

# Optimisation du traitement du signal sonore sous forme numérique.

Aujourd'hui les analyses et interprétations des signaux cadencent la vie politique, économique et scientifique des sociétés modernes. Toutes les données qui transitent et qui sont traduites de la forme analogique à numérique nécessitent d'être traitées, et beaucoup de modes opératoires furent découverts afin d'accélérer le traitement du signal.

Ce TIPE repose essentiellement sur l'utilisation de la transformée de Fourier, et le passage de la dimension temporelle en dimension fréquentielle. Je m'intéresse à un moyen de convertir un signal sonore en un identifiant unique de celui-ci grâce à une fonction de hachage.

## Positionnement thématique (ÉTAPE 1) :

- *INFORMATIQUE (Informatique Théorique)*
- *MATHEMATIQUES (Analyse)*
- *PHYSIQUE (Physique Ondulatoire)*

## Mots-clés (ÉTAPE 1) :

Mots-clés (en français)    Mots-clés (en anglais)

*Transformé de Fourier*    *Fourier Transform*

*Algorithme*    *Algorithm*

*fréquences*    *frequencies*

*spectre*    *spectral*

*signal*    *signal*

## Bibliographie commentée

Durant les deux dernières décennies certaines applications telles que Shazam ont pour but de donner le titre d'une musique à partir d'un échantillon sonore. Le logiciel développé par Shazam Entertainment Limited fut construit de façon à optimiser le traitement du signal audio et minimiser les risques d'erreur dans le résultat retourné.

En premier lieu, Shazam fait appel à la transformée de Fourier permettant de **convertir** un signal numérique de la dimension temporelle vers la dimension fréquentielle [1]. Il est assez facile de coder un algorithme naïf capable de convertir un signal numérique en spectre de

Fourier, mais un tel algorithme devient inefficace lorsqu'il s'agit d'analyser un échantillon entier [2]. Pour répondre à cet enjeu, des algorithmes, comme *Cooley–Tukey FFT algorithm*, furent mis au point [3]. Cette approche est formalisée dans un article de James W. Cooley et John W. Tukey intitulé *An Algorithm for the Machine Calculation of Complex Fourier Series* [4]. Lors de l'acquisition d'un signal, un phénomène de "bruit" peut apparaître, on utilise alors une fenêtre afin de lisser les données [5]. Grâce à une méthode de hachage publiée dans un article sur le blog de Michel Strauss sur le fonctionnement de Shazam, il est possible de créer une corrélation entre un son enregistré et une empreinte d'un signal numérique afin de pouvoir attribuer à chacun des signaux un identifiant unique, un *fingerprint* [6].

## Problématique retenue

Comment identifier un morceau de musique de manière unique et efficace à partir d'un échantillon sonore numérisé ?

## Objectifs du TIPE du candidat

1. Optimisation : Calcul du spectre de Fourier à partir de l'algorithme de Cooley-Tukey.
2. Création de la première version du *Fingerprint* : création d'une empreinte d'un échantillon sonore grâce aux pics d'intensités des fréquences et de la fonctions de hachage.
3. Création d'une deuxième version du *Fingerprint* : acquisition de multiples spectres de Fourier au cours du temps puis compilation de ces spectres pour créer une empreinte plus précise.
4. Conclusion sur la manière dont une machine peut efficacement reconnaître un échantillon sonore de façon unique.

## Références bibliographiques(ÉTAPE 1)

- [1] BDELDJALIL OUAHABI : Analyseurs de signaux à base de transformation de Fourier : <https://www.techniques-ingenieur.fr/doi/10.51257/a/v2/r1156>
- [2] DETREZ ERIC : TP FFT 2021 : [https://info.faidherbe.org/ITC2/TP\\_FFT\\_2021.pdf](https://info.faidherbe.org/ITC2/TP_FFT_2021.pdf)
- [3] WIKIPÉDIA : Cooley-tukey fft algorithm. : [https://en.wikipedia.org/w/index.php?title=Cooley%E2%80%93Tukey\\_FFT\\_algorithm&oldid=1237705262](https://en.wikipedia.org/w/index.php?title=Cooley%E2%80%93Tukey_FFT_algorithm&oldid=1237705262)
- [4] JAMES, W. COOLEY AND JOHN, W. TUKEY : An Algorithm for the Machine Calculation of Complex Fourier Series : *Mathematics of Computation* publié par American Mathematical Society
- [5] MICHAEL STRAUSS : How Shazam Works - An explanation in Python : <https://michaelstrauss.dev/shazam-in-python>

**[6]** WIKIPÉDIA : Fenêtrage : *<https://fr.wikipedia.org/w/index.php?title=Fen%C3%AAtirage&oldid=216528895>*