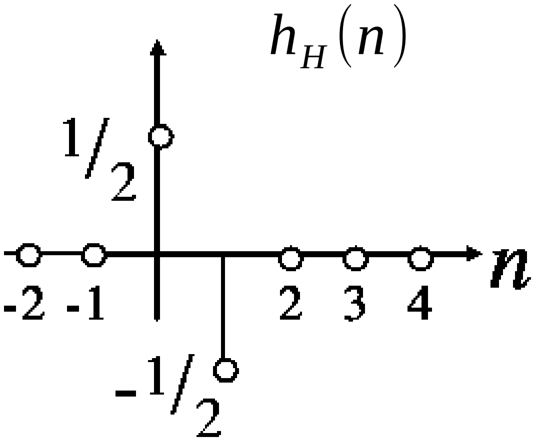
→ $h_{L2}(n)$ - principal impulse response for smoothing spatial filters (*image smoothing*)

Examples of LTI systems

- Example of finite impulse response LTI system, simple highpass filter
- $y_H(n) = 1/2\, x(n) - 1/2\, x(n-1)$

→ * Differentiator (first order),

* Preserves differences of values of neighboring samples,
* Highpass filtering effect,
* Keeps rapid changes of values of neighboring samples
* Rejects slowly varying trend of values of distant samples



→ *Sharpening effect*

$h_H(n) = 1/2\, \delta(n) - 1/2\, \delta(n-1)$

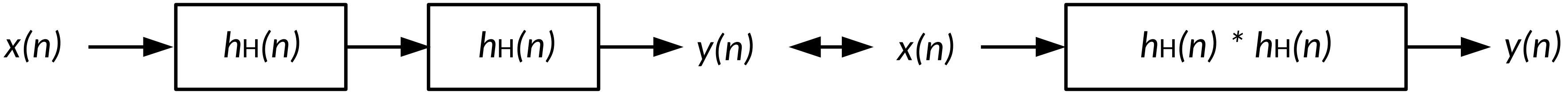
$h_H(n) = \{1/2, -1/2\}$

Examples of LTI systems

- Example of finite impulse response LTI system, simple highpass filter
- Repetitive use of *simple first-order differentiator highpass filter*, $h_H(n)$, will increase the *sharpening effect*

Since convolution is an associative operation

$$[\, x(n) \ast h_H(n) \,] \ast h_H(n) = x(n) \ast [\, h_H(n) \ast h_H(n) \,]$$



follows that the impulse response of the second-order differentiator filter, $h_{H2}(n)$, which will increase the highpass filtering effect, could be obtained by convolving $h_H(n)$ by itself

$$h_{H2}(n) = h_H(n) * h_H(n) = \{1/2, - 1/2\} * \{1/2, - 1/2\} = \{1/4, - 1/2, 1/4\}$$

Verify the highpass filtering effect of $h_{H2}(n)$ at http://www.fourier-series.com/

→ $h_{H2}(n)$ - principal impulse response for sharpening spatial filters (*image sharpening*)

Examples of LTI systems

- Further study, laboratory

1) Using the impulse response, $h_L(n) = \{1/2, 1/2\}$

of simple moving average lowpass filter, $y_L(n) = 1/2\, x(n) + 1/2\, x(n - 1)$,

derive the impulse response of the fourth order, i.e.,

$$h_{L4}(n) = h_L(n) * h_L(n) * h_L(n) * h_L(n)$$

Verify lowpass filtering effect of the filter $h_{L4}(n)$ at http://www.fourier-series.com/

2) Using the impulse response, $h_H(n) = \{1/2, - 1/2\}$

of simple differentiator highpass filter, $y_H(n) = 1/2\, x(n) - 1/2\, x(n - 1)$,

derive the impulse response of the fourth order, i.e.,

$$h_{H4}(n) = h_H(n) * h_H(n) * h_H(n) * h_H(n)$$

Verify highpass filtering effect of the filter $h_{H4}(n)$ at http://www.fourier-series.com/

Impulse response and stability

- An LTI system is said to be Bounded Input Bounded Output (BIBO) stable if and only if every bounded input produces a bounded output

$$\exists B_x, B_y: \quad |x(n)| \leqslant B_x < \infty, \quad |y(n)| \leqslant B_y < \infty$$

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

$$|y(n)| = \left|\sum_{k=-\infty}^{\infty} h(k)x(n-k)\right|$$

$$|y(n)| \leq \sum_{k=-\infty}^{\infty} |h(k)||x(n-k)|$$

$$|y(n)| \leq B_x \sum_{k=-\infty}^{\infty} |h(k)| \leq B_y$$

- An LTI system is stable, $B_y < \infty$, if and only if $\sum_{k=-\infty}^{\infty} |h(k)| < \infty$

- The system can give meaningful output only if it is stable

(Additional materials)

- Implementation of convolution
- Convolution, matrix interpretation
- Convolution, example
- Convolution, example
- Convolution in MATLAB
- Convolution and causality

Implementation of convolution

- Convolution sum for LTI systems

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

- Summation is symmetric in x and h

$$\text{i.e.} \quad l = n-k \rightarrow$$

$$x(n) * h(n) = \sum_{l=-\infty}^{\infty} x(n-l)\, h(l)$$

$$= \sum_{l=-\infty}^{\infty} h(l)\, x(n-l) = h(n) * x(n)$$

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

Demo: https://engineering.purdue.edu/VISE/ee438/demos/

Convolution, matrix interpretation

- Convolution interpretation 2, matrix interpretation

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

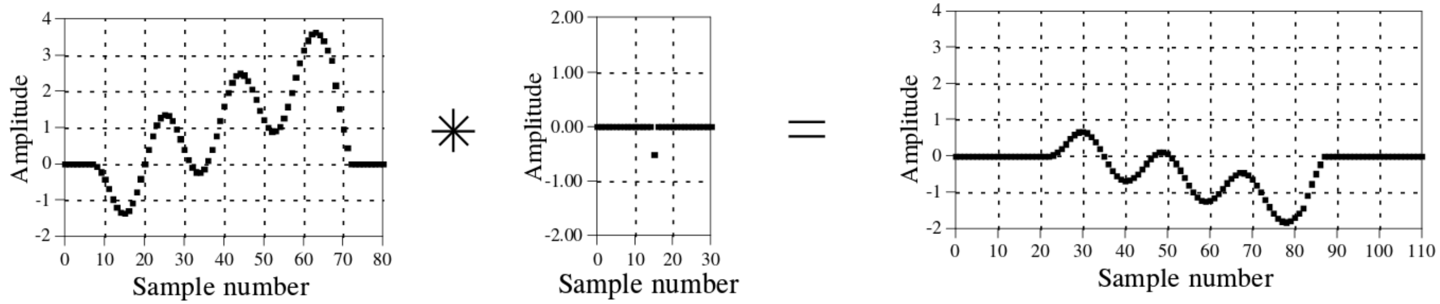
$$\begin{bmatrix} y(0) \\ y(1) \\ y(2) \\ \dots \end{bmatrix} = \begin{bmatrix} x(0) & x(-1) & x(-2) \\ x(1) & x(0) & x(-1) \\ x(2) & x(1) & x(0) \\ \dots & \dots & \dots \end{bmatrix} \begin{bmatrix} h(0) \\ h(1) \\ h(2) \end{bmatrix}$$

Diagonals in **X** matrix are equal

Convolution, example

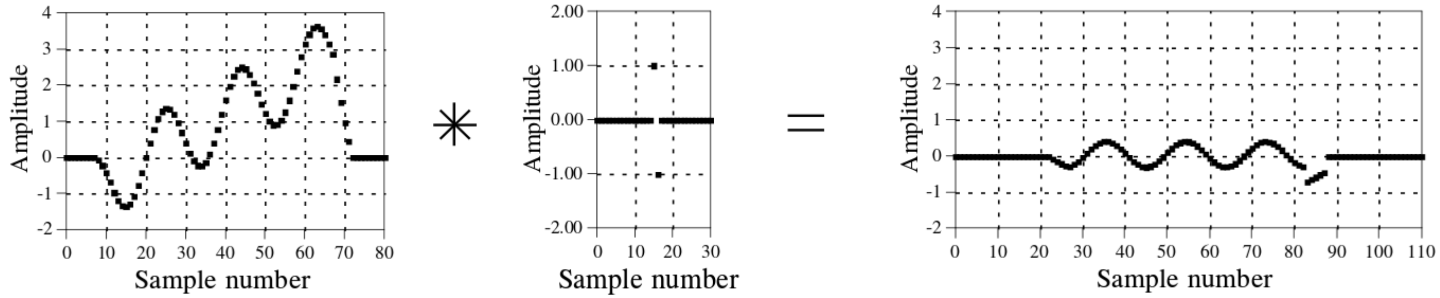
- Convolution example, inverting attenuator and discrete derivative

a. Inverting Attenuator



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

b. Discrete Derivative



Demo: http://www.fourier-series.com/

[GDSP]

Convolution, example

- Example

Derive (analytically) convolution for the following example

- Rectangular pulse of duration N

$$x[n] = u[n] - u[n-N] = \begin{cases} 1 & \text{if } 0 \leq n \leq N-1 \\ 0 & \text{otherwise} \end{cases}$$

is passed to the first-order low-pass filter of which impulse response is

$$h[n] = a^n u[n]$$

Solution: For $n < 0$ $y[n] = 0$

For $n \leq 0 \leq N - 1$

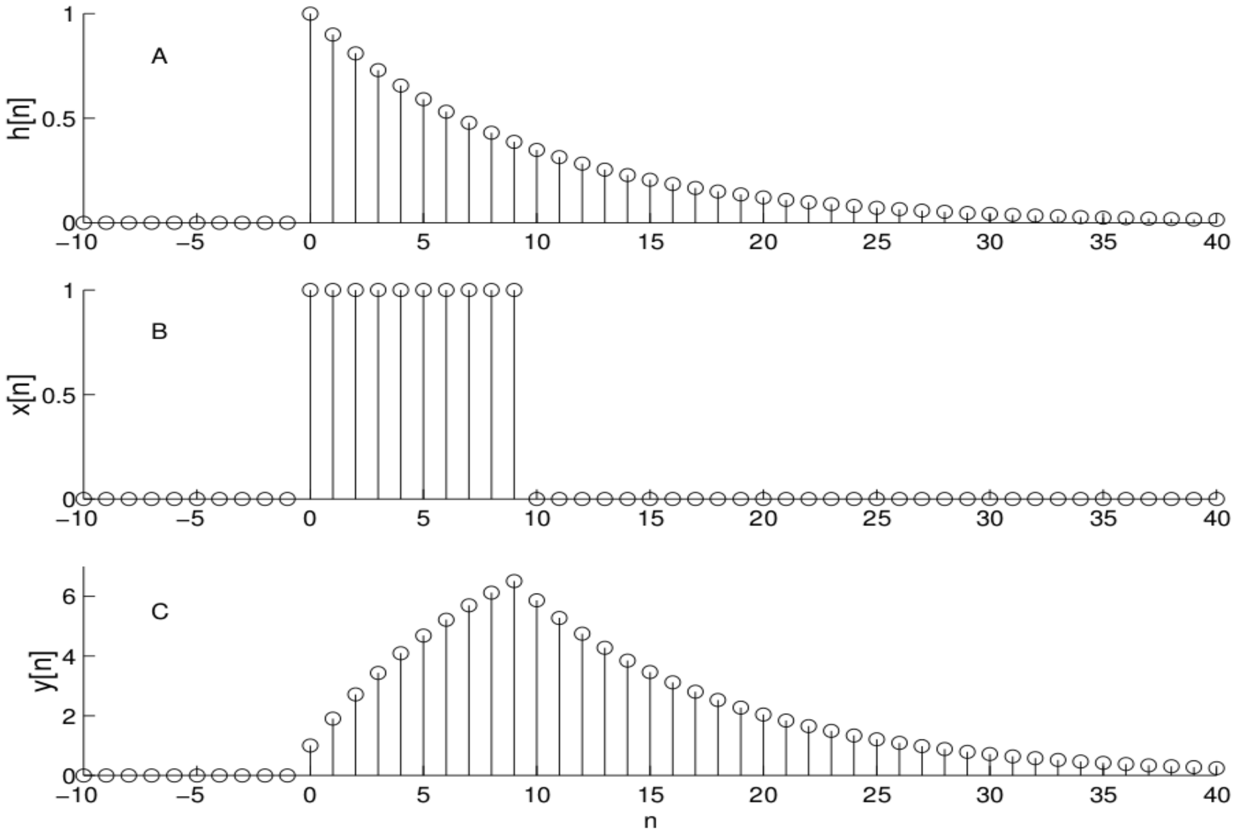
$$y[n] = \sum_{m=0}^{n} a^m = \frac{1 - a^{n+1}}{1 - a}$$

For $n \geq N$

$$y[n] = \sum_{m=n-N+1}^{n} a^m = a^n - N + 1 \frac{1 - a^N}{1 - a} = \frac{1 - a^- N}{1 - a^- 1} a^n$$

Convolution, example

- Example



Convolution example: (A) Unit-sample response of the first-order low-pass filter, $h[n] = a^n u[n]$, with $a = 0.9$. (B) Input signal, $x[n] = u[n] - u[n - N]$, with $N = 10$. (C) Output signal.

Convolution in MATLAB

- Convolution in MATLAB

  - The function conv() implements the convolution sum of two finite-length sequences

- *If* x = [0 4 1 -2 5]

  h = [1 2 3 1]

  then conv(x, h) *yields*

  [0 4 9 12 8 5 13 5]

Convolution and causality

- An LTI system is causal if and only if *h(n)* = 0 for *n* < 0

  - Convolution

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

  - Causal LTI system (*h(n)* causal) and causal input signal *x(n)*

$$y(n) = \sum_{k=0}^{n} x(k)\,h(n-k) = \sum_{k=0}^{n} h(k)\,x(n-k)$$