

Introduction au signal et bruit

Exercices

Gabriel Dauphin

November 23, 2025

Contents

1 Relations entrées-sorties sans effet mémoire	2
2 Signaux temps continu, fonction affine par morceaux	4
3 Définition et utilisation de la transformée de Fourier	6
4 Propriété de la transformée de Fourier	8
5 Diracs	11
6 Transformées de Fourier et dérivation	13
7 Équations différentielles	16
8 Filtres et effet mémoire	19
9 Description fréquentielle des filtres	22
10 Signaux périodiques	24
11 Filtres agissant sur des signaux périodiques	28
12 Échantillonnage d'un signal non-périodique	31
13 Modélisation stochastique du bruit	33
14 Autocorrélation et densité spectrale	37
15 Filtrage des processus aléatoires et application au bruit en $1/f$	41
16 Densité de probabilité et application au bruit de grenaille	45
A Supplément pour faire les pseudo-programmes	53
A.1 Outils	53
A.1.1 Commandes générales	53

Chapter 1

Relations entrées-sorties sans effet mémoire

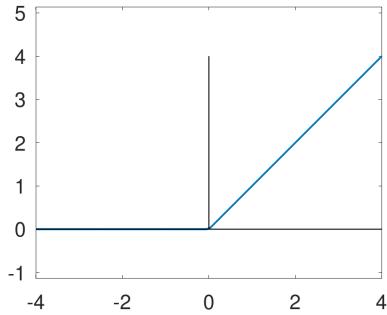


Figure 1.1: Relation entrée-sortie associée à un Relu (exercice 1)

Exercice 1 Le graphique représente la relation entrée-sortie d'un Relu pour Rectified Linear Unit.

1. En utilisant la figure 1.1, combien valent les signaux en sortie lorsque respectivement, les signaux en entrées valent -3 et 3 ?
2. Combien valent les puissances de ces signaux ?
3. Proposez une formule utilisant la valeur absolue, l'addition et la multiplication pour modéliser cette relation ?
4. On considère le filtre $\mathcal{H}_1(x) = 0.5x$ et $\mathcal{H}_2(x) = |x|$, montrez comment on peut fabriquer le filtre Relu.
5. Écrire le pseudo-code permettant de générer la figure 1.1.

Simulation de la figure 1.1.

```
x=linspace(-4,4,1e2);
y=zeros(size(x));
y(x<=0)=0;
y(x>0)=x(x>0);
figure(1); plot(x,y); figure_jolie(1);
xlabel('x'); ylabel('y'); axis('equal');
saveas(1,'./figures/fig_exSEB6a.png');
```

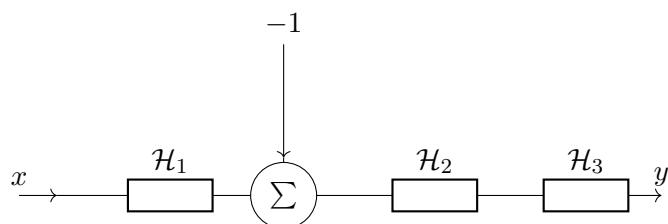


Figure 1.2: Schéma décrivant \mathcal{H} à partir de $\mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3$ pour l'exercice 2.

Exercice 2 Les filtres \mathcal{H}_1 , \mathcal{H}_2 et \mathcal{H}_3 sont définis par

$$\mathcal{H}_1(x) = |x| \quad \mathcal{H}_2(x) = \min(1, x) \quad \mathcal{H}_3(x) = \max(0, x) \quad (1.1)$$

On appelle \mathcal{H} le filtre décrit par la figure 1.2 et défini par les filtres $\mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3$. et associé à la relation transformant x en y .

1. Calculez les sorties y associées aux valeurs $-2, -1, 0, 1, 2$ pour x .
2. Écrivez la formule modélisant \mathcal{H} ?
3. Dessinez la relation associée à \mathcal{H} transformant x en y sur un graphe.

Chapter 2

Signaux temps continu, fonction affine par morceaux

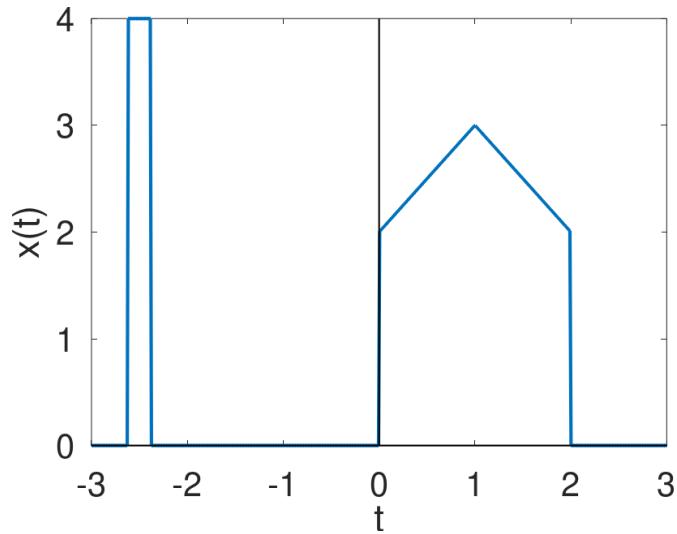


Figure 2.1: Visualisation de $x(t)$ qui a la forme d'une maison avec son lampadaire (exercice 3).

Exercice 3 On considère le signal $x(t)$ décrit par la figure 2.1.

1. Calculez les valeurs de $x(t)$ pour les valeurs de $t = -2.5, 0.5, 1, 2.5$.
2. Écrivez une formule décrivant $x(t)$ au moyen de différents intervalles de temps.
3. Utilisez quelques unes des fonctions de base présentées en cours pour définir $x(t)$.
4. Utilisez le crochet d'Iverson pour décrire $x(t)$.

Simulation de la figure 2.1

```
t=linspace(-3,3,500);
x=2*fonction_porte((t-1)/2)+fonction_T(t-1)+4*fonction_porte((t+2.5)*4);
figure(1); plot(t,x); figure_jolie(1);
xlabel('t'); ylabel('x(t)');
saveas(1,'./figures/fig_exSEB8a.png');
```

Exercice 4 On considère le signal $x(t)$ ainsi défini

$$x(t) = (at + b) \llbracket t_1 \leq t \leq t_2 \rrbracket \quad (2.1)$$

1. Représentez ce signal pour $a = 1$, $b = 0$ et $t_1 = 2$, $t_2 = 3$.

2. Représentez ce signal pour $a = -1$, $b = 1$ et $t_1 = 0$, $t_2 = 1$.

3. Montrez que pour $a = 0$, $x(t)$ peut se mettre sous la forme

$$x(t) = \alpha\Pi(\gamma t + \delta) \quad (2.2)$$

4. Montrez que pour $a > 0$, $x(t)$ peut se mettre sous la forme

$$x(t) = \alpha\Pi(\gamma t + \delta) + \beta\mathbb{C}(\gamma t + \delta) \quad (2.3)$$

5. Donnez un pseudo-code permettant de visualiser ce signal.

Chapter 3

Définition et utilisation de la transformée de Fourier

Exercice 5 On cherche à déterminer la transformée de Fourier de $s(t) = e^{-|t|}$.

1. Calculer la somme et l'énergie de ce signal.
2. On note $s_1(t) = s(t)\llbracket t \geq 0\rrbracket(t)$. Calculez la transformée de Fourier de $s_1(t)$ notée $\widehat{S}_1(f)$.
3. On note $s_2(t) = s(t)\llbracket t \leq 0\rrbracket(t)$. Calculez la transformée de Fourier de $s_2(t)$ notée $\widehat{S}_2(f)$.
4. On remarque $s(t) = s_1(t) + s_2(t)$ pour $t \neq 0$. Que peut-on en déduire sur la relation entre $\widehat{S}(f)$ et $\widehat{S}_1(f)$ et $\widehat{S}_2(f)$.
5. Déduisez $\widehat{S}(f)$.
6. En établissant le lien avec la première question, déterminez $\int_{-\infty}^{+\infty} \frac{1}{1+4\pi^2t^2} dt$.

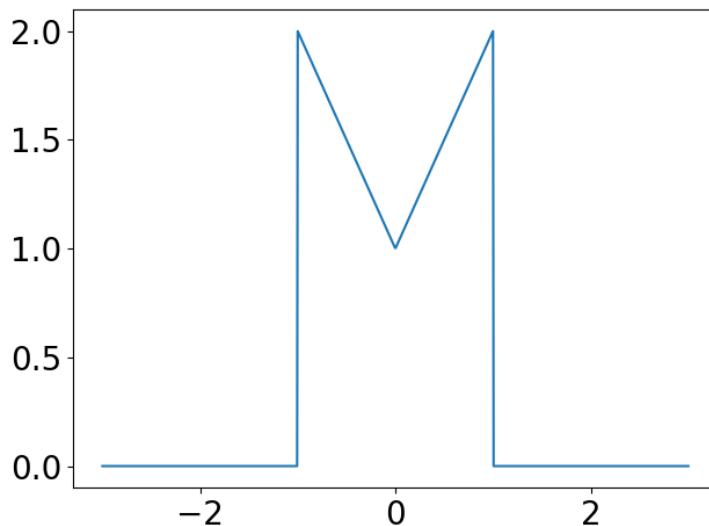


Figure 3.1: Visualisation du signal $x(t)$ (exercice 6).

Exercice 6 On considère le signal noté $x(t)$ et décrit par la figure 3.1. Donnez un pseudo-algorithme permettant de calculer sa transformée de Fourier.

```
t=np.linspace(-3,3,10**3)
x=2*seb.fonction_P(t/2)-seb.fonction_T(t)
plt,np = seb.debut()
fig,ax = plt.subplots()
ax.plot(t,x)
```

```

set.x_label('t')
set.y_label('x(t)')
plt.tight_layout()
fig.savefig('./figures/fig_exSEB25_fig1.png')
fig.show()

```

Exercice 7 On considère le signal $x(t) = e^{-|t|}$ dont la transformée de Fourier vaut $\widehat{X}(f) = \frac{2}{1+4\pi^2f^2}$. On considère

$$y(t) = \sum_{n=0}^{+\infty} \frac{x(t-n)}{2^n} \quad (3.1)$$

Montrez que la transformée de Fourier de $y(t)$ est

$$\widehat{Y}(f) = \frac{2}{1+4\pi^2f^2} \left(\frac{1}{1 - \frac{e^{-j2\pi f}}{2}} \right) \quad (3.2)$$

Chapter 4

Propriété de la transformée de Fourier

Exercice 8 Cet exercice cherche à illustrer la notion de parité.

1. On considère le signal $s(t) = e^{-|t|}$. Montrez que la transformée de Fourier de ce signal est à valeurs réelles.
2. En considérant différents fonctions de bases, proposez un algorithme montrant que la transformée de Fourier d'un signal pair est réel et que la transformée de Fourier d'un signal impair est imaginaire pur.

Exercice 9 On se donne des fonctions de bases et des tirages aléatoires. Montrez comment par simulation on peut confirmer que $\text{TF}[x(\frac{t}{a})](f) = a\hat{X}(af)$ pour $a > 0$.

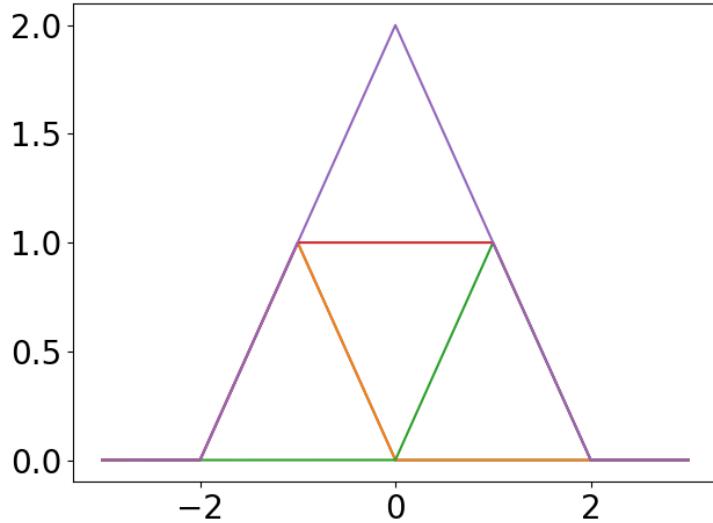


Figure 4.1: Figure associée à l'exercice 10.

Exercice 10 On considère le signal $x(t) = \mathbb{T}(t) = (1 - |t|)\llbracket -1 \leq t \leq 1 \rrbracket(t)$

1. En s'inspirant de la figure 4.1, montrez que

$$2x\left(\frac{t}{2}\right) = 2x(t) + x(t-1) + x(t+1) \quad (4.1)$$

2. Montrez en utilisant l'équation (4.1) que

$$2\hat{X}(2f) = \hat{X}(f) \left(2 + e^{-2j\pi f} + e^{2j\pi f} \right) \quad (4.2)$$

3. On remarque que $2 + e^{-2j\pi f} + e^{2j\pi f} = (e^{-j\pi f} + e^{j\pi f})^2 = 4 \cos^2(\pi f)$. En déduire l'expression de $\frac{\hat{X}(2f)}{\hat{X}(f)}$
4. On sait par ailleurs que $\hat{X}(f) = \frac{\sin(\pi f)}{\pi f}$, montrez que cette affirmation est compatible avec l'expression trouvée plus haut.

Simulation

```

plt,np = seb.debut()
t      = np.linspace(-3,3,10**4)
fig,ax = plt.subplots()
ax.plot(t,seb.fonction_T(t+1))
ax.plot(t,seb.fonction_T(t-1))
ax.plot(t,seb.fonction_T(t)+seb.fonction_T(t-1)+seb.fonction_T(t+1))
ax.plot(t,2*seb.fonction_T(t)+seb.fonction_T(t-1)+seb.fonction_T(t+1))
plt.tight_layout()
fig.savefig('./figures/fig_ex36_fig1.png')
fig.show()

```

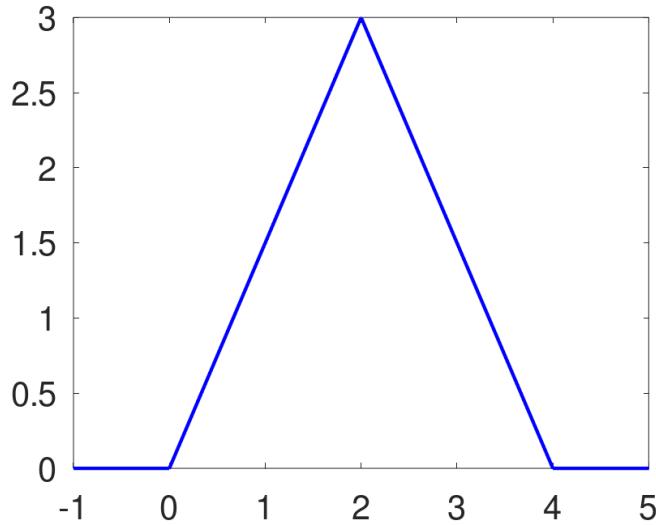


Figure 4.2: Graphe de $x(t)$ relatif à l'exercice 11.

Exercice 11 Le signal montré sur la figure 4.2 est noté $x(t)$. Sa transformée de Fourier est notée \widehat{X} .

1. Donnez une expression de $x(t)$ sous la forme de sa description sur plusieurs intervalles.
2. Donnez une expression de $x(t)$ en fonction de $\llbracket \cdot \rrbracket$.
3. Calculez $x(0)$, $x(1)$, E_x .
4. Calculez $\widehat{X}(0)$ et $\widehat{X}(1)$.
5. Construire $y_1(t) = x\left(\frac{t}{2}\right)$
6. Construire $y_1(t) = x(t - 1)$
7. Construire $y_1(t) = \frac{1}{2}x(t)$
8. Construire $y_1(t) = x(t) - x(t - 2)$

Simulation générant la figure 4.2 de l'exercice 11

```

t=linspace(-1,5,1e3);
x=3/2*t.*(t>=0).*(t<=2)+(4-t)*3/2.*(t>2).*(t<=4);
figure(1);
plot(t,x,'b-','linewidth',2);
set(gca,'fontsize',20);
saveas(1,'C:\A\SIMU\SEB\ex\exSEB1_fig1.png');

```

Solutions

1.

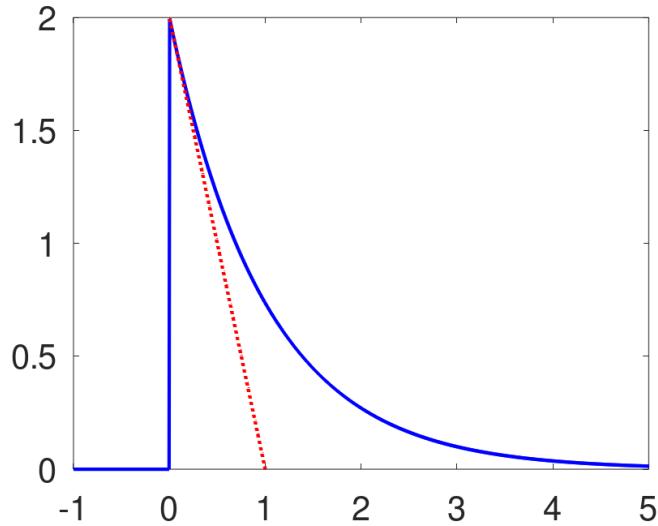


Figure 4.3: Graphe de $x(t)$ et de sa tangente pour l'exercice 12.

Exercice 12 Le signal montré sur la figure 4.3 est noté $x(t)$. Sa transformée de Fourier est notée \widehat{X} . Ce signal est de la forme $x(t) = ae^{-bt}\mathbf{1}(t \geq 0)$.

1. Justifiez la valeur de a avec la courbe exponentielle sur la figure 4.3.
2. Justifiez la valeur de b avec la ligne tangente à la courbe exponentielle sur la figure 4.3.
3. Donnez une expression de $x(t)$ en fonction de $\llbracket \cdot \rrbracket$.
4. Calculez $x(0)$, $x(1)$, E_x .
5. Calculez $\widehat{X}(0)$ et $\widehat{X}(1)$.
6. Construire $y_1(t) = x\left(\frac{t}{2}\right)$
7. Construire $y_1(t) = x(t - 1)$
8. Construire $y_1(t) = \frac{1}{2}x(t)$
9. Construire $y_1(t) = x(t) - x(t - 2)$

Simulation générant le graphe

```
t=linspace(-1,5,1e3);
x=2*exp(-t).*(t>=0);
t_tg=t((t>=0)&(t<=1));
x_tg=2-2*t_tg;
figure(1);
plot(t,x,'b-','linewidth',2,t_tg,x_tg,'r:','linewidth',2);
set(gca,'fontsize',20);
saveas(1,'C:\A\SIMU\SEB\ex\exSEB2_fig1.png');
```

Solutions

1.

Chapter 5

Diracs

Exercice 13 On considère le signal $x(t) = \Pi(t) = \llbracket -0.5 \leq t \leq 0.5 \rrbracket(t)$.

1. Calculez sa dérivée $y(t) = \frac{d}{dt}x(t)$.
2. Calculez $z(t) = \int_{-\infty}^t x(\tau) d\tau$.
3. Calculez la transformée de Fourier de $y(t)$ notée $\widehat{Y}(f)$ et en déduire celle de $x(t)$ notée $\widehat{X}(f)$.
4. Représentez les signaux $x(t), y(t), z(t)$.

Solution

1. $y(t) = \delta(t + 0.5) - \delta(t - 0.5)$
2. $z(t) = (t + 0.5)\llbracket -0.5 \leq t < 0.5 \rrbracket(t) + \llbracket 0.5 \leq t \rrbracket(t) = \mathbb{C}(t) + \mathbb{H}(t - 0.5)$
- 3.

$$\widehat{Y}(f) = \text{TF} [\delta(t + 0.5)](f) - \text{TF} [\delta(t - 0.5)](f) = e^{j\pi f} - e^{-j\pi f} = 2j \sin(\pi f) \quad (5.1)$$

Par conséquent,

$$\widehat{X}(f) = \frac{1}{j2\pi f} \widehat{Y}(f) = \frac{\sin(\pi f)}{\pi f} = \text{sinc}(f) \quad (5.2)$$

Exercice 14 On considère un oscillateur obtenu avec un comparateur (un amplificateur opérationnel monté en comparateur) et une capacité qui se charge et se décharge avec une résistance en fonction de la sortie du comparateur.

1. Proposez un montage électronique ou un schéma bloc.
 2. Donnez les équations de charge et décharge du condensateur.
 3. Donnez l'algorithme permettant de simuler le fonctionnement de cet oscillateur.
1. On utilise un amplificateur opérationnel monté en comparateur et réalisant

$$V_s = \begin{cases} +V_{cc} & \text{si } V_+ > V_- \\ -V_{cc} & \text{si } V_+ < V_- \end{cases} \quad (5.3)$$

où V_s est la tension à la sortie, $+V_{cc}$ et $-V_{cc}$ sont les tensions d'alimentation, V_+ et V_- sont les tensions aux entrées plus et moins.

On fixe la tension de référence du comparateur à $\frac{V_s}{2}$ grâce à un diviseur de tension, cette tension est relié à l'amplificateur sur la borne plus.

La tension mesurée par le comparateur avec sa borne moins est la tension du condensateur de capacité C qui est chargé et déchargé à partir de la tension de sortie de l'amplificateur par le biais d'une résistance R .

2. L'équation principale concerne la tension du condensateur :

$$C \frac{d}{dt} V_y = \frac{V_s - V_y}{R} \quad (5.4)$$

où V_y est la tension du condensateur. Celle-ci est continue. On a deux équations d'évolutions :

$$\begin{cases} \text{Tant que } V_y < \frac{V_{cc}}{2}, & V_s = V_{cc}, V_y(t) = V_{cc} + (V_y(t_0) - V_{cc})e^{-\frac{t-t_0}{RC}} \\ \text{Tant que } V_y > -\frac{V_{cc}}{2}, & V_s = -V_{cc}, V_y(t) = -V_{cc} + (V_y(t_0) + V_{cc})e^{-\frac{t-t_0}{RC}} \end{cases} \quad (5.5)$$

3. Le pseudo-programme est le suivant où y joue le rôle de V_y .

- Créer une échelle de temps t
- Créer un vecteur y de même taille que t
- Initialiser l'index `int` qui va parcourir y
- `While 1+ind < len(t)`
 - `While y[ind] < Vcc/2`
 - * Calcul de $y[ind+1]$ avec `Vs=+Vcc`
 - * Incrémentation de `ind`
 - `While y[ind] > -Vcc/2`
 - * Calcul de $y[ind+1]$ avec `Vs=-Vcc`
 - * Incrémentation de `ind`

Chapter 6

Transformées de Fourier et dérivation

Exercice 15 On considère le signal $x(t) = e^{-t} \llbracket t \geq 0 \rrbracket(t)$ et on note $\hat{X}(f)$ sa transformée de Fourier.

1. Calculez $\hat{X}(f)$.
2. Vérifiez qu'à partir de $\hat{X}(0) = 1$ vous retrouvez un résultat cohérent avec la définition de $x(t)$.
3. Un calcul mathématique ¹ montre que $\int_{-\infty}^{+\infty} \frac{dx}{1+x^2} = \pi$. Proposez un algorithme permettant de vérifier que ce résultat est correct.
4. Sachant que $x(t)$ est en fait réel, expliquez pourquoi

$$x(0) = \int_{-\infty}^{+\infty} \frac{df}{1 + 4\pi^2 f^2} \quad (6.1)$$

5. En utilisant le résultat précédent calculez la valeur théorique de $x(0) = \frac{1}{2}$.
6. Expliquez pourquoi on ne trouve pas le résultat $x(0) = 1$.

Solution

1. $\hat{X}(f) = \frac{1}{1+j2\pi f}$
2.

```
x=np.linspace(-100,100,10**8)
g=1/(1+x**2)
print(f"err={np.abs(seb.TF(x,g,0)-np.pi):.3g}")
err = 0.02
```

3.

$$x(0) = \operatorname{Re}(x(0)) = \operatorname{Re} \left[\int_{-\infty}^{+\infty} \frac{df}{1 + j2\pi f} \right] = \frac{1}{2} \int_{-\infty}^{+\infty} \left[\frac{1}{1 + j2\pi f} + \frac{1}{1 - j2\pi f} \right] df \quad (6.2)$$

4.

$$\int_{-\infty}^{+\infty} x(t) dt = \int_0^{+\infty} e^{-t} dt = 1 = \frac{1}{1 + j2\pi \times 0} = \hat{X}(0) \quad (6.3)$$

5. On fait le changement de variable $\omega = 2\pi f$, $d\omega = 2\pi df$.

6. On a bien

$$x(0) = \frac{1}{2} \lim_{t \rightarrow 0^-} + \frac{1}{2} \lim_{t \rightarrow 0^+} = \frac{1}{2} \times 0 + \frac{1}{2} \times 1 \quad (6.4)$$

¹Ce calcul utilise un simple changement de variable $x = \tan(\theta)$.

Exercice 16 On considère l'équation différentielle

$$\tau_2 \frac{d}{dt}y(t) + y(t) = e^{-\frac{t}{\tau_1}} \llbracket t \geq 0 \rrbracket(t) \quad (6.5)$$

On note la transformée de Fourier de $y(t)$, $\widehat{Y}(f)$.

1. Calculez $\widehat{Y}(f)$
2. Montrez que $\widehat{Y}(f)$ se met sous la forme

$$\widehat{Y}(f) = \frac{a}{1 + j2\pi f \tau_1} + \frac{b}{1 + j2\pi f \tau_2} \quad (6.6)$$

Trouvez les valeurs de a et b .

3. Donnez un algorithme permettant de confirmer la façon dont a et b dépendent de τ_1 et τ_2 en tirant aléatoirement des valeurs de τ_1 et τ_2 .
4. Donnez un algorithme permettant d'en déduire $y(t)$
5. Montrez que la solution de l'équation différentielle est

$$y(t) = \frac{\tau_1}{\tau_2 - \tau_1} \left(e^{-\frac{t}{\tau_1}} - e^{-\frac{t}{\tau_2}} \right) \llbracket t \geq 0 \rrbracket \quad (6.7)$$

Solution

1.

$$\widehat{Y}(f) = \frac{\tau_1}{1 + j2\pi f \tau_1} \frac{1}{1 + j2\pi f \tau_2} \quad (6.8)$$

2.

$$\frac{a}{1 + j2\pi f \tau_1} + \frac{b}{1 + j2\pi f \tau_2} = \frac{(a + b) + j2\pi f(\tau_1 b + a \tau_2)}{(1 + j2\pi f \tau_1)(1 + j2\pi f \tau_2)} \quad (6.9)$$

On en déduit que c'est possible si le système suivant a une solution

$$\begin{cases} a + b = \tau_1 \\ \tau_2 a + \tau_1 b = 0 \end{cases} \quad (6.10)$$

Ce système a une unique solution lorsque $\tau_1 \neq \tau_2$ et c'est $a = -\frac{\tau_1^2}{\tau_2 - \tau_1}$ et $b = \frac{\tau_2 \tau_1}{\tau_2 - \tau_1}$.

3.

```
import numpy.random as rd
import numpy.linalg as li
tau1,tau2 = rd.uniform(0,1),rd.uniform(0,1)
A=np.array([[tau1 ,tau2    ],
           [1      ,1      ]])
B=np.array([[tau1],
           [0      ]])
X=li.inv(A)@B
X_th=np.array([[-tau1/(tau2-tau1)], [tau2/(tau2-tau1)]])*tau1
print(f"err={np.sum(np.abs(X-X_th)):.3g}")
```
4.

```
plt.close('all')
import numpy.random as rd
tau1,tau2 = rd.uniform(0,1),rd.uniform(0,1)
X_th=np.array([[-tau1/(tau2-tau1)], [tau2/(tau2-tau1)]])*tau1
t=np.linspace(-1,5,10**2)
```

```

x_th=seb.val(X_th[0])*np.exp(-t/tau1)*(t>=0)/tau1+seb.val(X_th[1])*np.exp(-t/tau2)*(t>=0)/tau2
f=np.linspace(-50,50,10**3)
Y=lambda f: tau1/(1+1j*2*np.pi*f*tau1)/(1+1j*2*np.pi*f*tau2)
x=np.real(seb.TFI(f,Y(f),t))
fig,ax = plt.subplots()
ax.plot(t,x_th,label='th')
ax.plot(t,x,label='x')
ax.legend()
ax.set_xlabel('t')
plt.tight_layout()
fig.show()
print(f"err={np.max(x_th-x)}")

```

Chapter 7

Équations différentielles

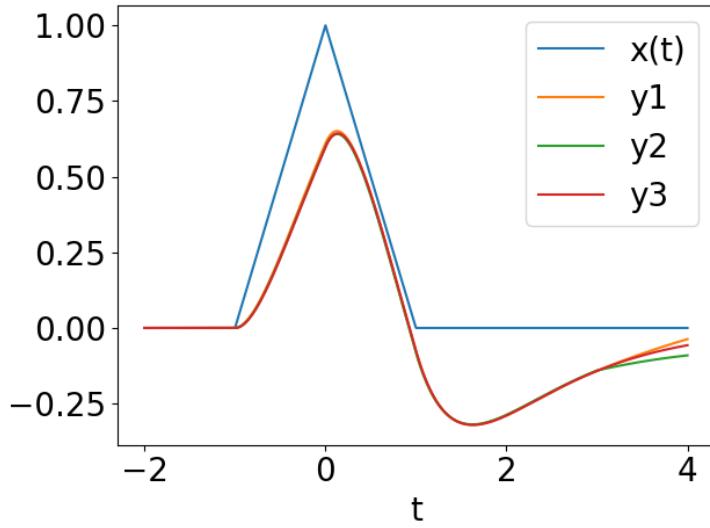


Figure 7.1: Visualisation de l'entrée $x(t)$ et de la sortie $y(t)$ illustrant l'exercice 17.

Exercice 17 On considère un filtre défini par l'équation différentielle

$$LC \frac{d^2}{dt^2}y(t) + RC \frac{d}{dt}y(t) + y(t) = RC \frac{d}{dt}x(t) \quad (7.1)$$

avec $R = 3$, $C = 0.5$, $L = 1$. On considère un signal en entrée défini par $x(t) = \mathbb{T}(t)$ et on cherche à simuler le signal de sortie $y(t)$ associé à ce filtre décrit par l'équation (7.1).

1. Montrez que

$$\frac{d}{dt}\mathbb{T}(t) = \int_{-\infty}^t [\delta(\tau - 1) - 2\delta(\tau) + \delta(\tau + 1)] d\tau \quad (7.2)$$

2. On appelle $\tilde{y}(t)$ la solution de cette deuxième équation différentielle

$$LC \frac{d^2}{dt^2}\tilde{y}(t) + RC \frac{d}{dt}\tilde{y}(t) + \tilde{y}(t) = \delta(t) \quad (7.3)$$

Exprimez $y(t)$ en fonction de $\tilde{y}(t)$.

- Cela peut se faire en considérant $a(t) = \tilde{y}(t-1)$, $b(t) = \tilde{y}(t+1)$ et $c(t) = a(t) + b(t) - 2\tilde{y}(t)$ et en calculant $LC \frac{d^2}{dt^2}a(t) + RC \frac{d}{dt}a(t) + a(t)$, $LC \frac{d^2}{dt^2}b(t) + RC \frac{d}{dt}b(t) + b(t)$ et $LC \frac{d^2}{dt^2}c(t) + RC \frac{d}{dt}c(t) + c(t)$.
- Cela peut se faire aussi en calculant d'une part $TF(\tilde{y}(t))(f)$ et d'autre part $TF(y(t))(f)$ à partir des équations différentielles puis en remarquant la ressemblance entre ces deux expressions, on peut calculer $\frac{TF(y(t))(f)}{TF(\tilde{y}(t))(f)}$, on peut trouver une relation entre $y(t)$ et $\tilde{y}(t)$.

3. En utilisant les fonctions `sol_eq_diff`, `deriver`, `integrer` et `retarder` de seb, donnez un pseudo-programme permettant de simuler $y(t)$.

Solution :

- On remarque que la fonction triangle dérivée une fois est une fonction porte avancée et une fonction porte retardée, (la porte étant définie $\Pi(t) = \llbracket |t| \leq 0.5 \rrbracket$).

$$\frac{d}{dt} \mathbb{T}(t) = \Pi(t + 0.5) - \Pi(t - 0.5) \quad (7.4)$$

Dérivée deux fois, ce sont trois, l'un avancé, le deuxième au milieu et un retardé.

$$\frac{d^2}{dt^2} \mathbb{T}(t) = \delta(t - 1) - 2\delta(t) + \delta(t + 1) \quad (7.5)$$

En intégrant cette expression, on trouve alors que

$$\frac{d}{dt} \mathbb{T}(t) = \int_{-\infty}^t [\delta(\tau - 1) - 2\delta(\tau) + \delta(\tau + 1)] d\tau \quad (7.6)$$

2.

$$y(t) = RC \int_{-\infty}^t [\tilde{y}(\tau - 1) - 2\tilde{y}(\tau) + \tilde{y}(\tau + 1)] d\tau \quad (7.7)$$

- Le pseudo-code est donné par

Algorithm 1 générant la figure 7.1.

Rentrer les valeurs de R,L,C

Créer une échelle de temps t entre -2 et 4 avec 1000 points

Calculer $\tilde{y}(t)$ en utilisant `sol_eq_diff` avec les coefficients LC, RC et 1 et l'échelle de temps t.

Utiliser `retarder` pour calculer $\tilde{y}_2(t) = \tilde{y}(t + 1) - 2\tilde{y}(t) + \tilde{y}(t - 1)$

Utiliser `integrer` pour calculer $y(t) = RC \int_{-\infty}^t \tilde{y}_2(\tau) d\tau$

```
def y(R,L,C,t):
    """réponse à une fonction triangle utilisant une équation différentielle"""
    import seb
    y_tilde=seb.sol_eq_diff((L*C,R*C,1),t)
    assert all(y_tilde[t<0]==0)
    y_tilde2=R*C*(seb.retarder(t,y_tilde,-1)-2*y_tilde+seb.retarder(t,y_tilde,1))
    y=seb.integrer(t,y_tilde2)
    return y

R,C,L = 3,0.5,1
t=np.linspace(-2,4,10**3)
fig,ax = plt.subplots()
ax.plot(t,seb.fonction_T(t),label='x(t)')
ax.plot(t,y(R,L,C,t),label='y')
ax.set_xlabel('t')
ax.legend()
plt.tight_layout()
fig.savefig('./figures/fig_exSeb11_fig1.png')
fig.show()
```

Exercice 18 On considère l'équation différentielle associée à une relation entrée-sortie :

$$\frac{d^2}{dt^2} y(t) + 3 \frac{d}{dt} y(t) + 2y(t) = \delta(t) \quad (7.8)$$

1. Écrivez le polynôme caractéristique.
2. Trouvez les solutions de ce polynôme.
3. En déduire la solution $y(t)$.
4. Calculez $\widehat{Y}(f)$ à partir de la solution trouvée
5. Comparez avec $\widehat{Y}(f)$ calculée à partir de l'équation différentielle.

Solution

1. Le polynôme caractéristique est $p^2 + 3p + 1$
2. Ses racines sont $p = -1$ et $p = -2$.
3. Je pose $p_1 = -1$ et $p_2 = -2$, je propose comme solution $y(t) = c_1 e^{p_1 t} \llbracket t \geq 0 \rrbracket + c_2 e^{p_2 t} \llbracket t \geq 0 \rrbracket$. Après calcul, je trouve

$$\begin{aligned} \frac{d^2}{dt^2}y(t) + 3\frac{d}{dt}y(t) + 2y(t) &= (p_1^2 + 3p_1 + 2)c_1 e^{p_1 t} + (p_2^2 + 3p_2 + 2)c_2 e^{p_2 t} \\ &\quad + ((p_1 + 3)c_1 + (p_2 + 3)c_2)\delta(t) + (c_1 + c_2)\delta'(t) \end{aligned} \quad (7.9)$$

Par identification, on obtient le système suivant

$$\begin{cases} c_1 + c_2 = 0 \\ 2c_1 + c_2 = 1 \end{cases} \quad (7.10)$$

La solution de ce système est $c_1 = 1$ et $c_2 = -1$. D'où $y(t) = (e^{-t} - e^{-2t}) \llbracket t \geq 0 \rrbracket$

4.

$$\widehat{Y}(f) = \text{TF}[(e^{-t} - e^{-2t}) \llbracket t \geq 0 \rrbracket](f) = \frac{1}{1 + j2\pi f} - \frac{1}{2 + j2\pi f} \quad (7.11)$$

Après calcul, on trouve

$$\widehat{Y}(f) = \frac{1}{2 + 3(j2\pi f) + (j2\pi f)^2} \quad (7.12)$$

5. Il se trouve qu'en convertissant les dérivées en $j2\pi f$ dans l'équation (7.8), on trouve

$$(j2\pi f)^2 \widehat{Y}(f) + 3(j2\pi f) \widehat{Y}(f) + 2\widehat{Y}(f) = 1 \quad (7.13)$$

Ce qui s'avère être la même expression que précédemment.

Chapter 8

Filtres et effet mémoire

Exercice 19 Dans cet exercice, on cherche à montrer par simulation que

$$\Pi(t) * \Pi(t) = \mathbb{T}(t) \quad (8.1)$$

où $\Pi(t) = [\|t| \leq 0.5](t)$ et $\mathbb{T}(t) = (1 - |t|)[\|t| \leq 1](t)$.

1. Montrez que $\text{TF}[\Pi(t) * \Pi(t)](f) = \text{sinc}^2(f)$ où $\text{sinc}(f) = \frac{\sin(\pi f)}{\pi f}$.
2. Proposez un algorithme utilisant la transformée de Fourier pour montrer l'équation (8.1).
3. Donnez un autre algorithme utilisant le produit de convolution pour démontrer aussi l'équation (8.1).

Solution

1. On sait d'après le cours que $\text{TF}[\Pi(t)](f) = \text{sinc}(f)$.

$$\text{TF}[\Pi(t) * \Pi(t)](f) = \text{TF}[\Pi(t)](f)\text{TF}[\Pi(t)](f) = \text{sinc}^2(f) \quad (8.2)$$

2. L'algorithme proposé utilise le fait qu'on sait d'après le cours que $\text{TF}[\Pi(t)](f) = \text{sinc}(f)$:

Créer une échelle de temps `tx` entre -2 et 2 avec 1000 points

Calculer `x` associé à `tx` en utilisant la fonction `fonction_T` de `seb.py`.

Créer une échelle de fréquence `f` entre -3 et 3 .

Calculez la transformée de Fourier de `x` appelé `X`.

Calculez `X_th` défini par $\hat{X}_{\text{th}}(f) = \text{sinc}^2(f)$.

Comparez `X` avec `X_th` en calculant le maximum de la valeur absolue de la différence.

Algorithm 2: associé à l'exercice 19

3. Créer une échelle de temps `tx` entre -2 et 2 avec 1000 points
Calculer `x` associé à `tx` en utilisant la fonction `fonction_P` de `seb.py`.
Utilisez `convolution` de `seb.py` pour en déduire $x'(t) = \Pi(t) * \Pi(t)$ sur l'échelle `tx`.
Comparer en calculant le maximum de la valeur absolue de la différence entre $x'(t)$ et $\mathbb{T}(t)$.

Algorithm 3: associé à l'exercice 19

```
tx=np.linspace(-2,2,1000)
x=seb.fonction_P(tx)
xp=seb.convolution(tx,x,tx,x,tx)
```

Exercice 20 Dans cet exercice, on cherche à montrer par simulation que

$$\Pi(t) * \Pi(t) = \mathbb{T}(t) \quad (8.3)$$

où $\Pi(t) = [\|t| \leq 0.5](t)$ et $\mathbb{T}(t) = (1 - |t|)[\|t| \leq 1](t)$.

1. On note $s(t) = \Pi(t) * \Pi(t)$, donnez une expression intégrale à $s(t)$.
2. Montrez que pour $t < -1$, $s(t) = 0$.
3. Montrez que $s(-t) = s(t)$ et que donc $s(t)$ est un signal pair.
4. En déduire que pour $t > 1$, $s(t) = 0$.
5. Montrez que $s(0) = 1$.
6. Montrez que $s(t) = 1 - t$ pour $t \in [0, 1]$.
7. Déduisez que $s(t) = \mathbb{T}(t)$.

Solution

1.

$$s(t) = \int_{-\infty}^{+\infty} \Pi(t - \tau) \Pi(\tau) d\tau \quad (8.4)$$

```
2. import seb
plt,np = seb.debut()
def s(t):
    tau = np.linspace(-20,20,10**4)
    x = seb.fonction_P(tau)*seb.fonction_P(t-tau)
    return seb.TF(tau,x,0)
import numpy.random as nr
K = 10**4
for k in range(K):
    t = nr.uniform(-100,-1)
    assert np.abs(s(t))<0.01
```

3. On rajoute

```
K = 10**4
for k in range(K):
    t = nr.uniform(-100,100)
    assert np.abs(s(t)-s(-t))<0.01
```

4. On rajoute `print(1-s(0))`

5. On rajoute

```
K = 10**4
for k in range(K):
    t = nr.uniform(0,1)
    assert np.abs(s(t)-(1-t))<0.01
```

6. On rajoute

```
K = 10**4
for k in range(K):
    t = nr.uniform(-100,100)
    assert np.abs(s(t)-seb.fonction_T(t))<0.01
```

Exercice 21 On cherche à trouver la réponse impulsionnelle associée à

$$y(t) = \int_{t-3}^t x(\tau) d\tau \quad (8.5)$$

1. Montrez que l'équation (8.5) peut se mettre sous la forme

$$y(t) = \int_{-\infty}^{+\infty} \llbracket t - 3 \leq \tau \leq t \rrbracket(\tau) x(\tau) d\tau \quad (8.6)$$

Mathématiquement, voici la définition d'une intégrale de a à b de $f(t)$

$$\int_a^b f(t) dt = \int_{-\infty}^{+\infty} \llbracket a \leq t \leq b \rrbracket(t) f(t) dt \quad (8.7)$$

2. En s'appuyant sur l'équation (8.6), trouvez $h(t)$ tel que

$$y(t) = \int_{-\infty}^{+\infty} h(t - \tau) x(\tau) d\tau \quad (8.8)$$

En déduire la réponse impulsionale $h(t)$.

3. On suppose maintenant que $x(t) = \delta(t)$. Trouvez une nouvelle façon de déterminer la réponse impulsionale en considérant successivement $t < 0$, $t \in [0, 1]$ et $t > 1$.

Exercice 22 On considère l'équation différentielle associée à une relation entrée-sortie :

$$\frac{d^2}{dt^2}y(t) + 3\frac{d}{dt}y(t) + 2y(t) = x(t) \quad (8.9)$$

1. Donnez la réponse fréquentielle.
2. Donnez un algorithme donnant la réponse impulsionale.
3. Écrivez le polynôme caractéristique.
4. Trouvez les solutions de ce polynôme.
5. En déduire la réponse impulsionale.

Soluton

1.

$$\hat{H}(f) = \frac{1}{2 + 3(j2\pi f) + (j2\pi f)^2} \quad (8.10)$$

2. `t = np.linspace(-2, 5, 200)`
`h = seb.sol_eq_diff((1, 3, 2), t)`
3. Le polynôme caractéristique est $p^2 + 3p + 1$
4. Ses racines sont $p = -1$ et $p = -2$.
5. Je pose $p_1 = -1$ et $p_2 = -2$, je propose comme solution $h(t) = c_1 e^{p_1 t} \llbracket t \geq 0 \rrbracket + c_2 e^{p_2 t} \llbracket t \geq 0 \rrbracket$. Après calcul, je trouve

$$\begin{aligned} \frac{d^2}{dt^2}h(t) + 3\frac{d}{dt}h(t) + 2h(t) &= (p_1^2 + 3p_1 + 2)c_1 e^{p_1 t} + (p_2^2 + 3p_2 + 2)c_2 e^{p_2 t} \\ &\quad + ((p_1 + 3)c_1 + (p_2 + 3)c_2)\delta(t) + (c_1 + c_2)\delta'(t) \end{aligned} \quad (8.11)$$

Par identification, on obtient le système suivant

$$\begin{cases} c_1 + c_2 = 0 \\ 2c_1 + c_2 = 1 \end{cases} \quad (8.12)$$

La solution de ce système est $c_1 = 1$ et $c_2 = -1$. D'où $h(t) = (e^{-t} - e^{-2t}) \llbracket t \geq 0 \rrbracket$

Chapter 9

Description fréquentielle des filtres

Exercice 23 On considère un filtre de réponse impulsionale

$$h(t) = \cos(2\pi t)e^{-t} \llbracket t \geq 0 \rrbracket(t) \quad (9.1)$$

Ce filtre est un passe-haut. Donnez un algorithme permettant de trouver les deux fréquences de coupure et sa bande passante.

Solution :

- Créer une échelle de fréquence f entre -5 et 5 avec 10^4 valeurs de fréquences
- Créer une échelle de temps t entre 0 et 100 avec 10^4 points.
- Calculer le module de la transformée de Fourier notée $|\hat{H}(f)|$ avec **séb.TF** et la réponse impulsionale
- Trouver la fréquence f_{\max} et la valeur du module en f_{\max} notée $|\hat{H}_{\max}|$.
- Trouver la fréquence f_0 entre 0 et f_{\max} qui minimise la valeur absolue de la différence entre $\frac{|\hat{H}_{\max}|}{\sqrt{2}}$ et $|\hat{H}(f)|$
- Trouver la fréquence f_2 entre f_{\max} et $+\infty$ qui minimise la valeur absolue de la différence entre $\frac{|\hat{H}_{\max}|}{\sqrt{2}}$ et $|\hat{H}(f)|$
- La bande passante est $f_2 - f_1$.

Exercice 24 On considère un filtre de réponse impulsionale

$$h(t) = \cos(2\pi t)e^{-|t|} \quad (9.2)$$

1. Montrer que la réponse fréquentielle de ce filtre est

$$\hat{H}(f) = \frac{1}{1 + 4\pi^2(f - 1)^2} + \frac{1}{1 + 4\pi^2(f + 1)^2} \quad (9.3)$$

Pour cela vous pouvez utiliser le fait que $\cos(2\pi t) = \frac{1}{2}e^{-j2\pi t} + \frac{1}{2}e^{j2\pi t}$ et que quand z est un complexe,

$$\frac{1}{z} + \frac{1}{\bar{z}} = \frac{2\Re(z)}{|z|^2}$$

2. Pourquoi en observant $h(t)$, on pouvait savoir que $\hat{H}(f) = |\hat{H}(f)|$

3. En observant l'équation (9.3), montrez trouvez la valeur de $f > 0$ qui maximise $|\hat{H}(f)|$.

4. On considère maintenant

$$|\hat{H}_1(f)| = \frac{1}{1 + 16\pi^2} + \frac{1}{1 + 4\pi^2(f - 1)^2} \quad (9.4)$$

Montrez que $|\hat{H}_1(f)|$ est une bonne approximation de $|\hat{H}(f)|$ autour de $f = 1$.

5. En utilisant cette nouvelle approximation, calculez les deux fréquences de coupures et la bande passante.

Exercice 25 On considère un filtre défini par

$$y(t) = \int_{t-3}^t x(\tau) d\tau = \int_{-\infty}^{+\infty} \llbracket t-3 \leq \tau \leq t \rrbracket(\tau) x(\tau) d\tau \quad (9.5)$$

La troisième expression étant une façon de donner un sens à la deuxième expression.

1. Calculez par rapport à τ , la dérivée de

$$z(\tau) = \llbracket t-3 \leq \tau \leq t \rrbracket(\tau) \quad (9.6)$$

t étant fixe dans ce calcul de dérivée.

2. Déduisez que

$$\frac{d}{dt} y(t) = x(t) - x(t-3) \quad (9.7)$$

3. Déduisez $\hat{H}(f)$.

4. On suppose maintenant qu'on connaisse la réponse impulsionale de ce filtre $h(t)$ défini par $\llbracket 0 \leq t \leq 3 \rrbracket(t)$, déduisez-en un autre de calcul de $\hat{H}(f)$.

Exercice 26 On considère un filtre dont la réponse fréquentielle vérifie

$$\hat{H}(f) = \frac{j2\pi f RC}{1 - 4\pi^2 f^2 + 4jRC\pi f} \quad (9.8)$$

1. Trouvez l'équation différentielle associée à la relation entrée-sortie ?
2. Trouvez l'équation différentielle associée à la réponse impulsionale ?
3. Proposez un algorithme permettant de calculer la réponse impulsuelle.

Chapter 10

Signaux périodiques

Exercice 27 On considère un signal non-périodique défini par $x(t) = e^{-t} \llbracket 0 \leq t < 1 \rrbracket(t)$ et un signal périodique obtenu en périodisant $x(t)$.

$$y(t) = \sum_{k=-\infty}^{+\infty} x(t-k) \quad (10.1)$$

1. Donnez l'algorithme permettant de tracer $y(t)$ pour $t \in [-3, 3]$.
2. Donnez l'algorithme permettant d'estimer M_y et P_y .
3. Donnez l'algorithme permettant de calculer la série de Fourier associée à $y(t)$.
4. Donnez un algorithme permettant de vérifier expérimentalement que $P_y = \sum_{k=-\infty}^{+\infty} |\hat{Y}_k|^2$ et que $M_y = \hat{Y}_0$ non pas seulement pour ce signal spécifiquement mais pour des signaux construits à partir de $x(t)$ et tirés aléatoirement.

Exercice 28 On considère un signal non-périodique défini par $x(t) = e^{-t} \llbracket 0 \leq t < 1 \rrbracket(t)$ et un signal périodique obtenu en périodisant $x(t)$.

$$y(t) = \sum_{k=-\infty}^{+\infty} x(t-k) \quad (10.2)$$

1. Représentez graphiquement $x(t)$ et $y(t)$ pour $t \in [-3, 3]$.
2. Calculez A_x et en déduire M_y .
3. Calculez E_x et en déduire P_y .
4. Montrez que les coefficients de la série de Fourier sont

$$\hat{Y}_k = \frac{1 - e^{-1}}{1 + j2\pi k} \quad (10.3)$$

5. En utilisant le fait que

$$\sum_{k=-\infty}^{+\infty} \frac{1}{1 + 4\pi^2 k^2} = \frac{1}{2} \frac{e+1}{e-1} \quad (10.4)$$

montrez qu'on retrouve le résultat précédent $P_y = \frac{1}{2} \frac{e^2 - 1}{e^2}$.

Solution :

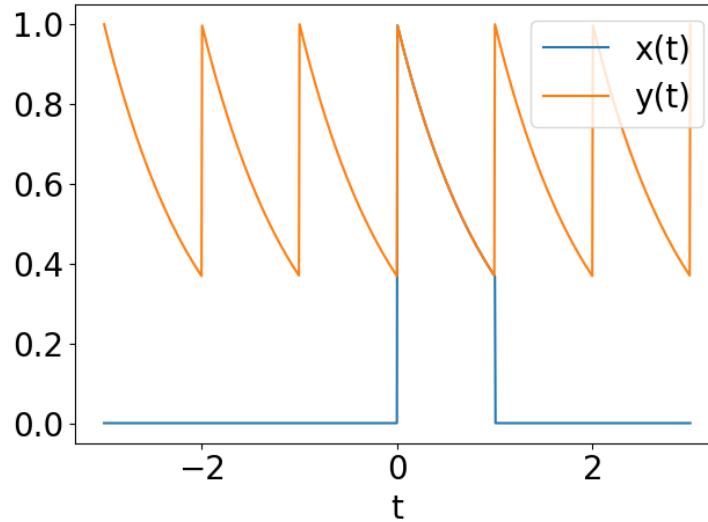


Figure 10.1: Graphe de $x(t)$ et $y(t)$ correspondant à l'exercice 28.

1. La figure 10.1 montre $x(t)$ en bleu et $y(t)$ en orange.

2.

$$A_x = \int_{-\infty}^{+\infty} x(t) dt = \int_0^1 e^{-t} dt = \frac{e-1}{e} \quad (10.5)$$

Il se trouve que $M_x = \int_0^1 e^{-t} dt = A_x$.

3.

$$E_x = \int_{-\infty}^{+\infty} x^2(t) dt = \int_0^1 e^{-2t} dt = \frac{1}{2} \frac{e^2 - 1}{e^2} \quad (10.6)$$

Il se trouve que $P_y = \int_0^1 e^{-2t} dt = E_x$.

4. On remarque que comme la période est de $T = 1$, $e^{j2\pi k} = 1$.

$$\hat{Y}_k = \int_0^1 e^{-t} e^{-j2\pi k} dt = \frac{1 - e^{-1}}{1 + j2\pi k} \quad (10.7)$$

```
import seb
plt,np = seb.debut()
tx=np.linspace(-3,3,10**3)
x=np.exp(-tx)*(tx>=0)*(tx<=1)
ty=seb.periodiser_ech_t(tx,(0,1))
assert all(ty>=0)&all(ty<=1)
y=np.exp(-ty)
plt.close('all')
fig,ax=plt.subplots()
ax.plot(tx,x,label='x(t)')
ax.plot(tx,y,label='y(t)')
ax.set_xlabel('t')
ax.legend()
plt.tight_layout()
fig.savefig('../figures/fig_exSEB22_fig1a.png')
fig.show()
ty2=tx[(tx>=0)*(tx<1)]
My=seb.TF(ty2,np.exp(-ty2),0)
```

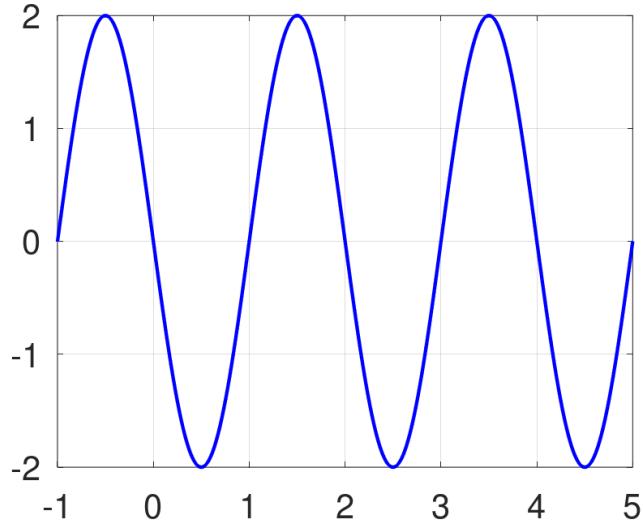


Figure 10.2: Graphe de $x(t)$ relatif à l'exercice 31.

```

k=np.arange(-10**3,10**3)
e=np.exp(1)
print(np.sum(1/(1+4*np.pi**2*k**2))-(e+1)/(e-1))
"""
2.0.0.
alors que numpy.__version__ 0.24.2
"""

```

Exercice 29 On considère un signal $x(t) = \llbracket 0 \leq t \leq 2 \rrbracket(t)$. On définit deux autres signaux à partir de $x(t)$.

$$y(t) = \sum_{k=-\infty}^{+\infty} x(t-3k) \text{ et } z(t) = \sum_{k=-\infty}^{+\infty} x(t-4k) \quad (10.8)$$

1. $x(t), y(t), z(t)$ sont-ils périodiques et si oui, quelles sont leurs périodes ?
2. Calculez M_y et P_y la moyenne et la puissance de $y(t)$.
3. Calculez M_z et P_z la moyenne et la puissance de $z(t)$.

Exercice 30 On considère un signal $x(t) = \llbracket 0 \leq t \leq 2 \rrbracket(t)$ dont la transformée de Fourier est notée \hat{X} . On définit un autre signal périodique à partir de $x(t)$.

$$y(t) = \sum_{k=-\infty}^{+\infty} x(t-3k) \quad (10.9)$$

Sa transformée de Fourier est notée \hat{Y} . On utilise dans les questions suivantes que

$$TF[\llbracket -0.5 \leq t \leq 0.5 \rrbracket(t)](f) = \frac{\sin(\pi f)}{\pi f} \quad (10.10)$$

1. Calculez \hat{X} .
2. Calculez \hat{Y} .

Exercice 31 Le signal montré sur la figure 10.2 est noté $x(t)$. Sa transformée de Fourier est notée \hat{X} . Ce signal est de la forme $x(t) = a \cos(bt + c)$.

1. $x(t)$ est-il un signal temps continu, temps discret, périodique, non-périodique, déterministe ou aléatoire.
2. Justifiez la valeur de a en observant la valeur maximale et minimale sur la figure 10.2.
3. Justifiez la valeur de b en mesurant la période sur la figure 10.2.
4. Justifiez la valeur de c en interprétant cette courbe comme en retard (ou en avance) par rapport à $a \cos(bt)$ sur la figure 4.3.
5. Calculez $x(0)$, $x(1)$, P_x .
6. Calculez \hat{X}_0 et \hat{X}_1 .
7. Dessiner sur le graphe $y_1(t) = x\left(\frac{t}{2}\right)$
8. Dessiner sur le graphe $y_1(t) = x(t - 1)$
9. Dessiner sur le graphe $y_1(t) = \frac{1}{2}x(t)$
10. Dessiner sur le graphe $y_1(t) = x(t) - x(t - 2)$

Simulation générant le graphe

```
t=linspace(-1,5,1e3);
x=2*cos(pi*t+pi/2);
figure(1);
plot(t,x,'b-','linewidth',2);
grid;
set(gca,'fontsize',20);
saveas(1,'C:\A\SIMU\SEB\ex\exSEB4_fig1.png');
```

Solutions

- 1.
2. $a = 2$
3. $b = \pi$
4. $c = \frac{\pi}{2}$.

Chapter 11

Filtres agissant sur des signaux périodiques

Exercice 32 On considère un signal défini par

$$x(t) = \Pi(t) - \frac{1}{2}\Pi\left(\frac{t}{3}\right) \quad (11.1)$$

On note $y(t)$ le signal périodisé en répétant l'intervalle $[-\frac{3}{2}, \frac{3}{2}]$. On considère le filtre défini par l'équation différentielle

$$\frac{d^2}{dt^2}z(t) + \frac{d}{dt}z(t) + y(t) = x(t) \quad (11.2)$$

On note $z(t)$ la sortie de ce filtre quand l'entrée est $y(t)$.

1. Représentez graphiquement $y(t)$ pour $t \in [-3, 3]$.
2. Montrez que $\widehat{X}(f) = \text{sinc}(f) - \frac{3}{2}\text{sinc}(3f)$.
3. En déduire que $\widehat{Y}_k = \frac{1}{3}\text{sinc}(\frac{k}{3}) - \frac{1}{2}\delta_k$, δ_k étant la suite nulle sauf en $k = 0$ ou elle vaut 1.
4. Calculez la réponse fréquentielle du filtre
5. En déduire la \widehat{Z}_k les coefficients de la série de Fourier de $z(t)$.
6. Proposez une approximation de $z(t)$.

Solution :

1. $x(t)$ vaut -0.5 sur $[-1.5, -0.5]$ et sur $[0.5, 1.5]$. Le signal vaut 0.5 sur $[-0.5, 0.5]$, et il est nul ailleurs. $y(t)$ est obtenu graphiquement en répétant ce signal à partir de son représentant défini sur $[-1.5, 1.5]$.
2. D'après le cours, $\text{TF}[\Pi(t)](f) = \text{sinc}(f)$, ce qui entraîne par dilatation de l'échelle des temps $\text{TF}[\Pi(t/3)](f) = 3\text{sinc}(3f)$. Par linéarité, on a $\widehat{X}(f) = \text{sinc}(f) - \frac{3}{2}\text{sinc}(3f)$.
3. Le cours donne une relation entre \widehat{Y}_k et $\widehat{X}(f)$

$$\widehat{Y}_k = \frac{1}{T}\widehat{X}\left(\frac{k}{3}\right) \quad (11.3)$$

En particulier, $\widehat{Y}_0 = \frac{1}{3}\widehat{X}(0) = -\frac{1}{3}$. Pour les autres valeurs de k , on remarque que $\text{sinc}\left(3\frac{k}{3}\right) = 0$ Ce qui donne finalement le résultat souhaité.

4. $\widehat{H}(f) = \frac{1}{-4\pi^2 f^2 + j2\pi f + 1}$
5. Le filtrage de signaux périodiques permet d'affirmer que

$$\widehat{Z}_k = \widehat{H}\left(\frac{k}{3}\right)\widehat{Y}_k \quad (11.4)$$

6. Je propose de considérer l'approximation

$$z(t) = \widehat{Z}_{-2}e^{-j4\pi t} + \widehat{Z}_{-1}e^{-j2\pi t} + \widehat{Z}_0 + \widehat{Z}_1e^{j2\pi t} + \widehat{Z}_2e^{j4\pi t} \quad (11.5)$$

La raison est qu'au delà de $|k| = 2$, \widehat{Z}_k devient très petit. En effet le dénominateur est une fonction décroissante de x quand on substitue $\pi^2 k^2$ par x dès que x est plus grand que π (et en fait bien avant). Cette décroissance s'observe en dérivant par rapport à x cette fonction.

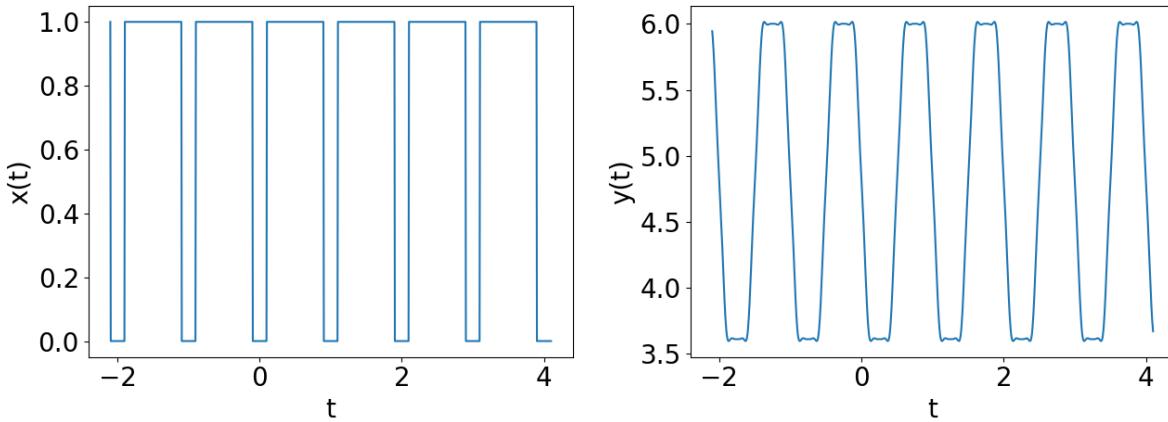


Figure 11.1: Courbe $x(t)$ et $y(t)$ pour l'exercice 11.1.

Exercice 33 La figure 11.1 représente le signal $x(t)$. On considère le filtre $y(t) = \int_{t-0.5}^t x(\tau) d\tau$ de réponse fréquentielle $\widehat{H}(f) = \frac{\sin(\pi f/2)}{\pi f} e^{-j\pi f/2}$.

1. Donnez une définition de $x(t)$ avec des équations, vous pouvez utiliser le fait que $\widehat{X}_0 = \frac{4}{5}$.
2. Calculez les coefficients de la série de Fourier $\widehat{X}_{-1}, \widehat{X}_0, \widehat{X}_1$. À quelle fréquences ces coefficients correspondent.
3. Expliquez une approximation permettant de calculer $y(t)$ en utilisant $\widehat{X}_{-1}, \widehat{X}_0, \widehat{X}_1$.
4. Calculez $y(t)$ avec cette approximation.
5. Donnez un algorithme permettant de calculer $y(t)$ sans utiliser cette approximation.

Simulation

```

plt,np = seb.debut()
t = np.linspace(-2.1,4.1,10**3)
tp = seb.periodiser_ech_t(t,1)
x = (0.1<=tp)*(tp<0.9)
plt.close('all')
fig,ax = plt.subplots()
ax.plot(t,x)
ax.set_xlabel('t')
ax.set_ylabel('x(t)')
plt.tight_layout()
fig.savefig('./figures/fig_exSEB42a.png')
fig.show()

k = np.arange(-10,10)
fk,Xk = seb.coef_serie_Fourier(tp,x,1,k)
Hk = np.sinc(fk/2)*np.exp(-1j*np.pi*fk/2)
Yk = Hk*Xk
y = np.zeros(len(t),dtype=complex)
for k_ind in range(len(k)):

```

```

k_val = k[k_ind]
y += Yk[k_ind]*np.exp(1j*2*np.pi*k_val*t)

y = np.real(y)
fig,ax = plt.subplots()
ax.plot(t,y)
ax.set_xlabel('t')
ax.set_ylabel('y(t)')
plt.tight_layout()
fig.savefig('./figures/fig_exSEB42b.png')
fig.show()

```

1. D'après le graphe on a $x(t)$ est périodique de période 1 et que pour $t \in [0, 1]$, $x(t) = \llbracket a \leq t \mathbf{1} - a \rrbracket(t)$. $\frac{4}{5} = \hat{X}_0$ est la moyenne de $x(t)$ et vaut donc $(1 - a - a)/1 = 1 - 2a$, donc $a = \frac{1}{10}$.
2. Les fréquences associées sont -1Hz , 0Hz , 1Hz et les valeurs sont

$$\hat{X}_1 = \int_a^{1-a} e^{-j2\pi t} dt = \frac{1}{2\pi j} (-e^{j2\pi a} + e^{-j2\pi a}) = -\frac{1}{\pi} \sin\left(\frac{\pi}{5}\right) \quad (11.6)$$

$x(t)$ étant pair et réel, on en déduit que \hat{X}_k est réel et pair et $\hat{X}_{-1} = \hat{X}_1$.

3. En ne considérant que $\hat{X}_{-1}, \hat{X}_0, \hat{X}_1$, on obtient une approximation de $y(t)$ en considérant que

$$\hat{Y}_k = \hat{H}\left(\frac{k}{1}\right) \hat{X}_k \text{ pour } k \in \{-1, 0, 1\} \quad (11.7)$$

4. En plaçant en entrée un Dirac, on en déduit que $y(t)$ vaut d'une part $h(t)$ la réponse impulsionnelle du filtre et d'autre part $\llbracket 0 \leq t \leq 0.5 \rrbracket(t)$. Le calcul de la réponse fréquentielle se fait à partir de $h(t)$

$$\hat{H}(f) = \text{TF}[\llbracket 0 \leq t \leq 0.5 \rrbracket](f) = \text{TF}[\Pi(2)](f) e^{-j2\pi f \frac{1}{4}} = \frac{1}{2} \text{TF}[\Pi(t)]\left(\frac{f}{2}\right) e^{-j\frac{\pi f}{2}} = \frac{1}{2} e^{-j\frac{\pi f}{2}} \text{sinc}\left(\frac{f}{2}\right) \quad (11.8)$$

Après calcul, on a $\hat{H}(0) = \frac{1}{2}$, $\hat{H}(1) = \frac{1}{j\pi}$, $\hat{H}(-1) = \frac{-1}{j\pi}$. En utilisant la série de Fourier,

$$y(t) = \hat{Y}_0 + \hat{Y}_1 e^{j2\pi t} + \hat{Y}_{-1} e^{-j2\pi t} = \frac{2}{5} + \frac{2}{\pi^2} \sin(\pi/5) \sin(2\pi t) \quad (11.9)$$

```

5. plt,np = seb.debut()
K = 20
k = np.arange(-K,K)
t = seb.linspace(0,1,10**3)
x = (t>=1/10)&(t<=9/10)+0
h = (t>=0)&(t<=0.5)
fk, Xk= seb.coef_serie_Fourier(t,x,1,k)
Hk= seb.TF(t,h,k)
Yk= Xk*Hk
y = np.zeros(len(t))
for k_ in range(len(k)):
    y = y + Yk[k_]*np.exp(1j*2*np.pi*k_[k_]*t)
y = np.real(y)

```

Chapter 12

Échantillonnage d'un signal non-périodique

Exercice 34 On considère le signal $x(t) = e^{-t} \llbracket t \geq 0 \rrbracket(t)$ et on souhaite illustrer la question du repliement spectral.

1. Donnez un algorithme permettant de simuler y_n le signal $x(t)$ échantillonné à la fréquence f_e .
2. Donnez un algorithme permettant de simuler $\hat{Y}(f)$ la transformée de Fourier de y_n .
3. Donnez un algorithme permettant de simuler $\hat{Z}(f)$, défini par

$$\hat{Z}(f) = \left| \frac{1}{f_e} \hat{Y}(f) - \hat{X}(f) - \hat{X}(f - f_e) - \hat{X}(f + f_e) \right| \quad (12.1)$$

$x(t)$ présente une discontinuité en $t = 0$, aussi il est nécessaire pour la valeur de $x(t)$ en $t = 0$ d'utiliser

$$\frac{1}{2} \lim_{t \rightarrow 0^-} x(t) + \frac{1}{2} \lim_{t \rightarrow 0^+} x(t) = \frac{1}{2} \quad (12.2)$$

Solution

```

1. def z(t):
    return np.exp(-t)*(t>=0)+0

T  = 10
Te = 1
te = seb.arange(0,T+Te,Te)
yn = z(te)
yn[0] = 0.5 #on corrige l'expression

2. fe = 1/Te
f   = np.linspace(-fe/2,fe/2,200)
Y  = np.real(seb.TFTD(te,yn,f))

3. X1 = seb.TF(t,x,f)+seb.TF(t,x,f+fe)+seb.TF(t,x,f-fe)
Z  = np.abs(1/fe*Y-np.real(X1))

```

Exercice 35 On considère le signal $x(t) = e^{-t} \llbracket t \geq 0 \rrbracket(t)$ et on souhaite calculer la transformée de Fourier du signal échantillonné. On considère f_e une fréquence d'échantillonnage. On note y_n le signal $x(t)$ échantillonné à la fréquence f_e .

1. Montrez que $y_n = e^{-nT_e} - \frac{1}{2}\delta_n$

2. Montrez que

$$\hat{Y}(f) = -0.5 + \frac{1}{1 - e^{-T_e - j2\pi f T_e}} \quad (12.3)$$

Solution

- $y_0 = \frac{1}{2}e^1 + \frac{1}{2} \times 0 = 0.5$ et $y_n = e^{-nT_e}$ pour $n \geq 1$. Donc $y_n = e^{-nT_e} \llbracket n \geq 0 \rrbracket - 0.5\delta_n$.
- En application de $1 + Z + Z^2 + \dots = \frac{1}{1-Z}$ pour $|Z| < 1$, on trouve le résultat souhaité en appliquant cela à $Z = e^{-T_e}e^{-j2\pi f T_e}$ et en rajoutant $\text{TFTD}[-0.5\delta_n](f) = -0.5$.

Exercice 36 Le signal étudié ici est $x(t) = t \llbracket t \in [0, 1] \rrbracket + (2-t) \llbracket t \in [1, 2] \rrbracket$. On considère $y(t)$ obtenu en périodisant le signal $x(t)$ pour $t \in [0, 3]$.

- $x(t)$ est-il un signal périodique ou non-périodique.
- $y(t)$ est-il un signal périodique ou non-périodique.
- Dessiner $x(t)$ pour $t \in [-1, 5]$ sur un graphe.
- Dessiner $y(t)$ pour $t \in [-1, 5]$ sur le même graphe.
- Calculez $x(0)$, $x(-2)$, E_x et P_x .
- Calculez $y(0)$, $y(-2)$, E_y et P_y .
- Calculez \hat{X}_0 et \hat{Y}_0 .
- Calculez \hat{X}_0 et \hat{Y}_0 .
- Dessiner sur le graphe $y_1(t) = y(\frac{t}{2})$
- Dessiner sur le graphe $y_2(t) = y(t-1)$
- Dessiner sur le graphe $y_3(t) = \frac{1}{2}y(t)$
- Dessiner sur le graphe $y_4(t) = y(t) - y(t-2)$

Simulation générant le graphe

```
t=linspace(-1,5,1e3);
x=2*cos(pi*t+0.5*pi);
figure(1);
plot(t,x,'b-','linewidth',2);
set(gca,'fontsize',20);
saveas(1,'C:\A\SIMU\SEB\ex\exSEB3_fig1.png');
```

Solutions

- non-périodique
- périodique
- C'est une fonction triangle centrée en $t = 1$ et de base de taille 2.
- C'est une suite de triangles sur $[3n, 3n + 2]$ et nul sur $[3n + 2, 3n + 3]$.
- $x(0) = 0 = x(-2)$. $P_x = 0$
Par symétrie, on a $E_x = 2 \int_0^1 t^2 dt = \frac{2}{3}$
- $y(0) = 0$ et $y(-2) = y(1) = 1$. $E_y = +\infty$ $P_y = \frac{E_x}{3} = \frac{2}{9}$.
- C'est la surface du triangle, $\hat{X}(0) = 1$.
- $\hat{Y}_0 = \frac{\hat{X}(0)}{3} = \frac{1}{3}$.
- Le signal $y(t)$ est dilaté d'un facteur 2 et de période 6.
- $y(t-1)$ est périodique de période 3, mais les triangles sont sur les intervalles $[3n + 1, 3n + 3]$.
- $y(t)$ est atténue d'un facteur 0.5, la pointe des triangles arrive en $1/2$.
- C'est un signal périodique de période 3, nul entre 0 et 1 et valant $2 - t$ entre 1 et 3.

Chapter 13

Modélisation stochastique du bruit

Exercice 37 On considère $\overset{r}{X}_1$ et $\overset{r}{X}_2$ deux variables aléatoires gaussiennes indépendantes de moyenne μ et σ . On définit

$$\begin{cases} \overset{r}{X} = \overset{r}{X}_1 + \overset{r}{X}_2 \\ \overset{r}{Y} = \overset{r}{X}_1 - \overset{r}{X}_2 \end{cases} \quad (13.1)$$

1. Calculez $\mu_X, \sigma_X, \mu_Y, \sigma_Y$ les moyennes statistiques et les racine carrés des variance de $\overset{r}{X}$ et $\overset{r}{Y}$.
2. Montrez que $\overset{r}{X}$ et $\overset{r}{Y}$ sont indépendantes au second ordre.
3. Donnez un algorithme permettant de tester que $\overset{r}{X}$ et $\overset{r}{Y}$ sont en fait indépendantes pas seulement au second ordre.

Solution :

1. Pour les espérances de $\overset{r}{X}$ et $\overset{r}{Y}$, on trouve

$$\begin{cases} \mu_X = E[\overset{r}{X}] = E[\overset{r}{X}_1] + E[\overset{r}{X}_2] = 2\mu \\ \mu_Y = E[\overset{r}{Y}] = E[\overset{r}{X}_1] - E[\overset{r}{X}_2] = 0 \end{cases} \quad (13.2)$$

Une première solution pour les variances de $\overset{r}{X}$ et $\overset{r}{Y}$.

$$\begin{cases} \sigma_X^2 = \text{Var}[\overset{r}{X}] = \text{Var}[\overset{r}{X}_1] + \text{Var}[\overset{r}{X}_2] = 2\sigma^2 \\ \sigma_Y^2 = \text{Var}[\overset{r}{Y}] = \text{Var}[\overset{r}{X}_1] + \text{Var}[-\overset{r}{X}_2] = 2\sigma^2 \end{cases} \quad (13.3)$$

Donc $\sqrt{\text{Var}[\overset{r}{X}]} = \sqrt{2}\sigma$.

2. Une deuxième solution est illustrée pour $\overset{r}{X}$.

$$E[\overset{r}{X}^2] = E[\overset{r}{X}_1^2] + 2E[\overset{r}{X}_1 \overset{r}{X}_2] + E[\overset{r}{X}_2^2] = (\mu^2 + \sigma^2) + (2\mu^2) + (\mu^2 + \sigma^2) = (2\mu)^2 + 2\sigma^2 = E[\overset{r}{X}]^2 + 2\sigma^2 \quad (13.4)$$

Donc $\sqrt{\text{Var}[\overset{r}{X}]} = \sqrt{E[\overset{r}{X}^2] - E[\overset{r}{X}]^2} = \sqrt{2}\sigma$.

```
import numpy.random as nr
K = 10**2
for k in range(K):
```

3. **for** $k \in \{0 \dots K - 1\}$ pour $K = 10^2$ **do**

Choisir aléatoirement des valeurs μ, σ suivant une loi uniforme entre 0 et 3.

Choisir aléatoirement des valeurs x_0, y_0 en tirant aléatoirement une valeur pour $\overset{r}{X}_1$ et $\overset{r}{X}_2$.

Tirer aléatoirement $N = 10^7$ valeurs pour $\overset{r}{X}_1$ et $\overset{r}{X}_2$.

Estimer $p_{xy} = P\left(\overset{r}{X} \leq x_0 \text{ et } \overset{r}{Y} \leq y_0\right)$.

Estimer $p_x = P\left(\overset{r}{X} \leq x_0\right)$ et $p_y = P\left(\overset{r}{Y} \leq y_0\right)$.

Vérifier que $|p_{xy} - p_{xy}| < 10^{-3}$.

end for

```

mu,sigma = nr.uniform(0,3), nr.uniform(0,3)
x1,x2   = nr.normal(mu,sigma), nr.normal(mu,sigma)
x0,y0   = x1+x2, x1-x2
N       = 10**7
x1,x2   = nr.normal(mu,sigma,N), nr.normal(mu,sigma,N)
x,y     = x1+x2, x1-x2
pxy     = np.mean((x<=x0)&(y<=y0))
px,py   = np.mean(x<=x0), np.mean(y<=y0)
assert np.abs(px*py-pxy) < 1e-3

```

Exercice 38 On considère un bruit blanc gaussien $\overset{r}{B}(t)$ de moyenne μ et d'écart-type σ que l'on observe entre $t = 0$ et $t = 1$. On l'échantillonne à $f_e = 10\text{Hz}$ et l'on note les variables aléatoires ainsi obtenues $\overset{r}{X}_n$. C'est le même bruit blanc que l'on échantillonne à $f_e^y = 20\text{Hz}$ et l'on note les variables aléatoires obtenues $\overset{r}{Y}_n$.

1. Combien y a-t-il de variables aléatoires $\overset{r}{X}_n$ et $\overset{r}{Y}_n$. Lesquelles sont identiques ?

2. On définit $\overset{r}{Z}_n$ de la façon suivante

$$\begin{cases} \overset{r}{Z}_0 = \overset{r}{Y}_0 \\ \overset{r}{Z}_1 = \overset{r}{Y}_1 + \overset{r}{Y}_2 \\ \overset{r}{Z}_2 = \overset{r}{Y}_3 + \overset{r}{Y}_4 \\ \vdots \end{cases} \quad (13.5)$$

Combien y a-t-il de variables aléatoires $\overset{r}{Z}_n$?

3. Calculez l'espérance et la variance de $\overset{r}{Z}_n$.

4. Calculez la corrélation entre $\overset{r}{Z}_n$ et $\overset{r}{X}_n$.

5. Calculez la corrélation entre $\overset{r}{Z}_n$ et $\overset{r}{Z}_{n-1}$.

6. Donnez l'algorithme permettant de répondre par simulation à ces trois dernières questions.

Solution :

1. 11 pour $\overset{r}{X}_n$, 21 pour $\overset{r}{Y}_n$ dont 11 sont identiques à $\overset{r}{X}_n$.

2. 11

3. $E[\overset{r}{Y}_n] = 0$ et $\text{Var}[\overset{r}{Y}_n] = \sigma^2$.

$\text{Var}[\overset{r}{Z}_0] = \sigma^2$ et $\text{Var}[\overset{r}{Z}_n] = 2\sigma^2$ en utilisant l'indépendance entre $\overset{r}{Y}_{2n-1}$ et $\overset{r}{Y}_{2n}$.

4. On s'appuie sur $\overset{r}{X}_n = \overset{r}{Y}_{2n}$. Quand $n \neq 0$,

$$\rho_{ZX} = \frac{\text{E}[\overset{r}{Z}_n \overset{r}{X}_n]}{\sqrt{\text{Var}[\overset{r}{Z}_n]} \sqrt{\text{Var}[\overset{r}{X}_n]}} = \frac{\sigma^2}{\sqrt{2}\sigma^2} = \frac{\sqrt{2}}{2} \quad (13.6)$$

Et quand $n = 0$, $\rho_{ZX} = 1$.

5. $\rho_{Z_n Z_{n-1}} = 0$

```
6. import numpy.random as nr
ty = seb.arange(0, 1+1/20, 1/20)
tx = ty[::2]
K = 100
r_ZX = np.zeros(len(tx))
r_ZZ = np.zeros(len(tx)-1)
E = np.zeros(len(ty))
Vy = np.zeros(len(ty))
Vx = np.zeros(len(tx))
Vz = np.zeros(len(tx))
for k in range(K):
    Yn = nr.normal(mu, sigma, len(ty))
    Xn = Yn[::2]
    Zn = np.zeros(len(tx))
    Zn[1:] = Yn[2::2]+Yn[1::2]
    Zn[0] = Yn[0]
    E += Yn/K
    Vy += Yn**2/K
    Vx += Xn**2/K
    Vz += Zn**2/K
    r_ZX += Zn*Xn/K
    r_ZZ += Zn[1:]*Zn[:-1]/K
Vy = Vy-E**2
Vx = Vx-mu**2
Vz[1:] = Vz[1:]-4*mu**2
Vz[0] = Vz[0] -mu**2
r_ZX = r_ZX/np.sqrt(Vx*Vz)
r_ZZ = r_ZZ/Vz
```

Exercice 39 On considère une variable aléatoire uniforme sur $[0, 2\pi]$, notée $\overset{r}{T}$. On note $\overset{r}{X}$ et $\overset{r}{Y}$ deux variables aléatoires construites à partir de $\overset{r}{T}$.

$$\overset{r}{X} = \cos(\overset{r}{T}) \text{ et } \overset{r}{Y} = \sin(\overset{r}{T}) \quad (13.7)$$

1. Donnez un algorithme permettant d'estimer la corrélation entre $\overset{r}{X}$ et $\overset{r}{Y}$ et aussi de vérifier si elles sont indépendantes.

2. Montrez que

$$E\left[\overset{r}{X} \overset{r}{Y}\right] = \frac{1}{2\pi} \int_0^{2\pi} \cos(t) \sin(t) dt \quad (13.8)$$

En déduire que $\overset{r}{X}$ et $\overset{r}{Y}$ ne sont pas corrélées.

3. Montrez que

$$\begin{cases} P(\overset{r}{X} \leq -\frac{\sqrt{3}}{2}) = \frac{1}{6} \\ P(\overset{r}{Y} \leq -\frac{\sqrt{3}}{2}) = \frac{1}{3} \\ P(\overset{r}{X} \leq -\frac{\sqrt{3}}{2} \text{ et } \overset{r}{Y} \leq -\frac{\sqrt{3}}{2}) = 0 \end{cases} \quad (13.9)$$

La dernière affirmation pourrait se montrer en utilisant le fait $\overset{r}{X}^2 + \overset{r}{Y}^2 = 1$. En déduire que $\overset{r}{X}$ et $\overset{r}{Y}$ ne sont pas indépendantes.

Solution

1. $\overset{r}{X}$ et $\overset{r}{Y}$ ne sont pas corrélés

```
K=10**4
import numpy.random as nr
T=nr.uniform(0,2*np.pi,K)
sigma_c,sigma_s = np.sqrt(np.mean((np.cos(T))**2)), np.sqrt(np.mean((np.sin(T))**2))
rho = np.mean(np.cos(T)*np.sin(T))/sigma_c/sigma_s
print(rho)
```

$\overset{r}{X}$ et $\overset{r}{Y}$ sont dépendants

```
import numpy.random as nr
K=10**4
x,y = np.cos(nr.uniform(0,2*np.pi)), np.sin(nr.uniform(0,2*np.pi)),
T=nr.uniform(0,2*np.pi,K)
X,Y = np.cos(T), np.sin(T)
Pxy = np.mean((X<=x)&(Y<=y))
Px,Py=np.mean((X<=x)), np.mean((Y<=y))
print(Pxy-Px*Py)
```

2. La densité de probabilité de T est $f_T(t) = \frac{1}{2\pi} [0 \leq t \leq 2\pi](t)$.

$$E[\overset{r}{X}\overset{r}{Y}] = E[\cos(\overset{r}{T}) \sin(\overset{r}{T})] = \frac{1}{2\pi} \int_0^{2\pi} \cos(t) \sin(t) dt = \frac{1}{2\pi} \int_0^{2\pi} \frac{1}{2} \sin(2t) dt \quad (13.10)$$

Cette dernière intégrale vaut 0 car sa primitive est périodique de période 2π .

- 3.

$$P(\overset{r}{X} \leq -\frac{\sqrt{3}}{2}) = P[\cos(T) \leq -\cos \frac{\pi}{6}] = P[\pi - \pi/6 \leq T \leq \pi + \pi/6] = \frac{1}{6} \quad (13.11)$$

$$P(\overset{r}{Y} \leq -\frac{\sqrt{3}}{2}) = P[\sin(T) \leq -\sin \frac{\pi}{3}] = P[\pi + \pi/3 \leq T \leq 2\pi - \pi/3] = \frac{1}{3} \quad (13.12)$$

Une première raison pour laquelle la probabilité conjointe est nulle est que les deux événements ne sont pas possible pour $\overset{r}{T}$. La deuxième raison est les deux événements impliquent $\overset{r}{X}^2 \geq \frac{3}{4}$ et $\overset{r}{Y}^2 \geq \frac{3}{4}$ et donc $1 = \overset{r}{X}^2 + \overset{r}{Y}^2 \geq \frac{3}{4}$.

Chapter 14

Autocorrélation et densité spectrale

Exercice 40 On considère un signal $x(t) = e^{-t} \mathbb{I}[t \geq 0](t)$ et une troncature sur $[0, T]$ définie par $x_T(t) = e^{-t} \mathbb{I}[0 \leq t \leq T](t)$. On note $R_x(t)$ et $R_{x_T}(t)$ leurs autocorrélations.

1. Montrez que pour $\tau \geq 0$, $R_x(\tau) = e^\tau \int_\tau^{+\infty} e^{-2t} dt$
2. Montrez que $R_x(t) = 0.5e^{-|t|}$
3. Montrez que $R_{x_T}(0) = 0.5(1 - e^{-2T})$.
4. Montrez que sa densité spectrale $S_x(f) = \frac{1}{1+4\pi^2f^2}$.
5. Donnez le pseudo-programme permettant de simuler $R_x(t)$ et $S_x(f)$.

Solution :

1. Pour $t > 0$,

$$R_x(t) = \int_0^{+\infty} x(\tau)x(\tau-t) d\tau = \int_t^{+\infty} e^{-\tau}e^{-(\tau-t)} d\tau = e^t \int_t^{+\infty} e^{-2\tau} d\tau \quad (14.1)$$

2. Le calcul montre $R_x(t) = \frac{1}{2}e^{-t}$ pour $t \geq 0$ et on sait qu'une autocorrélation est paire, on en déduit que pour $t \leq 0$, $R_x(t) = R_x(-t) = \frac{1}{2}e^t = \frac{1}{2}e^{-|t|}$ et de même pour $t \geq 0$, $R_x(t) = \frac{1}{2}e^{-t} = \frac{1}{2}e^{-|t|}$.

3.

$$R_{x_T}(0) = P_{x_T} = \int_0^{+\infty} x_T^2(t) dt = \int_0^T e^{-2t} dt = \frac{1}{2}(1 - e^{-2T}) \quad (14.2)$$

4. La densité spectrale est définie par $S_x(f) = \text{TF}[R_x(\tau)] = \text{TF}[\frac{1}{2}e^{-|t|}]$ Deux calculs montrent que

$$\text{TF}[e^{-t} \mathbb{I}[t \geq 0]] = \frac{1}{1 + j2\pi f} \text{ et } \text{TF}[e^t \mathbb{I}[t \leq 0]] = \frac{1}{1 - j2\pi f} \quad (14.3)$$

Par linéarité et en appliquant $\frac{1}{z} + \frac{1}{\bar{z}} = \frac{2\Re(z)}{|z|^2}$, on obtient le résultat souhaité.

```
t = seb.linspace(0,30,5000)
tr= seb.arange(-t[-1],t[-1]+(t[1]-t[0]),t[1]-t[0])
f = seb.linspace(-5,5,200)
h = np.exp(-t)*(t>=0)+0
Rx= seb.correlation(t,h,t,h,tr)
Sx= np.real(seb.TF(tr,Rx,f))
```

Exercice 41 On considère une particule qui à intervalles de temps réguliers, T_e , a une probabilité r de passer d'un état A à un état B . On suppose qu'à $t = 0$, cette particule est dans un état A . On note \vec{S}_n la variable aléatoire qui décrit l'état de cette particule à l'instant $t = nT_e$, $\vec{S}_n = 1$ si l'état est A et $\vec{S}_n = -1$ sinon. On note $p_n = P(S_n = 1)$.

1. Montrez que $p_{n+1} = (1 - r)p_n + r(1 - p_n)$
2. On remarque que si $p_n = p_{n+1}$ alors $p_n = \frac{1}{2}$ et on pose alors $q_n = p_n - \frac{1}{2}$, montrez que $q_{n+1} = q_n(1 - 2r)$.
3. Montrez que $p_n = \frac{1}{2} + \frac{1}{2}(1 - 2r)^n$
4. Montrez que $E[\overset{r}{S}_n] = 2p_n - 1$ et $\text{Var}[\overset{r}{S}_n] = 4p_n(1 - p_n)$.
5. Montrez que $E[\overset{r}{S}_n \overset{r}{S}_{n-1}] = 1 - 2r$.
6. Montrez que $E[\overset{r}{S}_n \overset{r}{S}_{n-2}] = (1 - 2r)^2$.
7. Donnez des algorithmes permettant de confirmer ces calculs.

Solution

```

import seb
import numpy.random as nr
plt,np = seb.debut()
r = 0.1
N,K= 7,10**5
pn = np.zeros(N)
ES = np.zeros(N)
VS = np.zeros(N)
RS1= np.zeros(N)
RS2= np.zeros(N)
T = np.zeros(N)
def actu(r,K,S=np.array([])):
    if len(S) == 0:
        S = np.ones(K)
    else:
        flip = 2*(nr.uniform(0,1,K)>r)-1
        S = flip*S
    return S
S = actu(r,K)
pn[0] = np.mean(S==1)
ES[0] = np.mean(S)
VS[0] = np.std(S)**2
RS1[0] = np.nan
RS2[0] = np.nan
S_anc = S
S_Tanc= S_anc
for n in range(1,N):
    S = actu(r,K,S)
    pn[n] = np.mean(S==1)
    ES[n] = np.mean(S)
    VS[n] = np.std(S)**2
    RS1[n]= np.mean(S*S_anc)
    if 1 == n:
        RS2[n] = np.nan
    else:
        RS2[n]= np.mean(S*S_Tanc)
    S_Tanc= S_anc
    S_anc = S
pn_th = 0.5+0.5*(1-2*r)**np.arange(N)
print(pn-pn_th)

```

```

ES_th = 2*pn_th-1
print(ES-ES_th)
VS_th = 4*(1-pn_th)*pn_th
print(VS-VS_th)
RS1_th = np.concatenate(([np.nan],(1-2*r)*np.ones(N-1)))
RS2_th = np.concatenate(([np.nan]*2,(1-2*r)**2*np.ones(N-1)))
print(RS1-RS1_th)
print(RS2-RS2_th)

```

1. L'énoncé affirme que

$$P\left(\overset{r}{S}_{n+1} = 1 | \overset{r}{S}_n = -1\right) = r \text{ et } P\left(\overset{r}{S}_{n+1} = 1 | \overset{r}{S}_n = 1\right) = 1 - r \quad (14.4)$$

On en déduit que

$$p_{n+1} = (1 - r)P\left(\overset{r}{S}_n = 1\right) + rP\left(\overset{r}{S}_n = -1\right) = (1 - r)p_n + r(1 - p_n) \quad (14.5)$$

2. Quand on remplace p_n et p_{n+1} par l , on trouve

$$l = (1 - r)l + r(1 - l) \Leftrightarrow l[1 - (1 - r) + r] = r \Leftrightarrow l = \frac{1}{2} \quad (14.6)$$

C'est la notion mathématique de "point fixe". Quand on remplace p_n par $q_n + \frac{1}{2}$, on trouve $q_{n+1} = (1 - 2r)q_n$ et $q_0 = p_0 - \frac{1}{2} = 1 - \frac{1}{2} = \frac{1}{2}$.

3. Par récurrence, on montre alors que $q_n = \frac{1}{2}(1 - 2r)^n$, et en remplaçant q_n par $p_n - \frac{1}{2}$, on obtient le résultat souhaité.

4. La définition de l'espérance montre que

$$E[\overset{r}{S}_n] = p_n \times 1 + (1 - p_n) \times (-1) = 2p_n - 1 \text{ et } E[\overset{r}{S}_n^2] = p_n \times 1 + (1 - p_n) \times (-1)^2 = 1 \quad (14.7)$$

On en déduit la variance

$$\text{Var}[\overset{r}{S}_n] = E[\overset{r}{S}_n^2] - \left(E[\overset{r}{S}_n]\right)^2 = 1 - (2p_n - 1)^2 = 4p_n(1 - p_n) \quad (14.8)$$

5.

$$E[\overset{r}{S}_n \overset{r}{S}_{n-1}] = P\left(\overset{r}{S}_n \overset{r}{S}_{n-1} = 1\right) - P\left(\overset{r}{S}_n \overset{r}{S}_{n-1} = -1\right) \quad (14.9)$$

L'événement $\overset{r}{S}_n \overset{r}{S}_{n-1} = 1$ se produit quand

- $\overset{r}{S}_n = 1, \overset{r}{S}_{n-1} = 1$ avec une probabilité de $p_{n-1}(1 - r)$
- $\overset{r}{S}_n = -1, \overset{r}{S}_{n-1} = -1$ avec une probabilité de $(1 - p_{n-1})(1 - r)$

L'événement $\overset{r}{S}_n \overset{r}{S}_{n-1} = -1$ se produit quand

- $\overset{r}{S}_n = 1, \overset{r}{S}_{n-1} = -1$ avec une probabilité de $(1 - p_{n-1})r$
- $\overset{r}{S}_n = -1, \overset{r}{S}_{n-1} = -1$ avec une probabilité de $p_{n-1}r$

On en déduit $E[\overset{r}{S}_n \overset{r}{S}_{n-1}] = (1 - 2r)$

6.

$$E[\overset{r}{S}_n \overset{r}{S}_{n-2}] = P\left(\overset{r}{S}_n \overset{r}{S}_{n-2} = 1\right) - P\left(\overset{r}{S}_n \overset{r}{S}_{n-2} = -1\right) \quad (14.10)$$

L'événement $\overset{r}{S}_n \overset{r}{S}_{n-2} = 1$ se produit quand

- $\overset{r}{S}_n = 1, \overset{r}{S}_{n-1} = 1, \overset{r}{S}_{n-2} = 1$ avec une probabilité de $p_{n-2}(1-r)^2$
- $\overset{r}{S}_n = 1, \overset{r}{S}_{n-1} = -1, \overset{r}{S}_{n-2} = 1$ avec une probabilité de $p_{n-2}r^2$
- $\overset{r}{S}_n = -1, \overset{r}{S}_{n-1} = 1, \overset{r}{S}_{n-2} = -1$ avec une probabilité de $(1-p_{n-2})r^2$
- $\overset{r}{S}_n = -1, \overset{r}{S}_{n-1} = -1, \overset{r}{S}_{n-2} = -1$ avec une probabilité de $(1-p_{n-2})(1-r)^2$

L'événement $\overset{r}{S}_n \overset{r}{S}_{n-2} = -1$ se produit quand

- $\overset{r}{S}_n = 1, \overset{r}{S}_{n-1} = 1, \overset{r}{S}_{n-2} = -1$ avec une probabilité de $(1-p_{n-2})r(1-r)$
- $\overset{r}{S}_n = 1, \overset{r}{S}_{n-1} = -1, \overset{r}{S}_{n-2} = -1$ avec une probabilité de $(1-p_{n-2})r(1-r)$
- $\overset{r}{S}_n = -1, \overset{r}{S}_{n-1} = 1, \overset{r}{S}_{n-2} = 1$ avec une probabilité de $p_{n-2}r(1-r)$
- $\overset{r}{S}_n = -1, \overset{r}{S}_{n-1} = -1, \overset{r}{S}_{n-2} = 1$ avec une probabilité de $p_{n-2}r(1-r)$

On en déduit $E[\overset{r}{S}_n \overset{r}{S}_{n-2}] = (1-2r)^2$

Exercice 42 On considère le signal $x(t) = \cos(2\pi t)$.

1. Calculez sa transformée de Fourier en tant que signal déterministe périodique.
2. Calculez son autocorrélation en tant que processus aléatoire d'énergie infinie. Ce signal est déterministe, mais peut être vu comme un processus aléatoire dont l'espérance est donne le signal lui-même : $E[\overset{r}{X}(t)] = x(t) = \cos(2\pi t)$. Ce signal n'est pas centré, mais ici on considère la même définition de l'autocorrélation associée aux processus aléatoire d'énergie infinie et centrés.
3. Calculez sa densité spectrale en tant que processus aléatoire d'énergie infinie.

Pour la résolution de l'exercice, on pourra utiliser

$$\cos(a)\cos(b) = \frac{1}{2}\cos(a+b) + \frac{1}{2}\cos(a-b) \quad (14.11)$$

1. Soit par identification avec la série de Fourier, soit parce que la transformée de Fourier inverse d'un Dirac est une exponentielle complexe,

$$\widehat{X}(f) = \frac{1}{2}\delta(f-1) + \frac{1}{2}\delta(f+1) \quad (14.12)$$

2. Le processus aléatoire $\overset{r}{X}(t)$ qui est en fait déterministe vérifie

$$E[\overset{r}{X}(t)] = \cos(2\pi t) \llbracket 0 \leq t \leq T \rrbracket(t) \text{ et } \text{Var}[\overset{r}{X}(t)] = 0 \quad (14.13)$$

$$\begin{aligned} E[\overset{r}{X}(t)\overset{r}{X}(t-\tau)] &= \cos(2\pi t) \llbracket 0 \leq t \leq T \rrbracket(t) \cos(2\pi(t-\tau)) \llbracket \tau \leq t \leq T+\tau \rrbracket(t) \\ &= \cos(2\pi t) \cos(2\pi(t-\tau)) \llbracket \tau \leq t \leq T \rrbracket(t) \\ &= \frac{1}{2} (\cos(4\pi t - 2\pi\tau) + \cos(2\pi\tau)) \llbracket \tau \leq t \leq T \rrbracket(t) \end{aligned} \quad (14.14)$$

Quand $T \rightarrow +\infty$,

$$\frac{1}{T} \int_{\tau}^T \cos(2\pi t) \cos(2\pi(t-\tau)) dt = \frac{1}{T} \left[\frac{1}{2} \frac{\sin(4\pi t - 2\pi\tau)}{4\pi} \right]_{\tau}^T + \frac{1}{2} \cos(2\pi\tau) \frac{T-\tau}{T} \rightarrow \frac{1}{2} \cos(2\pi\tau) \quad (14.15)$$

Donc $R_x(\tau) = \frac{1}{2} \cos(2\pi\tau)$

3.

$$S_x(f) = \text{TF}[R_x(\tau)](f) = \frac{1}{4}\delta(f-1) + \frac{1}{4}\delta(f+1) \quad (14.16)$$

Chapter 15

Filtrage des processus aléatoires et application au bruit en $1/f$

Exercice 43 On considère un processus aléatoire blanc $\tilde{X}(t)$ dont la puissance totale calculée entre la fréquence nulle et f_0 vaut P_x . On suppose qu'il est échantillonné à $f_e = 5f_0$ et alors notée \tilde{X}_n . On place \tilde{X}_n entrée d'un filtre passe-bas de réponse fréquentielle $\frac{1}{1+j2\pi f/f_1}$ et on note $\tilde{Y}(t)$ le processus aléatoire résultant. On pourra utiliser

$$\int_0^x \frac{1}{1+\tau^2} d\tau = \arctan(x), \quad TF[e^{-|t|}](f) = \frac{2}{1+4\pi f^2}, \text{ et } TF[e^{-t} \llbracket t \geq 0 \rrbracket](f) = \frac{1}{1+j2\pi f} \quad (15.1)$$

1. Quelle est l'autocorrélation $R_x[n]$?
2. Quelle est la densité spectrale $S_x(f)$?
3. Quelle est la densité spectrale $S_y(f)$?
4. Quelle la puissance P_y associée à \tilde{Y} ?
5. Quelle est l'autocorrélation $R_y(t)$?
6. Donnez l'algorithme permettant de simuler $\tilde{Y}(t)$.
7. Donnez l'algorithme permettant de vérifier la cohérence des calculs.

Solution :

1. \tilde{X}_n est un bruit blanc donc $R_x[n] = \delta_n$.

2.

$$S_x(f) = TFTD[R_x[n]](f) = TFTD[\delta_n](f) = 1 \quad (15.2)$$

3.

$$S_y(f) = S_x(f) \left| \widehat{H}(f) \right|^2 = \frac{1}{1+4\pi^2 f^2/f_1^2} \quad (15.3)$$

4.

$$P_y = \int_{-\infty}^{+\infty} S_y(f) df = \int_{-\infty}^{+\infty} \frac{df}{1+4\pi^2 f^2/f_1^2} \quad (15.4)$$

Grâce à un changement de variable $u = \frac{2\pi f}{f_1}$ et à $\int_{-\infty}^{+\infty} \frac{du}{1+u^2} = \pi$ on trouve $P_y = \frac{f_1}{2}$.

5. On observe que $\text{TF}[e^{-|t|}](f) = \frac{2}{1+4\pi^2 f^2}$.

On en déduit que $\text{TF}[e^{-f_1|t|}](f) = \frac{1}{f_1} \frac{2}{1+4\pi^2 \frac{f^2}{f_1^2}}$. Aussi

$$R_y(t) = \text{TF}^{-1}[S_y(f)](t) = \frac{f_1}{2} e^{-f_1|t_1|} \quad (15.5)$$

6. On peut calculer la réponse impulsionnelle du filtre analytiquement, observant que

$$\text{TF}[f_1 e^{-tf_1} \llbracket t \geq 0 \rrbracket](t)(f) = \text{TF}[e^{-t} \llbracket t \geq 0 \rrbracket](t) \left(\frac{f}{f_1} \right) = \hat{H}(f) \quad \text{On en déduit que}$$

$$h(t) = f_1 e^{-tf_1} \llbracket t \geq 0 \rrbracket(t) \quad (15.6)$$

```
import numpy.random as nr
T=3
f1 = 10
fe = 5*f1; Te = 1/fe
t = seb.arange(0,T,1/fe)
B = nr.normal(0,1,len(t))
h = f1*np.exp(-t*f1)*(t>=0)+0
Y = seb.convolution(t,B,t,h,t)/np.sqrt(Te)
```

On peut aussi récupérer h avec `seb.TFI` à partir de $\hat{H}(f)$.

On peut avoir une simulation plus précise du processus $\tilde{Y}(t)$ en ajoutant des zéros entre chaque échantillon en utilisant `seb.zero_aj` et en créant une nouvelle échelle de temps notée $t2$.

```
import numpy.random as nr
T=3
f1 = 10
fe = 5*f1; Te = 1/fe
t = seb.arange(0,T,1/fe)
t2 = seb.arange(0,T,1/fe/M); Te2=Te/M; fe2=fe*M
B = nr.normal(0,1,len(t))
h = f1*np.exp(-t2*f1)*(t2>=0)+0
Y = seb.convolution(t2,B,t2,h,t2)/Te2*np.sqrt(Te)
```

7. À partir de Y , on peut calculer sa puissance

$$Py = np.sum(Y**2)*(t2[1]-t2[0])/T$$

On peut calculer son autocorrélation avec une nouvelle échelle de temps `tr`, le fait de rajoute `Te2` sur le deuxième argument permet que `tr` soit symétrique par rapport à 0. On la compare avec `Ry_th` la valeur théorique.

```
tr = seb.arange(-t2[-1],t2[-1]+Te2,Te2)
Ry = seb.correlation(t2,Y,t2,Y,tr)/T
Ry_th = f1/2*np.exp(-np.abs(tr)*f1)
```

On retrouve la valeur de Py à partir de Ry

$$Ry[seb.where_nearest(tr,0)]$$

On peut en déduire la densité spectrale, que l'on peut aussi comparer à la valeur théorique

```
f = seb.linspace(-fe2/2,fe2/2,10000)
Sy = np.real(seb.TF(tr,Ry,f))
Sy_th = 1/(1+4*np.pi**2*f**2/f1**2)
```

On retrouve approximativement la puissance avec

```
np.sum(Sy)*(f[1]-f[0])
```

On peut aussi reprendre B2 et à partir de ces valeurs calculer une autocorrélation, convoluer avec Rh et retrouver Ry, ou bien calculer une densité spectrale Rb, multiplier par Sh et retrouver Sy.

On peut améliorer la précision de tous ces résultats en les moyennant sur plusieurs expérimentations (seul B est modifié).

Exercice 44 On considère un processus aléatoire en $1/f$ et noté $\tilde{X}(t)$ qui agit entre les fréquences f_0 et f_1 , sa puissance vaut P_x (i.e. c'est le bruit en $1/f$ du cours multiplié par une constante de façon à avoir la puissance souhaitée). On suppose qu'il est échantillonné à $f_e = 5f_1$ et alors notée \tilde{X}_n . On place \tilde{X}_n entrée d'un filtre passe-bas de réponse fréquentielle $\frac{1}{1+j2\pi\frac{f}{f_1}}$ et on note $\tilde{Y}(t)$ le processus aléatoire résultant. On pourra utiliser

$$\int_0^x \frac{1}{1+\tau^2} d\tau = \arctan(x), \quad TF[e^{-|t|}](f) = \frac{2}{1+4\pi f^2}, \text{ et } TF[e^{-t} \llbracket t \geq 0 \rrbracket](f) = \frac{1}{1+j2\pi f} \quad (15.7)$$

1. Quelle est la densité spectrale $S_x(f)$?
2. Donnez un algorithme permettant d'estimer l'autocorrélation $R_x[n]$?
3. Quelle est la densité spectrale $S_y(f)$?
4. Donnez un algorithme permettant d'estimer P_y associée à \tilde{Y} ?
5. Donnez l'algorithme permettant de simuler $\tilde{X}[n]$.
6. Donnez l'algorithme permettant de simuler $\tilde{Y}(t)$.
7. Donnez l'algorithme permettant de vérifier la cohérence des calculs.

Solution

1. J'appelle K une constante telle que $S_x(f) = \frac{K}{f} \llbracket f_0 \leq f \leq f_1 \rrbracket(f)$. K est fixée par P_x :

$$P_x = \int_{-\infty}^{+\infty} S_x(f) df = 2 \int_{f_0}^{f_1} \frac{K}{f} df = K (\ln(f_1) - \ln(f_0)) \quad (15.8)$$

Le coefficient 2 provient de la symétrie de $S_x(f)$ vis-à-vis de $f = 0$. Donc $K = \frac{P_x}{\ln(f_1) - \ln(f_0)}$ et

$$S_x(f) = \frac{1}{2 \ln(f_1) - \ln(f_0)} \frac{P_x}{f} \llbracket f_0 \leq f \leq f_1 \rrbracket(f) \quad (15.9)$$

2. L'autocorrélation se déduit de la densité spectrale avec

$$R_x(t) = \text{TFTD}^{-1} [S_x(f)](t) \quad (15.10)$$

```
Px = 2
f0,f1,fe = 2,20,40
M = 5
K = Px/(np.log(f1)-np.log(f0))
f = seb.linspace(-fe/2*M,fe/2*M,10**3)
Sx = np.zeros(len(f))
ind= np.nonzero((np.abs(f)>=f0)&(np.abs(f)<=f1))
Sx[ind] = K/np.abs(f[ind])/2
tr = seb.arange(-10,10,1/fe/M)
Rx = np.real(seb.TFTDI(f,Sx,tr))
```

3.

$$S_y(f) = S_x(f) \left| \frac{1}{1 + j2\pi \frac{f}{f_1}} \right|^2 = \frac{S_x(f)}{1 + 4\pi^2 \frac{f^2}{f_1^2}} \quad (15.11)$$

4. J'utilise $P_y = \int_{-\infty}^{+\infty} S_y(f) df$

```
Sy = Sx/(1+4*np.pi**2*f**2/f1**2)
Py = np.real(seb.TF(f,Sy,0))
print(f"Py={:3g} <= Px={:3g}"")
```

5. On récupère la réponse impulsionnelle à appliquer à un bruit blanc \vec{B}_n à partir de sa transformée de Fourier qui vaut $\sqrt{S_x(f)}$.

```
t = seb.arange(0,T,1/fe/M)
te = t[0::M]
assert np.abs(te[1]-te[0]-1/fe)<1e-10
B = np.zeros(len(t))
B[0::M] = nr.normal(0,1,len(te))
th = seb.arange(-T,T,1/fe/M)
h = seb.TFI(f,np.sqrt(Sy),th)
Y = seb.convolution(t,B,th,h,t)/np.sqrt(fe)*fe*M
Py1= np.real(seb.TF(t,Y**2,0))/T
print(f"Py1={:3g} Py={:3g}"")
```

6. On essaye de retrouver $S_y(f)$ à partir de $\vec{Y}(t)$.

```
Ry1= seb.correlation(t,Y,t,Y,th)/T
Sy1= seb.TF(th,Ry1,f)
plt.close('all')
fig,ax = plt.subplots()
ax.plot(f,Sy1)
ax.plot(f,Sy)
ax.set_xlim(-f1,f1)
plt.tight_layout()
fig.show()
```

Chapter 16

Densité de probabilité et application au bruit de grenaille

Exercice 45 On considère un bruit blanc échantillonné $\overset{r}{B}_n$ qui a une probabilité 0.5 de valoir 1 et 0.5 de valoir -1. Il est échantillonné à la fréquence $f_e = 1$. Et au moyen de $m(t) = \llbracket 0 \leq t < T_e/2 \rrbracket(t)$, il génère un bruit blanc avec forme.

$$\overset{r}{X}(t) = \sum_{n=-\infty}^{+\infty} \overset{r}{B}_n m(t - nT_e) \quad (16.1)$$

On considère deux filtres définis par leurs réponses impulsionales $h_1(t)$ et $h_2(t)$.

$$h_1(t) = \frac{\delta(t) + \delta(t - 1/2)}{2} \text{ et } h_2(t) = \llbracket 0 \leq t \leq 2 \rrbracket(t) \quad (16.2)$$

On note $\overset{r}{Y}(t)$ et $\overset{r}{Z}(t)$ les sorties des deux filtres lorsque l'entrée est $\overset{r}{X}(t)$. On pourra utiliser le fait que

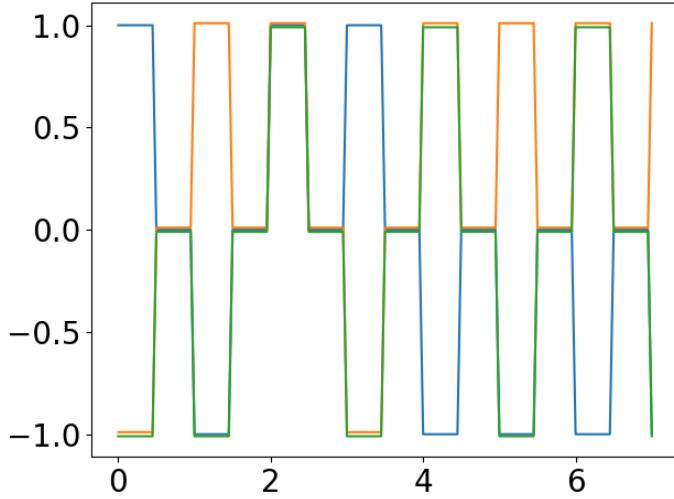
$$\Pi(t) * \Pi(t) = \mathbb{T}(t) \quad (16.3)$$

où $\Pi(t)$ est la porte entre -0.5 et 0.5 et \mathbb{T} est la fonction triangle entre -1 et 1 avec un sommet à 1 en $t = 0$.

1. Dessinez des trajectoires de $\overset{r}{X}(t)$.
2. Dessinez des trajectoires de $\overset{r}{Y}(t)$.
3. Dessinez des trajectoires de $\overset{r}{Z}(t)$.
4. Donnez un algorithme pour obtenir ces trajectoires
5. Calculez $S_m(f)$ à partir de $TF[m(t)](f)$ noté $\widehat{M}(f)$.
6. En déduire $R_m(t)$ en fonction de $\mathbb{T}(t)$ en utilisant que $TF[\mathbb{T}(t)](f) = |\Pi(t)|^2$.
7. Calculez $S_{h_1}(f)$ à partir de $TF[h_1(t)](f)$ noté $\widehat{H}_1(f)$.
8. En déduire $R_{h_1}(t)$ à partir de $S_{h_1}(f)$
9. Calculez $S_{h_2}(f)$ et $R_{h_2}(t)$ de la même façon que $S_m(f)$ et $R_m(t)$.
10. En déduire $R_z(t)$ sous la forme d'un produit de convolution de deux fonctions triangles et $S_z(f)$.
11. Donnez l'algorithme permettant de simuler ces notions.

Solution

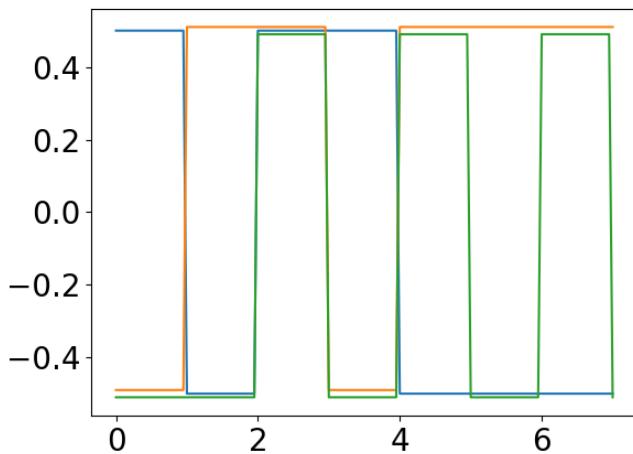
1. Toutes les trajectoires sont nulles sur $[n + 1/2, n + 1]$ et valent 1 ou -1 sur $[n, n + 1/2]$.



2. Si $t \in [n, n + 1/2]$, $\overset{r}{Y}(t) = \frac{1}{2} \overset{r}{B}_n$

Si $t \in [n + 1/2, n + 1]$, $\overset{r}{Y}(t) = \frac{1}{2} \overset{r}{B}_n$

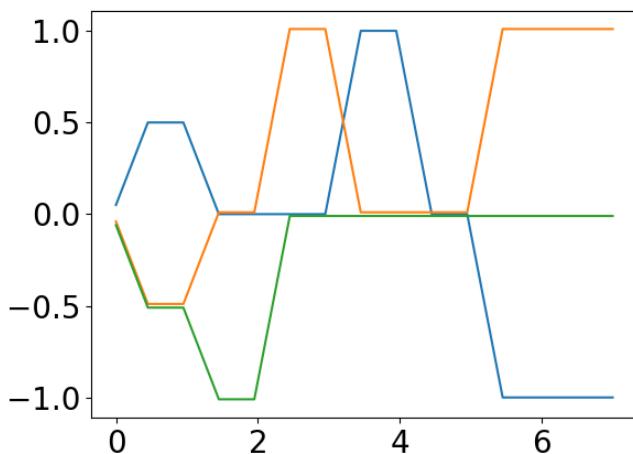
Les traits sont prolongés.



3. Si $t \in [n, n + 1/2]$, $\overset{r}{Z}(t) = (\overset{r}{B}_n - \overset{r}{B}_{n-2})(t - n) + \frac{1}{2}(\overset{r}{B}_{n-1} + \overset{r}{B}_{n-2})$

Si $t \in [n + 1/2, n + 1]$, $\overset{r}{Z}(t) = \frac{1}{2}(\overset{r}{B}_{n-1} + \overset{r}{B}_{n-2})$

Il y a une transition montante ou descendante sur les premiers intervalles et des plateaux sur les deuxièmes intervalles.



```

4. fe = 1
M = 20
T = 7
t = np.arange(0,T+1/fe/M,1/fe/M)
te = t[::M]
def B():
    B = np.zeros(len(t))
    B[::M] = 2*nr.randint(0,2,len(te))-1.0
    return B
tm = np.arange(0,1/fe+1/fe/M,1/fe/M)
m = (tm>=0)&(tm<1/fe/2)+0
B1,B2,B3 = B(),B(),B()
assert len(t)==len(B1)
assert len(tm)==len(m)
X1 = seb.convolution(t,B1,tm,m,t)*fe*M
X2 = seb.convolution(t,B2,tm,m,t)*fe*M
X3 = seb.convolution(t,B3,tm,m,t)*fe*M
plt.close('all')
fig,ax = plt.subplots()
ax.plot(t,X1)
ax.plot(t,X2+0.01)
ax.plot(t,X3-0.01)
plt.tight_layout()
fig.savefig('./figures/fig_ess61_a.png')
fig.show()
Y1 = 0.5*X1 + 0.5*seb.convolution(t,B1,tm,m,t-0.5)*fe*M
Y2 = 0.5*X2 + 0.5*seb.convolution(t,B2,tm,m,t-0.5)*fe*M
Y3 = 0.5*X3 + 0.5*seb.convolution(t,B3,tm,m,t-0.5)*fe*M
fig,ax = plt.subplots()
ax.plot(t,Y1)
ax.plot(t,Y2+0.01)
ax.plot(t,Y3-0.01)
fig.savefig('./figures/fig_ess61_b.png')
plt.tight_layout()
fig.show()
th = np.arange(0,2/fe+1/fe/M,1/fe/M)
h = (th>=0)&(th<2)+0
Z1 = seb.convolution(t,X1,th,h,t)
Z2 = seb.convolution(t,X2,th,h,t)
Z3 = seb.convolution(t,X3,th,h,t)
fig,ax = plt.subplots()
ax.plot(t,Z1)
ax.plot(t,Z2+0.01)
ax.plot(t,Z3-0.01)
fig.savefig('./figures/fig_ess61_c.png')
plt.tight_layout()
fig.show()

```

5. $m(t)$ est le signal $\Pi(t)$ retardé de $1/2$ puis dilaté d'un facteur 2 : $m(t) = \Pi\left(2t - \frac{1}{2}\right)$.

$$\text{TF}[\Pi(t - \frac{1}{2})](f) = \text{sinc}(f)e^{-j\pi f} \text{ et } \widehat{M}(f) = \text{TF}[\Pi(2t - \frac{1}{2})](f) = \frac{1}{2} \text{sinc}(f/2)e^{-j\pi f/2} \quad (16.4)$$

On en déduit $S_m(f) = |\widehat{M}(f)|^2 = \frac{1}{4} \text{sinc}^2(f/2)$.

6. On récupère la transformée de Fourier de $\mathbb{T}(t)$

$$\text{TF}[\mathbb{T}(t)](f) = \text{TF}[\Pi(t) * \Pi(t)](f) = (\text{TF}[\Pi(f)])^2 = \text{sinc}^2(f) \quad (16.5)$$

Puis on l'adapte pour trouver $R_m(t)$

$$\text{TF}[\mathbb{T}(2t)](f) = \frac{1}{2} \text{sinc}^2(f/2) \quad (16.6)$$

Aussi on a

$$R_m(t) = \text{TF}^{-1}[S_m(f)] = \text{TF}^{-1}\left[\frac{1}{2} \text{sinc}^2(f/2)\right] = \frac{1}{2} \mathbb{T}(2t) \quad (16.7)$$

7. La réponse fréquentielle du premier filtre est

$$\hat{H}_1(f) = \text{TF}\left[\frac{1}{2}\delta(t) + \frac{1}{2}\delta(t - \frac{1}{2})\right] = \frac{1}{2} + \frac{1}{2}e^{-j\pi f} \quad (16.8)$$

On en déduit

$$S_{h_1}(f) = |\hat{H}_1(f)|^2 = \left(\frac{1}{2} + \frac{1}{2} \cos(\pi f)\right)^2 + \left(\frac{1}{2} \sin(\pi f)\right)^2 = \frac{1}{2} + \frac{1}{2} \cos(\pi f) \quad (16.9)$$

Comme $\text{TF}[\delta(t)] = 1$, $\text{TF}[\delta(t - 0.5)](f) = e^{-j\pi f}$ et $\text{TF}[\delta(t + 0.5)](f) = e^{j\pi f}$, on en déduit que

$$R_{h_1}(t) = \text{TF}^{-1}\left[\frac{1}{2} + \frac{1}{2} \cos(\pi f)\right] = \frac{1}{2}\delta(t) + \frac{1}{4}\delta(t - 0.5) + \frac{1}{4}\delta(t + 0.5) \quad (16.10)$$

8. $h_2(t)$ est le signal $\Pi(t)$ retardé de $1/2$ puis dilaté d'un facteur 2 : $h_2(t) = \Pi\left(\frac{t}{2} - \frac{1}{2}\right)$.

$$\text{TF}[\Pi(t - \frac{1}{2})](f) = \text{sinc}(f)e^{-j\pi f} \text{ et } \hat{H}_2(f) = \text{TF}[\Pi(\frac{t}{2} - \frac{1}{2})](f) = 2 \text{sinc}(2f)e^{-2j\pi f} \quad (16.11)$$

On en déduit $S_{h_2}(f) = |\hat{H}_2(f)|^2 = 4 \text{sinc}^2(2f)$.

9.

$$R_{h_2}(t) = \text{TF}^{-1}[S_{h_2}(f)] = 2\text{TF}^{-1}[2 \text{sinc}^2(2f)] = 2\mathbb{T}\left(\frac{t}{2}\right) \quad (16.12)$$

10.

$$R_z(t) = \frac{1}{T_e} R_m(t) * R_{h_2}(t) = \frac{1}{2T_e} \mathbb{T}(2t) * 2\mathbb{T}\left(\frac{t}{2}\right) = f_e \mathbb{T}(2t) * \mathbb{T}\left(\frac{t}{2}\right) \quad (16.13)$$

$$S_z(f) = \frac{1}{T_e} S_m(f) S_{h_2}(f) = \frac{1}{T_e} \frac{1}{4} \text{sinc}^2(f/2) 4 \text{sinc}^2(2f) = f_e \text{sinc}^2(f/2) \text{sinc}^2(2f) \quad (16.14)$$

11. • Question 5 et 6

```

fe = 1
M = 20
T = 7
t = np.arange(0, T+1/fe/M, 1/fe/M)
te = t[:M]
K=100
def B():
    B = np.zeros(len(t))
    B[:M] = 2*nr.randint(0, 2, len(te))-1.0
    return B

```

```

tm = np.arange(0,1/fe+1/fe/M,1/fe/M)
m = (tm>=0)&(tm<1/fe/2)+0
def X():
    return seb.convolution(t,B(),tm,m,t)*fe*M
tr = seb.arange(-T,T,1/fe/M)
f = seb.linspace(-fe,fe,300)
Rx = np.zeros(len(tr))
Sx = np.zeros(len(f))
for k in range(K):
    X1 = X()
    Rx += seb.correlation(t,X1,t,X1,tr)/T/K
    Sx += np.abs(seb.TF(t,X1,f))**2/T/K
plt.close('all')
fig,ax = plt.subplots()
ax.plot(tr,Rx)
ax.plot(tr,fe*seb.fonction_T(tr*2)/2)
plt.tight_layout()
fig.show()
fig,ax = plt.subplots()
ax.plot(f,Sx)
ax.plot(f,1/4*(np.sinc(f/2))**2)
plt.tight_layout()
fig.show()

```

- Question 7 et 8.

```

fe = 1
M = 20
T = 70
t = np.arange(0,T+1/fe/M,1/fe/M)
te = t[:M]
K=100
def B():
    B = np.zeros(len(t))
    B[::M] = 2*nr.randint(0,2,len(te))-1.0
    return B
tm = np.arange(0,1/fe+1/fe/M,1/fe/M)
m = (tm>=0)&(tm<1/fe/2)+0
def X():
    return seb.convolution(t,B(),tm,m,t)*fe*M
def Y():
    B1 = B()
    X1 = seb.convolution(t,B1,tm,m,t)*fe*M
    X2 = seb.convolution(t,B1,tm,m,t-0.5)*fe*M
    return 0.5*X1 + 0.5*X2

tr = seb.arange(-T,T,1/fe/M)
f = seb.linspace(-fe,fe,300)
Ry = np.zeros(len(tr))
Sy = np.zeros(len(f))
for k in range(K):
    Y1 = Y()
    Ry += seb.correlation(t,Y1,t,Y1,tr)/T/K
    Sy += np.abs(seb.TF(t,Y1,f))**2/T/K
Ry_th= 0.5*(fe*seb.fonction_T((tr-0.5)*2)/2)+0.25*(fe*seb.fonction_T((tr-0.5)*2)/2)+0.25*(fe*seb.fonction_T((tr-0.5)*2)/2)

```

```

Sy_th=(0.5*np.cos(np.pi*f)+0.5)*1/4*(np.sinc(f/2))**2
plt.close('all')
fig,ax = plt.subplots()
ax.plot(tr,Ry)
ax.plot(tr,Ry_th)
plt.tight_layout()
fig.show()
fig,ax = plt.subplots()
ax.plot(f,Sy)
ax.plot(f,Sy_th)
ax.plot(f,np.real(seb.TF(tr,Ry_th,f))+0.005)
plt.tight_layout()
fig.show()

```

- Question 9 et 10.

```

fe = 1
M = 20
T = 70
t = np.arange(0,T+1/fe/M,1/fe/M)
te = t[:M]
K=100
def B():
    B = np.zeros(len(t))
    B[::M] = 2*nr.randint(0,2,len(te))-1.0
    return B
tm = np.arange(0,1/fe+1/fe/M,1/fe/M)
m = (tm>=0)&(tm<1/fe/2)+0
def X():
    return seb.convolution(t,B(),tm,m,t)*fe*M
th = np.arange(0,2/fe+1/fe/M,1/fe/M)
h = (th>=0)&(th<2)+0
def Z():
    return seb.convolution(t,X(),th,h,t)

tr = seb.arange(-T,T,1/fe/M)
f = seb.linspace(-fe,fe,300)
Rz = np.zeros(len(tr))
Sz = np.zeros(len(f))
for k in range(K):
    Z1 = Z()
    Rz += seb.correlation(t,Z1,t,Z1,tr)/T/K
    Sz += np.abs(seb.TF(t,Z1,f))**2/T/K
Rz_th= fe*seb.convolution(tr,seb.fonction_T(tr*2),tr,seb.fonction_T(tr/2),tr)
Sz_th= fe*(np.sinc(2*f))**2*(np.sinc(f/2))**2
plt.close('all')
fig,ax = plt.subplots()
ax.plot(tr,Rz)
ax.plot(tr,Rz_th)
plt.tight_layout()
fig.show()
fig,ax = plt.subplots()
ax.plot(f,Sz)
ax.plot(f,Sz_th)
ax.plot(f,np.real(seb.TF(tr,Rz_th,f))+0.005)

```

```

plt.tight_layout()
fig.show()

```

Exercice 46 On considère deux variables aléatoires $\overset{r}{X}$ qui suit une loi uniforme sur $[0, 1]$ et $\overset{r}{Y}$ qui suit une loi gaussienne centrée d'écart-type 1.

1. On définit $F_1(\lambda) = P\left(\left|\overset{r}{X} - 0.5\right| \leq \lambda\right)$ Montrez que $F_1(\lambda) = \min(1, 2\lambda)[\lambda \geq 0]$.
2. On définit $F_2(\lambda) = P\left(\left|\overset{r}{Y}\right| \leq \lambda\right)$ Montrez que $F_2(\lambda) = \text{erf}\left(\frac{\lambda}{\sqrt{2}}\right)$.
3. En utilisant l'inégalité sur l'écart à la moyenne donnez une borne inférieure à $F_1(\lambda)$ et à $F_2(\lambda)$.
4. Représentez sur un graphe ces différentes courbes.
5. Donnez le pseudo-algorithme permettant de générer ces courbes et vérifier ces calculs.

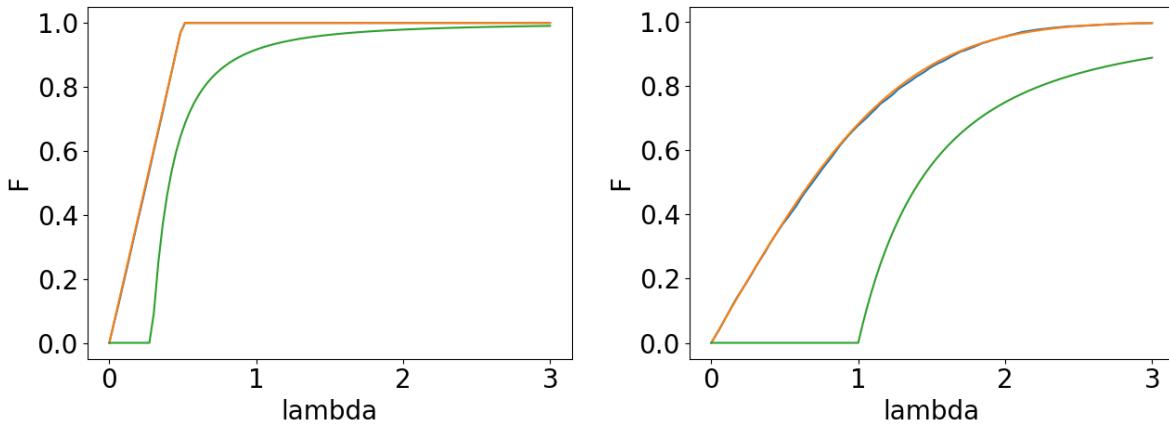


Figure 16.1

Simulation

```

import seb
import numpy.random as nr
plt,np = seb.debut()
K=10**4
M=100
X = nr.uniform(0,1,K)
Y = nr.normal(0,1,K)
lambda_l,F1,F2 = np.linspace(0,3,M),np.linspace(0,3,M),np.linspace(0,3,M)
F1_min,F2_min = np.zeros(M),np.zeros(M)
ind   = np.nonzero(lambda_l>=1/np.sqrt(12))
F1_min[ind] = 1-1/12/(lambda_l[ind]**2)
ind   = np.nonzero(lambda_l>=1)
F2_min[ind] = 1-1/(lambda_l[ind]**2)
for m in range(len(lambda_l)):
    F1[m] = np.mean(np.abs(X-0.5)<=lambda_l[m])
    F2[m] = np.mean(np.abs(Y)<=lambda_l[m])

def _min_(a,b):
    x = np.zeros(len(b))
    x = b*(b<=a)+a*(a<=b)+0
    return x

```

```
plt.close('all')
fig,ax = plt.subplots()
ax.plot(lambda_1,F1)
ax.plot(lambda_1,_min_(1,2*lambda_1))
ax.plot(lambda_1,F1_min)
ax.set_xlabel('lambda')
ax.set_ylabel('F')
plt.tight_layout()
fig.savefig('./figures/exSEB55_fig1a.png')
fig.show()

fig,ax = plt.subplots()
ax.plot(lambda_1,F2)
ax.plot(lambda_1,seb.erf(lambda_1/np.sqrt(2)))
ax.plot(lambda_1,F2_min)
ax.set_xlabel('lambda')
ax.set_ylabel('F')
plt.tight_layout()
fig.savefig('./figures/exSEB55_fig1b.png')
fig.show()
```

Appendix A

Supplément pour faire les pseudo-programmes

A.1 Outils

Python :

Lignes à mettre en début de séance

Les deux premières servent à faire en sorte que les programmes dans `rep_prg` soient utilisables lorsqu'on est dans le répertoire `rep_tra`. `rep_tra` et `rep_prg` sont deux noms de répertoires et de leurs chemins absolues. On peut aussi utiliser des chemins relatifs mais dans ce cas il faut veiller à l'endroit où l'on se trouve.

```
import sys
sys.path.append('rep_prg')
import os
os.chdir('rep_tra')
import seb
plt,np=seb.debut()
```

Cas d'un fichier Python avec extension .py.txt

```
seb.debut
```

La fonction `debut` renvoie un tuple (`plt,np`).

- `plt` sert pour faire des graphes et ses paramètres sont modifiés par `debut` de façon à rendre les graphes plus visibles.
- `np` correspond à `numpy`, c'est utile pour faire des calculs.

A.1.1 Commandes générales

- `help` suivi de parenthèses avec le nom de la fonction. Le contenu affiché est celui entre trois guillemets dans une fonction.
- `assert` provoque une erreur si ce qui suit n'est pas vrai.
- `len` suivi de parenthèses et le nom d'un vecteur ligne ou colonne, cela donne sa longueur.
- `1j` est le complexe imaginaire j .
- `round` qui fournit ¹ un entier à partir d'un nombre réel.
- `def` et `return` pour définir une fonction.

¹La fonction `round` de `numpy` fournit un entier de type `float`.

- `#` pour mettre une ligne en commentaire.
- `type` pour connaître le type d'une valeur ou d'une variable.
- `min` et `max`

Contenu susceptible d'être utilisé dans np pour numpy

- `linspace` permet de définir un ensemble de valeur régulièrement réparties en précisant la valeur initiale, la valeur finale et le nombre de ces valeurs. La dernière valeur est atteinte. Cette fonction n'entraîne pas en général que les valeurs soient des multiples d'un pas donné ce qui est contradictoire avec ce qui est contradictoire avec ce qu'on attend d'une échelle en temps ou en fréquence en traitement du signal. `seb.linspace` assure cette propriété.
- `arange` permet de définir un ensemble de valeur régulièrement réparties en précisant la valeur initiale, la valeur finale et l'espacement entre ces valeurs. La dernière valeur n'est jamais atteinte. Comme `linspace`, cette fonction n'est pas non plus conforme avec ce qu'on attend d'une échelle en temps ou en fréquence en traitement du signal. `seb.arange` assure cette propriété.
- `sqrt` pour racine carré
- `exp` pour exponentielle
- `array` suivi d'une liste entre crochets de valeurs espacées de virgules, pour définir un vecteur de type `numpy.ndarray`. On peut l'utiliser pour définir une matrice en utilisant deux séries de crochets.
- `real` suivi d'un complexe entre parenthèses pour prendre la partie réelle.
- `abs` suivi d'un complexe ou d'un réel entre parenthèses pour prendre le module ou la valeur absolue.
- `sinc` est la fonction sinus cardinal définie par $\frac{\sin(\pi t)}{\pi t}$
- `concatenate`
- `zeros`, `ones`, `zeros_like` et `ones_like`. Pour les deux premières le premier paramètre est un entier ou un tuple indiquant soit la taille du vecteur soit le nombre de lignes et de colonnes de la matrice à définir. Les deux dernières permettent de créer un vecteur ou une matrice ayant les mêmes dimension qu'un certain objet qu'on transmet en paramètre. Les valeurs de l'objet créés sont nulles si on utilise `zeros` ou `zeros_like` et 1 si on utilise `ones` ou `ones_like`.

Contenu susceptible d'être utilisé dans plt récupéré dans debut

- `subplots`
- `plot`
- `set_xlabel` et `set_ylabel`
- `set_legend`
- `tight_layout`
- `savefig`
- `show`

Contenu susceptible d'être utilisé dans seb

- Les fonctions pour définir un signal
 - `fonction_H` est la fonction échelon $\llbracket t \geq 0 \rrbracket(t)$.
 - `fonction_P` est la fonction porte sur $[-0.5, 0.5]$.
 - `fonction_T` est la fonction triangle sur $[-1, 1]$.
 - `fonction_C` est la demi-fonction triangle croissante sur $[-0.5, 0.5]$.
 - `fonction_D` est la demi-fonction triangle décroissante sur $[-0.5, 0.5]$.
 - `gaussian(x, mu, sigma)` renvoie la fonction associe a densité de probabilité de la Gaussienne de moyenne `mu` et d'écart-type `sigma`.
- Les fonctions pour calculer une transformée de Fourier
 - `TF` et `TFI` pour transformée de Fourier et transformée de Fourier inverse
 - `TFTD(t, x, f)` calcule la TFTD du signal temps discret défini par `t, x` en les fréquences `f`.
 - `TFTDI(f, X, t)` calcule la TFTD inverse du spectre complexe `X` définis pour les fréquences `f`. Le résultat est un signal temps discret pour les instants contenus dans `t`.
 - `TFD(t, s, T, bool)` calcule la TFD, `T` indique soit la période soit l'intervalle utilisée pour décrire le signal périodique `t, s` défini l'échelle de temps et le signal. `bool` vaut `True` si on veut une représentation centrée et `False` si on n'en veut pas. `f` est l'échelle de fréquence retournée. `S` est l'ensemble des coefficients associés `f, S=seb.TFD(t, s, T, bool)`
 - `coef_serie_Fourier(t, x, T, k)` calcule les coefficients de la série de Fourier X_k . `T` est soit la période du signal soit un tuple indiquant un intervalle sur lequel est défini $x(t)$. Si `T` est une valeur alors l'intervalle considéré est $[0, T]$. `k` est la liste des indices des fréquences calculées. Le programme retourne un tuple avec d'abord les fréquences et d'autre part les coefficients associés.
- Les fonctions relatives aux échelles
 - `synchroniser(t)` change l'échelle de temps de façon que le vecteur soit un multiple de la période d'échantillonnage.
 - `linspace(start, stop, num, dtype)` et `arange(start, stop, step, dtype)` similaires à celles définies dans `numpy` mais qui assurent que les échelles sont bien constituées de multiples d'un certain pas et ainsi conforme au traitement du signal.
- Les fonctions pour transformer un signal
 - `retarder(t, x, tau)` tarde le signal $x(t)$ défini par `t` et `x` de `tau` lorsque `tau` est positif et avance de `-tau` si `tau` est négatif.
 - `periodiser_ech_t(t, T)` produit un vecteur de même taille que `t` mais dont les valeurs sont entre 0 et `T` de façon à définir un signal périodique de période `T`. Si `T` est un intervalle alors c'est le motif entre `T[0]` et `T[1]` qui est répété.
- Les fonctions associées au produit de convolution
 - `convolution(tx, x, th, h, ty)` Le programme fournit le produit de convolution de `x` par `h` aux instants demandés par `ty`. `tx`, `th` et `ty` sont les échelles de temps de `x`, `h` et `y`. `x` et `tx` doivent être de même taille. `th` et `h` doivent être de même taille. `tx` doit contenir au moins deux composantes. `x` et `h` sont supposés d'énergie finie.
 - `correlation(tx, x, ty, y, tz)` calcule l'intercorrélation entre `tx, x` et `ty, y` en `tz`
- Les fonctions associées aux équations différentielles et au fait de dériver et d'intégrer.
 - `sol_eq_diff(coef, t)` `coef` sont les coefficients devant les termes de l'équation différentielle définis comme un tuple. `t` est l'ensemble des instants dont a cherché à calculer `y(t)`. `t` est un vecteur avec des points régulièrement espacés

- `deriver(t,x)` dérive le signal défini par `t,x` en renvoyant la dérivée en tous les instants de `t`.
- `integarer(t,x)` intègre le signal défini par `t,x` en renvoyant l'intégrale $\int_{-\infty}^t x(t) dt$ en tous les instants de `t`.
- Des fonctions pour trouver une valeur particulière dans un vecteur.
 - `find_nearest(array, value)` retourne la valeur de `array` la plus proche de `value`.
 - `where_nearest(array, value)` retourne l'indice dans `array` correspondant à la valeur la plus proche de `value`.
- `erreur_quad(fun,intervalle)` est une fonction pour calculer l'erreur quadratique. Cette fonction utilise une variable aléatoire sur un support uniforme pour calculer une erreur quadratique. `intervalle` doit indiquer avec un tuple contenant deux valeurs. `fun` est une fonction à transmettre qui estime l'erreur pour une valeur particulière.
- Des fonctions pour aider à gérer les types en Python. Python distingue une liste contenant une unique valeur de la valeur elle-même et d'un tableau de type `numpy` contenant cette valeur.
 - `val(x)` vérifie si `x` est un `numpy array` contenant une seule valeur, si c'est une seule valeur ou autre chose. Si c'est autre chose, cette fonction déclenche une erreur. Sinon elle renvoie cette unique valeur.
 - `vect(x)` vérifie si `x` est un `numpy array` contenant une ou plusieurs valeurs, une seule valeur ou autre chose. Si c'est autre chose, elle prend le premier élément et vérifie que celui-ci est bien un `numpy array` et elle renvoie ce vecteur. Sinon une erreur est déclenchée.
 - `milieux(b_val)` renvoie un vecteur ayant une composante en moins que `b_val` et correspondant aux milieux des termes consécutifs de `b_val`. Cette fonction suppose que `b_val` est régulièrement réparti.
- Les fonctions pour enregistrer des données et pour les récupérer à nouveau.
 - `save(nom_fichier,list_nom_var,list_var)` sauvegarde sous format binaire la liste des variables indiquées dans `list_var` le fichier s'appelle `nom_fichier` les noms des variables doivent être mises avec des apostrophes autour.
 - `load(nom_fichier)` lit le fichier binaire et renvoie un dictionnaire dont les clés sont les noms des variables enregistrées.

Contenu susceptible d'être utilisé dans `sympy`

- `Symbol`
- `solve`
- `simplify`
- `matrices.Matrix`