



ENSIM
École d'ingénieurs
Le Mans Université



MiCROdB

~ Travaux pratiques - Imagerie acoustique ~

ENSIM - Acoustique-Instrumentation
option : Diagnostic et Contrôle Vibratoire

Simon Bouley

✉ simon.bouley@microdb.fr

Dernière mise à jour : 23 février 2024

Table des matières

Imagerie acoustique	2
1.1 Géométrie	3
1.2 Signaux	3
1.3 Fonction de transfert	4
1.4 Formation de voies	5
1.5 Fonction de réponse d'antenne	6
1.6 Discussion	7
1.7 Fonctions Matlab / Python	7

※ Imagerie acoustique

L'imagerie acoustique a pour objectif de localiser les sources de bruit générées par un objet rayonnant à partir de mesures de pression (à l'aide d'un réseau de microphones, ou antenne microphonique).

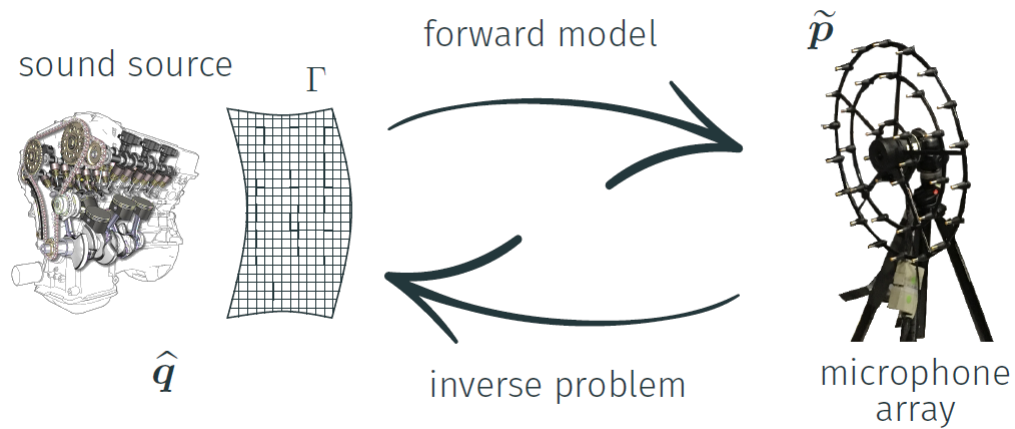


FIGURE 1 – Schéma général de l'imagerie acoustique : un objet rayonnant est défini par une grille de calcul, formée de sources potentielles ; un modèle définit la propagation acoustique ; une antenne microphonique mesure le champ de pression ; l'imagerie acoustique résout le problème inverse en déterminant les sources de bruit sachant les mesures et le modèle de propagation.



(a)



(b)

FIGURE 2 – (a) : Guitare étudiée, (b) : Antenne SoundCamera.

L'objectif de ces travaux pratiques est d'étudier la méthode d'imagerie acoustique la plus répandue : la formation de voies (en anglais *beamforming*). Cette méthode sera appliquée à des données réelles mesurées par une antenne plane (bidimensionnelle), placée devant une guitare à une distance égale à $d = 0.42$ m. Une note a été jouée lors de l'acquisition des signaux. Les signaux temporels vous seront fournis et vous devrez établir une cartographie sonore afin d'étudier le rayonnement de la guitare (Fig. (2)). Vous utiliserez le langage Matlab ou Python pour ce TP.

Remarque préliminaire : pour chaque tracé demandé, affichez les valeurs des axes, leur quantité physique et l'unité associée.

1.1 Géométrie

L'imagerie acoustique consiste à déterminer l'amplitude (parfois la phase) de sources potentielles positionnées sur un plan près de l'objet d'étude ou directement sur la peau de l'objet. Ces sources sont dites équivalentes car elles représentent les sources physiques à l'origine du rayonnement acoustique. Ces sources sont disposées sur une grille de calcul. Les microphones sont disposés en réseau sur un support nommé antenne microphonique, selon une géométrie connue.

La première étape de l'imagerie acoustique consiste à déterminer la position des microphones et des points de calcul.

Les positions de microphones sont définies dans le fichier *array.txt* tandis que celles des points de calcul sont définies dans le fichier *grid.txt*. L'antenne contient $N_{mic} = 81$ microphones et la grille $N_{node} = 1581$ points de calcul (grille de 51×31 nœuds). La distance entre l'antenne et la grille est égale à $d = 0.42$ m. Les deux ensembles de points sont dans le repère de l'antenne.

Tracez les positions des microphones et des points de calcul dans le repère de l'antenne.

1.2 Signaux

Les signaux acoustiques générés par la guitare ont été mesurés et stockés dans le fichier *data.mat*. Ce fichier contient plusieurs variables :

- *increment* : pas temporel Δt
- *number_of_channels* : nombre de microphones N_{mic}
- *number_of_values* : nombre d'échantillons temporels N
- *start_value* : valeur du premier échantillon temporel t_0
- *time* : échantillons temporels t
- *values* : matrice des signaux mesurés, de taille $[N \times N_{mic}]$

Importez les signaux temporels contenus dans le fichier des données mesurées. Tracez le signal temporel du 1^{er} microphone. Écoutez-le si besoin (fichier guitar_mic1.wav). Que pouvez-vous dire de ce signal ? Afin d'en connaître plus sur la composition fréquentielle du signal, quelle méthode mettriez-vous en place ? Est-ce que la méthode du périodogramme de Welch vous paraît adaptée ici ? Pourquoi ?

Nous souhaitons établir par la suite une carte d'imagerie acoustique qui évolue avec le temps. Nous avons besoin de découper les signaux acoustiques en différents blocs temporels successifs (pas de recouvrement) sur lesquels l'algorithme de formation de voies sera appliqué. La première étape est de définir la taille de ces blocs temporels. La résolution fréquentielle Δf souhaitée est de 10 Hz. *À combien d'échantillons par bloc correspond cette résolution ? Sans recouvrement, combien de blocs temporels complets comporte le signal ? Calculez le spectre fréquentiel correspondant au premier bloc du signal temporel du 1^{er} microphone. Normalisez-le par le nombre d'échantillons contenus dans un bloc temporel puis affichez sa valeur absolue en décibel (pression de référence $p_{ref} = 2e^{-5}$ Pa) sur une bande de fréquence exploitable que vous justifierez.*

Calculer le spectre de chaque microphone (non normalisé cette fois) et stockez-les dans une matrice de dimension $[N_{mic}, N_f]$ avec N_{mic} le nombre de microphones et N_f le nombre de raies fréquentielles.

1.3 Fonction de transfert

Pour appliquer la formation de voies aux données mesurées, le modèle de propagation entre l'antenne et le plan de calcul doit être connu. On considèrera ici que les sources se propagent en champ libre (sans obstacle) jusqu'à l'antenne. L'équation de propagation s'écrit :

$$G(\mathbf{r}, f) = \frac{e^{-jk\mathbf{r}}}{\mathbf{r}}, \quad \mathbf{r} = \|\mathbf{r}_m - \mathbf{r}_n\|_2, \quad (1)$$

avec j le nombre imaginaire, $k = 2\pi f / c_0$ le nombre d'onde total, $c_0 = 343 \text{ m.s}^{-1}$ la célérité du son dans l'air, f la fréquence d'étude, $\|\bullet\|_2$ la norme euclidienne, \mathbf{r} la distance entre la position d'un microphone \mathbf{r}_m et celle d'un nœud de la grille de calcul \mathbf{r}_n .

Calculez la matrice de fonction de transfert G entre tous les microphones et les points de calcul (taille $[N_{mic} \times N_{node}]$) à $f = 3900 \text{ Hz}$.

À partir de la matrice de mesure défini en section 1.2, créez un vecteur des pressions mesurées $\mathbf{p}(f)$ de dimensions $[N_{mic} \times 1]$ correspondant à la valeur du spectre à $f = 3900 \text{ Hz}$ pour chaque microphone.

1.4 Formation de voies

La formation de voies a pour objectif de déterminer la position de la source acoustique et son amplitude en résolvant un problème de minimisation. À une fréquence fixée f , pour un vecteur de pression mesurées $\mathbf{p}(f)$ et une matrice de fonction de transfert $\mathbf{G}(f)$, nous cherchons l'amplitude $\mathbf{b}(f)$ qui minimise l'erreur quadratique suivante :

$$\varepsilon(\mathbf{b}(f)) = \|\mathbf{p}(f) - \mathbf{b}(f)\mathbf{G}(f)\|_2. \quad (2)$$

La solution s'écrit (en omettant la dépendance à la fréquence) :

$$\varepsilon(\mathbf{b}) = (\mathbf{p} - \mathbf{b}\mathbf{G})^H(\mathbf{p} - \mathbf{b}\mathbf{G}) = \mathbf{p}^H\mathbf{p} - \mathbf{b}\mathbf{p}^H\mathbf{G} - \mathbf{b}^H\mathbf{G}^H\mathbf{p} + \mathbf{b}^H\mathbf{b}\mathbf{G}^H\mathbf{G}. \quad (3)$$

L'opérateur hermitien $(\bullet)^H$ définit la matrice transconjuguée (ou matrice adjointe) : $A^H = (\overline{A})^T$. Nous sommes libre de prendre \mathbf{b} et \mathbf{b}^H comme des variables indépendantes, ainsi :

$$\frac{\partial \varepsilon(\mathbf{b})}{\partial \mathbf{b}^H} = 0 \Leftrightarrow -\mathbf{G}^H\mathbf{p} + \mathbf{b}\mathbf{G}^H\mathbf{G} = 0, \quad (4)$$

$$\mathbf{b}(f) = \frac{\mathbf{G}^H(f)\mathbf{p}(f)}{\|\mathbf{G}\|_F^2}. \quad (5)$$

avec $\|\bullet\|_F$ la norme de Frobenius. Ce calcul doit être réalisé pour chaque point de la grille de calcul, $\mathbf{b}(f)$ est donc un vecteur de dimensions $[N_{node} \times 1]$ et

contient les amplitudes de sources sur la grille estimées par formation de voies à la fréquence f .

Calculez le vecteur de formation de voies $\mathbf{b}(f)$. Le vecteur $\mathbf{b}(f)$ fait intervenir la multiplication de matrices : utilisez `numpy.matmul`, `numpy.dot` ou `@` en Python et `` en Matlab.*

La grille étant de taille 51×31 noeuds, transformez ce vecteur $\mathbf{b}(f)$ de taille $[N_{node} \times 1]$ en matrice de taille $[51 \times 31]$.

Affichez :

- *Le signal temporel du microphone 1 avec les limites temporels du 1^{er} bloc en traits verticaux*
- *Le spectre correspondant, avec un trait pointillé vertical placé sur la fréquence 3900 Hz*
- *la carte de formation de voie à cette fréquence. Vous pouvez utiliser la fonction `plot_CBF` dans le script `TP_ENSIM_functions`.*

1.5 Fonction de réponse d'antenne

La qualité d'une antenne microphonique se définit en général par deux propriétés : la résolution et la dynamique. La résolution désigne la capacité de l'antenne à distinguer deux sources proches et est calculée comme la largeur du lobe principal à -3 dB du niveau maximal de la réponse d'antenne. La dynamique désigne la plage d'amplitude pour laquelle l'antenne affiche des sources réelles. Elle est définie par la différence d'amplitude entre le lobe principal et le premier lobe secondaire. La fonction de réponse de l'antenne (ou *point spread function*, *PSF*) pour le nœud d'indice i s'écrit :

$$PSF_i = \mathbf{G}^H(\mathbf{G}[:, i]), \quad (6)$$

et s'apparente à une carte de formation de voies pour une source monopolaire placée au nœud d'indice i .

Trouvez l'indice du point de la grille correspondant au maximum de la carte de formation de voies. Calculez la réponse d'antenne en ce point. Transformez le vecteur PSF_{imax} de taille $[N_{node} \times 1]$ en matrice de taille $[51 \times 31]$ et affichez en dB la réponse d'antenne en ce point. Vous pouvez utiliser la fonction `plot_PSF` dans le script

TP_ENSIM_functions. Calculez et affichez cette même fonction pour la fréquence 100 Hz.

1.6 Discussion

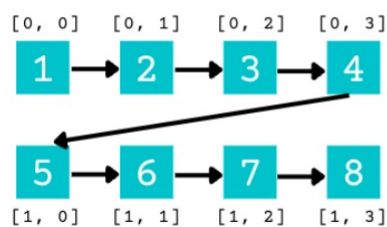
Appliquez le traitement sur différentes fréquences / différents blocs temporels. Commentez les cartes obtenues : analysez le comportement vibroacoustique de la guitare à l'origine des sources de bruit. À la lumière de ces résultats, quels sont les avantages et inconvénients de la méthode de formation de voies ?

1.7 Fonctions Matlab / Python

- importer un fichier texte : `importdata` / `numpy.loadtxt`
- importer un fichier .mat : `load` / `scipy.io.loadmat`
- calcul FFT : `fft(X)` / `numpy.fft.fft(X)`, X pouvant être un signal ou une collection de signaux
- conjuguée d'une matrice : `conj(A)` / `numpy.conj(A)`
- transposée d'une matrice : `transpose(A)` / `numpy.transpose(A)`
- norme de Frobenius : `norm(A, 'Fro')` / `numpy.linalg.norm(A)`
- multiplication matricielle : `*` / `numpy.dot` ou `numpy.matmul` ou `@`
- multiplication terme-à-terme : `.*` / `*`
- réorganisation de matrices : `reshape` / `numpy.reshape`

Attention : l'ordre de réorganisation des matrices diffère entre Matlab et Python. Matlab utilise l'indexation Fortran tandis que Python utilise l'indexation C (voir plus bas). Ainsi, l'équivalent Matlab du code Python `numpy.reshape(b,(51,31))` est `permute(reshape(b,31,51), [2,1])`.

C-like index order



Fortran-like index order

