Rapport TP

TL Traitement Du Signal

Réalisé par:
Nabil ABDELOUAHED
Yann MARTIN

May 21, 2025

Contents

1 Partie 1: Quelques opérations de base sur les signaux

1.1 Signal numérique de synthèse

1.1.1 Génération du signal

Un signal sinusoïdal de fréquence f_0 est généré par la fonction suivante:

$$x[n] = \sin\left(2\pi f_0 \frac{n}{f_e}\right)$$

où f_e est la fréquence d'échantillonnage et N le nombre d'échantillons.

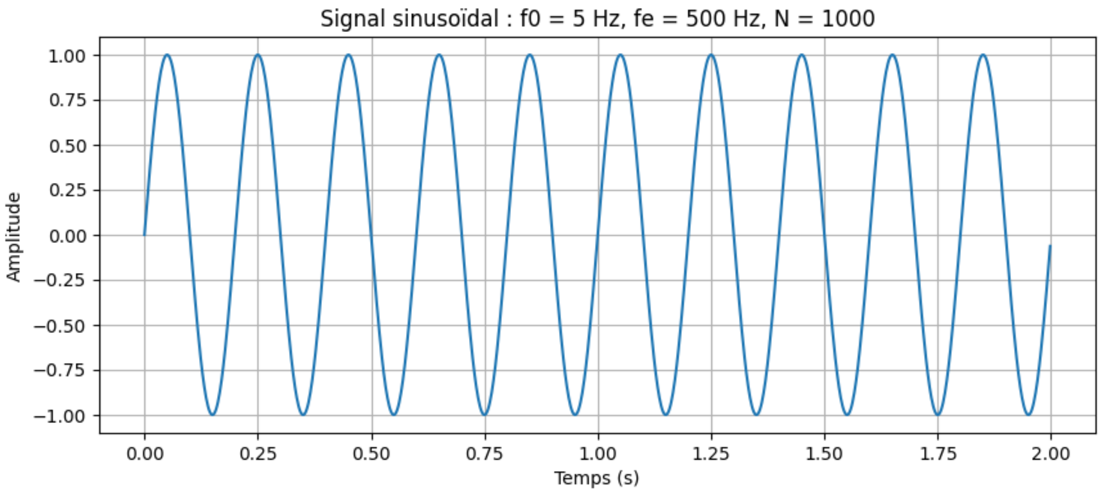


Figure 1: Signal sinusoïdal échantillonné

1.1.2 Énergie et puissance

L'énergie d'un signal discret $x[n]$ est donnée par :

$$E = \sum_{n=0}^{N-1} x[n]^2$$

Et la puissance moyenne par :

$$P = \frac{1}{N} \sum_{n=0}^{N-1} x[n]^2$$

Considérons un signal sinusoïdal discret de la forme :

$$x[n] = A \cdot \sin\left(2\pi f_0 \frac{n}{f_e}\right)$$

La puissance moyenne théorique d'un signal périodique est calculée par la formule:

$$P = \lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} x[n]^2$$

En utilisant l'identité trigonométrique :

$$\sin^2(\theta) = \frac{1 - \cos(2\theta)}{2}$$

on obtient :

$$x[n]^2 = A^2 \cdot \frac{1 - \cos\left(4\pi f_0 \frac{n}{f_e}\right)}{2}$$

Ainsi, la puissance devient :

$$P = \lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} A^2 \cdot \frac{1-\cos\left(4\pi f_0 \frac{n}{f_e}\right)}{2} = \lim_{N\to\infty} \frac{A^2}{2} - \frac{A^2}{2N} \sum_{n=0}^{N-1} \cos\left(4\pi f_0 \frac{n}{f_e}\right) = \frac{A^2}{2}$$

Donc dans notre cas la puissance moyenne théorique est égale à 0.5.

Pour la puissance moyenne calculée numériquement pour le signal échantillonné on a la méme formule mais sans la limite:

$$P = \frac{A^2}{2} - \frac{A^2}{2N} \sum_{n=0}^{N-1} \cos\left(4\pi f_0 \frac{n}{f_e}\right)$$

Cas idéal : si N est un multiple entier de la période du signal (i.e., N couvre un nombre entier de périodes), alors la somme des cosinus s'annule :

$$\sum_{n=0}^{N-1} \cos(4\pi f_0 t) = 0 \quad \Rightarrow \quad P = \frac{A^2}{2}$$

Cas général : si N n'est pas un multiple exact de la période, la somme ne s'annule pas et on observe une légère déviation de la puissance par rapport à $\frac{A^2}{2}$. Cela est dû au fait que le signal est tronqué entre deux points non symétriques.

En variant N on trouve plusieurs valeurs de la puissance moyenne qui restent proches de la valeur théorique:

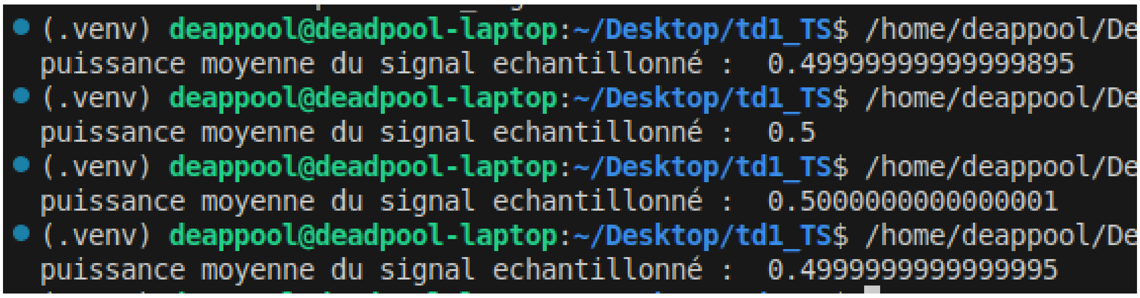


Figure 2: Énergie et puissance du signal

1.1.3 Quantification

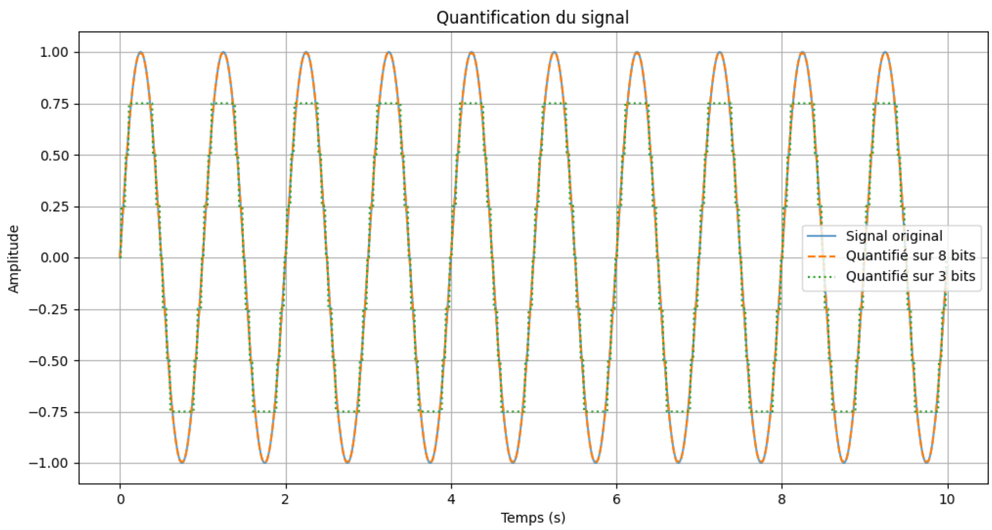


Figure 3: Quantification du signal à 3 et 8 bits

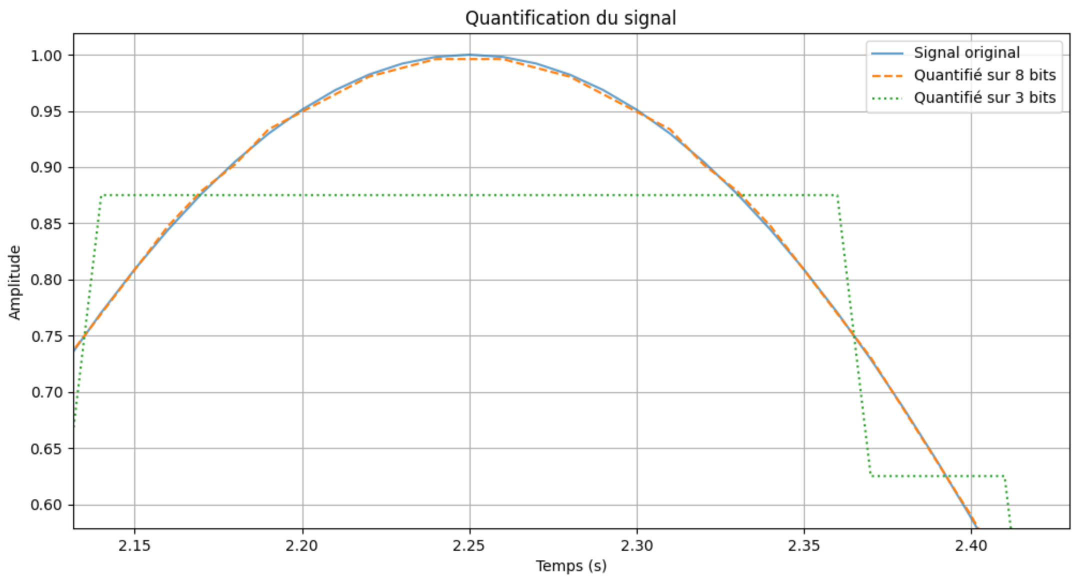


Figure 4: Zoom sur la quantification du signal à 3 et 8 bits

Le signal à 8 bits suit mieux la forme continue du signal d'origine mais on voit quand méme quelques erruers. À 3 bits, les marches sont plus visibles et le signal produit est significativement moins fidéle au signal d'origine.

SNR (Signal-to-Noise Ratio) :

$$\mathrm{SNR} = 10\log_{10}\left(\frac{E_{\text{signal}}}{E_{\text{bruit}}}\right)$$

```
Énergie du bruit de quantification (8 bits): 0.006400
SNR (8 bits): 49,92 dB

Énergie du bruit de quantification (3 bits): 6.236655
SNR (3 bits): 19,82 dB
```

Figure 5: SNR pour chaque niveau de quantification

On remarque que l'energie du bruit est plus élevée pour le signal quantifié à 3 bits. Le résultat est logique car on voit sur le graphe que ce signal est plus éloigné du signal d'origine comparé au signal quantifié à 8 bits. On a la méme conclusion en raisonant sur le SNR: SNRq8 > SNRq3.

1.2 Signal audio

1.2.1 Enregistrement

Les mots « Bonjour » et « ChatGpt » ont été enregistrés via Audacity.

1.2.2 Restitution à différentes fréquences

L'audio est lu à f_e, $2f_e$ et $\frac{f_e}{2}$.
Effets observés :

- **Durée :** doubler la fréquence de restitution divise la durée par deux (voix accélérée), tandis que la diviser par deux double la durée (voix ralentie).
- **Hauteur :** multiplier la fréquence de restitution rend la voix plus aiguë (fréquences doublées), la diminuer la rend plus grave (fréquences divisées).

- **Applications :** cette manipulation illustre le principe de transposition spectrale et d'étirement temporel.

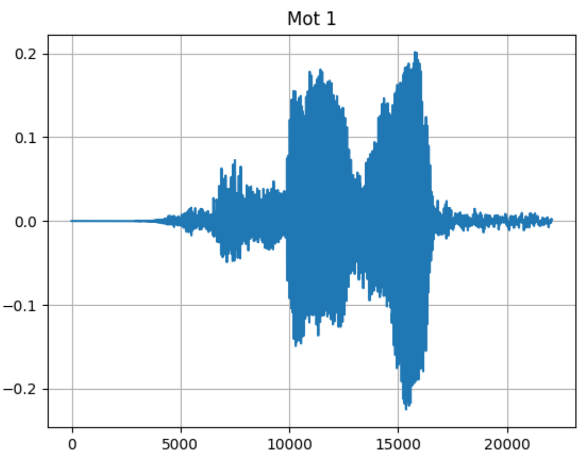
1.2.3 Quantification du signal audio

- À 3 bits : le son devient rugueux et très bruité, fortement altéré.
- À 8 bits : la voix reste compréhensible mais moins naturelle.
- À 16 bits (original) : qualité fidèle.

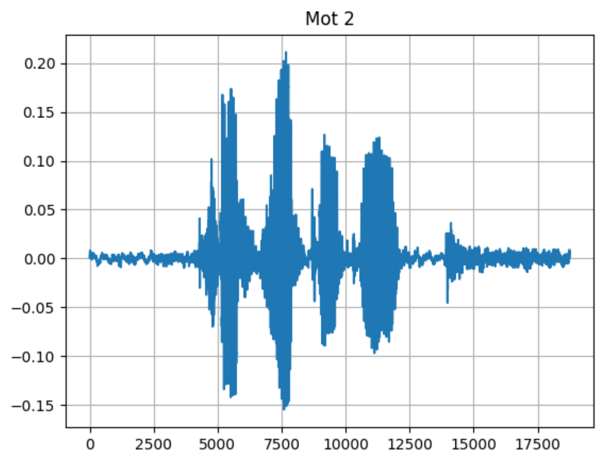
1.2.4 Extraction et séparation de mots

Après repérage visuel, les deux mots ont été extraits via tranches temporelles, puis enregistrés :

```
sf.write("mot1.wav", mot1, fe)
sf.write("mot2.wav", mot2, fe)
```



(a) Mot 1: "Bonjour"



(b) Mot 2: "ChatGpt"

Figure 6: Séparation des mots dans le signal audio

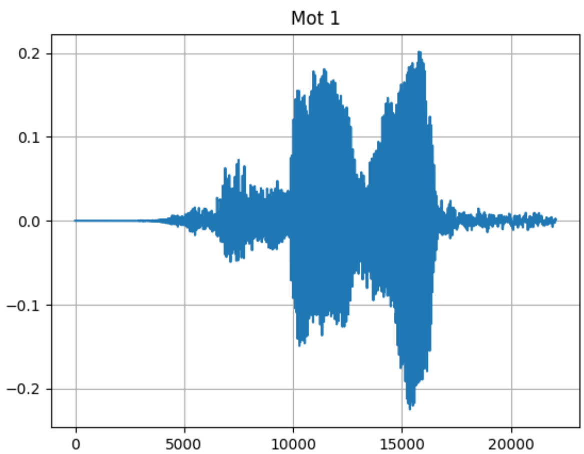
1.2.5 Extraction de mots

Aprés avoir identifié l'intervalle de temps pour chaque mot dans le signal enregistré on utilise ce code pour séparer les deux mots:

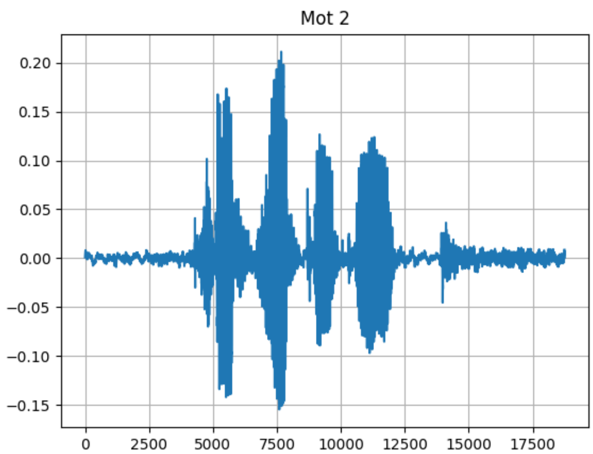
```
n1 = int(0.3 * fe)     #0.3s
n2   = int(2.3 * fe)   #2.3s
n3 = int(4 * fe)       #4s

mot1 = y[n1:n2]
mot2 = y[n2+1:n3]

print("Mot 1 :")
sd.play(mot1, fe)
sd.wait()

print("Mot 2 :")
sd.play(mot2, fe)
sd.wait()
```

Puis on enregistre les deux mots séparément dans des fichiers .wav :

```
sf.write("mot1.wav", mot1, fe)
sf.write("mot2.wav", mot2, fe)
```



(a) Mot 1



(b) Mot 2

Figure 7: Séparation des deux mots