

## TP Analyse et synthèse de sons de cloche

**Roland Badeau**



Les fichiers nécessaires à ce TP sont disponibles sur le Moodle ATIAM. Vous pourrez charger un fichier son sous Matlab en tapant la commande `[x,Fs] = wavread('cloche.wav')`. Pour l'écouter, vous pourrez taper `soundsc(x,Fs)`. En Python, vous pourrez utiliser le notebook qui vous est fourni `template-TP-HR.ipynb`.

## 1 Introduction

Les cloches sont parmi les instruments de musique les plus anciens et le son qu'elle produisent est souvent évocateur parce qu'il a bercé le quotidien des générations depuis environ 3000 ans, accompagnant petits et grands événements. Cette évocation tient en partie à la structure du spectre sonore : les modes propres de vibration sont en général accordés par les facteurs de cloches (dans le cas des bons instruments) pour que leurs fréquences suivent une série particulière, qui comporte notamment la tierce mineure (Mi bémol si la cloche est en Do). Cette série n'est pas harmonique mais les rapports entre fréquences propres sont tels qu'on perçoit une hauteur bien définie (celle à laquelle on chanterait la note entendue). Notamment la présence de la série 2-3-4, forte au début du son, vient renforcer la sensation de hauteur au voisinage du fondamental. Cette sensation est liée à un effet psychoacoustique (traitement du signal reçu par le cerveau).

Soit  $f_p$  la fréquence correspondant à la hauteur perçue. L'analyse de la série des fréquences propres se traduit par la donnée d'un tableau d'environ 15 facteurs  $\alpha_n = f_n/f_p$ . Un ordre de grandeur est donné ci-après : 0.5 (bourdon), 1 (fondamental), 1.2 (tierce mineure), 1.5 (quinte), 2 (nominal), 2.5, 2.6, 2.7, 3, 3.3, 3.7, 4.2 (faux double octave), 4.5, 5, 5.9. Le timbre du son correspondant dépend de l'amplitude et de la décroissance de chacun de ces partiels.

Ce TP vise à mettre en oeuvre une méthode d'estimation spectrale à haute résolution pour effectuer l'analyse / synthèse de sons de cloche. Comme on peut le constater sur la figure 1, ce type de sons présente une forte décroissance temporelle.

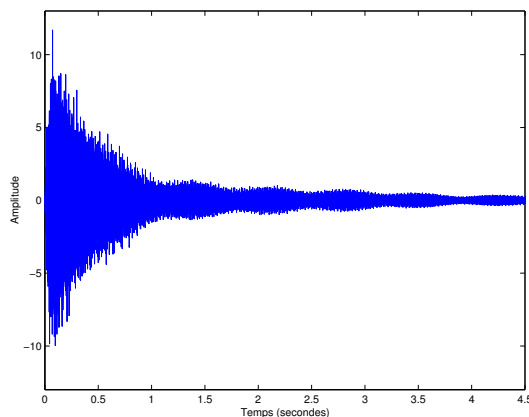


FIGURE 1 – Son de cloche

Pour tenir compte de cette atténuation, on utilise le modèle *Exponential Sinusoidal Model* (ESM) :

$$s[t] = \sum_{k=0}^{K-1} a_k e^{\delta_k t} e^{i(2\pi f_k t + \phi_k)},$$

où à chaque fréquence  $f_k \in ]-\frac{1}{2}, \frac{1}{2}]$  est associée une amplitude réelle  $a_k > 0$ , une phase  $\phi_k \in ]-\pi, \pi]$ , et un facteur d'amortissement  $\delta_k \in \mathbb{R}$ . En définissant les amplitudes complexes  $\alpha_k = a_k e^{i\phi_k}$  et les pôles complexes  $z_k = e^{\delta_k + i2\pi f_k}$ , ce modèle se réécrit sous la forme

$$s[t] = \sum_{k=0}^{K-1} \alpha_k z_k^t.$$

Le cas  $\delta_k < 0$  permet donc de modéliser des sinusoides à décroissance exponentielle, ce qui correspond aux solutions des équations de propagation physique. Les paramètres du modèle sont alors  $\{\delta_k, f_k, a_k, \phi_k\}_{k \in \{0 \dots K-1\}}$ . Pour les estimer, nous utiliserons la méthode ESPRIT présentée dans le cours. Dans un premier temps, nous l'appliquerons à un signal synthétique pour mettre en évidence la supériorité des méthodes à haute résolution sur l'analyse de Fourier en terme de résolution spectrale. Dans un deuxième temps, cette méthode sera appliquée à des sons de cloche.

## 2 Rappels de Matlab/Python

En Matlab :

- $A'$  : conjugué hermitien de la matrice  $A$  (inclut la conjugaison complexe);
- $A.'$  : transposé de la matrice  $A$  (sans conjugaison complexe);
- $A(l1:l2, c1:c2)$  : matrice extraite de  $A$  entre les lignes  $l_1$  et  $l_2$  (inclue) et les colonnes  $c_1$  et  $c_2$  (inclue).

En Python :

- $A.conj()$  : conjugué hermitien de la matrice  $A$  (inclut la conjugaison complexe);
- $A.T$  : transposé de la matrice  $A$  (sans conjugaison complexe);
- $A[l1:l2, c1:c2]$  : matrice extraite de  $A$  entre les lignes  $l_1$  et  $l_2$  (exclue) et les colonnes  $c_1$  et  $c_2$  (exclue).

## 3 Signal synthétique

Nous nous intéressons ici à un signal synthétique de longueur  $N$ , constitué d'une somme de deux exponentielles complexes, dont les fréquences sont séparées d'un intervalle  $\Delta f = \frac{1}{N}$  (ce qui correspond à la limite de la résolution de l'analyse de Fourier). Les phases seront tirées aléatoirement. On ne bruitera pas ce signal, de sorte que le signal observé  $x[t]$  sera égal à  $s[t]$ . On choisira par exemple les paramètres suivants :  $N = 63$ ,  $f_0 = \frac{1}{4}$ ,  $f_1 = f_0 + \frac{1}{N}$ ,  $a_0 = 1$ ,  $a_1 = 10$ ,  $\delta_0 = 0$ ,  $\delta_1 = -0.05$ . Pour le synthétiser, on pourra utiliser la fonction fournie

```
x = Synthesis(N,delta,f,a,phi);
```

dont les arguments sont  $N$ , le vecteur `delta` des atténuations  $\delta_k$ , le vecteur `f` des fréquences  $f_k$ , le vecteur `a` des amplitudes  $a_k$ , et le vecteur `phi` des phases  $\phi_k$ .

### 3.1 Analyse spectrale par transformation de Fourier

Observer le périodogramme de ce signal. On examinera brièvement la séparabilité des deux raies spectrales, sans zero-padding ( $N_{\text{fft}} = N$ ) et avec zero-padding ( $N_{\text{fft}} = 1024 > N$ ).



### 3.2 Méthodes à haute résolution

On se propose d'écrire des fonctions

$$\text{MUSIC}(\mathbf{x}, n, K) \text{ et } [\text{delta}, f] = \text{ESPRIT}(\mathbf{x}, n, K)$$

qui analysent le signal  $x$  de longueur  $N$  en utilisant les méthodes MUSIC et ESPRIT, avec un espace signal de dimension  $K$  et un espace bruit de dimension  $n - K$ , et des vecteurs de données de longueur  $n$  comprise entre  $K + 1$  et  $N - K + 1$ . Pour traiter nos signaux synthétiques, on pourra choisir  $n = 32$  et  $K = 2$ . Les deux méthodes partagent les étapes suivantes :

#### 1. Calcul de la matrice de corrélation

La matrice de corrélation du signal observé est définie par la relation

$$\widehat{\mathbf{R}}_{xx} = \frac{1}{l} \mathbf{X} \mathbf{X}^H$$

où  $\mathbf{X}$  est une matrice de Hankel de dimension  $n \times l$  contenant les  $N = n + l - 1$  échantillons du signal :

$$\mathbf{X} = \begin{bmatrix} x[0] & x[1] & \dots & x[l-1] \\ x[1] & x[2] & \dots & x[l] \\ \vdots & \vdots & \ddots & \vdots \\ x[n-1] & x[n] & \dots & x[N-1] \end{bmatrix}$$

La matrice  $\mathbf{X}$  pourra être construite avec la fonction `hankel`.

#### 2. Estimation de l'espace signal

On pourra diagonaliser la matrice  $\widehat{\mathbf{R}}_{xx}$  à l'aide de la commande `[U1, Lambda, U2] = svd(Rxx)`. La matrice  $\widehat{\mathbf{R}}_{xx}$  étant symétrique positive, les vecteurs colonnes des matrices  $\mathbf{U}_1$  et  $\mathbf{U}_2$  (de dimension  $n \times n$ ) sont vecteurs propres de  $\widehat{\mathbf{R}}_{xx}$ , associés aux  $n$  valeurs propres rangées dans la matrice diagonale  $\mathbf{\Lambda}$  par ordre décroissant (on a alors  $\widehat{\mathbf{R}}_{xx} = \mathbf{U}_1 \mathbf{\Lambda} \mathbf{U}_1^H = \mathbf{U}_2 \mathbf{\Lambda} \mathbf{U}_2^H$ ). On pourra ainsi extraire de  $\mathbf{U}_1$  (ou de  $\mathbf{U}_2$ ) une base de l'espace signal  $\mathbf{W}$  (de dimension  $n \times K$ ).

#### 3.2.1 Algorithme ESPRIT

L'algorithme ESPRIT vise dans un premier temps à estimer les fréquences et les facteurs d'amortissement :

#### 3. Estimation des fréquences et des facteurs d'amortissement

Pour estimer les fréquences, on pourra procéder de la façon suivante :

- extraire de  $\mathbf{W}$  les matrices  $\mathbf{W}_\downarrow$  (obtenue en supprimant la dernière ligne de  $\mathbf{W}$ ) et  $\mathbf{W}_\uparrow$  (obtenue en supprimant la première ligne de  $\mathbf{W}$ );
- calculer  $\mathbf{\Phi} = \left( (\mathbf{W}_\downarrow^H \mathbf{W}_\downarrow)^{-1} \mathbf{W}_\downarrow^H \right) \mathbf{W}_\uparrow = \mathbf{W}_\downarrow^\dagger \mathbf{W}_\uparrow$ , où le symbole  $\dagger$  est l'opérateur de pseudo inverse (fonction `pinv` de Matlab ou fonction `numpy.linalg.pinv` de Python).
- calculer les valeurs propres de  $\mathbf{\Phi}$  à l'aide de la fonction `eig` de Matlab ou de la fonction `numpy.linalg.eig` de Python (on rappelle que les valeurs propres de  $\mathbf{\Phi}$  sont les pôles  $z_k = e^{\delta_k + i2\pi f_k}$ ). En déduire  $\delta_k = \ln(|z_k|)$  et  $f_k = \frac{1}{2\pi} \text{angle}(z_k)$ .

#### 4. Estimation des amplitudes et des phases

Il s'agit à présent d'écrire une fonction



$$[a, \phi] = \text{LeastSquares}(x, \delta, f)$$

qui estime les amplitudes  $a_k$  et les phases  $\phi_k$  par la méthode des moindres carrés, connaissant le signal  $x$ , les atténuations  $\delta_k$  et les fréquences  $f_k$ . Les amplitudes complexes sont ainsi déterminées par la relation

$$\alpha = \left( (V^{NH} V^N)^{-1} V^{NH} \right) x = V^{N\dagger} x \quad (1)$$

où  $x$  est le vecteur  $[x[0], \dots, x[N-1]]^T$  et  $V^N$  est la matrice de Vandermonde de dimension  $N \times K$ , dont les coefficients vérifient la relation  $V_{(t,k)}^N = z_k^t$  pour tous  $(t, k) \in \{0 \dots N-1\} \times \{0 \dots K-1\}$ . Pour calculer la matrice  $V^N$ , il est possible d'éviter d'utiliser une boucle `for` en remarquant que  $\ln(V_{(t,k)}^N) = t(\delta_k + i2\pi f_k)$ . Ainsi, la matrice contenant les coefficients  $\ln(V_{(t,k)}^N)$  s'exprime comme le produit d'un vecteur colonne par un vecteur ligne. On en déduira  $a_k = |\alpha_k|$  et  $\phi_k = \text{angle}(\alpha_k)$  pour tout  $k \in \{0 \dots K-1\}$ .

### 5. Application aux signaux synthétiques

Appliquer les fonctions ESPRIT et LeastSquares au signal précédemment synthétisé. Commenter.

### 3.2.2 Méthode MUSIC

On rappelle que le pseudo-spectre MUSIC est défini par  $P(z) = \frac{1}{\|W_{\perp}^H v^n(z)\|^2}$ .

### 6. Affichage du pseudo-spectre MUSIC

Écrire une fonction MUSIC( $x, n, K$ ) qui affiche le logarithme du pseudo-spectre comme une fonction des deux variables  $f \in [0, 1]$  et  $\delta \in [-0.1, 0.1]$  (on pourra utiliser la fonction `surf` de Matlab ou la fonction `plot_surface` de Python Matplotlib). Appliquer la fonction MUSIC au signal précédemment synthétisé, et vérifier que le pseudo-spectre fait bien apparaître les deux pôles  $z_k = e^{\delta_k + i2\pi f_k}$ .

## 4 Signaux audio

Nous nous proposons maintenant d'appliquer les fonctions développées dans la partie précédente à des sons de cloche.

### 4.1 Analyse spectrale par transformation de Fourier

Examiner le périodogramme du signal `cloche.wav` et comparer la série de ses fréquences propres aux valeurs données en introduction.

### 4.2 Méthode à haute résolution

Il s'agit à présent d'appliquer l'algorithme ESPRIT à ce signal. On posera  $K = 54$ ,  $n = 512$  et  $l = 2n = 1024$  (d'où  $N = n + l - 1 = 1535$ ).

Afin de garantir que le modèle de signal soit vérifié sur la fenêtre d'analyse (amortissement exponentiel), on extraira un segment de longueur  $N$  dont le début sera postérieur au maximum de l'enveloppe de la forme d'onde. On pourra ainsi commencer au 10000<sup>ème</sup> échantillon.



Appliquer la fonction ESPRIT au signal extrait afin d'en estimer les fréquences propres et les atténuations correspondantes. En déduire les amplitudes et les phases à l'aide de la fonction LeastSquares. Enfin, écouter le signal resynthétisé à l'aide de la fonction Synthesis (sur une durée plus longue que le segment extrait, afin de bien mettre en évidence les résonnances du son), et commenter.



Contexte académique } sans modifications

***Par le téléchargement ou la consultation de ce document, l'utilisateur accepte la licence d'utilisation qui y est attachée, telle que détaillée dans les dispositions suivantes, et s'engage à la respecter intégralement.***

La licence confère à l'utilisateur un droit d'usage sur le document consulté ou téléchargé, totalement ou en partie, dans les conditions définies ci-après, et à l'exclusion de toute utilisation commerciale.

Le droit d'usage défini par la licence autorise un usage dans un cadre académique, par un utilisateur donnant des cours dans un établissement d'enseignement secondaire ou supérieur et à l'exclusion expresse des formations commerciales et notamment de formation continue. Ce droit comprend :

- le droit de reproduire tout ou partie du document sur support informatique ou papier,
- le droit de diffuser tout ou partie du document à destination des élèves ou étudiants.

Aucune modification du document dans son contenu, sa forme ou sa présentation n'est autorisée.

Les mentions relatives à la source du document et/ou à son auteur doivent être conservées dans leur intégralité.

Le droit d'usage défini par la licence est personnel et non exclusif. Tout autre usage que ceux prévus par la licence est soumis à autorisation préalable et expresse de l'auteur : [sitepedago@telecom-paristech.fr](mailto:sitepedago@telecom-paristech.fr)