Titre: Title:	Une généralisation de l'analyse en composantes indépendantes pour le débruitage des signaux de parole
Auteur: Author:	Mohamed Salah Ben Slimen
Date:	2021
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Ben Slimen, M. S. (2021). Une généralisation de l'analyse en composantes indépendantes pour le débruitage des signaux de parole [Master's thesis, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/6291/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/6291/
Directeurs de recherche: Advisors:	Antoine Saucier
Programme: Program:	Maîtrise recherche en mathématiques appliquées

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Une généralisation de l'analyse en composantes indépendantes pour le débruitage des signaux de parole

MOHAMED SALAH BEN SLIMEN
Département de mathématiques et de génie industriel

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
Mathématiques appliquées

Avril 2021

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Ce mémoire intitulé :

Une généralisation de l'analyse en composantes indépendantes pour le débruitage des signaux de parole

présenté par **Mohamed Salah BEN SLIMEN**
en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
a été dûment accepté par le jury d'examen constitué de :

Denis MARCOTTE, président
Antoine SAUCIER, membre et directeur de recherche
Roland MALHAMÉ, membre

DÉDICACE

À mes parents et mes soeurs,
vous me manquez. . .

REMERCIEMENTS

J'aimerais d'abord exprimer ma gratitude envers ma famille qui a toujours été la pour moi malgré la distance, mes parents qui m'ont toujours soutenu pendant les périodes les plus difficiles. Je les remercierai jamais assez pour tous leurs sacrifices, je leur dédie tout mon mémoire. Je remercie aussi mes soeurs Ines et Bahaa qui ont toujours été là pour moi et qui me manquent beaucoup. Je vous suis plus que reconnaissant.

Je voudrais adresser mes remerciements les plus chaleureux à mon directeur de recherche, Prof. Antoine Saucier pour tout le soutien et la confiance qu'il m'a apporté pendant tout le projet de recherche. Je le remercie pour sa disponibilité et ses conseils tout au long du projet.

Mes remerciements chaleureux vont aux membres du jury, Prof Denis Marcotte
et Prof Roland Malhamé de bien vouloir juger ce mémoire.
Je tiens à remercier nos partenaires M. Vikrant Singh Tomar de Fluent.ai et M.Wissem Maazoun de MITACS pour leur aide et le soutien financier.

Je remercie aussi mes amis Myriam, Ghassen, Oussama et Noursen que je considère comme ma deuxième famille au Canada et qui m'ont toujours soutenu. Je tiens aussi à remercier mes meilleurs amis Achref et Cyrine qui me manquent et que j'ai hate de retrouver bientôt.

RÉSUMÉ

Les systèmes de reconnaissance vocale existants sont très précis et atteignent des performances élevées lorsqu'ils traduisent le signal de parole en une intention. Ces appareils nécessitent souvent la réception de signaux vocaux sans bruit afin d e p rédire a vec précision l'intention du locuteur. La présence de bruit dans les signaux vocaux peut conduire à de fausses prédictions qui peuvent conduire le système à exécuter de fausses actions.

La recherche effectuée fait partie des stages Mitacs Accelerate, qui ont été menés en collaboration avec Fluent.ai, une startup montréalaise spécialisée en intelligence artificielle, plus précisément dans la reconnaissance vocale pour des appareils utilisés dans les maisons intelligentes. L'objectif principal de la recherche est de développer un nouvel algorithme agissant comme front-end pour réduire le bruit des signaux vocaux en utilisant la séparation des sources.

Le travail effectué introduit d'abord un examen critique des approches développées précédemment pour appliquer la séparation des sources et réduire le bruit des données. La revue de ces méthodes a permis de développer un algorithme capable de séparer les signaux en un signal de parole et un signal de bruit puis de reconstruire la source de parole débruitée.

Deux méthodes ont été proposées aux deux situations possibles : avec ou sans la présence de délais entre les microphones. Les deux algorithmes ont été testés et validés à l'aide d'enregistrements contenant du bruit fournis par notre partenaire industriel Fluent.ai. L'algorithme a été implémenté en tant qu'interface pour l'algorithme de Fluent.ai qui utilise les réseaux de neurones artificiels pour comprendre l'intention du locuteur. Pour cela, nous avons utilisé le même environnement utilisé par Fluent.ai qui a été entièrement implémenté en langage de programmation Python.

ABSTRACT

The existing speech recognition systems reach high and precise performances when understanding the intention of the speaker. These devices require often the reception of clean speech signals in order to accurately predict the intent of the speaker. The presence of noise in speech signals can lead to false predictions which can lead the system to execute false actions.

The research done is part of the Mitacs Accelerate internships, which were conducted in collaboration with Fluent.ai, a Montréalese startup specializing in speech recognition for smart home devices. The main purpose of the research is to develop a new front-end algorithm to help reducing the noise from speech signals using sources separation.

The work done introduces first a critical review of approaches developed previously to apply source separation and reduce the noise from data. The review of those methods helped to develop an algorithm able to separate signals to a speech signal and a noise signal then reconstruct a cleaner speech.

Two methods were proposed to both possible situations: with or without the presence of delays between the microphones. Both algorithms were investigated and validated using recordings containing noise that were provided by our industrial partner Fluent.ai. The algorithm was implemented as a front-end for a software that uses deep learning neural networks to understand the intent of the speaker. For that we used the same environment used by Fluent.ai which was entirely implemented in Python language.

TABLE DES MATIÈRES

LISTE DES FIGURES

LISTE DES SIGLES ET ABRÉVIATIONS

ACI	Analyse en Composantes Indépendantes
AI	Artificial Intelligence
ARM	Acorn RISC Machine
ASIR	Automatic Speech intent Recognition
ASR	Automatic Speech Recognition
BSS	Blind Source Separation
EEG	Electroencéphalographie
ICA	Independent Component Analysis
ICA/2S/2PM	Independent Component Analysis/2Sources /2Point Moments
MFCC	Mel-Frequency Cepstrum
NLU	Natural Language Understanding
SLU	Speech Language Understanding
SVM	Support Vector Machine
VAD	Voice Activity Detection

CHAPITRE 1 INTRODUCTION

La reconnaissance vocale est un des domaines de recherche les plus en vogue de ces dernières années. L'interaction entre l'humain et la machine devient de plus en plus facile grâce à la détection et la compréhension de la voix humaine. La voix humaine est une suite d'ondes créées par les cordes vocales grâce à la vibration. Cette voix est reçue par l'appareil de reconnaissance vocale sous forme de signal qui sera traité grâce à des techniques de traitement de signaux afin d'en comprendre l'intention. La machine peut apprendre à comprendre la voix humaine grâce à différentes techniques en utilisant l'apprentissage machine. La voix est une des interfaces naturelles les plus faciles à utiliser. Cela permet de remplacer plusieurs actions qui demandent l'interaction physique de la personne avec la machine, chose qui permet de faciliter l'utilisation de la machine dans des cas difficiles (e.g. contrôle des fonctionnalités d'une voiture en conduisant).

Le domaine de la reconnaissance vocale a vu naître des algorithmes performants pour comprendre l'intention de l'utilisateur, cependant ces algorithmes sont affectés par la présence de bruit et de réverbération dans les signaux de paroles et de bruit. Dans le cas général, la personne se tient à distance de l'appareil, chose qui dégrade le signal par l'ajout de bruit et de réverbération. La reconnaissance vocale automatique doit être robuste au bruit et à la réverbération afin de ne pas détériorer la précision de l'appareil. Dans le but d'atteindre cet objectif, l'utilisation de techniques de traitement des signaux devient impérative. En présence de bruit, l'appareil de reconnaissance vocale reçoit un signal de parole provenant d'une ou plusieurs sources de parole superposées à un autre signal provenant d'une source de bruit. Ces deux composantes des signaux reçus par l'appareil réduisent la précision de l'algorithme de compréhension de la voix humaine.

Pendant les récentes années, la recherche dans le domaine du traitement des signaux ainsi qu'en intelligence artificielle ont prêté beaucoup d'attention à la résolution du problème de BSS (Blind Source Separation). Dans le cas général, les microphones reçoivent une superposition de signaux qui peuvent être séparés en composantes indépendantes. On distingue essentiellement deux types d'algorithmes permettant de résoudre ce genre de problèmes : les algorithmes d'apprentissage non supervisés ou supervisés.

Les algorithmes d'apprentissage non supervisés permettent de séparer des signaux superposés sans avoir recours à des données d'entraînement au préalable ni à des données liées aux coefficients de mixage des microphones. L'Analyse en Composantes Indépendantes (ACI) est l'un des algorithmes les plus connus de ce genre. Les algorithmes d'apprentissage supervisés

comme les réseaux de neurones artificiels permettent d'apprendre à partir de données d'entraînements à estimer les paramètres de poids permettant de faire la séparation de signaux . Leur application à ce problème pourrait aider à séparer la superposition des signaux reçus par le micro en deux composantes indépendantes, la source de parole et la source de bruit.

Notre projet de recherche se fait dans le cadre d'une coopération avec notre partenaire industriel Fluent.ai. Notre partenaire utilise un système de reconnaissance automatique de l'intention Automatic Speech to Intent Recognition (ASIR). Ce système se base sur une architecture utilisant les réseaux de neurones profonds qui servent à reconnaître une succession de commandes. Ce système utilise un processeur ARM qui calcule les prédictions localement sans avoir besoin de puissance de calcul infonuagique. ARM est un type d'architecture de processeur très utilisé dans les petits appareils électroniques comme les téléphones intelligents ou les appareils de maisons intelligentes. Le système est équipé de deux microphones et a une position fixe dans une salle où il y aurait présence d'une personne ou plus. La forme géométrique de l'appareil peut changer, cependant l'appareil est relativement petit et les deux microphones sont très proches l'un de l'autre. L'appareil a une fonction de réveil ('wake word') qui permet d'activer la reconnaissance vocale de l'appareil grâce à un signal de réveil comme "Hey Fluent". L'appareil est conçu afin de recevoir des signaux vocaux d'intention, e.g. "Turn On coffee machine". Ce genre de signaux d'intention sont traduits en actions par un réseau de neurones profond. À part les signaux d'intention, il y a aussi présence de signaux comme de la musique, parasites ou du bruit venant d'autres appareils comme un ventilateur. Dans la situation typique que nous considérons, les interlocuteurs parlent à tour de rôle, avec de courtes périodes où on entendrait les personnes parler ensemble ou seulement du bruit. Cette situation typique diffère de la situation du problème de séparation aveugle des sources où on aurait des personnes qui parlent simultanément. On considère alors que l'appareil entend une personne à la fois parler en présence de bruit, avec la possibilité d'avoir un changement de locuteur qui serait plus loin ou plus proche de l'appareil. Dans ce cas, le système de Fluent.ai devrait pouvoir reconnaître le signal provenant de la personne qui parle comme étant le signal d'intention. Notre partenaire souhaite isoler seulement le signal d'intention afin de l'envoyer à l'algorithme de reconnaissance d'intention. De plus l'appareil en question possède deux microphones, ceci implique l'existence de délais lors de la réception des ondes sonores par les microphones. Plusieurs algorithmes ont vu le jour pour résoudre ce genre de problème, cependant aucune méthode ne réussit à parfaitement séparer des sources des signaux de microphones. Les délais entre les microphones changent en fonction de la distance qui sépare ces derniers ainsi que la distance qui sépare les sources des microphones. L'ajout de la contrainte des délais rend la réduction de bruit des signaux de parole plus compliquée. La réduction du bruit en tenant compte de délais entre deux microphones reste

encore un sujet de recherche qui n'a pas été exploré en détails.

Pour ces raisons, le développement d'un algorithme pour séparer le bruit de la source de parole en respectant l'existence de délais est nécessaire. Cet algorithme doit aussi pouvoir suivre la variation du délai si la source de parole change de position. Les signaux de parole et de bruits sont supposés inconnus. La relation entre les signaux ainsi que les paramètres de mixage sont aussi inconnus. Sachant que le bruit réduit les performances de la reconnaissance automatique de l'intention, cet algorithme devrait pouvoir estimer les différents paramètres de mixage ainsi que les délais afin de pouvoir reconstruire correctement les signaux et envoyer seulement le signal de parole à l'algorithme de reconnaissance d'intention de Fluent.ai.

L'objectif principal de ce mémoire est de développer une méthode basée sur une extension de l'ACI afin de séparer les signaux des microphones et enlever le bruit en présence de délais. Pour résoudre ce problème nous devons : - Formuler le problème sans délais entre les deux microphones

- Adapter la méthode pour tenir compte des délais entre les microphones
- Détecter les périodes calmes lorsque le locuteur ne parle pas.
- Estimer les coefficient de mixage ainsi que les délais pour la source de parole et la source de bruit.
- Reconstruire le signal de parole sans bruit.

CHAPITRE 2 DÉFINITION DU PROBLÈME

2.1 Introduction

L'appareil de reconnaissance vocale automatique permet de comprendre l'intention de la voix humaine. Ce genre d'appareil utilise généralement des algorithmes de traitement de la langue naturelle qui permettent de prédire l'intention de l'humain quand le signal de parole est reçu par l'appareil. Un appareil conventionnel de reconnaissance vocale comprend un système de compréhension de la langue parlée (*Spoken Language Understanding (SLU)*). Le système conventionnel intègre deux modules principaux. Le premier est un algorithme de reconnaissance automatique de la parole (*Automatic Speech Recognition (ASR)*) qui sert à traduire le signal en texte. Le deuxième module est un algorithme de compréhension du langage naturel (*Natural Language Understanding (NLU)*) qui est utilisé pour comprendre le sens de la phrase et l'intention du locuteur. L'appareil de notre partenaire industriel Fluent.ai a une architecture différente, il intègre un système complet de compréhension de langage parlé (*End-to-end SLU*) qui comprend un seul module principal. Ce module peut comprendre directement l'intention du locuteur à partir du signal de parole sans avoir besoin de le traduire en texte, comme le montre la figure (2.1) de Lugosch et al. [2].

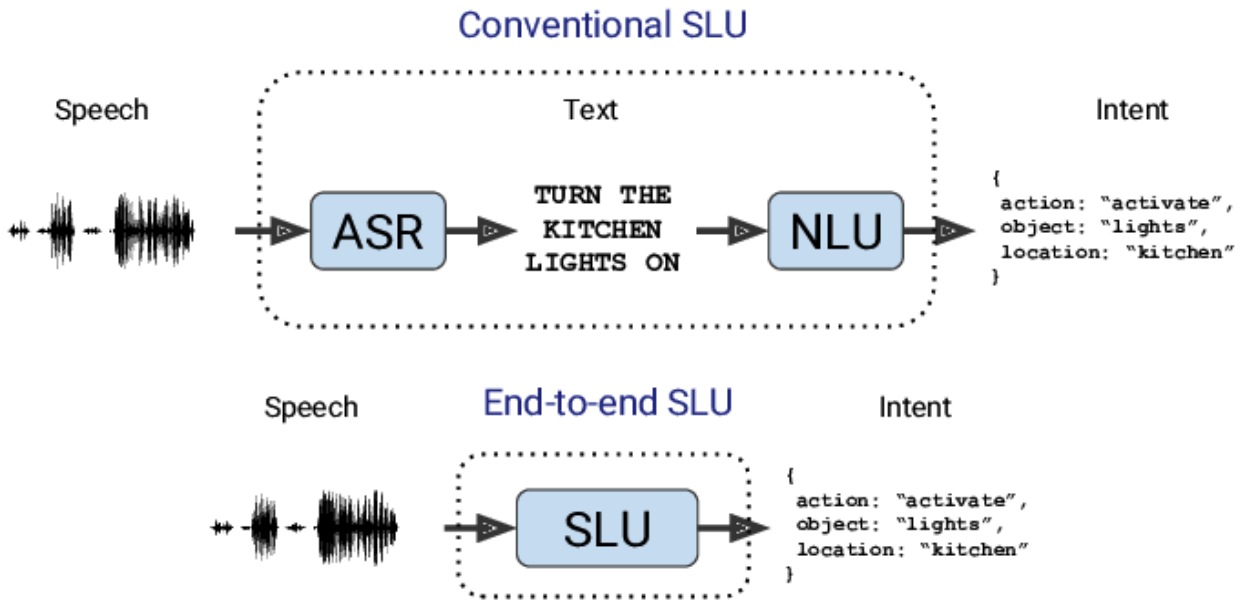

Figure 2.1 Différence entre l'architecture conventionnelle SLU et End-to-end SLU.

Ces algorithmes sont très sensibles à la présence de bruit dans les signaux car la prédiction de l'intention est estimée directement à partir du signal et non du texte. En présence de signaux autres que le signal de parole, i.e. bruit, la performance de l'appareil de reconnaissance vocale automatique diminue. Dans ce projet, on se focalise sur la réduction du bruit. Le problème est de séparer les signaux sources, i.e. signal de parole et signal de bruit en utilisant les deux signaux reçus par les microphones de l'appareil. Les signaux de parole et de bruit sont supposés être inconnus. La relation entre les signaux des sources et les microphones est aussi inconnue. Ce problème est connu comme le problème de séparation aveugle de sources (*Blind Source Separation (BSS)*).

2.2 Le *Cocktail Party Problem* et la séparation aveugle de sources

En 1953, Cherry [3] a introduit le problème du *Cocktail Party* qui décrit la situation où plusieurs personnes parlent en même temps. En présence de plusieurs microphones, on enregistre des superpositions des voix des personnes qui parlent. Dans ce contexte, la séparation aveugle de sources a vu le jour et est devenu un sujet de recherche actif. C'est une méthode de traitement des signaux qui permet de reconstruire les signaux sources qui sont inconnus à partir des signaux de microphones qui sont connus. Plusieurs méthodes de BSS ont été proposées par des chercheurs des domaines du traitement statistique des signaux, du traitement des signaux audio, de la psychologie cognitive ainsi que du traitement d'images. La séparation aveugle de sources est une méthode de séparation qui suppose que les signaux des sources sont indépendants, inconnus et sans délais d'un microphone à un autre.

On suppose que les signaux acoustiques reçus par les microphones sont des combinaisons linéaires des sources inconnues. Chaque signal est une suite de valeurs qui représente la lecture du microphone au temps $n \in \mathbb{N}$ et est défini comme une observation $x(n)$. Cette observation est formulée comme une combinaison linéaire des mêmes sources inconnues $s(n) \in \mathbb{R}$. Les sources sont inconnues et indépendantes au cours du temps $n \in \mathbb{N}$. On appelle A la matrice de mixage inconnue de dimensions $m \times n$. Chaque élément $a_{i,j}$ de la matrice A correspond à un des coefficients de mixage du microphone $x_m(n)$ pour la source $s_n(n)$. On peut modéliser le système par

$$x(n) = A \ \ s(n), \tag{2.1}$$

où

$$x(n) = \begin{bmatrix} x_1(n) \\ x_2(n) \\ \vdots \\ x_m(n) \end{bmatrix}, \tag{2.2}$$

$$A_{m,n} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{pmatrix}, \tag{2.3}$$

et

$$s(n) = \begin{bmatrix} s_1(n) \\ s_2(n) \\ \vdots \\ s_n(n) \end{bmatrix}. \tag{2.4}$$

L'objectif de la séparation aveugle des sources est de reconstruire les signaux des sources $s(n)$ à partir des signaux des microphones. Pour cela, il est nécessaire d'estimer la matrice A et ensuite de calculer son inverse W dans le cas où $m = n$:

$$W = A^{-1}, \tag{2.5}$$

$$s(n) = Wx(n). \tag{2.6}$$

Ainsi on peut reconstruire les signaux des sources $s(n)$ en suivant le diagramme de la figure (2.2) selon Rana et al. [4]

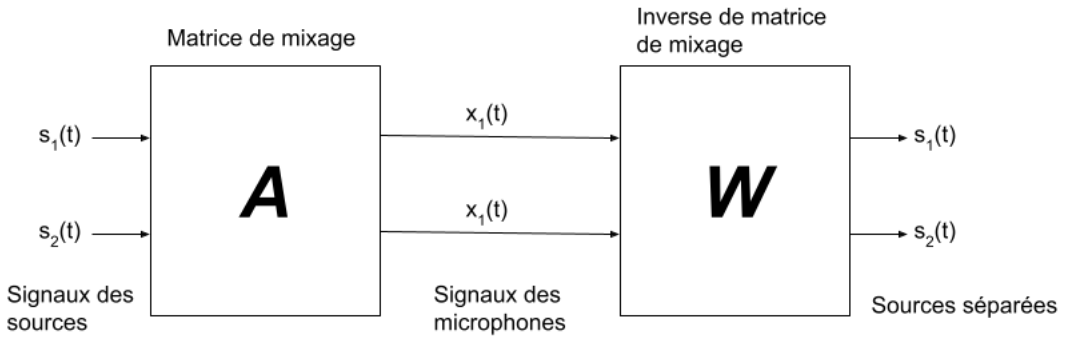

Figure 2.2 Diagramme de séparation aveugle des sources.

Plusieurs approches ont été développées pour résoudre le problème de séparation aveugle de sources. L'approche la plus connue est l'Analyse en Composantes Indépendantes (ACI) qui a été introduite par Hérault et Ans [5]. Plusieurs variétés de l'ACI ont vu le jour. La plus populaire est l'algorithme FastICA de Hyvärinen [6]. La méthode ACI est efficace et précise pour séparer des superpositions de sources indépendantes si les superpositions sont de simples combinaisons linéaires des sources. Cependant, l'ACI est une méthode qui a été modélisée pour traiter des signaux qui n'ont pas été enregistrés par des microphones distants. L'existence d'une distance entre les microphones crée des délais pendant la réception des signaux sonores. La présence de délais crée une limitation pour le modèle formulé par l'ACI et rend la méthode non utilisable dans un cas réel. De plus l'algorithme de l'ACI est plus performant si le nombre des signaux observés est supérieur aux nombre des sources, i.e si le nombre de microphones est plus grand que le nombre des sources de parole. Cette limitation cause problème dans le cas réel car on a un nombre fixe de microphones, deux dans le cas de notre projet.

Dans notre projet de recherche, l'utilisation de l'appareil de reconnaissance vocale de notre partenaire industriel se fait généralement dans une pièce. Ceci implique l'existence de réverbérations créées par la réflexion des ondes sonores des sources de paroles avec les murs de la pièce. Cette réverbération est un signal parasite qui vient rendre la séparation des sources encore plus compliquée. Dans notre projet, on suppose qu'il n'y a pas réverbération.

2.3 La séparation de sources dans le cas où m = 2 et n = 2

Pour une matrice A de dimensions $m \times n$, on peut distinguer trois cas :

— $m = n$, si le nombre de sources est égal au nombre de microphones.

— $m > n$, si le nombre de microphones est plus grand que le nombre de sources.

— $m < n$, si le nombre de sources est plus grand que le nombre de microphones.

Dans notre projet, l'appareil de notre partenaire possède deux microphones et nous supposons qu'on a une source de parole et une source de bruit. Nous considérons alors seulement le cas où $m = n = 2$.

Dans notre projet de recherche, il existe une petite distance entre les microphones de l'appareil. Cette distance crée des délais lors de la réception des ondes sonores des sources entre les deux microphones. Le modèle (2.1) peut alors être reformulé sous la forme équivalente

$$\begin{cases} x(n) = A(n) + B(n), \\ y(n) = \alpha \ A(n+i) + \beta \ B(n+j), \end{cases} \tag{2.7}$$

où $x(n)$ et $y(n)$ sont les signaux enregistrés par les deux microphones, $\alpha > 0$, $\beta > 0$ sont des constantes qui dépendent de la distance entre les microphones et les source, $A(n) \in \mathbb{R}$ et $B(n) \in \mathbb{R}$ sont les sources inconnues reçues par le premier microphone, et $(i, j) \in \mathbb{Z}^2$ sont les délais respectifs des sources de parole et source de bruit. Les variables $A(n)$ et $B(n)$ sont supposées avoir une espérance nulle $E\{A(n)\} = 0$, $E\{B(n)\} = 0$ pour tout n. Ces deux variables aléatoires sont aussi supposées être indépendantes dans le sens probabiliste.
On supposera que le signal $A(n)$ a des caractéristiques statistiques semblables à celles d'un signal de parole. Le signal $B(n)$ a une structure statistique inconnue. En tenant compte de la forme géométrique de l'appareil, nous pouvons réduire l'intervalle de variation des paramètres de mixage et des délais.
En se basant sur l'architecture de l'appareil de notre partenaire Fluent.ai, on peut supposer que la distance entre les deux microphones de l'appareil ne dépasse pas les 10 cm. Fluent.ai utilise des signaux avec une fréquence d'échantillonnage de 16 kHz. Sachant que la vitesse du son est de l'ordre de 343 m/s, on peut déduire que $i \leq 4.66$ et $j \leq 4.66$. Dans la suite, on

supposera que i et j sont dans l'intervalle $[-5, 5]$. Si la distance entre la source et l'appareil est beaucoup plus grande que 10 cm, alors on peut supposer que $\alpha \approx 1$ et $\beta \approx 1$. On suppose aussi que la source de parole $A(n)$ n'est pas constamment active et qu'il existe plusieurs périodes calmes pendant lesquelles le locuteur ne parle pas. Pendant ces périodes calmes, on peut supposer que $A(n) = 0$.

Dans notre projet de recherche, nous étudions le modèle sans réverbérations. Si on tenait compte des réverbérations alors le modèle prendrait plutôt la forme

$$\begin{cases} x(n) = \sum_{k=0}^{K} h_1(k)A(n-k) + \sum_{k=0}^{K} h_2(k)B(n-k), \\ y(n) = \sum_{k=0}^{K} h_3(k)A(n-k) + \sum_{k=0}^{K} h_4(k)B(n-k) \end{cases} \tag{2.8}$$

où les filtres $h_i(k) > 0$, $i \in \{1, 2, 3, 4\}$ sont inconnus et $K >> 1$. Sachant que les microphones sont proches l'un de l'autre, on peut déduire que $h_1 \approx h_3$ et $h_2 \approx h_4$.

En général, les distances entre les microphones et les sources sont légèrement différentes. En absence de réverbération, si on suppose que les microphones sont très proches l'un de l'autre, alors les délais satisfont $i = 0$ et $j = 0$ et le système d'équation (2.7) devient

$$\begin{cases} x(n) = A(n) + B(n), \\ y(n) = \alpha \ A(n) + \beta \ B(n). \end{cases} \tag{2.9}$$

CHAPITRE 3 REVUE DE LITTÉRATURE

L'ACI utilise l'indépendance des sources sonores pour faire la séparation des superpositions de signaux. Cette séparation se fait grâce à deux hypothèses : la non-gaussianité des sources et l'absence d'information mutuelle entre les sources.

3.1 L'analyse en composantes indépendantes

3.1.1 Introduction

Plusieurs approches de l'analyse en composantes indépendantes ont été développées pour la séparation des sources. Toutes les approches utilisent la même représentation du problème. On considère des signaux provenant de sources physiques. Plusieurs types de signaux peuvent être traités par l'ACI :

- Signaux de paroles enregistrés par des microphones ;
- Signaux enregistrés par des capteurs (e.g EEG) ;
- Ondes radios enregistrées par des récepteurs ;
- Images enregistrées par des capteurs photo

Dans le contexte de la séparation de sources, on assume que les capteurs sont placées dans des positions différentes de la pièce, ceci implique l'existence de coefficients de mixage différents pour chaque capteur. D'après Chien [7], chaque coefficient decrit la corrélation spatiale qui peut exister entre la source et le capteur récepteur. On note $x(t)$ et $y(t)$, les amplitudes de deux signaux enregistrés respectivement par les microphones 1 et 2 à chaque instant t. On note les deux sources inconnues $s_1(t)$ et $s_2(t)$ correspondant à une source de parole et une source de bruit. On peut écrire les signaux $x(t)$ et $y(t)$ comme étant la somme pondérée des sources inconnues $s_1(t)$ et $s_2(t)$

$$\begin{cases} x(t) = a_{11}s_1(t) + a_{12}s_2(t), \\ y(t) = a_{21}s_1(t) + a_{22}s_2(t). \end{cases} \tag{3.1}$$

Les coefficients de mixage a_{ij} sont constants et inconnus. Les a_{ij} forment la matrice de mixage A. La méthode ACI suppose que A est inversible. L'inverse de A sert à reconstruire les sources $s_1(t)$ et $s_2(t)$ si

$$W := A^{-1} = \begin{pmatrix} W_{1,1} & W_{1,2} \\ W_{2,1} & W_{2,2} \end{pmatrix}, \tag{3.2}$$

alors

$$\begin{cases} s_1(t) = w_{11}\ x(t) + w_{12}\ y(t), \\ s_2(t) = w_{21}\ x(t) + w_{22}\ y(t). \end{cases} \tag{3.3}$$

L'analyse en composantes indépendantes est une méthode statistique qui utilise les observations $x(t)$ et $y(t)$ afin d'estimer la matrice inverse W et reconstruire les sources.

3.1.2 Indépendance des sources

Si deux sources s_1 et s_2 sont considérées comme deux variables aléatoires différentes, alors l'indépendance statistique entre ces deux variables aléatoires implique que

$$p(s_1, s_2) = p_{s_1}(s_1)p_{s_2}(s_2), \tag{3.4}$$

où $p_{s_1,s_2}(s_1, s_2)$ est la densité conjointe de s_1 et s_2, et $p_{s_1}(s_1)$, $p_{s_2}(s_2)$ sont les probabilités marginales de variables s_1 et s_2.

3.1.3 Fonction coût

Afin de mesurer l'indépendance des sources, l'ACI utilise des fonctions indicatrices d'indépendances appelées fonctions coût. La méthode ACI cherche à minimiser ou maximiser la fonction coût, dépendamment de la fonction utilisée, à l'aide d'un algorithme d'optimisation. Plusieurs fonctions coût existent et le choix de cette dernière influe sur la variance asymptotique et la robustesse de l'ACI. Plusieurs fonctions coût ont été utilisées pour les différentes approches de l'ACI.

Mesure de la non-gaussiannité

Plusieurs fonctions coût existent pour mesurer la non-gaussiannité des signaux. Les plus utilisées sont :

— Coefficient d'aplatissement :
Le coefficient d'aplatissement est une mesure statistique qui décrit le degré d'aplatissement d'une distribution. Elle est définie par

$$\text{kurtose}(s_1) = E\left\{s_1^4\right\} - 3E\left\{s_1^2\right\}, \tag{3.5}$$

où $E\{.\}$ désigne l'espérance mathématique du signal s_1.
Si $E\{s_1\} = 0$ et $E\{s_1^2\} = 1$, alors

$$\text{kurtose}(s_1) = E\left\{s_1^4\right\} - 3. \tag{3.6}$$

Pour une variable gaussienne s_1, le coefficient d'aplatissement est nulle. Une variable aléatoire non-gaussienne a un coefficient d'aplatissement non nulle. On peut mesurer la non-gaussianité d'une variable aléatoire en calculant la valeur absolue ou le carré de son coefficient d'aplatissement.

— Néguentropie
La néguentropie est aussi une mesure de la non-gaussiannité d'une variable aléatoire souvent appelée entropie négative. Hyvärinen [8] définit la néguentropie comme la différence d'entropie entre une variable aléatoire et la variable aléatoire gaussienne qui a la même matrice de covariance. La néguentropie peut être calculée comme suit :

$$J(s_1) = H(s_{gauss}) - H(s_1), \tag{3.7}$$

où s_{gauss} la variable gaussienne qui a la même matrice de covariance que s_1 et $H(s_{gauss})$ l'entropie de s_{gauss} avec

$$H(s_{gauss}) = -\sum_{i=1}^{N} p(i)\ln(p(i)), \tag{3.8}$$

où $p(i) > 0$ sont les distributions de probabilités des $s(i)$.

L'information mutuelle

L'information mutuelle est une mesure très utilisée dans le domaine du traitement des signaux pour mesurer la dépendance statistiques entre des variables aléatoires. L'information mutuelle mesure l'information partagée par deux variables alátoires s_1 et s_2 et peut être utilisée comme une fonction coût. Hyvärinen [8] définit l'information mutuelle entre s_1 et s_2 comme

$$I(s_1, s_2) = \sum_{i=1}^{2} H(s_i) - H(s_1, s_2). \tag{3.9}$$

3.1.4 Les Algorithmes de l'ACI

Plusieurs algorithmes ont été développés pour résoudre le problème de séparation de sources indépendantes. On peut séparer ces algorithmes en deux catégories :

— Algorithmes qui mesurent l'indépendance des sources en minimisant l'information mutuelle.
— Algorithmes qui mesurent l'indépendance des sources en maximisant la non-gaussiannité.

Algorithme FastICA de Hyvärinen [9]

L'algorithme FastICA d'Hyvärinen [9] est l'approche la plus connue pour résoudre le problème de séparation de sources aveugles. C'est une méthode connue pour la rapidité de sa convergence et sa précision. Elle est très utilisée pour séparer les signaux enregistrés par les capteurs d'électroencéphalographie qui enregistrent l'activité électrique du cerveau. L'ACI permet d'enlever les artéfacts pour améliorer l'analyse de l'activité cérébrale. La méthode FastICA se base sur la maximisation de la non-gaussiannité des composantes. Cette méthode utilise le blanchiment des données comme prétraitement et la descente du gradient stochastique comme méthode d'optimisation pour maximiser la fonction coût. FastICA s'inspire des réseaux de neurones et utilise un apprentissage neural en ligne. FastICA est une méthode itérative de mise à jour à point fixe qui estime la matrice des poids w qui maximise la non-gaussiannité des composantes. Pour estimer les poids w, Hyvärinen [8] utilise la forme suivante :

$$w(i) = E\left\{xg(w(i-1)^T x)\right\} - E\left\{g'(w(i-1)^T x)\right\} w(i-1) \tag{3.10}$$

où y est le signal de sortie du réseau de neurones, $g(.)$ est une fonction d'activation et w la matrice de poids mise à jour à chaque itération i. Nous avons testé l'algorithme FastICA en utilisant trois signaux superposés à partir de trois signaux différents. Il y a trois signaux sources, dont un signal sinusoïdal, un signal en créneaux et une onde triangulaire. Les trois observations ont été créées en utilisant une matrice de mixage $M = \begin{pmatrix} 1 & 1 & 1 \\ 0.5 & 2 & 1 \\ 1.5 & 1 & 2 \end{pmatrix}$.

Dans la figure (3.1), le premier graphique montre trois observations créées à partir des sources. Le deuxième graphique montre les signaux des sources, le signal sinusoïdal est coloré en rouge, le signal en créneaux est coloré en bleu et l'onde triangulaire est colorée en vert. La figure (3.1) montre que les reconstructions ont des couleurs différentes des sources. La reconstruction du signal sinusoïdal est colorée en bleu, le signal en créneaux est coloré en vert et l'onde

triangulaire est colorée en rouge. On remarque que l'algorithme FastICA réussit à séparer les trois sources. Cependant, les couleurs des reconstructions sont différentes des couleurs des sources car la séparation par FastICA s'effectue d'une manière aveugle et il n'est pas possible de reconnaître l'ordre des sources. On remarque aussi que le signe de l'onde triangulaire a été inversé lors de la reconstruction. Dans un contexte de reconnaissance de la parole, FastICA ne permet pas de reconnaître la source de parole.

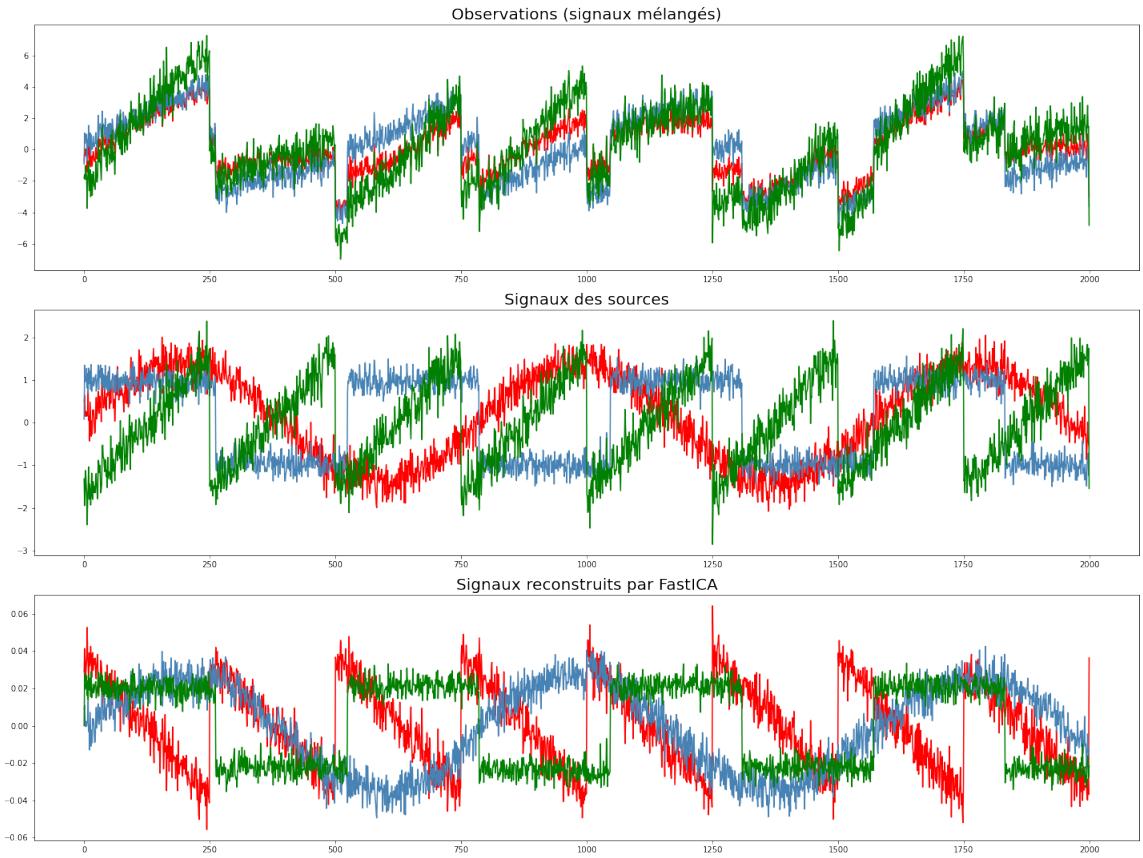

Figure 3.1 Résultats obtenus avec FastICA.

(L'unité de temps est la période d'échantillonnage)

Algorithme InfoMax de Bell et Sejnowski [10]

Cet algorithme est une approche itérative d'apprentissage développée par Bell et Sejnowski [10]. Cette variante de l'ACI s'inspire de l'optimisation utilisée pour les réseaux de neurones artificielles. Bell et Sejnowski [10] décrivent Infomax comme une méthode adaptative qui se base sur la maximisation de l'information apprise par un réseau de neurones. L'idée est de maximiser l'entropie jointe $H(y)$ des signaux résultants de la séparation. Cette séparation utilise la minimisation de l'information mutuelle entre les composantes indépendantes. Les poids du réseau de neurones artificiel sont mis à jour d'une manière itérative comme décrit par Hyvärinen [6]

$$w_{i+1} = w_i + \mu[I - 2g(y_i y_i^T I)]w_i, \tag{3.11}$$

avec i l'indice de l'itération, μ le taux d'apprentissage, y le signal de sortie du réseau de neurones et $g(.)$ une fonction d'activation non linéaire. Il existe différentes fonctions d'activations utilisées lors de l'apprentissage du réseau de neurones. La fonction d'activation du réseau de neurones utilisée par cette approche est la fonction logistique. La fonction logistique est une fonction non linéaire calculée par

$$g(y) = \frac{1}{1 + e^{-y}}, \tag{3.12}$$

On utilise les signaux de l'expérience précédente pour tester l'approche Infomax de Bell et Sejnowski [10]. La figure (3.2) montre les résultats de séparation obtenus par l'algorithme Infomax. Dans la figure (3.2), le premier graphique montre trois observations créées à partir des sources. Le deuxième graphique montre les signaux des sources, le signal sinusoïdal est coloré en rouge, le signal en créneaux est coloré en bleu et l'onde triangulaire est colorée en vert. Le troisième graphique de la figure (3.2) montre les reconstructions retrouvées par Infomax. On remarque qu'on ne réussit pas à retrouver des reconstructions similaires aux sources. Le résultat retrouvé par FastICA semble être bien meilleur que Infomax.

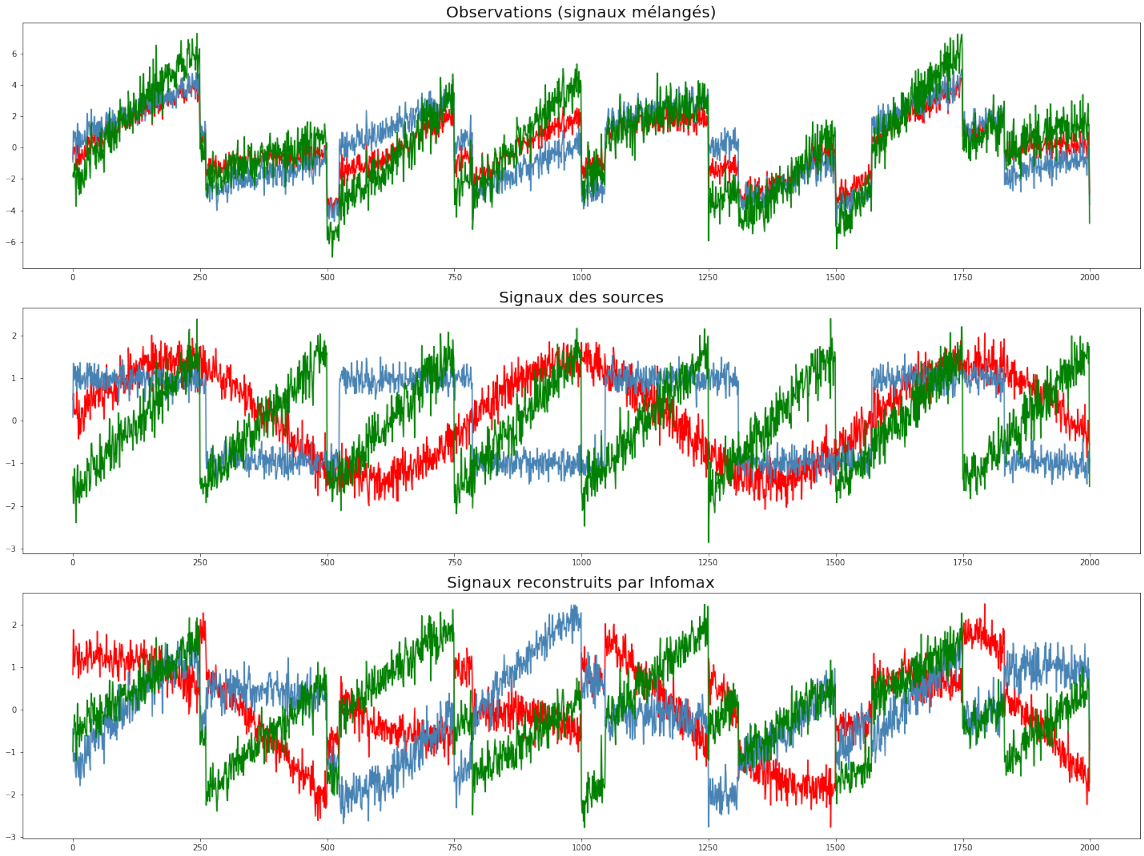

Figure 3.2 Résultats avec Infomax.

(L'unité de temps est la période d'échantillonnage)

Analyse en composantes indépendantes algébrique de Yamaguchi et Itoh [11]

Il existe une approche algébrique de l'analyse en composantes indépendantes qui permet d'extraire la matrice de mixage et de reconstruire directement les signaux sources. L'estimation se fait à partir de moments d'ordre quatre et conduit à des équations algébriques. Cette méthode non itérative développée par Yamaguchi et Itoh [11] utilisent les moments statistiques de quatrième ordre pour estimer les coefficients de mixage.
Pour deux signaux enregistrés x_1 et x_2 on a

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 & \alpha \\ \beta & 1 \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \end{pmatrix}, \tag{3.13}$$

où α et β sont deux paramètres de mixage. On peut reconstruire les signaux sources s_1 et s_2 en inversant la matrice de mixage. Yamaguchi et Itoh [11] proposent une technique pour estimer les paramètres inconnus α et β en utilisant seulement x_1 et x_2. En partant de l'idée

que s_1 et s_2 sont indépendants on peut dire que

$$E\{s_1 s_2\} = E\{s_1\} E\{s_2\}, \tag{3.14}$$

et

$$E\left\{s_1^3 s_2\right\} = E\left\{s_1^3\right\} E\{s_2\}. \tag{3.15}$$

En combinant les équations (3.13) et (3.14) on peut alors déduire que

$$\beta = \frac{\alpha C_2 - C_3}{\alpha C_3 - C_1}, \tag{3.16}$$

où

$$C_1 = E\left\{x_1^2\right\} - E\{x_1\}^2, \tag{3.17}$$

$$C_2 = E\left\{x_2^2\right\} - E\{x_2\}^2, \tag{3.18}$$

$$C_3 = E\{x_1 x_2\} - E\{x_1\} E\{x_2\}, \tag{3.19}$$

À partir des équations (3.13) et (3.15) on peut aussi dériver

$$-C_4\beta + C_5 + 3C_6\alpha\beta - 3C_7\alpha - 3C_8\alpha^2\beta - 3C_9\alpha^2 + C_{10}\alpha^{23}\beta - C_{11}\alpha^3 = 0, \tag{3.20}$$

où

$$C_4 = E\left\{x_1^4\right\} - E\left\{x_1^3\right\} E\{x_1\}, \tag{3.21}$$

$$C_5 = E\left\{x_1^3 x_2\right\} - E\left\{x_1^3\right\} E\{x_2\}, \tag{3.22}$$

$$C_6 = E\left\{x_1^3 x_2\right\} - E\left\{x_1^2 x_2\right\} E\{x_1\}, \tag{3.23}$$

$$C_7 = E\left\{x_1^2 x_2^2\right\} - E\left\{x_1^2 x_2\right\} E\{x_2\}, \tag{3.24}$$

$$C_8 = E\left\{x_1^2 x_2^2\right\} - E\left\{x_1 x_2^2\right\} E\{x_1\}, \tag{3.25}$$

$$C_9 = E\left\{x_1 x_2^3\right\} - E\left\{x_1 x_2^3\right\} E\{x_2\}, \tag{3.26}$$

$$C_{10} = E\left\{x_1 x_2^3\right\} - E\{x_1\} E\left\{x_2^3\right\}, \tag{3.27}$$

$$C_{11} = E\left\{x_2^4\right\} - E\left\{x_2^3\right\} E\{x_2\}, \tag{3.28}$$

Par la suite on peut simplement éliminer β de l'équation (3.16) en utilisant (3.20) de manière à retrouver

$$(C_2C_{10}-C_{11}C_3)\alpha^4+(3C_9C_3-3C_8C_2-C_3C_{10}+C_1C_{11})\alpha^3+(3C_6C_2+3C_8C_3-3C_9C_1-3C_7C_3)\alpha^2 + (C_5C_3+3C_7C_1-3C_6C_3-C_2C_4\alpha+C_3C_4-C_1C_5=0. \quad \text{- (3.29)}$$

Afin de résoudre l'équation de quatrième degré (3.28) et estimer α et β, Yamaguchi et Itoh [11] utilisent la méthode Ferrari pour résoudre une équation de quatrième degré. Cependant, cette méthode souffre du fait que les estimateurs des moments d'ordre quatre ont une variance élevée.

3.2 Estimation de délais

Dans le cadre de notre projet de recherche, la distance qui existe entre les microphones implique l'existence de délais à la réception des ondes sonores par les capteurs. Pour cela, nous avons investigué quelques approches afin d'estimer le délai entre les signaux lors de leur réception, dont une méthode basée sur la corrélation croisée entre les signaux et une approche basée sur l'estimation d'un filtre adaptatif utilisant les moindres carrés.

3.2.1 Corrélation croisée pour l'estimation de délais

L'approche utilisant la corrélation croisée entre les signaux est la méthode la plus simple pour estimer un délai existant entre deux signaux. On considère deux signaux s_1 et s_2 définis par

$$s_1(t) = A(t), \tag{3.30}$$

et

$$s_2(t) = \alpha \ A(t-i), \tag{3.31}$$

où $A(t)$ correspond aux mesures enregistrées par un capteur à chaque instant t et i est un délai, et α est un coefficient d'atténuation lié à la propagation de l'onde sonore entre la source et le de microphone. Dans le cas d'un délai i constant et si $A(t)$ est un processus stochastique stationnaire, on peut exprimer la corrélation croisée entre s_1 et s_2 comme le montre Marmaroli et al. [12]

$$R_{s_1s_2}[t] = E\left\{s_1[n]s_2[n+t]\right\}. \tag{3.32}$$

La corrélation croisée (3.32) atteint son maximum pour le délai qui met les deux signaux s_1 et s_2 en phase. On peut alors dériver l'estimation du délai i comme l'argument qui maximise la fonction de corrélation croisée entre les signaux s_1 et s_2 avec

$$\hat{i} \in \arg\max_t \hat{R}_{s_1 s_2}[t], \tag{3.33}$$

avec $t \in [-N, N]$ et N le délai maximum qu'on pourrait observer et qui dépend de la distance entre les deux capteurs utilisés. Pour des signaux données de longueur L on peut estimer $R_{s_1 s_2}[t]$ avec

$$R_{s_1 s_2}[t] = \frac{1}{N + 1 - |t|} \sum_{n=\max(0,-t)}^{\min(N-t,N)} s_1[n] s_2[n+t]. \tag{3.34}$$

3.2.2 Méthode du filtre adaptatif des moindres carrés

La méthode du filtre adaptatif basé sur les moindres carrés a été développée par Reed et al. [13]. Cette méthode utilise une fonction de mise à jour pour chaque itération t

$$g(t+1) = g(t) + \beta e(t) s_1(t), \tag{3.35}$$

avec $g(k)$ un filtre à réponse impulsionnelle finie et β le coefficient d'adaptation. Pour estimer le délai i entre deux signaux s_1 et s_2, on calcule l'erreur entre le signal $x_1(t)$ et un signal $y(t)$ avec

$$y(t) = g^T(t) s_1(t), \tag{3.36}$$

et l'erreur est calculée comme suit :

$$e[t] = s_2(t) - y(t). \tag{3.37}$$

Pour estimer le délai i entre les signaux x_1 et x_2, on minimise l'erreur des moindres carrées entre s_1 et s_2 après avoir appliqué le filtre $g(t)$

$$\hat{i} \in \arg\min_i \text{MSE}(s_1(i) - g^T s_2(i)). \tag{3.38}$$

3.3 Détection de l'activité vocale

Une des composantes de la méthode développée durant ce projet de recherche vise à retrouver les périodes calmes quand le locuteur de parle pas. Nous avons investigué quelques méthodes existantes pour détecter automatiquement les périodes calmes d'un enregistrement d'un microphone. La détection des périodes calmes est une partie importante de la méthode car elle permet d'estimer le délai lié à la source de bruit $B(n)$.
La détection de l'activité vocale (*Voice Activity Detection (VAD)*) est un sujet de recherche très populaire. Le VAD vise à classifier un morceau d'un signal comme un morceau actif ou calme. Cette technique est très utilisée dans le domaine de la reconnaissance vocale. Plusieurs approches de VAD ont été développées pour déterminer les périodes calmes. On peut diviser ces méthodes en deux catégories : les approches utilisant l'apprentissage automatique supervisé et les approches utilisant l'apprentissage automatique non supervisé. Les méthodes les plus populaires sont celles qui utilisent la machine à vecteur de support pour faire la classification des périodes calmes en utilisant l'énergie du signal et les coefficients MFCC.

3.3.1 Machine à vecteur de support pour la classification

La machine à vecteur de support (*Support Vector Machine (SVM)*) est une des méthodes d'apprentissage statistique supervisé utilisées pour faire de la classification. Dans le domaine de la VAD, le SVM est utilisé pour classifier les périodes calmes et actives. Il est possible d'entraîner un SVM sur un ensemble de données d'entraînement afin qu'il apprenne à segmenter l'enregistrement d'un microphone en périodes actives et périodes calmes. A partir d'enregistrements vocaux, il est possible de calculer plusieurs coefficients à partir des signaux pour construire un ensemble de données étiquetées. Ces données sont utilisées pour entraîner le SVM à faire la classification.

Machine à vecteur de support binaire

La machine à vecteur de support binaire est un classificateur binaire qui retourne une décision entre deux classes. Ces deux classes sont séparées par un hyperplan estimé pendant la phase d'entraînement du SVM. Pour apprendre au SVM à prendre une décision, on l'entraîne sur des données avec deux étiquettes différentes : une étiquette $y_l = +1$ associée à un morceau actif du signal, ou une étiquette $y_l = -1$ associée à un morceau calme. Afin de calculer une prédiction, le SVM utilise la fonction f comme définie par Evgeniou et Pontil [14] pour calculer la prédiction

$$f(x) = w^T x + b. \tag{3.39}$$

Au cours de la phase d'entraînement le SVM estime un hyperplan qui sépare les données en deux classes avec une marge formée par l'ensemble des points contenus par les vecteurs de supports comme le montre la figure (3.3)

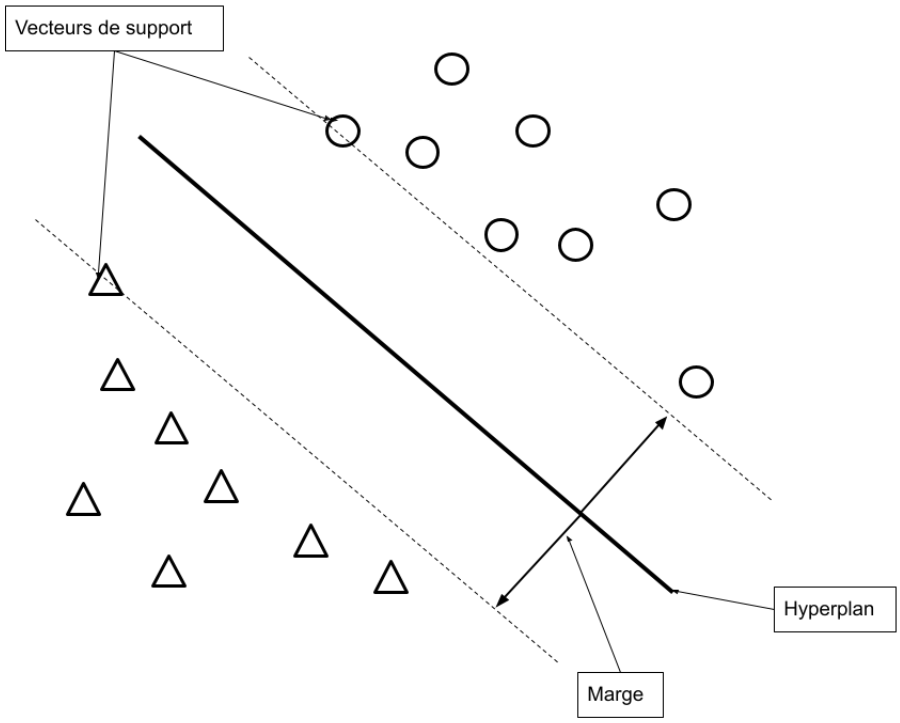

Figure 3.3 Fonctionnement d'une machine à vecteurs de support binaire.

Pour les deux classes $y = +1$ et $y = -1$ on retrouve

$$w^T x + b \geq 1 \quad \text{pour} \quad y = +1, \tag{3.40}$$

$$w^T x + b \leq 1 \quad \text{pour} \quad y = -1. \tag{3.41}$$

On peut combiner les deux inégalitées (3.40) et (3.41) pour obtenir

$$y(w^T x + b) - 1 \geq 0 \quad \text{pour} \quad y \in (-1, 1) \tag{3.42}$$

Afin d'estimer les paramètres w de l'hyperplan on mesure la distance D entre chaque point x_i et la marge en calculant

$$D(x, y) = \frac{y(w^T x + b)}{||w||_2}. \tag{3.43}$$

avec $||(.)||_2$ la norme euclidienne. Pour les points se retrouvant exactement sur la marge de l'hyperplan la distance $D = \frac{1}{||w||_{L2}}$. L'idée est de retrouver l'hyperplan qui va maximiser la distance entre les points de l'ensemble d'entraînement et la marge de l'hyperplan. On assume que la marge totale de l'hyperplan est calculée ainsi

$$m_+ + m_- = \frac{2}{||w||_{L2}}, \tag{3.44}$$

où $m_\pm$ est la marge de chaque côté de l'hyperplan du SVM.
On veut maximiser la largeur de la marge qui sépare les points et l'hyperplan. On minimise le dénominateur $||w||_{L2}$ sous la contrainte

$$y(w^T x + b) - 1 \geq 0. \tag{3.45}$$

On combine les équations (3.44) et (3.45) pour formuler le problème d'optimisation suivant :

$$\rho(w, b, \alpha) = \frac{||w||^2}{2} - \sum_{i=1}^{n} \alpha_i [y(w^T x + b) - 1], \tag{3.46}$$

où α_i sont les multiplicateur lagrangiens. Pour estimer les paramètre de l'hyperplan du SVM, on minimise ρ par rapport à w et b et on maximise ρ par rapport à α.

SVM Binaire utilisant les coefficients MFCC de Kinnunen et al. [15]

Le SVM binaire est une approche populaire pour classifier l'activité vocale de la personne. La méthode développée par Kinnunen et al. [15], utilise les coefficients MFCC pour entraîner un SVM à classifier les signaux en périodes silencieuses ou actives. Ces coefficients sont extraits à partir des signaux utilisés pour l'entraînement du SVM. Les signaux sont alors divisés en petites morceaux, pour chaque morceau Kinnunen et al. [15] calculent les coefficients MFCC et les enregistre comme données d'entraînement pour apprendre au SVM à classifier les morceaux du signal en calmes ou actifs.

Approche utilisant l'énergie du signal et les coefficients MFCC de Dey et al. [16]

Dey et al. [16] ont développé une architecture utilisant l'énergie du signal pour faire une classification binaire des signaux. L'idée est d'utiliser un seuil d'énergie pour séparer les données d'entraînement selon leur énergies. Puis ces données sont utilisées pour entraîner un classificateur pour classifier le signal en signal de parole ou non. La figure (3.4) représente l'architecture utilisée par Dey et al. [16] pour faire la détection d'activité vocale

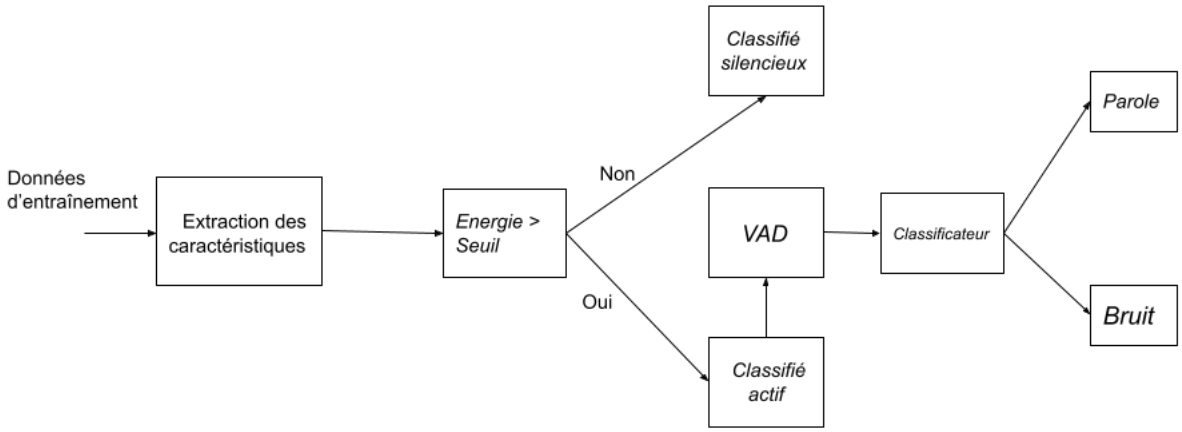

Figure 3.4 Diagramme de l'architecture du système VAD.

Pour entraîner le classificateur, Dey et al. [16] extraient les coefficients MFCC qu'ils calculent à partir des enregistrements d'entraînement. Puis dépendamment de son énergie, le signal est classifié en période silencieuse ou active. Si l'énergie du signal dépasse le seuil alors il est considéré comme une période active. Par la suite, le classificateur est entraîné en utilisant les coefficients MFCC afin de classifier le signal en signal de parole ou non . Pour cette méthode, deux algorithmes de classification sont proposés : un algorithme utilisant le SVM et un algorithme utilisant les réseaux de neurones profonds.

CHAPITRE 4 MÉTHODE ICA/2S/2PM SANS DÉLAIS

4.1 Introduction

Notre partenaire industriel Fluent.ai désire débruiter les signaux sonores reçus par les microphones de son appareil de reconnaissance vocale. Le but de la méthode ICA/2S/2PM est de séparer les signaux reçus en deux composantes indépendantes dont la source de parole $A(n)$ et la source de bruit $B(n)$. On commence d'abord par développer une approche qui ne tient pas compte des délais présents entre les microphones, par la suite on veut généraliser cette approche et l'adapter au cas où il y a présence de délais.

4.2 Formulation du problème ACI

Dans ce projet, nous implémentons l'algorithme de séparation des signaux pour un appareil qui contient deux microphones séparés par une distance qui ne dépasse pas 10 cm. Ceci implique que les délais existants entre les microphones sont très petits. Si on suppose que la fréquence d'échantillonnage utilisée est égale à 16 kHz et que la vitesse du son est égale à 343 m/s alors on peut déduire que le délai entre les microphones est égale à 0.00029 secondes. En utilisant la fréquence d'échantillonnage on déduit que les délais $i < 5$ et $j < 5$ avec i le délai lors de la réception des ondes sonores provenant de la voix locuteur et j le délai lors de la réception des ondes sonores provenants de la source de bruit aux microphones. Dans ce chapitre, nous commençons par le cas où $i = j = 0$. Nous utilisons la notation ICA/2S/2PM pour la méthode comme référence à l'analyse en composantes indépendantes pour deux sources en utilisant les moments à deux points. Cette méthode résout le problème classique de l'ACI en utilisant les corrélations entre deux variables aléatoires.

On commence par formuler notre modèle avec des délais $i = j = 0$. Les mesures $x(n)$ et $y(n)$ des deux microphones sont reliées aux sources $A(n)$ et $B(n)$ par

$$x(n) = A(n) + B(n), \tag{4.1}$$

$$y(n) = \alpha A(n) + \beta B(n), \tag{4.2}$$

où $\alpha > 0$ et $\beta > 0$, sont deux paramètres réels constants et correspondant aux coefficients d'amplification liés aux sources $A(n)$ et $B(n)$. Ces deux paramètres satisfont la condition $\alpha > \beta$ par convention. De plus, dans le cas de l'appareil de reconnaissance vocale utilisé, la distance entre les microphones est si petite qu'on peut supposer que $\alpha \approx 1$ et $\beta \approx 1$.

4.3 Approche des corrélations en deux points

L'approche développée pour résoudre le problème d'ACI utilise les corrélations en deux points avec un décalage temporel k entre deux variables aléatoires. On utilise les équations précédentes à un autre instant $n+k$, où $k>0$ un paramètre ajustable, pour obtenir

$$x(n+k) = A(n+k) + B(n+k), \tag{4.3}$$

$$y(n+k) = \alpha\, A(n+k) + \beta\, B(n+k). \tag{4.4}$$

Les variables aléatoires $A(n)$ et $B(n)$ sont deux sources indépendantes pour chaque échantillon $n \in \mathbb{Z}$. Les sources sonores $\{A(n), n \in \mathbb{Z}\}$ et $\{B(n), n \in \mathbb{Z}\}$ sont supposées être stationnaires. Cela implique aussi que les processus aléatoires $\{x(n), n \in \mathbb{Z}\}$ et $\{y(n), n \in \mathbb{Z}\}$ sont stationnaires. Toutes ces variables aléatoires ont une espérance nulle

$$\begin{aligned} E\{A(n)\} &= 0, \\ E\{B(n)\} &= 0, \\ E\{x(n)\} &= 0, \\ E\{y(n)\} &= 0. \end{aligned} \tag{4.5}$$

On commence par calculer les différents moments qu'on utilisera pour calculer les estimés des paramètres α et β

$$XX := E\left\{x(n)^2\right\} = E\left\{x(n+k)^2\right\}, \tag{4.6}$$

$$YY := E\left\{(n)^2\right\} = E\left\{y(n+k)^2\right\}, \tag{4.7}$$

$$XY := E\{x(n)\, y(n)\}, \tag{4.8}$$

$$XX_* := E\{x(n)\, x(n+k)\}, \tag{4.9}$$

$$YY_* := E\{y(n)\, y(n+k)\}, \tag{4.10}$$

$$XY_* := E\{x(n)\, y(n+k)\}, \tag{4.11}$$

$$AA := E\left\{A(n)^2\right\} = E\left\{A(n+k)^2\right\}, \tag{4.12}$$

$$BB := E\left\{B(n)^2\right\} = E\left\{B(n+k)^2\right\}, \tag{4.13}$$

$$AA_* := E\left\{A(n)\ A(n+k)\right\}, \tag{4.14}$$

$$BB_* := E\left\{B(n)\ B(n+k)\right\}. \tag{4.15}$$

En utilisant les carrés des équations (4.1) on peut alors réécrire XX comme

$$\begin{aligned} XX &= E\left\{x(n)^2\right\}, \\ &= E\left\{(A(n)+B(n))^2\right\}, \\ &= E\left\{A(n)^2 + 2A(n)B(n) + B(n)^2\right\}, \\ &= AA + BB. \end{aligned} \tag{4.16}$$

De même on peut réécrire YY comme

$$\begin{aligned} YY &= E\left\{y(n)^2\right\}, \\ &= E\left\{(\alpha A(n)+\beta B(n))^2\right\}, \\ &= E\left\{\alpha^2 A(n)^2 + 2\alpha(n)B(n) + \beta^2 B(n)^2\right\}, \\ &= \alpha^2 AA + \beta^2 BB. \end{aligned} \tag{4.17}$$

On multiplie les équations en (4.1) pour retrouver XY :

$$\begin{aligned} XY &= E\left\{x(n)y(n)\right\}, \\ &= E\left\{(A(n)+B(n))(\alpha A(n)+\beta B(n))\right\}, \\ &= E\left\{\alpha A(n)^2 + \beta A(n)B(n) + \alpha A(n)B(n) + \beta B(n)^2\right\}, \\ &= \alpha AA + \beta BB. \end{aligned} \tag{4.18}$$

On multiplie les équations (4.1) et (4.3) pour retrouver XX_* :

$$\begin{aligned} XX_* &= E\left\{x(n)x(n+k)\right\}, \\ &= E\left\{(A(n)+B(n))(A(n+k)+B(n+k))\right\}, \\ &= E\left\{A(n)A(n+k) + A(n)B(n+k) + B(n)A(n+k) + B(n)B(n+k)\right\}, \\ &= AA_* + BB_*. \end{aligned} \tag{4.19}$$

De même nous multiplions les équations (4.2) et (4.4) pour calculer YY_* :

$$\begin{aligned} YY_* &= E\left\{y(n)y(n+k)\right\}, \\ &= E\left(\alpha A(n)+\beta B(n))(\alpha A(n+k)+\beta B(n+k))\right\}, \\ &= E\left\{\alpha^2 A(n)A(n+k)+\alpha\beta A(n)B(n+k)+\alpha\beta B(n)A(n+k)+\beta^2 B(n)B(n+k)\right\}, \\ &= \alpha^2 AA_* + \beta^2 BB_*. \end{aligned} \tag{4.20}$$

De même on multiplie (4.1) et (4.4) afin d'obtenir XY_* :

$$\begin{aligned} XY_* &= E\left\{x(n)y(n+k)\right\}, \\ &= E\left(A(n)+B(n))(\alpha A(n+k)+\beta B(n+k))\right\}, \\ &= E\left\{\alpha A(n)A(n+k)+\beta A(n)B(n+k)+\alpha A(n+k)B(n)+\beta B(n)B(n+k)\right\}, \\ &= \alpha\ AA_* + \beta\ BB_*. \end{aligned} \tag{4.21}$$

Les équations calculées précédemment nous permettent de formuler un système d'équations. Résoudre ce système nous permettra d'estimer les paramètres inconnus α et β. On peut alors formuler ce système d'équations comme suit :

$$XX = AA + BB, \tag{4.22}$$

$$YY = \alpha^2 AA + \beta^2 BB, \tag{4.23}$$

$$XY = \alpha AA + \beta BB, \tag{4.24}$$

$$XX_* = AA_* + BB_*, \tag{4.25}$$

$$YY_* = \alpha^2 AA_* + \beta^2 BB_*, \tag{4.26}$$

$$XY_* = \alpha AA_* + \beta BB_*. \tag{4.27}$$

où $\{AA, BB, AA_*, BB_*, \alpha, \beta\}$ sont les six inconnues.

4.4 Résolution du système d'inconnues (4.22)-(4.27)

On résout d'abord les équations (4.22)-(4.23), (4.22)-(4.24), (4.23)-(4.24) afin de calculer les inconnues AA, BB, ceci nous permet d'obtenir trois paires de solutions (AA, BB).
En combinant les équations (4.22)-(4.23) on retrouve la première paire (AA, BB)

$$AA = \frac{YY - \beta^2 XX}{\alpha^2 - \beta^2}, BB = \frac{\alpha^2 XX - YY}{\alpha^2 - \beta^2}. \tag{4.28}$$

De même en utilisant les équations (4.22)-(4.24) on retrouve la deuxième paire (AA, BB)

$$AA = \frac{XY - \beta XX}{\alpha - \beta}, BB = \frac{XY - \alpha XX}{\beta - \alpha}. \tag{4.29}$$

Par la suite on utilise les équations (4.23)-(4.24) on retrouve la troisième paire (AA, BB)

$$AA = \frac{YY - \beta XY}{\alpha^2 - \alpha\beta}, BB = \frac{\alpha XY - YY}{(\alpha - \beta)\beta}. \tag{4.30}$$

La consistance du système d'équations implique que les trois paires de solutions retrouvées pour (AA, BB) soient égales. Ceci implique que

$$\frac{YY - \beta^2 XX}{\alpha^2 - \beta^2} = \frac{XY - \beta XX}{\alpha - \beta}, \frac{YY - \beta^2 XX}{\alpha^2 - \beta^2} = \frac{YY - \beta XY}{\alpha^2 - \alpha\beta}. \tag{4.31}$$

De même, pour BB on obtient

$$\frac{\alpha^2 XX - YY}{\alpha^2 - \beta^2} = \frac{XY - \alpha XX}{\beta - \alpha}, \frac{\alpha^2 XX - YY}{\alpha^2 - \beta^2} = \frac{\alpha XY - YY}{(\alpha - \beta)\beta}. \tag{4.32}$$

Les égalités précédentes sont satisfaites si et seulement si

$$\beta = \frac{YY - \alpha XY}{XY - \alpha XX}. \tag{4.33}$$

De la même manière, grâce aux équations (4.25)-(4.26), (4.25)-(4.27), (4.26)-(4.27) on retrouve trois paires de solutions pour les équations (AA_*, BB_*).

En combinant les équations (4.25)-(4.26) on retrouve la première paire (AA, BB)

$$AA* = \frac{YY_* - \beta^2 XX*}{\alpha^2 - \beta^2}, BB* = \frac{\alpha^2 XX* - YY*}{\alpha^2 - \beta^2}, \tag{4.34}$$

De même en utilisant les équations (4.25)-(4.27) on retrouve la deuxième paire (AA, BB)

$$AA* = \frac{XY_* - \beta XX_*}{\alpha - \beta}, BB* = \frac{XY_* - \alpha XX_*}{\beta - \alpha}, \tag{4.35}$$

Par la suite on utilise les équations (4.26)-(4.27) on retrouve la troisième paire (AA, BB)

$$AA* = \frac{YY_* - \beta XY_*}{\alpha^2 - \alpha\beta}, BB* = \frac{\alpha XY_* - YY_*}{(\alpha - \beta)\beta}. \tag{4.36}$$

De même la consistance du système d'équations implique que les trois paires de solutions retrouvées pour (AA_*, BB_*) soient égales. Ceci implique que

$$\frac{YY_* - \beta^2 XX_*}{\alpha^2 - \beta^2} = \frac{XY_* - \beta XX_*}{\alpha - \beta}, \quad \frac{YY_* - \beta^2 XX_*}{\alpha^2 - \beta^2} = \frac{YY_* - \beta XY_*}{\alpha^2 - \alpha\beta}. \tag{4.37}$$

Ainsi on retrouve que les égalités précédentes sont satisfaites si et seulement si

$$\beta = \frac{YY_* - \alpha XY_*}{XY_* - \alpha XX_*}. \tag{4.38}$$

En utilisant les équations (4.33) et (4.38) on retrouve l'égalité

$$\frac{YY - \alpha XY}{XY - \alpha XX} = \frac{YY_* - \alpha XY_*}{XY_* - \alpha XX_*}. \tag{4.39}$$

La dernière égalité peut être reformulée en une équation de second ordre pour l'inconnue α

$$\alpha^2(XY\ XX_* - XY_*\ XX) - \alpha(YY\ XX_* + XY\ XY_* - XX\ YY_* - XY\ XY_*) + (YY\ XY_* - YY_*\ XY) = 0. \tag{4.40}$$

On peut alors résoudre cette équation de second ordre en calculant δ :

$$\delta := \sqrt{(XX_*\ YY - XX\ YY_*)^2 + 4(XX_*\ XY - XX\ XY*)(XY\ YY_* - YY\ XY_*)}. \tag{4.41}$$

Les solutions de l'équation du second degrée sont donnée par

$$\alpha_\pm = \frac{XX_*\ YY - XX\ YY_* \pm \delta}{2(XY\ XX_* - XX\ XY_*)}, \tag{4.42}$$

Les solutions qui correspondent aux valeurs des inconnues α et β sont

$$\alpha = \frac{XX_*\ YY - XX\ YY_* + \delta}{2(XY\ XX_* - XX\ XY_*)}, \tag{4.43}$$

$$\beta = \frac{XX_* \, YY - XX \, YY_* - \delta}{2(XY \, XX_* - XX \, XY_*)}. \tag{4.44}$$

car $\alpha > \beta$. Pour calculer les estimés de α et β on doit estimer les variables $XX, XX_*, YY, YY_*, XY, XY_*$. Ces variables sont calculées en utilisant les estimateurs suivants :

$$XX = \frac{1}{N} \sum_{n=0}^{N-1} x(n) \, x(n), \tag{4.45}$$

$$YY = \frac{1}{N} \sum_{n=0}^{N-1} y(n) \, y(n), \tag{4.46}$$

$$XX_* = \frac{1}{N} \sum_{n=0}^{N-1-k} x(n) \, x(n+k), \tag{4.47}$$

$$YY_* = \frac{1}{N} \sum_{n=0}^{N-1-k} y(n) \, y(n+k), \tag{4.48}$$

$$XY = \frac{1}{(N-1)} \left[\sum_{n=0}^{N-1} x(n) \, y(n) \right], \tag{4.49}$$

$$XY_* = \frac{1}{2(N-k-1)} \left[\sum_{n=0}^{N-k-1} x(n) \, y(n+k) + \sum_{n=0}^{N-k-1} y(n) \, x(n+k) \right]. \tag{4.50}$$

Le paramètre k ne doit pas être très grand pour éviter de retrouver des corrélations nulles. k doit être plus petit que la longueur de corrélation de $x(n)$ et celle de $y(n)$.

4.5 Reconstruction des sources

Après avoir estimé les paramètres inconnus α et β, on peut écrire la matrice de mixage H suivant le système d'équations (4.1) comme

$$H = \begin{pmatrix} 1 & 1 \\ \alpha & \beta \end{pmatrix}, \tag{4.51}$$

Il suffit alors d'estimer l'inverse de la matrice de mixage H^{-1} afin de reconstruire les sources $A(n)$ et $B(n)$. H^{-1} peut être calculé comme suit :

$$H^{-1} = \frac{1}{\beta - \alpha} \begin{pmatrix} \beta & -1 \\ -\alpha & 1 \end{pmatrix}, \tag{4.52}$$

et les estimés $\hat{A}(n)$ et $\hat{B}(n)$ des sources $A(n)$ et $B(n)$ sont donnés par

$$\begin{bmatrix} \hat{A}(n) \\ \hat{B}(n) \end{bmatrix} = H^{-1} \begin{bmatrix} x(n) \\ y(n) \end{bmatrix}. \tag{4.53}$$

4.6 Test de la méthode ICA/2S/2PM

Afin de vérifier que l'approche ICA/2S/2PM fonctionne, on poursuit notre activité par des tests de vérification effectués avec des signaux enregistrés par les microphones de Fluent.ai. L'idée est de créer des signaux superposés qui correspondent à des signaux enregistrés par deux microphones dans un environnement avec du bruit. On effectue aussi un test de comparaison avec les méthodes FastICA de Hyvärinen [6] et Infomax de Bell et Sejnowski [10] pour voir si la méthode ICA/2S/2PM produit de meilleurs résultats.

4.6.1 Les signaux utilisés

On utilise des signaux enregistrés par les microphones de notre partenaire industriel Fluent.ai. Le premier signal $A(n)$ représente l'enregistrement de la voix d'un homme qui prononce une suite de signaux d'intention : "Alexa let me know the parking location, Alexa let me know the notice, Alexa call the elevator". Le deuxième signal $B(n)$ est le signal d'un bruit rose. Le bruit rose est un bruit aléatoire qui a une densité spectrale inversement proportionnelle à sa fréquence. Dans la figure (4.1) on a représenté les deux signaux qui ont une fréquence d'échantillonnage de 16 kHz et une longueur de 10 secondes correspondant à 160 000 échantillons.

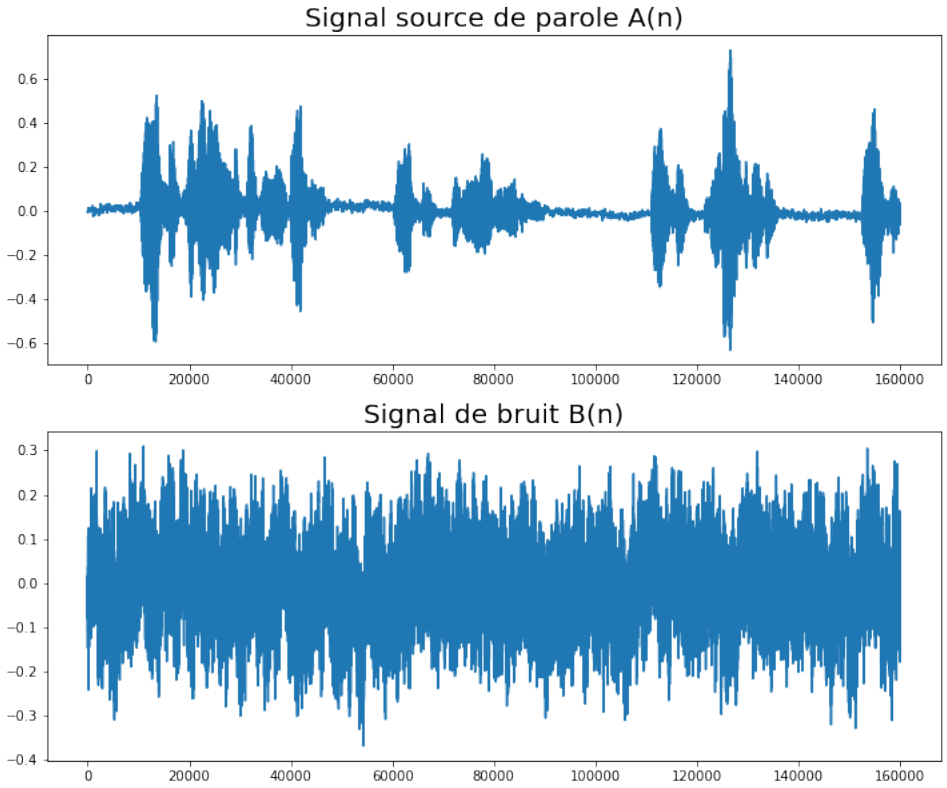

Figure 4.1 Signaux des sources.

(L'unité de temps est la période d'échantillonnage)

Pour créer des signaux superposés qui représentent ce que recevraient les microphones de l'appareil de Fluent.ai dans un environnement contenant du bruit, on utilise des coefficients de mixage $\alpha = 1.2$ et $\beta = 0.8$. En utilisant les signaux des sources $A(n)$ et $B(n)$ et une matrice de mixage H donnée par

$$H = \begin{pmatrix} 1 & 1 \\ 1.2 & 0.8 \end{pmatrix}, \tag{4.54}$$

on peut créer les signaux superposés représentés dans la figure (4.2).

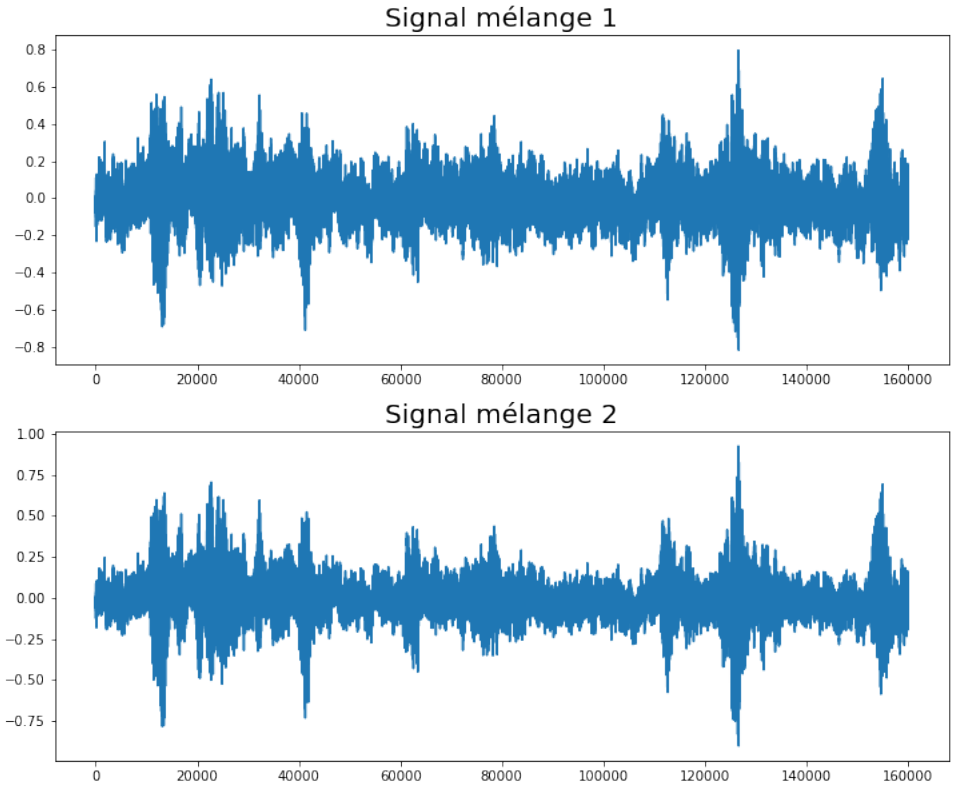

Figure 4.2 Signaux des microphones.

(L'unité de temps est la période d'échantillonnage)

4.6.2 Résultats de la méthode FastICA de Hyvärinen (1999)

On commence par faire un test de séparation en utilisant la méthode FastICA pour comparer les résultats qu'on retrouvera plus tard avec ICA/2S/2PM. En utilisant la méthode FastICA, on retrouve deux reconstructions très similaires au signaux $A(n)$ et $B(n)$ comme le montre la figure (4.3). Afin de bien comparer les résultats obtenus par les trois méthodes utilisées, on calcule une erreur e définie comme le pourcentage de différence d'amplitude entre le signal estimé $\hat{A}(n)$ et la source $A(n)$

$$e = ||\hat{A}(n) - A(n)||. \tag{4.55}$$

On utilise des signaux centrés et réduits pour les sources utilisées et leurs estimés avec

$$S = \frac{s - \mu}{\sigma}. \tag{4.56}$$

Pour le test de la méthode FastICA, on retrouve une erreur $e = 0.00017$ et un temps de calcul $T = 0.1$ seconde.

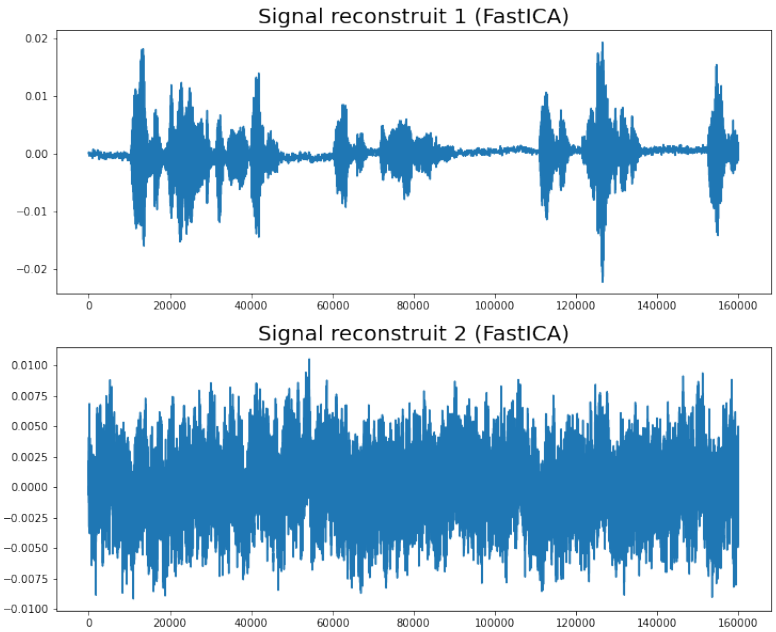

Figure 4.3 Signaux reconstruits par FastICA.

(L'unité de temps est la période d'échantillonnage)

4.6.3 Résultats de la méthode Infomax de Bell et Sejnowski (1995)

Par la suite, on effectue un autre test de séparation en utilisant la méthode Infomax pour comparer les résultats qu'on retrouvera plus tard avec ICA/2S/2PM. En utilisant la méthode Infomax, on retrouve deux reconstructions très similaires au signaux $A(n)$ et $B(n)$ comme le montre la figure (4.4). Pour le test de la méthode Infomax, on retrouve une erreur $e = 2.74e - 05$ et un temps de calcul $T = 4.3$ seconde.

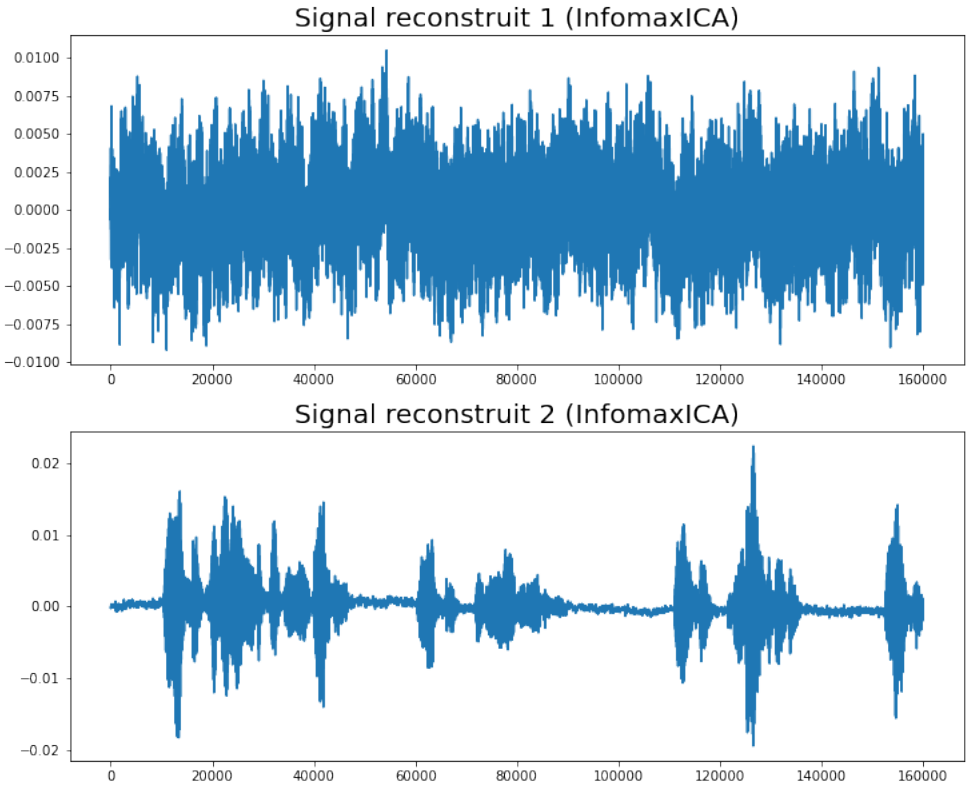

Figure 4.4 Signaux reconstruits par Infomax.

(L'unité de temps est la période d'échantillonnage)

4.6.4 Résultats de la méthode ICA/2S/2PM

En utilisant les mêmes signaux que dans le test précédent avec la méthode FastICA, nous avons effectué un deuxième test en utilisant l'approche ICA/2S/2PM et nous avons retrouvé des estimations exactes pour la matrice de mixage H

$$H_{\text{ICA/2S/2PM}} = \begin{pmatrix} 1 & 1 \\ 1.20 & 0.80 \end{pmatrix}. \tag{4.57}$$

De même on retrouve des reconstructions identiques aux signaux des sources $A(n)$ et $B(n)$ comme le montre la figure (4.5). Les signaux reconstruits ne représentent aucune atténuation ou amplification des amplitudes à l'instar de FastICA et Infomax. On retrouve une erreur $e = 1.95e - 06$ qui est plus faible que les erreurs retrouvées pour les tests de FastICA et

Infomax. Le temps de calcul $T = 0.007$ seconde est très faible en comparaison avec FastICA et Infomax.

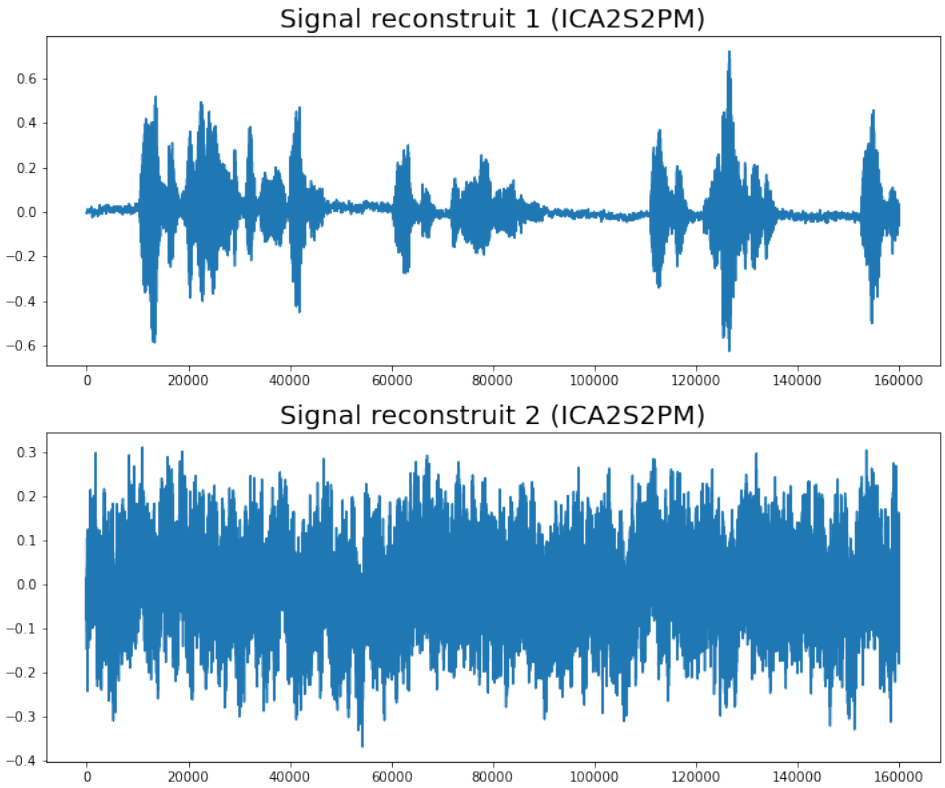

Figure 4.5 Signaux reconstruits par ICA2S2PM.

(L'unité de temps est la période d'échantillonnage)

4.7 Test de la méthode ICA/2S/2PM sur des signaux avec des délais

4.7.1 Test comparatif avec la méthode FastICA en présence de délais

Les méthodes ICA/2S/2PM, FastICA et Infomax n'ont pas été développée pour tenir compte de l'existence de délais lors de la réception des signaux par les microphones. Afin de simuler une situation réelle, on ajoute des délais aux sources lors de la création des mixages. On ajoute un délai $i \in \mathbb{Z}$ lié à la source de parole $A(n)$ et un délai $j \in \mathbb{Z}$ lié à la source de bruit $B(n)$. Pour nos tests, on garde les mêmes paramètres de mixage α et β et on utilise des délais $i = 2$ et $j = -1$ tout en gardant les mêmes signaux des sources $A(n)$ et $B(n)$. On retrouve une erreur $e = 2.45$ plus grande que dans le test précédent. La figure (4.6) montre que les délais ajoutés ont eu un impact sur les résultats de la reconstruction des signaux. Les signaux reconstruits contiennent encore le bruit qu'on a ajouté lors du mixage

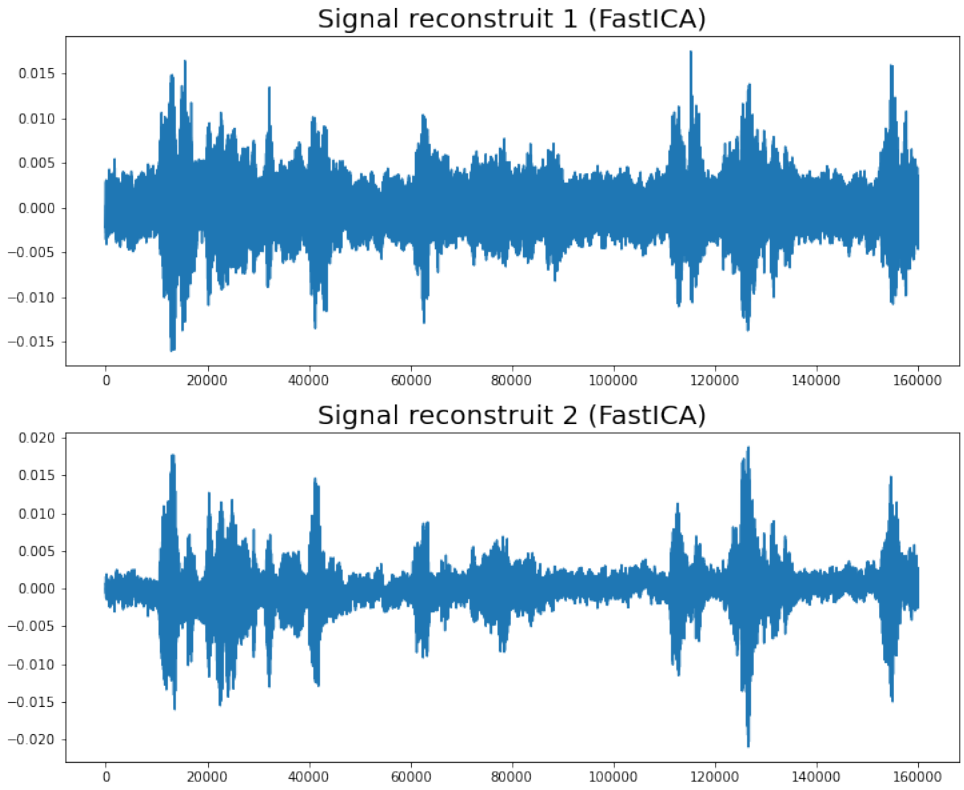

Figure 4.6 Signaux reconstruits par FastICA avec présence de délais.

(L'unité de temps est la période d'échantillonnage)

4.7.2 Test comparatif avec la méthode Infomax en présence de délais

On utilise les mêmes signaux ainsi que les mêmes paramètres pour ce test. En utilisant l'approche de Infomax, on retrouve une erreur $e = 1.81$ plus grande que dans le test avec les signaux sans délais. La figure (4.7) montre que les délais ajoutés ont eu un impact sur les résultats de la reconstruction des signaux. Les signaux reconstruits contiennent encore le bruit qu'on a ajouté lors du mixage

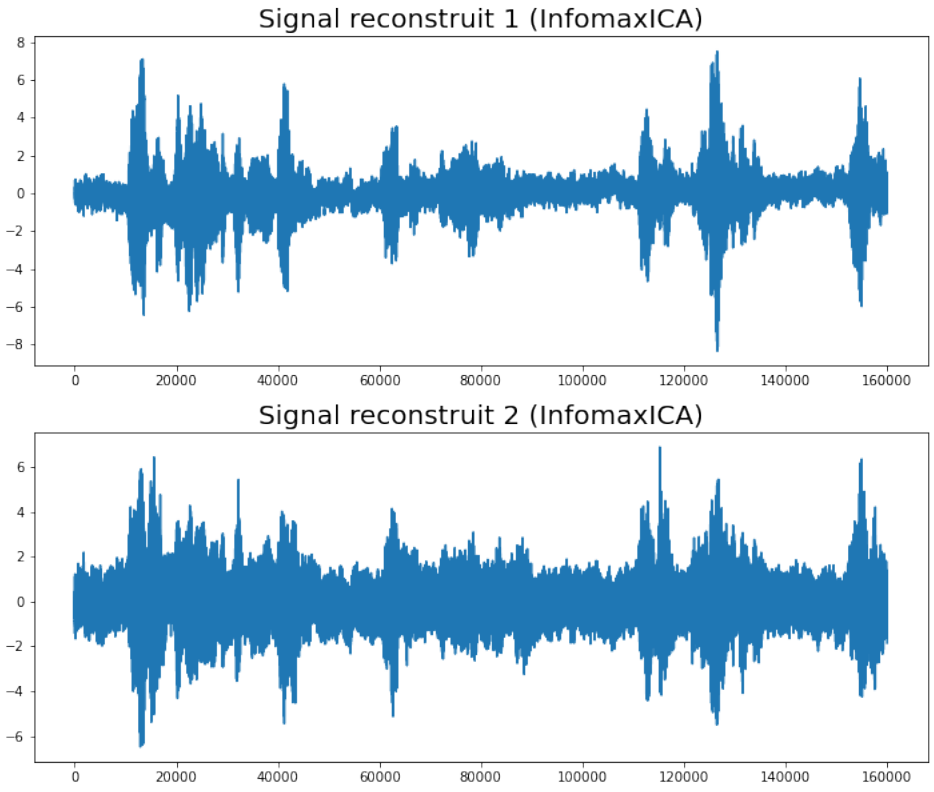

Figure 4.7 Signaux reconstruits par Infomax avec présence de délais.

(L'unité de temps est la période d'échantillonnage)

4.7.3 Test de la méthode ICA/2S/2PM

De même, on utilise les mêmes signaux avec les mêmes paramètres α, β, i et j afin de tester l'effet de la présence de délais lors du mixage sur la méthode ICA/2S/2PM. Pour ce test l'estimation de la matrice de mixage H donne

$$H_{\text{ICA/2S/2PM}} = \begin{pmatrix} 1 & 1 \\ 1.14 & -0.74 \end{pmatrix}. \tag{4.58}$$

Cette estimation n'est pas correcte et on peut constater sur la figure (4.8) que les délais ont beaucoup affecté la reconstruction des signaux. On voit que le bruit reste très présent dans les deux reconstructions, et l'erreur calculée $e = 1,64$ est aussi plus grande que dans le test avec les signaux sans délais.

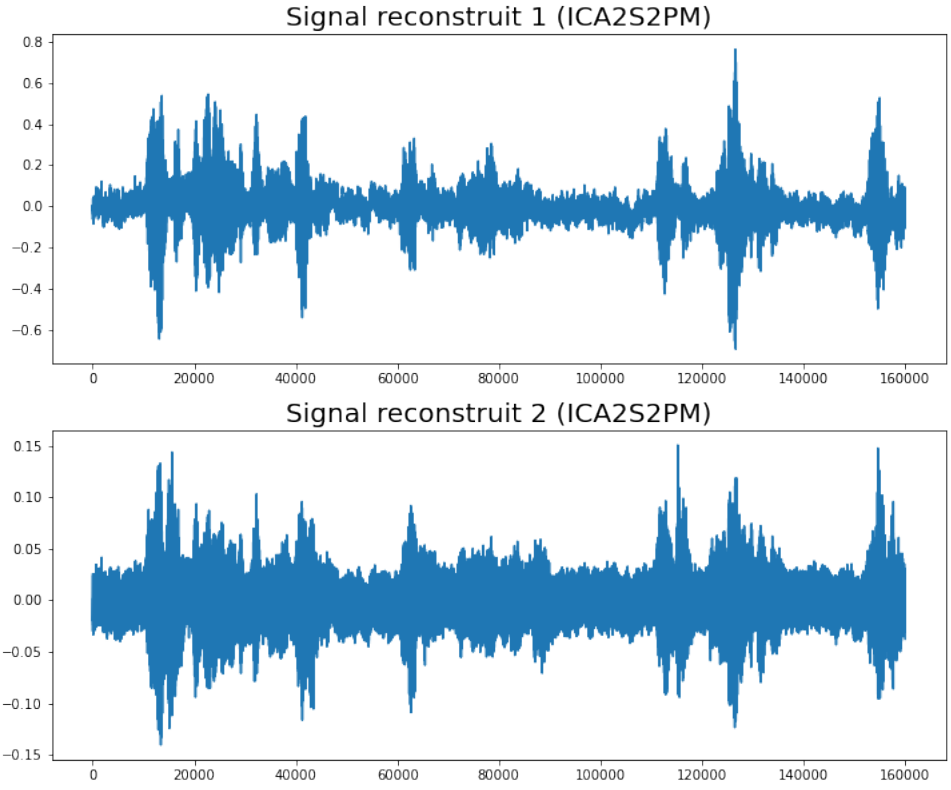

Figure 4.8 Signaux reconstruits par ICA/2S/2PM avec présence de délais.

(L'unité de temps est la période d'échantillonnage)

4.8 Conclusion

La méthode ICA/2S/2PM a été développée pour résoudre le problème ACI sans tenir compte des délais. En analysant les tests précédents, on voit que ICA/2S/2PM a permis de retrouver des estimations exactes pour les paramètres de mixage α et β même quand α est très proche de β. On a testé la méthode en utilisant d'autres structures de signaux et on a remarqué que la performance de séparation n'est pas sensible à la nature des signaux. L'erreur retrouvée avec ICA/2S/2PM et le temps de calcul sont plus faibles que celles retrouvées avec les deux autres approches d'ACI. Les résultats des tests prouvent que l'approche ICA/2S/2PM surpasse les méthodes FastICA et Infomax quand on utilise deux signaux superposés à partir de deux sources indépendantes.

CHAPITRE 5 MÉTHODE ICA/2S/2PM AVEC DÉLAIS

5.1 Introduction

Dans le chapitre précédent, nous avons montré que la méthode ICA/2S/2PM réussit à résoudre le problème de séparation de sources de l'ACI pour deux microphones et deux sources. Cependant les tests ont montré que l'existence de délais créait d'importantes imprécisions lors de la reconstruction des sources, chose qu'on ne peut négliger. On a décidé alors de s'inspirer de l'approche de l'ICA/2S/2PM pour développer une méthode générale qui tiendrait compte des délais existants entre les microphones.

5.2 Reformulation du problème ACI

L'existence des délais i et j implique la nécessité de reformuler le problème de séparation de sources. Si on désigne par i le délai associé à la source de parole $A(n)$ et par j le délai associé à la source de bruit $B(n)$, alors on peut réécrire le système d'équation de l'ACI comme suit :

$$\begin{cases} x(n) = A(n) + B(n), \\ y(n) = \alpha \ A(n+i) + \beta \ B(n+j), \end{cases} \tag{5.1}$$

avec $i \in \mathbb{Z}$, $j \in \mathbb{Z}$, $\alpha \in \mathbb{R}$ et $\beta \in \mathbb{R}$.
Afin de reconstruire les sources $A(n)$ et $B(n)$, et réduire le bruit, nous devons commencer par estimer les inconnues i, j, α et β. Cependant, même si ces paramètres sont connus, l'utilisation d'une inverse de la matrice de mixage n'est pas appropriée pour reconstruire les sources à cause des délais i et j. En effet, les sources ont subi un décalage avant le mixage et on ne peut plus utiliser l'approche ICA/2S/PM car les composantes des signaux des microphones $x(n)$ et $y(n)$ sont différentes. Dans ce chapitre nous allons adapter la méthode ICA/2S/2PM pour résoudre le problème de séparation de sources en tenant compte des délais.

5.3 Reconstruction des sources avec une pseudo-inverse

Dans cette section, on présente l'approche développée pour reconstruire les signaux $A(n)$ et $B(n)$ en tenant compte des délais. On suppose ici que les variables i, j, α et β sont connues. On peut alors réécrire le système d'équations (5.1) en remplaçant n par $n-j$ dans la seconde équation pour obtenir

$$\begin{cases} x(n) = A(n) + B(n), \\ z(n) := y(n-j) = \alpha\ A(n+l) + \beta\ B(n), \end{cases} \tag{5.2}$$

où $l = i - j$ et $l \in \mathbb{Z}$.
Dépendamment du signe du paramètre l, on distingue deux formes d'inverses possibles. Considérons un exemple où $n \in \{0,1,2,3\}$ et $l = 1$. On peut alors réécrire le système d'équations (5.2) sous la forme explicite suivante :

$$\begin{pmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \\ z(0) \\ z(1) \\ z(2) \\ z(3) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & \alpha & 0 & 0 & 0 & \beta & 0 & 0 & 0 \\ 0 & 0 & \alpha & 0 & 0 & 0 & \beta & 0 & 0 \\ 0 & 0 & 0 & \alpha & 0 & 0 & 0 & \beta & 0 \\ 0 & 0 & 0 & 0 & \alpha & 0 & 0 & 0 & \beta \end{pmatrix} \begin{pmatrix} A(0) \\ A(1) \\ A(2) \\ A(3) \\ A(4) \\ B(0) \\ B(1) \\ B(2) \\ B(3) \end{pmatrix}. \tag{5.3}$$

Considérons un autre exemple avec $n \in \{1,2,3,4\}$ et $l = -1$. On peut alors réécrire le système (5.2) sous la forme explicite suivante :

$$\begin{pmatrix} x(1) \\ x(2) \\ x(3) \\ x(4) \\ z(1) \\ z(2) \\ z(3) \\ z(4) \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ \alpha & 0 & 0 & 0 & 0 & \beta & 0 & 0 & 0 \\ 0 & \alpha & 0 & 0 & 0 & 0 & \beta & 0 & 0 \\ 0 & 0 & \alpha & 0 & 0 & 0 & 0 & \beta & 0 \\ 0 & 0 & 0 & \alpha & 0 & 0 & 0 & 0 & \beta \end{pmatrix} \begin{pmatrix} A(0) \\ A(1) \\ A(2) \\ A(3) \\ A(4) \\ B(1) \\ B(2) \\ B(3) \\ B(4) \end{pmatrix}. \tag{5.4}$$

En fonction du signe du délai l, la matrice de mixage H peut prendre une forme similaire à (5.3) ou (5.4).
En supposant qu'on connaît déjà les coefficients de mixage α et β et si $l = 1$, on peut constater que le système d'équations (5.3) comprend 8 équations et 9 inconnues :
$\{A(0), A(1), A(2), A(3), A(4), B(0), B(1), B(2), B(3)\}$.
Si $l = -1$ alors le système d'équations (5.4) comprend 8 équations et 9 inconnues :
$\{A(0), A(1), A(2), A(3), A(4), B(1), B(2), B(3), B(4)\}$.
Pour un signal de longueur N, le nombre d'inconnues est égal à $(2N + |l|)$ et le nombre d'équations est égal à $(2N - |l|)$. Si on note par $X \in \mathbb{R}^8$ le vecteur qui contient les valeurs des $x(i)$ et $z(i)$ et $\theta \in \mathbb{R}^9$ le vecteur qui comprend les valeurs des inconnues $A(i)$ et $B(i)$ alors on peut alors réécrire les équations (5.3) et (5.4) sous la forme

$$X = H_{\pm}\theta, \tag{5.5}$$

où $H_{\pm}$ la matrice de mixage de dimensions (8×9). H_+ désigne la matrice mixage pour un délai positif et H_- la matrice de mixage pour un délai négatif. Comme le système en (5.5) est sous-déterminé, on considère plutôt un estimé $\hat{\theta}$ de θ qui est la solution du problème

$$\begin{aligned} &\arg\min_{\theta \in \mathbb{R}^9} ||\theta||^2, \\ \text{sous la contrainte} \quad &X = H_{\pm}\theta. \end{aligned} \tag{5.6}$$

En d'autres termes, parmi tous les θ qui satisfont la contrainte (5.5), on choisit celui qui a la norme la plus petite. La solution du problème (5.6) est

$$\hat{\theta}_{\pm} = H_{\pm}^T (H_{\pm} H_{\pm}^T)^{-1} X, \tag{5.7}$$

si et seulement si $|H_{\pm}^T H_{\pm}| \neq 0$. La matrice $A_{\pm} := H_{\pm}^T (H_{\pm} H_{\pm}^T)^{-1}$ est la pseudo-inverse de Penrose [1].
Pour calculer $\hat{\theta}_{\pm}$, on résout le système en (5.5) avec

$$H_{\pm} H_{\pm}^T V = X, \tag{5.8}$$

et on retrouve la solution

$$\hat{\theta}_{\pm} = H_{\pm}^T V. \tag{5.9}$$

Cela permet de retrouver le même résultat plus rapidement qu'en inversant la matrice $H_{\pm} H_{\pm}^T$. Si la matrice $H_{\pm}$ est constante, on peut alors calculer l'inverse car on peut utiliser la même matrice sur les nouveaux morceaux du signal. Si la matrice $H_{\pm}$ varie contuellement, alors on devrait plutôt résoudre le système d'équations. Dans la partie qui suit, on prend un exemple plus simple, où la longueur du signal est $K = 3$ et le délai est $l = 1$. On peut alors exprimer la matrice de mixage H_+ comme suit :

$$H_+ = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & \alpha & 0 & 0 & \beta & 0 & 0 \\ 0 & 0 & \alpha & 0 & 0 & \beta & 0 \\ 0 & 0 & 0 & \alpha & 0 & 0 & \beta \end{pmatrix}, \tag{5.10}$$

avec

$$|H_+^T H_+| = 2\alpha^6 + 2\alpha^4\beta^2 + 2\alpha^2\beta^4 + \beta^6. \tag{5.11}$$

Sachant que $\alpha > 0$ et $\beta > 0$, alors on peut conclure que $|H_+^T H_+| \neq 0$ et la pseudo-inverse de Moore-Penrose prend la forme

$$A_{\pm} = \begin{pmatrix} 2 & 0 & 0 & \beta & 0 & 0 \\ \alpha\beta & \alpha^2+2 & 0 & \alpha(\alpha^2+\beta^2+1) & \beta & 0 \\ 0 & \alpha\beta & \alpha^2+2 & 0 & \alpha(\alpha^2+\beta^2+1) & \beta \\ 0 & 0 & \alpha\beta & 0 & 0 & \alpha(\alpha^2+\beta^2) \\ \beta^2+2 & \alpha\beta & 0 & \beta(\alpha^2+\beta^2+1) & 0 & 0 \\ 0 & \beta^2+2 & \alpha\beta & \alpha & \beta(\alpha^2+\beta^2+1) & 0 \\ 0 & 0 & \beta^2+2 & 0 & \alpha & \beta(\alpha^2+\beta^2+1) \end{pmatrix}. \tag{5.12}$$

5.3.1 Généralisation de la pseudo-inverse de Moore-Penrose [1] pour tous les délais

On considère le système d'équations (5.2), avec un délai $l \in \mathbb{Z}$ qui satisfait $|l| < 5$ et un indice $n \in \{0, 1, ..., N-1\} =: I$ avec $N >> 5$. Pour le signal $B(n)$, l'indice temporel n satisfait

$$0 < n < N - 1 + l. \tag{5.13}$$

Pour le signal $A(n)$, l'indice temporel n satisfait

$$0 < n < N - 1. \tag{5.14}$$

Comme dans les exemples de la section précédente, on distingue deux cas possibles :
$l > 0$ et $l < 0$.

Cas du délai positif

Si $l > 0$, on définit le vecteur d'inconnues suivant :

$$\theta_+ := (A(0), A(1), ..., A(N-1+l), B(0), B(1), ..., B(N-1)) \quad \in \mathbb{R}^{2N+l}. \tag{5.15}$$

on note aussi

$$a_+ = (A(0), A(1), ..., A(N-1+l)) \quad \in \mathbb{R}^{N+l}. \tag{5.16}$$

et

$$b_+ = (B(0), B(1), ..., B(N-1)) \quad \in \mathbb{R}^{N}. \tag{5.17}$$

Le nombre d'inconnues associé à θ_+ est $2N + l$. On peut écrire la matrice de mixage sous la forme

$$H_+ = \begin{pmatrix} H_x \\ H_y \end{pmatrix} \in \mathbb{R}^{2N \times (2N+l)}, \tag{5.18}$$

où $H_x \in \mathbb{R}^{N \times (2N+l)}$ et $H_y \in \mathbb{R}^{N \times (2N+l)}$ sont les matrices de mixage qui correspondent respectivement aux signaux $x(n)$ et $y(n)$. Le système $X = H\theta$ prend donc la forme

$$X = \begin{pmatrix} H_x \\ H_y \end{pmatrix} \begin{pmatrix} a_+ \\ b_+ \end{pmatrix}. \tag{5.19}$$

Dans ce qui suit, on indexe les composantes d'une matrice en commençant par 0, e.g. A(0) est

la première composante de θ_+. L'équation pour $x(n)$ correspond à nième ligne de la matrice H_x, les sources $A(n)$ et $B(n)$ sont les composantes de longueur respectives $n+1$ et $N+l+n+1$ du vecteur de composantes θ_+, par conséquent les composantes non nulles de la nième ligne de H_x sont données par

$$H_x(n+1, n+1) = 1, \text{ et } H_x(n+1, N+l+n+1) = 1, \ n \in \mathbb{I}. \tag{5.20}$$

L'équation pour $y(n)$ correspond à la nième ligne de la matrice H_y. $A(n+l)$ et $B(n)$ sont les composantes de longueur respectives $(n+l+1)$ et $(N+l+n+1)$ du vecteur θ_+, alors les composantes non nulles de la nième ligne de H_y sont données par

$$H_y(n+1, n+l+1) = \alpha, \ H_x(n+1, N+l+n+1) = \beta, \ n \in \mathbb{I}, \tag{5.21}$$

et la pseudo-inverse peut être calculée avec

$$A_+ = H_+^T (H_+ H_+^T)^{-1} \in \mathbb{R}^{(2N+l)\times 2N}. \tag{5.22}$$

Cas du délai négatif

Si $l < 0$, on définit le vecteur d'inconnues θ_- comme suit :

$$\theta_- := (A(l), A(l+1), ..., A(N-1), B(0), B(1), ..., B(N-1)) \quad \in \mathbb{R}^{2N+l}. \tag{5.23}$$

On écrira aussi

$$a_- = (A(l), A(l+1), ..., A(N-1)) \quad \in \mathbb{R}^{N+l}, \tag{5.24}$$

et

$$b_- = (B(0), B(1), ..., B(N-1)) \quad \in \mathbb{R}^{N}. \tag{5.25}$$

Le nombre d'inconnues est $2N + |l|$. On on peut écrire la matrice de mixage sous la forme

$$H_- = \begin{pmatrix} H_x \\ H_y \end{pmatrix} \in \mathbb{R}^{2N \times (2N-l)}, \tag{5.26}$$

où $H_x \in \mathbb{R}^{N \times (2N-l)}$ et $H_y \in \mathbb{R}^{N \times (2N-l)}$ sont les matrices de mixage qui correspondent respectivement aux signaux $x(n)$ et $y(n)$. On peut réécrire le système $X = H\theta$ comme suit :

$$X = \begin{pmatrix} H_x \\ H_y \end{pmatrix} \begin{pmatrix} a_- \\ b_- \end{pmatrix}. \tag{5.27}$$

Dans qui suit, on indexe les composantes d'une matrice en partant de 0, e.g. A(0) est la première composante de θ_+. L'équation pour $x(n)$ correspond à nième ligne de la matrice H_x, les sources $A(n)$ et $B(n)$ sont les composantes de longueurs $(n-l+1)$ et $(N-1-l+1+n+1 = N-l+n+1)$ du vecteur de composantes θ_-, alors les composantes non nulles de la nième ligne de H_x sont données par

$$H_x(n+1, n-l+1) = 1, \quad H_x(n+1, N-l+n+1) = 1, n \in \mathbb{I}. \tag{5.28}$$

L'équation pour $y(n)$ correspond à la nième ligne de la matrice H_y. $A(n+l)$ et $B(n)$ sont les composantes $n+l-l+1 = n+1$ et $N-l+n+1$ du vecteur θ_-, alors les composantes non nulles de la nième ligne de H_y sont données par

$$H_y(n+1, n+1) = \alpha, \quad H_x(n+1, N-l+n+1) = \beta, n \in \mathbb{I}, \tag{5.29}$$

et la pseudo-inverse est obtenue avec

$$A_{-} = H_{-}^{T}(H_{-}H_{-}^{T})^{-1} \in \mathbb{R}^{(2N+l)\times 2N}. \tag{5.30}$$

On n'inverse pas la matrice, on résout plutôt un système d'équations car la complexité numérique est plus faible. Dans le cas réel, la matrice H ne change pas beaucoup et on peut calculer l'inverse et l'utiliser sur les morceaux subséquents. Si la matrice H change, alors on peut la mettre à jour et recalculer sa pseudo-inverse.

Cas du délai nul

Dans le cas où $l = 0$, on peut simplement calculer directement l'inverse de la matrice de mixage H comme le suggère la méthode ICA/2S/2PM.

5.3.2 Test de la méthode de pseudo-inverse de Moore-Penrose [1]

L'approche de la pseudo-inverse expliquée précédemment permet de résoudre le système d'équations qui prend en considération les délais entre les microphones. Cependant la pseudo-inverse de Moore-Penrose [1] est une estimation de l'inverse de la matrice de mixage H et elle ne donne pas des résultats exacts. Dans cette section nous testons cette méthode afin de connaître ses limitations et pour l'adapter afin de retrouver une reconstruction correcte des signaux sources $A(n)$ et $B(n)$. On commence par exécuter un test préliminaire en utilisant un signal sinusoidal auquel on additionne un signal de bruit blanc avec les paramètres de mixage $\alpha = 1.1$ et $\beta = 0.9$ et un délai $l = 5$. La figure (5.1) montre les reconstructions des deux signaux $A(n)$ et $B(n)$.

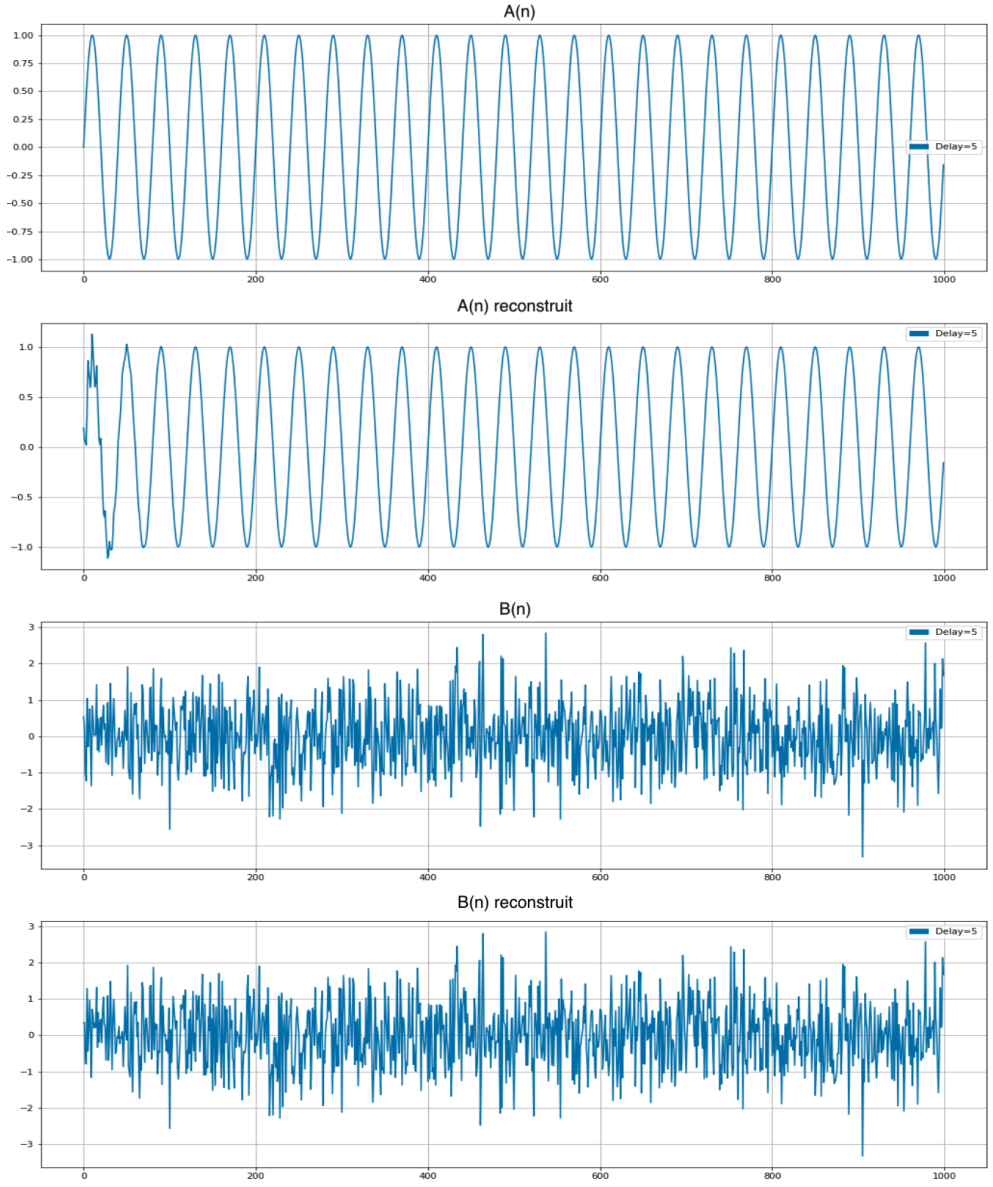

Figure 5.1 Signaux reconstruits avec la pseudo-inverse de Moore-Penrose.

(L'unité de temps est la période d'échantillonnage)

Dans la figure (5.1), on constate que la reconstruction de $A(n)$ présente des imprécisions au début du signal. Elles sont dues à l'erreur induite par l'estimation de la pseudo-inverse. Le temps de calcul de la pseudo-inverse de Moore-Penrose [1] dépend des dimensions de la matrice de mixage H. L'estimation de la pseudo-inverse pour un signal d'une longueur de 1000 échantillons prend un temps considérable (>0.5 seconde). Il faut donc utiliser des signaux de petites tailles pour réduire la complexité du calcul. Dans la suite, on utilise une pseudo-inverse sur de petits intervalles pour reconstruire les signaux par morceaux et réduire le temps de calcul. Pour l'expérience de la figure (5.2), nous avons utilisé un délai $l = 1$. Nous avons reconstruit le signal $A(n)$ en utilisant des fenêtres disjointes juxtaposées de taille $L = 100$ échantillons. Les expériences effectuées ont permis d'estimer la pseudo-inverse et de reconstruire les signaux en moins de 0.1 seconde pour des signaux de taille inférieure à 500 échantillons (équivalant à 0.03125 seconde).
On a observé dans la figure (5.2) une erreur périodique à chaque reconstruction de morceau du signal.

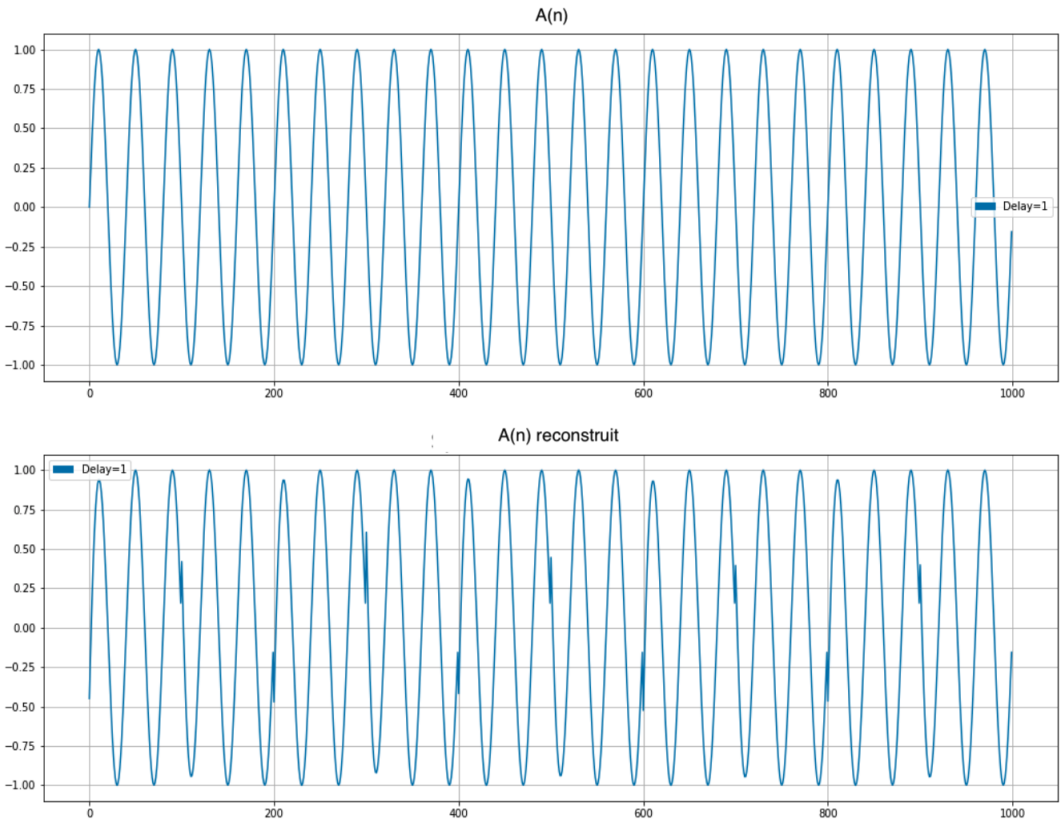

Figure 5.2 Signaux reconstruits avec la pseudo-inverse Moore-Penrose par morceaux.

(L'unité de temps est la période d'échantillonnage)

La figure (5.2) montre que la reconstruction du signal $A(n)$ présente une discontinuité causée par une erreur périodique lors de la reconstruction par morceaux. La figure (5.3) montre que la différence calculée entre le signal $A(n)$ et son estimé. On remarque un pic d'erreur au début de chaque morceau qui diminue par la suite. Cette erreur est élevée et il est nécessaire d'adapter notre approche pour la réduire.

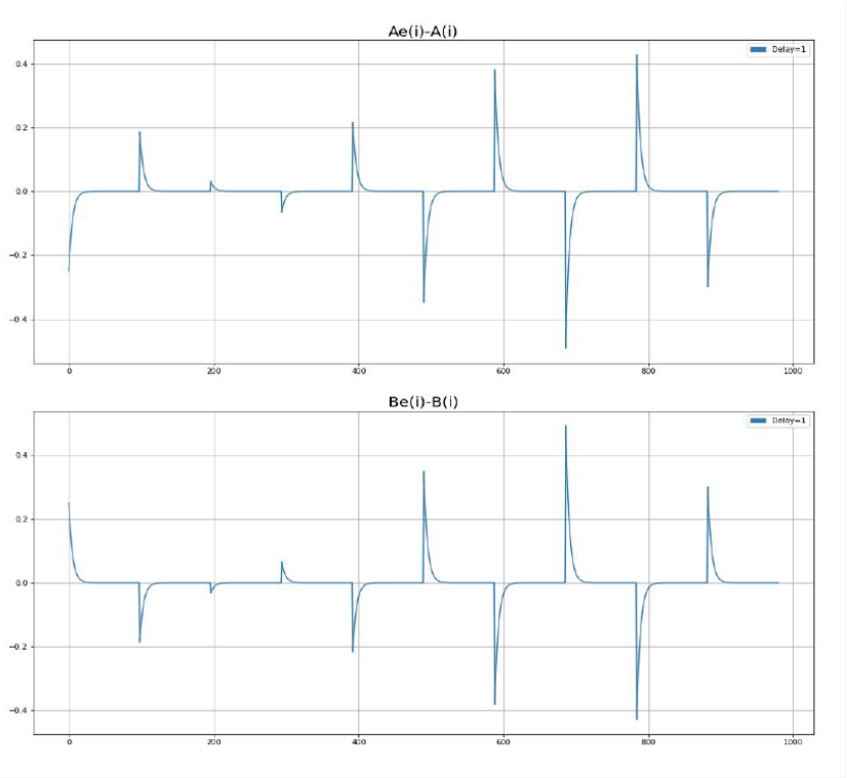

Figure 5.3 Différence entre A(n) et son estimé.

(L'unité de temps est la période d'échantillonnage)

Cette erreur peut se situer à droite ou à gauche de la reconstruction de chaque morceau dépendamment du signe du délai. Comme le montre la figure (5.1), si le délai est positif, alors l'erreur se trouve au début de la reconstruction. Sinon si le délai est négatif, alors l'erreur se trouve à la fin de la reconstruction. Cette erreur peut être réduite si on utilise une reconstruction de morceaux avec un recouvrement entre les morceaux. On a alors ajouté un paramètre $r > 0$ de recouvrement entre les morceaux consécutifs. Cela permet à chaque itération de choisir seulement la partie de la reconstruction qui ne contient pas d'erreur. Si le délai est positif, alors on remplacera l'erreur qui se trouve dans la première partie de la reconstruction par la dernière partie de la reconstruction du morceau précédent. Si le délai est négatif, alors on remplace l'erreur qui se trouve dans la dernière partie de la reconstruction par la première partie de la reconstruction du morceau successif.

La figure (5.4) montre que cette méthode permet de réduire l'erreur périodique observée précédemment. Ici on utilise un recouvrement $r = 10$ entre deux morceaux consécutifs, ce qui représente 10% de la taille du morceau reconstruit qui a une longueur de 100 échantillons. On peut constater une nette amélioration, la reconstruction est presque identique à la source $A(n)$

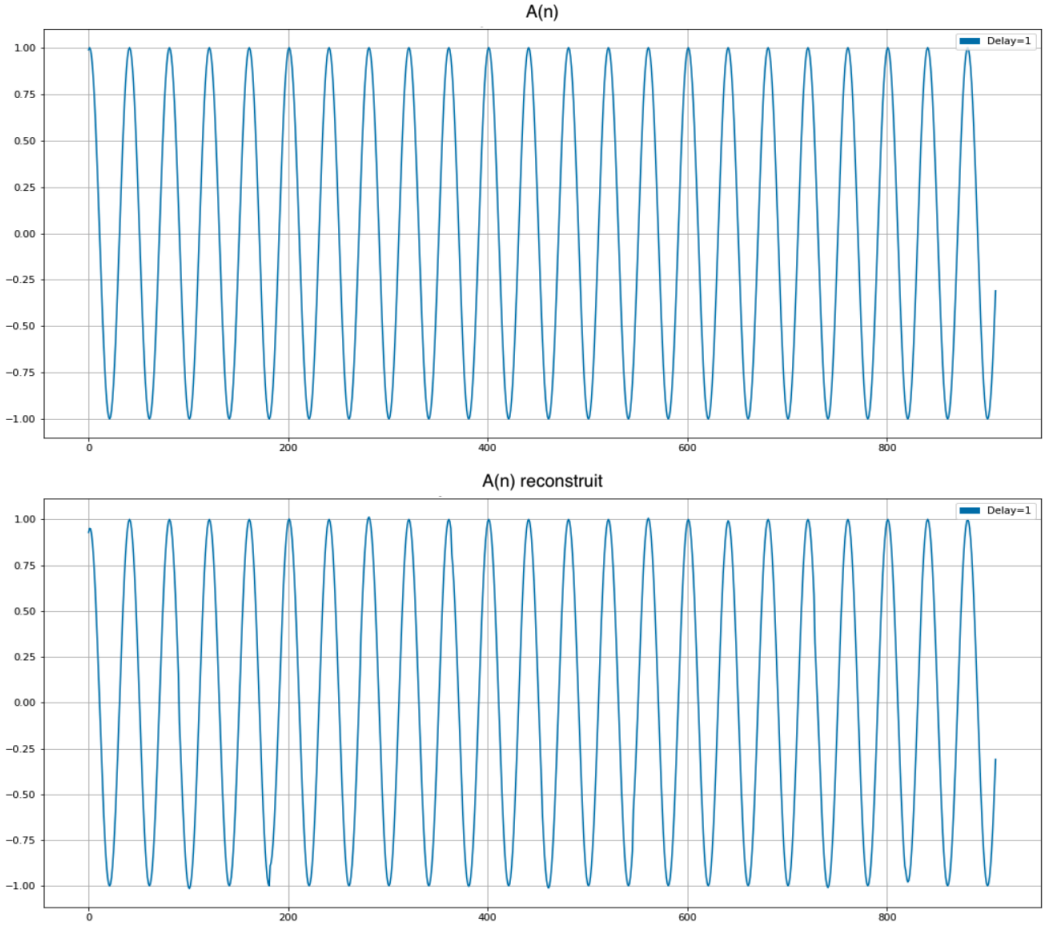

Figure 5.4 Signaux reconstruits avec la pseudo-inverse Moore-Penrose par morceaux avec recouvrement.

(L'unité de temps est la période d'échantillonnage)

Pour vérifier la précision de l'approche de la pseudo-inverse avec recouvrement, on a testé la méthode sur un signal de parole auquel on a additionné un signal de bruit de TV. La figure (5.5) représente les signaux utilisés pour créer les deux superpositions utilisés.

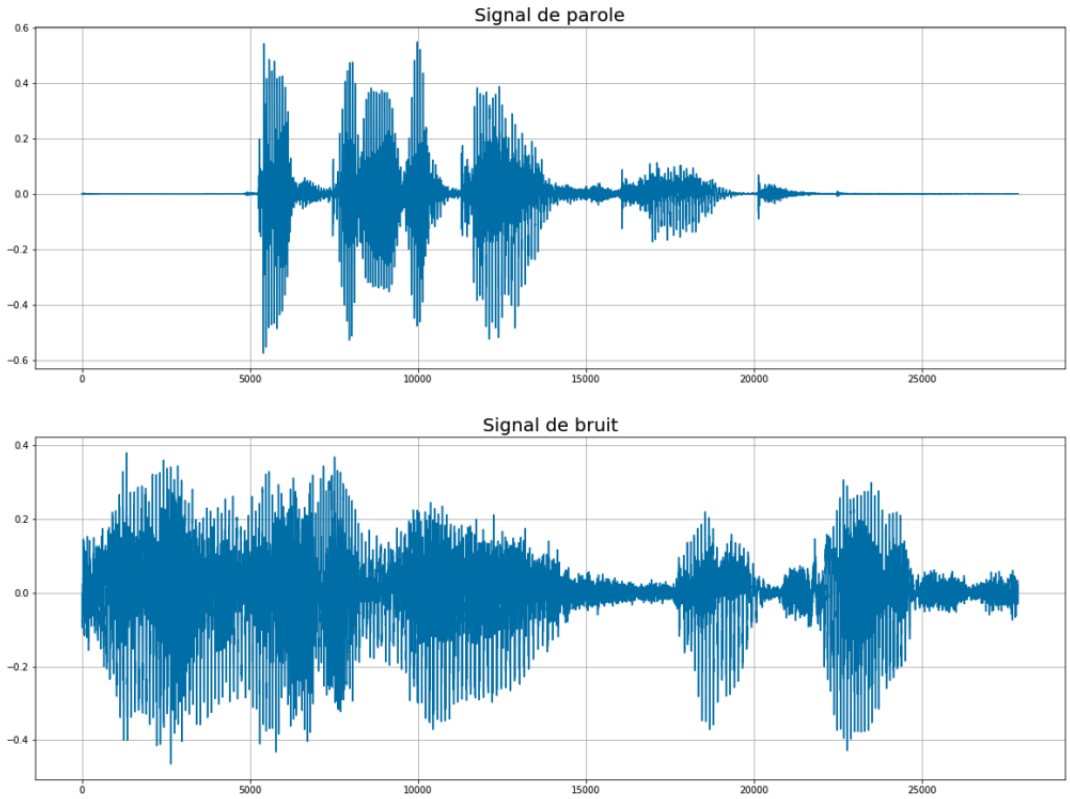

Figure 5.5 Signaux des sources utilisées pour le test de la pseudo-inverse Moore-Penrose par morceaux avec recouvrement.

(L'unité de temps est la période d'échantillonnage)

On a utilisé un délai $l = 5$ (délai maximal entre les microphones) entre les signaux des microphones et une pseudo-inverse avec une taille de 500 échantillons et un recouvrement de 250 échantillons entre les morceaux consécutifs . Pour tester la précision de la méthode on a calculé l'erreur entre les amplitudes de l'estimé $\hat{A}(n)$ et $A(n)$. La figure (5.6) montre que l'erreur périodique est très faible si on utilise la pseudo-inverse avec recouvrement.

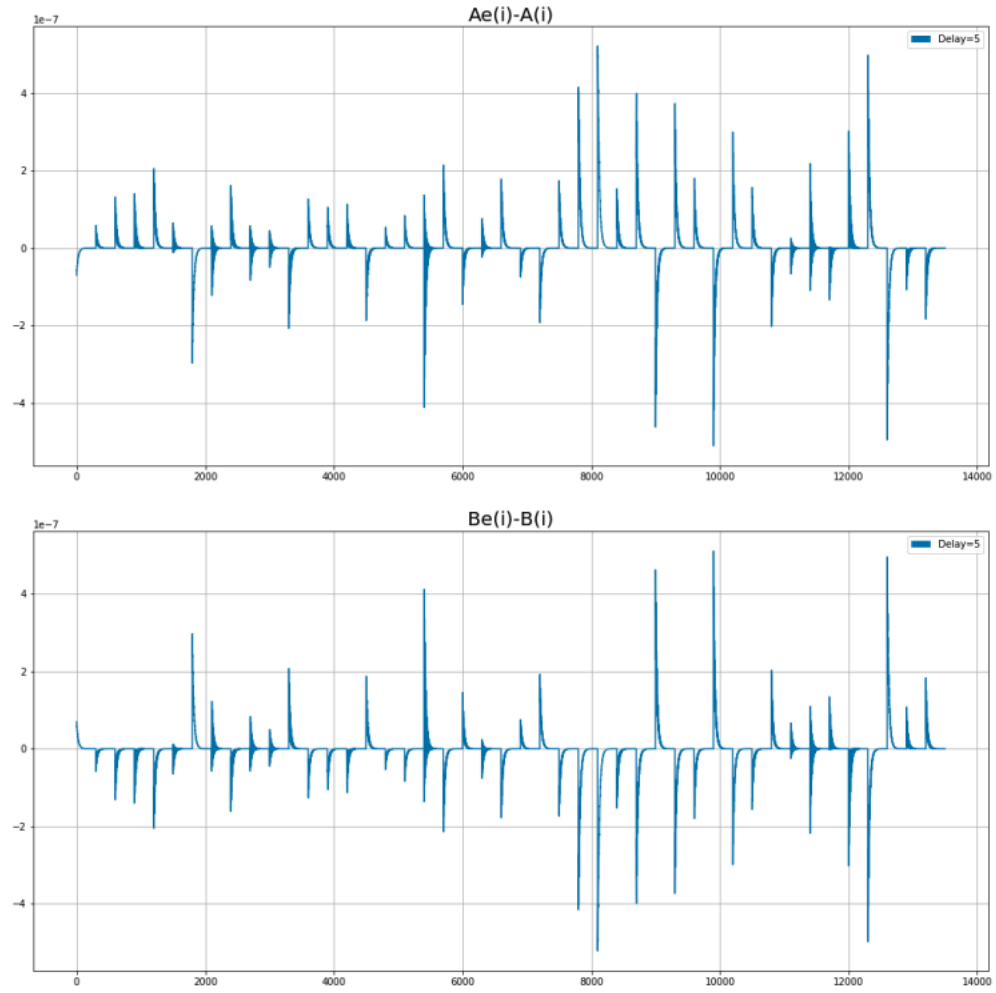

Figure 5.6 Signaux reconstruits avec la pseudo-inverse Moore-Penrose par morceaux avec recouvrement.

(L'unité de temps est la période d'échantillonnage)

5.3.3 Conclusion

La méthode développée permet de reconstruire de manière précise les signaux des sources $A(n)$ et $B(n)$. L'approche qui utilise une reconstruction par morceaux avec recouvrement a permis de corriger les erreurs liées à l'estimation de la pseudo-inverse de Moore-Penrose [1]. Grâce à cette approche, on peut reconstruire avec précision les signaux des sources même en présence de délai.

5.4 Estimation de j et β

Durant les périodes calmes, le locuteur ne parle pas et on peut alors supposer que cette période est caractérisée par $A(n) = 0$. Ainsi le problème de l'ACI peut être réécrit de la manière suivante :

$$\begin{cases} x(n) = B(n), \\ y(n) = \beta \ B(n+j). \end{cases} \tag{5.31}$$

On considère un intervalle de temps I_k des signaux des microphones $x(n)$ et $y(n)$ défini par

$$I_k = [m_k, n_k]. \tag{5.32}$$

En supposant que le délai satisfait $j \in \{-5, -4, ..., 4, 5\}$, on peut définir les deux vecteurs $x_k \in \mathbb{R}^N$ et $y_{k,j} \in \mathbb{R}^N$ avec $N = n_k - m_k + 1$ comme suit :

$$\begin{cases} x_k(n) = x(n), \quad n \in I_k, \\ y_{k,j}(n) = y(n+j), \quad n \in I_k. \end{cases} \tag{5.33}$$

Nous voulons calculer le coefficient de corrélation maximal $\rho^*(k)$ donné par

$$\rho^*(k) := \max_{j \in J} \ \rho(x_k, y_{k,j}). \tag{5.34}$$

On utilise l'estimateur du coefficient de corrélation entre deux vecteurs arbitraires $X \in \mathbb{R}^N$ et $Y \in \mathbb{R}^N$

$$\rho(X, Y) = \frac{\sum_{i=1}^{N} (X(i) - \hat{X})(Y(i) - \hat{Y})}{\sqrt{\sum_{i=1}^{N} (X(i) - \hat{X})^2} \sqrt{\sum_{i=1}^{N} (Y(i) - \hat{Y})^2}}, \tag{5.35}$$

où $\hat{X} = \frac{1}{N} \sum_{i=1}^{N} X(i))$ et $\hat{Y} = \frac{1}{N} \sum_{i=1}^{N} Y(i))$.

On définit l'estimé $j^*(k)$ du délai j pendant une période calme par

$$j^*(k) = \arg\max_{j \in J} \ \rho(x_k, y_{k,j}). \tag{5.36}$$

On effectue un test de vérification sur des signaux superposés en utilisant un délai $j = 5$. La figure suivante montre que le coefficient de corrélation atteint son maximum pour la valeur $j^* = 5$

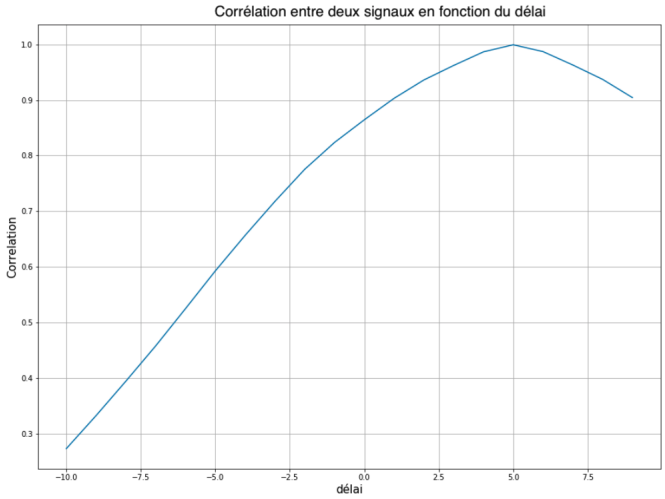

Figure 5.7 Coefficient de corrélation en fonction du délai.

Supposons qu'on a réussi à détecter une période calme et à estimer la vraie valeur du délai j. Il devient alors facile d'estimer le paramètre de mixage β. On peut utiliser le modèle suivant

$$y(n) = \beta \ x(n+j) + e(j), \tag{5.37}$$

où $e(j)$ est une erreur .On peut alors utiliser une régression linéaire entre $y(n)$ et $x(n+j)$ pour estimer le coefficient β, ce qui donne l'estimé

$$\hat{\beta} = \frac{\sum_{i=1}^{N}(y(i)-\hat{y})(x(i+j)-\hat{x})}{\sum_{i=1}^{N.}(x(i)-\hat{x})^2}, \tag{5.38}$$

où $\hat{x}$ et $\hat{y}$ sont les moyennes des signaux $x(n)$ et $y(n)$.

5.4.1 Détection des périodes calmes

Afin de pouvoir estimer le coefficient β et le délai j, il est impératif de détecter une période calme pendant l'enregistrement du microphone. Dans cette section, nous avons investigué différentes approches pour détecter la présence d'une période calme dans un enregistrement où une personne parle en prenant des pauses.

Approche de la maximisation du coefficient de corrélation

Si on se situe dans une période calme, alors pour un délai estimé $j^* = j$, le coefficient de corrélation entre les signaux $x(n)$ et $y(n)$ atteint une valeur proche de 1 (1 étant la valeur maximale). Une approche serait de calculer le coefficient de corrélation pour les différentes valeurs possibles du délai $j_k \in [-5, ..., 5]$ entre les différents morceaux x_k et y_k des signaux $x(n)$ et $y(n)$. On peut alors classifier une période comme calme si et seulement si son coefficient de corrélation $\rho_k \approx 1$. La figure (5.8) montre les résultats obtenus pour le calcul du coefficient de corrélation maximal retrouvé pour différents morceaux de longueur de 10 000 échantillons.

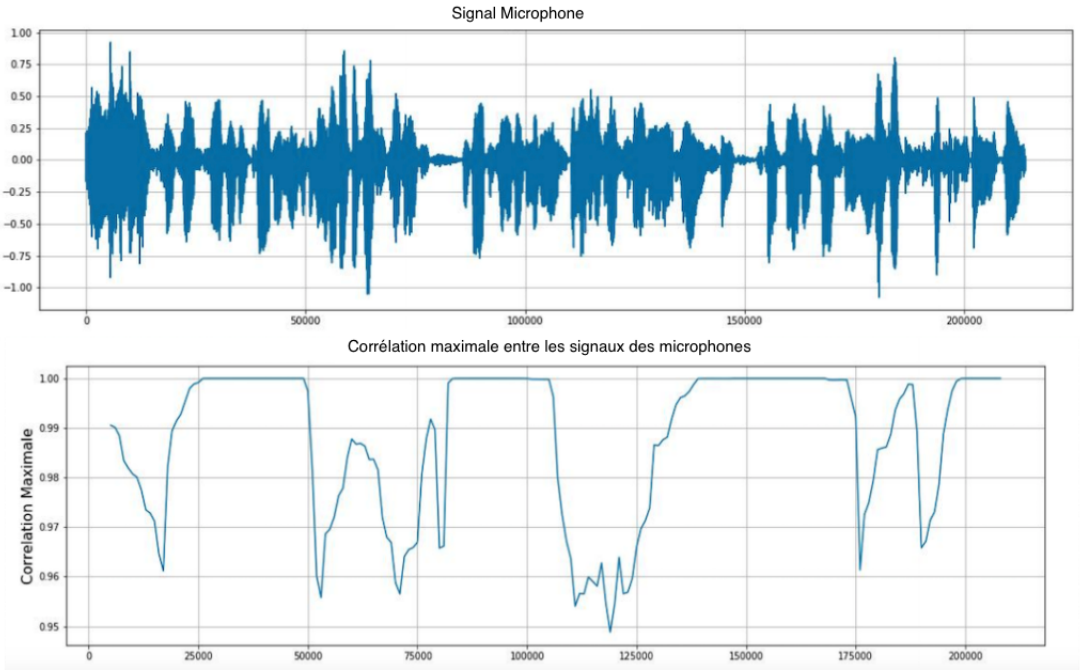

Figure 5.8 Variation du coefficient de corrélation maximal par rapport au temps.

(L'unité de temps est la période d'échantillonnage)

On voit bien que la courbe de corrélation maximale entre les signaux des microphones (5.9) présente des plateaux tels que $\rho_k \approx 1$. Si on représente seulement le signal $A(n)$ avec la

courbe du coefficient de corrélation maximal, on peut constater que les plateaux coïncident exactement avec les périodes où la personne ne parle pas, i.e période calme

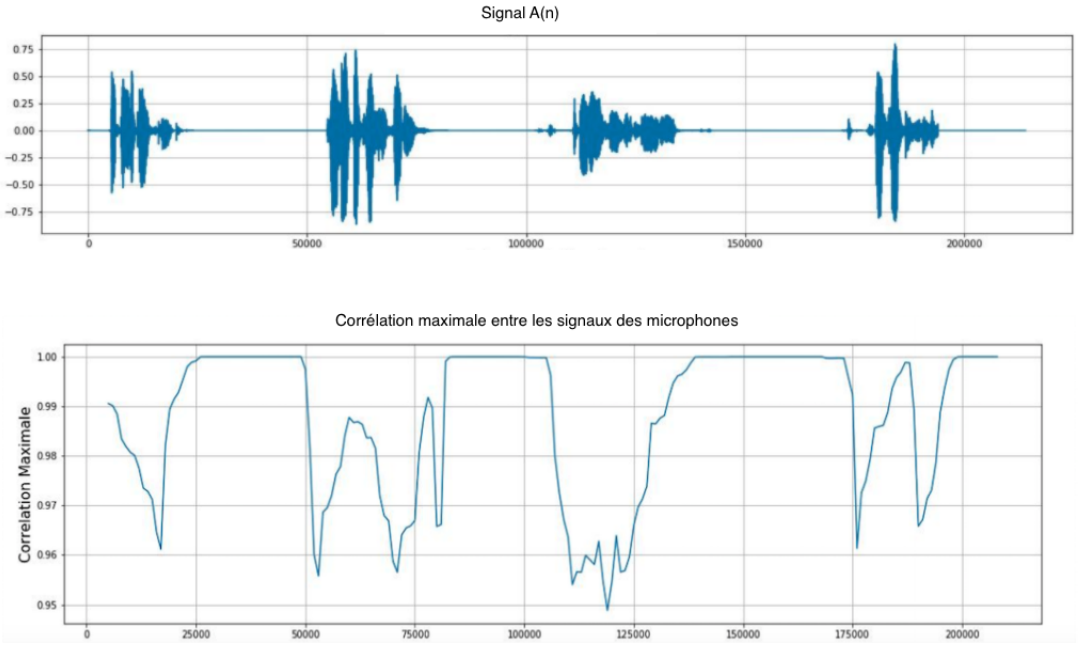

Figure 5.9 Coefficient de corrélation maximal en fonction du temps.

(L'unité de temps est la période d'échantillonnage)

Une idée serait de classifier comme périodes calmes les morceaux du signal du microphone qui ont une valeur $\rho_k \approx 1$ et qui seraient aussi entourée par deux autres morceaux ayant aussi un coefficient de corrélation très proche de 1, i.e $\rho_{k-1} \approx 1$ et $\rho_{k+1} \approx 1$.
Cependant on voit sur la figure (5.9) que le coefficient de corrélation atteint des valeurs proches de 1 même pendant les périodes actives. Cette approche nécessite l'utilisation d'un seuil qui doit être initialisé à l'avance. Si le signal de parole est beaucoup plus intense que le signal de la source de bruit, le coefficient de corrélation maximal reste élevé même pendant les périodes actives, contrainte qui rend le choix du seuil difficile et rend la méthode sensible au changements de la nature des signaux. Quand le signal de la source de bruit est négligeable par rapport au signal de parole, l'approche de la corrélation maximale ne permet plus de différencier les périodes calmes des periodes actives car le coefficient de corrélation est très proche de 1 même pendant les périodes actives.

L'approche de la machine à vecteur de support

La détection automatique de l'activité vocale est un domaine de recherche très actif. Plusieurs approches ont été développées afin de détecter la voix d'une personne quand elle parle. L'approche utilisant la machine à vecteur de support ainsi que celle utilisant les réseaux de neurones profonds sont les approches les plus connues pour reconnaître la voix humaine. Ces méthodes nous ont inspiré à utiliser la machine à vecteur de support pour détecter les périodes calmes. La machine à vecteur de support est une méthode statistique d'apprentissage automatique supervisée qui nécessite des données d'entraînement. L'idée est d'entraîner une machine à vecteur de support binaire pour classifier différents morceaux d'un signal en période calme ou période active. Cet algorithme est entraîné à différencier les morceaux du signal qui présentent une forte énergie contre les morceaux qui ont une faible énergie. L'énergie d'un signal x est définie par

$$E = \sum_{i=1}^{N} |x(n)|^2. \tag{5.39}$$

Pour segmenter un signal en périodes silencieuses et périodes calmes, nous utilisons une approche développée par Giannakopoulos [17]. Cette approche consiste à construire une base de données d'entraînement à partir d'un enregistrement vocal d'entraînement. Pour construire cet ensemble de données, il est important de bien choisir les caractéristiques du signal qui seront utilisées pour faire la classification. On utilise l'approche et le code de la librairie pyAudio développée par Giannakopoulos [17] pour choisir et calculer les différentes caractéristiques. On commence par diviser l'enregistrement en plusieurs morceaux, puis on calcule pour chaque morceau k son énergie E_k. Pour chaque morceau, on calcule aussi plusieurs caractéristiques qui vont constituer nos données d'entraînement. Giannakopoulos [17] suggère d'utiliser les caractéristiques de 10% des morceaux qui ont la plus forte énergie et les 10% des morceaux qui ont la plus faible énergie parmi tous les morceaux du signal d'entraînement.
Les caractéristiques définies par Giannakopoulos [18] sont les suivantes :
- Le taux de changement du signe d'un signal appelé aussi Zero Crossing Rate (ZCR) mesure le nombre de fois que le signal change de signe pendant une fenêtre précise

$$\mathrm{ZCR}(x_k) = \sum_{n=-\infty}^{+\infty} |\mathrm{sign}[x(n)] - \mathrm{sign}[x(n-1)]| t(k-n), \tag{5.40}$$

où

$$\mathrm{sign}[x(n)] = \begin{cases} 1, & \text{si } x > 0, \\ -1, & \text{si } x < 0, \end{cases} \tag{5.41}$$

et

$$t[x(n)] = \begin{cases} \frac{1}{2N}, \text{ pour } 0 < n < N-1, \\ 0, \text{ sinon.} \end{cases} \tag{5.42}$$

- L'entropie de l'énergie décrit la dispersion de l'énergie d'une fenêtre d'un signal, elle peut aussi indiquer des changements brusques dans le signal. L'entropie H d'un signal x_k est définie par

$$H(x_k) = -\sum_{i=1}^{N} p(i) \ln(p(i)). \tag{5.43}$$

avec $p(i) > 0$ les distributions de probabilités des $x(i)$.

- Le centroïde spectral est une mesure qui caractérise le spectre d'un signal et indique la localisation du centre du spectre d'un signal. Le Centroïde C d'un signal de fréquence f_i ayant une transformée de Fourier discrete $F_i(n)$ peut être calculé avec

$$C = \frac{\sum_{i=1}^{N} F_i f_i}{\sum_{i=1}^{N} F_i}. \tag{5.44}$$

- La propagation spectrale mesure la moyenne de déviation autour du centroïde. Les signaux qui contiennent du bruit ont en général une large propagation spectrale alors que la voix humaine a une propagation spectrale faible. La propagation spectrale *Spread* d'un signal de fréquence f_i avec un centroïde C peut être calculée avec

$$\text{Spread} = \sqrt{\frac{\sum_{i=1}^{N} F_i (f_i - C)^2}{\sum_{i=1}^{N} F_i}}. \tag{5.45}$$

- Le flux spectral est une mesure de la fluctuation de la magnitude du spectre d'un signal, elle mesure la différence carrée entre deux magnitude de spectres de deux fenêtres consécutives d'un signal. Pour un signal $x(n)$ on peut calculer le flux spectral comme suit :

$$F_s = \sum_{i=1}^{N} (E_k(n) - E_{k-1}(n))^2. \tag{5.46}$$

avec $E_k(n) = \frac{x_k(n)}{\sum_{n=1}^{N} x_k(n)}$.

- Les coefficients MFCC sont les coefficients de fréquence cepstral de Mel et décrivent une représentation cepstrale des bandes de fréquences non-linéaire distribuées selon l'échelle de Mel. Ces coefficients sont calculés à partir de la transformation en cosinus de l'énergie spectrale.

- Le vecteur Chroma et sa déviation est une propriété du signal qui décrit le ton d'un enregistrement audio et ainsi la qualité de ce dernier. Le vecteur Chroma est en général un vecteur à 12 éléments qui décrit l'énergie des différentes classes de hauteur (C, C, D, D, E, ..., B). La déviation de Chroma décrit tout simplement l'écart type du vecteur Chroma.

On définit une étiquette $y_k = -1$ (comme période active) pour les caractéristiques qui sont associées à une haute énergie et une étiquette $y_k = +1$ aux caractéristiques qui sont associées à une faible énergie. Par la suite on entraîne le SVM en utilisant les données d'entraînement calculées précédemment afin de calculer nos prédictions. On effectue un test de vérification de l'approche implémentée sur un enregistrement d'un microphone de notre partenaire Fluent.ai. On a entraîné un SVM sur des morceaux de 5 000 échantillons et un pas de 1000 échantillons. Pour l'entraînement, on a utilisé un signal de longueur de 1 600 000 échantillons (100 secondes). La figure suivante montre que l'approche réussit bien à détecter les périodes calmes colorées en rouge.

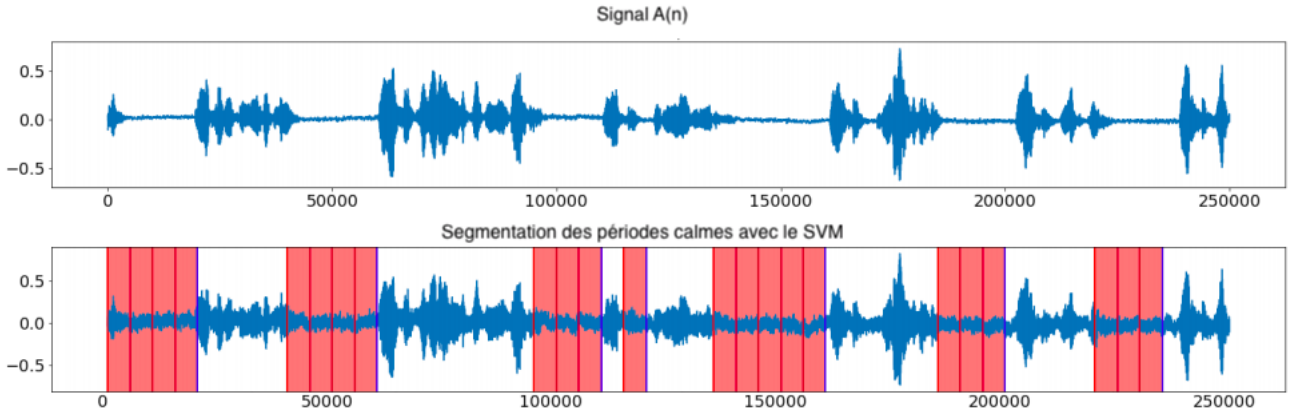

Figure 5.10 Détection des périodes calmes avec le SVM.

(L'unité de temps est la période d'échantillonnage)

5.5 Estimation de i et α

Après avoir réussi à estimer le délai j et le coefficient β, nous présentons dans cette partie l'approche utilisée pour estimer le délai i ainsi que le coefficient α.
Nous utilisons les paramètres j et β, maintenant connus, pour réécrire le système d'équations en (5.1). En soustrayant j on peut retrouver le système suivant :

$$\begin{cases} x(n) = A(n) + B(n), \\ z(n) := y(n-j) = \alpha\ A(n+i-j) + \beta\ B(n). \end{cases} \tag{5.47}$$

On utilise $l = i - j$ afin de simplifier l'écriture des équations. Pour cette partie de la méthode, on utilise similairement à ICA/2S/2PM sans délais les corrélations entre x et z. On utilise le paramètre de décalage $m \in \mathbb{Z}$ pour dériver les équations suivantes à partir de (5.47)

$$\begin{cases} x(n+m) = A(n+m) + B(n+m), \\ z(n-m) = \alpha\ A(n+l-m) + \beta\ B(n-m). \end{cases} \tag{5.48}$$

En combinant les systèmes en (5.47) et (5.48), on retrouve le système d'équations suivant :

$$\begin{cases} x(n) = A(n) + B(n), \\ z(n) := y(n-j) = \alpha\ A(n+l) + \beta\ B(n), \\ x(n+m) = A(n+m) + B(n+m), \\ z(n-m) = \alpha\ A(n+l-m) + \beta\ B(n-m). \end{cases} \tag{5.49}$$

On définit les différents moments qu'on utilisera pour estimer i et α :

$$AA := E\left\{A(n)A(n)\right\}, \tag{5.50}$$

$$AA_m := E\left\{A(n)A(n+m)\right\}, \tag{5.51}$$

$$BB := E\left\{B(n)B(n)\right\}, \tag{5.52}$$

$$BB_m := E\left\{B(n)B(n+m)\right\}, \tag{5.53}$$

$$XX := E\left\{x(n)x(n)\right\}, \tag{5.54}$$

$$XX_m := E\left\{x(n)x(n+m)\right\}, \tag{5.55}$$

$$ZZ := E\left\{z(n)z(n)\right\}, \tag{5.56}$$

$$ZZ_m := E\{z(n)z(n+m)\}, \tag{5.57}$$

$$XZ := E\{x(n)z(n)\}, \tag{5.58}$$

$$XZ_m := E\{x(n)z(n+m)\}, \tag{5.59}$$

$$ZX := E\{z(n)x(n)\}, \tag{5.60}$$

$$ZX_m := E\{z(n)x(n+m)\} = E\{z(n-m)x(n)\}. \tag{5.61}$$

En utilisant les équations en (5.49), on obtient

$$XX = AA + BB, \tag{5.62}$$

$$XX_m = AA_m + BB_m, \tag{5.63}$$

$$ZZ = \alpha^2 \ AA + \beta^2 \ BB, \tag{5.64}$$

$$ZZ_m = \alpha^2 \ AA_m + \beta^2 \ BB_m, \tag{5.65}$$

$$XZ = \alpha \ AA_l + \beta \ BB, \tag{5.66}$$

$$XZ_m = \alpha \ AA_{l-m} + \beta \ BB_m. \tag{5.67}$$

Les équations précédentes décrivent un système d'équations non linéaire avec les inconnues

$$\{AA, AA_m, AA_l, AA_{l-m}, BB, BB_m, \alpha\}. \tag{5.68}$$

Si $m \neq l$, alors le système d'équations (5.62)-(5.67) est sous-déterminé car il contient six équations et sept inconnues. On sait déjà que les statistiques XX_m et ZZ_m ne dépendent pas du signe de m car les processus aléatoires $A(n)$ et $B(n)$ sont stationnaires. Par contre les variables XZ_m dépendent bien du signe de m car

$$\begin{aligned} XZ_{-m} - XZ_m &= (\alpha AA_{l+m} + \beta BB_{-m}) - (\alpha AA_{l-m} + \beta BB_m), \\ &= (\alpha AA_{l+m} - \alpha AA_{l-m}) \ \text{car} \ B_{-m} = B_m, \\ &\text{pour tout } m \ \neq 0. \end{aligned} \tag{5.69}$$

Si $l = 0$, alors

$$XZ_{-m} - XZ_m = 0 \ \text{pour tout m} \ . \tag{5.70}$$

La sensibilité de XZ_m au signe de l et la sous-détermination du système d'équations (5.62)-(5.67) suggère l'addition de la statistique XZ_{-m} afin d'augmenter le nombre d'équations. On aura ainsi

$$XX = AA + BB, \tag{5.71}$$

$$XX_m = AA_m + BB_m, \tag{5.72}$$

$$ZZ = \alpha^2 AA + \beta^2 BB, \tag{5.73}$$

$$ZZ_m = \alpha^2 AA_m + \beta^2 BB_m, \tag{5.74}$$

$$XZ = \alpha\ AA_l + \beta\ BB, \tag{5.75}$$

$$XZ_m = \alpha\ AA_{l-m} + \beta\ BB_m, \tag{5.76}$$

$$XZ_{-m} = \alpha\ AA_{l+m} + \beta\ BB_m. \tag{5.77}$$

Ainsi on retrouve un système d'équations sous-déterminé qui inclut sept équations et huit inconnues données par

$$\{AA, AA_m, AA_l, AA_{l-m}, AA_{l+m}, BB, BB_m, \alpha\}. \tag{5.78}$$

Si $m = l$, alors le système d'équations prend la forme plus simple

$$XX = AA + BB, \tag{5.79}$$

$$XX_l = AA_l + BB_l, \tag{5.80}$$

$$ZZ = \alpha^2 AA + \beta^2 BB, \tag{5.81}$$

$$ZZ_l = \alpha^2 AA_l + \beta^2 BB_l, \tag{5.82}$$

$$XZ = \alpha\ AA_l + \beta\ BB, \tag{5.83}$$

$$XZ_l = \alpha\ AA + \beta\ BB_l, \tag{5.84}$$

$$XZ_{-l} = \alpha\ AA_{2l} + \beta\ BB_l. \tag{5.85}$$

Le système d'équations qui précède est non linéaire et est composé maintenant de sept équations et six inconnues

$$\{AA, AA_l, AA_{2l}, BB, BB_l, \alpha\}. \tag{5.86}$$

On constate que ce système d'équations est sur-déterminé et on peut même suggérer que le paramètre β puisse être estimé si ce dernier était inconnu. Ainsi on peut estimer α et par la même occasion le délai j car ce système d'équations n'est soluble que si $j = l$. Une approche pour résoudre ce système d'équations non linéaire est de minimiser les résidus dans le sens des moindres carrés en utilisant la méthode d'optimisation de Levenberg-Marquardt. On utilise les conditions initiales suivantes :

$$\begin{cases} AA = 0, \\ AA_l = 0, \\ AA_{2l} = 0, \\ BB = 0, \\ BB_l = 0, \\ \alpha = 1. \end{cases} \tag{5.87}$$

On a commencé par résoudre le système sur-déterminé au complet et choisir le délai l qui donne le résidu le plus faible de tout le système. Cette approche fonctionne bien sauf pour le cas où $l = 0$. Si le délai l est nul, l'algorithme fonctionne mais retrouve de temps en temps un estimé faux $l = 1$ ou $l = -1$. Ce problème nous a poussé à utiliser une méthode basée sur la sur-détermination du système d'équations. Sachant que le délai $j \in [-5, ..., 5]$, on peut alors résoudre le système d'équations pour les différentes valeurs possibles de j. Pour chaque itération, on résout le système d'équations pour les sept combinaisons possibles de six équations prises parmis les sept équations (5.79)-(5.85) et on calcule la somme des résidus à chaque fois. Après avoir résolu le système d'équations pour les sept combinaisons possibles, on calcule la variance des estimés de α obtenues pour toutes combinaisons d'équations. Après avoir calculé la variance de α pour les différentes valeurs possibles de j, notre estimé j^* de j est celui qui produit la variance de α la plus faible.

5.6 Méthode ICA/2S/2PM avec délais par morceau

Après avoir développé les approches nécessaires pour l'estimation des coefficients de mixage α et β ainsi que les délais i et j, nous avons regroupé les différentes parties de la méthode ICA/2S/2PM avec délais dans un seul bloc.
Notre partenaire Fluent.ai utilise des algorithmes de traitement de la voix naturelle qui reçoivent comme entrée des petits morceaux d'enregistrement. Ainsi Fluent.ai souhaiterait avoir une approche qui fonctionne par morceau. Afin de satisfaire ce besoin, nous avons adapté notre algorithme afin de calculer les estimations nécessaires et générer les reconstructions des signaux $A(n)$ et $B(n)$ par morceau. Nous avons aussi ajouté quelques contraintes afin de bien vérifier qu'un morceau de signal est bel et bien classé comme une période calme. Pour l'expérience suivante, nous avons utilisé un enregistrement de longueur de 100 secondes pour entraîner un SVM comme décrit par Giannakopoulos [17] afin de détecter les périodes calmes d'un enregistrement d'un microphone dans un environnement bruyant. Cet SVM a été entraîné sur des fenêtres de 0.3125 secondes et un pas de 0.0625 secondes. La figure (5.11) montre les résultats obtenus pour la segmentation des périodes calmes sur une partie de l'enregistrement du microphone

Figure 5.11 Segmentation des périodes calmes.

(L'unité de temps est la période d'échantillonnage)

On remarque que la segmentation des périodes calmes fonctionne bien malgré l'existence de quelques fausses prédictions signalant la présence de périodes calmes. Afin d'éviter ce genre de fausses prédictions, nous avons ajouté une contrainte lors de la classification du SVM, un morceau du signal est classé comme période calme si et seulement si le morceau précédent et le morceau suivant est aussi classé comme période calme par le SVM. On utilise le critère

$$f_{calm}(x_k) = +1 \text{ si et seulement si } f_{svm}(x_k) = f_{svm}(x_{k-1}) = f_{svm}(x_{k+1}) = +1, \qquad (5.88)$$

où f_{calm} est la fonction de classification des périodes calmes et f_{svm} la fonction de prédiction du SVM. Si $f_{calm} = +1$, alors l'algorithme estime les paramètres β et le délai j grâce à l'approche utilisant la maximisation de la corrélation et la régression linéaire. Cependant, il n'est pas possible de calculer une reconstruction des sources $A_k(n)$ et $B_k(n)$ car $A_k(n) = 0$ pendant la période calme.

Si $f_{calm}(x_k) = -1$, alors l'algorithme classifie le morceau x_k comme une période non calme. Tant que l'algorithme n'a pas retrouvé des estimés de tous les paramètres α, β, i et j, alors il est impossible de calculer une reconstruction et donc l'algorithme retourne le signal du microphone. Si $f_{calm}(x_k) = -1$ et $\alpha_{k-1} \neq 0$ (estimé de α du morceau x_{k-1}), alors l'algorithme estime la pseudo-inverse en utilisant les paramètres prédits et enregistrés précédemment et calcule la reconstruction des sources.

Si $f_{calm}(x_k) = -1$ et $\alpha_{k-1} = 0$ alors l'algorithme retourne le signal du microphone car l'estimé de α est nul.

Dans le cas où le SVM retourne une prédiction $f_{svm}(x_k) = -1$ et classifie la période directement comme une période active alors l'algorithme estime les inconnues α_k et i_k seulement si $\beta_{k-1} \neq 0$. Si $\beta_{k-1} \neq 0$ alors l'algorithme estime par la suite la pseudo-inverse et calcule la reconstruction des sources $A_k(n)$ et $B_k(n)$ du morceau.

Si $f_{svm}(x_k) = -1$ et $\beta_{k-1} = 0$ alors il est impossible d'estimer la pseudo-inverse car il n'y a pas eu présence de période calme avant. Le diagramme (5.12) décrit le fonctionnement de la méthode ICA/2S/2PM par morceau.

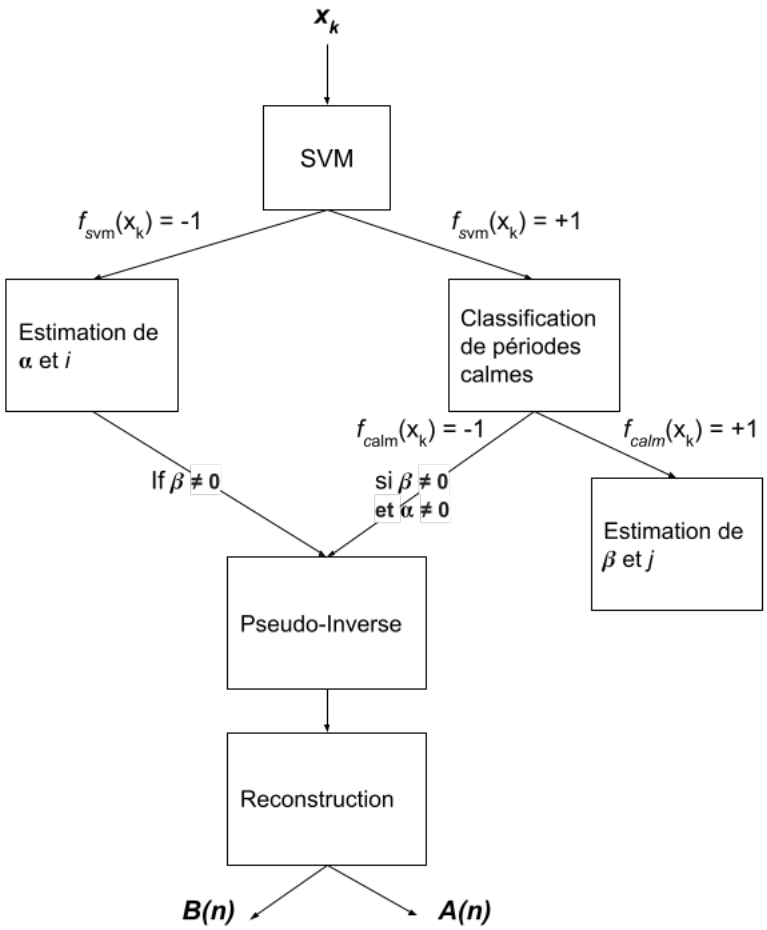

Figure 5.12 Diagramme en flux de ICA/2S/2PM avec délais par morceaux.

5.7 Test de la méthode au complet

Dans cette section nous avons effectué le test de la méthode ICA/2S/2PM avec délais par morceaux au complet. Le signal $x(n)$ utilisé pour entraîner le SVM est l'enregistrement d'un microphone qui représente la voix d'un homme en train de dire une suite de commandes d'intentions comme "Open the window" dans un environnement contenant du bruit rose. L'entraînement du SVM s'est fait par morceaux de longueur $K = 5000$ échantillons et un pas $p = 1000$ échantillons. Pour effectuer le test nous avons utilisé deux parties de l'enregistrement de Fluent.ai. qu'on a superposé à du bruit rose additif en utilisant les paramètres suivants :
- $\alpha = 1.1$, $\beta = 0.9$, $j = 1$, i =-3 pour le morceau entre 0 et 40 000 échantillons.
- $\alpha = 1.2$, $\beta = 0.9$, $j = 1$, i = -4 pour le morceau entre 40 000 et 80 000 échantillons.
- $\alpha = 1.3$, $\beta = 0.9$, $j = 1$, i = -5 pour le morceau entre 80 000 et 150 000 échantillons.

- $\alpha = 1.4$, $\beta = 0.9$, $j = 1$, i =6 pour le morceau entre 150 000 et 200 000 échantillons.
- $\alpha = 1.5$, $\beta = 0.9$, $j = 1$, i =7 entre 200 000 et 250 000 échantillons.

Ces enregistrements permettent de simuler une situation où une personne parle en changeant de position ou bien une situation où plusieurs personnes situés à des endroits différents parlent successivement. Dans les résultats suivants nous avons représenté les estimations enregistrées par l'algorithme ainsi que la reconstruction du signal désiré $A(n)$. Les deux figures (5.13) et (5.14) montrent les résultats obtenus en utilisant l'algorithme ICA2S2PM avec délais au complet. On voit que les estimations des paramètres α, β, i et j pour les différents morceaux étaient correctes et que les reconstructions du signal estimé $A^*(n)$ sont presque identiques au signal source $A(n)$.

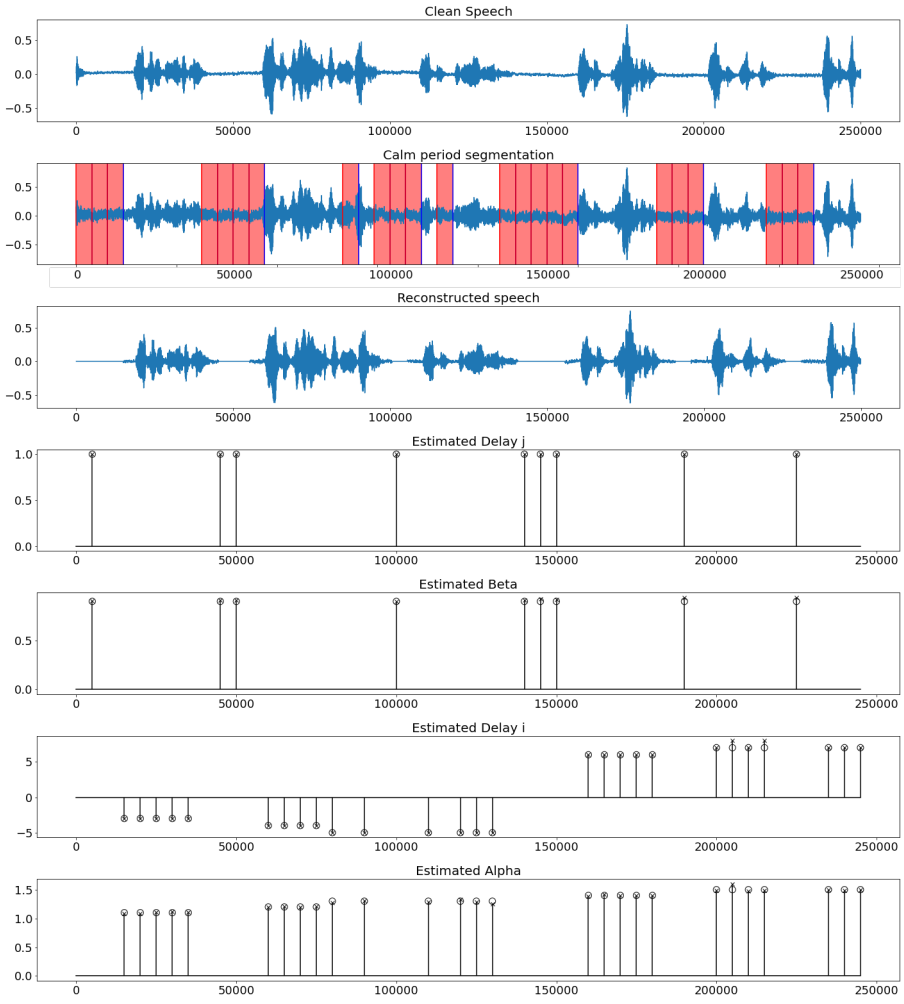

Figure 5.13 Estimés j, β, i, α et reconstruction du signal de parole pour le premier enregistrement.

(L'unité de temps est la période d'échantillonnage)

Les figures (5.13) et (5.14) contiennent chacune sept graphiques au total. Le premier graphique représente le signal de parole $A(n)$ qui ne contient pas de bruit. Le signal de parole représente une suite des signaux d'intention enregistrés par Fluent.ai. Ce signal a une fréquence d'échantillonnage de 16 kHz et a une taille de 250 000 échantillons (équivalant à 15,62 secondes). Le deuxième graphique représente le signal d'un des deux microphones avec la segmentation calculée par le SVM. Le graphique montre des régions en rouge qui correspondent aux morceaux x_k avec $f_{svm}(x_k) = +1$. Les graphiques 4, 5, 6 et 7 représentent les estimés de j_k, β_k, ß$_k$ et α_k calculés pour les morceaux x_k de taille de 5 000 échantillons. Le symbole $\circ$ montre la vraie valeur du paramètre quand on a créé les signaux superposés, le symbole $\times$ montre la valeur de l'estimé du paramètre. Le troisième graphique représente le résultat de la reconstruction du signal de parole $A(n)$ sans le bruit $B(n)$.

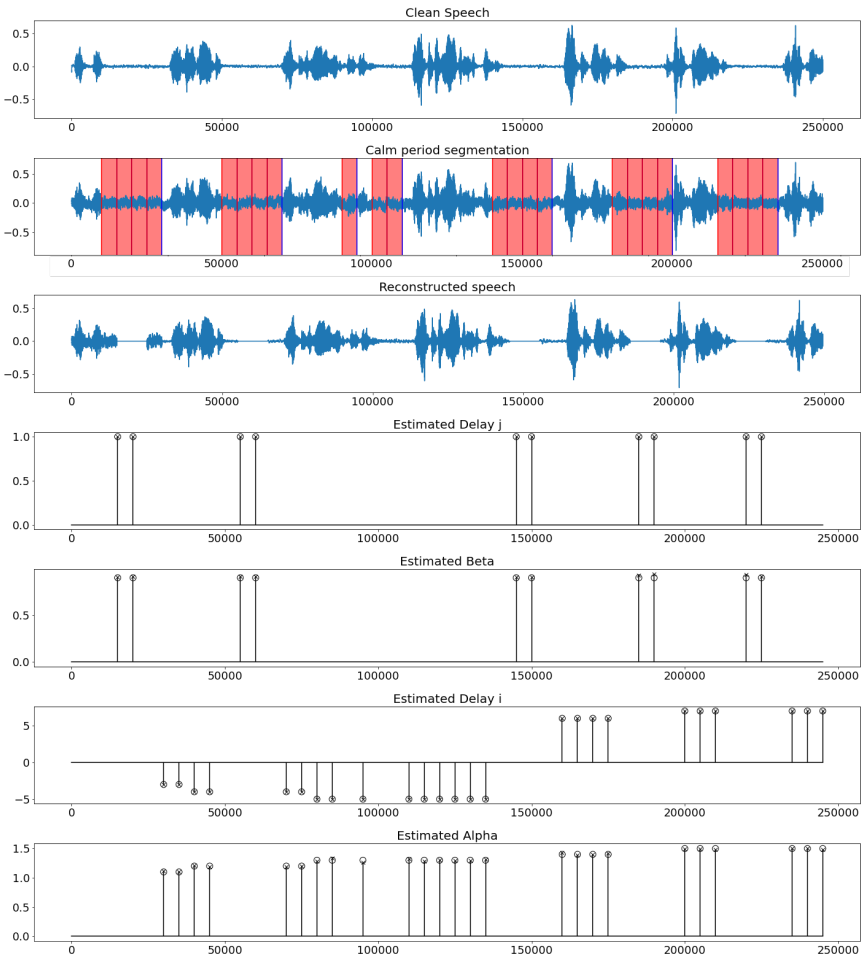

Figure 5.14 Estimés j, β, i, α et reconstruction du signal de parole pour le deuxième enregistrement.

(L'unité de temps est la période d'échantillonnage)

On constate que cette méthode nous permet de séparer parfaitement les signaux $A(n)$ et $B(n)$. Grâce à cette approche, on peut à la fois identifier le signal de parole $A(n)$ et le reconstruire sans le bruit $B(n)$. Cet algorithme a permis de réduire le bruit d'un enregistrement d'une longueur de 15,625 secondes en 3 secondes de temps d'exécution. Pour chaque morceau x_k de longueur de 5 000 échantillons, le temps d'exécution pour calculer les estimés j_k, β_k, ß$_k$ et α_k et calculer les signaux $A_k(n)$ et $B_k(n)$ est de 0.06 seconde.

5.8 Résultats préliminaires pour des signaux réels

Dans le but de tester l'approche ICA/2S/2PM avec délais, nous avons effectué quelques tests en utilisant des signaux réels enregistrés par des microphones. Dans le cas des signaux réels, le bruit n'est pas additionné manuellement au signal de parole. Fluent.ai nous a procuré un enregistrement produit dans une salle comprenant une source de bruit provenant d'un haut parleur placé à une distance des microphones et une source de parole qui provient d'un deuxième haut parleur situé dans un autre endroit. Les signaux des sources de parole et de bruit sont identiques aux signaux utilisés pour créer les superpositions synthétiques dans les tests précédents. Les résultats obtenus en utilisant l'approche ICA/2S/2PM avec délais ont montré que les estimés des délais i et j étaient très volatiles et avaient plusieurs fois une valeur nulle. Les estimés des coefficients α et β avaient une grande variance et les reconstructions étaient très similaires aux signaux des microphones. Cependant ces résultats peuvent s'expliquer par la présence de réverbérations dans les enregistrements. En effet, enregistrer des sources provenant de hauts parleurs dans une salle crée des réverbérations qui viennent fausser les résultats des estimés de l'algorithme ICA/2S/2PM.

CHAPITRE 6 CONCLUSION ET RECOMMANDATIONS

L'objectif principal de ce mémoire est de réduire le bruit des signaux enregistrés par les deux microphones de l'appareil de notre partenaire Fluent.ai. On a présenté les approches permettant de séparer les signaux de microphones en deux signaux sources : le signal de parole d'un locuteur et le signal d'une source de bruit.

6.1 Synthèse des travaux

Nous avons développé deux méthodes pour séparer les signaux des microphones en deux composantes indépendantes. Ces deux méthodes utilisent les corrélations entre les deux signaux des microphones.

Au chapitre 4, nous avons présenté la méthode ICA/2S/2PM pour la séparation des signaux qui arrivent à deux microphones sans délais. On a expliqué comment utiliser la corrélation entre les signaux à des temps différents pour formuler un système d'équations qui nous permet d'estimer les paramètres de mixage. La méthode qu'on a implémentée respecte les suppositions du problème de l'analyse en composantes indépendantes. On a aussi présenté le test de la méthode ainsi que des comparaisons avec les deux approches existantes les plus utilisées. Les tests de vérification nous ont permis de constater que la première méthode fonctionne parfaitement si on utilise des signaux qui ont été reçus par les microphones sans délais. Le test nous a aussi montré que notre méthode présente des résultats meilleurs que les deux autres méthodes étudiées pendant notre revue de litérature. On a aussi remarqué les limitations de ces méthodes quand on a introduit des délais dans les signaux des microphones.

Dans le chapitre 5, on a présenté les différentes parties de l'algorithme qu'on a développé pour la séparation des signaux des microphones quand il y a présence de délais entre les microphones. La méthode utilise aussi la corrélation à deux points similaire à ICA/2S/2PM sans délais. La méthode inclut un algorithme utilisant la machine à vecteur de support qui permet de détecter les périodes calmes. On a aussi réussi à estimer les paramètres de mixage ainsi que les délais grâce à une régression linéaire simple et la méthode d'optimisation de Levenberg-Marquardt.

Dans le chapitre 5, nous avons aussi présenté une adaptation de la méthode permettant de faire la séparation et la reconstruction des signaux par morceaux. Cette adaptation est nécessaire pour un traitement du signal en temps réel. L'approche développée permet d'estimer les paramètres dans un contexte dynamique où les paramètres changent au cours du temps.

On a trouvé que les reconstructions étaient pécises et permettent de séparer clairement le signal de parole du bruit. Cette adaptation a permis d'avoir un temps d'exécution faible. Cette méthode peut fonctionner en temps réel pour faire la réduction de bruit. Pour tester notre méthode, nous avons utilisé des enregistrements fournis par notre partenaire Fluent.ai. Ces données contiennent l'enregistrement d'une personne qui parle sans bruit et un autre enregistrement d'un bruit rose. Les tests ont montré que l'approche qu'on a développé permet de calculer les estimés des paramètres inconnus avec précision. On a aussi montré que l'algorithme permet d'estimer des paramètres qui pourraient changer suite au changement de la distance entre le locuteur et les microphones et ainsi permetterait de tenir compte de la non-stationarité du signal de parole.

6.2 Limitations de la solution proposée

Pour estimer les paramètres de mixage ainsi que les délais, nous avons utilisé des enregistrements superposés synthétiques. Cette méthode de mixage ne permet pas d'avoir des enregistrements qui contiennent de la réverbération. Nos expériences préliminaires avec des signaux contenant de la réverbération montrent que la précision de la séparation se détériore significativement. En particulier, les délais estimés sont souvent nuls. En fait, il existe plusieurs délais associés aux multiples réverbérations reçues par les microphones au cours de l'enregistrement. La présence de réverbération n'a pas été inclue dans la formulation du problème de l'analyse en composantes indépendantes. La réverbération cause un problème lorsqu'on utilise des signaux qui ont été enregistrés par des microphones dans une pièce fermée où le locuteur se tient à distance des deux microphones. Les réverbérations sont créées par la réflexion des signaux de paroles et de bruits sur les murs et les objects de la pièce où se trouve le locuteur.

6.3 Améliorations futures

Dans le cadre d'améliorations futures, on pourrait modifier la formulation du problème de l'analyse en composantes indépendantes pour inclure les réverbérations dans notre système d'équations. Une autre possibilité serait de réduire la réverbération avec un algorithme approprié avant d'appliquer notre méthode.

RÉFÉRENCES

[1] R. Penrose, "A generalized inverse for matrices," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 51, n°. 3, p. 406–413, 1955.

[2] L. Lugosch, M. Ravanelli, P. Ignoto, V. Tomar et Y. Bengio, "Speech model pre-training for end-to-end spoken language understanding," 04 2019.

[3] E. C. Cherry, "Some experiments on the recognition of speech, with one and with two ears," *The Journal of the acoustical society of America*, vol. 25, n°. 5, p. 975–979, 1953.

[4] M. Rana, M. M. Rahman et M. Hasnain, "Comparison study between independent component analysis and principle component analysis in the context of hidden source separation," *World Journal of Pharmaceutical Research*, vol. 7, p. 178–196, 09 2018.

[5] J. Hérault et B. Ans, "Réseau de neurones à synapses modifiables : décodage de messages sensoriels composites par apprentissage non supervisé et permanent," 1984.

[6] A. Hyvärinen et E. Oja, "Independent component analysis : algorithms and applications," *Neural Networks*, vol. 13, n°. 4, p. 411–430, 2000. [En ligne]. Disponible : https://www.sciencedirect.com/science/article/pii/S0893608000000265

[7] J.-T. Chien, "Chapter 4 - independent component analysis," dans *Source Separation and Machine Learning*, J.-T. Chien, édit. Academic Press, 2019, p. 99–160. [En ligne]. Disponible : https://www.sciencedirect.com/science/article/pii/B9780128045664000164

[8] A. Hyvärinen, "Survey on independent component analysis," 1999.

[9] A. Hyvarinen, "Fast and robust fixed-point algorithms for independent component analysis," *IEEE transactions on Neural Networks*, vol. 10, n°. 3, p. 626–634, 1999.

[10] A. J. Bell et T. J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," *Neural computation*, vol. 7, n°. 6, p. 1129–1159, 1995.

[11] T. Yamaguchi et K. Itoh, "An algebraic solution to independent component analysis," *Optics Communications*, vol. 178, n°. 1, p. 59–64, 2000. [En ligne]. Disponible : https://www.sciencedirect.com/science/article/pii/S0030401800006428

[12] P. Marmaroli, X. Falourd et H. Lissek, "A comparative study of time delay estimation techniques for road vehicle tracking," 04 2012.

[13] F. Reed, P. Feintuch et N. Bershad, "Time delay estimation using the lms adaptive filter–static behavior," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, n°. 3, p. 561–571, 1981.

[14] T. Evgeniou et M. Pontil, “Support vector machines : Theory and applications,” vol. 2049, 01 2001, p. 249–257.

[15] T. Kinnunen, E. Chernenko, M. Tuononen et H. Li, “Voice activity detection using mfcc features and support vector machine,” vol. 2, 03 2012.

[16] J. Dey, M. S. B. Hossain et M. Haque, “An ensemble svm-based approach for voice activity detection,” 02 2019.

[17] T. Giannakopoulos, “pyaudioanalysis : An open-source python library for audio signal analysis,” *PLOS ONE*, vol. 10, n°. 12, p. 1–17, 12 2015. [En ligne]. Disponible : https://doi.org/10.1371/journal.pone.0144610

[18] T. Giannakopoulos et A. Pikrakis, “Introduction to audio analysis,” dans *Introduction to Audio Analysis*, T. Giannakopoulos et A. Pikrakis, édit. Oxford : Academic Press, 2014, p. i. [En ligne]. Disponible : https://www.sciencedirect.com/science/article/pii/B9780080993881000091