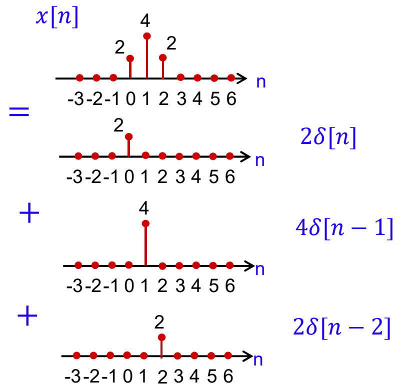
Superposition

U1

- If the system is linear, superposition can be applied: Break input into additive parts and sum the responses to the parts

- All signals can be expressed as a linear combination of shifted impulses

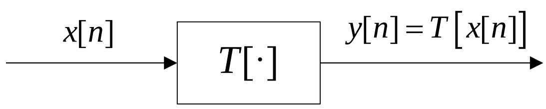
$$x[n] = \sum_{k=-\infty}^{\infty} x[k]\, \delta[n-k]$$



Convolution equation

U1

- Response of a discrete-time LTI system to any input



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

linearity

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

convolution

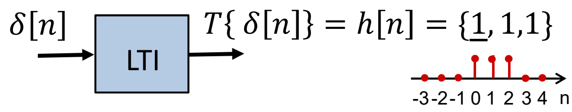
time-invariance

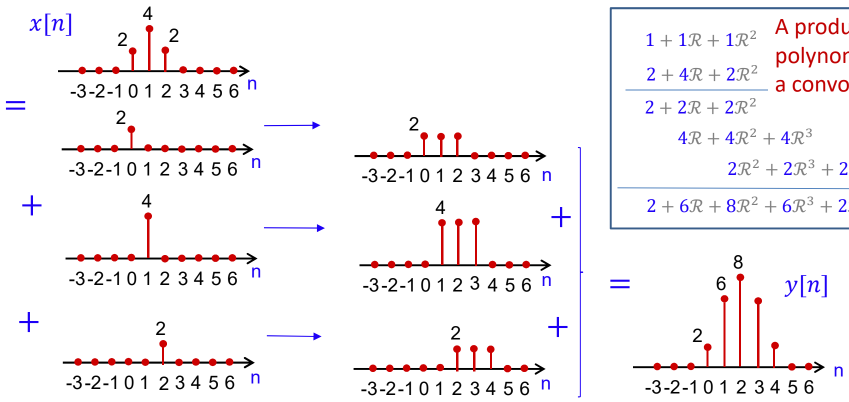
- For analog LTI systems: $y(t) = \int_{-\infty}^{\infty} x(\tau)h(t-\tau)d\tau \triangleq x(t) * h(t)$

Computation of convolution (1)

U1

If the system is linear and time invariant (LTI), the computation of the output is easy



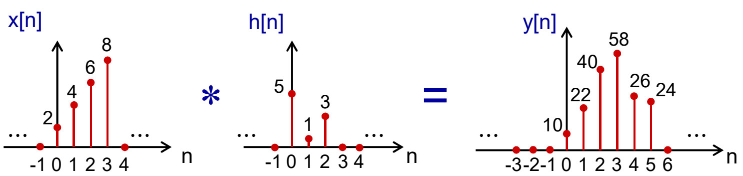


$$
\begin{array}{l}
1 + 1\mathcal{R} + 1\mathcal{R}^2 \\
2 + 4\mathcal{R} + 2\mathcal{R}^2 \\
\hline
2 + 2\mathcal{R} + 2\mathcal{R}^2 \\
\quad 4\mathcal{R} + 4\mathcal{R}^2 + 4\mathcal{R}^3 \\
\qquad 2\mathcal{R}^2 + 2\mathcal{R}^3 + 2\mathcal{R}^4 \\
\hline
2 + 6\mathcal{R} + 8\mathcal{R}^2 + 6\mathcal{R}^3 + 2\mathcal{R}^4
\end{array}
$$

A product of polynomials is a convolution

Computation of convolution (2)

U1



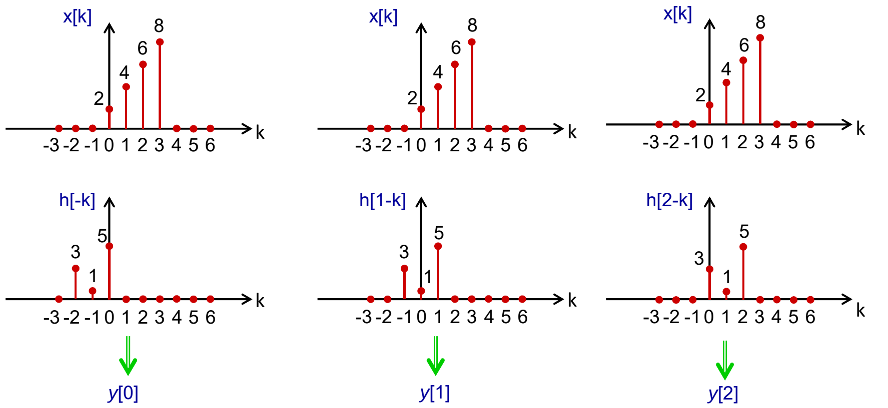
$$y[n] = x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k]h[n-k]$$

We already have seen a way to interpret/compute convolution in the discrete-time case, now we will see another

Reflection, shift, multiply and add

U1

$$y[0]=\sum_{k=-\infty}^{\infty}x[k]h[0-k] \qquad y[1]=\sum_{k=-\infty}^{\infty}x[k]h[1-k] \qquad y[2]=\sum_{k=-\infty}^{\infty}x[k]h[2-k]$$

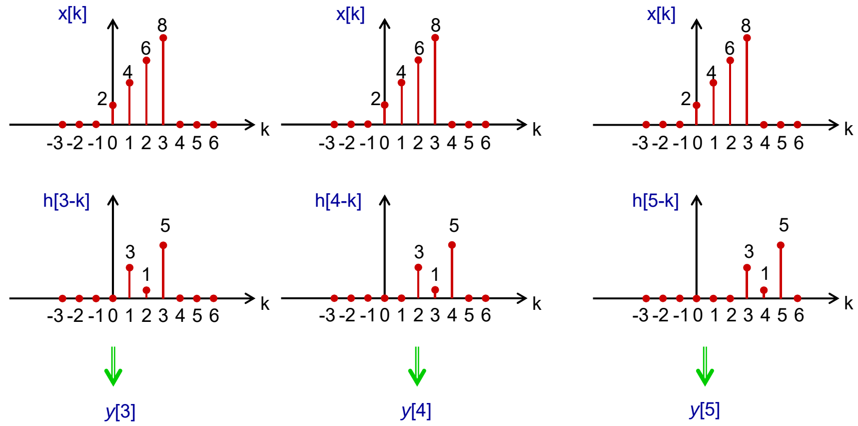


y[0]=2·5+4·0+6·0+8·0=10 y[1]=2·1+4·5+6·0+8·0=22 y[2]=2·3+4·1+6·5+8·0=40

Reflection, shift, multiply and add

U1

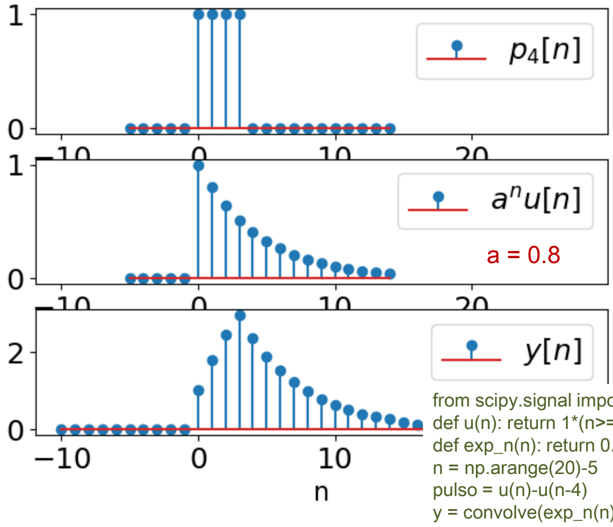
$$y[3]=\sum_{k=-\infty}^{\infty}x[k]h[3-k] \qquad y[4]=\sum_{k=-\infty}^{\infty}x[k]h[4-k] \qquad y[5]=\sum_{k=-\infty}^{\infty}x[k]h[5-k]$$



y[3]=2·0+4·3+6·1+8·5=58 y[4]=2·0+4·0+6·3+8·1=26 y[5]=2·0+4·0+6·0+8·3=24

Example

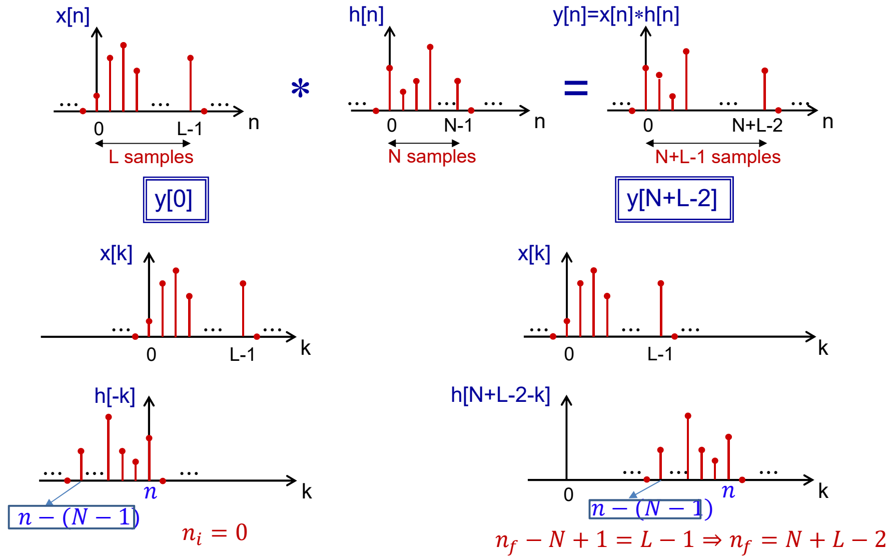
U1




```
from scipy.signal import convolve
def u(n): return 1*(n>=0)
def exp_n(n): return 0.8**n*(n>=0)
n = np.arange(20)-5
pulso = u(n)-u(n-4)
y = convolve(exp_n(n),pulso)
```


Convolution length: L+N-1

U1



Convolution properties

U1

- Distributive property: a[n] ∗ (b[n] + c[n]) = (a[n] ∗ b[n]) + (a[n] ∗ c[n])
- Associative property: a[n] ∗ (b[n] ∗ c[n]) = (a[n] ∗ b[n]) ∗ c[n]
- Conmutative property: a[n] ∗ b[n] = b[n] ∗ a[n]
- Identity (neutral) element: $x[n] * \delta[n] = \delta[n] * x[n] = x[n]$

  Delay : $x[n-n_0] = x[n] * \delta[n-n_0]$

✓ The same properties in the analog case

About notation

U1

- A conventional notation for convolution is $x[n] * h[n]$

  It looks like an operation between samples, but it is not

If $y[n] = x[n] * h[n]$
$x[1] * h[1] \neq y[1]$

Delay:
If $y[n] = x[n] * h[n]$
$x[n-n_0] * h[n] = y[n-n_0]$
$x[n-n_0] * h[n-n_0] = y[n-2n_0]$

Convolution operates on signals not samples
Do not be fooled by the notation!

- Another notation is: $(x * h)[n]$

Summary of convolution for time-discrete signals

U1

Different ways of computing convolution, $x[n] = \{\underline{1}, 2,3,4\};\ h[n] = \{\underline{1}, 0, -1\}$

- Finite-differences equation
  $y[n] = x[n] * (h[0]\delta[n] + \cdots)$
  $= h[0]x[n]+h[1]x[n-1]+\ldots$

$y[n] = x[n] - x[n-2]$
$y[0] = x[0] - x[-2] = 1$
$y[1] = x[1] - x[-1] = 2$
….

Useful for step by step analysis

- Product of polynomials
  $y[n] = \sum_m x[m]h[n-m]$
  $= \sum_m h[m]x[n-m]$

$$
\begin{array}{l}
1 + 2\mathcal{R} + 3\mathcal{R}^2 + 4\mathcal{R}^3 \\
\underline{1 + 0\mathcal{R} - 1\mathcal{R}^2} \\
1 + 2\mathcal{R} + 3\mathcal{R}^2 + 4\mathcal{R}^3 \\
\quad 0\mathcal{R} + 0\mathcal{R}^2 + 0\mathcal{R}^3 + 0\mathcal{R}^4 \\
\quad\quad \underline{-1\mathcal{R}^2 \ - 2\mathcal{R}^3 - 3\mathcal{R}^4 - 4\mathcal{R}^5} \\
1 + 2\mathcal{R} + 2\mathcal{R}^2 + 2\mathcal{R}^3 \ - 3\mathcal{R}^4 - 4\mathcal{R}^5
\end{array}
$$

Convolution is equivalent to a multiplication of polynomials (commutative)

- Reflection, shift, multiply and add:
  - Gives intuition
  - Useful for signals expressed in analytical form and also for analog signals

$-1,0,\underline{1} \rightarrow$
$x[n] = \underline{1}, 2,3,4$
$y[n] = \underline{1}, 2,2,2,-3,-4$

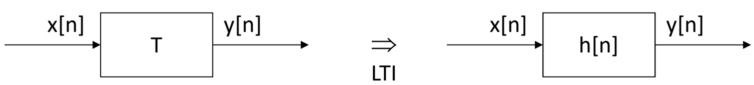
Output at time n computed as a linear combination of input at time n and surrounding samples

LTI systems and impulse response

U1

Then, if the system is linear and time invariant (LTI)

- it is uniquely defined by the impulse response, $h[n] = \mathrm{T}\{\delta[n]\}$, (known, measured, or identified from the data)



- and the output can be computed through convolution

Properties of LTI systems

U1

- Causal system
  - In general: $y[n] = f\{x[n-k], \quad k \geq 0\}$
  - A LTI system is causal iff (if, and only if) $h[n] = 0 \quad \forall n < 0$

Future samples

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

$$= \ldots + h[2]x[n-2] + h[1]x[n-1] + h[0]x[n] + \boxed{h[-1]x[n+1] + h[-2]x[n+2] + \ldots}$$

- Stable system
  - In general: if $x[n]$ is bounded $\Rightarrow y[n]$ is bounded
  - A LTI system is stable iff $\sum_{n=-\infty}^{+\infty} |h[n]| < \infty$

sufficient

$$\text{for } |x[n]| < B, \forall n, \qquad |y[n]| \leq B \sum_{k=-\infty}^{+\infty} |h[k]|$$

necessary

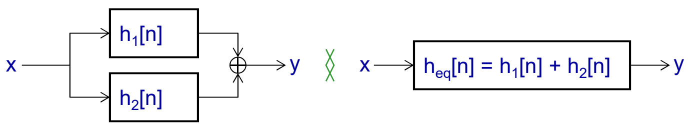
$$\text{for } x[n] = \begin{cases} h^*[-n] / |h[-n]| & \text{if } h[-n] \neq 0 \\ 0 & \text{otherwise} \end{cases} \qquad y[0] = \sum_{k=-\infty}^{+\infty} |h[k]|$$

Connection of LTI systems

U1

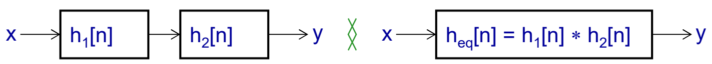
- Distributive property: a[n] ∗ (b[n] + c[n]) = (a[n] ∗ b[n]) + (a[n] ∗ c[n])

Conexión en paralelo

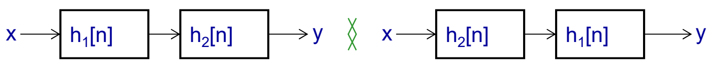


- Associative property: a[n] ∗ (b[n] ∗ c[n]) = (a[n] ∗ b[n]) ∗ c[n]

Conexión en serie o cascada



- Commutative property: a[n] ∗ b[n] = b[n] ∗ a[n]



LTI systems are always commutative !!!