



HAL
open science

Algorithmes rapides pour la modélisation d'une réponse de salle dont l'atténuation dépend de la fréquence

Achille Aknin, Roland Badeau

► To cite this version:

Achille Aknin, Roland Badeau. Algorithmes rapides pour la modélisation d'une réponse de salle dont l'atténuation dépend de la fréquence. 16e Congrès Français d'Acoustique (CFA 2022), Apr 2022, Marseille, France. hal-03559398

HAL Id: hal-03559398

<https://hal.telecom-paris.fr/hal-03559398>

Submitted on 17 May 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



16^{ème} Congrès Français d'Acoustique
11-15 Avril 2022, Marseille

Algorithmes rapides pour la modélisation d'une réponse de salle dont l'atténuation dépend de la fréquence

A. Aknin, R. Badeau

LTCI, Télécom Paris, Institut Polytechnique de Paris



En traitement du signal audio, la modélisation mathématique de la réponse de salle permet d'améliorer la qualité de l'estimation de signaux sources à partir de signaux réverbérés, afin par exemple d'effectuer une déréverbération ou de séparer un mélange de sources sonores. Dans un article précédent, nous avons travaillé sur l'implémentation d'un modèle stochastique de réponse impulsionnelle de salle, dans lequel l'atténuation exponentielle de la puissance au cours du temps dépend de la fréquence. En effet cette caractéristique est particulièrement importante si l'on veut prendre en compte la dépendance fréquentielle de l'absorption des murs, qui est généralement supérieure en hautes fréquences par rapport aux basses fréquences. Nous avons présenté une nouvelle structure de matrice, paramétrée par un unique filtre p , qui réalise cette atténuation exponentielle dépendant de la fréquence, et nous avons montré qu'elle pouvait être utilisée pour estimer les paramètres d'une réponse de salle. Cependant, cette matrice P étant de taille $T \times T$, où T est la longueur de la réponse impulsionnelle de la salle (généralement de l'ordre du millier ou de la dizaine de milliers d'échantillons en pratique), nous ne pouvons pas calculer directement des produits matriciels impliquant cette matrice P dans des conditions réelles pour des raisons de coût de calcul. Dans cet article, nous présentons donc plusieurs algorithmes rapides de produit matrice-vecteur que nous avons développés, qui exploitent la structure particulière de cette matrice, et dont la complexité est seulement de $O(T \log(T))$ ou $O(T \log^2(T))$ opérations au lieu de $O(T^2)$. Grâce à ces algorithmes, il devient possible d'exploiter la matrice P pour estimer les paramètres de vraies réponses de salle, sans être limité par la complexité de calcul.

1 Introduction

Les algorithmes de traitement du signal audio pour des applications telles que la séparation de sources ou la déréverbération reposent souvent sur la modélisation de la réponse de salle. En effet, le signal observé est le résultat de la somme de signaux sources filtrés par les réponses de salle correspondantes, éventuellement bruitée par un bruit de mesure. L'estimation conjointe des signaux sources et des réponses de salles nécessite donc de les modéliser correctement.

Le modèle le plus simple [12, 11] de réponse de salle $h(t)$ est un processus Gaussien aux échantillons indépendants et identiquement distribués (i.i.d.) dont la variance décroît exponentiellement : $\forall t > 0$,

$$\begin{aligned} h(t) &= e^{-at} b(t) \\ b(t) &\sim \mathcal{N}(0, \sigma^2). \end{aligned}$$

avec $a, \sigma > 0$.

Ce modèle est une bonne approximation pour la partie tardive de la réverbération, qui se trouve à la fin de la réponse de salle. Cependant, il ne représente pas correctement les réflexions précoces, qui sont très importantes pour l'auditeur. D'autres modèles stochastiques de réverbération existent, comme par exemple ceux utilisant des matrices de covariance spatiale [7], des variables latentes Gaussiennes complexes [8] ou encore une loi t de Student [10].

Dans une publication précédente [3], nous avons proposé un modèle stochastique de réverbération basé sur les travaux présentés dans [4, 5]. Nous avons ensuite étendu ce modèle [2] pour prendre en compte la dépendance en fréquence de la décroissance exponentielle présente dans la réponse de salle.

En effet cette décroissance exponentielle est due à l'absorption partielle de l'énergie sonore lors de la réflexion sur les surfaces de la salle. Or cette absorption dépend de la fréquence : les hautes fréquences sont généralement plus absorbées que les basses fréquences. Nous avons donc introduit un nouvel opérateur linéaire sous forme de matrice permettant de modéliser une décroissance exponentielle dont la vitesse de décroissance dépend de

la fréquence. Un nouveau modèle utilisant cette matrice a permis de mieux modéliser des réponses de salles et d'en estimer les paramètres en utilisant un algorithme d'Espérance-Maximisation [6].

Cependant, cet algorithme d'estimation des paramètres était particulièrement lent à cause des temps de calculs impliquant la matrice de décroissance exponentielle dépendant de la fréquence. Nous allons donc présenter ici des algorithmes rapides de calcul permettant d'évaluer rapidement des produits matrice-vecteur et matrice-matrice impliquant cette matrice particulière.

Nous rappellerons tout d'abord dans la section 2 la définition de la matrice de décroissance exponentielle dépendant de la fréquence et quelques résultats obtenus en utilisant cette matrice dans le cadre de la modélisation stochastique de réponses de salles. Puis nous présenterons des méthodes de calcul rapide de produits matrice-vecteur ou matrice-matrice dans la section 3. Enfin nous résumerons les résultats obtenus dans la section 4.

Notations

- p^{*n} : filtre p convolé avec lui-même n fois (fonction delta de Kronecker si $n = 0$),
- $\mathcal{F}_x(f)$: transformée de Fourier du signal x à la fréquence f ,
- $\text{TFD}_T(x)$: transformée de Fourier discrète d'ordre T du signal x ,
- $X(z) = \sum_{u \in \mathbb{Z}} x(u)z^{-u}$: transformée en Z du signal x prise en $z \in \mathbb{C}$.

2 Matrice modélisant une décroissance exponentielle dépendant de la fréquence

Dans cette section, nous introduirons tout d'abord la matrice qui nous permet d'appliquer une décroissance temporelle exponentielle, dont la vitesse de décroissance

dépendant de la fréquence. Comme mentionné en section 1, nous avons déjà utilisé cette matrice dans le cadre de la modélisation stochastique de réponses de salles dans un article précédent [2] et nous rappellerons ici quelques résultats obtenus comme exemples d'utilisation de la matrice.

2.1 Définition de la matrice

Supposons que nous souhaitions appliquer à un signal $x \in \mathbb{R}^T$ une décroissance exponentielle, dont la vitesse de décroissance dépend de la fréquence. Une solution serait par exemple d'appliquer une Transformée de Fourier à Court Terme (TFCT) à x puis d'appliquer une décroissance exponentielle différente dans chaque bande de fréquences. Cependant cette méthode pose problème : d'une part il faudrait choisir un découpage optimal des fréquences, ce qui pourrait dépendre du problème et du signal x , et d'autre part cette approche pourrait créer des discontinuités au niveau de la limite entre deux bandes de fréquences.

Au lieu de cela, nous avons décidé d'utiliser une nouvelle structure matricielle, que nous allons maintenant introduire. Soit \mathcal{P} l'ensemble des matrices de taille $T \times T$, avec $T \in \mathbb{N}^*$, tel que $P \in \mathcal{P}$ si et seulement si il existe une réponse impulsionnelle p causale telle que :

$$P = \begin{pmatrix} 1 & 0 & \dots & \dots & 0 \\ 0 & p(0) & 0 & \dots & \vdots \\ 0 & p(1) & p^{*2}(0) & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & p(T-2) & p^{*2}(T-3) & \dots & p^{*(T-1)}(0) \end{pmatrix}$$

où $p(0) \neq 0$ et p^{*u} est p convolé par lui-même u fois (si $u = 0$, alors $p^{*u} = \delta$ est la fonction delta de Kronecker).

Étant donnée une matrice $P \in \mathcal{P}$, si on en extrait la sous-matrice P_1 obtenue en supprimant la première ligne et la première colonne, alors on peut constater que $P_1 = AP'$, où A est une matrice de Toeplitz triangulaire inférieure correspondant au filtre p et P' est la matrice de taille $T-1 \times T-1$ structurée comme P et toujours construite avec le filtre p . Cela signifie qu'appliquer la matrice P à un signal x transforme le signal d'entrée au temps u de la même manière que filtrer par p^{*u} le ferait. Si la réponse en fréquence du filtre p est inférieure à 1 à toutes les fréquences, cela correspond donc à une décroissance exponentielle au cours du temps.

La figure 1 permet de visualiser l'effet de la matrice P : elle montre le log-spectrogramme de Px où $P \in \mathcal{P}$ et x est un bruit blanc Gaussien de variance 1, à côté de la réponse en fréquence du filtre p ayant servi à construire la matrice P . On peut constater que la matrice P applique bien une décroissance exponentielle (qui apparaît comme une décroissance linéaire dans le log-spectrogramme) et que la vitesse de la décroissance dépend de $\mathcal{F}_p(f)$: plus $\mathcal{F}_p(f)$ est élevé, plus la décroissance exponentielle à la fréquence f dans le log-spectrogramme est lente.

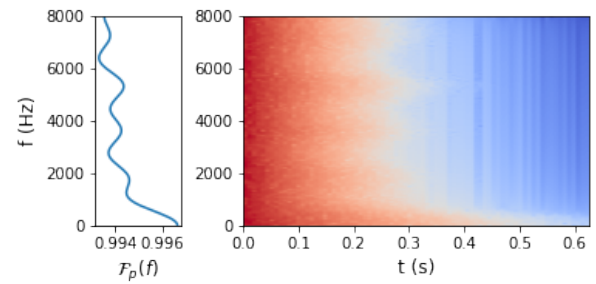


FIGURE 1 – Comparaison de la réponse en fréquence du filtre p (gauche) au log-spectrogramme de Px (droite) où $P \in \mathcal{P}$ est la matrice correspondant au filtre p et x est un bruit blanc Gaussien de variance 1.

2.2 Exemple d'utilisation de la matrice P

Les matrices de l'ensemble \mathcal{P} peuvent être utilisées dès qu'on observe une décroissance exponentielle dépendant de la fréquence. Par exemple nous pouvons modéliser l'enveloppe temps-fréquence d'une note de piano, où les différentes harmoniques ne s'atténuent pas forcément à la même vitesse. Mais nous décrivons ici une application explorée dans un article précédent [2] : la modélisation d'une réponse de salle, qui caractérise le phénomène de réverbération.

En effet comme une partie de l'énergie est absorbée à chaque réflexion du son sur les murs, cela crée une atténuation exponentielle de l'amplitude de la réponse de salle au fur et à mesure du temps. Dans un premier modèle de réponse de salle [3], nous avons simplifié le modèle pour considérer une atténuation exponentielle indépendante de la fréquence. Nous avons alors constaté que, dans le cas de réponses impulsionnelles correspondant à de vraies salles, le modèle était inadapté.

En effet, en pratique, le coefficient d'absorption de l'énergie des murs dépend de la fréquence : les basses fréquences sont en général moins absorbées que les hautes fréquences. Cela correspond donc à une atténuation exponentielle qui dépend de la fréquence également, ce qui a motivé la création d'un nouveau modèle incluant une matrice $P \in \mathcal{P}$ pour modéliser la dépendance en fréquence de l'atténuation exponentielle.

À partir de ce modèle, nous avons pu apprendre les paramètres d'une réponse de salle de la base de données MARDY [13] en utilisant un algorithme d'Espérance-Maximisation (EM) [6]. Comme le modèle est un modèle stochastique, nous avons ensuite pu générer une réponse de salle avec les paramètres appris afin de comparer la vraie réponse de salle à une réponse de salle synthétique. Nous avons également pu faire la même chose dans le cas du modèle où l'atténuation exponentielle ne dépend pas de la fréquence, afin de vérifier l'apport de la matrice $P \in \mathcal{P}$.

La figure 2 montre les trois log-spectrogrammes obtenus : la figure 2c correspond à la vraie réponse de salle tandis que la figure 2a (resp. 2b) correspond à la réponse de salle synthétique obtenue avec (resp. sans) la décroissance exponentielle dépendant de la fréquence.

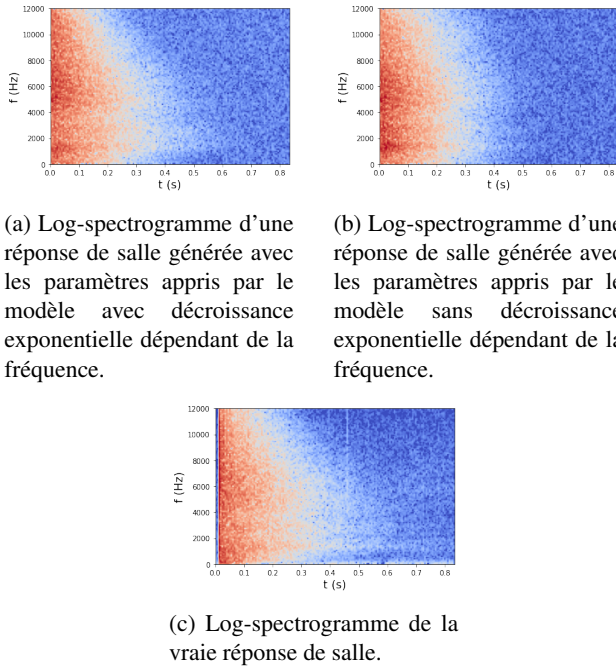


FIGURE 2 – Trois log-spectrogrammes en décibels (dB) : la vraie réponse de salle (en bas) et les réponses de salles générées par le modèle avec décroissance exponentielle dépendant de la fréquence (à gauche) et avec décroissance exponentielle constante (à droite).

Comme le montrent les log-spectrogrammes, la réponse de salle simulée à partir du modèle avec décroissance exponentielle dépendant de la fréquence est similaire à la vraie réponse de salle, mis à part le court silence avant les premières réflexions. Le modèle sans décroissance exponentielle dépendant de la fréquence, par contre, ne peut correctement représenter la décroissance exponentielle dont la vitesse dépend de la fréquence. Il tente donc de compenser cette lacune en imposant une énergie plus faible dans les hautes fréquences que dans les basses fréquences dès le début de la réponse de salle en figure 2b. On peut donc observer une grande différence dans les valeurs aux temps faibles entre la figure 2b et les figures 2a et 2c.

Cette différence peut également être vérifiée par un test auditif, en filtrant un enregistrement anéchoïque par chaque réponse de salle séparément pour comparer l'effet des trois réponses de salles¹. On constate alors que la réponse de salle générée par le modèle avec décroissance exponentielle dépendant de la fréquence est plus proche de la vraie réponse de salle que celle générée par l'ancien modèle.

Ces résultats confirment l'intérêt des matrices de l'ensemble \mathcal{P} pour la modélisation d'une décroissance exponentielle dont la vitesse dépend de la fréquence. Cependant, une lacune importante du nouveau modèle se trouve dans le temps de calcul mis par l'algorithme EM : pour une réponse de salle de 8000 échantillons, il nous faut 2 minutes pour une seule itération de l'algorithme sur un

1. Le code et quelques exemples audio sont disponibles sur GitHub : https://github.com/Aknin/P_matrix_model

processeur AMD Ryzen 5 3600 3.6 GHz et il faut environ 200 itérations pour converger. Or, une grande partie de ce temps de calcul est due au calcul de produit matriciel PM où $P \in \mathcal{P}$ et M est une matrice carrée de taille $T \times T$, où T est la longueur de la réponse de salle.

En effet, un produit de deux matrices de taille $T \times T$ a une complexité en $O(T^3)$, ce qui rend notre algorithme inutilisable sur des réponses de salle plus longues, dont la longueur est en général de plus de 15000 échantillons à 44.1 kHz. En comparaison, si on multipliait une matrice de Toeplitz à une autre matrice carrée de taille $T \times T$, le produit serait de complexité $O(T^2 \log(T))$ puisque cela revient à effectuer des produits de convolution. De la même manière, les matrices de l'ensemble \mathcal{P} étant particulièrement structurées, nous pouvons réduire la complexité des calculs de produit matrice-vecteur ou matrice-matrice impliquant une matrice $P \in \mathcal{P}$.

3 Algorithmes de calcul rapide

Dans cette section, nous parlerons des méthodes permettant de calculer rapidement des produits matrices-vecteurs ou matrice-matrice impliquant une matrice $P \in \mathcal{P}$. Nous commencerons par exposer certaines propriétés des matrices P qui montrent qu'un produit matrice-vecteur est équivalent à une évaluation de polynômes dans le domaine de Fourier, puis nous présenterons une méthode encore plus rapide utilisant une approximation de Taylor.

3.1 Propriétés des matrices P

Pour le reste de cet article, nous considérerons que P est une matrice de l'ensemble \mathcal{P} de taille $T \times T$ et construite à partir d'un filtre causal p de support fini $\llbracket 0, L_p - 1 \rrbracket$ avec $L_p \ll T$. Nous utiliserons aussi la matrice \tilde{P} de taille $\tilde{T} \times \tilde{T}$ avec $\tilde{T} = (T - 1)L_p + 1$ qui est construite comme la matrice P . Le nombre de lignes est prévu pour ne pas tronquer les filtres p^{*u} pour $0 \leq u < T$ afin d'éviter les effets de bords :

$$\tilde{P} = \begin{pmatrix} 1 & 0 & \dots & \dots & & 0 \\ 0 & p(0) & 0 & \dots & & \vdots \\ 0 & p(1) & p^{*2}(0) & \ddots & & \vdots \\ \vdots & \vdots & \vdots & \ddots & & 0 \\ \vdots & \vdots & \vdots & \ddots & & p^{*(T-1)}(0) \\ \vdots & \vdots & \vdots & \ddots & & \vdots \\ 0 & 0 & 0 & \dots & p^{*(T-1)}((T-1)L_p - 1) & \end{pmatrix}$$

En effet, p^{*u} est de support $\llbracket 0, u(L_p - 1) \rrbracket$ donc $p^{*(T-1)}$ est de support $\llbracket 0, (T-1)(L_p - 1) \rrbracket$, ce qui donne une hauteur totale de $\tilde{T} = (T-1)L_p + 1$ si on prend en compte les $T-1$ zéros au début de la dernière colonne de \tilde{P} .

Étant donné un vecteur $x \in \mathbb{R}^T$, on considérera $y = Px \in \mathbb{R}^T$ et $\tilde{y} = \tilde{P}x \in \mathbb{R}^{\tilde{T}}$. On peut facilement obtenir y à partir de \tilde{y} en ne gardant que les T premiers coefficients. Les algorithmes que nous allons présenter permettent de calculer

\tilde{y} mais, à cause des effets de bord, ils ne fonctionneraient pas si on essayait de calculer y directement.

Théorème 1. Si $\tilde{y} = \tilde{P}x$, alors sa transformée en Z vérifie : pour tout $z \in \mathbb{C}$,

$$\tilde{Y}(z) = X\left(\frac{z}{P(z)}\right)$$

où $X(z)$ est la transformée en Z du signal x et $P(z)$ est la transformée en Z de la réponse impulsionnelle p .

Démonstration. Par définition de la matrice \tilde{P} , nous avons : pour tout $u \in \llbracket 0, \tilde{T} - 1 \rrbracket$,

$$\tilde{y}(u) = \sum_{v=0}^{\tilde{T}-1} p^{*v}(u-v)x(v)$$

Nous avons alors, pour la transformée en Z de \tilde{y} : pour tout $z \in \mathbb{C}$,

$$\begin{aligned} \tilde{Y}(z) &= \sum_{u=0}^{\tilde{T}-1} \tilde{y}(u)z^{-u} \\ &= \sum_{u=0}^{\tilde{T}-1} \sum_{v=0}^{\tilde{T}-1} p^{*v}(u-v)x(v)z^{-u} \\ &= \sum_{u=0}^{\tilde{T}-1} \sum_{v=0}^{\tilde{T}-1} p^{*v}(u-v)z^{-(u-v)}x(v)z^{-v} \\ &= \sum_{v=0}^{\tilde{T}-1} x(v)z^{-v} \sum_{u=0}^{\tilde{T}-1} p^{*v}(u-v)z^{-(u-v)} \\ &= \sum_{v=0}^{\tilde{T}-1} x(v)z^{-v} \sum_{u=-v}^{\tilde{T}-v-1} p^{*v}(u)z^{-u} \\ &= \sum_{v=0}^{\tilde{T}-1} x(v)z^{-v} P^v(z) \end{aligned}$$

car le support de p^{*v} est $\llbracket 0, v(L_p - 1) \rrbracket$, ce qui est inclus dans $\llbracket -v, \tilde{T} - v - 1 \rrbracket$ pour tout $v \in \llbracket 0, \tilde{T} - 1 \rrbracket$. Donc

$$\tilde{Y}(z) = \sum_{v=0}^{\tilde{T}-1} x(v) \left(\frac{z}{P(z)}\right)^{-v} = X\left(\frac{z}{P(z)}\right)$$

□

Grâce à ce résultat, nous pouvons calculer la Transformée de Fourier Discrète (TFD) de \tilde{y} . En effet, celle-ci correspond à la transformée en Z prise en certains points du cercle unité : pour $k \in \llbracket 0, \tilde{T} - 1 \rrbracket$, on note $e_k = e^{2i\pi k/\tilde{T}}$

$$\begin{aligned} \text{TFD}_{\tilde{T}}(\tilde{y})(k) &= \tilde{Y}(e_k) \\ &= X\left(\frac{e_k}{P(e_k)}\right) \\ \text{TFD}_{\tilde{T}}(\tilde{y})(k) &= \sum_{u=0}^{\tilde{T}-1} x(u)z_k^{-u} \end{aligned}$$

où $z_k = \frac{e_k}{P(e_k)} = \frac{e_k}{\text{TFD}_{\tilde{T}}(p)(k)}$.

Autrement dit, calculer la TFD de \tilde{y} revient à évaluer le polynôme de degré $\tilde{T} - 1$ de coefficients $(x(u))_{u=0}^{\tilde{T}-1}$ en $\tilde{T}/2 + 1$

points si \tilde{T} est pair² : les z_k^{-1} pour $k \in \llbracket 0, \tilde{T}/2 \rrbracket$. Notons qu'il est inutile de calculer les valeurs pour k plus grand car, \tilde{y} étant réel, $\text{TFD}_{\tilde{T}}(\tilde{y})(\tilde{T} - k) = \overline{\text{TFD}_{\tilde{T}}(\tilde{y})(k)}$.

Afin d'éviter les effets de bords dans la preuve du théorème 1, nous avons fixé $\tilde{T} = (T - 1)L_p + 1$. Cependant, en pratique, il suffit que \tilde{T} soit légèrement plus grand que T car $p^{*(T-1)}$ converge généralement rapidement vers 0. Dans notre cas, nous avons fixé $\tilde{T} \approx 1.2T$ ³ et avons constaté que, pour $T \geq 100$, l'erreur est inférieure à 10^{-7} avec le filtre p obtenu par le modèle de réponse de salle présenté en section 2.2. Autrement dit, nous considérerons pour les calculs de complexité que $\tilde{T} = O(T)$.

L'approche classique pour évaluer des polynômes est la méthode de Horner [9, Chap. 4], qui permettrait de calculer $\text{TFD}_{\tilde{T}}(\tilde{y})$ en $O(T^2)$ opérations. Cela correspond à la complexité du produit matrice-vecteur classique. À la place, nous pouvons utiliser un algorithme plus avancé [1, Chap. 8] dont la complexité est seulement de $O(T \log^2(T))$.

Cependant, cet algorithme est difficile à implémenter efficacement car il nécessite d'optimiser les accès mémoire pour éviter de trop grands temps de calculs. Nous avons donc développé une autre méthode de calcul, dérivée de cette même approche d'évaluation de polynômes en introduisant une approximation de Taylor, que nous exposons maintenant.

3.2 Calcul rapide par une approximation de Taylor

Comme expliqué en section 3.1, calculer un produit matrice-vecteur impliquant une matrice $P \in \mathcal{P}$ est équivalent à évaluer X , la transformée en Z de x , en \tilde{T} points de \mathbb{C} : les points $z_k = \frac{e_k}{\text{TFD}_{\tilde{T}}(p)(k)}$ pour $k \in \llbracket 0, \tilde{T} - 1 \rrbracket$. On note :

$$A(z) = X(z^{-1}) = \sum_{u=0}^{\tilde{T}-1} x(u)z^u \quad (1)$$

On cherche donc à évaluer le polynôme A en z_k^{-1} pour $k \in \llbracket 0, \tilde{T}/2 \rrbracket$.

Cette évaluation de polynôme peut être coûteuse mais nous savons rapidement évaluer ce même polynôme aux points e_k^{-1} avec l'algorithme de transformée de Fourier rapide puisque $A(e_k^{-1}) = X(e_k) = \text{TFD}_{\tilde{T}}(x)(k)$. Or, en pratique, le point z_k^{-1} est proche de e_k^{-1} car la réponse en fréquence du filtre p est proche de 1. En effet, si la réponse en fréquence du filtre p était très inférieure à 1, cela correspondrait à une décroissance exponentielle très rapide, ce qui n'est pas réaliste dans l'application envisagée en section 2.2.

La figure 3 le confirme : à partir d'un filtre p correspondant à une vraie réponse de salle, on calcule les points z_k correspondant et on affiche également les points e_k . Comme on peut l'observer, z_k est très proche de e_k pour tout $k \in \llbracket 0, \tilde{T} - 1 \rrbracket$. Une rapide approximation nous donne par conséquent $A(z_k^{-1}) = A(e_k^{-1}) + O(|z_k^{-1} - e_k^{-1}|)$.

2. Si \tilde{T} est impair, il suffit d'utiliser $\tilde{T} + 1$ et tous les résultats restent valables.

3. Notons que la transformée de Fourier rapide est plus efficace lorsque l'ordre \tilde{T} est un multiple de petits nombres premiers, il sera donc parfois intéressant d'augmenter encore \tilde{T} si cela résulte en un multiple de petits nombres premiers.

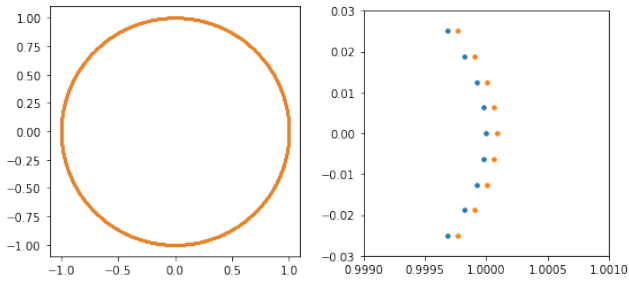


FIGURE 3 – Comparaison des points z_k (orange) aux points e_k (bleu) dans l'espace des complexes dans le cas d'un filtre p appris à partir d'une vraie réponse de salle. La figure de droite est centrée sur le point $(1, 0)$ pour mieux distinguer les points.

Une meilleure approximation peut être obtenue en utilisant la formule de Taylor-Young :

$$A(z_k^{-1}) = \sum_{n=0}^N \frac{A^{(n)}(e_k^{-1})}{n!} (z_k^{-1} - e_k^{-1})^n + o(|z_k^{-1} - e_k^{-1}|^N) \quad (2)$$

où $A^{(n)}$ est la n -ième dérivée du polynôme A défini dans l'équation (1).

Le calcul de $A^{(0)}(e_k^{-1})$ est rapide : il correspond à la transformée de Fourier rapide du signal x en \tilde{T} points. Pour ce qui est des dérivées, elles correspondent également à des transformées de Fourier :

$$A^{(n)}(e_k^{-1}) = \sum_{u=0}^{T-1-n} \frac{(u+n)!}{u!} x(u+n) (e_k^{-1})^u \quad (3)$$

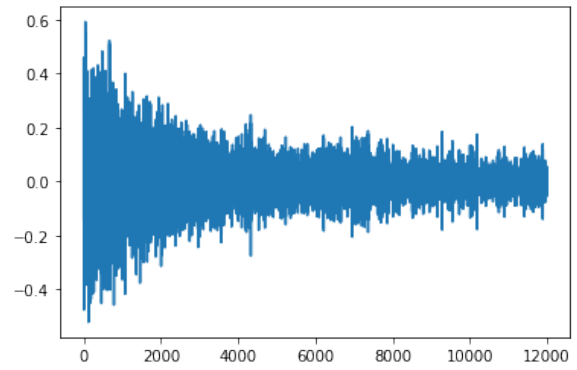
Ainsi, si on transforme le signal x en $x_1(u) = (u+1)x(u+1)$ pour tout $u \in \llbracket 0, T-2 \rrbracket$, alors $A^{(1)}(e_k^{-1}) = \text{TFD}_{\tilde{T}}(x_1)(k)$. En répétant l'opération de décalage et multiplication point par point, on peut ainsi obtenir récursivement toutes les dérivées $A^{(n)}(e_k^{-1})$ pour $n \in \llbracket 0, N \rrbracket$.

L'approximation de Taylor nous permet donc de calculer efficacement le produit matrice-vecteur Px en calculant d'abord $\text{TFD}_{\tilde{T}}(\tilde{y})(k) = A(z_k^{-1})$ puis en procédant à une TFD inverse.

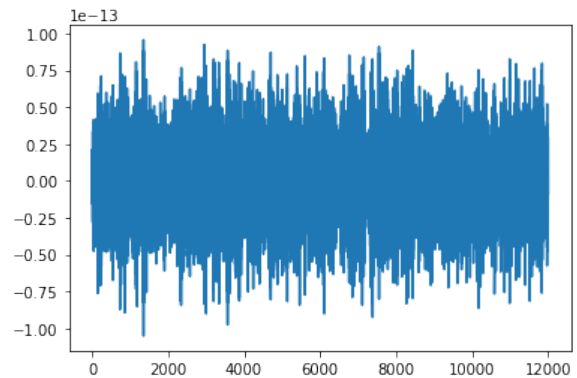
La partie la plus coûteuse de l'algorithme est le calcul des $N+1$ TFD correspondant à chacune des dérivées du polynôme A aux points e_k . Grâce à la transformée de Fourier rapide, la TFD d'ordre \tilde{T} peut être calculée en $O(\tilde{T} \log(\tilde{T}))$ opérations, ce qui signifie que la complexité du produit matrice-vecteur Px est $O(NT \log(T))$ (on rappelle que $\tilde{T} = O(T)$). La complexité du produit matrice-matrice sera donc de $O(NT^2 \log(T))$.

En pratique, le choix de l'ordre dépendra du filtre p et de l'horizon temporel T . Dans nos tests, nous avons utilisé un filtre p appris sur une réponse de salle de la base de données MARDY [13] et nous nous sommes limités à $T \leq 12000$.

Avec ce filtre p , si $T = 12000$ et x est un bruit blanc Gaussien de variance 1, la figure 4b montre la différence entre $y = Px$ et y_{taylor} , l'estimation de y faite par une approximation de Taylor d'ordre 30. On constate qu'un



(a) $y = Px$ dans le cas où p est un filtre appris sur une vraie réponse de salle et x est un bruit blanc Gaussien aux échantillons i.i.d.



(b) Erreur ($y_{\text{taylor}} - y$) commise sur l'estimation de $y = Px$ par une approximation de Taylor d'ordre 30 pour $T = 30$.

FIGURE 4 – Comparaison de $y = Px$ (haut) et ($y_{\text{taylor}} - y$) l'erreur commise sur l'estimation de y (bas).

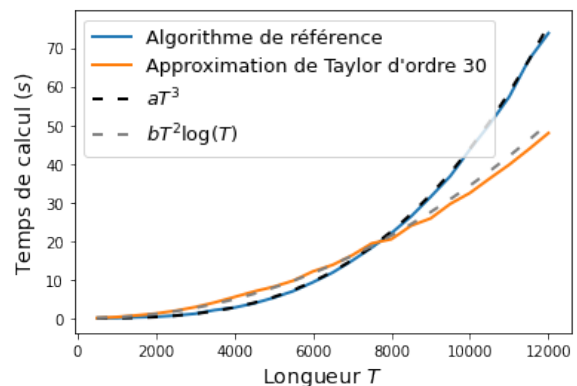


FIGURE 5 – Comparaison du temps de calcul (haut) du produit matrice-matrice sous MATLAB dans le cas de l'algorithme de référence (bleu) et de l'algorithme basé sur l'approximation de Taylor à l'ordre 30 (orange) pour plusieurs horizons temporels T .

ordre $N = 30$ permet de calculer le produit matrice-vecteur avec une erreur négligeable devant y (de l'ordre de 10^{-13}), et cela de manière stable au cours du temps.

La figure 5 présente le temps de calcul d'un produit matrice-matrice PM , où M est une matrice de bruit blanc Gaussien aux échantillons i.i.d. Ce temps de calcul est représenté pour des horizons temporels T allant de 100 à 12000 dans deux cas. L'algorithme de référence calcule la matrice P en effectuant récursivement des produits de convolutions pour remplir la matrice colonne par colonne, ce qui a une complexité de $O(T^2 \log(T))$. Il effectue ensuite le produit matriciel implémenté dans MATLAB, avec une complexité de $O(T^3)$. L'algorithme basé sur l'approximation de Taylor utilise quant à lui l'équation (2) pour calculer les produits matrice-vecteur pour chaque colonne de la matrice M en temps $O(T^2 \log(T))$. Ces temps de calculs ont été mesurés sous MATLAB avec un processeur AMD Ryzen 5 36000 3.6 GHz. La figure inclut également la courbe aT^3 (resp. $bT^2 \log(T)$) où nous avons choisi le paramètre a (resp. b) pour correspondre au mieux à la courbe de temps de calcul de l'algorithme de référence (resp. l'algorithme basé sur l'approximation de Taylor).

Comme le montre la figure ??, le temps de calcul est à l'avantage de l'algorithme de référence pour $T < 8000$ mais l'algorithme basé sur l'approximation de Taylor permet d'accélérer le calcul lorsque $T \geq 8000$. Les courbes aT^3 et $bT^2 \log(T)$ confirment quant à elles la complexité théorique puisqu'elles suivent correctement les courbes de temps de calcul. L'algorithme présente donc un intérêt significatif lorsqu'on travaille avec de grands horizons temporels, et on gagnerait certainement encore plus de temps de calcul avec une implémentation optimisée sur un langage compilé.

4 Conclusion

Dans cet article, nous sommes revenus sur une méthode de modélisation d'une décroissance exponentielle dont la vitesse dépend de la fréquence. Nous avons rappelé dans la section 2 la matrice $P \in \mathcal{P}$ nous permettant d'effectuer cette modélisation et un exemple d'application, puis nous avons présenté en section 3 deux algorithmes nous permettant d'évaluer rapidement des produits matrice-vecteur ou matrice-matrice impliquant la matrice P de dimension $T \times T$.

Le premier algorithme de calcul matrice-vecteur, basé sur des évaluations rapides de polynômes, a une complexité de $O(T \log^2(T))$ au lieu de $O(T^2)$ mais nous ne sommes pas parvenus à l'implémenter efficacement car il nécessite d'optimiser le code et les accès mémoire. Le deuxième algorithme, basé sur une approximation de Taylor, est plus simple à implémenter car il utilise des transformées de Fourier rapides, qui sont déjà implémentées en MATLAB. Cela nous permet d'évaluer un produit matrice-vecteur en $O(TN \log(T))$ opérations, où N est l'ordre du développement de Taylor. En pratique, on constate en effet que le temps de calcul de produits matrice-matrice est à l'avantage de cet algorithme pour $T \geq 8000$ par rapport à l'algorithme de référence.

Nous avons montré qu'il est possible de calculer rapidement des produits matrice-vecteur et matrice-matrice impliquant une matrice $P \in \mathcal{P}$. Il serait intéressant

d'implémenter de manière optimale les deux algorithmes dans un langage compilé afin d'accélérer les calculs.

Nos travaux futurs se concentreront sur l'utilisation des matrices de l'ensemble \mathcal{P} dans le cas de la modélisation de réponses de salles. Avec ces nouveaux algorithmes de calcul rapide, nous pourrions utiliser le modèle présenté dans notre article précédent [2] pour résoudre des problèmes plus complexes, comme la déréverbération ou la séparation de sources audio, en utilisant un algorithme EM pour estimer conjointement le signal de la source audio et la réponse de salle.

Références

- [1] A. V. Aho and J. E. Hopcroft. *The design and analysis of computer algorithms*. Pearson Education India, 1974.
- [2] A. Akinin and R. Badeau. Stochastic reverberation model with a frequency dependent attenuation. In *2021 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 351–355. IEEE, 2021.
- [3] A. Akinin, T. Dupré, and R. Badeau. Evaluation of a stochastic reverberation model based on the image source principle. In *International Conference on Digital Audio Effects*, 2020.
- [4] R. Badeau. Unified stochastic reverberation modeling. In *Proc. of 26th European Signal Processing Conference (EUSIPCO)*, Rome, Italy, Sept. 2018.
- [5] R. Badeau. Common mathematical framework for stochastic reverberation models. *The Journal of the Acoustical Society of America*, 2019. Special issue on room acoustics modeling and auralization.
- [6] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1) :1–38, 1977.
- [7] N. Q. Duong, E. Vincent, and R. Gribonval. Under-determined reverberant audio source separation using a full-rank spatial covariance model. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(7) :1830–1840, 2010.
- [8] L. Girin and R. Badeau. On the use of latent mixing filters in audio source separation. In *International Conference on Latent Variable Analysis and Signal Separation*, pages 225–235. Springer, 2017.
- [9] D. E. Knuth. *Art of computer programming, volume 2 : Seminumerical algorithms*. Addison-Wesley Professional, 2014.
- [10] S. Leglaive, R. Badeau, and G. Richard. Student's t source and mixing models for multichannel audio source separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(6) :1154–1168, 2018.
- [11] J. A. Moorer. About this reverberation business. *Computer Music Journal*, 3(2) :13–28, 1979.
- [12] M. R. Schroeder. Frequency-correlation functions of frequency responses in rooms. *The Journal of the Acoustical Society of America*, 34(12) :1819–1823, 1962.
- [13] J. Y. Wen, N. D. Gaubitch, E. A. Habets, T. Myatt, and P. A. Naylor. Evaluation of speech dereverberation algorithms using the MARDY database. In *Proc. Intl. Workshop Acoust. Echo Noise Control (IWAENC)*, 2006.