# Image Processing (CSE281)

## Fall 2025/2026


## Dr. Essam Abdellatef

Contact Number: 012 8 192 55 90

Email: eabdellatef@Aiu.edu.eg

# Image Re-size

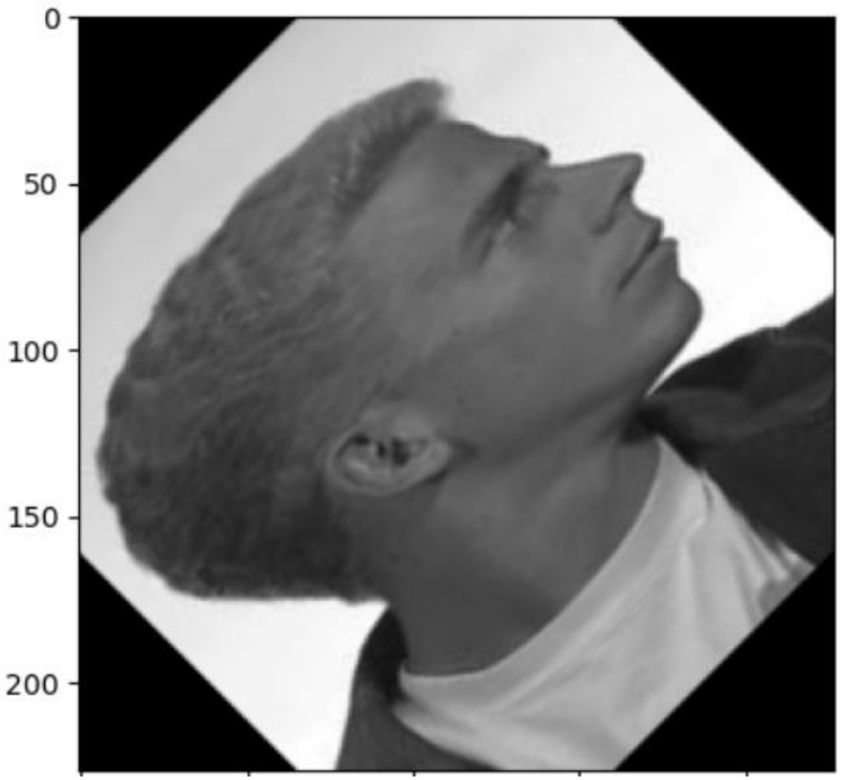
```
from PIL import Image #Python Imaging Library
image_path = "F:/13.jpg"
my_image = Image.open(image_path)
my_image.show()
new_size = (100, 100)
resized_image = my_image.resize(new_size)
resized_image.show()
```

# RGB to Gray

```python
from PIL import Image #Python Imaging Library
image_path = "F:/13.jpg"
my_image = Image.open(image_path)
Gray_image = my_image.convert('L')
Gray_image.show()
```

# Image Rotation

```
from PIL import Image #Python Imaging Library
image_path = "F:/13.jpg"
my_image = Image.open(image_path)
Rotated_image = my_image.rotate(45)
Rotated_image.show()
```



# Drawing on images

```python
from PIL import Image, ImageDraw
my_image = Image.open("F:/13.jpg")
draw = ImageDraw.Draw(my_image)
draw.line((10, 20, 400, 20), fill = (0, 255, 0), width = 4)
my_image.show()
```

draw.line((10, 20, 400, 20), fill = (0, 255, 0), width = 4)
Start point = (10, 20)
End point = (400, 20)
Fill = (0, 255, 0) → Green
Width = 4 → The Thickness of the Line

# Drawing on images

```python
from PIL import Image, ImageDraw
my_image = Image.open("F:/13.jpg")
draw = ImageDraw.Draw(my_image)
draw.rectangle((10, 20, 40, 45), fill = (0, 255, 0), width = 4)
my_image.show()
```

**draw.rectangle((10, 20, 40, 45), fill = (0, 255, 0), width = 4)**
**Start point = (10, 20)**
**End point = (40, 45)**
**Fill = (0, 255, 0) → Green**
**Width = 4 → The Thickness**

# Drawing on images

```
from PIL import Image, ImageDraw
my_image = Image.open("F:/13.jpg")
draw = ImageDraw.Draw(my_image)
draw.circle((40, 80), 30, fill = (0, 255, 0), width = 4)
my_image.show()
```

**draw.circle((40, 80), 30, fill = (0, 255, 0), width = 4)**
**Center = (40, 80)**
**30 → Radius of Circle**
**Fill = (0, 255, 0) → Green**
**Width = 4 → The Thickness**

# Image Cropping

```python
from PIL import Image
my_image = Image.open("F:/13.jpg")
crop_box = (100, 50, 400, 300)
cropped_image = my_image.crop(crop_box)
cropped_image.show()
```

**Crop_box = (left, upper, right, lower)**
**Left = 100 → Start cropping 100 pixels from the left edge of the image.**
**Upper = 50 → Start cropping 50 pixels from the top edge.**
**Right = 400 → Stop cropping at 400 pixels from the left edge.**
**Lower = 300 → Stop cropping at 300 pixels from the top edge.**

# Image Blurring

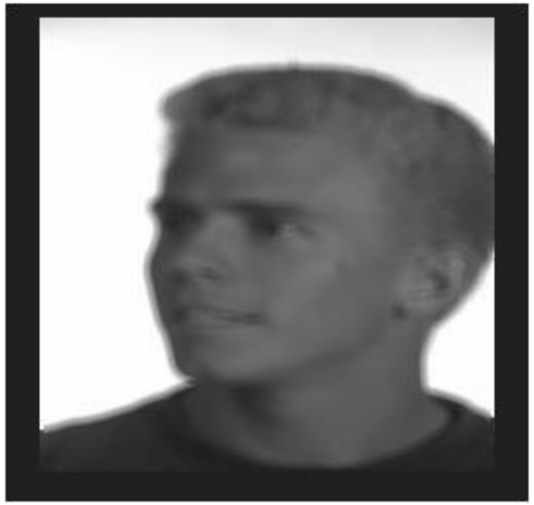
A technique that smooths an image by reducing sharp transitions in intensity.

***Purpose:*** Used to reduce noise, remove fine details, or prepare images for further processing such as edge detection.

***Operation:*** Convolution with a blur kernel (such as Gaussian).

# Image Blurring

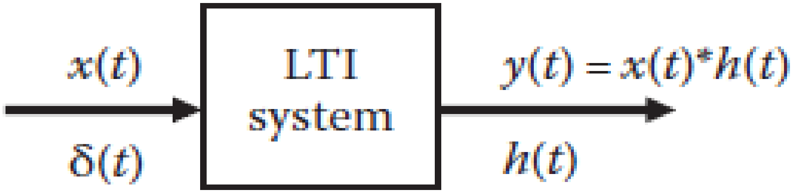
```
from PIL import Image, ImageFilter
my_image = Image.open("F:/13.jpg")
blurred_image = my_image.filter(ImageFilter.BLUR)
blurred_image.show()
```



# Convolution

The convolution of two signals $x(t)$ and $h(t)$ is usually written in terms of the operator * as

$$y(t) = x(t) * h(t) = \int_{-\infty}^{\infty} x(\tau)h(t-\tau)d\tau$$



# Convolution

## Properties of the Convolution Integral

1. $x(t) * h(t) = h(t) * x(t)$ (Commutative)

2. $f(t) * [x(t) + y(t)] = f(t) * x(t) + f(t) * y(t)$ (Distributive)

3. $f(t) * [(x(t) * y(t)] = f[(t) * x(t)] * y(t)$ (Associative)

4. $f(t) * \delta(t) = \int_{-\infty}^{\infty} f(\tau)\delta(t-\tau)d\tau = f(t)$

5. $f(t) * \delta(t-t_o) = f(t-t_o)$

6. $f(t) * \delta'(t) = \int_{-\infty}^{\infty} f(\tau)\delta'(t-\tau)d\tau = f'(t)$

7. $f(t) * u(t) = \int_{-\infty}^{\infty} f(\tau)u(t-\tau)d\tau = \int_{-\infty}^{t} f(\tau)d\tau$

8. $u(t) * \delta'(t) = \int_{-\infty}^{\infty} u(\tau)\delta'(t-\tau)d\tau = u'(t) = \delta(t)$

9. If $x_1(t) * x_2(t) = y(t)$, then
   $x_1(t + t_1) * x_2(t + t_2) = y(t + t_1 + t_2)$

# Convolution

**Example**

The input $x(t)$ and the impulse response $h(t)$ of an LTI system are given by $x(t) = u(t)$ and $h(t) = e^{-3}tu(t)$. Find the output response.

**Solution**

From Equation 2.6,

$$y(t) = \int_{-\infty}^{\infty} x(\tau)h(t-\tau)d\tau = \int_{-\infty}^{\infty} u(\tau)e^{-3(t-\tau)}u(t-\tau)d\tau$$

In this integral, $t$ is a constant so that we can pull out the factor $e^{-3t}$.

By definition,

$$u(\tau) = \begin{cases} 1, & \tau > 0 \\ 0, & \tau < 0 \end{cases}$$

$$u(t-\tau) = \begin{cases} 1, & t-\tau > 0 \\ 0, & t-\tau < 0 \end{cases} = \begin{cases} 1, & \tau < t \\ 0, & \tau > t \end{cases}$$

Thus,

$$u(\tau)u(t-\tau) = \begin{cases} 1, & 0 < t < \tau \\ 0, & \text{otherwise} \end{cases}$$

# Convolution

The limits on the integral now becomes $0 < \tau < t$. Therefore, Equation 2.2.1 becomes

$$y(t) = e^{-3t} \int_0^t e^{3\tau} d\tau = \frac{e^{-3t}}{3} e^{3\tau} \Big|_0^t$$

$$= \frac{1}{3}(1 - e^{-3t}), \quad t > 0$$

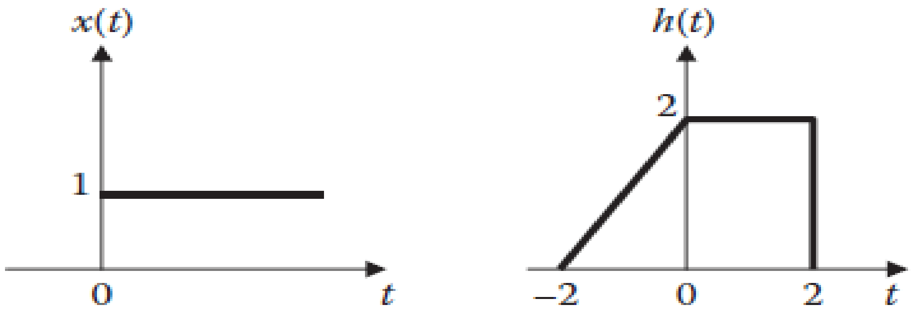
or

$$y(t) = \frac{1}{3}(1 - e^{-3t})u(t)$$

# Graphical Convolution

**Example**

Obtain the convolution of the two signals in Figure 2.4.




**Solution**

Let $y(t) = x(t)*h(t)$, where $x(t)$ is a unit step and

$$h(t) = \begin{cases} 2+t, & -2 < t < 0 \\ 1, & 0 < t < 2 \end{cases}$$

# Graphical Convolution

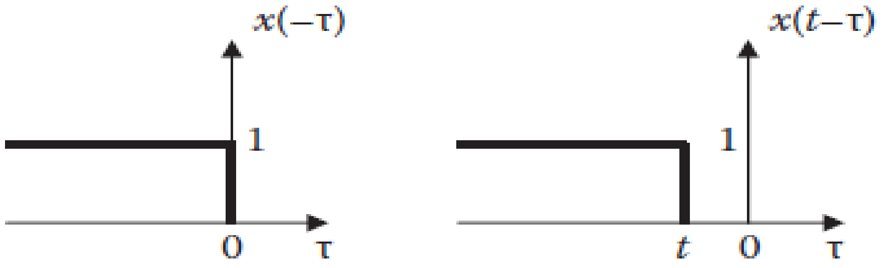
In this case, it is easy to fold $x(t)$, the unit step function. Let

$$y(t) = x(t) * h(t) = \int x(t-\tau)h(\tau)d\tau$$

We follow the four steps to get $y(t)$. First, we fold $x(t)$ as shown in Figure 2.5a and shift it by $t$ as shown in Figure 2.5b.

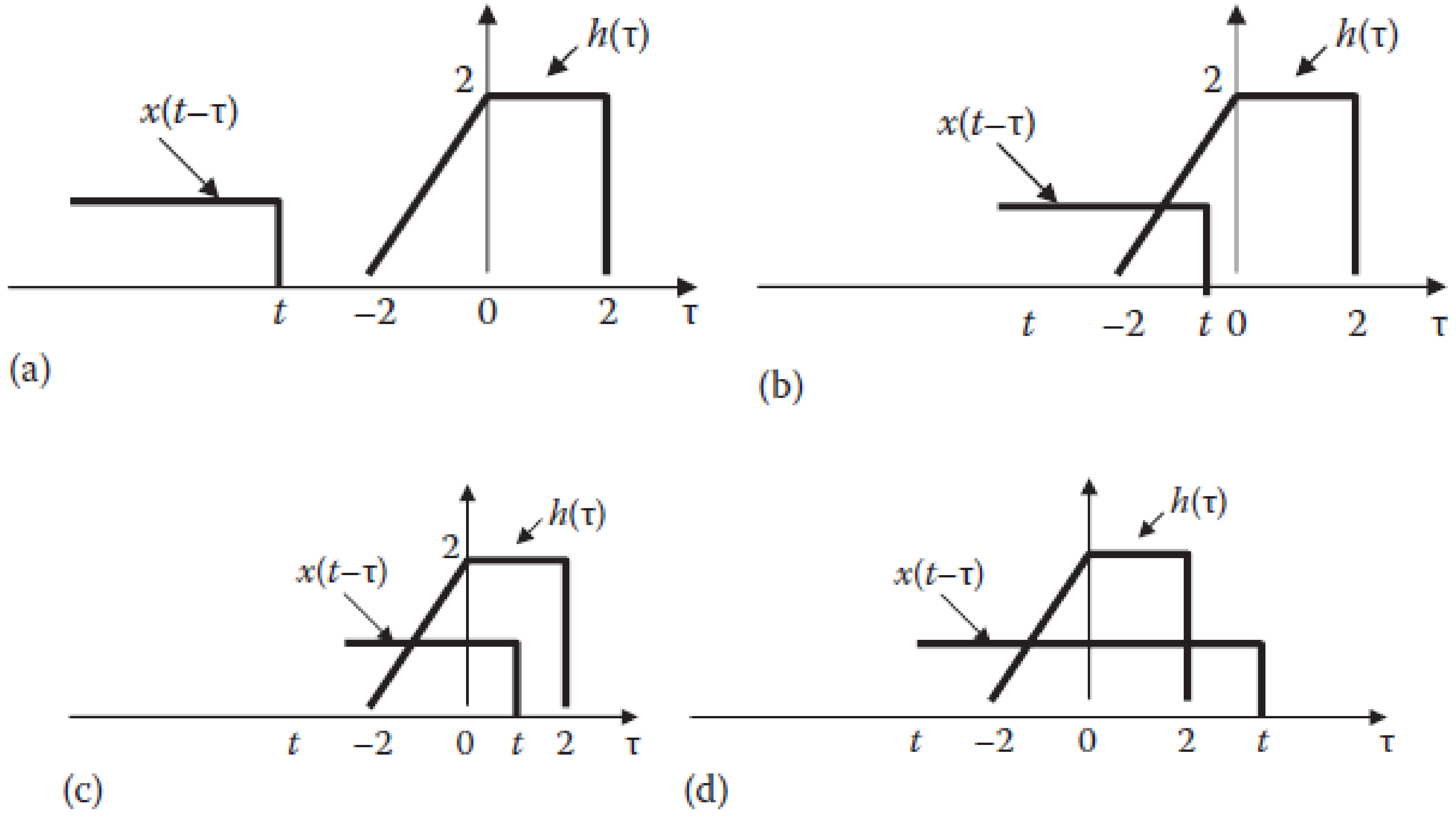
For $t < -2$, there is no overlap of the two signals, as shown in Figure 2.6a. Hence,

$$y(t) = x(t) * h(t) = 0, \qquad t < -2$$



(a) (b)

# Graphical Convolution



# Graphical Convolution

For $-2 < t < 0$, the two signals overlap between $-2$ and $t$, as shown in Figure 2.6b.

$$y(t) = \int_{-2}^{t} h(\tau)x(t-\tau)d\tau = (2+\tau)(1)d\tau\Big|_{-2}^{t} = 2\tau + \frac{\tau^2}{2}\Big|_{-2}^{t}$$

$$= 0.5t^2 + 2t + 2, \quad -2 < t < 0$$

For $0 < t < 2$, the two signals overlap between $-2$ and $t$, as shown in Figure 2.6c.

$$y(t) = \int_{-2}^{0} (2+\tau)(1)d\tau + \int_{0}^{t} (2)(1)d\tau$$

$$= \left(2\tau + \frac{\tau^2}{2}\right)\Bigg|_{-2}^{0} + 2\tau\Big|_{0}^{t} = 4 - 2 + 2t = 2(t+1), \quad 0 < t < 2$$

# Graphical Convolution

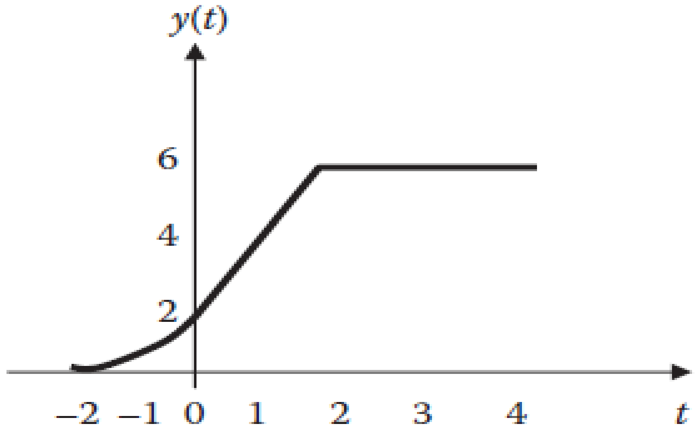
For $t > 2$, the two signals overlap between –2 and 2, as shown in Figure 2.6d.

$$y(t) = \int_{-2}^{0} (2+\tau)(1)d\tau + \int_{0}^{2} (2)(1)d\tau$$

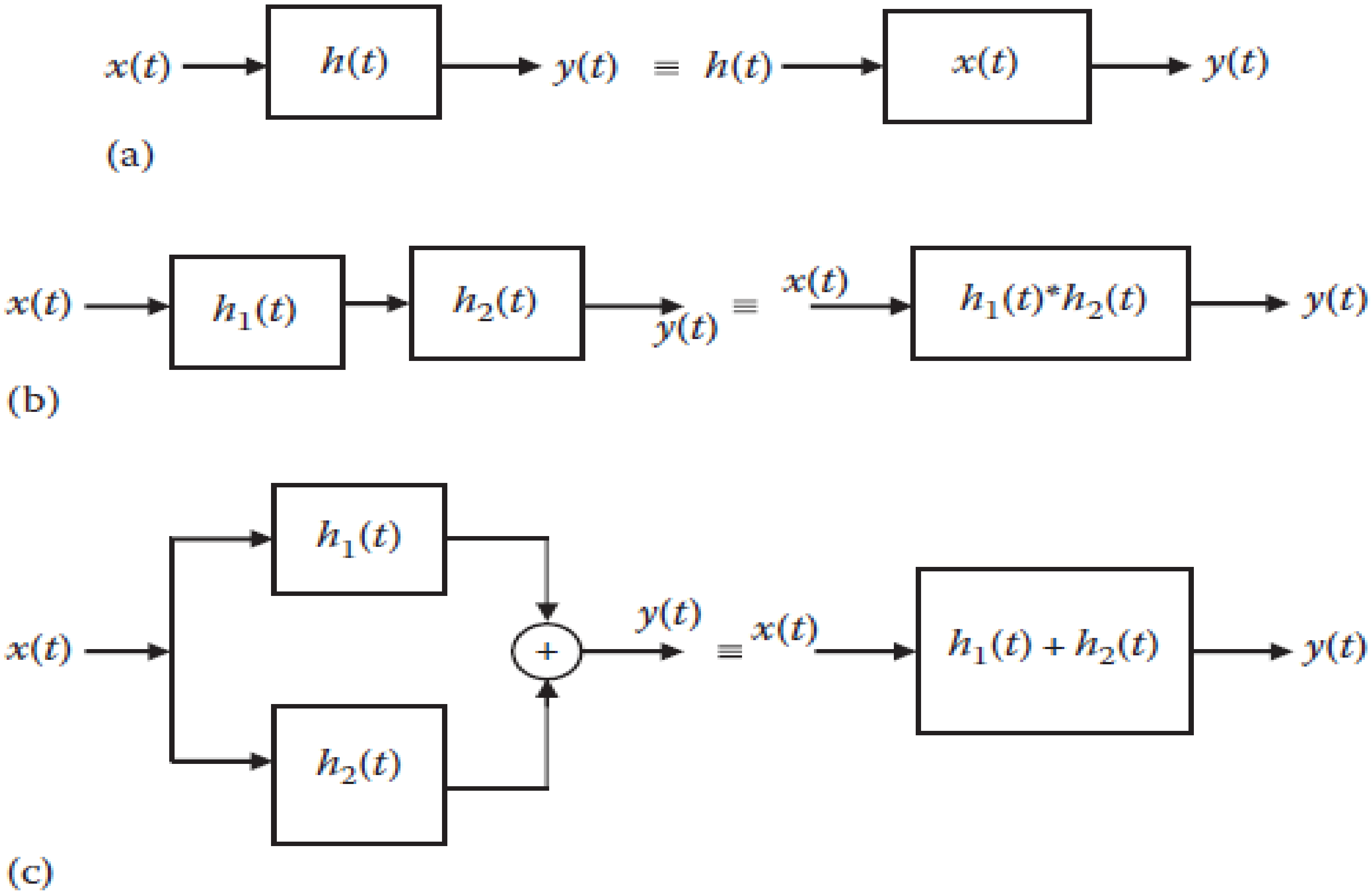
$$= \left(2\tau + \frac{\tau^2}{2}\right)\Bigg|_{-2}^{0} + 2\tau\Big|_{0}^{2} = 4 - 2 + 4 = 6, \quad t > 2$$

# Graphical Convolution

$$y(t) = \begin{cases} 0.5t^2 + 2t + 2, & -2 \le t \le 0 \\ 2(t+1), & 0 \le t \le 2 \\ 6, & t > 2 \\ 0, & \text{otherwise} \end{cases}$$



# Convolution



# Convolution

**Example**

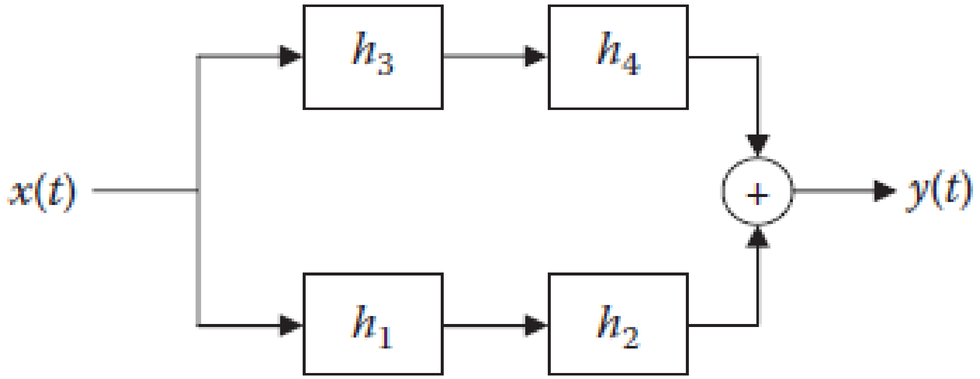
Find the impulse response of the system shown in Figure 2.13. Let

$$h_1(t) = 3\delta(t)$$

$$h_2(t) = 2e^{-t}u(t)$$

$$h_3(t) = 4e^{-2t}u(t)$$

$$h_4(t) = e^{-3t}u(t)$$



# Convolution

Using the fact that $x(t)*\delta(t) = x(t)$, we obtain the first convolution on the right-hand side as

$$h_1(t) * h_2(t) = 3\delta(t) * \left\{2e^{-t}u(t)\right\} = 6e^{-t}u(t)$$

For the second convolution,

$$h_3(t) * h_4(t) = \int_{-\infty}^{\infty} 4e^{-2\tau}u(\tau)e^{-3(t-\tau)}u(t-\tau)d\tau$$

but

$$u(\tau)u(t-\tau) = \begin{cases} 1, & 0 < t < \tau \\ 0, & \text{otherwise} \end{cases}$$

# Convolution

Therefore,

$$h_3(t) * h_4(t) = \int_0^t 4e^{-2\tau}e^{-3(t-\tau)}d\tau = 4e^{-3t}\int_0^t e^{(3-2)\tau}d\tau$$

$$= 4e^{-3t}e^{\tau}\Big|_0^t = 4(e^{-2t} - e^{-3t}), \quad t > 0$$

$$= 4(e^{-2t} - e^{-3t})u(t)$$

Substituting Equations 2.5.2 and 2.5.3 into Equation 2.5.1 gives

$$h(t) = 6e^{-t}u(t) + 4(e^{-2t} - e^{-3t})u(t)$$

# Discrete Convolution

When an input is applied to a discrete-time system, the response or output sequence can be determined in a way similar to using the impulse response and the convolution integral for continuous-time systems.

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

The unit impulse sequence is also redefined as

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

The signal only has value at $n = 0$. The displaced delta function is

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

# Discrete Convolution

The convolution of the discrete input signal $x[n]$ and the impulse response $h[n]$ is

$$y[n] = x[n] * h[n]$$

and is defined as

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

Thus,

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

# Discrete Convolution

## Properties of the Convolution Sum

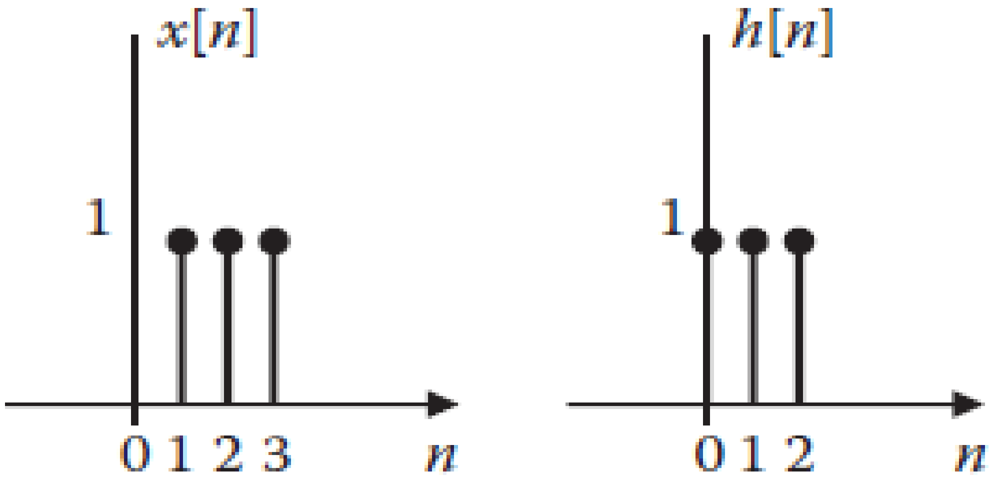
1. $x[n] * h[n] = h[n] * x[n]$ (Commutative)
2. $f[n] * \left[x[n] + y[n]\right] = f[n] * x[n] + f[n] * y[n]$ (Distributive)
3. $f[n] * \left[x[n] * y[n]\right] = \left[f[n] * x[n]\right] * y[n]$ (Associative)
4. $x[n-m] * h[n-k] = y[n-m-k]$ (Shifting)
5. $x[n] * \delta[n] = x[n]$

# Discrete Convolution

**Example**

Consider $x[n]$ and $h[n]$ as shown in Figure 2.17. (The signals are all zero outside the ranges indicated.) Find $y[n] = x[n]*h[n]$:

(a) Analytically and (b) graphically.



# Discrete Convolution

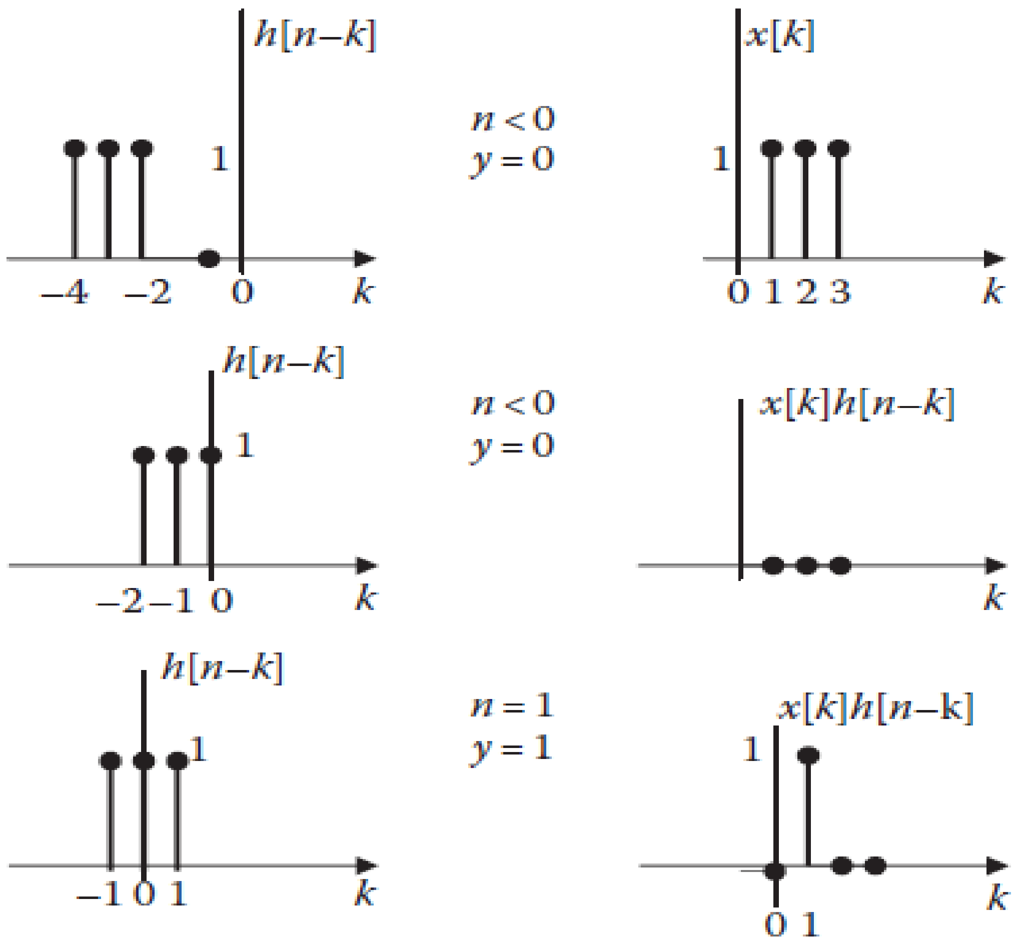
(a) Using Equation 2.19, we can write

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

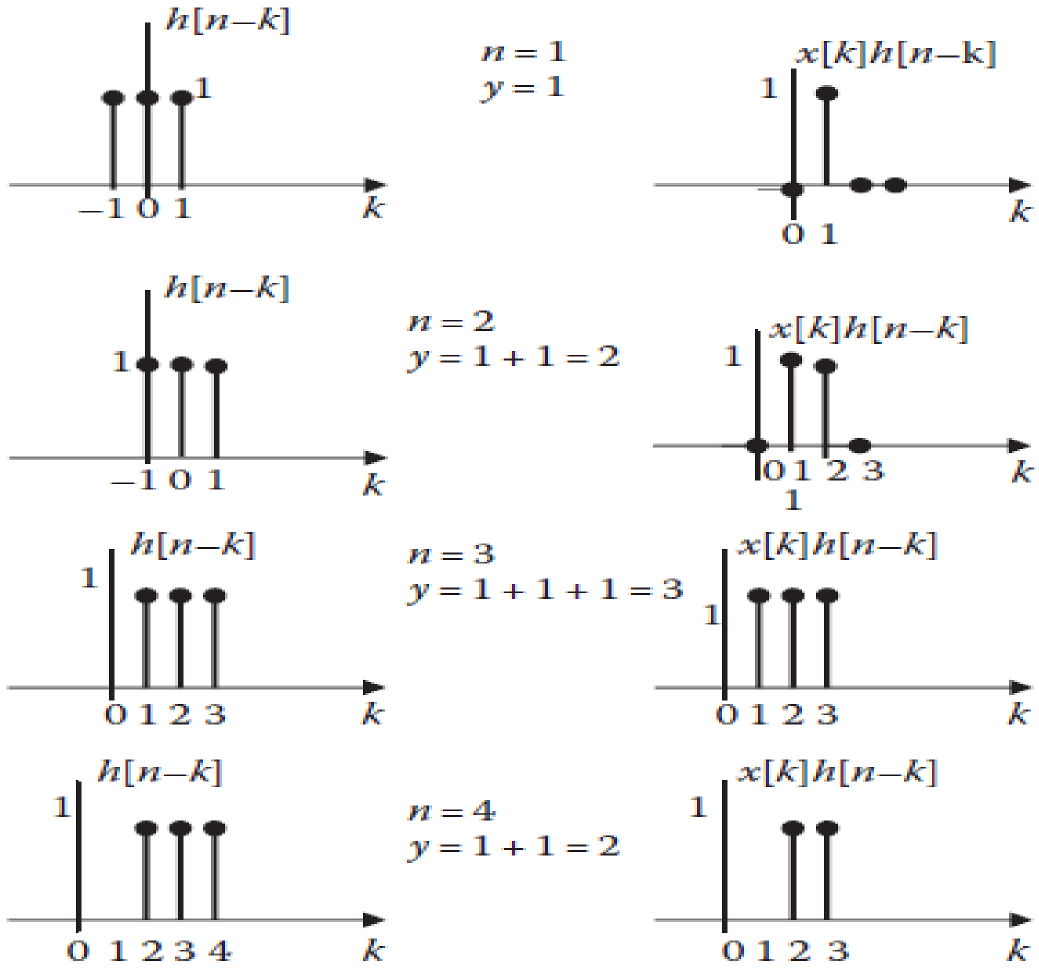
$$h[n] = \delta[n] + \delta[n-1] + \delta[n-2]$$

$$\begin{aligned}
y[n] &= x[n] * h[n] = x[n] * \{\delta[n] + \delta[n-1] + \delta[n-1]\} \\
&= x[n] + x[n-1] + x[n-2] \\
&= \delta[n-1] + \delta[n-2] + \delta[n-3] \\
&\quad +\delta[n-2] + \delta[n-3] + \delta[n-4] \\
&\quad +\delta[n-3] + \delta[n-4] + \delta[n-5] \\
&= \delta[n-1] + 2\delta[n-2] + 3\delta[n-3] + 2\delta[n-4] + \delta[n-5]
\end{aligned}$$

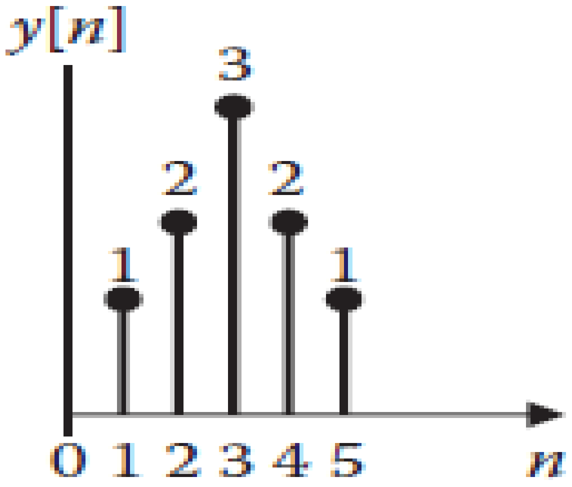
# Discrete Convolution



# Discrete Convolution



# Discrete Convolution



# Discrete Convolution

**Example**

Consider a discrete-time LTI system for which the input and impulse functions are given by

$$x[n] = (0.7)^n u[n], \qquad h[n] = (0.2)^n u[n]$$

Find the response $y[n]$.

**Solution**

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

But $x[k] = (0.7)^k u[k]$ and $h[n-k] = (0.2)^{n-k} u[n-k]$
Because $x[n]$ and $h[n]$ are causal, we can use Equation 2.27.

$$y[n] = x[n] * h[n] = \sum_{k=0}^{n} (0.7)^k (0.2)^{n-k}, \quad n = 0, 1, 2, \ldots$$

$$= (0.2)^n \sum_{k=0}^{n} \left(\frac{0.7}{0.2}\right)^k$$