

**Année universitaire  
2025 - 2026**

---

**Signaux analogiques et  
numériques — 3ETI**  
**Séance pratique 2**

---

**Analyse fréquentielle**



**Serge Mazauric  
Eric Van Reeth**

## Introduction

L'objectif principal de cette séance est de vous familiariser avec les représentations fréquentielles des signaux. Dans une première partie, vous aurez à effectuer l'analyse spectrale de signaux d'origines diverses afin d'étudier un certain nombre de leurs caractéristiques. Par la suite, vous réaliserez un projet de reconnaissance automatique (ou classification) de voyelles basé sur l'analyse spectrale.

## Évaluation

**La partie 2 de ce projet sera évaluée lors d'une soutenance orale d'une durée de 15 minutes** (moitié pour la présentation, moitié pour les questions). Le projet sera présenté en binôme sous forme de 4 diapositives qui devront contenir les informations suivantes (vous pourrez ajouter du contenu que vous jugerez nécessaire mais toujours en gardant 4 diapos) :

- **Diapo 1** : Affichage du signal à classifier (domaines temporel et fréquentiel)
- **Diapo 2** : Un spectre par voyelle de la base d'apprentissage (un spectre par son, donc 5 spectres au total)
- **Diapo 3** : Présentation de la stratégie de classification mise en œuvre :
  - Harmoniques extraites : afficher la moyenne des harmoniques pour chaque voyelle
  - Justifier le nombre d'harmoniques utilisé, la méthode de normalisation
- **Diapo 4** : Présentation et analyse des résultats
  - Numéro de signal attribué et voyelle identifiée
  - Distance/similarité avec les voyelles de la base d'apprentissage
  - Conclusion

**Le code développé devra également être rendu sur e-campus la veille de la séance d'évaluation orale.**

## Règles et bonne pratiques

Voici quelques règles et bonnes pratiques pour le déroulement du projet :

- Veillez à ce que vos diapos soient clairs et lisibles. La qualité de la présentation sera prise en compte dans la note.
- L'usage d'IA pour l'implémentation du code du projet et la rédaction des diapos est interdit.
- Le code développé doit être propre à chaque binôme. Tout plagiat détecté sera sanctionné.
- Pensez à vous référez à l'aide des fonctions Python en cas de doute sur leur utilisation.

## Mise en place

Ce TP sera codé en Python, en utilisant l'environnement virtuel `env_ms_i` mis en place en début d'année. Avant de démarrer, vous effectuerez la mise à jour de la librairie `msicpe` pour avoir accès à certaines fonctions utiles au TP.

Rappel : Mise à jour de la librairie `msicpe` :

- Ouvrez un terminal ou une invite de commandes et placez vous dans le répertoire où l'environnement `msicpe` a été installé :

```
cd chemin_dossier_contenant_env_ms_i
```

- Activez votre environnement virtuel :

Pour Windows : `. \env_msi\Scripts\activate`

Pour Linux/Mac : `source env_msi/bin/activate`

- Exécutez la commande suivante dans le même terminal :

`python -m pip install -U msicpe`

Les librairies Python à importer sont les suivantes :

```
import numpy as np
import msicpe.san as san
from plotly import express as px
import scipy.signal as signal
```

## 1 Partie 1 - Introduction à l'analyse spectrale

Cette partie est une introduction à l'analyse spectrale. L'objectif est d'afficher puis d'analyser les spectres de différents signaux afin de retrouver dans le domaine fréquentiel les caractéristiques de ces signaux.

Le chargement des différents signaux se fera de la manière suivante :

```
signal_dict = np.load('signal.npz') # signal_dict est un dictionnaire
# contenant les données contenues dans le fichier signal.npz
s = signal_dict['s'] # extraction du signal utile
t = signal_dict['t'] # extraction du vecteur temps associé au signal
```

Le calcul des transformées de Fourier se fera simplement se fera grâce à la fonction `san.trans_fourier()` dont l'aide peut être obtenue ainsi : `help(san.trans_fourier)`.

**Travail demandé :** Tracer le signal temporel ainsi que le spectre d'au moins 3 signaux du dossier `data_partie_1`. Analyser les tracés obtenus.

## 2 Partie 2 - Reconnaissance automatique de voyelle

La reconnaissance vocale est aujourd'hui un outil permettant de retranscrire de manière automatique un signal de parole en texte. Pour réaliser cela, il est nécessaire d'utiliser des outils de traitement du signal (aujourd'hui très souvent combinés à des outils d'intelligence artificielle) pour associer une lettre, une syllabe ou un mot au son étudié.

Votre objectif dans ce projet est de réaliser un algorithme de classification de voyelles capable de reconnaître automatiquement les sons /a/ /e/ /i/ /o/ et /u/. Pour cela, vous aurez à votre disposition plusieurs bases de données :

- **Une base d'entraînement** (ou base d'apprentissage) qui contient 5 signaux pour chaque son. Vous utiliserez cette base pour **définir les caractéristiques type de chaque son**, et votre **stratégie de classification**.
- **Une base de test** qui contient 2 signaux pour chaque son. Vous l'utiliserez pour vérifier que votre stratégie de classification est fonctionnelle.
- **Une base d'étude**, différente pour chaque binôme, qui contiendra un son inconnu que vous aurez à classifier correctement.

Les sections suivantes vous guideront dans la réalisation du projet.

## 2.1 Lecture des fichiers et affichages

1. Les signaux des différentes bases sont fournis au format .npz. Voici le code permettant de lire ce format et d'importer les signaux en tant que variable Python.

```
dat = np.load('son_a_charger.npz') # dat est un dictionnaire contenant
    ↵ les données contenues dans le fichier .npz
s = dat['s'] # extraction du signal utile
t = dat['t'] # extraction du vecteur temps associé au signal
```

À partir du code ci-dessus, ouvrez un signal au choix parmi ceux de la base d'apprentissage.

2. Affichez le signal chargé en fonction du temps.
3. Grâce à la fonction `san.trans_fourier()`, calculez la transformée de Fourier du signal et affichez le spectre en fonction des fréquences réelles (en Hz). Expliquer la présence des différents pics observés.
4. Tracer maintenant le logarithme (base 10) du spectre et commenter les différences avec le tracé précédent.

## 2.2 Apprentissage – Caractérisation des voyelles

Dans cette section, votre objectif est de définir un vecteur contenant un certain nombre de valeurs caractéristiques de chaque voyelle de la base d'apprentissage. Plus précisément, ce vecteur contiendra les amplitudes moyennes des spectres de la voyelle d'intérêt aux fréquences suivantes :

- à la fréquence fondamentale
- aux fréquences d'un certain nombre d'harmoniques (à vous de déterminer le nombre d'harmoniques à considérer)

Pour extraire les amplitudes du spectre à ces fréquences, vous avez à votre disposition les fonctions suivantes :

- `san.detect_fondamentale()`
- `san.detect_pics()`

1. Choisir une voyelle et un nombre d'harmoniques (appelé  $H$ , qui sera amené à être modifié par la suite). Pour chacun des 5 sons de la voyelle choisie, construire un vecteur de dimension  $(H+1)$  contenant les valeurs d'amplitude du logarithme (base 10) du spectre aux fréquences d'intérêt.
2. Calculez le vecteur contenant la valeur moyenne des spectres (toujours en logarithme) de la voyelle choisie pour chaque fréquence d'intérêt. Affichez le contenu des 5 vecteurs obtenus en fonction des indices des harmoniques considérées, ainsi que le vecteur moyen.
3. Répétez les opérations précédentes les autres voyelles et affinez le choix de  $H$ .

Aide : Pensez à normaliser les différents vecteurs obtenus pour faciliter la comparaison, et limiter les biais dus aux différences d'amplitude entre les différents signaux.

## 2.3 Test de la méthode

Il s'agit maintenant de voir si le vecteur moyen obtenu pour chaque voyelle vous permet de classifier correctement les voyelles de la base de test.

1. Pour chaque voyelle de la base de test, calculer le vecteur de caractéristiques spectrales de la même manière que dans la partie précédente.

2. Calculer la distance (à déterminer) entre le vecteur obtenu et les vecteurs moyens obtenus lors de la phase d'apprentissage. Vous considérerez que la classification est correcte si la distance minimale est obtenue avec le vecteur moyen correspondant à la bonne voyelle.
3. Vérifier que chaque voyelle de la base de test est correctement classifiée. Si ce n'est pas le cas, revoir votre stratégie de classification (faire varier  $H$ , la méthode de normalisation et/ou la distance choisie).

## 2.4 Application au cas d'étude

Dans cette dernière partie, vous considérerez que votre méthode de classification est fonctionnelle. Appliquez-la maintenant au son propre à votre binôme qui vous a été attribué par l'intervenant en séance, et donner le son auquel vous l'associez.