

# TD 3

---

## Analyse spectrale du chant d'une baleine bleue & d'une cigale - Détermination d'un numéro masqué & premier filtrage

### Objectifs

- Analyser le chant d'une baleine bleue dans le domaine spectral
- Analyser le chant d'une cigale dans le domaine spectral
- Déterminer un numéro de téléphone par analyse spectrale
- Supprimer par filtrage un harmonique parasite dans un morceau de musique

Téléchargez le fichier TD3.zip sur le site web du cours.

Vous sauvegarderez l'ensemble de vos commandes dans des programmes `td3_1.mlx`, et `td3_2.mlx` qui débiteront ainsi :

```
% Fichier td3_X.mlx
% Votre prénom nom
% Date du jour

clear
clc
close all

...
```

### 3.1 Analyse spectrale du chant d'une baleine bleue

Chargez dans MATLAB l'enregistrement du chant d'une baleine bleue en saisissant la commande :

```
ChantBaleine=fullfile(matlabroot,'examples','matlab','bluewhale.au');
```

```
[x,fs] = audioread(ChantBaleine);
```

Relevez la fréquence d'échantillonnage.

Utilisez la fonction `sound` pour écouter le chant de la baleine :

```
sound(x,fs);
```

Lancez l'analyseur de signaux `SignalAnalyzer`.

Visualisez les 20 secondes enregistrées et affichez le spectre du chant de baleine.

On rappelle que la bande de fréquences audibles par l’oreille humaine s’étale de 20 Hz à 20 kHz. Lorsque la fréquence est faible, le son est grave (de 20 à 200 Hz). On parle de son médium pour une fréquence comprise entre 200 et 1000 Hz et de son aigu lorsque la fréquence est comprise entre 1000 et 15000 Hz.

Relevez la plage de fréquences et en déduire le type de son produit par le chant d’une baleine.

Il est possible de restreindre l’analyse d’un signal sur une durée d’observation donnée en cliquant sur l’icône **Panner** dans le menu **Display**.

Réalisez l’analyse du signal sur la fenêtre de 0 à 5s permettant de se focaliser sur le premier morceau du chant. Observez les harmoniques principaux de ce premier son émis par la baleine. Faites glisser la fenêtre du panner pour observer les caractéristiques des 3 autres sons émis par la baleine.

L’outil **Panner** sera exploité avantageusement dans l’exercice 3.3.

## 3.2 Analyse spectrale du chant d’une cigale

Chargez dans MATLAB l’enregistrement du chant d’une cigale en saisissant la commande :

```
[y,fe] = audioread('crickets.wav');
```

Relevez la fréquence d’échantillonnage.

Utilisez la fonction **sound** pour écouter le chant de la cigale :

```
sound(x,fe);
```

Lancez l’analyseur de signaux **SignalAnalyzer**.

Visualisez les 2 secondes enregistrées et affichez le spectre du chant de la cigale.

Relevez la plage de fréquences et en déduire le type de son produit par le chant d’une cigale.

Réalisez l’analyse du signal sur la fenêtre de 0 à 0.5s à l’aide du **Panner**. Faites glisser la fenêtre du panner pour observer les caractéristiques des 3 autres sons émis par la cigale et relevez le caractère périodique du son.

## 3.3 Détermination d’un numéro de téléphone par analyse spectrale

Un code DTMF (dual-tone multi-frequency) ou FV (Fréquences Vocales) est une combinaison de fréquences utilisée pour la téléphonie filaire traditionnelle utilisée sur le Réseau téléphonique commuté (RTC) (sauf voix sur IP). Ces codes sont émis lors de l’appui sur une touche du clavier téléphonique, et sont utilisés pour la composition des numéros de téléphones et de fax.

Techniquement, chaque touche d’un téléphone correspond à un couple de deux fréquences audibles qui sont émises simultanément. On s’intéresse ici au code DTMF utilisé aux Etats-Unis qui utilise sept fréquences bien distinctes qui permettent de coder les douze touches du clavier téléphonique représenté sur la figure 3.1. Ces sept fréquences sont visibles sur le bord gauche et sur le bas de la figure 3.1.

Elles sont également précisées dans le tableau ci-dessous.

697 Hz	770 Hz	852 Hz	941 Hz	1209 Hz	1336 Hz	1477 Hz
--------	--------	--------	--------	---------	---------	---------

Le son généré par l’appui d’une touche résulte de la somme de deux sinusoïdes aux fréquences associées. Ainsi l’appui de la touche 1 produira le signal suivant :

$$y_1 = \frac{1}{2}(\sin(2\pi \times 697 \times t) + \sin(2\pi \times 1209 \times t));$$

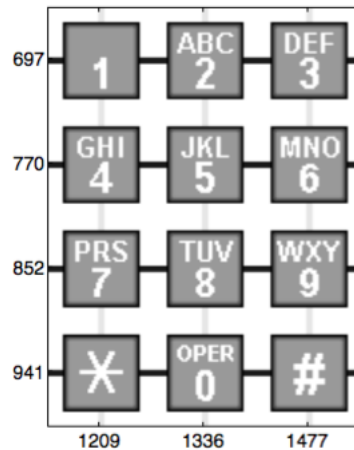


Figure 3.1 – Diapason

La figure 3.2 représente le signal et son spectre d’amplitude lorsqu’on appuie sur la touche ”1” du clavier. Les deux fréquences peuvent être clairement identifiées à partir du spectre.

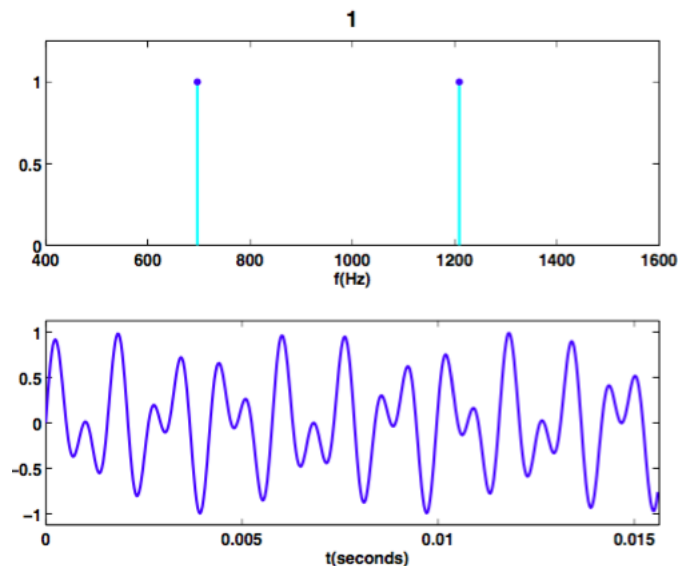


Figure 3.2 – Evolution temporelle et spectre du signal résultant de l’appui de la touche 1 du clavier

Un fichier `numero_tel.mat` contenant un numéro de téléphone américain est disponible sur le site web du cours.

Dans ce fichier se trouvent les vecteurs colonne `y` et `t` contenant les échantillons du numéro composé et les instants d’enregistrement ainsi que la fréquence d’échantillonnage `fe`.

Créez un nouveau fichier `td3_1.m`.

Chargez le numéro de téléphone dans MATLAB en saisissant la commande :

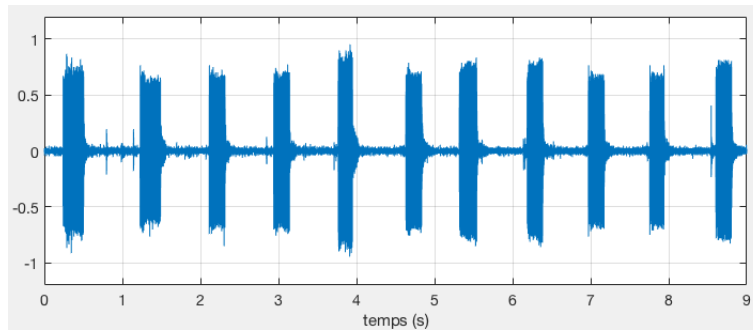


Figure 3.3 – Evolution temporelle du numéro de téléphone composé

```
load numero_tel;
```

Tracez l'évolution temporelle du numéro de téléphone sur les 9 s d'enregistrement disponibles à l'aide de la commande suivante :

```
plot(t,y),shg
```

Vérifiez que vous obtenez un tracé similaire à celui représenté sur la figure 3.3.

En zoomant, observez la forme de chaque son. Que constatez-vous ?

De combien de chiffres est composé le numéro de téléphone ?

Est-il possible de déterminer le numéro de téléphone d'après l'évolution temporelle du signal composé ?

Ecoutez le numéro composé en saisissant la commande :

```
sound(y,fe);
```

On peut tracer la représentation temps-fréquence du numéro composé en saisissant la commande :

```
spectrogram(y,kaiser(128,18),120,128,fe,'yaxis');
```

La figure permet de visualiser le contenu fréquentiel du signal au cours du temps mais la précision sur l'axe des fréquences n'est pas suffisante pour identifier les deux fréquences.

Utilisez l'outil graphique d'analyse de signaux **SignalAnalyzer** pour déterminer le numéro de téléphone (le panner sera en particulier très utile).

A l'aide d'Internet, déterminez le nom de la société à qui appartient le numéro. Trouvez son adresse aux Etats-Unis et utilisez Google Maps pour visualiser la porte d'entrée principale de l'entreprise américaine. Un indice est fourni sur la figure 3.4.

En pratique, la détection de tonalités DTMF peut être effectuée en utilisant différents algorithmes. Pour en savoir plus, parcourez l'article publié en 1996 qui présente l'analyse comparative des performances de trois algorithmes :

[http://ptolemy.eecs.berkeley.edu/papers/96/dtmf\\_ict/ict96.pdf](http://ptolemy.eecs.berkeley.edu/papers/96/dtmf_ict/ict96.pdf)

Relevez le nom des trois algorithmes comparés.

Trouvez en surfant sur Internet le code DTMF ou combinaison de fréquences utilisée pour la téléphonie filaire traditionnelle utilisé en France. Générer le numéro de l'accueil de Polytech Nancy. Tracez son évolution temporelle et son spectrogramme à l'aide de l'outil graphique



Figure 3.4 – Indice photographique pour trouver à qui appartient le numéro à identifier

d’analyse de signaux de MATLAB.

Faites valider vos résultats par l’enseignant.

### 3.4 Traitement d’un extrait musical parasité

Lors de l’enregistrement d’un signal musical, un signal parasite haute fréquence est venu perturber l’enregistrement. L’objectif est donc d’identifier par analyse spectrale du signal enregistré la perturbation puis de la supprimer au travers d’un filtrage afin de retrouver le signal d’origine.

Créez un nouveau fichier `td3_2.mlx`.

Chargez le fichier `extrait_musique.wav`, à l’aide de la fonction Matlab `audioread` :

```
[y,fe] = audioread('extrait_musique.wav');
```

Quelle est la fréquence d’échantillonnage utilisée pour enregistrer le signal ?

Utilisez la fonction `sound` pour écouter l’extrait musical bruité (attention au volume car cela peut s’avérer désagréable à entendre) :

```
sound(y,fe);
```

Observez le spectre d’amplitude du signal. Vous pouvez utiliser l’outil `SignalAnalyzer` pour visualiser le spectre. Déterminez la fréquence de l’harmonique parasite.

On propose de supprimer la perturbation par application d’un filtrage passe-bas de Butterworth en étudiant l’influence du choix de l’ordre. Matlab comporte une fonction `butter` (`help butter`) qui fournit les paramètres de la fonction de transfert du filtre à partir de la spécification de l’ordre, de la fréquence de coupure du filtre normalisée et du type de filtre (low, high,...). On pourra s’inspirer de la solution suivante en spécifiant la fréquence coupure souhaitée :

```
fc=????; % fréquence de coupure du filtre
fcn=fc/(fe/2); % fréquence de coupure du filtre normalisée par rapport à fe/2
n=1; % ordre du filtre
[b,a] = butter(n,fcn,'low'); % Calcul du numérateur et du dénominateur du filtre
```

```
freqz(b,a,512,fe) % tracé de la réponse fréquentielle du filtre  
yf=filter(b,a,y); % produit de convolution du filtre et du signal bruité  
sound(yf,fe); % pour écouter l'extrait filtré
```

Renouvelez la procédure en augmentant l'ordre du filtre jusqu'à déterminer l'ordre qui permet de supprimer totalement l'harmonique perturbateur et ainsi écouter le signal musical d'origine.

MATLAB possède la fonction `buttord` qui fournit l'ordre du filtre de Butterworth qui permet de respecter l'atténuation en dB souhaitée. Utilisez cette fonction pour déterminer l'ordre minimal qui permet de ne plus entendre la perturbation dans l'extrait musical.

Faites valider votre filtrage par l'enseignant.