REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Ecole Nationale Polytechnique
Département d'Electronique

End-of-studies project dissertation in partial fulfilment of the requirements for the State Engineer Degree in Electronics

Blind Speech Separation: Algorithm Improvement and Implementation using Raspberry Pi with UMA-8-SP Mic Array Testbed

Realized by:
Lynda *BERRAH*
Nacira *MENDJEL*

Under the supervision of:
Pr. Adel *BELOUCHRANI*
Dr. Soufiane *TEBACHE*

Publicly presented and defended on June 30^{th}, 2022

Composition of the Jury:

President	Mr. Sid-Ahmed *BERRANI*	PhD.	ENP
Examiner	Mr. Abdelouahab *BOUDJELLAL*	PhD.	EMP
Supervisor	Mr. Adel *BELOUCHRANI*	Prof.	ENP
Supervisor	Mr. Soufiane *TEBACHE*	PhD.	LDCCP/ENP
Guest member	Mr. Kamel *REMILI*	Magister	
Guest member	Mr. Karim *ABED-MERAIM*	Prof.	Polytech Orléans

ENP 2022

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Ecole Nationale Polytechnique
Département d'Electronique

End-of-studies project dissertation in partial fulfilment of the requirements for the State Engineer Degree in Electronics

Blind Speech Separation: Algorithm Improvement and Implementation using Raspberry Pi with UMA-8-SP Mic Array Testbed

Realized by:
Lynda *BERRAH*
Nacira *MENDJEL*

Under the supervision of:
Pr. Adel *BELOUCHRANI*
Dr. Soufiane *TEBACHE*

Publicly presented and defended on June 30^{th}, 2022

Composition of the Jury:

President	Mr. Sid-Ahmed *BERRANI*	PhD.	ENP
Examiner	Mr. Abdelouahab *BOUDJELLAL*	PhD.	EMP
Supervisor	Mr. Adel *BELOUCHRANI*	Prof.	ENP
Supervisor	Mr. Soufiane *TEBACHE*	PhD.	LDCCP/ENP
Guest member	Mr. Kamel *REMILI*	Magister	
Guest member	Mr. Karim *ABED-MERAIM*	Prof.	Polytech Orléans

ENP 2022

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Ecole Nationale Polytechnique
Département d'Electronique

Mémoire de Projet de Fin d'Etudes en vue de l'obtention du diplôme d'Ingénieur d'Etat en Electronique

Séparation Aveugle de Signaux Vocaux : Amélioration de l'Algorithme et Implémentation à l'aide d'un Raspberry Pi et d'un Réseau de Capteurs UMA-8-SP

Réalisé par :
Lynda *BERRAH*
Nacira *MENDJEL*

Sous la direction de :
Pr. Adel *BELOUCHRANI*
Dr. Soufiane *TEBACHE*

Présenté et soutenu publiquement le: 30/06/2022

Composition du Jury:

Président	M. Sid-Ahmed *BERRANI*	Docteur	ENP
Examinateur	M. Abdelouahab *BOUDJELLAL*	Docteur	EMP
Promoteur	M. Adel *BELOUCHRANI*	Professeur	ENP
Promoteur	M. Soufiane *TEBACHE*	Docteur	LDCCP/ENP
Invité	M. Kamel *REMILI*	Magistère	
Invité	M. Karim *ABED-MERAIM*	Professeur	Polytech Orléans

ENP 2022

ملخص

في العالم الحقيقي ، لا تسجل الميكروفونات إشارة الكلام المستهدفة فحسب ، بل تسجل أيضا المصادر الأخرى ، والتأثيرات الصوتية للغرفة ، وضوضاء الخلفية. ومن ثم ، فإن استخراج الكلام المستهدف من الخلائط الصاخبة أمر مرغوبٍ فيه للغاية للعديد من التطبيقات. يهدف هذا العمل إلى المعالجة الأعمى للكلام . أولاً ، قمنا بدراسة ومقارنة ثلاث خوارزميات: IVA **و** Fast IVA **و** ILRMA. **بعد ذلك ، عملنا على تحسين أداء هذه الخوارزميات باستخدام عمليتين مختلفتين: تقليل الضوضاء الصوتية ومعادلة** SIMO. **تظهر النتائج تحسنا كبيرا في الأداء. أخيراً ، تم تنفيذ الفصل على نظام مضمن واختباره.**

الكلمات الرئيسية: فصل الكلام الأعمى، IVA، Fast IVA، ILRMA، SIMO **معادلة وتقليل الضوضاء والأنظمة المدمجة.**

Résumé

Dans un environnement réel, les microphones n'enregistrent pas seulement le signal de parole cible mais aussi d'autres sources indésirables, les effets acoustiques de la pièce et le bruit de fond. Par conséquent, extraire le signal cible à partir de mélanges convolutifs bruyants est hautement souhaitable pour de nombreuses applications. Ce travail a pour but de traiter la séparation aveugle des signaux parole. Tout d'abord, nous avons étudié et comparé trois algorithmes de séparation aveugle de sources: IVA, Fast IVA, et ILRMA. Ensuite, nous avons travaillé sur l'amélioration des performances de ces algorithmes en utilisant deux post-traitements différents : le débruitage et l'égalisation SIMO. Les résultats montrent une amélioration significative des performances. Enfin, le schéma de séparation sélectionné a été implémenté sur système embarqué et testé sur des signaux réels.

Mots clés: Séparation aveugle de parole, IVA, Fast IVA, ILRMA, égalisation SIMO, débruitage et système embarqué.

Abstract

In a real-world environment, microphones record not only the target speech signal but also other available sources, the room acoustic effects, and background noise. Hence, extracting target speech from noisy convolutive mixtures is highly desirable for many applictions. This work aims to address the convolutive blind source separation of speech signals. First, we studied and compared three frequency-domain blind speech separation algorithms: IVA, Fast IVA, and ILRMA. Then, we worked on improving the performances of these algorithms using two different post-processings: speech denoising and SIMO equalization. The results demonstrate a significant improvement in performance. Finally, the selected separation scheme was implemented on an embedded system and tested on real-world signals.

Keywords: Blind Speech Separation, IVA, Fast IVA, ILRMA, SIMO equalization, denoising and embedded systems.

Dedication

I dedicate this work,
To my lovely mother and father, words cannot express my endless gratitude for their constant encouragement, attention, and prayers throughout my educational career. I have been truly blessed to have them as parents. They have provided me with unending love and support throughout the years and encouraged me to follow my passion and go after my dreams.
To my sisters: Alycia, Flora, Sofia, and Amylia, for their immense support and for always being there for me.
To my beloved ones: Azzeddine, Dihia, and Lysa for their support and encouragement.
To all my teachers, in the electronics department of the Ecole Nationale Polytechnique.
And to all my classmates, who have made these 3 years a special experience for me.
To all my friends I met at every stage of my education and my life.
To my work partner Lynda, for sharing this work and its difficulties, ups, and downs with me. Thank you for your unique motivation and determination to make this work valuable. I could not have wished for a better work partner.
This work is for all those who supported me.

Nacira

Dedication

I dedicate this work
To my dear parents, whom I love with all my heart, for all their sacrifices, love, support, and prayers throughout my studies.
To my dear sisters, Naila and Nesrine, for their support and encouragement.
To my nieces and nephews, Yasmine, Wassim, and Meriem, what would my life be without you!
To the memory of my grandparents, may God welcome them into his vast paradise, I hope I have made you proud.
To my aunts, uncles, Kamel, Fayçal, Lamia and all my family for their support throughout my university career.
To my beloved ones: Ramzi, Aicha, Chiraz, Lyna, Malika, Maya, Sabrina, Hynd, Sarah, Roza, Nadir, and Hamza.
To all my classmates and to all the students of the department of electronics in particular Aziz, Malik, Ahmed, Mohammed, Raouf, Anes, Riad, Massylia, and Samy.
To all the teachers of the National Polytechnic School.
To Nacira, my coworker, for sharing this work and its difficulties with me. Thank you for your patience, perseverance, devotion, and determination to achieve the best work possible. I couldn't have asked for a more ideal partner.
And to all those who have contributed in some way to make me the person I am today.

Lynda

Acknowledgement

We would like to express our gratitude to everyone who helped to realize this work, beginning with our dear parents, without whose sacrifices we would not be where we are today.

We express our sincere gratitude to our promoters, Adel BELOUCHRANI, Professor at the Ecole Nationale Polytechnique and a model of excellence as a researcher, advisor, and instructor, and Soufiane TEBACHE, Doctor at the LDCCP laboratory of the Ecole Nationale Polytechnique, for their rigorous supervision and motivating encouragement. We would like to thank them for their advice, which was crucial to the achievement of this work. We also thank Mr. ABDI Rabah, who planned the purchase of all the equipment used in our project.

We would also like to thank Mr. Sid-Ahmed BERRANI for having accepted to chair our jury, Mr. Abdelouahab BOUDJELLAL, our examiner, for his interest in our work, Mr. Kamel REMILI and Mr. Karim ABED-MERAIM for having honored us with their presence and giving their time.

Particular thought is also addressed to all the teachers and students of the electronics engineering department, with whom we shared three wonderful years.

Contents

List of Tables

List of Figures

List of Acronyms

AMUSE	A Minimally-Unsatisfiable Subformula Extractor
BSI	Blind System Identification
BSS	Blind Source Separation
CBSS	Convolutive Blind Source Separation
CR	Cross-Relation
DFT	Discrete Fourier Transform
EVD	EigenValue Decomposition
FDBSS	Frequency-Domain Blind Source Separation
FDICA	Frequency-Domain Independent Component Analysis
FIR	Finite Impulse Response
Fast IVA	Fixed-point/Fast Independent Vector Analysis
GGD	Generalized Gaussian Distribution
HOS	Higher-Order Statistics
ICA	Independent Component Analysis
IDFT	Inverse Discrete Fourier Transform
ILRMA	Independent Low-Rank Matrix Analysis
IS	Itakura-Saito divergence
ISM	Image Source Method
ISTFT	Inverse Short-Time Fourier Transform
IVA	Independent Vector Analysis
KLD	Kullback-Leiber divergence
LMS	Least Mean Square
Log-MMSE	Minimum Mean Square Estimators Log Spectral Amplitude
LS	Least Squares
MCFLMS	Multi-Channel Frequency-domain LMS
MDP	Minimal Distortion Principle
ML	Maximum Likelihood

MSE	Mean-Square Error
NMCFLMS	Normalized Multi-Channel Frequency-domain LMS
NG	Natural Gradient
NMF	Non-Negative Matrix Factorization
PCA	Principle Component Analysis
PDF	Probability Density Function
RIR	Room Impulse Response
RNMCFLS	Noise Robust NMCFLMS
SGD	Super-Gaussian Distribution
SIMO	Single-Input Muliple-output
SIRP	Spherically Invariant Random Processes
SOS	Second-Order Statics
SOBI	Second Order Blind Identification
SS	Source Separation
SSL	Spherically Symmetric Laplace distribution
STFT	Short-Time Fourier Transform
T-F	Time-Frequency
VSS-MCFLMS	Variable Step Size MCFLMS

List of symbols

Scalar variables are denoted by plain letters, (e.g. x), vectors by bold-face lower-case letters, (e.g. $\mathbf{x}$), and matrices by bold-face upper-case letters, (e.g., $\mathbf{X}$). In this document, the following notations are used:

$\mathbb{R}$	Set of real numbers
$\mathbb{C}$	Set of complex numbers
$\|.\|$	Absolute value
$\|\|.\|\|_2$	Euclidean norm
$(.)^*$	Complex conjugate operator
$(.)^T$	Transpose operator
$(.)^H$	Hermitian operator
$(.)^-1$	Inverse operator
$(.)^\#$	Pseudo-inverse operator
$(.)^*$	Complex conjugate operator
$det(.)$	Matrix determinant operator
$E(.)$	Statistical expectation operator
$H(.)$	Entropy function
$\nabla\ f$	Gradient operator of the function f
$\circ$	Hadamard product
$\circledast$	Linear convolution operator
M	Number of microphones
L	Number of sources
F	Number of frequency bins in the time-frequency representation
N	Number of time frames in the time-frequency representation
$\mathbf{x}$	Mixtures in the time domain
$\mathbf{s}$	Original source signals in the time domain
$\mathbf{y}$	Estimated sources in the time domain
$\mathbf{y}_{tf_l}$	l^{tf} output signal in the T-F domain

$\boldsymbol{\varphi}^f(\mathbf{y}_{tf_l})$	Score function at frequency bin f
$\mathbf{I}_N$	$N \times N$ identity matrix
$\mathbf{D}$	Diagonal matrix
$\mathbf{P}$	Permutation matrix
$\mathcal{C}$	Contrast function
$\mathbf{A}$	Mixing matrix
$\mathbf{W}$	Demixing/Unmixing matrix
$\mathbf{Rxx}$	Autocorrelation matrix
$\mathbf{Q}$	Whitening matrix
$\mathbf{T}$	Basis matrix in the NMF decomposition
$\mathbf{V}$	Activation matrix in the NMF decomposition
$\mathbf{h}$	Channel's impulse response
$\mathbf{g}$	Equalization filter

Introduction

Signal processing techniques are applied for data acquisition, analysis, and transmission chain. As a result, these methods have applications in almost every technology area, particularly the audio field, where the goal is to achieve the best sound quality.

Many systems in this field work quite well when there is only one source and almost no echo. However, their performance degrades in highly reverberant environments or when several speakers talk simultaneously. Since most audio signals found in the real world are mixtures to which several sources contribute, it would be highly desirable to separate audio signals. This problem is termed the cocktail party problem.

The cocktail party problem can be formulated as follows: "How can you understand what your neighbor is saying over the other voices, music, and background noise? "

Human has an incredible ability to focus on a particular sound of interest in the presence of many unwanted and distracting sounds and cancel out the other sounds. Human hearing performs selective listening in this case based on the spatial and spectro-temporal characteristics of the sound sources present. Additionally, it takes into account prior knowledge such as learned features of speech and language as well as the source spatial position provided by vision.

Much effort over the past decades has been devoted to understanding the capabilities of humans. The aim of these studies is to mimic this behavior onto an artificial system for source separation. However, the performance of the machines is poor compared to human performance.

In this work, we consider the problem of separating audio sources in a reverberant environment where mixtures are recorded from several microphones. In the following, we present an overview of the different chapters of this thesis.

- In the first chapter, we present the source separation problem by explaining its principle and giving its mathematical model.

- In the second chapter, we give a structured presentation of several blind source separation methods, focusing in particular on the frequency-domain speech source separation techniques.

- Then, in chapter 3, we present some post-processings that aim to improve the separation quality.

- In chapter 4, we present the methods and tools used for the data generation and the evaluation of the algorithms.

- In chapter 5, we propose an analysis of the performance of all the algorithms studied using synthetic signals. After this step, a separation algorithm has been selected for implementation.

- In chapter 6, this chosen algorithm is tested using real-world recorded signals. Then, the algorithm is implemented using Raspberry Pi and a UMA-8-SP microphone array testbed.

- We end our report with a conclusion that summarises the contributions of our work and opens up a set of perspectives on the audio source separation in particular and in source separation in general.

Chapter 1

Background and Related Literature Survey

Hearing aids, automatic speech recognition, human-machine interaction, and many other systems work effectively when there is only one source, but their performance degrades when multiple voices are present simultaneously. Therefore, it is highly desirable to separate the source signals before performing audio processing. This problem is known as the cocktail party problem. A promising technique to solve it is blind source separation (BSS). This first chapter presents an overview of BSS, the general mathematical concepts required to understand the following chapters, and a summary of the current state of the art.

1.1 Cocktail Party Problem

The cocktail party effect occurs when sounds from different sources mix in the air before reaching the ear. Consider trying to carry on a conversation with another person or a group of people at a cocktail party while there are a variety of sounds coming from the environment: speech, music, and even a whistle from outside the window (Yu et al., 2014b). Because of the amazing capability of the human auditory system, you will be able to distinguish the sounds and focus your attention on one speaker with ease.

Over the past few decades, several studies have been conducted to mimic this human auditory ability. Colin Cherry (Cherry, 1953) was the first to introduce this problem as the cocktail party problem and described it as a problem in which multiple human speakers are talking simultaneously, each speaker's voice must be isolated (separated) from the other present sounds, similar to the way how the human sensory system can identify and listen to individual speakers in a crowded party.

1.2 Blind Source Separation

Source separation refers to a set of problems aimed at separating individual source components from their observed mixtures. Blind source separation (BSS) refers to a powerful technique of separating mixtures of sources. These mixtures are obtained from a set of sensors, each of which receives different mixtures of the source signals since their position is not the same. The term "blind" implies that neither the sources nor the mixing parameters are known. There is only a small amount of prior knowledge, such as the statistical independence of the source signals (Comon and Jutten, 2010).

Over the past decades, enormous progress has been achieved in the field of BSS, which has emerged as one of the most promising and exciting topics, with solid theoretical foundations, in the domains of neural computing, advanced statistics, and signal processing. BSS has been employed effectively in various fields, including speech recognition, cocktail party problems, image processing, remote sensing, communication systems, exploration seismology, geophysics, econometrics, data mining, and neural networks (Gao, 2011).

1.3 Mixtures and Separation Models

The BSS problem can be stated as an estimation of L unknown source signals from mixtures that are unknown functions of the original sources.

Suppose we use M microphones to record the mixtures. At a particular time t, we have M observed signals $x_1(t), ..., x_M(t)$ which are assumed to be the mixtures of L independent source signals $s_1(t), ..., s_L(t)$. Then, $\mathbf{x}(t) = [x_1(t), x_2(t), ..., x_M(t)]^T \in \mathbb{R}^{M \times 1}$ and $\mathbf{s}(t) = [s_1(t), s_2(t), ..., s_L(t)]^T \in \mathbb{R}^{L \times 1}$ are called *the observation vector* and *the source vector*, respectively, where $(.)^T$ represents the transpose operator. Hence, $\mathbf{x}(t) = \mathcal{F}(\mathbf{s}(t))$, where $\mathcal{F}$ is the unknown *mixing system.*

The goal of BSS is to construct a *separating system* $\mathcal{G}$, in order to retrieve unobserved mixed signals, $\mathbf{y}(t) = \mathcal{G}(\mathbf{x}(t))$, where $\mathbf{y}(t) = [y_1(t), y_2(t), ..., y_L(t)]^T \in \mathbb{R}^{L \times 1}$ represents *the estimates of the source signals* (output vector of the BSS algorithm).

The blind source separation task depends strongly on the characteristics of the sources and the way in which they are mixed within the physical environment. In order to choose an appropriate BSS method, several parameters must be considered.

First, the number of observations M compared to the number of sources L is an

important parameter in BSS algorithms. The mixture is said to be *under-determined* when it contains more sources than sensors (L>M); *over-determined* when it contains fewer sources than sensors (L < M); *determined* when the number of sources is equal to the number of sensors (L = M) (Comon and Jutten, 2010, Ch. 4, p. 108). In most studies, the determined case is considered. The over-determined case can be converted to the determined one using dimensionality reduction techniques such as Principle Component Analysis (PCA)(Bro and Smilde, 2014). However, the under-determined case can only be solved with significant prior knowledge about the sources.

Secondly, mixtures can be either *linear* or *non-linear*. The case of a non-linear mix occurs when, for example, the microphones have a non-linear behavior. This non-linearity of the mixing system $\mathcal{F}$ is often negligible compared to the intensity of the recorded data. Therefore, the linear model (the observed data is a linear combination of the sources) is generally considered. In the remainder of this document, we will consider the case of linear mixtures.

Finally, the mixing can be either *instantaneous* or *convolutive*. In the case of instantaneous mixing, at each instant t, the observations are linear combinations of the sources at the same instant t. The algorithms designed for this model, while valuable for narrowband signals, have limited practical applicability in speech separation problems, since the latter are wideband signals with respect to the propagation channel. When BSS is used to solve the cocktail party problem, in a real reverberant environment, mixtures of audio sources are convolutive rather than instantaneous due to propagation delays (Douglas and Gupta, 2007). This means that the microphones capture not only the direct source signals, but also the attenuated and delayed versions of the source signals reflected from the walls and ceiling.

1.3.1 Instantaneous Linear Mixing

The noiseless instantaneous linear mixing model is defined as:

$$x_m(t) = \sum_{l=1}^{L} a_{ml} s_l(t) \quad m = 1, .., M \tag{1.1}$$

Where $x_m(t)$ is the m^{th} element of the mixture vector, $s_l(t)$ is the l^{th} element of the source vector, and a_{ml} are the coefficients of the linear time-invariant mixing system represented by the $M \times L$ matrix $\mathbf{A}$, called *the mixing matrix*. In matrix form:

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) \tag{1.2a}$$

$$= \sum_{l=1}^{L} \mathbf{a}_l s_l(t) \tag{1.2b}$$

Where $\mathbf{a}_l$ is the l^{th} column of matrix $\mathbf{A}$. The goal of BSS for instantaneous mixtures is to adjust the coefficients of an $L \times M$ *separation* or *unmixing matrix* $\mathbf{W}$ such that:

$$y_l(t) = \sum_{m=1}^{M} w_{lm} x_m(t) \quad l = 1, .., L \tag{1.3}$$

Where $y_l(t)$ represents an estimate of a single original source and w_{lm} are the entries of matrix $\mathbf{W}$. In matrix form:

$$\mathbf{y}(t) = \mathbf{W}\mathbf{x}(t)$$

$$= \sum_{m=1}^{M} \mathbf{w}_m x_m(t) \tag{1.4}$$

Where $\mathbf{w}_m$ is the m^{th} column of matrix W.

1.3.2 Convolutive Mixing

In the case of acoustics sources, the mixed signals are a linear mixture of filtered versions of each of the source signals. The noiseless convolutive linear mixing model is given by:

$$x_m(t) = \sum_{l=1}^{L} \sum_{p=0}^{P-1} a_{ml}(p) s_l(t-p) \quad m = 1, .., M \tag{1.5}$$

$$\mathbf{x}(t) = \sum_{p=0}^{P-1} \mathbf{A}(p)\mathbf{s}(t-p) \tag{1.6}$$

Where $\mathbf{A}$(p), $p = 0, ..., (P-1)$, is the $M \times L$ *transfer function matrix/ multichannel FIR filter representing the room impulse response (RIR)*, whose elements are denoted $a_{ml}(p)$, $\mathbf{s}$(t) is the source vector, and P is *the mixing filter length in time*, i.e. the number of samples that represents the delay and reverberations in a real-room situation. The p^{th} slice of the mixing filter $\boldsymbol{A}$(p) is:

$$\mathbf{A}(p) = \begin{bmatrix} a_{11}(p) & \dots & a_{1L}(p) \\ \vdots & \ddots & \vdots \\ a_{M1}(p) & \dots & a_{ML}(p) \end{bmatrix} \tag{1.7}$$

In time domain Convolutive BSS, the sources are estimated using a set of inverse filter matrices $\mathbf{W}$(q), q = 0, . . . , (Q -1) such that:

$$y_l(t) = \sum_{m=1}^{M} \sum_{q=0}^{Q-1} w_{lm}(q) x_m(t-q) \quad l = 1, .., L \tag{1.8}$$

$$\mathbf{y}(t) = \sum_{q=0}^{Q-1} \mathbf{W}(q)\mathbf{x}(t-q) \tag{1.9}$$

Where $w_{lm}(q)$ represents *the separating filter coefficient* from the mixture m to the output source l and Q is *the unmixing filter length in time.* The q^{th} slice of the unmixing filter $\mathbf{W}$(q) is:

$$\mathbf{W}(q) = \begin{bmatrix} w_{11}(q) & \dots & w_{1M}(q) \\ \vdots & \ddots & \vdots \\ w_{L1}(q) & \dots & w_{LM}(q) \end{bmatrix} \tag{1.10}$$

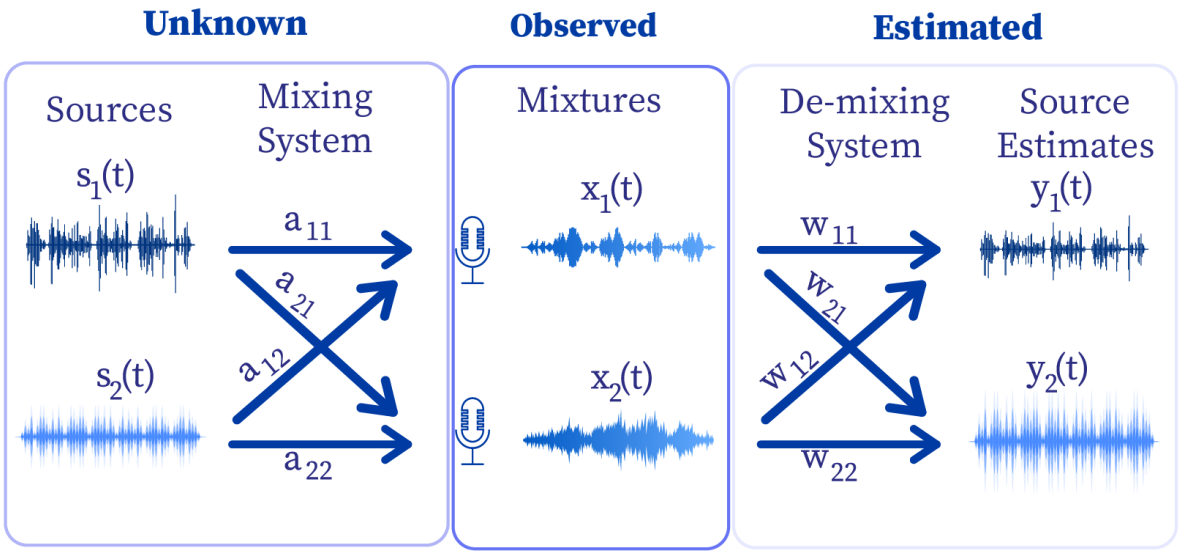

Figure 1.1: Mixing and unmixing system in the case of instantaneous mixtures

1.4 Ambiguities

BSS algorithms assume as little knowledge as possible of the source signals and the mixing matrix $\mathbf{A}$. This lack of prior information leads to several ambiguities regarding the possible solutions provided by a BSS algorithm. Indeed, it is not always possible to uniquely identify the original source signals. Instead, it is only feasible to recover the sources and the mixing matrix up to certain indeterminacies.

1.4.1 Scale Ambiguity

We cannot determine the variances (or energies) of the recovered independent sources. The reason is that the mixing matrix $\mathbf{A}$ and the sources $\mathbf{s}(t)$ are both unknown. Thus, any non-zero scalar multiplier α in one of the sources $s_l(t)$ could be canceled by dividing the corresponding column $\mathbf{a}_l$ of $\mathbf{A}$ by the same scalar without changing the observations.

$$\mathbf{x}(t) = \sum_{l=1}^{L} (\frac{1}{\alpha}\mathbf{a}_l)(\alpha s_l(t)) \tag{1.11}$$

This shows that the sources can only be estimated up to a scaling constant.

1.4.2 Permutation Ambiguity

We cannot determine the order of the recovered independent sources. This is because reordering the sources and columns of the mixture matrix accordingly leaves the observations unchanged. Formally, a permutation matrix $\mathbf{P}$, which is a square binary matrix that has one entry of 1 in each row and each column, and its inverse $\mathbf{P}^{-1} = \mathbf{P}$ can be substituted into the model to give:

$$\mathbf{APPs}(t) \tag{1.12}$$

Where $\mathbf{Ps}(t)$ contains the original signals $\mathbf{s}(t)$ but in a different order, and the matrix $\mathbf{AP}$ is just a mixing matrix with permuted columns. This implies that the sources can only be recovered up to a permutation.

1.5 Frequency-Domain Convolutive BSS

According to their processing domain, the major approaches to solving the convolutive blind source separation problem fall into two main categories: the so-called temporal methods, which perform the separation in the time domain, and the so-called frequency methods, which operate in the time-frequency domain.

The first attempts to address the separation of convolutive mixtures used time-domain methods, in which the BSS algorithms are directly applied to the mixture model. Once the algorithm converges, this approach effectively separates the sources. However, the computational cost associated with estimating the filter coefficients for the deconvolution operation can be very high, especially when dealing with reverberant mixtures using filters with long delays. Indeed, the convergence rate decreases when the channel order increases.

As we know from the basics of signal processing, convolving in the time domain is equivalent to multiplying in the frequency domain. Therefore, to overcome the computational complexity of the source separation of convolutive mixtures in the time domain, the frequency domain methods convert the time mixing into the time-frequency domain using an appropriate transformation so that temporal convolution is converted into simple multiplications. Thus, the problem of separating convolutive mixtures is reduced into several independent complex-valued instantaneous BSS problems. Then, several well-established instantaneous BSS algorithms can be applied to each frequency bin separately, which greatly simplifies the separation algorithm.

1.5.1 Time-Frequency Representation

When applying convolutive BSS (CBSS) to speech signal mixtures, a long multi-channel finite impulse response (FIR) is used to achieve separation since these signals are characterized by the fact that all their properties, such as amplitude, frequency, and phase, change over time. It would be interesting to consider the use of a time-frequency representation, which is much more practical.

The transformation of time-domain signals into the time-frequency domain is usually performed via the short-time Fourier transform (STFT), which will be described hereafter.

1.5.1.1 Short-Time Fourier Transform

The Short-Time Fourier Transform (STFT) is the simplest and most commonly used time-frequency representation. It consists of applying a moving window to the signal, then performing the Discrete Fourier Transform (DFT) of the signal within the window as the window moves. This enables us to relax the assumption of stationarity made when calculating the DFT since it analyzes frequency content only over short intervals. The STFT of a signal is computed as follows:

1. In the first stage of the STFT analysis, the input signal is segmented into fixed-length frames of short duration, and each frame of the segmented signal is multiplied with an appropriate window function $w_a(t)$. The values of the window functions $w_a(t)$ are zero outside the interval $t \in [0, T-1]$, where T is the number of samples in a frame. Two commonly-used windows are the rectangular window, which truncates the signal segment, and the Hamming window, which applies a taper to the ends to avoid unnatural discontinuities in the speech segment.

$$\mathbf{x}_n(t) = \mathbf{x}(t + \tau_0 + nR).w_a(t) \quad n = 0, ..., N-1, \;\; t = 0, ..., T-1 \tag{1.13}$$

   Where: the variable n, referred to as a time frame, represents the frame index of the input signal, N is the number of time frames, τ_0 is the positions of the first sample of the first frame, and the variable R denotes the hop size, which represents the number of time advances from one frame to the next one in samples.

2. After windowing, the DFT of each windowed segment $\mathbf{x}_n(t)$ is taken, resulting in complex-valued STFT coefficients.

$$\mathbf{x}_{tf}(f, n) = \sum_{t=0}^{T-1} \mathbf{x}_n(t) exp(\frac{-2j\pi tf}{F}) \qquad f = 0, ..., F-1 \tag{1.14}$$

   The variable f, referred to as a frequency bin, is a frequency index, F is the number of frequency bins, and j is the imaginary unit.

   F must be a power of 2 and is usually equal to T, the number of samples in a frame.

3. If F is larger than the frame length T, we have to extend $\mathbf{x}_n(t)$ with zeros on both sides before applying the DFT.

As a result of applying the STFT to equation (1.6), the linear convolution in the time domain can be written in the frequency domain as separate multiplications for each frequency bin as:

$$\mathbf{x}_{tf}(f,n) = \mathbf{A}(f)\mathbf{s}_{tf}(f,n) \tag{1.15}$$

Where: $\mathbf{x}_{tf}(f,n) = [x_{tf_1}(f,n), ..., x_{tf_M}(f,n)]^T$ whose elements $x_{tf_m}(f,n)$ are the $(f,n)^{th}$ element of the T-F representations $\mathbf{X}_{tf_m}$ of the microphone signals $x_m(t)$, $\mathbf{s}_{tf}(f,n) = [s_{tf_1}(f,n), ..., s_{tf_L}(f,n)]^T$ whose elements $s_{tf_l}(f,n)$ are the $(f,n)^{th}$ element of the T-F representations $\mathbf{S}_{tf_m}$ of the source signals $s_l(t)$.

The matrix $\mathbf{A}(f)$ is the $(M \times L)$ mixing matrix at frequency bin f.

For a particular frequency bin f, (1.15) represents an instantaneous mixing system.

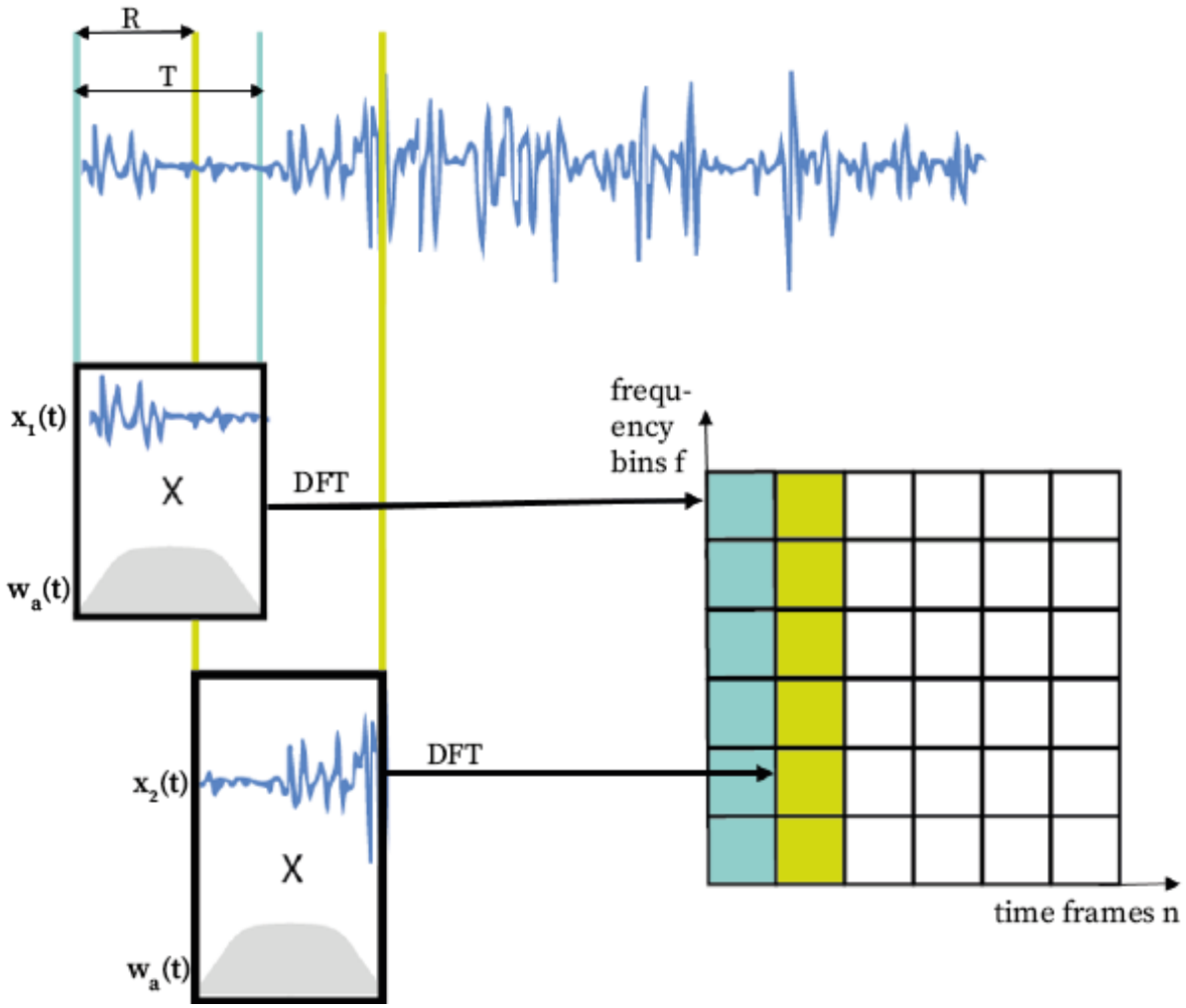

Figure 1.2: Short Time Fourier Transform Analysis

It is worthwhile to note that the previous equation (1.15) is considered to be valid only for signals $\mathbf{s}(t)$ that permit Fourier Transforms. Also, it is approximately valid if the time-convolution is circular (Albataineh and Salem, 2021). However, in practice, ensuring that the time convolution is circular requires making the Fourier transform length significantly larger than the maximum length of the FIR filter.

1.5.1.2 Inverse Short-Time Fourier Transform

Once the process of source separation is completed and in order to reconstruct the signal in the time domain from the obtained spectrograms, an inverse process of the STFT is necessary. This operation is called the Inverse Short-Time Fourier Transform (ISTFT).

The ISTFT can be calculated by several methods. We will describe the weighted overlap-add method (Crochiere, 1980) in the following:

1. For each time frame n of the STFT, the Inverse Discrete Fourier Transform (IDFT) is applied to come back to the time domain, i.e.

$$\mathbf{y}_n(t) = \frac{1}{F}\sum_{f=0}^{F-1} \mathbf{x}_{tf}(f,n)exp(\frac{2j\pi ft}{F}) \qquad ,n = 0,...,N-1, \;\; t = 0,...,T-1 \quad (1.16)$$

2. Nevertheless, the STFT domain filtering used to estimate the source signal STFT coefficients might introduce artifacts that affect all time samples in a given frame. These artifacts are particularly audible at the frame boundaries, which is why another windowing is performed at this level using a synthesis window $w_s(t)$ as $\mathbf{y}_n(t)w_s(t)$. Just like the analysis window, the values of the synthesis window functions $w_s(t)$ are zero outside the interval $t \in [0, T-1]$.

3. After that, these IDFTs are summed to produce the final output in the time domain $\mathbf{y}(t)$.

$$\mathbf{y}(t) = \sum_{n=0}^{N-1} \mathbf{y}_n(t - \tau_0 - nR)w_s(t - \tau_0 - nR) \quad (1.17)$$

   The analysis window $\mathbf{w}_a$ and synthesis window $\mathbf{w}_s$ are generally chosen to satisfy the perfect reconstruction property. Indeed, the entire STFT and ISTFT procedure must allow for the time domain recovery of the original signal, i.e. $\mathbf{y}(t) = \mathbf{x}(t)$. This perfect reconstruction is only possible if and only if the following condition is

satisfied:

$$\sum_{n=0}^{N-1} w_a(t - \tau_0 - nR)w_s(t - \tau_0 - nR) = 1 \quad \forall t \in [0, T-1] \tag{1.18}$$

Since each frame is multiplied by both the analysis and synthesis windows (Vincent et al., 2018).

1.5.1.3 Time-Frequency Trade-Off

A large number of frequency bins F provides a high-frequency resolution. This comes at the cost of reduced temporal resolution because large values of F result in longer time windows T. Thus, a given sample $\mathbf{x}(t)$ will be covered by more frames, resulting in temporal blur. Therefore, there is a trade-off between temporal resolution and frequency resolution.

1.5.2 Permutation and Scaling Ambiguities in Frequency-Domain BSS

The frequency-domain BSS provides considerable advantages. Nevertheless, it is not without its drawbacks. Indeed, since the convolutive problem is treated as a separate problem in each frequency band, the source signals in each frequency bin are estimated with arbitrary permutation and scaling. Thus, unlike in the standard instantaneous mixing model, where these permutation and scaling ambiguities are negligible, in CBSS, they turn into major inconsistencies, which affect separation performance significantly.

- **Scale ambiguity** results from the fact that the variances of the separated signals in each frequency bin cannot be uniquely determined. If the separated signals are transferred into the time domain without fixing this problem, the recovered signals will be distorted versions of the original source signals.
- **Permutation ambiguity** is due to the fact that the order of separated signals across various frequency bins will most likely be inconsistent and cannot be identified exactly. When transforming back to the time domain, this causes severe distortion because the estimated source signals, unless correctly aligned, still contain interference from other sources.

High performance of BSS methods in the frequency domain usually requires a method to overcome permutation and scaling ambiguities.

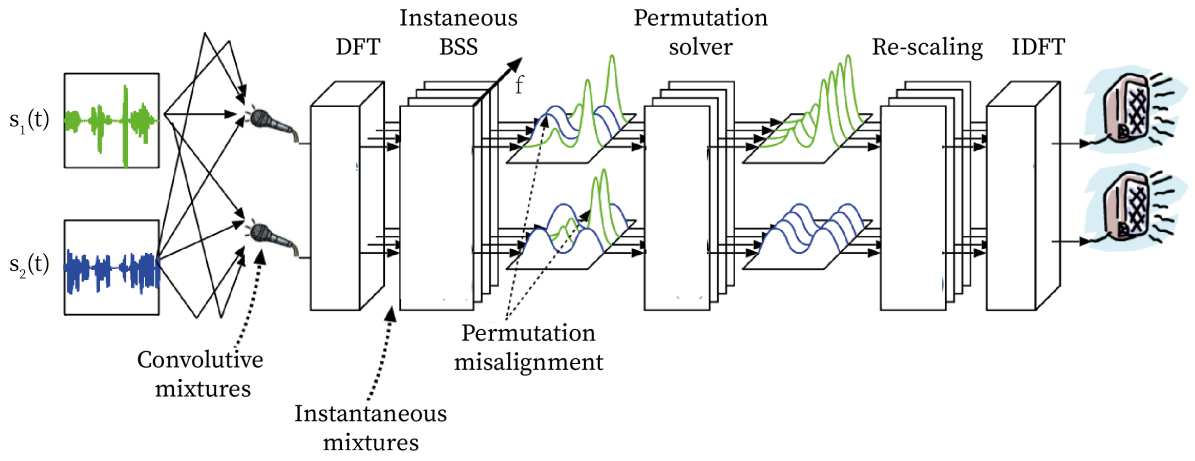

Figure 1.3: Permutation ambiguities in the time-frequency domain (Mukai et al., 2004)

1.6 Measure of Statistical Independence

The most common method to solve the convolutive BSS problem is to exploit the statistical independence of the sources. The usual assumption is the mutual statistical independence between the unknown sources. Although sometimes difficult to implement, this assumption is realistic and entirely justified in many problems. To do this, we must first select a function $\mathcal{C}(\mathbf{W})$, called the contrast function, to measure the independence of the estimated/output signals. There are several measures of dependency, including the Kullback-Leibler (KL) divergence between the joint distribution and the product of the marginal distributions of the outputs, which will be explored in the following chapter. Then, we need to find a learning algorithm for the unmixing matrix W which minimizes the dependency among the outputs.

1.6.1 Statistical Independence

Statistical independence between the source signals is expressed in terms of the probability density functions (PDF). The source signals are said independent, if and only if, the joint PDF, denoted by $p(\mathbf{s}_1, \ldots, \mathbf{s}_L)$ can be written as the product of the marginal PDF's of the sources $\mathbf{s}_l$, i.e. the PDF of the l^{th} source when it is considered alone.

$$p(\mathbf{s}_1, ..., \mathbf{s}_L) = \prod_{l=1}^{L} p(\mathbf{s}_l) \tag{1.19}$$

This is equivalent to stating that model sources $\mathbf{s}_l$ do not carry mutual information, i.e. information on the vector $\mathbf{s}_l$ does not give any information about the vector $\mathbf{s}_{l'}$, $l \neq l'$.

1.6.2 Contrast Functions

In order to maximize the independence between the source signals, we need to define an optimization criterion. In signal processing, the mean square error is frequently used as an optimization criterion. In the present BSS problem, such a criterion cannot be employed since the inputs are not observed. Therefore, criteria called contrast functions, or simply contrasts, are utilized (Jain and Rai, 2012). The maxima or minima of these contrast functions correspond to a separation of all sources.

A contrast is a functional $\mathcal{C}$: $\mathbb{E}^N \to \mathbb{R}$, defined on random variables $\mathbf{x} \in \mathbb{E}^N$ (Palmer and Makeig, 2012), that satisfies the following conditions (Comon, 1994):

1. $\mathcal{C}(\mathbf{x})$ does not change if the components x_i are permuted:

$$\mathcal{C}(\mathbf{Px}) = \mathcal{C}(\mathbf{x}) \ \ ,\forall \ \mathbf{P} \text{ permutation matrix}$$

2. $\mathcal{C}(\mathbf{x})$ is invariant by scale change:

$$\mathcal{C}(\mathbf{Dx}) = \mathcal{C}(\mathbf{x}) \ \ ,\forall \ \mathbf{D} \text{ diagonal matrix}$$

3. If $\mathbf{x}$ has independent components, then:

$$\mathcal{C}(\mathbf{Mx}) \leq \mathcal{C}(\mathbf{x}) \ \ ,\forall \ \mathbf{M} \text{ Complex or real matrix}$$

1.7 Whitening

Most blind source separation methods benefit considerably from preprocessing the data to facilitate separation. The fundamental assumption of the BSS technique is the statistical independence of the sources to be estimated. A weaker form of independence is non-correlation. A slightly stronger property than non-correlation is whitening. Therefore, whitening of the observed data $\mathbf{x}$ is a very useful preprocessing that significantly simplifies the source separation problem.

The whiteness of a zero-mean random vector $\mathbf{x}$ means that its components are uncorrelated, and their variances equal unity (Hyvärinen et al., 2001). In other words, the covariance matrix, as well as the correlation matrix of $\mathbf{x}$, denoted $\mathbf{R}_{xx}$, equals the identity matrix:

$$\mathbf{R}_{xx} = E[\mathbf{x}(t)\mathbf{x}(t)^H] = \mathbf{I}_M \tag{1.20}$$

Where M is the number of sensors used to record the mixtures, $E[.]$ is the statistical expectation operator, $(.)^H$ is the hermitian operator, and $\boldsymbol{I}_M$ the $M \times M$ identity matrix.

Thus, before applying BSS algorithms to fit the separation matrix. We first center the data, then whiten the observed signals by removing the cross-correlation between them and ensuring that they have unit variance. These will result in a better-conditioned problem and increased learning speed.

One of the most commonly used methods for whitening is the Eigen-Value Decomposition (EVD) of the covariance matrix. Its principle is the same as that of Principal Component Analysis (PCA), which allows us to find a lower-dimensional subspace to project our data into. Suppose we have M recorded mixture signals $\mathbf{x}_1, ..., \mathbf{x}_M$ of two sources. The goal is to find L=2 (number of sources) directions $\mathbf{v}_1$ than $\mathbf{v}_2$ where the data vary much more. For this purpose, we first compute the covariance matrix $\mathbf{R}_{xx}$ as follows:

$$\mathbf{R}_{xx} \approx \frac{1}{T} \sum_{t=1}^{T} \mathbf{x}(t)\mathbf{x}(t)^H \tag{1.21}$$

where $\mathbf{x}(t) = [x_1(t), ..., x_M(t)]^T$ and T is the number of samples in the time domain.

The covariance matrix defines both the variance and the orientation of our data. The EVD of this matrix allows us to find two additional elements: a representative vector pointing in the direction of the greater spread of data and a value indicating the spread of data in this direction. These two elements are known, respectively, as *eigenvectors* and *eigenvalues*. Thus, the first principal component which is required to have the largest possible variance is the top eigenvector of $\mathbf{R}_{xx}$, corresponding to the largest eigenvalue λ_1, and the second component, which must be orthogonal to the first component, is the second eigenvector.

Following that, we arrange the L largest values in decreasing order in a diagonal matrix $\mathbf{D}$ and we stack the L corresponding eigenvectors in columns to construct the matrix $\mathbf{E}$, such as:

$$\mathbf{R}_{xx} = \mathbf{E}\mathbf{D}\mathbf{E}^H \tag{1.22}$$

Then, we project our data in the new sub-space by multiplying the observed data $\mathbf{x}$ by the matrix of eigenvectors $\mathbf{E}$, and to make each of our input features have unit variance, we simply re-scale each mixture by multiplying the projected data by $\mathbf{D}^{-\frac{1}{2}}$. Thus, *the whitening matrix* $\mathbf{Q} \in \mathbb{R}^{L \times M}$ can be formulated as:

$$\mathbf{Q} = \mathbf{D}^{-\frac{1}{2}}\mathbf{E}^H \tag{1.23}$$

The whitened data is given by:

$$\mathbf{X}_p = \mathbf{Q}\mathbf{X} \tag{1.24}$$

It results that the new mixing matrix $\mathbf{A}_p$ is orthogonal, since:

$$E[\mathbf{x}_p(t)\mathbf{x}_p(t)^H] = \mathbf{A}_p E[\mathbf{s}(t)\mathbf{s}(t)^H]\mathbf{A}_p^T = \mathbf{A}_p\mathbf{A}_p^T = \mathbf{I}_L \tag{1.25}$$

This means that instead of estimating the $M \times L$ parameters that are the elements of the original matrix $\mathbf{A}$, we just need to estimate an orthogonal mixing matrix $\mathbf{A}_p$ which contains L(L-1)/2 degrees of freedom.

After finding the mixing matrix, we go back to our original space before calculating the estimated sources, using the whitening matrix $\mathbf{Q}$.

$$\mathbf{A} = \mathbf{Q}^{\#}\mathbf{A}_p \tag{1.26}$$

Where $(.)^{\#}$ denotes the pseudo-inverse.

1.8 Literature Survey

The first study of the BSS problem dates back to 1986 when Herault and Jutten presented the H-J algorithm (Herault and Jutten, 1986). This work marked the beginning of a new era in signal processing. Since then, the BSS problem has attracted researchers' interest.

In 1989, the first international workshop on higher-order spectral analysis was organized. At this workshop, Cardoso (Cardoso, 1989) and Comon (Comon, 1989) presented the first papers on Independent Component Analysis (ICA). These works provided a clear general framework for the well-known ICA separation algorithm. The latter is essentially based on the statistical independence of the sources $\boldsymbol{s}(t)$. The unmixing matrix is constructed by optimizing an objective function so that the resulting components of the output vector $\boldsymbol{y}(t)$ become as independent as possible.

Since then, the theory of ICA has been refined progressively through the development of a variety of algorithms. In 1994, Comon (Comon, 1994) proposed a popular minimum mutual information-based ICA method. The following year, Bell and Sejnowski (Bell and Sejnowski, 1995a) proposed the Infomax principle-based maximum entropy approach. Later, Amari and colleagues improved this algorithm using the natural gradient (Amari et al., 1995a),(Amari, 1998). A few years later, Hyvärinen, Oja,

and Pajunen presented the fixed-point or FastICA algorithm (Hyvärinen and Oja, 1997), (Oja and Hyvarinen, 2000), (Hyvarinen, 1999a), (Hyvärinen and Pajunen, 1999), which contributed to the application of ICA to large-scale problems due to its computational efficiency.

In addition to ICA, many other popular algorithms have been proposed to solve source separation in the case of instantaneous mixtures, including the Algorithm for Multiple Unknown Signals Extraction (AMUSE) (Tong et al., 1990) and its generalization, Second Order Blind Identification (SOBI) (Belouchrani et al., 1997), which use the time dependence of the components via the joint diagonalization of one or more autocovariance matrices, respectively.

For acoustic applications, we deal with the convolutive mixture case. Although most of the efforts have been directed toward solving the simple case of instantaneous mixtures, several algorithms have also been proposed to solve the CBSS problem.

In the time domain, sparse component analysis (Gribonval and Lesage, 2006), (Yu et al., 2014a) is an example of a widely used algorithm for speech signal separation. Another robust solution is the convolutive generalization of the popular SOBI algorithm (Bousbia-Salah et al., 2001). Despite the excellent separation performance of these algorithms, they suffer, like the rest of the time-domain BSS methods, from a high computation load.

To increase the computational efficiency of CBSS, Frequency-Domain BSS (FDBSS) approaches have been proposed. FDBSS transforms the mixtures in the frequency domain before applying a complex-valued instantaneous ICA algorithm to each frequency bin. Therefore, the permutation problem must be solved in post-processing (e.g. (Sawada et al., 2004)).

To solve the permutation problem in FDICA, a more elegant solution called Independent Vector Analysis (IVA) was proposed by Kim (Kim et al., 2006a). IVA solves the permutation problem by employing a multivariate source prior where the sources are considered as random vectors. This method allows for independence between multivariate source signals and preserves the dependence within each source vector. The original IVA algorithm uses the natural gradient method to optimise the contrast function. The fast fixed-point IVA (FastIVA) algorithm (Lee et al., 2007a) is a fast version of the IVA algorithm and it uses the Newton method to minimise the contrast function.

The source prior used to model the dependence structure within the source vectors is crucial to the separation performance. IVA typically uses a spherical multivariate distribution (e.g., a spherical Laplace distribution) as the source model to ensure

higher-order correlations between the frequency bins within each source. This latter can be used for a wide range of sounds because it does not include specific information about the spectral structures of the sources. However, some sources have specific spectral structures, such as the harmonic structure of instrumental or musical sounds. Therefore, the introduction of a better source model has the potential to improve the performance of source separation. In 2016, a new FDBSS method known as Independent Low-Rank Matrix Analysis (ILRMA) was introduced (Kitamura et al., 2016). This algorithm uses Non-Negative Matrix Factorization (NMF) to capture the spectral structures of each source as the generative source model in IVA.

1.9 Conclusion

In this first chapter, we have provided a general overview of blind source separation and discussed different mixing models. We also highlighted the fact that, in a blind context, complete identification of the mixing matrix is impossible. Indeed, the latter can only be predicted up to one scalar and one column permutation. Moreover, we have seen that the source separation problem can be reduced to the search for independent components in a linear mixture. Finally, some separation methods have been discussed.

In the following chapters, we will only discuss blind separation problems of convolutive mixtures, since we are dealing with speech signals, with a focus on methods that operate in the time-frequency domain.

Chapter 2

Some Blind Speech Separation Algorithms

The most common method for BSS is to separate the independent components so that they are as close as possible to the source signals; this is known as ICA-based BSS. In these methods, there are two key aspects: the objective function and the optimization algorithm (Yu et al., 2014b).

The definition of an objective function involves choosing a suitable independence measure and a source model. The choice of the latter is crucial but not immediate since we have to make assumptions about the statistical properties of the source signals. An appropriate selection of these two elements results in a well-posed problem and has a significant impact on the quality of the separation.

Once the objective function is defined, we need to select a learning algorithm to optimize it. We typically use the conventional gradient, the stochastic gradient (Bell and Sejnowski, 1995b), the relative gradient (Cardoso, 1997), the natural gradient (Cichocki and Unbehauen, 1996), or another heuristic learning algorithm.

In the following, we will detail the objective function and the learning algorithm used for three popular methods in speech signal separation: Natural Gradiant IVA (NG-IVA), Fast IVA (FIVA), and Independent Low-Rank Matrix Analysis (ILRMA).

2.1 Independent Vector Analysis

Independent Vector Analysis (IVA), first proposed in (Kim et al., 2006a), is an extension of Frequency Domain Independent Component Analysis (FDICA) that uses the entire frequency spectrum as an input to solve the permutation problem. In IVA, the sources are not simply single variables as in ICA but rather multidimensional random vectors where all frequency components of each source signal are considered

together. Since the elements of a random vector are related to each other, the elements of a single source vector are dependent. Thus, the algorithm aims to maximize the independence between the different source signals while retaining the dependency within each vector. Hence, instead of using a contrast function that measures the component-wise independence in each frequency bin, a contrast function that measures the whole independence among the multivariate sources is applied.

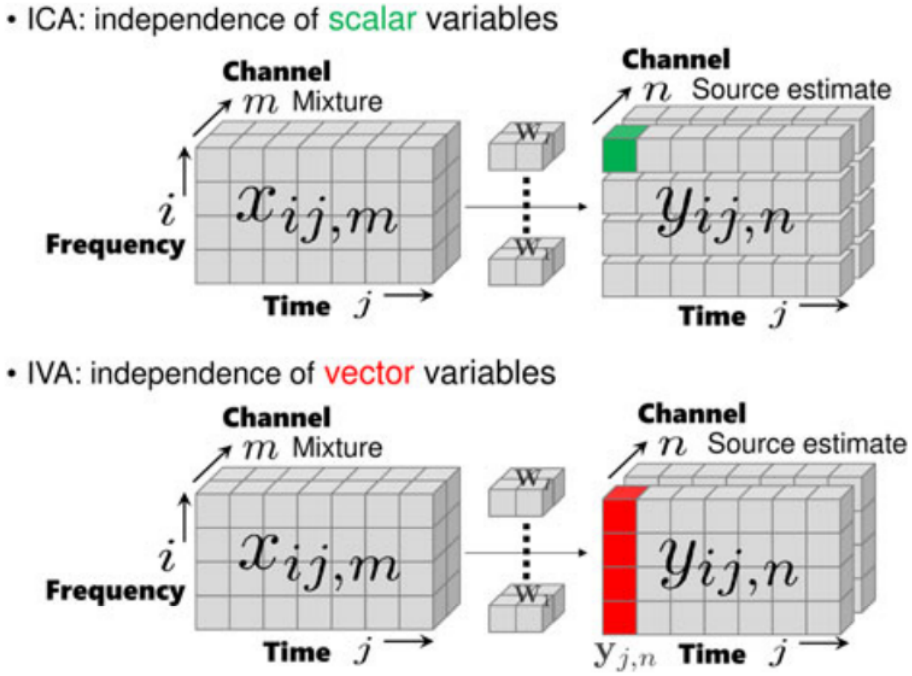

Figure 2.1: Independence in IVA (Sawada et al., 2019)

2.1.1 The IVA Model

In order to implement the IVA algorithm for the convolutive BSS, the Short Time Fourier Transform (STFT) is used to convert the problem from the time domain to the frequency domain. The noiseless FDBSS model is the following:

$$\mathbf{x}_{tf}(f,n) = \mathbf{A}(f)\mathbf{s}_{tf}(f,n) \tag{2.1}$$

In order to separate the source signals from the observed mixtures, an unmixing matrix must be estimated. As seen in the first chapter, the separation model is given as:

$$\mathbf{y}_{tf}(f,n) = \mathbf{W}(f)\mathbf{x}_{tf}(f,n) \tag{2.2}$$

2.1.2 Formula for the Whole Unmixing

In the classical approach, to estimate the unmixing matrix at the frequency bin f, we apply the ICA algorithm with only the mixtures at the frequency bin f. In contrast, in the IVA model, we apply the standard ICA by considering the entire spectrum of mixtures and learn each group as a whole by defining a dependence between the multivariate sources (Hiroe, 2006).

$$\begin{bmatrix} \begin{bmatrix} y_{tf_1}(1,n) \\ \vdots \\ y_{tf_1}(F,n) \end{bmatrix} \\ \vdots \\ \begin{bmatrix} y_{tf_L}(1,n) \\ \vdots \\ y_{tf_L}(F,n) \end{bmatrix} \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} w_{11}(1) & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & w_{11}(F) \end{bmatrix} & \dots & \begin{bmatrix} w_{1M}(1) & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & w_{1M}(F) \end{bmatrix} \\ \vdots & \ddots & \vdots \\ \begin{bmatrix} w_{L1}(1) & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & w_{L1}(F) \end{bmatrix} & \dots & \begin{bmatrix} w_{LM}(1) & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & w_{LM}(F) \end{bmatrix} \end{bmatrix} \begin{bmatrix} \begin{bmatrix} x_{tf_1}(1,n) \\ \vdots \\ x_{tf_1}(F,n) \end{bmatrix} \\ \vdots \\ \begin{bmatrix} x_{tf_M}(1,n) \\ \vdots \\ x_{tf_M}(F,n) \end{bmatrix} \end{bmatrix} \tag{2.3}$$

$$\iff \begin{bmatrix} \mathbf{y}_{tf_1}(n) \\ \vdots \\ \mathbf{y}_{tf_L}(n) \end{bmatrix} = \begin{bmatrix} \mathbf{W}_{11} & \dots & \mathbf{W}_{1M} \\ \vdots & \ddots & \vdots \\ \mathbf{W}_{L1} & \dots & \mathbf{W}_{LM} \end{bmatrix} \begin{bmatrix} \mathbf{x}_{tf_1}(n) \\ \vdots \\ \mathbf{x}_{tf_M}(n) \end{bmatrix} \tag{2.4}$$

$$\iff \mathbf{Y}_{tf}(n) = \mathbf{W}\ \mathbf{X}_{tf}(n) \tag{2.5}$$

where $\mathbf{Y}_{tf}(n)$, whose elements $\mathbf{y}_{tf_l}(n) \in \mathbb{C}^{F \times 1}$, and $\mathbf{X}_{tf}(n)$, whose elements $\mathbf{x}_{tf_m}(n) \in \mathbb{C}^{F \times 1}$, represent the time-frequency representation of the estimated sources and the observed mixtures, at the time frame n, respectively.

Note that while the dependence between all frequency bins of a source signal is maintained by using an appropriate source model, mixing and unmixing in IVA is restricted to each frequency bin in order to simplify the derivation of the learning rules, as we will see later.

2.1.3 IVA Assumptions

The IVA method, and more generally, the ICA-based BSS algorithms, make the following assumptions (Kim et al., 2006b):

1. Elements of a source vector $\mathbf{s}_l$ are mutually independent of elements of the other

source vectors $\mathbf{s}_{l'}$ $, l \neq l'$

$$p(\mathbf{s}_1, ..., \mathbf{s}_L) = p(\mathbf{s}_1) \times ... \times p(\mathbf{s}_L) \tag{2.6}$$

Where $p(\mathbf{s}_1, ..., \mathbf{s}_L)$ is the joint probability density function of the sources, and $p(\mathbf{s}_l)$ is the marginal probability density function of the l^{th} source signal $\mathbf{s}_l$.

2. Within a source vector, the elements depend on the others.

3. The number of sources L is less than or equal to the number of microphones M (L $\leq$ M).

2.1.4 Pre-processing

After applying the STFT, we center and whiten the observed mixtures to make the problem well conditioned. In IVA, since mixing and unmixing are restricted to each frequency bin, we keep the output signals $\mathbf{y}_l(f)$, $l \in [1, L]$ zero-mean and white by preprocessing the observed data in each frequency bin $\mathbf{X}(f)$ to be zero-mean and white, and by constraining the unmixing matrix $\mathbf{W}(f)$'s to be orthogonal (Makino et al., 2007):

$$E\left[\mathbf{X}(f)\mathbf{X}(f)^H\right] = \mathbf{I}_M \quad , f = 1, ..., F \tag{2.7}$$

$$\mathbf{W}(f)\mathbf{W}(f)^H = \mathbf{I}_L \quad , f = 1, ..., F \tag{2.8}$$

2.1.5 Cost Function

Separating multivariate sources from multivariate observations requires a cost function for multivariate random variables to measure the statistical dependence between the output signals. In IVA, the time structure of the signals is ignored. Indeed, we deal with the signals in the time-frequency domain as observations of random vectors. Thus, for convenience we omit the time structure n and denote the l^{th} output signal in the time-frequency domain $\mathbf{y}_{tf_l} = [y_{tf_l}(1), y_{tf_l}(2), \ldots, y_{tf_l}(F)]^T$.

When the estimated sources $\mathbf{y}_{tf_1}, ..., \mathbf{y}_{tf_L}$ are mutually independent, the joint PDF $p(\mathbf{y}_{tf_1}, ..., \mathbf{y}_{tf_L})$ should be decomposed to the product of the marginal PDFs $\prod_l p(\mathbf{y}_{tf_l})$ i.e. $p(\mathbf{y}_{tf_1}, ..., \mathbf{y}_{tf_L}) = \prod_l p(\mathbf{y}_{tf_l})$.

Therefore, to measure the independence between estimated sources, we need to measure the relative dependency between the two distributions. The Kullback-Leibler Divergence (KLD) is used in the standard IVA (Kim et al., 2006a). The latter can

be explained as the distance between these two PDFs. This contrast $\mathcal{C}$ reduces to the minimum, zero, if and only if the outputs $\mathbf{y}_{tf_l}$'s are mutually independent.

$$\mathcal{C} = KLD\left(p(\mathbf{y}_{tf_1}, ..., \mathbf{y}_{tf_L}) \middle\| \prod_l p(\mathbf{y}_{tf_l})\right) = KLD\left(p(\mathbf{y}_{tf}) \middle\| \prod_l p(\mathbf{y}_{tf_l})\right) \tag{2.9}$$

Where $\mathbf{y}_{tf} = [\mathbf{y}_{tf_1}, ..., \mathbf{y}_{tf_L}]^T$.

$$\mathcal{C} = \int p(\mathbf{y}_{tf}) log\left(\frac{p(\mathbf{y}_{tf})}{\prod_l p(\mathbf{y}_{tf_l})}\right) \mathrm{d}\mathbf{y}_{tf} \tag{2.10}$$

The interesting part of this contrast function is that each source is multivariate, this makes it so that when minimizing this function to remove the dependency between sources, the dependency between the elements of each vector is not affected. To confirm this statement, Hiroe in his article (Hiroe, 2006), calculated the KLD from artificially permuted spectrograms, and found that less permuted spectrograms lead to lower value of KLD. Therefore, the contrast function preserves the inherent dependence of each source vector, while removing the dependence across sources. The previously defined contrast function may be written as follows:

$$\mathcal{C} = \int p(\mathbf{y}_{tf}) \log\Big(p(\mathbf{y}_{tf})\Big) d\mathbf{y}_{tf} - \int p(\mathbf{y}_{tf}) \log\left(\prod_l p(\mathbf{y}_{tf_l})\right) d\mathbf{y}_{tf} \tag{2.11a}$$

$$= \sum_l H(\mathbf{y}_{tf_l}) - H(\mathbf{y}_{tf}) \tag{2.11b}$$

Where $H(.)$ represents the entropy function.

$$H(\mathbf{y}_{tf}) = -\int p(\mathbf{y}_{tf}) \log\Big(p(\mathbf{y}_{tf})\Big) d\mathbf{y}_{tf} \tag{2.12}$$

$$H(\mathbf{y}_{tf_l}) = -\int p(\mathbf{y}_{tf_l}) \log\Big(p(\mathbf{y}_{tf_l})\Big) d\mathbf{y}_{tf_l} \tag{2.13}$$

The entropy of a linear transformation $\mathbf{y}_{tf} = \mathbf{W}\mathbf{x}_{tf}$ is given by:

$$H(\mathbf{y}_{tf}) = H(\mathbf{x}_{tf}) + \log\Big|\det \mathbf{W}\Big| \tag{2.14}$$

For a more detailed proof about this result refer to (Hyvärinen et al., 2001, Ch. 5, p. 109).

Note that the term $H(\mathbf{x}_{tf})$ is constant with respect to $\mathbf{W}$. Hence, minimizing the KL divergence is equivalent to minimizing the following term:

$$argmin_{\{\mathbf{W}(f)\}}\ \mathcal{C} = argmin_{\{\mathbf{W}(f)\}}\ \sum_{l} H(\mathbf{y}_{tf_l}) - \log\left|\det \mathbf{W}\right| \tag{2.15a}$$

$$= argmin_{\{\mathbf{W}(f)\}}\ \sum_{l} E\left[-\log\left(p(\mathbf{y}_{tf_l})\right)\right] - \log\left|\det \mathbf{W}\right| \tag{2.15b}$$

With the constraint of $\mathbf{W}(f)$'s orthogonal.

2.1.6 Learning Algorithm

Now that the objective function for IVA has been defined, an optimization method must be selected to minimize the contrast function and therefore seek the unmixing matrix $\mathbf{W}$. A number of algorithms can be used including natural gradient (Amari et al., 1995b), relative gradient (Cardoso, 1995), fixed-point iteration (Hyvärinen and Oja, 1997), or Newton's method (Hyvarinen, 1999b). In the standard IVA, we apply the Natural Gradient, which is well known as a fast convergence method.

By differentiating the cost function $\mathcal{C}$ with respect to the unmixing matrix $\mathbf{W}$, we obtain the gradient descent matrix. The natural gradient $\Delta\mathbf{W}$ can be obtained by multiplying the gradient descent matrix by the scaling matrices $\mathbf{W}^H\mathbf{W}$.

$$\Delta\mathbf{W} = -\frac{\partial\mathcal{C}}{\partial\mathbf{W}}\mathbf{W}^H\mathbf{W} \tag{2.16}$$

As seen in the equation (2.15), the contrast function is given by:

$$\mathcal{C} = \sum_{l=1}^{L} H(\mathbf{y}_{tf_l}) - log\left|det\mathbf{W}\right| \tag{2.17}$$

This implies that the gradient descent of the contrast function $\mathcal{C}$ with respect to $\mathbf{W}$ is the following:

$$\frac{\partial\mathcal{C}}{\partial\mathbf{W}} = \frac{\partial\left(\sum_{l=1}^{L} H(\mathbf{y}_{tf_l})\right)}{\partial\mathbf{W}} - \frac{\partial \log\left|det\mathbf{W}\right|}{\partial\mathbf{W}} \tag{2.18}$$

The second part of the equation (2.18), can be simplified using the equation (57) in (Petersen and Pedersen, 2012, ch. 1, p. 9).

$$\frac{\partial \log |det\mathbf{W}|}{\partial \mathbf{W}} = \left(\mathbf{W}^{-1}\right)^H = \left(\mathbf{W}^H\right)^{-1} \tag{2.19a}$$

$$\Rightarrow \quad \frac{\partial \log |det\mathbf{W}|}{\partial \mathbf{W}}\mathbf{W}^H\mathbf{W} = \left(\mathbf{W}^H\right)^{-1}\mathbf{W}^H\mathbf{W} = \mathbf{W} \tag{2.19b}$$

$$\Rightarrow \quad \Delta\mathbf{W} = -\frac{\partial\left(\sum_{l=1}^{L} H(\mathbf{y}_{tf_l})\right)}{\partial \mathbf{W}}\mathbf{W}^H\mathbf{W} + \mathbf{W} \tag{2.19c}$$

The first part of the equation (2.18) is difficult to express in a simple formula, instead of $\Delta\mathbf{W}$, a rule $\Delta\mathbf{W}(f)$ is derived for each frequency bin f:

$$-\frac{\partial\left(\sum_{l=1}^{L} H(\mathbf{y}_{tf_l})\right)}{\partial \mathbf{W}(f)}\mathbf{W}(f)^H = E\left\{\boldsymbol{\varphi}^f(\mathbf{y}_{tf})\mathbf{y}_{tf}(f)^H\right\} \tag{2.20}$$

Where $\boldsymbol{\varphi}^f(\mathbf{y}_{tf})$ represents the score function at the frequency bin f:

$$\boldsymbol{\varphi}^f(\mathbf{y}_{tf}) = \left[\varphi_1^f(\mathbf{y}_{tf_1}), ..., \varphi_L^f(\mathbf{y}_{tf_L})\right]^T \tag{2.21a}$$

$$\varphi_l^f(\mathbf{y}_{tf_l}) = \frac{\partial}{\partial y_{tf_l}(f)} \log\left(p(\mathbf{y}_{tf_l})\right) \tag{2.21b}$$

We can see from equation (2.21) that the score function takes the whole spectrum of the l^{th} output, i.e., all frequency bins, as its argument, which is the main difference from the conventional frequency-domain ICA that considers the frequency bins separately. Using this new dependent model for IVA, each frequency bin depends on the entire spectrum, thus solving the permutation problem.

From (2.19) and (2.20), we can write the natural gradient $\Delta\mathbf{W}(f)$ in each frequency bin f as follows:

$$\Delta\mathbf{W}(f) = \left\{\mathbf{I} + E\left[\boldsymbol{\varphi}^f(\mathbf{y}_{tf})\mathbf{y}_{tf}(f)^H\right]\right\}\mathbf{W}(f) \tag{2.22}$$

Once, the natural gradient is computed in each frequency bin f, the unmixing matrix is updated as in the classical gradient algorithm:

$$\mathbf{W}(f) = \mathbf{W}(f) + \eta\Delta\mathbf{W}(f) \tag{2.23}$$

Where $\eta \in [0, 1]$ is the step size. It is a tuning parameter that imposes a trade-off between

convergence speed and stability.

2.1.7 Multivariate Probability Density Functions

In the light of the equations (2.21) and (2.22), it appears that in order to design a speech separation system, it is crucial to have a reasonable approximation for the marginal multivariate PDFs of the sources $p(\mathbf{y}_{tf_l})$, where $\mathbf{y}_{tf_l} = [y_{tf_l}(1), ..., y_{tf_l}(F)]^T$ is a vector across all frequency bins.

As explained previously, the inter-frequency dependency is preserved in IVA by using the multivariate source prior. Previous works observed that speech has such property of spherical symmetry and introduced spherically invariant random processes (SIRP) to model band-limited speech (Brehm and Stammler, 1987). Thus, a spherically symmetric multivariate source distribution is adopted. The spherically symmetric property is represented as an assignment of the vector's norm into a proper scalar function $f(.)$, such as:

$$p(\mathbf{y}_{tf_l}) = \alpha f\left(||\mathbf{y}_{tf_l}||_2\right) \tag{2.24}$$

Where: $||\mathbf{y}_{tf_l}||_2 = \sqrt{\sum_{f=1}^{F} |y_{tf_l}(f)|^2}$

The choice of the function f(.) generates various PDFs including Spherically Symmetric Laplacian distribution (SSL), Symmetric Exponential Norm Distribution (SEND), Generalized Gaussian distribution (GGD).

Speech signals have super-Gaussian characteristics (Tashev and Acero, 2010), which means that the data is highly peaked at zero and that asymptotically falls off more slowly than the Gaussian distribution as the distance from zero increases. Hence, the original IVA method uses a the multivariate Spherically Symmetric Laplacian distribution (SSL) (Kim et al., 2006a) for the source priors, since this latter captures the super-Gaussian property of speech. Assuming this source model, the marginal PDFs of the estimated sources can be written as follows:

$$p(\mathbf{y}_{tf_l}) \propto \exp\left\{-||\mathbf{y}_{tf_l}||_2\right\} = \exp\left\{-\sqrt{\sum_{f=1}^{F} |y_{tf_l}(f)|^2}\right\} \tag{2.25}$$

As a result, the score function at a particular frequency bin f is the following:

$$\varphi_l^f(\mathbf{y}_{tf_l}) = \frac{y_{tf_l}(f)}{\sqrt{\sum_{f=1}^{F} |y_{tf_l}(f)|^2}} \tag{2.26}$$

2.1.8 Scaling

As discussed earlier in Section 1.5.2, separation methods operating in the frequency domain cannot determine the scales of the estimated sources. Therefore a rescaling method is needed to correct the amplitudes before transforming the signals into the time domain after separation. Indeed, if the scaling in the various bins is not rectified, the recovered signals will only be a filtered version of the sources and will generally not sound natural.

Several methods for correcting the scaling ambiguity have been proposed in the literature. Hereafter, the well-known Minimal Distortion Principle (MDP) method is employed. Its principle is as follows: from a set of proper separators, choose the one that minimizes the squared distance between the separated source and the input signals $E[|\mathbf{y}(t) - \mathbf{x}(t)|^2]$.

This MDP method uses the following unmixing matrix for $f \in [1, F]$:

$$\mathbf{W}_s(f) = diag(\mathbf{W}^{-1}(f))\mathbf{W}(f) \tag{2.27}$$

Let's explain in the following why this new unmixing matrix can solve the scaling problem. Remember that we have scaling and permutation ambiguities in ICA-based approaches, thus we have:

$$\mathbf{W}(f)\mathbf{A}(f) = \mathbf{D}(f)\mathbf{P}(f) \quad f \in [1, F] \tag{2.28}$$

Where $\mathbf{D}(f)$ is a diagonal scaling martix which represents the scaling indeterminacy of IVA and $\mathbf{P}(f)$ is a permutation matrix, but we consider only the case of $\mathbf{P}(f) = I$ to make the description below simple.

$$diag(\mathbf{W}^{-1}(f)) = diag(\mathbf{A}(f)\mathbf{D}^{-1}(f)) \tag{2.29a}$$
$$= diag(\mathbf{A}(f))\mathbf{D}^{-1}(f) \tag{2.29b}$$

Therefore,

$$\mathbf{W}_s(f)\mathbf{A}(f) = diag(\mathbf{W}^{-1}(f))\mathbf{W}(f)\mathbf{A}(f) \tag{2.30a}$$
$$= diag(\mathbf{A}(f))\mathbf{D}^{-1}(f)\mathbf{D}(f) \tag{2.30b}$$
$$= diag(\mathbf{A}(f)) \tag{2.30c}$$

This last equation shows that the choice of the unmixing matrix $\mathbf{W}_s(f)$ has removed the effect of the scaling matrix $\mathbf{D}(f)$.

2.1.9 Summary of Algorithm

Algorithme 1 : IVA algorithm

input : Observed mixtures $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_M]^T$, number of sources L , number of iterations $maxIter$

output : Estimated sources $\mathbf{y}_1, ..., \mathbf{y}_L$

Calculate the STFT $\mathbf{X}_{tf} \in \mathbb{C}^{M \times F \times N}$ of the observed mixtures $\mathbf{X}$
for $f \leftarrow 1$ **to** F **do**
  `// Whitening the data using PCA`
  $\mathbf{X}_{tf}(f) = \mathbf{X}_{tf}(f) - mean_n(\mathbf{X}_{tf}(f))$ `//` $mean_n$`: along the time axis`
  $\mathbf{R}_{XX} = E[\mathbf{X}_{tf}(f)\mathbf{X}_{tf}^H(f)]$
  $[\mathbf{E}, \mathbf{D}] = EVD(\mathbf{R}_{XX})$
  $\mathbf{Q}(f) = \mathbf{D}^{-\frac{1}{2}}\mathbf{E}^T$
  $\mathbf{X}_p(f) = \mathbf{Q}(f)\mathbf{X}_{tf}(f)$
  `// Initialization of` $\mathbf{W}_p$
  $\mathbf{W}_p(f) = I$
end
`// Learning rules`
for $iter \leftarrow 1$ **to** $maxIter$ **do**
  for $f \leftarrow 1$ **to** F **do**
    $\mathbf{Y}_{tf}(f) = \mathbf{W}_p(f)\mathbf{X}_p(f)$
    Calculate $\varphi_l^f(\mathbf{y}_{tf_l}) = \frac{\mathbf{y}_{tf_l}(f)}{\sqrt{\sum_{f=1}^{F} |\mathbf{y}_{tf_l}(f)|^2)}}$ for all l
    Define $\boldsymbol{\varphi}^f(\mathbf{y}_{tf}) = [\varphi_1^f(\mathbf{y}_{tf_1}), ..., \varphi_L^f(\mathbf{y}_{tf_L})]^T$
    $\Delta\mathbf{W}_p(f) = \left\{\mathbf{I} + E\left[\boldsymbol{\varphi}^f(\mathbf{y}_{tf})\mathbf{y}_{tf}(f)^H\right]\right\}\mathbf{W}_p(f)$
    Update $\mathbf{W}_p(f)$: $\mathbf{W}_p(f) = \mathbf{W}_p(f) + \eta\Delta\mathbf{W}_p(f)$
  end
end
`// Rescaling`
for $f \leftarrow 1$ **to** F **do**
  $\mathbf{W}(f) = \mathbf{W}_p(f)\mathbf{Q}(f)$
  $\mathbf{W}(f) = diag(\mathbf{W}^{-1}(f))\mathbf{W}(f)$ `// Apply minimal distortion principle`
  $\mathbf{Y}_{tf}(f) = \mathbf{W}(f)\mathbf{X}(f)$
end
Calculate the ISTFT of $\mathbf{Y}_{tf}(f)$

2.2 Fast Independent Vector Analysis

Several researchers have worked on optimizing the IVA algorithm seen earlier, so there have been several new algorithms. In this section, we will deal with Fast Independent Vector Analysis (FastIVA) (Lee et al., 2007a). In this algorithm, the optimization of the contrast function is done using the update rules of Newton's method, which, compared

to other gradient-descent methods, converges quickly and does not require learning rate selection. In addition, Newton's method finds stationary points rather than specified maxima or minima and therefore has more flexibility in separating sources.

When applying Newton's method to the contrast function, which is the likelihood contrast function with an appropriate multivariate PDF in the case of Fast IVA, the standard approach for optimizing a real-valued function of complex variables is used. Then, a second-order Taylor polynomial, a useful tool for deriving complex Newton-like IVA algorithms for convolutive BSS, is introduced.

2.2.1 Cost Function

2.2.1.1 Maximum Likelihood Method

Maximum likelihood estimation is a statistical method for estimating the unknown parameters of a model that we will denote θ. The parameter values are found to maximize the likelihood that the model's process produced the observed data (Hyvärinen et al., 2001, Ch. 4, p. 90). This implies that to implement maximum likelihood estimation we must:

- Assume a model for our data.
- Be able to derive the likelihood function for our data, given our assumed model.

Once the likelihood function is derived, maximum likelihood estimation is nothing more than a simple optimization problem.

Assume that we have T observations of $\mathbf{x}$, denoted by $x(1), ..., x(T)$. The likelihood function is defined by:

$$p(\mathbf{x}|\theta) = p(x(1), ..., x(T)|\theta) \tag{2.31}$$

The application of the ML method almost always assumes that the observations x(t) are statistically independent of each other. Fortunately, this holds quite often in practice. Assuming independence, the likelihood function can be obtained as the product of the conditional PDF of the single scalar measurement x(t) evaluated at the T points.

$$p(\mathbf{x}|\theta) = \prod_{t=1}^{T} p(x(t)|\theta) \tag{2.32}$$

Because many density functions contain an exponential function, it is often more

convenient to deal with the log-likelihood function $log\ \ p(\mathbf{x}|\theta)$.

$$log\ \ p(\mathbf{x}|\theta) = \sum_{t=1}^{T} log\ \ p(x(t)|\theta) \tag{2.33}$$

The maximum likelihood estimate $\hat{\theta}$ of the parameter vector θ is chosen to be the value that maximizes the likelihood function. Clearly, this estimate also maximizes the log-likelihood. Therefore, the ML estimator is usually found from the solutions of the following equation:

$$\frac{\partial}{\partial\theta} log\ \ p(\mathbf{x}|\theta)|_{\theta=\widehat{\theta}} = 0 \tag{2.34}$$

2.2.1.2 Likelihood Contrast Function for CBSS

Since the observed vectors $\mathbf{x}_{tf}$ and the estimated signals $\mathbf{y}_{tf}$ are linearly related with the matrix $\mathbf{W}(f)$ (equation (2.2)). We can obtain the joint PDF of the observed mixtures $p(\mathbf{x}_{tf_1}(n), ..., \mathbf{x}_{tf_M}(n))$ by using the well-known result on the density of a linear transform (Hyvärinen et al., 2001, Ch. 2, p. 35) and by considering that the Jacobian for a complex-valued variable is the square of the Jacobian for a real-valued variable (Adali et al., 2008).

$$p(\mathbf{x}_{tf_1}(n), ..., \mathbf{x}_{tf_M}(n)) = p(\mathbf{y}_{tf_1}(n), ..., \mathbf{y}_{tf_L}(n)) \prod_f \left| det\mathbf{W}(f) \right|^2 \tag{2.35}$$

As the conventional IVA, the separated signals $\mathbf{y}_{tf_l}$ are assumed to be independent of each other, thus:

$$p(\mathbf{x}_{tf_1}(n), ..., \mathbf{x}_{tf_M}(n)) = \prod_l p(\mathbf{y}_{tf_l}(n)) \prod_f \left| det\mathbf{W}(f) \right|^2 \tag{2.36}$$

The likelihood function of the parameters $\mathbf{W}(f)\ \ , f = 1, ..., F$, denoted $\mathcal{L}(\mathbf{W})$, is given as:

$$\mathcal{L}(\mathbf{W}) = \prod_n p\left(\mathbf{x}_{tf_1}(n), ..., \mathbf{x}_{tf_M}(n) \middle| \mathbf{W}\right) \tag{2.37}$$

Using the equation (2.36), the likelihood function can be written as:

$$\mathcal{L}(\mathbf{W}) = \prod_n \left\{ \left(\prod_l p(\mathbf{y}_{tf_l}(n)) \right) \left(\prod_f \left| det\mathbf{W}(f) \right|^2 \right) \right\} \tag{2.38}$$

Therefore, the negative log-likelihood function, which represents the FastIVA cost function, can be calculated as:

$$\mathcal{C} = -\log \mathcal{L}(\mathbf{W}) = -\sum_{f,n} \log \left| det \mathbf{W}(f) \right|^2 - \sum_{n,l} \log p(\mathbf{y}_{tf_l}(n)) \tag{2.39a}$$

$$= -2N \sum_{f} \log \left| det \mathbf{W}(f) \right| - \sum_{n,l} \log p(\mathbf{y}_{tf_l}(n)) \tag{2.39b}$$

For notational simplicity, let us divide the negative log-likelihood by N and replace the sum over the time frame index n by an expectation operator. Then, the cost function to be optimize is described by:

$$\mathcal{C} = -2 \sum_{f} \log \left| det \mathbf{W}(f) \right| - \sum_{l} E \left\{ \log p(\mathbf{y}_{tf_l}(n)) \right\} \tag{2.40}$$

Since we keep the unmixing matrices orthogonal during the learning, such that $\log \left| det(\mathbf{W}(f)) \right| = 0, f = 1, ..., F$, the contrast function is given by:

$$\mathcal{C} = -\sum_{l} E \left\{ \log p(\mathbf{y}_{tf_l}) \right\} \tag{2.41}$$

After replacing $p(\mathbf{y}_{tf_l})$ in the likelihood contrast with a spherically symmetric distribution, like SSL or SEND, the contrast function can be written as:

$$\mathcal{C} = \sum_{l} E \left\{ G(\sum_{f} |y_{tf_l}(f)|^2) \right\} \tag{2.42}$$

With: $G(\sum_f |y_{tf_l}(f)|^2) = -log\ \ p(\mathbf{y}_{tf_l})$ is a nonlinear function which corresponds to the source prior.

The problem is now reduced to a minimization problem of the previously defined negative log-likelihood function.

$$argmin_{\{\mathbf{W}(f)\}} \sum_{l=1}^{L} E \left\{ G(\sum_{f} |\mathbf{y}_{tf_l}(f)|^2) \right\} \tag{2.43}$$

under the constraint $\mathbf{w}_l(f)^H \mathbf{w}_l(f) = 1$

Where $\mathbf{w}_l(f)$ denotes the l^{th} row of the unmixing matrix $\mathbf{W}$(f).

By using the Lagrange multiplier, we convert the constrained optimization problem to a simple optimization problem.

$$\sum_{l=1}^{L}\left[E\left\{G(\sum_{f}|\mathbf{y}_l(f)|^2)\right\}-\sum_{f}\lambda_l(f)((\mathbf{w}_l(f))^H\mathbf{w}_l(f)-1)\right] \tag{2.44}$$

Where: $\lambda_l(f)$ is the lagrangian multiplier.

2.2.2 Learning Algorithm

Once the contrast function is selected, we can derive our separation algorithm by choosing the optimization method.

We start from the Taylor expansion of a real-valued function f($\mathbf{w}$) around $\mathbf{w}_0$, where $\mathbf{w}$ is complex. Using the definitions for complex derivatives and complex gradients (Lee et al., 2007b), it can be shown that the Taylor expansion of f($\mathbf{w}$) up to the second order is given as fellow:

$$\begin{aligned}f(\mathbf{w})\approx f(\mathbf{w}_0)&+\frac{\partial f(\mathbf{w}_0)}{\partial \mathbf{w}^T}(\mathbf{w}-\mathbf{w}_0)+\frac{\partial f(\mathbf{w}_0)}{\partial \mathbf{w}^H}(\mathbf{w}-\mathbf{w}_0)^*+\frac{1}{2}(\mathbf{w}-\mathbf{w}_0)^T\frac{\partial^2 f(\mathbf{w}_0)}{\partial \mathbf{w}\partial \mathbf{w}^T}(\mathbf{w}-\mathbf{w}_0)\\&+\frac{1}{2}(\mathbf{w}-\mathbf{w}_0)^H\frac{\partial^2 f(\mathbf{w}_0)}{\partial \mathbf{w}^*\partial \mathbf{w}^H}(\mathbf{w}-\mathbf{w}_0)^*+(\mathbf{w}-\mathbf{w}_0)^H\frac{\partial^2 f(\mathbf{w}_0)}{\partial \mathbf{w}^*\partial \mathbf{w}^T}(\mathbf{w}-\mathbf{w}_0)\end{aligned} \tag{2.45}$$

The $\mathbf{w}$ that optimizes the function f($\mathbf{w}$) will set the gradient $\frac{\partial f(\mathbf{w})}{\partial \mathbf{w}^*}$, to zero and hence

$$\frac{\partial f(\mathbf{w})}{\partial \mathbf{w}^*}\approx\frac{\partial f(\mathbf{w}_0)}{\partial \mathbf{w}^*}+\frac{\partial^2 f(\mathbf{w}_0)}{\partial \mathbf{w}^*\partial \mathbf{w}^T}(\mathbf{w}-\mathbf{w}_0)+\frac{\partial^2 f(\mathbf{w}_0)}{\partial \mathbf{w}^*\partial \mathbf{w}^H}(\mathbf{w}-\mathbf{w}_0)^*\equiv 0 \tag{2.46}$$

Using this, the fast algorithm is derived by setting $\mathbf{w}\equiv\mathbf{w}_l(f)$ and $f(\mathbf{w}_l(f))\equiv E\left\{G(\sum_f|\mathbf{y}_l(f)|^2)\right\}-\sum_f\lambda_l(f)((\mathbf{w}_l(f))^H\mathbf{w}_l(f)-1)$.

After computing the derivatives in (2.46) and making some approximations, we can obtain the following learning rule:

$$\begin{aligned}\mathbf{w}_l(f)=E\left\{G'(\sum_f|y_{l,0}(f)|^2)+|y_{l,0}(f)|^2G''(\sum_f|y_{f,0}(f)|^2)\right\}\mathbf{w}_{l,0}(f)\\-E\left\{(y^*_{l,0}(f))G'(\sum_f|y_{l,0}(f)|^2)\mathbf{x}_{tf}(f)\right\}\end{aligned} \tag{2.47}$$

where G'( ) and G'' represent the first and second derivative of G( ), respectively.

It should be noted that the rows of the unmixing matrix $\mathbf{W}$ need to be

decorrelated. For the maximum likelihood approach, symmetric decorrelation is applied in each frequency bin.

$$\mathbf{W}(f) \leftarrow (\mathbf{W}(f)\mathbf{W}^H(f))^{-\frac{1}{2}}\mathbf{W}(f) \tag{2.48}$$

As was the case in IVA, a rescaling method is also needed in FastIVA to correct the amplitudes. The same one used in IVA (Section 2.1.8): the Minimal Distortion Principle (MDP), will be used in FastIVA.

2.2.3 Summary of Algorithm

Algorithme 2 : FastIVA Algorithm

input : Observed mixtures $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_M]^T$, number of sources L , number of iterations $maxIter$

output : Estimated sources $\mathbf{y}_1, ..., \mathbf{y}_L$

Calculate the STFT $\mathbf{X}_{tf} \in \mathbb{C}^{M \times F \times N}$ of the observed mixtures $\mathbf{X}$
for $f \leftarrow 1$ **to** F **do**
  // Whitening the data using PCA
  $\mathbf{X}_{tf}(f) = \mathbf{X}_{tf}(f) - mean_n(\mathbf{X}_{tf}(f))$ // $mean_n$: along the time axis
  $\mathbf{R}_{XX} = E[\mathbf{X}_{tf}(f)\mathbf{X}_{tf}^H(f)]$
  $[\mathbf{E}, \mathbf{D}] = EVD(\mathbf{R}_{XX})$
  $\mathbf{Q}(f) = \mathbf{D}^{-\frac{1}{2}}\mathbf{E}^T$
  $\mathbf{X}_p(f) = \mathbf{Q}(f)\mathbf{X}_{tf}(f)$
  // Initialization of $\mathbf{W}_p$
  $\mathbf{W}_p(f) = I$
end
// Learning rules
for $iter \leftarrow 1$ **to** $maxIter$ **do**
  for $f \leftarrow 1$ **to** F **do**
    $\mathbf{Y}_{tf}(f) = \mathbf{W}_p(f)\mathbf{X}_p(f)$
    // Update unmixing matrices
    $\mathbf{w}_l(f) = E\left\{G'(\sum_f |y_{l,0}(f)|^2) + |y_{l,0}(f)|^2 G''(\sum_f |y_{l,0}(f)|^2)\right\}\mathbf{w}_{l,0}(f) - E\left\{(y_{l,0}^*(f))G'(\sum_f |y_{l,0}(f)|^2)\mathbf{X}_p(f)\right\}$
    // Decorrelation
    $\mathbf{W}_p(f) \leftarrow (\mathbf{W}_p(f)\mathbf{W}_p^H(f))^{-\frac{1}{2}}\mathbf{W}_p(f)$
  end
end

```
// Rescaling
for f ← 1 to F do
    W(f) = W_p(f)Q(f);
    W(f) = diag(W^{-1}(f))W(f)   // Apply minimal distortion principle;
    Y_tf(f) = W(f)X(f);
end
Calculate the ISTFT of Y_tf(f);
```

2.3 Independent Low-Rank Matrix Analysis

The conventional IVA models the sources using a stationary distribution that does not include any specific information about the sources' spectral structures. However, some sources have specific spectral patterns. In 2016, Daichi Kitamura presented Independent Low-Rank Matrix Analysis (ILRMA), a new efficient method for convolutional BSS that provides a better source model (Kitamura et al., 2016). This method unifies Independent Vector Analysis (IVA) and Non-negative Matrix Factorization (NMF). The latter allows capturing the spectral structures of each source as a source model in the IVA.

2.3.1 Itakura-Saito NMF

Non-negative Matrix Factorization (Virtanen, 2007) is a type of sparse representation algorithm that decomposes a non-negative matrix $\widetilde{\mathbf{Y}} \in \mathbb{R}^{F \times N}$, i.e. a matrix which does not contain negative elements, into the product of two non-negative matrices. The first matrix, called the basis matrix $\mathbf{T} \in \mathbb{R}^{F \times K}$, acts as a dictionary of spectral patterns in $\widetilde{\mathbf{Y}}$, such as notes, chords, percussive sounds, or more complex adaptive structures. The second one called the activation matrix $\mathbf{V} \in \mathbb{R}^{K \times N}$, involves time-varying gains of each basis in $\mathbf{T}$ as row vectors. It is worth noting that the number of bases K must be set to a much smaller number than F or N.

$$\widetilde{\mathbf{Y}} \approx \widehat{\mathbf{Y}} \stackrel{\text{def}}{=\!=} \mathbf{TV} \tag{2.49}$$

Where $\widehat{\mathbf{Y}}$ is the low-rank approximation of $\widetilde{\mathbf{Y}}$.

$$\widetilde{y}(f,n) \approx \sum_{k=1}^{K} t(f,k)v(k,n) \tag{2.50}$$

The following figure depicts the decomposition model of NMF for K=2. The basis matrix $\mathbf{T}$ includes two types of spectral patterns as the bases to represent the observed

matrix using time-varying gains in the activation matrix $\mathbf{V}$.

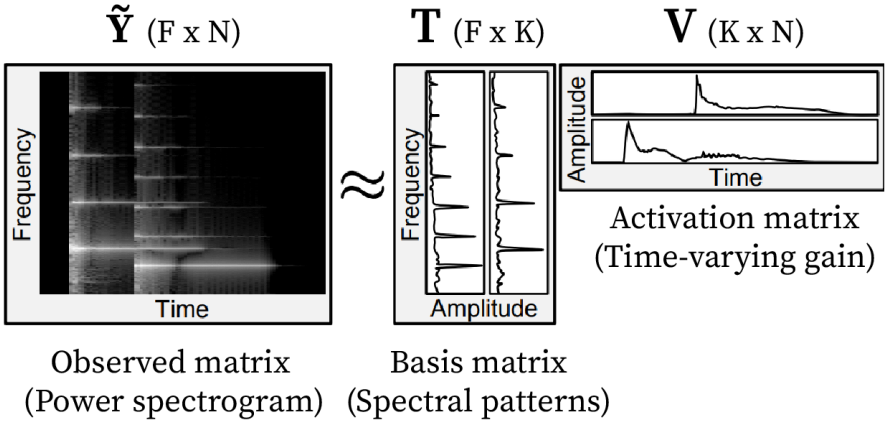

Figure 2.2: NMF decomposition with K=2 bases (Kitamura et al., 2016)

Let us assume that we have a single-channel signal $\mathbf{y}$, to which we apply the STFT. As a result, we have the $(f,n)^{th}$ element of the T-F representations $y_{tf}(f,n)$. However, as we have seen before, the entries for the NMF algorithm must be non-negative. Thus, to apply NMF, we have to first convert $y_{tf}(f,n)$ to a non-negative value $\widetilde{y}(f,n) \in \mathbb{R}_+$. Typically, we can take the squared value:

$$\widetilde{y}(f,n) = |y_{tf}(f,n)|^2 = y_{tf}(f,n)y_{tf}^*(f,n) \;\; , f = 1, ...F \quad , n = 1, ..., N \tag{2.51}$$

Then, a matrix $\left[\widetilde{\mathbf{Y}}\right]_{fn} = \widetilde{y}(f,n)$, is constructed with all the pre-processed values.

Once the pre-processing is completed, we apply the NMF method which consists of minimizing an error of fit between the power spectrogram of the mixture, denoted $\widetilde{\mathbf{Y}}$, and its approximate low-rank matrix $\hat{\mathbf{Y}} \stackrel{\text{def}}{=\!=} \mathbf{TV}$. Thus, the problem can be formulated by defining a cost function $\mathcal{D}$, which is a scalar error measure between the input matrix $\widetilde{\mathbf{Y}}$ and the output product $\hat{\mathbf{Y}}$ but subject to the non-negativity of the values of $\mathbf{T}$ and $\mathbf{V}$. The non-negativity of $\mathbf{T}$ ensures the interpretability of the dictionary, in the sense that the extracted patterns $\mathbf{t}_k$ (k^{th} column of $\mathbf{T}$) remain non-negative, like the data samples, while the non-negativity of $\mathbf{V}$ ensures that approximation $\hat{\mathbf{Y}}$ remains non-negative, like $\widetilde{\mathbf{Y}}$.

$$\mathbf{T}, \mathbf{V} = argmin_{\mathbf{T},\mathbf{V}} \;\; \mathcal{D}\left(\widetilde{\mathbf{Y}}, \mathbf{TV}\right) \tag{2.52}$$

Where:

$$\mathcal{D}\left(\widetilde{\mathbf{Y}}, \mathbf{TV}\right) = \sum_{f=1}^{F}\sum_{n=1}^{N} d\left(\widetilde{y}(f,n), \hat{y}(f,n)\right) \tag{2.53}$$

There are several choices for the distance/divergence measures used in the NMF cost function, including the Euclidean distance (Lee and Seung, 2000), the generalized Kullback-Leibler (KL) divergence (Lee and Seung, 2000), and the Itakura-Saito (IS) divergence (Févotte et al., 2009). Hereafter, we will only focus on the IS divergence cost function. In order to define this cost function, the probability distribution of the STFT coefficients $y_{tf}(f,n)$ is modeled by a zero-mean complex Gaussian distribution as follows:

$$\mathcal{N}(y_{tf}(f,n)|0,\hat{y}(f,n)) \propto \frac{1}{\hat{y}(f,n))} \exp\left\{-\frac{|y_{tf}(f,n)|^2}{\hat{y}(f,n)}\right\} \tag{2.54}$$

We notice in equation (2.54) that the lower-rank approximation $\hat{y}(f,n)$ of the power spectrogram $\widetilde{y}(f,n)$ represents the variance of the distribution.

The IS cost function can be derived as the difference between the log-likelihoods of $\widetilde{y}(f,n)$ and $\hat{y}(f,n)$.

$$d\left(\widetilde{y}(f,n),\hat{y}(f,n)\right) = \log\left\{\mathcal{N}(y_{tf}(f,n)|0,\widetilde{y}(f,n))\right\} - \log\left\{\mathcal{N}(y_{tf}(f,n)|0,\hat{y}(f,n))\right\} \tag{2.55a}$$

$$= -\log\left\{\widetilde{y}(f,n)\right\} - \frac{\widetilde{y}(f,n)}{\widetilde{y}(f,n)} + \log\left\{\hat{y}(f,n)\right\} + \frac{\widetilde{y}(f,n)}{\hat{y}(f,n)} \tag{2.55b}$$

$$= \frac{\widetilde{y}(f,n)}{\hat{y}(f,n)} - \log\left\{\frac{\widetilde{y}(f,n)}{\hat{y}(f,n)}\right\} - 1 \tag{2.55c}$$

This distance/divergence can be minimized according to (Févotte et al., 2009), in the following manner: First, the elements of $\mathbf{T}$ and $\mathbf{V}$ are randomly initialized with non-negative values. Then, the following multiplicative update rules are applied until convergence:

$$t(f,k) \leftarrow t(f,k)\sqrt{\frac{\sum_n \widetilde{y}(f,n)v(k,n)(\hat{y}(f,n))^{-2}}{\sum_n v(k,n)(\hat{y}(f,n))^{-1}}} \tag{2.56}$$

$$v(k,n) \leftarrow v(k,n)\sqrt{\frac{\sum_f \widetilde{y}(f,n)t(f,k)(\hat{y}(f,n))^{-2}}{\sum_f t(f,k)(\hat{y}(f,n))^{-1}}} \tag{2.57}$$

These update rules are called multiplicative because they update each element by multiplying it by a positive scalar value. As a result, these update rules guarantee the non-negativity of all matrices' elements while providing fast convergence.

2.3.2 Cost Function

The classical IVA models the sources with a stationary distribution, where the variance is uniformly fixed at unity over the frequency bins and is not estimated. In the time-varying Gaussian IVA (Ono et al., 2012), a new source model is introduced in which the variance is shared across frequency bins but changes over time. The Itakura-Saito NMF (IS-NMF) introduced a more flexible source model, where the variance is blindly estimated in each time-frequency slot by low-rank decomposition using the NMF (Makino, 2018, Ch. 6). As a result, we can model the specific time-frequency structure with a limited number of bases and activations.

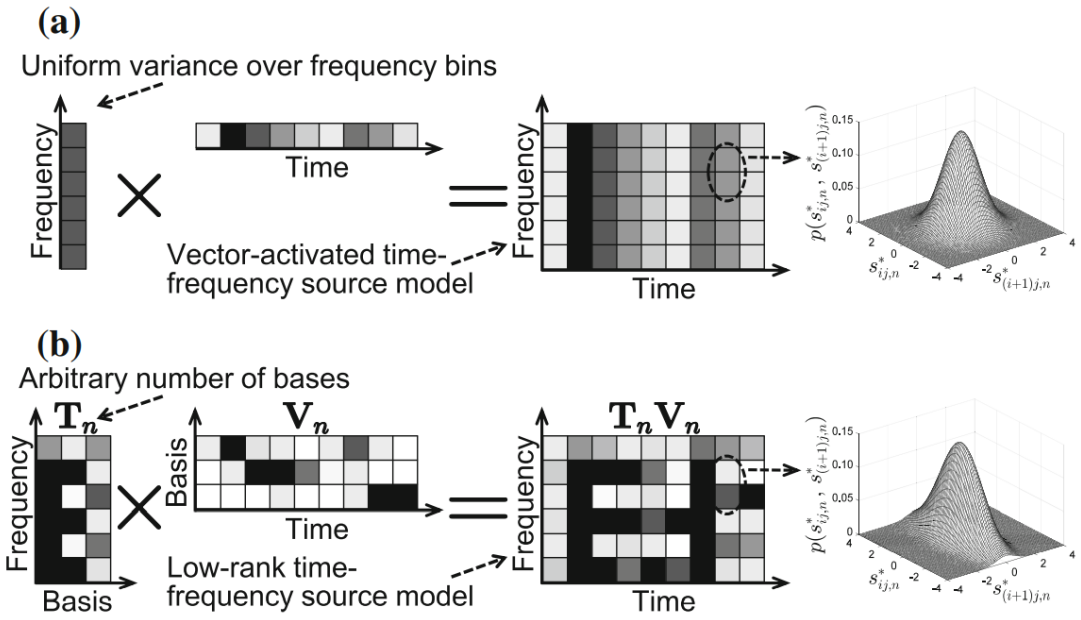

Figure 2.3: Comparison of source models (variance structures) a) time-varying Gaussian IVA and b) Itakura–Saito NMF, where grayscale in each time-frequency slot indicates scale of variance (Makino, 2018)

Therefore, in ILRMA, IS-NMF is used for source estimation in order to develop more accurate spectral models. This is accomplished by decomposing the source variance $\hat{y}_l(f,n)$ using a limited number of NMF bases. The source model in each time-frequency slot is assumed to be the circularly symmetric complex Gaussian distribution, as in IS-NMF. As a result, the joint PDF of the separated signals $\mathbf{y}_{tf_l}$ is as follows:

$$p(\mathbf{y}_{tf_1}(n),...,\mathbf{y}_{tf_L}(n)) = \prod_{l=1}^{L} p(\mathbf{y}_{tf_l}(n)) \tag{2.58a}$$

$$= \prod_{l,f} \frac{1}{\pi \hat{y}_l(f,n)} \exp\left\{-\frac{|y_{tf_l}(f,n)|^2}{\hat{y}_l(f,n)}\right\} \tag{2.58b}$$

As explained earlier, unlike IVA where the sourcewise variance is uniform, in ILRMA

this latter corresponds to the decomposition of the power spectrograms of the separated sources at each time-frequency slot $y_{tf_l}(f,n)$.

The cost function in ILRMA is the negative log-likelihood with the previously defined PDF (equation (2.58)). As it is the same as the Fast IVA cost function, simply replacing the PDF in equation (2.39) will allow us to define the objective function to minimize.

$$\mathcal{C} = -2N \sum_{f} \log \left| det \mathbf{W}(f) \right| + \sum_{f,n,l} \left[\log \hat{y}_l(f,n) + \frac{|y_{tf_l}(f,n)|^2}{\hat{y}_l(f,n)} \right] \tag{2.59}$$

In the previous cost function (2.59), the variance $\hat{y}_l(f,n)$ as well as the demixing matrices $\mathbf{W}(f)$ are both unknown. Therefore, in ILRMA, the demixing matrix $\mathbf{W}(f)$ and the source model $p(\mathbf{y}_{tf_1}, ..., \mathbf{y}_{tf_L})$ are simultaneously estimated in a fully blind manner.

2.3.3 Update Rules

Hereafter, the update rules based on the auxiliary function technique (Ono, 2011) are used for the optimization IVA. These update rules are faster than conventional update rules, and the step size parameter can be omitted in each iteration.

$$\mathbf{U}_l(f) = \frac{1}{N} \sum_{n} \frac{1}{\hat{y}_l(f,n)} \mathbf{x}_{tf}(f,n) \mathbf{x}_{tf}^H(f,n) \tag{2.60}$$

$$\mathbf{w}_l(f) = \left(\mathbf{W}(f) \mathbf{U}_l(f) \right)^{-1} \mathbf{e}_l \tag{2.61}$$

$$\mathbf{w}_l(f) \leftarrow \mathbf{w}_l(f) \left(\mathbf{w}_l^H(f) \mathbf{U}_l(f) \mathbf{w}_l(f) \right)^{-\frac{1}{2}} \tag{2.62}$$

Where $\mathbf{e}_n$ denotes the $N \times 1$ unit vector with the n^{th} element equal to unity.

After the update $\mathbf{W}(f)$, the separated signal $\mathbf{y}_{tf}(f,n)$ should be updated as:

$$y_{tf_l}(f,n) = \mathbf{w}_l^H(f) \mathbf{x}_{tf}(f,n) \tag{2.63}$$

Then, once the separated signals are updated, the variance of the outputs at each time-frequency slot $\hat{y}_l(f,n)$ is updated after applying the multiplicative update rules of the NMF algorithm, that we will detail below.

It should be noted that there are two models in ILRMA: ILRMA without partitioning function (ILRMA-1) and ILRMA with partitioning function (ILRMA-2),

where the partitioning function clusters the total number of bases into each source.

2.3.3.1 ILRMA-1 / Without Partitioning Function

In ILRMA-1, a fixed number of bases, K, is used to decompose each separated source spectrogram, i.e. the NMF bases K are not shared among the L source estimates through the optimization process. In this case, the decomposition and multiplicative update rules are as follows (Kitamura et al., 2016):

$$t_l(f,k) \leftarrow t_l(f,k)\sqrt{\frac{\sum_n |y_{tf_l}(f,n)|^2 v_l(k,n)(\hat{y}_l(f,n))^{-2}}{\sum_n v_l(k,n)(\hat{y}_l(f,n))^{-1}}} \tag{2.64}$$

$$v_l(k,n) \leftarrow v_l(k,n)\sqrt{\frac{\sum_f |y_{tf_l}(f,n)|^2 t_l(f,k)(\hat{y}_l(f,n))^{-2}}{\sum_f t_l(f,k)(\hat{y}_l(f,n))^{-1}}} \tag{2.65}$$

$$\hat{y}_l(f,n) = \sum_k t_l(f,k)v_l(k,n) \tag{2.66}$$

2.3.3.2 ILRMA-2 / With Partitioning Function

In ILRMA-2, we only set the total number of bases K and the algorithm adaptively estimates the optimal number of NMF bases for each separate source. Thus, we also need to optimize the latent cluster indicator variables $z(l,k)$, in the same way as $t(f,k)$ and $v(k,n)$. The latter have a continuous value such that $z(l,k) \in [0,1]$ and that $\sum_l z(l,k) = 1$. In this case, the decomposition and multiplicative update rules are as follows (Kitamura et al., 2016):

$$z(l,k) \leftarrow z(l,k)\sqrt{\frac{\sum_{f,n} |y_{tf_l}(f,n)|^2 t(f,k)v(k,n)(\hat{y}_l(f,n))^{-2}}{\sum_{f,n} t(f,k)v(k,n)(\hat{y}_l(f,n))^{-1}}} \tag{2.67}$$

$$z(l,k) \leftarrow \frac{z(l,k)}{\sum_{l'} z(l',k)} \tag{2.68}$$

$$t(f,k) \leftarrow t(f,k)\sqrt{\frac{\sum_{n,l} |y_{tf_l}(f,n)|^2 z(l,k)v(k,n)(\hat{y}_l(f,n))^{-2}}{\sum_{n,l} z(l,k)v(k,n)(\hat{y}_l(f,n))^{-1}}} \tag{2.69}$$

$$v(k,n) \leftarrow v(k,n)\sqrt{\frac{\sum_{f,l}|y_{tf_l}(f,n)|^2 z(l,k)t(f,k)(\hat{y}_l(f,n))^{-2}}{\sum_{f,l} z(l,k)t(f,k)(\hat{y}_l(f,n))^{-1}}} \tag{2.70}$$

$$\hat{y}_l(f,n) = \sum_k z(l,k)t(f,k)v(k,n) \tag{2.71}$$

2.3.4 Normalization

Once the objective function is minimized using the update rules defined earlier, a scaling ambiguity occurs between $\mathbf{W}(f)$ and $\hat{y}_l(f,n)$ because both are unknown and they can both determine the scale of the separate signal $y_{tf_l}(f,n)$. To avoid this problem, the following normalization should be applied at each iteration:

$$\mathbf{w}_l(f) \leftarrow \mathbf{w}_l(f)\lambda_l^{-1} \tag{2.72}$$

$$y_{tf_l}(f,n) \leftarrow y_{tf_l}(f,n)\lambda_l^{-1} \tag{2.73}$$

$$\hat{y}_l(f,n) \leftarrow \hat{y}_l(f,n)\lambda_l^{-2} \tag{2.74}$$

- **ILRMA-1:**

$$t_l(f,k) \leftarrow t_l(f,k)\lambda_l^{-2} \tag{2.75}$$

- **ILRMA-2:**

$$t(f,k) \leftarrow t(f,k)\sum_l z(l,k)\lambda_l^{-2} \tag{2.76}$$

$$z(l,k) \leftarrow \frac{z(l,k)\lambda_l^{-2}}{\sum_{l'} z(l',k)\lambda_{l'}^{-2}} \tag{2.77}$$

Where λ_l is an arbitrary sourcewise normalization coefficient, such as the sourcewise average power:

$$\lambda_l = \left[(FN)^{-1}\sum_{f,n}|y_{tf_l}(f,n)|^2\right]^{\frac{1}{2}} \tag{2.78}$$

2.3.5 Back-projection Technique

Similarly to IVA, a rescaling method is required to correct the amplitudes of the separated signals $\mathbf{y}_{tf_l}$ before transforming the signals into the time domain. In ILRMA, we apply the back-projection technique (Murata et al., 2001). This technique restores the scale of the output signals to their amplitudes observed in $\mathbf{x}_{tf}(f,n)$, and this by multiplying these signals by the inverse matrix of the demixing matrix.

$$\overline{\mathbf{y}}_{tf_l}(f,n) = \mathbf{w}_l(f)^{-1} y_{tf_l}(f,n) \tag{2.79}$$

Where $\overline{\mathbf{y}}_{tf_l}(f,n) = [\overline{y}_1(f,n), ..., \overline{y}_M(f,n)]^T$ represents the separated signals whose scale is fitted to the observed signals at each microphone and $w_l(f)^{-1}$ is the l^{th} column of the inverse of the unmixing matrix at the frequency bin f.

2.3.6 Summary of Algorithm

This section summarizes the ILRMA algorithms (Kitamura, 2018). As previously stated, ILRMA has two models depending on whether or not the partitioning function is included.

Algorithme 3 : ILRMA-1

input : Observed mixtures $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_M]^T$, number of sources L
output : Estimated sources $\mathbf{y}_1, ..., \mathbf{y}_L$

Calculate the STFT $\mathbf{X}_{tf} \in \mathbb{C}^{M \times F \times N}$ of the observed mixtures $\mathbf{x}_m$
Initialize $\mathbf{W}(f)$ with identity matrix $\mathbf{I}_L \in \mathbb{R}^{L \times L}$, for all f
Initialize $\mathbf{T}_l$ and $\mathbf{V}_l$ with non-negative random values, for all l
Calculate $\mathbf{y}_{tf}(f,n) = \mathbf{W}(f)\mathbf{x}_{tf}(f,n)$, for all f and n
Calculate $\mathbf{P}_l = |\mathbf{Y}_{tf_l}|^2$, for all l
Calculate the variance of the source priors $\mathbf{R}_l = \mathbf{T}_l\mathbf{V}_l$ for all l
repeat
  for $l \leftarrow 1$ **to** L **do**
    `// NMF multiplicative rules`
    $\mathbf{T}_l = \mathbf{T}_l \circ \left[\frac{\left(\mathbf{P}_l \circ \mathbf{R}_l^{-2}\right)\mathbf{V}_l^T}{\mathbf{R}_l^{-1}V_l^T}\right]^{\frac{1}{2}}$ `// Update of basis matrix`
    $\mathbf{R}_l = \mathbf{T}_l\mathbf{V}_l$
    $\mathbf{V}_l = \mathbf{V}_l \circ \left[\frac{\mathbf{T}_l^T\left(\mathbf{P}_l \circ \mathbf{R}_l^{-2}\right)}{\mathbf{T}_l^T\mathbf{R}_l^{-1}}\right]^{\frac{1}{2}}$ `// Update of activation matrix`
    $\mathbf{R}_l = \mathbf{T}_l\mathbf{V}_l$

13
14
15 **for** $f \leftarrow 1$ **to** F **do**
16 `// Learning of the unmixing matrix W`
17 $\mathbf{U}_l(f) = \frac{1}{N}\left\{\mathbf{X}_{tf}(f)^H\left[\mathbf{X}_{tf}(f) \circ \left(\mathbf{R}_l(f)^{-1}\mathbf{1}^{(1\times M)}\right)\right]\right\}^T$
18 `//` $\mathbf{1}^{(1\times M)}$ `denotes` $(1 \times M)$ `matrix of ones`
19 $\mathbf{w}_l(f) = \left(\mathbf{W}(f)\mathbf{U}_l(f)\right)^{-1}\mathbf{e}_l$
20 $\mathbf{w}_l(f) = \mathbf{w}_l(f)\left(\mathbf{w}_l(f)^H\mathbf{U}_l(f)\mathbf{w}_l(f)\right)^{-\frac{1}{2}}$
21 **end**
22 **end**
23 Calculate $\mathbf{y}_{tf}(f,n) = \mathbf{W}(f)\mathbf{x}_{tf}(f,n)$, for all f and n `// New estimates` Calculate $\mathbf{P}_l = |\mathbf{Y}_{tf_l}|^2$, for all l
24 **for** $l \leftarrow 1$ **to** L **do**
25 $\lambda_l = \sqrt{\frac{1}{FN}\sum_{f,n} p_l(f,n)}$
26 **for** $f \leftarrow 1$ **to** F **do**
27 $\mathbf{w}_l(f) = \mathbf{w}_l(f)\lambda_l^{-1}$
28 **end**
29 $\mathbf{P}_l = \mathbf{P}_l\lambda_l^{-2}$ $\mathbf{R}_l = \mathbf{R}_l\lambda_l^{-2}$ $\mathbf{T}_l = \mathbf{T}_l\lambda_l^{-2}$
30 **end**
31 **until convergence**

Algorithme 4 : ILRMA-2

input : Observed mixtures $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_M]^T$, number of sources L
output : Estimated sources $\mathbf{y}_1, ..., \mathbf{y}_L$

1 Calculate the STFT $\mathbf{X}_{tf} \in \mathbb{C}^{M\times F\times N}$ of the observed mixtures $\mathbf{x}_m$ Initialize $\mathbf{W}(f)$ with identity matrix $\mathbf{I}_L \in \mathbb{R}^{L\times L}$, for all f
2 Initialize $\mathbf{T}_l$ and $\mathbf{V}_l$ with non-negative random values, for all l, and $\mathbf{Z}$ with random values $\in [0,1]$ $\mathbf{Z} = \mathbf{Z} \circ \left(\mathbf{1}^{(N\times N)}\mathbf{Z}\right)^{-1}$
3 Calculate $\mathbf{y}_{tf}(f,n) = \mathbf{W}(f)\mathbf{x}_{tf}(f,n)$, for all f and n Calculate $\mathbf{P}_l = |\mathbf{Y}_{tf_l}|^2$, for all l Calculate the variance of the source priors $\mathbf{R}_l = \left[\left(\mathbf{1}^{(F\times 1)}\mathbf{z}_l^T\right) \circ \mathbf{T}\right]\mathbf{V}$ for all l
4 **repeat**
5 **for** $l \leftarrow 1$ **to** L **do**
6 $\mathbf{b}_l^{(\mathbf{Z})} = \left(\frac{\left\{\left[\mathbf{T}^T\left(\mathbf{P}_l \circ \mathbf{R}_l^{-2}\right)\right] \circ \mathbf{V}\right\}\mathbf{1}^{(N\times 1)}}{\left[\left(\mathbf{T}^T\mathbf{R}_l^{-1}\right) \circ \mathbf{V}\right]\mathbf{1}^{(N\times 1)}}\right)^{\frac{1}{2}}$
7 **end**

12

13 $\mathbf{Z} = \mathbf{Z} \circ \mathbf{B}^{(\mathbf{Z})}$, where $\mathbf{B}^{(\mathbf{Z})} = [\mathbf{b}_1^{(\mathbf{Z})}, ..., \mathbf{b}_L^{(\mathbf{Z})}]^T$ $\mathbf{Z} = \mathbf{Z} \circ \left(\mathbf{1}^{(N \times N)}\mathbf{Z}\right)^{-1}$ Calculate $\mathbf{R}_l = \left[\left(\mathbf{1}^{(F \times 1)}\mathbf{z}_l^T\right) \circ \mathbf{T}\right] \mathbf{V}$ for all l

14 **for** $f \leftarrow 1$ **to** F **do**

15 $\mathbf{b}_f^{(\mathbf{T})} = \left(\frac{\left\{\left[\mathbf{V}\left(\mathbf{P}_f \circ \mathbf{R}_f^{-2}\right)\right] \circ \mathbf{Z}^T\right\}\mathbf{1}^{(L \times 1)}}{\left[\left(\mathbf{V}\mathbf{R}_f^{-1}\right) \circ \mathbf{Z}^T\right]\mathbf{1}^{(L \times 1)}}\right)^{\frac{1}{2}}$

16 **end**

17 $\mathbf{T} = \mathbf{T} \circ \mathbf{B}^{(\mathbf{T})}$, where $\mathbf{B}^{(\mathbf{T})} = [\mathbf{b}_1^{(\mathbf{T})}, ..., \mathbf{b}_F^{(\mathbf{T})}]^T$

18 Calculate $\mathbf{R}_l = \left[\left(\mathbf{1}^{(F \times 1)}\mathbf{z}_l^T\right) \circ \mathbf{T}\right] \mathbf{V}$ for all l

19 **for** $n \leftarrow 1$ **to** N **do**

20 $\mathbf{b}_n^{(\mathbf{V})} = \left(\frac{\left\{\left[\mathbf{T}^T\left(\mathbf{P}_n \circ \mathbf{R}_n^{-2}\right)\right] \circ \mathbf{Z}^T\right\}\mathbf{1}^{(L \times 1)}}{\left[\left(\mathbf{T}^T\mathbf{R}_n^{-1}\right) \circ \mathbf{Z}^T\right]\mathbf{1}^{(L \times 1)}}\right)^{\frac{1}{2}}$

21 **end**

22 $\mathbf{V} = \mathbf{V} \circ \mathbf{B}^{(\mathbf{V})}$, where $\mathbf{B}^{(\mathbf{V})} = [\mathbf{b}_1^{(\mathbf{V})}, ..., \mathbf{b}_N^{(\mathbf{V})}]$

23 Calculate $\mathbf{R}_l = \left[\left(\mathbf{1}^{(F \times 1)}\mathbf{z}_l^T\right) \circ \mathbf{T}\right] \mathbf{V}$ for all l **for** $l \leftarrow 1$ **to** L **do**

24 **for** $f \leftarrow 1$ **to** F **do**

25 $\mathbf{U}_l(f) = \frac{1}{N}\left\{\mathbf{X}_{tf}(f)^H \left[\mathbf{X}_{tf}(f) \circ \left(\mathbf{R}_l(f)^{-1}\mathbf{1}^{(1 \times M)}\right)\right]\right\}^T$

26 $\mathbf{w}_l(f) = \left(\mathbf{W}(f)\mathbf{U}_l(f)\right)^{-1}\mathbf{e}_l$

27 $\mathbf{w}_l(f) = \mathbf{w}_l(f)\left(\mathbf{w}_l(f)^H\mathbf{U}_l(f)\mathbf{w}_l(f)\right)^{-\frac{1}{2}}$

28 **end**

29 **end**

30 Calculate $\mathbf{y}_{tf}(f,n) = \mathbf{W}(f)\mathbf{x}_{tf}(f,n)$, for all f and n `// New estimates` Calculate $\mathbf{P}_l = |\mathbf{Y}_{tf_l}|^2$, for all l

31 **for** $l \leftarrow 1$ **to** L **do**

32 $\lambda_l = \sqrt{\frac{1}{FN}\sum_{f,n} p_l(f,n)}$

33 **for** $f \leftarrow 1$ **to** F **do**

34 $\mathbf{w}_l(f) = \mathbf{w}_l(f)\lambda_l^{-1}$

35 **end**

36 $\mathbf{P}_l = \mathbf{P}_l\lambda_l^{-2}$ $\mathbf{R}_l = \mathbf{R}_l\lambda_l^{-2}$

37 **end**

38 Calculate $t(f,k) = t(f,k)\sum_l z(l,k)\lambda_l^{-2}$ for all f and k Calculate $z(l,k) = \frac{z(l,k)\lambda_l^{-2}}{\sum_{l'} z(l',k)\lambda_{l'}^{-2}}$ for all l and k

39 **until convergence**

40 Calculate $\overline{\mathbf{y}}_{tf_l}(f,n) = \mathbf{w}_l(f)^{-1}y_{tf_l}(f,n)$ for all f,n and l `// Back-projection`

2.4 Conclusion

This chapter covers three popular frequency-domain BSS algorithms: Independent Vector Analysis, Fixed Point Independent Vector Analysis, and Independent Low-Rank Matrix Analysis. For each algorithm, we described the contrast functions and the optimization methods. In the following chapters, we will simulate and compare the performance of these three algorithms using various metrics.

Chapter 3

IVA-based BSS Algorithm Improvements

After studying a set of algorithms for blind speech separation, we suggest some additional processings to improve the separation process. First, we propose to recover multiple signals for each estimated source signal while applying back-projection. Following that, we will demonstrate that the well-known minimal distortion principle is just a particular case of this method. After that, using Single-Input Multiple-Output (SIMO) deconvolution, we will exploit the spatial diversity provided by the previous approach to eliminate the room effect and reduce the variance of the separation algorithm. Finally, we propose to add a de-noising module to improve the quality of the output estimated signals.

3.1 Frequency Domain Reconstruction

In Chapter 2, we discussed two distinct ways of dealing with the scale ambiguity in convolutive BSS. Indeed, IVA and Fast IVA are based on the Minimal Distortion Principle (section 2.1.8), whereas ILRMA employs the back-projection (section 2.3.5). In this section, we will focus on the second method.

In ILRMA, back-projection is achieved by multiplying the l^{th} source by the l^{th} column of the mixing matrix, which allows us to find each source at its observed amplitude at L microphones, then we select the signal observed at the m^{th} microphone, termed reference microphone. We propose to modify this technique by multiplying the outputs by the pseudo-inverse of the unmixing matrix. Therefore, each estimated speech signal will be recovered at all microphones as if the other sources were absent. We explain below how this multiplication solves the scaling indeterminacy and prove that it is a generalization of the MDP.

Let $\mathbf{A}(f)$ be the mixing matrix (without scaling ambiguity), $\hat{\boldsymbol{A}}(f)$ be the estimated mixing matrix , at the f^{th} frequency bin, and $y_{tf_l}(f,n)$ be the estimate of the l^{th} source signal $s_{tf_l}(f,n)$ at the time-frequency slot (f,n). First, we start by estimating the mixing matrix by the pseudo-inverse of the unmixing matrix:

$$\hat{\boldsymbol{A}}(f) = \mathbf{W}(f)^{\#} \tag{3.1}$$

Since there is a scaling indeterminacy, we have:

$$y_{tf_l}(f,n) = \alpha_l(f) s_{tf_l}(f,n) \tag{3.2a}$$

$$\hat{\boldsymbol{a}}_l(f) = \frac{1}{\alpha_l(f)} \boldsymbol{a}_l(f) \tag{3.2b}$$

To recover the separated signal, $\underline{\mathbf{s}}_{tf_l}(f,n)$, and avoiding the scaling indeterminacy, we multiply the l^{th} separated signal $y_{tf_l}(f,n)$ by the corresponding column of the estimated mixing matrix $\hat{\boldsymbol{a}}_l(f)$, as follows:

$$\underline{\mathbf{s}}_{tf_l}(f,n) = \hat{\boldsymbol{a}}_l(f) y_{tf_l}(f,n) = \frac{1}{\alpha_l(f)} \boldsymbol{a}_l(f) \alpha_l(f) s_{tf_l}(f,n) = \boldsymbol{a}_l(f) s_{tf_l}(f,n) \tag{3.3}$$

The method explained previously is a generalization of MDP, since it allows to find $\underline{\mathbf{s}}_{tf_l}(f,n) = [\underline{\mathrm{s}}_{tf_{l_1}}(f,n), ..., \underline{\mathrm{s}}_{tf_{l_M}}(f,n)]^T$, whereas the MDP allow us to find only the l^{th} component, $\underline{\mathrm{s}}_{tf_{l_l}}(f,n)$. From equation (3.3), we have that the l^{th} signal recovered by back-projection is the following:

$$\underline{\mathrm{s}}_{tf_{l_l}}(f,n) = \hat{a}_{ll}(f) y_{tf_l}(f,n) \tag{3.4}$$

Thus, the l^{th} component of all the L sources, denoted $\underline{\mathbf{s}}_{tf}(f,n)\Big|_l = [\underline{\mathrm{s}}_{tf_{1_1}}(f,n), \underline{\mathrm{s}}_{tf_{2_2}}(f,n), ..., \underline{\mathrm{s}}_{tf_{L_L}}(f,n)]^T$, can be obtained using back-projection, as:

$$\underline{\mathbf{s}}_{tf}(f,n)\Big|_l = \begin{bmatrix} \hat{a}_{11}(f) & 0 & \dots & 0 \\ 0 & \hat{a}_{22}(f) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \hat{a}_{LL}(f) \end{bmatrix} \mathbf{y}_{tf}(f,n) \tag{3.5a}$$

$$= diag(\hat{\boldsymbol{A}}(f)) \mathbf{y}_{tf}(f,n) \tag{3.5b}$$

$$= diag(\mathbf{W}^{-1}(f)) \mathbf{W}(f) \mathbf{x}_{tf}(f,n) \tag{3.5c}$$

$$= \mathbf{W}_s(f) \mathbf{x}_{tf}(f,n) \tag{3.5d}$$

It appears from (3.5) that the MDP is a particular case of the back-projection since we get the unmixing matrix used for MDP, $\mathbf{W}_s(f) = diag(\mathbf{W}^{-1}(f))(\mathbf{W})(f)$.

In conclusion, using the approach presented in this section, we recover M signals rather than a single signal for each distinct source, which provides us with spatial diversity. The latter will allow us to improve the intelligibility of the estimated speech signal. This by removing reverberations, on the one hand, and by reducing the variance of the estimation on the other hand.

3.2 Single-Input Multiple-Output Deconvolution

After propagation through a convolutive channel, the intelligibility of a signal can be improved using channel equalization. The latter requires two steps: first, channel estimation and, second, an efficient way to design an equalizer (Vincent et al., 2018), which essentially consists of inverse filtering of the room impulse response. In the case of speech dereverberation, the channel represents the effect of sound propagation from source to microphone. Several microphones are used to record a single source. As a result, the acoustic system has a single-input multiple-output structure, while the dereverberation system has a multiple-input single-output structure.

3.2.1 Blind System Identification

Blind System Identification (BSI) aims to recover the source signal only through multiple observations by estimating the channels. BSI methods can be classified into second-order statics-based methods (SOS) and higher-order statistics-based methods (HOS). Since HOS cannot be computed accurately from a small number of observations, convergence is slow. Furthermore, a cost function based on HOS information is generally not convex. Therefore, an algorithm based on HOS information may converge to a local minimum. Since it has been recognized that the identification problem can be solved using SOS, research on blind channel identification has been directed towards SOS methods, whose motivation is the potential for fast convergence and better accuracy.

There is plenty of literature about SOS-based blind channel identification, such as the subspace (SS) algorithm (Moulines et al., 1995b), the cross-relation (CR) algorithm (Liu et al., 1993), and the least-squares component normalization (LSCN) algorithm (Avendano et al., 1999). Here we focus on BSI methods that jointly minimize the cross-relation error between different pairs of sensor signals.

3.2.1.1 Problem Formulation

In this section, we consider the blind estimation of the impulse responses of a SIMO FIR system. The i^{th} observation $x_i(n)$ is the result of a linear convolution between the original source signal $s(n)$ and the corresponding channel response h_i, corrupted by an additive noise $b_i(n)$ (Huang and Benesty, 2002).

$$x_i(n) = h_i \circledast s(n) + b_i(n) \tag{3.6a}$$

$$= \sum_{l=0}^{L-1} h_i(l)s(n-l) + b_i(n) \quad , i = 1, ..., M \tag{3.6b}$$

where the symbol $\circledast$ denotes the linear convolution operator, L is the length of the impulse response, and M is the number of channels.

In our case, $x_i(n)$ represents the signals after back-projection, and $b_i(n)$ is the noise introduced by the estimation process.

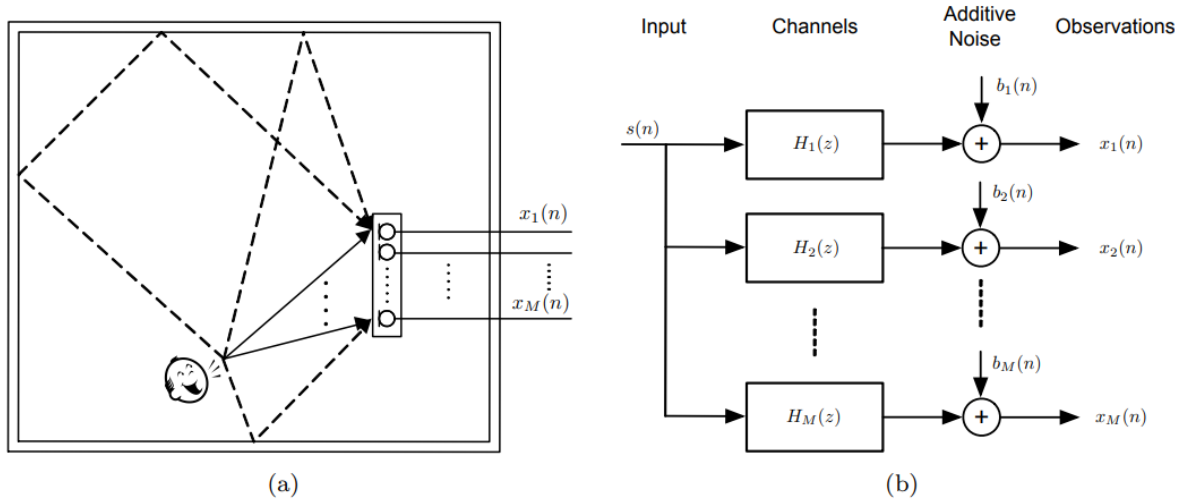

Figure 3.1: (a) SIMO acoustic system diagram; (b) Channel equalization problem formulation

In vector form, (3.6) can be written as follows:

$$x_i(n) = \mathbf{h}_i^T \mathbf{s}(n) + b_i(n) \tag{3.7}$$

Where $\mathbf{s}(n) = [s(n), s(n-1), ..., s(n-L+1)]^T$ and $\mathbf{h}_i = [h_i(0), h_i(1), ..., h_i(L-1)]^T$ is the impulse reponse of the i^{th} channel.

The aim of a BSI algorithm is to estimate $\mathbf{h} = [\mathbf{h}_1^T, \mathbf{h}_2^T, ..., \mathbf{h}_M^T]^T$ from the noisy observations $x_i(n)$ $i = 1, ..., M$, $n = 1, ..., N$, where N denotes the length of the observed signal.

3.2.1.2 Channel Identifiability Conditions

The identifiability of a channel corresponds to the existence of a unique solution to the unknown system's impulse responses with respect to a particular type of algorithm. According to (Xu et al., 1995), two conditions are necessary and sufficient to guarantee an identifiable system.

1. **Channel diversity:** refers to the fact that the channels $\mathbf{h}_i$, $i = 1, ..., M$ are co-prime, i.e., the multi-channel transfer functions do not share common zeros (Moulines et al., 1995a).

2. **Condition for the input signals:** first, the input data must be non-zero, otherwise, no channel information is contained in the output. Second, the Hankel matrix $\mathbf{S}(n)$ of the source signal must be of full rank. If not, there will be an insufficient number of equations $\mathbf{S}(n)\mathbf{h}_i = \mathbf{x}_i(n)$, where $\mathbf{x}_i(n) = [x_i(n), x_i(n-1), ..., x_i(n-L+1)]^T$ and the Hankel matrix $\mathbf{S}(n)$ is given by :

$$\mathbf{S}(n) = \begin{bmatrix} s(n) & s(n-1) & \dots & s(n-L+1) \\ s(n-1) & s(n-2) & \dots & s(n-L) \\ \vdots & \vdots & \ddots & \vdots \\ s(n-L+1) & s(n-L) & \dots & s(n-2L+2) \end{bmatrix} \tag{3.8}$$

   And thus no unique solutions.

3.2.1.3 Cross-Relation Method

Cross-relation error based BSI algorithms (Xu et al., 1995) exploit the Single-Input-Multiple-Output (SIMO) structure to estimate the room impulse responses.

From equation (3.6) , for any pair of two noise-free outputs $x_i(n)$ and $x_j(n)$

$$x_i(n) = h_i \circledast s(n) \quad and \quad x_j(n) = h_j \circledast s(n) \quad 1 \leq i \neq j \leq M \tag{3.9}$$

Then,

$$h_j \circledast x_i(n) = h_j \circledast \underbrace{(h_i \circledast s(n))}_{x_i(n)} \tag{3.10}$$

Using the commutativity of convolution, it follows:

$$h_j \circledast x_i(n) = h_i \circledast x_j(n) \tag{3.11}$$

where h_i and h_j denote the i^{th} and j^{th} acoustic impulse responses, respectively.

Equation (3.11) is a linear equation satisfied by every pair of channels, and it shows that the outputs of each channel pair are related by their channel responses. If we have adequate data samples of the outputs, we can write out an over-determined set of linear equations involving h_i and h_j.

At time n, in vector form, equation (3.11) can be expressed as :

$$\mathbf{x}_i^T(n)\mathbf{h}_j = \mathbf{x}_j^T(n)\mathbf{h}_i \quad 1 \leq i \neq j \leq M \tag{3.12}$$

Using this relation, an error term can then be defined as :

$$e_{ij}(n) = \begin{cases} \mathbf{x}_i^T(n)\mathbf{h}_j - \mathbf{x}_j^T(n)\mathbf{h}_i & i \neq j, i, j = 1, 2, ..., M \\ 0 & i = j, i, j = 1, 2, ..., M \end{cases} \tag{3.13}$$

Here, we have $\frac{(M-1)M}{2}$ distinct error signals $e_{ij}(n)$, which exclude the case $e_{ij}(n) = 0$ and count the $e_{ij}(n) = -e_{ji}(n)$ pair only once.

In order to avoid the trivial estimate with all zero elements, a unit-norm constraint is imposed on $\mathbf{h}$, and the normalized error signal becomes:

$$\epsilon_{ij}(n) = \begin{cases} \frac{\mathbf{x}_i^T(n)\mathbf{h}_j}{||\mathbf{h}||} - \frac{\mathbf{x}_j^T(n)\mathbf{h}_i}{||\mathbf{h}||} & i \neq j, i, j = 1, 2, ..., M \\ 0 & i = j, i, j = 1, 2, ..., M \end{cases} \tag{3.14}$$

If the channel identifiability conditions are satisfied, $\mathbf{h}_i$ and $\mathbf{h}_j$ can be determined uniquely up to an unknown scaling factor by minimizing a total squared error $J(n)$ across all unique microphone pairs.

$$\widehat{\mathbf{h}} = argmin_{\mathbf{h}} E(\mathcal{J}(n)), \quad \text{subject to } ||\widehat{\mathbf{h}}|| = 1 \tag{3.15}$$

where the cost function is specified as

$$\mathcal{J}(n) = \sum_{i=1}^{M-1} \sum_{j=i+1}^{M} \epsilon_{ij}^2(n) \tag{3.16}$$

3.2.1.4 Noise Robust Multichannel Frequency-Domain LMS (RNMCFLMS)

In order to determine the error signal in equation (3.14), we need to calculate a linear convolution of the i^{th} channel output x_i and the j^{th} channel's impulse response h_j. In speech signal dereverberation, channel length is large, thus these convolutions are

computationally intensive in the time domain. Therefore, we prefer to perform digital filtering in the frequency domain. When computing the DFT, we have to make linear convolution behaves like circular convolution so that the convolution becomes multiplication in the frequency domain. One method that makes it possible is the overlap-save method (Daher et al., 2010).

The overlap-save procedure cuts the signal up into equal-length segments with some overlap. Then it takes the DFT of the signal segments and saves the parts of the convolution that correspond to the circular convolution. In our case, the input data blocks are overlapped by L points, where L represents the length of the channel. For each block of length L, where we have 2L data inputs, we discard the L first convolution results and retain only the L last components, which are identical to the results of linear convolution.

A bar below for vectors and a calligraphic upper-letter for matrices will be used in the following to indicate that they are in the frequency domain. Let:

- $\mathbf{x}_i(m) = [x_i(mL-L), ..., x_i(mL), ..., x_i(mL+L-1)]^T$ be the overlapped i^{th} channel output in the m^{th} block of length $2L$.
- $\boldsymbol{\mathcal{D}}_{x_i}(m)$ a diagonal matrix whose elements are given by the DFT of $\mathbf{x}_i(m)$.
- $\widehat{\underline{\boldsymbol{h}}}_j(m)$ consist of the 2L-point DFT of the vector $[\widehat{\boldsymbol{h}}_j^T(m)\ \mathbf{0}_{1\times L}]^T$, where $\mathbf{0}_{1\times L}$ is a zero vector of length L and $\widehat{\boldsymbol{h}}_j(m) = [\widehat{h}_{j,0}(m), \widehat{h}_{j,1}(m), ..., \widehat{h}_{j,L-1}(m)]^T$ is the j^{th} channel's impulse response at the m^{th} block.

According to equation (3.12) the block error in the frequency domain is given by:

$$\underline{\boldsymbol{e}}_{ij}(m) = \boldsymbol{\mathcal{D}}_{x_i}(m)\widehat{\underline{\boldsymbol{h}}}_j(m) - \boldsymbol{\mathcal{D}}_{x_j}(m)\widehat{\underline{\boldsymbol{h}}}_i(m) \tag{3.17}$$

Similar to its counterpart in the time domain, the mean square error criterion in the frequency domain is defined as follows:

$$argmin_{\widehat{\underline{\boldsymbol{h}}}}\mathcal{J}_f = E\{\mathcal{J}_f(m)\} \tag{3.18}$$

Where : $\widehat{\underline{\boldsymbol{h}}}(m) = [\widehat{\underline{\boldsymbol{h}}}_1^T(m), \widehat{\underline{\boldsymbol{h}}}_2^T(m), ..., \widehat{\underline{\boldsymbol{h}}}_M^T(m)]^T$ and the instantaneous square error at the m^{th} block:

$$\mathcal{J}_f(m) = \sum_{i=1}^{M-1}\sum_{j=i+1}^{M} \underline{\boldsymbol{e}}_{ij}^H(m)\underline{\boldsymbol{e}}_{ij}(m) \tag{3.19}$$

In the standard LMS algorithm (Huang and Benesty, 2003), the expectation is estimated by a single sample, and the optimal value is determined by moving in the

opposite direction of the error gradient at each iteration:

$$\underline{\widehat{\boldsymbol{h}}}(m+1) = \underline{\widehat{\boldsymbol{h}}}(m) - \mu_f \nabla \mathcal{J}_f(m) \tag{3.20}$$

Where $\nabla \mathcal{J}_f(m) = \frac{\partial \mathcal{J}_f(m)}{\partial \underline{\widehat{\boldsymbol{h}}}^*(m)}$ is the gradient and μ_f the step size.

The step size in the LMS algorithm must be chosen such that it makes a trade-off between rate of convergence, mean-square error, and the algorithm's ability to track the system as its impulse responses change. To achieve a balance between these three design goals, we often use a variable step size $\mu_f(m)$:

$$\underline{\widehat{\boldsymbol{h}}}(m+1) = \underline{\widehat{\boldsymbol{h}}}(m) - \mu_f(m) \nabla \mathcal{J}_f(m) \tag{3.21}$$

In the Variable Step Size MCFLMS (VSS-MCFLMS)(Haque and Hasan, 2007), $\mu_f(m)$ is defined as follows:

$$\mu_f(m) = \frac{\underline{\widehat{\boldsymbol{h}}}^H(m)}{||\nabla \mathcal{J}_f(m)||^2 + \epsilon} \nabla \mathcal{J}_f(m) \tag{3.22}$$

Where ϵ is a small positive real number used to prevent singularity.

To avoid a trivial estimate, we apply the following normalization at each step:

$$\underline{\widehat{\boldsymbol{h}}}(m+1) = \frac{\underline{\widehat{\boldsymbol{h}}}(m+1)}{\sqrt{ML}||\underline{\widehat{\boldsymbol{h}}}(m)||} \tag{3.23}$$

In Normalized Multi-Channel Frequency-domain LMS (NMCFLMS) algorithm (Huang and Benesty, 2003), the Newton's method is first used in order to accelarate the convergence. To do so, the variable step size $\mu_{f_k}(m)$ used for the estimation of the k^{th} channel is given by:

$$\underline{\widehat{\boldsymbol{h}}}_k(m+1) = \underline{\widehat{\boldsymbol{h}}}_k(m) - \mu_{f_k}(m) \frac{\partial \mathcal{J}_f(m)}{\partial \underline{\widehat{\boldsymbol{h}}}_k^*(m)} \quad , k = 1, ..., M \tag{3.24}$$

$$\mu_{f_k}(m) = \rho \quad \left[\frac{\partial}{\partial \underline{\widehat{\boldsymbol{h}}}_k^T(m)} \left\{ \frac{\partial \mathcal{J}_f(m)}{\partial \underline{\widehat{\boldsymbol{h}}}_k^*(m)} \right\} \right]^{-1} \tag{3.25}$$

Where $0 < \rho < 2$ is the step-size for the NMCFLMS algorithm. After expressing the gradient and the hessian of $\mathcal{J}_f(m)$ with respect to the filter coefficients, we find that the update equation of the NMCFLMS algorithm is given as:

$$\underline{\widehat{\boldsymbol{h}}}_k(m+1) = \underline{\widehat{\boldsymbol{h}}}_k(m) - \rho \boldsymbol{P}_k^{-1}(m) \times \sum_{i=1}^{M} \boldsymbol{D}_{x_i}^*(m) \underline{\boldsymbol{e}}_{ik}^{01}(m) \quad k = 1, ..., M \tag{3.26}$$

Where

$$\boldsymbol{P}_k(m) = \sum_{i=1, i\neq k}^{M} \boldsymbol{D}^*_{x_i}(m)\boldsymbol{D}_{x_i}(m)$$

$$\underline{\boldsymbol{e}}^{01}_{ik}(m) = DFT_{2L\times 2L}\left(\mathbf{0}_{1\times L} \quad \left[IDFT_{L\times L}(\underline{\boldsymbol{e}}_{ik}(m))\right]^T\right)^T$$

In practice, a recursive scheme is used to estimate a stable power spectrum :

$$P_k(m) = \lambda P_k(m-1) + (1-\lambda) \times \sum_{i=1, i\neq k}^{M} \boldsymbol{D}^*_{x_i}(m)\boldsymbol{D}_{x_i}(m) \quad , \quad k = 1, 2, ..., M \tag{3.27}$$

Where λ is a forgetting factor.

In the presence of noise, all of these algorithms suffer from misconvergence. This characteristic is due to the non-uniform spectral attenuation of the estimated channel coefficients. A new algorithm, termed Robust NMCFLMS (Haque and Hasan, 2008), proposed a modified objective function by adding a penalty cost function $J_p(m)$ to the original cost function $J_f(m)$ previously defined. This ensures the robustness of the adaptive algorithm by requiring the spectral energy to be uniformly distributed across all frequencies in the estimated channel response.

$$J_{\text{mod}}(m) = J_f(m) + \beta(m)(-J_p(m)) \tag{3.28}$$

The penalty function proposed to this novel algorithm is defined as follow:

$$maximize \quad J_p(m) = \prod_{i=1}^{ML} |\underline{\widehat{h}}_i(m)|^2 \tag{3.29}$$

Subject to :

$$|\underline{\widehat{h}}_1(m)|^2 + |\underline{\widehat{h}}_2(m)|^2 + ... + |\underline{\widehat{h}}_{ML}(m)|^2 = \frac{1}{ML} \tag{3.30}$$

Where $\underline{\widehat{\mathbf{h}}}(m)$ is redefined as $\underline{\widehat{\mathbf{h}}} = [\underline{\widehat{\mathbf{h}}}_1(m), \underline{\widehat{\mathbf{h}}}_2(m), ..., \underline{\widehat{\mathbf{h}}}_M(m)]^T$.

Substituting the expression of $|\underline{\widehat{h}}_{ML}(m)|^2$ from equation (3.29) into (3.30), we obtain:

$$J_p(m) = |\underline{\widehat{h}}_1(m)|^2 \times |\underline{\widehat{h}}_2(m)|^2 \times ... \times |\underline{\widehat{h}}_{ML-1}(m)|^2 \times \left(\frac{1}{ML} - |\underline{\widehat{h}}_1(m)|^2 - |\underline{\widehat{h}}_2(m)|^2 - \quad ... - |\underline{\widehat{h}}_{ML-1}(m)|^2\right) \tag{3.31}$$

By differentiating this equation with respect to $\underline{\widehat{h}}_k^*(m)$, we obtain the following:

$$\nabla J_{p_k}(m) = 2\underline{\widehat{h}}_k(m)\left\{\frac{1}{ML} - |\underline{\widehat{h}}_1(m)|^2 - ... - 2|\underline{\widehat{h}}_k(m)|^2 - ... - |\underline{\widehat{h}}_{ML-1}(m)|^2\right\}\prod_{i=1,i\neq k}^{ML}|\underline{\widehat{h}}_i(m)|^2 \tag{3.32}$$

To get the maximum of this penalty function $J_p(m)$, we cancel out its derivative $\nabla J_{p_k}(m) = 0$.

It is easily seen from equation (3.32) that this is only possible if

$$|\underline{\widehat{h}}_1(m)|^2 + ... + 2|\underline{\widehat{h}}_k(m)|^2 + ... + |\underline{\widehat{h}}_{ML-1}(m)|^2 = \frac{1}{ML} \tag{3.33}$$

In this way, one can construct (ML-1) simultaneous linear equations of the same form as (3.33) for each value of k. Now, adding all these equations together, we get the following:

$$|\underline{\widehat{h}}_1(m)|^2 + ... + |\underline{\widehat{h}}_k(m)|^2 + ... + |\underline{\widehat{h}}_{ML-1}(m)|^2 = \frac{ML-1}{M^2L^2} \tag{3.34}$$

Taking the two equations (3.33) and (3.34) and subtracting the second from the first, we obtain the condition for penalty function maximization as $|\underline{\widehat{h}}_k(m)|^2 = \frac{1}{M^2L^2}$. This shows that the penalty function will be at its maximum when the estimated channel coefficients exhibit uniform magnitude spectra in the frequency-domain. The update rule can, be obtained as:

$$\underline{\widehat{\boldsymbol{h}}}(m+1) = \frac{\underline{\widehat{\boldsymbol{h}}}(m) - \mu_f(m)\nabla J_f(m) + \beta(m)\mu_f(m)\nabla J_p(m)}{\sqrt{ML}||\underline{\widehat{\boldsymbol{h}}}(m)||} \tag{3.35}$$

In order to simplify the expression of the penalty gradient, we apply a natural logarithm to both sides of equation (3.29) and the penalty cost function can be rewritten as follows:

$$\widetilde{J}_p(m) = \sum_{i=1}^{ML} ln(|\underline{\widehat{h}}_i(m)|^2) \tag{3.36}$$

The penalty function gradient is obtained as:

$$\nabla\widetilde{J}_p(m) = [\nabla\widetilde{J}_{p1}^T(m), ..., \nabla\widetilde{J}_{pk}^T(m), ..., \nabla\widetilde{J}_{pM}^T(m)]^T \tag{3.37}$$

Where:

$$\nabla\widetilde{J}_{p_k}^T(m) = \frac{\partial\widetilde{J}_p(m)}{\partial real\{\underline{\widehat{h}}_k^*(m)\}} + j\frac{\partial\widetilde{J}_p(m)}{\partial imag\{\underline{\widehat{h}}_k^*(m)\}} = \frac{2}{|\underline{\widehat{h}}_k(m)|^2}\underline{\widehat{h}}_k(m)$$

The coupling factor, $\beta(m)$, is estimated such that the total gradient becomes zero,

$\nabla J_{mod}(m) = 0$. This gives, $\nabla J_f(m) = \beta(m)\nabla \widetilde{J}_p(m)$, thus we can obtain $\beta(m)$ as:

$$\beta(m) = \left| \frac{\nabla \widetilde{J}_p^H(m) \nabla J_f(m)}{||\nabla \widetilde{J}_p(m)||^2} \right| \tag{3.38}$$

The update equation for the robust NMCFLMS algorithm cas be expressed as follow:

$$\widehat{\underline{\boldsymbol{h}}}_k(m+1) = \widehat{\underline{\boldsymbol{h}}}_k(m) - \rho \boldsymbol{P}_k^{-1}(m) \times \sum_{i=1}^{M} \boldsymbol{D}_{x_i}^*(m) \underline{\boldsymbol{e}}_{ik}^{01}(m) + \rho \beta_n(m) \nabla \widetilde{J}_{pk}(m) \quad k = 1, ..., M \tag{3.39}$$

Where $\beta_n(m)$ is estimated similar to (3.38) but using the NMCFLMS algorithm update parameters.

3.2.2 System Equalization

The second step in this two-step dereverberation procedure is to design a multichannel equalizer $\mathbf{g} = [\mathbf{g}_1^T, \mathbf{g}_2^T, ..., \mathbf{g}_M^T]^T$ where $\mathbf{g}_m = [g_m(0), g_m(1), ..., g_m(L_i - 1)]^T$ is of length L_i.

When the equalizer is the inverse of the system, $\mathbf{g}$ and $\widehat{\mathbf{h}}$ must satisfy the relationship (Miyoshi and Kaneda, 1988):

$$\sum_{m=1}^{M} \widehat{h}_m(l) \circledast g_m(l) = d(l), \quad l = 0, ..., L + L_i - 2 \tag{3.40}$$

Where $\mathbf{d} = [d(0), ..., d(L + L_i - 2]^T$ is the target response vector:

$$d(l) = \begin{cases} 0 & \text{if} \quad 0 \leq l < \tau \\ 1 & \text{if} \quad l = \tau \\ 0 & \text{otherwise} \end{cases} \tag{3.41}$$

τ represents the delay of the target response, which is generally considered to be zero.

In matrix form, the equation system (3.40) is written as:

$$\widehat{\mathbf{H}}\mathbf{g} = \mathbf{d} \tag{3.42}$$

With $\widehat{\mathbf{H}} = [\widehat{\mathbf{H}}_1, ..., \widehat{\mathbf{H}}_M]$, where $\widehat{\mathbf{H}}_m$, is an $(L + L_i - 1) \times L_i$ convolution matrix of m^{th}

channel impulse response $\widehat{\mathbf{h}}_m$:

$$\widehat{\mathbf{H}}_m = \begin{bmatrix} \widehat{h}_m(0) & 0 & \dots & 0 \\ \widehat{h}_m(1) & \widehat{h}_m(0) & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \widehat{h}_m(L-1) & \dots & \vdots & \vdots \\ 0 & \widehat{h}_m(L-1) & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \widehat{h}_m(L-1) \end{bmatrix} \tag{3.43}$$

The equalization filter $\mathbf{g}$ can be obtained by minimizing the least squares (LS) cost function:

$$\mathcal{C} = \left|\left|\widehat{\mathbf{H}}\mathbf{g} - \mathbf{d}\right|\right|_2^2 \tag{3.44}$$

Hence, the LS solution is given by:

$$\mathbf{g} = \widehat{\mathbf{H}}^{\#}\mathbf{d} \tag{3.45}$$

3.3 Denoising

Minimum Mean Square Estimators Log Spectral Amplitude (log-MMSE) is commonly used for the enhancement of noisy speech signals. The latter reduces the residual noise without affecting the speech signal itself, that is, without introducing much speech distortion. The log-MMSE consists of an estimation problem in which the clean signal is estimated from a given function of the noisy signal. Let assume that the noisy signal $y(t)$ is the sum of the clean speech signal $x(t)$ and the noise signal $n(t)$.

$$y(t) = x(t) + n(t) \qquad 0 \leq t \leq T \tag{3.46}$$

The goal is to minimize the expected value of some distortion measure between the clean and estimated signals. For this approach to work, a perceptually meaningful distortion measure and a reliable statistical model for the signal and noise must be specified. In the following, the STFT representations of signal and noise are assumed to be statistically independent zero-mean complex Gaussian random variables, and the mean-square error (MSE) of the log-spectra distortion measure is used:

$$\hat{\boldsymbol{X}}(f) = argmin_{\hat{\boldsymbol{X}}(f)} E\left\{\left(\log \boldsymbol{X}(f) - \log \hat{\boldsymbol{X}}(f)\right)^2\right\} \tag{3.47}$$

Where :

$\hat{\boldsymbol{X}}(f)$ is the estimate spectral magnitude at the f^{th} frequency bin.

$\boldsymbol{X}(f)$ is the true magnitude of the clean signal at the f^{th} frequency bin.

In order to solve the minimization problem, we have to express the expectation. This is done with respect to the joint PDF $p(\mathbf{Y}, \mathbf{X}(f))$ and is given by:

$$\mathbf{MSE} = \int\int \left(\log \boldsymbol{X}(f) - \log \hat{\boldsymbol{X}}(f)\right)^2 p(\mathbf{Y}, \mathbf{X}(f)) \; d\mathbf{Y} d\mathbf{X}(f) \tag{3.48}$$

Minimization of MSE with respect to $\hat{\boldsymbol{X}}(f)$ leads to the optimal MMSE estimator $\hat{\boldsymbol{X}}(f)$ given by:

$$\log \hat{\boldsymbol{X}}(f) = E\left[\log \boldsymbol{X}(f)|\mathbf{Y}\right] = E\left[\log \boldsymbol{X}(f)|\mathbf{Y}(1)...\mathbf{Y}(F)\right] = \int \log \mathbf{X}(f) p(\mathbf{X}(f)|\mathbf{Y}) \; d\boldsymbol{X}(f) \tag{3.49}$$

Thus the optimal log-MMSE estimator can be obtained by evaluating the conditional mean of $\log \boldsymbol{X}(f)$. Assuming statistical independence between the frequency components, we have that:

$$\log \hat{\boldsymbol{X}}(f) = E\left[\log \boldsymbol{X}(f)|\mathbf{Y}(f)\right] \tag{3.50}$$

$$\hat{\boldsymbol{X}}(f) = \exp\left\{E\left[\log \boldsymbol{X}(f)|\mathbf{Y}(f)\right]\right\} \tag{3.51}$$

The evaluation of the previous conditional mean is not straightforward but can be simplified using the moment-generating function of $\boldsymbol{X}(f)$. The resulting estimator is the following:

$$\hat{\boldsymbol{X}}(f) = \frac{\xi_f}{\xi_f + 1} \exp\left\{\frac{1}{2}\int_{v_f}^{\infty} \frac{e^{-t}}{t} \; dt\right\} \mathbf{Y}(f) \tag{3.52}$$

Where:

- $\lambda_x(f)$ and $\lambda_n(f)$ represents the variance of the f^{th} spectral component of the clean signal and the noise, respectively.
- $\xi_f = \frac{\lambda_x(f)}{\lambda_n(f)}$ is the a priori SNR, i.e., the true SNR of the f^{th} spectral component.
- $v_f = \frac{\xi_f}{1+\xi_f}\gamma_f$, where: $\gamma_f = \frac{\mathbf{Y}(f)^2}{\lambda_n(f)}$ is the a posteriori SNR, i.e., the observed SNR of the f^{th} spectral component.

3.4 Conclusion

In this chapter, we suggested new processing techniques to improve the separation quality and intelligibility of the estimated source signals. First, we have exploited the

spatial diversity resulting from the application of back-projection. Indeed, the latter gives rise to a SIMO system for which we apply a deconvolution algorithm that removes the room effect while also improving estimation quality. Then, we proposed to use log-MMSE as a denoising algorithm since this latter is well-known for its excellent performance in the case of speech signals. High separation performance can be achieved even in a noisy environment; by using this denoising module before or after the BSS algorithm. This module's placement will be decided at a later stage. In chapter 5, we will evaluate the effect of these improvements using objective performance measures.

Chapter 4

Softwares, Data Generation and Evaluation Criteria

The evaluation of blind speech separation algorithms requires an experimental setup that typically includes speech signals, acoustic environments, and separation performance criteria. This chapter describes the dataset, tools, and methods used for convolutive speech separation.

First, the software tools and environments used to conduct the simulation and the experimental test are presented. Then, the materials utilized to generate the synthetic signals for performance evaluation, including the dataset used to obtain the audio signals and the library used to model the Room Impulse Responses (RIR), are discussed. Finally, the performance measures employed to evaluate and analyze the separation performance of the algorithms are explained.

4.1 Software Tools

This section discusses the programming languages, libraries, and environments used to conduct performance evaluations of previously defined algorithms (Chapters 2 and 3) and implement real-world tests.

4.1.1 MATLAB

MATLAB, an acronym for Matrix Laboratory, is a programming platform used particularly in engineering applications such as signal processing and data science. The

core of MATLAB is the MATLAB language, a high-performance matrix programming language developed by MathWorks to analyze data, develop and optimize algorithms, and design models while giving speed, accuracy, and precision to the results and allowing their visualization. There are many toolboxes in this program that considerably enhance its functionality. For example, the one we will use in our work: BSS Eval Toolbox distributed online under the GNU Public License.

4.1.2 Python

Python is a high-level, interpreted, interactive, object-oriented scripting language. It has extensive support and a wide selection of libraries for mathematical computation and audio processing, making it suitable for BSS. In our work, we used Python 3.9 to generate synthetic speech mixtures and implement the code on a Raspberry Pi to perform real-world tests. The libraries and the text editor we used are described below.

4.1.2.1 Libraries

- **NumPy** (Harris et al., 2020) is an open-source, powerful Python library that stands for Numerical Python. It is a library consisting of multidimensional array objects and a collection of routines for processing those arrays, such as algebraic operations.

- **SciPy** (Virtanen et al., 2020) is an open-source Python library, which is dedicated to scientific computing and mathematics. We used SciPy in our codes for audio support, such as reading and writing speech signals.

- **Pyroomacoustics** is a software package for audio algorithms simulation. The package includes both a fast RIR generator and several reference implementations of popular algorithms for beamforming, Direction Of Arrival (DOA) finding, and adaptive filtering. We used in our codes the RIR generator in order to create synthetic speech mixtures.

4.1.2.2 Visual Studio Code

In our work, we used Visual Studio Code (VS Code), a lightweight but powerful source-code editor developed by Microsoft for Windows, macOS, and Linux. It can be used with a variety of programming languages, including Python.

4.2 Data Generation

4.2.1 Database

The speech signals used for all experiments in Chapter 5 were obtained during the 2011 Signal Separation Evaluation Campaign (SiSEC2011) (Araki et al., 2012). SiSEC is the first community-based signal separation evaluation campaign from which we can draw rigorous scientific conclusions. SISEC2011 includes, among other things, single voice 10-second recordings of four male and four female speakers sampled at 16 kHz.

4.2.2 Room Impulse Responses

When a conversation occurs inside a room, the presence of nearby reflecting walls distorts the speech signals. Sounds do not only follow the direct path from the source to the microphone but also reach the microphone after bouncing off one or more walls (Neely and Allen, 1979). The "room effect" can be viewed as a convolution in the time domain of the speech signal with a room impulse response, which represents the transfer function between the sound source and microphone. The degree of mixing of speech sources is determined by the room's reverberation time (RT). The latter is the time required for an impulse response's energy to decay below a certain level. For example, the RT60 reverberation time is defined as the time it takes for the impulse response to decay by 60 dB from its initial level.

In our work, Pyroomacoustics (Scheibler et al., 2017) is used to create artificial RIR between the sources and microphones. This toolbox aims to accurately model real-world conditions in order to evaluate rigorously different BSS algorithms.

In Pyroomacoustics, we create a simulation scenario to generate mixtures of different speakers. To do so, we first define the size of a three-dimensional room to which a few sound sources and a microphone array are attached. Next, we determine the positions and directivity patterns of the sources and microphones. After that, we must specify the RT60 and use Sabine's formula to calculate the energy absorption of the walls. Finally, the image source method is used to model the RIR and generate artificial mixtures.

4.2.3 Image Source Method

The image source method (ISM) (Allen and Berkley, 1979) is a simulation method for small rooms that have been widely used in room acoustics. This model replaces reflections on walls with virtual sources playing the same sound as the source and builds an RIR from the corresponding delays and attenuations. The model is accurate only as long as the wavelength of the sound is small relative to the size of the reflectors, which it assumes to be uniformly absorbing across frequencies. Nevertheless, these assumptions are not too far from reality in many environments of interest, such as offices.

4.3 Performance Measures

In the case of audio source separation, the performance of a BSS algorithm can be evaluated using different metrics that measure the quality of the separation process. These metrics are generally classified into objective and subjective metrics.

Subjective measurement of audio quality is usually done through listening tests. However, in the source separation community, listening tests have not been widely used so far.

Objective measurement is carried out, for example, using the BSS Eval toolbox (Vincent et al., 2006). The latter provides a set of four performance measures that evaluate various source separation algorithms in an evaluation framework where the original sources, and perhaps even the noise that disturbed the mix, are available for comparison.

The processing of this measure consists of two successive steps. The first one attempts to decompose the estimated source signal $\widehat{s}_i$ as follows:

$$\widehat{s}_i = s_{target} + e_{interf} + e_{artif} \tag{4.1}$$

Where $s_{target} = f(s_i)$ is a version of the original source s_i modified by an allowed distortion f. e_{interf} and e_{artif} are the interference and artifact terms, respectively. The decomposition method into these three terms is based on orthogonal projections. Let us assume that the source signals s_i are mutually orthogonal and denote $\prod\{y_1, y_2, ..., y_k\}$ the orthogonal projector onto the subspace spanned by the vectors $y_1, y_2, ..., y_k$. The projector is a $T \times T$ matrix, where T is the length of these vectors. We consider the two orthogonal projectors:

$$\boldsymbol{P}_{s_i} = \prod\{s_i\} \tag{4.2}$$

$$\boldsymbol{P_s} = \prod\{(s_{i'})_{1\leq i'\leq n}\} \tag{4.3}$$

and then we have:

$$s_{target} = \boldsymbol{P}_{s_i}\widehat{s}_i \tag{4.4}$$

$$e_{interf} = \boldsymbol{P_s}\widehat{s}_i - \boldsymbol{P}_{s_i}\widehat{s}_i \tag{4.5}$$

$$e_{artif} = \widehat{s}_i - s_{target} - e_{interf} \tag{4.6}$$

Thus, the computation of s_{target} and e_{interf} is as follows:

$$s_{target} = \frac{\langle \widehat{s}_i, s_i\rangle\, s_i}{||s_i||^2} \tag{4.7}$$

$$e_{interf} = \sum_{i'\neq i} \frac{\langle \widehat{s}_i, s_{i'}\rangle\, s_{i'}}{||s_{i'}||^2} \tag{4.8}$$

Where we denote $\langle a, b\rangle = \sum_{t=0}^{T-1} a(t)b^*(t)$ the inner product between two complex-valued vectors a and b of length T.

The second step consists of estimating the energy ratios of each of these three terms. In our performance measures, we consider two performance measures, the Signal-to-Distortion Ratio (SDR) and the Signal-to-Interference ratio (SIR), which are expressed as follows:

$$SDR = 10\log_{10}\frac{||s_{target}||^2}{||e_{interf} + e_{artif}||^2} \tag{4.9}$$

$$SIR = 10\log_{10}\frac{||s_{target}||^2}{||e_{interf}||^2} \tag{4.10}$$

The quality of the source separation is directly proportional to the SIR and SDR values. The higher the value, the more accurate the estimation. Since the BSS algorithm does not completely eliminate the other sources but strongly attenuates them, the SIR represents an essential criterion for evaluating separation quality. The SDR is also a good performance criterion because it measures overall performance by considering interference, noise, and artifacts.

4.4 Conclusion

The set of tools used for data generation and code execution, as well as the different separation performance measures used for speech source separation systems, have been described in this chapter.

In the following chapter, they will be used for the comparative study of the algorithms presented in Chapter 2 and to determine the influence of the pre-and post-processing proposed in Chapter 3.

Chapter 5

Performance Study and Comparison of BSS Algorithms

In this chapter, we explore the capabilities of several convolutive BSS algorithms: IVA, Fast IVA, and ILRMA, using artificially generated mixtures with the Image Source method. These experiments allow us to properly assess SIR, SDR, and the runtime performances of these methods. Then, based on these results, an algorithm among the previously cited methods is selected for the hadware implementation. The latter will be subjected to multiple experiments with respect to the separation task challenge (an increase of the RT60, the addition of white noise,...) in order to evaluate its capabilities and robustness.

After that, the post-processing we recommended in chapter 3 has been assessed objectively. The addition of a denoising module is expected to improve the performance of the separation method in a noisy environment, and the SIMO deconvolution should improve the quality of the output signals. To confirm this, we conducted several numerical evaluations, where we compared the algorithm with and without those post-processings.

5.1 Experimental Setup

The RIRs of two simulation scenarios and the corresponding convolutive mixtures were generated using Pyroomacoutics (Annex A) to perform objective evaluations. In both cases, we choose a 5.5 m×3.5 m×3 m room with a reverberation time (RT60) set to 130 ms. Then, we placed a microphone array in the center of the room. The latter consists of 7 microphones, one in the center and the other six spaced equally around a circle of a 4.5 cm radius. This chosen disposition is similar to the one we use in the real-world tests. That is, to rigorously test each algorithm before selecting the one to implement.

In the first simulation scenario, two sources are present in the room, while in the second one, three are present. These sources were positioned at various angles, 0.5 m away from the microphone array (Figures 5.1 and 5.2). The mixtures are then produced using the speaking utterances of four males and four females. The detailed experiment parameters are given in Table 5.1.

Reverberation time	130 ms
Room dimensions	5.5m × 3.5m × 3 m
Positions of microphones	[2.727, 1.789, 1.1], [2.772, 1.789, 1.1], [2.705, 1.750, 1.1], [2.750, 1.750, 1.1], [2.795, 1.750, 1.1], [2.727, 1.711, 1.1], [2.772, 1.711, 1.1]
Sources positions	Case1: [3, 2.183, 1.2], [2.317, 1.5, 1.2] Case2: [2.428, 2.133, 1.2],[3.259, 1.921, 1.2], [2.663, 1.257, 1.2]
Signal duration	10 s
Sampling rate	16 000 Hz

Table 5.1: Experiment parameters

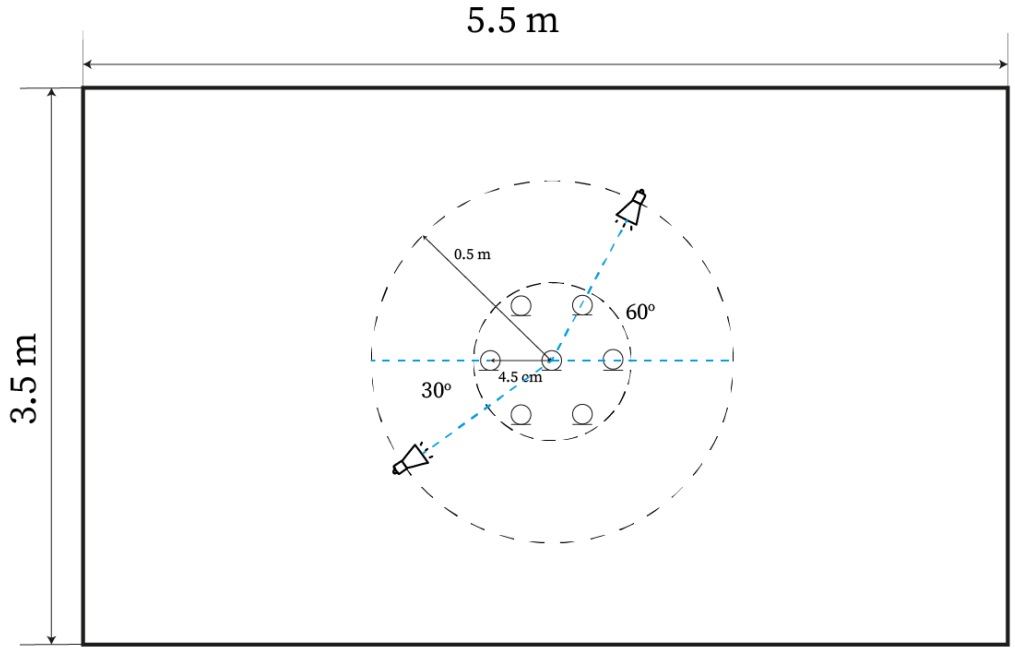

Figure 5.1: Case of 2 sources: Room environment showing the locations of sources and microphones.

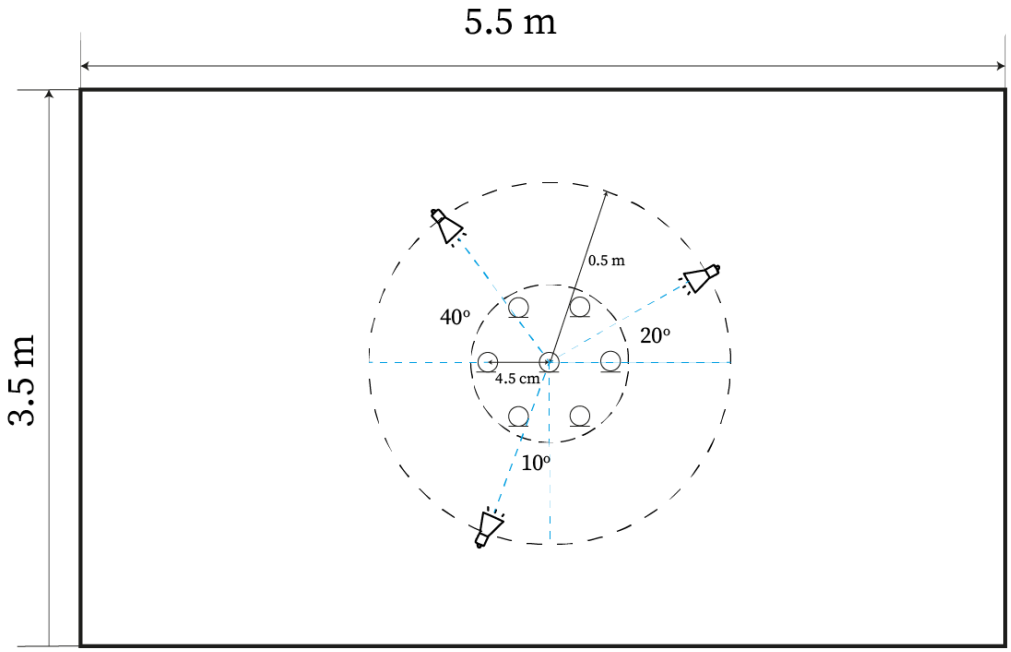

Figure 5.2: Case of 3 sources: Room environment showing the locations of sources and microphones.

5.2 Simulation Results

5.2.1 Comparison of the Algorithms' Performances

In the first experiment, the three previously described algorithms (IVA, Fast IVA, and ILRMA) are evaluated. The table 5.2 describes the parameters used for each BSS method.

IVA	**NFFT**	:	1024
	Learning rate	:	0.1
	Window type	:	Hanning
	Window length	:	1024
	Hop size	:	256
FAST-IVA	**NFFT**	:	1024
	Window type	:	Hanning
	Window length	:	1024
	Hop size	:	256

ILRMA	**NFFT**	:	1024
	Window type	:	Hamming
	Window length	:	1024
	Hop size	:	512
	Number of bases	:	10

Table 5.2: Algorithm parameters

The algorithm iterations are stopped based on a convergence criterion. When the difference between the value of the objective function at iterations i and $i+1$ is less than a predetermined tolerance, the algorithm stops. In our case, we have set this value to 10^{-6} for all algorithms. The separation performances of the algorithms for synthetic mixture signals of 2 source signals, expressed in SDR (dB) and SIR (dB), and the actual computational time for each method, are presented in Tables 5.3. The calculations were performed using MATLAB 2020 (64-bit) with an Intel Core i5-7200 (2.50 GHz) CPU.

		SIR (dB)		SDR (dB)		
Case	Methods	Source 1	Source 2	Source 1	Source 2	Time (s)
2 males	IVA	15.18	18.28	14.93	17.88	23.64
	Fast IVA	15.38	18.87	15.20	18.57	9.06
	ILRMA-1	15.40	18.82	15.24	18.54	87.22
	ILRMA-2	15.30	18.76	15.13	18.21	35.88
2 females	IVA	16.48	13.62	14.86	12.51	64.49
	Fast IVA	22.07	18.75	20.40	17.76	97.95
	ILRMA-1	2.17	1.38	0.60	0.62	79.05
	ILRMA-2	0.27	0.22	−0.22	− 0.73	90.49
1 male 1 female	IVA	13.40	20.76	12.84	17.17	19.68
	Fast IVA	14.13	21.08	13.64	18.21	5.71
	ILRMA-1	21.45	29.25	20.43	23.87	111.36
	ILRMA-2	25.57	24.78	23.44	22.26	72.92

Table 5.3: Case 2 sources: The algorithms' performances

Figures: 5.3 and 5.4 illustrate the performances of the algorithms in the case of two-sources separation.

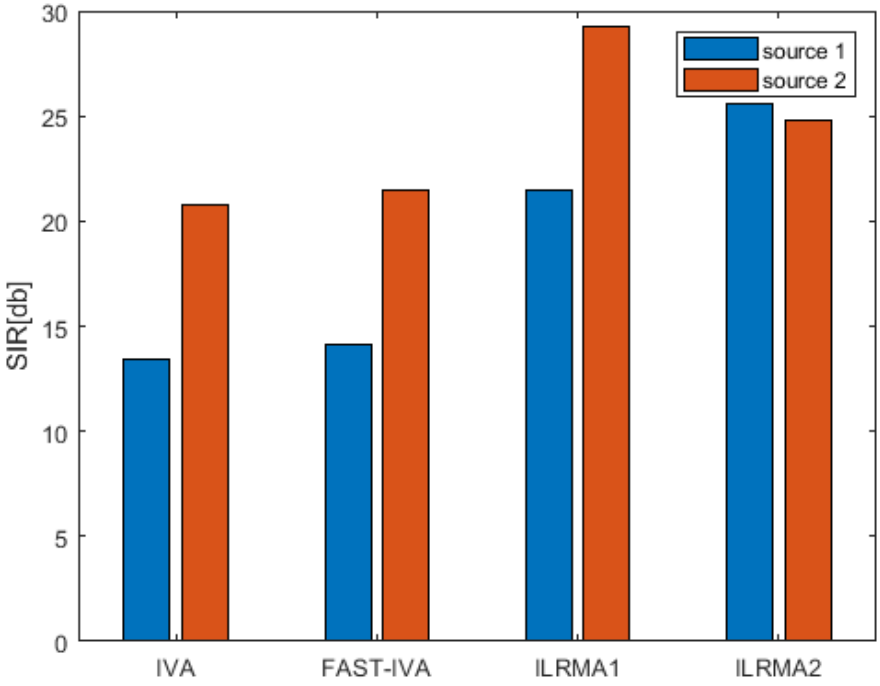

Figure 5.3: SIR (dB) in the case of two-source separation (1 male - 1 female).

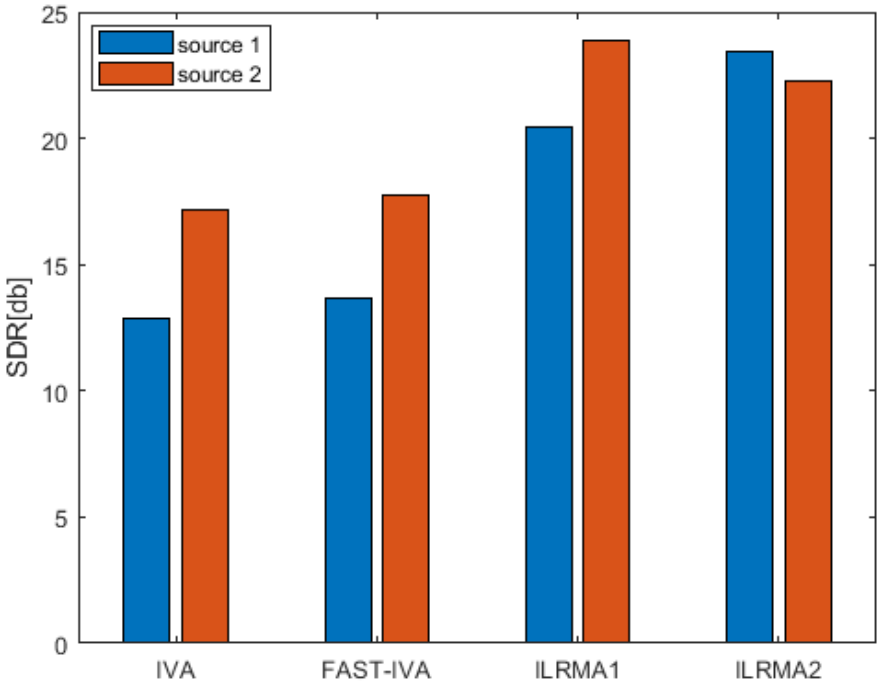

Figure 5.4: SDR (dB) in the case of two-source separation (1 male - 1 female).

The performance criteria evaluated in the scenario of separating three various mixtures of three source signals are shown in table 5.4.

		SIR (dB)			SDR (dB)			
Case	Methods	source 1	source 2	source 3	source 1	source 2	source 3	Time (s)
3 males	IVA	7.42	7.80	15.69	6.38	6.61	15.16	28.06
	Fast IVA	13.74	13.40	16.71	12.11	11.79	15.97	3.66
	ILRMA-1	10.30	9.37	17.68	7.88	8.14	17.33	123.42
	ILRMA-2	6.61	8.51	9.73	4.88	6.90	8.82	54.12
2 males 1 female	IVA	19.46	15.15	18.24	17.59	14.35	16.93	24.09
	Fast IVA	21.34	16.31	19.15	18.58	15.29	17.57	4.56
	ILRMA-1	26.27	20.48	25.22	24.14	19.58	23.62	137.12
	ILRMA-2	21.18	15.63	25.79	17.76	14.51	23.56	40.04
2 females 1 male	IVA	14.05	13.64	10.16	11.76	12.66	9.38	62.45
	Fast IVA	17.19	13.74	11.9	15.23	12.17	10.35	12.60
	ILRMA-1	19.14	6.34	5.82	16.03	4.14	4.27	142.11
	ILRMA-2	17.31	10.47	9.08	13.64	7.71	7.11	41.69

Table 5.4: Case 3 sources: The algorithms' performances.

Figures 5.5 and 5.6 illustrate the performances of the algorithms in the case of three-sources separation.

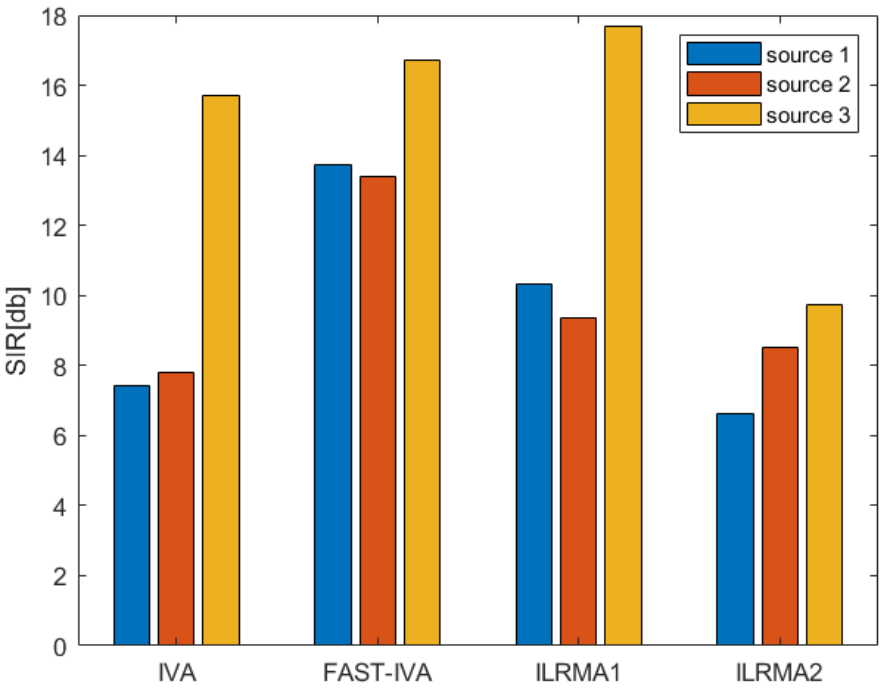

Figure 5.5: SIR (dB) in the case of separation of 3 source mixtures (3 males).

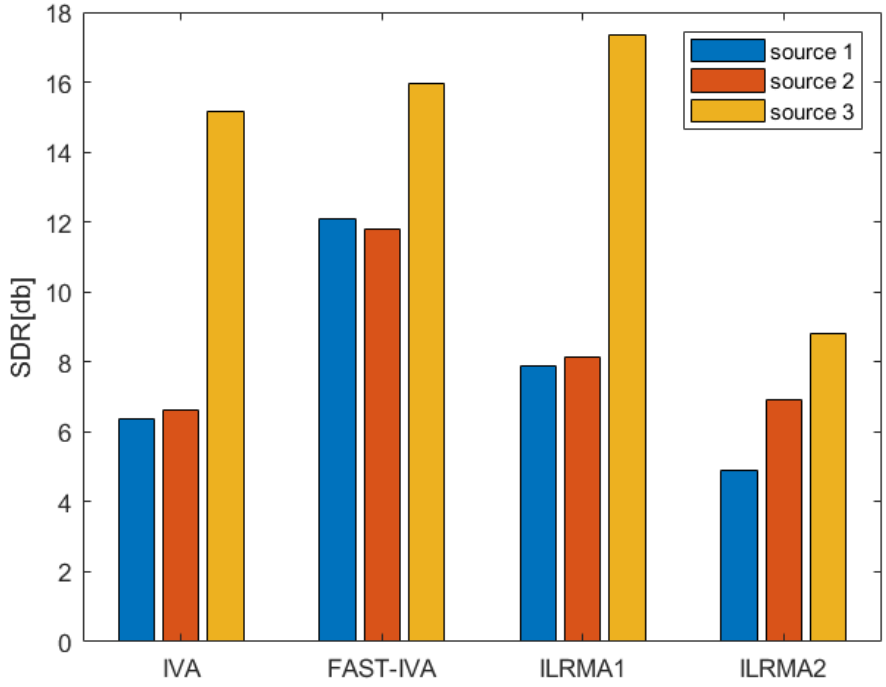

Figure 5.6: SDR (dB) in the case of separation of 3 source mixtures (3 males).

- The first remark to be drawn from these results is related to the execution time of the Fast-IVA. In both the two-sources and the three-sources separation scenarios, this technique can separate the mixtures in a very short amount of time.

- The ILRMA algorithm, with its two variations, ILRMA1 and ILRMA2, often takes the longest time to execute since it requires an additional step at each iteration, which is to estimate the variance of the speech signals.

- Both ILRMA algorithms appear to be unstable. In some cases, they fail to separate even one source from the mixture. That is because the ILRMA algorithm needs to capture the source model correctly in order to separate the sources. When it accurately models the speech signals, it outperforms the other algorithms, but this is not always the case. Indeed, due to the pitch's variation over time, it is sometimes difficult to capture speech spectrograms using NMF decomposition.

- ILRMA without partitioning (type 1) produces better simulation results than ILRMA type 2. Indeed, the partitioning function leads to instability in the speech separation. This might be a result of the sensitivity of the performance to the number of bases.

- While comparing IVA and Fast-IVA, we find that Fast IVA outperforms IVA, in most cases, in terms of separation performance and execution time.

- According to the obtained results, the main drawback of Fast IVA is that it takes longer to perform the separation when there is an all-female mix than in the other cases. This fact should be investigated in further studies.

5.2.2 Effect of Room Reverberation

In this second experiment, we evaluate the robustness of the Fast IVA algorithm regarding the reverberation time. To do so, the reverberation time (RT60) is varied when generating the RIRs. To construct the following graphs, the SDRs and SIRs were calculated for different RT60s ranging from 150 ms to 500 ms. Figures 5.7, 5.8, 5.9, and 5.10 depict the evolution of the BSS performances as the RT60 increases in the case of two-source separation and three-source mixtures.

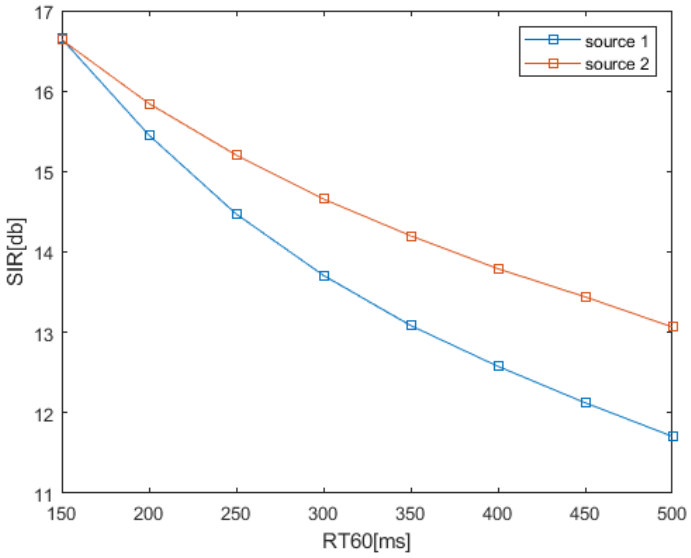

Figure 5.7: Case of 2 sources: Effect of reverberation on the SIR of the separated signals

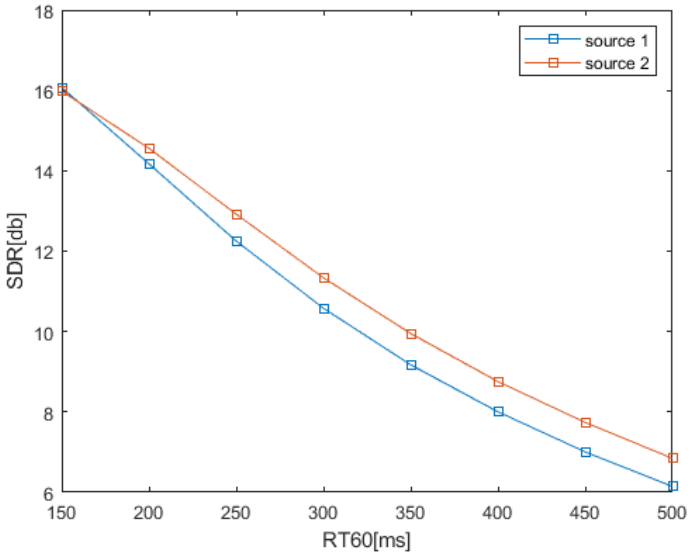

Figure 5.8: Case of 2 sources: Effect of reverberation on the SDR of the separated signals

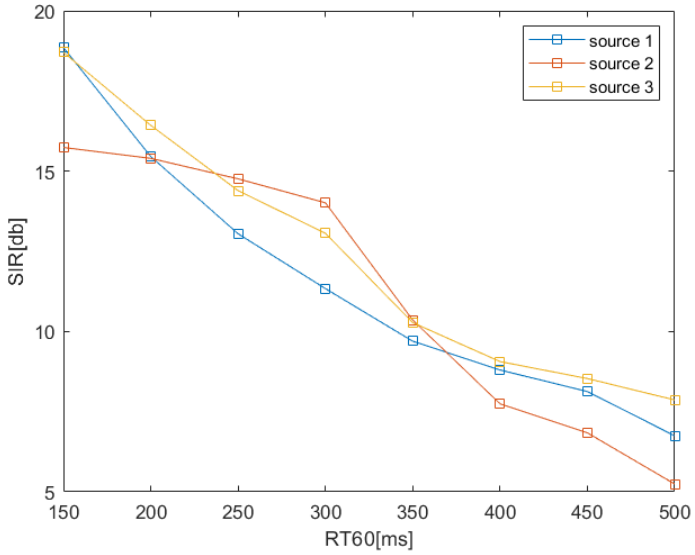

Figure 5.9: Case of 3 sources: Effect of reverberation on the SIR of the separated signals

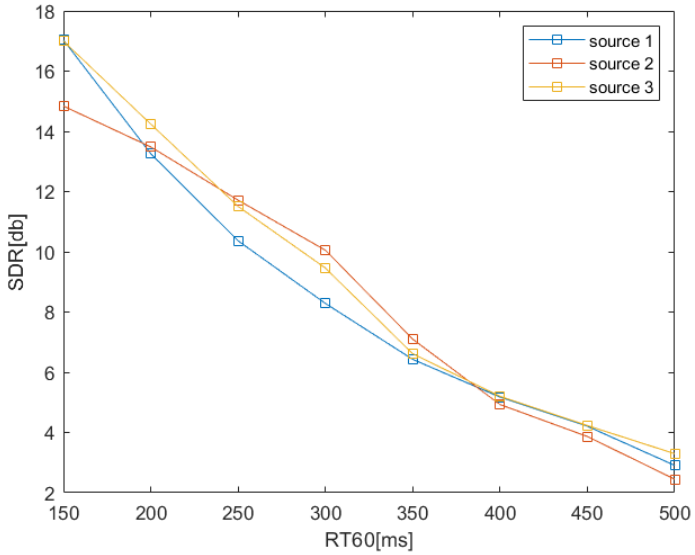

Figure 5.10: Case of 3 sources: Effect of reverberation on the SDR of the separated signals

The graphs above show that the performances gradually degrade as the RT60 increases, which is not surprising given the rise in sound reflections associated with higher room reverberations.

5.2.3 Evaluation of the Denoising Algorithm

In this third evaluation, we aim first to assess the impact of noise on the Fast-IVA algorithm. For this purpose, the speech mixture signals were degraded by computer-

generated white noise with SNR values ranging from -20 to 30 dB. To do this, we plot the SDR of the output signals using the Monte-Carlo Method $M = 20$, where M represents the number of Monte-Carlo realizations.

Figure 5.11 shows that the algorithm's performance increases when the noise level decreases.

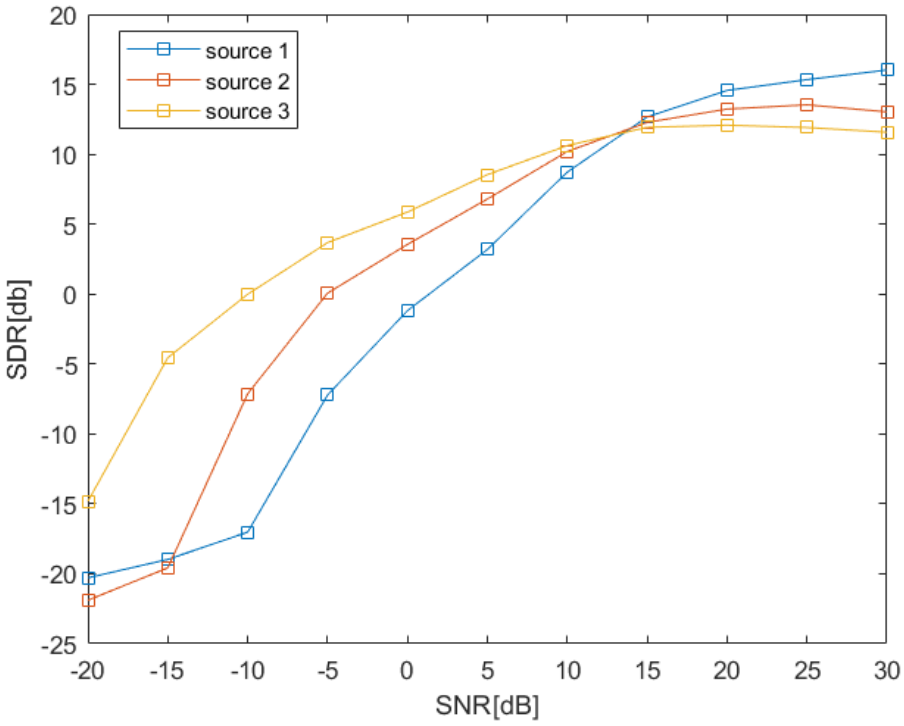

Figure 5.11: Effect of SNR on the SDR of the separated signals (case of 3 males) using Monte-Carlo runs.

Next, we compare the outcomes of the separation without denoising to the ones of two distinct combinations of the BSS model and the log-MMSE denoising technique to assess the impact of adding this denoising module in the separation algorithm and then select the appropriate denoising scheme.

In the first scheme, the convolutive mixtures are first processed by the Fast-IVA method for estimating the speech sources. The noisy separated speech signals are next processed by the denoising module, where noisy components are eliminated to enhance the quality of the estimated signals.

In the second model, the received observed speech mixtures are denoised using log-MMSE in the first step. Then, in the second step, the enhanced convolutive speech mixtures are processed by the Fast-IVA algorithm. That is to get the noise-free estimated speech signals from the enhanced mixtures.

Figure 5.12 shows a typical example of this performance evaluation in the case of three-source separation, where the SNR of the input mixtures is fixed at 10 dB.

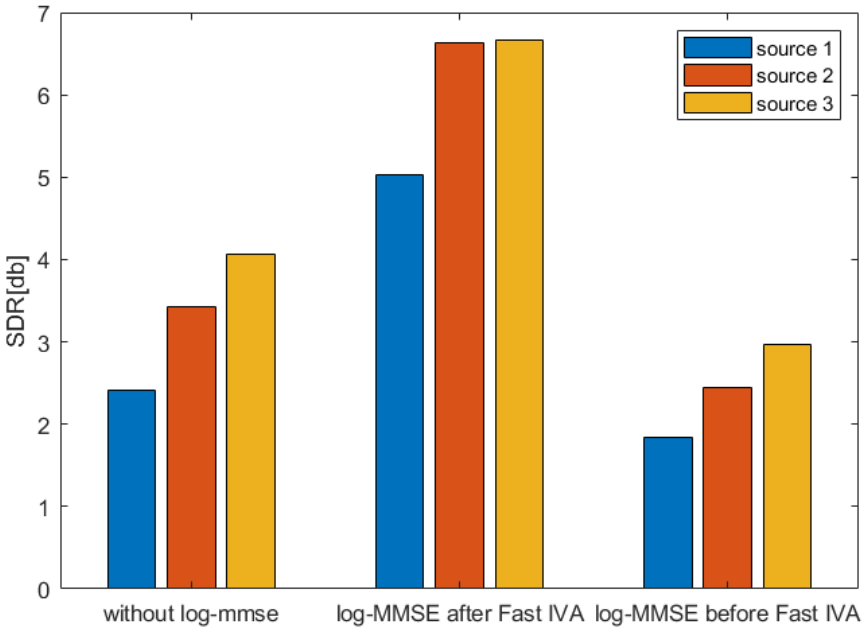

Figure 5.12: Effect of log-MMSE bloc on the SDR of the separated signals (case of 3 males)

From the results of the two proposed models, it appears that the application of the first model, which corresponds to the application of log-MMSE after the BSS algorithm, outperforms the second model and significantly improves the results obtained in the absence of the denoising algorithm. In that case, the speech sources are estimated first from the noisy mixture. Then, the noise is removed individually from the output signals, which leads to better performance.

5.2.4 Evaluation of the SIMO Equalization

Herein, the effect of applying the SIMO equalization to the separated speech signals is evaluated. In this proposed separation scheme, we use the back-projection technique after applying the Fast IVA to the mixture signals, which gives us the inputs of a SIMO deconvolution system. Those signals are used to improve the separation algorithm performances.

Figures 5.13 and 5.14 illustrate the SIRs and SDRs improvements obtained on the two-sources mixture data. Table 5.6 lists the performances and the increase in the execution time resulting from this suggested method.

step-size ρ	0.01
exponential forgetting-factor λ	0.93
Channel length L	256
length of the equalization filters Li	254

Table 5.5: SIMO parameters

		SIR (dB)		SDR (dB)		
Case	Methods	source 1	source 2	source 1	source 2	Time (s)
2 males	MDP	16.55	18.97	16.11	18.13	10.12
	SIMO equalization	23.13	23.34	22.15	21.46	13.3
	difference	**6.58**	**4.37**	**6.04**	**3.33**	3.18
2 females	MDP	21.31	21.46	9.57	19.46	61.18
	SIMO equalization	28.32	24.91	9.71	21.64	64.73
	difference	**7.01**	**3.45**	**0.14**	**2.18**	3.55
1 male 1 female	MDP	17.74	12.94	17.18	12.59	7.72
	SIMO equalization	18.52	19.58	17.80	18.28	10.5
	difference	**0.78**	**6.64**	**0.62**	**5.69**	2.78

Table 5.6: Case of two sources: SIMO equalization's outcomes

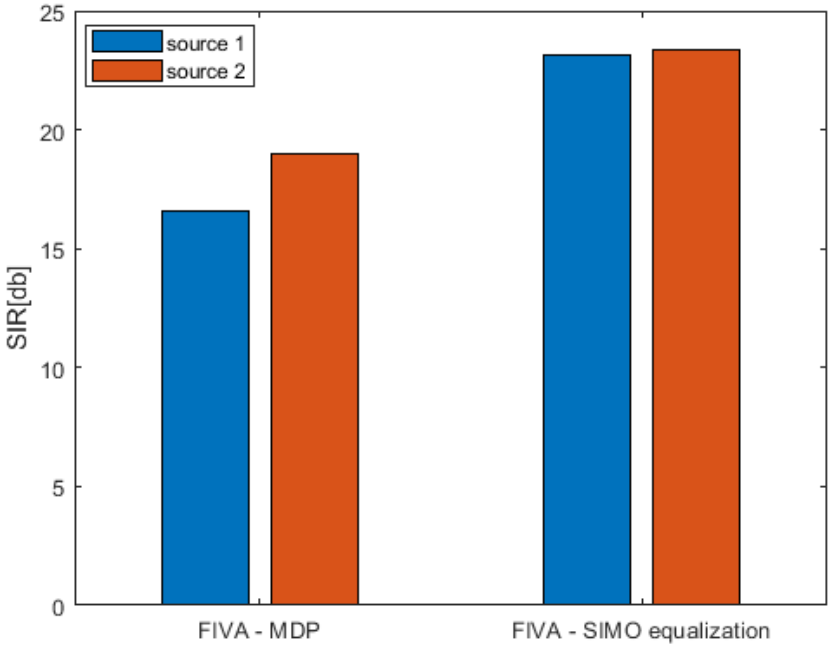

Figure 5.13: Case of 2 sources (2 males) : increase in SIR (dB)

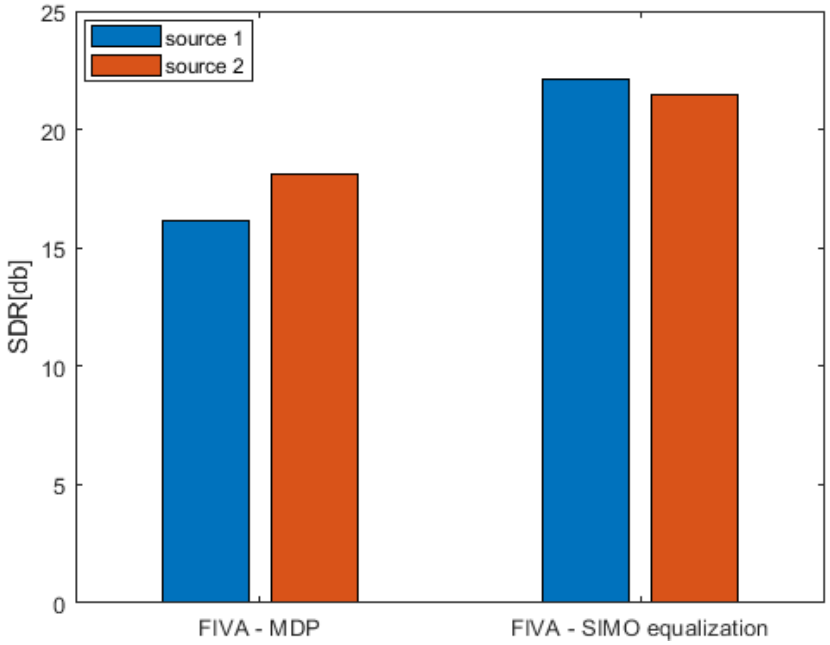

Figure 5.14: Case of 2 sources (2 males): increase in SDR (dB)

The performance improvements evaluated in the scenario of separating mixtures of three source signals are presented in Table 5.7 and Figures 5.15 and 5.16

		SIR (dB)			SDR (dB)			
Case	Methods	source 1	source 2	source 3	source 1	source 2	source 3	Time (s)
3 males	MDP	18.30	17.80	18.71	16.90	13.97	18.10	18.10
	SIMO Equalization	18.53	18.87	22.64	17.05	14.59	21.36	26.09
	Difference	**0.23**	**1.07**	**3.93**	**0.15**	**0.62**	**3.26**	7.99
2 males 1 female	MDP	11.84	15.22	15.14	11.58	14.02	14.26	88.83
	SIMO Equalization	17.60	18.32	16.80	16.81	16.22	15.67	93.95
	Difference	**5.76**	**3.1**	**1.66**	**5.23**	**2.2**	**1.41**	5.12
2 females 1 male	MDP	10.11	11.38	11.73	9.49	10.55	11.34	5.95
	SIMO Equalization	10.69	12.29	18.69	9.99	11.32	17.08	10.72
	Difference	**0.58**	**0.91**	**6.96**	**0.5**	**0.77**	**5.74**	4.77

Table 5.7: Case of three sources: SIMO equalization's outcomes

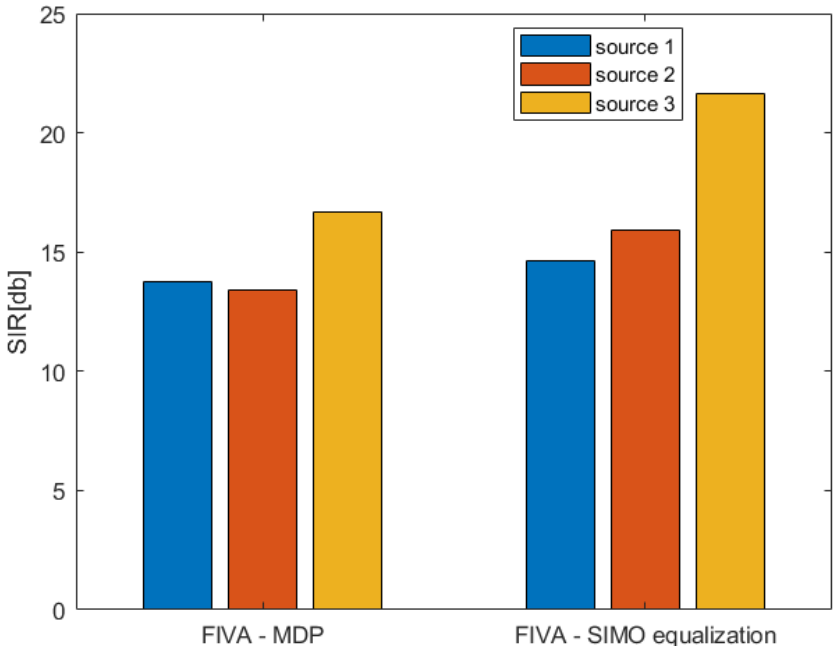

Figure 5.15: Case of 3 sources : increase in SIR (dB) (case of 3 males)

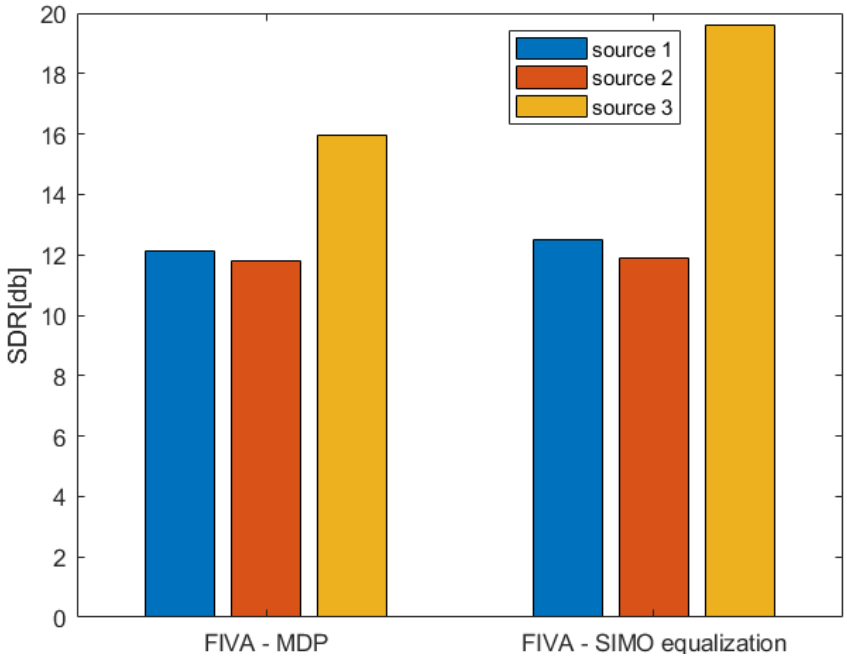

Figure 5.16: Case of 3 sources : increase in SDR (dB)(case of 3 males)

The results show that the proposed post-processing technique significantly improves the separation performance up to 7 dB without much influence on the execution time for all tested mixtures, both in the case of the two sources and three sources.

5.3 Conclusion

In this chapter, the IVA, Fast-IVA, and ILRMA algorithms were evaluated. The results show that the Fast-IVA outperforms the other two methods, especially in terms of execution time. Furthermore, it was concluded that the log-MMSE denoising module enhances the separation performance when placed after the separation algorithm in a noisy environment. Finally, we assessed the effectiveness of the SIMO equalization-based post-processing proposal. The latter technique significantly improves separation performance.

The results obtained in this chapter have allowed us to select a good algorithm for speech signals separation: the Fast fixed-point Independent Vector Analysis (Fast IVA). In the following, our goal is to implement this chosen algorithm on a Raspberry pi 4 to be able to separate real-world signals captured by the UMA-8 microphone array. The details and results of this implementation will be presented in the next chapter.

Chapter 6

Real-world Tests and Hardware Implementation of the Fast Fixed-Point IVA Algorithm

After studying a set of speech separation algorithms, we choose to use Fast IVA for the rest of our work because of its high performance and computational speed. Furthermore, from the previous simulations, we concluded that the BSS performs much better when log-MMSE and SIMO equalization are applied to the separated speech signals.

In this chapter, we will present the results of applying the Fast IVA algorithm with the suggested post-processing to real-world recorded signals using the UMA-8-SP microphone array. Then, this algorithm is implemented using Raspberry Pi 4. This implementation aims to leverage BSS as pre-processing in systems such as noise-robust speech recognition, crosstalk separation in telecommunications, and high-quality hearing aids equipment.

6.1 Hardware Devices

6.1.1 UMA 8 Microphone Array

Speech separation requires a microphone array to record multiple signals for a given mixture. In our work, we used the UMA-8, a high-performance, low-cost multi-channel USB microphone array. The latter has a plug-and-play USB audio connectivity; the microphone starts working once it is connected to a computer or a Raspberry Pi.

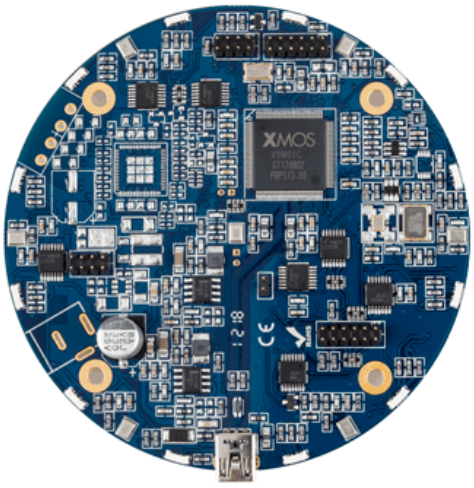

Figure 6.1: MiniDSP UMA-8 USB microphone array

The UMA-8 contains seven high-performance MEMS microphones configured in a circular pattern. It has two distinct operation modes: the DSP mode and the RAW mode. When the UMA-8 is in the raw mode, each MEMS microphone's audio signal is recorded as a distinct channel. Therefore, the recording, in this case, consists of 8 channels, since there is an additional channel, which is the output of a spare PDM port that has no transducer attached to the UMA-8 board. Multiple sample rates in the range of 11.2 - 16 -32 -44.1- 48kHz are available in this first operating mode. In the DSP mode, the UMA-8 converts the seven MEMS into a mono signal by performing tasks like beamforming, noise reduction, and acoustic echo cancelation using the XMOS Vocal Fusion DSP processing library. The sampling rate, in this case, is set to 48 kHz.

We tested the microphone's sensitivity, and the results allowed us to conclude that, in an indoor environment, the sensor network can record sounds up to a few tens of meters away. While in an outdoor environment, it can only record sounds less than 10 meters away.

6.1.2 Raspberry Pi

Figure 6.2: Raspberry Pi

The Raspberry Pi is neither a microcontroller nor a microprocessor but a single-board computer that, connected to a mouse, keyboard, and screen, works like any computer. But compared to a laptop or desktop computer, the Raspberry Pi is slower. However, it is still a computer capable of providing all the expected functionalities with low power consumption. In our work, for the

Fast-IVA implementation, we choose the
Raspberry Pi 4 board. This choice is due to its compatibility with the UMA-8 sensor array, its computational power, the ease of computation it allows, and its reasonable cost.

Raspberry Pi 4 is the latest product in the popular Raspberry Pi boards, which was introduced in June 2019. It offers revolutionary increases in processor speed, multimedia performance, memory, and connectivity over the previous generation while maintaining similar power consumption.

The product's key technical specifications can be find below:

Processor	Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
Memory	2GB
Input power	5V DC via USB-C connector (minimum 3A1) 5V DC via GPIO header (minimum 3A1) Power over Ethernet (PoE)–enabled (requires separate PoE HAT)
Connectivity	2.4 GHz and 5.0 GHz IEEE 802.11b/g/n/ac wireless LAN, Bluetooth 5.0, BLE Gigabit Ethernet 2 × USB 3.0 ports 2 × USB 2.0 ports.

Table 6.1: Key technical features of the Raspberry Pi 4

6.2 Real-world Tests

6.2.1 Experimental Setup

The experiment was conducted in a room with dimensions of 5.5 m x 4 m x 3 m, where the UMA-8 array microphone was placed on a table in the middle of the room. About 50 cm from the sensor array, two or three people were sitting and talking simultaneously. The UMA-8, connected to a computer, records the data at a sampling rate of 16 kHz for 10 seconds. All real-world signals used in our work were recorded using the RAW mode of the UMA-8 with only the 7 MEMS microphones without considering the eight channel. The separation is then performed using Fast IVA with MATLAB, with the same parameters as in the previous chapter.

6.2.2 Experimental Results

Hereafter, three cases have been considered. The first is a dialogue between two women, the second is a simultaneous discussion of two women, while the third is a mixture of three sources, two of which are men, and the third is a woman.

Case 1: Dialogue between two females

Figure 6.3 shows one of the signals recorded at the UMA-8 microphones and the two separate speech signals using: standard Fast IVA and Fast IVA with SIMO equalization and denoising. In this case, the standard Fast IVA algorithm took 58.96 seconds to run, while the SIMO equalization and log-MMSE denoising algorithm took 3.07 seconds and 0.89 seconds, respectively.

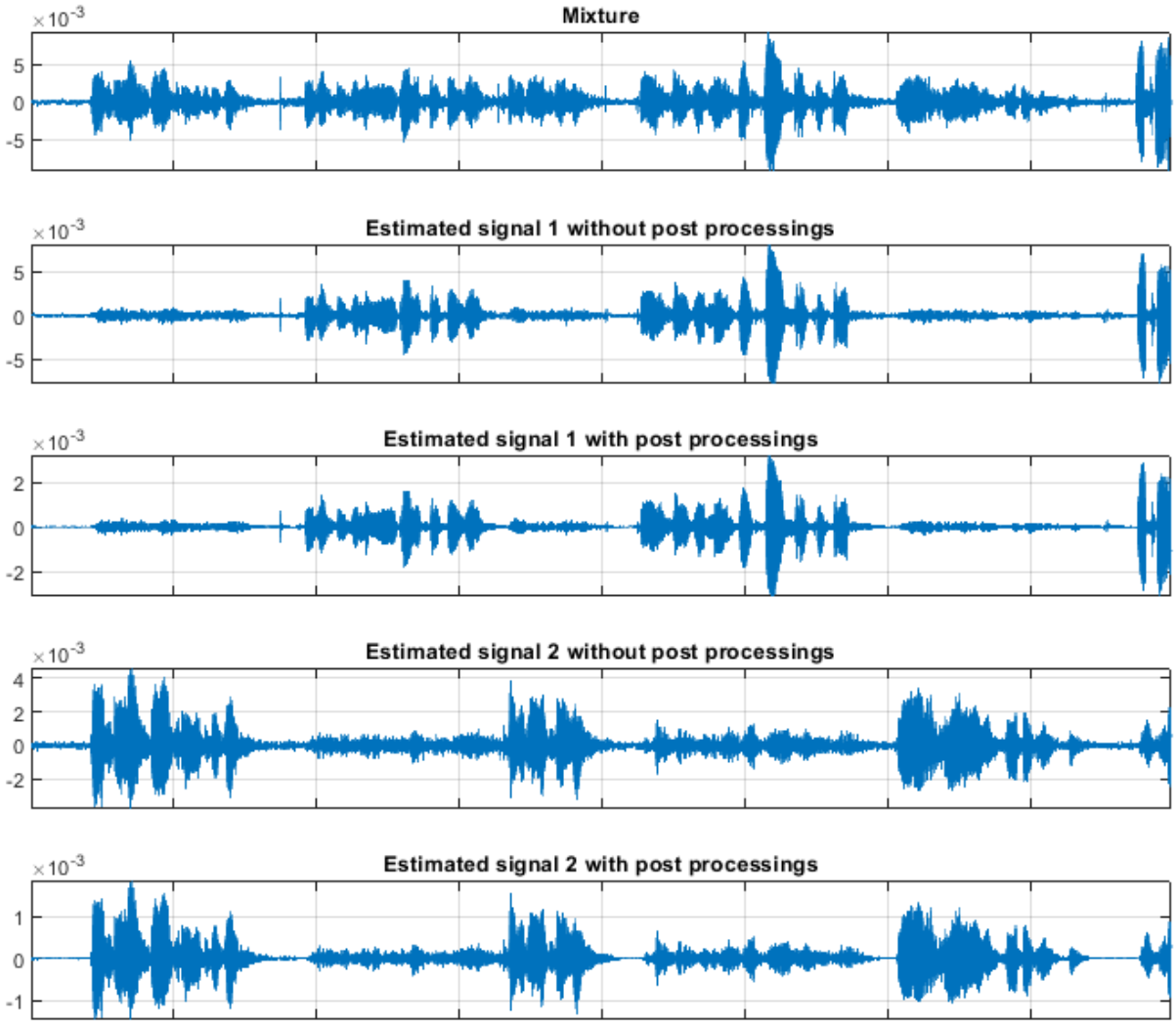

Figure 6.3: Experiments with real-world acoustic recordings: the mixture recorded by the UMA-8 microphone, the separation results of Fast IVA and the improved Fast-IVA algorithm.

From a subjective point of view, the algorithm can be considered to achieve a blind separation of the source speech signals by listening to the outputs. Furthermore, from the previous results, the following can be mentioned:

- The separation is clearly visible when comparing the two separated outputs to the corresponding mixed voice (figure 6.3). Since it is a dialogue, we can also tell when a person is silent and when they speak by separating the signals from the background noise.
- The voice is much more intelligible when listening to these two estimated sources.
- The results of the log-MMSE clearly appear when we visualize the signals; we can see, for example, that noise is eliminated from very low amplitude silent moments.

Case 2: two female mixture

Figure 6.4 shows one of the recorded signals in the microphones and the two separated speech signals using the proposed separation approach. The computations took 15.24 seconds.

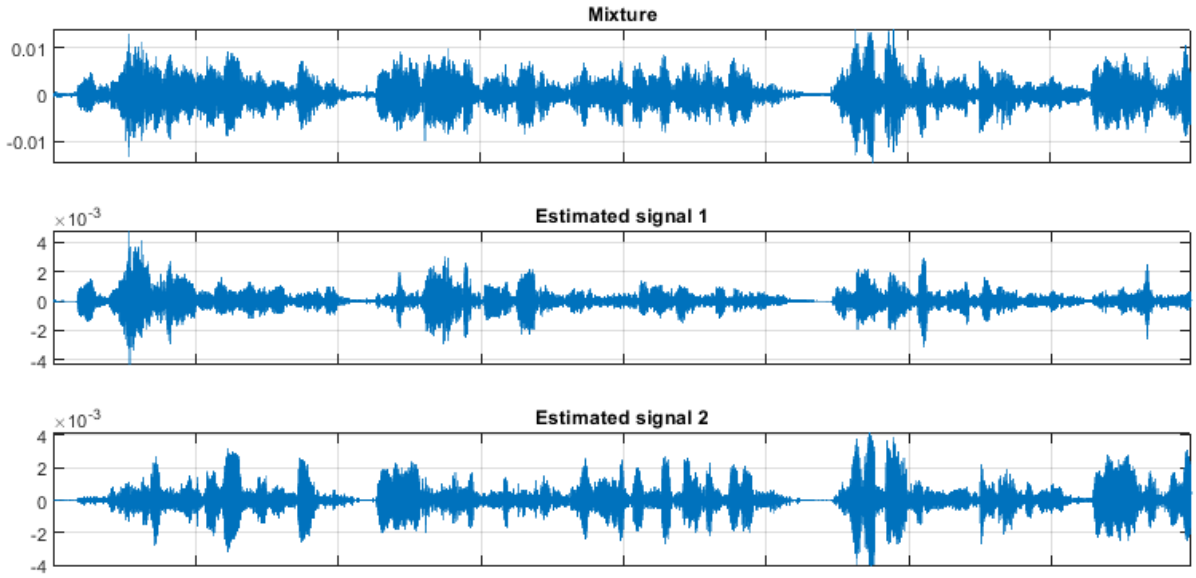

Figure 6.4: Mixture and separation signals in the case of two females source separation.

From the above figure and from listening to the two Fast IVA outputs, it can be seen that the improved Fast IVA separate well the speech signals.

Case 3: three sources - 2 males and 1 female

The results below was obtained using the improved Fast IVA algorithm, which took 14.87 seconds to run.

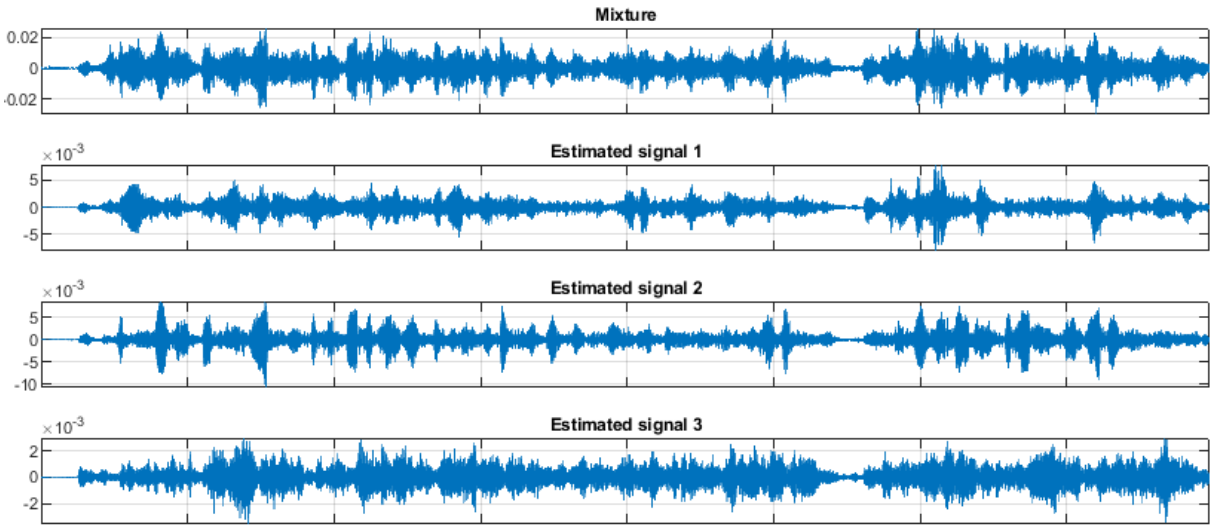

Figure 6.5: Mixture and separation signals in the case of 3 sources separation (2 males - 1 female).

In the case of three-source separation, we can see that the first two sources are effectively separated while the third is less well separated. Even though more than 2 sources are being separated, the execution time is still good.

6.3 Hardware Implementation

6.3.1 Experimental Setup

The Fast-IVA separation algorithm with the denoising module and without SIMO equalization was rewritten with the python language and then implemented on a Raspberry Pi 4 board. The UMA-8 microphone array was connected to the Raspberry Pi to perform the recording. Two speakers were also connected to the board to listen to the signals after separation. The same experimental setup mentioned above for the real-world separation test will be applied here. By running the separation algorithm on the Raspberry Pi, the following results were obtained.

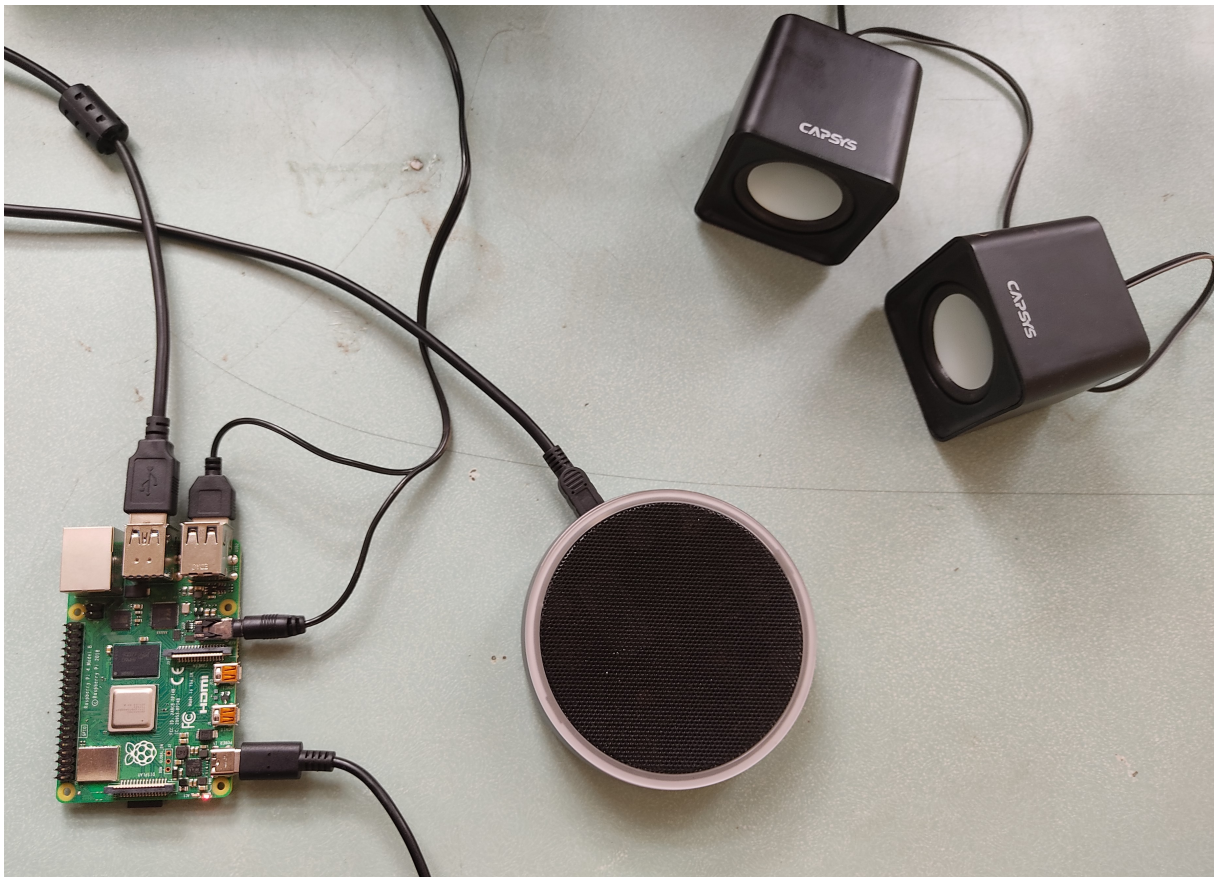

Figure 6.6: Experimental setup.

6.3.2 Experimental Results

Three cases have been considered in this part. The sentences in this section remain the same and were pronounced by the same speakers as in the previous section.

Case 1: Dialogue between two women

For this first experiment, two women were positioned on either side of the AMU8 and conducted a dialogue.

The figure 6.7 shows one mixture recorded by the UMA-8 microphones array and the two separated speech signals using: standard Fast IVA and the new improved. The standard Fast IVA algorithm took 70 seconds to run.

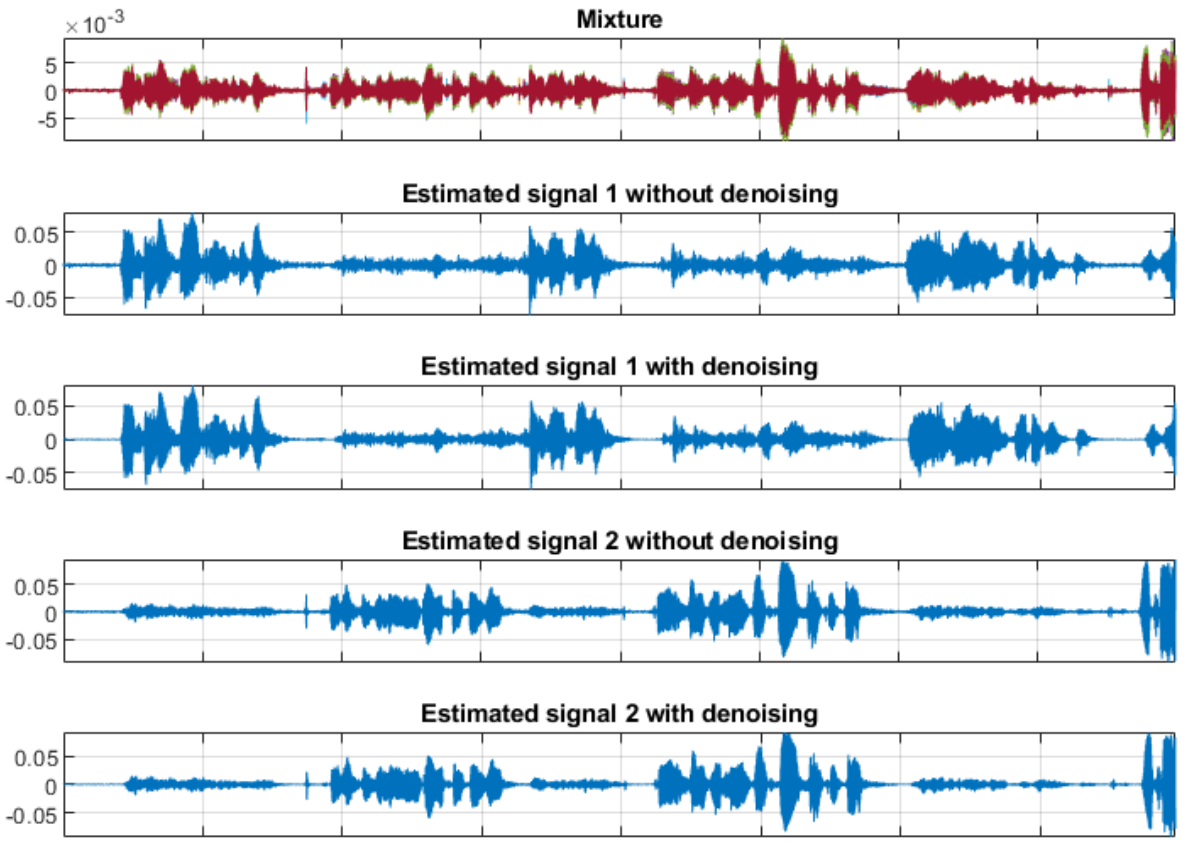

Figure 6.7: Mixture and separation signals in the case of a dialogue between two females.

The figure 6.7 shows that the system manages to separate the two sources. Indeed, by comparing the mixture with the two estimated sources, we can clearly see that the sources have been separated and that it is a dialogue.
By listening to the estimated sources, we find that the separation is well done, and the voices are much more intelligible.
Now, comparing the estimated signals with and without denoising, we can see the effect of the Log-MMSE as it has removed some noise.

Case 2: two females mixture

In this part, unlike in the previous case, the two women spoke simultaneously. Figure 6.8 shows one of the recorded signals in the microphones UMA-8 array and the two separated speech signals using the proposed separation approach. The algorithm took 60 seconds to run.

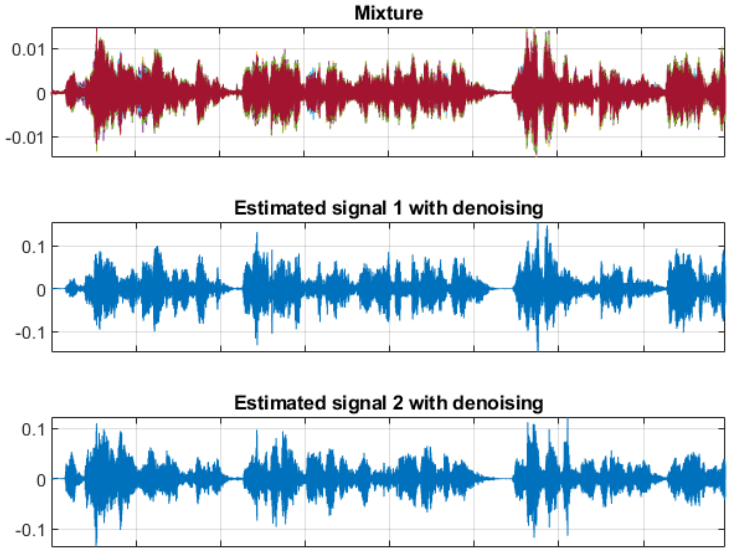

Figure 6.8: Mixture and separation signals in the case of two females source separation.

It can be seen from, this figure that the mixing was separated, and this was confirmed by listening to the two estimated signals. They were well separated.

Case 3: Three sources mixture

Finally, the separation system was tested on three sources, two males and one female. The following figure shows the graph of the mixture and the three estimated sources. The computations of Fast-IVA algorithm took 121 seconds.

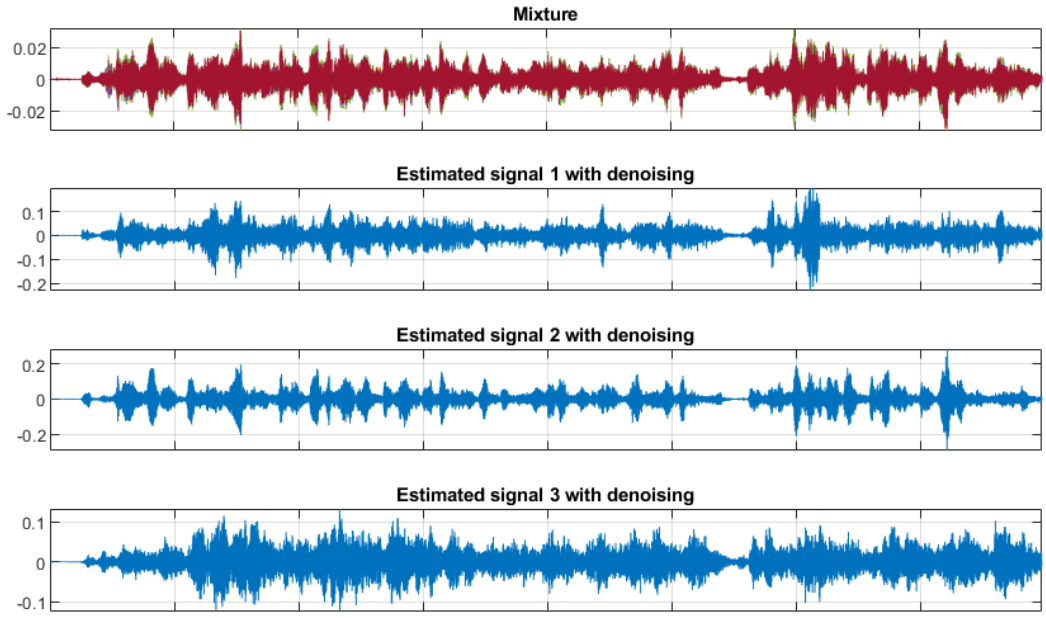

Figure 6.9: Mixture and separation signals in the case of 3 sources separation(2 males - 1 female).

Through these graphs, it can be seen that the first two estimated sources have been separated while the last one was not well separated. From listening to these signals, our observation from these graphs is correct. The latest estimated source has a lot of background noise.

6.4 Conclusion

In this chapter, we first performed some Fast-IVA tests on signals recorded by the UMA-8. The results showed that the separation algorithm works well even in these real cases.

Next, the algorithm was implemented on a Raspberry board, and further tests were performed. Again, the results show that the separation algorithm works well on a Raspberry board. The only drawback is that, in this case, the algorithm execution on the board takes a bit longer than the case on a laptop.

This separation algorithm's implementation allows us to have an independent system that can serve as a pre-processing for several applications.

Conclusion

In this work, we have addressed the problem of blind speech separation. Several methods were first studied. Subsequently, the Fast Fixed-Point Independent-Vector analysis method was selected for a potential implementation on the Raspberry Pi for its high performance and computational speed.

A modified version of this algorithm was proposed by adding two post-processings to improve the quality of the separation of multiple speech sources from their noisy reverberant mixtures. It consists of a multi-stage algorithm that operates as follows. First, the convolutive mixtures are subjected to the Fast-IVA algorithm to estimate the speech sources. Then, a SIMO system is obtained using the back-projection rescaling method, for which a deconvolution algorithm is applied. This extra step allows for the exploitation of spatial diversity and subsequently improves the separation performance. Finally, log-MMSE filtering is applied to reduce the noise from the estimated output signals.

The experimental results have demonstrated that the proposed algorithm offers significantly higher separation performance without substantially increasing computation time.

In addition, we have completed a hardware implementation of the Fast-IVA algorithm, which demonstrated the feasibility of separating speech sources in real-world environments and provided a ready-to-use and functional system that can be employed in multiple applications, including hearing aids and speech recognition.

Future work

The results presented in this dissertation demonstrated encouraging results. But still, we have noted that there is still a long way to go before a complete solution for speech source separation. The goal of the research community interested in the cocktail party problem was always to find a more "elegant" solution, potentially mimicking the deep-rooted biological and sensory mechanisms that a human being can use.

- Separation of the mixture in the under-determined case is one of the desired goals in the same way that a human being has only two ears but can separate more than two speakers.

- Another perspective is to be able to provide potential solutions for the mobile source case. Humans can use prior knowledge about the target speaker, such as familiarity with the speaker's voice.

- Another step that can be added to our proposed algorithm is the cross-talk suppression, which will allow us to remove the speech source residuals that remained in the background.

- In many speech processing applications, deep neural networks have demonstrated outstanding performances but at the expense of interpretability. Therefore, it would also be interesting to investigate the use of deep learning methods in source separation and why not combine deep learning and classical signal processing techniques to take advantage of both.

- It would also be useful to add SIMO deconvolution in the hardware implementation to get better results.

Bibliography

T. Adali, H. Li, M. Novey, and J.-F. Cardoso. Complex ica using nonlinear functions. *IEEE Transactions on Signal Processing*, 56(9):4536–4544, 2008.

Z. Albataineh and F. M. Salem. A robustica-based algorithmic system for blind separation of convolutive mixtures. *International Journal of Speech Technology*, 24(3):701–713, 2021.

J. B. Allen and D. A. Berkley. Image method for efficiently simulating small-room acoustics. *The Journal of the Acoustical Society of America*, 65(4):943–950, 1979.

S. Amari, A. Cichocki, and H. H. Yang. A new learning algorithm for blind signal separation. In *NIPS*, 1995a.

S.-i. Amari. Natural Gradient Works Efficiently in Learning. *Neural Computation*, 10 (2):251–276, 02 1998. ISSN 0899-7667. doi: 10.1162/089976698300017746. URL https://doi.org/10.1162/089976698300017746.

S.-i. Amari, A. Cichocki, and H. Yang. A new learning algorithm for blind signal separation. *Advances in neural information processing systems*, 8, 1995b.

S. Araki, F. Nesta, E. Vincent, Z. Koldovský, G. Nolte, A. Ziehe, and A. Benichoux. The 2011 signal separation evaluation campaign (sisec2011): - audio source separation - . In F. Theis, A. Cichocki, A. Yeredor, and M. Zibulevsky, editors, *Latent Variable Analysis and Signal Separation*, pages 414–422, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. ISBN 978-3-642-28551-6.

C. Avendano, J. Benesty, and D. R. Morgan. A least squares component normalization approach to blind channel identification. In *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No. 99CH36258)*, volume 4, pages 1797–1800. IEEE, 1999.

A. J. Bell and T. J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1129–1159, 1995a. doi: 10.1162/neco.1995.7.6.1129.

A. J. Bell and T. J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural computation*, 7(6):1129–1159, 1995b.

A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, and E. Moulines. A blind source separation technique using second-order statistics. *IEEE Transactions on signal processing*, 45(2):434–444, 1997.

H. Bousbia-Salah, A. Belouchrani, and K. Abed-Meraim. Jacobi-like algorithm for blind signal separation of convolutive mixtures. *Electronics Letters*, 37(16):1, 2001.

H. Brehm and W. Stammler. Description and generation of spherically invariant speech-model signals. *Signal Processing*, 12(2):119–141, 1987.

R. Bro and A. K. Smilde. Principal component analysis. *Analytical methods*, 6(9):2812–2831, 2014.

J.-F. Cardoso. Blind identification of independent components with higher-order statistics. In *Workshop on Higher-Order Spectral Analysis*, pages 157–162, 1989. doi: 10.1109/HOSA.1989.735288.

J.-F. Cardoso. The invariant approach to source separation. *Proceedings of the International Symposium on Nonlinear Theory and Applications NOLTA*, 1:55–60, 1995.

J.-F. Cardoso. Estimating equations for source separation. In *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages 3449–3452. IEEE, 1997.

E. C. Cherry. Some Experiments on the Recognition of Speech, with One and with Two Ears. *The Journal of the Acoustical Society of America*, 25(5):975–979, Sept. 1953. ISSN 0001-4966. doi: 10.1121/1.1907229. URL https://asa.scitation.org/doi/abs/10.1121/1.1907229. Publisher: Acoustical Society of America.

A. Cichocki and R. Unbehauen. Robust neural networks with on-line learning for blind identification and blind separation of sources. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 43(11):894–906, 1996.

P. Comon. Separation of stochastic processes. In *Workshop on Higher-Order Spectral Analysis*, pages 174–179, 1989. doi: 10.1109/HOSA.1989.735291.

P. Comon. Independent component analysis, a new concept? *Signal Processing*, 36(3):287–314, 1994. ISSN 0165-1684. doi: https://doi.org/10.1016/0165-1684(94)90029-9. URL https://www.sciencedirect.com/science/article/pii/0165168494900299. Higher Order Statistics.

P. Comon and C. Jutten. *Handbook of Blind Source Separation: Independent component analysis and applications*. Academic press, 2010.

R. Crochiere. A weighted overlap-add method of short-time fourier analysis/synthesis. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(1):99–102, 1980.

A. Daher, E. H. Baghious, G. Burel, and E. Radoi. Overlap-save and overlap-add filters: Optimal design and comparison. *IEEE Transactions on Signal Processing*, 58(6):3066–3075, 2010. doi: 10.1109/TSP.2010.2044260.

S. C. Douglas and M. Gupta. Convolutive blind source separation for audio signals. In *Blind speech separation*, pages 3–45. Springer, 2007.

C. Févotte, N. Bertin, and J.-L. Durrieu. Nonnegative matrix factorization with the itakura-saito divergence: With application to music analysis. *Neural computation*, 21 (3):793–830, 2009.

B. Gao. *Single channel blind source separation*. PhD thesis, Newcastle University, 2011.

R. Gribonval and S. Lesage. A survey of sparse component analysis for blind source separation: principles, perspectives, and new challenges. In *ESANN'06 proceedings-14th European Symposium on Artificial Neural Networks*, pages 323–330. d-side publi., 2006.

M. Haque and M. Hasan. Variable step-size multichannel frequency-domain lms algorithm for blind identification of finite impulse response systems. *IET Signal Processing*, 1(4): 182–189, 2007.

M. A. Haque and M. K. Hasan. Noise robust multichannel frequency-domain lms algorithms for blind channel identification. *IEEE Signal Processing Letters*, 15:305–308, 2008.

C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, Sept. 2020. doi: 10.1038/s41586-020-2649-2. URL https://doi.org/10.1038/s41586-020-2649-2.

J. Herault and C. Jutten. Space or time adaptive signal processing by neural network models. In *AIP Conference Proceedings 151 on Neural Networks for Computing*, page 206–211, USA, 1986. American Institute of Physics Inc. ISBN 088318351X.

A. Hiroe. Solution of permutation problem in frequency domain ica, using multivariate probability density functions. In J. Rosca, D. Erdogmus, J. C. Príncipe, and S. Haykin, editors, *Independent Component Analysis and Blind Signal Separation*, pages 601–608, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-32631-1.

Y. Huang and J. Benesty. Adaptive blind channel identification: multi-channel least mean square and newton algorithms. *Signal Processing*, 82:1127–1138, 08 2002. doi: 10.1016/S0165-1684(02)00247-5.

Y. Huang and J. Benesty. A class of frequency-domain adaptive approaches to blind multichannel identification. *IEEE Transactions on signal processing*, 51(1):11–24, 2003.

A. Hyvarinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10(3):626–634, 1999a. doi: 10.1109/72.761722.

A. Hyvarinen. Fast and robust fixed-point algorithms for independent component analysis. *IEEE transactions on Neural Networks*, 10(3):626–634, 1999b.

A. Hyvärinen and E. Oja. A fast fixed-point algorithm for independent component analysis. *Neural computation*, 9(7):1483–1492, 1997.

A. Hyvärinen and P. Pajunen. Nonlinear independent component analysis: Existence and uniqueness results. *Neural networks*, 12(3):429–439, 1999.

A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*, volume 26. 06 2001. ISBN 9780471405405. doi: 10.1002/0471221317.

S. N. Jain and C. Rai. Blind source separation and ica techniques: a review. *International Journal of Engineering Science and Technology*, 4(4):1490–1503, 2012.

T. Kim, H. T. Attias, S.-Y. Lee, and T.-W. Lee. Blind source separation exploiting higher-order frequency dependencies. *IEEE transactions on audio, speech, and language processing*, 15(1):70–79, 2006a.

T. Kim, I. Lee, and T.-W. Lee. Independent vector analysis: definition and algorithms. In *2006 Fortieth Asilomar Conference on Signals, Systems and Computers*, pages 1393–1396. IEEE, 2006b.

D. Kitamura. Algorithms for Independent Low-Rank Matrix Analysis. page 7, 2018.

D. Kitamura, N. Ono, H. Sawada, H. Kameoka, and H. Saruwatari. Determined blind source separation unifying independent vector analysis and nonnegative matrix factorization. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(9):1626–1641, 2016.

D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. *Advances in neural information processing systems*, 13, 2000.

I. Lee, T. Kim, and T.-W. Lee. Fast fixed-point independent vector analysis algorithms for convolutive blind source separation. *Signal Processing*, 87(8):1859–1871, 2007a. ISSN 0165-1684. doi: https://doi.org/10.1016/j.sigpro.2007.01.010. URL https://www.sciencedirect.com/science/article/pii/S0165168407000163. Independent Component Analysis and Blind Source Separation.

I. Lee, T. Kim, and T.-W. Lee. Fast fixed-point independent vector analysis algorithms for convolutive blind source separation. *Signal Processing*, 87(8):1859–1871, 2007b.

H. Liu, G. Xu, and L. Tong. A deterministic approach to blind equalization. In *Proceedings of 27th Asilomar Conference on Signals, Systems and Computers*, pages 751–755. IEEE, 1993.

S. Makino. *Audio source separation*, volume 433. Springer, 2018.

S. Makino, T.-W. Lee, and H. Sawada. *Blind speech separation*, volume 615. Springer, 2007.

M. Miyoshi and Y. Kaneda. Inverse filtering of room acoustics. *IEEE Transactions on acoustics, speech, and signal processing*, 36(2):145–152, 1988.

E. Moulines, P. Duhamel, J.-F. Cardoso, and S. Mayrargue. Subspace methods for the blind identification of multichannel fir filters. *IEEE Transactions on Signal Processing*, 43(2):516–525, 1995a. doi: 10.1109/78.348133.

E. Moulines, P. Duhamel, J.-F. Cardoso, and S. Mayrargue. Subspace methods for the blind identification of multichannel fir filters. *IEEE Transactions on signal processing*, 43(2):516–525, 1995b.

R. Mukai, H. Sawada, S. Araki, and S. Makino. Frequency domain blind source separation using small and large spacing sensor pairs. In *2004 IEEE International Symposium on Circuits and Systems (IEEE Cat. No. 04CH37512)*, volume 5, pages V–V. IEEE, 2004.

N. Murata, S. Ikeda, and A. Ziehe. An approach to blind source separation based on temporal structure of speech signals. *Neurocomputing*, 41(1):1–24, 2001. ISSN 0925-2312. doi: https://doi.org/10.1016/S0925-2312(00)00345-3. URL https://www.sciencedirect.com/science/article/pii/S0925231200003453.

S. T. Neely and J. B. Allen. Invertibility of a room impulse response. *Journal of the Acoustical Society of America*, 66:165–169, 1979.

E. Oja and A. Hyvarinen. Independent component analysis: algorithms and applications. *Neural networks*, 13(4-5):411–430, 2000.

N. Ono. Stable and fast update rules for independent vector analysis based on auxiliary function technique. In *2011 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 189–192, 2011. doi: 10.1109/ASPAA.2011.6082320.

T. Ono, N. Ono, and S. Sagayama. User-guided independent vector analysis with source activity tuning. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2417–2420, 2012. doi: 10.1109/ICASSP.2012.6288403.

J. A. Palmer and S. Makeig. Contrast functions for independent subspace analysis. In *International Conference on Latent Variable Analysis and Signal Separation*, pages 115–122. Springer, 2012.

K. B. Petersen and M. S. Pedersen. The matrix cookbook, nov 2012. URL http://www2.compute.dtu.dk/pubdb/pubs/3274-full.html. Version 20121115.

H. Sawada, R. Mukai, S. Araki, and S. Makino. A robust and precise method for solving the permutation problem of frequency-domain blind source separation. *IEEE transactions on speech and audio processing*, 12(5):530–538, 2004.

H. Sawada, N. Ono, H. Kameoka, D. Kitamura, and H. Saruwatari. A review of blind source separation methods: two converging routes to ilrma originating from ica and nmf. *APSIPA Transactions on Signal and Information Processing*, 8, 2019.

R. Scheibler, E. Bezzam, and I. Dokmanic. Pyroomacoustics: A python package for audio room simulations and array processing algorithms. *CoRR*, abs/1710.04196, 2017. URL http://arxiv.org/abs/1710.04196.

I. Tashev and A. Acero. Statistical modeling of the speech signal. In *International Workshop on Acoustic, Echo, and Noise Control (IWAENC)*, 2010.

L. Tong, V. Soon, Y. Huang, and R. Liu. Amuse: a new blind identification algorithm. In *IEEE International Symposium on Circuits and Systems*, pages 1784–1787 vol.3, 1990. doi: 10.1109/ISCAS.1990.111981.

E. Vincent, R. Gribonval, and C. Févotte. Performance measurement in blind audio source separation. *IEEE transactions on audio, speech, and language processing*, 14(4): 1462–1469, 2006.

E. Vincent, T. Virtanen, and S. Gannot. *Audio source separation and speech enhancement*. John Wiley & Sons, 2018.

P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

T. Virtanen. Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria. *IEEE transactions on audio, speech, and language processing*, 15(3):1066–1074, 2007.

G. Xu, H. Liu, L. Tong, and T. Kailath. A least-squares approach to blind channel identification. *IEEE Transactions on signal processing*, 43(12):2982–2993, 1995.

K. Yu, K. Yang, and Y. Bai. Estimation of modal parameters using the sparse component analysis based underdetermined blind source separation. *Mechanical Systems and Signal Processing*, 45(2):302–316, 2014a.

X. Yu, D. Hu, and J. dong Xu. *Blind Source Separation: Theory and Applications*. John Wiley & Sons, 2014b.

Annexes

Annexe A: Pyroomacoustics

To generate a Room impulse Response and the corresponding convolutional mixtures using Pyroomacoustics, we proceed as follows:

1. Specify the lengths of the walls of a 3D room in a single Vector. In the following example, we define a 5.5m x 4m x 3m room.

```
room_dim = [5.5, 4, 3]
```

2. Use Sabine's formula to determine the wall energy absorption and the maximum order of the Image Source Method (ISM) needed to produce the desired reverberation time (RT60). Note that the ISM's maximum order corresponds to the permitted maximum number of reflections.

```
rt60 = 0.5
e_absorption, max_order = pra.inverse_sabine(rt60, room_dim)
```

3. Create the room by specifying its size, the wall energy absorption, the maximum order of the Image Source Method, and the sampling frequency.

```
room = pra.ShoeBox(
    room_dim, fs=16000, materials=pra.Material(e_absorption),
   max_order=max_order
)
```

4. Create as many sources as you want, simply by specifying their positions, the audio signal that the sources will emit and the start time. Hereafter, we create a source located at [2.3, 1.6, 1.62] in the room, which will emit the content of the audio file "speech.wav" from 0.7 s.

```
from scipy.io import wavfile
_, audio = wavfile.read('speech.wav')

room.add_source([2.3, 1.6, 1.62], signal=audio, delay=0.7)
```

5. Add a microphone array in the room by defining a nd-array of size (3, M), where each column contains the coordinates of one microphone. Here, we create an array with two microphones:

```
import numpy as np
mic_locs = np.c_[
    [6.3, 4.87, 1.2],  # mic 1
    [6.3, 4.93, 1.2],  # mic 2
]

room.add_microphone_array(mic_locs)
```

   Note that in the two previously defined functions, *add_source* and *add_microphone* we can specify the directivity of these two elements. Multiple directivity patterns are available such as hypercardioid, cardioid, and subcardioid.

6. Create the room Impulse Response by using the Image Source Method. The attribute *rir* of the object room, is a list of lists, where the inner list is on sources and the outer list is on microphones, containing the coefficients of the channels.

```
room.simulate()
```

7. Write the convolutive mixtures recorded by the microphones in wav files.

```
room.mic_array.to_wav(
   "output.wav",
    norm=True,
    bitdepth=np.int16,
)
```