CSx25: Digital Signal Processing
NCS224: Signals and Systems

Dr. Ahmed Shalaby

http://bu.edu.eg/staff/ahmedshalaby14#

Outline

- ~~Digital Signal Processing Introduction~~
    - ~~Mathematical modeling~~
    - ~~Continuous Time Signals~~
    - ~~Discrete Time Signals~~
- ~~Analyzing Continuous-Time Systems in the Time Domain~~
- **Analyzing Discrete Systems in the Time Domain**

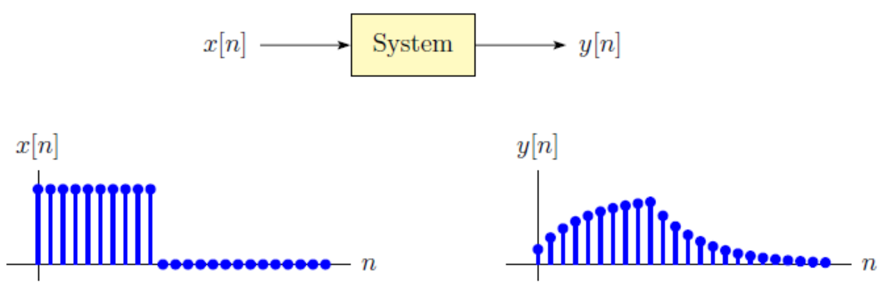
2.1 Intro to System

Introduction

System

In general, a discrete-time system is a mathematical formula, method or algorithm that defines a cause-effect relationship between a set of discrete-time input signals and a set of discrete-time output signals.

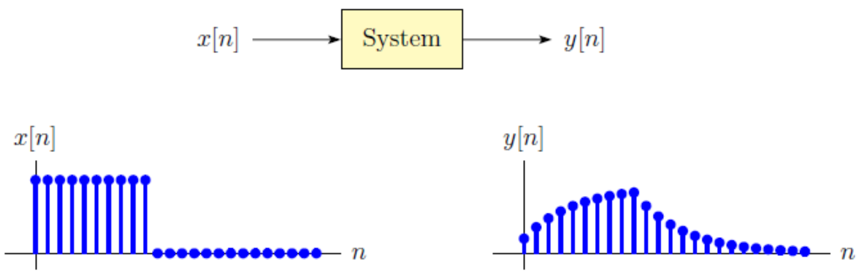
Illustration of signal-system interaction involving a single-input single-output discrete-time system.



$$y[n] = \text{Sys}\{x[n]\}$$

2.1 Intro to System

Introduction (continued)



Some examples:

$$y[n] = K\,x[n]$$

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

$$y[n] = K\,\left[x[n]\right]^2$$

2.2 Linearity and Time Invariance

Linearity in discrete-time systems

Conditions for linearity

$$\text{Sys}\{x_1[n] + x_2[n]\} = \text{Sys}\{x_1[n]\} + \text{Sys}\{x_2[n]\}$$

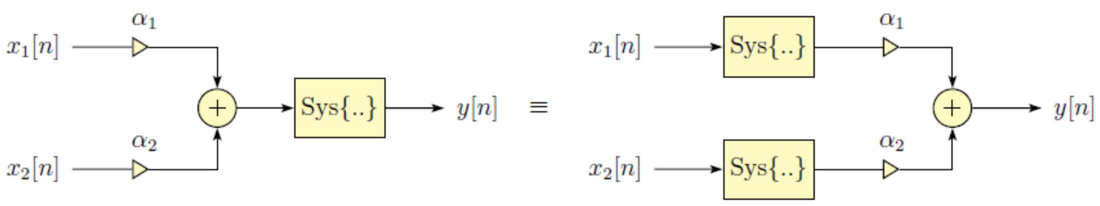
$$\text{Sys}\{\alpha_1\, x_1[n]\} = \alpha_1\ \text{Sys}\{x_1[n]\}$$

$x_1[n]$, $x_2[n]$: Any two signals; α_1: Arbitrary constant gain factor.

Superposition principle (combine the two conditions into one)

$$\text{Sys}\{\alpha_1\, x_1[n] + \alpha_2\, x_2[n]\} = \alpha_1\ \text{Sys}\{x_1[n]\} + \alpha_2\ \text{Sys}\{x_2[n]\}$$

$x_1[n]$ and $x_2[n]$: Any two signals; α_1, α_2: Arbitrary constant gain factors.

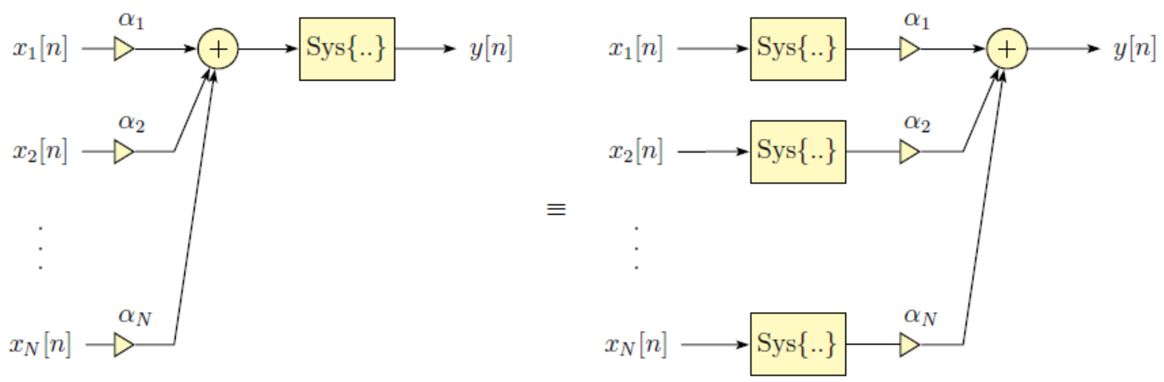


2.2 Linearity and Time Invariance

Linearity in discrete-time systems (continued)

If superposition works for the weighted sum of any two input signals, it also works for an arbitrary number of input signals.

$$\mathrm{Sys}\left\{\sum_{i=1}^{N} \alpha_i\, x_i[n]\right\} = \sum_{i=1}^{N} \alpha_i\ \mathrm{Sys}\left\{x_i[n]\right\} = \sum_{i=1}^{N} \alpha_i\, y_i[n]$$



2.2 Linearity and Time Invariance

Example 3.1

Testing linearity in discrete-time systems

For each of the discrete-time systems described below, determine whether the system is linear or not:

a. $y[n] = 3\,x[n] + 2x[n-1]$

b. $y[n] = 3\,x[n] + 2\,x[n+1]\,x[n-1]$

c. $y[n] = a^{-n}\,x[n]$

<u>Solution:</u>

a.

$$\begin{aligned} y[n] &= 3\,x[n] + 2x[n-1] \\ &= 3\,(\alpha_1\,x_1[n] + \alpha_2\,x_2[n]) + 2\,(\alpha_1\,x_1[n-1] + \alpha_2\,x_2[n-1]) \\ &= \alpha_1\,(3\,x_1[n] + 2\,x_1[n-1]\,) + \alpha_2\,(3\,x_2[n] + 2\,x_2[n-1]\,) \\ &= \alpha_1\,y_1[n] + \alpha_2\,y_2[n] \end{aligned}$$

Superposition principle holds, and therefore the system is linear.

2.2 Linearity and Time Invariance

Example 3.1 (continued)

b.

$$
\begin{aligned}
y[n] =& 3\,x[n] + 2\,x[n+1]\,x[n-1] \\
=& 3\,(\alpha_1\,x_1[n] + \alpha_2 x_2[n]) \\
&+ 2\,(\alpha_1\,x_1[n+1] + \alpha_2\,x_2[n+1])\,(\alpha_1\,x_1[n-1] + \alpha_2\,x_2[n-1])
\end{aligned}
$$

Superposition principle does not hold true. The system in part (b) is not linear.

c.

$$
\begin{aligned}
y[n] =& a^{-n}\,x[n] \\
=& a^{-n}\,(\alpha_1\,x_1[n] + \alpha_2\,x_2[n]) \\
=& \alpha_1\,a^{-n}\,x_1[n] + \alpha_2\,a^{-n}\,x_2[n] \\
=& \alpha_1\,y_1[n] + \alpha_2\,y_2[n]
\end{aligned}
$$

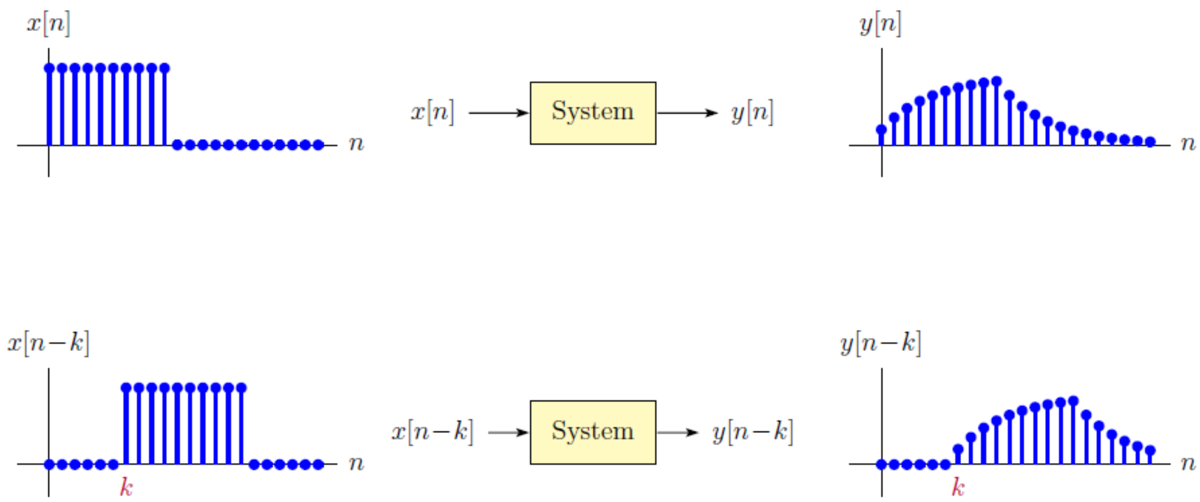
Superposition principle holds. The system in part (c) is linear.

2.2 Linearity and Time Invariance

Time-invariance in discrete-time systems

Condition for time-invariance

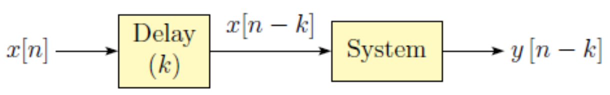
$$\text{Sys}\{x[n]\} = y[n] \quad \text{implies that} \quad \text{Sys}\{x[n-k]\} = y[n-k]$$

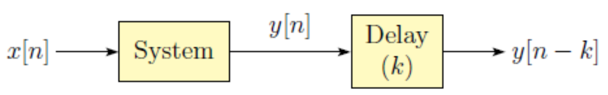


2.2 Linearity and Time Invariance

Time-invariance in discrete-time systems (continued)

Alternatively, time-invariance can be explained by the equivalence of the two system configurations shown:





2.2 Linearity and Time Invariance

Example 3.2

Testing time-invariance in discrete-time systems

For each of the discrete-time systems described below, determine whether the system is time-invariant or not:

a. $y[n] = y[n-1] + 3\,x[n]$

b. $y[n] = x[n]\,y[n-1]$

c. $y[n] = n\,x[n-1]$

<u>Solution:</u>

a. $\text{Sys}\left\{x[n-k]\right\} = y[n-k-1] + 3\,x[n-k] = y[n-k]$ Time-invariant.

b. $\text{Sys}\left\{x[n-k]\right\} = x[n-k]\,y[n-k-1] = y[n-k]$ Time-invariant.

c. $\text{Sys}\left\{x[n-k]\right\} = n\,x[n-k-1] \neq y[n-k]$ Not time-invariant.

2.3 Difference Equations for Discrete-Time Systems

we have discussed methods of representing continuous-time systems with ***differential equations***. Using a similar approach, discrete-time systems can be modeled with ***difference equations*** involving current, past, or future samples of input and output signals.

In the sections that follow we will focus our attention on ***difference equations*** for DTLTI systems and develop solution techniques for them using an approach that parallels our study of differential equations for CTLTI systems (***RCL Circuits – differential equation***)

A ***length-N moving average filter*** is a simple system that produces an output equal to the arithmetic average of the most recent N samples of the input signal.

2.3 Difference Equations for Discrete-Time Systems

Example 3.3

Moving-average filter

Explore the operation of a length-N moving average filter.

Solution:

The output signal is the arithmetic average of the most recent N samples of the input signal.

Compute $y[100]$:

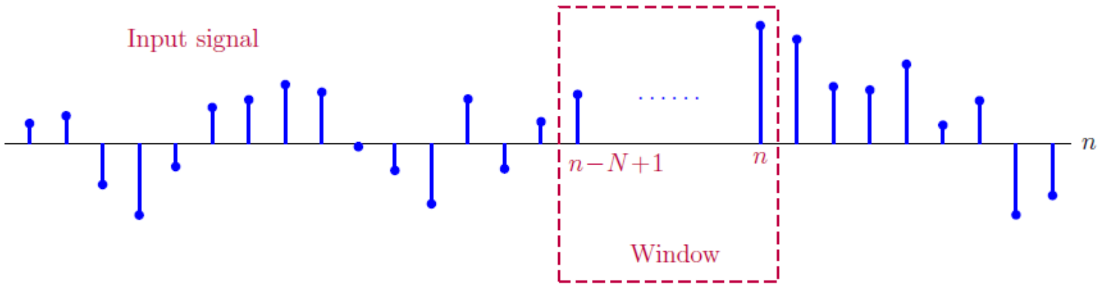
$$y[100] = \frac{x[100] + x[99] + \ldots + x[100-(N-1)]}{N} = \frac{1}{N}\sum_{k=0}^{N-1} x[100-k]$$

The general expression for the length-N moving average filter:

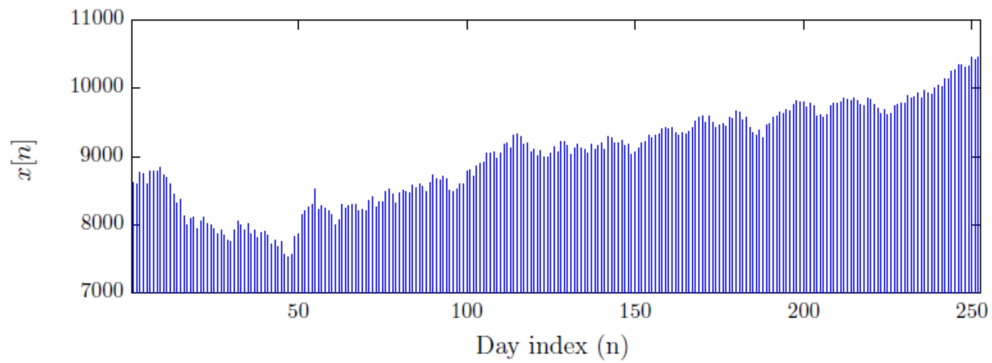
$$y[n] = \frac{x[n] + x[n-1] + \ldots + x[n-(N-1)]}{N} = \frac{1}{N}\sum_{k=0}^{N-1} x[n-k]$$

2.3 Difference Equations for Discrete-Time Systems

Example 3.3 (continued)



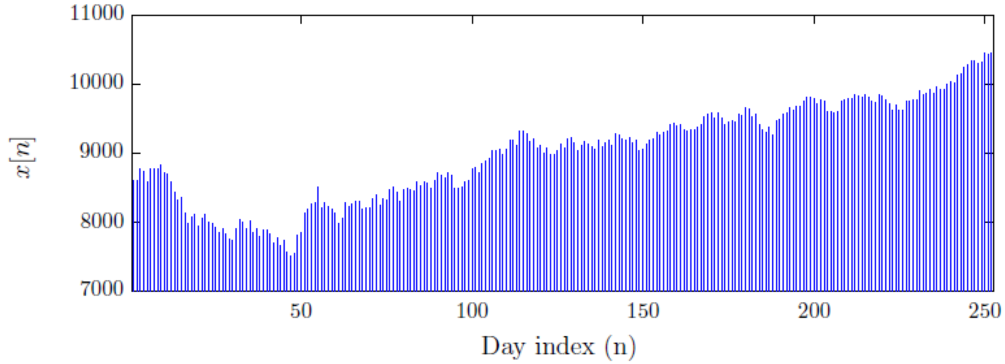
Example: Dow Jones index



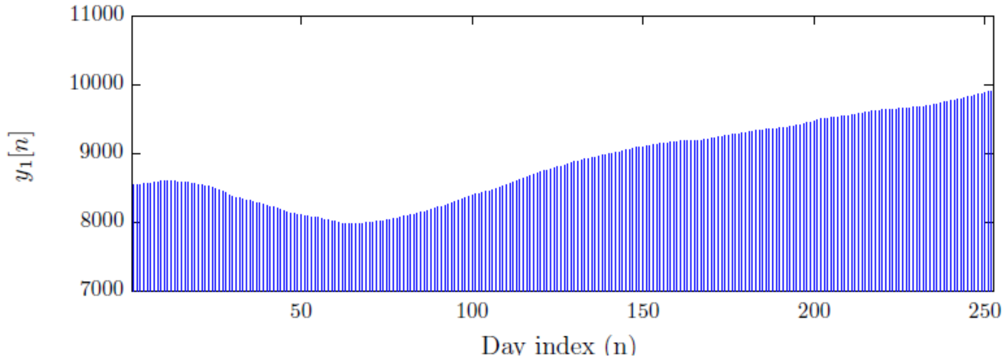
2.3 Difference Equations for Discrete-Time Systems

Example 3.3 (continued)

Example: 50-Day moving average of the Dow Jones index



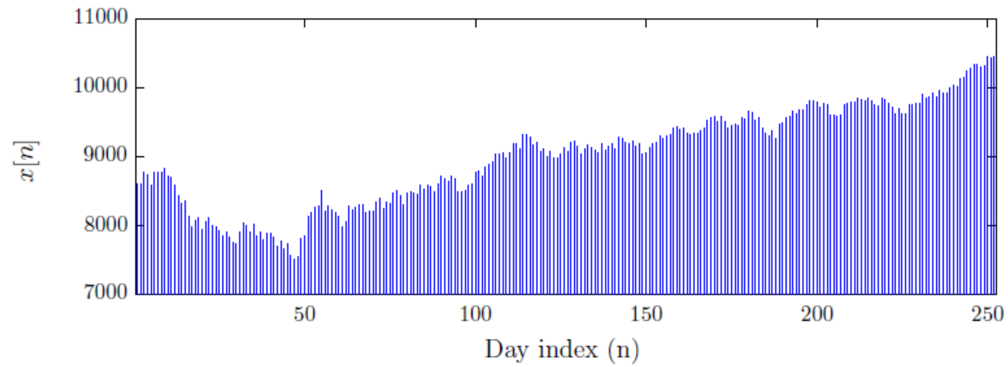
$$y_1[n] = \frac{1}{50} \sum_{k=0}^{49} x[n-k]$$



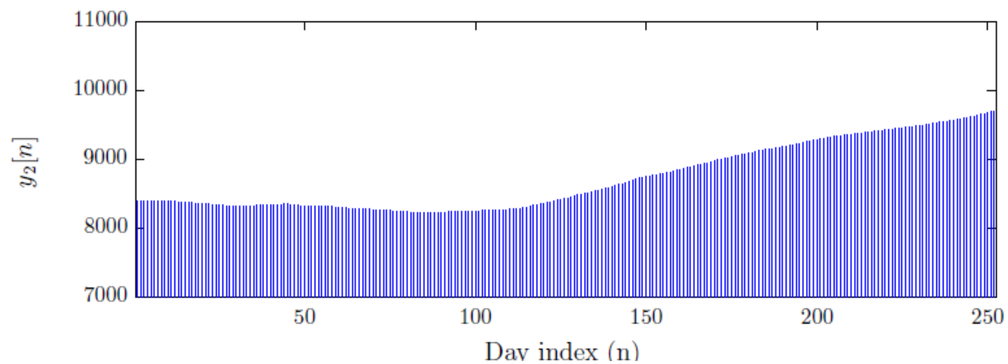
2.3 Difference Equations for Discrete-Time Systems

Example 3.3 (continued)

Example: 100-Day moving average of the Dow Jones index



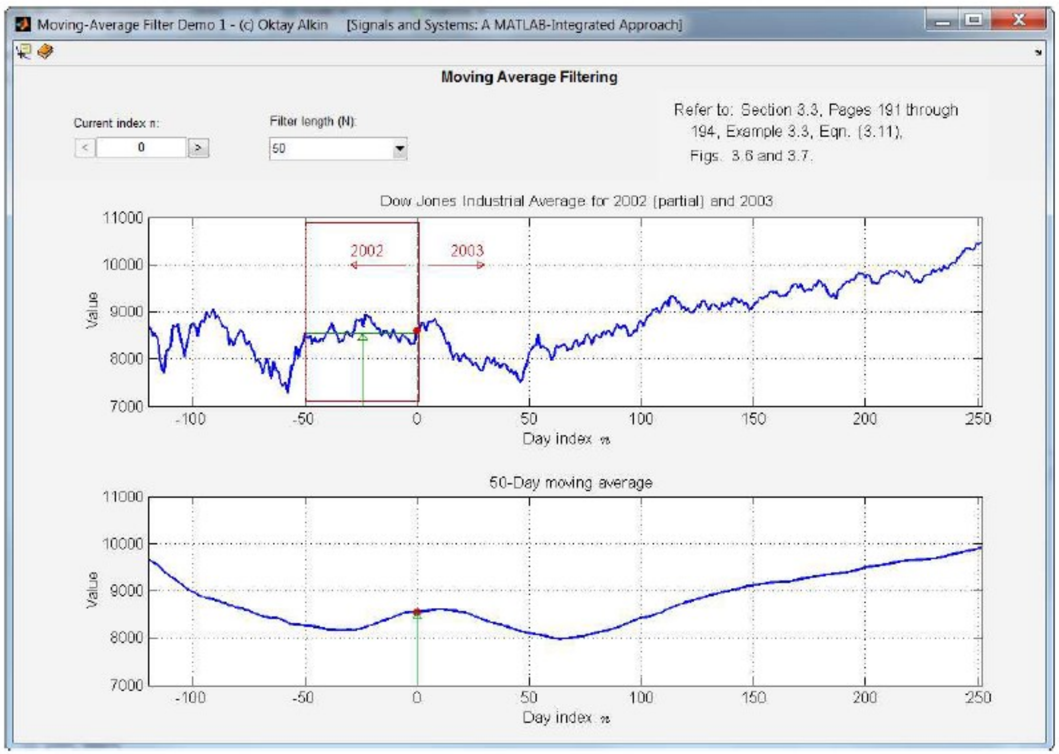
$$y_2[n] = \frac{1}{100} \sum_{k=0}^{99} x[n-k]$$



2.3 Difference Equations for Discrete-Time Systems

Interactive demo: ma_demo1.m

Observe the operation of the length-N moving average filter discussed in Example 3.3.



2.3 Difference Equations for Discrete-Time Systems

Example 3.4

Length-2 moving-average filter

Explore the operation of a length-2 moving average filter.

Solution:

A length-2 moving average filter produces an output by averaging the current input sample and the previous input sample.

$$y_[n] = \frac{1}{2}\,x[n] + \frac{1}{2}\,x[n-1]$$

For a few values of the index we can write the following set of equations:

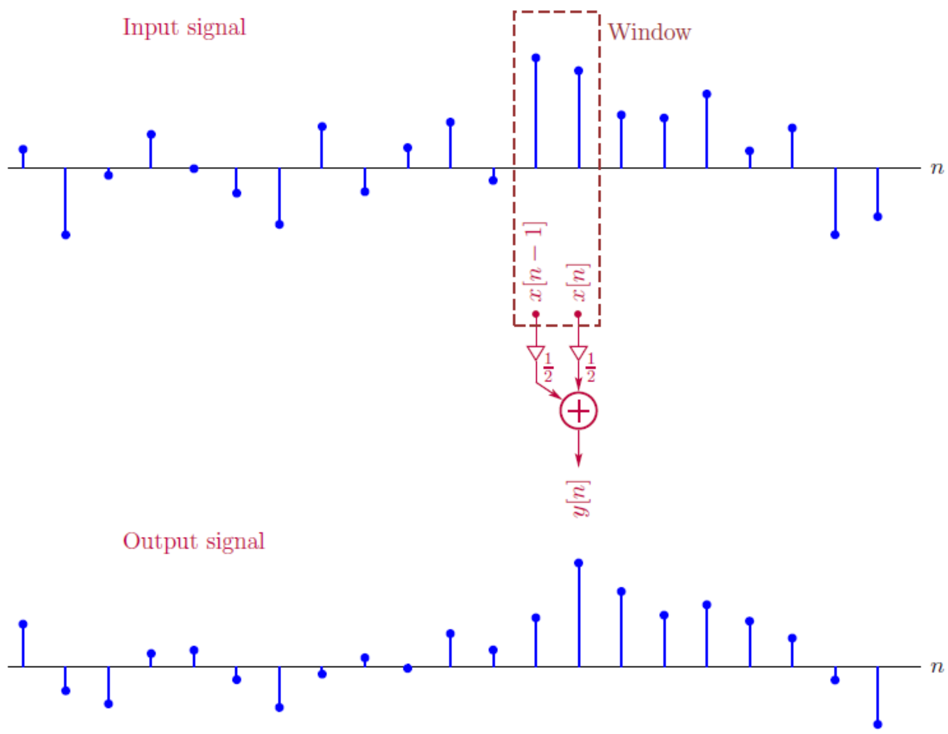
$$n = 0: \qquad y[0] = \tfrac{1}{2}\,x[0] + \tfrac{1}{2}\,x[-1]$$

$$n = 1: \qquad y[1] = \tfrac{1}{2}\,x[1] + \tfrac{1}{2}\,x[0]$$

$$n = 2: \qquad y[2] = \tfrac{1}{2}\,x[2] + \tfrac{1}{2}\,x[1]$$

2.3 Difference Equations for Discrete-Time Systems

Example 3.4 (continued)



2.3 Difference Equations for Discrete-Time Systems

Example 3.5

Length-4 moving average filter

Explore the operation of a length-4 moving average filter.

Solution:

A length-4 moving average filter produces an output by averaging the current input sample and the previous three input samples.

$$y_[n] = \frac{1}{4}\,x[n] + \frac{1}{4}\,x[n-1] + \frac{1}{4}\,x[n-2] + \frac{1}{4}\,x[n-3]$$

For a few values of the index we can write the following set of equations:

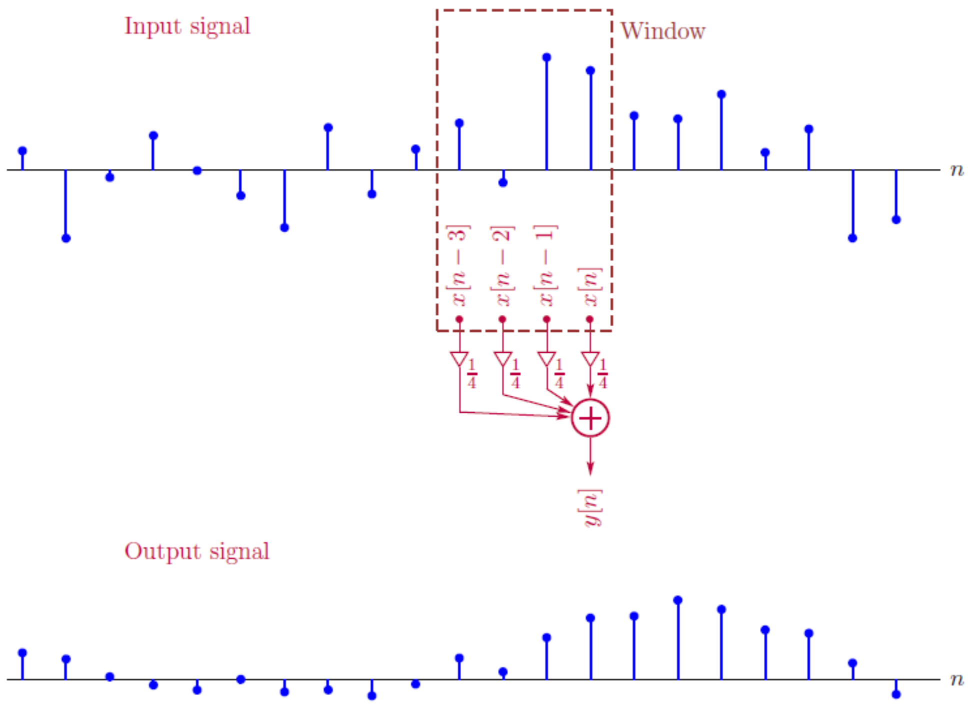
$$n = 0: \qquad y[0] = \tfrac{1}{4}\,x[0] + \tfrac{1}{4}\,x[-1] + \tfrac{1}{4}\,x[-2] + \tfrac{1}{4}\,x[-3]$$

$$n = 1: \qquad y[1] = \tfrac{1}{4}\,x[1] + \tfrac{1}{4}\,x[0] + \tfrac{1}{4}\,x[-1] + \tfrac{1}{4}\,x[-2]$$

$$n = 2: \qquad y[2] = \tfrac{1}{4}\,x[2] + \tfrac{1}{4}\,x[1] + \tfrac{1}{4}\,x[0] + \tfrac{1}{4}\,x[-1]$$

2.3 Difference Equations for Discrete-Time Systems

Example 3.5 (continued)



▸ MATLAB Exercise 2.1

▸ MATLAB Exercise 2.2

2.3 Difference Equations for Discrete-Time Systems

Example 3.6

Exponential smoother

Explore the operation of an exponential smoother.

Solution:

The current output sample is computed as a mix of the current input sample and the previous output sample through the equation

$$y[n] = (1 - \alpha)\, y[n-1] + \alpha\, x[n]$$

The parameter α is a constant in the range $0 < \alpha < 1$, and it controls the degree of smoothing.

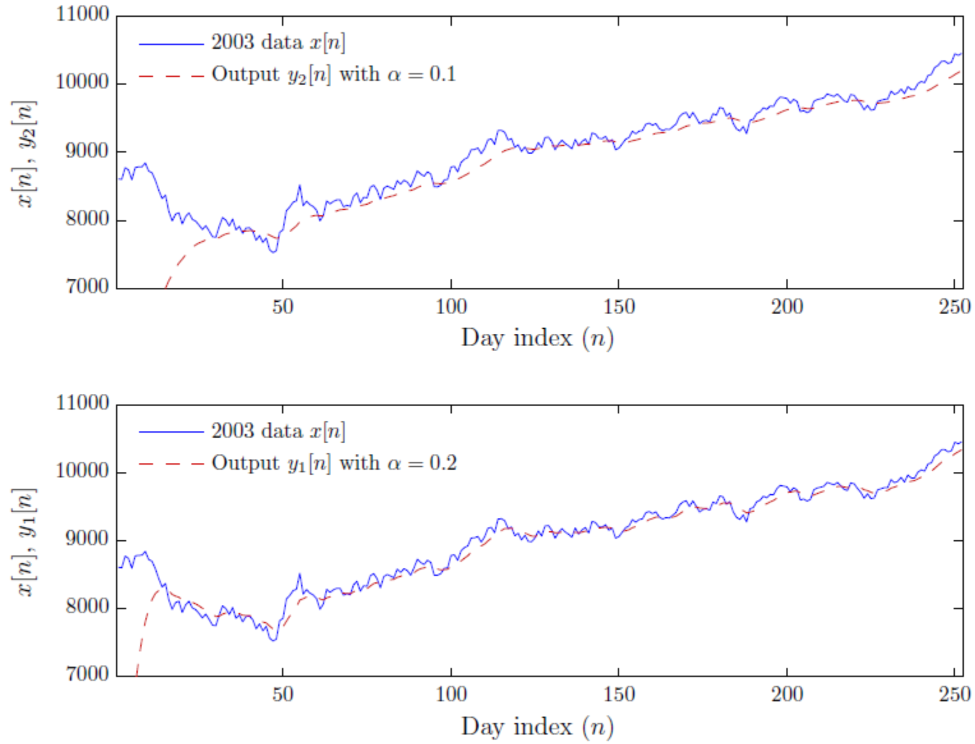
$$n = 0: \qquad y[0] = (1 - \alpha)\, y[-1] + \alpha\, x[0]$$

$$n = 1: \qquad y[1] = (1 - \alpha)\, y[0] + \alpha\, x[1]$$

$$n = 2: \qquad y[2] = (1 - \alpha)\, y[1] + \alpha\, x[2]$$

2.3 Difference Equations for Discrete-Time Systems

Example 3.6 (continued)



▸ MATLAB Exercise 2.3

2.4 Constant-Coefficient Linear Difference Equations

Constant-coefficient difference equations (continued)

Constant-coefficient difference equation for a DTLTI system

The difference equation

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

represents a DTLTI system provided that all initial conditions are equal to zero:

$$y[n_0-1]=0\,,\quad y[n_0-2]=0\,,\quad \ldots,\quad y[n_0-N]=0$$

The order of the difference equation (and therefore the order of the system it represents) *is the larger of N and M*. For example, the length-2 moving average filter discussed in Example 3.4 is a *first-order system*.

$$y[n] = \frac{1}{2}\,x[n] + \frac{1}{2}\,x[n-1]$$

2.5 Solving Difference Equations

Iteratively solving difference equations

Consider the difference equation for the exponential smoother of Example 3.6:

$$y[n] = (1-\alpha)\, y[n-1] + \alpha\, x[n]$$

- Given the initial value $y[-1]$ of the output signal, $y[0]$ is found by

$$y[0] = (1-\alpha)\, y[-1] + \alpha\, x[0]$$

- Knowing $y[0]$, the next output sample $y[1]$ is found by

$$y[1] = (1-\alpha)\, y[0] + \alpha\, x[1]$$

- Knowing $y[1]$, the next output sample $y[2]$ is found by

$$y[2] = (1-\alpha)\, y[1] + \alpha\, x[2]$$

- Continue for as many samples as needed.

MATLAB Exercise 3.4

2.5 Solving Difference Equations

Solution of the general difference equation

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

Initial conditions:

$$y[n_0 - 1] \, , \quad y[n_0 - 2] \, , \quad \ldots , \quad y[n_0 - N]$$

General solution:

$$y[n] = y_h[n] + y_p[n]$$

- $y_h[n]$ is the *homogeneous solution* of the difference equation (natural response).
- $y_p[n]$ is the *particular solution* of the difference equation.
- $y[n] = y_h[n] + y_p[n]$ is the *forced solution* of the difference equation (forced response).

2.5 Solving Difference Equations

Finding the forced response of a discrete-time system

Choosing a particular solution for various discrete-time input signals

Input signal	Particular solution
K (constant)	k_1
$K\, e^{an}$	$k_1\, e^{an}$
$K\, \cos(\Omega_0 n)$	$k_1\, \cos(\Omega_0 n) + k_2\, \sin(\Omega_0 n)$
$K\, \sin(\Omega_0 n)$	$k_1\, \cos(\Omega_0 n) + k_2\, \sin(\Omega_0 n)$
$K\, n^m$	$k_m n^m + k_{m-1} n^{m-1} + \ldots + k_1 n + k_0$

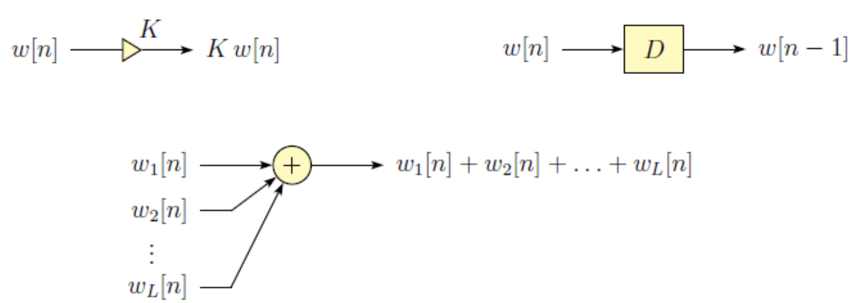
2.6 Block Diagram Representation of Discrete-Time Systems

Block diagram representation of discrete-time systems

- A discrete-time system can also be represented with a block diagram, and multiple solutions exist that are functionally equivalent.
- Block diagrams provide additional insight into the operation of a system.
- We often use a block diagram as the first step in developing the computer code for implementing a discrete-time system.

Block diagrams for discrete-time systems are constructed using three types of components:

- Constant-gain amplifiers
- Signal adders
- Delay elements

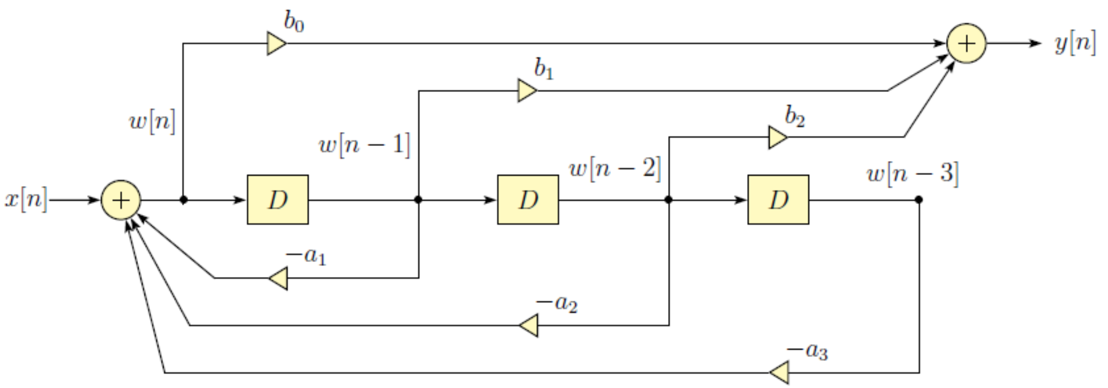


2.6 Block Diagram Representation of Discrete-Time Systems

Block diagram representation of discrete-time systems (continued)

It can be shown that

$$y[n] = b_0\, w[n] + b_1\, w[n-1] + b_2\, w[n-2]$$



2.6 Block Diagram Representation of Discrete-Time Systems

Example 3.16

Block diagram for discrete-time system

Construct a block diagram to solve the difference equation

$$y[n] - 0.7\,y[n-1] - 0.8\,y[n-2] + 0.84\,y[n-3] = 0.1\,x[n] + 0.2\,x[n-1] + 0.3\,x[n-2]$$

with the input signal $x[n] = u[n]$ and subject to initial conditions

$$y[-1] = 0.5\;, \quad y[-2] = 0.3\;, \quad y[-3] = -0.4$$

Solution:

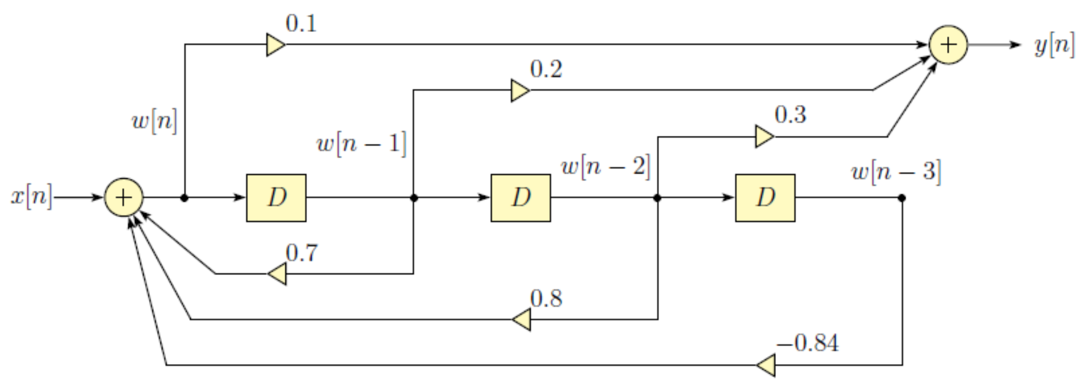
Using the intermediate variable $w[n]$:

$$w[n] - 0.7\,w[n-1] - 0.8\,w[n-2] + 0.84\,w[n-3] = x[n]$$

$$y[n] = 0.1\,w[n] + 0.2\,w[n-1] + 0.3\,w[n-2]$$

2.6 Block Diagram Representation of Discrete-Time Systems

Example 3.16 (continued)



Initial conditions specified in terms of the values of $y[-1]$, $y[-2]$ and $y[-3]$ need to be translated to corresponding values of $w[-1]$, $w[-2]$ and $w[-3]$:

$$w[-1] = 1.0682\,, \quad w[-2] = 1.7149\,, \quad w[-3] = 0.1674$$

In the block diagram the outputs of the three delay elements should be set equal to these values before starting the simulation.

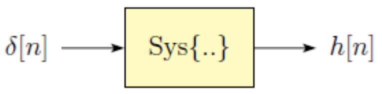
▸ MATLAB Exercise 3.5

▸ MATLAB Exercise 3.6

2.7 Impulse Response and Convolution

Impulse response

$$h[n] = \text{Sys}\{\delta[n]\}$$



For a DTLTI system: The impulse response also constitutes a complete description of a DTLTI system.

Finding the impulse response from the difference equation:

- Finding the impulse response of a DTLTI system amounts to finding the forced response of the system when the forcing function is a unit-impulse:

$$x[n] = \delta[n]$$

- In the case of a difference equation with no feedback, the impulse response is found by direct substitution of the input signal $x[n] = \delta[n]$ into the difference equation.
- If the difference equation has feedback, then it is easier to first find the unit-step response of the system as an intermediate step, and to determine the impulse response from the unit-step response.
- A more practical method will be presented in Chapter 8 through the use of the z-transform.

2.7 Impulse Response and Convolution

Example 3.17

Impulse response of moving average filters

Find the impulse response of the length-2, length-4 and length-N moving average filters.

Solution:

Difference equation for length-2 moving average filter:

$$y[n] = \frac{1}{2}\,x[n] + \frac{1}{2}\,x[n-1]$$

Let $h_2[n]$ denote the impulse response of the length-2 moving average filter.

$$h_2[n] = \text{Sys}\left\{\delta[n]\right\} = \frac{1}{2}\,\delta[n] + \frac{1}{2}\,\delta[n-1]$$

In tabular form:

$$h_2[n] = \{\underset{\uparrow}{1/2},\ 1/2\}$$

2.7 Impulse Response and Convolution

Convolution operation for DTLTI systems

The output signal $y[n]$ of a DTLTI system is obtained by *convolving* the input signal $x[n]$ and the impulse response $h[n]$ of the system.

Discrete-time convolution

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

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

2.7 Impulse Response and Convolution

Convolution operation for DTLTI systems (continued)

Steps involved in computing the convolution of two signals

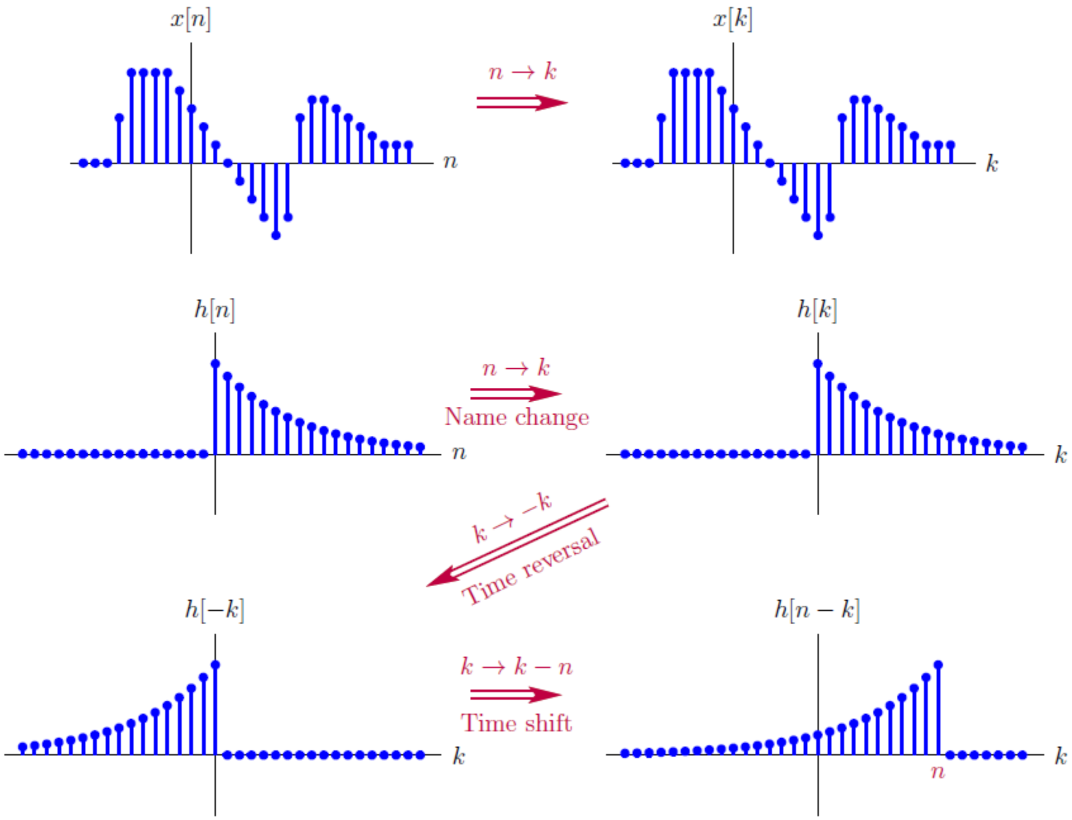
1. Sketch the signal $x[k]$ as a function of the independent variable k. This corresponds to a simple name change on the independent variable, and the graph of the signal $x[k]$ appears identical to the graph of the signal $x[n]$.
2. For one specific value of n, sketch the signal $h[n-k]$ as a function of the independent variable k. This task can be broken down into two steps as follows:
   - 2a. Sketch $h[-k]$ as a function of k. This step amounts to time-reversal of the signal $h[k]$.
   - 2b. In $h[-k]$ substitute $k \rightarrow k-n$. This step yields

     $$h[-k]\Big|_{k \rightarrow k-n} = h[n-k] \tag{1}$$

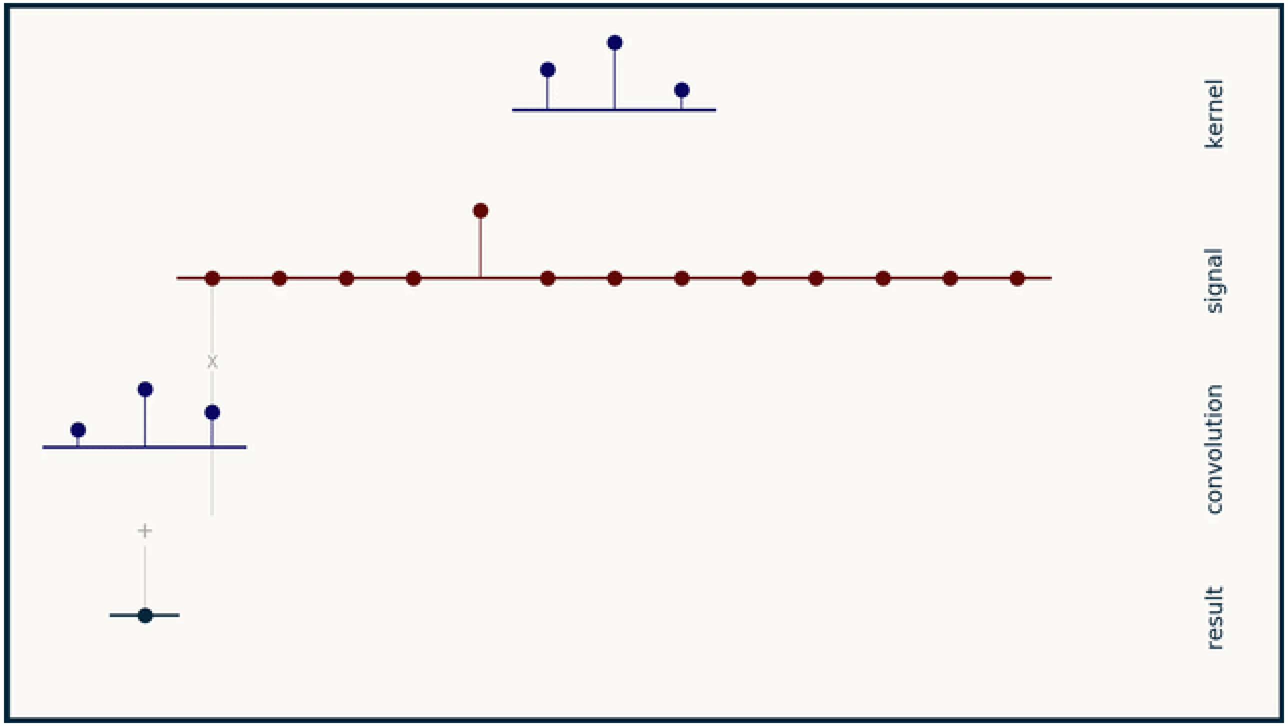
     and amounts to time-shifting $h[-k]$ by n samples.
3. Multiply the two signals in 1 and 2 to obtain the product $x[k]\,h[n-k]$.
4. Sum the sample amplitudes of the product $x[k]\,h[n-k]$ over the index k. The result is the amplitude of the output signal at the index n.
5. Repeat steps 1 through 4 for all values of n that are of interest.

2.7 Impulse Response and Convolution

Convolution operation for DTLTI systems (continued)

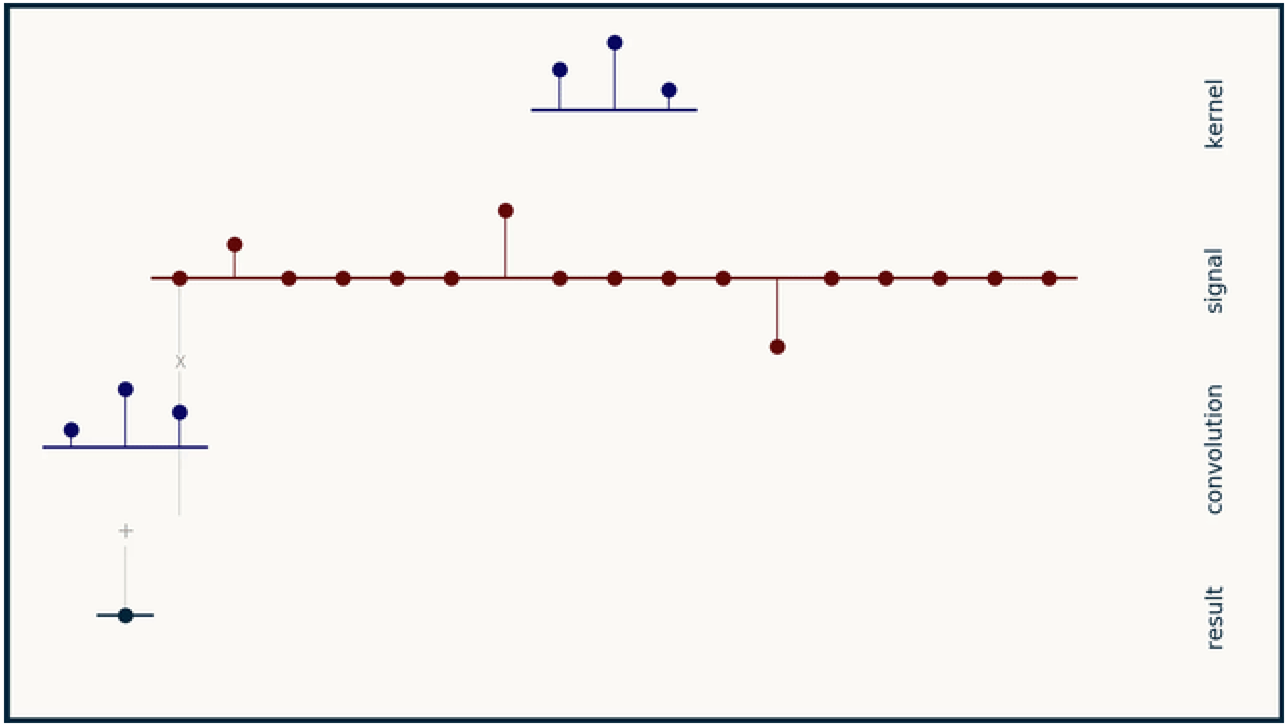


2.7 Impulse Response and Convolution



https://e2eml.school/convolution_one_d.html

2.7 Impulse Response and Convolution



https://e2eml.school/convolution_one_d.html

2.7 Impulse Response and Convolution

Example 3.19

A simple discrete-time convolution example

A discrete-time system is described through the impulse response

$$h[n] = \{ \underset{\substack{\uparrow \\ n=0}}{4}, 3, 2, 1 \}$$

Use the convolution operation to find the response of the system to the input signal

$$x[n] = \{ \underset{\substack{\uparrow \\ n=0}}{-3}, 7, 4 \}$$

<u>Solution:</u>

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

Express $x[k]$ and $h[n-k]$ as functions of k:

$$x[k] = \{ \underset{\substack{\uparrow \\ k=0}}{-3}, 7, 4 \}$$

$$h[-k] = \{ 1, 2, 3, \underset{\substack{\uparrow \\ k=0}}{4} \}$$

$$h[n-k] = \{ 1, 2, 3, \underset{\substack{\uparrow \\ k=n}}{4} \}$$

Since $x[k] = 0$ for $k < 0$ and $k > 2$

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

2.7 Impulse Response and Convolution

For $n = 0$:

$$y[0] = \sum_{k=0}^{0} x[k]\, h[0-k]$$
$$= x[0]\, h[0] = (-3)\ (4) = -12$$

For $n = 1$:

$$y[1] = \sum_{k=0}^{1} x[k]\, h[1-k]$$
$$= x[0]\, h[1] + x[1]\, h[0]$$
$$= (-3)\ (3) + (7)\ (4) = 19$$

For $n = 2$:

$$y[2] = \sum_{k=0}^{2} x[k]\, h[2-k]$$
$$= x[0]\, h[2] + x[1]\, h[1] + x[2]\, h[0]$$
$$= (-3)\ (2) + (7)\ (3) + (4)\ (4) = 31$$

For $n = 3$:

$$y[3] = \sum_{k=0}^{2} x[k]\, h[3-k]$$
$$= x[0]\, h[3] + x[1]\, h[2] + x[2]\, h[1]$$
$$= (-3)\ (1) + (7)\ (2) + (4)\ (3) = 23$$

For $n = 4$:

$$y[4] = \sum_{k=1}^{2} x[k]\, h[4-k]$$
$$= x[1]\, h[3] + x[2]\, h[2]$$
$$= (7)\ (1) + (4)\ (2) = 15$$

For $n = 5$:

$$y[5] = \sum_{k=2}^{2} x[k]\, h[5-k]$$
$$= x[2]\, h[3] = (4)\ (1) = 4$$

Thus the convolution result is

$$y[n] = \{\, \underset{\substack{\uparrow \\ n=0}}{-12}\,,\ 19,\ 31,\ 23,\ 15,\ 4\,\}$$

2.7 Impulse Response and Convolution

Example 3.21

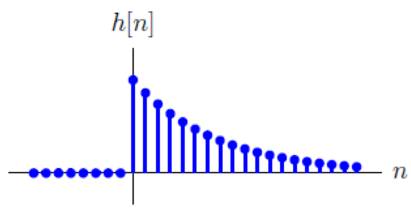
A more involved discrete-time convolution example

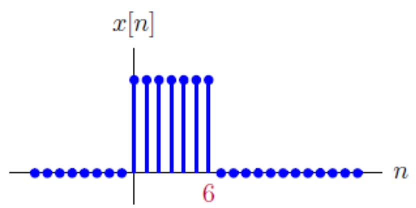
A discrete-time system is described through the impulse response

$$h[n] = (0.9)^n \; u[n]$$

Use the convolution operation to find the response of a system to the input signal

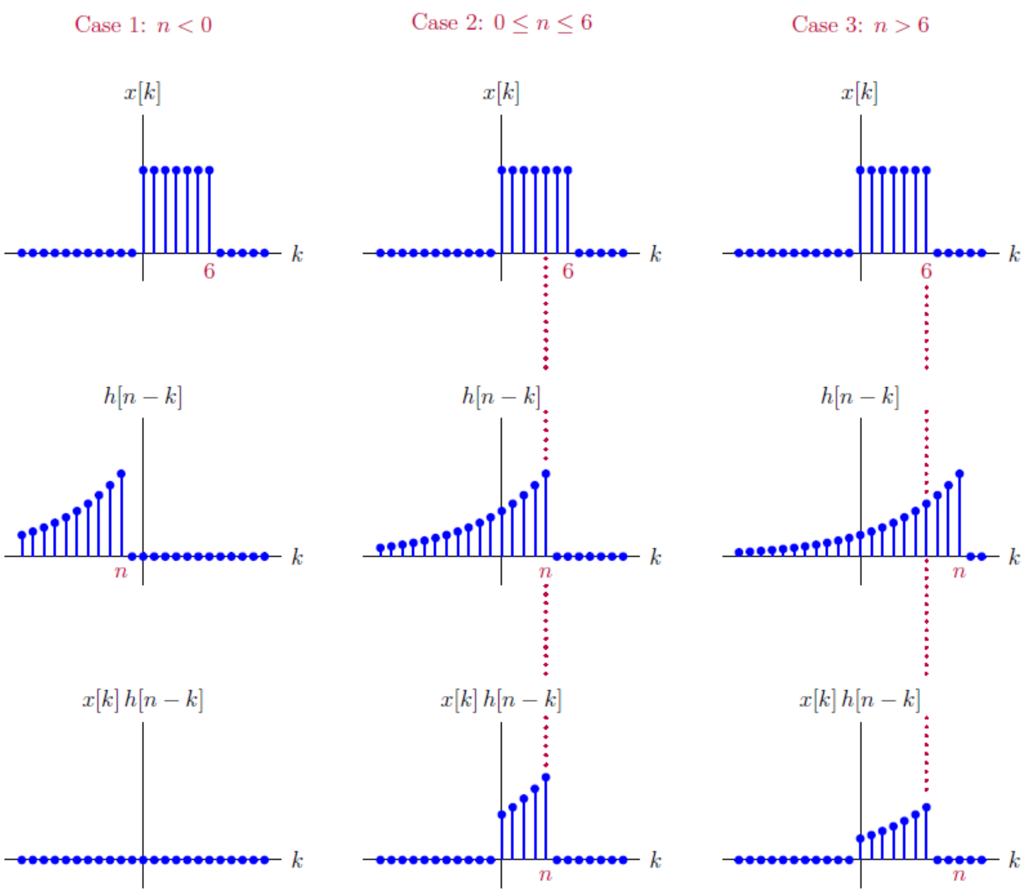
$$x[n] = u[n] - u[n-7]$$





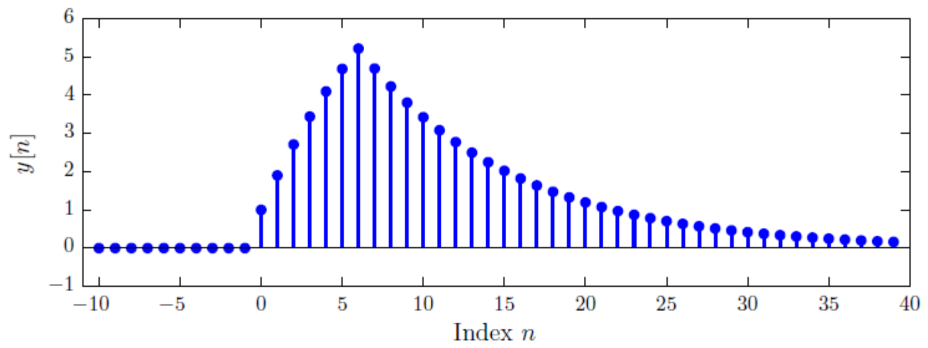
2.7 Impulse Response and Convolution

Example 3.21 (continued)



2.7 Impulse Response and Convolution

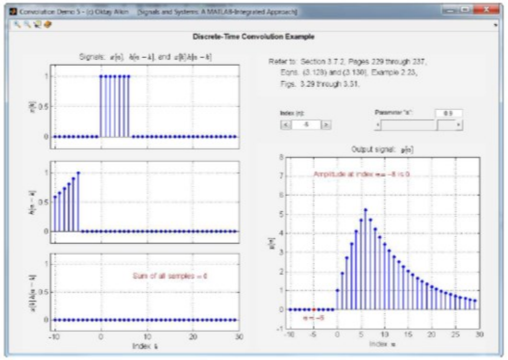
Example 3.21 (continued)



Interactive demo: conv_demo5.m

MATLAB Exercise 3.8

Change n and observe the waveforms and their overlaps.



2.8 Causality in Discrete-Time Systems

Causality in discrete-time systems

Causal system

A system is said to be causal if the current value of the output signal depends only on current and past values of the input signal, but not on its future values.

DTLTI system:

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

For $k < 0$, the term $x[n-k]$ refers to future values of the input signal.

Causality in DTLTI systems

For a DTLTI system to be causal, the impulse response of the system must be equal to zero for all negative index values.

$$h[n] = 0 \quad \text{for all} \quad n < 0$$

2.9 Stability in Discrete-Time Systems

Stability in discrete-time systems

Stable system

A system is said to be *stable* in the *bounded-input bounded-output (BIBO)* sense if any bounded input signal is guaranteed to produce a bounded output signal.

$$|x[n]| < B_x < \infty \quad \text{implies that} \quad |y[n]| < B_y < \infty$$

DTLTI system:

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

Stability in DTLTI systems

For a DTLTI system to be stable, its impulse response must be *absolute summable*.

$$\sum_{k=-\infty}^{\infty} |h[k]| < \infty$$

2.9 Stability in Discrete-Time Systems

Example 3.22

Stability of a length-2 moving-average filter

Comment on the stability of the length-2 moving-average filter described by the difference equation

$$y[n] = \frac{1}{2}\,x[n] + \frac{1}{2}\,x[n-1]$$

Solution:

Let us check directly to see if any arbitrary bounded input signal is guaranteed to produce a bounded output signal.

$$|y[n]| = \left|\frac{1}{2}\,x[n] + \frac{1}{2}\,x[n-1]\right|$$

$$|y[n]| \le \frac{1}{2}\,|x[n]| + \frac{1}{2}\,|x[n-1]|$$

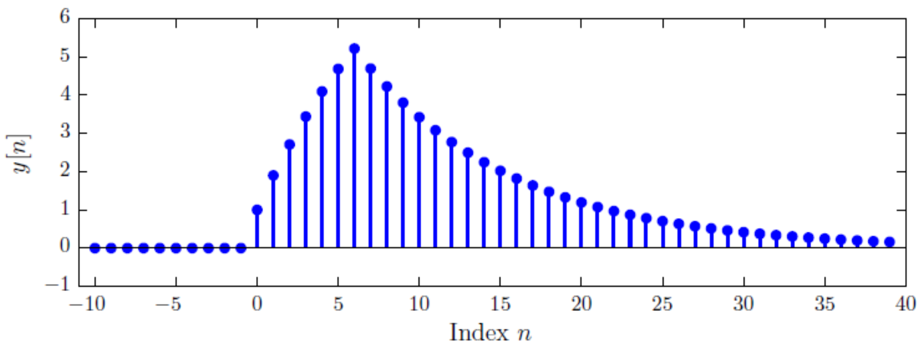
Since $|x[n]| < B_x$ for all n

$$|y[n]| \le \frac{1}{2}\,B_x + \frac{1}{2}\,B_x = B_x$$

Conclusion

2.7 Impulse Response and Convolution

Example 3.21 (continued)



Case 1: $n < 0$
There is no overlap between $x[k]$ and $h[n-k]$.

MATLAB Exercise 3.8

$$y[n] = 0, \quad \text{for} \quad n < 0$$

Case 2: $0 \leq n \leq 6$
For this case, the two functions overlap for $0 \geq k \leq n$.

$$y[n] = \sum_{k=0}^{n} (1)\,(0.9)^{n-k} = -9\left[(0.9)^n - \frac{1}{0.9}\right], \quad \text{for} \quad 0 \leq n \leq 6$$

Case 3: $n > 6$ For this case the two functions overlap for $0 \leq k \leq 6$.

$$y[n] = \sum_{k=0}^{6} (1)\,(0.9)^{n-k} = 9.8168\,(0.9)^n, \quad \text{for} \quad n > 6$$