

Examen Signal et bruit (2 heures et 30 minutes)

Sujet à rendre avec la copie

Nom :
Prénom :

Les téléphones portables ne sont pas autorisés. Une copie double manuscrite est autorisée.

Exercice 1. On considère une réponse impulsionnelle $x(t) = \mathbb{I}[|t| \leq 0.5](t)$ et $y(t) = \mathbb{I}[|t| \leq 1](t)$. On note $R_x(t)$ et $R_y(t)$ les autocorrélations de $x(t)$ et $y(t)$.

$$R_x(t) = \int_{-\infty}^{+\infty} x(\tau)x(\tau - t) d\tau \text{ et } R_y(t) = \int_{-\infty}^{+\infty} y(\tau)y(\tau - t) d\tau \quad (1)$$

On a vu en cours que $R_x(t) = (1 - |t|)\mathbb{I}[|t| \leq 1](t)$

1. Proposez et démontrez une relation entre $x(t)$ et $y(t)$
2. Écrivez un programme permettant de visualiser $y(t)$.
3. Calculez $R_y(t)$.
4. Donnez un programme en Python permettant de simuler $R_y(t)$.

1.

$$y(t) = x\left(\frac{t}{2}\right) \quad (2)$$

```
2. import seaborn as seb
plt, np = seb.debut()
t = seb.linspace(-2, 2, 500)
y = (np.abs(t) <= 1)
plt.close('all')
fig, ax = plt.subplots()
ax.plot(t, y)
plt.tight_layout()
fig.show()
```

3. J'effectue le changement de variable $\tau' = \tau/2$.

$$\begin{aligned} R_y(t) &= \int_{-\infty}^{+\infty} x(\tau/2)x(\tau/2 - t/2) d\tau = \int_{-\infty}^{+\infty} x(\tau')x(\tau' - t/2) 2d\tau' \\ &= 2R_x(t/2) = 2(1 - |t|/2)\mathbb{I}[|t|/2 \leq 1](t) = (2 - |t|)\mathbb{I}[|t| \leq 2](t) \end{aligned} \quad (3)$$

```
4. import seaborn as seb
plt, np = seb.debut()
t = seb.linspace(-2, 2, 500)
y = (np.abs(t) <= 1).astype(float)
Ry = seb.correlation(t, y, t, y, t)
plt.close('all')
fig, ax = plt.subplots()
ax.plot(t, Ry)
plt.tight_layout()
fig.show()
```

Exercice 2. On considère un processus aléatoire défini par

$$\tilde{X}(t) = 2 \cos(2\pi(f_0 + \tilde{A})t) \text{ où } \tilde{A} \text{ est une loi uniforme sur } [0, 2] \quad (4)$$

On utilise les valeurs de $f_0 = 3$.

1. Dessinez deux trajectoires possible sur un graphe pour $t \in [0, 1]$.
2. Calculez $E[\tilde{X}(t)]$.
3. Donnez un programme en Python permet d'estimer $E[\tilde{X}(t)]$ et de le représenter graphiquement.

Solution :

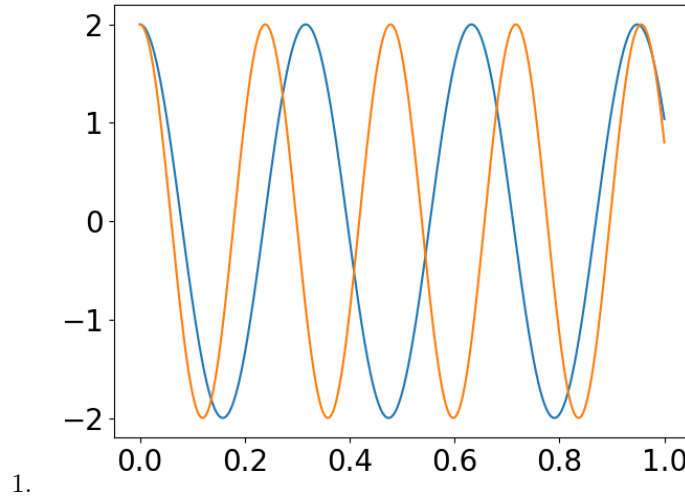


Figure 1: Deux exemples de trajectoires, corrigé exercice ??.

```
import seb
import random as rd
plt, np = seb.debut()
f0=3
t = seb.linspace(0,1,500)
a1,a2 = rd.uniform(0,2), rd.uniform(0,2)
x1,x2 = 2*np.cos(2*np.pi*(f0+a1)*t), 2*np.cos(2*np.pi*(f0+a2)*t)
plt.close('all')
fig, ax = plt.subplots()
ax.plot(t, x1, t, x2)
plt.tight_layout()
fig.show()
fig.savefig('../figures/exSEB66_fig1.png')
```

2.

$$E[\tilde{X}(t)] = \frac{1}{2} \int_0^2 2 \cos(2\pi(f_0 + a)t) da = \left[\frac{\sin(2\pi(f_0 + a)t)}{2\pi t} \right]_0^2 = \frac{\sin(2\pi(f_0 + 2)t) - \sin(2\pi f_0 t)}{2\pi t} \quad (5)$$

```
3. import seb
import random as rd
plt, np = seb.debut()
f0=3
t = seb.linspace(1/1000,1,500)
y = np.zeros(shape=(len(t),), dtype=float)
K = 100
for _ in range(K):
    a = rd.uniform(0,2)
```

```

x = 2*np.cos(2*np.pi*(f0+a)*t)
y += x/K

y_th = (np.sin(2*np.pi*(f0+2)*t)-np.sin(2*np.pi*f0*t))/(2*np.pi*t)
plt.close('all')
fig,ax = plt.subplots()
ax.plot(t,y,t,y_th)
plt.tight_layout()
fig.show()

```

Exercice 3. On considère un filtre défini par

$$\frac{d}{dt}y(t) + y(t) = x(t) \quad (6)$$

On place en entrée $x(t) = \mathbb{I}[0 \leq t \leq 1](t)$ et on note A_x et A_y les valeurs de $\int_0^1 x(t) dt$ et $\int_0^1 y(t) dt$.

1. Écrivez un programme en Python permettant de simuler $h(t)$ la réponse impulsionnelle du filtre.
2. Calculez A_x .
3. Calculez la réponse fréquentielle du filtre considéré.
4. Calculez A_y .
5. Adaptez le programme pour estimer A_y .

Solution :

1. On peut choisir d'estimer la réponse fréquentielle et en déduire une estimation de $h(t)$. Ici j'utilise la résolution de l'équation différentielle.

```

import seb
plt,np = seb.debut()
t = seb.linspace(-1,3,1000)
h = seb.sol_eq_diff((1,1),t)
plt.close('all')
fig,ax = plt.subplots()
ax.plot(t,h)
plt.tight_layout()
fig.show()

```

2. C'est la surface d'un carré.

$$A_x = \int_0^1 x(t) dt = 1 \times 1 = 1 \quad (7)$$

- 3.

$$\hat{H}(f) = \frac{1}{1 + j2\pi f} \quad (8)$$

4. Il serait tentant d'utiliser la relation $A_y = \hat{H}(0)A_x$, mais en réalité

$$\int_0^{+\infty} y(t) dt = \hat{H}(0) \int_0^{+\infty} x(t) dt \hat{H}(0) \int_0^1 x(t) dt = \hat{H}(0)A_x = 1 \times 1 = 1 \quad (9)$$

Ici, j'utilise la réponse fréquentielle pour en déduire la réponse impulsionnelle $h(t) = e^{-t}\mathbb{I}[t \geq 0]$. J'en déduis un calcul avec le produit de convolution.

$$y(t) = h(t) * x(t) = \int_0^t h(t - \tau)x(\tau) d\tau \quad (10)$$

Pour $t \in [0, 1]$,

$$y(t) = \int_0^t e^{-(t-\tau)} d\tau = e^{-t} \int_0^t e^{\tau} d\tau = e^{-t}(e^t - 1) = 1 - e^{-t} \quad (11)$$

Et finalement

$$A_y = \int_0^1 y(t) dt = 1 - [-e^{-t}]_0^1 = 1 + (e^{-1} - 1) = e^{-1} \quad (12)$$

D'ailleurs si on fait le calcul complètement, on retrouve effectivement $\int_0^{+\infty} y(t) dt = 1$

5. J'estime $y(t)$ avec la convolution de $x(t)$ par $h(t)$ et j'en déduis A_y .

```
import seb
plt, np = seb.debut()
t = seb.linspace(-1, 10, 1000)
h = seb.sol_eq_diff((1, 1), t)
x = ((t >= 0) & (t <= 1)).astype(float)
y = seb.convolution(t, x, t, h, t)
Ay = np.real(seb.TF(t, y * ((t >= 0) & (t <= 1)).astype(float), 0))
print(f"{Ay:.3g}")
```

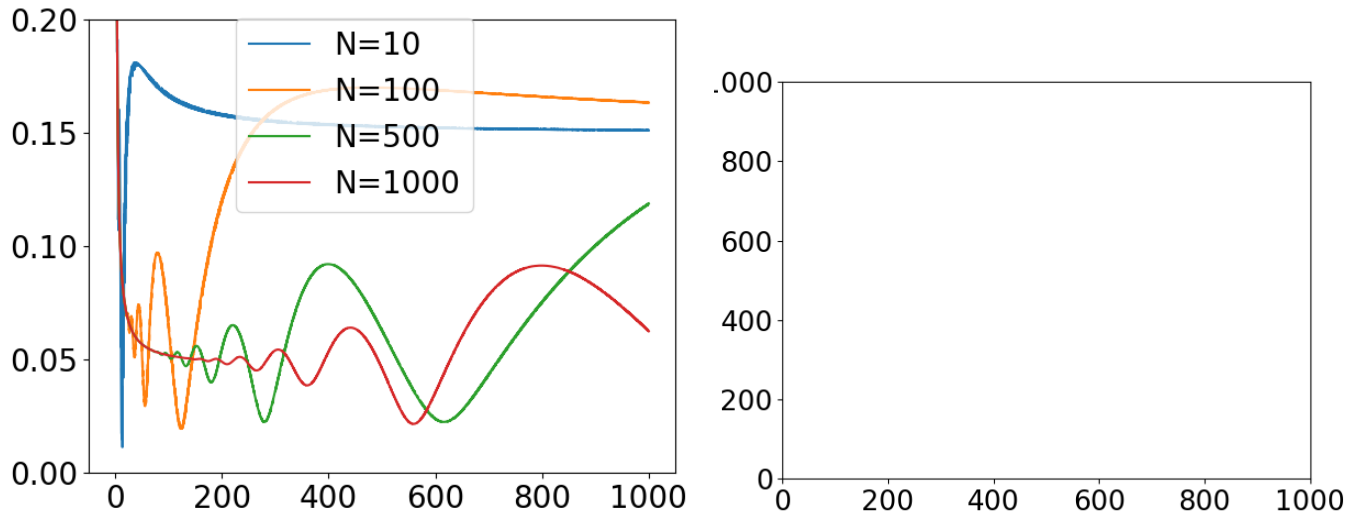


Figure 2: À gauche : évolution de err en fonction de f_e pour différentes valeurs de N .

Exercice 4. On considère un signal défini par $x(t) = e^{-t} \mathbb{I}[t \geq 0](t)$ et on approche ce signal avec $y_n = e^{-nT_e} \mathbb{I}[0 \leq n \leq N-1][n]$. T_e est la période d'échantillonnage associée à la fréquence d'échantillonnage f_e .

1. Donnez l'échelle de temps associée au signal y_n sous la forme d'une suite t_n en fonction de f_e ?
2. Calculez $\hat{X}(1)$ la transformée de Fourier de $x(t)$ en $f = 1$ en fonction de N et f_e .
3. Calculez $\hat{Y}(1)$ la transformée de Fourier de y_n en $f = 1$ en fonction de N et f_e .
4. Calculez $err = \left| \hat{X}(1) - \frac{1}{f_e} \hat{Y}(1) \right|$ en fonction de N et f_e , et expliquez pourquoi il y a ce coefficient $\frac{1}{f_e}$?
5. Donnez le programme permettant d'estimer err sans utiliser les calculs précédents.
6. La gauche de la figure 2 représente différentes courbes de err en fonction de f_e pour $N = 10, 100, 500, 1000$. Il est possible de retrouver quelle courbe correspond à quelle valeur de N en se basant sur l'information suivante : plus la valeur de N est grande plus les oscillations des courbes vis-à-vis de f_e sont dilatées. Ainsi la courbe associée à $N = 10$ ressemble à un échelon tandis que celle associée à $N = 1000$ est celle dont la valeur en $f_e = 1000$ est la plus faible. Représentez grossièrement sur la droite de la figure 2, la courbe de f_e minimisant err en fonction de N en utilisant la gauche de cette figure.

Solution :

1. $t_n = \frac{n}{f_e}$
- 2.

$$\hat{X}(1) = \int_0^{+\infty} e^{-t} e^{-j2\pi t} dt = \left[-\frac{1}{1+j2\pi} e^{-(1+j2\pi)t} \right]_0^{+\infty} = \frac{1}{1+j2\pi} \quad (13)$$

- 3.

$$\hat{Y}(1) = \sum_{n=0}^{N-1} e^{-nT_e} e^{-j2\pi nT_e} = \sum_{n=0}^{N-1} \left(e^{-(1+j2\pi)T_e} \right)^n = \frac{1 - e^{-(1+j2\pi)NT_e}}{1 - e^{-(1+j2\pi)T_e}} \quad (14)$$

- 4.

$$\text{err} = \left| \frac{1}{1+j2\pi} - \frac{1}{f_e} \frac{1 - e^{-(1+j2\pi)NT_e}}{1 - e^{-(1+j2\pi)T_e}} \right| \quad (15)$$

On peut d'ailleurs remarquer que quand $N \rightarrow +\infty$ et $T_e = \frac{1}{f_e} \rightarrow 0$, on a bien $\text{err} \rightarrow 0$.

5. Dans le programme qui suit, j'ai mis la division par f_e dans $Y1[k]$.

```
import seb
plt,np = seb.debut()
plt.close('all')
fig,ax = plt.subplots()
N_list = [10,100,500,1000]
t = seb.linspace(0,1000,10**4)
X1 = seb.TF(t,np.exp(-t),1)
for l in range(4):
    N=N_list[l]
    fe = np.linspace(1e-2,1000,10**4)
    Y1 = np.zeros(shape=(len(fe)),dtype=complex)
    for k in range(len(fe)):
        tk = seb.arange(0,N/fe[k],1/fe[k])
        Y1[k] = seb.TFTD(tk,np.exp(-tk),1)/fe[k]
    err = np.abs(X1-Y1)
    str = f"{N=}"
    ax.plot(fe,err)

plt.tight_layout()
fig.legend(('N=10','N=100','N=500','N=1000'),loc='upper center')
ax.set_ylim([0,0.2])
fig.show()
plt.savefig('../figures/fig_exSEB68_fig1.png')
```

6. On obtient la courbe souhaitée en trouvant pour chaque valeur de N , la valeur de f_e qui minimise l'erreur. Cette valeur f_e en première approximation augmente avec N , on peut donc associer chaque courbe à la valeur de N en fonction de la localisation de son minimum.

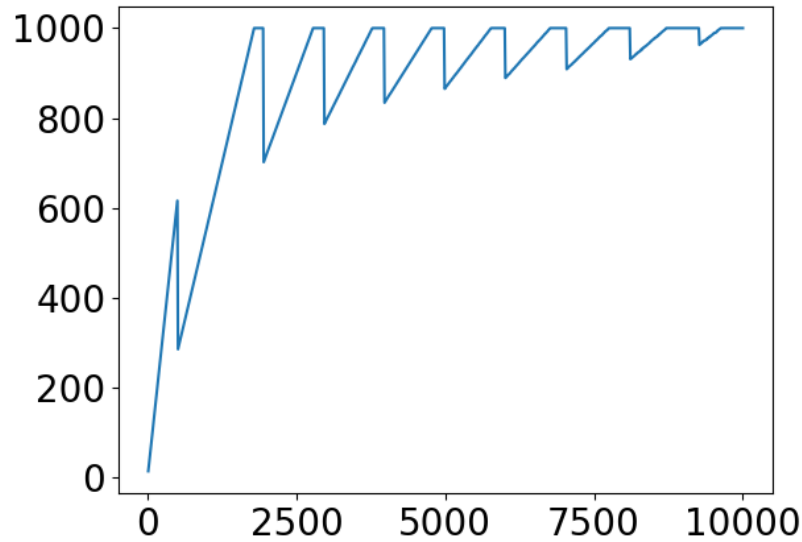


Figure 3: Simulation obtenue avec le programme qui suit.

```
import seb
import numpy as np
N_list = np.linspace(10,10**4,1000)
fe_N = np.zeros(shape=(len(N_list),),dtype=float)
t = seb.linspace(0,1000,10**4)
X1 = seb.TF(t,np.exp(-t),1)
for ind,N in enumerate(N_list):
    print(f"{N=:}")
    assert np.abs(N_list[ind]-N)<1e-5
    fe = np.linspace(1e-2,1000,10**4)
    Y1 = np.zeros(shape=(len(fe),),dtype=complex)
    for k in range(len(fe)):
        tk = seb.arange(0,N/fe[k],1/fe[k])
        Y1[k] = seb.TFTD(tk,np.exp(-tk),1)/fe[k]
    err = np.abs(X1-Y1)
    fe_N[ind] = fe[np.argmin(err)]

plt,np = seb.debut()
plt.close('all')
fig,ax = plt.subplots()
ax.plot(N_list,fe_N)
fig.show()
plt.savefig('../figures/fig_exSEB68_fig3.png')
```

Exercice 5. Expliquez du point de vue du cours de Signal et Bruit ce qu'est le bruit de grenaille.