

EXAMEN TRAITEMENT NUMÉRIQUE DU SIGNAL **Aucun** document autorisés

Durée totale de l'épreuve : 2 heures

1. ANALYSE SPECTRALE

1.1. Questions théoriques.

- (1) Soit  $x \in \mathbb{R}^N$  un signal discret Fini de taille  $N$ . Donner sa transformée de Fourier  $\hat{x}$ .
- (2) Cette transformée est-elle continue ? Discrète ? Périodique (préciser la période si oui) ?
- (3) Donner la transformée de Fourier inverse.
- (4) Soit le signal porte de fréquence de coupure  $\nu_0$ , donné dans le domaine fréquentiel :

$$\hat{x}(\nu) = \begin{cases} 1 & \text{si } |\nu| < \nu_0 \\ 0 & \text{sinon} \end{cases}$$

Exprimer un filtre passe-bande laissant intacte les fréquences entre  $[\nu_0, \nu_1]$  et annulant les autres, en fonctions de deux passe-bas de fréquence de coupure respective  $\nu_0$  et  $\nu_1$

- (5) Calculer les réponses impulsionnelles  $h_0$  et  $h_1$  des filtres passe-bas précédents.
- (6) En déduire la réponse impulsionnelle du passe bande.

1.2. Questions pratiques. Écrire un script matlab permettant de

- (1) Lire un fichier audio stéréo au format .wav (exemple, le fichier 'rock.wav'), en conservant son taux d'échantillonnage.
- (2) Le convertir en un signal monophonique
- (3) Représenter ses échantillons sur une échelle adaptée
- (4) Calculer et représenter la module de son spectre avec une échelle adaptée en fréquence.
- (5) On représente sur la figure 1 le spectre d'un signal audio, entre les fréquences 210 Hz et 450 Hz. À l'aide du tableau 1, quelles sont les quatre notes principales jouées dans ce morceau ?

Do	Do #	Re	Re #	Mi	Fa	Fa #	Sol	Sol #	La	La #	Si
262 Hz	277 Hz	294 Hz	311 Hz	330 Hz	349 Hz	370 Hz	392 Hz	415 Hz	440 Hz	466 Hz	494 Hz

TABLE 1. Correspondance notes / fréquences

2. FILTRAGE

2.1. Questions théoriques. Dans la suite, on considère un filtre numérique de réponse impulsionnelle  $h$ ,  $x$  un signal numérique, et on note  $y$  le signal obtenu en filtrant  $x$  par  $h$ .

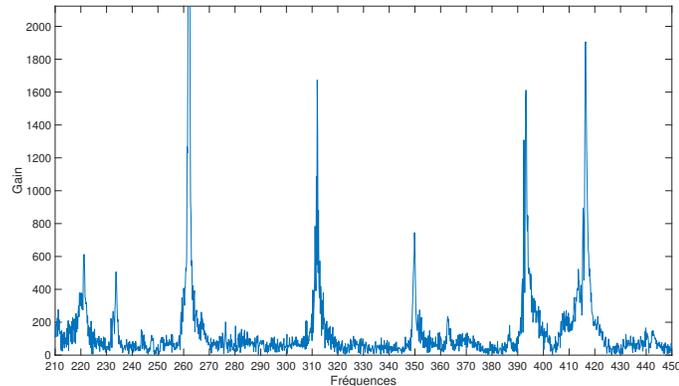


FIGURE 1. Spectre d'un signal audio : zoom entre les fréquences 210 Hz et 450 Hz.

#### 2.1.1. Questions générales.

- (1) Rappeler l'équation de filtrage (ou équation aux différences) dans le domaine temporel
- (2) Qu'est-ce qu'un filtre réalisable ?
- (3) Un filtre qui admet une transformée de Fourier est-il nécessairement réalisable ?
- (4) Qu'est-ce que la fonction de transfert d'un filtre ?

### 3. FILTRAGE

**3.1. Questions théoriques.** Dans la suite, on considère un filtre numérique de réponse impulsionnelle  $h$ ,  $x$  un signal numérique, et on note  $y$  le signal obtenu en filtrant  $x$  par  $h$ .

#### 3.1.1. Questions générales.

- (1) Rappeler l'équation de filtrage (ou équation aux différences) dans le domaine temporel
- (2) Qu'est-ce qu'un filtre réalisable ?
- (3) Qu'est-ce qu'un filtre FIR ? Comment s'écrit son équation aux différences ?
- (4) Qu'est-ce qu'un filtre IIR ? Comment s'écrit son équation aux différences ?
- (5) Donner les avantages et les inconvénients pratiques des approches FIR et IIR.

**3.2. Questions pratiques.** Compléter le script matlab de la question 1.2 afin de permettre d'implémenter la synthèse d'un filtre FIR réalisable par fenêtrage, et d'effectuer le filtrage dans le domaine temporel.

Vous pouvez utiliser comme bon vous semble les fonctions matlab suivantes. Cette liste n'est pas exhaustive, vous pouvez utiliser d'autres fonctions que vous connaissez.

- `[s,fe] = audioread('file.wav')` pour lire le fichier `file.wav`. `s` est une matrice  $N$  lignes 2 colonnes. La colonne 1 correspond au canal gauche et la colonne 2 au canal droit. `fe` est la fréquence d'échantillonnage du fichier.
- `T = length(x)` qui renvoie le nombre d'éléments  $T$  du vecteur `x`

- `V = linspace(a,b,N)` qui crée un vecteur `V` de taille `N` éléments espacés selon une échelle linéaire, allant de `a` à `b`
- `xfft = fft(x)` qui renvoie la transformée de Fourier `xfft` de `x`.
- `xfft_shift = fftshift(xfft)` réordonne le vecteur `xfft` afin que la fréquence 0 soit au centre du vecteur `xfft_shift`.
- `fen = fenetre('nom',T)` renvoie dans le vecteur `fen` la fenêtre correspondant à `'nom'`, de taille `T`
- `Z = zeros(N,1)` crée un vecteur `Z` de taille `N` contenant que des 0.