

Matière :
Traitement du Signal
Master PR — M1

Pr B. BOUZID & Dr. S.K. MEHDI

Série de Travaux Dirigés

Série	Thème	Exercices
1	Énergie, Convolution, Corrélation, Séries de Fourier	7
2	Représentation Fréquentielle, TF, SLI, Filtres	10
3	Signaux Numériques et Systèmes Discrets (SLID)	12
4	Transformée en Z	6
5	Échantillonnage et Transformée de Fourier Discrète	4
Total		39

Objectifs pédagogiques

- Maîtriser les outils mathématiques du traitement du signal
- Analyser et caractériser les signaux dans les domaines temporel et fréquentiel
- Concevoir et analyser des systèmes linéaires invariants (continus et discrets)
- Appliquer les méthodes numériques (TFD, TZ, échantillonnage)

USTHB – Faculté de Physique

Matière : Traitement du Signal

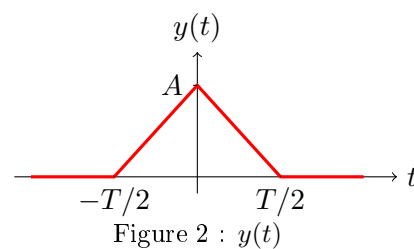
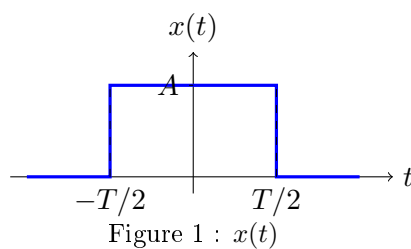
Master PR-M1

Série d'exercices – Énergie, Signaux, Systèmes et Traitement Numérique

SÉRIE 1 Énergie, Convolution, Corrélation et Séries de Fourier

Exercice 1

Soient les signaux $x(t)$ et $y(t)$ représentés sur les figures 1 et 2.



1. Calculer, pour chacun des signaux, son énergie totale et sa puissance totale.
2. Représenter le signal résultant de la convolution de x par y .
3. Calculer et représenter l'autocorrélation de x .
4. Calculer et représenter la corrélation de x par y .

Exercice 2

Soit f une fonction périodique de période 2π définie sur $[-\pi, \pi]$ par :

$$f(0) = 0, \quad f(-\pi) = 0, \quad f(\pi) = 0, \quad \forall t \in]-\pi, 0[, f(t) = -a \quad (a > 0), \quad \forall t \in]0, \pi[, f(t) = a.$$

1. Représenter la fonction.
2. Caractériser énergétiquement le signal représenté.
3. Développer f en série de Fourier.
4. Représenter le spectre de f .

Exercice 3

Soit f une fonction périodique de période 2π , définie sur $[0, 2\pi[$ par $f(x) = e^x$.

1. Représenter la fonction f .
2. Calculer les coefficients complexes de la série de Fourier associée à f .
3. En déduire les sommes :

$$\sum_{n=1}^{+\infty} \frac{1}{1+n^2} \quad \text{et} \quad \sum_{n=1}^{+\infty} \frac{(-1)^n}{1+n^2}.$$

Exercice 4

Évaluer les expressions suivantes :

$$1^\circ) \int \delta(t-4) \cdot \Gamma(t-2) dt$$

$$2^\circ) \int e^{j2\pi f_0 t} \delta(t-t_0) dt$$

$$3^\circ) \int \cos(s) \delta(s - \frac{\pi}{2}) ds$$

$$4^\circ) \int \delta(k-5) \cdot \cos(2\pi k + \frac{\pi}{4}) dk$$

$$5^\circ) \int \delta(t-3) \cdot \Pi(t-2) dt$$

$$6^\circ) \int \delta(t) \cdot \text{sinc}(t) dt$$

Exercice 5

Soit le signal $x(t)$ défini par :

$$x(t) = \begin{cases} \lim_{\tau \rightarrow 0} \frac{1}{\tau} \left(1 - \frac{|t|}{\tau}\right) & \text{si } |t| < \tau \\ 0 & \text{si } |t| > \tau \end{cases}$$

Montrer que ce signal peut être assimilé à une représentation de l'impulsion de Dirac.

Exercice 6

Un détecteur à scintillation NaI(Tl) produit une impulsion de courant modélisée par :

$$x(t) = A \cdot e^{-t/\tau_1} (1 - e^{-t/\tau_2}) \Gamma(t)$$

avec $\tau_1 = 250$ ns (temps de décroissance du scintillateur), $\tau_2 = 50$ ns (temps de montée), $A = 1$ mA.

1. Calculer l'énergie totale de $x(t)$. Ce signal est-il un signal à énergie finie ? Justifier.
2. Calculer et exprimer analytiquement l'autocorrélation $R_{xx}(\tau)$ de $x(t)$.
3. Deux événements produisent des impulsions $x(t)$ et $y(t) = x(t - t_0)$ avec $t_0 = 100$ ns. Calculer la corrélation croisée $R_{xy}(\tau)$. En quelle valeur de τ est-elle maximale ? Interpréter physiquement.
4. À l'aide de Python (NumPy/SciPy/Matplotlib), simuler $x(t)$ sur $[0, 2\mu\text{s}]$ avec un pas d'échantillonnage de 1 ns, puis tracer $x(t)$, $R_{xx}(\tau)$ et $R_{xy}(\tau)$.

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 from scipy.signal import correlate
4
5 tau1 = 250e-9; tau2 = 50e-9; A = 1e-3; t0 = 100e-9
6 dt = 1e-9
7 t = np.arange(0, 2e-6, dt)
8
9 x = A * np.exp(-t/tau1) * (1 - np.exp(-t/tau2))
10 y = np.where(t >= t0, A * np.exp(-(t-t0)/tau1) * (1 - np.exp(-(t-t0)/tau2)), 0)
11
12 Rxx = correlate(x, x, mode='full')
```

```

13 Rxy = correlate(x, y, mode='full')
14 lags = np.arange(-(len(t)-1), len(t)) * dt
15
16 fig, axes = plt.subplots(3, 1, figsize=(10, 8))
17 axes[0].plot(t*1e9, x*1e3); axes[0].set_xlabel('t (ns)'); axes[0].set_ylabel('x
    (mA)')
18 axes[1].plot(lags*1e9, Rxx); axes[1].set_xlabel('tau (ns)'); axes[1].set_ylabel('
    Rxx')
19 axes[2].plot(lags*1e9, Rxy, color='orange'); axes[2].set_xlabel('tau (ns)')
20 for ax in axes: ax.legend(); ax.grid(True)
21 plt.tight_layout(); plt.show()

```

Exercice 7

En spectrométrie gamma, le spectre en énergie d'une source de ^{60}Co présente deux pics photoélectriques à 1173 keV et 1332 keV. Chaque pic est modélisé par une gaussienne :

$$f_i(t) = A_i \exp\left(-\frac{(t - \mu_i)^2}{2\sigma^2}\right), \quad i = 1, 2$$

avec $\mu_1 = 1173$, $\mu_2 = 1332$ (en keV), $\sigma = 30$ keV, $A_1 = 1$, $A_2 = 0.85$. On note $f(t) = f_1(t) + f_2(t)$ le spectre total.

1. Montrer que f est un signal à énergie finie et calculer son énergie totale.
2. Développer f_1 en série de Fourier généralisée sur $[0, 2000]$ keV. Combien de termes sont nécessaires pour une représentation fidèle à 1% près ?
3. Calculer et représenter l'autocorrélation de f . Interpréter les deux pics qui apparaissent dans $R_{ff}(\tau)$.
4. En Python, tracer $f(t)$, son autocorrélation, et vérifier numériquement le théorème de Wiener-Khintchine.

```

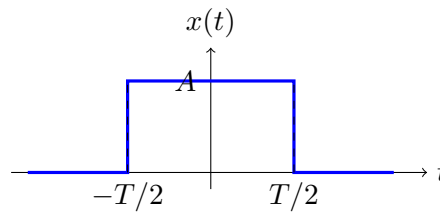
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 t = np.linspace(0, 2000, 10000) # energie en keV
5 mu1, mu2 = 1173, 1332; sigma = 30; A1, A2 = 1.0, 0.85
6 f = A1*np.exp(-(t-mu1)**2/(2*sigma**2)) + A2*np.exp(-(t-mu2)**2/(2*sigma**2))
7
8 F = np.fft.fft(f)
9 DSE = np.abs(F)**2
10 Rff = np.fft.fftshift(np.fft.ifft(DSE).real)
11
12 fig, axes = plt.subplots(2, 1, figsize=(10, 6))
13 axes[0].plot(t, f); axes[0].set_xlabel('Energie (keV)'); axes[0].grid(True)
14 axes[1].plot(Rff); axes[1].set_title('Autocorrelation de f'); axes[1].grid(True)
15 plt.tight_layout(); plt.show()

```

SÉRIE 2 Représentation Fréquentielle – TF – SLI – Filtres Analogiques

Exercice 1

Soit le signal porte $x(t) = A\Pi_T(t)$ représenté ci-dessous :



- 1°) Calculer la TF de ce signal.
- 2°) La représenter ainsi que son spectre d'amplitude.
- 3°) En déduire sa densité spectrale d'énergie.
- 4°) En déduire son énergie totale.
- 5°) En déduire sa fonction d'autocorrélation. Retrouver l'énergie à partir de cette dernière.
- 6°) Déduire du 1°) la TF de $\text{sinc}(t)$ et la représenter.

Exercice 2

À partir de la propriété de translation fréquentielle de la TF et de la TF d'une impulsion de Dirac :

- 1°) Déterminer la TF du signal exponentiel : $x(t) = e^{j2\pi f_0 t}$.
- 2°) En déduire la TF du signal sinusoïdal : $y(t) = \cos(2\pi f_0 t)$.
- 3°) On observe ce signal $y(t)$ sur une durée finie T (différente de la période T_0). En considérant que cette observation correspond à la multiplication par un signal porte, calculer sa TF.
- 4°) Que faudrait-il faire pour que ce spectre se rapproche le plus possible de celui obtenu par la décomposition en série de Fourier ?
- 5°) Interpréter ces résultats du point de vue du compromis de précision temps-fréquence.
- 6°) Calculer la TF d'une fonction gaussienne centrée sur 0, la représenter et montrer qu'elle constitue une meilleure fonction de fenêtrage que la fonction porte.

Exercice 3

Un circuit non linéaire de type parabolique est modélisé par la caractéristique de transfert :

$$u_2(t) = \alpha \cdot u_1(t) + \beta \cdot u_1^2(t)$$

Sachant qu'on lui applique $u_1(t) = A \sin(\omega_0 t)$:

- 1°) Déterminer les composantes spectrales obtenues à la sortie.
- 2°) Quelle est la puissance normalisée P_2 du signal de sortie ?
- 3°) Que vaut-elle par rapport à celle du signal d'entrée P_1 ?

Application numérique : $A = 10 \text{ V}$, $\omega = 200\pi \text{ rad/s}$, $\alpha = 1$, $\beta = 0.2 \text{ V}^{-1}$.

Exercice 4

On applique une exponentielle décroissante $x_1(t) = U_0 e^{-at} \Gamma(t)$ d'amortissement $a = 100 \text{ s}^{-1}$ à un filtre passe-bas de constante de temps $\tau = 1 \text{ ms}$ avec :

$$H(f) = \frac{1}{1 + j2\pi f\tau}$$

- 1°) Calculer la TF $Y(f)$ de la tension de sortie $y(t)$ du filtre.
- 2°) Utiliser le tableau des TF inverses pour déduire l'expression temporelle de $y(t)$.

Exercice 5

Soit le signal :

$$u(t) = \begin{cases} U_0 \cdot \cos(2\pi f_0 t) & \text{si } |t| \leq t_0 \\ 0 & \text{si } |t| > t_0 \end{cases}$$

- 1°) Représenter $u(t)$.
- 2°) Calculer sa TF $U(f)$.
- 3°) Représenter $|U(f)|$ pour $U_0 = 1 \text{ V}$, $T = 1/f_0 = 1 \text{ ms}$ et $t_0 = 10 \text{ ms}$.
- 4°) Montrer que l'analyse spectrale d'une sinusoïde pendant une durée finie revient à remplacer les raies spectrales situées en $f = \pm f_0$ par la fonction sinus cardinal.

Exercice 6

Soit un système analogique d'entrée $x(t)$ et de sortie $y(t)$ décrit par l'équation différentielle :

$$\frac{d^3 y}{dt^3} + 8 \frac{d^2 y}{dt^2} + 17 \frac{dy}{dt} + 10y = \frac{d^2 x}{dt^2} + 2 \frac{dx}{dt} - 15x$$

- 1°) Donner la fonction de transfert $H(s)$. En déduire ses pôles et ses zéros.
- 2°) En déduire l'équation différentielle d'ordre deux du système équivalent.
- 3°) Déterminer l'expression de la réponse impulsionnelle $h(t)$. Ce filtre est-il réalisable ? Justifier.

Le signal d'entrée est $x(t) = 4e^{-4t} \Gamma(t)$.

- 4°) Déterminer le signal de sortie $y(t)$ pour cette entrée.

Exercice 7

Un circuit non linéaire de type parabolique est modélisé par :

$$y(t) = a x(t) + b x^2(t)$$

Sachant qu'on lui applique $x(t) = X_0 \sin(2\pi f_0 t)$:

- 1°) Déterminer les composantes spectrales obtenues à la sortie.
- 2°) Quelle est la puissance normalisée P_2 du signal de sortie ?
- 3°) Que vaut-elle par rapport à celle du signal d'entrée P_1 ?

Application numérique : $X_0 = 1 \text{ V}$, $f_0 = 100 \text{ Hz}$, $a = 1$, $b = 0.2 \text{ V}^{-1}$.

Exercice 8

Soit un système analogique décrit par l'équation différentielle :

$$\frac{d^3y}{dt^3} + 5\frac{d^2y}{dt^2} + 9\frac{dy}{dt} + 5y = 2\frac{dx}{dt} - 5x$$

- 1°) Donner la fonction de transfert $H(s)$. En déduire ses pôles et ses zéros. Écrire le script **Python** permettant d'obtenir ses pôles et ses zéros :

```
1 import numpy as np
2 num = [2, -5]
3 den = [1, 5, 9, 5]
4 zeros = np.roots(num)
5 poles = np.roots(den)
6 print("Zeros :", zeros)
7 print("Poles :", poles)
```

- 2°) Déterminer l'expression de la réponse impulsionnelle $h(t)$. Ce filtre est-il réalisable ? Justifier. Écrire le script **Python** permettant de déterminer $h(t)$ pour $t = 0, 1, 10, 100$ et 1000 :

```
1 from scipy.signal import residue
2 import numpy as np
3 num = [2, -5]; den = [1, 5, 9, 5]
4 R, P, K = residue(num, den)
5 t_vals = np.array([0, 1, 10, 100, 1000])
6 h_vals = np.real(np.sum(R[:, None] * np.exp(P[:, None] * t_vals), axis=0))
7 print("h(t) =", h_vals)
```

- 3°) Déterminer le signal de sortie $y(t)$ pour l'entrée $x(t) = 2e^{-t}\Gamma(t)$.
 4°) Écrire le script **Python** (en utilisant `residue` de SciPy) permettant d'obtenir $y(t)$ pour $t = 0, 1, 10, 100$ et 1000.

Exercice 9

Un détecteur HPGe (germanium hyperpure) est modélisé comme un système linéaire invariant dont la réponse impulsionnelle est :

$$h(t) = \frac{1}{\tau_c} e^{-t/\tau_c} \Gamma(t), \quad \tau_c = 2 \mu\text{s}$$

Le signal d'entrée correspond à une impulsion de courant générée par un photon gamma de 662 keV (source ^{137}Cs) :

$$x(t) = Q_0 \cdot \delta(t), \quad Q_0 = 1.6 \times 10^{-14} \text{ C}$$

- 1°) Déterminer analytiquement la réponse $y(t) = x(t) * h(t)$.
 2°) Calculer la TF $H(f)$ du filtre et identifier son type. Déterminer sa fréquence de coupure f_c .
 3°) Calculer la TF $Y(f)$ du signal de sortie.
 4°) En Python, simuler et tracer $x(t)$, $h(t)$ et $y(t)$ sur $[0, 20 \mu\text{s}]$. Vérifier numériquement que $Y(f) = X(f) \cdot H(f)$.

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from scipy.signal import fftconvolve
4
5 dt = 1e-8; t = np.arange(0, 20e-6, dt); tau_c = 2e-6; Q0 = 1.6e-14
6 sigma = 5e-8
```

```

7 x = Q0/(sigma*np.sqrt(2*np.pi)) * np.exp(-t**2/(2*sigma**2))
8 h = (1/tau_c)*np.exp(-t/tau_c)
9 y = fftconvolve(x, h, mode='full')[:len(t)] * dt
10
11 fig, axes = plt.subplots(3, 1, figsize=(10, 8))
12 for ax, sig, lbl in zip(axes, [x, h, y], ['x(t)', 'h(t)', 'y(t)']):
13     ax.plot(t*1e6, sig, label=lbl); ax.legend(); ax.grid(True)
14     ax.set_xlabel('t (us)')
15 plt.tight_layout(); plt.show()

```

Exercice 10

Un système de comptage de particules alpha enregistre un signal bruité. Le signal utile est $s(t) = e^{-t/T}$ avec $T = 5$ ms, et le bruit est un bruit blanc gaussien centré de densité spectrale $N_0/2$.

- 1°) Exprimer la densité spectrale de puissance $S_{ss}(f)$ du signal utile.
- 2°) Rappeler l'expression du filtre de Wiener optimal $H_{\text{opt}}(f)$ dans le cas signal + bruit blanc.
- 3°) Montrer que pour ce signal, le filtre optimal est un filtre passe-bas du premier ordre. Identifier sa fréquence de coupure en fonction de T et N_0 .
- 4°) Pour $N_0 = 0.01$ et $T = 5$ ms, tracer en Python $H_{\text{opt}}(f)$ et le comparer au filtre passe-bas simple de même fréquence de coupure.

```

1 import numpy as np; import matplotlib.pyplot as plt
2 T = 5e-3; N0 = 0.01
3 f = np.linspace(-1000, 1000, 10000)
4 Sss = (2*T) / (1 + (2*np.pi*f*T)**2)
5 H_opt = Sss / (Sss + N0/2)
6 fc = 1/(2*np.pi*T)
7 H_pb = 1/np.sqrt(1 + (f/fc)**2)
8 plt.plot(f, H_opt, label='Filtre de Wiener')
9 plt.plot(f, H_pb, '--', label='Filtre passe-bas')
10 plt.xlabel('Frequence (Hz)'); plt.legend(); plt.grid(True); plt.show()

```

SÉRIE 3 Signaux Numériques et Systèmes Linéaires Invariants Discrets (SLID)

Exercice 1

Soit un système LIT dont la réponse impulsionnelle $h(n)$ est telle que :

$$h(n) = 1 \text{ pour } 0 \leq n \leq 3, \quad h(n) = 0 \text{ pour } n < 0 \text{ et } n > 3.$$

Calculer la réponse $y(n)$ à la suite $x(n)$ telle que :

$$x(n) = a^n \text{ avec } a = 0.7 \text{ pour } 0 \leq n \leq 5, \quad x(n) = 0 \text{ ailleurs.}$$

Exercice 2

Montrer que la Transformée en Z de la suite causale $x(n)$ définie par :

$$x(n) = nT e^{-anT} \text{ pour } n \geq 0, \quad x(n) = 0 \text{ pour } n < 0$$

a pour expression :

$$X(Z) = \frac{T e^{-aT} Z^{-1}}{(1 - e^{-aT} Z^{-1})^2}$$

Calculer les transformées inverses de $\ln(Z - a)$ et $\frac{Z}{(Z - a)(Z - b)}$, et établir les conditions sur a et b pour que la suite obtenue converge.

Exercice 3

Calculer la Transformée en Z de la réponse impulsionnelle :

$$h(n) = r^n \frac{\sin[(n + 1)\theta]}{\sin(\theta)}, \quad n \geq 0 \quad ; \quad h(n) = 0, \quad n < 0$$

Quel est le domaine de convergence de la fonction obtenue ? Placer ses pôles et zéros dans le plan des Z.

Exercice 4

Soit un système LIT dont la fonction de transfert $H(Z)$ s'écrit :

$$H(Z) = \frac{1}{1 - 1.6 Z^{-1} + 0.92 Z^{-2}}$$

Auquel est appliqué un signal à spectre uniforme et de puissance unité. Calculer la puissance du signal en sortie du système et donner la répartition spectrale.

Exercice 5

Utiliser la Transformation en Z monolatérale pour calculer la réponse du système défini par :

$$y(n) = x(n) + y(n - 1) - 0.8 y(n - 2)$$

avec les conditions initiales $y(-1) = a$ et $y(-2) = b$, à la suite :

$$x(n) = e^{jn\omega} \text{ pour } n \geq 0, \quad x(n) = 0 \text{ pour } n < 0.$$

Mettre en évidence la réponse due aux conditions initiales et la réponse en régime permanent.

Exercice 6

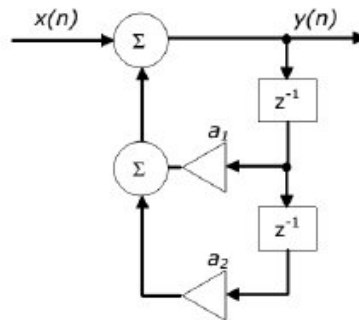
Un SLI est décrit par l'équation :

$$y(n) + a_1 y(n - 1) = b_0 x(n) + b_2 x(n - 2)$$

- Dessiner la structure du filtre correspondant.
- Trouver l'expression analytique de sa réponse impulsionnelle.
- En déduire une condition pour que le filtre soit stable.
- Vérifier cette condition à partir de la fonction de transfert en z .

Exercice 7

Le système suivant constitue, pour certaines valeurs de ses coefficients, un oscillateur numérique implémentant un calcul purement récursif de $\cos(x)$:



On demande :

- 1°) D'écrire la récurrence numérique correspondante.
- 2°) De trouver les valeurs de a_1 et a_2 permettant à ce filtre d'avoir une réponse impulsionnelle de pulsation réduite φ et d'amortissement nul.
- 3°) De calculer les pôles de ce système et de les dessiner dans le plan complexe.

Exercice 8

Un homme place un couple de lapins dans un enclos. Chaque couple donne naissance à un nouveau couple chaque mois, et chaque nouveau couple devient productif à partir de son troisième mois.

$y(n)$ est le nombre de couples au n -ième mois, $x(n)$ le nombre de couples ajoutés. Le système est caractérisé par :

$$y(n) = y(n - 1) + y(n - 3) + x(n)$$

- 1°) Déterminer la réponse impulsionnelle $h(n)$ pour $n < 7$.
- 2°) Écrire le code **Python** permettant de déterminer $h(n)$ pour $n < 7$:

```

1 import numpy as np
2 from scipy.signal import lfilter
3 b = [1]; a = [1, -1, 0, -1]
4 n = np.arange(7)
5 delta = (n == 0).astype(float)
6 h = lfilter(b, a, delta)
7 print("h(n) =", h)

```

3°) On introduit dans l'enclos un couple le 1^{er} mois et 2 le troisième mois. Déterminer le nombre de couples pour les 5 premiers mois.

4°) Écrire le code **Python** correspondant :

```
1 x = np.array([1, 0, 2, 0, 0], dtype=float)
2 y = lfilter(b, a, x)
3 print("y(n) =", y)
```

5°) Donner la transformée en Z $H(Z)$ du système.

Exercice 9

La réponse d'un système numérique est décrite par :

$$y(n) = -y(n-1) + 2y(n-3) + 3x(n)$$

1°) Déterminer les valeurs de $h(n)$ pour $n < 7$. Écrire le code **Python** permettant d'obtenir $h(n)$ pour $n < 7$:

```
1 from scipy.signal import lfilter
2 import numpy as np
3 b = [3]; a = [1, 1, 0, -2]
4 n = np.arange(7); delta = (n==0).astype(float)
5 h = lfilter(b, a, delta)
6 print("h(n) =", h)
```

L'entrée $x(n)$ est définie par $x(0) = 2$, $x(2) = 1$, $x(n) = 0$ ailleurs.

2°) Déterminer les valeurs de $y(n)$ pour $n < 7$.

3°) Écrire le code **Python** permettant d'obtenir $y(n)$ pour $n < 16$:

```
1 x = np.zeros(16); x[0]=2; x[2]=1
2 y = lfilter(b, a, x)
3 print("y(n) =", y)
```

4°) Représenter le graphe de fluence de ce système.

5°) Déterminer $H(z)$ et ses pôles et zéros.

6°) En déduire l'expression de la réponse impulsionnelle causale $h(n)$.

Le signal d'entrée est $x(n) = \delta(n) + 2\delta(n-1)$.

7°) Déterminer $Y(z)$ et en déduire les expressions des réponses causale et anticausale de $y(n)$.

Exercice 10

La réponse d'un système numérique est décrite par :

$$y(n) = 16y(n-4) + x(n) + 3x(n-1)$$

1°) Écrire le code **Python** permettant d'obtenir $h(n)$ pour $n < 9$:

```
1 from scipy.signal import lfilter
2 import numpy as np
3 b = [1, 3]; a = [1, 0, 0, 0, -16]
4 n = np.arange(9); delta = (n==0).astype(float)
5 h = lfilter(b, a, delta)
6 print("h(n) =", h)
```

L'entrée $x(n)$ est définie par $x(0) = 2$, $x(1) = 1$, $x(3) = 2$, $x(n) = 0$ ailleurs.

- 2°) Écrire le code **Python** permettant d'obtenir $y(n)$ pour $n < 9$.
- 3°) Représenter le graphe de fluence de ce système.
- 4°) Déterminer $H(z)$, ses pôles et ses zéros.
- 5°) En déduire la réponse impulsionnelle causale $h(n)$.

Le signal d'entrée est $x(n) = \delta(n) + 2\delta(n-1)$.

- 6°) Déterminer $Y(z)$. En déduire les réponses causale et anticausale $y(n)$.

Exercice 11

En tomographie par émission de positons (TEP), le signal numérique acquis par un détecteur est modélisé par la suite causale :

$$x(n) = A \cdot (0.9)^n \cdot u(n)$$

représentant la décroissance de l'activité d'un radio-isotope (^{18}F , $T_{1/2} = 110$ min, normalisé sur une fenêtre d'acquisition discrète).

On souhaite traiter ce signal avec le filtre numérique récursif du premier ordre :

$$y(n) = \alpha y(n-1) + (1-\alpha)x(n), \quad 0 < \alpha < 1$$

- 1°) Déterminer la fonction de transfert $H(Z)$ de ce filtre. Identifier son type. Donner la condition de stabilité sur α .
- 2°) Calculer la réponse $y(n)$ pour $x(n) = (0.9)^n u(n)$ et $\alpha = 0.5$.
- 3°) En Python, simuler $x(n)$ et $y(n)$ pour $n = 0, \dots, 50$ et $\alpha \in \{0.2, 0.5, 0.8\}$. Tracer les résultats et commenter l'effet de α sur le lissage du signal.

```

1 import numpy as np; import matplotlib.pyplot as plt
2 from scipy.signal import lfilter
3 N=51; n=np.arange(N); x=0.9**n
4 fig, axes = plt.subplots(1,3,figsize=(14,4),sharey=True)
5 for i, alpha in enumerate([0.2,0.5,0.8]):
6     y = lfilter([1-alpha],[1,-alpha],x)
7     axes[i].stem(n,x,linewidth='C0-',markerfmt='C0o',label='x(n)')
8     axes[i].stem(n,y,linewidth='C1-',markerfmt='C1s',label=f'y(n), a={alpha}')
9     axes[i].set_title(f'alpha={alpha}'); axes[i].legend(); axes[i].grid(
10    True)
11 plt.tight_layout(); plt.show()

```

- 4°) Quel est l'effet du paramètre α sur la résolution temporelle de la détection ? Quel compromis cela implique-t-il en imagerie TEP ?

Exercice 12

En résonance magnétique nucléaire (RMN), le signal de précession libre (FID – *Free Induction Decay*) est modélisé par :

$$s(n) = A e^{-n/\tau} \cos(\omega_0 n + \varphi)$$

avec $A = 1$, $\tau = 100$ échantillons, $\omega_0 = \pi/8$ rad/échantillon, $\varphi = 0$.

- 1°) Montrer que ce signal satisfait une équation aux différences récursive du second ordre. Donner les coefficients de cette récurrence.
- 2°) Écrire la fonction de transfert $H(Z)$ associée. Calculer ses pôles et les placer dans le plan complexe. À quelle condition le signal est-il stable ?

- 3°) En Python, générer $s(n)$ pour $n = 0, \dots, 200$ et tracer son évolution temporelle ainsi que son spectre (module de la TFD).

```
1 import numpy as np; import matplotlib.pyplot as plt
2 N=201; n=np.arange(N)
3 s = np.exp(-n/100)*np.cos(np.pi/8*n)
4 S = np.fft.fft(s, n=1024)
5 freqs = np.fft.fftfreq(1024)
6 fig, (ax1,ax2) = plt.subplots(2,1,figsize=(10,6))
7 ax1.plot(n,s); ax1.set_title('Signal FID (RMN)'); ax1.grid(True)
8 ax2.plot(freqs[:512],np.abs(S[:512]))
9 ax2.set_xlabel('Frequence normalisee'); ax2.grid(True)
10 plt.tight_layout(); plt.show()
```

- 4°) Comparer la largeur spectrale du pic de résonance avec la valeur théorique $1/(\pi\tau)$. Interpréter le résultat en termes de résolution fréquentielle en RMN.

SÉRIE 4 Transformée en Z

Exercice 1

Calculer les TZ des fonctions suivantes (donner le domaine de convergence) :

- $\delta[n - a]$ avec $\delta[n] = 1$ pour $n = 0$, 0 sinon
- $U[n] = 1, \forall n \geq 0$
- $U[n] - U[n - a], a \in \mathbb{N}$
- $U[n] \cdot \alpha^n$

Pour $x[n] = U[n] - U[n - a]$, calculer $x * x[n]$.

Exercice 2

Calculer la transformée en Z du signal porte discret $x[k] = \Pi_T[k]$:

- en appliquant directement la définition ;
- en utilisant l'échelon de Heaviside discret $\varepsilon[k]$ et le théorème du retard.

Comparer les deux résultats.

Exercice 3

Soit la fonction en Z :

$$F(z) = \frac{z^{-5}}{z + 2}$$

1. Calculer les développements de Laurent sur les domaines $D_1 = \{|z| > 2\}$ et $D_2 = \{|z| < 2\}$.
2. En déduire les inverses causale et anti-causale de $F(z)$.

Exercice 4

Soit l'EDF suivante :

$$y[k] - y[k - 1] - y[k - 2] = u[k]$$

1. Résoudre l'EDF par calcul direct.
2. Calculer les réponses causales pour les entrées $u[k] = 1, \forall k \geq 0$ et $u[k] = a^k, \forall k \geq 0$.
3. Retrouver les résultats en utilisant la TZ.

Exercice 5

Un compteur Geiger-Müller est modélisé par un système discret dont la sortie $y[n]$ représente le nombre de coups détectés par intervalle de temps. Le temps mort du détecteur est modélisé par l'équation aux différences :

$$y[n] = x[n] + 0.3y[n - 1] - 0.1y[n - 2]$$

où $x[n]$ est l'entrée (coups incidents) et $y[n]$ la sortie (coups enregistrés).

- 1°) Calculer la fonction de transfert $H(Z)$. Déterminer ses pôles et ses zéros. Ce système est-il stable ?
- 2°) Calculer la TZ de la réponse impulsionnelle $h[n]$. En déduire son expression temporelle.
- 3°) Si l'entrée est un échelon $x[n] = u[n]$, calculer $Y(Z)$ et en déduire $y[n]$. Interpréter physiquement la valeur finale.

4°) En Python, simuler $h[n]$ et la réponse à l'échelon pour $n = 0, \dots, 30$:

```

1 from scipy.signal import lfilter
2 import numpy as np; import matplotlib.pyplot as plt
3 b=[1]; a=[1,-0.3,0.1]; N=31; n=np.arange(N)
4 delta=(n==0).astype(float)
5 h=lfilter(b,a,delta)
6 y_step=lfilter(b,a,np.ones(N))
7 fig,(ax1,ax2)=plt.subplots(2,1,figsize=(9,6))
8 ax1.stem(n,h); ax1.set_title('Reponse impulsionnelle h[n]'); ax1.grid(True)
9 ax2.stem(n,y_step); ax2.set_title('Reponse a l echelon y[n]'); ax2.grid(
    True)
10 plt.tight_layout(); plt.show()

```

Exercice 6

En dosimétrie, les capteurs de rayonnement enregistrent souvent une perturbation sinusoïdale à 50 Hz provenant du réseau électrique. On souhaite concevoir un filtre coupe-bande numérique (*notch filter*) pour éliminer cette composante.

Le filtre est défini par :

$$H(Z) = \frac{1 - 2 \cos(\omega_0)Z^{-1} + Z^{-2}}{1 - 2r \cos(\omega_0)Z^{-1} + r^2Z^{-2}}$$

avec $\omega_0 = 2\pi \times 50/F_e$, $F_e = 1000$ Hz, $r = 0.95$.

- 1°) Identifier les pôles et les zéros de $H(Z)$. Montrer que les zéros sont exactement sur le cercle unité à la pulsation ω_0 .
- 2°) Calculer $|H(e^{j\omega})|$ et vérifier que $H(e^{j\omega_0}) = 0$.
- 3°) Écrire l'équation aux différences correspondante à ce filtre.
- 4°) En Python, tracer la réponse fréquentielle et simuler son application sur un signal somme d'un signal utile et d'un bruit à 50 Hz.

```

1 import numpy as np; import matplotlib.pyplot as plt
2 from scipy.signal import freqz, lfilter
3 Fe=1000; omega0=2*np.pi*50/Fe; r=0.95
4 b=[1,-2*np.cos(omega0),1]; a=[1,-2*r*np.cos(omega0),r**2]
5 w,H=freqz(b,a,worN=2048,fs=Fe)
6 t=np.arange(0,1,1/Fe)
7 sig=np.exp(-t/0.2)+0.5*np.sin(2*np.pi*50*t)
8 sig_f=lfilter(b,a,sig)
9 fig,axes=plt.subplots(3,1,figsize=(10,9))
10 axes[0].plot(w,20*np.log10(np.abs(H)+1e-10))
11 axes[0].set_title('Reponse frequentielle (dB)'); axes[0].grid(True)
12 axes[1].plot(t,sig,label='Signal brute'); axes[1].legend(); axes[1].grid(
    True)
13 axes[2].plot(t,sig_f,label='Signal filtre',color='green')
14 axes[2].legend(); axes[2].grid(True)
15 plt.tight_layout(); plt.show()

```

SÉRIE 5 Échantillonnage et Transformée de Fourier Discrète (TFD)

Exercice 1

- 1°) Fabriquer 4000 échantillons d'un signal x correspondant à la somme de 6 sinusoïdes d'amplitudes relatives telles que les rapports d'énergie ($20 \log_{10}(A/A_0)$) soient 0, -5, -10, -15, -20 et -25 dB, et de fréquences respectives $F_0, 2F_0, 3F_0, 4F_0, 5F_0, 6F_0$, avec $F_0 = 445$ Hz, échantillonné à $F_e = 16$ kHz. On suppose que 0 dB correspond à $A_0 = 1$.

```

1 import numpy as np
2 Fe=16000; F0=445; N=4000
3 t=np.arange(N)/Fe
4 dB=[0, -5, -10, -15, -20, -25]
5 A=[10**(d/20) for d in dB]
6 x=sum(A[i]*np.sin(2*np.pi*(i+1)*F0*t) for i in range(6))

```

- 2°) Calculer l'énergie du signal (directement ou en utilisant `np.linalg.norm`).
- 3°) Fabriquer 4000 échantillons d'un signal aléatoire centré (utiliser `np.random.rand` et `np.mean`). Calculer son énergie. Ajouter aux sinusoïdes x le bruit blanc b en lui donnant une énergie de 20 dB en dessous de celle des sinusoïdes : $10 \log_{10}(E_b/E_x) = -20$ dB.
- 4°) Afficher le signal x_b entre les échantillons 1100 et 1600 : `plt.plot(xb[1100:1600])`.
- 5°) Contrôler les axes avec `plt.axis([1100, 1600, -1, 3])` après `plt.plot(xb[:4000])`.
- 6°) En utilisant une TFD sur $N = 2048$ points, comparer les spectres X et X_b des signaux x et x_b en affichage $20 \log_{10}$:

```

1 import matplotlib.pyplot as plt
2 X=np.fft.fft(x,2048); Xb=np.fft.fft(xb,2048)
3 freqs=np.fft.fftfreq(2048,1/Fe)
4 plt.plot(freqs[:1024],20*np.log10(np.abs(X[:1024])+1e-10),label='X')
5 plt.plot(freqs[:1024],20*np.log10(np.abs(Xb[:1024])+1e-10),label='Xb')
6 plt.legend(); plt.grid(True); plt.show()

```

- 7°) Générer une porte rectangulaire de longueur $P = 5$ échantillons.
- 8°) Calculer la TFD de la porte sur $N = 1000$ points : `Pf = np.fft.fft(P, 1000)`.
- 9°) Afficher le module de la transformée : `plt.plot(np.abs(Pf))`.
- 10°) À quoi est due la symétrie observée ?
- 11°) Quelle est la largeur $\Delta\nu$ du lobe principal ? (Utiliser `np.argmax`.)
- 12°) Afficher le module en décibels entre 0 et 15 dB.
- 13°) Vérifier qu'il y a environ 13 dB entre le max du lobe principal et le max du lobe secondaire. Vérifier que pour $P = 50$ échantillons, on a toujours environ 13 dB.

Exercice 2

Soit le signal :

$$s(t) = A \cos(2\pi f_1 t) + 3A \cos(2\pi \cdot 2f_1 t) + 2A \cos(2\pi \cdot 3f_1 t) + 4A \cos(2\pi \cdot 4f_1 t), \quad f_1 = 10 \text{ Hz}$$

Le signal $s_e(t)$ représente $s(t)$ échantillonné (échantillonnage parfait) avec une fréquence f_e .

- 1°) Donner l'expression de $S(f)$ (Transformée de Fourier de $s(t)$).
- 2°) Donner l'expression de $S_e(f)$ (Transformée de Fourier de $s_e(t)$).
- 3°) Pour $A = 1$ et $f_e = 100$ Hz, représenter sur deux graphes côte-à-côte les spectres de $S(f)$ et $(1/f_e) \cdot S_e(f)$ entre 0 et 150 Hz.

- 4°) Expliquer les différences entre $S(f)$ et $S_e(f)$. Quel est l'effet de l'échantillonnage sur le spectre du signal ?
- 5°) À partir des échantillons du signal numérisé, pourra-t-on restituer correctement $s(t)$? Pourquoi ?

Exercice 3

Un détecteur à scintillation NaI(Tl) couplé à un photomultiplicateur acquiert un signal numérique correspondant à la superposition de deux impulsions gamma de ^{241}Am (59.5 keV) et de ^{133}Ba (81 keV). Le signal numérisé est modélisé par :

$$x[n] = A_1 \sin\left(\frac{2\pi f_1 n}{F_e}\right) + A_2 \sin\left(\frac{2\pi f_2 n}{F_e}\right)$$

avec $f_1 = 59.5$ kHz, $f_2 = 81$ kHz, $A_1 = 1$, $A_2 = 0.7$, $F_e = 500$ kHz, $N = 2048$ points.

- 1°) Quelle est la condition de Shannon pour ce signal ? Est-elle respectée ?
- 2°) Calculer la résolution fréquentielle $\Delta f = F_e/N$ de la TFD. Est-il possible de distinguer les deux composantes ?
- 3°) En Python, générer $x[n]$, calculer sa TFD et tracer son spectre en amplitude. Identifier les deux pics.

```

1 import numpy as np; import matplotlib.pyplot as plt
2 Fe=500e3; N=2048; f1,f2=59.5e3,81e3; A1,A2=1.0,0.7
3 n=np.arange(N)
4 x=A1*np.sin(2*np.pi*f1*n/Fe)+A2*np.sin(2*np.pi*f2*n/Fe)
5 X=np.fft.fft(x); freqs=np.fft.fftfreq(N,1/Fe)
6 plt.plot(freqs[:N//2]/1e3,np.abs(X[:N//2]))
7 plt.xlabel('Frequence (kHz)'); plt.title('Spectre TFD -- NaI(Tl)')
8 plt.grid(True); plt.show()

```

- 4°) Appliquer un fenêtrage de Hann avant la TFD. Comparer le spectre obtenu avec et sans fenêtrage.
- 5°) En doublant N à 4096 points (*zero-padding*), peut-on améliorer la résolution spectrale ? Discuter.

Exercice 4

En imagerie par rayons X, un détecteur linéaire échantillonne un profil de transmission $I(x)$ le long d'une rangée de pixels. Le profil est modélisé par :

$$I(x) = I_0 \left[1 - 0.5 \cos\left(\frac{2\pi x}{D_1}\right) - 0.3 \cos\left(\frac{2\pi x}{D_2}\right) \right]$$

avec $D_1 = 2$ mm, $D_2 = 8$ mm, $I_0 = 1000$, $x \in [0, 100]$ mm.

- 1°) Quelle est la fréquence spatiale maximale de $I(x)$? En déduire le pas d'échantillonnage maximal Δx_{\max} selon le théorème de Shannon.
- 2°) Comparer les reconstructions obtenues avec $\Delta x = 0.5$ mm, $\Delta x = 1$ mm et $\Delta x = 3$ mm. En Python, tracer les trois cas.

```

1 import numpy as np; import matplotlib.pyplot as plt
2 I0=1000; D1=2e-3; D2=8e-3
3 x_c=np.linspace(0,0.1,10000)
4 I_c=I0*(1-0.5*np.cos(2*np.pi*x_c/D1)-0.3*np.cos(2*np.pi*x_c/D2))
5 fig,axes=plt.subplots(1,3,figsize=(14,4),sharey=True)
6 for ax,dx,lbl in zip(axes,[0.5e-3,1e-3,3e-3],
7                       ['dx=0.5mm','dx=1mm (Shannon)','dx=3mm (sous-ech.)']):

```

```
8     x_s=np.arange(0,0.1,dx)
9     I_s=I0*(1-0.5*np.cos(2*np.pi*x_s/D1)-0.3*np.cos(2*np.pi*x_s/D2))
10    ax.plot(x_c*1e3,I_c,'k-',alpha=0.4,label='continu')
11    ax.stem(x_s*1e3,I_s,linewidth='C1-',markerfmt='C1o',label='echantillons')
12    ax.set_title(lbl); ax.set_xlabel('x (mm)'); ax.legend(); ax.grid(True)
13 plt.tight_layout(); plt.show()
```

- 3°) Interpréter le phénomène de repliement (*aliasing*) dans le cas du sous-échantillonnage. Quel artefact cela génère-t-il en imagerie médicale (CT-scan) ?
- 4°) On souhaite appliquer un filtre anti-repliement avant l'échantillonnage à $\Delta x = 3$ mm. Quelle doit être la fréquence de coupure ? En Python, appliquer ce filtre et comparer le résultat.