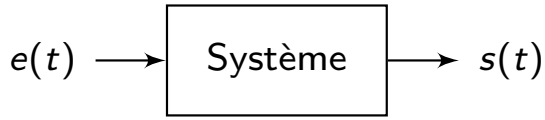
Signaux & Systèmes

Produit de convolution

K. Boudjelaba

Produit de convolution

Introduction



- **Problème :**
  - Dans le contexte générale des systèmes LTIs, comment s'exprime la sortie $s(t)$ en fonction du signal d'entrée $e(t)$ et des "caractéristiques" du système ?

Objectifs

- la sortie peut s'exprimer comme la convolution du signal $e(t)$ avec la réponse impulsionnelle du système $h(t)$.
- Dans le domaine fréquentielle, l'opération de convolution correspond à la multiplication de la TF de $e(t)$ par la TF de $h(t)$.

Produit de Convolution

Définition

Le produit de convolution est défini mathématiquement par :

$$s(t) = (h * e)(t) = \int_{-\infty}^{\infty} h(t-u) \cdot e(u) du = \int_{-\infty}^{\infty} h(u) \cdot e(t-u) du$$

- **WARNING**: L'intégration se fait par rapport à une variable "muette" u qui n'apparait pas dans le résultat du calcul. Le résultat est une fonction qui dépend du temps t.

Produit de convolution discret (signaux numériques)

Soient $e[n]$ et $h[n]$ deux signaux discrets. Le produit de convolution discret entre ces deux signaux est donné par :

$$s[n] = (h \otimes e)[n] = \sum_{k=-\infty}^{\infty} h[n-k] \cdot e[k] = \sum_{k=-\infty}^{\infty} h[k] \cdot e[n-k]$$

Produit de Convolution

Propriété (Convolution avec une impulsion de Dirac)

$$x(t) * \delta(t-\tau) = x(t-\tau)$$

- L'impulsion de Dirac est l'élément neutre du produit de convolution $(x(t) * \delta(t) = x(t))$

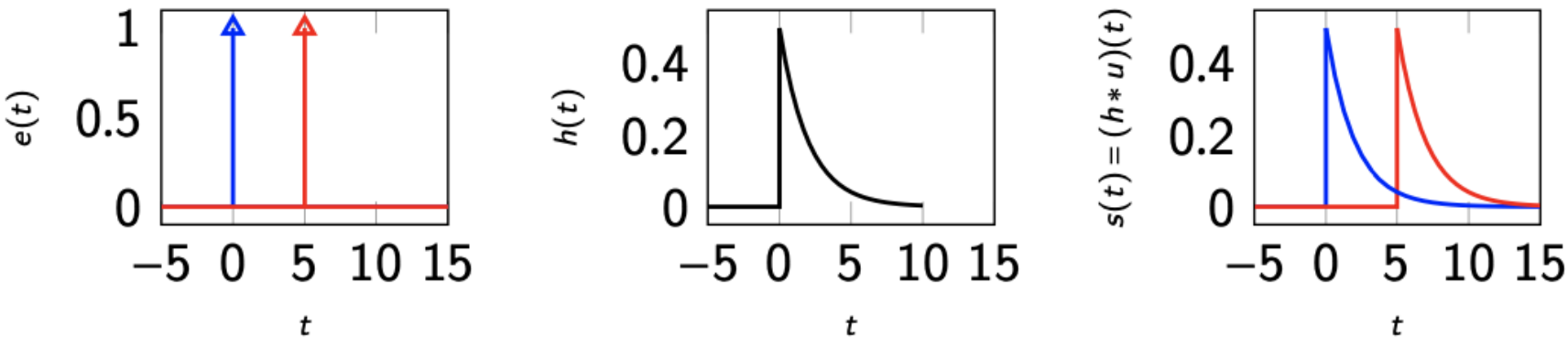


Figure 1: Réponse du système

Produit de Convolution

Propriété (Théorème Fondamental de la Convolution)

Si $s(t) = (h * e)(t)$, alors nous obtenons dans le domaine fréquentiel

$$\boxed{S(f) = H(f)E(f)}$$

- $E(f) = TF[e(t)]$ et $S(f) = TF[s(t)]$ désignent les Transformées de Fourier de $e(t)$ et de $s(t)$.
- $H(f) = TF[h(t)]$ est appelée fonction de transfert du système.

Remarque

- Convoluer deux signaux dans le domaine temporel revient à les multiplier dans le domaine fréquentiel → notion de filtrage.
- (Dualité): Multiplier deux signaux dans le domaine temporel revient à les convoluer dans le domaine fréquentiel.

Produit de Convolution

Exercice

Soient $h[n] = [\ \underset{\underset{0}{\uparrow}}{1},\ \underset{\underset{1}{\uparrow}}{2},\ \underset{\underset{2}{\uparrow}}{2},\ \underset{\underset{3}{\uparrow}}{3}\ ]$ et $e[n] = [\ \underset{\underset{0}{\uparrow}}{2},\ \underset{\underset{1}{\uparrow}}{-1},\ \underset{\underset{2}{\uparrow}}{3}\ ]$

Calculer le produit de convolution $(h \otimes e)[n]$.

Produit de Convolution

Solution 1

$h[n]$	=	1	2	2	3
$e[n]$	=	2	-1	3	

Produit de Convolution

Solution 1

$h[n]$	=	1	2	2	3
$e[n]$	=	2	-1	3	
		2	4	4	6

Produit de Convolution

Solution 1

$h[n]$	=	1	2	2	3	
$e[n]$	=	2	-1	3		
		2	4	4	6	
			-1	-2	-2	-3

Produit de Convolution

Solution 1

$h[n]$	=	1	2	2	3		
$e[n]$	=	2	-1	3			
		2	4	4	6		
			-1	-2	-2	-3	
				3	6	6	9

Produit de Convolution

Solution 1

$h[n]$	=	1	2	2	3		
$e[n]$	=	2	-1	3			
		2	4	4	6		
	+		-1	-2	-2	-3	
	+			3	6	6	9
$y[n]$	=	2	3	5	10	3	9
		↑	↑	↑	↑	↑	↑
		0	1	2	3	4	5

Produit de Convolution

Solution 2

	3	2	2	1	$\Leftarrow$ $h[n]$
		3	-1	2	$\Leftarrow$ $e[n]$

Produit de Convolution

Solution 2

	3	2	2	1	$\Leftarrow$	$h[n]$
		3	-1	2	$\Leftarrow$	$e[n]$
	6	4	4	2		

Produit de Convolution

Solution 2

		3	2	2	1	$\Leftarrow$ $h[n]$
			3	-1	2	$\Leftarrow$ $e[n]$
		6	4	4	2	
	-3	-2	-2	-1		

Produit de Convolution

Solution 2

		3	2	2	1	$\Leftarrow$	$h[n]$
			3	-1	2	$\Leftarrow$	$e[n]$
		6	4	4	2		
	-3	-2	-2	-1			
9	6	6	3				

Produit de Convolution

Solution 2

			3	2	2	1	$\Leftarrow$	$h[n]$
				3	-1	2	$\Leftarrow$	$e[n]$
			6	4	4	2		
+		-3	-2	-2	-1			
+	9	6	6	3				
=	9	3	10	5	3	2	$\Leftarrow$	$y[n]$
	↑	↑	↑	↑	↑	↑		
	5	4	3	2	1	0		

$$y[n] = [\ \underset{0}{2},\ \underset{1}{3},\ \underset{2}{5},\ \underset{3}{10},\ \underset{4}{3},\ \underset{5}{9}\ ]$$

Produit de Convolution

Code Python

On utilise la fonction `convolve` disponible dans la librairie `numpy`.

```
import numpy as np

h = np.array([1, 2, 2, 3])
e = np.array([2, -1, 3])
y = np.convolve(h,e)

print(y)
```

```
[ 2  3  5 10  3  9]
```

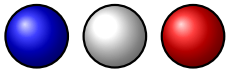
On crée une fonction `convolution` qui calcule le produit de convolution à partir de la formule :

$$(h \otimes e)[n] = \sum_{k=-\infty}^{\infty} h[k] \cdot e[n-k]$$

```
import numpy as np

def convolution(A,B):
    tailleA = np.size(A)
    tailleB = np.size(B)
    C = np.zeros(tailleA + tailleB -1)
    for m in np.arange(tailleA):
        for n in np.arange(tailleB):
            C[m+n] = C[m+n] + A[m]*B[n]
    return C

h = np.array([1, 2, 2, 3])
e = np.array([2, -1, 3])
y = convolution(h,e)

print(y)
```

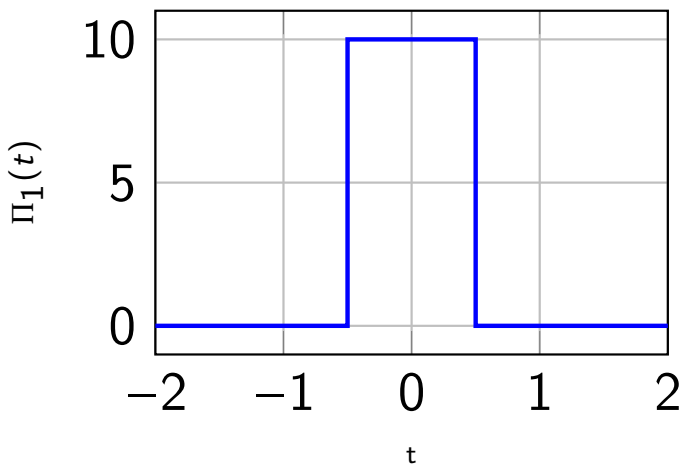
```
[ 2.  3.  5. 10.  3.  9.]
```

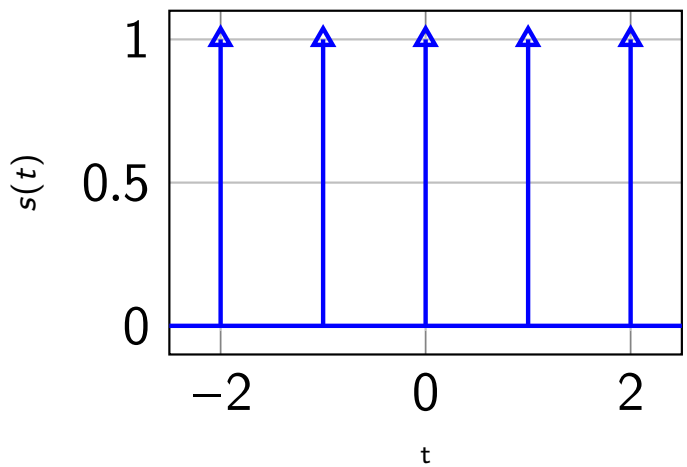
Produit de Convolution



Exercice : sous Python

- Calculer et tracer le produit de convolution de deux portes rectangulaires identiques.
- Calculer et tracer le produit de convolution de la porte rectangulaire et le peigne de Dirac.



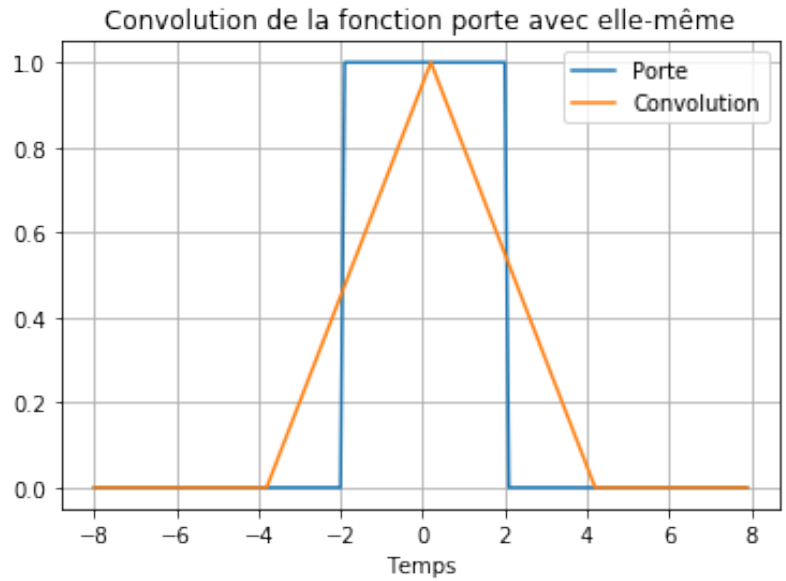


Produit de Convolution

Code Python

```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

a = 2.
t = np.arange(-8,8,0.1)
p = np.zeros(len(t))
p[np.abs(t)<a] = 1

c = np.convolve(p,p,'same')

plt.figure()
plt.plot(t,p,label='Porte')
# Ou
# plt.plot(t,c/c.max(),label='
                    Convolution')
plt.plot(t,c/sum(p),label='
                    Convolution')
plt.xlabel('Temps')
plt.title('Convolution de la fonction
                     porte avec elle-m
                    ême')
plt.legend()
plt.grid()
plt.show()
```

On utilise la fonction convolve disponible dans la librairie numpy.



Produit de Convolution

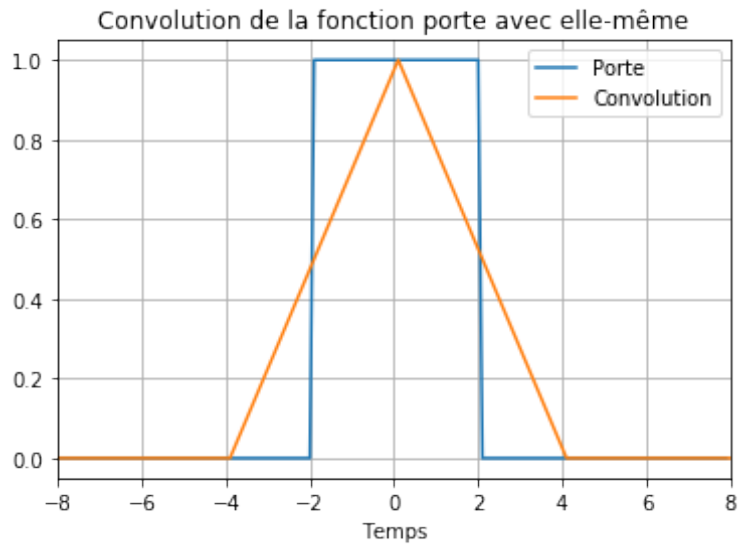
Code Python

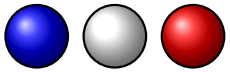
```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
def convolution(A,B):
    tailleA = np.size(A)
    tailleB = np.size(B)
    C = np.zeros(tailleA + tailleB -1)
    for m in np.arange(tailleA):
        for n in np.arange(tailleB):
            C[m+n] = C[m+n] + A[m]*B[n]
    return C
a = 2.
t = np.arange(-8,8,0.1)
t1 = np.arange(-16,16-0.1,0.1)
y = np.zeros(len(t))
y[np.abs(t)<a] = 1
z = convolution(y,y)
plt.figure()
plt.plot(t,y,label='Porte')
plt.plot(t1,z/z.max(),label='Convolution')
plt.xlabel('Temps')
plt.title('Convolution de la fonction porte
                        avec elle-même')
plt.xlim([-8,8])
plt.legend()
plt.grid()
plt.show()
```

On crée une fonction `convolution` qui calcule le produit de convolution à partir de la formule :

$$(h \otimes e)[n] = \sum_{k=-\infty}^{\infty} h[n-k] \cdot e[k]$$

$$= \sum_{k=-\infty}^{\infty} h[k] \cdot e[n-k]$$





Exercice : sous Python

Tracer $s(t) = \sum_{n=-N}^{N} C_n e^{j2\pi n f_0 t}$ pour $N = 3$ puis $N = 7$ et $N = 11$.

Avec $f_0 = 10$ Hz.

- Coefficients de la DSF complexe d'un signal carré ($A = 2$):

$$C_n = \begin{cases} \frac{A}{2} & \text{pour } n = 0 \\ 0 & \text{pour } n \text{ pair et } \neq 0 \\ -j\frac{A}{n\pi} & \text{pour } n \text{ impair} \end{cases}$$

- Coefficients de la DSF complexe d'un signal triangulaire ($A = 2$):

$$C_n = \begin{cases} \frac{A}{2} & \text{pour } n = 0 \\ 0 & \text{pour } n \text{ pair et } \neq 0 \\ -\frac{2A}{n^2\pi^2} & \text{pour } n \text{ impair} \end{cases}$$

Produit de Convolution