IV – Sistemas
Filtros FIR

Processamento Digital de Sinais

André Lourenço

Sumário

- Sistemas
  - Sistemas FIR
  - Resposta Impulsional
  - Convolução Linear
- Implementação de Sistemas Discretos
  - Diagramas de Blocos
- Propriedades dos Sistemas
- Sistemas Lineares e Invariantes no Tempo
  - Associação entre Sistemas

SISTEMAS

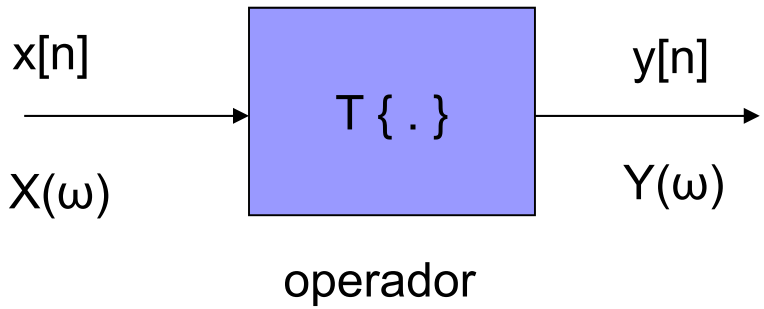
Sistema

- Definição:
- Um sistema tranforma uma sucessão que é o sinal de entrada $x[n]$ num outro sinal que é o de saída $y[n]$ por meio de uma função ou operação que se representa por:

$$y[n] = T\{x[n]\} \quad \text{ou} \quad y(t) = T\{x(t)\}$$

Onde x: sinal de entrada

y: sinal de saída



A caixa azul é o sistema (filtro) representado matematicamente pelo operador T

Neste capítulo, a nossa atenção vai para os sistemas em vez dos sinais.

Os sistemas são os FIR (Finite Impulse Response) digital filters

Porquê FIR?

Porque o sinal de saída do filtro é o resultado do cálculo de uma soma pesada finita de valores passados, presentes ou futuros do sinal de entrada do filtro.

$$y[n] = \sum_{k=-M_1}^{M_2} b_k x[n-k]$$

M_1, M_2 finitos.

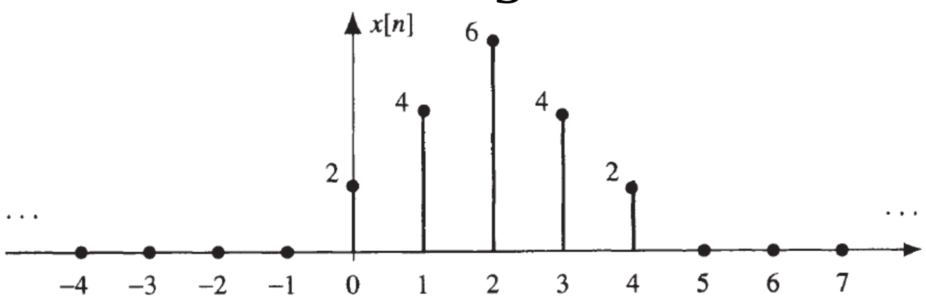
Exemplos

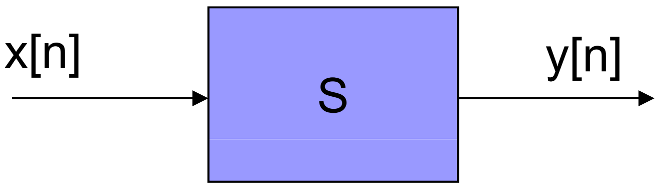
Um dos filtros FIR mais simples é a média que é usada para alisar ou suavizar dados muito flutuantes antes da sua interpretação.

- Filtro média móvel de 3 termos
  - Cálcula a “média móvel” de 3 ou mais amostras consecutivas de uma sequência, formando uma nova sequência com os valores médios. Neste caso

$$y[n]=\sum_{k=0}^{2} b_k x[n-k]=\frac{1}{3}x[n]+\frac{1}{3}x[n-1]+\frac{1}{3}x[n-2]=$$

$$=\frac{1}{3}\left(x[n]+x[n-1]+x[n-2]\right)$$





Exemplo (cont.)

- Outros exemplos:

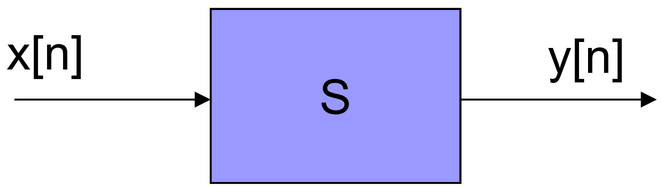
$$y[n]=x[n]-x[n-1]$$

$$y[n]=\left(x[n-1]\right)^3$$

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

$$y[n]=\left(x[n-1]\right)^3$$

- De notar que nestes filtros não entram na soma valores do sinal de saída.

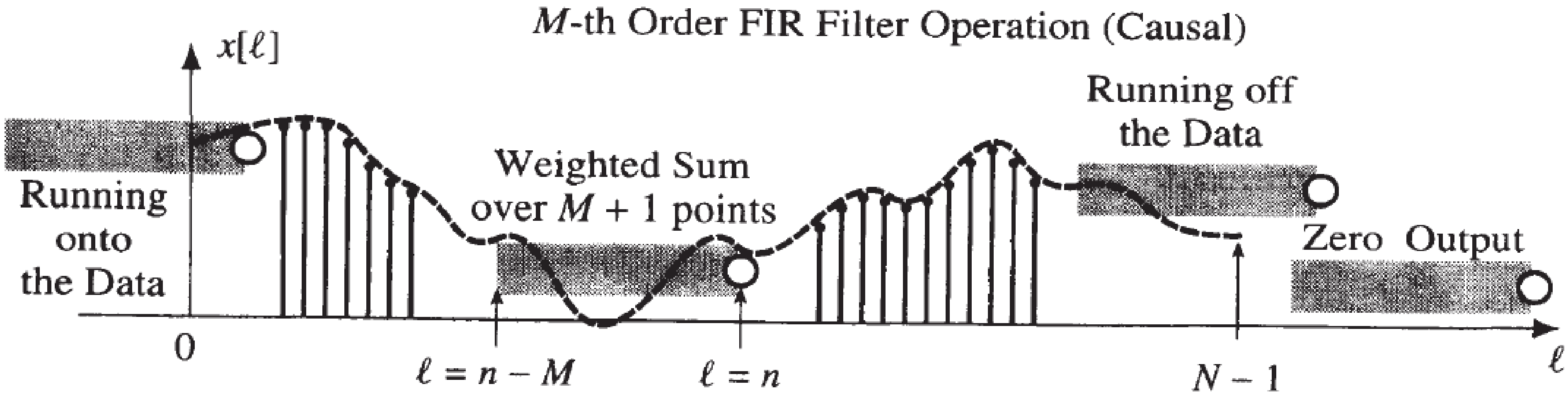


SISTEMAS FIR

Sistemas FIR

- Fórmula geral: $$y[n]=\sum_{k=0}^{M} b_k x[n-k]$$

Sliding window



M é a ordem do filtro e L (nº de coef. Do filtro) é o comprimento do filtro

Algumas definições

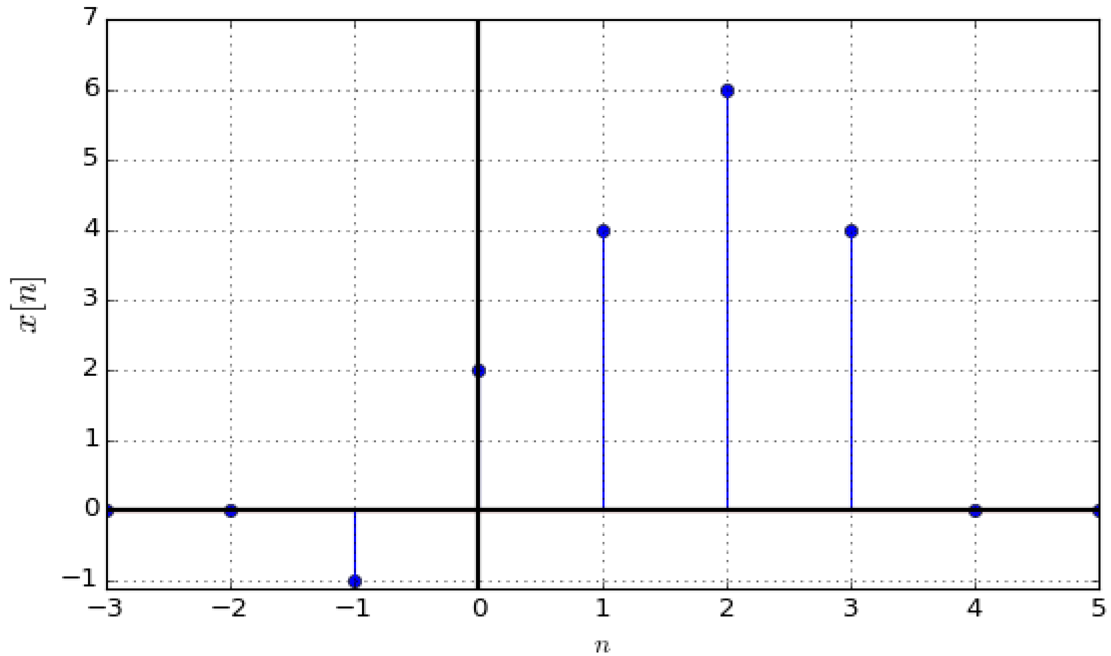
Filtro causal: Filtro cujo sinal de saída $y[n]$ no instante presente depende apenas dos valores do sinal de entrada no instante presente $x[n]$ e em instantes passados $x[n-k],\ k>0$.

Exemplo: $y[n]=\frac{1}{2}\left(x[n]+x[n-1]\right)$

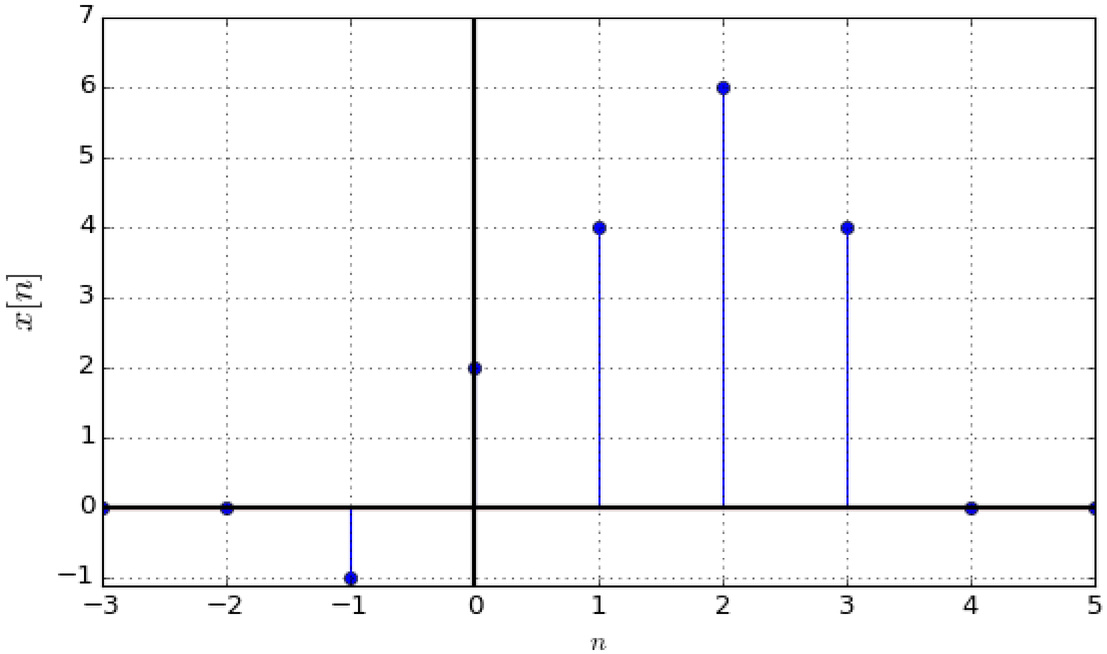
Suporte do sinal de entrada: valores de n para os quais o sinal não se torna totalmente nulo:

Suporte de $x[n]$:

$-1\leq n\leq 3$



Exemplo:
Considere o seguinte sinal discreto de entrada:



Obtenha o sinal de saída do filtro causal e finito de média móvel de 3 pontos:

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

M=2 ordem do filtro
L=3 nº coef. filtro

$$y[-1] = \frac{1}{3}\left(x[-1] + x[-2] + x[-3]\right) = \frac{1}{3}(-1) = -\frac{1}{3}$$

$$y[0] = \frac{1}{3}\left(x[0] + x[-1] + x[-2]\right) = \frac{1}{3}(-1+2) = \frac{1}{3}$$

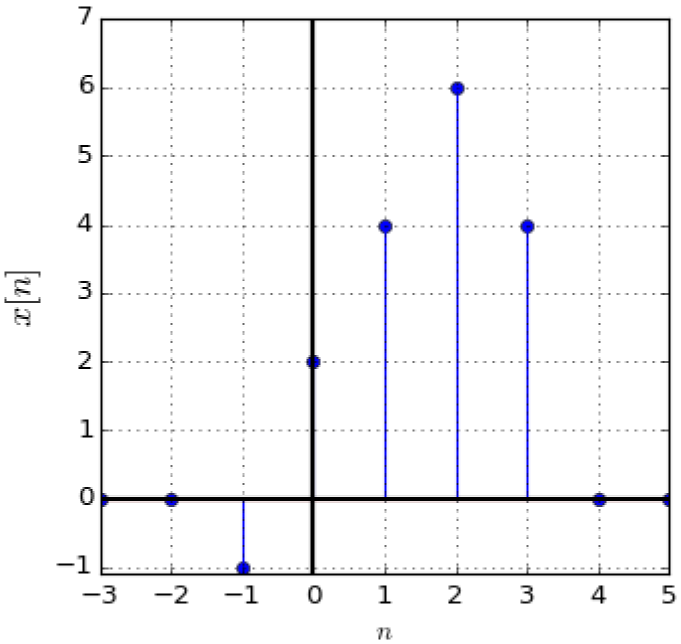
$$y[1]=\frac{1}{3}\left(x[1]+x[0]+x[-1]\right)=\frac{1}{3}(4+2-1)=\frac{5}{3}$$

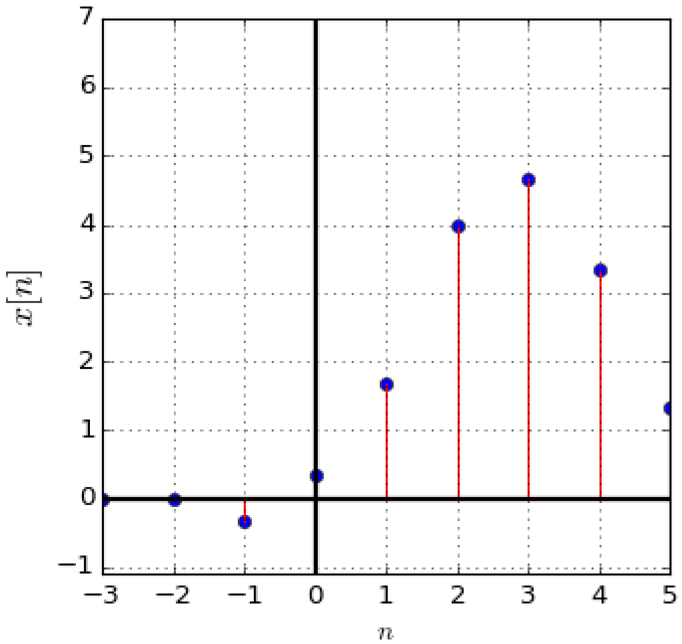
$$y[2]=\frac{1}{3}\left(x[2]+x[1]+x[0]\right)=\frac{1}{3}(6+4+2)=4$$

$$y[3]=\frac{1}{3}\left(x[3]+x[2]+x[1]\right)=\frac{1}{3}(4+6+4)=\frac{14}{3}$$

$$y[4]=\frac{1}{3}\left(x[4]+x[3]+x[2]\right)=\frac{1}{3}(0+4+6)=\frac{10}{3}$$

$$y[5]=\frac{1}{3}\left(x[5]+x[4]+x[3]\right)=\frac{1}{3}(0+0+4)=\frac{4}{3}$$





Um FIR é completamente definido através do conjunto dos seus coeficientes $\{b_k\}$

Exemplo: FIR com coeficientes $\{b_k\} = \{3, -1, 2, 1\}$

L=4 coef. -> M=3 -> FIR de ordem 3.
Preencha os espaços em amarelo na tabela.

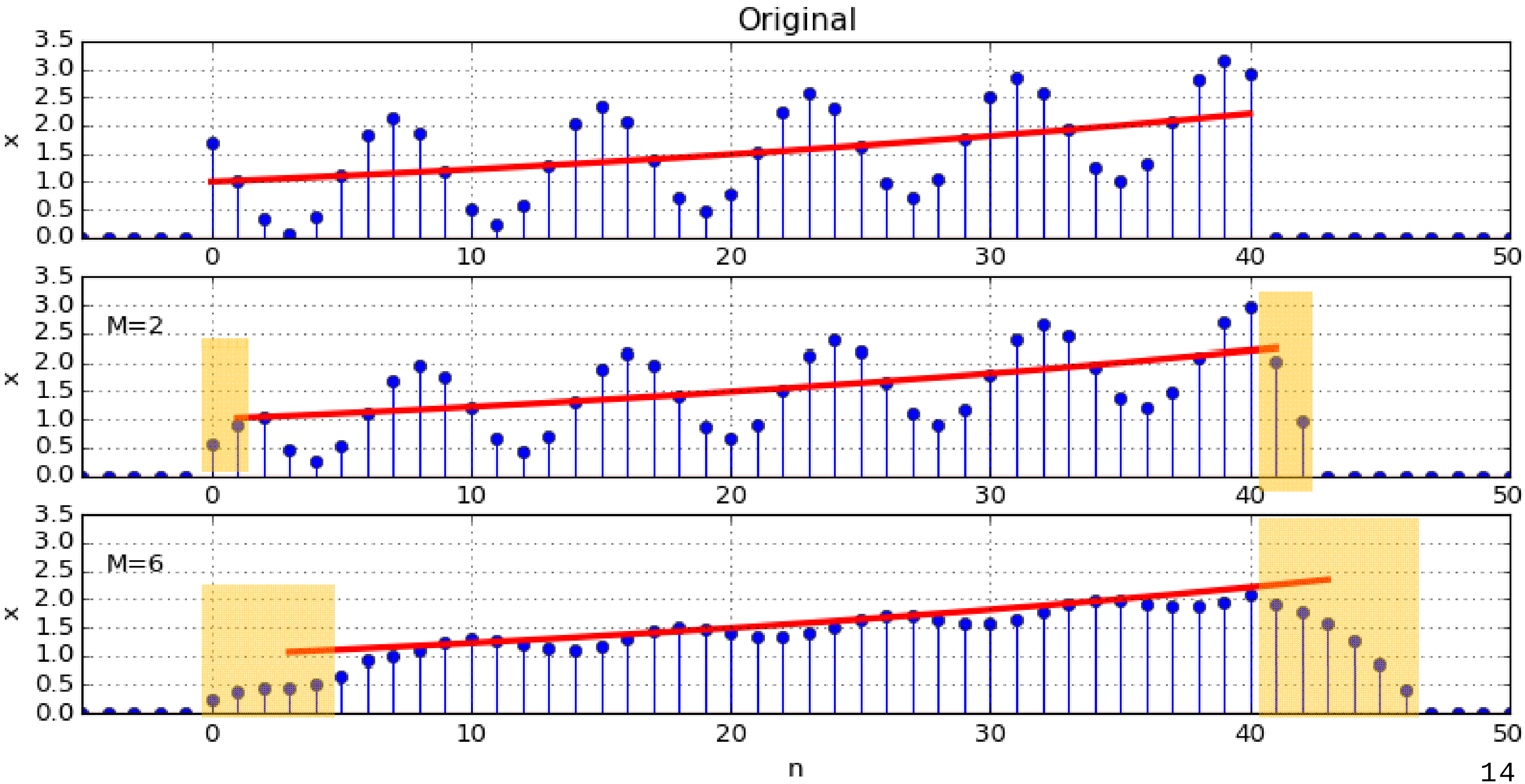
$$y[n] = \sum_{k=0}^{3} b_k x[n-k] = 3x[n] - 1x[n-1] + 2x[n-2] + 1x[n-3]$$

n	n<0	0	1	2	3	4	5	6	7	8	n>8
x[n]	0	2	4	6	4	2	0	0	0	0	0
y[n]	0	6	10	18				8	2	0	0

Considere o sinal:

$$x[n] = \begin{cases} 1.02^n + 0.5\cos\left(\dfrac{2\pi n}{8} + \dfrac{\pi}{4}\right) & , \quad 0 \le n \le 40 \\ 0 \quad , \quad \text{otherwise} \end{cases}$$

Utilize filtros FIR média móvel de 3 e 7 pontos para alisar o sinal de modo a detectar claramente a sua tendência exponencial.



Observações:

1. A sucessão de entrada é toda zeros antes de n=0, pelo que sendo o filtro causal, a sucessão de saída também é zero para n<0.

2. As zonas a amarelo assinalam valores de sinal de saída calculados com zeros do sinal de entrada no início e no fim.

3. A linha a vermelho representa a parcela do sinal 1.02^n

Três conceitosimportantes

Impulso Unitário

Resposta do filtro ao impulso unitário

Convolução

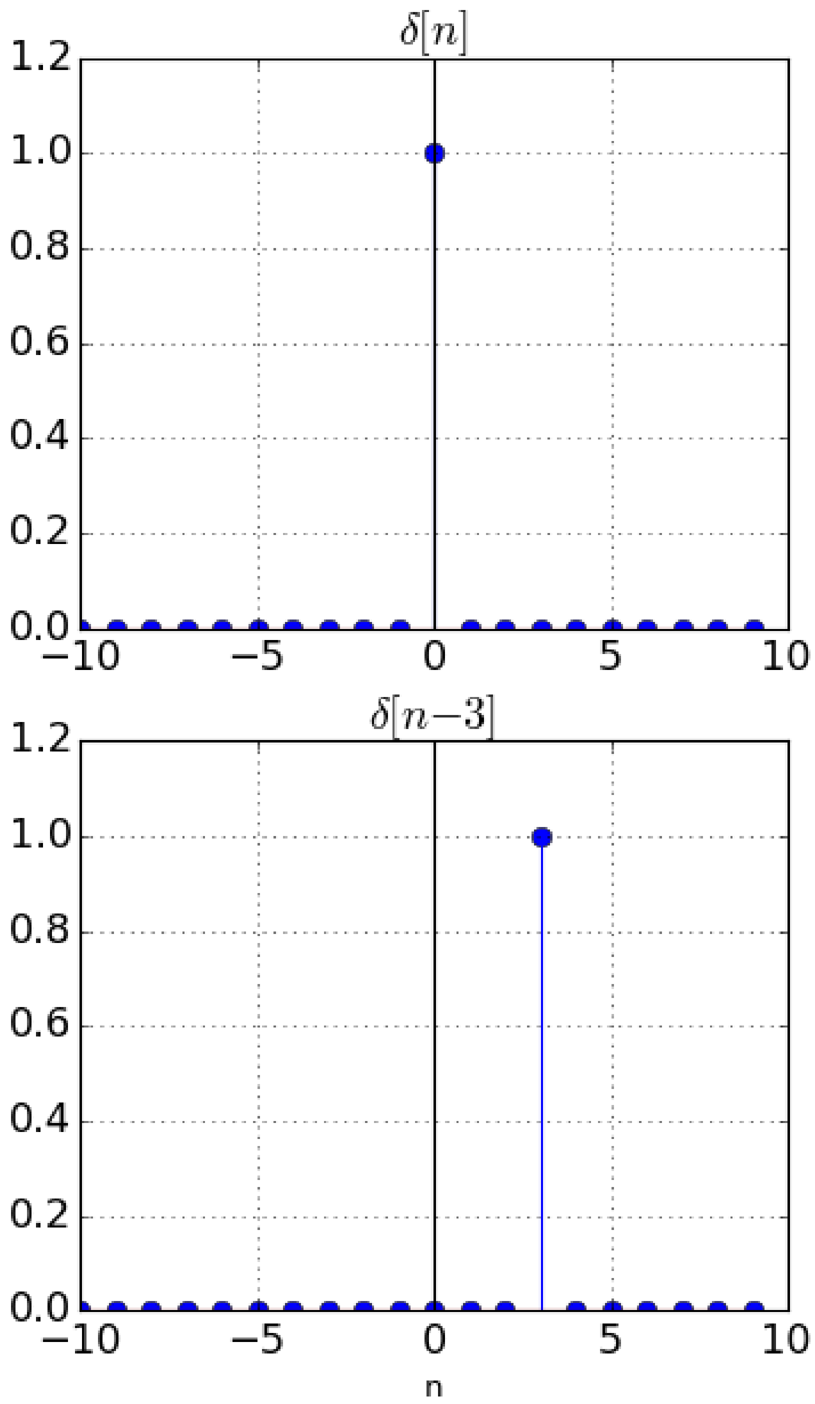
Impulso Unitário

- Tempo Discreto

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

Impulso unitário deslocado

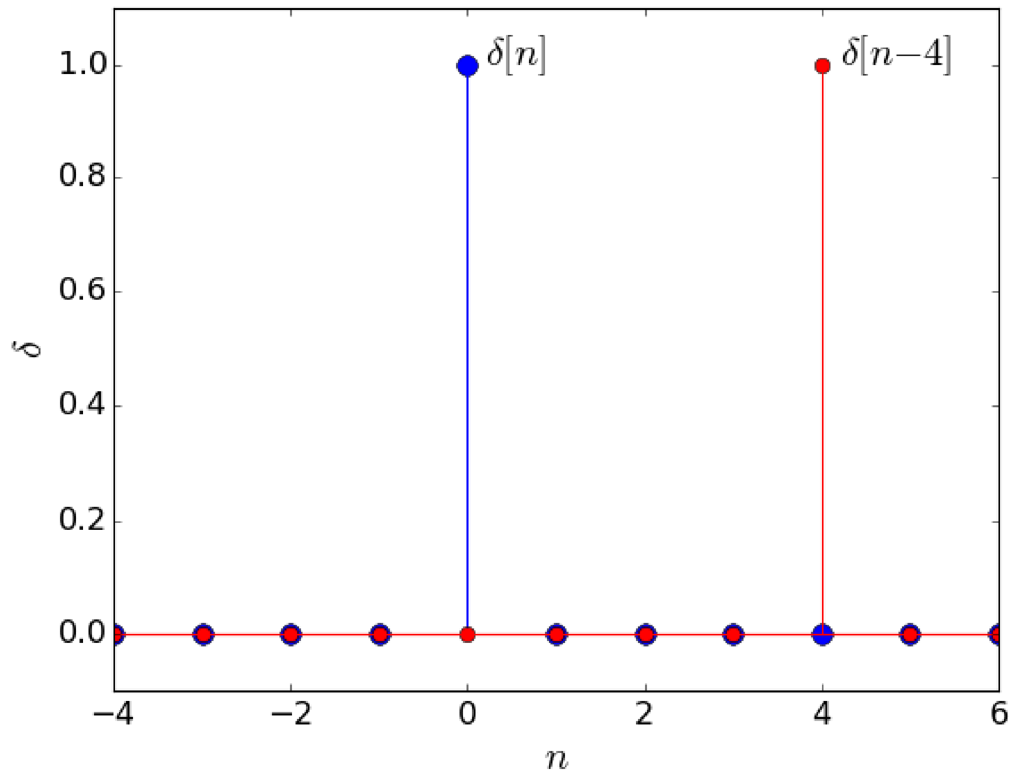
$$\delta[n-k]=\begin{cases}1 \;, & n=k\\0 \;, & n\neq k\end{cases}$$



André Lourenço

Exemplo

n	...	-4	-3	-2	-1	0	1	2	3	4	5	...
δ[n]	0	0	0	0	0	1	0	0	0	0	0	0
δ[n-4]	0	0	0	0	0	0	0	0	0	1	0	0



Exemplo

Seja o sinal discreto:

$$x[-1]=-1,\ \ x[0]=2,\ \ x[1]=4,\ \ x[2]=6,\ \ x[3]=4,\ \ x[4]=2 \text{ e zero cc}$$

Podemos escrever este sinal em termos do impulso unitário como:

$$x[n]=-1\delta[n+1]+2\delta[n]+4\delta[n-1]+6\delta[n-2]+4\delta[n-3]+2\delta[n-4]$$

No caso geral de um sinal discreto, podemos escrever:

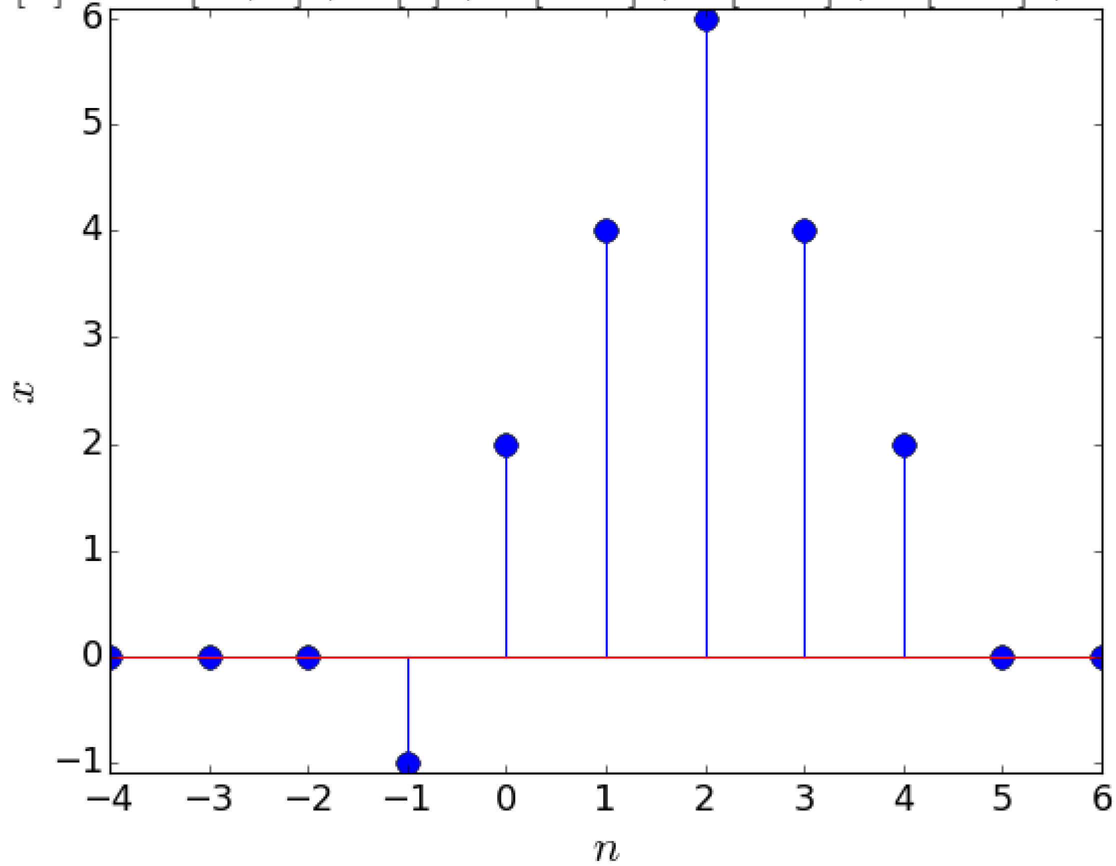
$$x[n]=\sum_k x[k]\delta[n-k]$$

Exemplo:

$$x[n] = x[-1]\delta[n] + x[0]\delta[n] + x[1]\delta[n-1] + x[2]\delta[n-2] + x[3]\delta[n-3] + x[4]\delta[n-4] =$$

$$= \sum_{k=-1}^{4} x[k]\delta[n-k]$$

$x[n] = -1\delta[n+1] + 2\delta[n] + 4\delta[n-1] + 6\delta[n-2] + 4\delta[n-3] + 2\delta[n-4]$



$x[-1] = -1$

$x[0] = 2$

$x[1] = 4$

$x[2] = 6$

$x[3] = 4$

$x[4] = 2$

Impulso Unitário

- Tempo Contínuo

$$\delta(t) = 0 \quad , \quad t \neq 0$$

$$\delta(at) = \frac{1}{|a|}\delta(t)$$

$$\int_{-\infty}^{+\infty} \delta(t)\,dt = 1$$

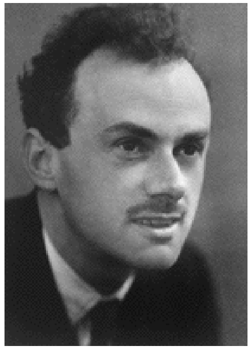
$$\int_{-\infty}^{+\infty} k\delta(t)\,dt = k$$

$$\int_{-\infty}^{+\infty} \delta(t - t_0)\,dt = 1$$

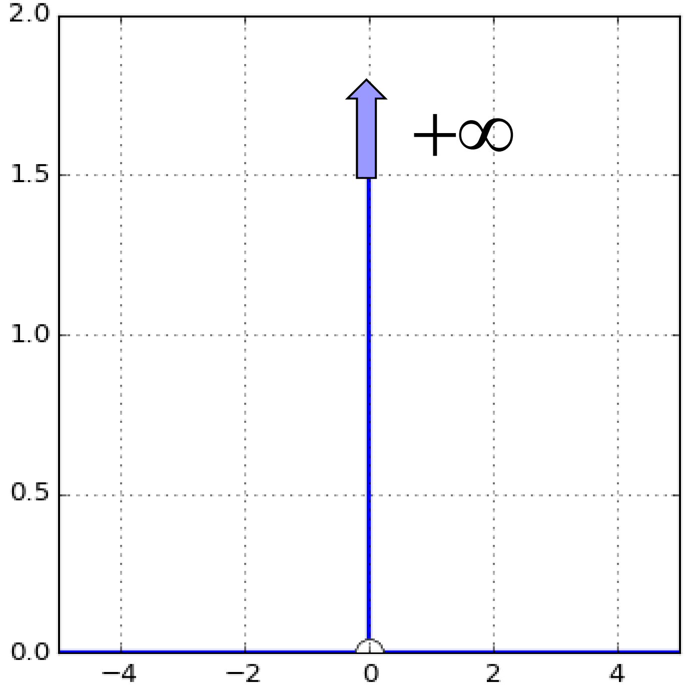
$$\int_{-\infty}^{+\infty} \delta(\tau - t_0)\,d\tau = u(t - t_0)$$

Paul Dirac
(1902-1984)

Dirac shared the 1933 Nobel Prize for physics with Erwin Schrödinger

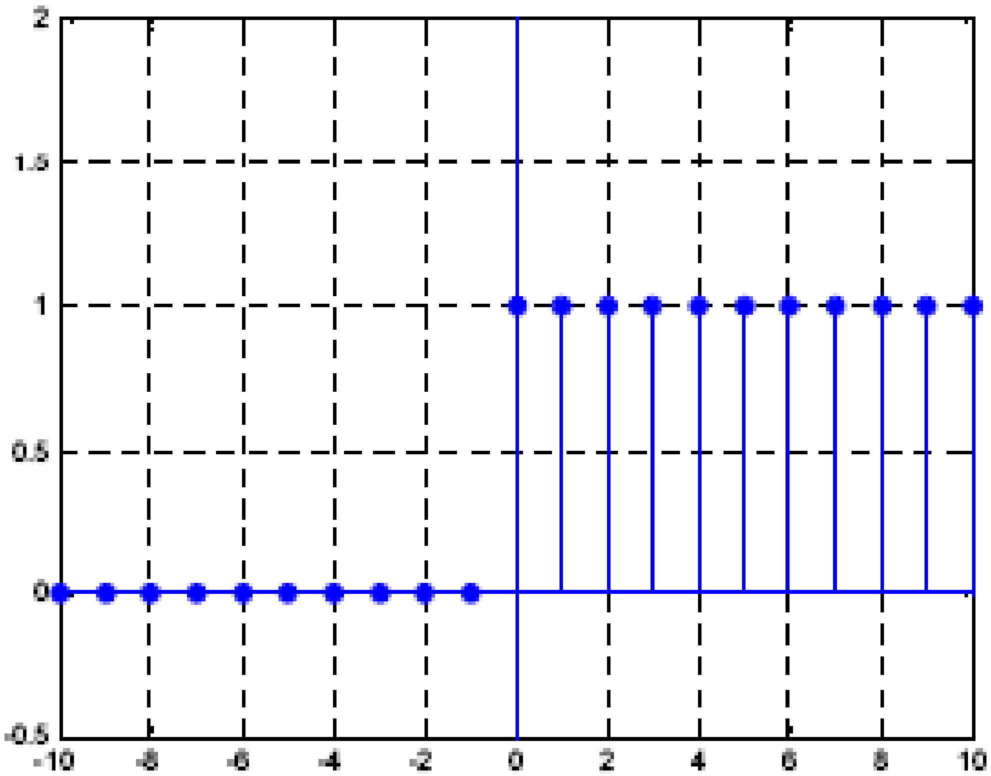


Delta de *Dirac*



Escalão Unitário

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



- Relações úteis

$$\delta[n] = u[n] - u[n-1]$$

$$u[n] = \sum_{m=-\infty}^{n} \delta[m] = \sum_{i=0}^{\infty} \delta[n-i]$$

$$x[n]\delta[n] = x[0]\delta[n]$$

$$\delta[n-n_0] = u[n-n_0] - u[n-n_0-1]$$

$$u[n-n_0] = \sum_{m=-\infty}^{n} \delta[m-n_0] = \sum_{i=0}^{\infty} \delta[n-n_0-i]$$

$$x[n]\delta[n-n_0] = x[0]\delta[n-n_0]$$

Resposta Impulsional

- h[n] : resposta impulsional (ou impulsiva)
  Resposta do sistema a um impulso unitário

$$x[n]=\delta[n] \longrightarrow \boxed{\mathbf{T\{.\}}} \longrightarrow y[n]=h[n]$$

Sistemas FIR e Resposta Impulsional

- FIR: Finite Impulse Response

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

$$x[n]=\delta[n] \longrightarrow \boxed{\mathbf{T\{.\}}} \longrightarrow y[n]=h[n]$$

- Resposta ao impulso unitário de um FIR

de coef. $\{b_k\}$

$$h[n]=\sum_{k=0}^{M} b_k \delta[n-k]=\begin{cases} b_n & n=0,1,2,\ldots,M \\ 0 & \text{cc} \end{cases}$$

Como os coeficientes do filtro são iguais à resposta impulsional do filtro $h[k]$, podemos substituir em:

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

b_k por $h[k]$ e escrever:

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

A saída do FIR fica expressa em termos do sinal de entrada e da resposta impulsional.
Esta operação é a ***convolução finita*** da resposta impulsional do FIR com o sinal de entrada.

What terms are used in describing FIR filters?

•***Impulse Response*** - The "impulse response" of a FIR filter is actually just the set of FIR coefficients. (If you put an "impulse" into a FIR filter which consists of a "1" sample followed by many "0" samples, the output of the filter will be the set of coefficients, as the 1 sample moves past each coefficient in turn to form the output.)

•***Tap*** - A FIR "tap" is simply a coefficient/delay pair. The number of FIR taps, (often designated as "N") is an indication of :

1) the amount of memory required to implement the filter,
2) the number of calculations required, and
3) the amount of "filtering" the filter can do; in effect, more taps means more stop-band attenuation, less ripple, narrower filters, etc.

•***Transition Band*** -

•The band of frequencies between passband and stopband edges.

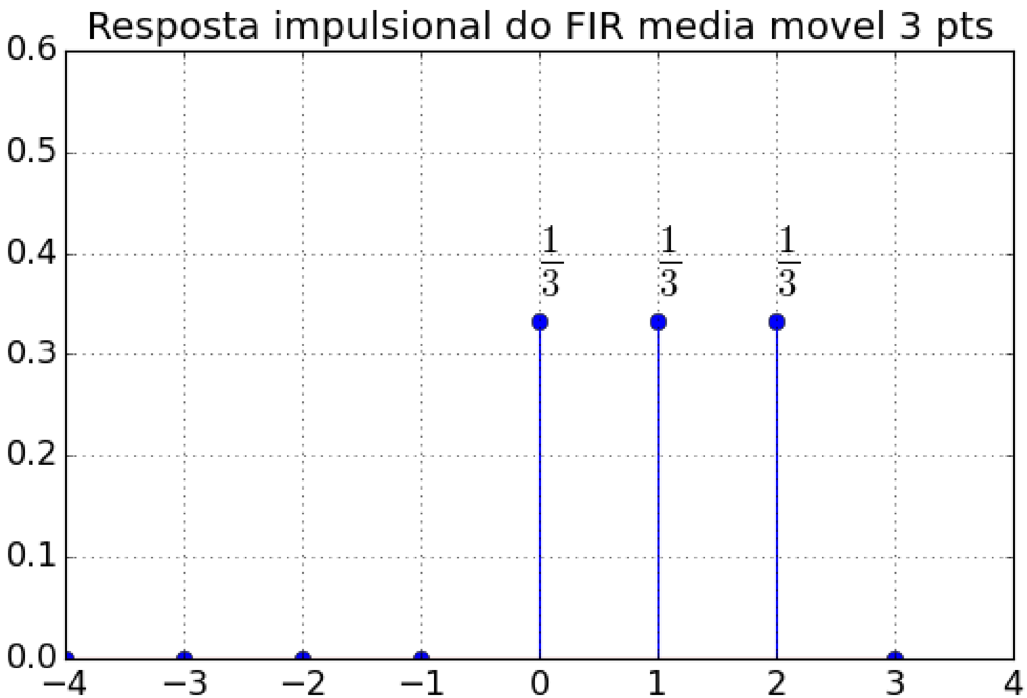
•The narrower the transition band, the more taps are required to implement the filter. (A "small" transition band results in a "sharp" filter.)

Exemplo

Determinar e representar graficamente a resposta impulsional do filtro de média móvel de 3 pontos.

$$\{b_k\} = \left\{ \tfrac{1}{3}, \tfrac{1}{3}, \tfrac{1}{3} \right\} \qquad M = 2 \quad , \quad L = 3$$

$$h[n] = \sum_{k=0}^{2} b_k \delta[n-k] = \underbrace{\tfrac{1}{3}\delta[n]}_{\frac{1}{3} \text{ em n=0}} + \underbrace{\tfrac{1}{3}\delta[n-1]}_{\frac{1}{3} \text{ em n=1}} + \underbrace{\tfrac{1}{3}\delta[n-2]}_{\frac{1}{3} \text{ em n=2}}$$



Exercício

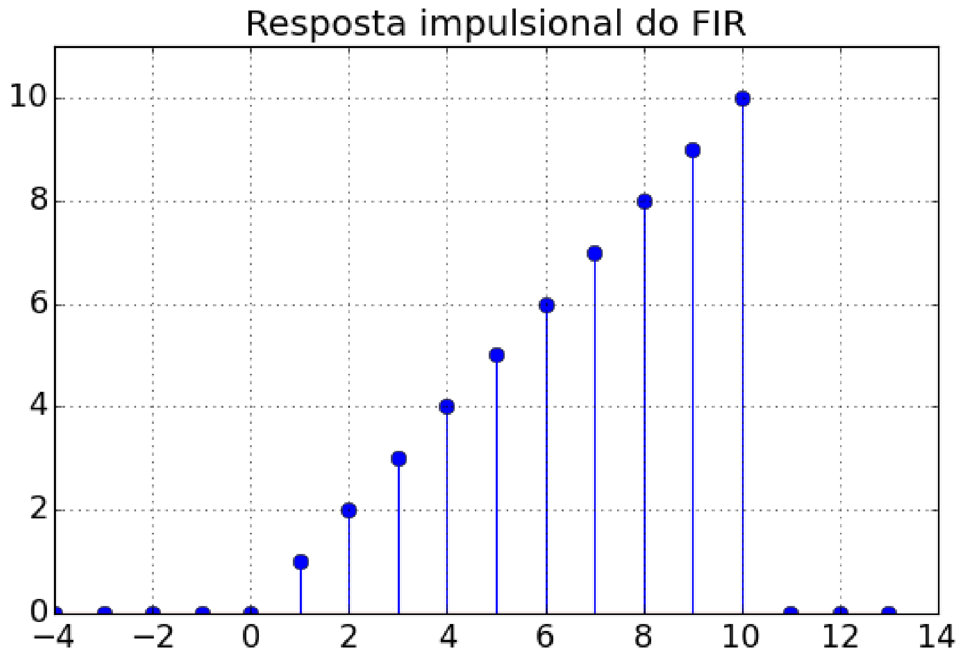
Determinar e representar graficamente a resposta impulsional do filtro FIR

$$y[n] = \sum_{k=0}^{10} kx\left[n-k\right]$$

Res.

Substitui-se $x\left[n-k\right]$ pelo respectivo impulso $\delta\left[n-k\right]$ obtendo-se:

$$h[n] = \sum_{k=0}^{10} k\delta\left[n-k\right] = 0\delta\left[n\right] + 1\delta\left[n-1\right] + 2\delta\left[n-2\right] + \ldots + 10\delta\left[n-10\right]$$



Exercício

Determinar os coeficientes do filtro FIR que produz um atraso de 3 pontos no sinal.

Res.

Pretende-se que a sequência $x[n]$ seja transformada na sequência $y[n] = x[n-3]$ ou seja o valor de x que estava em zero passa a estar em 3, o que estava em 1 passa a estar em 4, etc.

$$y[n] = \sum_{k=0}^{3} b_k x[n-k] = 0x[n] + 0x[n-1] + 0x[n-2] + 1x[n-3] =$$

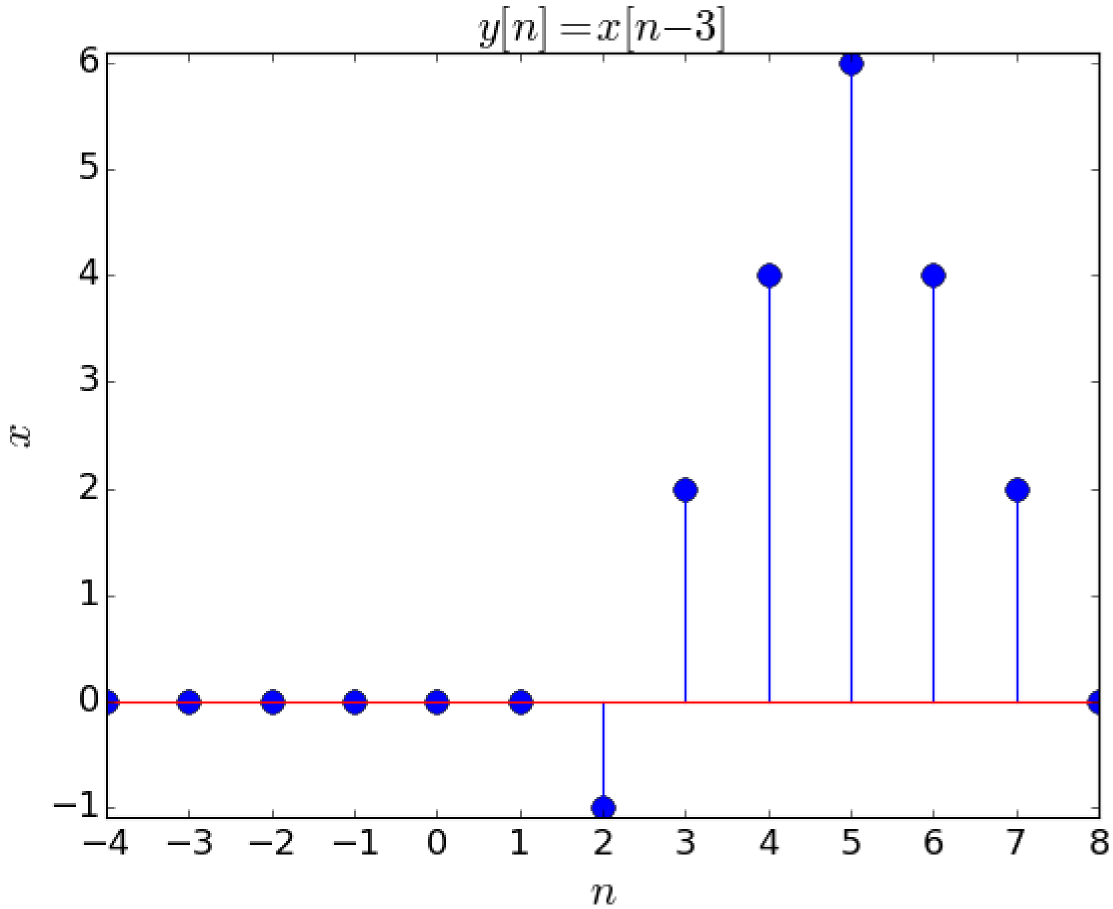
$$= x[n-3] \qquad \{b_k\} = \{0,0,0,1\}$$

Neste caso a resposta impulsional deste filtro é :

$$h[n] = \delta[n-3]$$

Exemplo:

$$y[n] = x[n-3]$$



CONVOLUÇÃO LINEAR

Convolução Linear

- Chama-se convolução linear à operação:

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

- Permite conhecendo a resposta impulsional, h[n], e a entrada x[n], determinar a saída y[n]

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

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

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

Esta equação só pode ser implementada directamente no computador se os sinais estiverem limitados a um suporte finito.

•Escolhe-se n = 0 para início das duas sucessões x e h;
•Seja K + 1 o valor para o qual y [n] = 0 para todo n > K + 1;
•Seja M + 1 valor para o qual x [n] = 0 para todo n > M + 1;

then the discrete convolution expression is:

$$y[n] = \sum_{k=\max(n-M,0)}^{\min(n,K)} x[k]\, h[n-k]$$

NOTA: A convolução 1D está implementada em SciPy por meio da função signal.convolve ou em Numpy pela função numpy.convolve

Suponhamos que K>= M: explicitando os cálculos tem-se:

$$
\begin{aligned}
y[0] &= x[0]\,h[0] \\
y[1] &= x[0]\,h[1] + x[1]\,h[0] \\
y[2] &= x[0]\,h[2] + x[1]\,h[1] + x[2]\,h[0] \\
\vdots \quad & \vdots \quad \vdots \\
y[M] &= x[0]\,h[M] + x[1]\,h[M-1] + \cdots + x[M]\,h[0] \\
y[M+1] &= x[1]\,h[M] + x[2]\,h[M-1] + \cdots + x[M+1]\,h[0] \\
\vdots \quad & \vdots \quad \vdots \\
y[K] &= x[K-M]\,h[M] + \cdots + x[K]\,h[0] \\
y[K+1] &= x[K+1-M]\,h[M] + \cdots + x[K]\,h[1] \\
\vdots \quad & \vdots \quad \vdots \\
y[K+M-1] &= x[K-1]\,h[M] + x[K]\,h[M-1] \\
y[K+M] &= x[K]\,h[M]\,.
\end{aligned}
$$

Assim, a convolução discreta completa de duas sucessões finitas de comprimentos K + 1 e M + 1 respectivamente, origina uma sucessão finita de comprimento **(K + 1) + (M + 1) – 1= K + M + 1**

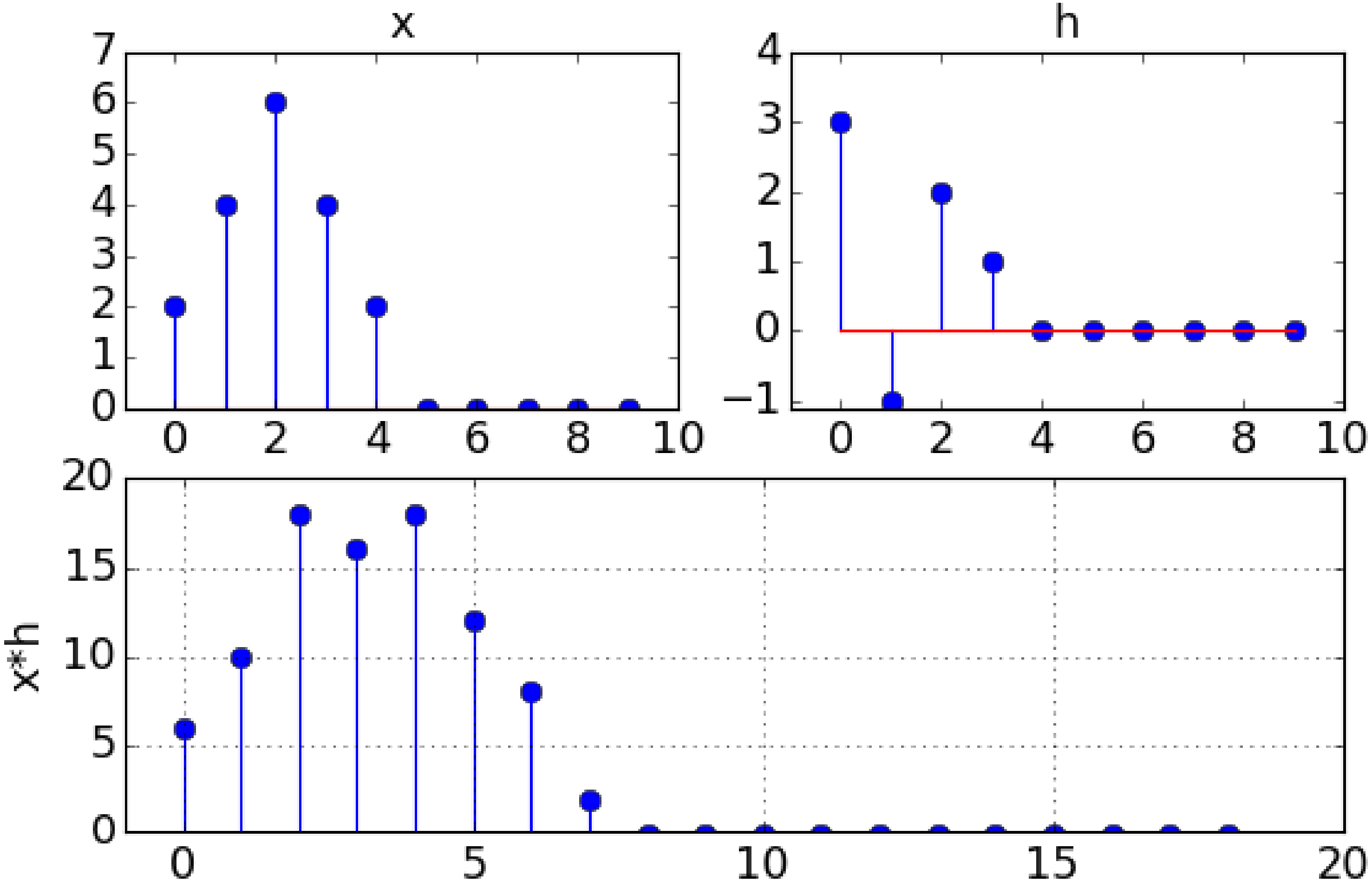
Estas operações podem-se por exemplo, traduzir no seguinte algoritmo prático:

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

n	0	1	2	3	4	5	6	7	8
x[n]	2	4	6	4	2				
h[n]	3	-1	2	1					
h[0]x[n-0]	6	12	18	12	6				
h[1]x[n-1]		-2	-4	-6	-4	-2			
h[2]x[n-2]			4	8	12	8	4		
h[3]x[n-3]				2	4	6	4	2	
y[n]	6	10	18	16	18	12	8	2	

Vejamos o que se passou com o sinal $x[n]$ quando convoluído com $h[n]$

$$x = [2, 4, 6, 4, 2] \qquad h = [3, -1, 2, 1]$$



numpy.convolve

numpy.convolve(*a, v, mode='full'*) [source]

Returns the discrete, linear convolution of two one-dimensional sequences.

The convolution operator is often seen in signal processing, where it models the effect of a linear time-invariant system on a signal [R17]. In probability theory, the sum of two independent random variables is distributed according to the convolution of their individual distributions.

If *v* is longer than *a*, the arrays are swapped before computation.

Parameters: **a** : *(N,) array_like*

First one-dimensional input array.

v : *(M,) array_like*

Second one-dimensional input array.

mode : *{'full', 'valid', 'same'}, optional*

'full':

By default, mode is 'full'. This returns the convolution at each point of overlap, with an output shape of (N+M-1,). At the end-points of the convolution, the signals do not overlap completely, and boundary effects may be seen.

'same':

Mode *same* returns output of length `max(M, N)`. Boundary effects are still visible.

'valid':

Mode *valid* returns output of length `max(M, N) - min(M, N) + 1`. The convolution product is only given for points where the signals overlap completely. Values outside the signal boundary have no effect.

Returns: **out** : *ndarray*

Discrete, linear convolution of *a* and *v*.

scipy.signal.convolve

scipy.signal.convolve(*in1*, *in2*, *mode='full'*) [so

Convolve two N-dimensional arrays.

Convolve *in1* and *in2*, with the output size determined by the *mode* argument.

Parameters:

in1 : *array_like*

First input.

in2 : *array_like*

Second input. Should have the same number of dimensions as *in1*; if sizes of *in1* and *in2* are not equal then *in1* has to be the larger array.

mode : *str {'full', 'valid', 'same'}, optional*

A string indicating the size of the output:

`full`

The output is the full discrete linear convolution of the inputs. (Default)

`valid`

The output consists only of those elements that do not rely on the zero-padding.

`same`

The output is the same size as *in1*, centered with respect to the 'full' output.

Returns:

convolve : *array*

An N-dimensional array containing a subset of the discrete linear convolution of *in1* with *in2*.

Propriedades da Convolução

- Linearidade

$$h[n] * \left(Ka[n] + Jb[n] \right) = h[n] * Ka[n] + h[n] * Jb[n]$$

- Comutatividade

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

- Associatividade

$$\left(x_1[n] * x_2[n] \right) * x_3[n] = x_1[n] * \left(x_2[n] * x_3[n] \right)$$

- Convolução com Impulso unitário

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

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

Exemplo

- Cálcule a convolução linear entre x[n] e h[n]
- $h[n]=\delta[n]+2\delta[n-1]$,
- $x[n] = \delta[n]+\delta[n-1] - \delta[n-3]$

Algoritmo

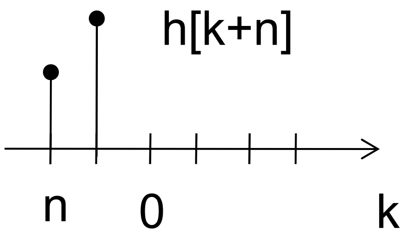
- Para cada n:
  - Obter $h[n-k]$
  - Multiplicar $x[k]\, h[n-k]$
  - Somar o valor obtido

De forma alternativa:

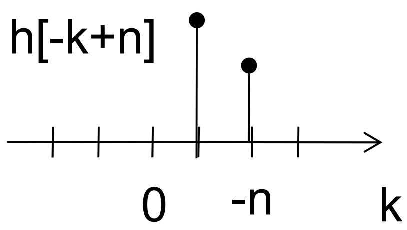
- Para cada n:
  - Obter $x[n-k]$
  - Multiplicar $h[k]\, x[n-k]$
  - Somar o valor obtido

Link –
[Joy of Convolution (discrete time)]

1º: deslocamento



2º: reflexão



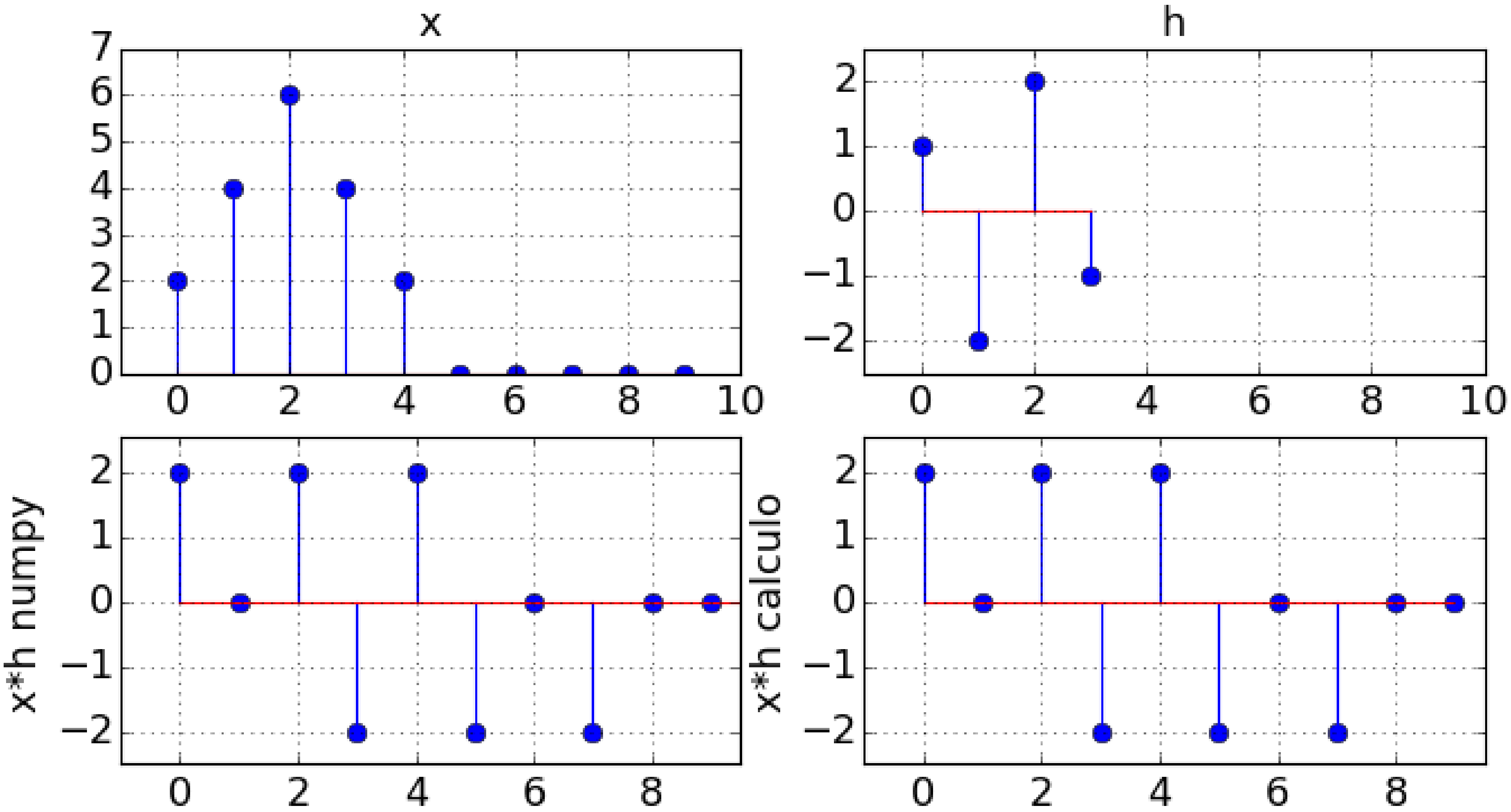
Exercício

Use o algorítmo da convolução para determinar o sinal de saida y[n] para o filtro de ordem 3 de coeficientes $\{b_k\} = \{1, -2, 2, -1\}$ e sinal de entrada $x = [2, 4, 6, 4, 2]$

x	2	4	6	4	2			
h	1	-2	2	-1				
	2	4	6	4	2			
		-4	-8	-12	-8	-4		
			4	8	12	8	4	
				-2	-4	-6	-4	-2
x*h	2	0	2	-2	2	-2	0	-2

Comprimento de x = 5 -> M=4
Comprimento de h = 4 -> K=3

comprimento da convolução = 4 + 3 + 1



x
h
x*h numpy
x*h calculo

@python

```
x=array([2,4,6,4,2])
h=array([1,-2,2,-1])
y=numpy.convolve(x,h ,mode='full')
y=[2,0,2,-2,2,-2,0,-2]
```

Exemplo: Usando a convolução, calcule o seguinte produto de polinómios:

$$p(x) = \left(1 + 2x + 3x^2 + 5x^4\right)\left(1 - 3x - x^2 + x^3 + 3x^4\right)$$

p1	1	2	3	0	5				
p1	1	-3	-1	1	3				
	1	2	3	0	5				
		-3	-6	-9	0	-15			
			-1	-2	-3	0	-5		
				1	2	3	0	5	
					3	6	9	0	15
P(x)	**1**	**-1**	**-4**	**-10**	**7**	**-6**	**4**	**5**	**15**

```
x=array([1,2,3,0,5])
h=array([1,-3,-1,1,3])
y=numpy.convolve(x,h ,mode='full')
y=[1,-1,-4,-10,7,-6,4,5,15]
```

Em Python os sistemas FIR são implementados com np.convolve().
Considere o seguinte excerto de script Python que permite calcular a convolução de hh que é a resposta impulsional do sistema da média móvel 11 pontos, com o sinal sinusoidal xx de 51 pontos

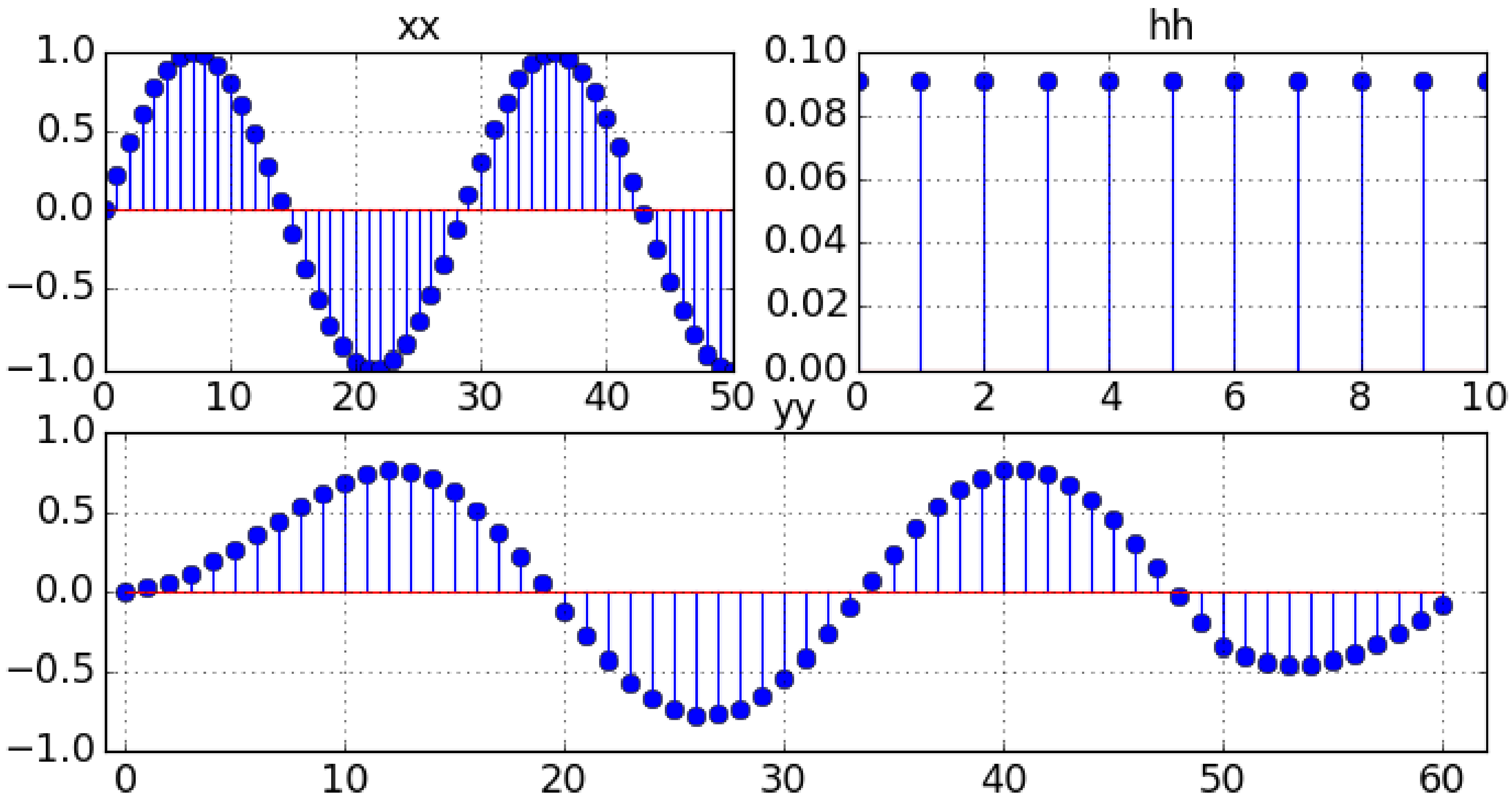
```
import numpy as np
from matplotlib import pyplot as plt

n=np.arange(0,51)
xx=np.sin(0.07*np.pi*n)
hh=np.ones(11)/11.
yy=np.convolve(xx,hh,mode='full')
```

Determine o comprimento do sinal de saída yy

Res:
Compr(yy)=compr(xx) + compr(hh) – 1= 51 +11 – 1 = 61 pontos

Pode confirmar na fig do slide seguinte



Nota: de 0 a 50 temos 51 pontos, de 0 a 10 temos 11 pontos e de 0 a 60 temos 61 pontos

DIAGRAMAS DE BLOCOS

Atendendo à definição de um FIR:

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

Assim precisamos de um modo de fazer o seguinte:

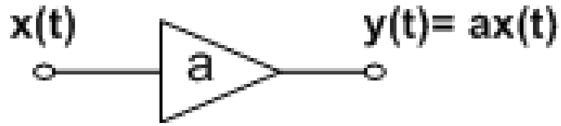
•Multiplicar sinais de entrada deslocados pelos respectivos valores dos coeficientes do filtro
•Adicionar sucessões
•Obter versões deslocadas do sinal de entrada.

Para representar esquematicamente este tipo de operações usam-se diagramas de blocos.

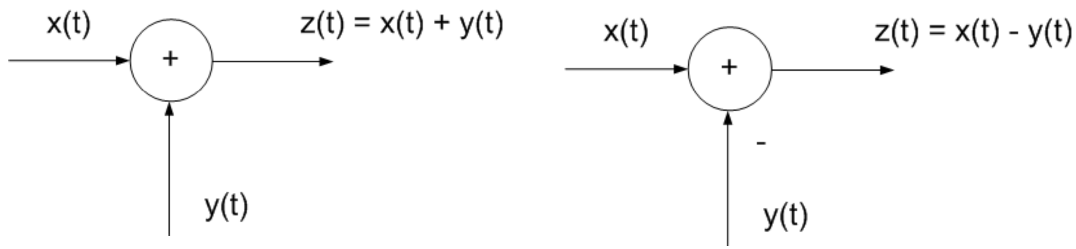
Diagramas de Blocos

Simbologia usada:

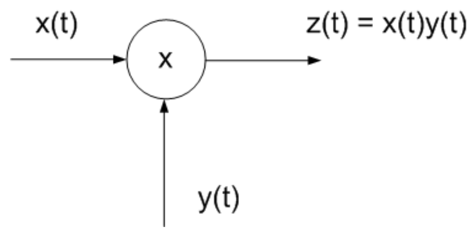
- Amplificação



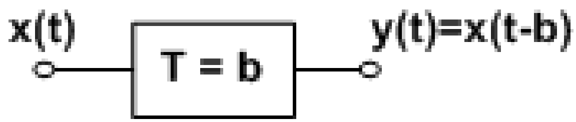
- Soma / subtracção



- Multiplicação



- Atraso

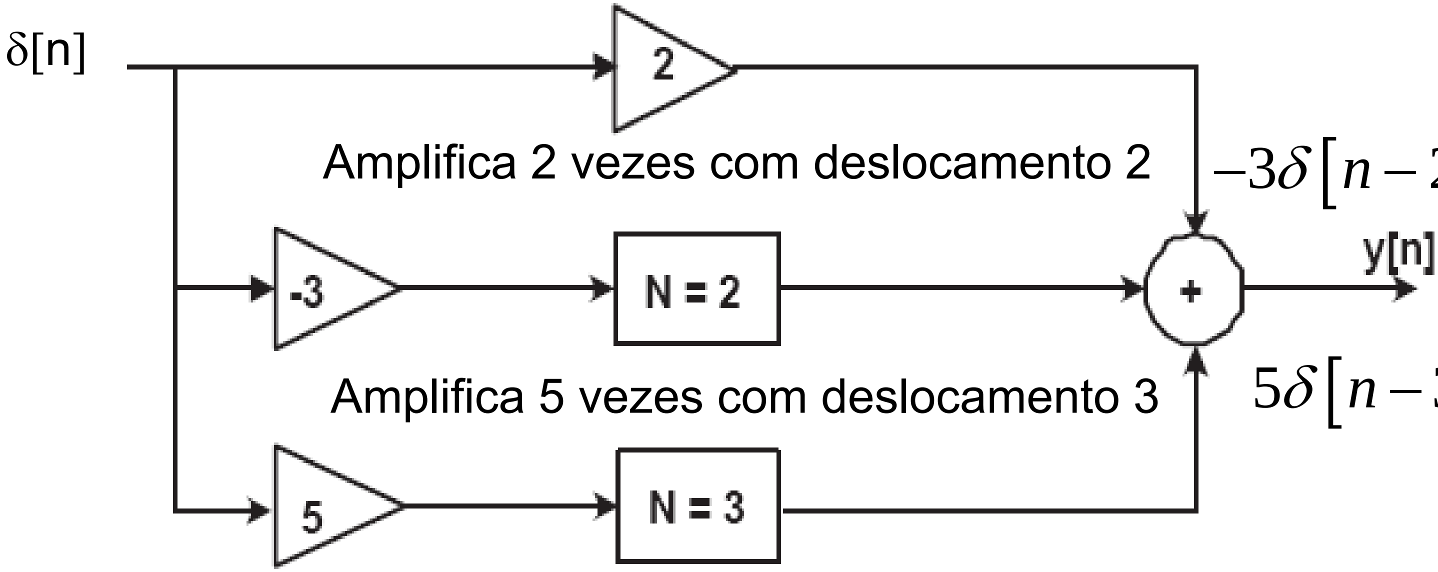


Resposta Impulsional

Amplifica 2 vezes sem deslocamento $2\delta[n]$

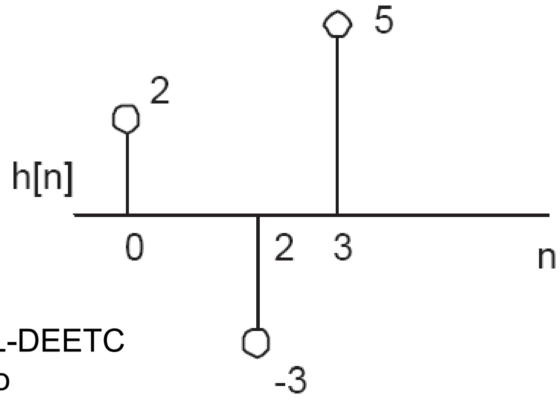
Amplifica 2 vezes com deslocamento 2 $-3\delta[n-2]$

Amplifica 5 vezes com deslocamento 3 $5\delta[n-3]$



e soma

- Qual a resposta impulsional?

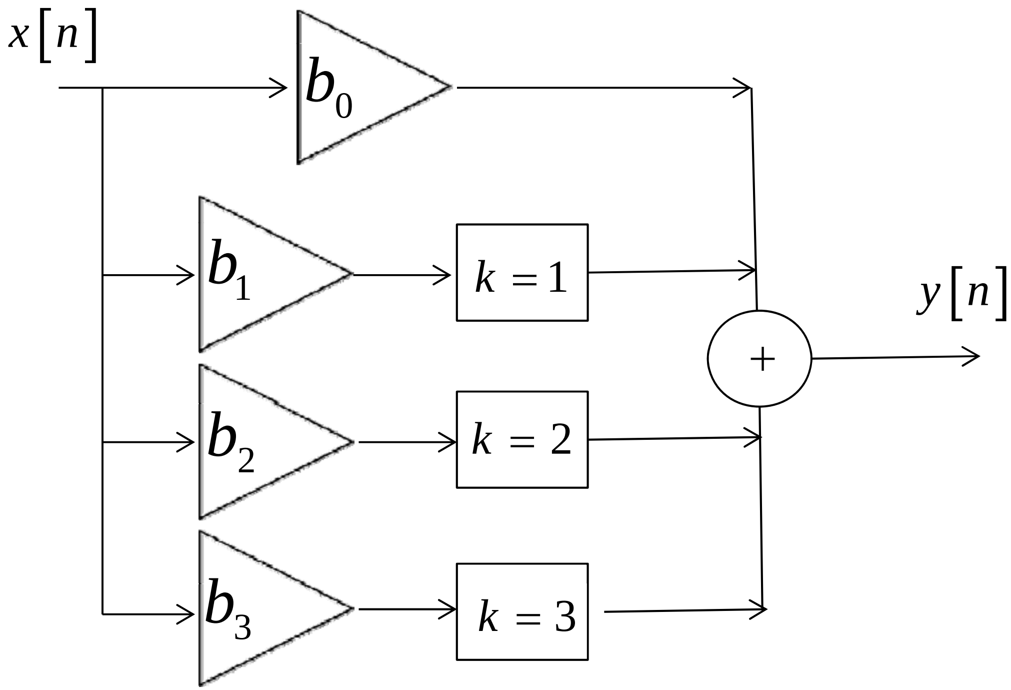


$$h[n] = 2\delta[n] - 3\delta[n-2] + 5\delta[n-3]$$

Exemplo

Considere o seguinte diagrama de blocos.

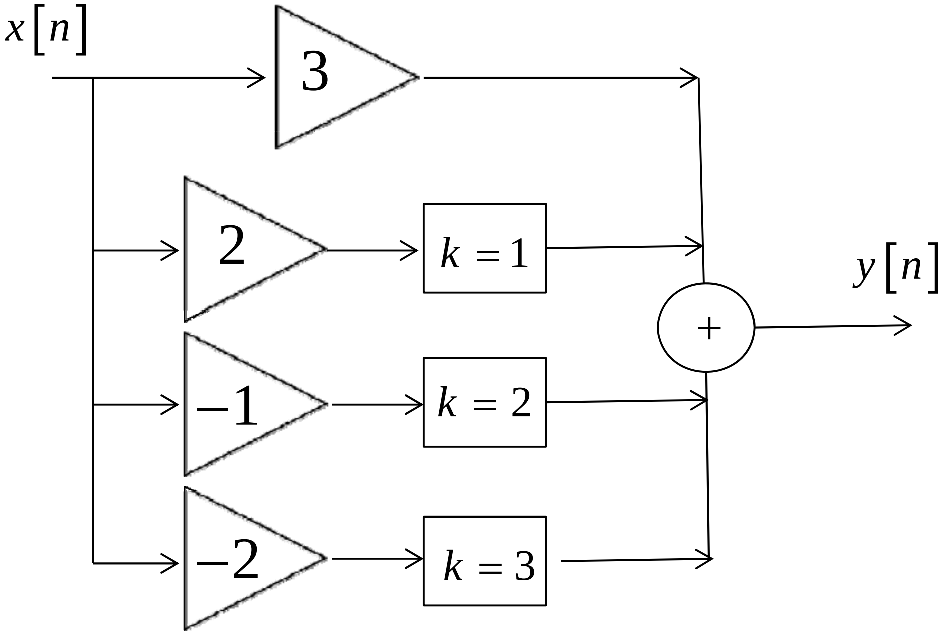
Obtenha uma expressão analítica que traduza o conteúdo do diagrama.



$$y[n]=b_0x[n]+b_1x[n-1]+b_2x[n-2]+b_3x[n-3]$$

Exemplo

Determine a equação às diferenças para o seguinte diagrama de blocos e escreva os coeficientes do filtro.



Eq. às diferenças $y[n]=3x[n]+2x[n-1]-x[n-2]-2x[n-3]$

Coef. filtro $b_0=3, b_1=2, b_2=-1, b_3=-2$

PROPRIEDADES DOS SISTEMAS

Linearidade

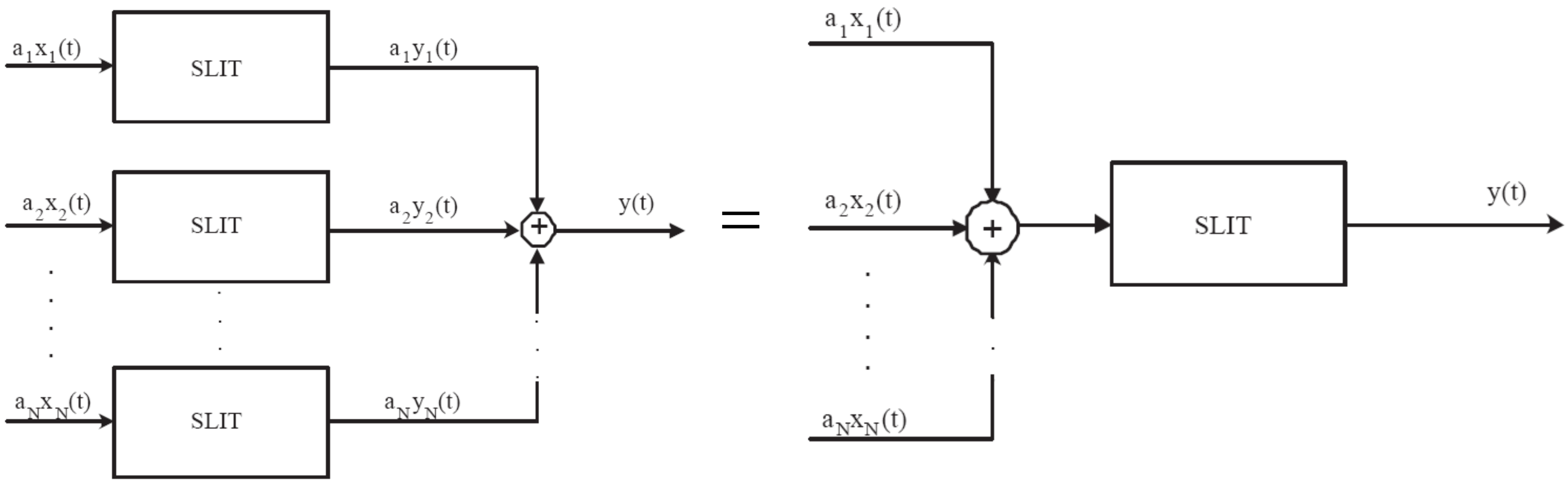
- Um sistema tal que:

$$x_1[n] \rightarrow y_1[n] \quad \text{e} \quad x_2[n] \rightarrow y_2[n]$$

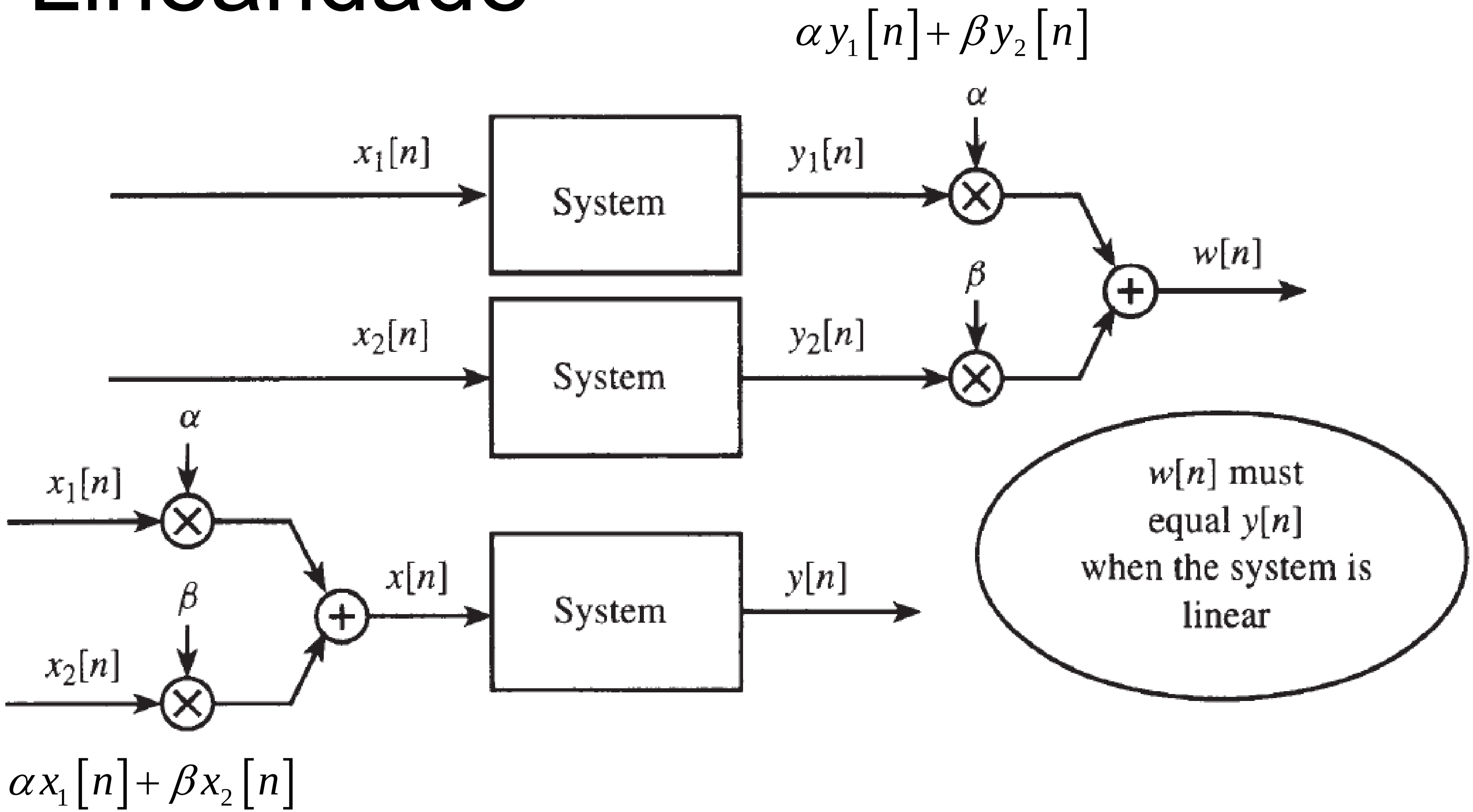
- é linear se:

$$x[n] = \alpha x_1[n] + \beta x_2[n] \rightarrow y[n] = \alpha y_1[n] + \beta y_2[n]$$

- Por outras palavras, um sistema é linear se verificar o principio da sobreposição



Linearidade



Linearidade

Exemplos:

1 - Verifique se o sistema dado é linear

$$y[n] = 2x[n] - x[n-1]$$

Res

$$x_1[n] \to 2x_1[n] - x_1[n-1] \qquad\qquad x_2[n] \to 2x_2[n] - x_2[n-1]$$

$$\begin{aligned} w[n] = \alpha x_1[n] + \beta x_2[n] &\to \alpha 2x_2[n] - \alpha x_2[n-1] + \beta 2x_2[n] - \beta x_2[n-1] = \\ &= \alpha\left(2x_2[n] - x_2[n-1]\right) + \beta\left(2x_2[n] - x_2[n-1]\right) \end{aligned}$$

$$y[n] \to \alpha\left(2x_1[n] - x_1[n-1]\right) + \beta\left(2x_2[n] - x_2[n-1]\right)$$

Como $y[n]$ e $w[n]$ são iguais o sistema é linear.

Linearidade

2 - Verifique se o sistema dado é linear

$$y[n] = x[n] + 1$$

$$y_1[n] = x_1[n] + 1 \qquad y_2[n] = x_2[n] + 1$$

$$w[n] = x_1[n] + 1 + x_2[n] + 1 = x_1[n] + x_2[n] + 2$$

$$y[n] = x_1[n] + x_2[n] + 1$$

Como $y[n]$ e $w[n]$ são diferentes o sistema não é linear.

3 - Verifique se o sistema dado é linear

$$y[n] = \left(x[n]\right)^2$$

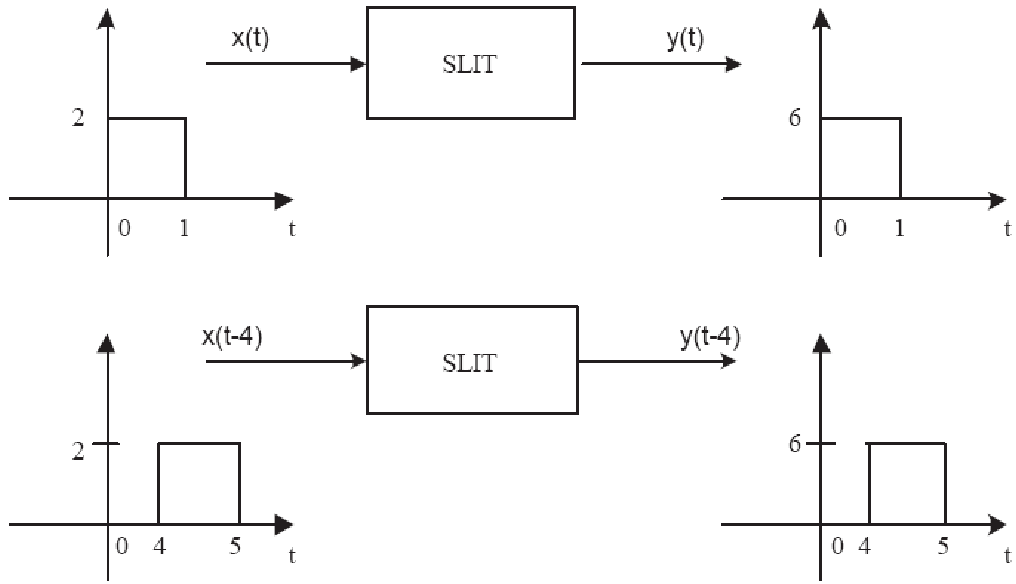
Invariância Temporal

Um sistema diz-se invariante no tempo se deslocando o sinal de entrada de n_0 (atraso ou avanço), o sinal de saída apresenta o mesmo deslocamento

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

Caso seja invariante a sua resposta a $x[n-n_0]$ é:

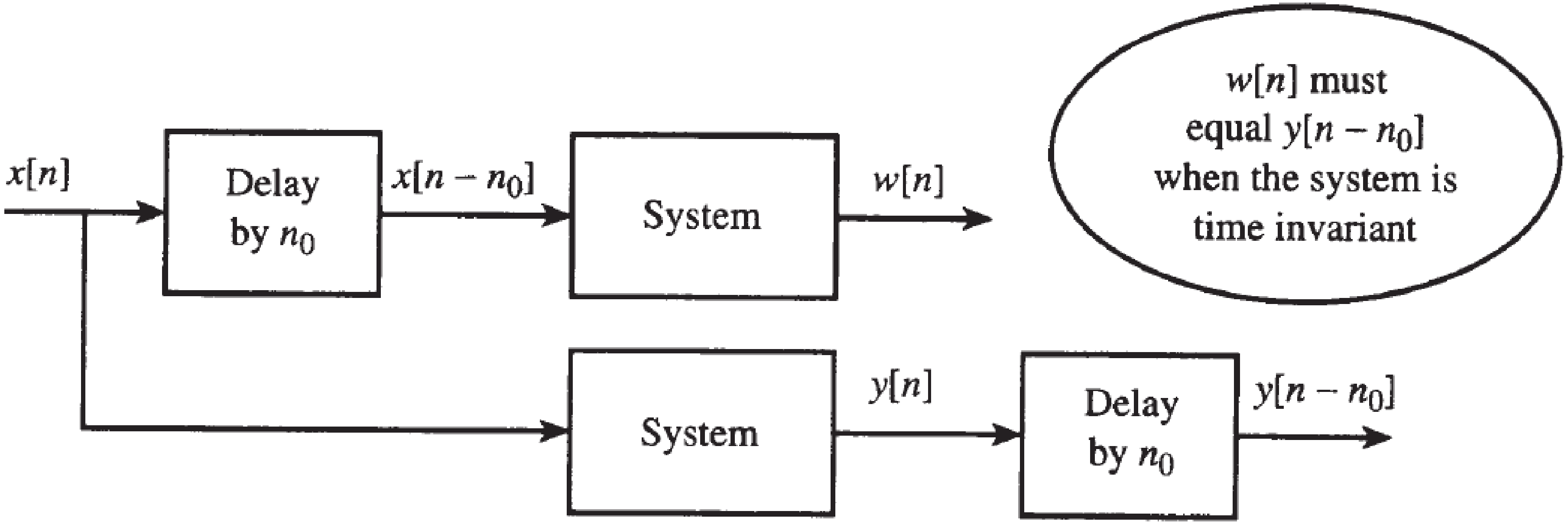
$$y[n-n_0] = S(x[n-n_0])$$



- Exemplos

$$y[n] = \sum_{k=-\infty}^{\infty} x[k] \qquad y[n] = (x[n])^2$$

Invariância Temporal



Exemplos:

1 --Vejamos a invariância no tempo de $y[n] = (x[n])^2$

Se tivermos como entrada o sinal de entrada deslocado, tem-se à saída: $w[n] = (x[n-n_0])^2$

Se tivermos à entrada o sinal de entrada sem deslocamento, então

$y[n] = (x[n])^2$. Aplicando o deslocamento tem-se:

$$y[n-n_0] = (x[n-n_0])^2$$

que é igual a $w[n]$ pelo que o sistema é invariante no tempo

Exemplos:

2 --Vejamos a invariância no tempo de $y[n] = x[-n]$

Se tivermos como entrada o sinal de entrada deslocado, tem-se à saída:

$$w[n] = x[(-n) - n_0] = x[-n - n_0]$$

Se tivermos à entrada o sinal de entrada sem deslocamento, então

$y[n] = x[-n]$. Aplicando o deslocamento tem-se:

$$y[n - n_0] = x[-(n - n_0)] = x[-n + n_0]$$

que é diferente de $w[n]$ pelo que o sistema não é invariante no tempo.

Exemplos:

3 --Vejamos a invariância no tempo de $y[n] = nx[n]$

Se tivermos como entrada o sinal de entrada deslocado, tem-se à saída:

$$w[n] = nx[n - n_0]$$

Se tivermos à entrada o sinal de entrada sem deslocamento, então

$y[n] = nx[n]$. Aplicando o deslocamento tem-se:

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

que é diferente de $w[n]$ pelo que o sistema não é invariante no tempo

Causalidade

- Um sistema diz-se um causal se a saída em qualquer instante n_0 só depender dos valores da entrada nesse instante ou nos instantes anteriores ($n \le n_0$).

  Sistema não antecipativo

- Exemplos

causal

$$y[n] = x[n] + 2x[n-2]$$

não causal

$$y[n] = x[n+1]$$

Estabilidade

- Um sistema diz-se um sistema estável se para qualquer sinal de entrada limitado em amplitude,

$$x[n] \leq M \quad , \quad \forall n$$

o sinal de saída for também limitado em amplitude,

$$y[n] \leq N \quad , \quad \forall n$$

Um sistema que não é estável diz-se um sistema instável.

- Exemplos

$$y(n) = x(n)^{100}$$

Estável

$$y(n) = nx(n)$$

Instável

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

A estabilidade depende da forma de x(n)

Invertibilidade

- Um sistema diz-se um sistema invertível se entradas distintas produzirem saídas distintas. O mesmo é dizer, conhecida que seja a saída do sistema é possível saber qual a entrada que lhe deu origem.

- Exemplos

$$y(n) = cos(x(n)) \qquad\qquad y(n) = 3x(n) + 1$$

Sistemas FIR

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

- Lineares
- Invariantes no Tempo

SISTEMAS LINEARES E INVARIANTES NO TEMPO (SLITS)

Sistemas Lineares e Invariantes no Tempo (SLITs)

- h[n] : resposta impulsional (ou impulsiva)

$$x[n]=\delta[n] \longrightarrow \boxed{T\{.\}} \longrightarrow y[n]=h[n]$$

Sistema é Invariante
$x[n]=\delta[n-k]$ $\longrightarrow$ $y[n]=h[n-k]$

Sistema é Linear e Invariante:

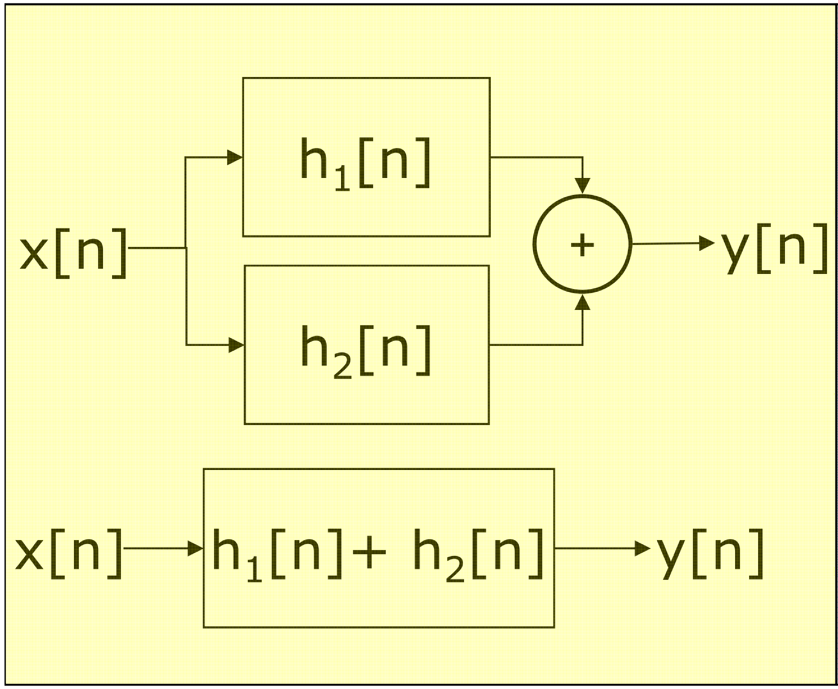
$$x[n]=\sum_{k=-\infty}^{\infty} x[k]\delta[n-k] \longrightarrow y[n]=T\left\{\sum_{k=-\infty}^{\infty} x[k]\delta[n-k]\right\}=$$

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

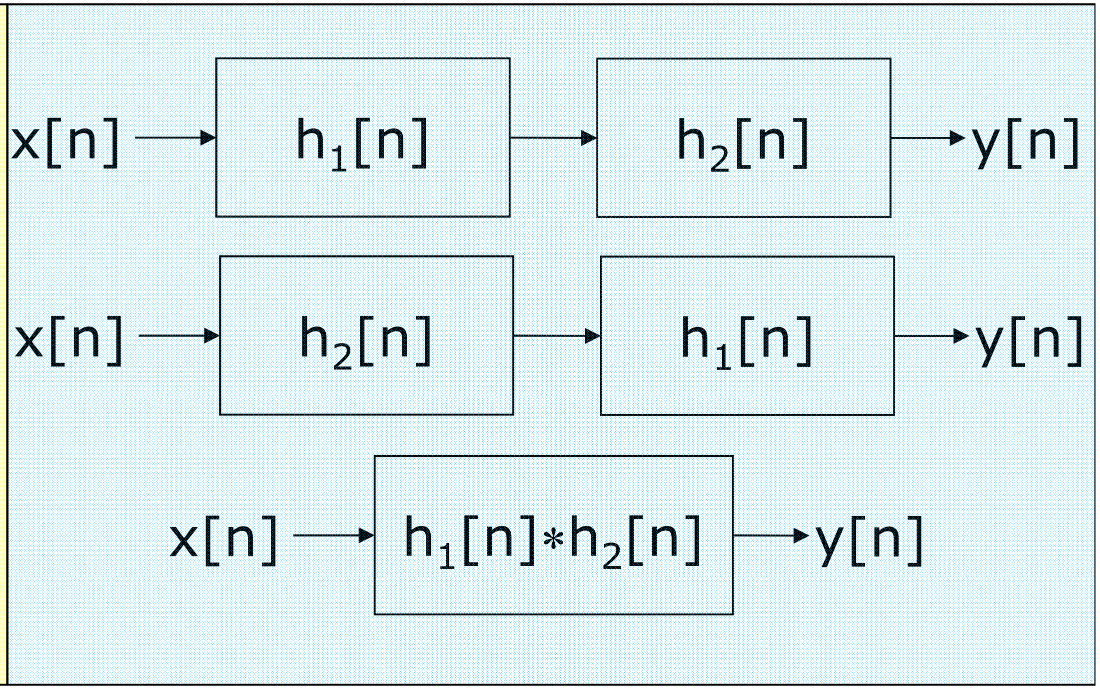
ASSOCIAÇÃO ENTRE SISTEMAS

Associação entre Sistemas

- Paralelo



- Série



Exemplo:

Considere o SLIT formado por dois sistema SLIT em série (em cascata) definidos por:

$$h_1[n] = \begin{cases} 1 & 0 \le n \le 3 \\ 0 & \text{cc} \end{cases} \qquad h_2[n] = \begin{cases} 1 & 0 \le n \le 2 \\ 0 & \text{cc} \end{cases}$$

Determine a resposta impulsional do sistema resultante.

Res.

$$h[n] = h_1[n] * h_2[n]$$

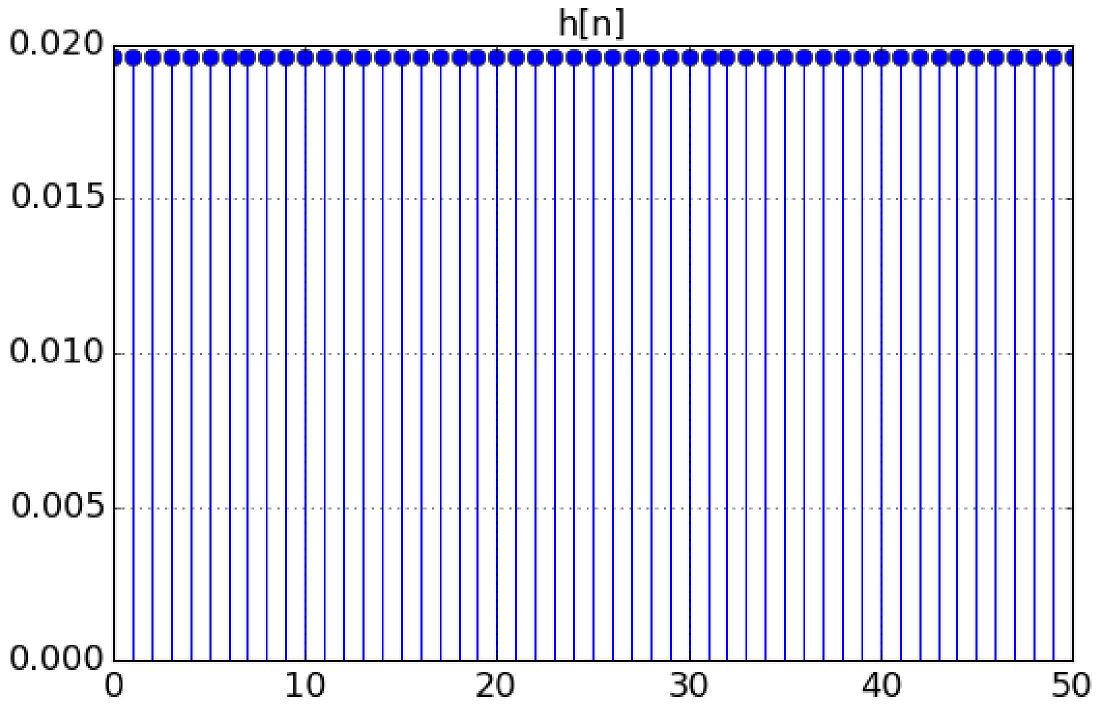
n	0	1	2	3	4	5
h_2[n]	1	1	1	1		
h_1[n]	1	1	1			
	1	1	1	1		
		1	1	1	1	
			1	1	1	1
h_[n]	1	2	3	3	2	1

$$h[n] = \sum_{k=0}^{5} b_k \delta[n-k] = \delta[n] + 2\delta[n-1] + 3\delta[n-2] + 3\delta[n-3] + 2\delta[n-4] + \delta[n-5]$$

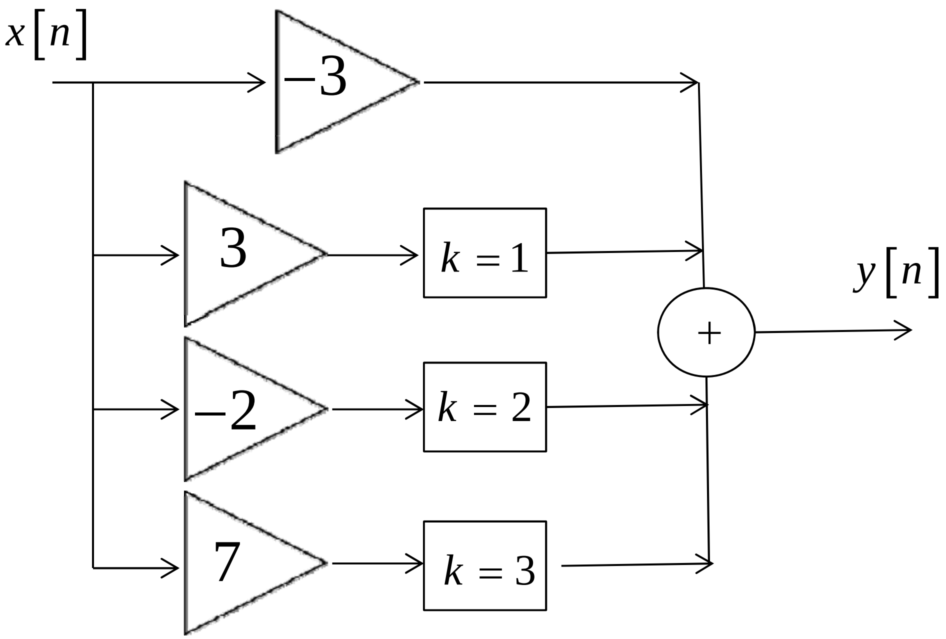
Exercícios

1 – Determine a resposta impulsional $h[n]$ de um sistema causal de média móvel de 51 pontos.

$$y[n] = \sum_{k=0}^{51} \frac{1}{51} x[n-k] \qquad\qquad h[n] = \sum_{k=0}^{51} \frac{1}{51} \delta[n-k]$$



2 - O seguinte diagrama de blocos define um SLIT.
a) Determine a equação às diferenças para este sistema



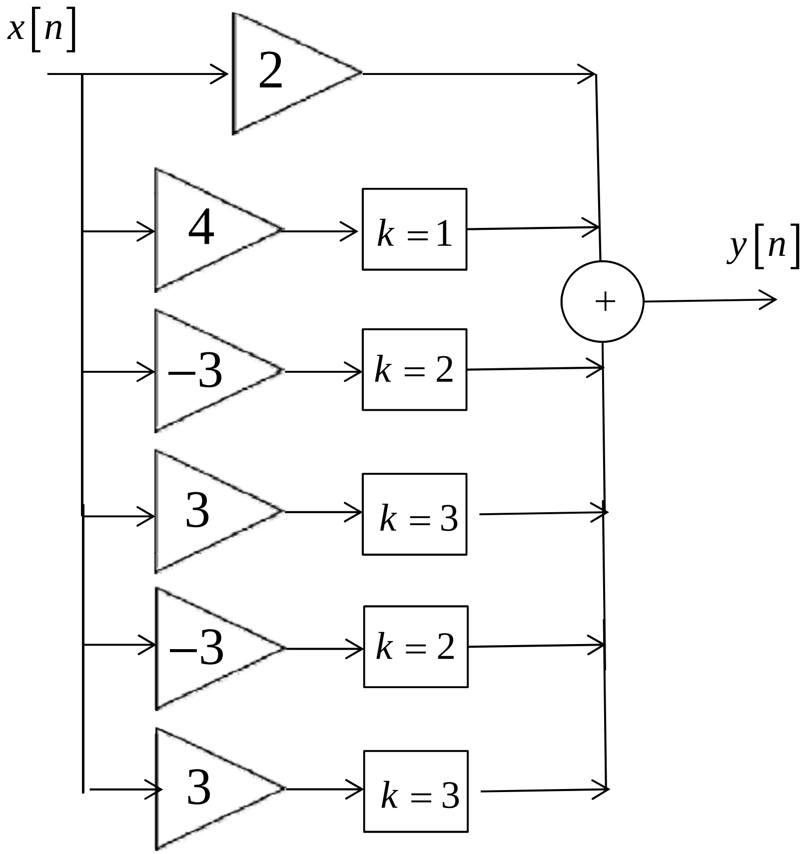
Eq. às diferenças $y[n] = -3x[n] + 3x[n-1] - 2x[n-2] + 7x[n-3]$

Coef. filtro $b_0 = -3, b_1 = 3, b_2 = -2, b_3 = 7$

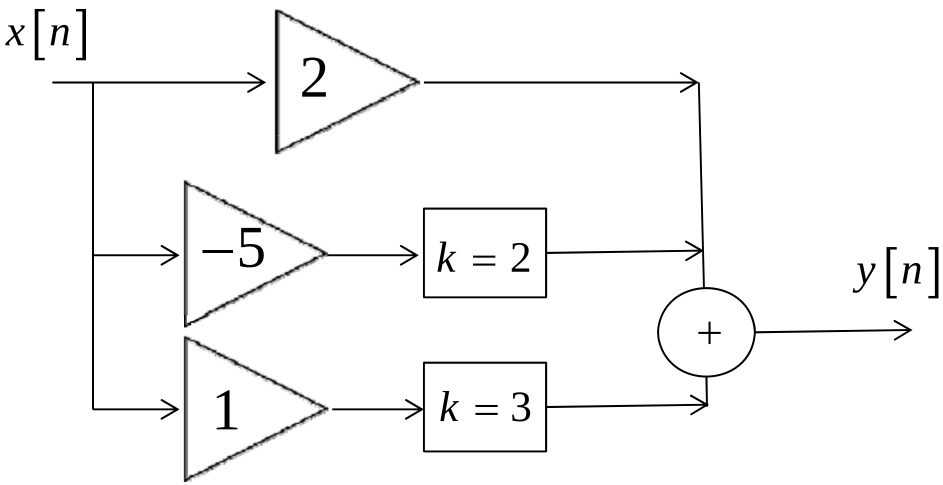
b) Desenhe o diagrama de blocos que representa o sistema cuja equação às diferenças é:

$$y[n]=2x[n]+4x[n-1]-3x[n-2]+3x[n-3]-4x[n-4]-2x[n-5]$$

Res



3 - O seguinte diagrama de blocos define um SLIT.
Determine a equação às diferenças para este sistema



Eq. às diferenças $$y[n] = 2x[n] - 5x[n-2] + x[n-3]$$

Coef. filtro $$b_0 = 2, b_1 = 0, b_2 = -5, b_3 = 1$$

4 – Um filtro FIR é descrito pela equação às diferenças:

$$y[n] = 3x[n] + 2x[n-3] - 3x[n-5]$$

a) Determine e represente graficamente a resposta impulsional.

b) Seja o sinal de entrada:

$$x[n] = 3e^{j\left(0.4\pi n - \pi/2\right)} \qquad \forall n$$

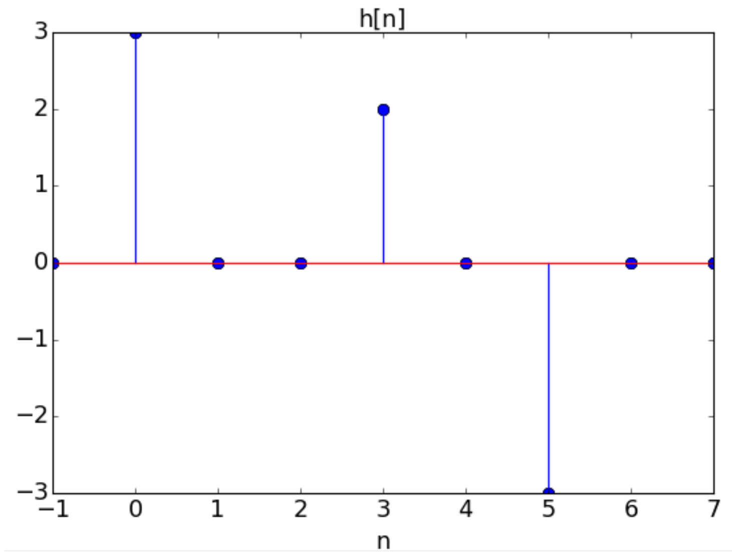
Nestas condições o sinal de saída é da forma:

$$y[n] = Ae^{j\left(2\pi \hat{f}_0 n + \phi\right)}$$

Determine os valores de $A, \phi, \hat{f}_0$

Res:

a) $h[n] = 3\delta[n] + 2\delta[n-3] - 3\delta[n-5]$



b) Uma das maneiras de resolver é introduzir o sinal de entrada na eq. às diferenças

$$y[n] = 3\times 3e^{j\left(0.4\pi n-\pi/2\right)} + 2\times 3e^{j\left(0.4\pi(n-3)-\pi/2\right)} - 3\times 3e^{j\left(0.4\pi(n-5)-\pi/2\right)} =$$

$$= 9e^{-j\pi/2}e^{j0.4\pi n} + 6e^{-j\pi/2}e^{-j1.2\pi}e^{j0.4\pi n} - 9e^{-j\pi/2}e^{j0.4\pi n} =$$

$$= 3e^{-j\pi/2}e^{j0.4\pi n}\left(\cancel{3} + 2e^{-j1.2\pi} - \cancel{3}\right) = 6e^{-j1.7\pi}e^{j0.4\pi n} =$$

$$= 6e^{j\left(0.4\pi n-1.7\pi\right)}$$

$$A = 6 \ \ ,\phi = -1.7\pi, \quad \widehat{f}_0 = 0.2 \ \text{ ou } \ \widehat{\omega}_0 = 0.4\pi$$