Rapport TP

TL Traitement Du Signal

Réalisé par:
Nabil ABDELOUAHED
Yann MARTIN

May 27, 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.

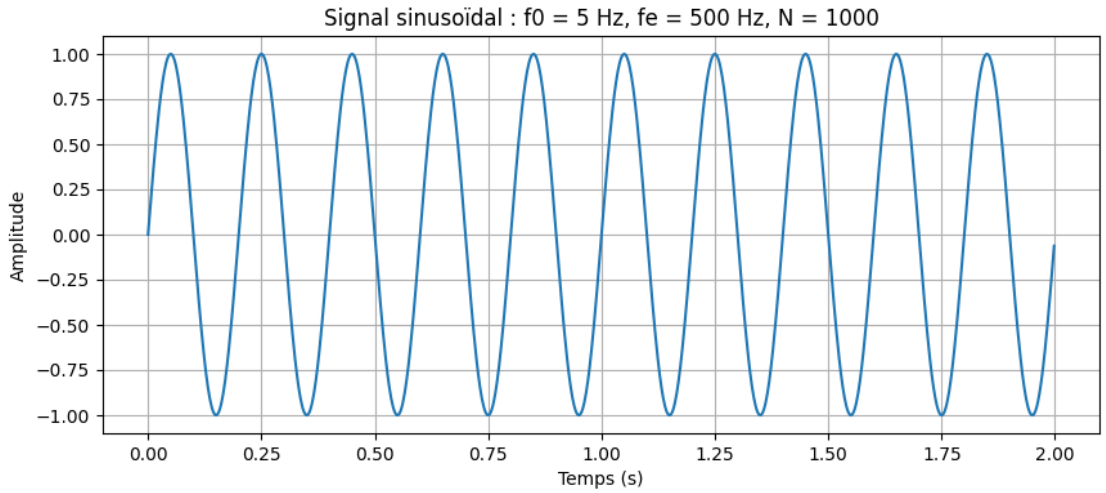


Figure 1: Signal sinusoïdal échantillonné

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 :

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

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:

```
(.venv) deappool@deadpool-laptop:~/Desktop/td1_TS$ /home/deappool/De
puissance moyenne du signal echantillonné :  0.49999999999999895
(.venv) deappool@deadpool-laptop:~/Desktop/td1_TS$ /home/deappool/De
puissance moyenne du signal echantillonné :  0.5
(.venv) deappool@deadpool-laptop:~/Desktop/td1_TS$ /home/deappool/De
puissance moyenne du signal echantillonné :  0.5000000000000001
(.venv) deappool@deadpool-laptop:~/Desktop/td1_TS$ /home/deappool/De
puissance moyenne du signal echantillonné :  0.4999999999999995
```

Figure 2: Énergie et puissance du signal

1.1.3 Quantification

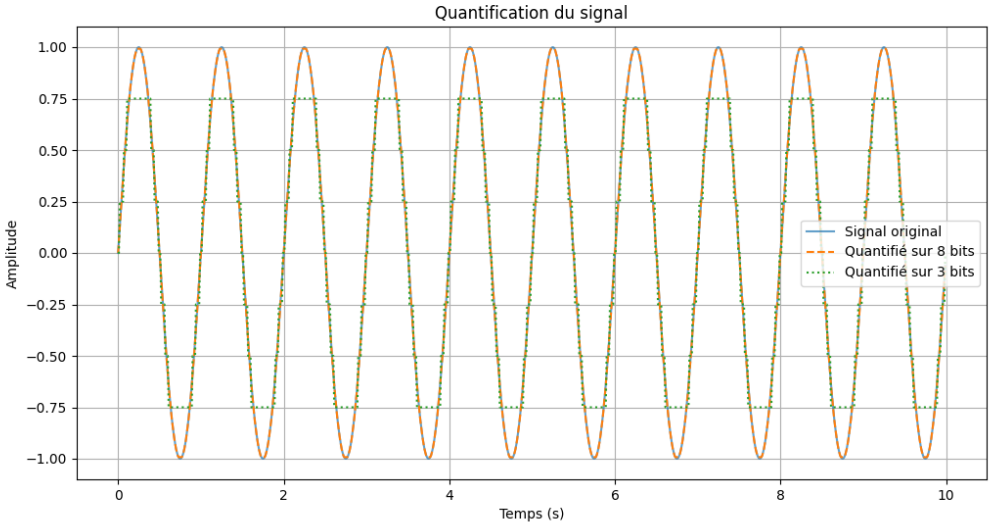


Figure 3: Quantification du signal à 3 et 8 bits

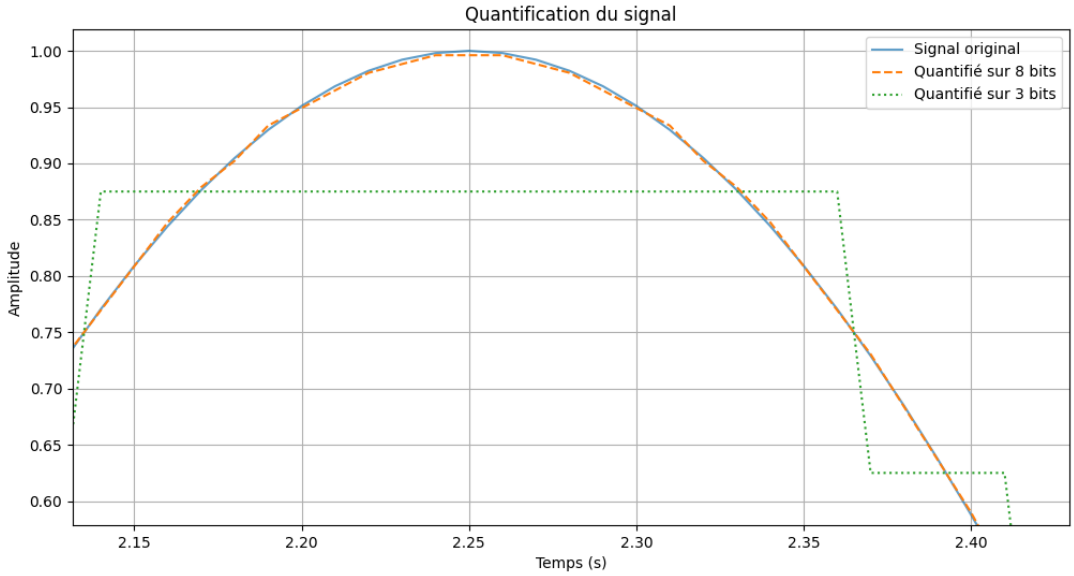


Figure 4: Zoom sur la quantification du signal à 3 et 8 bits

Pour la quantification à 3 bits, on retrouve bien 8 niveaux de quantification.
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 (voir figure 6).

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).

Lorsque la fréquence de restitution est trop grande ou trop petite comparée à la fréquence d'échantillonnage, le son devient incompréhensible. Le son est trop rapide ou trop lent, mais aussi on ne reconnaît plus les mots. Cela s'explique notamment par le déplacement des formants, c'est-à-dire des pics de résonance caractéristiques des voyelles et consonnes, qui changent de position dans le spectre. Leur modification rend la parole méconnaissable, car ce sont eux qui permettent d'identifier les sons du langage.

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.

1.2.4 Extraction et séparation de mots

Après repérage visuel, les deux mots ont été extraits via tranches temporelles.

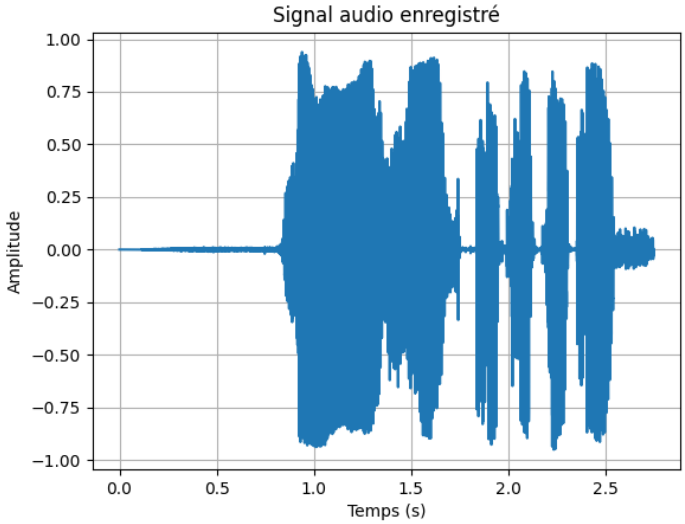
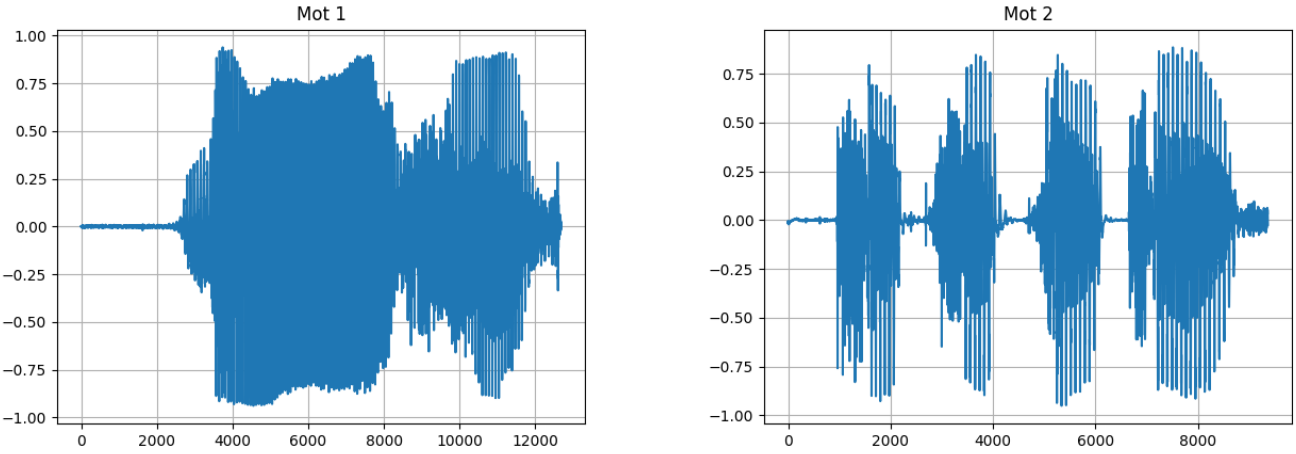


Figure 6: Signal audio enregistré



(a) Mot 1: "Bonjour"

(b) Mot 2: "ChatGpt"

Figure 7: Séparation des mots dans le signal audio

2 Partie 2: Classification des signaux

2.1 Exemple de calcul théorique

Soit un signal sinusoïdal discret défini par :

$$x[n] = A\sin(2\pi f n T_e)$$

L'autocorrélation théorique du signal discret est définie par :

$$R_{xx}[k] = \lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} x[n] \cdot x[n+k]$$

En remplaçant l'expression de $x[n]$:

$$R_{xx}[k] = \lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} A\sin(2\pi f n T_e) \cdot A\sin(2\pi f (n+k) T_e)$$

$$= A^2 \cdot \lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N-1} \sin(2\pi f n T_e) \cdot \sin(2\pi f n T_e + 2\pi f k T_e)$$

En utilisant l'identité trigonométrique :

$$\sin(a)\sin(b) = \frac{1}{2}[\cos(a-b) - \cos(a+b)]$$

On a :

$$\sin(2\pi f n T_e) \cdot \sin(2\pi f n T_e + 2\pi f k T_e) = \frac{1}{2}\left[\cos(2\pi f k T_e) - \cos(4\pi f n T_e + 2\pi f k T_e)\right]$$

$$\Rightarrow R_{xx}[k] = \frac{A^2}{2}\cos(2\pi f k T_e)$$

car la somme de $\cos(4\pi f n T_e + \cdot)$ sur une grande fenêtre N tend vers 0.
Conclusion : l'autocorrélation théorique du signal sinusoïdal discret est donnée par :

$$\boxed{R_{xx}[k] = \frac{A^2}{2}\cos(2\pi f k T_e)}$$

où :

- A est l'amplitude du signal,
- f est la fréquence du signal en Hz,
- $T_e = \frac{1}{f_e}$ est la période d'échantillonnage,
- $k \in \mathbb{Z}$ est le décalage (lag) discret.

2.2 Programmation

On calcule l'autocorrélation pour un signal sinusoïdal échantillonné en utilisant la formule :

$$R_{xx}[k] = \frac{1}{N} \sum_{n=0}^{N-1} x[n] \cdot x[n+k]$$

```
def autocorrelation_manual(x):
N = len(x)
r = np.zeros(2*N - 1)
lags = np.arange(-N + 1, N)
for k in range(-N + 1, N):
    somme = 0
    for n in range(N - abs(k)):
        somme += x[n] * x[n + k] if k >= 0 else x[n - k] * x[n]
    r[k + N - 1] = somme
return lags, r
```

On compare avec l'autocorrelation calculée avec numpy et on trace la différence entre les deux résultats pour visualiser les écarts. Pour convertir les abcisses en secondes, on utilise la formule :

$$\mathrm{t} = \mathrm{k}.T_e = \frac{\mathrm{k}}{f_e}$$

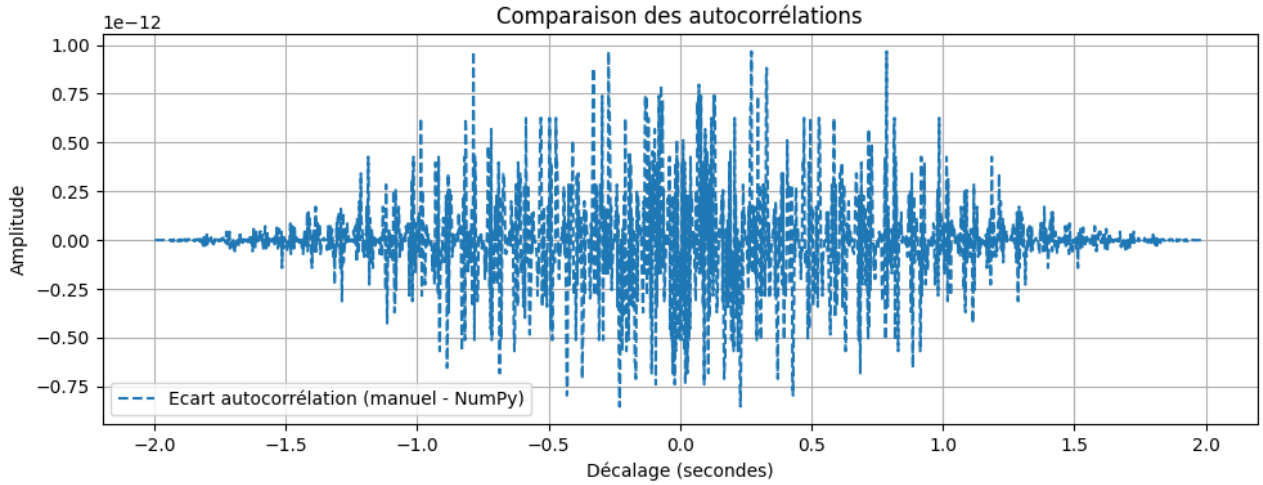


Figure 8: Ecart entre l'autocorrélation manuelle et celle de numpy

L'ecart est trop petit entre les deux méthodes (de l'ordre de 10^{-12}). Ceci est dû à la précision numérique des calculs et au fait que l'autocorrelation numpy est calculée différemment en utilisant plusieurs techniques d'optimisation.

On calcule le rapport des énergies entre la différence entre les deux méthodes et l'autocorrelation de numpy:

$$\text{rapport} = \frac{\sum_{k=0}^{N-1} \text{diff}[k]^2}{\sum_{k=0}^{N-1} \text{autocorr}[k]^2}$$

On trouve plusieurs valeurs de rapport pour différentes valeurs de N. Ceci est dû à la variation de l'énergie du signal en fonction de N, car l'autocorrelation et l'énergie sont sensibles à la longueur du signal (somme de 0 à N-1). On rajoute donc des valeurs au numérateur et au dénominateur dans le calcul de ce rapport (qui ne sont pas équivalentes vu l'écart entre les deux méthodes) ce qui expliques des valeurs de rapport différentes.

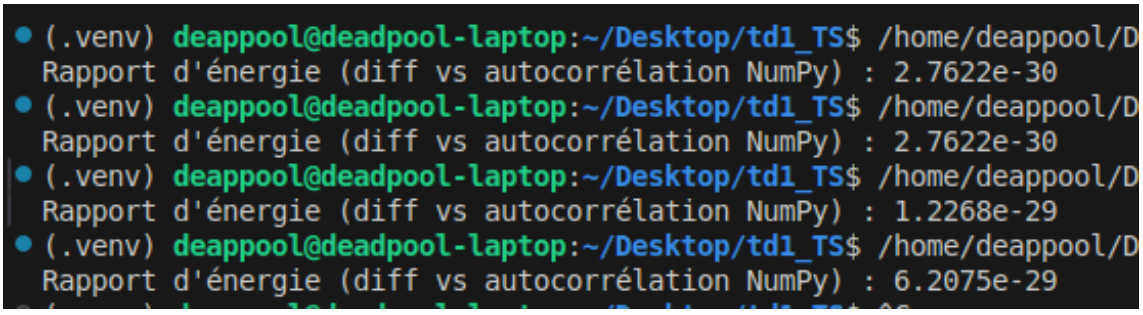


Figure 9: Valeurs du rapport pour plusieurs valeurs de N

2.3 Application à la classification de quelques signaux simples

On génére les signaux demandés avec le code suivant :

```python
# Parametres communs
fe = 11000           # frequence d'echantillonnage
duration = 1      # duree en secondes
t = np.linspace(0, duration, int(fe * duration), endpoint=False)

# 1. Signal sinusoidal de 200 Hz
f = 200
sinus = np.sin(2 * np.pi * f * t)

# 2. Signal triangulaire centre en 0 (serie de Fourier avec 10 termes)
tri = np.zeros_like(t)
for k in range(1, 11):
    n = 2 * k - 1  # uniquement les harmoniques impaires
    tri += ((-1)**((k+1)) / n**2) * np.sin(2 * np.pi * n * f * t)
tri *= (8 / (np.pi**2))  # normalisation serie de Fourier

# 3. Bruit blanc gaussien
taille = fe  # 1 seconde
bruit = np.random.randn(taille)
bruit /= np.max(np.abs(bruit))  # normalisation pour eviter la saturation
```

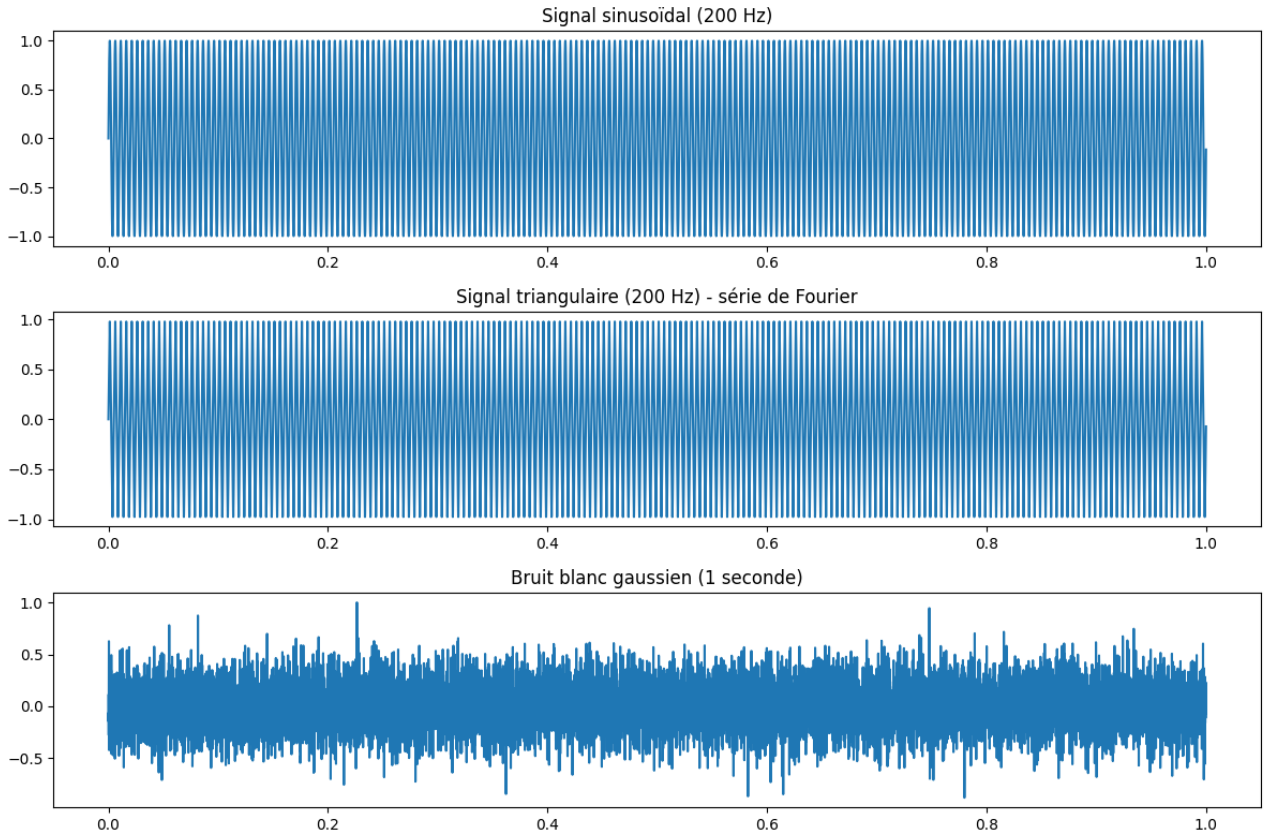


Figure 10: Signaux générés

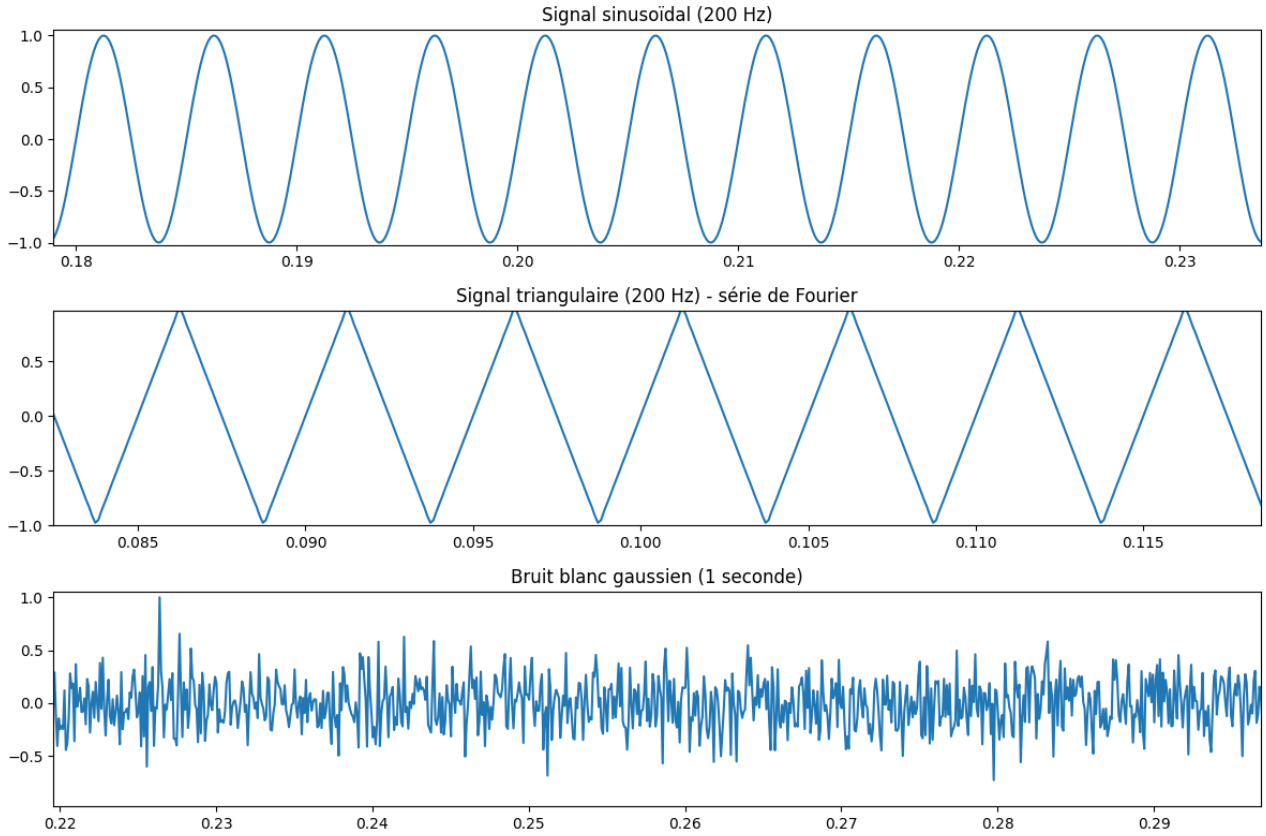


Figure 11: Signaux générés (zoom)

On calcule l'autocorrélation pour chaque signal sur une partie de durée de 30ms. C'est une durée typique en traitement du signal pour analyser la parole (équilibre entre résolution temporelle et fréquence). Elle est suffisante pour voir une structure (formants, périodicité...) sans mélanger des parties trop différentes du signal.

Un signal est dit **stationnaire** si ses propriétés statistiques, telles que la moyenne, la variance et l'autocorrélation, ne varient pas dans le temps.

L'analyse des autocorrélations montre que :

- Le signal vocal « aa » est **non stationnaire** car ses caractéristiques changent entre différentes tranches temporelles (voix humaine évolue dans le temps).
- Le signal de bruit blanc gaussien est **stationnaire**, ses propriétés statistiques étant constantes dans le temps (Pic au centre, décroissance rapide).
- Les signaux sinusoïdal et triangulaire, synthétisés sur quelques périodes, sont également **stationnaires** car ils sont périodiques et leurs statistiques ne varient pas dans le temps.

2.4 Classification de signaux de parole voisés ou non voisés

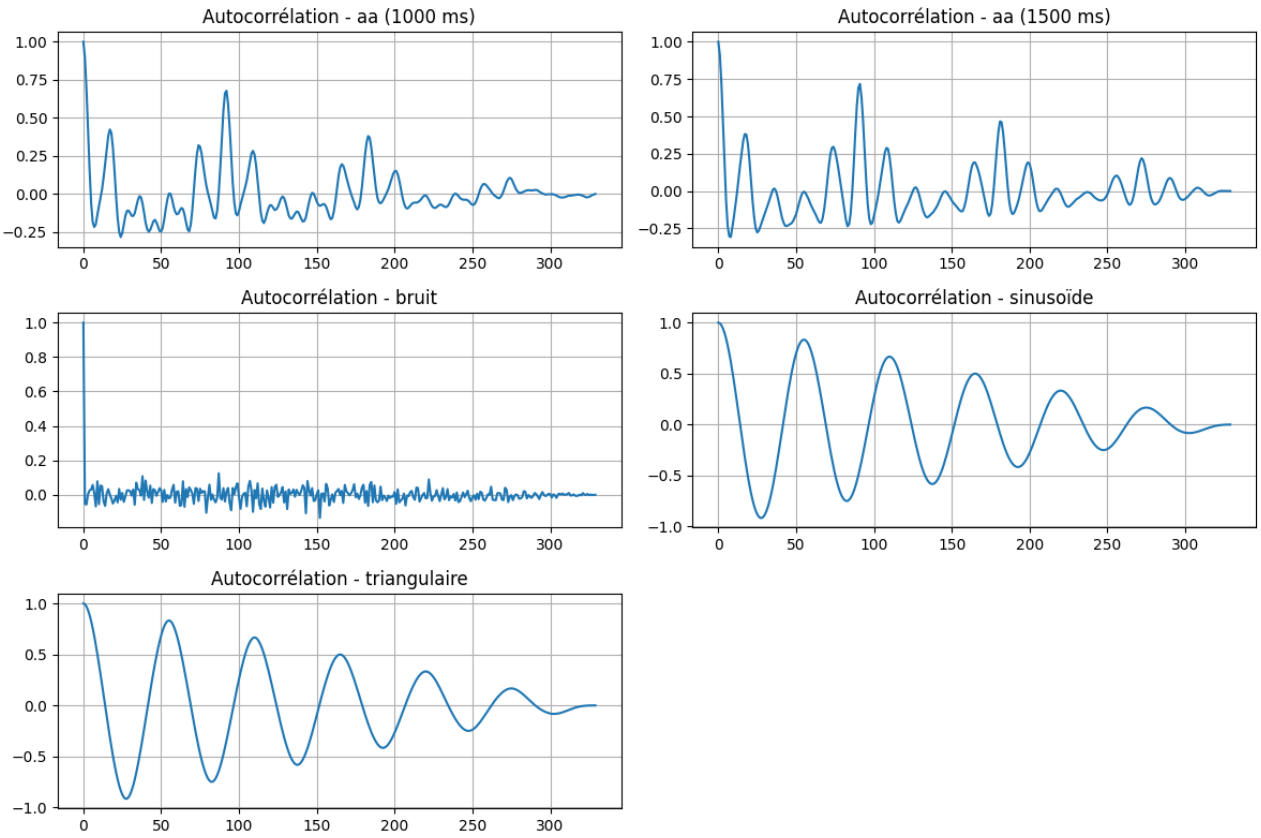


Figure 12: autocorrélation des signaux sur 30ms

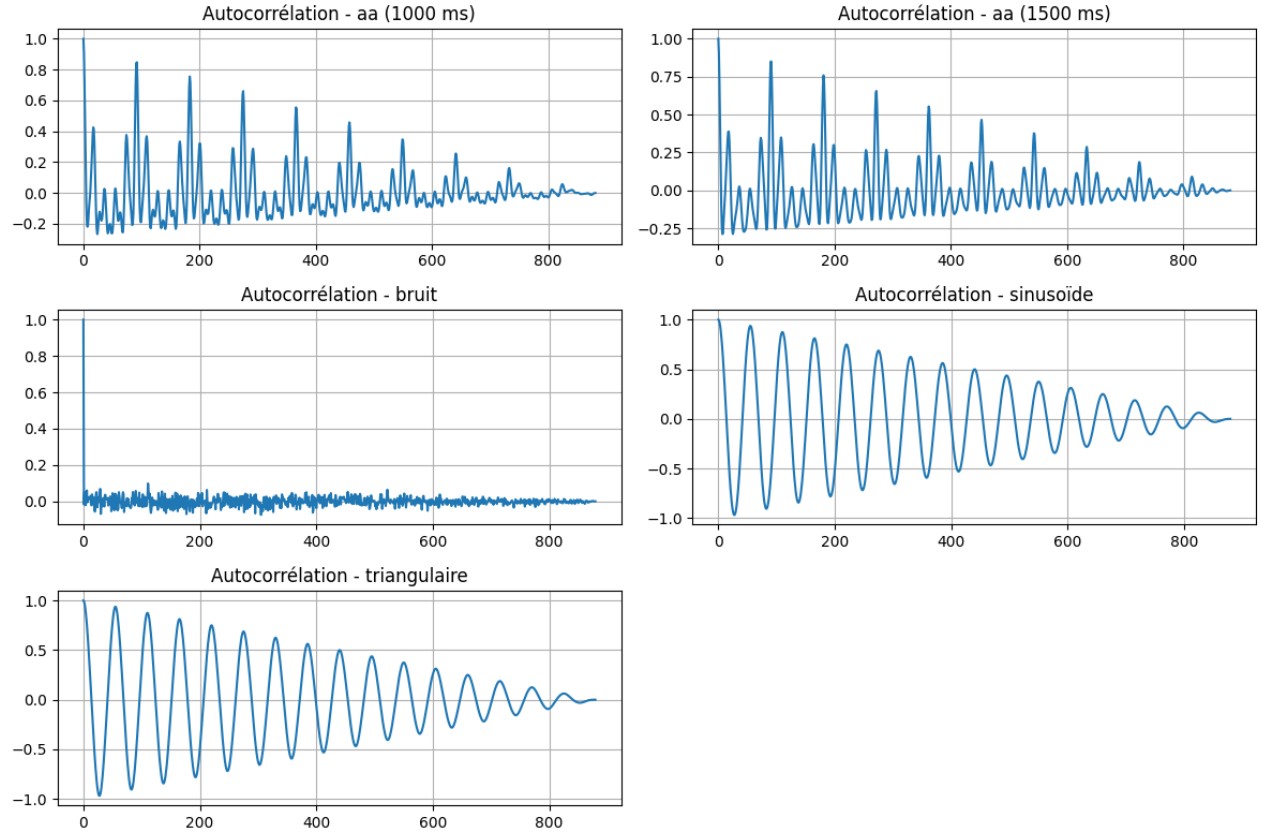


Figure 13: autocorrélation des signaux sur 80ms