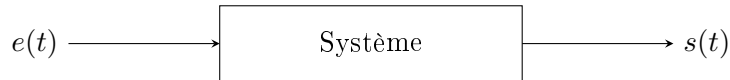


Dans ce TP, on présente la transformation bilinéaire qui permet de donner une solution approximative des équations différentielles.

1 Problèmes

On considère un système (par exemple un circuit électronique) recevant $e(t)$ en entrée et produisant $s(t)$ en sortie.



Le système obéit à une certaine équation différentielle.

$$(\mathcal{E}) : \quad \tau \cdot s' + s = e$$

On sait également que $s(0) = 0$.

2 Principe

Passant l'équation différentielle dans le domaine de Laplace on obtient

$$\tau p \cdot S(p) + S(p) = E(p) \Rightarrow S(p) = \frac{1}{1 + \tau p} E(p)$$

On déduit que la fonction de transfert du système est $H(p) = \frac{1}{1 + \tau p}$.

La transformation bilinéaire consiste à chercher une version en z de $H(p)$. On pourra noter $\hat{H}(z)$ cette fonction. On pose, considérant la période d'échantillonnage T_e :

$$\hat{H}(z) = H\left(\frac{2}{T_e} \frac{1 - z^{-1}}{1 + z^{-1}}\right)$$

Dans le cadre d'une simulation sur ordinateur, au lieu de fixer T_e , on choisit la durée T de l'expérience et le nombre N d'échantillons de sorte que $T_e = \frac{T}{N}$.

- (1) Exprimez $\hat{H}(z)$ en fonction de T_e, z, τ .
- (2) Déduisez-en la récurrence entre l'entrée e_n et la sortie s_n .
Cette récurrence dépend de T_e et τ .

Bien que tout cela devrait conduire à $s_0 \neq 0$, on impose $s_0 = 0$ car l'expérience montre que les résultats sont meilleurs.

3 Tableur

À chaque fois, vous pensez à étendre les formules jusqu'en bas.

- (1) Ouvrez le tableur et recopier ce tableau

	A	B	C	D	E	F
1	tau	2	T	10	N	1000
2						
3	n	t0	e(n)	s(n)		
4	0			0		
5	1					

Il faut étendre la colonne A pour que n aille jusque 1000

- (2) Écrivez en B4 la formule permettant de calculer le temps t_0 en fonction de n et $T_e = \frac{T}{N}$.

- (3) Pour e on prendra $e(t) = \begin{cases} 1 & \text{si } 0 \leq t \leq 2.5 \\ 0 & \text{sinon} \end{cases}$ Complétez la colonne C.
- (4) En D5, écrivez la récurrence trouvée précédemment.
- (5) Tracez les courbes de e et s avec t en abscisse. Vous pourrez ensuite modifier la valeur de τ pour voir l'effet.

4 Code Python

Les numéros à gauche sont les numéros de ligne. Ne pas les recopier.

- (1) On commence par importer le module graphique

```
Py 1 import matplotlib.pyplot as plt # graphique
```

- (2) On définit T et N ce qui permet de calculer dt .

```
Py 2 T = 10
3 N = 1000
4 dt = T/N
```

- (3) On fournit les paramètres de l'équation : il faut indiquer la valeur de τ et l'expression de $e(t)$. On reprend le même que précédemment.

```
Py 5 tau = 2
6 def e(t):
7     if 0 <= t <= 2.5:
8         return 1
9     else:
10        return 0
```

- (4) On va utiliser des **tableaux**. Ces tableaux sont des listes de nombres que l'on va compléter au fur et à mesure. Les tableaux sont initialisés avec []

```
Py 11 liste_t = [0] # liste des temps
12 liste_e = [e(0)] # liste des valeurs de e
13 liste_s = [0] # liste des valeurs de s
```

- (5) Pour ajouter une valeur à une liste, on utilise `append`. Quand on veut lire le dernier élément d'une liste, on utilise `[-1]`. Pour parcourir tous les instants de la simulation, on utilise une boucle `for`.

```
Py 14 for i in range(N): # répète N fois
15     nouveau_s = ...
16     nouveau_t = ...
17     nouveau_e = ...
18     liste_s.append(nouveau_s)
19     liste_t.append(nouveau_t)
20     liste_e.append(nouveau_e)
```

Complétez les ...

- (6) Il ne reste qu'à représenter les courbes.

```
Py 21 plt.plot(liste_t, liste_e, 'r', label='entrée')
22 plt.plot(liste_t, liste_s, 'b', label='sortie')
23 plt.legend()
24 plt.show()
```

Exécutez pour visualiser les courbes.

Vous pouvez répéter l'opération en modifiant des paramètres comme τ .