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

End-of-study project dissertation for obtaining the State Engineer's degree in Electronics

Blind Speech Separation: Adaptive algorithm and Implementation using UMA-16 v2 Mic Array Testbed

Realized by:
Idriss *MERAH*
Ahmed-Zakaria *GHECHAM*

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

Publicly presented and defended on June 26^{th}, 2023

Composition of the Jury:

President	Mrs. Nesrine *BOUADJENEK*	PhD.	ENP
Examiner	Mr. Mourad *ADNANE*	Prof.	ENP
Supervisor	Mr. Adel *BELOUCHRANI*	Prof.	ENP
Supervisor	Mr. Soufiane *TEBACHE*	PhD.	LDCCP/ENP

ENP 2023

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

End-of-study project dissertation for obtaining the State Engineer's degree in Electronics

Blind Speech Separation: Adaptive algorithm and Implementation using UMA-16 v2 Mic Array Testbed

Realized by:
Idriss *MERAH*
Ahmed-Zakaria *GHECHAM*

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

Publicly presented and defended on June 26^{th}, 2023

Composition of the Jury:

President	Mrs. Nesrine *BOUADJENEK*	PhD.	ENP
Examiner	Mr. Mourad *ADNANE*	Prof.	ENP
Supervisor	Mr. Adel *BELOUCHRANI*	Prof.	ENP
Supervisor	Mr. Soufiane *TEBACHE*	PhD.	LDCCP/ENP

ENP 2023

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 : Algorithme adaptatif et Implémentation à l'aide d'un Réseau de Capteurs UMA-16 v2

Réalisé par :
Idriss *MERAH*
Ahmed-Zakaria *GHECHAM*

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

Présenté et soutenu publiquement le: 26/06/2023

Composition du Jury:

Président	Mme. Nesrine *BOUADJENEK*	Docteur	ENP
Examinateur	M. Mourad *ADNANE*	Professeur	ENP
Promoteur	M. Adel *BELOUCHRANI*	Professeur	ENP
Promoteur	M. Soufiane *TEBACHE*	Docteur	LDCCP/ENP

ENP 2023

ملخص

في بيئة حيث يتحدث عدة أشخاص مسجلين في نفس الوقت، يكون من الصعب التمييز بين كل صوت. وبالتالي، فإن استخراج كل إشارة صوتية من هذا الاختلاط أمر أساسي وله عدة تطبيقات. تهدف هذه الدراسة إلى تنفيذ فصل المصادر الصوتية الأعمى بطريقة تكيفية. لقد درسنا في المقام الأول خوارزمية تحليل الاشعة المستقلة لفهم مبدأها بشكل جيد. ثم قمنا بتعديل الخوارزمية للحصول على نسختها التكيفية وأضفنا إليها تبييض البيانات التكيفي. أخيرًا، قمنا بمقارنة تأثير هذا التبييض على أداء الخوارزمية وقمنا بتنفيذ هذه الطريقة باستخدام إشارات حقيقية مسجلة باستخدام جهاز الميكروفون.

كلمات مفتاحية : إشارة صوتية، فصل المصادر الصوتية الأعمى، تحليل الاشعة المستقلة، تبييض

Résumé

Dans un environnement où plusieurs personnes enregistrées parlent en même temps, il est difficile de discerner chaque voix. De ce fait, extraire chaque signal de parole à partir de ce mélange convolutif est primordial et possède plusieurs applications. Ce travail a pour objectif de procéder à la séparation de sources aveugles d'une manière adaptative. Nous avons, en premier lieu, étudié l'algorithme Independent Vector Analysis (IVA) afin de bien assimiler son principe. Ensuite, nous avons modifié l'algorithme afin d'obtenir une version adaptative et ajouté à ce dernier un blanchiment de données adaptatif. Enfin, nous avons comparé les effets de ce blanchiment sur les performances de notre algorithme et implémenté cette méthode en utilisant des signaux réels enregistrés grâce à un dispositif de microphones.

Mots clés : IVA, séparation de sources aveugle, blanchiment.

Abstract

In an environment where multiple recorded individuals are speaking simultaneously, it is difficult to discern each voice. Therefore, extracting each speech signal from this convoluted mixture is crucial and has several applications. The objective of this work is to perform blind source separation in an adaptive manner. First, we studied the Independent Vector Analysis (IVA) algorithm to fully understand its principle. Then, we modified the algorithm to obtain its adaptive version and added adaptive data whitening to it. Finally, we compared the effects of this whitening on the performance of our algorithm and implemented this method using real signals recorded through an array of microphones.

Key words: IVA, Blind source separation, whitening.

Dedication

I dedicate this work,
To my parents, who have always encouraged me throughout my educational journey, to my two little sisters Amina and Sarah, to my grandparents, and to my entire family.
To the people who are dear to me: Okba, Raouf Boudia, Kenzi, Rahim, Yacine, Moncef, Abdelkrim, Younes, Midou, Yanis, Anis, Raouf Anseur, Katia, Samy, Walid, Thiziri, as well as all my classmates from the 'ELN MATCH' promotion with whom I have shared great moments.
To our supervisors, Professor Adel Belouchrani and doctor TEBACHE Soufiane, who have supported us throughout this project.
To all the people who have helped me in one way or another to become the person I am today.
To my co-worker, Idriss Merah, without whom I would not have been able to accomplish this work.
Lastly, I would like to pay tribute to my deceased grandfather, Mostefa Benzerga, may God welcome him into his vast paradise.

Ahmed-Zakaria

Dedication

I dedicate this work,
To my late mum Macouta whom I couldn't get to know and my late grandmother Yamina who raised me since I was a kid and to whom I owe most what I have achieved in my life may God welcome them into his vast paradise and I hope they are proud of what I have become today.
To my aunt Mounira whom I love most dearly to my brother Mounir and my father Nadir, my aunt Hawa and all the members of my family.
To my dear beloved friends: Karine, Anis, Rayane Rouizi, Aya, Hayat, Riad, Samy, Raouf, Zizou, Massylia, Katia, Rayane Terkmani, Nacim, Thizri, Lyna, Lynda and Imene.
To the Irmouli's who have a special place in my heart : Mourad, Shehenaz, Lina, Wafa, Nour, Fadila and Imene.
To my beloved Vision and Innovation Club (VIC) and all its members with whom I spent the best 3 years of my life and especially to the comittee I worked with: Farouk, Imane, Nadhir, Melissa, Imed, Riad, Seifo, Nesrine and Ahmed.
To the best classmates in the world and my second family 'ELN MATCH' promotion with whom I shared my best academic years I can never be thankful enough for having you in my life.
To Dr. Sid-Ahmed Berroudji who made me discover my passion for physics and engineering which helped me to chose electronics engineering and to all the teachers who made me love what I am studying.
To my supervisors, Professor Adel Belouchrani and doctor TEBACHE Soufiane, who have supported us throughout this project.
To Zaki, 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.
To all the people who have helped me in one way or another to become the person I am today. I love you all from the bottom of my heart.

Idriss

Acknowledgement

We would like to extend our heartfelt appreciation to all those who contributed to the realization of this work, starting with our beloved parents, whose unwavering support and sacrifices have played an integral role in shaping our journey and accomplishments.

We would also like to express our sincere gratitude to our esteemed mentors, Adel BELOUCHRANI, a distinguished Professor at the Ecole Nationale Polytechnique, who serves as an exemplary researcher, advisor, and instructor, and Dr. Soufiane TEBACHE who has a brilliant future ahead of him, for their meticulous guidance and unwavering motivation. Their rigorous supervision and invaluable encouragement have been instrumental in the successful completion of this work.

We would like to express our gratitude to Mrs. Nesrine BOUADJENEK for graciously accepting the role of jury chair for our final project and Mr. Mourad ADNANE our examiner, for his keen interest in our work.

We would like to extend our appreciation to all the teachers and students of the electronics engineering department. Their unwavering support, knowledge sharing, and collaborative spirit have played a significant role in shaping our academic journey and fostering an environment of continuous learning and growth.

Contents

List of Tables

List of Figures

List of Acronyms

AMUSE	A Minimally-Unsatisfiable Subformula Extractor
ASIO	Audio Stream Input/Output
BSS	Blind Source Separation
CBSS	Convolutive Blind Source Separation
CDF	Cummulative Distrubtion Function
FDBSS	Frequency-Domain Blind Source Separation
FDICA	Frequency-Domain Independent Component Analysis
FIR	Finite Impulse Response
Fast IVA	Fixed-point/Fast Independent Vector Analysis
FFT	Fast Fourier Transform
IC	Integrated Circuit
ICA	Independent Component Analysis
ILRMA	Independent Low-Rank Matrix Analysis
ISTFT	Inverse Short-Time Fourier Transform
IVA	Independent Vector Analysis
JADE	Joint Approximate Diagonalization of Eigenmatrices
KLD	Kullback-Leiber divergence
MDP	Minimal Distortion Principle
MSSG	Mean Squared Sum of Gradients
NG	Natural Gradient
PCA	Principle Component Analysis
PCB	Printed Circuit Board
PDM	Pulse Density Modulation
PDF	Probability Density Function
RIR	Room Impulse Response
RT60	Reverberation time (-60 dB)
RLS	Recursive Least Squares

SDR	Signal to Distortion Ratio
SIR	Signal to Interference Ratio
SAR	Signal to Artifact Ration
STFT	Short-Time Fourier Transform
UMA	Uniform Microphone Array

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. $\mathbf{X}$). Time domain vectors and matrices are denoted in italic bold letters and time index is between square brackets (e.g. $\boldsymbol{x}[n], \boldsymbol{s}[t], \boldsymbol{A}[p]$). Whereas in time-frequency domain they are denoted in straight bold letters, time frequency and time frame indices are between round brackets. (e.g., $\mathbf{x}_{TF}(f,n), \mathbf{s}_{TF}(f,n), \mathbf{W}(f,n)$).

In this document, the following notations are used:

$\mathbb{R}$	Set of real numbers
$\mathbb{C}$	Set of complex numbers
$\mathbb{N}$	Set of natural numbers
$\|.\|$	Absolute value
$\|\|.\|\|_2$	Euclidean norm
$\langle \quad \rangle$	Inner product
$(.)^*$	Complex conjugate operator
$(.)^T$	Transpose operator
$(.)^H$	Hermitian operator
$(.)^-1$	Inverse operator
$(.)^\#$	Pseudo-inverse operator
det(.)	Matrix determinant operator
tr(.)	Trace operator
$\mathbb{E}(.)$	Statistical expectation operator
$H(.)$	Entropy function
$I(.)$	Mutual information function
M	Number of microphones
L	Number of sources
F	Number of frequency bins in the time-frequency representation
P	Mixing filter's length
Q	Demixing filter's length

N	Number of time frames in the time-frequency representation
$\boldsymbol{x}$	Mixtures in the time domain
$\boldsymbol{s}$	Original source signals in the time domain
$\boldsymbol{v}$	Additive noise
$\boldsymbol{y}$	Estimated sources in the time domain
$\boldsymbol{x}_{TF}$	Mixtures in the time frequency domain
$\boldsymbol{y}_{TF}$	Output signal in the T-F domain
$\boldsymbol{\varphi}^{(f)}$	Score function at frequency bin f
$\mathbf{I}_L$	$L \times L$ identity matrix
$\mathbf{D}$	Diagonal matrix
$\mathbf{P}$	Permutation matrix
$\mathcal{C}$	Contrast function
$\boldsymbol{A}$	Mixing matrix
$\boldsymbol{W}$	Demixing/Unmixing matrix
$\mathbf{Rxx}$	Covariance matrix
$\mathbf{J}$	Jacobian matrix of transformation
$\mathfrak{R}$	Correlation matrix between sources vector and score function vector
$\boldsymbol{\mu}$	Mean vector
δ	Kronecker delta
$\mathbf{Q}$	Whitening matrix
$\mathbf{g}$	Activation function
Δ	Natural gradient operator
η	Learning rate
$\Pi\ \{s_l\}$	Projector operator
β, α	Forgetting (smoothing) factor
α_l	Scaling of the l^{th} source
λ	Eigenvalue or wavelength
$\sqrt{\xi}$	Root mean squared
γ	STFT window
ϵ	Small constant
$\mathcal{KL}(.)$	Kullback-Liebler divergence function
$\sigma(.)$	Sigmoid funcion
$\mathbf{\Phi}(f)$	Score function matrix at frequency bin f
argmin	Argument which minimizes
$\mathcal{J}_{IVA}$	IVA objective funcion
$L_2(\mathbb{R})$	Space of square integrable real functions
f_s, T_s	Sampling frequency/period

Introduction

Human beings are endowed with a multitude of senses. One of them is hearing, which is one of the most essential sensing systems that provide crucial inputs for one's perception.

Real-world sound signals are in most cases a mixture of different sound sources and luckily for us, human beings have a tremendous ability to locate, identify, separating each source and focus on one desired sound while eliminating the unwanted ones in real-time while receiving them simultaneously.

Machines however, are less successful at doing that, although some audio systems do a good task for studio recordings with a small number of sources, their performance drops massively when dealing with a large number of sources in a real-world environment that contains many signal reflections and background noise.

Signal processing-based methods have been used in the audio field in order to obtain good audio systems with the best possible sound quality.

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

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

- In the first chapter, we formulate the Blind Source Separation (BSS) problem while giving its mathematical model.

- In the second chapter, we give a structured presentation of an offline blind frequency-domain speech source separation algorithms, particularly on the Independent Vector Analysis (IVA) giving a detailed state of art starting from its origin the Independent Component Analysis (ICA) then two the frequency domain ICA which extends to speech signals by IVA.

- Then, in chapter 3, we give a structured presentation of adaptive (online) speech speech source separation algorithms which is natural gradient based adaptive IVA and we propose to add to it an adaptive whitening. Afterwards we analyse the algorithm performances using synthetic mixtures

- In chapter 4, we give a detailed technical review of the 16 microphone array and use it for real world tests and show the obtained results.

Chapter 1

Background and Related Literature Survey

Blind Source separation was first considered in the 80s after a simple discussion between Bernard Ans, Christian Jutten, and Jeanny Hérault with Jean-Pierre Roll, a neuroscientist, about motion decoding in vertebrates [1].

Blind source separation (BSS) is a technique used to separate a set of mixed signals into their individual sources. The main idea behind BSS is to find an estimate of the original sources from the mixed signals, even when the sources are not known and are only observed through their mixtures. This is a challenging problem because the sources are typically correlated and their number is usually unknown. BSS algorithms use various techniques, such as statistical signal processing and machine learning, to separate the sources. In the past decade, the field of BSS has achieved tremendous development, and BSS has become one of the most promising and exciting topics with solid theoretical foundation and potential applications In the fields of neural computing, advanced statistics, and signal processing. BSS has been successfully applied in various fields such as speech enhancement, recognition, biomedical imaging, image processing, remote sensing, communications systems, exploration seismology, geophysics, econometrics, data mining, and neural networks.

1.1 The cocktail party problem

One well-known example of the BSS problem is the cocktail party problem illustrated in figure 1.1, which involves separating the sounds of individual speakers at a party, where multiple sources of sound are present and interfere with each other. This problem was first introduced by Colin Cherry in 1953 [2] and it has various applications, such as in speech recognition systems, hearing aids, and audio signal processing. The goal of the cocktail party problem is to separate the sounds of each speaker from the mixture of sounds, similar to how the sources in a BSS problem are separated from the mixed signals. The question is, how can we recover the individual speaker?

Figure 1.1: Cocktail party problem.

1.2 Adaptive Blind Source Separation

Adaptive Blind Source Separation consists of separating the sources vector adaptively that is when having access only to the current samples only and not the whole signal. The term "adaptive" in adaptive blind source separation signifies that the separation algorithm can adapt and update its parameters or estimates based on the observed signals, allowing it to dynamically adjust its processing to improve the separation performance.

1.3 The BSS Model

It is assumed that a number of M microphones are being used. An electrical microphone signal is denoted as $x_m[n]$, where n: $n \in \mathbb{N}$ denotes a sample index, and the index m: $m \in \mathbb{N}, m \leq M$ is used to designate the m^{th} microphone signal. In vector notation the *array vector* is $\boldsymbol{x}[n] = \Big(x_1[n]\ x_2[n] \dots\ x_M[n]\Big)^T$.

Let L speech sources be in a *source* vector at discrete time $[n]$ after sampling, $\boldsymbol{s}[n] = \Big(s_1[n] \quad s_2[n] \quad \dots \quad s_L[n]\Big)^T$. $s_l[n]$ denotes the l^{th} source signal at discrete time $[n]$ and the index l: $l \in \mathbb{N}, l \leq L$.

One can assume that there is some unknown mixing model $\boldsymbol{\mathcal{F}}$ such that $\boldsymbol{x}[n] = \boldsymbol{\mathcal{F}}(\boldsymbol{s}[n])$. This unknown mixing system may depend on a variety of things, like the geometry of the array, the physics model involved, the nature of emitted waves, etc.

BSS aims to invert the model (if possible) in order to recover the original signals i.e: identify $\boldsymbol{\mathcal{F}^{-1}}$ such that $\boldsymbol{y}[n] = \boldsymbol{\mathcal{F}^{-1}}(\boldsymbol{x}[n]) = \hat{\boldsymbol{s}}[n]$, where the $\hat{}$ subscript denotes an "estimate". All this while having no *a priori* knowledge of the mixing system and thus the term "blind".

BSS issue can be divided into three categories depending upon the number of sources L and the number of sensors M used to detect the same.

(1). Over-determined mixing: Number of sources $L <$ number of sensors M.
(2). Determined mixing: Number of sources $L =$ number of sensors M.
(2). Under-determined mixing: Number of sources $L >$ number of sensors M.

1.3.1 The instantaneous model

In instantaneous mixing, L unknown source signals $\{s_l[n]\}_{1 \leq l \leq L}$ are combined to yield the M measured sensor signals $\{x_m[n]\}_{1 \leq m \leq M}$ as:

$$x_m[n] = \sum_{l=1}^{L} a_{ml} s_l[n] + v_m[n] \quad m = 1, .., M \tag{1.1}$$

Where a_{ml} are the coefficients of the linear time-invariant mixing system represented by the matrix $\mathbf{A} \in \mathbb{R}^{M \times L}$, called *the mixing matrix*. It is assumed that M noise signals are present, i.e., one noise signal $\{v_m[n]\}_{1 \leq m \leq M}$ per microphone element. In matrix form:

$$\boldsymbol{x}[n] = \mathbf{A}\mathbf{s}[n] + \boldsymbol{v}[n] \tag{1.2a}$$

$$= \sum_{l=1}^{L} \boldsymbol{a}_l s_l[n] + \boldsymbol{v}[n] \tag{1.2b}$$

where $\boldsymbol{a}_l$ is the l^{th} column of matrix $\mathbf{A}$.
BSS for instantaneous mixtures aims to recover the original sources by estimating the coefficients of a filter matrix called *separation matrix* (or *unmixing matrix*) $\mathbf{W} \in \mathbb{R}^{M \times L}$:

$$y_l[n] = \sum_{m=1}^{M} w_{lm} x_m[n] \quad l = 1, .., L \tag{1.3}$$

where $\{y_l[n]\}_{1 \leq l \leq L}$ represents an estimate of a single original source (i.e $\boldsymbol{y}[n] = \hat{\boldsymbol{s}}[n]$) and w_{lm} are the entries of matrix $\mathbf{W}$. In matrix form:

$$\begin{aligned} \boldsymbol{y}[n] &= \mathbf{W}\boldsymbol{x}[n] \\ &= \sum_{m=1}^{M} \mathbf{w}_m x_m[t] \end{aligned} \tag{1.4}$$

where $\mathbf{w}_m$ is the m^{th} column of matrix $\mathbf{W}$ as seen in figure 1.2.

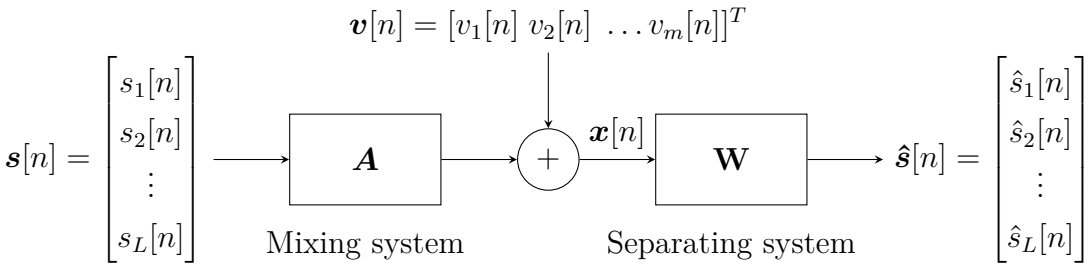


Figure 1.2: Block diagram for instantaneous BSS mixing and demixing processing.

1.3.2 The convolutive model

The received signal for m^{th} microphone, due to the spatial propagation of a source signal related to the l^{th} spatial source, is denoted as $x_{m,l}[n]$. Due to the linearity of the wave equation [11], the propagation of a spatial source signal to a microphone is modeled according to a linear causal convolution:

$$x_{ml}[n] = \sum_{p=0}^{P-1} a_{ml}[p] s_l[n-p]. \tag{1.5}$$

The received microphone signal for a set of L spatial sources, with an additive noise signal $v_m[n]$, is constructed by a linear superposition of all signals:

$$x_m[n] = \sum_{l=1}^{L} x_{ml}[n] + v_m[n] \tag{1.6a}$$

$$= \sum_{l=1}^{L} \sum_{p=0}^{P-1} a_{ml}[p] s_l[n-p] + v_m[n] \tag{1.6b}$$

The coefficient $a_{ml}[p]$ here is an impulse response function that describes the acoustical propagation path between the spatial source number l and microphone m. The length of a propagation path is here assumed to be finite and restricted to P samples, or $\frac{P}{f_s}$ seconds, where f_s is the sampling frequency. In matrix form we get:

$$\boldsymbol{x}[n] = \sum_{p=0}^{P-1} \mathbf{A}[p] \boldsymbol{s}[n-p] \tag{1.7}$$

where $\mathbf{A}[p] \in \mathbb{R}^{M \times L}$ is the *transfer function matrix/ multichannel FIR filter representing the room impulse response (RIR)* for the p^{th} delay, whose elements are denoted $a_{ml}[p]$ as illustrated in figure 1.3.

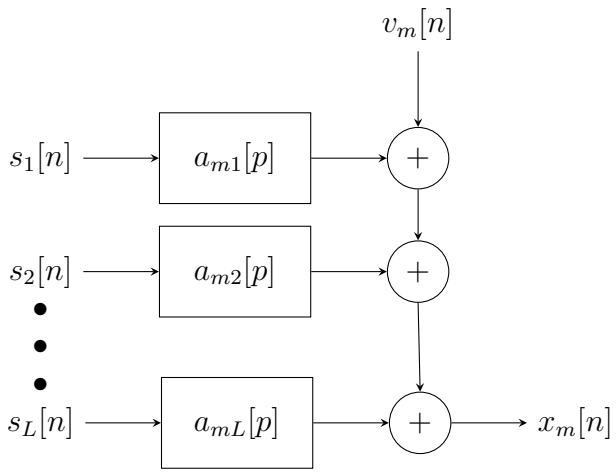


Figure 1.3: Block diagram for convolutive BSS mixing processing.

Convolutive BSS algorithms must exploit both spatial and temporal signal characteristics to function properly, which is why they are sometimes referred to as *spatio-temporal* BSS algorithms.
Unmixing process shown in figure 1.4 consists of estimating an *inverse multichannel separation FIR filter* $\{\mathbf{W}(q)\}_{0 \leq q \leq Q-1} \in \mathbb{R}^{L \times M}$.

The estimated source signal is:

$$y_l[n] = \sum_{m=1}^{M} \sum_{q=0}^{Q-1} w_{lm}[q] x_m[n-q] \tag{1.8}$$

$w_{lm}[q]$ is the inverse filer response of the l^{th} source to the m^{th} microphone at q^{th} delay. Here $\frac{Q}{f_s}$ represents the inverse filter's length in time in seconds. In matrix notation:

$$\boldsymbol{y}[n] = \sum_{q=0}^{Q-1} \mathbf{W}[q] \boldsymbol{x}[n-q] \tag{1.9}$$

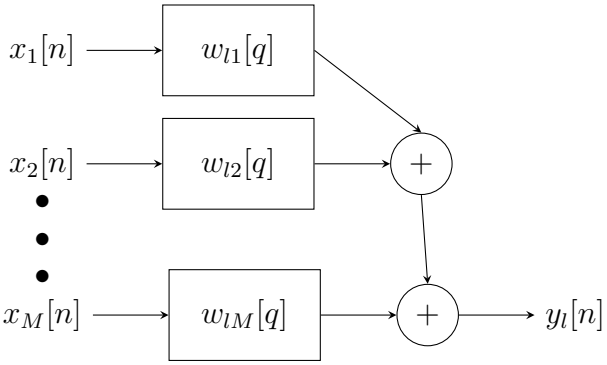


Figure 1.4: Block diagram for convolutive BSS demixing processing.

1.3.3 Why are speech signals convolutive?

Speech signals are inherently convolutive due to the physical mechanisms involved in sound propagation. When a person speaks, their vocal cords produce vibrations that generate sound waves that propagate through the air. These sound waves then reach the listener's ears after undergoing reflections, diffractions, and scattering due to the interaction with the environment, such as walls, ceilings, and objects [3]. These interactions cause the sound waves to propagate through different paths with different delays and attenuations, resulting in a convolutive mixture of the speech signal as shown in figure 1.5.

In addition to the environmental factors, the shape of the human head, mouth, and throat also contributes to the convolutiveness of speech signals. The human vocal tract acts as a filter that shapes the speech signal, resulting in a unique spectral signature for each speaker [4]. This spectral signature changes over time as the speaker changes the shape of their mouth and throat to produce different sounds. These changes in the spectral signature, known as formant transitions, are essential for speech recognition and make the speech signal even more convolutive [5].

The convolutiveness of speech signals presents a significant challenge for speech processing applications such as speech recognition, speaker identification, and speech enhancement. Blind source separation (BSS) algorithms aim to separate the original speech signals from their convolutive mixtures. To achieve this goal, BSS algorithms exploit the statistical independence of the speech sources and the sparsity of the speech signal in some transformed domains.

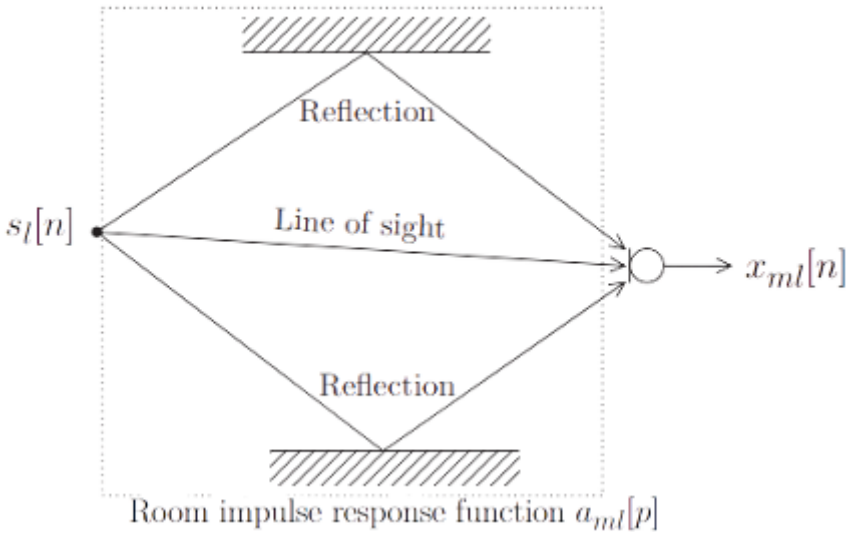


Figure 1.5: A propagation model from a spatial source l, with source signal $s_l[n]$, to a microphone m with signal $x_{ml}[n]$ according to a linear filter $a_{ml}[p]$ for three propagation path components, one line-of-sight component, and two reflection components.

1.3.4 Time-Frequency domain processing

Algorithms that operate in the time domain may suffer from a heavy computational load. This problem is significant even for a moderately advanced task such as computing a matrix multiplication between a square matrix and a vector, which is the case in, e.g., the Recursive Least Squares (RLS) algorithm [6]. The number of operations required for this task is proportionally quadratic to the number of filter coefficients. In addition, the rate of convergence for adaptive filters is generally reduced for long filters since the step-size is often inversely proportional to the number of filter taps [7]. A popular approach in modern signal processing taken in order to circumvent the drawbacks associated with time domain processing, is to introduce a time-frequency representation of the observed signal.

1.3.4.1 Non-stationarity and characteristics of speech signals

It is fortunate that speech signals are feature-rich and possess certain characteristics that enable BSS systems to be developed. Despite the wide-band nature of the voiced speech spectrum, it is well-known that band-limiting speech to frequencies between 300 Hz and 3500 Hz does not significantly harm its intelligibility [8]. The spectrum of unvoiced speech is not inherently band-limited but instead tends to fall off rapidly at the upper and lower frequency edges of human hearing.

Speech is inherently a non-stationary signal, and amplitude modulations are largely responsible for this characteristic. Additional properties of speech signals that are relevant to speech separation include the following:
1. Speech signals originating from different talkers at different spatial locations in an acoustic environment can be considered to be statistically independent.
2. Each speech signal typically has a unique temporal structure over short time frames (less than 1 second).
3. Speech signals are quasi-stationary for small time duration ($\approx$ 10 ms) but non-stationary over longer periods.

Theoretically, all of the above properties can simultaneously be exploited by a separation system [9], although it is possible to design systems that use only one of these features to achieve adequate separation.

1.3.4.2 The Short-Time Fourier Transform

The short-time Fourier transform (STFT) is the classical method of time frequency analysis. The concept is very simple as illustrated in figure 1.6. We multiply $x(t)$, which is to be analysed, with an analysis window $\gamma^*(t-\tau)$ and then compute the Fourier transform:

$$X(\tau, f) = \langle x(t), \gamma(t)_{\tau;f} \rangle = \int_{-\infty}^{\infty} x(t)\gamma^*(t-\tau)e^{-j2\pi ft}dt \tag{1.10}$$

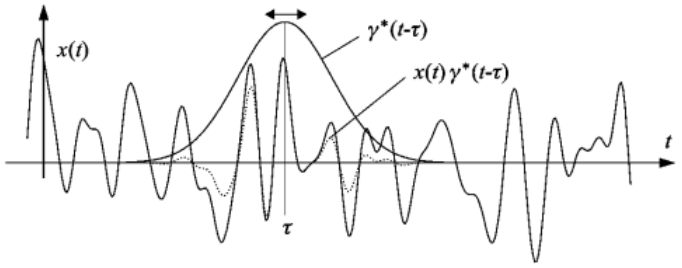
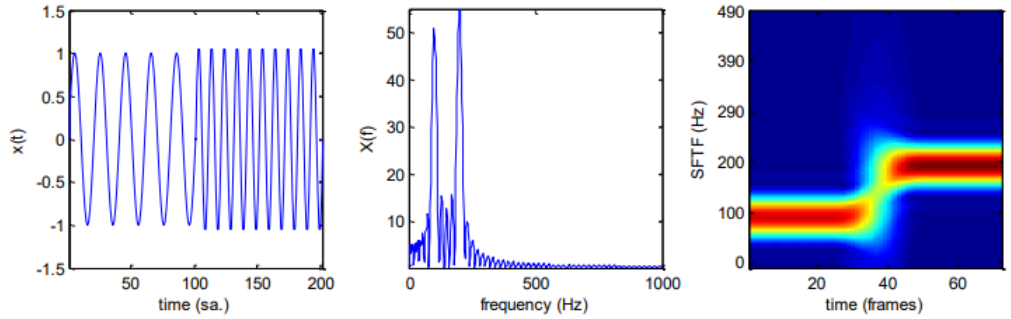


Figure 1.6: Short-time Fourier transform.

Because the STFT usually complex-valued in, we often use the so-called *spectrogram* for display. That is the squared modulus of the STFT:

$$S_x(\tau, f) = |X(\tau, f)|^2 = \left| \int_{-\infty}^{\infty} x(t)\gamma^*(t-\tau)e^{-j2\pi ft}dt \right|^2 \tag{1.11}$$



(a).Time representation (b).Frequency representation (c).Time-frequency representation

Figure 1.7: STFT and spectrogram of a chirp signal.

Fourier transform cannot provide information about both time and frequency – i.e. cannot provide simultaneous time and frequency localisation. And since speech signals are Non-stationary signals (as mentioned before) they may contain varying frequency content at different times, one considers using the STFT instead to capture more information about the signal. Figure 1.7 illustrates an example of a STFT for a chirp signal.

STFT for discrete time signals

Consider a multichannel signal, that is a vector of M observed signals at discrete time $t = nT_s$ by the microphones $\boldsymbol{x}[t] = (x_1[t]\ x_2[t] \dots\ x_M[t])^T$. For discrete time signals the integral in equation (1.10) turns into a summation.

$$\mathbf{x}_{TF}(f, n) = \sum_{t=0}^{T-1} \boldsymbol{x}[t]\gamma[t - nR]e^{-j2\pi\frac{ft}{F}} \tag{1.12}$$

• $n = 1, \dots, N$ is the current time frame index where N is the total number of time frames.

• $f = 1, \dots, F$ indicates the current frequency bin index where K is the total number of frequency bins.

• $t = 0, \dots, T-1$ is a sample in the frame and T is the total number of samples within the same frame.

• R represents the shift in the window or the number of advances in samples between the previous and next frame and $T - R$ is the number of overlapped samples.

• $\gamma[n]$ is the Analysis window which should be zero outside of the time interval $t \in [0, T-1]$.

The choice of the window

There is a trade-off between time and frequency resolution depending on the choice of the window. Basic requirements for $\gamma^*(t)$ to be called a time window are $\gamma^*(t) \in L_2(\mathbb{R})$ and $t\gamma^*(t) \in L_2(\mathbb{R})$. However, the uncertainty principle applies, giving a lower bound for the area of the window:

$$\Delta t \Delta f \geq \frac{1}{4\pi} \tag{1.13}$$

Equation (1.13) translates as follows: Choosing a short time window leads to good time resolution Δt and, inevitably, to poor frequency resolution Δf. On the other hand, a long time window yields poor time resolution, but good frequency resolution.

For Discrete STFT, if F is chosen to be a power of 2 and is generally equal to T and If F is larger than the frame length T, we have to extend $\boldsymbol{x}[t]$ with zeros on both sides before applying the DFT (zero padding).

1.3.4.3 Inverse Short-Time Fourier Transform

The ISTFT can be calculated by several methods. We will describe the weighted overlap- add method [10] in the following: For each time frame n, the IDFT is computed:

$$\mathbf{y}_n[t] = \sum_{f=1}^{F} \mathbf{x}_{TF}(f,n) e^{j2\pi \frac{ft}{F}} \tag{1.14}$$

In order to remove the artifacts effects which are more remarkable at the edges of the frames, another window is applied called the reconstruction window $g[t]$ which also has values zero outside of the time interval $[0, T-1]$. Those IDFT are then added and weighted by the reconstruction window:

$$\mathbf{y}[t] = \sum_{n=0}^{N-1} \mathbf{y}_n[t - nR] g[t - nR] \tag{1.15}$$

In order to have perfect reconstruction (i.e $\boldsymbol{y}[t] = \boldsymbol{x}[t]$), the perfect reconstruction condition is given by:

$$\sum_{n=0}^{N-1} g[t - nR]\gamma[t - nR] = 1 \tag{1.16}$$

Generally, a good choice for the windows is the *Hann window* shown in figure 1.8, as it removes the discontinuities effects at the edges caused by framing, since the window

tapers down to zero at the boarders. It is given by equation:

$$\gamma[t] \triangleq \begin{cases} \frac{1}{2}\left(1-\cos\left(\frac{2\pi t}{T}\right)\right) & 0 \leq t \leq T-1 \\ 0 & \text{elsewhere} \end{cases} \tag{1.17}$$

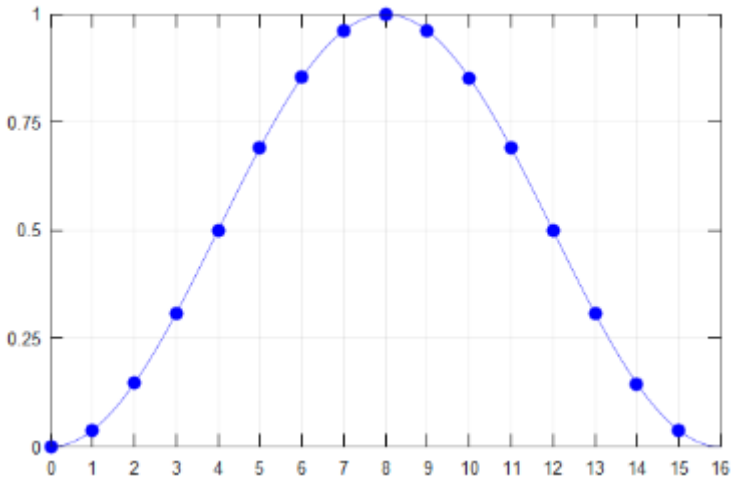


Figure 1.8: Hann window.

1.3.5 The BSS ambiguities

Ideally, BSS algorithms assume as little knowledge as possible about the mixing system or the sources being mixed. This lack of knowledge leads to several ambiguities regarding the possible solutions provided by a BSS algorithm. As most of the information is carried by the shape of the waveform [11], these amplitude and permutation ambiguities do not impact significantly the separation problem in practice.

1.3.5.1 Scaling ambiguity

One cannot recover the exact amplitudes of the original sources, this ambiguity arises from the fact that we can always introduce then cancel out a multiplicative factor without changing the observation vector:

$$\boldsymbol{x}[n] = \sum_{l=1}^{L} \left(\frac{\boldsymbol{a}_l}{\alpha_l}\right) (\alpha_l s_l[n]) \tag{1.18}$$

In matrix form this can be written as

$$\boldsymbol{x}[n] = \mathbf{D}^{-1}\mathbf{D}\boldsymbol{A}\boldsymbol{s}[n] \tag{1.19}$$

where D is a square diagonal non singular matrix $\mathbf{D} = \text{diag}(\alpha_1,\ \alpha_2,\ \ldots,\ \alpha_L)$ that contains the variance (energy) scaling of the sources. To be precise the unmixing model can be formally written as:

$$\hat{\boldsymbol{s}}[n] \stackrel{\circ}{=} \mathbf{D}\boldsymbol{s}[n] \tag{1.20}$$

1.3.5.2 Permutation ambiguity

We cannot determine the order of the estimated sources, this is because re-arraging the sources in the mixture process leaves the observation unchanged. To be precise the unmixing model can be formally written as:

$$\hat{\boldsymbol{s}}[n] \stackrel{\circ}{=} \mathbf{P}\boldsymbol{s}[n] \tag{1.21}$$

where $\mathbf{P}$ is a permuation matrix which is a square matrix with binary enteries which is:

$$[\mathbf{P}_\sigma]_{ij} = \delta_{i,\sigma(j)} = \begin{cases} 1 & \text{if } i = \sigma(j) \\ 0 & \text{otherwise.} \end{cases} \tag{1.22}$$

for any permutation $\{\sigma(i)\}_{i\in(1,2,\ldots,L)}$ where δ is the Kronecker delta.

1.4 Literature survey and related work

J. Hérault et al. were the pioneers in addressing the blind source separation problem, as evidenced by their seminal work in [1] [12]. Their contributions marked the inception of a new field in signal processing, which has since been extensively studied by the research community.

In 1989, Cardoso [13] and Comon [14] made significant breakthroughs by proposing Independent Component Analysis (ICA) as a solution to the blind source separation problem. ICA aims to separate signals by maximizing their independence. Their work laid the foundation for subsequent advancements in blind source separation, with researchers refining and expanding upon the ICA framework. Comon further extended ICA in 1994 with the minimum mutual information approach [15], while Bell et al. proposed the Infomax approach in the following year by maximizing entropy [16]. It was later established that these two approaches were equivalent. Additionally, Amari et al. introduced a natural gradient-based learning rule in 1995 to enhance the ICA algorithm [17]. Aiming to improve the computational efficiency of ICA, Hyvärinen et al. presented the Fixed Point ICA or Fast ICA method in 1997 and 1999 [18] [19].

Other notable algorithms addressing instantaneous mixture blind source separation include Joint Approximate Diagonalization of Eigenmatrices (JADE) by Cardoso et al. in 1993 [20] and 1998 [21], Second Order Blind Identification (SOBI) by Belouchrani et al. in 1997 [22], and the Algorithm for Multiple Unknown Signals Extraction (AMUSE) proposed by Tong et al. in 1990 [23].

Considering speech signals as convolutive mixtures, the problem falls within the domain of Convolutive Blind Source Separation (CBSS). In 2004, Sawada [24] addressed Frequency-Domain Blind Source Separation (FDBSS) by transforming mixtures into the frequency domain and treating each frequency bin as an independent complex-valued ICA problem. However, his approach encountered the permutation problem across frequency bins, which required post-processing solutions. In 2006, Kim [25] proposed an elegant algorithm that tackled the permutation problem by introducing a multivariate probability density function, enforcing dependencies between frequency bins and treating sources as random vectors. To enhance the computational efficiency of this algorithm, Lee et al. introduced FastIVA in 2007 using the Newton method [26]. Berrah and Mendjel proposed a Single Input Multiple Outputs (SIMO) equalization method in [27] to improve the performance of Fast-IVA, which was later published in a conference paper by Belouchrani et al. [28]. Non-Negative Matrix Factorization (NMF), introduced by Lee et al. in 1999 [29], is another well-known algorithm for FDBSS, predominantly used in music. NMF aims to capture the spectral structure of sources by factorizing the observation matrix into the product of two positive definite matrices [29]. Furthermore, an emerging FDBSS method called Independent Low-Rank Matrix Analysis (ILRMA) [30] provides a unified framework that combines IVA and NMF techniques.

In the context of adaptive blind source separation (BSS) of speech signals, notable advancements have been made by several researchers. Parra et al. conducted early work in 2000, where they introduced online BSS techniques for non-stationary signals utilizing decorrelation methods [31]. Building upon this foundation, Kim proposed an adaptive variant of the Independent Vector Analysis (IVA) algorithm in 2010 [32]. In 2014, Taniguchi et al. proposed an adaptive IVA-based algorithm using the auxiliary functions method [33]. However, this algorithm was found to suffer from substantial computational costs. To address this issue, Nakashima et al. subsequently proposed an inverse-free version of the aforementioned algorithm, which significantly reduces the computational burden [34].

1.5 Conclusion

In the first chapter, we have presented a comprehensive introduction to blind source separation and examined various mixing models. We have emphasised the inherent limitation in blind scenarios, where it is impossible to fully identify the mixing matrix. Instead, the identification is achieved up to scaling and permutation. We presented the short time Fourier transform which will be used in the next chapters for frequency domain BSS of speech signals. We also provided the related works and algorithms in blind source separation.

Chapter 2

Independent Vector Analysis state of art

In this chapter, we present the most widely used algorithm for blind source separation of speech signals that is Independent Vector analysis (IVA), but before this we introduce the Independent Component Analysis (ICA) which is at the origin of IVA

2.1 Independent Component Analysis

The concept of independent component analysis (ICA) was first introduced by the researcher J. C. Comon in his 1994 paper "Independent Component Analysis, a new concept?" published in Signal Processing [15]. However, the application of ICA to blind source separation (BSS) was first introduced by A. J. Bell et al. in the 1995 paper "An Independent Component Analysis Framework for Blind Signal Separation." [16]

One of the most widely used techniques for BSS is independent component analysis (ICA), which separates signals based on their statistical independence. ICA is a linear technique that transforms the original mixture of signals into a new representation, where the sources are as statistically independent as possible. This makes it possible to separate signals even when the sources are highly correlated, which is a common scenario in many real-world applications.

2.1.1 ICA model

Assume that there is an L-dimensional zero mean vector $\boldsymbol{s}[n] = \begin{pmatrix} s_1[n], & s_2[n] & \dots, s_L[n] \end{pmatrix}^T$, whose components are *mutually independent*. The vector $\boldsymbol{s}$ corresponds to L independent scalar valued source signals $s_l[n]$. A data vector $\boldsymbol{x}[n] = \begin{pmatrix} x_1[n] & \dots & x_M[n] \end{pmatrix}^T$ is observed at each time point t, such that:

$$\boldsymbol{x}[n] = \boldsymbol{A}\boldsymbol{s}[n] + \boldsymbol{v}[n]$$

where $\boldsymbol{A}$ is an $M \times L$ scalar matrix. The mixing is assumed to be instantaneous so there is no time-delay between the source l mixing into channel m. Once the sources are mixed, the observed signals at each channel are no longer independent, so the goal is to find a demixing matrix $\boldsymbol{W}$ which when applied separates the mixtures in the sense of miximizing an independance criterion (contrast function) such as information maximization, negenentropy maximization and likelihood maximization.

The goal of ICA is to find a linear transformation $\boldsymbol{W}$ of the dependent sensor signals $\boldsymbol{x}$ that makes the outputs as independent as possible:

$$\boldsymbol{y}[n] = \boldsymbol{W}\boldsymbol{x}[n] = \boldsymbol{W}\boldsymbol{A}\boldsymbol{s}[n],$$

where $\boldsymbol{y}$ is an estimate of the sources. The sources are exactly recovered when $\boldsymbol{W}$ is the inverse of $\boldsymbol{A}$ up to a permutation and scale change.

$$\boldsymbol{W}\boldsymbol{A} = \mathbf{PD}$$

2.1.2 The ICA assumptions

Independent compenent analysis has four assumptions:

(1). Mutually Independent sources.
(2). Overdetermined mixing scenario.
(3). Non Gaussian sources.
(4). Additive noise.

2.1.2.1 Mutually independent sources

Independence of random variables is a more general concept than decorrelation. Roughly speaking, we say that random variables y_i and y_j are statistically independent if knowledge of the values of y_i provides no information about the values of y_j.
Mathematically, the independence of the sources $\boldsymbol{s}[t] = \Big(s_1[t], \quad s_2[t] \quad \ldots, s_L[t]\Big)^T$, means that we can write the multivariate probability density function, of the vector as the product of marginal independent distributions i.e:

$$p(\mathbf{s}) = \prod_{i=1}^{L} p_i(s_i) \tag{2.1a}$$

$$p(s_1, s_2, \ldots, s_L) = p_1(s_1)p_2(s_2)\ldots p_L(s_L) \tag{2.1b}$$

This assumption is essential for ICA and everyrthing is based on it.

2.1.2.2 Overdetermined mixing scenario

The number of sensors is greater than or equal to the number of sources $M \geq L$, this is needed to make $\mathbf{A}$ full rank.

2.1.2.3 Non Gaussian sources

ICA uses high order statistics, hence it allows at most one source signal to be Gaussian since a Gaussian process is fully described by second order statistics.

2.1.2.4 Additive noise

No sensor noise or only low additive noise signals are permitted.

2.1.3 The Contrast function

A contrast function is a functional $\mathcal{C}\colon \mathbb{E}^M \to \mathbf{R}$, that measures the independence between N random variables of a random vector $\mathbf{x} \in \mathbb{E}^M$. For ICA, the maxima or minima of these contrast functions correspond to a successful separation of all sources. It should satisfy the following properties:

(1). $\mathcal{C}$ is invariant under a permutation $\mathbf{P}$:

$$\mathcal{C}(\mathbf{Px}) = \mathcal{C}(\mathbf{x})$$

(2). $\mathcal{C}$ is invariant under a scaling $\mathbf{D}$:

$$\mathcal{C}(\mathbf{Dx}) = \mathcal{C}(\mathbf{x})$$

(3). $\mathcal{C}$ decreases under a linear combination $\mathbf{M}$:

$$\mathcal{C}(\mathbf{Mx}) \leq \mathcal{C}(\mathbf{x})$$

If the equality holds i.e $\mathcal{C}(\mathbf{Mx}) = \mathcal{C}(\mathbf{x})$ then the matrix $\mathbf{M}$ is a separating (demixing) matrix.

2.1.4 Data Whitenning

A whitening or sphering transformation is a type of linear transformation that takes a set of random variables with a known covariance matrix and converts them into a new set of variables that are uncorrelated and each has a variance of 1. The transformation is known as "whitening" because it results in the input variables becoming a white noise vector.

The first step is to "center" the data, that is to substract the mean along the time axis that is:

$$\boldsymbol{x}[n] := \boldsymbol{x}[n] - \boldsymbol{\mu} \quad \forall n \tag{2.2}$$

An estimate for the mean vector is:

$$\hat{\boldsymbol{\mu}} = \frac{1}{N} \sum_{n=0}^{N-1} \boldsymbol{x}[n] \tag{2.3}$$

where N represents the total number of samples.

Next, the goal is to find a *whitening* transformation matrix $\mathbf{Q} \in \mathbb{C}^{L \times M}$ such that when applied to $\boldsymbol{x}(t)$ results on a unit covariance matrix.

$$\mathbb{E}[\mathbf{Q}\boldsymbol{x}\boldsymbol{x}^{\boldsymbol{H}}\mathbf{Q}^{H}] = \mathbf{I}_L \tag{2.4}$$

2.1.4.1 Determining the whitenning matrix

The aim is to force the covariance of the white processes to be unity, meaning:

$$\mathbb{E}[\underline{\boldsymbol{x}}[n]\underline{\boldsymbol{x}}[n]^H] = \mathbf{Q}\mathbb{E}[\boldsymbol{x}\boldsymbol{x}^{\boldsymbol{H}}]\mathbf{Q}^H = \mathbf{Q}\boldsymbol{R}_{\boldsymbol{xx}}\boldsymbol{Q}^H = \mathbf{I}_L \tag{2.5}$$

An estimate for the covariance matrix for the pre-whitened data is:

$$\hat{\boldsymbol{R}}_{\boldsymbol{xx}} = \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{x}[n]\boldsymbol{x}[n]^H = \frac{1}{N}\boldsymbol{X}\mathbf{X}^H \tag{2.6}$$

Where $\boldsymbol{X} \in \mathbb{C}^{M \times N}$ is the data matrix containing N samples of the process:

$$\boldsymbol{X} = \begin{pmatrix} \boldsymbol{x}[1] & \boldsymbol{x}[2] & \dots & \mathbf{x}[N] \end{pmatrix} \tag{2.7}$$

One can decompose $\boldsymbol{R}_{\boldsymbol{xx}}$ using Eigenvalue Decomposition (EVD):

$$\boldsymbol{R}_{\boldsymbol{xx}} = \mathbf{E}\mathbf{D}\mathbf{E}^H = \begin{bmatrix} \tilde{\mathbf{E}} & \tilde{\boldsymbol{E}}^{\perp} \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{D}} & 0 \\ 0 & \sigma^2\mathbf{I} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{E}}^H \\ \tilde{\mathbf{E}}^{\perp H} \end{bmatrix}$$

where:
• $\tilde{\mathbf{E}} = \begin{bmatrix} \mathbf{e}_1, \mathbf{e}_2, \dots \mathbf{e}_L \end{bmatrix} \in \mathbb{C}^{M \times L}$ is an isometry contains the column basis of the L sources subspace such that $\tilde{\mathbf{E}}^H\tilde{\mathbf{E}} = \boldsymbol{I}$ whereas $\tilde{\mathbf{E}^{\perp}} \in \mathbb{C}^{M \times (M-L)}$ is the orthogonal complementary matrix, its columns form a basis for the noise subspace. the matrix $\mathbf{E}$ is unitary.

• $\tilde{\mathbf{D}} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_L) \in \mathbb{R}^{L \times L}$ are the L largest eigenvalues of $\boldsymbol{R}_{\boldsymbol{xx}}$ which correspond to the L signals power.

Just like for the Principal Component Analysis (PCA) we choose the L most significant basis vectors of $\mathbf{E}$ this yields:

$$\mathbf{R}_{xx} \approx \tilde{\mathbf{E}}\tilde{\mathbf{D}}\tilde{\mathbf{E}}^H = (\tilde{\mathbf{E}}\tilde{\mathbf{D}}^{\frac{1}{2}})(\tilde{\mathbf{E}}\tilde{\mathbf{D}}^{\frac{1}{2}})^H \tag{2.8}$$

Now determining the whitening matrix $\mathbf{Q}$

$$\mathbf{Q}(\tilde{\mathbf{E}}\tilde{\mathbf{D}}^{\frac{1}{2}})(\tilde{\mathbf{E}}\tilde{\mathbf{D}}^{\frac{1}{2}})^H\mathbf{Q}^H = \mathbf{I}_L \tag{2.9}$$

Since $\tilde{\mathbf{E}}$ is an isometry, a choice for $\mathbf{Q}$ that verifies equation (2.9) is:

$$\hat{\mathbf{Q}} = (\tilde{\mathbf{E}}\tilde{\mathbf{D}}^{-\frac{1}{2}})^H = \left(\lambda_1^{-\frac{1}{2}}\mathbf{e_1} \quad \lambda_2^{-\frac{1}{2}}\mathbf{e_2} \quad \dots \quad \lambda_L^{-\frac{1}{2}}\mathbf{e_L}\right)^H \tag{2.10}$$

The last step is then to project the data and to perform *dimension reduction*:

$$\mathbf{X}_p = \mathbf{Q}\mathbf{X} \tag{2.11}$$

The new projected matrix $\mathbf{X}_p \in \mathbb{C}^{L \times N}$, this turns estimating the separating matrix $\boldsymbol{W}$ into a square matrix problem $\boldsymbol{W} \in \mathbb{C}^{L \times L}$.

2.1.5 Information maximization

As the components of the observed vectors are no longer independent, the multivariate probability density function, will not satisfy the product equality in equation (2.1a). The mutual information $I(\mathbf{x})$ of the observed vector is given by the Kullback-Leibler (KL) divergence $\mathcal{KL}(.||.)$ of the multivariate density from the density written in product form:

$$I(\mathbf{x}) = \int p(\mathbf{x}) \log \frac{p(\mathbf{x})}{\prod_{i=1}^{L} p_i\left(x_i\right)} d\mathbf{x} = \mathcal{KL}\left(p(\mathbf{x}) \| \prod_{i=1}^{L} p_i\left(x_i\right)\right) \tag{2.12}$$

The mutual information is a contrast and is positive and is equal to zero (minimum) only when the components x_i are independent.

The KL divergence can be thought of as a measure of "distance" between two different pdf's, the more the two pdf's are similar the less the distance between them. Of course it is not really a distance in the mathematical sense as it doesn't verify the symmetry property since:

$$\mathcal{KL}(p(a)\|p(b)) \neq \mathcal{KL}(p(b)\|p(a)) \tag{2.13}$$

Nadal et al. [35] showed that in the low-noise case, the minimum of the mutual information between the inputs x and outputs y of a neural processor implied that the output distributions were factorial. So in order to estimate the source vector $\boldsymbol{y} = \boldsymbol{W}\boldsymbol{x}$ so that its components y_l are independent, one should maximize the mutual information $I(\boldsymbol{y})$ which would imply that $p(\mathbf{y}) = \prod_{l=1}^{L} p_l(y_l)$ and thus sources are mutually independent.

Roth et al. [36] and Bell et al.[37] independently derived stochastic gradient learning

rules for this maximization and applied them, respectively, to forecasting, time series analysis, and the blind separation of sources. Bell et al. [37] proposed a simple learning algorithm for a feedforward neural network that blindly separates linear mixtures $\boldsymbol{x}$ of independent sources $\boldsymbol{s}$ using information maximization. They show that maximizing the joint entropy $H(\boldsymbol{y})$ of the output of a neural processor can approximately minimize the mutual information among the output components $u_i = g(y_i)$ where $g(y_i)$ is an invertible monotonic nonlinearity and $\boldsymbol{y} = \boldsymbol{W}\boldsymbol{x}$. The joint entropy at the outputs of a neural network is:

$$H(\boldsymbol{u}) = H(u_1, u_2, \ldots, u_L) = \sum_{l=1}^{L} H(u_l) - I(\boldsymbol{u}) \tag{2.14}$$

where:

$$H(\mathbf{x}) = -\int p(\mathbf{x}) \log(p(\mathbf{x}))d\mathbf{x} \tag{2.15a}$$

$$H(x_i) = -\int p(\mathbf{x}) \log(p(x_i))d\mathbf{x} \tag{2.15b}$$

The maximal value for $H(u_1, \ldots, u_L)$ is achieved when the mutual information among the bounded random variables $\mathbf{u} = \begin{pmatrix} u_1 & \ldots & u_L \end{pmatrix}$ is zero and their marginal distribution is uniform. As we will show below, this implies that the nonlinearity $g(y_i)$ has the form of the cumulative density function (cdf) of the true source distribution s_i [16]. Bell et al. choose the nonlinearity to be a fixed logistic function. This is equivalent to assuming a prior distribution of the sources to be a super-Gaussian distribution with heavy tails and a peak centred at the mean. The relationship between u_i and y_i is

$$p(u_i) = \frac{p(y_i)}{\left|\dfrac{\partial g(y_i)}{\partial y_i}\right|} \tag{2.16}$$

For a uniform distribution of u_i, it follows that

$$p(y_i) = \left|\frac{\partial g(y_i)}{\partial y_i}\right| \tag{2.17}$$

2.1.6 Deriving the gradient of the entropy

The probability density function of $\mathbf{u} = \boldsymbol{g}(\mathbf{y})$ and $\boldsymbol{y}$ are related by:

$$p(\mathbf{g}(\mathbf{y})) = \frac{p(\boldsymbol{y})}{|\det(\mathbf{J}(\boldsymbol{y})|} \tag{2.18}$$

where $\mathbf{J}(\boldsymbol{y})$ is the Jacobian matrix of the transformation between $\mathbf{g}(\boldsymbol{y})$ and $\boldsymbol{y}$ which is a diagonal matrix of the form

$$\begin{pmatrix} \frac{\partial g_1}{\partial y_1} & 0 & \dots & 0 \\ \vdots & \frac{\partial g_2}{\partial y_2} & \dots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & \dots & \frac{\partial g_L}{\partial y_L} \end{pmatrix} \tag{2.19}$$

And the probability density function of $\boldsymbol{y}$ and $\boldsymbol{x}$ are related by

$$p(\boldsymbol{y}) = \frac{p(\boldsymbol{x})}{|\det(\boldsymbol{W})|} \tag{2.20}$$

The expression of the entropy is therefore

$$H(\mathbf{g}(\boldsymbol{y})) = -\mathbb{E}[\log p(\mathbf{g}(\boldsymbol{y}))] = \mathbb{E}[\log|\det(\mathbf{W})|] + \mathbb{E}[\log|\det(\mathbf{J}(\boldsymbol{y}))|] - \mathbb{E}[\log p(\boldsymbol{x})] \tag{2.21}$$

Taking the derivative with respect to $\boldsymbol{W}$

$$\frac{\partial H(\mathbf{g}(\boldsymbol{y}))}{\partial \boldsymbol{W}} = \frac{\partial}{\partial \boldsymbol{W}} \log|\det(\boldsymbol{W})| + \frac{\partial}{\partial \mathbf{W}} \log \prod_{l=1}^{L} \left| \frac{\partial g_l}{\partial y_l} \right| \tag{2.22}$$

$$= \frac{\partial H(\mathbf{g}(\boldsymbol{y}))}{\partial \boldsymbol{W}} = \frac{\partial}{\partial \boldsymbol{W}} \log|\det(\boldsymbol{W})| + \frac{\partial}{\partial \mathbf{W}} \sum_{l=1}^{L} \log\left(p_l(y_l)\right) \tag{2.23}$$

For the first term in equation (2.22): $\frac{\partial}{\partial \boldsymbol{W}} \log|\det(\boldsymbol{W})| = \left(\boldsymbol{W}^H\right)^{-1}$. In the second term, the product splits up into sums of log-terms, in which only one is dependent on a particular $\boldsymbol{W}_{ij}$, and hence,

$$\frac{\partial}{\partial \mathbf{W}} \log \prod_{l=1}^{L} \left| \frac{\partial g_l}{\partial y_l} \right| = -\boldsymbol{\varphi}(\boldsymbol{y})\boldsymbol{x}^H \tag{2.24}$$

where $\boldsymbol{\varphi}(\boldsymbol{y})$ is the gradient vector of the log likelihood called the *score funcion*

$$\boldsymbol{\varphi}(\boldsymbol{y}) = -\frac{\partial \log\left(p(\boldsymbol{y})\right)}{\partial \boldsymbol{y}} = \begin{pmatrix} \varphi_1(y_1) & \varphi_2(y_2) & \dots & \varphi_L(y_L) \end{pmatrix}^T \tag{2.25}$$

The final expression for the gradient is therefore:

$$\nabla H(\boldsymbol{y}) = \left(\boldsymbol{W}^H\right)^{-1} - \boldsymbol{\varphi}(\boldsymbol{y})\boldsymbol{x}^H \tag{2.26}$$

Since entropy is equal to mutual information up to a constant term as shown in equation (2.14) they have the same gradient as in equation (2.26).In what follows we use the mutual information as an objective function and we won't need to derive the gradient again.

2.1.7 Deriving the natural gradient learning rule

Consider the loss function to be this time the mutual information $I(\boldsymbol{y})$ of the output sources

$$I(\boldsymbol{y}, \boldsymbol{W}) = -\log|\det(\boldsymbol{W})| - \sum_{l=1}^{L} \log\left(p_l(y_l)\right) + const \tag{2.27}$$

We can then apply the stochastic gradient descent learning method to derive a learning rule. In order to calculate the gradient of I, we derive the total differential dI of l when $\boldsymbol{W}$ is changed from $\boldsymbol{W}$ to $\boldsymbol{W} + d\boldsymbol{W}$. In component form,

$$dI = I(\mathbf{y}, \boldsymbol{W} + d\boldsymbol{W}) - I(\boldsymbol{y}, \boldsymbol{W}) = \sum_{i,j} \frac{\partial I}{\partial w_{ij}} dw_{ij} \tag{2.28}$$

where $\dfrac{\partial I}{\partial w_{ij}} dw_{ij}$ represents the gradient of I. Simple differential calculus yields

$$dI = -\mathrm{tr}(d\boldsymbol{W}\boldsymbol{W}^{-1}) + \boldsymbol{\varphi}(\boldsymbol{y})^H d\mathbf{y} \tag{2.29}$$

From $\boldsymbol{y} = \boldsymbol{W}\boldsymbol{x}$, we have

$$d\boldsymbol{y} = d\boldsymbol{W}\boldsymbol{x} = d\boldsymbol{W}\boldsymbol{W}^{-1}\boldsymbol{y} \tag{2.30}$$

We set $d\boldsymbol{X} = d\boldsymbol{W}\boldsymbol{W}^{-1}$, the gradient dI in equation (2.28) is expressed in the differential form:

$$dI = -\mathrm{tr}(d\boldsymbol{X}) + \boldsymbol{\varphi}(\boldsymbol{y})^H d\boldsymbol{X}\boldsymbol{y} \tag{2.31}$$

This leads to the stochastic gradient learning algorithm,

$$\Delta\boldsymbol{X}(k) = \boldsymbol{X}(k+1) - \boldsymbol{X}(k) = -\eta(k)\frac{dI}{d\boldsymbol{X}} = \eta(k)\left[\boldsymbol{I} - \boldsymbol{\varphi}(\boldsymbol{y}(k))\boldsymbol{y}(k)^H\right] \tag{2.32}$$

in terms of $\Delta\boldsymbol{X}(k) = \Delta\boldsymbol{W}(k)\boldsymbol{W}^{-1}(k)$ this is rewritten as

$$\Delta\boldsymbol{W}(k) = \boldsymbol{W}(k+1) - \boldsymbol{W}(k) = \eta(k)\left[\boldsymbol{I} - \boldsymbol{\varphi}(\boldsymbol{y}(k))\boldsymbol{y}(k)^H\right]\boldsymbol{W}(k) \tag{2.33}$$

This yields

$$\boldsymbol{W}(k+1) = W(k) + \eta(k)\left[\boldsymbol{I} - \boldsymbol{\varphi}(\boldsymbol{y}(k))\boldsymbol{y}(k)^H\right]\boldsymbol{W}(k) \tag{2.34}$$

The last step is to use the expectation value $\mathbb{E}$ of the natural gradient with respect to $\mathbf{y}$ for the batch (offline) processing. This yields the following well known ICA learning rule

$$\boldsymbol{W}(k+1) = W(k) + \eta(k)\Big[\boldsymbol{I} - \mathbb{E}[\boldsymbol{\varphi}(\boldsymbol{y}(k))\boldsymbol{y}(k)^H]\Big]\boldsymbol{W}(k) \tag{2.35}$$

Here the index k denotes the k^{th} iteration of the batch mode natural gradient descent.

2.2 Independent Vector Analysis (IVA)

Independent vector Analysis first proposed by Kim et all in [25], it extends the work done on Frequency Domain Independent Component Analysis FDCICA where the whole frequency spectrum is exploited to separate the sources from the convolutive mixtures. The IVA model assumes a linear mixing scenario in each frequency bin separately. However, 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. This solves the permutation problem across the frequency bins and that's by assuming dependency between the elements of a source vector by appropriate choice of the probability density function.

2.2.1 Batch IVA model

By applying the STFT to the linear model, and by assuming that the window length is sufficiently larger than the filter's length. One gets at frame index n:

$$\mathbf{x}_{TF}(f,n) = \mathbf{A}(f)\mathbf{s}_{TF}(f,n) \tag{2.36}$$

In matrix form for all N time frames one gets:

$$\mathbf{X}_{TF}(f) = \mathbf{A}(f)\mathbf{S}_{TF}(f) \tag{2.37}$$

where $\mathbf{X}_{TF}(f) = \Big(\mathbf{x}_{TF}(f,1) \quad \dots \quad \mathbf{x}_{TF}(f,N)\Big) \in \mathbb{C}^{M \times N}$ and $\mathbf{S}_{TF}(f) = \Big(\mathbf{s}_{TF}(f,1) \quad \dots \quad \mathbf{s}_{TF}(f,N)\Big) \in \mathbb{C}^{L \times N}$

One can treat this problem as several numbers (F) of ICA problems, because (2.35) can be rewritten as:

$$\mathbf{X}_{TF}(1) = \mathbf{A}(1)\mathbf{S}_{TF}(1) \quad , \quad \dots \quad , \quad \mathbf{X}_{TF}(F) = \mathbf{A}(F)\mathbf{S}_{TF}(F) \tag{2.38}$$

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

$$\mathbf{y}_{TF}(f,n) = \mathbf{W}(f)\mathbf{x}_{TF}(f,n) \tag{2.39}$$

2.2.2 Permutation ambiguity and scaling ambiguity in FDICA

One approach to address the challenge of convolutive mixtures in blind source separation is to work in the frequency domain. An initial strategy could be to estimate the unmixing matrix separately for each frequency bin, treating them as independent instantaneous problems. While this may seem promising, it becomes apparent that the permutation ambiguity, a common issue in ICA, has a significant impact and this would require post processing to solve permutation misalignement as shown in figure 2.2.

When solving the problem independently for each frequency bin, it is highly unlikely that the estimated mixtures at different frequency bins will be consistently ordered. This inconsistency is illustrated in Figure 2.1, where source s_2 is erroneously positioned as source s_1, and source s_L is placed incorrectly as well.

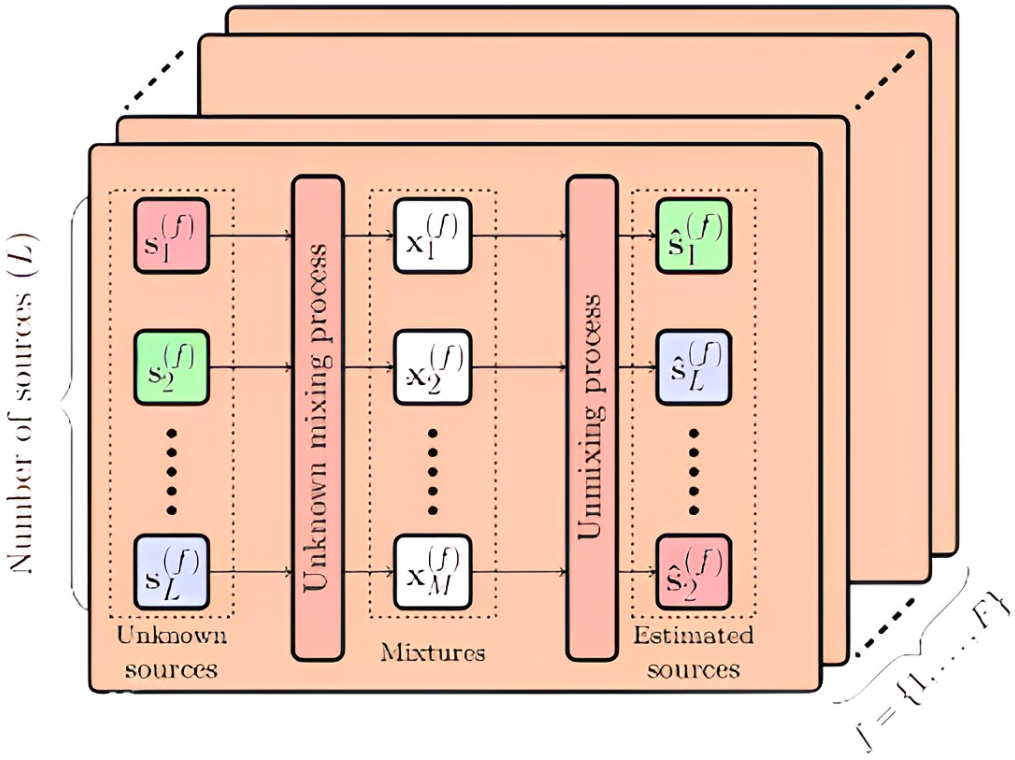


Figure 2.1: Permutation problem in frequency domain ICA (FDICA).

And this also of s_2 and s_1 is in the position of s_L . The estimated source's order would likely be different in each frequency bin. This is an inherent problem to ICA and the permutation is not known at each frequency bin (illustrated by the pink 'slices' in Figure 2.1). The instantaneous model is not suited to realistic mixing environments due to time delays in the convolutive mixing model.

Early attempts to address this challenge involved canceling 4th-order cross cumulants to estimate sources, as described in [38]. However, the frequency domain is often preferred for convolutive mixtures, as represented in Equation (2.35) for the associated mixing model. Early methods, such as those presented in [39] and [40] introduced feedback networks based approaches [37].

Parra and Spence's method, introduced in 2000 [31], imposed restrictions on filter lengths in the time domain to enforce "smoothness" across frequency bins. In 2004, another robust approach combined direction of arrival with interfrequency correlation [24].

An attempt to model multidimensional ICA was made in [21], which considered dependencies between frequency bins. This method utilizes independent subspace analysis (ISA) [41], which does not require independence between sources but relies on independence in the projections onto subspaces to model dependencies found in frequency domain speech signals.

However, the idea of independent vector analysis (IVA), introduced in [25], explicitly models dependencies between frequency bins. It formulates the algorithm by considering dependencies within vector sources and independence between vector sources, using a multivariate probability density function (pdf). IVA is currently the most promising method within the ICA-style framework, addressing the permutation problem in frequency domain blind source separation (FD-BSS).

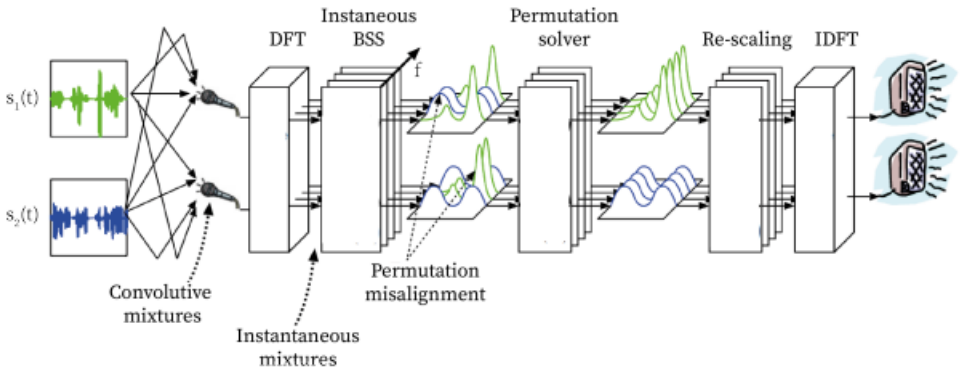


Figure 2.2: Solving the permutation ambiguity with post processing [42].

As for the scaling ambiguity, one way to take advantage of it, is in pre-processing. That is by keeping the output signal vectors $\mathbf{y}_{TF_l}$ zero-mean and white, and by constraining the whitening matrix $\mathbf{W}(f)$ to be orthogonal [43]:

$$\mathbb{E}[\mathbf{x}_{TF}(f)\mathbf{x}_{TF}^{H}(f)] = \mathbf{I}_M \quad f = 1 \dots F \tag{2.40}$$

$$\mathbf{W}(f)\mathbf{W}^{H}(f) = \mathbf{I}_L \quad f = 1 \dots F \tag{2.41}$$

2.2.3 Multivariate Probability Density Function

As seen in the previous section, in order to perform separation based on the ICA algorithm, one needs to assign a good approximation for the p.d.f of the sources $p(\mathbf{y}_{TF_l})$ where $\mathbf{y}_{TF_l} = \begin{pmatrix} y_{TF_l}(1) & y_{TF_l}(2) & \dots & y_{TF_l}(F) \end{pmatrix}^T$ is the l^{th} source estimate vector across all F frequency bins. Here the time structure n is omitted for convenience, as it is treated as an observation of the random vector $\mathbf{y}_{TF_l}$.

The chosen pdf should force the inter-frequency dependency in order to overcome the permutation ambiguity across frequency bins. Previous research such as the work by Brehm and Stammler [44] has identified the spherical symmetry characteristic of speech signals, leading to the introduction of spherically invariant random processes (SIRP) as a modeling framework for band-limited speech. This is done by introducing a spherically symmetric pdf for the sources that is a function of the norm of the random vector

$$p(\mathbf{y}_{TF_l}) \propto f(||\mathbf{y}_{TF_l}||_2) = f\left(\sqrt{\sum_{f=1}^{F} |y_{TF_l}(f)|^2}\right) \tag{2.42}$$

where $||\mathbf{y}_{TF_l}||_2$ is the 2-norm of the vector.
In the literature [25] [45], an usual choice for the spherically symmetric pdf is the Laplace distribution.

$$p(\mathbf{y}_{TF_l}) = \rho \exp\left\{-\left(\sqrt{\sum_{f=1}^{F} |y_{TF_l}(f)|^2}\right)\right\} \tag{2.43}$$

The score function of the l^{th} source at frequency bin f can therefore be immediately obtained by:

$$\varphi_l^{(f)} = -\frac{\partial \log p(y_{TF_l}(f))}{\partial y_{TF_l}} = \frac{y_{TF_l}(f)}{\sqrt{\sum_{f=1}^{F} |y_{TF_l}(f)|^2}} \tag{2.44}$$

And the score function vector for all sources $\boldsymbol{\varphi}^{(f)} = \begin{pmatrix} \varphi_1^{(f)} & \varphi_2^{(f)} & \dots & \varphi_l^{(f)} \end{pmatrix}^T$
In matrix form for all N time observations:

$$\mathbf{\Phi}(f) = \begin{pmatrix} \boldsymbol{\varphi}^{(f)}(1) & \boldsymbol{\varphi}^{(f)}(2) & \dots & \boldsymbol{\varphi}^{(f)}(N) \end{pmatrix} \tag{2.45}$$

Another common choice for the pdf is the derivative of the sigmoide function [37]:

$$p(y_{TF_l}(f)) = \frac{\partial}{\partial y_{TF_l}} \sigma(y_{TF_l}(f)) = \sigma(y_{TF_l}(f))(1 - \sigma(y_{TF_l}(f))) \tag{2.46}$$

where $\sigma(y_{TF_l}(f)) = \dfrac{1}{1+e^{-y_{TF_l}(f)}}$ is a cdf and widely used for neural networks. The corresponding score function is therefore:

$$\varphi_l^{(f)} = 1 - 2\sigma(y_{TF_l}(f)) \tag{2.47}$$

2.2.4 Cost function

As derived previously for ICA, the cost function to be optimized is either the mutual information $I(\mathbf{y}_{TF})$ or the entropy $H(\mathbf{y}_{TF})$ since minimizing the first is equivalent to maximazing the second:

$$\mathcal{J}_{IVA} = I(\mathbf{W}(f), \mathbf{y}_{TF}) = \mathcal{KL}\left(p(\mathbf{y}_{TF}) \| \prod_{l=1}^{L} p_i\left(y_{TF_l}\right)\right) \tag{2.48a}$$

$$= \int p(\mathbf{y}_{TF_1}, \ldots, \mathbf{y}_{TF_L}) \log\left(\frac{p(\mathbf{y}_{TF_1}, \ldots, \mathbf{y}_{TF_L})}{\prod_{l=1}^{L} p(\mathbf{y}_{TF_l})}\right) d\mathbf{y}_{TF_1} \ldots \mathbf{y}_{TF_L} \tag{2.48b}$$

$$= \int p(\mathbf{x}_{TF_1}, \ldots, \mathbf{x}_{TF_M}) \log\left(\frac{p(\mathbf{x}_{TF_1}, \ldots, \mathbf{x}_{TF_M})}{\prod_{l=1}^{L} p(\mathbf{x}_{TF_l})}\right) d\mathbf{x}_{TF_1} \ldots \mathbf{x}_{TF_M} \tag{2.48c}$$

$$- \sum_f \log|\det(\mathbf{W}(f)| - \sum_l^L \log p(\mathbf{y}_{TFl})$$

$$\mathcal{J}_{IVA} = -\sum_f \log|\det(\mathbf{W}(f)| - \sum_l^L \log p(\mathbf{y}_{TFl}) + const \tag{2.49}$$

When the estimated sources $\mathbf{y}_{TF1}, \mathbf{y}_{TF2}, \ldots \mathbf{y}_{TFL}$ are mutually independent, the joint pdf should be the product of the marginal pdfs, i.e., $p(\mathbf{y}_{TF1}, \ldots, \mathbf{y}_{TFL}) = \prod_{l=1}^{L} p(\mathbf{y}_{TFl})$ and:

$$\hat{\mathbf{W}}(f) = \text{argmin}_{\mathbf{W}(f)} \mathcal{J}_{IVA} \tag{2.50}$$

2.2.5 Update rule

The same update rule derived for the ICA is used for the IVA across each frequency bin f using the natural gradient:

$$\mathbf{W}(f) = \mathbf{W}(f) + \eta\left(\mathbf{I} - \mathbb{E}[\boldsymbol{\varphi}^{(f)} \mathbf{y}_{TF}(f)^H]\right) \mathbf{W}^H(f) \tag{2.51}$$

The expectation value can be estimated over the N time observations:

$$\mathbb{E}[\boldsymbol{\varphi}^{(f)}\mathbf{y}_{TF}^{H}(f)] = \frac{1}{N}\sum_{n=0}^{N-1}\varphi^{(f)}(n)\mathbf{y}_{TF}^{H}(f,n) = \frac{1}{N}\boldsymbol{\Phi}(f)\mathbf{Y}_{TF}^{H}(f) \tag{2.52}$$

2.2.6 Re-scaling

In order to fix the scaling ambiguity due to the separation process that is we cannot determine the scales of the sources exactly, one uses a rescaling method, in IVA the Minimal Distortion Principal (MDP) [46] is used, the MDP uses the following transformation to correct the scaling of the sources:

$$\mathbf{W}_s(f) = \mathrm{diag}(\hat{\mathbf{A}}(f))\mathbf{W}(f) \tag{2.53}$$

where $\hat{\mathbf{A}}(f) = \mathbf{W}(f)^{\#}$.

The MDP consists of the following: In a set of valid separators, choose $\mathbf{W}$ such that it minimises $\mathbb{E}[||\mathbf{y} - \mathbf{P}\mathbf{x}||^2]$ for some permutation matrix $\mathbf{P}$. One can show that this leads to equation (2.53) in the case were we do not consider any permutation i.e $\mathbf{P} = \mathbf{I}$.

In this case the proof can be derived easily. Recall that the separation process is achieved up to a diagonal matrix $\mathbf{D}(f)$:

$$\mathbf{W}(f)\mathbf{A}(f) = \mathbf{D}(f) \tag{2.54}$$

which yields:

$$\mathrm{diag}(\mathbf{W}^{\#}(f)) = \mathrm{diag}(\mathbf{A}(f)D^{-1}(f)) = \mathrm{diag}(\mathbf{A}(f))\mathbf{D}^{-1}(f) \tag{2.55}$$

and one gets:

$$\mathbf{W}_s(f)\mathbf{A}(f) = \mathrm{diag}(\mathbf{A}(f))\mathbf{D}^{-1}(f)\mathbf{D}(f) \tag{2.56a}$$

$$\mathbf{W}_s(f)\mathbf{A}(f) = \mathrm{diag}(\mathbf{A}(f)) \tag{2.56b}$$

Equation (2.54) shows that the scaling ambiguity is therefore solved using the MDP.

2.2.7 Algorithm summary

Here below the pseudo-code for the batch Independent vector analysis algorithm.

Algorithme 1 : IVA algorithm

input : Observed mixtures $\boldsymbol{X} = [\boldsymbol{x}_1, ..., \boldsymbol{x}_M]^T$, number of sources L , number of iterations N_{Iter}, learning rate $\eta, NDFT, window$

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

Compute thr STFT $\mathbf{X}_{TF} \in \mathbb{C}^{M \times F \times N}$ of the mixtures data $\boldsymbol{X}$
for $f \leftarrow 1$ **to** F **do**
  // Data whitening
  $\mathbf{X}_{TF}(f) = \mathbf{X}_{TF}(f) - \boldsymbol{\mu}(f) * ones(1, N)$ // $\mu(f)$: along the time axis
  $\mathbf{R}_{XX}(f) = \frac{1}{N}\mathbf{X}_{TF}(f)\mathbf{X}_{TF}^H(f)$
  $[\mathbf{E}(f), \mathbf{D}(f)] = Eig(\mathbf{R}_{XX}(f))$
  $\mathbf{Q}(f) = \mathbf{D}^{-\frac{1}{2}}(f)\mathbf{E}^T(f)$
  $\mathbf{X}_p(f) = \mathbf{Q}(f)\mathbf{X}_{TF}(f)$
  // Initialize of $\mathtt{W}_p$
  $\mathbf{W}_p(f) = I$
end
// Learning rules
for $iter \leftarrow 1$ **to** N_{iter} **do**
  for $f \leftarrow 1$ **to** F **do**
    $\mathbf{Y}_{TF}(f) = \mathbf{W}_p(f)\mathbf{X}_p(f)$
    Calculate the score function $\varphi_l^{(f)} = \frac{y_{TF_l}(f)}{\sqrt{\sum_{f=1}^{F} y_{TF_l}(f)|^2)}}$ for all l
    Define $\boldsymbol{\varphi}^{(f)}(n) = [\varphi_1^{(f)}(n), ..., \varphi_L^{(f)}(n)]$
    Concatenate N frames $\boldsymbol{\Phi}(f) = \left[\boldsymbol{\varphi}^{(f)}(1) \quad \dots \quad \boldsymbol{\varphi}^{(f)}(N)\right]$
    Estimate $\mathbb{E}\left[\boldsymbol{\varphi}^{(f)}\mathbf{y}_{TF}^H\right] \approx \frac{1}{N}\boldsymbol{\Phi}(f)\mathbf{Y}_{TF}^H(f)$
    Compute the gradient $\Delta\mathbf{W}_p(f) = \left\{\mathbf{I} - \mathbb{E}\left[\boldsymbol{\varphi}^{(f)}\mathbf{y}_{TF}^H(f)\right]\right\}\mathbf{W}_p(f)$
    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**
  $\mathbf{W}(f) = \mathbf{W}_p(f)\mathbf{Q}(f)$ // Go back to original dimension
  $\mathbf{W}(f) = \mathrm{diag}(\mathbf{W}^{\#}(f))\mathbf{W}(f)$ // Apply minimal distortion principle
  $\mathbf{Y}_{TF}(f) = \mathbf{W}(f)\mathbf{X}_{TF}(f)$
end
Calculate the ISTFT of $\mathbf{Y}_{TF}(f)$

2.3 Conclusion

In this chapter, our focus was on the exploration of the renowned Independent Component Analysis (ICA) algorithm and its extension to the frequency domain using the Short-Time Fourier Transform (STFT). Subsequently, we delved into the realm of offline Independent Vector Analysis (IVA) and provided a comprehensive discussion on its mathematical model and iterative update rule. Notably, we highlighted the elegant solution offered by IVA for addressing the permutation problem encountered across frequency bins, due to the sources multivariate priors. Additionally, we elucidated the utilization of the MDP to correct the scaling of the output signal. This will be useful for the next chapter when deriving the adaptive version of the Algorithm.

Chapter 3

Adaptive Independent Vector Analysis

In this chapter we study the adaptive version of natural gradient based Independent Vector Analysis mainly the one introduced by Kim in [32] which we in addition propose an adaptive whitening to it in addition. The block index (n) is introduced to the unmixing model to highlight the iterative nature over time of the adaptive version. We afterwards generate synthetic mixtures in different scenarios (two and three speakers of the two different genders) using Python and measure the separation performance and also test it in noisy environment.

3.1 Natural Gradient based adaptive Independent Vector Analysis

In order to implement a real-time BSS system, it is necessary to extract the outputs before the next inputs come in. Thus, the learning process must be an adaptive algorithm. In an adaptive algorithm, the coefficients of the separation-filter matrices are updated at every frame [32].

3.1.1 Mathematical Model

Recall that the time domain convolutive model is given by:

$$\boldsymbol{x}[n] = \sum_{p=0}^{P-1} \boldsymbol{A}[p]\boldsymbol{s}[n-p] \tag{3.1}$$

where $\boldsymbol{x}[n]$ is the signal mixture, $\boldsymbol{s}[n]$ is the sources signal and the matrix $\boldsymbol{A}[p]$ represents the room response.

After receiving each portion of the mixture signal, we proceed to compute the STFT in the current frame:

$$\mathbf{x}_{TF}(f,n) = \sum_{t=0}^{T-1} \gamma[t]\mathbf{x}[t-nR]e^{-j\frac{2\pi ft}{F}} \tag{3.2}$$

thus, the convolutive model becomes an instantaneous one at each frequency bin f:

$$\mathbf{x}_{TF}(f,n) = \mathbf{A}(f,n)\boldsymbol{s}_{TF}(f,n) \tag{3.3}$$

3.1.2 Adaptive Whitening

Since the separation is online, we cannot apply (PCA) since we would need the whole data overtime which we don't have access to in real time. Therefore, one needs to perform whitening in an adaptive manner, hence the need to compute both the mean vector $\boldsymbol{\mu}$ and the covariance matrix. Thus, we propose an adaptive whitening which is our main contribution.

3.1.2.1 Mean vector

The mean vector at frame n can be estimated by:

$$\boldsymbol{\mu}(f,n) = \frac{1}{n}\sum_{n'=0}^{n-1} \mathbf{x}_{TF}(f,n') \tag{3.4}$$

one can easily obtain the following recursive formula:

$$\boldsymbol{\mu}(f,n) = \frac{n-1}{n}\boldsymbol{\mu}(f,n-1) + \frac{1}{n}\mathbf{x}_{TF}(f,n) \tag{3.5}$$

as n grows the value of the mean won't get updated as the second term goes to 0. In order to avoid that we opt for an exponential window:

$$\boldsymbol{\mu}(f,n) = \alpha\boldsymbol{\mu}(f,n) + (1-\alpha)\mathbf{x}_{TF}(f,n) \tag{3.6}$$

Where $0 < \alpha < 1$ is a smoothing factor. Then we subtract the mean from the data:

$$\mathbf{x}_{TF}(f,n) := \mathbf{x}_{TF}(f,n) - \boldsymbol{\mu}(f,n) \tag{3.7}$$

3.1.3 Covariance matrix

In order to whiten the data, we need to compute the covariance matrix $\mathbf{R_{xx}}(f,n)$ which has as expression the following:

$$\mathbf{R_{xx}}(f,n) = \frac{1}{n}\sum_{n'=0}^{n-1} \mathbf{x}_{TF}(f,n')\mathbf{x}_{TF}(f,n')^H \tag{3.8}$$

which can be computed using the recursive formula:

$$\mathbf{R_{xx}}(f,n) = \frac{n-1}{n}\mathbf{R_{xx}}(f,n-1) + +\frac{1}{n}\mathbf{x}_{TF}(f,t)\mathbf{x}_{TF}(f,n)^H \tag{3.9}$$

A good choice would be to use an exponential window:

$$\mathbf{R_{xx}}(f,n) = \alpha\mathbf{R_{xx}}(f,n-1) + (1-\alpha)\mathbf{x}_{TF}(f,n)\mathbf{x}_{TF}(f,n)^H \tag{3.10}$$

Using EVD we obtain:

$$\Big[\mathbf{D}(f,n), \mathbf{E}(f,n)\Big] = Eig(\mathbf{R_{xx}}(f,n)) \tag{3.11}$$

where $\mathbf{D}(f,n)$ is the diagonal matrix which contains the L largest eigenvalues of $\mathbf{R_{xx}}(f,n)$ and $\mathbf{E}(f,n)$ is the matrix containing the corresponding eigenvectors.

The whitening matrix is $\mathbf{Q}(f,n)$ at each new time frame n is:

$$\mathbf{Q}(f,n) = \mathbf{D}(f,n)^{-\frac{1}{2}}\mathbf{E}(f,n)^H \tag{3.12}$$

Then, the whitening data $\mathbf{x}_{TF_p}(f,n)$ will be:

$$\mathbf{x}_{TF}(f,n) := \mathbf{Q}(f,n)\mathbf{x}_{TF}(f,n) \tag{3.13}$$

3.1.4 Updating the separation filter

Each output signal will be computed in real-time using the following equation:

$$\mathbf{y}_{TF}(f,n) = \mathbf{W}(f,n)\mathbf{x}_{TF}(f,n) \tag{3.14}$$

Our purpose here, is to update the separating filter using an adaptive natural gradient descent.

To do so, we calculate the new filter at each frame with the learning rule:

$$\mathbf{W}(f,n) = \mathbf{W}(f,n-1) + \eta(f,n)\Delta\mathbf{W}(f,n) \tag{3.15}$$

where $\eta(f,n) = \dfrac{\eta}{\sqrt{\xi(f,n)}}$ is an adaptive learning rate.

The normalisation factor $\xi(f,n)$ is defined as:

$$\xi(f,n) = \beta\xi(f,n-1) + (1-\beta)\sum_{i=1}^{L} |x_i(f,n)|^2 / L \tag{3.16}$$

β is a forgetting factor. Introducing a normalisation factor not only enhances the algorithm's robustness but also increases its ability to withstand abrupt changes in input signal energy. This is achieved by dividing the input signal's sample Root Mean Square (RMS). To prevent the occurrence of a scenario where $\xi(f,n)^{-1}$ becomes infinite, a small constant ϵ is added to the term $\xi(f,n)$ to prevent division by zero.

$\Delta\mathbf{W}(f,n)$ denotes the natural gradient at the current frame which has for expression:

$$\Delta\mathbf{W}(f,n) = \Big(\mathbf{I} - \mathbb{E}[\boldsymbol{\varphi}^{(f)}(n)\mathbf{y}_{TF}(f,n)^H]\Big)\mathbf{W}(f,n) \tag{3.17}$$

where $\mathbf{I}$ is the identity matrix and $\boldsymbol{\varphi}(f,n)$ is the score function which is given by:

$$\boldsymbol{\varphi}^{(f)}(n) = \begin{pmatrix} \varphi_1^{(f)}(n) & \dots & \varphi_L^{(f)}(n) \end{pmatrix}^T \tag{3.18}$$

where:

$$\varphi_l^{(f)}(n) = \frac{\mathrm{y}_{TF_l}(f,n)}{\sqrt{\sum_{f=1}^{F} |\mathrm{y}_{TFl}(f,n)|^2}} \tag{3.19}$$

In order to estimate the expectation we need the whole ensemble of N time frames.

$$\mathfrak{R}(f) = \mathbb{E}[\boldsymbol{\varphi}^{(f)}\mathbf{y}_{TF}(f)^H] = \frac{1}{N}\sum_{n=0}^{N-1} \boldsymbol{\varphi}(f,n)\mathbf{y}_{TF}(f,n)^H \tag{3.20}$$

In online IVA this is not suitable, so we use an instantaneous stochastic gradient.

Here, the online version of the expectation becomes:

$$\mathfrak{R}(f,n) = \mathbb{E}\left[\boldsymbol{\varphi}^{(f)}(n)\mathbf{y}_{TF}(f,n)^H\right] \approx \boldsymbol{\varphi}^{(f)}(n)\mathbf{y}_{TF}(f,n)^H \tag{3.21}$$

And thus one obtains the following:

$$\mathbf{W}(f,n) = \mathbf{W}(f,n-1) + \eta(f,n)\left(\mathbf{I} - \boldsymbol{\varphi}^{(f)}(n)\mathbf{y}_{TF}(f,n)^H\right)\mathbf{W}(f,n) \tag{3.22}$$

3.1.4.1 Nonholonomic constraint

In many applications, such as speech signals, when a source signal becomes very small suddenly, the corresponding coefficients of separation filters tend to be large in the learning process to compensate for this changes and to emit the output signal larger. In particular, when one source signal becomes silent, the separation filters diverge. Therefore, we adopt a nonholonomic constraint[47] to avoid this phenomenon.

It was shown that the diagonal term in equation (3.22) can be set arbitrarily [48],[49],[50] . The problem has been analysed also based on the information geometry of semi-parametric statistical models ([51] [52] [53]). Therefore, the above algorithm can be generalised in a more flexible and universal form as:

$$\boldsymbol{W}(t+1) = \boldsymbol{W}(t) + \eta(t)\left[\boldsymbol{\Lambda}(t) - \boldsymbol{\varphi}(\boldsymbol{y}(t))\boldsymbol{y}^H(t)\right]\boldsymbol{W}(t),$$

where $\boldsymbol{\Lambda}(t)$ is any positive definite scaling diagonal matrix. By determining $\boldsymbol{\Lambda}$ correctly depending on $\boldsymbol{y}$, we have various algorithms. Amari et al. proposed in [47]:

$$\boldsymbol{\Lambda}(t) = \mathrm{diag}(\boldsymbol{\varphi}(\boldsymbol{y}(t))\boldsymbol{y}^H(t))$$

The new constraints lead to a new learning algorithm, written as:

$$\mathbf{W}(t+1) = \mathbf{W}(t) + \eta(t)\mathbf{F}[\mathbf{y}(t)]\mathbf{W}(t) \tag{3.23}$$

where all the elements on the main diagonal of the $L \times L$ matrix $\mathbf{F}(\mathbf{y})$ are put equal to zero, that is:

$$f_{ii} = 0 \tag{3.24}$$

and:

$$f_{ij} = \varphi_i y_l \quad i \neq j \tag{3.25}$$

The learning rule in equation 3.23, is called the orthogonal nonholonomic natural gradient descent algorithm. In consequence, we obtain the following gradient with the constraint by simply replacing the identity matrix $\mathbf{I}$ in (3.17) with $\mathbf{\Lambda}(f,n)$.

Where:

$$\mathbf{\Lambda}(f,n) = \text{diag}(\mathfrak{R}(f,n)) \tag{3.26}$$

Then, the natural gradient becomes:

$$\Delta\mathbf{W}(f,n) = (\mathbf{\Lambda}(f,n) - \mathfrak{R}(f,n))\mathbf{W}(f,n) \tag{3.27}$$

3.1.5 Rescaling

The last step in the separation process is to adjust the scaling of the sources and that is using the MDP principle discussed previously.

$$\mathbf{W}_s(f,n) = \text{diag}(\mathbf{W}'(f,n)^{\#})\mathbf{W}'(f,n) \tag{3.28}$$

Where $\mathbf{W}'(f,n) = \mathbf{W}(f,n)\mathbf{Q}(f,n)$ denotes the separation filter projected back into the original dimension ($L \times M$).

3.1.6 Signal reconstruction

Once the output signals are estimated in frequency domain, we convert back to time domain by calculating the inverse short time Fourier transform:

$$\boldsymbol{y}[n] = \hat{\boldsymbol{s}}[n] = \text{ISTFT}(\mathbf{W}_s(f,n)\mathbf{x}_{TF}(f,n)) \tag{3.29}$$

3.1.7 Algorithm summary

Herein, we present a concise pseudo-code (algorithm summary) for the adaptive natural gradient based IVA (NG IVA) with whitening process. The pseudo code captures and summarises every equation and step given at the beginning of this chapter.

Algorithme 2 : Adaptive IVA

input : T samples of observed mixtures $x_1[t], ..., x_M[t]$ $t = 1 \dots T$, number of sources L, NFFT,window, learning rate η, forgetting factors β, α

output : T samples of estimated sources $y_1[t], ..., \mathbf{y}_L[t]$ in time domain $t = 1 \dots T$

Initialize $\mathbf{R}_{\mathbf{x}}(f, 0)$ with I_M for all f

Initialize $\mathfrak{R}(f, 0)$ and $\mathbf{W}(f, 0)$ with I_L for all f

Calculate the M-channel forward FFT; $\mathbf{x}_{TF}(f, n) = FFT(\boldsymbol{x}_{\{1,...,M\}})$

Whitening

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

  compute the mean recursively: $\boldsymbol{\mu}(f, n) = \alpha\boldsymbol{\mu}(f, n-1) + (1-\alpha)\mathbf{x}(f, n)$

  $\mathbf{x}_{TF}(f, n) = \mathbf{x}_{TF}(f, n) - \boldsymbol{\mu}(f, n)$

  compute covariance matrix recursively
  $\mathbf{R}_{\mathbf{x}}(f, n) = \alpha\mathbf{R}_{\mathbf{x}}(f, n-1) + (1-\alpha)\mathbf{x}_{TF}(f, n)\mathbf{x}_{TF}^H(f, n)$

  $\Big[\mathbf{D}(f, n), \mathbf{E}(f, n)\Big] = Eig(\mathbf{R}_{\mathbf{x}}(f, n))$ `// Eigenvalue decomposition`

  $\mathbf{x}_{TF}(f, n) = \mathbf{D}(f, n)^{-\frac{1}{2}}\mathbf{E}(f, n)^H\mathbf{x}_{TF}(f, n)$ `// dimension reduction`

  $\mathbf{y}_{TF}(f, n) = \mathbf{W}(f, n)\mathbf{x}_{TF}(f, n)$

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

Perform separation

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

  compute the sources priors $\varphi_l^{(f)}(n) = \frac{y_l(f,n)}{\sqrt{\sum_{f=1}^{F} |y_l(f,n)|^2}}$

  Concatenate $\boldsymbol{\varphi}(f, n) = \Big(\varphi_1(f, n) \quad \dots \quad \varphi_L(f, n)\Big)^T$ `// score function`

  $\mathfrak{R}(f, n) = \boldsymbol{\varphi}(f, n)\mathbf{y}(f, n)^H$ `// Correlation matrix`

  $\Delta\mathbf{W}(f, n) = \Big(diag(\mathfrak{R}(f, n)) - \mathfrak{R}(f, n)\Big)\mathbf{W}^{(f)}[n]$ `// natural gradient`

  $\xi(f, n) = \beta\xi(f, n-1) + (1-\beta)\sum_{i=1}^{L} |x_i(f, n)|^2 / L$ `// normalization`

  $\mathbf{W}(f, n) = \mathbf{W}(f, n) + \frac{\eta}{\sqrt{\xi(f, n) + \epsilon}}\Delta\mathbf{W}(f, n-1)$ `// separation matrix`

Rescaling *(for all f)*

$\mathbf{W}(f, n) = \mathbf{W}(f, n)\mathbf{D}(f, n)^{-\frac{1}{2}}\mathbf{E}(f, n)^H$ `// Back to original dimension`

$\mathbf{W}(f, n) = diag(\mathbf{W}(f, n)^{\#})\mathbf{W}(f, n)$ `// Apply MDP to correct scaling`

$\mathbf{y}_{TF}(f, n) = \mathbf{W}(f, n)\mathbf{x}_{TF}(f, n)$ `// Estimate output signal`

Compute ISTFT *(for all l)*

$n = n + 1$ `// increment time block (n)`

$y_l[t] = IFFT(y_{TF_l}(f, n)) \quad t = 1 \dots T$ `// T samples of estimate sources`

3.2 Experimental results

3.2.1 Softwares

3.2.1.1 Data generation Python

Python is an interactive, high-level, object-oriented scripting language that is widely used for various applications, including mathematical computation and audio processing. It offers extensive support and a diverse range of libraries, making it well-suited for tasks related to Blind Source Separation (BSS). In our project, we utilized Python 3.9 to generate synthetic speech mixtures. The Library used is outlined down below.

Pyroomacoustics [54] is audio simulation software that includes a quick Room Impulse Response (RIR) generator and reference implementations of common algorithms like beamforming, Direction Of Arrival (DOA) estimation, and adaptive filtering. We utilized Pyroomacoustics' RIR generator in our code to generate synthetic mixtures.

3.2.1.2 Reverberation time RT60

The reverberation time (RT) refers to the duration it takes for the energy of an impulse response to decrease below a specific threshold, typically expressed in decibels. A commonly used threshold is -60 dB, denoted as RT60. In this report, the calculation of reverberation time is performed using the Schroeder integral method [55]. This method defines a continuous decay curve (E) to quantify the decay characteristics.

$$E(t) = \int_t^\infty a^2(t')dt' \tag{3.30}$$

Another discrete normalised version is

$$E[n] = \frac{\sum_{n'=n}^{\infty} a^2[n']}{\sum_{n'=0}^{\infty} a^2[n']}$$

To estimate a line that intersects the horizontal axis at -60dB, linear regression analysis can be employed. A MATLAB implementation of this approach can be found in [56]. For more comprehensive information on measuring reverberation time and analysing decay curves, refer to [57].

3.2.1.3 MATLAB

MATLAB, short for Matrix Laboratory, is a programming platform extensively used in engineering applications like signal processing and data science. It features a high-performance matrix programming language developed by MathWorks, enabling efficient data analysis, algorithm optimization, and model design. MATLAB's toolboxes, including the BSS Eval Toolbox, enhance its functionality, with the latter being distributed online under the GNU Public License. We used an HP core i5-2035G4 with a CPU 1.5 GHz.

3.2.2 Performance parameters

The Signal-to-Distortion-Ratio (SDR), Signal-to-Interference-Ratio (SIR) and Signal-to-Artefact-Ratio (SAR) defined in [58] are used throughout this thesis to evaluate the separation performances.

Based on the following model, the decomposition of an estimated signal is:

$$\hat{s}_l(t) = s_{target}(t) + e_{interf}(t) + e_{artif}(t) \tag{3.31}$$

where $s_{target} = \mathcal{F}(s_l(t))$ represents the original signal modified by an allowed distortion $\mathcal{F}$, e_{inter} is an allowed deformation of the sources which accounts for the interferences of the unwanted sources whereas e_{artif} is an "artifact" term that may correspond to artifacts of the separation algorithm such as musical noise, etc. or simply to deformations induced by the separation algorithm that are not allowed [59].

SDR is usually considered to be an overall measure of how good a source sounds. It measures how well the desired source has been extracted while suppressing unwanted components. It is computed as follows:

$$\text{SDR} = 10\log_{10}\frac{||s_{target}||^2}{||e_{inter} + e_{artif}||^2} \tag{3.32}$$

SIR is usually interpreted as the amount of other sources that can be heard in a source estimate. This is most close to the concept of "bleed", or "leakage". It quantifies the ability of the separation algorithm to suppress unwanted sources and isolate the desired source. SIR is computed as follows:

$$\text{SIR} = 10\log_{10}\frac{||s_{target}||^2}{||e_{inter}||^2} \tag{3.33}$$

SAR is usually interpreted as the amount of unwanted artifacts a source estimate has with relation to the true source. The formula for SAR is:

$$\mathrm{SAR} = 10 \log_{10} \frac{||s_{target}||^2}{||e_{artif}||^2} \tag{3.34}$$

these measures provide a quantitative assessment of the separation quality, allowing for comparisons and evaluations of different BSS algorithms. While they don't directly correspond to physical phenomena, they serve as useful indicators of the quality of the separated sources based on their power characteristics. Of course, the higher the values are the better are the performances.

$||a||^2 = \langle a, a \rangle$ is the energy of a and $\langle a, b \rangle = \sum_{t=0}^{T-1} a[t]b^*[t]$ denotes the inner product between two complex valued signals of length T.

The terms in the right hand side of equation (3.29) are estimated using the decomposition method. The decomposition is based on orthogonal projections. Let $\Pi\{y_1, \ldots, y_k\}$ denote the orthogonal projector onto the subspace spanned by the vectors $y_1, \ldots, y_k$. The projector is a $T \times T$ matrix, where T is the length of these vectors. We consider the two orthogonal projectors:

$$P_{s_l} := \Pi\{s_l\} \tag{3.35}$$

$$P_{\mathbf{S}} := \Pi\left\{(s_{l'})_{1 \le l' \le L}\right\} \tag{3.36}$$

and we decompose $\hat{s}_l$ as the sum of the three terms:

$$s_{\text{target}} := P_{s_l} \widehat{s}_l \tag{3.37}$$

$$e_{\text{interf}} := P_{\mathbf{s}} \widehat{s}_l - P_{s_l} \widehat{s}_l \tag{3.38}$$

$$e_{\text{artif}} : = \widehat{s}_l - s_{target} - e_{artif} \tag{3.39}$$

if the sources are mutually independent, then the last step is straightforward and the terms are computed using inner products are projectors as follows:

$$s_{\text{target}} = \frac{\langle \widehat{s}_l, s_l \rangle}{||s_l||^2} s_l \tag{3.40}$$

$$e_{\text{interf}} = \sum_{l' \neq l} \frac{\langle \widehat{s}_l, s_{l'} \rangle}{||s'_l||^2} s_{l'} \tag{3.41}$$

3.2.2.1 Experimental setup

In our simulation, we generated Room Impulse Responses (RIRs) using Pyroomacoustics with an RT60 of 150ms for a room measuring 5.5 × 3.5 × 2.75 meters. A circular microphone array consisting of one central microphone and six equidistantly spaced microphones was used.

We conducted two mixing scenarios: a two-source scenario and a three-source scenario. The two-source scenario involved different combinations of male (M) and female (F) speakers, while the three-source scenario included two females with one male and two males with one female.

For the two-source scenario, the speech signals had a duration of 27 seconds. We evaluated the performance of the classic online Natural Gradient-based Independent Vector Analysis (online NG IVA) algorithm and the proposed online NG IVA algorithm with whitening. We measured SIR, SDR and SAR values at one-second intervals using BSS eval toolbox [58]. Interpolated graphs were generated to visualize the temporal evolution of SIR, SAR, and SDR, and their average values after convergence (around 7 seconds and 4 seconds respectively) were represented using bar diagrams.

To accommodate the classic online NG IVA algorithm without whitening, we used two symmetrically placed microphones and deactivated the remaining ones. The learning rate (η) was initially set to the largest value before divergence, and then reduced by 50% after each additional 25% of the total number of frames (N). The signals were sampled at a rate of 16000kHz, and a 256-point FFT with a 75% overlap (shift size of 64 samples) was employed. The forgetting factors (β and α) were set to 0.5 and 0.985, respectively.

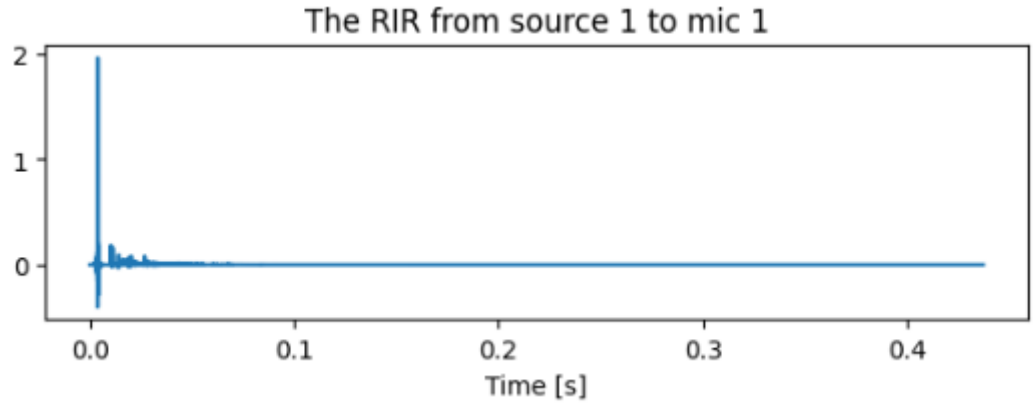


Figure 3.1: Room Impulse Response from source 1 to mic 1.

Figure 3.1 shows the generated RIR from source 1 to microphone 1.

The experiment setup for two sources scenario are resumed in the table 3.1 down below:

Reverberation time	150 ms
Room dimensions	5.5m $\times$ 3.5m $\times$ 2.75 m
Positions of microphones	[2.75, 1.75, 1.4], [2.795, 1.75, 1.4] [2.705, 1.75, 1.4] , [2.773, 1.788, 1.4],[2.773, 1.711, 1.4], [2.72, 1.78, 1.4], [2.72, 1.71, 1.4]
Sources positions	[2.9, 2.183, 1.5], [4.2, 2, 1.5]
Signal duration	27 s
Sampling rate	16 000 Hz
Learning rate η	IVA with whitening 3 then 1.5 then 0.75 then 0.375 IVA without whitening 2.5 then 1.25 then 0.625 then 0.3125
NFFT	256 samples
Window	2 Hanning(t)/NFFT
Shift size	64 samples
Forgetting factors	$\beta = 0.5$, $\alpha = 0.985$
Initalisations	$\mathbf{W}(f,0) = \mathbf{I}_L$, $\mathbf{R}_{\mathbf{x}}(f,0) = \mathbf{I}_M$, $\mathfrak{R}(f,0) = \mathbf{I}_L$ and $\boldsymbol{\mu}(f,0) = \mathbf{0}$ for all frequencies F

Table 3.1: Experiment parameters

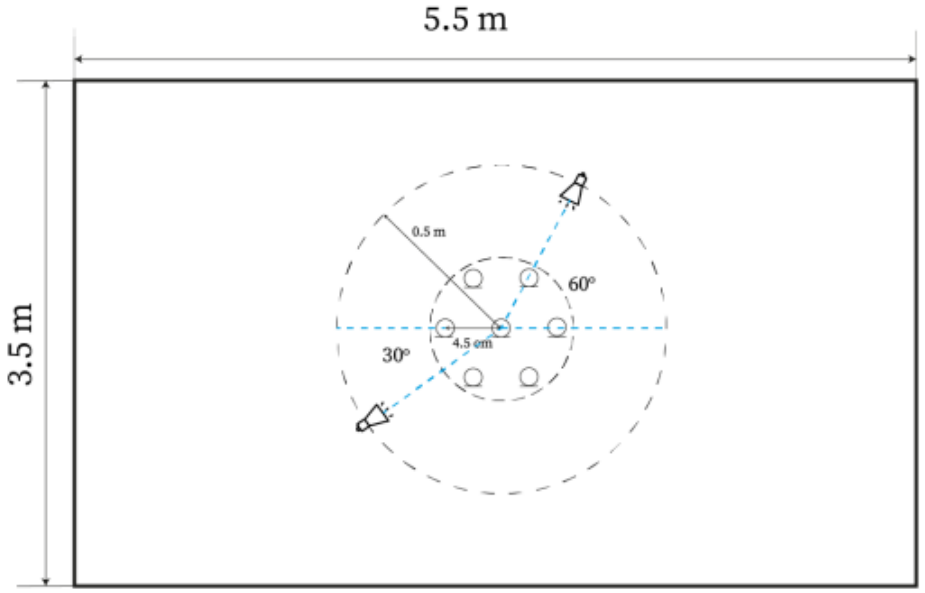


Figure 3.2: Two sources configuration.

Figure 3.2 illustrates the microphones and sources positions for two sources scenario.

The experiment setup for three sources scenario are resumed in the table 3.2 down below.

Reverberation time	150 ms
Room dimensions	5.5m $\times$ 3.5m $\times$ 2.75 m
Positions of microphones	[2.75, 1.75, 1.4], [2.799, 1.75, 1.4], [2.5, 1.79, 1.4], [2.81, 1.8, 1.4] , [2.69, 1.711, 1.4], [3.1, 1.58, 1.4],[3.2, 1.78, 1.4]
Sources positions	[3.21, 1.921, 1.5], [2.36, 2.07, 1.5], [2.66, 1.25, 1.5]
Signal duration	27 s
Sampling rate	16 000 Hz
Learning rate η	IVA with whitening 2.5 then 1.25 then 0.625 then 0.3125 IVA without whitening 2 then 1 then 0.5 then 0.25
NFFT	256 samples
Window	2 Hanning(t)/NFFT
Shift size	64 samples
Forgetting factors	$\beta = 0.5$, $\alpha = 0.985$
Initalisations	$\mathbf{W}(f,0) = \mathbf{I}_L$, $\mathbf{R}_{\mathbf{x}}(f,0) = \mathbf{I}_M$, $\mathfrak{R}(f,0) = \mathbf{I}_L$ and $\boldsymbol{\mu}(f,0) = \mathbf{0}$ for all frequencies F

Table 3.2: Experiment parameters for three sources scenario

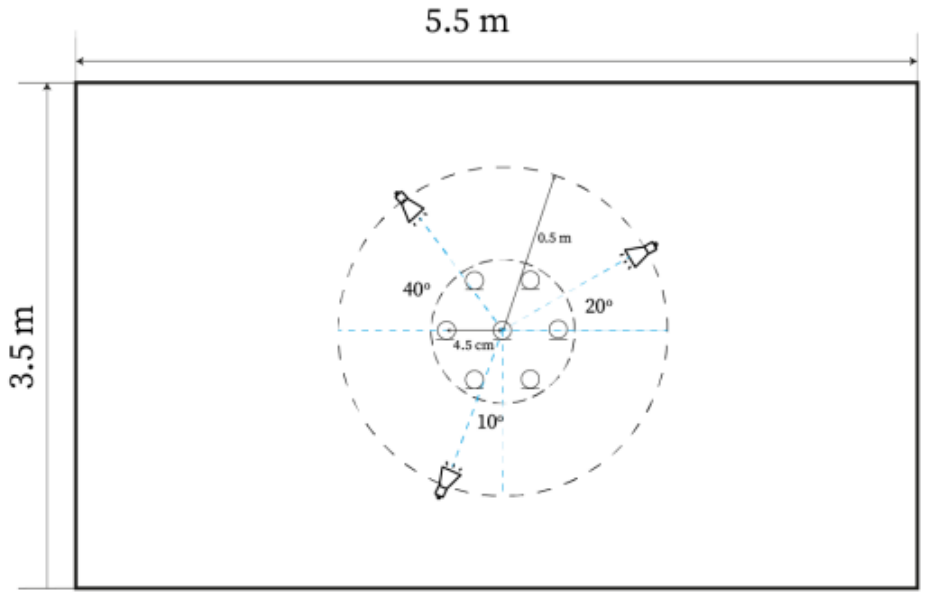


Figure 3.3: Three sources configuration.

Figure 3.3 illustrates the microphones and sources positions for three sources scenario.

3.2.3 Results

3.2.3.1 Two sources scenario

Two females case: Here below, we present the performances in the case of two female speakers.

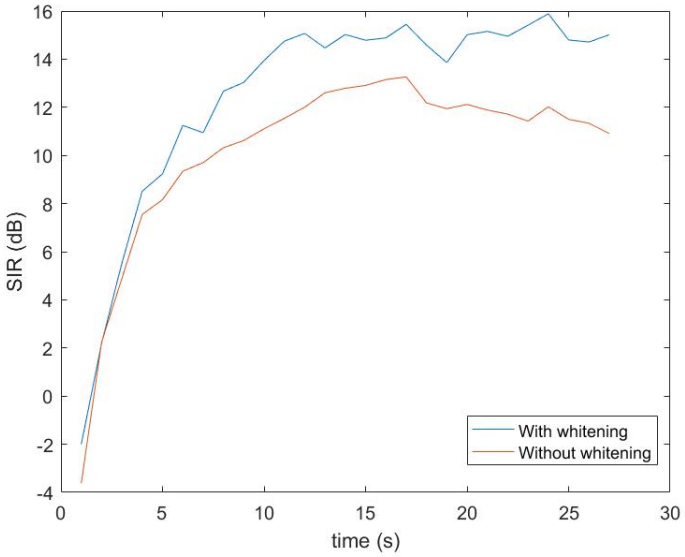


Figure 3.4: SIR of source 1 (dB) evolution in time (s) 2F.

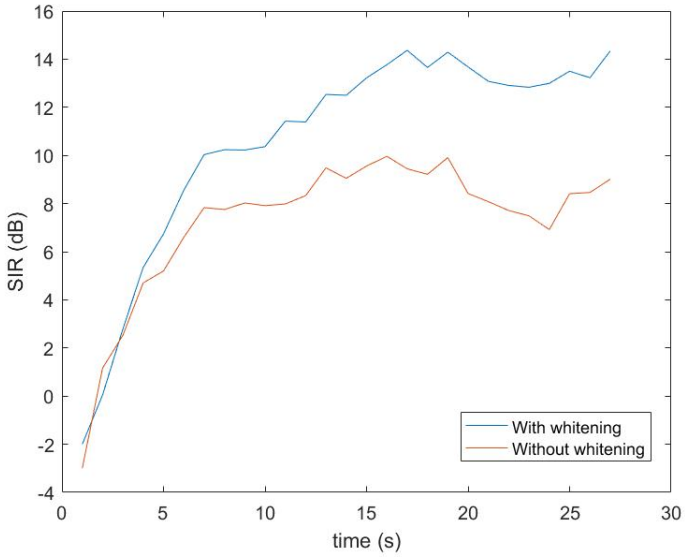


Figure 3.5: SIR of source 2 (dB) evolution in time (s) 2F.

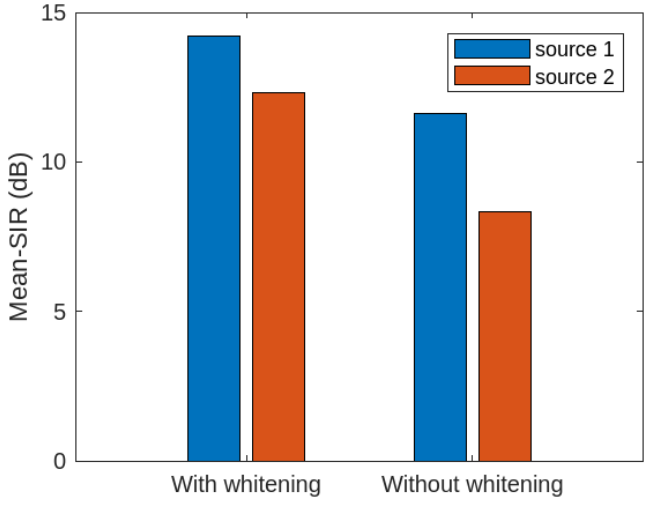


Figure 3.6: Mean values of SIR in the case of two female speakers.

Figures 3.4, 3.5 and 3.6 show SIR values (both over time and mean values in bars with and without whitening respectively) for two female speakers case, show that whitening increases SIR of the estimated signals. Both algorithms converge over time to approximately 16 dB and 14 dB for source 1 and to 14 dB and 8 dB for source 2. However, there are some fluctuations because the learning rule is stochastic.

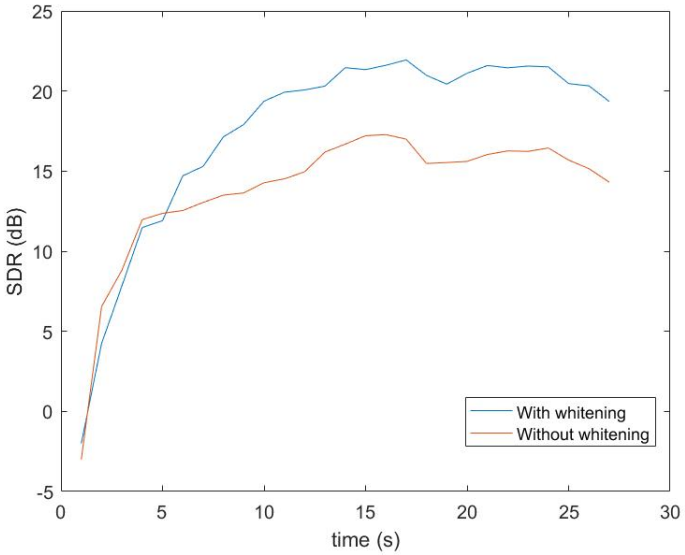


Figure 3.7: SDR of source 1 (dB) evolution in time (s) 2F.

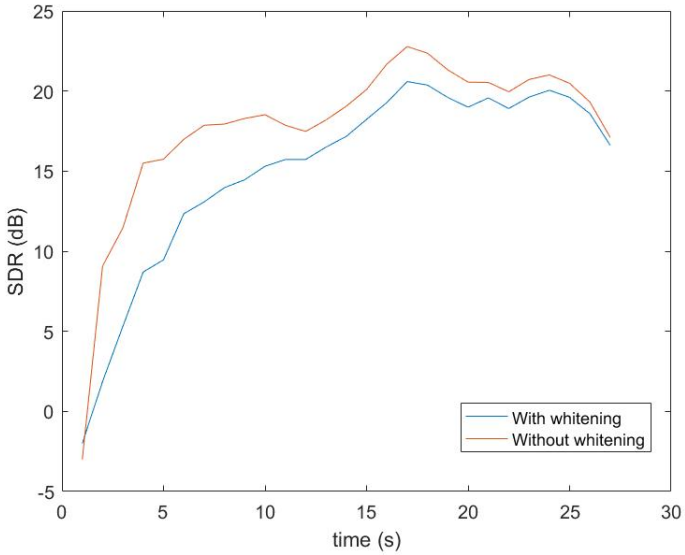


Figure 3.8: SDR of source 2 (dB) evolution in time (s) 2F.

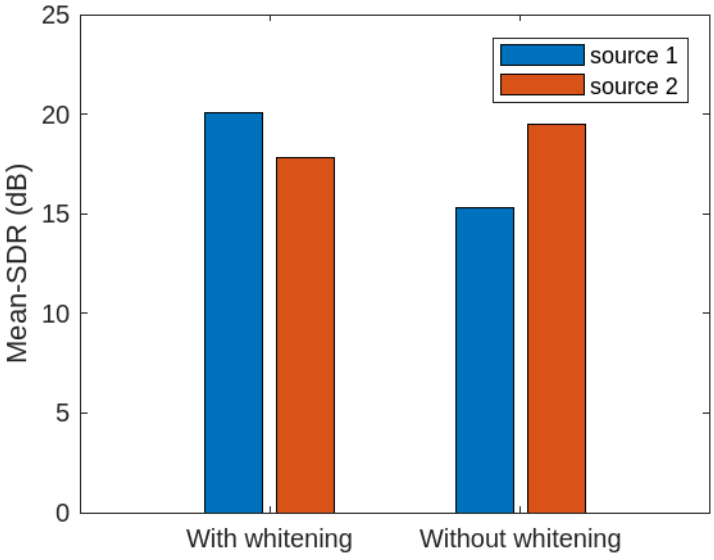


Figure 3.9: Mean values of SDR in the case of two female speakers

Figures 3.7, 3.8 and 3.9 show SDR values (both over time and mean values in bars with and without whitening respectively) for two female speakers case, show that whitening increases SIR of the estimated source 1 while classic NG IVA outperforms slightly NG IVA with whitening. Both algorithms converge over time to approximately 21 dB and 15 dB for source 1 and to 20 dB and 19 dB for source 2. However, there are some fluctuations because the learning rule is stochastic.

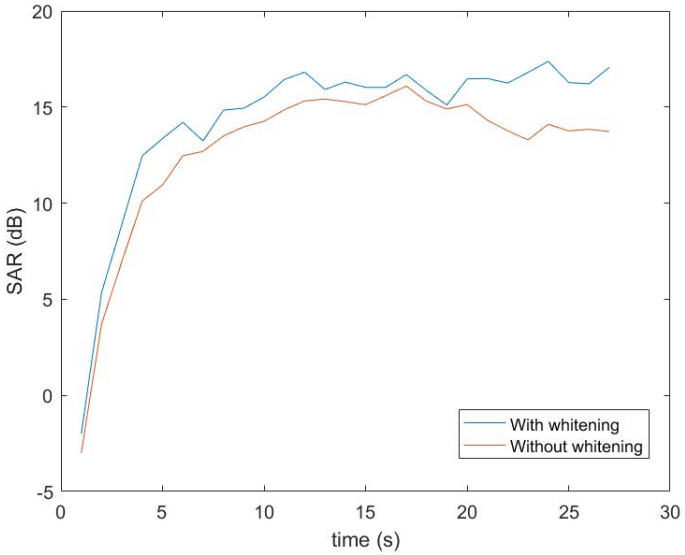


Figure 3.10: SAR of source 1 (dB) evolution in time (s) 2F.

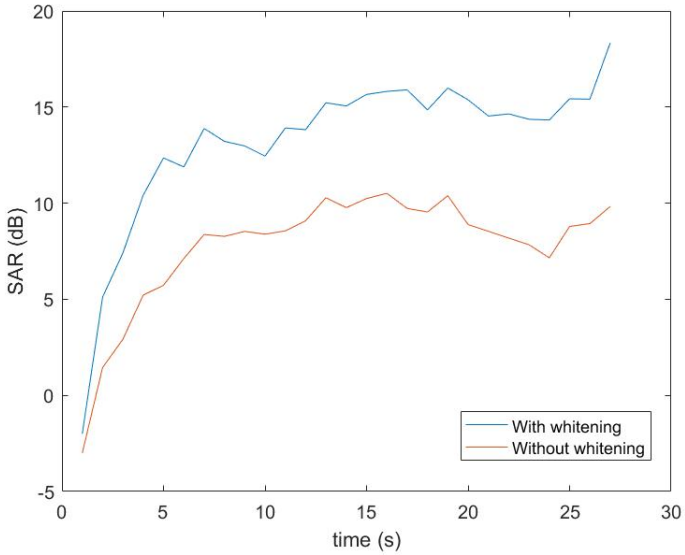


Figure 3.11: SAR of source 2 (dB) evolution in time (s) 2F.

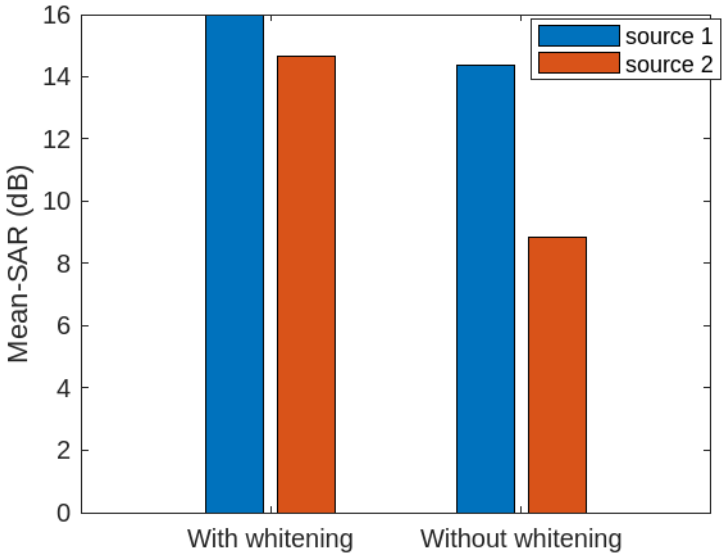


Figure 3.12: Mean values of SAR in the case of two female speakers.

Figures 3.10, 3.11 and 3.12 show SAR values (both over time and mean values in bars with and without whitening respectively) for two female speakers case, show that whitening increases SAR of the estimated sources (although just slightly for the second source). Both algorithms converge over time to approximately 16 dB and 15 dB for source 1 and to 15 dB and 10 dB for source 2. However, there are some fluctuations because the learning rule is stochastic.

Two males case: Here below,, we present the performances in the case of two male speakers.

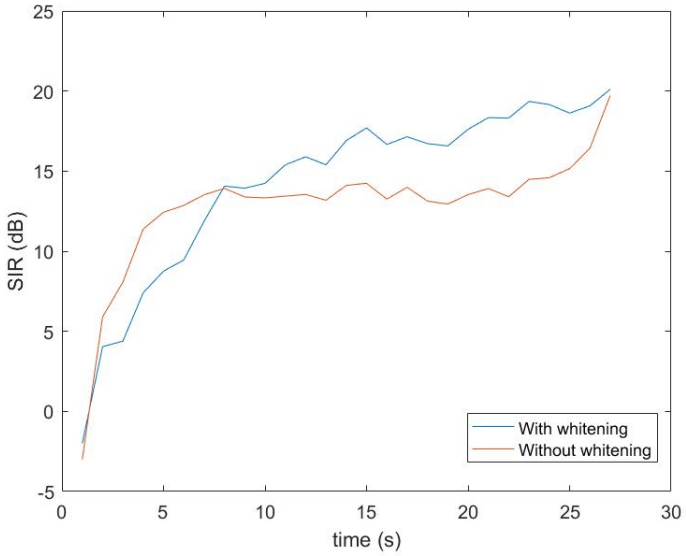


Figure 3.13: SIR of source 1 (dB) evolution in time (s) 2M.

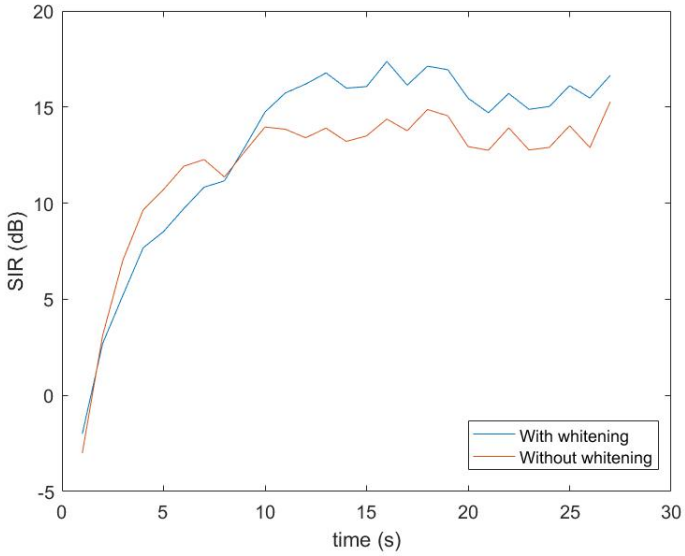


Figure 3.14: SDR of source 2 (dB) evolution in time (s) 2M.

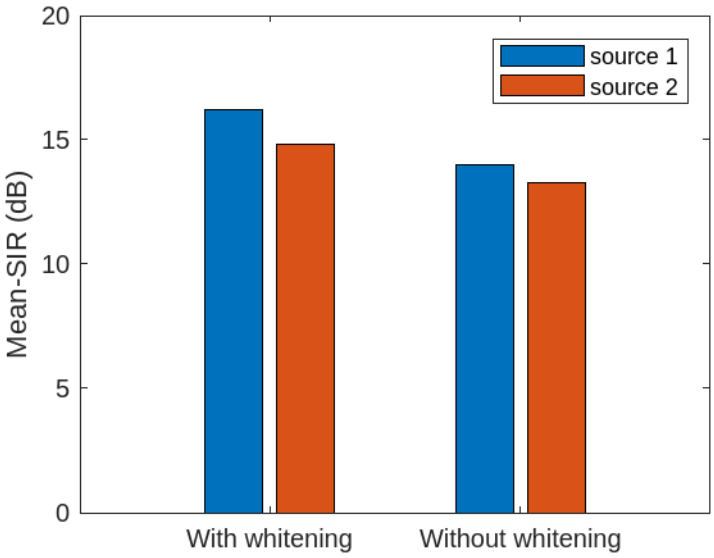


Figure 3.15: Mean values of SIR in the case of two male speakers.

Figures 3.13, 3.14 and 3.15 show SIR values (both over time and mean values in bars with and without whitening respectively) for two male speakers case, show that whitening increases SIR of the estimated sources (although just slightly for the second source). Both algorithms converge over time to approximately 18 dB and 15 dB for source 1 and to 16 dB and 15 dB for source 2. However, there are some fluctuations because the learning rule is stochastic.

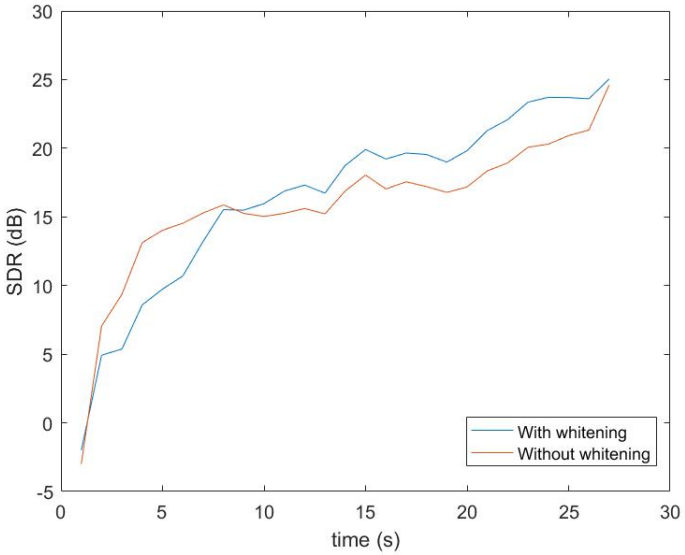


Figure 3.16: SDR of source 1 (dB) evolution in time (s) 2M.

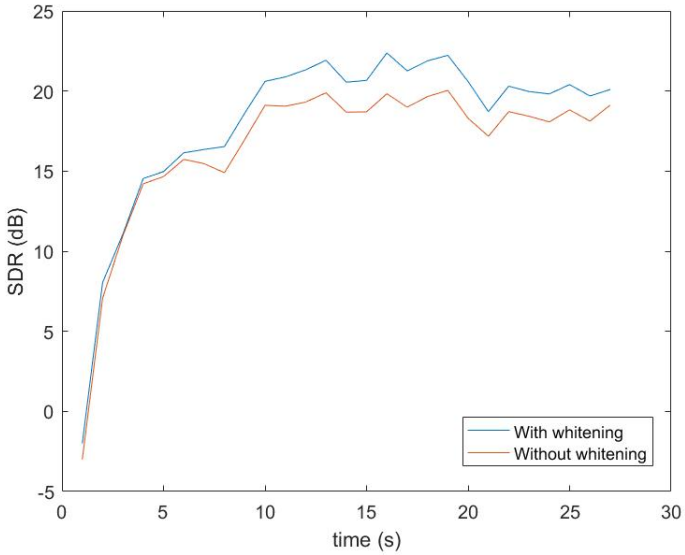


Figure 3.17: SDR of source 2 (dB) evolution in time (s) 2M.

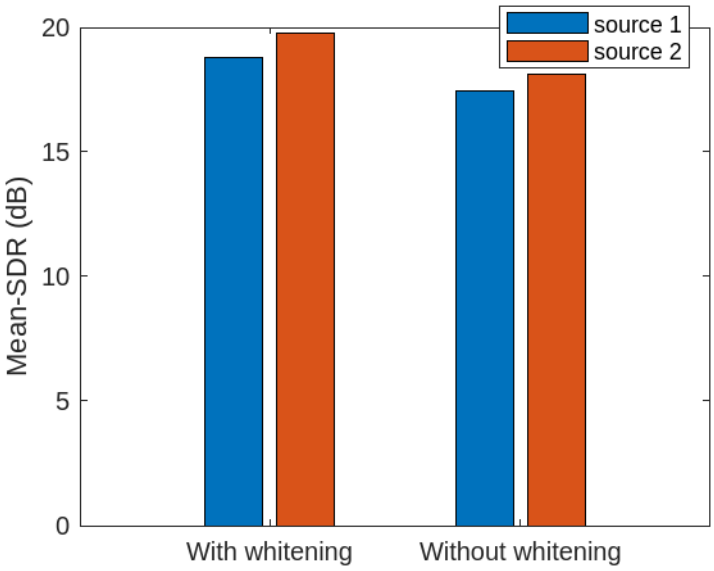


Figure 3.18: Mean values of SDR in the case of two male speakers.

Figures 3.16, 3.17 and 3.18 show SDR values (both over time and mean values in bars with and without whitening respectively) for two male speakers case, show that whitening increases SDR of the estimated sources (although just slightly). Both algorithms converge over time to approximately 20 dB and 19 dB for source 1 and to 21 dB and 19 dB for source 2. However, there are some fluctuations because the learning rule is stochastic.

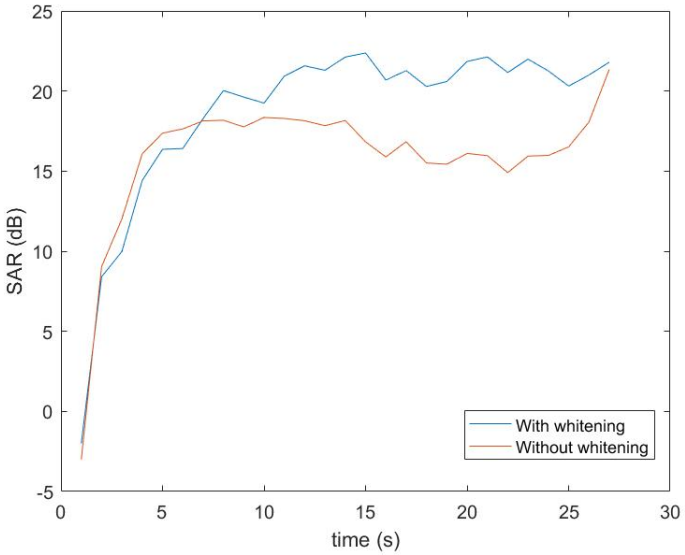


Figure 3.19: SAR of source 1 (dB) evolution in time (s) 2M.

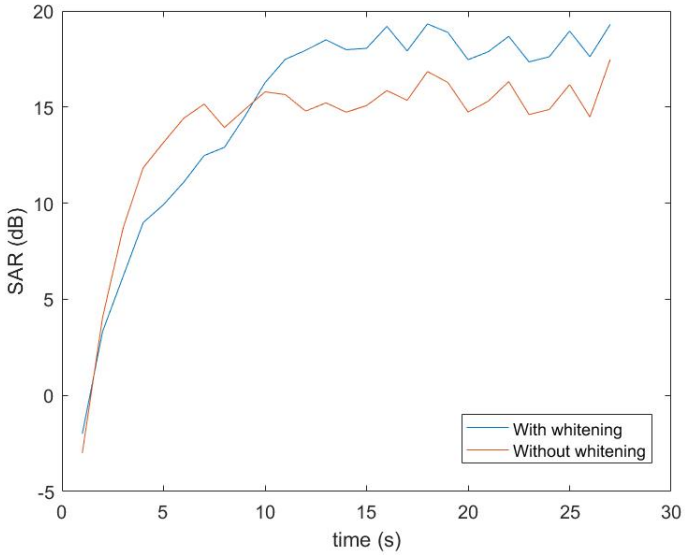


Figure 3.20: SAR of source 2 (dB) evolution in time (s) 2M.

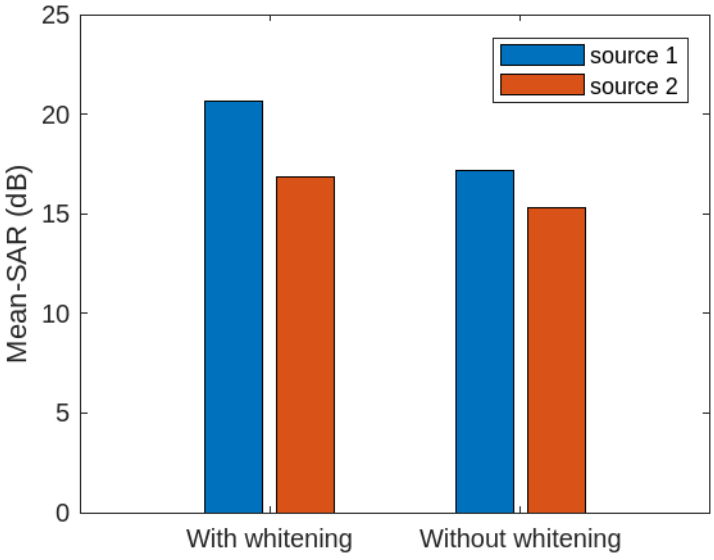


Figure 3.21: Mean values of SAR in the case of two male speakers.

Figures 3.19, 3.20 and 3.21 show SAR values (both over time and mean values in bars with and without whitening respectively) for two male speakers case, show that whitening increases SAR of the estimated sources. Both algorithms converge over time to approximately 21 dB and 15 dB for source 1 and to 20 dB and 15 dB for source 2. However, there are some fluctuations because the learning rule is stochastic.

One male and one female case: Here below, we present the performances in the case of one male and one female speakers.

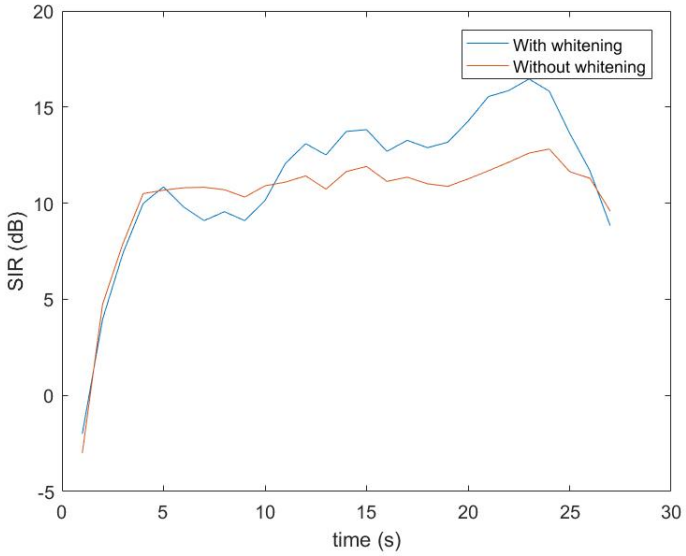


Figure 3.22: SIR of source 1 (dB) evolution in time (s) 1M+1F.

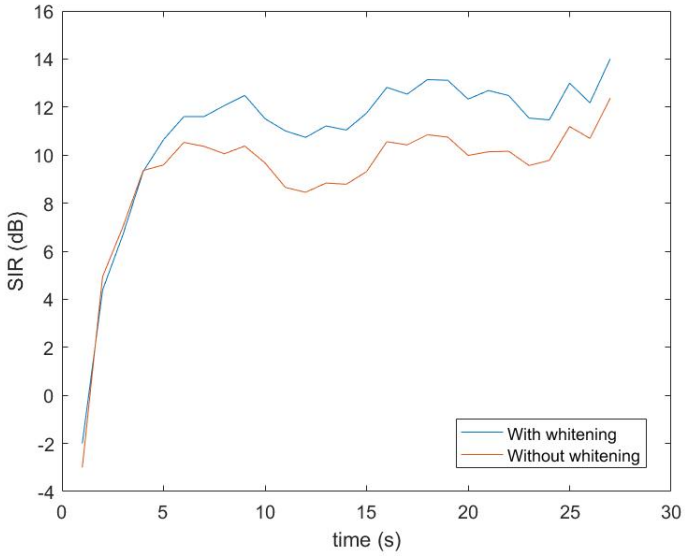


Figure 3.23: SIR of source 2 (dB) evolution in time (s) 1M+1F.

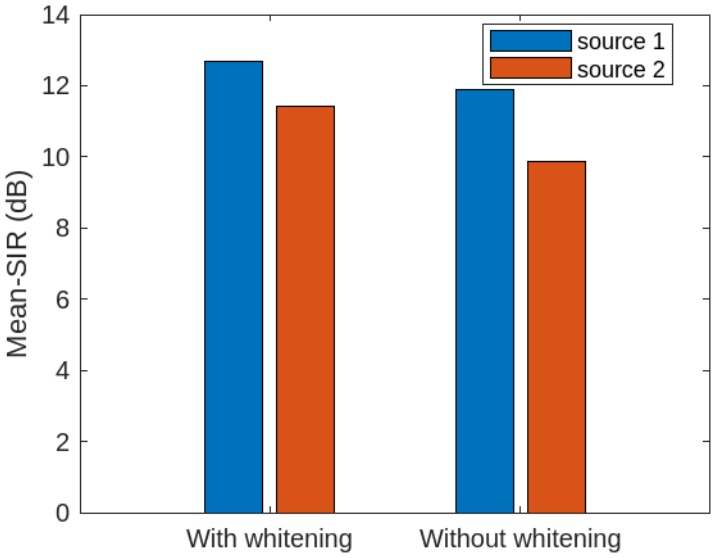


Figure 3.24: Mean values of SIR in the case of two male speakers.

Figures 3.22, 3.23 and 3.24 show SIR values (both over time and mean values in bars with and without whitening respectively) for one male and one female speakers case, show that whitening increases SAR of the estimated sources. Both algorithms converge over time to approximately 16 and 11 dB for source 1 and to 13 and 10 dB for source 2. However, there are some fluctuations because the learning rule is stochastic.

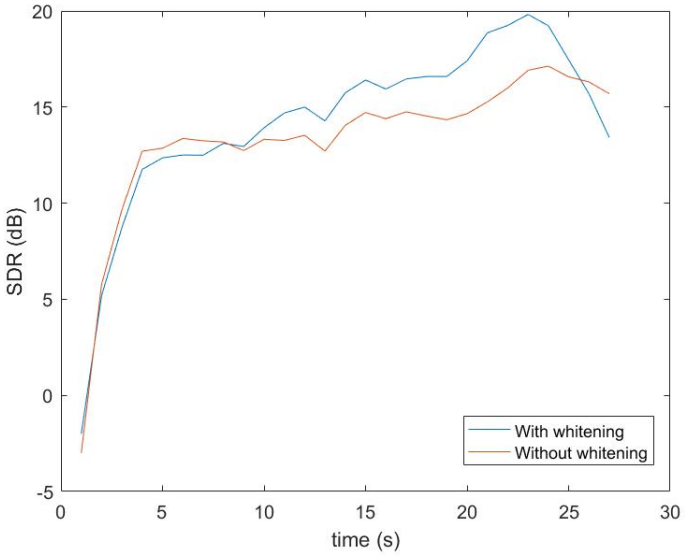


Figure 3.25: SDR of source 1 (dB) evolution in time (s) 1M+1F.

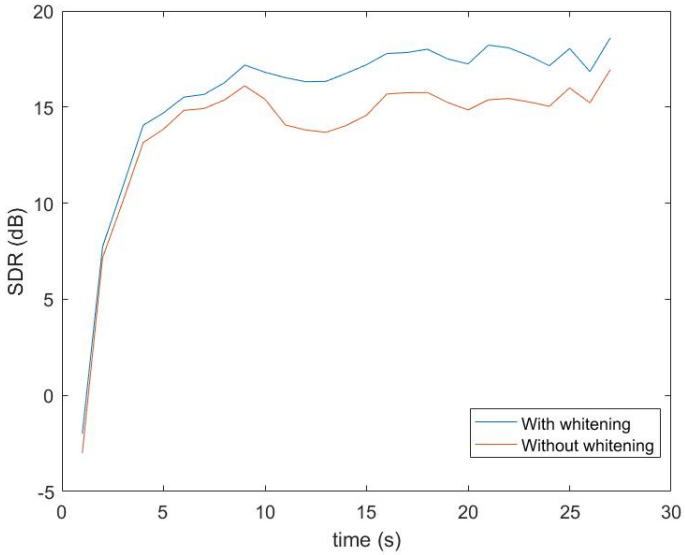


Figure 3.26: SDR of source 2 (dB) evolution in time (s) 1M+1F.

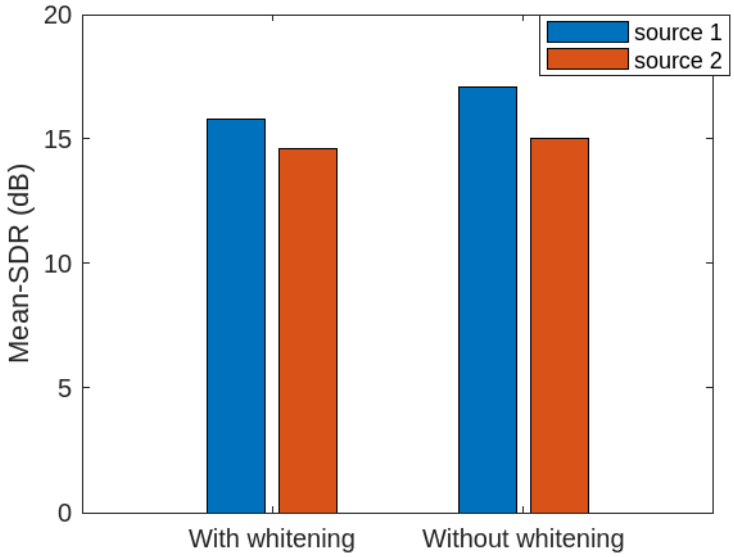


Figure 3.27: Mean values of SDR in the case of two male speakers.

Figures 3.25, 3.26 and 3.27 show SDR values (both over time and mean values in bars with and without whitening respectively) for one male and one female speakers case, show that whitening increases SAR of the estimated sources. Both algorithms converge over time to approximately 20 and 15 dB for source 1 and to 17 and 15 dB for source 2. However, there are some fluctuations because the learning rule is stochastic.

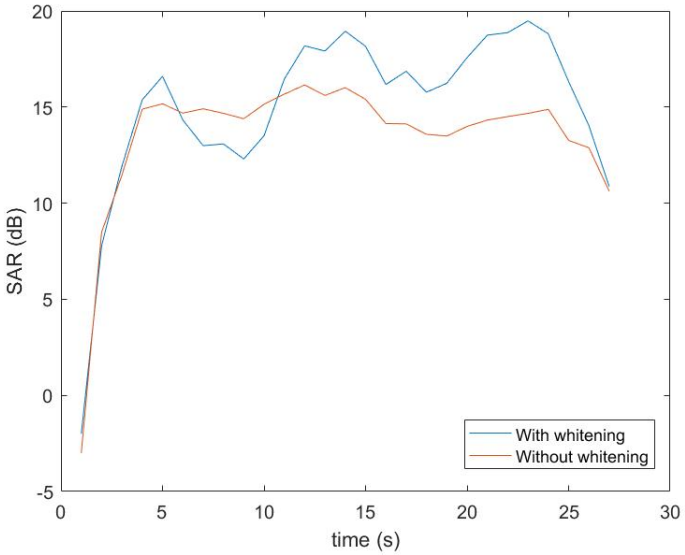


Figure 3.28: SAR of source 1 (dB) evolution in time (s) 1M+1F.

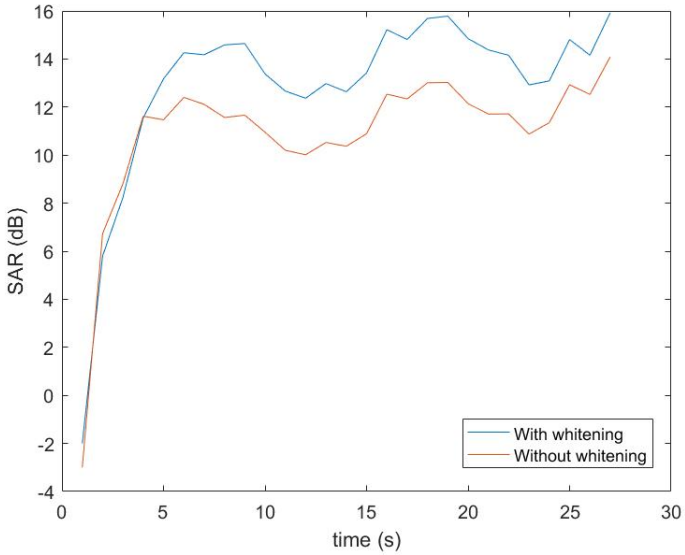


Figure 3.29: SAR of source 2 (dB) evolution in time (s) 1M+1F.

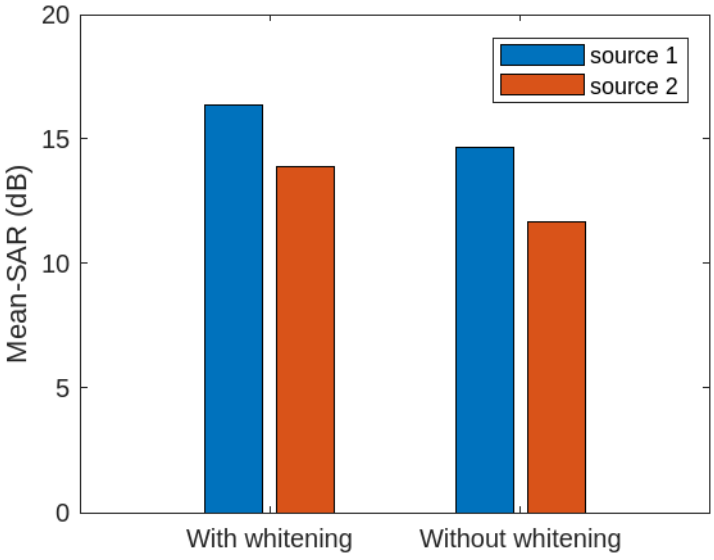


Figure 3.30: Mean values of SAR in the case of one male and one female speakers.

Figures 3.28, 3.29 and 3.30 SAR values (both over time and mean values in bars with and without whitening respectively) for one male and one female speakers case, show that whitening increases SAR of the estimated sources. Both algorithms converge over time to approximately 20 and 15 dB for source 1 and to 17 and 15 dB for source 2. However, there are some fluctuations because the learning rule is stochastic.

The table below provides the overall average-values of SIR SDR and SAR for all three cases of two speakers' scenario for NG online IVA with (w) and without (w/o) whitening.

		SIR (dB)		SDR (dB)		SAR (dB)	
Case	Methods	S1	S2	S1	S2	S1	S2
2 females	NG IVA w whitening	14.20	12.31	20.10	17.88	16.05	14.67
	NG IVA w/o whitening	11.60	8.34	15.30	19.51	14.38	8.85
2 males	NG IVA w whitening	16.23	14.85	18.75	19.79	20.62	16.86
	NG IVA w/o whitening	13.90	13.28	17.43	18.11	17.14	15.28
1 male 1 female	NG IVA w whitening	12.7	11.41	15.78	14.36	16.37	13.91
	NG IVA w/o whitening	11.80	9.86	17.10	15.01	14.65	11.67

Table 3.3: Two sources: the algorithms' performances (average SIR SDR and SAR).

Comment:

The results in table 3.3 demonstrate that both algorithms performed well in separating the sources in the two-speaker scenario, with the majority of values exceeding 10 dB for all three performance criteria (Average SIR, SDR and SAR values). Notably, the NG IVA algorithm with whitening consistently outperformed the NG IVA algorithm without whitening, except on two occasions where the performance was comparable.

The highest recorded values for SIR, SDR, and SAR were 16.23 dB, 20.10 dB, and 20.62 dB, respectively, while the lowest values were 8.34 dB, 14.36 dB, and 8.85 dB, respectively. It is worth mentioning that the adaptive NG IVA algorithm with whitening achieved the highest values, whereas the adaptive NG IVA algorithm without whitening attained the lowest values among the evaluated configurations.

These findings indicate that the algorithms exhibit satisfactory performance in separating sources in the two-speaker scenario with different male and female combinations, as evidenced by the high SDR, SAR, and SIR values.

3.2.3.2 Three sources scenario

Two females and one male case: Here below, we present the performances in the case of two female and one male speakers.

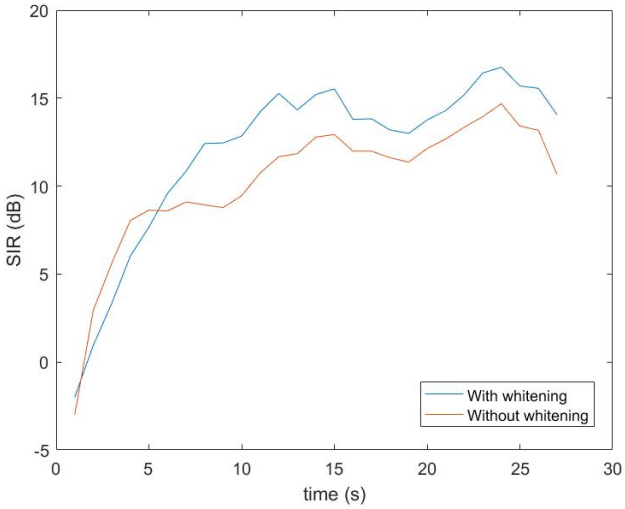


Figure 3.31: SIR of source 1 (dB) evolution in time (s) 2F+1M.

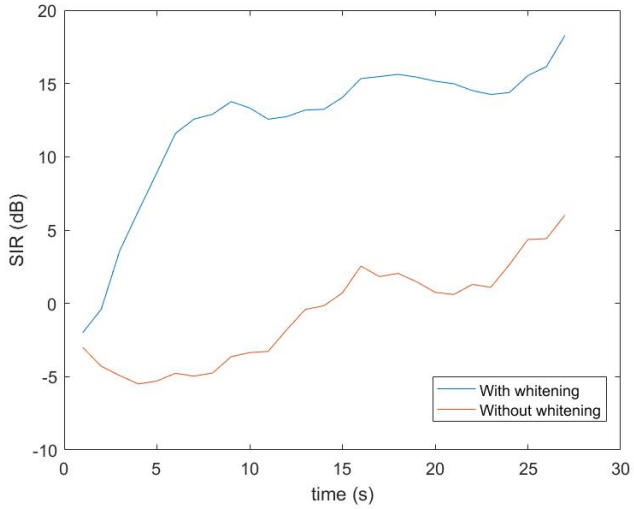


Figure 3.32: SIR of source 2 (dB) evolution in time (s) 2F+1M.

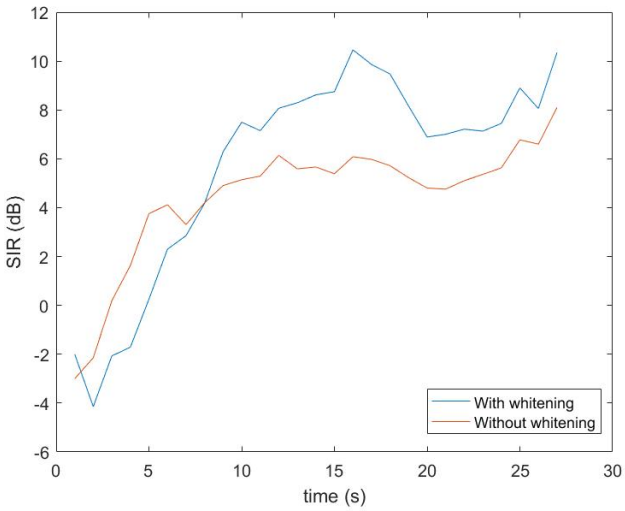


Figure 3.33: SIR of source 3 (dB) evolution in time (s) 2F+1M.

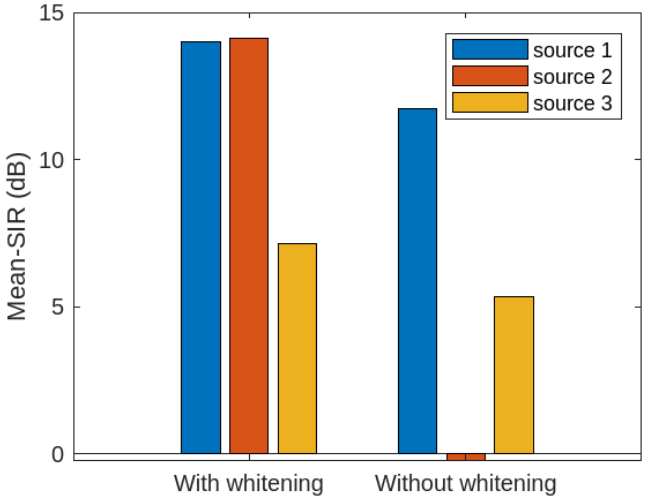


Figure 3.34: Mean values of SIR in the case of two females and one male speakers.

Figures 3.33, 3.32, 3.33 and 3.34 show SIR values (both over time and mean values in bars with and without whitening respectively) for two females one male speakers case, show that whitening increases SIR of the estimated sources. Both algorithms converge over time to approximately 15 and 12 dB for source 1, to 15 and 1 dB for source 2 and to 8 and 6 dB for source 3. However, there are some fluctuations because the learning rule is stochastic.

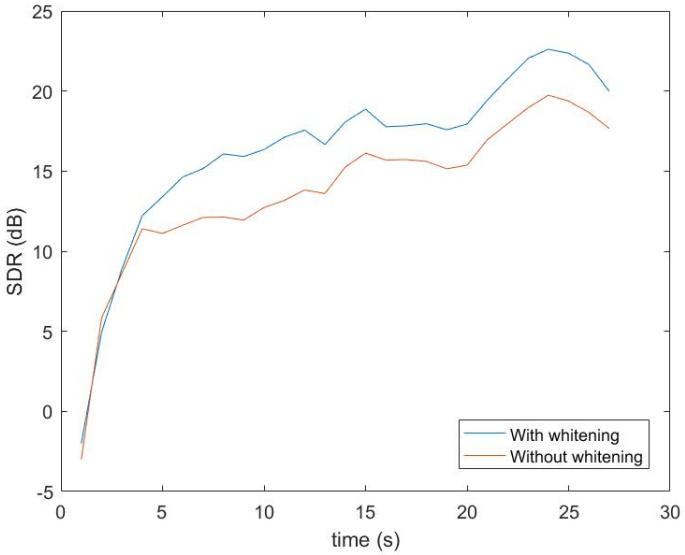


Figure 3.35: SDR of source 1 (dB) evolution in time (s) 2F+1M.

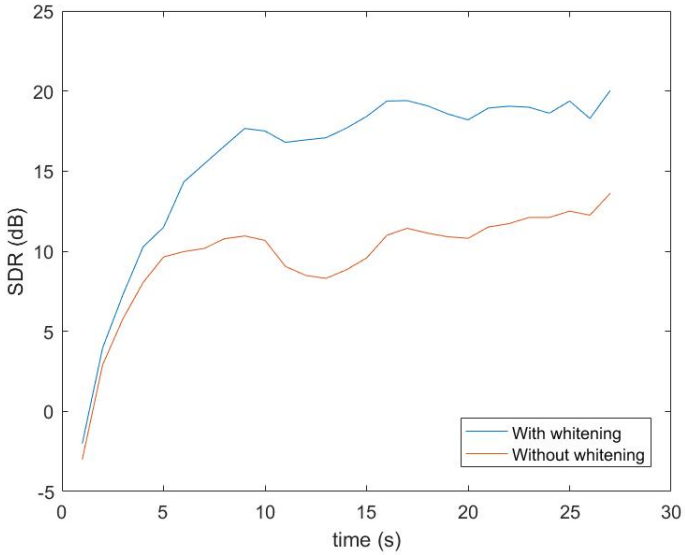


Figure 3.36: SDR of source 2 (dB) evolution in time (s) 2F+1M.

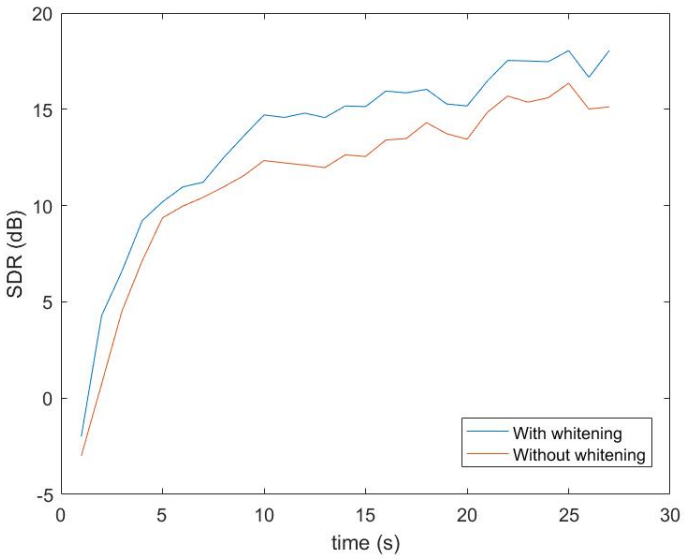


Figure 3.37: SDR of source 3 (dB) evolution in time (s) 2F+1M.

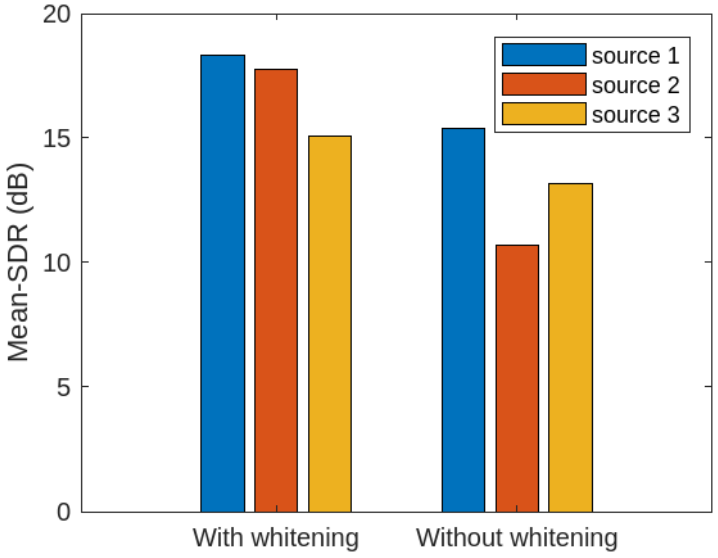


Figure 3.38: Mean values of SDR in the case of two females and one male speakers.

Figures 3.33, 3.32, 3.33 and 3.34 show SDR values (both over time and mean values in bars with and without whitening respectively) for two females one male speakers case, show that whitening increases SDR of the estimated sources. Both algorithms converge over time to approximately 20 and 16 dB for source 1, to 20 and 10 dB for source 2 and to 15 and 13 dB for source 3. However, there are some fluctuations because the learning rule is stochastic.

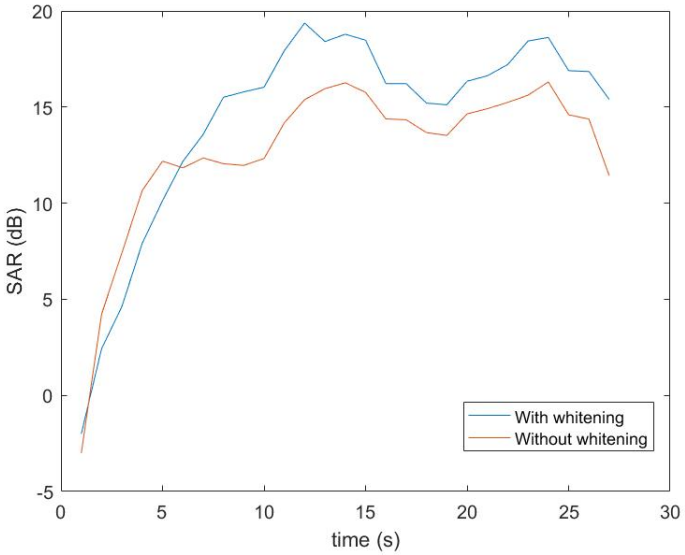


Figure 3.39: SAR of source 1 (dB) evolution in time (s) 2F+1M.

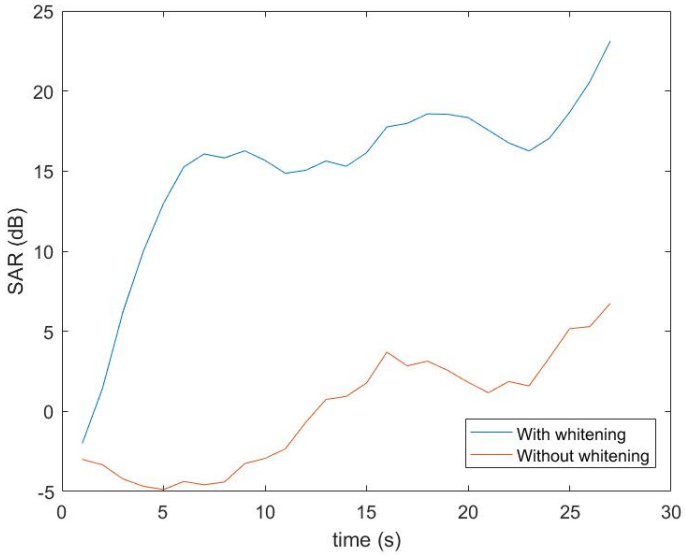


Figure 3.40: SAR of source 2 (dB) evolution in time (s) 2F+1M.

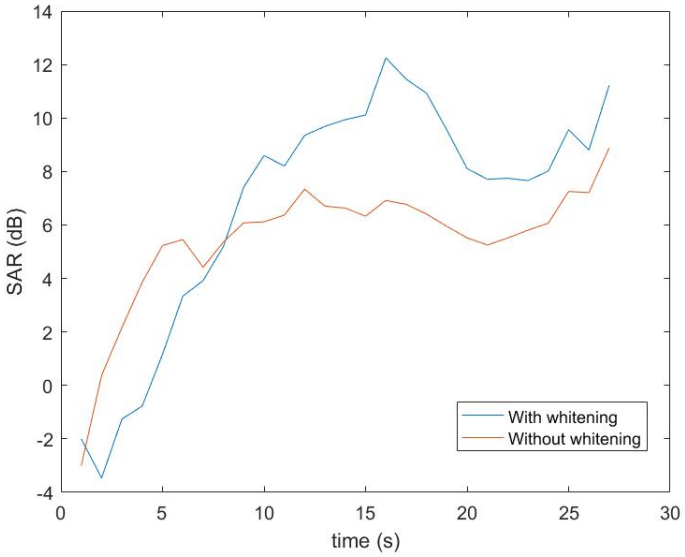


Figure 3.41: SAR of source 3 (dB) evolution in time (s) 2F+1M

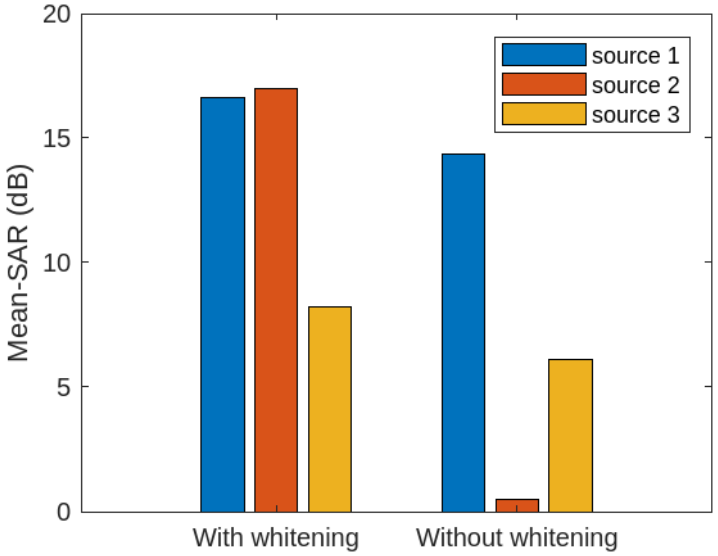


Figure 3.42: Mean values of SAR in the case of two females and one male speakers.

Figures 3.39, 3.40, 3.41 and 3.42 show SAR values (both over time and mean values in bars with and without whitening respectively) for two females one male speakers case, show that whitening increases SAR of the estimated sources. Both algorithms converge over time to approximately 19 and 15 dB for source 1, to 18 and 0 dB for source 2 and to 11 and 6 dB for source 3. However, there are some fluctuations because the learning rule is stochastic.

Two males and one female: Here below, we present the performances in the case of two female and one male speakers.

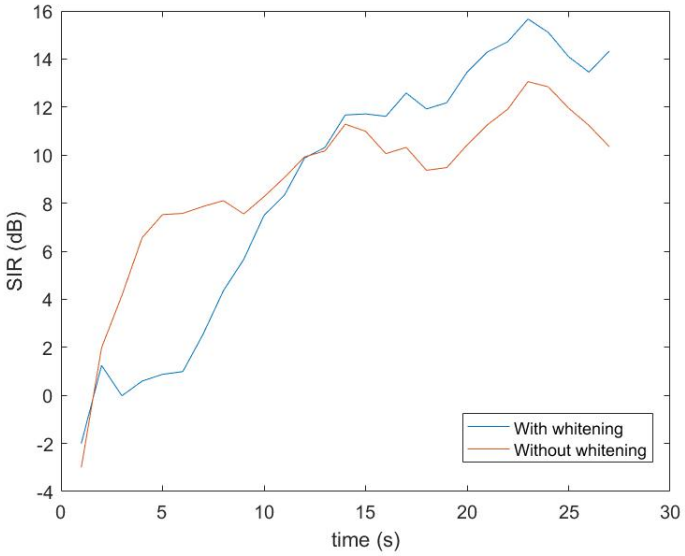


Figure 3.43: SIR of source 1 (dB) evolution in time (s) 2M+1F.

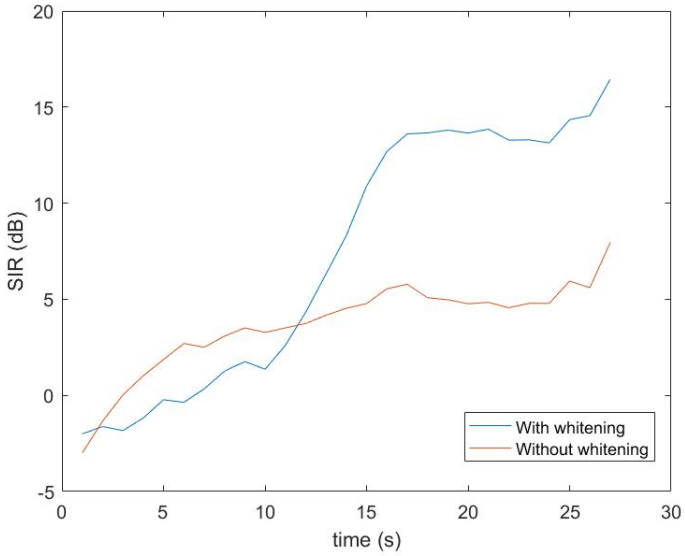


Figure 3.44: SIR of source 2 (dB) evolution in time (s) 2M+1F.

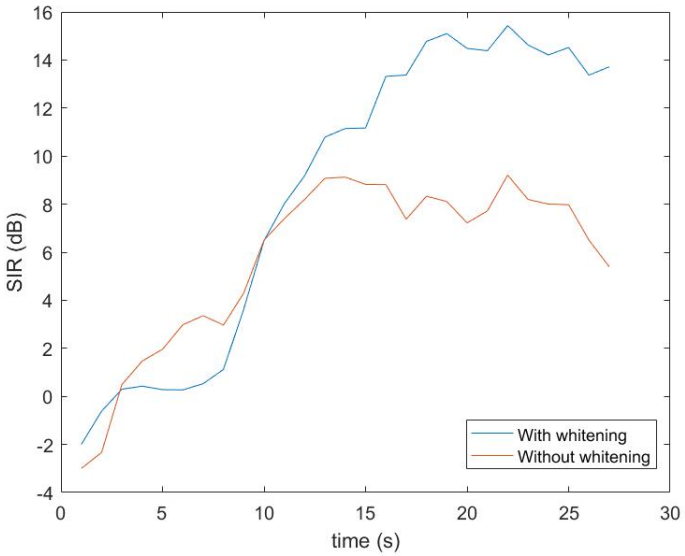


Figure 3.45: SIR of source 3 (dB) evolution in time (s) 2M+1F

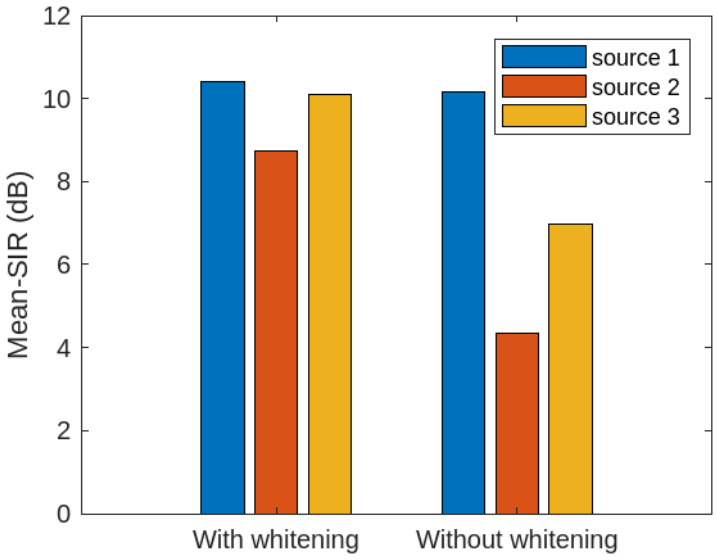


Figure 3.46: Mean values of SIR in the case of two males and one female speakers.

Figures 3.43, 3.44, 3.45 and 3.46 show SIR values (both over time and mean values in bars with and without whitening respectively) for two males and one female speakers case, show that whitening increases SIR of the estimated sources. Both algorithms converge over time to approximately 16 and 12 dB for source 1, to 15 and 5 dB for source 2 and to 16 and 10 dB for source 3. However, there are some fluctuations because the learning rule is stochastic.

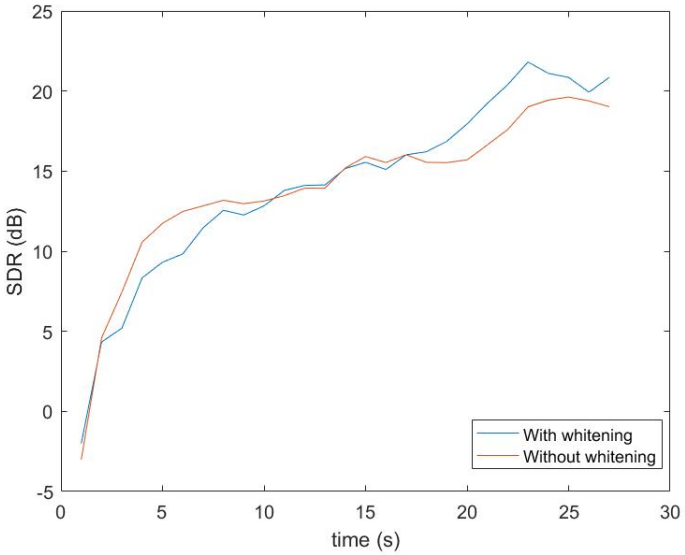


Figure 3.47: SDR of source 1 (dB) evolution in time (s) 2M+1F

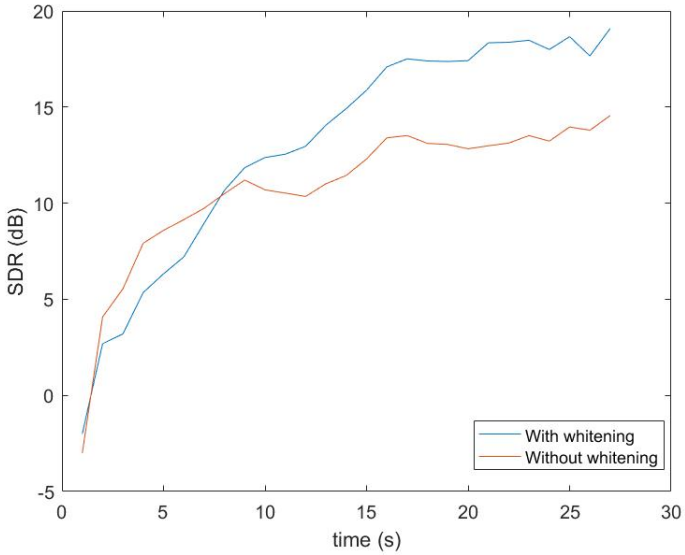


Figure 3.48: SDR of source 2 (dB) evolution in time (s) 2M+1F.

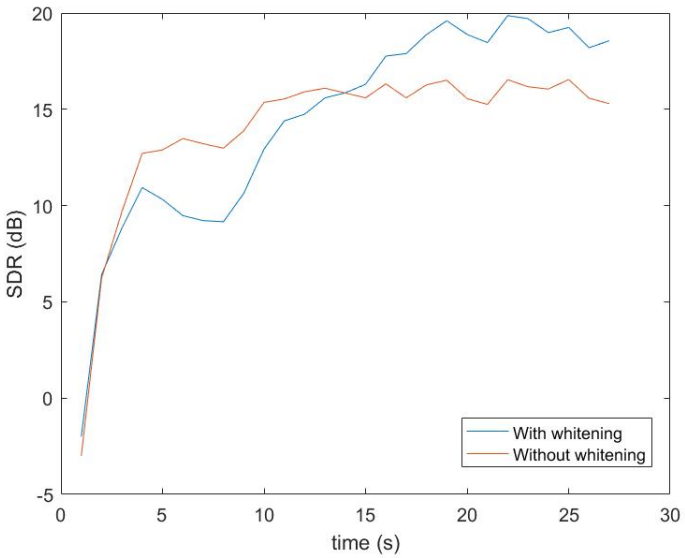


Figure 3.49: SDR of source 3 (dB) evolution in time (s) 2M+1F

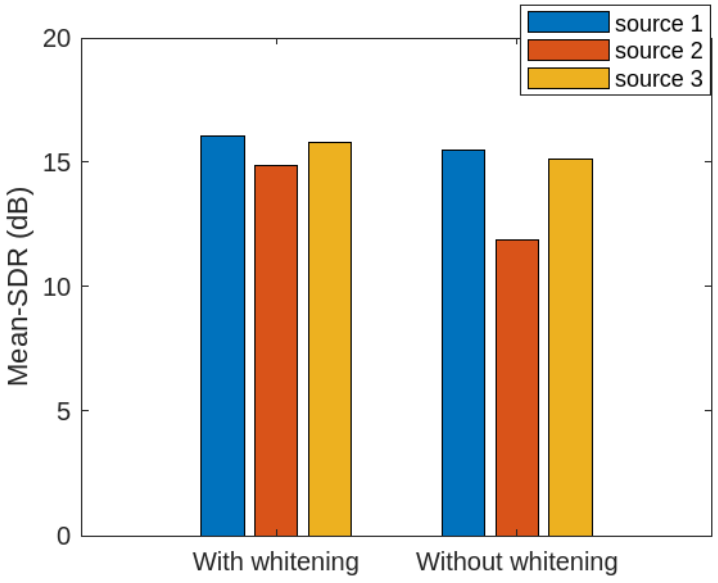


Figure 3.50: Mean values of SDR in the case of two males and one female speakers.

Figures 3.47, 3.48, 3.49 and 3.50 show SDR values (both over time and mean values in bars with and without whitening respectively) for two males and one female speakers case, show that whitening increases SDR of the estimated sources. Both algorithms converge over time to approximately 20 and 18 dB for source 1, to 18 and 14 dB for source 2 and to 20 and 15 dB for source 3. However, there are some fluctuations the learning rule is stochastic.

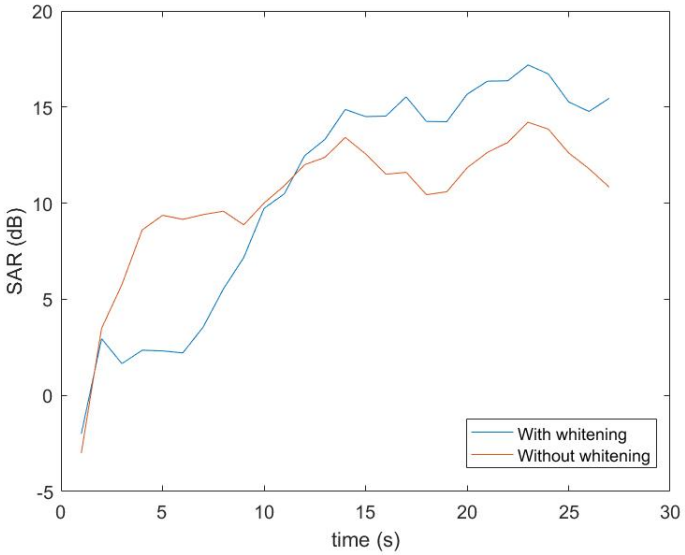


Figure 3.51: SAR of source 1 (dB) evolution in time (s) 2M+1F

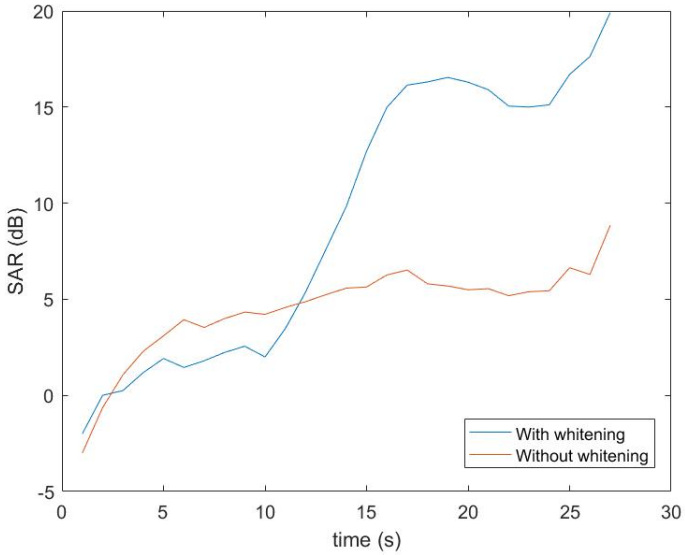


Figure 3.52: SAR of source 2 (dB) evolution in time (s) 2M+1F.

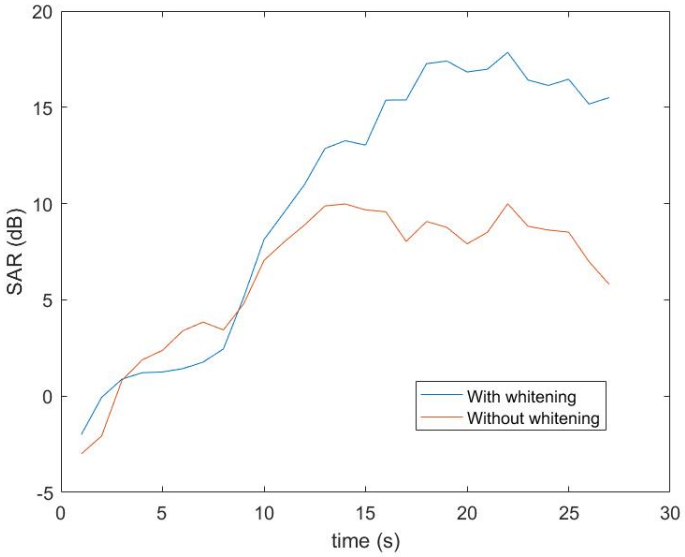


Figure 3.53: SAR of source 3 (dB) evolution in time (s) 2M+1F

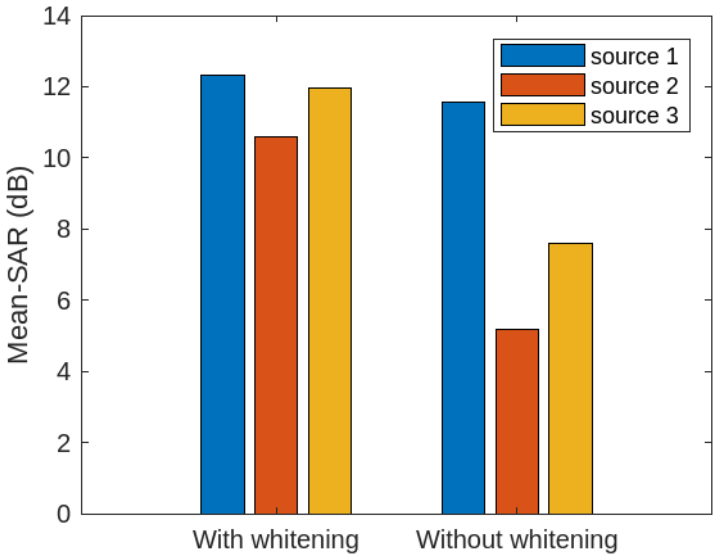


Figure 3.54: Mean values of SAR in the case of two males and one female speakers.

Figures 3.51, 3.52, 3.53 and 3.54 show SAR values (both over time and mean values in bars with and without whitening respectively) for two males and one female speakers case, show that whitening increases SAR of the estimated sources. Both algorithms converge over time to approximately 15 and 13 dB for source 1, to 15 and 5 dB for source 2 and to 18 and 10 dB for source 3. However, there are some fluctuations the learning rule is stochastic.

The table 3.4 below provides the overall average-values of SIR SDR and SAR for all two cases of three speakers' scenario for NG online IVA with (w) and without (w/o) whitening.

		SIR (dB)			SDR (dB)			SAR (dB)		
Case	Methods	S1	S2	S3	S1	S2	S3	S1	S2	S3
2F + 1M	NG IVA w whitening	14.01	14.12	7.13	18.32	17.78	15.08	16.62	17.00	8.20
	NG IVA w whitening	11.74	−0.21	5.32	15.39	10.68	13.19	14.33	0.56	6.16
2M + 1F	NG IVA w whitening	10.39	8.75	10.11	16.07	14.86	15.79	12.34	10.64	11.97
	NG IVA w whitening	10.16	4.33	6.96	15.65	11.95	15.34	11.57	5.19	7.59

Table 3.4: Three sources: the algorithms' performances (average SIR SDR and SAR).

Comment:

The obtained results in table 3.4 demonstrate the effectiveness of both algorithms in the task of source separation, with the majority of values surpassing the threshold of 10dB for all three performance criteria, namely SDR, SAR SIR.

Notably, the algorithm incorporating whitening in adaptive NG IVA consistently outperforms the algorithm without whitening, demonstrating superior separation performance in the majority of cases. This observation suggests that incorporating whitening into the NG IVA algorithm enhances its ability to discriminate and separate sources in a more robust manner.

Although the performance of the algorithms slightly declined when confronted with the scenario involving three speakers, they still exhibited noteworthy separation capabilities. The highest recorded values for SIR, SDR, and SAR were 14.12dB, 18.32dB, and 17.00dB, respectively, while the lowest values were -0.21dB, 10.68dB, and 0.56dB, respectively. It is worth noting that the adaptive NG IVA algorithm with whitening achieved the highest values, while the adaptive NG IVA algorithm without whitening attained the lowest values among the tested configurations.

3.2.3.3 Noise effect

Next on, we evaluated the algorithm's performance in a noisy environment, for that we generated white noise with SNR values ranging from -20 to 30 dB and plotted average values over time for SIR, SDR and SAR with Monte-Carlo realization (N_{MC}=20).

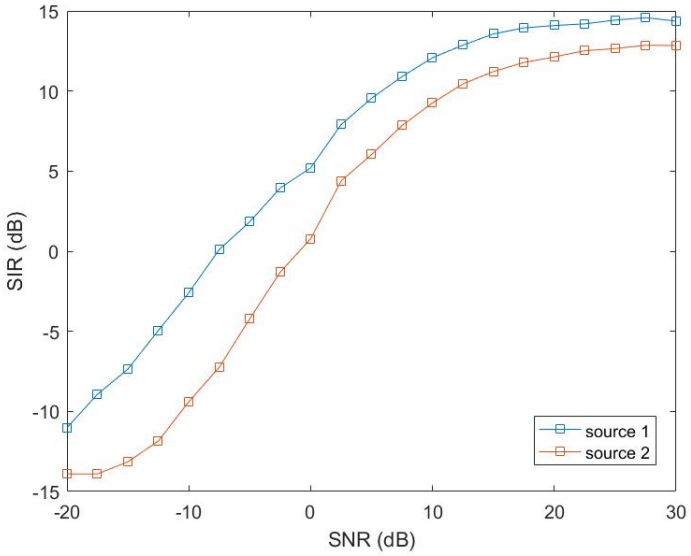


Figure 3.55: Effect of SNR on the SIR of separated signals (2 males) using MC runs.

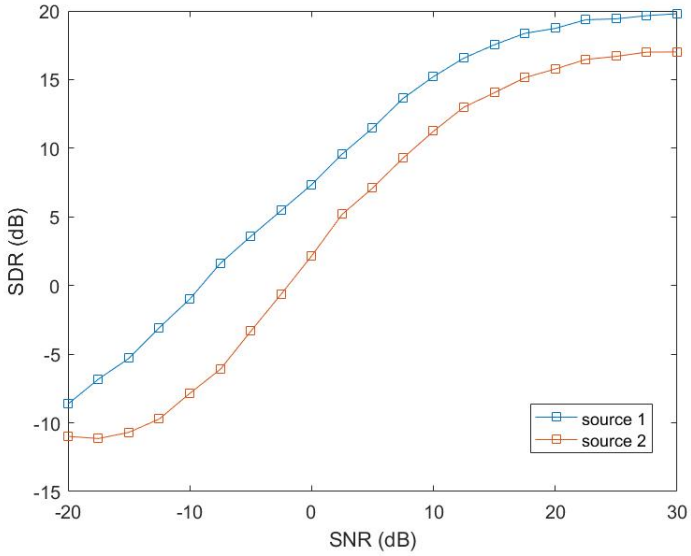


Figure 3.56: Effect of SNR on the SDR of separated signals (2 males) using MC runs.

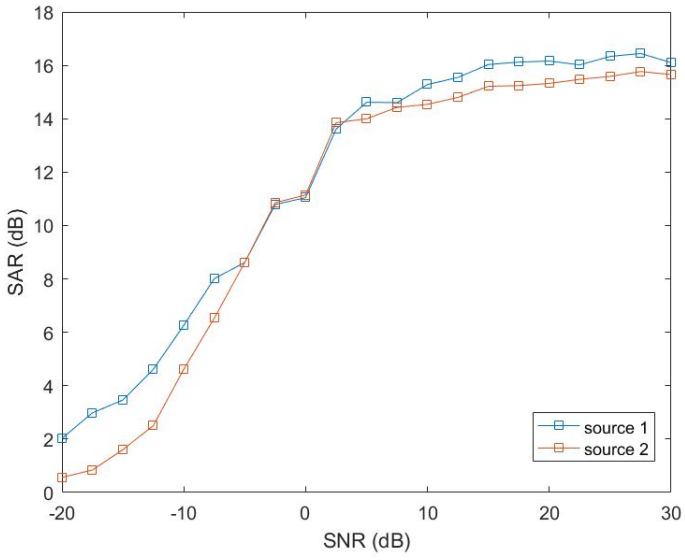


Figure 3.57: Effect of SNR on the SAR of the separated signals (2 males) using MC runs.

Figures 3.55, 3.56 and 3.57 show that the algorithm's performance increases when the noise level decreases.

Finally to prove the convergence of the algorithm over time blocs indexes (n), we computed the $MSSG$ (Mean Squared Sum of Gradient) of instantaneous natural gradient matrix:

$$MSSG(n) = \frac{1}{NFL} \sum_{i,j,f} |\Delta \mathrm{w}_{i,j}(f,n)|^2 \tag{3.42}$$

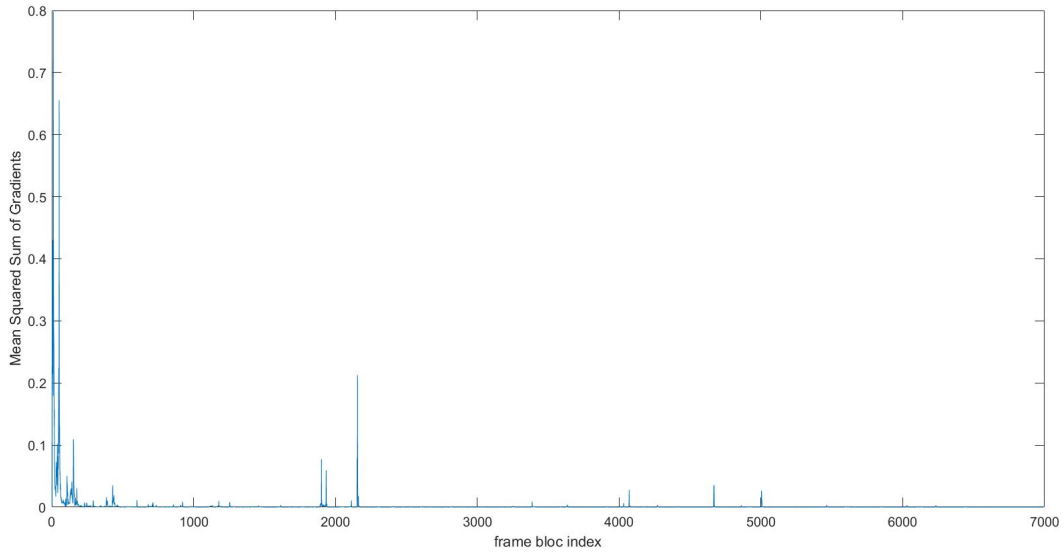


Figure 3.58: MSSG of adaptive NG IVA convergence for the case of two female speakers.

As shown by figure (3.58) the sum convergence to zero over the adaptation, which confirms the algorithm convergence, however the convergence was a bit erratic especially in early stages but afterwards it settles quickly.

3.3 Conclusion

In this chapter we presented at the mathematical formulation for the adaptive version of natural gradient IVA and proposed an adaptive whitening which allows us to have more microphones than sources unlike the standard adaptive Natural Gradient IVA in [32] where the number of sources was forced to be equal to the number of microphones.

Adaptive NG IVA was evaluated with both whitening and without whitening. The results show that the algorithm achieves good separation in both cases. Although adaptive NG IVA with whitening outperforms adaptive NG IVA without whitening, it comes with a computational cost since adaptive NG IVA takes twice the time the standard adaptive NG IVA whcih is due mainly to eigenvalue decomposition at each frame.

Chapter 4

Real world tests using UMA-16 v2

To validate the algorithm's efficacy in real-life scenarios, we conducted experiments using recorded mixtures of multiple speakers captured with a UMA-16 v2 microphone array. In this context, we provide a comprehensive description of the UMA-16, including its technical specifications, physical characteristics, and detailed operating mode guidelines. By extending the evaluation to real-life mixtures, we aim to assess the algorithm's performance under more realistic conditions and validate its suitability for practical applications.

4.1 UMA-16 v2

4.1.1 Who are miniDSP

miniDSP, a prominent manufacturer in the field of Digital Audio Signal Processors, has established itself as a leader in providing solutions for various markets including Home Theater, HiFi, headphones, and the automotive industry. Founded in 2009, miniDSP is a technology company that specializes in developing Digital Signal Processing (DSP) platforms for audio applications. Headquartered in Hong Kong, a bustling and vibrant city, miniDSP benefits from its close proximity to Shenzhen, China's largest electronic manufacturing hub, allowing the company to actively engage in the dynamic industry. The company's growth and success are driven by a passion for technology and an agile product development approach, which has enabled the development of valuable in-house intellectual property. [60]

4.1.2 UMA-16 v2 USB

The Uniform Microphone Array (UMA)-16 v2 shown in figure 4.1 is the latest Digital Audio Signal Processor device developed by **miniDSP** it is a sixteen-channel microphone array with plugplay USB audio connectivity. With its onboard XMOS interface, the UMA-16 is the perfect fit for the development of beamforming algorithms or for DIY acoustic camera. Its system architecture consists of two core elements:

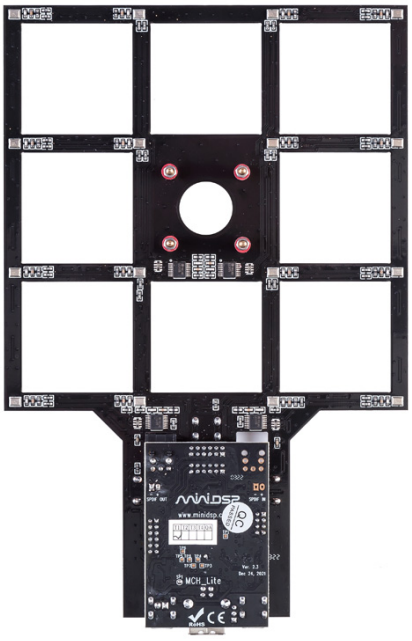


Figure 4.1: UMA-16 front.

• A microphone PCB with 16 x Knowles SPH1668LM4H MEMS microphones in a uniform rectangular array (URA). A center hole fits an optional USB camera for applications such as acoustic cameras. The microphone PCB is a simple 2-layer design that can easily be customized to your needs by following the schematics included in the user manual.

• Stacked on top of the mic array is the MCHStreamer Lite USB interface. This XMOS XCORE interface allows for a high quality PDM to PCM conversion and presents all 16 channels of raw audio to the USB interface.

Features

① Multichannel USB microphone array for voice command.

② 16 channels of RAW audio for development of custom beamforming algorithms

③ High quality MEMS from Knowles SPH1668LM4H.

④ USB to PDM conversion for up to 16 x PDM MEMS microphones. Center Hole for USB camera (not provided).

⑤ Sample Matlab code to get started.

Technical specifications

Below in table 4.1 the technical specifications of UMA-16 v2 USB [61].

Item	Description
USB audio input	XMOS Xcore200 asynchronous USB audio up to 48 kHz, USB Audio Class 2 compliant ❍ ASIO driver for Windows ❍ Driverless for macOS
PDM	inputs Up to 16 x MEMS microphone connections (8 x stereo PDM data lines)
MEMS microphone	16 x SPH1668LM4H - Acoustic Overload @ 120dBSPL / High SNR of 65dB / RF shielded
ADC/DAC Sample rate	Sample rate: 8, 11.025, 12, 16, 32, 44.1 or 48 kHz
Resolution	Resolution: 24 bit
USB port	USB port type Mini-B for audio streaming and firmware upgrade
Power supply	USB powered
Dimensions	132 x 202 x 18 mm (H x W x D)
Mounting	4 x M3 holders for front panel mounting / CAD drawings available on demand.

Table 4.1: Key technical features of the UMA-16 v2.

Discovering the UMA-16 with Matlab

Inside MATLAB, we can define our recording interface using the device reader object

```
fs = 48000;
audioFrameLength = 1024;
deviceReader = audioDeviceReader(...
 'Device', 'miniDSP ASIO Driver',...
 'Driver', 'ASIO', ...
 'SampleRate', fs, ...
 'NumChannels', 16 ,...
 'OutputDataType','double',...
 'SamplesPerFrame', audioFrameLength);
```

Listing 4.1: Recording interface code in MATLAB.

Microphone array mechanical drawing

Down below the microphone array mechanical drawing, the inter-element spacement between two microphones along the x-axis is $d_x = 42$mm and the distance along the y-axis is $d_y = 42$mm, these distances are chosen as to avoid spatial aliasing in both direction. This known as the Nyquiest spatial criterion:

$$d_x \leq \frac{\lambda}{2} \tag{4.1a}$$

$$d_y \leq \frac{\lambda}{2} \tag{4.1b}$$

where $\lambda = \dfrac{c}{f_{max}}$ is the wavelength of the greatest frequency component f_{max}. In audio processing the bandwidth allocated for a single voice-frequency transmission channel is usually 4 kHz [62], $c = 343\text{ms}^{-1}$ is sound speed in the air. Under those conditions

$$\lambda = \frac{343}{4000} = 85.75\text{mm} \tag{4.2}$$

half this wavelength is equal to 42.875mm. And thus a value of interelement spacement $d_x = d_y = d = 42$mm is well less than, half this wavelength. This is shwon in Figure 4.4

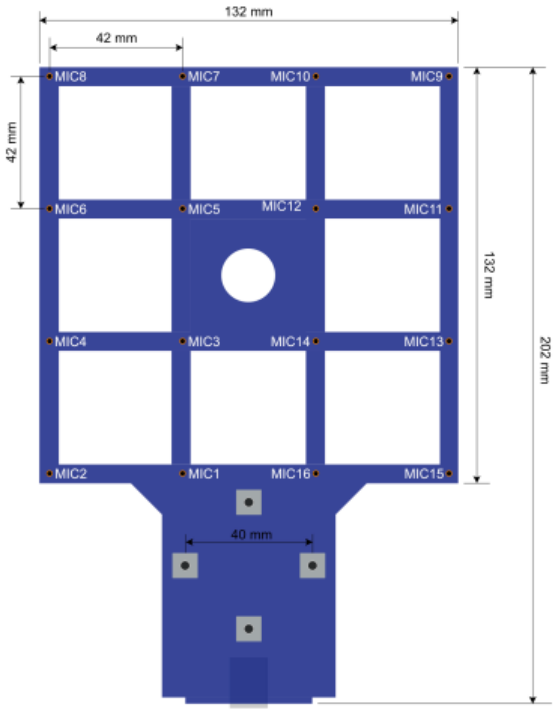


Figure 4.2: Front.

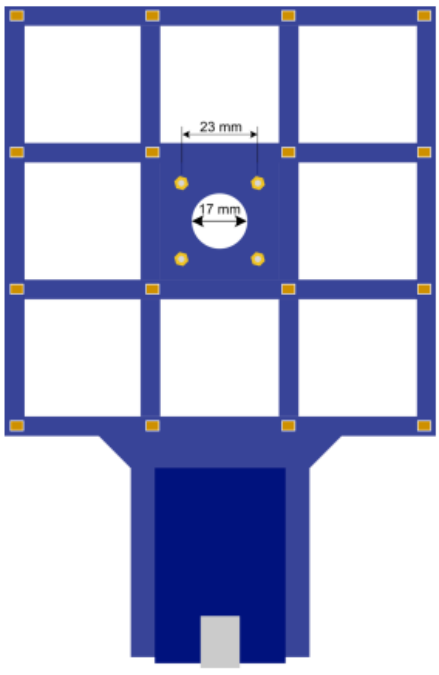


Figure 4.3: Back

Figure 4.4: Uma-16 mechanical drawing.

Circuit Schematics

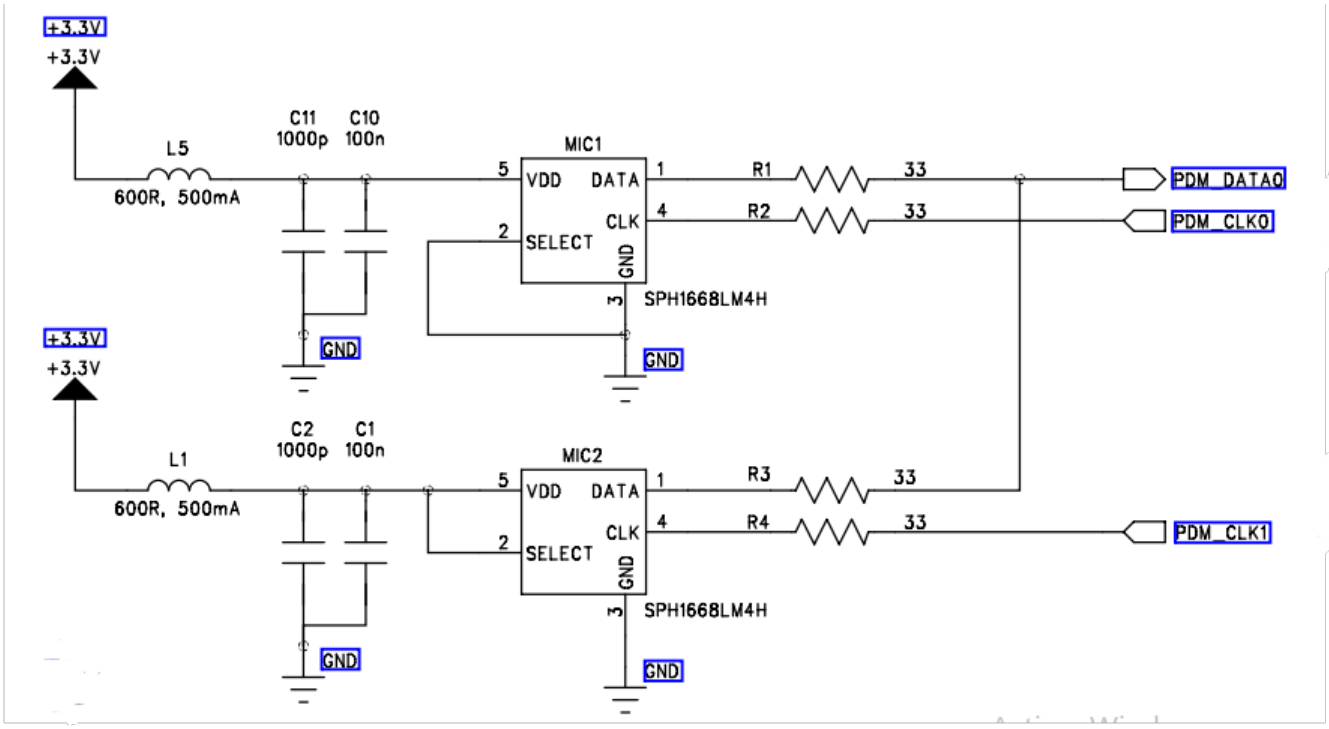


Figure 4.5: PDM-Microphone schematic.

An example of a PDM (Pulse Density Modulation) microphone is shown in figure 4.5. PDM microphones are digital microphones that produce a bitstream, or stream of single-bit digital data. The digital representation of the analog sound signal is what the PDM microphone outputs. A MEMS (Micro-Electro-Mechanical System) microphone sensor and an integrated circuit (IC) that transforms the sensor's output into a PDM signal are both included in the PDM microphone. The circuit is composed of an *LC* the inductance is 500mH and the first capacitor is 1000pF and the second is 100nF. The microphone used is SPH1668LM4H (figure 4.8) . And it has the following features

- Low Distortion of 1.6% at 120dB SPL and High SNR of 65.5dB
- Flat Frequency Response and RF Shielded
- Supports Dual Multiplexed Channels and Omnidirectional.

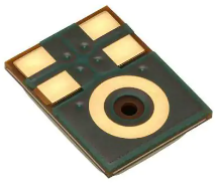


Figure 4.6: Front.

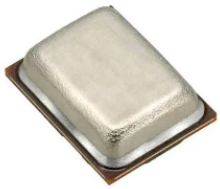
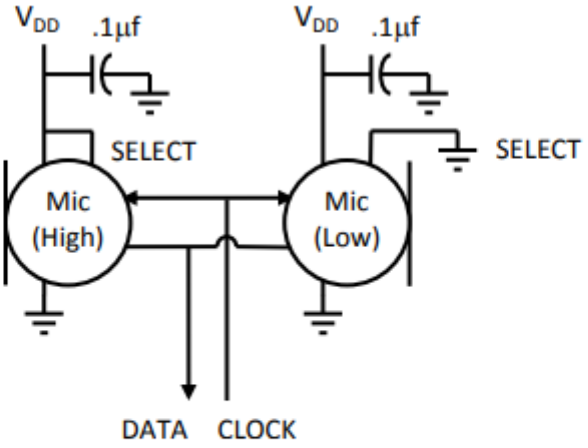


Figure 4.7: Back.

Figure 4.8: SPH1668LM4H-1 microphones.

Each microphone is made of silicon by knowless electronics has a voltage V_{DD} in range 1.62V $\sim$ 3.6V and a clock that works on a frequency f_{clock} in range 100Hz $\sim$ 10 kHz. The circuitry is given down below in figure 4.9:



Microphone	SELECT	Asserts DATA On	Latch DATA On
Mic (High)	V_{DD}	Rising Clock Edge	Falling Clock Edge
Mic (Low)	GND	Falling Clock Edge	Rising Clock Edge

Figure 4.9: Microphone's circuit diagram.

The timing diagram for the microphone is the shown in figure 4.10. [63]

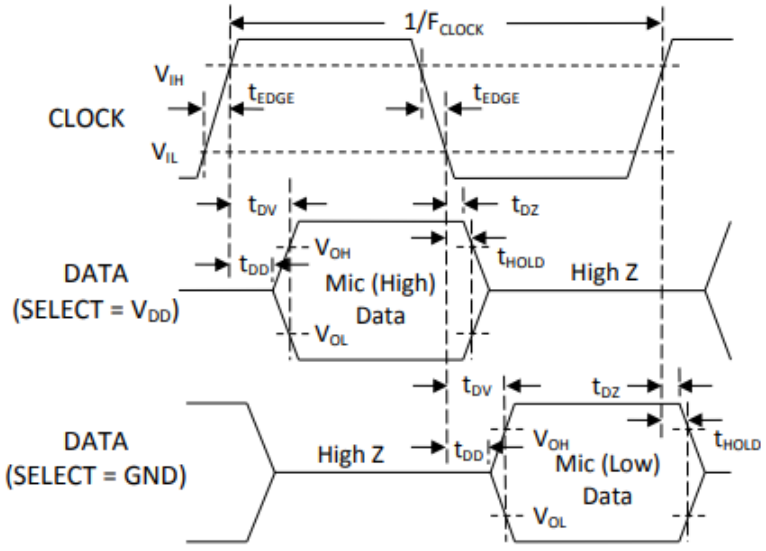


Figure 4.10: Microphone's timing diagram.

4.2 UMA-16 v2 operating mode

After providing a detailed physical description of the UMA-16 v2 array, the subsequent section will focus on elucidating its operating mode. This section is intended to serve as a comprehensive manual, offering instructions and guidelines on effectively utilising the microphone array.

4.2.1 Connectivity and USB Driver

The initial step involves the installation of the USB driver specifically associated with the Windows operating system. This driver installation can be accomplished redeeming the coupon received together with UMA-16 v2 and following the provided instructions. Subsequently, the microphone array is connected to the computer via a USB type A to type B cable (figure 4.11). Once the connection is established, the computer system should be capable of detecting the presence of the microphone array, which is visually indicated by the initiation of a blinking blue LED light on the array as shown in figure 4.12.

Figure 4.11: USB type A to type B cable.

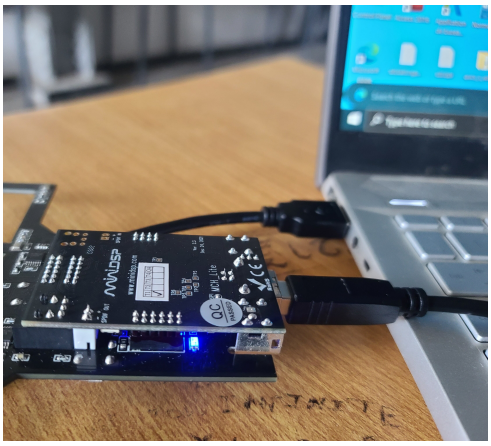


Figure 4.12: UMA-16 successfully connected.

4.2.2 Control panel

To configure the parameters of the UMA-16 v2 array, one should navigate to the control panel menu on the system, open the control panel, select Hardware and Sound, and choose Manage Audio Devices. In the list of audio devices, locate "MCHStreamer Multi-channels" as shown in figure 4.13 which represents the UMA-16 v2 array. Selecting this option provides access to a range of configurable parameters for the array. These parameters allow adjustments to settings such as input/output levels, sample rates, audio formats, buffer sizes, and more, tailored to specific needs.

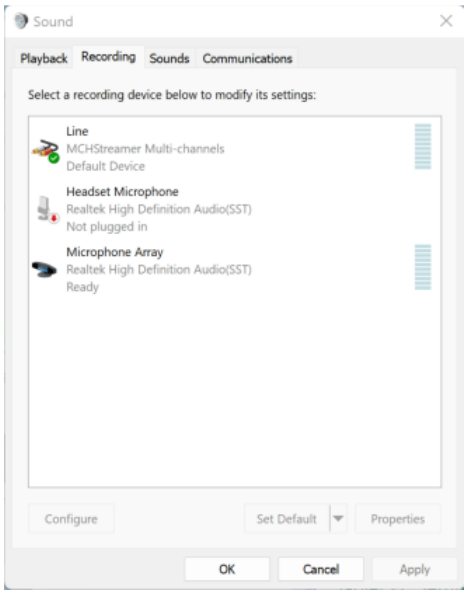


Figure 4.13: Control Panel UMA-16 v2.

4.2.2.1 Sampling rate and bit depth

The input signal of the UMA-16 v2 array is characterised by a flexible sampling rate that spans from 8kHz to 48kHz accommodating a wide range of audio applications as shown in figure 4.14. This flexibility in the sampling rate can be easily configured through the control panel, enabling users to adapt it based on their specific usage scenarios and requirements. Similarly, the UMA-16 v2 array provides users with the flexibility to select the desired audio depth. The audio depth refers to the number of bits used to represent each audio sample. With the UMA-16 v2 array, users can choose between two options: 16 bits or 24 bits.

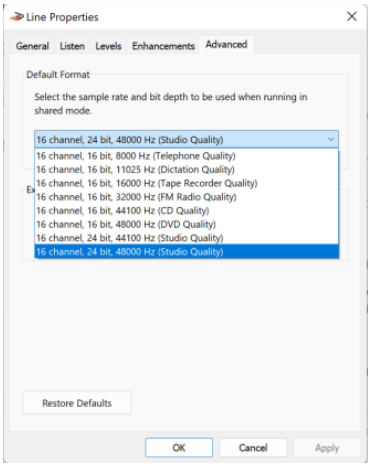


Figure 4.14: Sampling rate and depth adjustment for UMA-16 v2.

4.2.2.2 Adjusting the volume

In order to adjust the volume levels of the UMA-16 v2 array, users can utilise the control panel interface. By navigating to the designated "volume" window within the control panel as shown in figure 4.15. One can also deactivate any microphone by clicking down below at the two brackets.

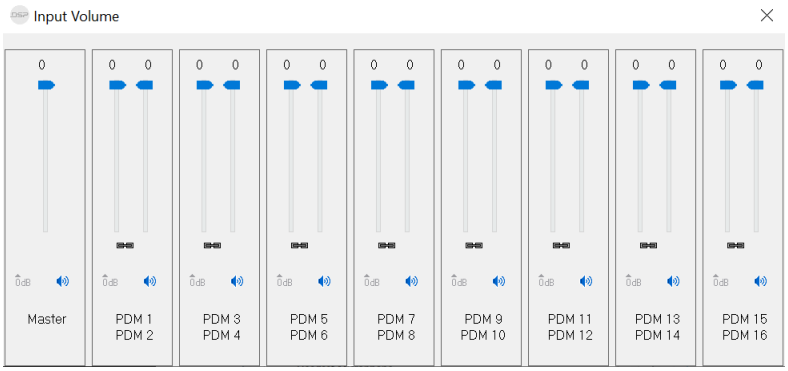


Figure 4.15: Volume adjustment for UMA-16 v2.

4.2.2.3 Buffer settings

The UMA-16 v2 array allows users to optimise latency by adjusting the buffer size using the ASIO (Audio Stream Input/Output) protocol [64]. While it is recommended to keep the default buffer settings for non-professional audio applications, professionals can fine-tune the buffer size to meet specific latency requirements as illustarted in figure 4.16.

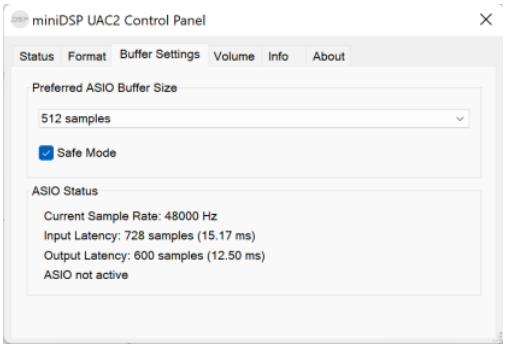


Figure 4.16: Buffer settings for UMA-16 v2.

4.2.3 Data acquisition

To record sound signals using the UMA-16 v2 array, users need to ensure that all relevant parameters have been properly configured. Once the necessary parameters are verified, the acquisition of data can be initiated using MATLAB.

```
deviceReader = audioDeviceReader(...
 'Device', 'miniDSP ASIO Driver',...
 'Driver', 'ASIO', ...
 'OutputDataType','double');
setup(deviceReader)
fileWriter = dsp.AudioFileWriter('raouf_samy_zahra.wav','FileFormat','
   WAV', 'DataType', 'int16');
disp('Speak into microphone now.')
tic %the timer start counting the recording time %
while toc < 27 %this is to specify the duration of the recording%
    acquiredAudio = step(deviceReader);
    step(fileWriter, acquiredAudio);
end
release(deviceReader);
release(fileWriter);
disp('Recording complete.')
```

Listing 4.2: Acquisition code on MATLAB.

In this code, the **audioDeviceReader** function is used to create an audio device reader object with the appropriate settings. The 'Device' parameter specifies the UMA-16 v2 array as the audio device, and the 'Driver' parameter sets the driver to ASIO.
The code then creates a **dsp.AudioFileWriter** object named **fileWriter** to save the recorded audio. The file name and format are specified within the user. The recording process starts with the **disp** function displaying a message to indicate that the user should start speaking into the microphone. The tic function starts the timer to measure the recording time.
Inside the while loop, the acquired audio is read using **step(deviceReader)** and converted to int16 data type using **int16(acquiredAudio)**. This is done to match the data type expected by the **dsp.AudioFileWriter** object. The **step(fileWriter, scaledAudio)** writes the scaled audio data to the .WAV file.
The while loop continues until the desired recording duration, specified by the **toc<27** (here 27 corresponds to 27 seconds which is the duration of our recorded signals for the tests) condition, is reached.
After the loop, the device reader and file writer objects are released using the release function, and a message is displayed to indicate that the recording is complete as shown in figure 4.17.

```
>> Record
Speak into microphone now.
Recording complete.
>>
```

Figure 4.17: Mixture successfully recorded on MATLAB using UMA-16 v2.

4.3 Real world tests

4.3.1 Experimental setup

The experiment was conducted in a room with dimensions of 7.5 m x 5 m x 3 m, where the UMA-16 v2 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-16 v2, connected to a computer, records the data at a sampling rate of 16 kHz for 27 seconds. All real-world signals used in our work were recorded using the RAW mode of the UMA-16 v2 with all the 16 MEMS microphones. The separation is then performed using NG IVA (both with whiening and without whitening process) using MATLAB , with the same parameters as in the previous chapter.

Figure 4.18: Two speakers case.

Figure 4.19: Three speakers case.

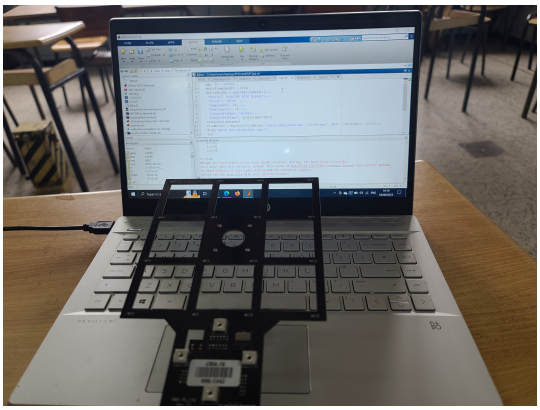


Figure 4.20: UMA-16 v2 and MATLAB setup.

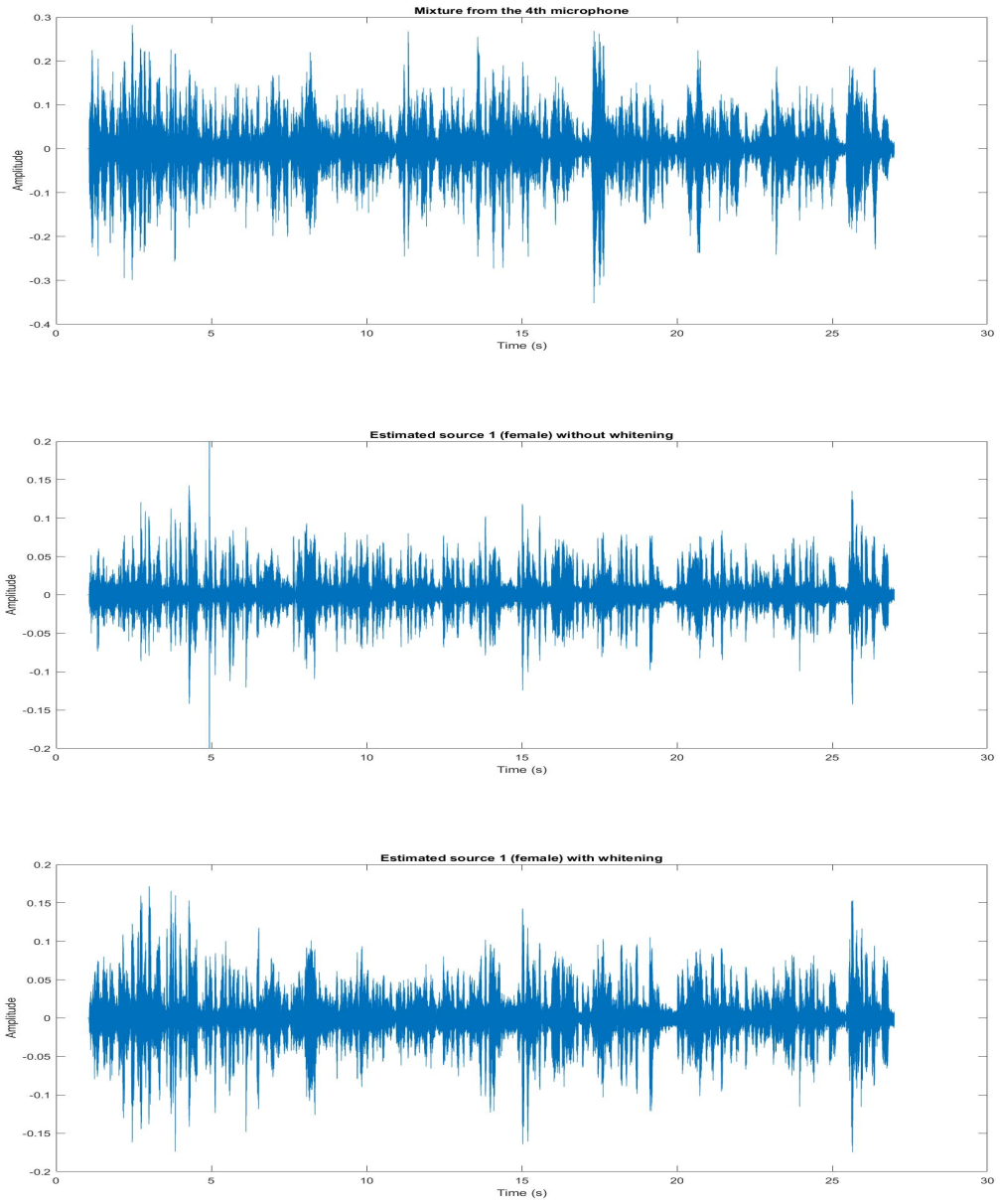
Figures 4.18, 4.19 illustrate real life mixtures experiments for the case of two speakers (one male and one female) and Three speakers case (two males and one female) respectively. Figure 4.20 show the microphone array along with MATLAB software for data acquisition.

4.3.2 Experimental results

Hereafter, two cases were considered. The first is a simultaneous discussion between one woman and one man, the second is a simultaneous discussion of three speakers two of which are men, and the third is a woman.

Case 1: One male speaker and one female speaker mixture

Figure 4.21 shows the 4th channel's signal recorded at the UMA-8 microphones and the two separate speech signals using: NG IVA with whitening and NG IVA without whitening. In this case, the standard NG IVA algorithm took 78 seconds to run, while the whitened version took twice that time 150 seconds.



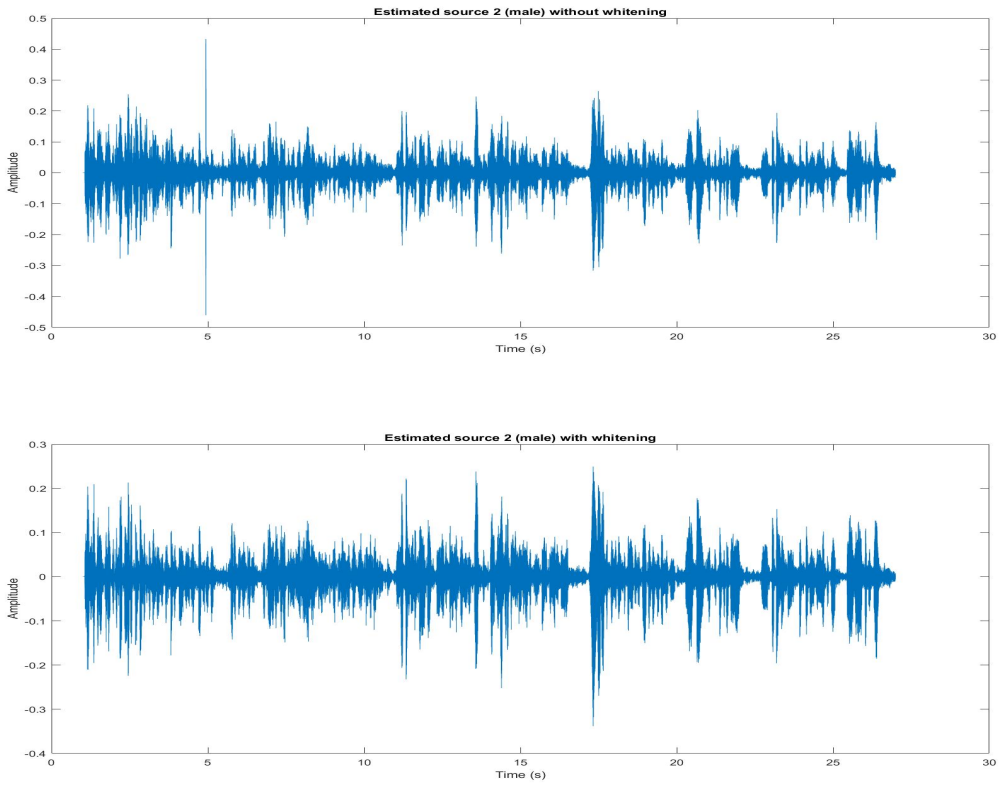


Figure 4.21: 1st experiment with real-world acoustic recordings: two sources mixture recorded by the 4th UMA-16 v2 microphone array, the separation results of NG IVA with whitening and without whitening algorithm.

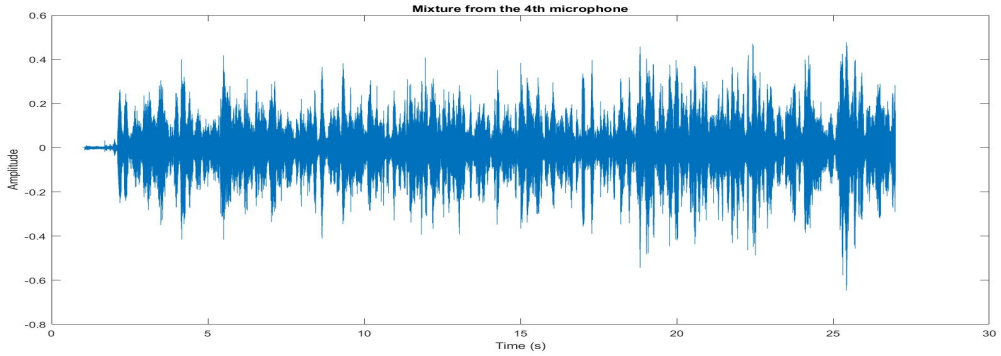
From a subjective perspective, the algorithm demonstrates successful separation of the source speech signals by listening to the outputs. these observations can be made:

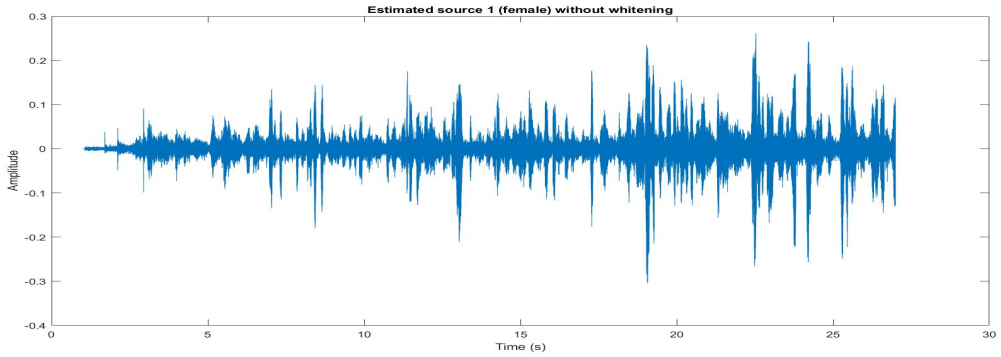
- When listening to the outputs; one can notice that NG IVA with whitening takes less to separate the sources (after about 4 seconds) whilst the standard adaptive NG IVA takes longer to start separating (after about 8.5 seconds). However this comes at the cost of processing time as mentioned at the beginning.

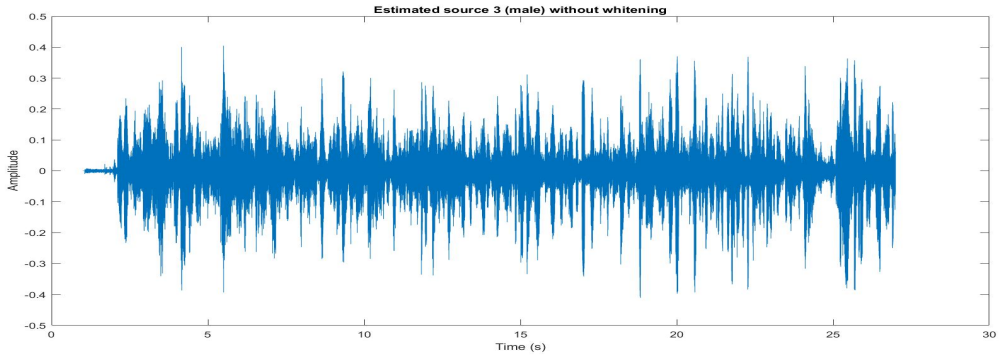
- From the figure 4.21, we can distinguish the sources' signature from the mixture, and one can see that the sources have been estimated with the correct scaling.

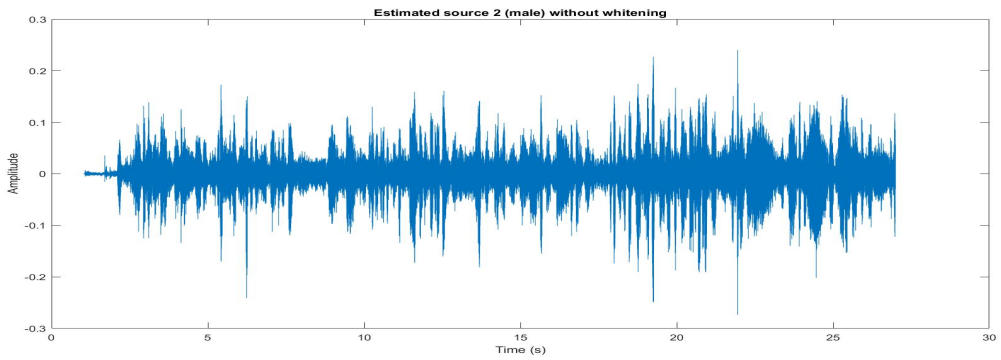
- Whitening improved sound quality and achieved better separation by listening to both of them.

Case 2: Three speakers, two male and one female: Down below the mixtures and separated signals for Three speakers case.









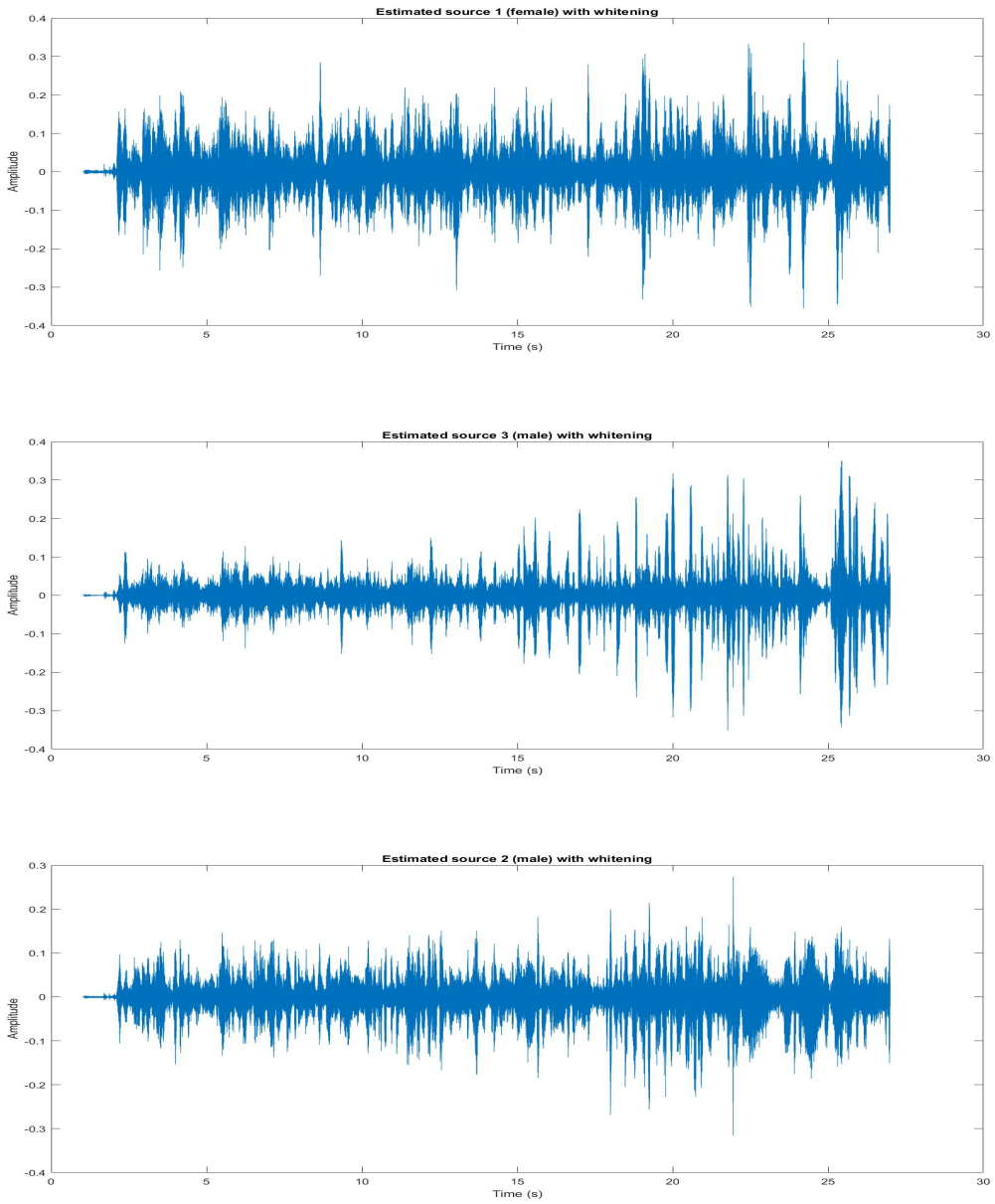


Figure 4.22: 2nd experiment with real-world acoustic recordings: three sources mixture recorded by the 4th UMA-16 v2 microphone array, the separation results of NG IVA with whitening and without whitening algorithm.

Again the algorithm separated the signals fairly well, one can distinguish which source is the heard audio, however when dealing with three sources case, the algorithm suffered a bit and one can hear the other two sources in the background. Unlike when using generated mixtures where it did successfully separate each one of them without background interference (or too little to be noticed). The signals waveforms are shown in figure 4.22.

4.4 Conclusion

In this chapter, our focus was on the UMA-16 v2 microphone, emphasizing its capabilities and highlighting its significance in our work. We provided a comprehensive overview of its physical characteristics and electronic components, shedding light on its advanced features and functionalities.

Furthermore, we presented a detailed operating mode guide for the UMA-16 v2 microphone, offering step-by-step instructions on how to utilize it effectively for recording real-world tests. By conducting experiments using real-life signals, we aimed to validate and reinforce the findings obtained in the previous chapter, this time in a practical setting. The results obtained from the real-life signal recordings once again affirmed the effectiveness of the separation algorithm. The algorithm demonstrated its ability to successfully separate and distinguish sources in complex and dynamic acoustic environments, as observed in the recorded signals using the UMA-16 v2 microphone.

The successful application of the separation algorithm to real-life signals underscores its practical viability and robustness. It highlights the algorithm's adaptability and reliability in various scenarios.

Conclusion

In this finale project, we have assessed the Blind Source Separation problem. At first, we gave a brief introduction to the BSS problem with its mathematical formulation and its ambiguities along with related work on BSS for speech signals both non adaptive and adaptive versions.

We studied the adaptive natural gradient based Independent Vector Analysis and proposed a modification to the classic adaptive NG IVA, that is an adaptive data whitening which allows us to use more microphones than sources which allows us to exploit spatial diversity of the array we afterwards implemented both of them writing the codes from scratch as they are not available in open source, the results show that whitening improves separation performances noticeably and speeds up convergence of the separation as shown experimentally. However, this effects heavily the computational cost of the algorithm, as it takes longer to run the code (twice the time of the standard adaptive NG IVA).

Finally we got to use and see how powerful is the UMA-16 v2 array and to test it in real world scenarios which resulted in good results.

Future work

Despite the encouraging results shown by the algorithm, however, one major problem which the introduction of whitening causes is the computational cost (the run time of the algorithm) which mainly due to the eigenvalue decomposition at each frame. for the future we will be working on the two following point.

- Reduce the run time of adaptive NG IVA with whitening, that is by working on adaptive eigenvalue decomposition.

- Implement the algorithm on an embedded system which will allow us to have an independent system that can serve as a pre-processing for several applications.

Bibliography

[1] J. Hérault and C. Jutten, "Blind separation of sources, part i: An adaptive algorithm based on neuromimetic architecture", *Signal Processing*, vol. 10, no. 3, pp. 149–158, 1986.

[2] E. C. Cherry, "Some experiments on the recognition of speech, with one and with two ears", *The Journal of the Acoustical Society of America*, vol. 25, no. 5, pp. 975–979, 1953.

[3] R. Martin, *Speech Separation by Humans and Machines*. Springer Science Business Media, 2005.

[4] D. E. Johnson and D. E. Dudgeon, *Array Signal Processing–Concepts and Techniques*. Prentice Hall, 1993.

[5] P. Mermelstein, "Articulatory model for the study of speech production", *Journal of the Acoustical Society of America*, vol. 53, no. 4, pp. 1070–1082, 1973.

[6] M. H. Hayes, "Recursive least squares", in *Statistical Digital Signal Processing and Modeling*, Wiley, 1996, pp. 541–551, ISBN: 0-471-59431-8.

[7] S. Haykin, *Adaptive Filter Theory*. Prentice Hall, 2002, ISBN: 0-13-048434-2.

[8] L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals*. Prentice Hall, 1993.

[9] H. Buchner, R. Aichner, and W. Kellermann, "Blind source separation for convolutive mixtures exploiting nongaussianity, nonwhiteness and nonstationarity", in *IEEE International Workshop on Acoustic Echo and Noise Control (IWAENC)*, IEEE, 2003, pp. 275–278.

[10] R. Crochiere, "A weighted overlap-add method of short-time fourier analysis/synthesis", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 28, no. 1, pp. 99–102, 1980.

[11] L. Tong, Y. Inouye, and R. Liu, "Waveform-preserving blind estimation of multiple independent sources", *IEEE Transactions on Signal Processing*, vol. 41, no. 7, pp. 2461–2470, 1993.

[12] 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*, American Institute of Physics Inc., USA, 1986, pp. 206–211, ISBN: 088318351X.

[13] J.-F. Cardoso, "Blind identification of independent components with higher-order statistics", in *Workshop on Higher-Order Spectral Analysis*, 1989, pp. 157–162. DOI: 10.1109/HOSA.1989.735288.

[14] P. Comon, "Separation of stochastic processes", in *Workshop on Higher-Order Spectral Analysis*, 1989, pp. 174–179. DOI: 10.1109/HOSA.1989.735291.

[15] P. Comon, "Independent component analysis, a new concept?", *Signal Processing*, vol. 36, no. 3, pp. 287–314, 1994.

[16] A. J. Bell and T. J. Sejnowski, "An independent component analysis framework for blind signal separation", *Advances in neural information processing systems*, pp. 525–532, 1995.

[17] S.-i. Amari, A. Cichocki, and H. Yang, "A new learning algorithm for blind signal separation", in *Advances in Neural Information Processing Systems*, vol. 8, 1995.

[18] A. Hyvärinen and E. Oja, "A fast fixed-point algorithm for independent component analysis", *Neural Computation*, vol. 9, no. 7, pp. 1483–1492, 1997.

[19] A. Hyvärinen and E. Oja, "Fast and robust fixed-point algorithms for independent component analysis", *IEEE Transactions on Neural Networks*, vol. 10, no. 3, pp. 626–634, 1999. DOI: 10.1109/72.761722.

[20] J.-F. Cardoso and A. Souloumiac, "Blind beamforming for non-gaussian signals", *IEE Proceedings F - Radar and Signal Processing*, vol. 140, no. 6, pp. 362–370, 1993.

[21] J.-F. Cardoso, "Multidimensional independent component analysis", in *1998 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, vol. 4, 1998, pp. 1941–1944.

[22] 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*, vol. 45, no. 2, pp. 434–444, 1997. DOI: 10.1109/78.554307.

[23] L. Tong, V. Soon, Y.-F. Huang, and R. Liu, "Amuse: A new blind identification algorithm", in *IEEE International Symposium on Circuits and Systems*, vol. 3, 1990, pp. 1784–1787. DOI: 10.1109/ISCAS.1990.111981.

[24] 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*, vol. 12, no. 5, pp. 530–538, 2004.

[25] T. Kim, I. Lee, and T.-W. Lee, "Independent vector analysis: Definition and algorithms", in *2006 Fortieth Asilomar Conference on Signals, Systems and Computers*, IEEE, 2006, pp. 1393–1396.

[26] I. Lee, T. Kim, and T.-W. Lee, "Fast fixed-point independent vector analysis algorithms for convolutive blind source separation", *Signal Processing*, vol. 87, no. 8, pp. 1859–1871, 2007.

[27] L. Berrah and N. Mendjel, *Blind speech separation: Algorithm improvement and implementation using raspberry pi with UMA-8-SP mic array testbed*, Electronics Department, Ecole Nationale Polytechnique, Engineering final project, 2022.

[28] A. Belouchrani, N. Mendjel, L. Berrah, and S. Tebache, "Independent vector analysis based mimo deconvolution: Exploiting spatial diversity through back projection", in *22nd IEEE Statistical Signal Processing Workshop*, Hanoi, Vietnam, Jul. 2023.

[29] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization", *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[30] 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*, vol. 24, no. 9, pp. 1626–1641, 2016.

[31] L. Parra and C. Spence, "Convolutive blind separation of non-stationary sources", *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 3, pp. 320–327, 2000.

[32] T. Kim, "Real-time independent vector analysis for convolutive blind source separation", *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 7, pp. 1431–1438, 2010.

[33] T. Taniguchi, N. Ono, A. Kawamura, and S. Sagayama, "An auxiliary-function approach to online independent vector analysis for real-time blind source separation", in *2014 4th Joint Workshop on Hands-free Speech Communication and Microphone Arrays (HSCMA)*, IEEE, 2014, pp. 107–111.

[34] T. Nakashima and N. Ono, "Inverse-free online independent vector analysis with flexible iterative source steering", in *2022 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, IEEE, 2022, pp. 749–753.

[35] J. P. Nadal and N. Parga, "Non-linear neurons in the low-noise limit: A factorial code maximizes information transfer", *Network*, vol. 4, pp. 295–312, 1994.

[36] Z. Roth and Y. Baram, "Multidimensional density shaping by sigmoids", *IEEE Trans. on Neural Networks*, vol. 7, no. 5, pp. 1291–1298, 1996.

[37] A. Bell and T. Sejnowski, "An information maximization approach to blind separation and blind deconvolution", *Neural Computation*, vol. 7, pp. 1129–1159, 1995.

[38] H.-L. Thi and C. Jutten, "Blind source separation for convolutive mixtures", *Signal Processing*, vol. 45, no. 2, pp. 209–229, 1995.

[39] J. Xi and J. Reilly, "Blind separation and restoration of signals mixed in convolutive environment", in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, IEEE, 1997, p. 1327.

[40] K. Torkkola, "Blind separation of convolved sources based on information maximization", in *1996 IEEE Signal Processing Society Workshop, Neural Networks for Signal Processing*, IEEE, 1996, pp. 423–432.

[41] A. Hyvärinen and U. Köster, "Fastisa: A fast fixed-point algorithm for independent subspace analysis", in *14th European Symposium on Artificial Neural Networks*, 2006, pp. 371–376.

[42] J. Benesty, S. Makino, J. Chen, *et al.*, "Frequency-domain blind source separation", *Speech enhancement*, pp. 299–327, 2005.

[43] S. Makino, T.-W. Lee, and H. Sawada, *Blind Speech Separation*. Springer, 2007, vol. 615.

[44] H. Brehm and W. Stammler, "Description and generation of spherically invariant speech-model signals", *Signal Processing*, vol. 12, no. 2, pp. 119–141, 1987.

[45] 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*, vol. 15, no. 1, pp. 70–79, 2007.

[46] K. Matsnoka, "Minimal distortion principle for blind source separation", *SICE 2002. Proceedings of the 41st SICE Annual Conference*, vol. 41, Sep. 2002. DOI: 10.1109/SICE.2002.1195729.

[47] S.-i. Amari, T.-P. Chen, and A. Cichocki, "Nonholonomic orthogonal learning algorithms for blind source separation", *Neural computation*, vol. 12, no. 6, pp. 1463–1484, 2000.

[48] A. Cichocki, R. Unbehauen, L. Moszczynski, and E. Rummert, "A new on-line adaptive algorithm for blind separation of source signals", in *Proc. of 1994 Int. Symposium on Artificial Neural Networks ISANN-94*, 1994, pp. 406–411.

[49] 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*, vol. 43, pp. 894–906, 1996.

[50] S.-i. Amari and J.-F. Cardoso, "Blind source separation—semi-parametric statistical approach", *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2692–2700, 1997.

[51] S.-i. Amari and M. Kawanabe, "Estimating functions in semi-parametric statistical models", in *Estimating functions*, I. V. Basawa, V. P. Godambe, and R. L. Taylor, Eds., vol. 3, Cambridge University Press, 1997, pp. 65–81.

[52] S.-i. Amari and M. Kawanabe, "Information geometry of estimating functions in semiparametric statistical models", *Bernoulli*, vol. 3, pp. 29–54, 1997.

[53] P. J. Bickel, C. A. J. Klaassen, Y. Ritov, and J. A. Wellner, *Efficient and adaptive estimation for semi-parametric models*. Baltimore, MD: Johns Hopkins University Press, 1993.

[54] *Pyroomacoustics*, `https://pyroomacoustics.readthedocs.io/en/pypi-release/`, Accessed: 10-04-2023.

[55] M. R. Schroeder, "New method of measuring reverberation time", *The Journal of the Acoustical Society of America*, vol. 37, no. 3, pp. 409–412, 1965.

[56] C. Brown, *t60.m*, MATLAB Central, Available at `http://de.mathworks.com/matlabcentral/fileexchange/1212-t60-m`, 2002.

[57] International Organization for Standardization, *ISO 3382-2 (2008)*, Acoustics – Measurement of room acoustic parameters – Part 2: Reverberation time in ordinary rooms, Switzerland: ISO, 2008.

[58] C. Févotte, R. Gribonval, and E. Vincent, "BSS EVAL toolbox user guide", IRISA, Rennes, France, Tech. Rep. Technical Report 1706, 2005.

[59] C. Févotte, R. Gribonval, and E. Vincent, "Bss_eval toolbox user guide–revision 2.0", 2005.

[60] miniSDP Homepage, *About us*, `https://www.minidsp.com/aboutus/aboutus`, 2009.

[61] MINIDSP, *MINIDSP UMA-16 Microphone Array specifications*, `https://www.minidsp.com/products/usb-audio-interface/uma-16-microphone-array`, 22-05-2023.

[62] Wikipedia, *Voice frequency-frequency band*, `https://shorturl.at/nuHI7`, 2023.

[63] K. electronics, *Datasheet*, `https://shorturl.at/rxCU0`, 2015.

[64] *What is ASIO?*, `https://www.r-tt.com/technology-articles/what-is-asio.html`, Accessed: 14-05-2023.