

# Processamento Digital de Sinais - PDS

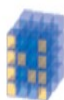
---

## LAB002 - Python em PDS:

Vamos usar a Linguagem Python para visualizar e compreender os conceitos de PDS

Bibliotecas de Python::

- Numpy e Scipy - <https://scipy.org/>
- Matplotlib - <https://matplotlib.org/>



NumPy  
Base N-dimensional  
array package



SciPy library  
Fundamental library  
for scientific  
computing



Matplotlib  
Comprehensive 2-D  
plotting

## Exercícios:

1. Suponha que um sistema  $H$  do tipo LTI com resposta impulso  $h[n]$  é aplicado um sinal  $x[n]$ .

$$h[n] = \begin{cases} 1, & 0 \leq n \leq 2 \\ 0, & \text{caso contrario} \end{cases} \quad x[n] = \begin{cases} 0.5, & n = 0 \\ 2, & n = 1 \\ 0, & \text{caso contrario} \end{cases}$$

Apresente um programa que apresente a resposta  $y[n]$ .

(Utilize a função *convolve* do *numpy*)

2. Suponha que em um sistema  $H$  do tipo LTI com a resposta impulso  $h[n]$  é aplicado o sinal  $x[n]$ .

$$h[n] = u[n] - u[n - 10] \quad x[n] = u[n - 2] - u[n - 7]$$

Crie um programa que apresente graficamente a resposta de  $y[n]$ .

Utilize o intervalo de  $n = -10$  a  $100$  para gerar os sinais de entrada e resposta ao impulso.

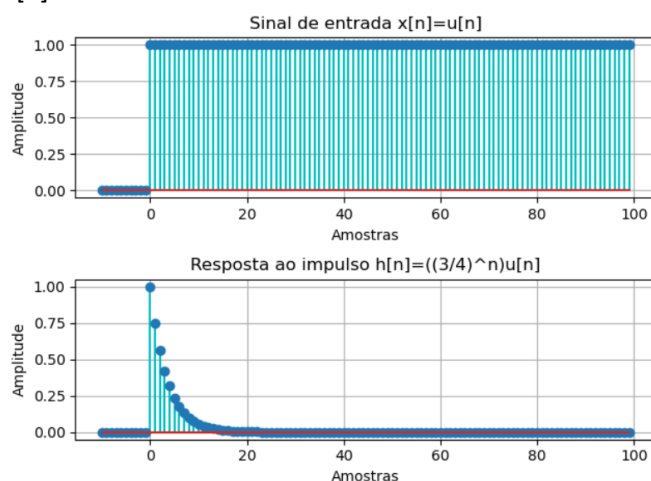
Cuidado com o intervalo de amostras  $n$  da resposta  $y[n]$

# Processamento Digital de Sinais - PDS

Cuidado com a resposta da função *convolve*.

- ✓ O início do tempo de  $y[n]$  é dado pela soma dos inícios dos tempos dos sinais em convolução  $h[n]$  e  $x[n]$ .
  - $ny[0] = nh[0] + nx[0]$
  - ex.:
    - Se  $x = [1, 2, -1, 2]$  e  $nx = [1, 2, 3, 4]$
    - Se  $h = [1, 2, 0, 0]$  e  $nh = [0, 1, 2, 3]$
    - Então a resposta de  $y$  vai iniciar em  $1(\text{de } x) + 0(\text{de } h) = 1$ ;
    - ficando  $ny = [1, 2, 3, \dots]$
- ✓ Cuidado também com o tamanho da resposta, pois o comprimento total da resposta pode não corresponder a resposta que os sinais que vc queria, pois eles foram truncados. O próximo exercício irá demonstrar este problema.

3. Suponha que um sistema H do tipo LTI com resposta impulso  $h[n]$  é aplicado o sinal  $x[n]$ .



$$x[n] = u[n]$$

$$h[n] = \left(\frac{3}{4}\right)^n u[n]$$

Figura 1 – Sinais do exercício 3

Ao usarmos o função *convolve* e ajustarmos o início de ( $ny = nx(0) + nh(0)$ ) obtemos um sinal mostrado na figura 2. Esta não é a resposta correta para os sinais apresentados.

# Processamento Digital de Sinais - PDS

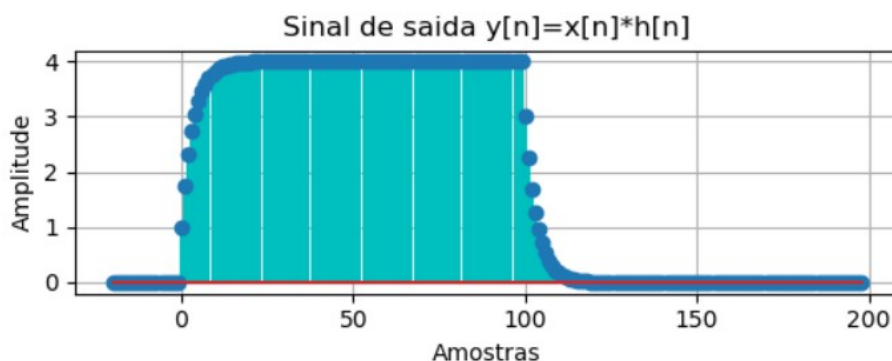


Figura 2 – Resposta completa da função convolve

A forma correta é apresentada abaixo

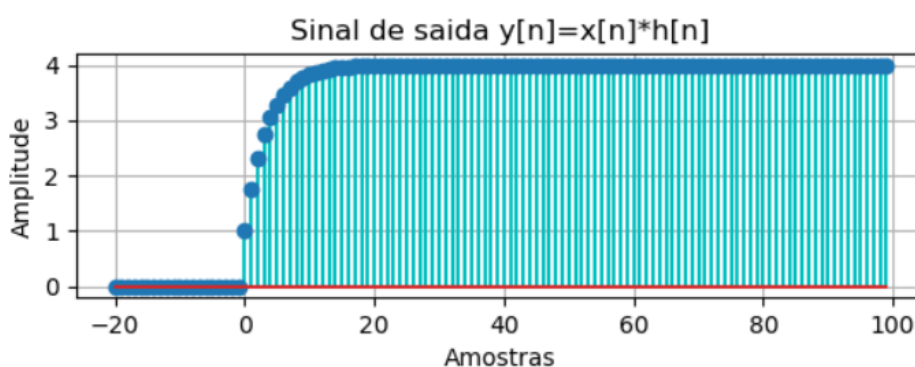


Figura 3 – Resposta válida para a convolução entre os sinais envolvidos

O problema ocorre porque tanto o sinal de  $x[n]$  e como de  $h[n]$  são definidos com valores até o infinito, mas como isso não pode ser representado no computador eles são truncados a partir da amostra 100. E a função convolve retorna um vetor de tamanho de  $(N+M-1)$ , sendo  $N$  a quantidade de amostras de  $x[n]$  e  $M$  a quantidade de amostras de  $h[n]$ . Para isso ele precisa que  $x[n]$  e  $h[n]$  tenham esse tamanho  $(N+M-1)$  de amostras, o que ele faz preenchendo os valores de  $x[n]$  e  $h[n]$  que ela não conhece com zeros.

Assim em vez de  $x[n]$  ser um função degrau unitário ele vira um pulso retangular de 0 até 100. Não é um problema muito complexo para ser resolvido, geralmente só precisa de uma boa análise dos limites dos sinais envolvidos na convolução.

- Então agora crie um programa que apresente graficamente  $x[n]$ ,  $h[n]$  e a resposta  $y[n]$ , sendo que

$$h[n] = 0.96^n \sin\left(\frac{\pi}{16}n\right) [u[n] - u[n-10]]$$

$$x[n] = u[n-2]$$

Utilize o intervalo de -10 a 100 para  $n$ .

Deixe bem claro qual intervalo do sinal  $y[n]$  é válido.

# Processamento Digital de Sinais - PDS

---

4. Apresente o programa em Python que esboce a resposta ao degrau dos sistemas abaixo: (Os gráficos devem estar no intervalo  $[-2 \leq n \leq 20]$  )

a.  $h[n] = \left(\frac{1}{2}\right)^n u[n]$

b.  $h[n] = \delta[n] - \delta[n-1]$

c.  $h[n] = (-1)^n [u[n+2] - u[n-3]]$

d.  $h[n] = u[n]$

e.  $h[n] = (-n)u[n]$

f.  $h[n] = \text{sen}\left[\frac{1}{12}\pi n\right] u[n-3]$