Controle de Sistemas I

Sistemas Lineares Invariantes no Tempo

Renato Dourado Maia

Faculdade de Ciência e Tecnologia de Montes Claros

Fundação Educacional Montes Claros

Introdução

$h(t),\ h[n]$ ⇒ Resposta do sistema quando a entrada é um impulso unitário, $\delta(t),\ \delta[n]$.

A Resposta ao Impulso caracteriza um sistema LTI: dada uma entrada x, pode-se, conhecendo-se h, determinar-se y. Esse método é denominado convolução.

Sinais Discretos e Soma de Impulsos

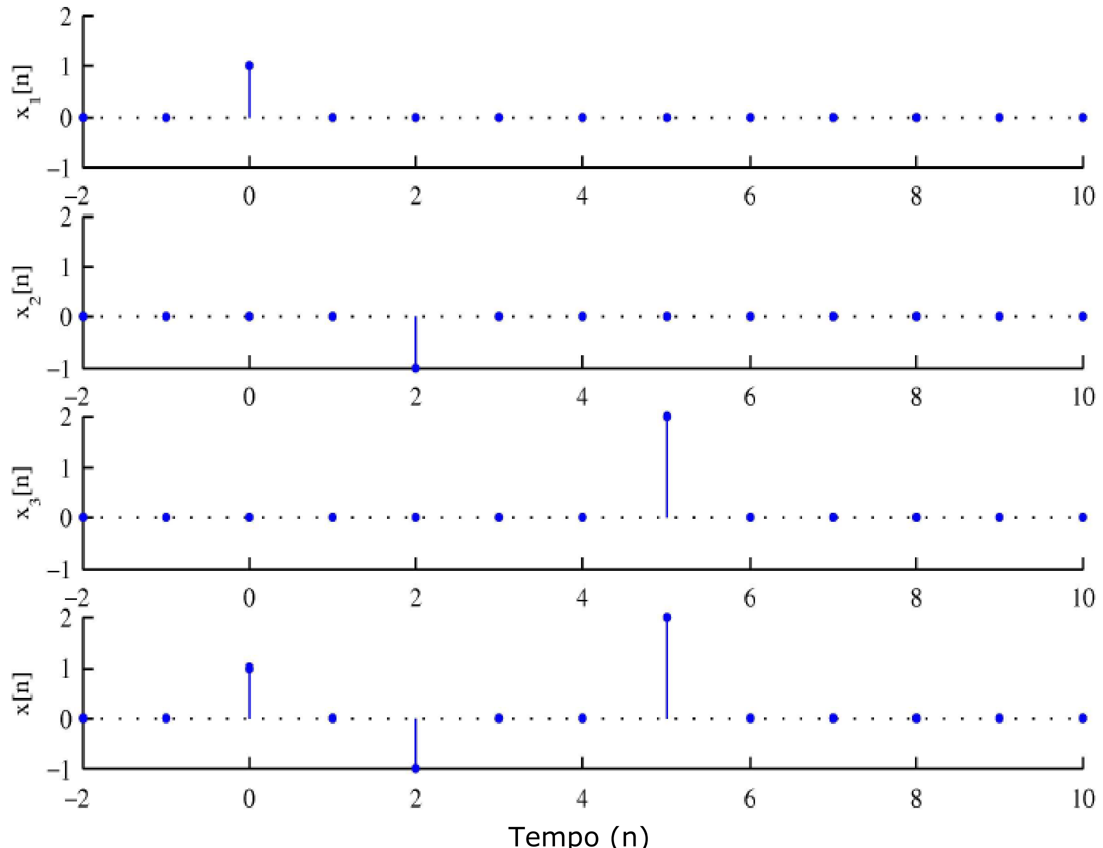
- Seja o seguinte sinal: $x[n] = \begin{cases} 1, & n = 0 \\ -1, & n = 2 \\ 2, & n = 5 \\ 0, & caso\ contrário \end{cases}$

O sinal pode ser escrito como uma soma de impulsos?

SIM!!! ⟹ $x[n] = 1\delta[n] - 1\delta[n-2] + 2\delta[n-5] = x_1[n] + x_2[n] + x_3[n]$

Sinais Discretos e Soma de Impulsos

$$x[n] = 1\delta[n] - 1\delta[n-2] + 2\delta[n-5] = x_1[n] + x_2[n] + x_3[n]$$



Sinais Discretos e Soma de Impulsos

- Todo sinal discreto limitado pode ser escrito como uma soma ponderada de impulsos unitários:

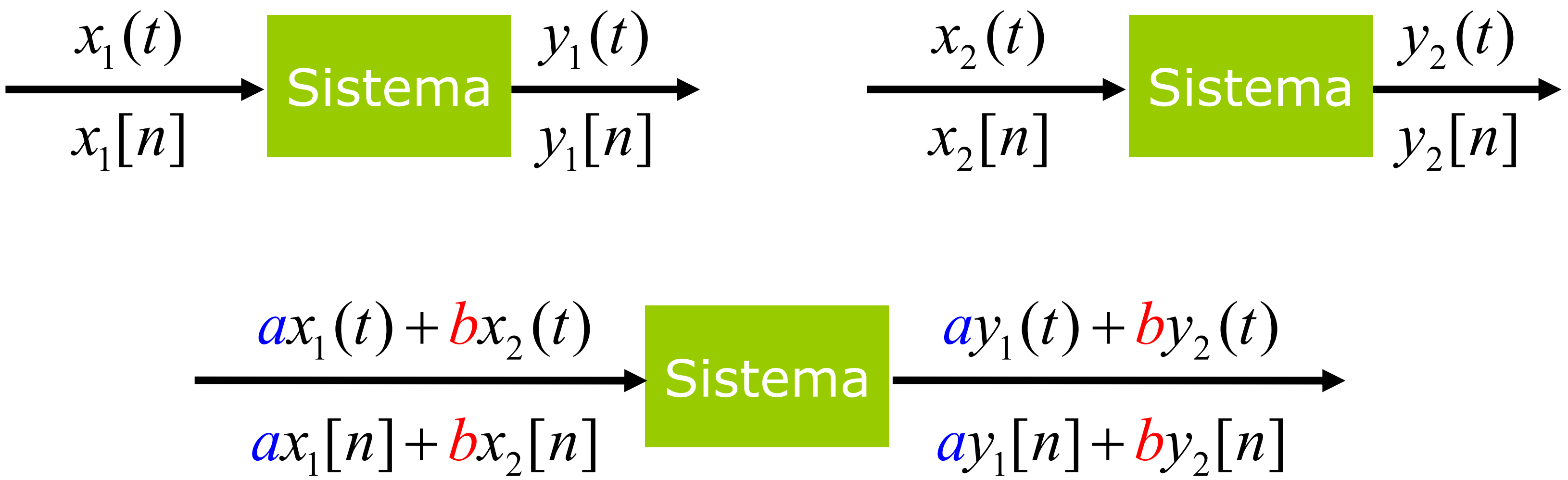
Impulso Deslocado

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

Peso

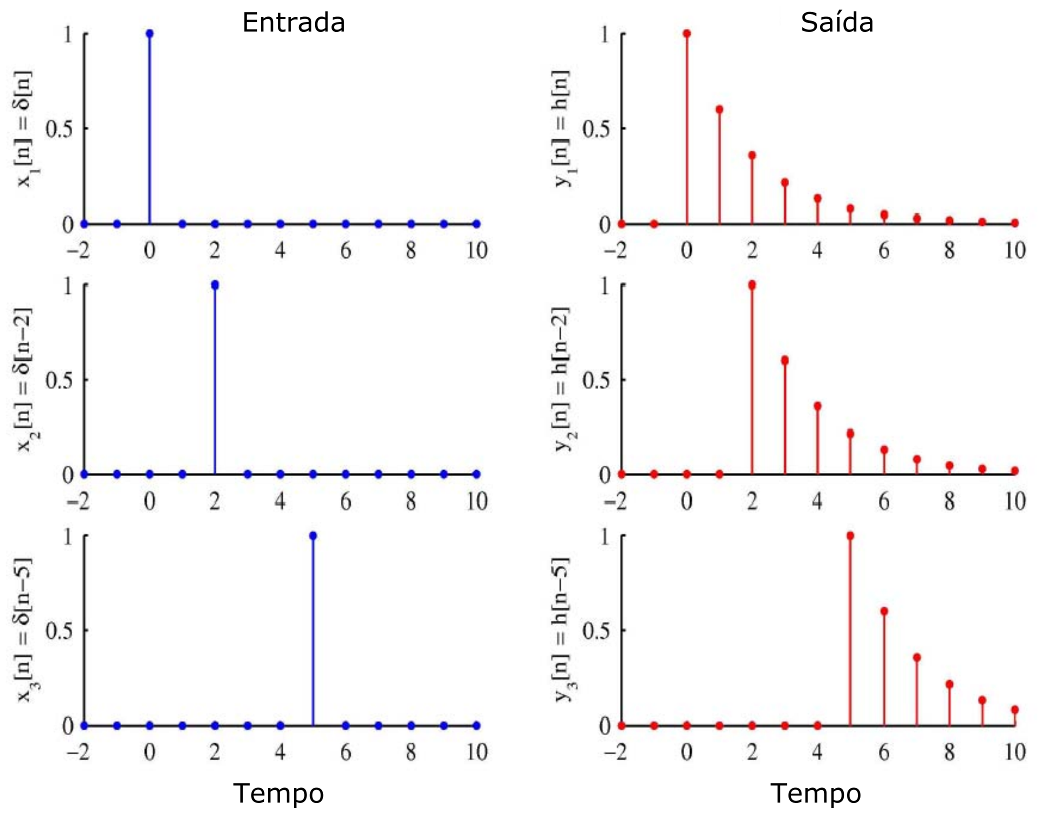
Lembrando...

- Linearidade:



Lembrando...

- Invariância no Tempo:



Somatório de Convolução

- Retomando o exemplo:

$$x[n] = 1\delta[n] - 1\delta[n-2] + 2\delta[n-5] = x_1[n] + x_2[n] + x_3[n]$$

Considerando a LINEARIDADE e a INVARIÂNCIA NO TEMPO:

$$y[n] = y_1[n] + y_2[n] + y_3[n]$$

$$\left.\begin{aligned} x_1[n] = \delta[n] &\rightarrow y_1[n] = 1h[n] \\ x_2[n] = -\delta[n-2] &\rightarrow y_2[n] = -1h[n-2] \\ x_2[n] = 2\delta[n-5] &\rightarrow y_3[n] = 2h[n-5] \end{aligned}\right\} y[n] = 1h[n] - 1h[n-2] + 2h[n-5]$$

A saída é uma soma ponderada das saídas devidas a cada entrada, ou seja, um somatório de respostas ao impulso deslocadas e ponderadas!!!

Somatório de Convolução

- Generalizando para qualquer sinal discreto limitado:

$$x[n]=\sum_{k=-\infty}^{+\infty} x[k]\delta[n-k] \longrightarrow$$ Sinal Discreto Limitado

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

→ Somatório de Convolução

UM SISTEMA LTI É COMPLETAMENTE CARACTERIZADO POR SUA RESPOSTA AO IMPULSO UNITÁRIO!!!

Somatório de Convolução - Resumo

$$x[n] \longrightarrow \boxed{h[n]} \longrightarrow y[n]$$

$$\delta[n] \longrightarrow \boxed{h[n]} \longrightarrow h[n]$$ → Definição de $h[n]$

$$\delta[n-k] \longrightarrow \boxed{h[n]} \longrightarrow h[n-k]$$ → Invariância no Tempo

$$x[k]\delta[n-k] \longrightarrow \boxed{h[n]} \longrightarrow x[k]h[n-k]$$ → Linearidade

$$\sum_{k=-\infty}^{+\infty} x[k]h[n-k] \longrightarrow \boxed{h[n]} \longrightarrow \sum_{k=-\infty}^{+\infty} x[k]h[n-k]$$ → Linearidade

$$x[n] \longrightarrow \boxed{h[n]} \longrightarrow \sum_{k=-\infty}^{+\infty} x[k]h[n-k]$$ → Definição de $\delta[n]$

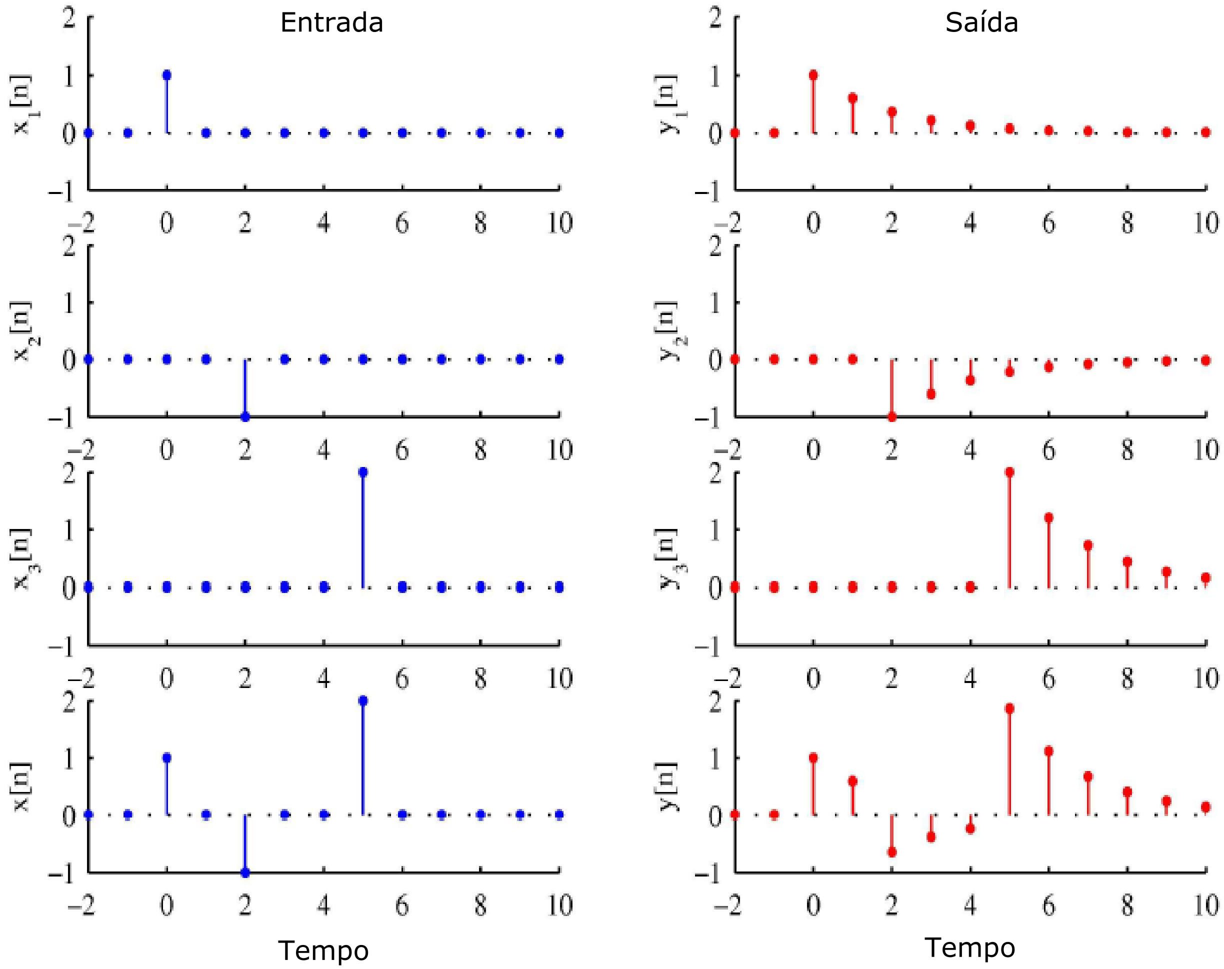
Somatório de Convolução

Exemplo

$$x[n] = \begin{cases} 1, & n = 0 \\ -1, & n = 2 \\ 2, & n = 5 \\ 0, & caso\ contrário \end{cases}$$

$$h[n] = a^n u[n] = \begin{cases} a^n, & n \geq 0 \\ 0, & n < 0 \end{cases}, \quad onde\ a = 0.6$$

Somatório de Convolução



Somatório de Convolução

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

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

O que acontece para cada valor de n, se imaginarmos os sinais em função da variável k?

Vejamos uma animação em Java para compreendermos a segunda interpretação do somatório de convolução: rebate, desloca, multiplica e soma...

Somatório de Convolução

Exemplo – O Mesmo ☺

$$x[n] = \begin{cases} 1, & n = 0 \\ -1, & n = 2 \\ 2, & n = 5 \\ 0, & \textit{caso contrário} \end{cases}$$

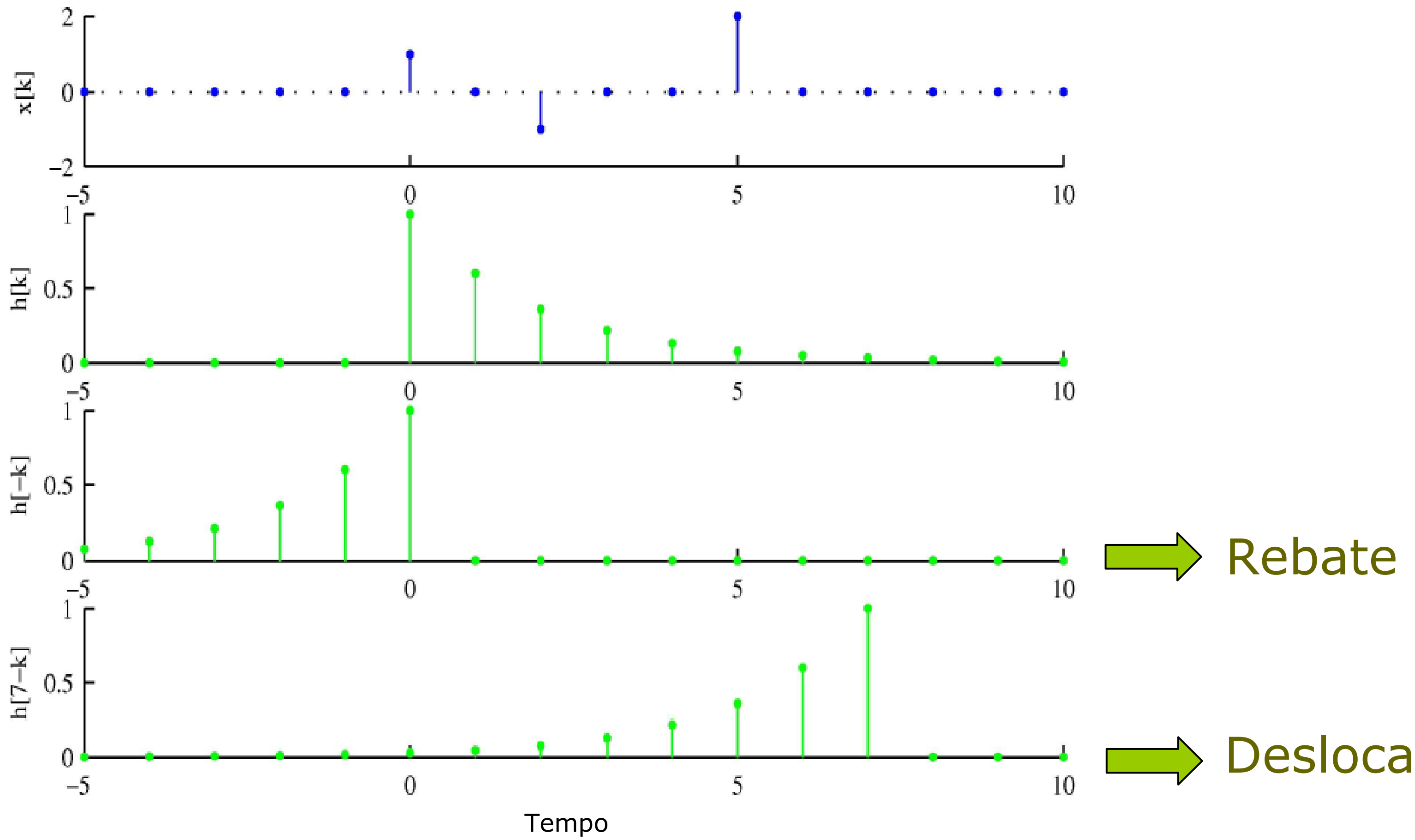
$$h[n] = a^n u[n] = \begin{cases} a^n, & n \geq 0 \\ 0, & n < 0 \end{cases}, \quad \textit{onde } a = 0.6$$

Somatório de Convolução

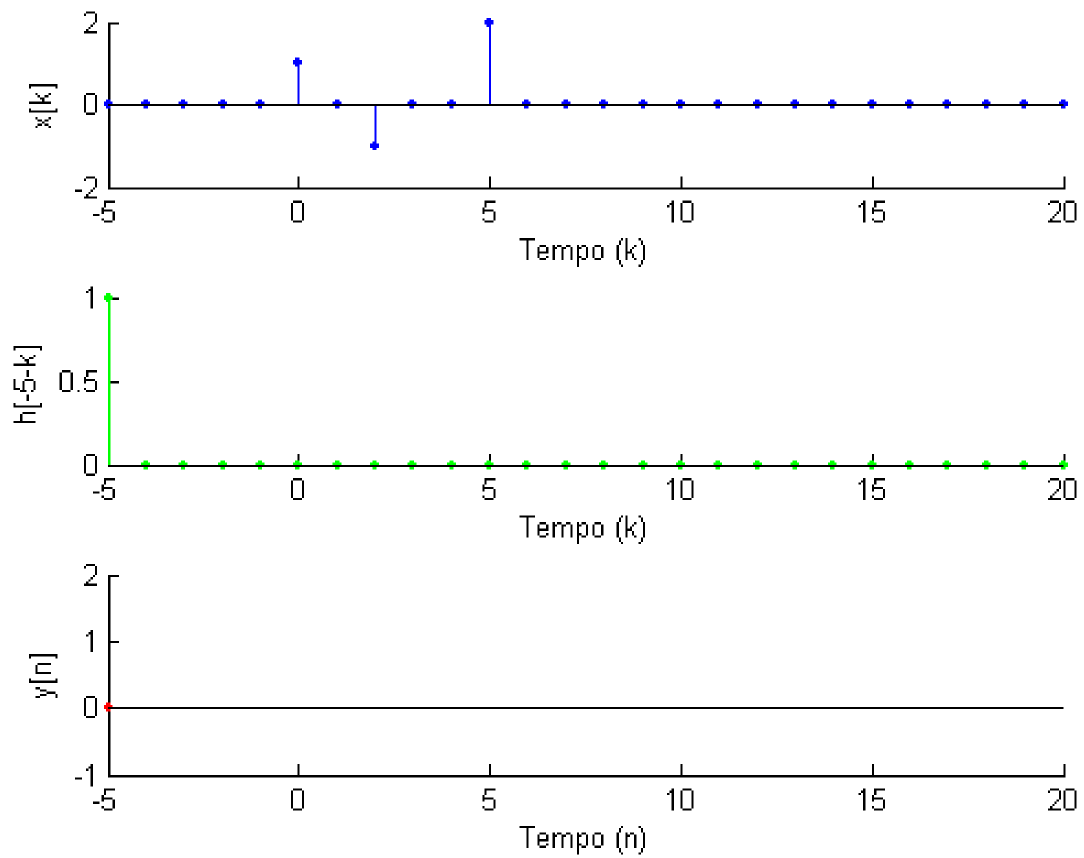
- Vamos observar graficamente a resolução do exemplo utilizando a interpretação rebate, desloca, multiplica e soma.

→ Script em Matlab: M_6_SistemasLTIProg1.m

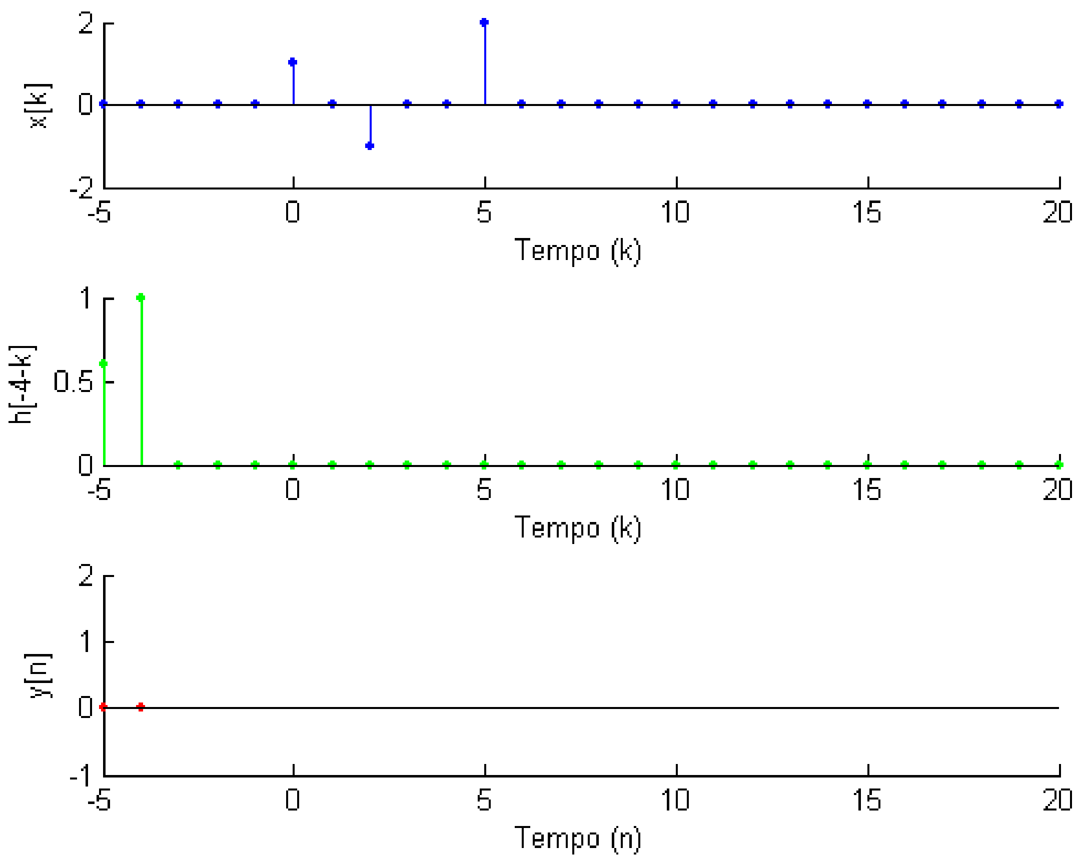
Somatório de Convolução



n = -5

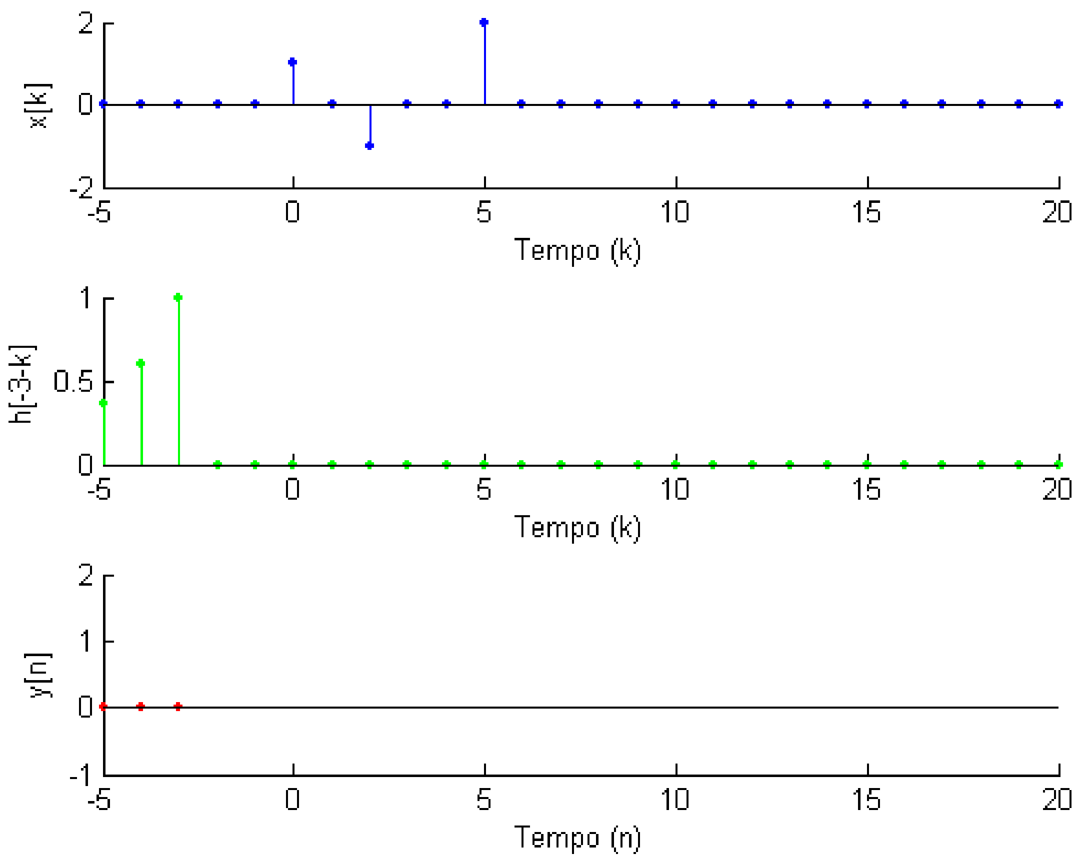


n = -4

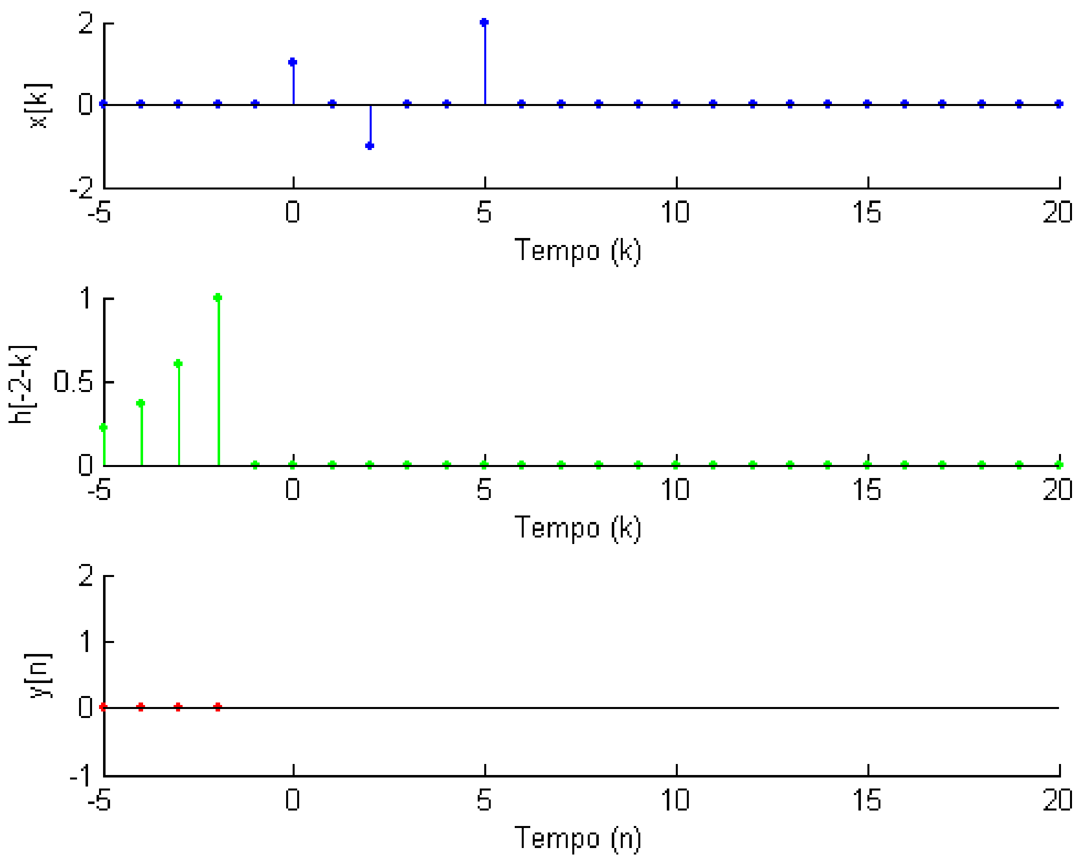


2
0
-2
x[k]
-5
0
5
10
15
20
Tempo (k)
1
0.5
0
h[-4-k]
-5
0
5
10
15
20
Tempo (k)
2
1
0
-1
y[n]
-5
0
5
10
15
20
Tempo (n)

n = -3

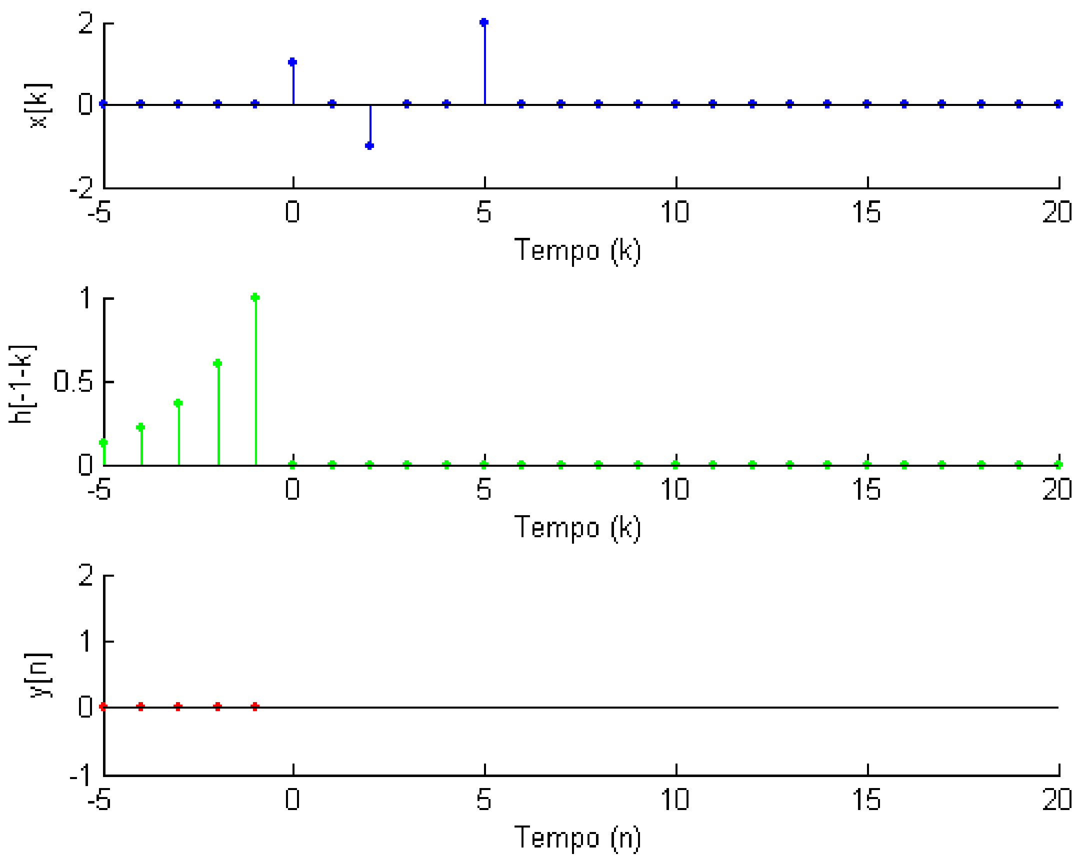


n = -2



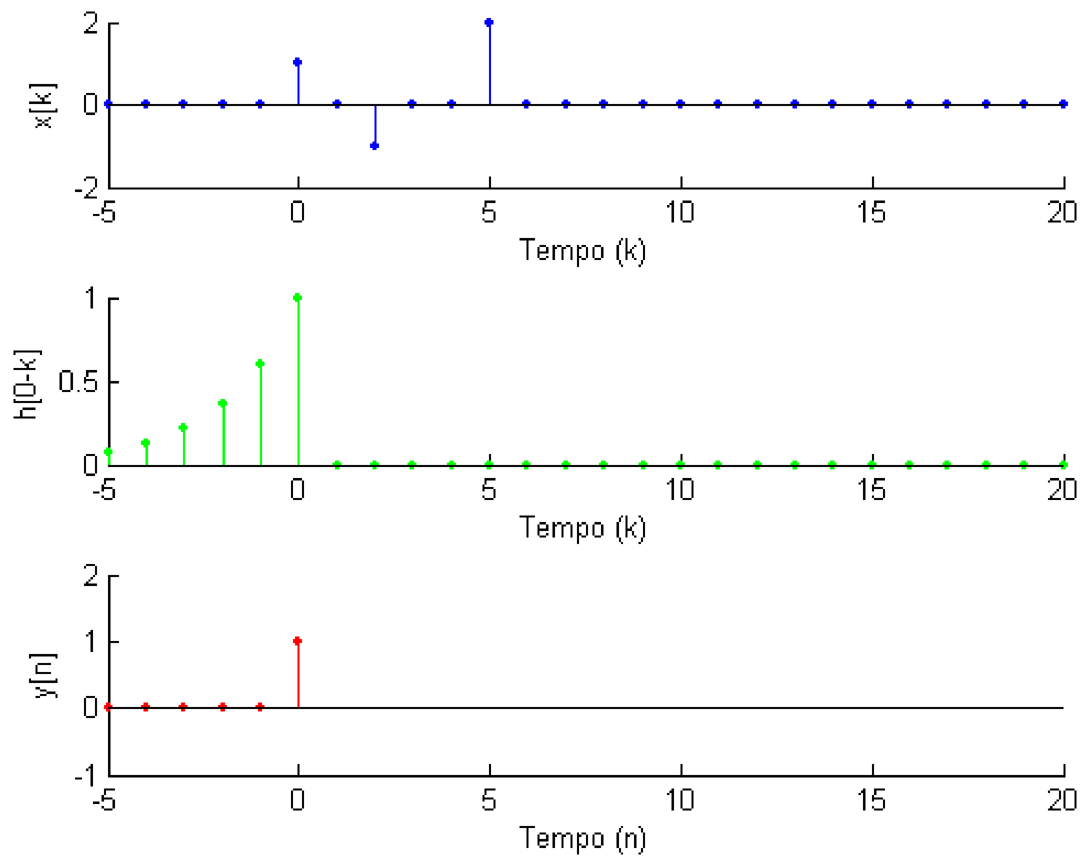
2
0
-2
x[k]
-5
0
5
10
15
20
Tempo (k)
1
0.5
0
h[-2-k]
-5
0
5
10
15
20
Tempo (k)
2
1
0
-1
y[n]
-5
0
5
10
15
20
Tempo (n)

n = -1



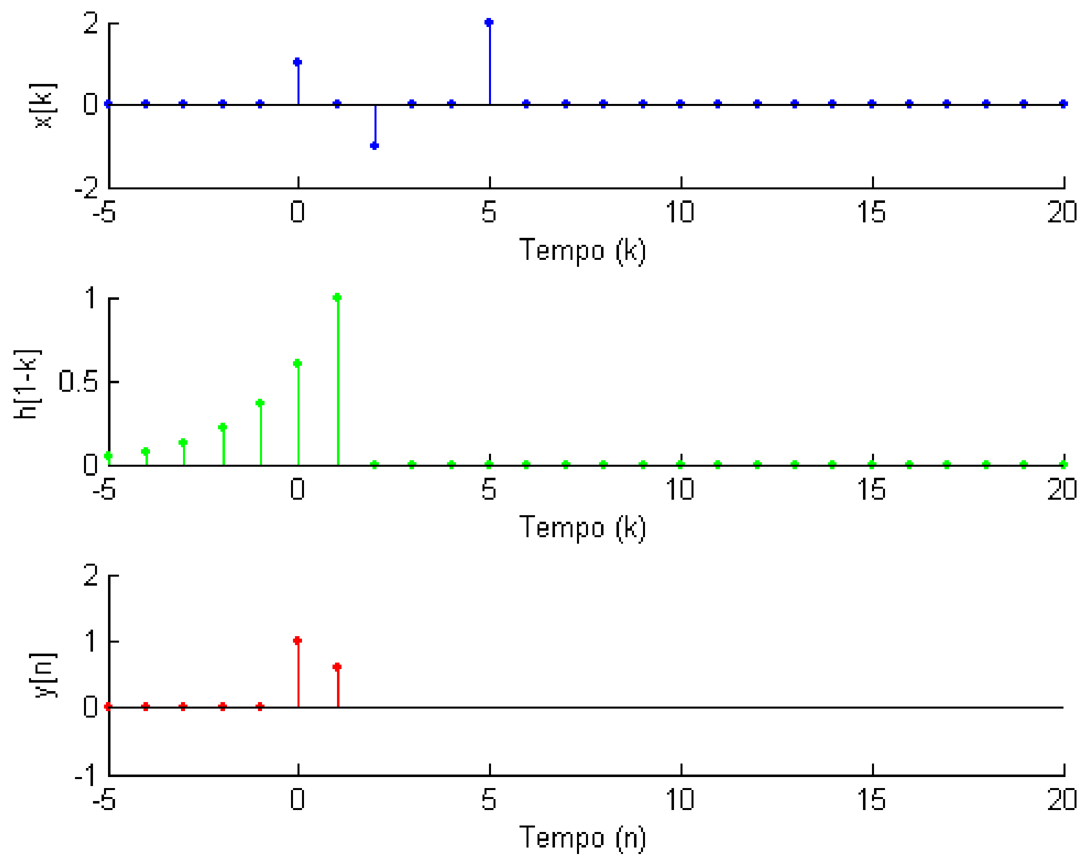
x[k]
2
0
-2
-5
0
5
10
15
20
Tempo (k)
h[-1-k]
1
0.5
0
-5
0
5
10
15
20
Tempo (k)
y[n]
2
1
0
-1
-5
0
5
10
15
20
Tempo (n)

n = 0

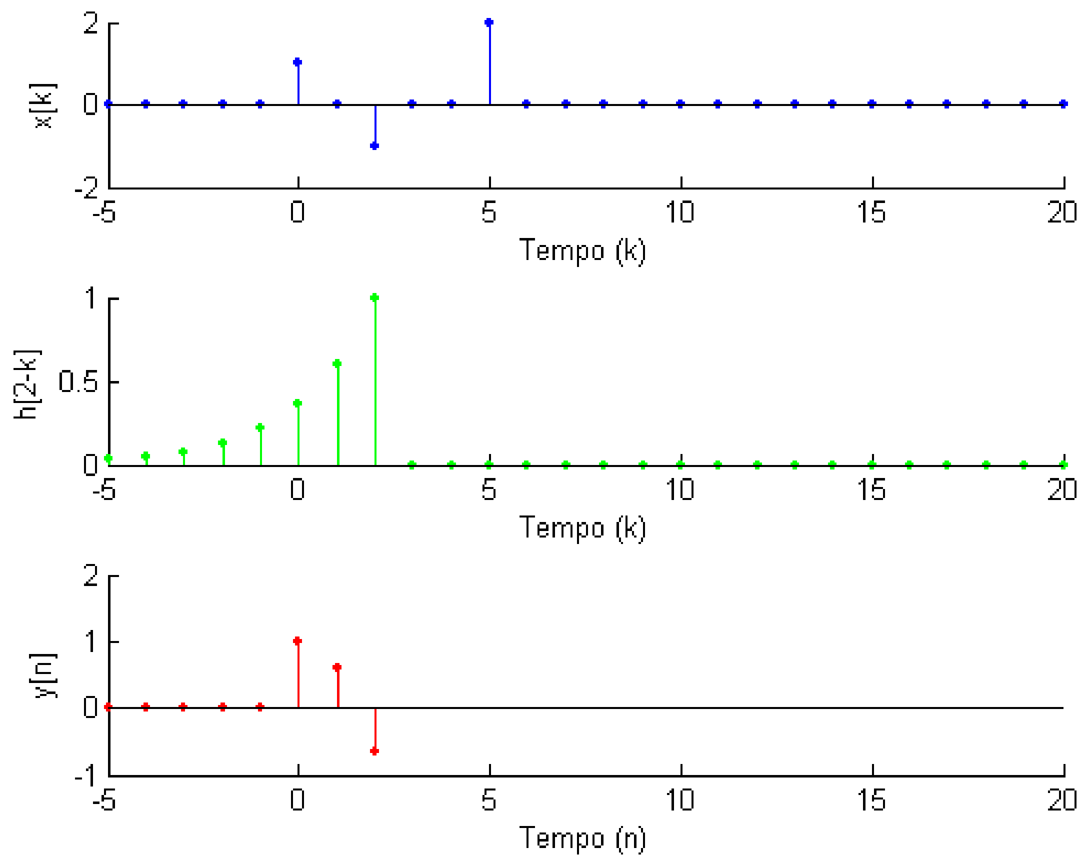


2
0
-2
x[k]
-5
0
5
10
15
20
Tempo (k)
1
0.5
0
h[0-k]
-5
0
5
10
15
20
Tempo (k)
2
1
0
-1
y[n]
-5
0
5
10
15
20
Tempo (n)

n = 1

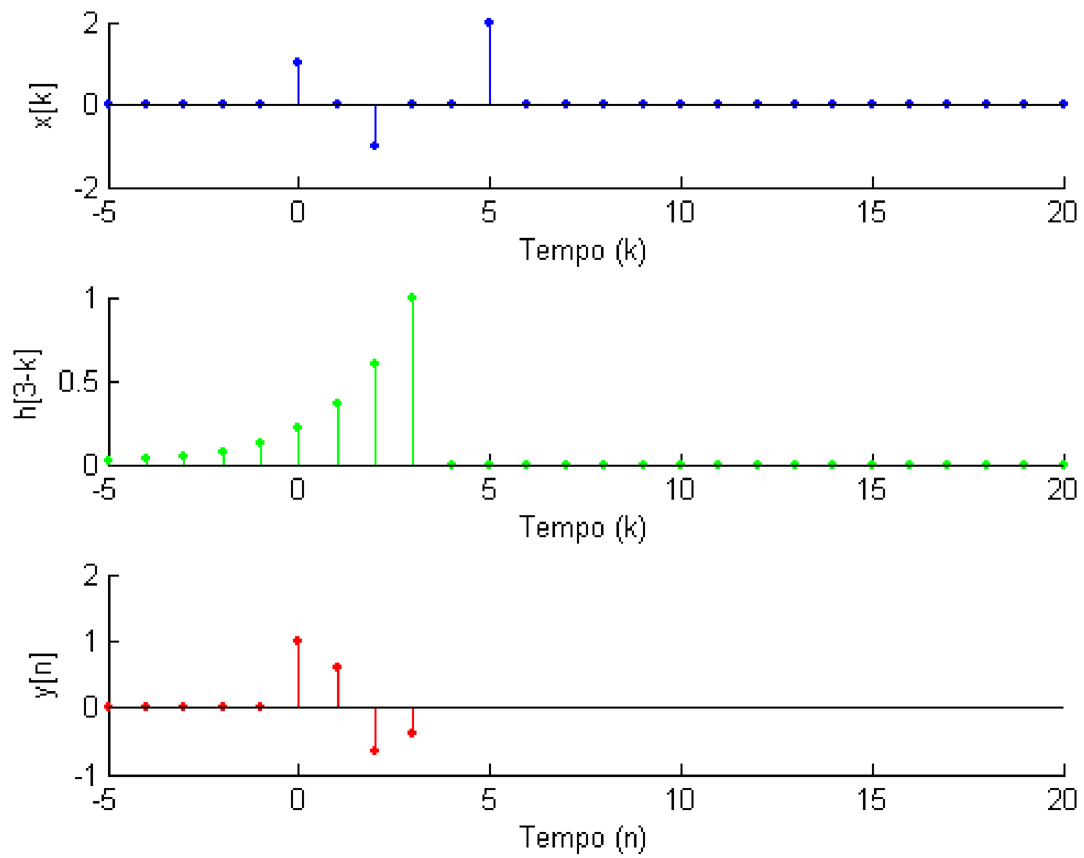


n = 2



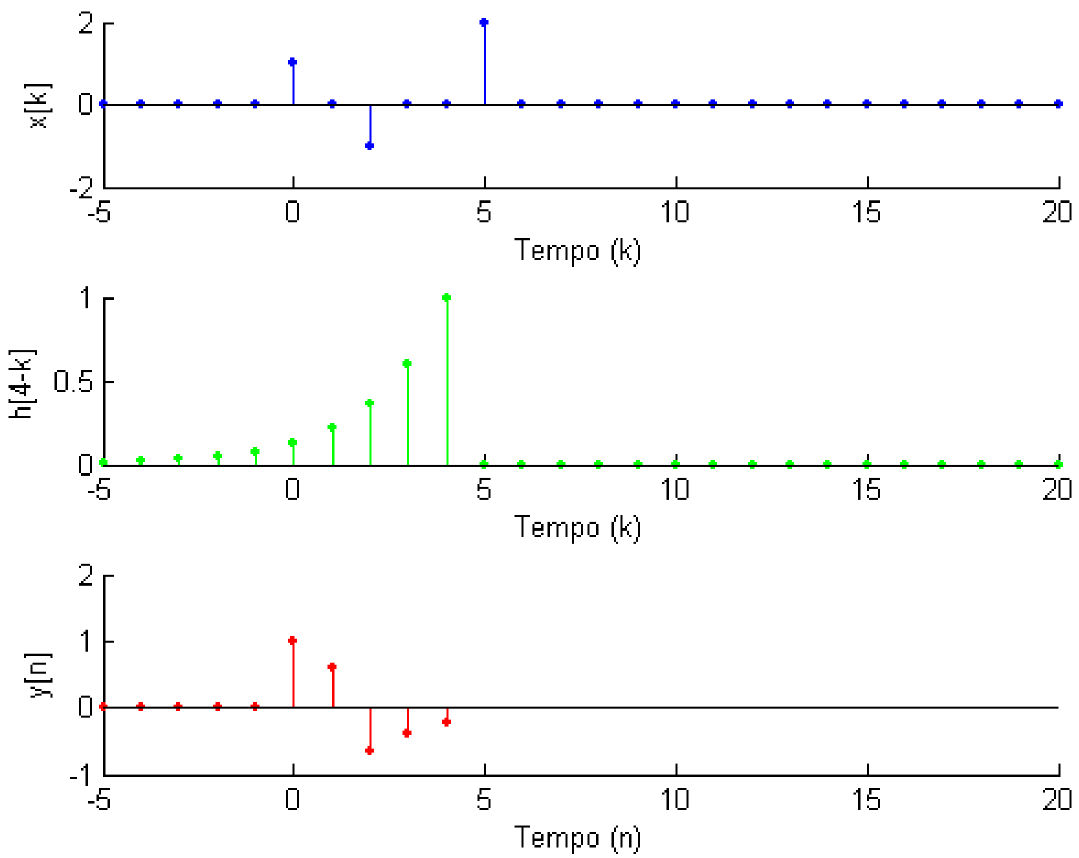
x[k]
2
0
-2
-5
0
5
10
15
20
Tempo (k)
h[2-k]
1
0.5
0
-5
0
5
10
15
20
Tempo (k)
y[n]
2
1
0
-1
-5
0
5
10
15
20
Tempo (n)

n = 3



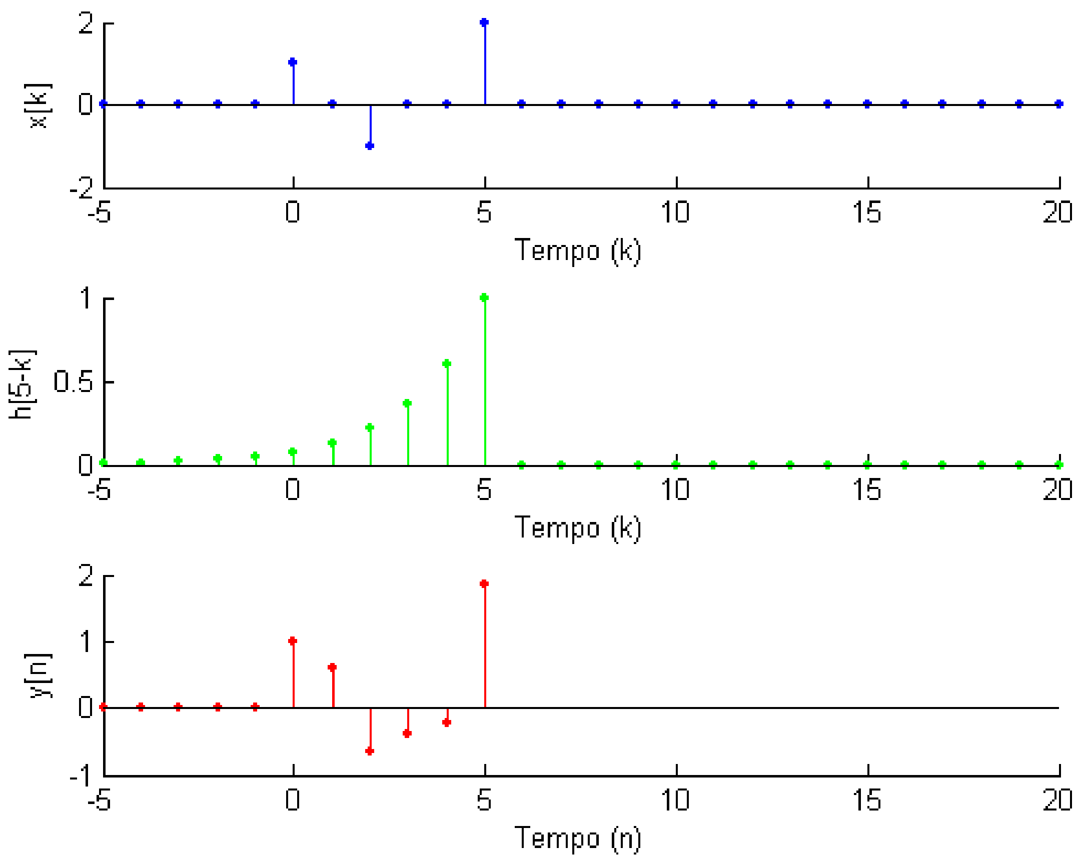
x[k]
2
0
-2
-5
0
5
10
15
20
Tempo (k)
h[3-k]
1
0.5
0
-5
0
5
10
15
20
Tempo (k)
y[n]
2
1
0
-1
-5
0
5
10
15
20
Tempo (n)

n = 4



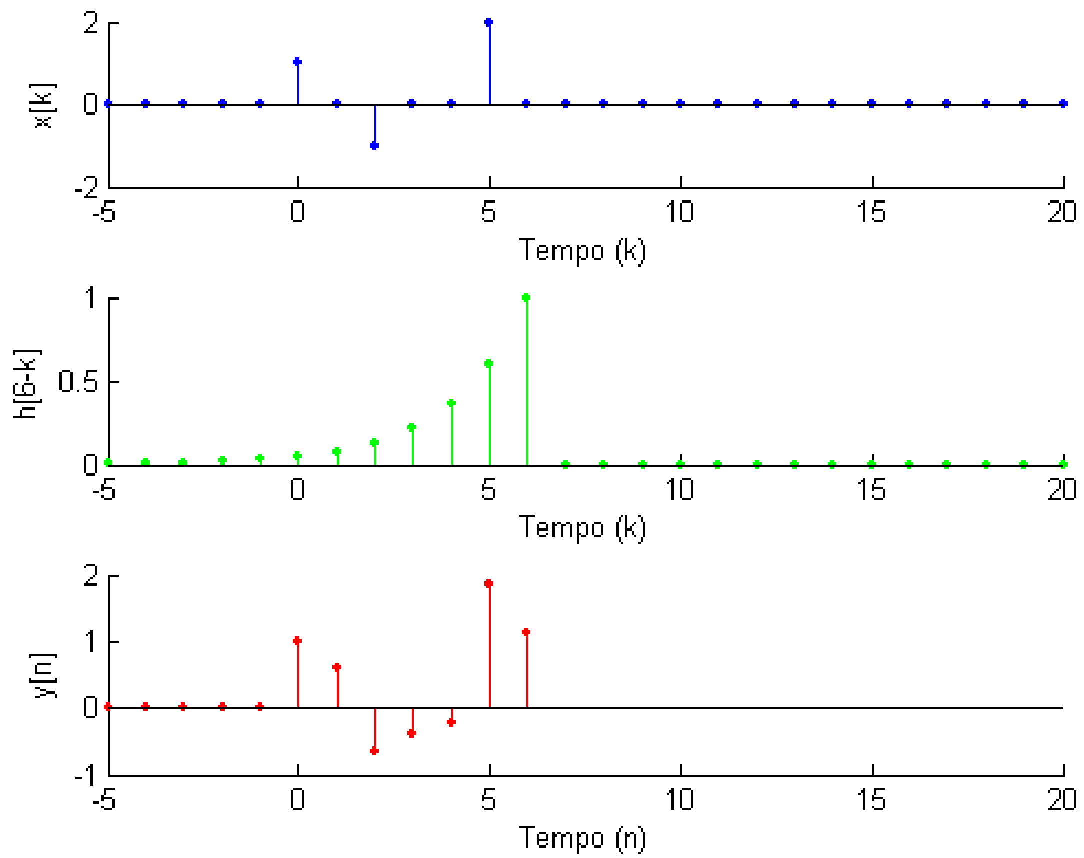
x[k]
2
0
-2
-5
0
5
10
15
20
Tempo (k)
h[4-k]
1
0.5
0
-5
0
5
10
15
20
Tempo (k)
y[n]
2
1
0
-1
-5
0
5
10
15
20
Tempo (n)

n = 5

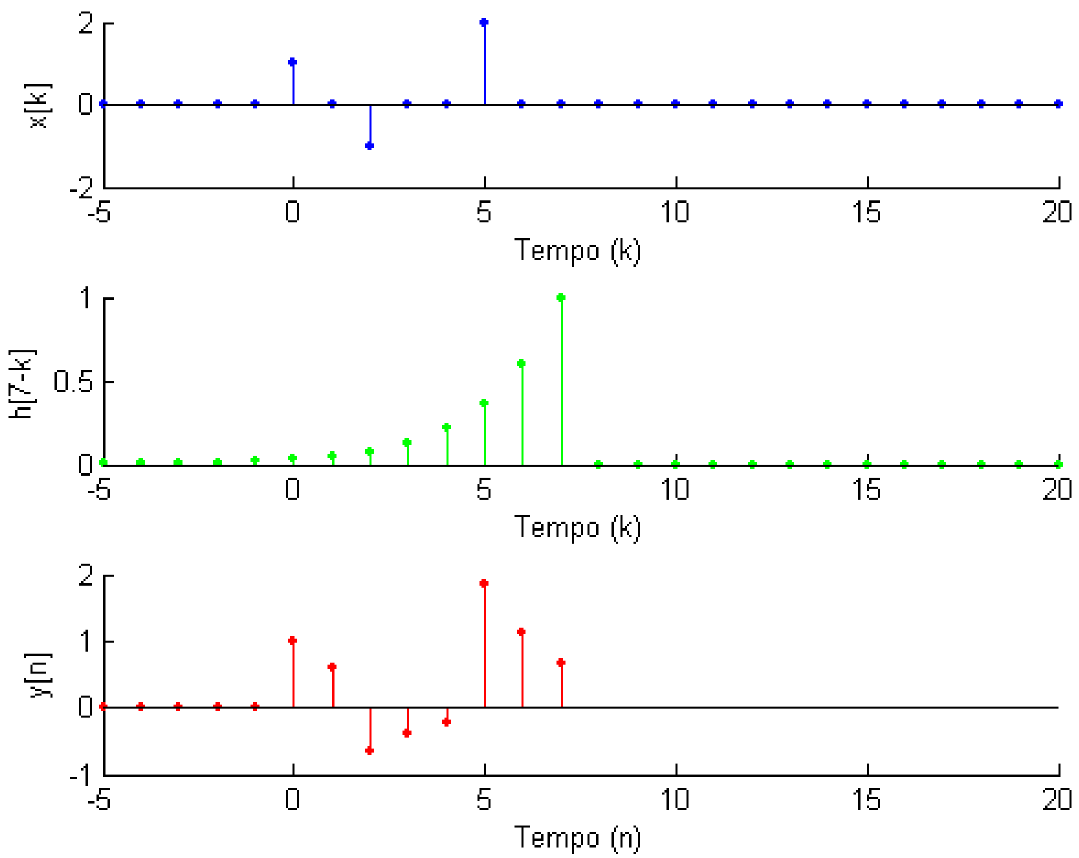


2
0
-2
x[k]
-5
0
5
10
15
20
Tempo (k)
1
0.5
0
h[5-k]
-5
0
5
10
15
20
Tempo (k)
2
1
0
-1
y[n]
-5
0
5
10
15
20
Tempo (n)

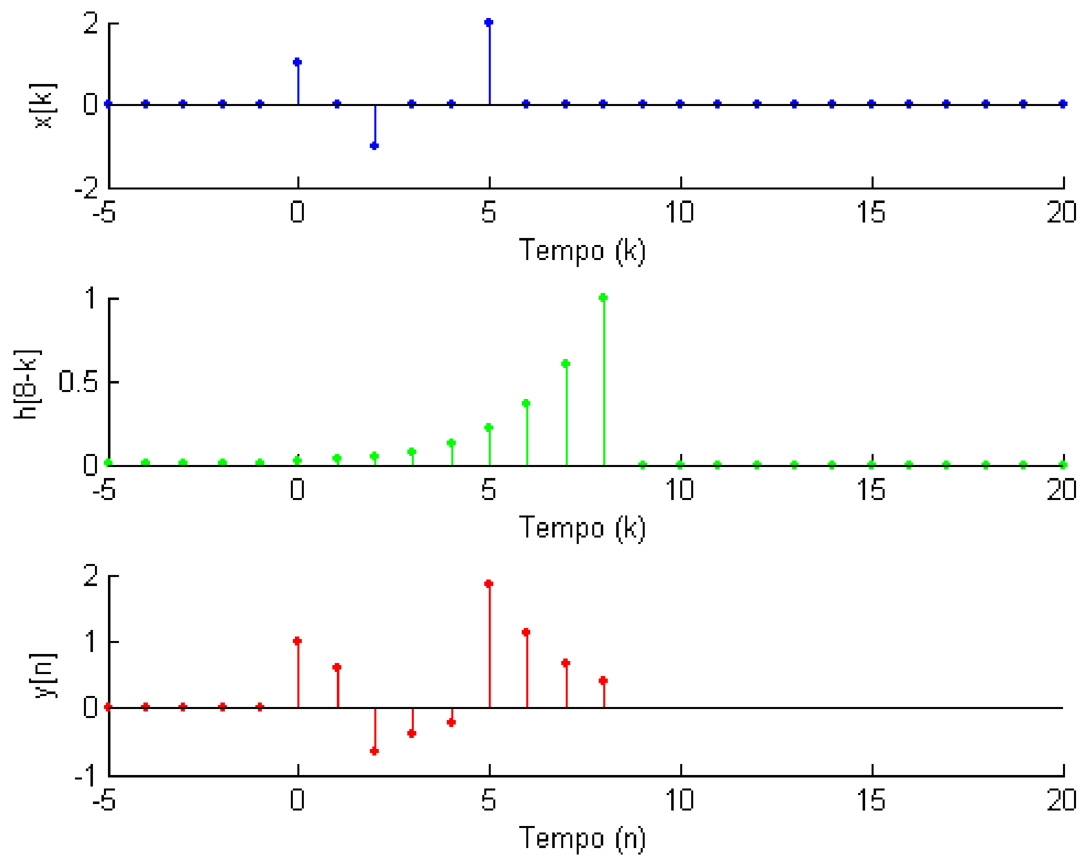
n = 6



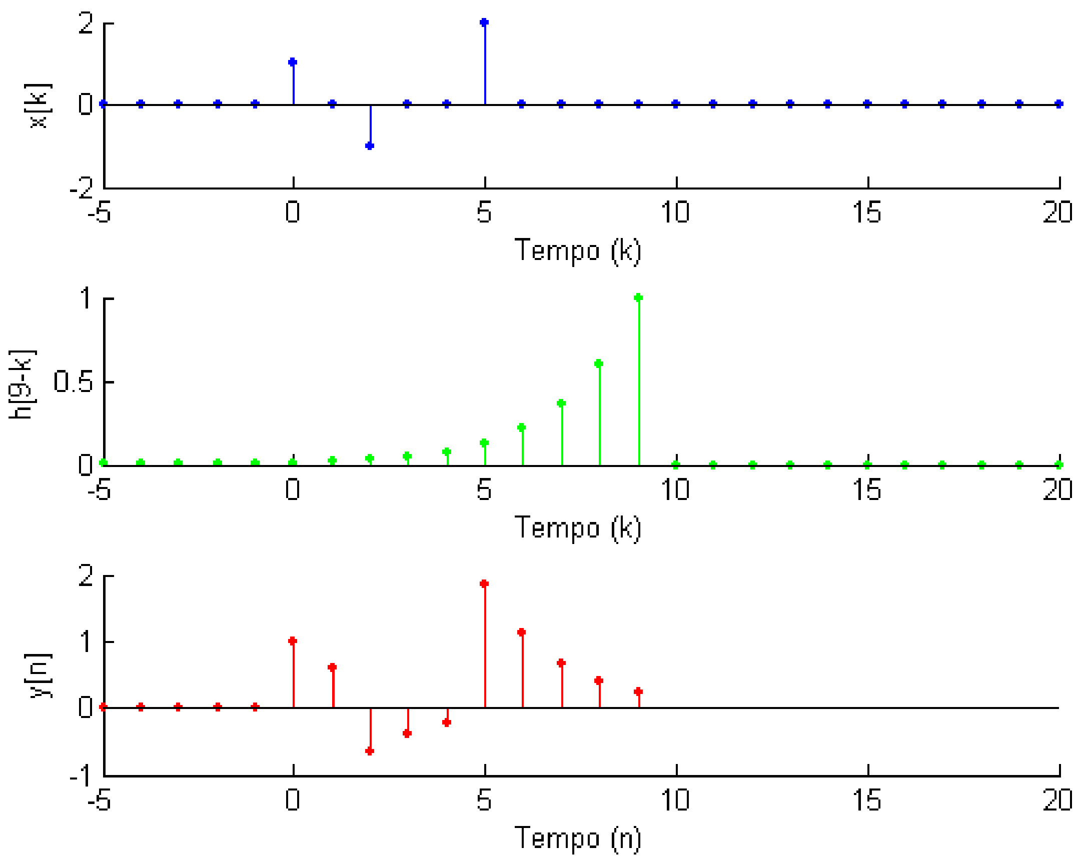
n = 7



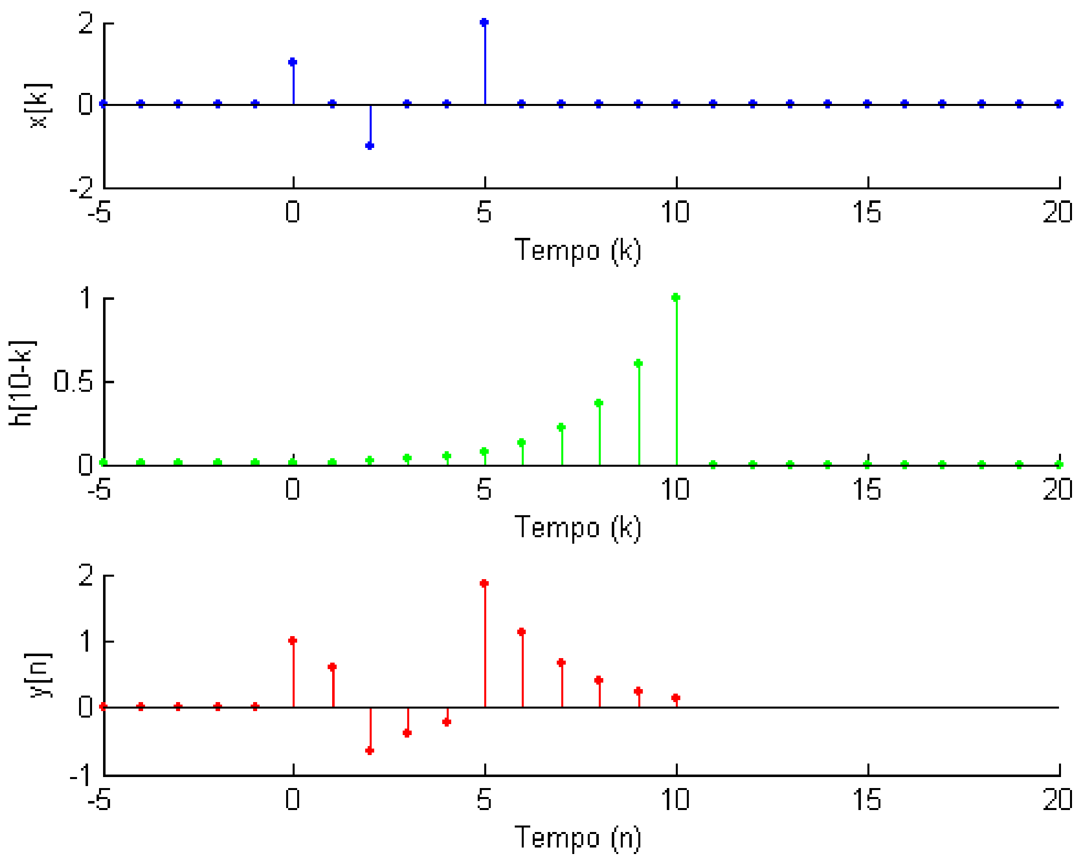
n = 8


x[k]
2
0
-2
-5
0
5
10
15
20
Tempo (k)
h[8-k]
1
0.5
0
-5
0
5
10
15
20
Tempo (k)
y[n]
2
1
0
-1
-5
0
5
10
15
20
Tempo (n)

n = 9

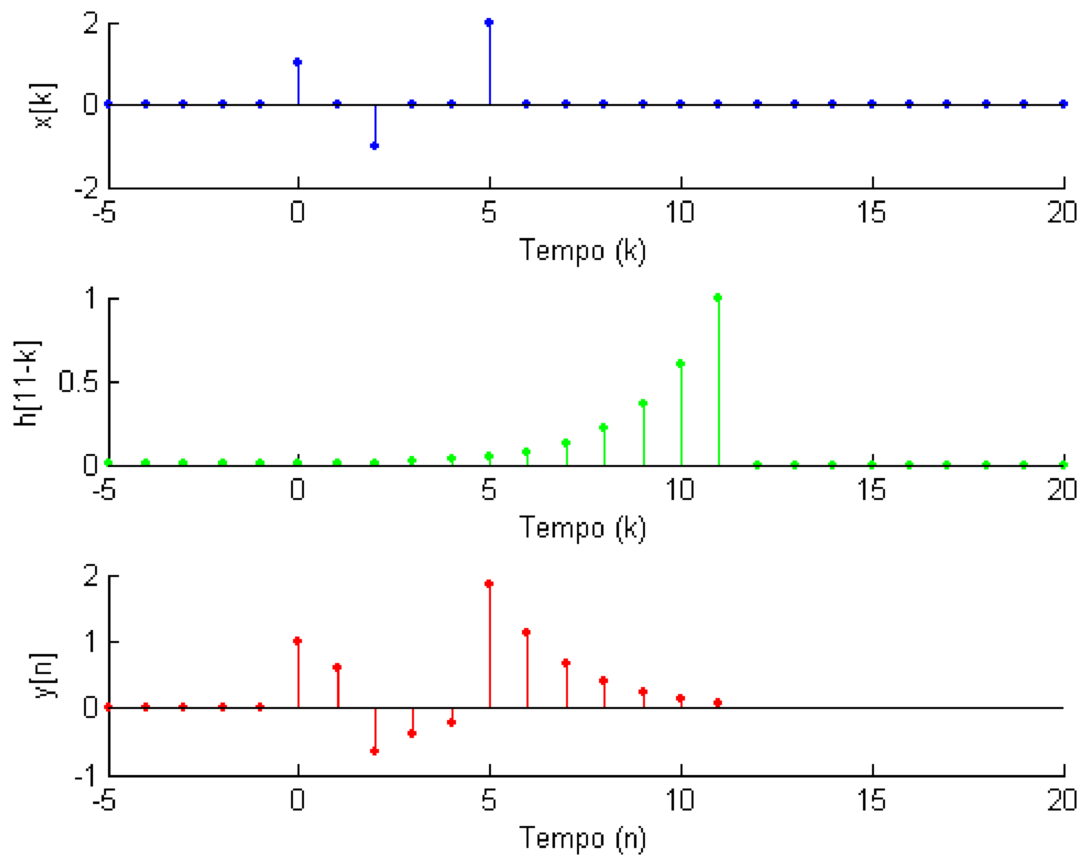


n = 10



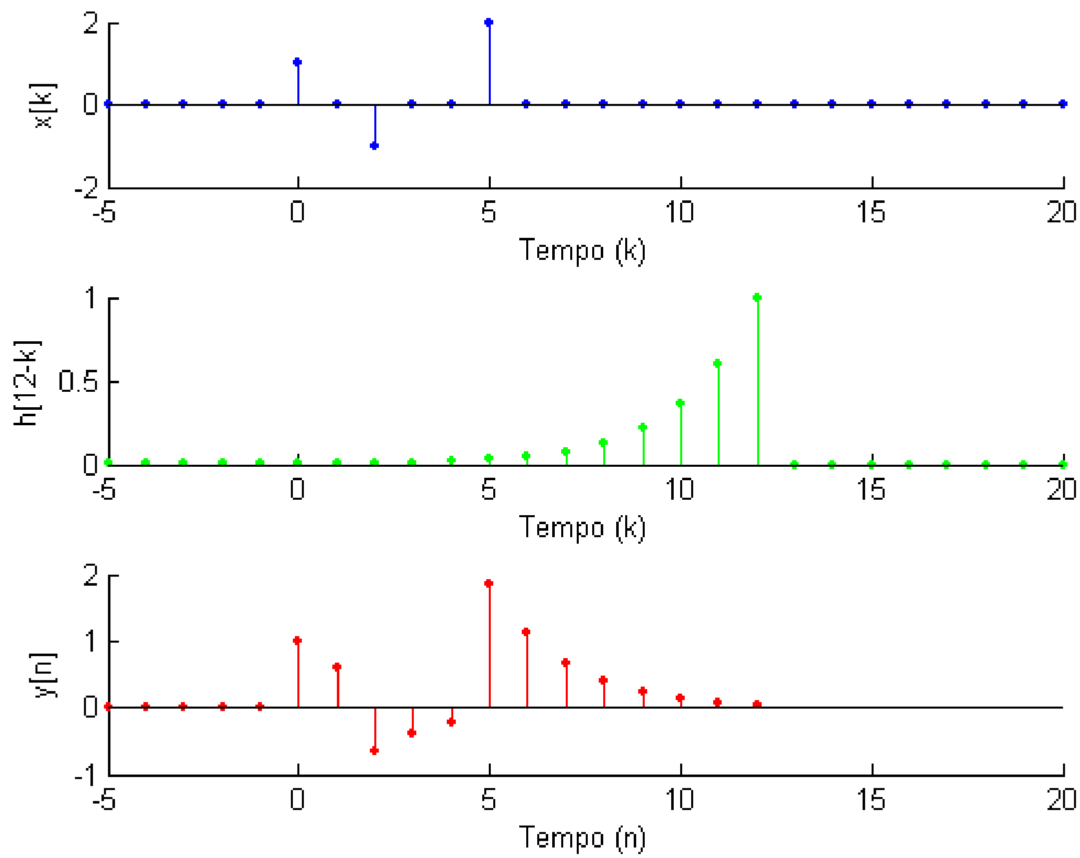
2
0
-2
x[k]
-5
0
5
10
15
20
Tempo (k)
1
0.5
0
h[10-k]
-5
0
5
10
15
20
Tempo (k)
2
1
0
-1
y[n]
-5
0
5
10
15
20
Tempo (n)

n = 11



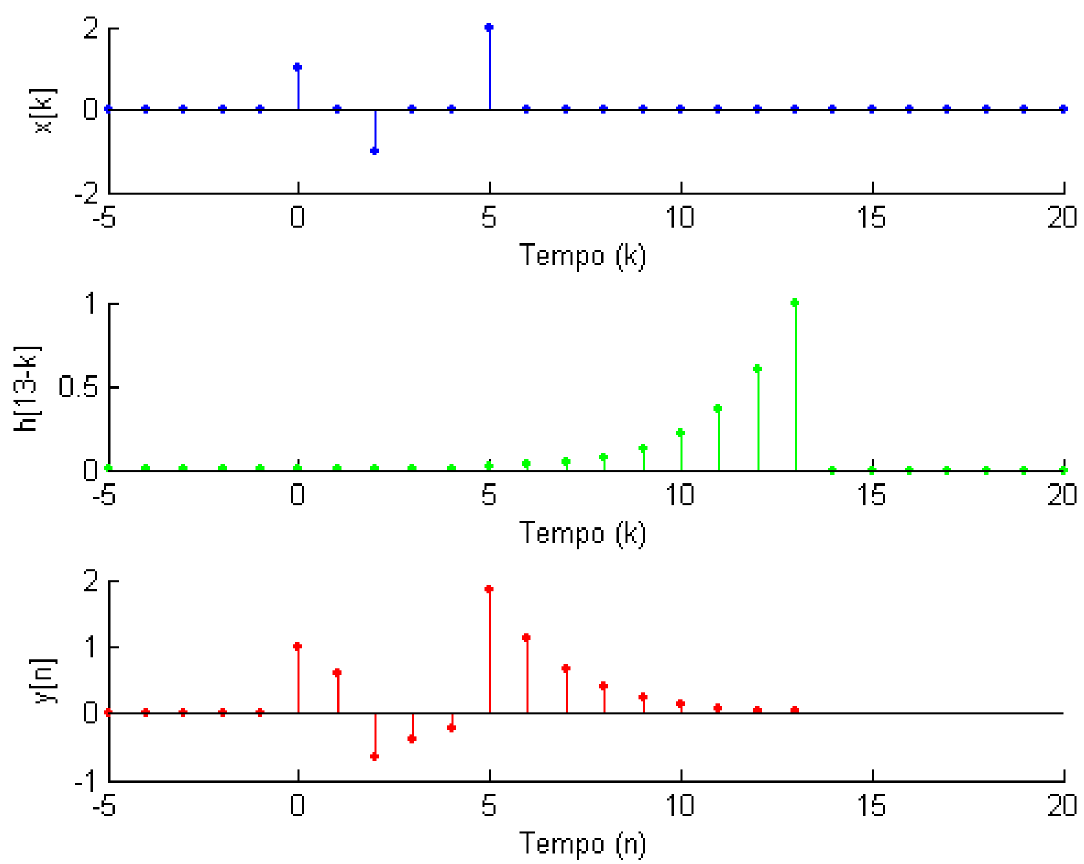
2
0
-2
x[k]
-5
0
5
10
15
20
Tempo (k)
1
0.5
0
h[11-k]
-5
0
5
10
15
20
Tempo (k)
2
1
0
-1
y[n]
-5
0
5
10
15
20
Tempo (n)

n = 12



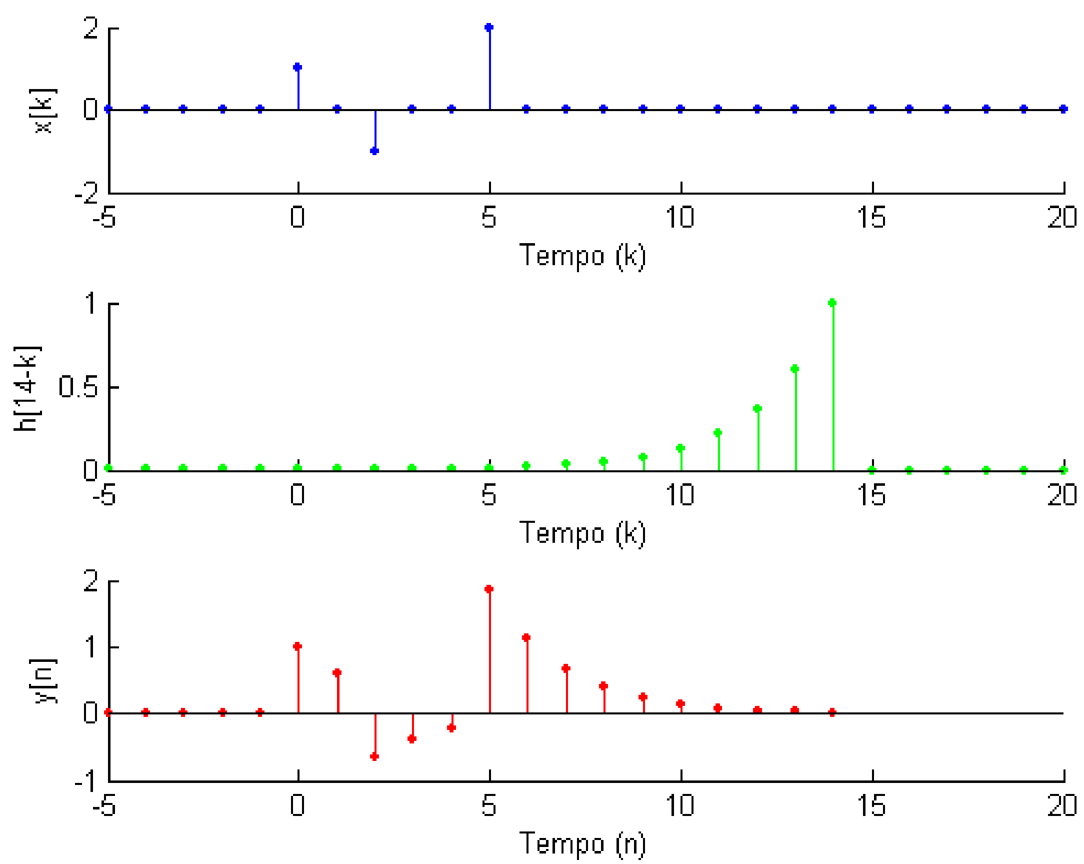
2
0
-2
x[k]
-5
0
5
10
15
20
Tempo (k)
1
0.5
0
h[12-k]
-5
0
5
10
15
20
Tempo (k)
2
1
0
-1
y[n]
-5
0
5
10
15
20
Tempo (n)

n = 13

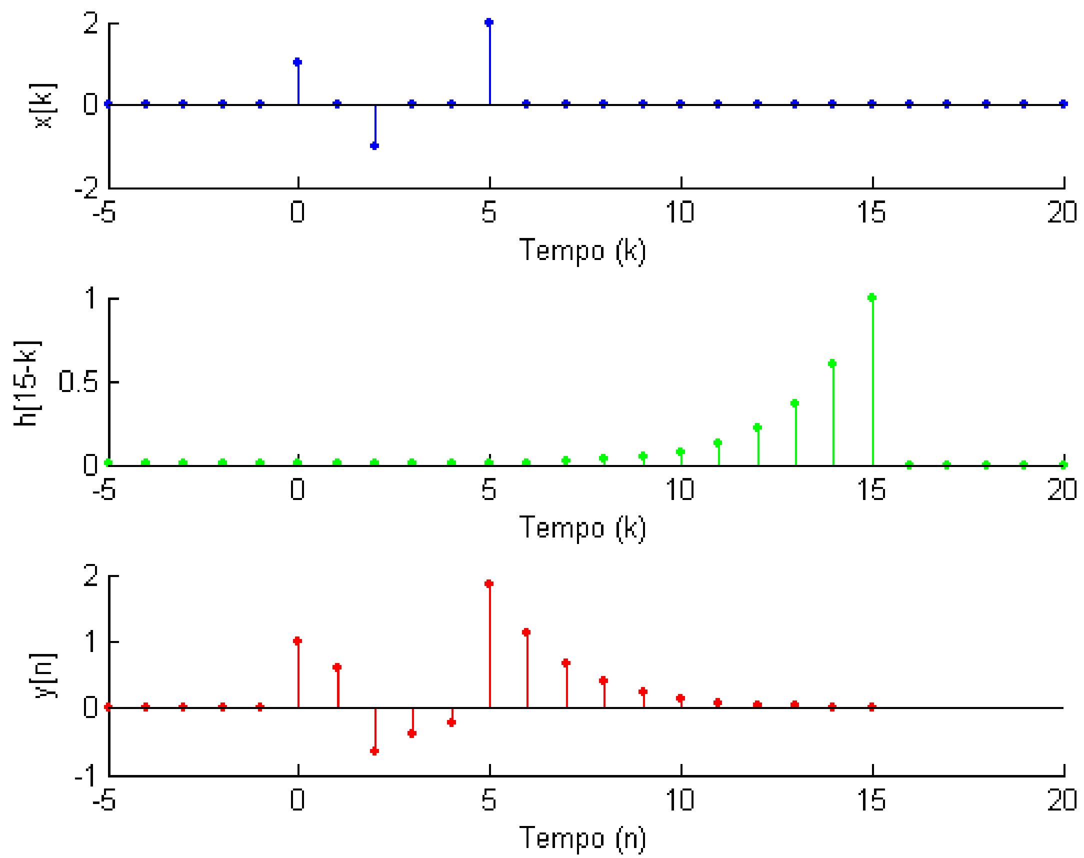


2
0
-2
x[k]
-5
0
5
10
15
20
Tempo (k)
1
0.5
0
h[13-k]
-5
0
5
10
15
20
Tempo (k)
2
1
0
-1
y[n]
-5
0
5
10
15
20
Tempo (n)

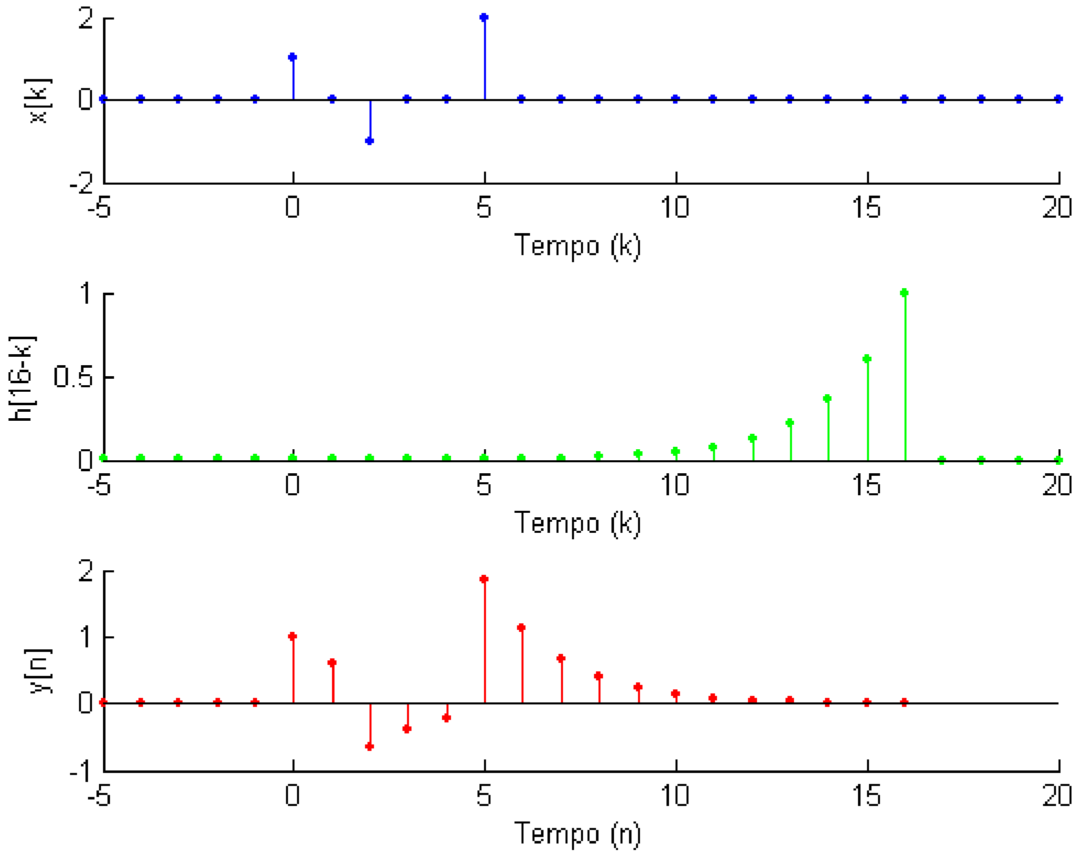
n = 14



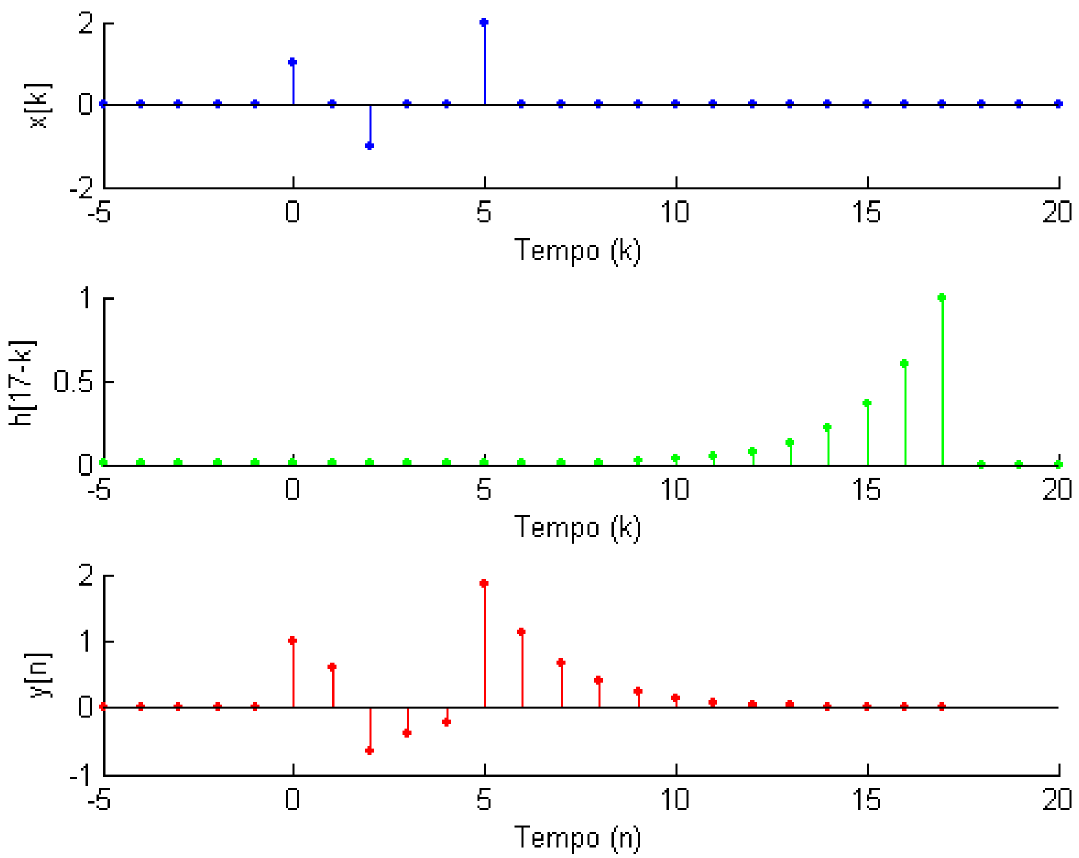
n = 15



n = 16

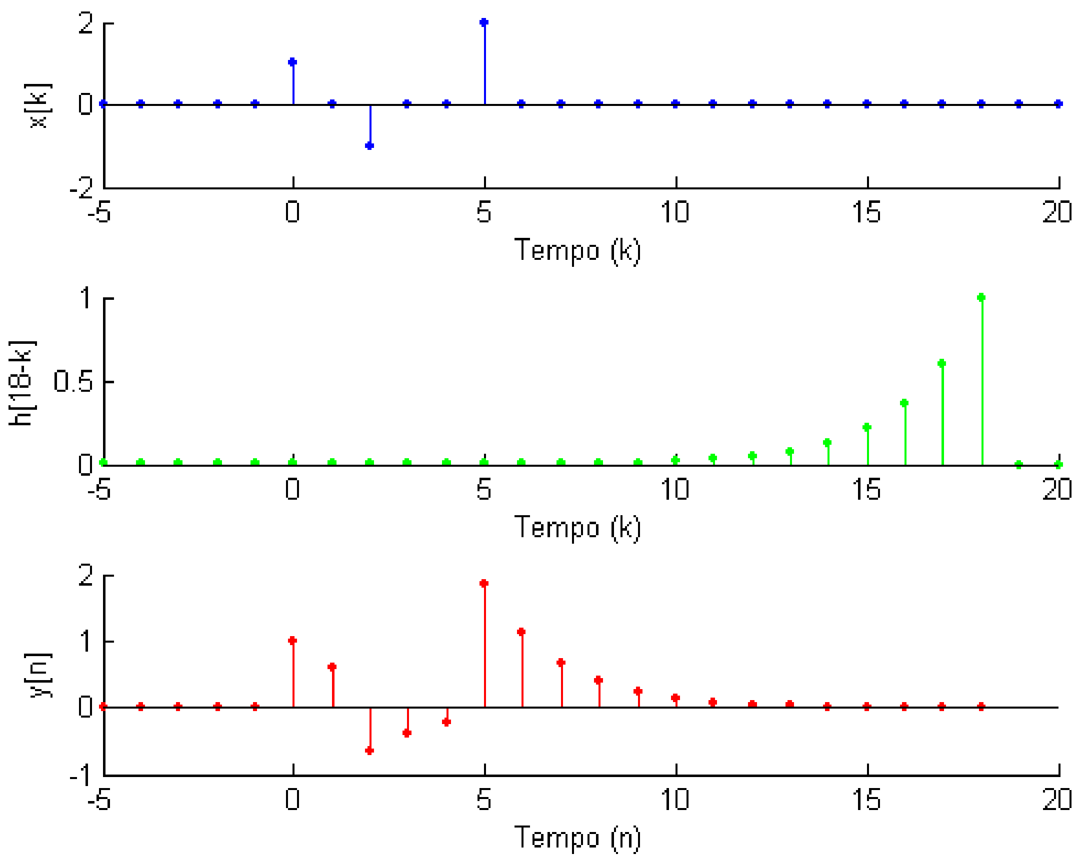


n = 17

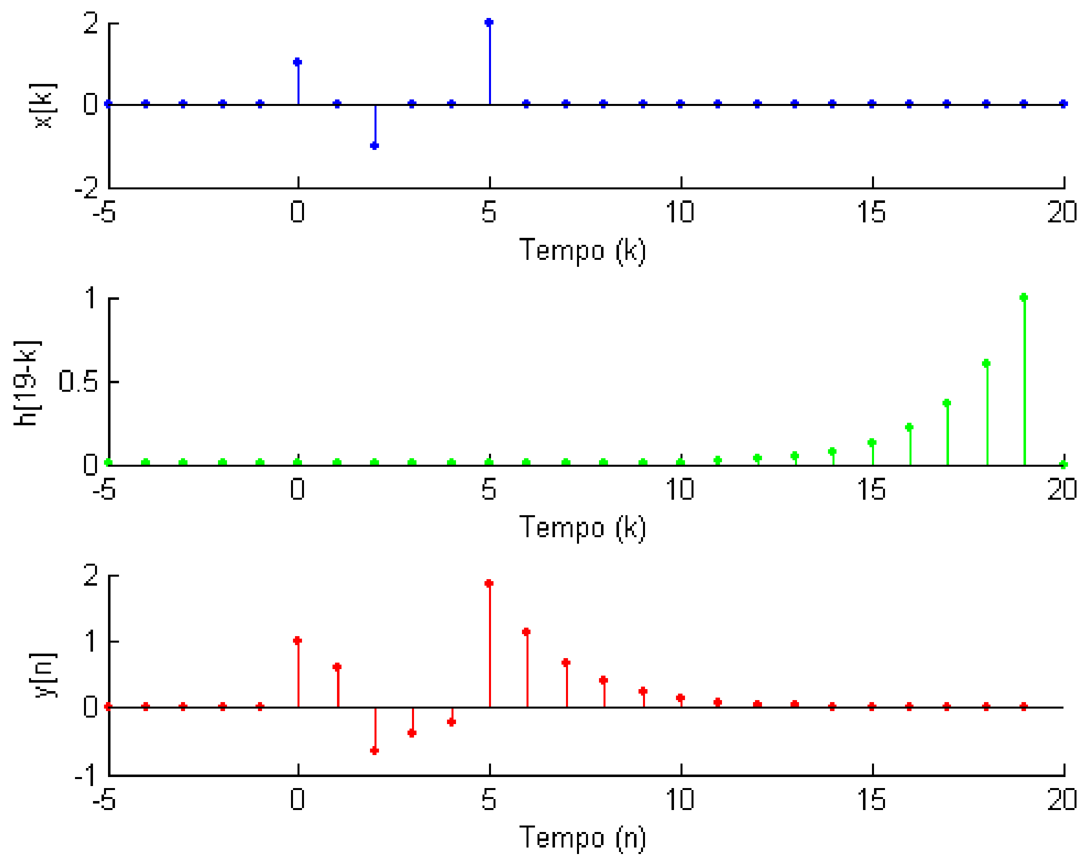


x[k]
2
0
-2
-5
0
5
10
15
20
Tempo (k)
h[17-k]
1
0.5
0
-5
0
5
10
15
20
Tempo (k)
y[n]
2
1
0
-1
-5
0
5
10
15
20
Tempo (n)

n = 18

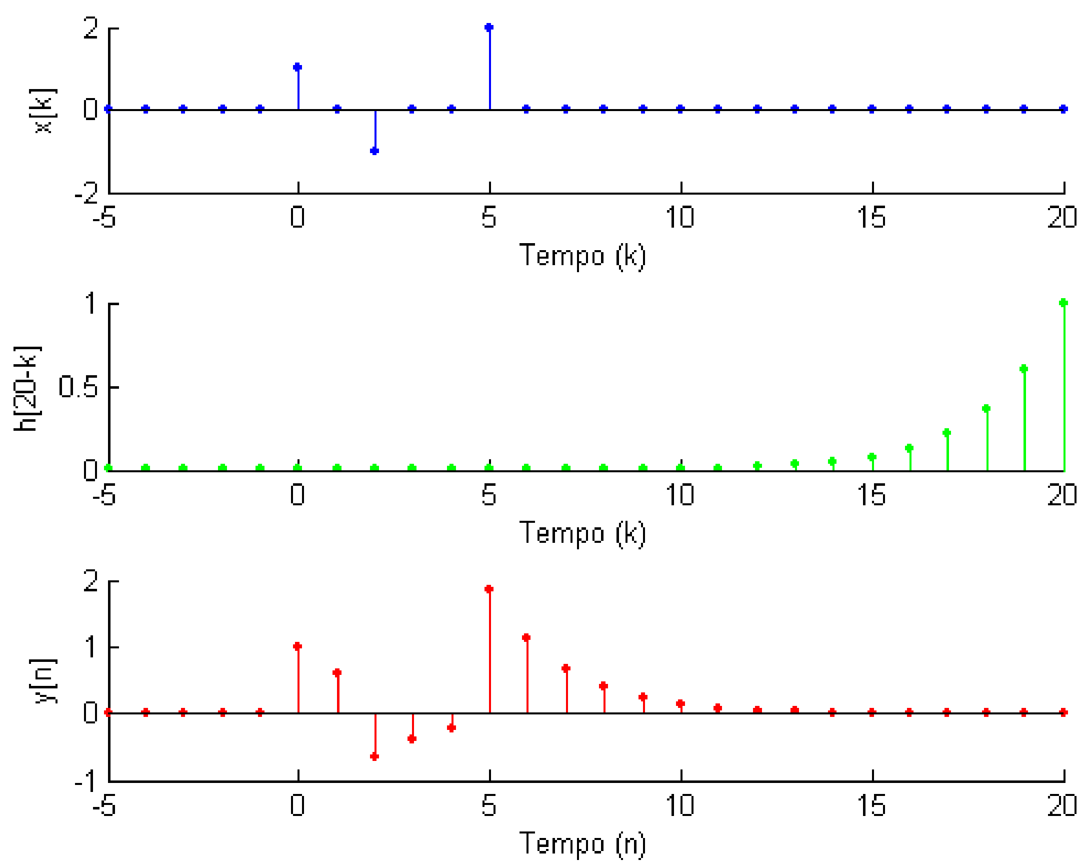


n = 19

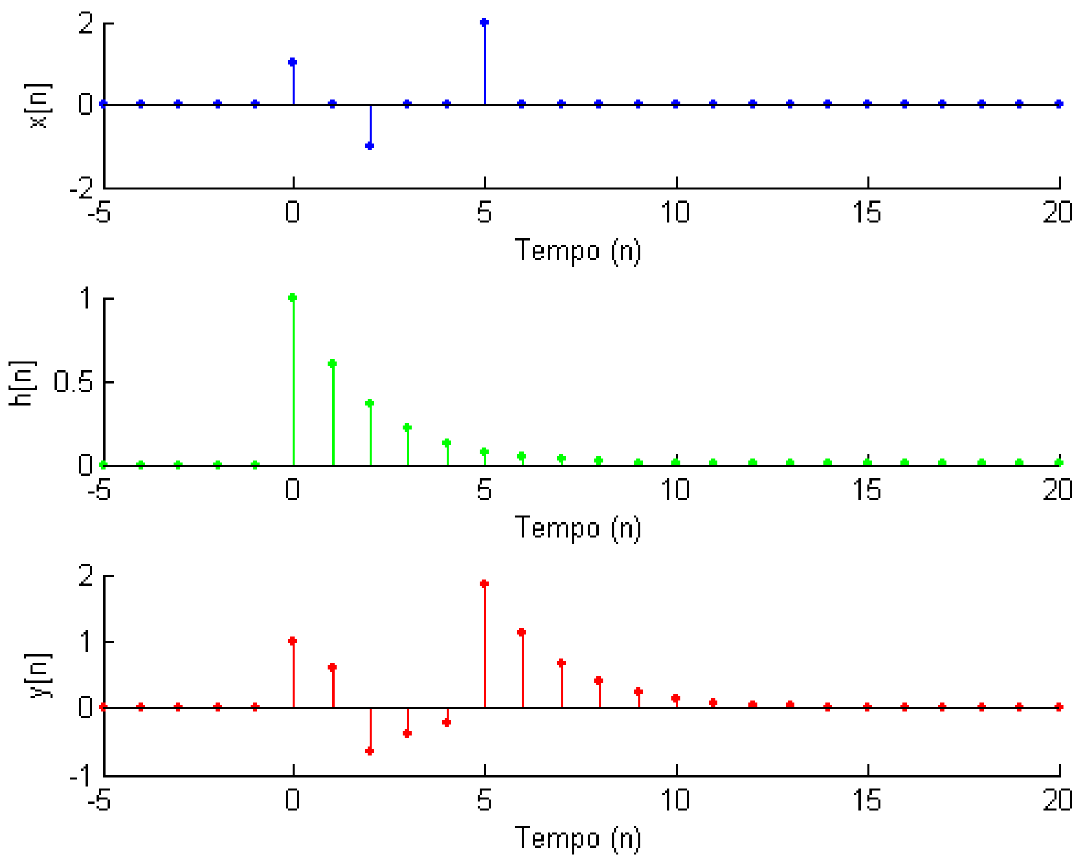


x[k]
2
0
-2
-5
0
5
10
15
20
Tempo (k)
h[19-k]
1
0.5
0
-5
0
5
10
15
20
Tempo (k)
y[n]
2
1
0
-1
-5
0
5
10
15
20
Tempo (n)

n = 20


2
0
-2
x[k]
-5
0
5
10
15
20
Tempo (k)
1
0.5
0
h[20-k]
-5
0
5
10
15
20
Tempo (k)
2
1
0
-1
y[n]
-5
0
5
10
15
20
Tempo (n)

Resumindo...



x[n]
2
0
-2
-5
0
5
10
15
20
Tempo (n)
h[n]
1
0.5
0
-5
0
5
10
15
20
Tempo (n)
y[n]
2
1
0
-1
-5
0
5
10
15
20
Tempo (n)

Educational Matlab GUIs

- Demos sobre Processamento de Sinais: Convolução, Série de Fourier, Transformadas, etc...

http://users.ece.gatech.edu/mcclella/matlabGUIs/index.html

(Acesso em 03/03/2007)

- Vamos brincar um pouco com a Discrete Convolution Demo! ☺