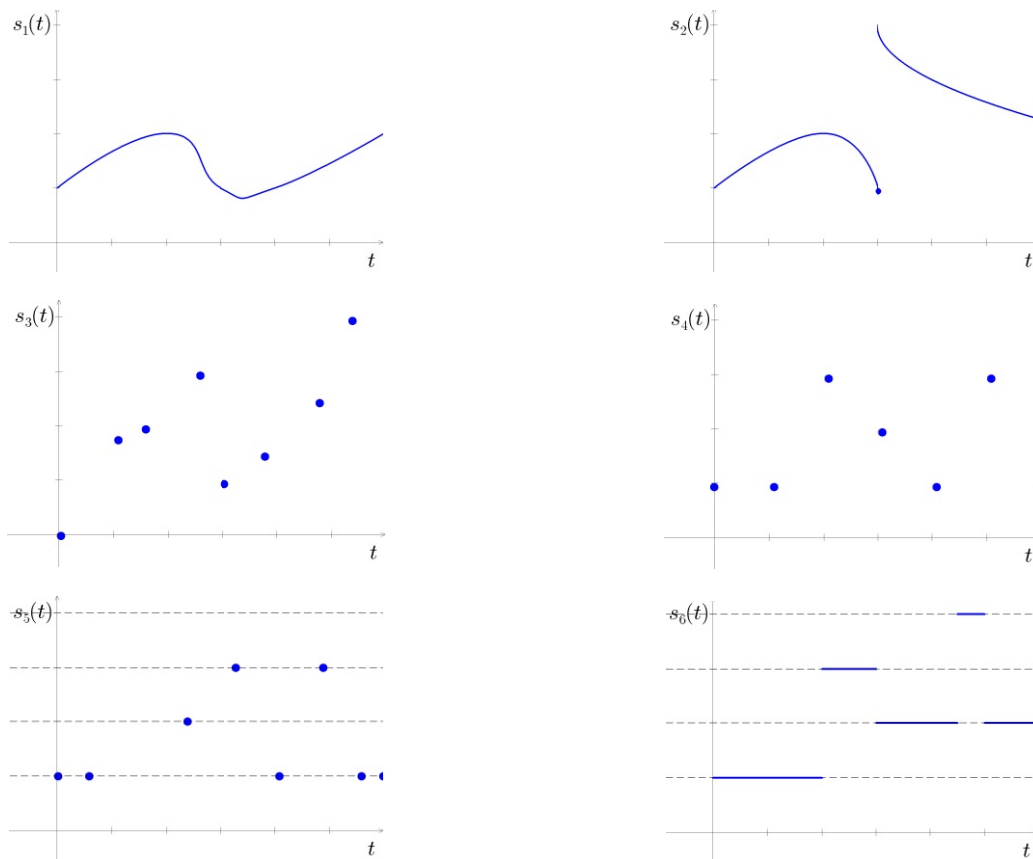


Classification des signaux unidimensionnels

1 En théorie

Exercice 1 (*). Pour chacun des signaux s_1 à s_6 représentés ci-dessous, répondre aux questions suivantes. Les pointillés sur certaines représentations indiquent les seuls niveaux disponibles pour le signal.



- Parmi les signaux suivants, donner ceux qui sont :
 - continus ?
 - discrets ?
 - analogiques ?
 - quantifiés ?
 - échantillonnés ?
 - numériques ?
- Parmi ces signaux, lequel est issu de l'échantillonnage d'un signal continu avec une période d'échantillonnage T_e fixe ?

Exercice 2 (*). Soit le signal f défini par

$$f(t) = A \sin(2\pi\lambda t + \phi), \text{ avec } A, \lambda \in \mathbb{R}^{+*} \text{ et } \phi \in \mathbb{R}.$$

- Représenter schématiquement le signal f et en donner certaines de ses caractéristiques (temporelles et morphologiques).
- Calculer l'énergie totale de f .

3. Calculer la puissance moyenne totale de f .
4. Donner sa représentation spectrale.

Exercice 3 (*). Reprendre l'exercice précédent avec les fonctions suivantes (sauf la question 4).

1. Le signal carré g , impair, T -périodique et d'amplitude $A > 0$ défini par :

$$g(x) = \begin{cases} A & \text{si } x \in \left]0; \frac{T}{2}\right[\\ -A & \text{si } x \in \left]\frac{T}{2}; T\right[. \end{cases}$$

2. Le signal h défini par

$$h(t) = \frac{\Gamma(t)}{(t+1)^2} \quad \forall t \in \mathbb{R}.$$

2 En pratique

Exercice 4 ()**. L'utilisation de Python est indispensable pour cette partie. Les fichiers `my_signal_td1.py` et `td1_squelette.py` sont disponibles sur Moodle. On a vu en CM qu'un signal sonore peut être modélisé par une fonction définie sur \mathbb{R}^+ , et que l'échantillonnage de cette fonction puis sa quantification permet sa numérisation. On s'intéresse dans cette partie à des fichiers sons au format `.wav` échantillonnés à la fréquence $f_e = 44100$ Hz. L'import des libraires `sounddevice` et `scipy.io.wavfile` permet de lire, de créer et de jouer directement des sons à partir de Python.

1. Compléter les fonctions `read_test`, qui s'appuie sur la méthode `read` de `scipy.io.wavfile`, et qui, à un fichier sonore `wav`, renvoie l'échantillon issu du son du fichier s'il a été échantillonné à la fréquence $f_e = 44100$ Hz, et rien sinon.
2. Utiliser les fonctions `read_test` et `play` pour lire à partir de Python le fichier `forest5s.wav`. Remarquer que ce fichier contient deux pistes et les comparer.
3. Compléter la fonction `plot`, qui, à un vecteur de données `data` correspondant à un signal échantillonné, affiche la représentation de ce signal en fonction du temps (et donc en tenant compte de la fréquence d'échantillonnage). Tester la fonction sur `data = read_test('forest5s.wav')`.
4. (a) Compléter la fonction `sinus`, qui, à une fréquence `freq` et un instant `t`, retourne l'image de t par la fonction $t \mapsto \sin(2\pi \times \text{freq} \times t)$.
 (b) Compléter les méthodes `samples` et `plot` de la classe `my_signal_td1.py`. Tester la fonction sur `data = read_test('forest5s.wav')`.
 (c) En utilisant la méthode `partial` et la fonction `sinus`, créer la fonction $t \mapsto \sin(2\pi \times 440 \times t)$ ¹, puis l'objet `Signal` correspondant. L'utiliser pour représenter le signal sur un intervalle bien choisi, puis pour jouer le son associé à ce signal.
5. Compléter la fonction `chord`, qui, à une liste `frequency_list` de fréquences et un instant `t`, retourne la somme des sinus à ces fréquences. Utiliser cette fonction pour représenter et jouer l'accord majeur de Do, constitué des notes suivantes de même intensité : un do à 262 Hz, un mi à 330 Hz et un sol à 392 Hz.
6. Le code DTMF (dual-tone multi-frequency) ou FV (Fréquences Vocales) est une combinaison de fréquences utilisée pour la téléphonie fixe classique. Ces codes sont émis lors de la pression sur une touche du clavier téléphonique, et sont utilisés pour la composition des numéros de téléphones. Techniquement, chaque touche d'un téléphone correspond à un couple de deux fréquences audibles qui sont émises simultanément selon la table ci-dessous. (Wikipedia).

¹Le *la 440* est une note de musique utilisée comme hauteur de référence et de fréquence 440 Hz (Wikipedia)

	1209 Hz	1336 Hz	1477 Hz	1633 Hz
697 Hz	1	2	3	A
770 Hz	4	5	6	B
852 Hz	7	8	9	C
941 Hz	*	0	#	D

Compléter la fonction DTMF qui demande à un utilisateur son numéro de téléphone et qui permet de représenter et de jouer le signal associé, par la stratégie de votre choix.