POLITECNICO DI MILANO

Facoltà di Ingegneria

Scuola di Ingegneria Industriale e dell'Informazione

Dipartimento di Elettronica, Informazione e Bioingegneria

Master of Science in

Computer Science and Engineering


# Source extraction using informed independent vector analysis

Supervisor:

PROF. GIUSEPPE BERTUCCIO

Assistant Supervisor:

PROF. EMANUEL A.P. HABETS

Master Graduation Thesis by:

LUIS GERMAIN ARANGO

Student Id n. 850564

Academic Year 2017-2018

# ACKNOWLEDGMENTS

I would like to thank prof. Habets for giving me the opportunity to work on this thesis.

I also want to thank prof. Bertuccio for his support and to Soumitro for his precious help and suggestions.

A special thank goes to my mother which has always been at my side, instructing me by words and examples and supporting me since the day I was born.

Another special thank goes to Sara for her constant motivation and support through my whole thesis.

# CONTENTS

# LIST OF FIGURES



# LIST OF TABLES

# ABSTRACT

Blind audio source separation aims at extracting a certain number of acoustic source signals from a set of observation signals; the term "blind" comes from the fact that in the separation process no (or very little) information about the sources or the mixing system is available. The interaction of the acoustic signals with the surrounding environment causes time delays and reverberations which involve long filter lengths to be estimated in the time domain. Although the convolutive mixtures can be separated efficiently by Frequency Domain Independent Component Analysis (FDICA) algorithms, all ICA based algorithms suffer from a permutation ambiguity, which for FDICA algorithms is present at every frequency bin. To solve this problem, the independent vector analysis (IVA), which employs a multivariate dependency model to capture inter-frequency dependencies, has been proposed.

In this thesis we focus on an extension of IVA, called Supervised Independent Vector Analysis, in which the multidimensional source model of IVA is extended by adding pilot components which are statistically dependent on the desired sources. These pilot component signals act as a prior knowledge which enforces the natural gradient to converge in a limited solution space: thanks to this property, we are able to perform Audio Source Extraction, i.e. separating and extracting one particular desired audio source. We investigate the Supervised IVA and the influence of the pilot components on the convergence of the algorithm, starting by some simple oracle models for the pilot components and, after assessing the improvement provided to the IVA by adding the pilots, we implement a version in which a Convolutional Neural Network (CNN) Localizer is used to detect the Direction-of-Arrival (DOA) and to track the activity of the sources so that the correspondent pilot component can be added to the basic IVA. We name this algorithm Informed Independent Vector Analysis (IIVA). Our model is simple and flexible: we are able to improve the extraction of a speech signal which direction-of-arrival is approximately known.

We simulate realistic scenarios to asses the performances of the proposed method: the experimental results show that the convergence is stable with respect to the IVA and the objective performances are in

line with those of an existing source extraction algorithm in the literature. We also show that the frequency components are separated and included in the solution with high fidelity. Furthermore, we prove that our algorithm is able to rapidly converge, allowing a real-time implementation and thus it can be used for several real world applications.

# ESTRATTO

La separazione alla cieca di sorgenti audio mira a estrarre un certo numero di segnali sorgente da un insieme di segnali di osservazione; il termine "alla cieca" è utilizzato perché nessuna (o poca) informazione a riguardo delle sorgenti o del sistema di mescolamento è disponibile. L'interazione del segnale sonoro con l'ambiente circostante causa ritardi temporali e riverberazione che richiedono la stima nel dominio temporale di filtri le cui dimensioni sono molto grandi. Nonostante le miscelazioni convolutive possano essere separate efficientemente da algoritmi di analisi delle componenti indipendenti (ICA) nel dominio della frequenza, tutti gli algoritmi ICA soffrono della ambiguità delle permutazioni delle soluzioni che nel caso di ICA nel dominio della frequenza si presenta ad ogni banda di frequenze.

Per risolvere questo problema, è stata proposta l'analisi dei vettori indipendenti (IVA) che utilizza un modello di dipendenza multivariata per catturare dipendenze inter-frequenziali. In questa tesi ci concentriamo su un'estensione di IVA, chiamata IVA supervisionata (SIVA), nella quale il modello multidimensionale delle sorgenti viene esteso aggiungendo i cosiddetti componenti piloti che sono statisticamente dipendenti dalle sorgenti audio desiderate. I segnali dei piloti attuano come conoscenza a priori e forzano il gradiente naturale a convergere in uno spazio di soluzioni limitate: grazie a ciò, siamo in grado di eseguire l'estrazione di sorgente, cioè estrarre una particolare sorgente desiderata. Investighiamo SIVA e l'influenza dei piloti sulla convergenza dell'algoritmo, iniziando da semplici modelli oracolo e, verificato il miglioramento rispetto all'IVA basico, implementiamo una versione che chiamiamo analisi informata dei vettori indipendenti (IIVA), in cui un localizzatore basato su rete neurale convolutiva viene utilizzato per rilevare la direzione d'arrivo delle sorgenti, tracciandone l'attività in modo che il pilota corrispondente possa essere aggiunto al basico IVA. Il nostro modello è semplice e flessibile: siamo in grado di migliorare l'estrazione di un segnale vocale la cui direzione di arrivo è approssimativamente nota.

Nel nostro lavoro, simuliamo scenari realistici per verificare le prestazioni del metodo proposto: i risultati sperimentali mostrano che la convergenza è stabile rispetto a quella dell'algoritmo basico e le performance oggettive sono in linea con quelle di un altro algoritmo

presente in letteratura. Mostriamo anche che le componenti frequenziali vengono separate ed incluse nella soluzione con alta fedeltà. Inoltre l'algoritmo è in grado di convergere rapidamente, permettendo un'implementazione in tempo reale e può quindi essere utilizzato per diverse applicazioni nel mondo reale.

# 1

# INTRODUCTION

Speech signals in real-world are subject to what in the literature is referred to as distortion. A *distortion* is a modification (usually unwanted) of the waveform of a signal (in our case an audio signal) due to the interaction of the propagating wave with real-world elements: reflections on walls and objects creates reverberation, other sound sources create interferences and environmental sounds (e.g. wind, traffic due to cars) generate noise. Due to the enlisted distortions, microphone signals need to be cleaned before transmission, storage or reproduction.

The term Speech Enhancement is used to indicate the set of techniques and algorithms used to improve the intelligibility and the perception of speech signals by making use of signal processing tools. For example, in a phone call it is necessary to enhance the voice of the desired speaker source over other interfering sources and over the noises around him prior the transmission to the listener at the other end. In a video-conference scenario, the cancellation of echo-feedback due to the loudspeakers emitting the voice captured at the other side of the communication system, the elimination of the reverberation of the target source due to the walls and in general a target speech enhancement are crucial steps for a correct communication. These methods need to be efficient and robust. If we think about an astronaut which needs to communicate with the base when exploring an unknown space or to formula-1 drivers which may need to report a failure or a problem they are facing to their team, then it is easy to understand how important and critical is the research and the improvement of speech enhancement techniques.

With the advent of artificial-intelligent systems which aim to help users by taking as input commands given through their voice, speech enhancement is a required pre-processing step for automatic speech recognition. There are two common approaches to face this problem:

one is the use of microphone array processing techniques (*Beamforming* or *Spatial Filtering*) and the other is the use of *Blind Source Separation* techniques.

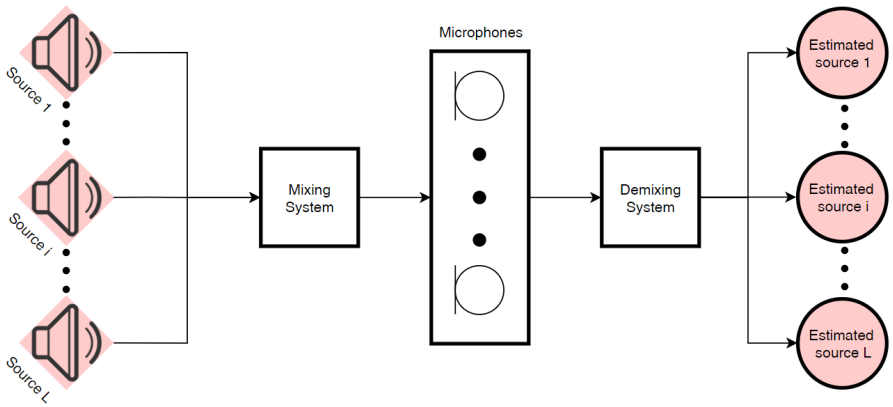
Beamforming techniques, such as [37], [39], exploit the combination of elements in an antenna array in such a way that signals at particular angles are enhanced by constructive interference while others are canceled by destructive interference. Beamformers can be classified in two main groups: conventional (or fixed) and adaptive. Conventional beamformers use a fixed set of weightings and time-delays to combine the signals from the sensors in the array, primarily using only information about the location of the sensors in space and the DOA of the desired source(s). In contrast, adaptive beamforming techniques generally exploit some properties, generally Second Order Statistics, of the signals received by the array, typically to improve rejection of unwanted signals from other DOAs.

Blind Source Separation is the separation of a set of source signals from a set of mixed signals (to which we refer as mixture); the term blind is used because the sources are determined without the use of any (or very little) prior knowledge of the data structure and the mixing process, through the application of (typically) an internal measure. In this thesis we focus on blind audio source separation, i.e. the separation of acoustic source signals. This problem is in general underdetermined (there are more sources than sensors) but useful solutions can be derived under a variety of conditions: for example, some BSS methods seek to narrow the set of possible solutions minimizing the risk of excluding the desired solution. Famous and widely pursued approaches are given by Principal Component Analysis [26] and Independent Component Analysis [20] where one seeks source signals characterized by minimal correlation or maximal independence in a probabilistic or information-theoretic sense. A second approach is given by Non-negative Matrix Factorization (NMF) [31] in which structural constraints are imposed on the source signals: a common theme in this approach is to impose some kind of low-complexity constraint on the signal, such as sparsity, in some basis for the signal space. This approach can be particularly effective if one requires not the whole signal, but merely its most salient features.

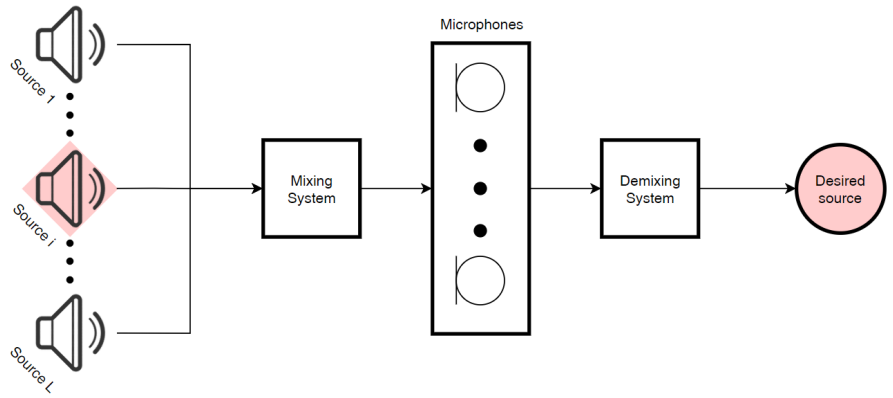
In this thesis we focus on BSS techniques, in particular on the *Independent Vector Analysis* proposed by *Kim et al.* [30] in which it is assumed, similarly to ICA (IVA can be seen as a generalization of ICA), that the "subcomponents" in a mixture are non-Gaussian signals and that they are statistically independent from each other. The model proposed in IVA makes possible to avoid the indeterminacy of permutation inherent to the ICA algorithms which should be corrected to obtain a proper separation of the signal in the time domain. In the

IVA formulation, the permutation problem of the ICA solutions over different frequency bins, also known as the local permutation problem, given by the Frequency Domain ICA (FDICA) [38] is completely avoided by modeling the sources probability distribution functions as multivariate super-Gaussian distribution [30] and the algorithm.

IVA has captured the attention of many researchers for the above mentioned characteristic and has been investigated and extended to optimize its convergence properties. Some of the most interesting extensions of the algorithm propose to declare a cost function using an auxiliary-function technique [40] which is free from step-size parameter tuning, or to add geometrical constrains to the standard IVA algorithm [27] so that the desired speech signal is always delivered at the output of the corresponding separation filter. We focus our work on the investigation of an interesting extension, called Supervised IVA [34], which extend the multidimensional source model of IVA by adding the so called pilot components which are statistically dependent on the target and/or the noise sources. These pilot component signals act as a prior knowledge which enforces the natural gradient to converge in a limited solution space, without imposing any explicit constraint to the demixing system. Since the pilots need to be one for each of the sources to be separated, we are able to perform Source Extraction - the difference between Source Extraction and Source Separation is explained in the next section - by focusing on the pilot component of the desired source. In chapter 4 we proposed different models for the pilot components and the activation procedure of these pilots. To validate the SIVA [34], we decided at first to develop an oracle version of the algorithm in which the activation of the pilot is decided by evaluating the energy content of the desired clean source signal at a certain time-frame. We investigated some simple oracle models for the pilot components by adding information coming from the noiseless desired signal. Once assessed the boost that the pilots provide to the standard IVA, we implemented a version, which we named Informed Independent Vector Analysis, in which a Convolutional Neural Network (CNN) Localizer [17] was used to detect the Direction-of-Arrival (DOA) of the sources at a certain time frame $n$ so that if the DOA of the source is detected then the source is considered as active and the correspondent pilot component would influence the basic IVA. Our model is simple and flexible: we are able to improve the extraction of a speech signal which direction-of-arrival is approximately known and the only needed parameter is the Region-of-interest of the desired source in terms of angular position with respect to the set of sensors used.




**(a)** Blind Source Separation




**(b)** Blind Source Extraction

**Figure 1.1:** (a) shows a schema representing a typical BSS structure while (b) represents a typical BSE structure. In both diagrams pink is used to mark the sources to be estimated: the main difference between BSS and BSE is the number of outputs at the end of the processing chain.

## 1.1 SOURCE SEPARATION VS SOURCE EXTRACTION

Blind source separation (BSS) is a major area of research in signal processing with a vast literature regarding various areas, including wireless communication, biomedical engineering, image processing and also acoustic signal processing which is the area in which we focus on.

In BSS, given a set of combinations of a certain number of source signals (mixture), the objective is that of separating and reconstructing the source signals which compose the mixture. The separated signals are approximations of the original source signals. The term blind comes from the fact that the source separation process is accomplished without the use of any (or very little) prior knowledge about the source signals and/or the mixing process. One particular

application of BSS in audio is to extract a desired audio source from mixtures involving noise, background or unwanted sources.

Compared to BSS, the objective of blind source extraction (BSE) is that of extracting only one of the sources, to which we refer to as desired source, among the sources originating the mixture. The main difference between BSS and BSE is in the number of outputs of the processing chains: in the case of BSS the number of outputs is equal to the number of sources which compose the observed mixture signals while in BSE the output is usually one or anyway less than the number of source signals composing the mixture. In Fig. 1.1 we show the difference between a BSS structure (Top) and a BSE structure (Bottom) in which the desired source is only one of the source signals.

Since in BSE the aim is that of recovering only a single source from the observation set, this leads to reduced computational complexity and more flexibility compared to BSS. This is useful in many practical applications in which we desire to extract only one precise source such as in mobile and, in particular, in hands-free communication systems where the background noises may in fact be stronger, even much stronger sometimes, than the desired signal and furthermore composed by several signals to which we are not interested a.k.a. unwanted sources, e. g. motor noise, other passenger voices, environmental sounds, etc.

## 1.2 OUTLINE

In *Chapter 2* we provide an overview of the main theoretical background on which our work is based. First we describe the Blind Source Separation problem, starting from the mixture models used for the formal problem formulation, then we briefly describe the statistical properties on which most of the BSS algorithm use to perform the separation. In this chapter, we also shortly describe the Independent Component Analysis (ICA), which is a special case of BSS, which provides the basis for the development of the IVA, and we discuss the main limitations of the algorithms based on ICA which motivates the extension to the IVA.

In *Chapter 3* we introduce the concept of Independent Vector Analysis with a formal mathematical description and explanation. In this chapter we describe the difference and novelty of IVA with respect to ICA, describing the advantages of using a multi-variate distribution to model the source priors. We also explain the gradient optimization used to minimize the multi-variate cost function and we describe the issues of the standard IVA. Furthermore, we describe some interesting extensions [40], [27], [34], of the IVA which aim to solve some of

the open issues in IVA and provide the basis and motivation for the method proposed in this thesis.

In *Chapter 4* we describe the proposed approach, starting from the early stages of the development, by using some prior information about the sources to be estimated, to arrive to the version proposed in this thesis. In this chapter we also describe the multi-channel CNN localizer and how we used it in our work, and at the end of the chapter we propose some models for the pilots components.

In *Chapter 5* we describe the settings of the experiments and the performance criteria that we use. We start the experiments by comparing the early oracle versions of the proposed methods with the CNN-based version. Once established the effectiveness of the CNN-based version, we investigate the influence of the parameters on the performances of the proposed method and we finally compare the algorithm with the CIVA [27].

In *Chapter 6* we draw our conclusions based on the results obtained, and we discuss some possible future developments and applications.

# 2

# BACKGROUND

## 2.1 BLIND SOURCE SEPARATION

Consider a situation in which there are a certain number of signals emitted by physical elements or sources. These sources might be different areas of the brain generating electric signals, devices emitting radio-waves or people speaking in a room producing acoustic (speech) signals. Assume the latter case, i.e. the acoustic case, and assume there are a certain number of sensors or receivers: the sensors are placed in different spatial locations so that each sensor observes a mixture of the original source signals with some different weights due to the different propagation path of the acoustic waves traveling from the sources to the receivers. Given a set of mixed signals received by the sensors, to which we refer to as *mixture*, we would like to retrieve and separate each of the sources which generates the mixture signals. This problem is known as *Blind Source Separation*: the term blind comes from the fact that the sources are determined without the use of any (or very little) prior knowledge of the source signals and the mixing process.

The BSS problem, in real world applications, is generally underdetermined, i. e. there are more sources than sensors, but useful solutions can be derived under a variety of conditions: methods for BSS generally seek to narrow the set of possible solutions by minimizing the risk of excluding the desired ones. The problem of source separation in the auditory domain has received a lot of attention and numerous approaches have been developed.

Famous and widely pursued approaches are given by Principal Component Analysis (PCA) [26] and Independent Component Analysis (ICA) [20] where one seeks source signals characterized by minimal correlation or maximal independence in a probabilistic or infor-

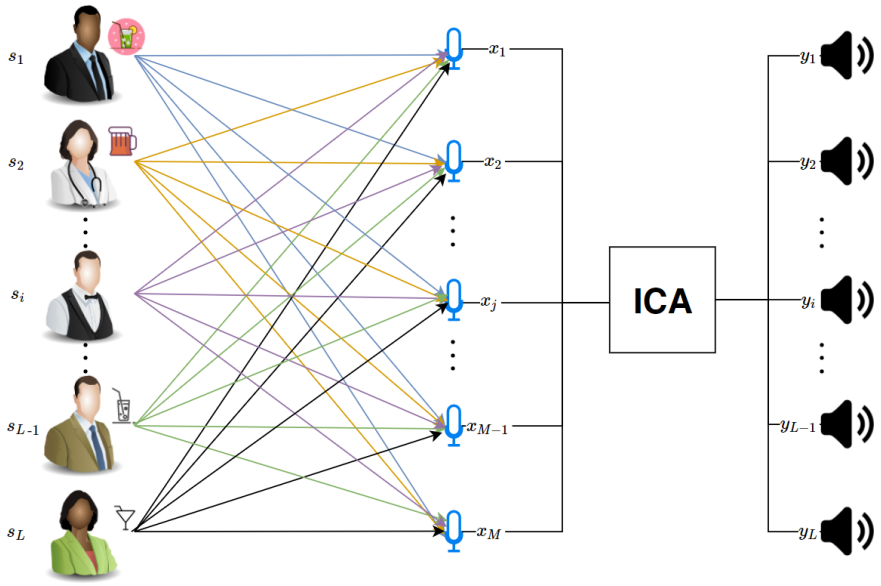



**Figure 2.1:** A representation of the cocktail party problem: L people in a room speak at the same time and, given the M observations of the superposition of their voices, the objective is that of separating the sources and hear them individually.

mation theoretic sense. Another approach is given by Non-negative Matrix Factorization (NMF) [31] in which structural constraints are imposed on the source signals: generally the approach is to impose some constraint on the signal, such as sparsity, so that the complexity of the solution space is reduced. NMF can be described as a group of algorithms a matrix $\mathbf{V}$ is factorized into (usually) two matrices $\mathbf{W}$ and $\mathbf{H}$, with the property that all three matrices have no negative elements: this non-negativity makes the resulting matrices easier to inspect.

In this thesis we focus on ICA, in particular to a generalization named *Independent Vector Analysis* [30] which is discussed in the following sections.

### COCKTAIL PARTY

To give a clear example of a task to be solved in the BSS scenario, consider the famous *Cocktail party problem* defined by Cherry [18]. Imagine being at a cocktail party where a number of people is talking simultaneously inside a room and you try to follow one conversation: for the human brain it is an easy task to recognize the various sources and to focus on a specific one, filtering the unwanted sources, but this is a really challenging problem in digital signal processing.

To better describe the scenario in terms of digital signal processing, we need to better define the environment. Let us assume that

there are $L$ sources and $M$ microphones recording the surrounding scenario: each microphone catches a superposition of the $L$ sources as shown in fig Fig. 2.1. The problem is to try to separate the sources and properly hear them individually: the idea is either to identify the mixing matrix which describes how the different conversations got mixed at each different microphone given just a set of observations (mixture of signals) in order to reconstruct the source signals or to directly estimate the de-mixing matrix which gives us the separated sources as output. Unless some assumptions are made, this is an *ill-posed problem*.

The most common assumption made to solve the problem is to consider that the source signals are statistically independent, i. e. knowing the value of one of the sources does not give any information about another. The methods which rely on this assumption are referred to as *Independent Component Analysis* (ICA): these are statistical techniques which aim to decompose a complex data set into independent sub sets. It can be shown that under some reasonable conditions, if the ICA assumption holds, then the source signals can be recovered up to permutation and scaling. The cocktail party problem has been deeply investigated since 80's and it is still a hot topic in the research community.

### 2.1.1 *Mixture models*

Given the set of sources signals at the discrete time $t$, a mixture of the $L$ source signals given by $\mathbf{s}(t) = [s_1(t), s_2(t), \cdots, s_L(t)]^T$ are captured by an array of $M$ microphones. The captured signals are referred to as *observations* and they are indicated with $\mathbf{x}(t) = [x_1(t), x_2(t), \cdots, x_M(t)]^T$.

In a reverberant environment, source signals are filtered. The sensors used to capture the signal add a noisy component $v(t)$ to the observed signals. The $i$th observation signal $x_i(t)$ at time $t$ is given by

$$x_i(t) = \sum_{j=1}^{L} \sum_{\tau=0}^{T-1} h_{ij}(\tau) s_j(t-\tau) + v_i(t), \tag{2.1}$$

where $h_{ij}(t)$ is a time-domain transfer function from the $j$th source to the $i$th receiver, which has a length of $T$ samples, $s_j(t)$ is the $j$th source signal at time $t$, and $L$ is the number of sources. The coefficients $h_{ij}(\tau)$ are usually a function of time because they might be subject to variations over time; however, for simplicity, the mixing model is frequently assumed to be stationary. Theoretically the filters may have infinite length and they could be implemented using *Infinite Impulse Response (IIR)* filters, but in practice they can be modeled with *Finite Impulse Response (FIR)* filters, simply considering $T < \infty$.

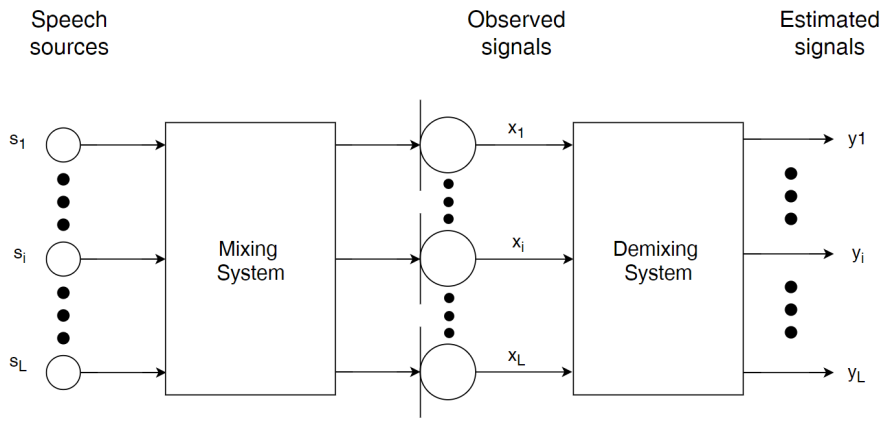



**Figure 2.2:** The typical structure of a determined Blind Source Separation problem (L=M).

Equation (2.1) can be written in a more compact form using the matrix notation:

$$x_i(t) = \sum_{\tau=0}^{T-1} \mathbf{H}_\tau s_j(t-\tau) + v_i(t). \tag{2.2}$$

where $\mathbf{H}_\tau$ represents the $M \times L$ matrix containing the FIR polynomial coefficients at time $\tau$. For simplicity of notation, in the following we consider the observation signals as *stationary* (and this is acceptable for short-time intervals) and *noise-free*: Nevertheless, it is important to notice that, in any real world applications, noise is always present due to the electronic components of the sensors.

### INSTANTANEOUS SOURCE SEPARATION

Assuming an instantaneous mix of the signals, e.g. the signals which are captured by the sensors arrive at the same time without being filtered, then it is possible to write (2.2) as

$$\mathbf{x}(t) = \mathbf{H}\mathbf{s}(t), \tag{2.3}$$

where $\mathbf{H}$ represents the $M \times L$ matrix containing the mixing coefficients. The objective is to find an estimate of the matrix $\mathbf{H}$, which is considered a stable and stationary system, so that we can compute its inverse (or pseudo-inverse in the case of under-determined or over-determined systems) $\mathbf{G}$, also known as the *demixing matrix*, to obtain, at the output $\mathbf{y}(t) = [y_1(t), \cdots, y_N(t)]^T$ of our processing chain, an estimation of the sources. In Fig. 2.2 is illustrated a typical structure of a determined system. The output signal $\mathbf{y}(t)$ is the reconstructed source signal $\hat{\mathbf{s}}(t)$, so that

$$\mathbf{y}(t) = \hat{\mathbf{s}}(t) = \mathbf{G}\mathbf{x}(t) = \mathbf{G}\mathbf{H}\mathbf{s}(t). \tag{2.4}$$

Assuming that a matrix $\mathbf{G}$ exists such that $\mathbf{GH} = \mathbf{I}$, in the case in which we perfectly succeeded on the estimation of the mixing matrix, we would have that the sources are perfectly reconstructed, i. e. $\hat{\mathbf{s}}(t) = \mathbf{y}(t) = \mathbf{s}(t)$.

This is an ideal result which is not feasible due to different causes, the main ones are the following:

- the noise introduced by the sensors and other environmental sounds;
- the delays due to the finite speed propagation of the sound in the air (the instantaneous model is not valid in most of real world applications);
- the reverberation due to the reflections of the sound waves;
- the length of the FIR filters used to model the mixing channel which can have more than 2000 taps, leading to reduced efficiency;
- the fact that we may try to estimate more sources than the number of sensors we are using, leading to an under-determined system.

Thus, we can only try to achieve an approximated solution, so that $\hat{\mathbf{s}}(t) = \mathbf{y}(t) \approx \mathbf{s}(t)$.

The instantaneous mixture has been investigated intensively and various algorithms, e. g. the natural gradient based algorithm [5], the decorrelation-based BSS [19] among others, have been developed to deal with it.

### CONVOLUTIVE SOURCE SEPARATION

In many real-world applications the sources are said to be *convolutively* mixed and this is also the case in acoustics. In such systems, the mixtures are weighted and delayed, and each source contributes to the sum with multiple delays corresponding to the multiple paths by which an acoustic signal propagates to a microphone.

The convolutive mixing process defined in (2.1) can be simplified by transforming the mixture signals into the frequency domain. In fact, in the frequency domain, the convolution becomes a simple multiplication for each frequency, so that we can rewrite the convolutive mixing process as

$$\mathbf{X}(\omega_k) = \mathbf{H}(\omega_k)\mathbf{S}(\omega_k), \tag{2.5}$$

where at each frequency sample $\omega_k = \frac{2\pi k}{NT}$, $\mathbf{H}(\omega_k)$ is a complex MxN matrix, $\mathbf{X}(\omega_k)$ is a complex Mx1 vector while $\mathbf{S}(\omega_k)$ is a complex Nx1 vector. To map the signal into the frequency domain the Discrete

Fourier Transform (DFT) is usually used, using very efficient algorithms to implement it as the Fast Fourier Transform (FFT); the signal is typically *windowed*, i. e. the DFT is computed using a time interval of length T resulting in

$$\mathbf{X}(\omega_k, t) = \sum_{\tau=0}^{T-1} w(\tau)\mathbf{x}(t+\tau)e^{-i\omega_k\tau/T}, \tag{2.6}$$

in which the window function $w(\tau)$ is chosen to minimize the overlap between different frequency bands.

### 2.1.2 *Statistical properties*

BSS algorithms are based on assumptions on the sources and the mixing system: in general, the sources are assumed to be *independent* or at least *uncorrelated*, while the mixing system is frequently assumed to be linear and time invariant. In convolutive separation another assumption that is common is that the receivers capture M independent linearly mixed versions of the sources: this means that the origins of the sources are located in different positions; this is known as *spatial diversity* assumption.

The independence assumption can be simply stated for two random vectors $\mathbf{x}$ and $\mathbf{y}$: if knowing $\mathbf{x}$ doesn't give any information on $\mathbf{y}$ then we can say that they are mutually independent. More precisely, we can use the Probability Density Function (PDF) of the random variables to define the statistical independence concept: $\mathbf{x}$ and $\mathbf{y}$ are statistically independent if and only if their joint probability density can be decomposed into the product of their individual marginal densities, i. e.

$$p_{x,y}(x, y) = p_x(x)\, p_y(y), \tag{2.7}$$

where $p_x(x)$ and $p_y(y)$ are the PDFs of $x$ and $y$ respectively.

The BSS methods which exploit the statistical properties of the sources can be divided in two groups depending on the separation criteria used, which can either be Second-Order-Statistics (SOS) as in [9, 14] or Higher-Order-Statistics (HOS) [15, 21].

#### MOMENTS

A probability distribution can be characterized by its *moments*: these are statistical parameters used to measure distributions and they are mathematically equivalent to moments in physics, if the probability density function is interpreted as a mass density function.

Given the probability density function (*pdf*) $p_x(x)$, the *r*-th order *raw moment* is given by its expected value about zero

$$m_{x,r} = E[X^r] = \int x^r p_x(x) dx. \tag{2.8}$$

The moment of order zero is always 1 while the first order moment $m_1$ represents the mean value $\mu_x$ of the distribution

$$m_{x,1} = \mu_x = E[X] = \int x\, p_x(x) dx. \tag{2.9}$$

Higher-order moments can be more easily interpreted if they are referred to the mean value, so we need to introduce the *central moments* of a distribution, which are the expected values of the distribution about their mean; the *r*-th order central moment is defined as

$$\mu_{x,r} = E[(X - \mu_x)^r] = \int (x - \mu_x)^r p_x(x) dx. \tag{2.10}$$

Moments can be used to determine the characteristic of a set of data, namely to describe their generating stochastic process. Within the many BSS algorithms, extensive use of SOS (*variance*) [9, 14] and HOS (in particular the *4-th order moment* known as the *Kurtosis*) [15, 21] has been made since they are both useful to determine whether a distribution is a Gaussian or not. The fact that we are interested in evaluating the Gaussianity of a distribution is justified by the assumptions made on the source signals which is better explained in section 2.2.

### SECOND-ORDER STATISTICS

The second-order central moment of a random variable X is the variance of that random variable:

$$\mu_{x,2} = \mathrm{Var}(X) = \sigma_x^2 = E\left[(X - \mu_x)^2\right], \tag{2.11}$$

where $\sigma_x$ is the standard deviation of the random variable $x$. Given two random processes X and Y we can extend the concept of variance to the concept of covariance.

The covariance between two random variables is given as

$$\mathrm{cov}(X, Y) = E\left[(X - \mu_x)^2 (Y - \mu_y)^2\right] \tag{2.12}$$

For two vectors $\bar{x}$ and $\bar{y}$, each composed of $m$ random samples coming from two distributions, we can define the covariance matrix

$$\Sigma_{xy} = \mathrm{cov}(\bar{x}, \bar{y}), \tag{2.13}$$

and it gives a measure of how two random variables will change together. By looking at the covariance matrix of we can understand

whether there is correlation between two (or more) variables. Some BSS techniques are based on SOS by requiring uncorrelated sources but it is important to notice that if two variables are uncorrelated it does not necessarily mean that they are independent: independence implies uncorrelatedness while the vice-versa is not true. This means that, by their-selves, SOS are not sufficient for separation [23]. The advantage of SOS is that they are less sensitive to outliers and noise hence less data is required for their estimation.

### HIGHER-ORDER STATISTICS

High-order moments have been successfully used for assessing independence in the data: one of the ways of expressing independence is that all the cross-moments between the sources are zero; however, these parameters are not easy to understand and there is a lot of confusion on their meaning and interpretation. As already mentioned, the kurtosis is particularly useful and lot of different definitions can be found on Internet and on textbooks, most of them claiming that kurtosis represents the "peakedness" of a distribution.

Dr. Westfall published in [42] several arguments (and proofs) addressing why kurtosis cannot be interpreted as a measure of peakedness: even if there is correlation between peakedness and kurtosis, this relationship is not a direct one. Dr. Wheeler defines kurtosis in [43] as a parameter that is a measure of the combined weight of the tails relative to the rest of the distribution. Kurtosis is about the tails of the distribution (and not the peakedness or flatness): it is correctly interpreted as a measure of the tail-heaviness of the distribution.

The kurtosis is the fourth standardized moment, defined as

$$\frac{\mu_4}{\sigma^4} = \text{Kurt}[X] = \mathrm{E}\left[\left(\frac{X - \mu_x}{\sigma}\right)^4\right], \tag{2.14}$$

and it is usually measured in relationship with the kurtosis of the normal distribution, following Pearson measure of kurtosis, so that it becomes

$$\mathrm{E}\left[\left(\frac{X - \mu_x}{\sigma}\right)^4\right] - 3\,, \tag{2.15}$$

where the constant number 3 is the kurtosis of normal distributions. Following Pearson's convention, when the kurtosis is close to zero then a normal distribution is often assumed: these are called mesokurtic distributions. If the kurtosis is less than zero, then the distribution has light tails and is called a platykurtic distribution. If the kurtosis is greater than zero, then the distribution has heavier tails and is called a leptokurtic distribution. We can see in Fig. 2.3 a comparison between a leptokurtic, a mesokurtic and a platykurtic distribution.

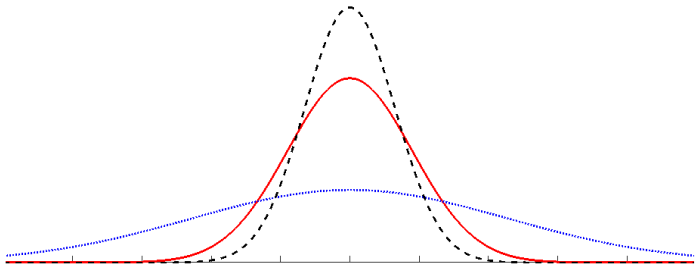


**Figure 2.3:** Comparison of the shape of a leptokurtic distribution (dashed black line), mesokurtic distribution (solid red line) and a platykurtic distribution (dotted blue line).

## 2.2 INDEPENDENT COMPONENT ANALYSIS

A special case of BSS which has been extensively investigated and successfully used is the so called "independent component analysis" (ICA): this is a computational method which aims to separate a multivariate signal into several independent subcomponents which are non-Gaussian. The main assumption of this set of techniques is that the subcomponents are non-Gaussian signals and that they are statistically independent from each other [25]. A common example is given by sounds: a sound, in general, can be represented as a signal which is the composition of the superposition of several source signals. The question then is whether it is possible to separate these contributing sources from the observed total signal.

Conventional ICA algorithms are usable when the number of sources and the number of observations are the same: if L sources are present, at least $M = L$ observations (e.g. microphones) are needed to recover the original signals. Cases of overdetermined ($L < M$) and underdetermined ($L > M$) ICA algorithms have also been attempted and investigated exploiting some assumptions, for example sparseness in data.

The ICA separation of mixed signals delivers very good results whenever the assumptions are satisfied; we can enlist two important assumptions and three effects of mixing source signals.

- The two assumptions are:
  1. the *source signals* are *independent* of each other;
  2. the *values* in each *source signal* have *non-Gaussian distributions*.
- The three effects of mixing source signals:

1. *Dependence*: even if per assumption 1 the source signals are independent, their signal mixtures are not and this is due to the fact that the mixtures share the same source signals.
2. *Normality*: according to the *Central Limit Theorem*, the sum of independent random variables (with finite variance) tends towards a Gaussian distribution. In other words, the sum of two or more independent random variables usually has a distribution which is closer to a Gaussian than any of the two originating distributions.
3. *Complexity*: The temporal complexity of a mixture signal is greater than that of its originating source signals.

These principles form the basic formulation of ICA. Thus, if the extracted signals are independent or have non-Gaussian distributions or have low complexity then they must be source signals.

The independent components are found by maximizing the statistical independence of the estimated components: it is possible to choose one of many ways to measure the independence and this choice influences the development and the form of the ICA algorithm. The two most common used definitions of independence for ICA are:

- minimization of mutual information (MMI);
- maximization of non-Gaussianity.

The MMI family uses measures like the Kullback-Leibler Divergence and maximum entropy while the maximization of non-Gaussianity family, motivated by the central limit theorem, makes use of kurtosis and negentropy.

Typical algorithms use some preprocessing steps to simplify and reduce the complexity of the problem such as trend removal (to create a zero mean signal), whitening (with the eigenvalue decomposition), and dimensionality reduction. Whitening ensures that all dimensions are treated equally before the algorithm is applied. Well-known algorithms for ICA include Infomax [8], FastICA [24], and JADE [16].

The ICA formulation is generally addressed in the time-domain or in the frequency-domain; in the following paragraphs we briefly resume the differences within these two approaches. In general, both in the time and frequency versions, ICA cannot identify the actual number of source signals, nor the proper scaling (including sign) neither the proper permutations of the source signals. These problems need to be carefully addressed in order to have a proper reconstruction of the source signals.

### 2.2.1 *Ambiguities of ICA*

In the ICA model in (2.3), it is easy to see that the following ambiguities (also known as indeterminacies) hold:

1. *Permutation ambiguity*: We cannot determine the order of the independent components. Even if we permute the rows of $\mathbf{G}$, it is still an ICA solution. We can characterize the permutation ambiguity using a permutation matrix $\mathbf{P}$ so that

$$\mathbf{G} = \mathbf{P}\mathbf{H}^{-1}. \tag{2.16}$$

   This means that the solutions found are totally negligent with respect to the order of the sources since we only seek for independence across the generating data sets.

2. *Scaling ambiguity*: given a non-singular diagonal matrix $\mathbf{D}$, if $\mathbf{G} = \mathbf{H}^{-1}$ is a valid separator, i. e. each of the source signals appears at an output terminal of the separator, then it still remains a valid separator no matter the linear transformation given by $\mathbf{D}$ so that

$$\mathbf{G} = \mathbf{D}\mathbf{H}^{-1} \tag{2.17}$$

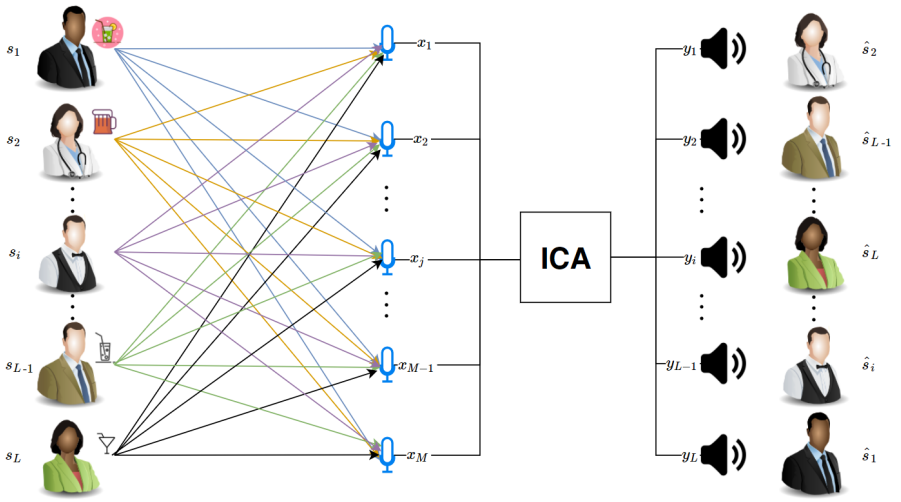
   is also a valid separator. This means that we cannot determine if the solution has a correct scaling, i. e. even if we succeed in separating the sources, they might be multiplied by different unknown scalars. This problem has found several solutions, a famous one given by Matsuoka in [33] that is shortly described in the next section.

BSS is considered to be successful if the output $\mathbf{y}(n)$ is at most a permuted and filtered version of the signal sources $\mathbf{s}(n)$, in which case $\mathbf{G}$ is a product of a permutation matrix $\mathbf{P}$ and a diagonal matrix $\mathbf{D}$:

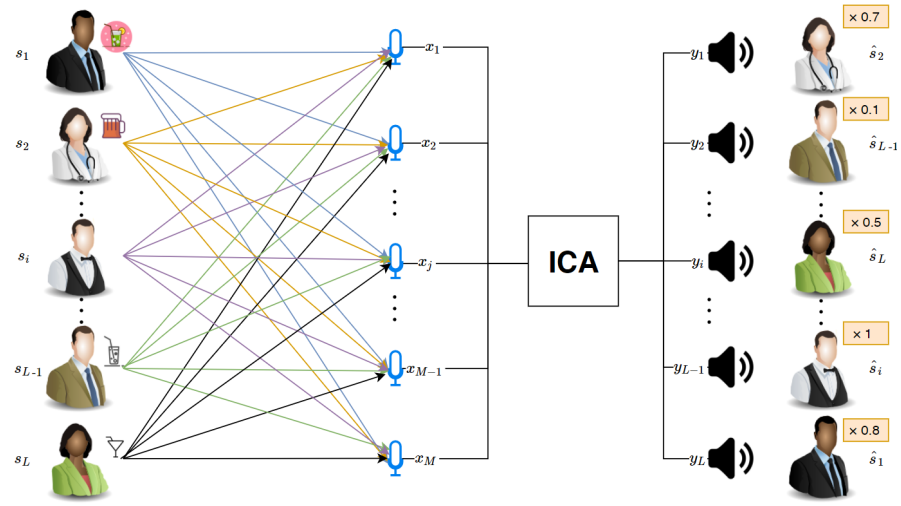
$$\mathbf{G} = \mathbf{P}\mathbf{D}\mathbf{H}^{-1}. \tag{2.18}$$

For ease of understanding, we can see a representation of the effect of the permutation ambiguity in Fig. 2.4(a) and the scaling ambiguity in Fig. 2.4(b): the separation is successful but the order of the separated sources is different from the order of the original sources (permutation ambiguity) and each of the separated sources is multiplied by a different scalar which is not known a priori (scaling ambiguity).

A two group classification of ICA algorithms can be based on the domain in which they work; Time-Domain ICA (TDICA) in which the inverse system of the mixing filter is performed in the time domain




**(a)** Permutation Ambiguity




**(b)** Scaling Ambiguity

**Figure 2.4:** The sources are properly separated but each source is multiplied by a random scalar which is unknown a priori.

and the Frequency-Domain ICA (FDICA) in which the operation of inversion is done in the frequency domain. A brief overview is given in the following sections.

#### 2.2.1.1 *Time Domain ICA*

The first ICA algorithms as InfoMax [7], have been developed in the time-domain for instantaneous mixtures. The problems arising in this formulation are that in most of the real-world applications, the observation signals are noisy and filtered because of the interaction with the surrounding environment.

In TDICA the inverse filter system is performed using the full-band observed signals and this can be an advantage since with the full-band speech signals the independence assumption of sources usually holds. Nevertheless, TDICA has some severe computational problems

due to the high complexity of the iterative rules for the FIR-filter estimation and to the degradation of the convergence when dealing with reverberant environments. TDICA algorithms are efficient only in the case of mixtures with a short-tap FIR filter, i. e. less than 100 taps but, because of the above mentioned problems, they fail to separate source signals under real acoustic environments.

#### 2.2.1.2 *Frequency Domain ICA*

Some limitations of the ICA can be overcome by the use of the FDICA: Fourier transform techniques are useful in dealing with convolutive mixtures since convolutions in the time domain become products between Fourier transforms in the frequency domain. Applying Fourier transform to the data does not change the mixing matrix since this operation is a linear one. It is possible to use standard ICA algorithms in the Fourier domain, taking the STFT [2, 3, 1] of the data, instead of the global transform. This means that the Fourier transform is applied separately to each data window and the ICA algorithms can run on each frequency bin, giving the separation per each frequency band. This allow the complexity of the filter and the convergence speed to be much reduced but a major problem arises: the mixing matrix is now a function of the angular frequency while in the standard ICA/BSS problem it is constant. In fact, the problem with the FDICA approach is that of the indeterminacy of permutation and scale: since we run the ICA for every discrete frequency bin $[k]$, the indeterminacies are usually different in each frequency interval so we now have that

$$\mathbf{G}[k] = \mathbf{P}[k]\mathbf{D}[k]\mathbf{H}^{-1}[k]. \tag{2.19}$$

This problem is now magnified compared to the one we had in the TDICA since to reconstruct a source signal in the time domain, we need all its frequency components in the correct order. If we would sum the separated components given by the application of the ICA to each frequency bin to reconstruct the sources, we would not know to which source the separated component belongs to, so that the result would be an unknown signal which would be composed of frequency components coming from different signal sources as represented in Fig. 2.5.

Thus, several approaches for choosing which source signals in different frequency intervals belong together have been proposed and this methods have been categorized in [10] in two main categories: methods based on consistency of filter coefficients, and methods based on consistency of the spectrum of the recovered signals.

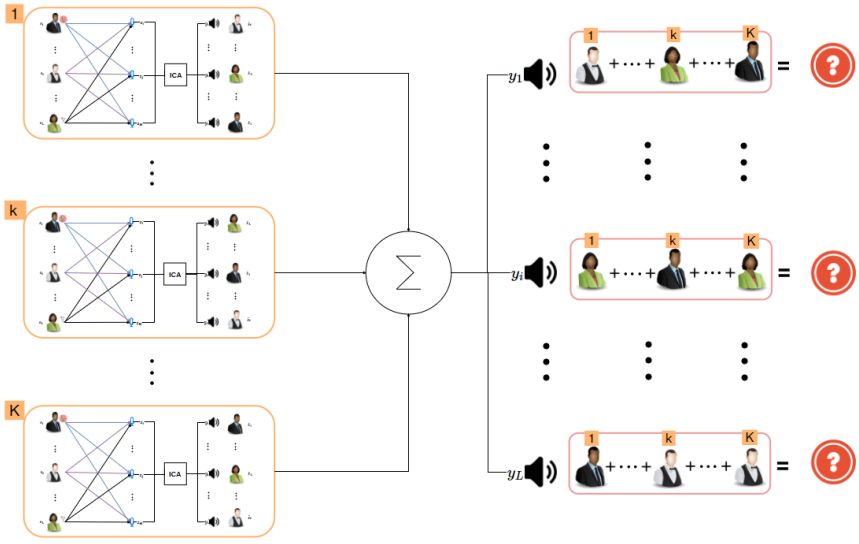



**Figure 2.5:** The sources are properly separated but each source is multiplied by a random scalar which is unknown a priori.

### SCALING PROBLEM SOLUTION: THE MINIMAL DISTORTION PRINCIPLE

In section 2.2 we described the scaling ambiguities of ICA. We can note that (2.18) in practice states that in BSS all the valid separators are usually considered essentially equivalent. The Minimal Distortion Principle [33] by Matsuoka states the following separator has a special meaning:

$$\mathbf{G}^\star \triangleq \operatorname{diag}\mathbf{H} \cdot \mathbf{H}^{-1} = \mathbf{D}\mathbf{H}^{-1}, \tag{2.20}$$

where $D = \operatorname{diag}\mathbf{H}$. We call this separator the optimal (valid) separator. It should be noted that this definition of the optimal separator has no indeterminacy; it is uniquely determined independently of in the indeterminacy in the definition of the source signals because the following holds for any diagonal matrix $\mathbf{E}$:

$$\operatorname{diag}\mathbf{H}\mathbf{E} \cdot (\mathbf{H}\mathbf{E}^{-1}) = \operatorname{diag}\mathbf{H} \cdot \mathbf{H}^{-1} \tag{2.21}$$

The optimal separator $\mathbf{G}^\star$ can be characterized by either of the following two propositions.

**Proposition 1:** The optimal separator $\mathbf{G}^\star$ is the valid separator that minimizes $|\mathbf{G}\mathbf{H} - \mathbf{H}|^2$.

**Proposition 2:** The optimal separator $\mathbf{G}^\star$ is the valid separator that minimizes $E\left[|\mathbf{y}(t) - \mathbf{x}(t)|^2\right]$.

These two propositions state the minimal distortion principle in two manners. Namely, the optimal separator is determined such that the

overall transfer function **GH** be as close to **H** as possible, or equivalently the separator's output $\mathbf{y}(t)$ be as close to $\mathbf{x}(t)$ as possible. The optimal separator can also be characterized as a direct constraint on matrix **G** .

**Proposition 3:** The optimal separator $\mathbf{G}^{\star}$ is the valid separator that satisfies $\operatorname{diag} \mathbf{G}^{-1} = \mathbf{I}$.

The optimal separator has some properties that are favorable in actual implementation of BSS.

1. The output of the separator then becomes

$$\mathbf{y}(t) = \operatorname{diag}(\mathbf{H})\mathbf{H}^{-1}\mathbf{H}\mathbf{s}(t) = \operatorname{diag}(\mathbf{H})\mathbf{s}(t). \tag{2.22}$$

   This implies that output $y_i(t)$ is $a_{ii}s_i(t)$ , which is the *i*-th source that would be observed at the *i*-th sensor when there were no other source signals. This property will be convenient for interpretation of the signals separated and later processing.

2. The optimal separator does not depend on the properties of the sources; it depends on the mixing process **H** only. So, even for such non-stationary signals as voices, the optimal separator is invariant with time as long as the mixing process is fixed.

3. In actual implementation, the separator needs to be realized with an FIR filter. It is desirable that the degree of the filter is as low as possible. Based on the minimal distortion principle, the separator is chosen such that the output of the separator becomes as close to the output of the sensor as possible. So, it can be expected that the separator will be realized with a relatively low degree.

# 3

# INDEPENDENT VECTOR ANALYSIS

Independent vector analysis (IVA) has been proposed by Kim, Eltoft, and Lee in [29] as an extension of ICA to solve the frequency permutations (also referred to as *local permutations*). IVA has captured the attention of many researchers for the above mentioned characteristic and has been investigated and extended to optimize its convergence. In the following sections we describe the standard IVA [29] and we shortly describe some of the most interesting extensions of the algorithm [40, 27, 34].

## 3.1 STANDARD IVA

Applying the ICA algorithm to instantaneous mixtures in each frequency bin would lead to a reduction of the computational complexity and a faster convergence but then the problem would be the permutation of the ICA solutions over different frequency bins: this is due to the indeterminacy of permutation inherent in the ICA algorithm which should be corrected to obtain a proper separation of the signal in the time domain. Kim et al. [30] reformulated the cost function in ICA and proposed a dependency model which captures inter-frequency dependencies in data: these dependencies are related to an improved model for the source signal prior. While the source priors are defined as independent priors at each frequency bin in conventional algorithms, higher order dependencies are used across frequency. Thus, it is possible to define each source prior as a multivariate super-Gaussian distribution[1]. IVA is able to preserve higher-order

1 the multivariate super-Gaussian distribution is an extension of the independent Laplacian distribution

dependencies and structures of frequencies so that the local permutation problem is completely avoided, and the separation performances are comparably high even in severely ill-posed conditions.
The method proposed in [30] consists of a mixing and separating procedure in a convolutive environment, the definition of a cost function, and an algorithm for learning the parameters of the separating filters.

### 3.1.1 *Frequency Domain IVA*

Using the model defined in section 2.1.1, we know that in a convolutive environment, source signals are time delayed and convolved. Consider the *i*th observation signal $x_i(t)$ at time $t$

$$x_i(t) = \sum_{j=1}^{L} \sum_{\tau=0}^{T-1} h_{ij}(\tau) s_j(t-\tau), \tag{3.1}$$

where $h_{ij}(t)$ is a time-domain transfer function from the *j*th source to the *i*th observation, which has $T$ length in time, $s_j(t)$ is the *j*th source signal at time $t$, and $L$ is the number of sources. Applying the STFT, the time-domain signal $x_i(t)$ is converted to the frequency-domain signal $x_i[n,k]$

$$x_i[n,k] = \sum_{t=0}^{K-1} w(t) x_i(nJ+t) e^{-j\omega_k t}, \tag{3.2}$$

where $n$ is the frame index, $\omega_k = 2\pi(k-1)/K$ is the $k$-th frequency sample where $k = 1, \cdots, K$; $J$ is the shift size and $w(t)$ is a window function. If the window length $K$ is sufficiently longer than the length of the mixing filter $h_{ij}(t)$, the convolution in the time domain is approximately converted to multiplication in the frequency domain

$$x_i[n,k] \approx \sum_{j=1}^{L} h_{ij}[k]\, s_j[n,k]. \tag{3.3}$$

If the separating filter matrices exist, i. e. the inverses or pseudo-inverses of the mixing matrices at each frequency exist ($L <= M$), then the separated *i*th source signal is given as

$$y_i[n,k] = \sum_{j=1}^{M} g_{ij}[k]\, x_j[n,k] \approx s_i[n,k], \tag{3.4}$$

where $g_{ij}[k]$ is the separating filter at the $k$-th frequency bin and $M$ is the number of observed signals. The problem is that of determining the separating filter and to solve it is necessary to define a proper objective function for multivariate random variables.

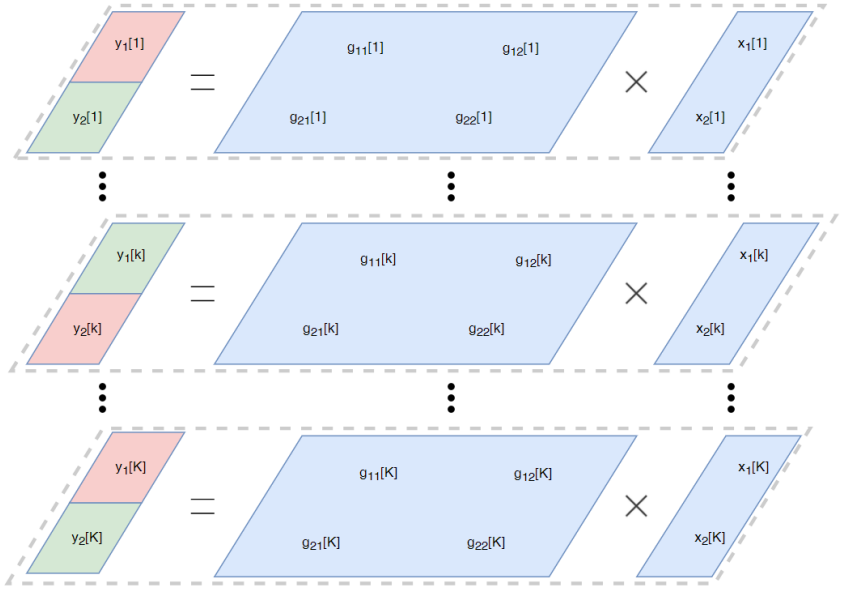
### 3.1.2 *IVA cost function*

To separate multivariate sources from multivariate observations, in [30] was defined a cost function for multivariate random variables: the *Kullback–Leibler divergence* between two functions is used as the measure of independence. One function is an exact joint probability density function $p(\hat{\mathbf{s}}_1, \cdots, \hat{\mathbf{s}}_L)$, and the other is a nonlinear function which is the product of approximated probability density functions of individual source vectors $\prod_{i=1}^{L} q(\hat{\mathbf{s}}_i)$. This can be seen as an extension of mutual information between multivariate random variables:

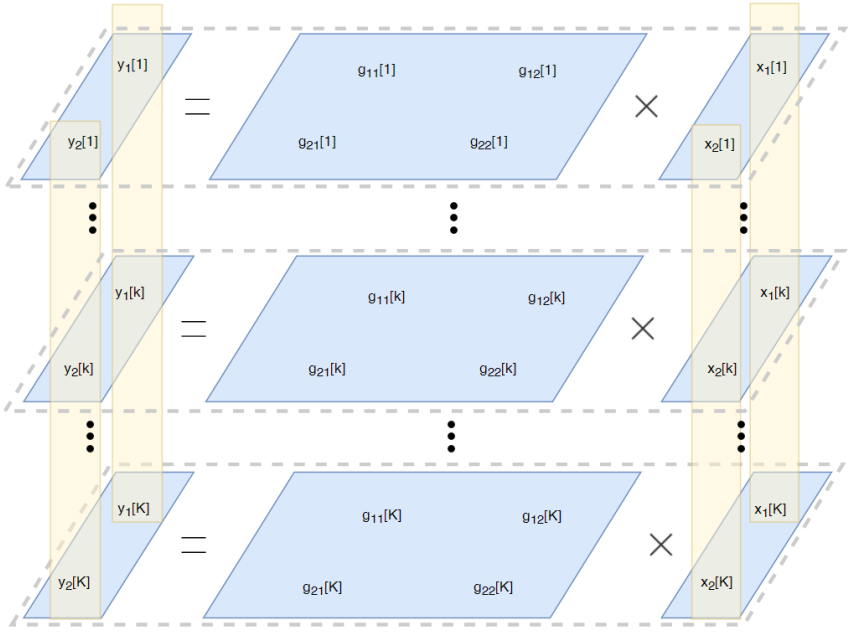
$$\begin{aligned}
\mathcal{C} &= \mathcal{KL}\left(p(\hat{\mathbf{s}}_1, \cdots, \hat{\mathbf{s}}_L) \parallel \prod_{i=1}^{L} q(\hat{\mathbf{s}}_i)\right) \\
&= \int p(\hat{\mathbf{s}}_1, \cdots, \hat{\mathbf{s}}_L) \log \frac{p(\hat{\mathbf{s}}_1, \cdots, \hat{\mathbf{s}}_L)}{\prod_{i=1}^{L} q(\hat{\mathbf{s}}_i)} d\hat{\mathbf{s}}_1, \cdots, d\hat{\mathbf{s}}_L \\
&= \int p(\mathbf{x}_1, \cdots, \mathbf{x}_M) \log p(\mathbf{x}_1, \cdots, \mathbf{x}_M) d\mathbf{x}_1, \cdots, d\mathbf{x}_M + \\
&\quad - \sum_{k=1}^{K} \log |\det \mathbf{G}[k]| - \sum_{i=1}^{L} \int p(\hat{\mathbf{s}}_i) \log q(\hat{\mathbf{s}}_i) d\hat{\mathbf{s}}_i \\
&= \text{const} - \sum_{k=1}^{K} \log |\det \mathbf{G}[k]| - \sum_{i=1}^{L} E\left[\log q(\hat{\mathbf{s}}_i)\right],
\end{aligned} \tag{3.5}$$

where $\mathbf{G}[k]$ is the separating matrix at the $k$-th frequency bin, the term $const = \int p(\mathbf{x}_1, \cdots, \mathbf{x}_M) \log p(\mathbf{x}_1, \cdots, \mathbf{x}_M) d\mathbf{x}_1, \cdots, d\mathbf{x}_M = H(\mathbf{x})$ is the entropy (used in information theory) of the given observations, which is a constant because the observed signals will not change in the optimization procedure; the function $H(\cdot)$ represents the entropy while $E[\cdot]$ represents the expectation. The third step in (3.5) is derived using $H(\mathbf{Gx}) = log(det|\mathbf{G}|) + H(\mathbf{Gx})$ which holds for a linear invertible transformation $\mathbf{G}$, and the determinant of the block diagonal matrix $\mathbf{G}$ is $det(\mathbf{G}) = \prod_{k=1}^{K} \det(\mathbf{G}[k])$.

It is important to note that the random variables in the above equations are multivariate: each source is multivariate and it is minimized when the dependency between the source vectors is removed but the dependency between the components of each vector does not need to be removed. Therefore, the cost function preserves the inherent frequency dependency within each source, but it removes dependency between the sources. An example of a $2 \times 2$ demixing model is shown in Fig. 3.1. In this example, each horizontal layer is an ICA demixing model in for each frequency bin, and the demixing procedure is carried out in layers independently. Since ICA in different layers may output the separated results in different order, the permutation ambiguity will occur, which is indicated by the different color of $\mathbf{y}[k]$ Fig. 3.1. The permutation ambiguity must be carefully addressed before the inverse STFT is performed, or else the separation procedure




**(a)** ICA spearation procedure




**(b)** IVA separation procedure

**Figure 3.1:** Comparison between the ICA demixing procedure (a) and the IVA demixing procedure (b). The figure shows that the the ICA procedure suffers from the permutation problem (green and red colors) while IVA procedure is computed by considering the input as a vector (vertical yellow bars), preserving the order of the sources during the process.

would fail. In addition to separate sources in each frequency bin, IVA utilizes inter-frequency bin information to solve the permutation problem in the separation procedure. The IVA model is very similar with the ICA model, as shown in Fig. 3.1. Their difference is that signals are considered as vectors in IVA, i.e. $\mathbf{x}_i = [x_i[1] \cdots, x_i[K]]^T$, $\mathbf{y}_i = [y_i[1], \cdots, y_i[K]]^T$ (vertical bars in Fig. 3.1(b)), and they will be optimized as multivariate variables, instead of independent scalars like in ICA.

### 3.1.3 *Learning algorithm*

#### GRADIENT METHOD

The derivation of the learning algorithm is done by using a gradient descent method to minimize the cost function; differentiating the cost function with respect to the coefficients of the separating matrices, it is possible to obtain the gradients for the coefficients as follows:

$$\Delta g_{ij}[k] = -\frac{\partial \mathcal{C}}{\partial g_{ij[k]}} = g_{ij}^{-H[k]} - E\left[\varphi[k]\,(\hat{\mathbf{s}}_i[1], \cdots, \hat{\mathbf{s}}_i[K])\,\mathbf{x}_j^\star[k]\right] \quad (3.6)$$

where $g^H$ is used to indicate the conjugate transpose (Hermitian) of $g$ while $x^\star$ indicates the conjugate of $x$ and $(\mathbf{G}^{-1}[k])^H = g_{ij}^{-H}[k]$. By multiplying scaling matrices $G^\dagger[k]G[k]$ to the gradient matrices $\Delta G[k] \equiv \Delta g_{ij}[k]$, we can obtain the *Natural Gradient*, which is well known as a fast convergence method [6], so

$$\Delta g_{ij}[k] = \sum_{l=1}^{L} (I_{il} - E\,[\varphi[k]\;(\hat{\mathbf{s}}_i[1], \cdots, \hat{\mathbf{s}}_i[K])\,\hat{\mathbf{s}}_l^\star[k]\,]\,)\,g_{lj}[k], \quad (3.7)$$

where $I_{il}$ is 1 only when $i = l$, otherwise 0, and the nonlinear function $\varphi[k](\cdot)$ is given as

$$\varphi[k](\hat{\mathbf{s}}_i[1], \cdots, \hat{\mathbf{s}}_i[k]) = -\frac{\partial \log q(\hat{\mathbf{s}}_i[1], \cdots, \hat{\mathbf{s}}_i[k])}{\partial \hat{\mathbf{s}}_i[k]}. \quad (3.8)$$

The term (3.8) is referred to as *multivariate score function*, and it corresponds to the score function in the conventional ICA.

To compute the batch version of the algorithm, the expected value in (3.7) is calculated by summing the product of the value of the random variable and its associated probability, taken over all of the values of the random variable. The batch update rule to update the coefficients of separating matrices is given as

$$g_{ij}^{new}[k] = g_{ij}^{old}[k] + \eta \Delta g_{ij}[k] \quad (3.9)$$

where $\eta$ is the learning rate. The online update rule can be obtained by omitting the expectation in (3.7) and updating at every time sample.

### MULTIVARIATE SCORE FUNCTION

The only difference between the IVA approach and the conventional ICA is the form of the score function. In fact, if the multivariate score function $\varphi[k](\hat{s}_i[1], \cdots, \hat{s}_i[k]$ is defined as a single-variate score function, then the algorithm would reduce to the same in the conventional ICA: the fact that the score function is a multivariate function is the most important point in IVA.

According to many ICA literatures, a score function is closely related to a source prior. For example, when the sources have super-Gaussian distribution, Laplacian distribution is widely used as a source prior. In IVA, a multivariate score function is also closely related to a source prior since the cost function (3.5) includes $q(\hat{\mathbf{s}}_i)$, which is an approximated probability density function of a source vector, that is, $q(\mathbf{s}_i) \approx p(\mathbf{s}_i)$ . Thus, as shown in (3.8), a multivariate score function can be obtained by differentiating the log prior with respect to each element of a source vector.

In most BSS approaches, the source prior for a super-Gaussian signal is defined by a Laplacian distribution: suppose that the source prior of a vector is independent Laplacian distribution in each frequency bin. This can be written as:

$$p(\mathbf{s}_i) = \prod_{k=1}^{K} p(s_i[k]) = \alpha \prod_{k=1}^{K} \exp\left(-\frac{|s_i[k] - \mu_i[k]|}{\sigma_i[k]}\right), \tag{3.10}$$

where $\alpha$ is a normalization term, and $\mu_i[k]$ and $2\sigma_i[k]^2$ are a mean and a variance of the $i$th source signal at the $k$th frequency bin, respectively. Assuming zero mean and unit variance, the score function is given by

$$\begin{aligned} \varphi[k](\hat{\mathbf{s}}_i[1] \cdots \hat{\mathbf{s}}_i[k]) &= \frac{\partial \sum_{k=1}^{K} |\hat{s}_i[k]|}{\partial \hat{s}_i[k]} = \frac{\hat{s}_i[k]}{|\hat{s}_i[k]|} \\ &= \exp\left(j \cdot \arg(\hat{s}_i[k])\right), \end{aligned} \tag{3.11}$$

but (3.11) is not a multivariate function, because the function depends on only a single variable $\hat{s}_i[k]$.

Instead of using an independent prior, we have to define a new prior, which is highly dependent on the other elements of a source vector; in IVA the source prior is defined as a dependent multivariate super-Gaussian distribution

$$p(\mathbf{s}_i) = \alpha \exp(-\sqrt{(\mathbf{s}_i - \mu_i)^H \mathbf{\Sigma}_i^{-1} (\mathbf{s}_i - \mu_i)}), \tag{3.12}$$

where $\mu_i$ and $\mathbf{\Sigma}_i$ are a mean vector and a covariance matrix of the $i$th source signal, respectively.

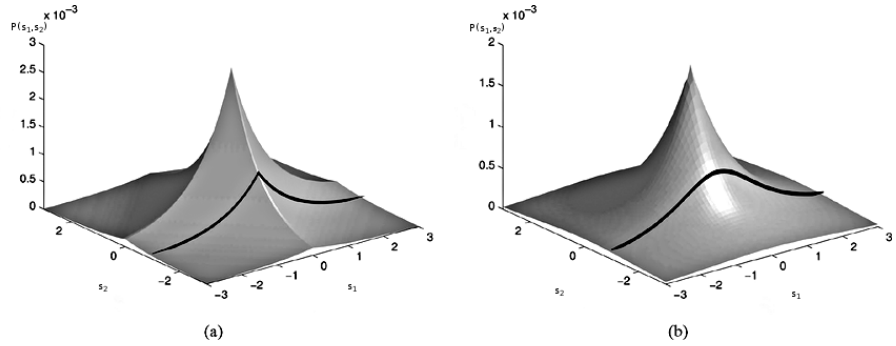



**Figure 3.2:** Comparison between (a) an independent Laplacian distribution and (b) a dependent multivariate super-Gaussian distribution. The figure shows the dependency between only two arbitrary elements of a multidimensional variable $\mathbf{s} = [s[1] \cdots s[K]]^T$ . $s_1$ can be considered as either real or imaginary part of $s[1]$ , and also $s_2$ can be considered as either real or imaginary part of $s^{(2)}$. The black line indicates $p(s_1|s_2 = 1)$. In (a), the probability of $s_1$ always has Laplacian distribution regardless of $s_2$. In (b), however, the probability of $s_1$ given $s_2 = 1$ does not have Laplacian distribution even though the probability of $s_1$ given $s_2 = 0$ has Laplacian distribution. [30]

In Fig. 3.2 is shown the difference between the assumption of independent Laplacian distribution and a dependent multivariate super-Gaussian distribution. In Fig. 3.2(b), the joint distribution of $x_1$ and $x_2$ does not display any directionality which means $x_1$ and $x_2$ are uncorrelated. However, the marginal distribution of $x_1$ is different from the joint distribution of $x_1$ and $x_2$, that is $x_1$ and $x_1$ are highly dependent. In contrast to the distribution shown in Fig. 3.2(a), Fig. 3.2(b) has a radial shape, which is similar to Gaussian distribution, but has higher peak and heavier tail. Thinking in a different way, one can notice that the distribution shown in Fig. 3.2(b) can be obtained by a scale mixture of Gaussians with a fixed mean and a variable variance, as we describe next.

Suppose that there is a $K$-dimensional random variable, which is defined by

$$\mathbf{s}_i = \sqrt{v} \cdot \mathbf{z}_i + \boldsymbol{\mu}_i, \tag{3.13}$$

where $v$ is a scalar random variable, $\mathbf{z}_i$ is a $K$-dimensional random variable, and $\boldsymbol{\mu}_i$ is a $k$-dimensional deterministic variable. Here, the random variable, $\mathbf{z}_i$, has Gaussian distribution with zero mean and $\Sigma_i$ covariance matrix, so that

$$p(\mathbf{z}_i) = \alpha_z \exp\left(-\frac{\mathbf{z}_i^H \boldsymbol{\Sigma}_i^{-1} \mathbf{z}_i}{2}\right), \tag{3.14}$$

where $\alpha_z$ is a normalization term.

Suppose that $\nu$ has a kind of Gamma distribution

$$p(\nu) = \alpha_\nu \nu^{\frac{(K-1)}{2}} \exp(-\frac{\nu}{2}), \tag{3.15}$$

where $\alpha_\nu$ is a normalization term. Then, the given random variable $\mathbf{s}_i$ has Gaussian distribution. Its mean and covariance are $\mu_i$ and $\Sigma_i$, respectively. In this model, the used distribution can be obtained by integrating the joint distribution of $\mathbf{s}_i$ and $\nu$ over $\nu$

$$\begin{aligned} p(\mathbf{s}_i, \nu) &= \int_0^\infty p(\mathbf{s}_i, \nu|\nu) p(\nu) d\nu \\ &= \hat{\alpha} \int_0^\infty \sqrt{\nu} \exp\left(-\frac{1}{2}\left(\frac{(\mathbf{s}_i - \mu_i)^H \mathbf{\Sigma}^{-1} (\mathbf{s}_i - \mu_i)}{\nu} + \nu\right)\right) d\nu \\ &= \alpha \exp\left(-\sqrt{(\mathbf{s}_i - \mu_i)^H \mathbf{\Sigma}^{-1} (\mathbf{s}_i - \mu_i)}\right). \end{aligned} \tag{3.16}$$

Therefore, each component of $\mathbf{s}_i$ is not only correlated to others caused by $\mathbf{\Sigma}_i$, but also has variance dependency generated by $\nu$. Even though the covariance matrix $\Sigma_i$ is assumed to be identity, that is, each component of $\mathbf{s}_i$ is uncorrelated, the components are dependent on each other. Most natural signals have inherent dependencies between frequency bins such as the variance dependency above modeled. In other words, when one frequency component has a larger variance, the other frequency components have larger variances as well. From a theoretical point of view, each frequency bin is uncorrelated to the others, because the Fourier bases are orthogonal bases; this is not completely true when dealing with finite observations and with STFT approximations. Supposing to be in the ideal case of infinite observations and using the Fourier Transform, it is possible to set the covariance term $\mathbf{\Sigma}_i$ as a diagonal matrix. Since Fourier outputs have zero means, it is possible to write (3.12) as follows:

$$p(\mathbf{s}_i) = \alpha \exp\left(-\sqrt{\sum_k \left|\frac{s_i[k]}{\sigma_i[k]}\right|^2}\right) \tag{3.17}$$

where $\sigma_i[k]$ is the standard deviation of the $i$th source at the $k$th frequency bin, which determines the scale of each element of a source vector. In the algorithm, $\sigma_i[k]$ is set to 1, because of the adjustment of the scale based on the Minimal distortion principle [33] after learning the separating filters. Consequently, the multivariate score function used is given as

$$\varphi[k](\hat{s}_i[1] \cdots \hat{s}_i[k]) = \frac{\partial \sqrt{\sum_{k=1}^K |\hat{s}_i[k]|^2}}{\partial \hat{s}_i[k]} = \frac{\hat{s}_i[k]}{\sqrt{\sum_{k=1}^K |\hat{s}_i[k]|^2}} \tag{3.18}$$

Kim, Eltoft, and Lee algorithm described in in [29] uses a fixed form of a multivariate score function (3.8), but this does not mean that the

used form is appropriate for separating source signals. Since the form of a multivariate score function is related to dependency of sources, the proper form of a multivariate score function might vary with different types of dependency.

### ONLINE LEARNING ALGORITHM

*Kim* proposed an online version of the IVA algorithm in [28]: the coefficients of the separation-filter matrices are updated at every frame. Thus, (3.3) should be slightly modified as follows:

$$y_i[n,k] = \sum_{j=1}^{M} g_{ij}[n,k]\, x_j[n,k] \tag{3.19}$$

where $n$ denotes the frame index. Therefore, the filter coefficients are updated as:

$$g_{ij}[n,k] = g_{ij}[n-1,k] + \eta \Delta g_{ij}[n,k] \tag{3.20}$$

where $\Delta g_{ij}[n,k]$ denotes the gradient of the current frame, which is the most critical part of the algorithm and it is discussed in the following subsections.

The Natural Gradient in (3.7) needs to be modified as well: there, the ensemble of the estimated outputs are needed to calculate the expectation $E\left[\varphi[k]\,(\hat{\mathbf{s}}_i[1],\cdots,\hat{\mathbf{s}}_i[k])\,\hat{\mathbf{s}}_l^{\star}[k]\right]$ that we refer to as the *scored correlation* and assign it to $\mathfrak{R}_{il}[k]$.
For the batch learning, it is simply obtained by taking a sample mean of them as follows:

$$\mathfrak{R}_{il}[k] = \frac{1}{N}\sum_{n=0}^{N-1} \varphi[k]\,(\mathbf{y}_i[n,1],\cdots,\mathbf{y}_i[n,K])\;\mathbf{y}_l^{\star}[n,k] \tag{3.21}$$

However, estimating the online version of the scored correlation $\mathfrak{R}_{il}[k]$ becomes more complicated.

This achieved by *two assumptions*:

1. The first assumption is that the scored correlation $\mathfrak{R}_{il}[k]$ depends on only the previous frames. Instead of considering all the time frames, it is possible to use only some previous frames from current time and then calculate the exact scored correlation $\mathfrak{R}_{il}[k]$ for those limited number of frames. However, in this case, there is the necessity to calculate the previous outputs using updated filter coefficients. This makes the algorithm inefficient.
2. The second assumption is given by the fact that a simple stochastic gradient is adopted by omitting the expectation in (3.21).

The online version of the scored correlation at a current frame can be calculated as:

$$\mathfrak{R}_{il}[n,k] = \varphi[k]\,(\,\mathbf{y}_i[n,1],\cdots,\mathbf{y}_i[k,n]\,)\;\mathbf{y}_l^{\star}[k,n] \tag{3.22}$$

Thus, the online Natural Gradient learning rule is given as

$$\Delta g_{ij}[k] = \sum_{l=1}^{L} (I_{il} - \mathfrak{R}_{il}[n,k]\,)\,g_{lj}[n,k] \tag{3.23}$$

This works well to extract the source signals when it is applied to batch learning. In the case of online learning, a stability problem may occur. Looking at (3.23) it is possible to observe that the gradient converges to zero when the scored correlation $\mathfrak{R}_{il}[n,k]$ approaches the identity matrix $I_{il}$: this means that if the source signals change their local average magnitudes, the gradient may fluctuate according to that. In many applications, such as speech signals processing and evoked potentials, when a source signal becomes suddenly very small, 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 a non-holonomic constraint [4] is adopted to avoid this phenomenon. In consequence, the following gradient is obtained with the constraint by replacing the identity matrix $I_{il}$ with $\Lambda_{il}[n,k]$:

$$\begin{aligned}\Delta g_{ij}[k] &= \sum_{l=1}^{L} (\Lambda_{il}[n,k] - \mathfrak{R}_{il}[n,k])\,g_{lj}[n,k] \\ &= \begin{cases} 0, & \text{if } i = l \\ \sum_{l=1}^{L} (-\mathfrak{R}_{il}[n,k])\,g_{lj}[n,k], & \text{if } i \neq l \end{cases}\end{aligned} \tag{3.24}$$

where $\Lambda_{il}[n,k]$ is equal to $\mathfrak{R}_{il}[n,k]$ when $i = l$ and is zero when $i$ is not equal to $l$. Thus, $L$ multiplications can be omitted at every frequency bin, which is more efficient when compared with (3.23).

In order to improve the convergence properties and obtain a robustness to input level, one may consider a second-order gradient. In the case of batch algorithm, we could obtain a faster algorithm by adopting the Newton method as shown in [32]. However, this approach has some constraints such that if the inputs are supposed to be spatially whitened, then the separation matrix is orthogonal. Thus, applying it to a real-time online algorithm would not be efficient. Instead, we would follow the gradient derived in the previous sections and adjust only the learning rate with a normalization factor as

$$g_{ij}[n,k] = g_{ij}[n-1,k] + \eta\sqrt{\xi^{-1}[n,k]}\,\Delta g_{ij}[n,k] \tag{3.25}$$

where $\epsilon^{-1}[n,k]$ denotes the normalization factor. Here, we would normalize the gradient with respect to the input level and apply the same factors to all corresponding sources. Therefore, the normalization factor is given as

$$\xi[n,k] = \beta\xi[n-1,k] + \frac{(1-\beta)}{L}\sum_{i=0}^{L}|x_i[n,k]|^2 \tag{3.26}$$

where $\beta$ is a smoothing factor.
Equivalently with the batch algorithm of IVA, the proposed online algorithm also avoids the permutation problem. However, the scales of the outputs may be different from the original ones. In particular, the different scale in each frequency bin causes frequency distortion when the signal is reconstructed. To avoid this problem it is possible to adjust the learned separation-filter matrix. The method using the MDP, that we shortly described in the last paragraph of Section 2.1, is a well known solution [33]. Accordingly, we adjust the output signal by multiplying the scale factor $diag\,(G^{-1}[k])$.

The final process is the reconstruction of the time-domain version of the estimated signal by performing an inverse Fourier transform and overlap-add method as follows:

$$y_i(t) = \sum_{n=0}^{N-1}\sum_{k=0}^{K} y_i[n,k]e^{j\omega_k(t-nk)}. \tag{3.27}$$

#### 3.1.3.1 *IVA issues*

While IVA is theoretically "permutation free" across the frequencies, it does not solve the ambiguity order of the full-band output signals (i. e. the "global permutation") which is a complicated problem in time-varying conditions as the output order might change over time. In addition, in IVA it is assumed that the mixture is a linear combination of a known number of sources but in real-world the source activity is likely to vary over time and this could lead the algorithm to diverge.

## 3.2 IVA EXTENSIONS

In the following sections, we report some interesting extensions of the standard IVA. One of the reviewed techniques uses a different approach to minimize the cost function in IVA by using an auxiliary function approach [40], while the other extensions [27, 34] try to solve some of the IVA issues such as the global permutation problem and they try to improve its convergence properties.

### 3.2.1 *Auxiliary Function approach IVA*

Taniguchi et al. proposed in [40] a stable online IVA algorithm for super-Gaussian convolutive mixtures based on a fast and stable batch IVA algorithm using an auxiliary-function technique [35].
AuxIVA requires no environment-sensitive parameters such as the step-size parameter used in natural gradient IVA and the convergence speed is much faster than the conventional gradient optimization. The formulation follows from the standard IVA cost function (3.5); assuming that the number of microphones M is equal to the number of sources L, then (3.5) becomes

$$\mathbf{J}(\mathbf{G}) = \text{const} - \sum_{k=1}^{K} \log|\det \mathbf{G}[k]| - \sum_{i=1}^{L} E[\log q(\hat{\mathbf{s}}_i)], \tag{3.28}$$

where the term $C(\hat{\mathbf{s}}_i) = -E[\log q(\hat{\mathbf{s}}_i)]$ is referred to as contrast function. Since IVA assumes a multivariate super-Gaussian distribution as source prior, combining (3.28) and the source prior model defined in (3.17), we can define an auxiliary variable $r_i[n]$ as

$$r_i[n] = \sqrt{\sum_{k=1}^{K} |\hat{s}_i[k]|^2} = \sqrt{\sum_{k=1}^{K} |\mathbf{g}^H[k]\mathbf{x}[n,k]|^2}. \tag{3.29}$$

Since the minimization of the standard IVA cost function (3.5) is a nonlinear optimization problem, in general no closed-form solution is available. To find a solution, an iterative application of the gradient-based update rule is needed but this is operation brings a trade-off between convergence speed and stability which is dependent on the value of the step-size parameter. In AuxIVA, instead of directly decreasing the cost function in (3.28), the demixing matrix is estimated by calculating the auxiliary variable (3.29) and decreasing the following auxiliary function with respect to the demixing matrix $\mathbf{G}$:

$$Q(\mathbf{G}, \mathbf{r}) = \frac{1}{2} \sum_{i=1}^{L} \sum_{k=1}^{K} \mathbf{g}_i^H[k]\mathbf{V}[k]\mathbf{g}_i[k] - \sum_{k=1}^{K} \log|\det \mathbf{G}[k]| + \text{const}\,, \tag{3.30}$$

where $\mathbf{r}$ represents the set of auxiliary variables $r_i[n]$ for $i = 1, \cdots, L$. The auxiliary variables $r_i[n]$ are included in the statistics, with weighted covariances $\mathbf{V}[k]$ as follows:

$$\mathbf{V}[k] = \sum_{k=1}^{K} \left[ \frac{C'(r_i[n])}{r_i[n]} \mathbf{x}[n,k]\, \mathbf{x}^H[k,n] \right], \tag{3.31}$$

where $\mathbf{V}[k]$ can be considered as a covariance weighted by the scalar $\frac{C'(r_i[n])}{r_i[n]}$: this cost function is guaranteed to decrease monotonically.

An online version has been proposed in [40] by approximation of the auxiliary variable in the auxiliary function by autoregressive estimation of its related statistics. This is a natural extension of the offline

AuxIVA and algorithm can be efficiently implemented. However, the theoretical correctness of the approximation, i.e. whether the online version converges as well as the offline AuxIVA does, has not been proven at this moment.

### 3.2.2 *Geometrically Constrained IVA*

Khan, Taseska, and Habets proposed in [27] a geometrically constrained IVA (CIVA) algorithm that works in the frequency domain to extract the desired source whose DOA is known.

Taking the first microphone of the array of sensors as reference, the *relative transfer function* (RTF) is given by

$$\mathbf{z}_1[k] = [1, e^{j\frac{\omega}{c}[\mathbf{d}_2-\mathbf{d}_1]^T\mathbf{q}_1}, \cdots, e^{j\frac{\omega}{c}[\mathbf{d}_M-\mathbf{d}_1]^T\mathbf{q}_1}]^T, \tag{3.32}$$

where $\mathbf{d}_m$ is the location of the $m$-th microphone, $\mathbf{q}_1$ represents a unit-norm vector pointing in the direction of the desired source, $c$ is the speed of sound and $\omega = 2\pi f = 2\pi k F_s (2K)^{-1}$ where $f$ is the frequency in Hertz with $F_s$ being the sampling frequency.

The Euclidean angle between between the separation filter $\mathbf{g}_1$ and the far-field steering vector $\mathbf{z}_1$ is defined as

$$\cos\zeta[k] = \frac{\mathrm{Re}\left\{\mathbf{g}_1^H[k]\mathbf{z}_1[k]\right\}}{\|\mathbf{g}_1[k]\|\,\|\mathbf{z}_1[k]\|} \tag{3.33}$$

A broadband penalty function which restricts the Euclidean angle between $\mathbf{g}_1[k]$ and $\mathbf{z}_1[k]$, steering the filter of interest $\mathbf{g}_1$ in the direction of the desired source is then given by

$$J_p(\mathbf{g}_1) = \sum_{k=1}^{K} [\cos\zeta[k] - 1]^2 . \tag{3.34}$$

Thus, the standard cost function of IVA (3.5) is augmented with the penalty term (3.34), resulting in the constrained IVA cost function

$$J_{civa}(\mathbf{G}) = J_{iva}(\mathbf{G}) + \lambda\, J_p(\mathbf{g}_1), \tag{3.35}$$

where $\lambda(\lambda \geqslant 0)$ is a penalty term. Taking the gradient of (3.35) with respect to the elements of the demixing matrix $\mathbf{G}[k]$, we obtain:

$$\nabla\mathbf{G}_{civa}[k] = \nabla\mathbf{G}_{iva}[k] + \lambda\nabla\mathbf{G}_p[k], \tag{3.36}$$

where $\mathbf{G}_{iva}[k]$ is the gradient of $J_{iva}$ and $\mathbf{G}_p[k]$ is the gradient of the penalty term (3.34). The penalty term is a function only of $\mathbf{g}_1[k]$ so that the gradient with respect to the filters coefficients is given by

$$\nabla\mathbf{G}_p = \begin{bmatrix} \nabla\mathbf{g}_1^H[k] \\ 0_{M-1\times M} \end{bmatrix}, \tag{3.37}$$

where $\nabla \mathbf{g}_1[k]$ is the gradient of the penalty function (3.34) with respect to $\mathbf{g}_1^*[k]$ and it is derived based on [13]. It is useful to define $C = 1/\left(\|\mathbf{g}_1[k]\| \cdot \|\mathbf{z}_1[k]\|^2\right)$ so that the gradient of the penalty term is given by

$$\nabla \mathbf{g}_1[k] = C \cdot \left[ (\cos\zeta[k] - 1) \left( \mathbf{z}_1[k] - \frac{\mathbf{g}_1[k]}{\|\mathbf{g}_1[k]\|^2} \operatorname{Re}\left\{ \mathbf{g}_1^H[k] \mathbf{z}_1[k] \right\} \right) \right]. \tag{3.38}$$

The CIVA ensures that the desired speech signal is always delivered at the output of the corresponding separation filter with small distortion and without the knowledge of the number of interferers. In contrast, the unconstrained IVA algorithm introduces higher distortion of the desired speech signal in non-determined and reverberant scenarios.

### 3.2.3 *Supervised IVA*

Nesta and Koldovskj proposed in [34] to extend the multidimensional source model of IVA by adding pilot components statistically dependent on the target and noise sources.

The injected pilot signals act as a prior knowledge enforcing the natural gradient to converge in a limited solution space, without imposing any explicit constraint to the demixing system. The extension follows from (3.18) in the standard IVA learning algorithm described in section 3.1.3. In (3.18) the denominator on the right-hand side corresponds to a factor that binds all the frequency bins together: without this factor, the decorrelation of the outputs will be achieved in each bins separately but the full wide-band source would be affected by the permutation problem. By following this observation, in [34] the adaptation to enforce another level of dependence, namely, between the separated components and pilot signals which are designed to capture high level spectral or spatial differences between the target and the interfering sources.

Nesta and Koldovskj propose to extend the multivariate model in (3.17), by injecting an additional "Pilot" component $P_i$ into the source vector $\mathbf{s}_i$ so that the exended source vector $\widetilde{\mathbf{s}}_i$ can be written as

$$\widetilde{\mathbf{s}}_i = [s_i[1], \cdots, s_i[K], \gamma P_i], \tag{3.39}$$

Thus, the multivariate source prior becomes

$$p(\widetilde{\mathbf{s}}_i) = \alpha \exp\left( - \sqrt{\sum_{k=1}^{K} |s_i[k]|^2 + \gamma^2 |P_i|^2} \right), \tag{3.40}$$

where $\gamma$ is a hyper-parameter controlling the influence of the "Pilots". By indicating with $\widetilde{\mathbf{y}}_i = [y_i[1], \cdots, y_i[K], \gamma P_i]$ the extended output

vector and by noting that $P_i$ is independent on $\mathbf{G}[k]$, the ML update is derived by using (3.40) and (3.5).
The new score function is obtained as:

$$\begin{aligned}\varphi[k](\widetilde{\mathbf{y}}_i) &= \varphi[k](y_i[1], \cdots, y_i[K], \gamma P_i) \\ &= \frac{\hat{s}_i[k]}{\sqrt{\sum_{k=1}^{K} |\hat{s}_i[k]|^2 + \gamma^2 |P_i|^2}},\end{aligned} \tag{3.41}$$

in which it is possible, by controlling $\gamma$, to trade the importance of the mutual frequency self-dependence versus the dependence on the pilot component $P_i$. In the extreme cases, if $\gamma$ is set to a small value, then the standard IVA is realized and the order of recovered sources would depend only on the initialization of $\mathbf{G}[k]$. On the other hand, if $\gamma$, is chosen to be a large value, the alignment of the frequency components is forced to follow the one of the pilot signal $P_i$. The component $P_i$ needs to be designed in order to be statistically dependent on the $i$-th source; a possible pilot signal can be defined as

$$P_i = p_i[n] \sqrt{\sum_{k=1}^{K} |x_i[n,k]|^2}, \tag{3.42}$$

where $p_i[n]$ is the posterior probability to observe the $i$th source at the STFT frame $n$. The posteriors can be estimated by learning the distributions of discriminative spectral or spatial features computed from the input mixture $\mathbf{x}[k]$. It has to be noted that (3.42) is only a special case which has been chosen by Nesta and Koldovskj to show a connection between the weighted ICA and S-IVA [34] but other pilots can be defined, with the constraint that they need to be dependent on the source components. The vector of the features are indicated as $\mathbf{v}[n] = [v_1[n] \cdots v_F[n]]$ and the source classes are defined as "$i = 1$" for the "desired" target source, "$i > 1$" for the "noise" sources. The parameters of a supervised classifier are learned beforehand from training data in order to produce the posteriors $p_i(n)$ associated to each class. Any sort of hard or soft classifier can be used, such as Gaussian Mixture Models, SVM or discriminatively trained Deep Neural Network [41].

# 4

# INFORMED IVA

To validate the Supervised IVA proposed by Nesta and Koldovskj, we decided at first to develop an "oracle" version of the algorithm in which the activation of the pilot(s) described in (3.42) is(are) decided by evaluating the energy content of the desired source(s) at a certain time frame $n$.

After the validation of the oracle algorithm, we implemented a version in which a Convolutional Neural Network (CNN) [17] was used to detect the DOAs of the sources at a certain time frame $n$ so that if the DOA of the source is detected then the source is considered active, the posterior probability is set to a value greater than zero and consequently the correspondent pilot component influences the basic IVA. At the end of the chapter we also propose some possibilities to model the pilot components and their activations.

## 4.1 SIVA BASED ON ORACLE DETECTION

In this realization of the SIVA, the pilot components have been implemented tracking the activation of the desired source by evaluating the clean (a.k.a. noiseless) speech source signal received at the reference microphone: the oracle term is used since we assumed to know the noiseless source signal received at the microphone, which is not the case in real-world applications in which we would only capture the mixture signal because of the the mutual interaction of the different sources and the interaction of the sound waves with the surrounding environment [11].

Since we investigate different models for the pilots, for ease of notation, we refer to the second term of the pilots, which is the weighting

introduced by the pilot, as $b_i[n]$ so that the pilots can be expressed as

$$P_i = p_i[n]b_i[n]. \tag{4.1}$$

### 4.1.1 *Noiseless pilot component*

In this version of the oracle-SIVA, the pilot components have been implemented using the noiseless signal of the desired source(s) received at the reference microphone, which is indicated by $\bar{x}_{i,1}$ where $i$ indicates that the desired source is the $i$-th source and the index $1$ is used to indicate that we are taking the reference microphone, so that $b_i[n]$ can be written as

$$b_i[n] = \sqrt{\sum_{k=1}^{K} |\bar{x}_{i,1}[n,k]|^2}. \tag{4.2}$$

The evaluation of the activation of the pilots $P_i$ is done by the use of a threshold $\xi_{SIVA}$ which is compared to $b_i[n]$, so that we can write the posterior $p_i[n]$ as

$$p_i[n] = \begin{cases} 1 & \text{if} \quad \sum_{k=1}^{K} |\bar{x}_{i,1}[n,k]|^2 \geqslant \xi_{SIVA}, \\ 0 & \text{otherwise}, \end{cases} \tag{4.3}$$

thus the pilot $P_i$ of the $i$-th source becomes active when $b_i[n]$ is greater than $\xi_{SIVA}$ and the algorithm would follow the additional information provided by $P_i$.

### 4.1.2 *Noisy pilot component*

In this version of the oracle-SIVA, the decision for the posteriors is the same as in the noiseless case, i. e. the decision is done using the noiseless signal of the desired source(s) received at the reference microphone, while the pilot information $b_i[n]$ is given by the contribution of all the frequency components given by the mixtures received by the microphones, so that we can write

$$b_i[n] = \sqrt{\sum_{k=1}^{K} |x_1[n,k]|^2}, \tag{4.4}$$

where the index $1$ indicates that we take only the the information given by the reference microphone while the posteriors $p_i[n]$ are given by (4.3).

## 4.2 SIVA BASED ON CNN ACTIVITY DETECTION

Using a Uniform Linear Array of microphones and assuming that the DOA of the desired source is known, we used a DOA estimator

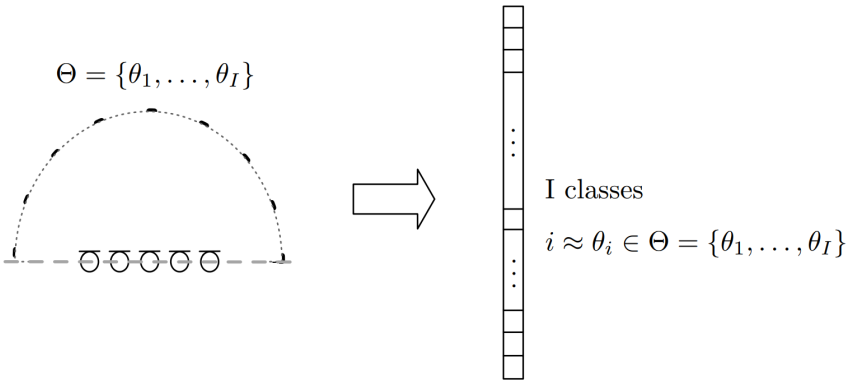



**Figure 4.1:** The multi-speaker DOA estimation problem is formulated as an I class multi-label classification problem. The i-th class corresponds to i-th DOA $\theta_i$: each class corresponds to a possible DOA among the set of possible DOA.

based on CNN [17] to establish whether the source is active or it is not so that we can use this information to decide the posteriors $p_i[n]$. We assume to approximately know $\theta_i$ the DOA of the desired i-th source, with a Region of Interest (ROI) $\Delta\theta_i$ which is $\theta_i$ - 5 $\leqslant \Delta\theta_i \leqslant$ $\theta_i$ + 5.

The assumption of the knowledge of $\Delta\theta_i$ is justified by the fact that our aim is to do source extraction: infact, this allow us to find a solution even in the case in which there are more sources than microphones. Note that the knowledge of $\Delta\theta_i$ is not necessary since the CNN-DOA estimator in [17] computes the probabilities to find a source in an angular region which goes from $[0, \pi]$ so that the algorithm is feasible for BSE and also for BSS (in the cases determined and over-determined cases).

### 4.2.1 *Multi-Speaker Localization CNN for Activity Detection*

To decide the posteriors $p_i[n]$ of the Informed IVA (IIVA) algorithm, we decided to use a Multi-Speaker Localizer which makes use of a Convolutional Neural Network [17] proposed by Chakrabarty and Habets. The reason for which we opted to utilize this CNN-based localizer is that given an angular ROI $\Delta\theta_i$ it allows us to determine whether the source in $\Delta\theta_i$ is active or if it is not: this can be useful for many applications in which the aim is that of performing source extraction knowing the angular region from where a speech signal is expected to be emitted, i. e. in video/audio calls using laptops, conferences halls where the speaker is in front of the microphone, hand-free communications systems, etc.

The localizer proposed by Chakrabarty and Habets is a CNN based supervised learning method for DOA estimation which aims to esti-

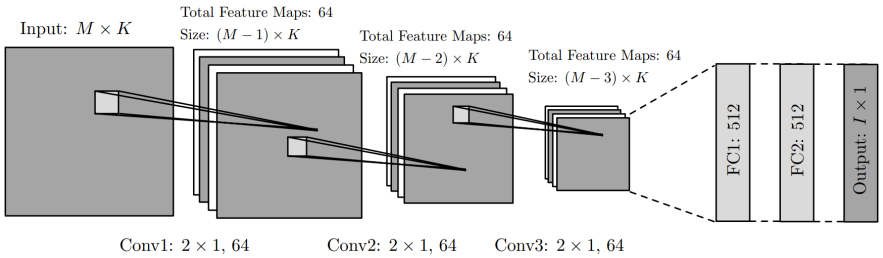



**Figure 4.2:** CNN architecture [17].

mate multiple DOAs per time frame given the STFT representation of the observed signals. The CNN has been trained using only synthesized white noise signals and by the use of a simple input representation it learns useful features during its training.

To realize the CNN-based localizer, considering an independent source DOA model, the multi-speaker DOA estimation problem has been translated to an I class multi-label classification problem as shown in Fig. 4.1: the range of possible DOAs is discretized into I discrete values to obtain a set of possible DOA values $\Theta = [\theta_1 \cdots \theta_I]$ so that each one of the classes corresponds to a possible DOA value among the set $\Theta$.

Given the input at each time frame the objective is to compute the probability for each of the I classes using I binary classifiers. The input of the CNN is given by features coming from the STFT representation of the microphone signal: these observed signals can be written in their phasor representation as

$$Y_m[n,k] = A_m[n,k]e^{j\phi_m[n,k]}, \tag{4.5}$$

where $A_m[n,k]$ represents the magnitude component and $\phi_m[n,k]$ denotes the phase component of the STFT coefficient of the received signal at the m-th microphone for the n-th time frame and k-th frequency bin. The input feature for the n-th time frame is formed by arranging $\phi_m[n,k]$ for each time-frequency bin $[n,k]$ and each microphone m into a matrix of size M × K, which Chakrabarty and Habets called the phase-map. The main assumption on which the method relies is that of the speakers not being simultaneously active per time-frequency unit: this condition is also known as W-disjoint orthogonality and it has been shown to hold approximately for speech signals in [36].

With the phase map as input, the task of the CNN is to generate the posterior probabilities for each of the DOA classes: indicating with $\Phi[n]$ the phase-map at the n-th time frame, then the posterior probability provided by the CNN at the output is indicated as $p(\theta_i|\Phi[n])$, where $\theta_i$ is the DOA corresponding to the i-th class.

In Fig. 4.2, is shown the CNN architecture employed. In the convolution layers (Conv layers in Fig. 4.2), small filters (also known as local filters) of size $2 \times 1$ are applied to learn the local correlations between the phase components of neighboring microphones at local frequency regions. The learned local structures are then eventually combined by the fully connected layers (FC layers in Fig. 4.2) for the final classification task.

Finally, since what we are doing in our work is to use the CNN as an activity detector, it is sufficient to evaluate if the desired source is revealed or if it is not at a certain time frame $n$ and this can be done since we assumed to know the ROI of the desired source $\Delta\theta_i$. Thus we are able to decide whether the pilot of the i-th source is active or not by assigning a value to the posterior $p_i$ after comparing the posterior probability $p(\Delta\theta_i|\Phi[n])$ of the DOA to come from $\Delta\theta_i$ with a threshold level $\xi_{SIVA}$ which lies in the interval $[0, 1]$. In the following section we propose two models to assign the posterior probabilities $p_i[n]$ of the pilots $P_i$ and two models for the second term $b_i[n]$ of the pilots.

### 4.2.2 *Pilots modeling*

If we call $p_{DOA}[n]$ the posterior probability of the i-th source to be active at the n-th time frame in the set of DOAs $\Phi[n]$, then we can simply evaluate to set the posteriors $p_i$ of the SIVA by comparing it to the threshold $\xi_{SIVA}$.

We propose two models to assign the posteriors $p_i[n]$ and 2 models for the term $b_i[n]$ of (4.1) that will be described in the following section.

#### 4.2.2.1 *Posterior probability modeling*

We propose two models for the pilots: a *soft* version and a *hard* version. In the following paragraphs we give a formal description of the two models.

##### SOFT PILOT ACTIVATION

In this version we assign the value of the posterior $p_{DOA}[n]$ of the ROI $\Delta\theta_i$ given by the CNN to the posteriors $p_i[n]$ of the pilot components. When the probability $p_{DOA}[n]$ is greater than a given threshold $\xi_{SIVA}$ then the posterior $p_i[n]$ of the i-th desired source is set to the same value $p_{DOA}[n]$ given by the CNN while if the probability of the i-th desired source is lower than $\xi_{SIVA}$ than the posterior $p_i[n]$ is set to zero; the probability of the undesired sources is distributed equally given that the remaining probability is equal to $1 - p_i[n]$. If we suppose to

be interested in extracting only one source and we indicate with $p_1[n]$ the posterior probability associated to the desired source which DOA is $\Delta\theta_1$, it is possible to write the model for the soft version as

$$p_1[n] = \begin{cases} p_{DOA}[n], & \text{if} \quad p_{DOA}[n] \geqslant \xi_{SIVA}, \\ 0, & \text{otherwise}, \end{cases} \tag{4.6}$$

$$p_{i\neq 1}[n] = \frac{1 - p_1[n]}{1 - L}.$$

Notice that in (4.6), when the probability $p_{DOA}[n]$ is lower than the threshold $\xi_{SIVA}$ than the posterior $p_1[n]$ is set to zero and the other sources are set so that $p_{i\neq i}[n] = \frac{1}{1-L}$ where $L$ is the number of sources.

#### HARD PILOT ACTIVATION

In this version we set the posterior $p_1[n]$ of the desired source to 1 when the probability $p_{DOA}[n]$ of the source to be in the ROI $\Delta\theta_i$ is above the threshold $\xi_{SIVA}$ and to zero all the the unwanted sources so that it is possible to write

$$p_1[n] = \begin{cases} 1, & \text{if} \quad p_{DOA}[n] \geqslant \xi_{SIVA}, \\ 0, & \text{otherwise}, \end{cases} \tag{4.7}$$

$$p_{i\neq 1}[n] = \frac{1 - p_1[n]}{1 - L}.$$

As in the soft version, when the probability $p_DOA[n]$ is lower than the threshold $\xi_{SIVA}$ than the posterior $p_1[n]$ is set to zero and the other sources are set so that $p_{i\neq i}[n] = \frac{1}{1-L}$ where $L$ is the number of sources.

### 4.2.2.2 *Pilot weighting modeling*

We propose two models for the term $b_i[n]$ in (4.4): the *All-mics* version and the *Single-mic* version with their formal description in the following paragraphs.

#### SINGLE-MICROPHONE

This is is the first of the two proposed models for the term $b_i[n]$. In the *Single-mic* version we model the additive term introduced by the pilots with the use of the information captured only by the reference microphone, that we indicate as $x_1$, so that we can write $b_i[n] = \overline{b}_i[n]$ where $\overline{b}_i[n]$ is given by

$$\overline{b}_i[n] = \sqrt{\sum_{k=1}^{K} |x_1[n,k]|^2}. \tag{4.8}$$

In (4.8) the term $b_i[n]$ is modeled as the sum of all the frequency content of the observation captured at the reference microphone $x_1$ at the n-th time-frame.

ALL-MICROPHONES

In the second version we model the term $b_i[n]$ by using the observation given by all the microphones so that $b_i[n] = \widetilde{b}_i[n]$, where $\widetilde{b}_i[n]$ is defined as

$$\widetilde{b}_i[n] = \sqrt{\frac{1}{M}\sum_{j=1}^{M}\sum_{k=1}^{K}|x_j[n,k]|^2}, \tag{4.9}$$

In (4.9) the information is provided by the weighted average over all the observation set and all the frequency components. In chapter 5 a comparison to evaluate which is the better model is conduced.

#### 4.2.2.3 *Gradient update normalization*

To improve the convergence properties of the algorithm, we decided to modify the gradient update rule in (3.20) with a normalization term as in [27] so that the normalized update rule becomes

$$g_{ij}[n+1,k] = g_{ij}[n+1,k] + \eta\frac{\Delta g_{ij}[n,k]}{||\Delta g_{ij}[n,k]||_F}. \tag{4.10}$$

# 5

# EXPERIMENTS

In this chapter we describe the experiments which have been conducted to test the properties of the SIVA, starting from a brief description of the preliminary steps which involve the 2 oracle versions described in the previous chapter which are compared to the proposed method in order to verify if the algorithm works in more realistic scenarios. Successively we describe the experiments that we had to understand the behavior of the proposed method depending on the influence of the parameters which characterize it and which are the best tuning of these parameters. We investigate the amount of time necessary to accomplish the separation depending on the learning rate and number of iterations, to assess the online separation properties of the proposed method. Furthermore, we compare the proposed algorithm with the CIVA [27] from the state-of-the-art of the IVA algorithms for online source extraction.

## 5.1 SETTING AND PERFORMANCE CRITERIA

To evaluate the proposed algorithm, several experiments were conducted using a simulated room which dimensions are $7.5 \times 5.5 \times 3$ with a $T_{60} = 150\text{ms}$ on a 20s speech segment with 2 different configurations for the source positions:

1) In configuration 1, as depicted in Fig. 5.1(a), we have a speech signal of an English woman and another speech signal of an English man. The signal coming from the woman is positioned at 35° with respect to the reference microphone of the ULA while the man is positioned at 75°.

2) In configuration 2, as depicted in Fig. 5.1(b), the speech signal of the English woman is positioned at 130° with respect to the

reference microphone of the ULA while the speech signal of the English man is positioned at 65° .

The number of sources used for the investigations is equal to 2, so that the tuning of all the parameters has been investigated for this particulat case which is an over-determined case (more microphones than sources).

The sensors used are simulated microphones, more precisely we simulated a Uniform Linear Array (ULA) [12] composed by 4 omni-directional microphones, where the inter-microphone distance $d_{mic}$ is 0.08m and the ULA hase been positioned at the center of the room as shown in Fig. 5.1.

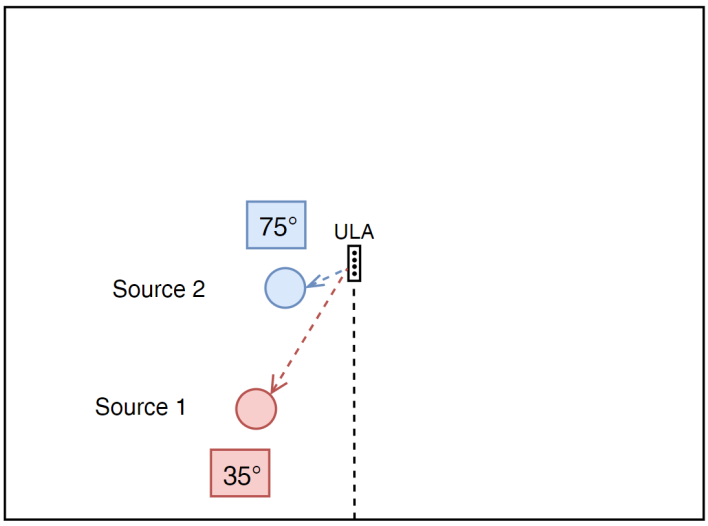
The quality of the desired speech signal at the output of the proposed algorithm was evaluated using simulated audio data: to obtain the microphone signals, we convolved clean speech signals sampled at 16 kHz with simulated room impulse responses. The room impulse responses (RIRs) were generated using [22]. The implementation has been done in the time-frequency domain using the STFT. The STFT frame size is of 512 samples with 50% overlap. In all the experiments, a diffuse noise with 30dB signal-to-noise ratio (SNR) and a sensor noise with 40dB SNR was added to the microphone signals. The signal-to-interference (SIR) ratio and speech distortion index (SD) where used as defined in [28]. The performance was measured by the improvement of signal-to-interference ratio (SIR), which is the difference between input and output SIR, defined as $SIR_{improvement} = SIR_{out} - SIR_{in}$ in [28] where

$$SIR_{in} = 10\log\left(\frac{\sum_{n,k}|\bar{x}_i[n,k]|^2}{\sum_{n,k,i}\left|\sum_{j\neq i}\bar{x}_j[n,k]\right|^2}\right),$$
$$SIR_{out} = 10\log\left(\frac{\sum_{n,k,i}|\bar{y}_i[n,k]|^2}{\sum_{n,k,j}\left|\sum_{j\neq i}\bar{y}_j[n,k]\right|^2}\right). \tag{5.1}$$

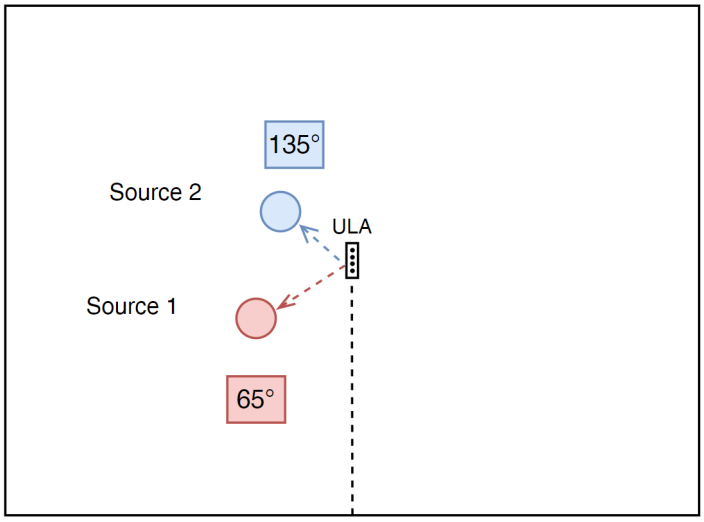
While the SD is given by

$$SD = \frac{1}{n}\left(\frac{\sum_{n,k}|\bar{x}_i[n,k]-\bar{y}_i[n,k]|^2}{\sum_{n,k}|\bar{x}_i[n,k]|^2}\right) \tag{5.2}$$

Unless stated otherwise, for each configuration we measured the performances taking as the desired source individually each of the sources and then we averaged the results to obtain the performances for a specific configuration.




**(a)** Configuration 1




**(b)** Configuration 2

**Figure 5.1:** In (a) Configuration 1: source 1 is positioned at 75° with respect to the center of the microphone array while source 2 at 35°. In (b) Configuration 2: source 1 is positioned at 65° while source 2 is positioned at 135°.

## 5.2 SIVA: PILOT INFLUENCE

The novelty introduced by Nesta and Koldovskj is that of the pilot components. As already mentioned in section 3.2.3, the pilots need to be statistically dependent on the sources: for this reason we decided to try to investigate which pilot could be used. The pilots in (3.42) are dependent on the posteriors $p_i[n]$ and a second term which is the contribution of all the frequency components given by the observation set $\sqrt{\sum_{k=1}^{K} |x_i[n,k]|^2}$. Both this terms can be designed arbitrarily with the only constrain that the pilots should be statistically dependent from the sources.

We investigated some possibilities for the design of the pilots, starting from simple "oracle" cases (assuming some prior knowledge about the source distributions) to more realistic cases and evaluating how the algorithm is influenced by them. In the experiments in this section, unless clearly stated, the learning rate $\eta$ has been set to 8, the threshold $\xi_{SIVA}$ to evaluate the power of the desired active source signal was set to -40 dB and in both algorithms the parameter $\beta_{SIVA}$ was set to 0.35 which is the value which provides best performances for both oracle versions.

### 5.2.1 *SIVA versions comparison*

In this experiment we compared the two oracle versions of the SIVA and the proposed CNN-based version: we report the best performances achieved using the noiseless pilot, the noisy pilot versions and the CNN-based SIVA defined in section 4.1.1, the section 4.1.2 and in section 4.2 respectively.

In table 5.1 we can see that the noisy oracle version, i. e. the version in which the pilots are given from the observation signals, provides slightly lower performances than the noiseless version and that, as expected, the CNN-based version provides lower performances than the oracle versions but good results are still achieved.

The experiments have been done for the two different configurations as usual. These results show that the CNN-based SIVA can be used in real world scenarios: the influence of the pilots allows to have a great boost in the performances with respect to the standard IVA.

### 5.2.2 *Conclusions*

We analyzed the performances of various SIVA versions starting from the Oracle-versions which are using informations about the source signals (which are not normally available in real world applications)

| | Algorithm | $\beta_{SIVA}$ | $SIR_{imp}$ | SD |
|---|---|---|---|---|
| config1 | $SIVA_{Noiseless}$ | 0.35 | 7.29 | 0.17 |
| | $SIVA_{Noisy}$ | 0.35 | 6.97 | 0.19 |
| | $SIVA_{CNN}$ | 0.6 | 6.75 | 0.17 |
| | IVA | – | 1.92 | 0.31 |
| config2 | $SIVA_{Noiseless}$ | 0.35 | 8.76 | 0.19 |
| | $SIVA_{Noisy}$ | 0.35 | 8.42 | 0.17 |
| | $SIVA_{CNN}$ | 0.5 | 8.15 | 0.17 |
| | IVA | – | 2.15 | 0.33 |

**Table 5.1:** Comparison in terms of SIR improvement and SD between the noiseless-oracle, the noisy-oracle SIVA versions, the CNN-based SIVA and the IVA.

to the CNN-based version which is possible to use in a real world scenario. Our results shows that good performances are obtained in realistic scenarios in which only the ROI is known and no other additional information is available. It is important to note that the standard IVA most of the times diverged, thus not arriving to a proper solution: for this reason, in the following experiments, the standard IVA will not be considered for comparison. The algorithm needs some prior tuning in order to work properly: this led us to investigate, in the next section, how the various parameters influence the CNN-based SIVA.

## 5.3 SIVA: PARAMETERS INVESTIGATION

In this section we show the results of the experiments that we conducted to evaluate the behavior of the CNN-based SIVA, which we refer to as IIVA, using the different models proposed in section 4.2.2 and also in function of the various hyper-parameters which compose it. Since the oracle versions use information gathered from the noiseless source signals, their tuning is totally different from the tuning of the IIVA: in the next we report and discuss only the IIVA tuning.

### 5.3.1 *Soft vs Hard pilot activation*

In this experiment we compare the performances given by two different models for the posterior probability $p_i[n]$ in (3.41) which are computed through the use of the CNN-DOA localizer [17] explained in

| | Algorithm | η | Iterations | $\beta_{SIVA}$ | $SIR_{imp}$ | SD |
|---|---|---|---|---|---|---|
| config1 | $SIVA_{soft}$ | 8 | 3 | 0.6 | 6.64 | 0.17 |
| | $SIVA_{hard}$ | 8 | 3 | 0.6 | 6.65 | 0.17 |
| | $SIVA_{soft}$ | 10 | 1 | 0.5 | 5.71 | 0.19 |
| | $SIVA_{hard}$ | 10 | 1 | 0.5 | 5.73 | 0.19 |
| config2 | $SIVA_{soft}$ | 2 | 8 | 0.6 | 8.15 | 0.17 |
| | $SIVA_{hard}$ | 2 | 8 | 0.6 | 8.19 | 0.17 |
| | $SIVA_{soft}$ | 10 | 1 | 0.5 | 7.90 | 0.17 |
| | $SIVA_{hard}$ | 10 | 1 | 0.5 | 7.91 | 0.17 |

**Table 5.2:** Comparison of the performances in terms of SIR improvement and SD given by the *Soft* and the *Hard* model of the posteriors with the tuning of the parameters used to obtain the best performances for configuration 1 (top) and configuration 2 (bottom)

section 4.2.1; the posteriors given by the CNN have been normalized at each time frame so that $\sum_{i=1}^{I} p_i[n] = 1$ for each time-frame. We investigated the version proposed for the posteriors in section 4.2.2.1: the *soft* pilot activation and the *hard* pilot activation. The setting of the parameters are reported with the achieved results in table 5.2: as we can see from the results obtained, the performances of the IIVA using the different models for the activation of the posterior probabilities are practically the same for both models.

Since the performances are practically the same for both versions, we decided to use the hard activation version for all the successive experiments: this decision has been taken because using the hard version can better show the behaviors caused by other parameters or weightings like the $\beta_{SIVA}$ and the $\gamma_{SIVA}$ parameters that are investigated in the following sections.

### 5.3.2 *Single-microphone vs All-microphones*

In this experiments we compared the noisy CNN-based SIVA with two different models for $b_i[n]$ that we named *Single-microphone* and *All-microphones* versions in section 4.2.2.2. The difference between the two investigated versions relies in the number of observations used for the additive normalization term introduced by the pilot as given in (3.41). As usual, the performances have been investigated for the both configurations: in table 5.3 we show the SIR and SD obtained for each proposed version with the value of the parameters which gives the best performance.

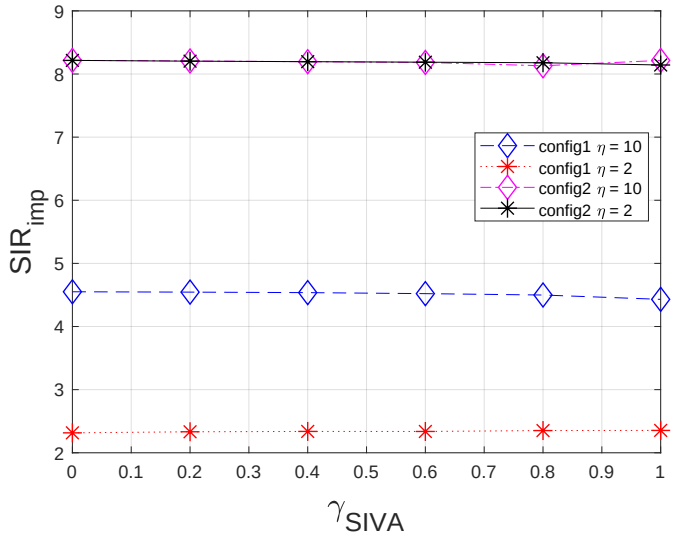
| | Algorithm | $\eta$ | Iterations | $\beta_{SIVA}$ | $SIR_{imp}$ | SD |
|---|---|---|---|---|---|---|
| config1 | $IIVA_{Single\text{-}Mic}$ | 8 | 3 | 0.6 | 6.43 | 0.17 |
| | $IIVA_{All\text{-}mics}$ | 8 | 3 | 0.6 | 6.62 | 0.17 |
| | $IIVA_{Single\text{-}Mic}$ | 10 | 1 | 0.5 | 5.53 | 0.19 |
| | $IIVA_{All\text{-}mics}$ | 10 | 1 | 0.5 | 5.71 | 0.19 |
| config2 | $IIVA_{Single\text{-}Mic}$ | 2 | 8 | 0.6 | 8.10 | 0.17 |
| | $IIVA_{All\text{-}mics}$ | 2 | 8 | 0.6 | 8.18 | 0.17 |
| | $IIVA_{Single\text{-}Mic}$ | 10 | 1 | 0.5 | 7.93 | 0.17 |
| | $IIVA_{All\text{-}mics}$ | 10 | 1 | 0.5 | 7.99 | 0.16 |

**Table 5.3:** Comparison of the performances in terms of SIR improvement and SD given by the *Single-mic* and the *All-mics* models with the tuning of the parameters used to obtain the best performances for configuration 1 (top) and configuration 2 (bottom).

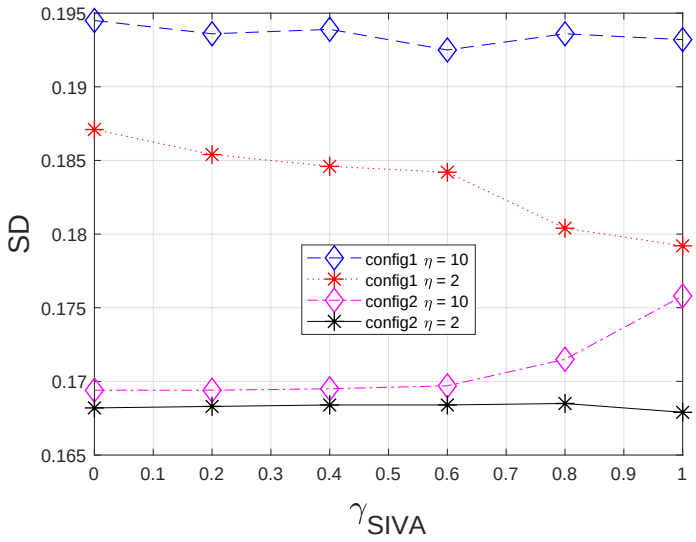
As we can see, the all-microphones version gives better performances in terms of SIR improvement while the SD has similar values for both models: in the following experiments we chose to use the All-microphones version which takes into account all the available information given by all the observation set.

### 5.3.3 *Weighting parameter influence*

In this experiment we evaluate the influence of $\gamma_{SIVA}$ in (3.41) which is a weighting hyper-parameter that allows us to vary the amount of dependence on the pilot component $P_i$. If $\gamma_{SIVA}$ is set to zero, then the standard IVA is realized and thus the separation of the sources would depend only on the initialization of the demixing matrix $\mathbf{G}$ while if $\gamma_{SIVA}$ is chosen to be a large value, the alignment of the frequency components is forced to follow the one of the pilots. In Fig. 5.2(a) and Fig. 5.2(b) are respectively reported the SIR improvement and the SD of the IIVA for a set of values of $\gamma_{SIVA} \in [0, 1]$ for both configurations and for 2 different learning rates, while in Fig. 5.3(a) and Fig. 5.3(b) we have the same setting for a set of $\gamma_{SIVA} \in [2, 20]$. We can see that $\gamma_{SIVA}$, so the weight of the pilots, does not influence much the performances of the algorithm. Nevertheless, we can notice that a large value for $\gamma_{SIVA}$ results in slightly worst performances in terms of SIR improvement apart in the case in which the learning rate is set to 10 for configuration 1 (blue dashed line) in Fig. 5.3(a) in which the performances get clearly worst for high values of $\gamma_{SIVA}$. We can also observe in Fig. 5.3(a) that smaller values of $\gamma_{SIVA}$ do not show any

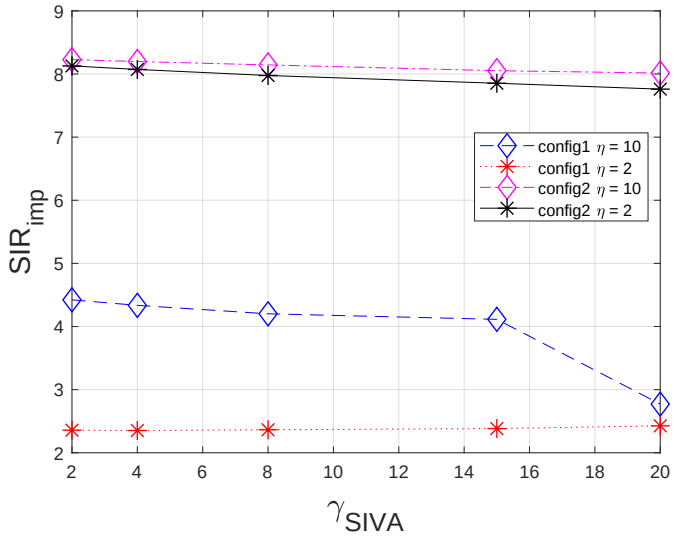



**(a)** SIR comparison

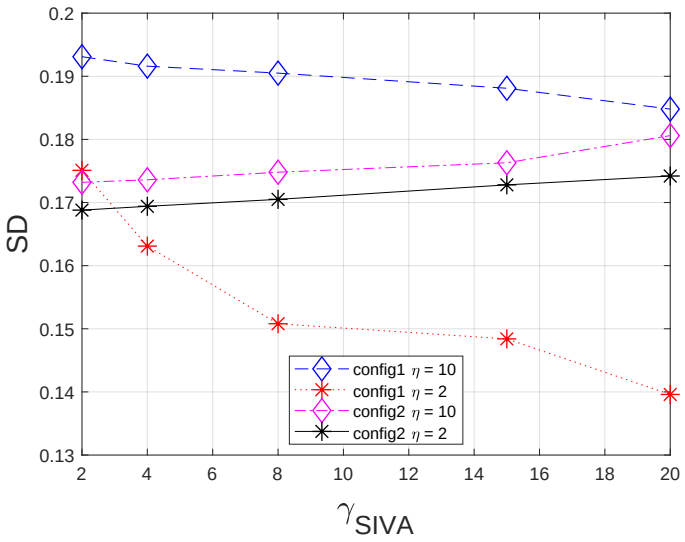



**(b)** SD comparison

**Figure 5.2:** Performances in terms of SIR improvement (a) and SD (b) given by varying the weighting parameter $\gamma_{\text{SIVA}}$ for a set of values $\in [0, 1]$ for configuration 1 (config1) and configuration 2 (config2).




**(a)** SIR comparison




**(b)** SD comparison

**Figure 5.3:** Performances in terms of SIR improvement (a) and SD (b) given by varying the weighting parameter $\gamma_{\text{SIVA}}$ for a set of values $\in [2, 20]$ for configuration 1 (config1) and configuration 2 (config2).

improvement nor different behavior in terms of SIR while the SD in Fig. 5.3(b) shows similar results.

### 5.3.4 *Smoothing parameter influence*

In this experiment we investigated a modified version of the SIVA algorithm, proposed by Nesta and Koldovskj in [34]. The investigated version is an heuristic modification which is given by the introduction of the smoothing parameter $\beta_{\text{SIVA}}$ which influences the impact of the standard IVA normalization given by the first and the second members of the denominator in (3.41) instead of the parameter $\gamma_{\text{SIVA}}$ in section 5.3.3 which gives weight only to the second term of the denominator. Using the parameter $\beta_{\text{SIVA}}$, the modified extended score function becomes

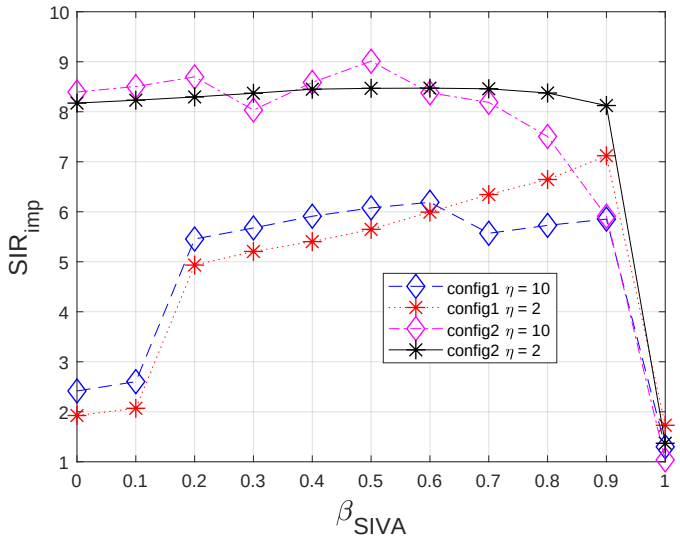
$$\varphi[k](\widetilde{\mathbf{y}}_i) = \frac{\hat{s}_i[k]}{\sqrt{(1-\beta_{\text{SIVA}})\sum_{k=1}^{K}|\hat{s}_i[k]|^2 + \beta_{\text{SIVA}}\,|P_i|^2}}, \tag{5.3}$$

where $\beta_{\text{SIVA}}$ is a trade-off parameter which is set in the range between 0 and 1 in order to weight the influence of the pure IVA versus the weight of the pilot. In the extreme cases, i. e. when $\beta_{\text{SIVA}}$ is equal to 0 or when it is equal to 1, we have that the score correlation in (3.18) becomes

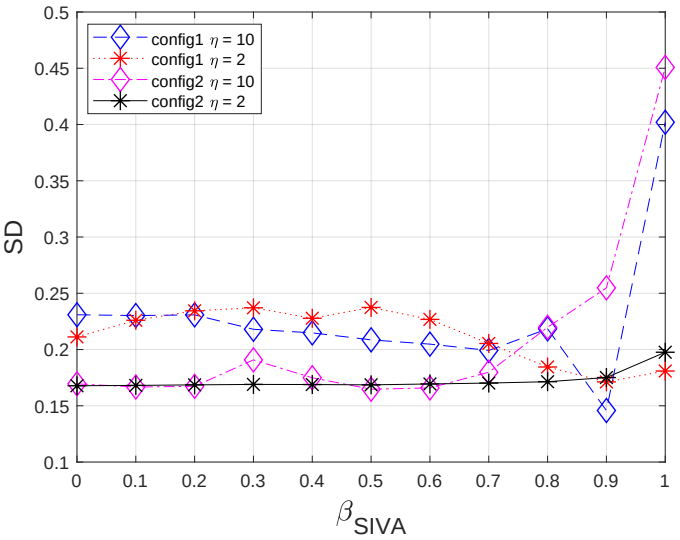
$$\varphi[k](\widetilde{\mathbf{y}}_i) = \begin{cases} \frac{\hat{s}_i[k]}{\sqrt{\sum_{k=1}^{K}|\hat{s}_i[k]|^2}}, & \text{if} \quad \beta_{\text{SIVA}} = 0, \\ \frac{\hat{s}_i[k]}{\sqrt{|P_i|^2}}, & \text{if} \quad \beta_{\text{SIVA}} = 1, \end{cases} \tag{5.4}$$

so that when $\beta_{\text{SIVA}}$ is 0, then the standard IVA is realized while when $\beta_{\text{SIVA}}$ is 1 the algorithm follows only the information provided by the pilots.

As we can see from Fig. 5.4, the experiments have been conducted for different learning rates and for both configurations. In Fig. 5.4(a) and Fig. 5.4(b) are respectively the SIR improvement and the SD of the IIVA for $\eta = 2$ and $\eta = 10$ for both usual configurations. It is interesting to notice that the influence of the parameter $\beta_{\text{SIVA}}$, and thus of the pilot components, has more impact for configuration 1 with respect to configuration 2 in which a good perfomance is obtained also in the case of the basic IVA (which is the case of $\beta_{\text{SIVA}} = 0$). The additional information of the pilots, smoothed by the parameter $\beta_{\text{SIVA}}$, seems to be of great help whenever there is a mismatch in the power of the signal which are captured by the ULA: in fact as we can see in Fig. 5.1(a) in configuration 1, source 1 (at 35° with respect to the ULA) is almost 3 times further than source 2 (at 75°) so the signal coming from source 1 would be much more attenuated when received by the ULA then in the case of configuration 2 in Fig. 5.1(b). The

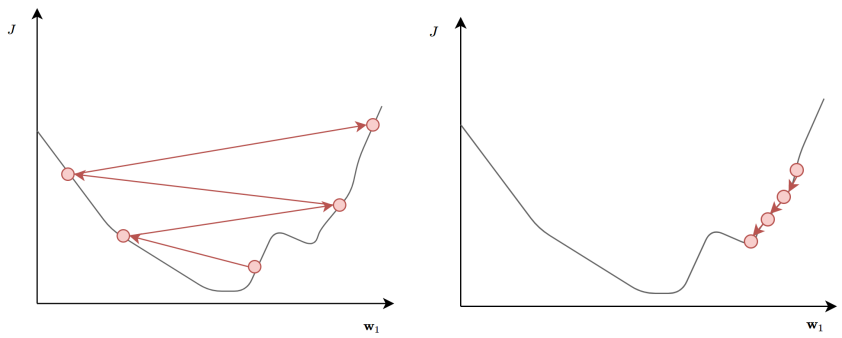



**(a)** SIR




**(b)** SD

**Figure 5.4:** The upper window (a) shows the Signal-to-Interference Ratio improvement with respect to the variations of the hyper-parameter $\beta_{SIVA}$ while in the lower window (b) we can see correspondent Signal Distortion measurements of the SIVA algorithm




(a) Large η (b) Small η

**Figure 5.5:** Representation of the influence of the value of the learning rate in a non-convex optimization problem. For a large learning rate (a) we might diverge from the optimal solution while with a small learning rate (b) the optimization process might get into a local minima.

difference with the parameter $\gamma_{SIVA}$, that we investigated in section 5.3.3, is that $\beta_{SIVA}$ allows to smooth the importance of following the pure IVA versus the importance of the pilots. For configuration 1, as we can see in Fig. 5.4(a), the $\beta_{SIVA}$ parameter shows an important improvement of the performances with respect to the performances given by $\gamma_{SIVA}$ reported in the previous section section 5.3.3. For this reason, for the following experiments we took in account the version using the $\beta_{SIVA}$ parameter.

### 5.3.5 *Learning rate*

As in every gradient-based update algorithm, the choice of the learning rate is crucial: for the gradient descent to work we must set η (learning rate) to an appropriate value. The learning rate determines how fast or slow we move towards the optimal weights but it also influences the achievable performance: if η is very large we would follow the gradient descent direction with a big step-size so that the number of iterations needed to reach the minimum would be small, causing the algorithm to rapidly converge, but on the other hand there could be the risk to skip the optimal solution because the large step-size would not allow the descent to reach the global minimum and in some cases we might also diverge if the learning is really high as shown in Fig. 5.5(a). In the case in which η is a really small value, we may need many iterations to converge to the global minimum so that the algorithm would be really slow to arrive to convergence and, furthermore, there might also be the risk to get trapped in some local minima as shown in Fig. 5.5(b). The choice of the learning rate is

| | $\eta$ | Iterations | $SIR_{imp}$ | SD | Time(s) |
|---|---|---|---|---|---|
| config1 | **0.5** | 35 | 7.37 | 0.17 | 424.67 |
| | **1** | 22 | 7.31 | 0.17 | 282.43 |
| | **2** | 12 | 7.28 | 0.17 | 239.42 |
| | **5** | 4 | 7.24 | 0.18 | 58.92 |
| | **8** | 3 | 7.07 | 0.18 | 35.92 |
| | **10** | 1 | 6.84 | 0.17 | 24.78 |
| | **15** | 1 | 6.57 | 0.19 | 24.78 |
| | **20** | 1 | 5.94 | 0.19 | 24.78 |
| | **30** | 1 | 5.67 | 0.20 | 24.78 |
| config2 | **0.5** | 18 | 8.71 | 0.16 | 245.67 |
| | **1** | 10 | 8.69 | 0.16 | 222.01 |
| | **2** | 8 | 8.56 | 0.17 | 129.29 |
| | **5** | 2 | 8.51 | 0.17 | 52.24 |
| | **8** | 2 | 8.28 | 0.17 | 48.56 |
| | **10** | 1 | 8.19 | 0.17 | 25.14 |
| | **15** | 1 | 8.12 | 0.19 | 25.14 |
| | **20** | 1 | 7.38 | 0.19 | 25.14 |
| | **30** | 1 | 5.29 | 0.26 | 25.14 |

**Table 5.4:** Comparison of the performances given by varying the learning rate $\eta$. On the right is reported the time spent to carry the whole procedure which is dependent on the number of iterations required to reach the best convergence possible at that specific learning rate.

crucial for all the algorithms which make use of it and IVA makes no exception from this point of view.

We investigated the learning rate to understand what is the rate which allows a good convergence while keeping the number of iterations low: this condition, in fact, is crucial to implement whichever real-time application since it determines the speed of convergence of the algorithm.

In table 5.4 we report the investigation on the learning rate $\eta$ for a set of values in the interval $[0.5, 30]$: for each learning rate, by using the minimum number of iterations necessary to converge, we reached the maximum SIR improvement possible for that specific learning rate. The parameter $\beta_{SIVA}$ has been tuned to achieve the best possible

performance. As we can see in table 5.4, choosing a small $\eta$ allows us to obtain a slightly better performance in terms of SIR but on the other hand the amount of time required to accomplish the optimization is highly increased.

For learning rates higher than 10, the best performances achievable start to get much worse: for high learning rate values, if we increase the number of iterations the performances get worst, similarly to the case of large learning rate in Fig. 5.5(a), and if we increase to values above 100 it is likely to obtain a divergent (so not valid) solution already at the first iteration. It is also worth to notice that for $\eta = 10$ only 1 iteration is required: while there is a slight decrease in the performance in terms of SIR, the time required to compute the optimization is much decreased, allowing a possible real-time implementation.

### 5.3.6 *Conclusions*

In this section we investigated the influence of the various parameters on the performances of the proposed methods. We first verified that the method works by using the oracle-versions in section 4.1, at the beginning with ideal conditions, i. e. knowing the noiseless signal of the desired source captured at the ULA, and later on we validated the oracle version using the true observations for the pilots.

After the assessment of the oracle version, we used a CNN localizer to verify if, in a known desired ROI, the source is active or not in a certain time-frame and to compute the posterior probabilities needed for the IIVA implementation.

We investigated various models for the pilot components of the algorithm and we studied the parameters which influence the behavior of the algorithm: we discovered that the parameter $\beta_{\text{SIVA}}$ helps to improve the performance of the IIVA as shown in section 5.3.4, with a high boost in terms of SIR in the cases in which there is a relevant difference between the power of the interferences and the power of the desired source to be extracted (due to different distance with respect to the ULA). Furthermore, in section 5.3.5 we verified that the choice of a proper learning rate is crucial in order to obtain good performances while keeping low the required time for the optimization and that the proposed algorithm is suitable for real-time implementations.

| | | Algorithm | $\beta_{\text{SIVA}}$ | $\text{SIR}_{\text{imp}}$ | SD |
|---|---|---|---|---|---|
| config1 | **source 1** | CIVA | – | 5.92 | 0.37 |
| | **source 1** | IIVA | 0.6 | 6.99 | 0.17 |
| | **source 2** | CIVA | – | 4.57 | 0.38 |
| | **source 2** | IIVA | 0.6 | 4.46 | 0.02 |
| config2 | **source 1** | CIVA | – | 7.91 | 0.32 |
| | **source 1** | IIVA | 0.6 | 9.85 | 0.33 |
| | **source 2** | CIVA | – | 11.25 | 0.18 |
| | **source 2** | IIVA | 0.6 | 8.23 | 0.17 |

**Table 5.5:** Comparison in terms of SIR improvement and SD between the noisy-oracle and the CNN-based SIVA versions.

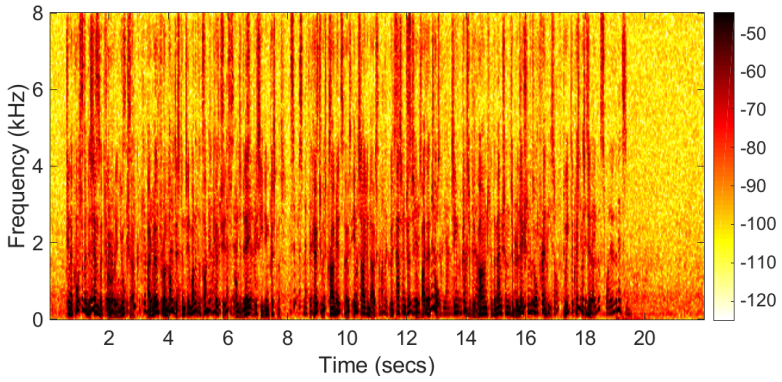
## 5.4 ALGORITHMS COMPARISON

In this section we compare the performances of the CIVA [27] and the proposed IIVA. The comparison has not been done with respect to the standard IVA [28] for the following reasons:

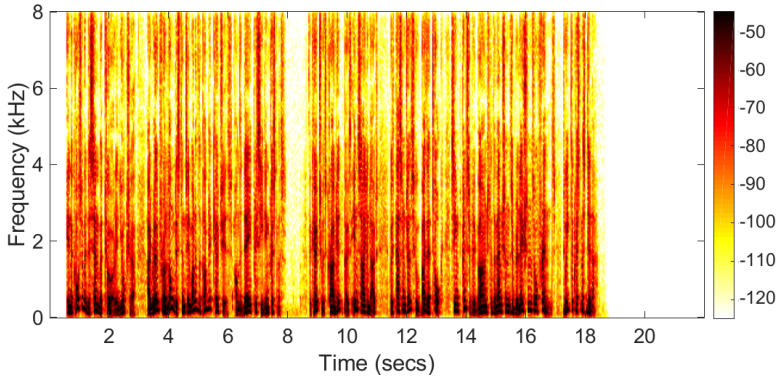
(i) The standard IVA is not suitable for source extraction since the order of the solution at the output suffers from the so called global permutation problem described in section 2.2.1 so that it is impossible to know which source has been extracted at a certain output.

(ii) The standard IVA is unstable and frequently diverges, specially for learning rates larger than 1.

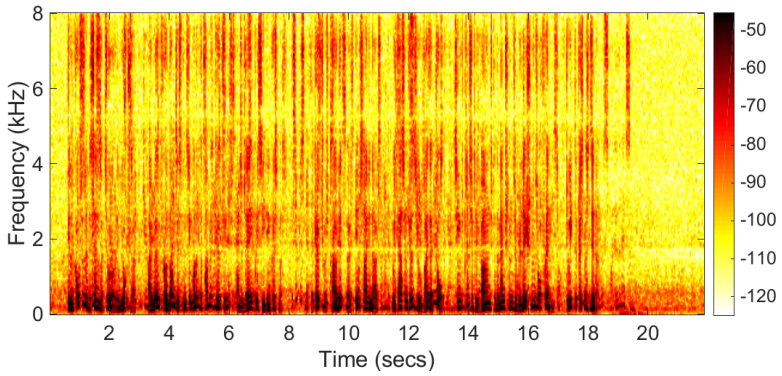
In our experiments both the compared algorithms have been tuned to obtain the best performances possible. In table 5.5, considering the case of configuration 1 (config1) in Fig. 5.1(a) in which there is a mismatch in the power emitted by the sources due to the different distances with respect to the ULA position and the difference of the DOA is of 40°, in the case in which the desired source is source 1, the proposed method performs better than the CIVA while when the desired source is the second, the CIVA shows a similar performance in terms of SIR improvement, while the SD of the IIVA is much lower than the one of CIVA. In the case of configuration 2 (config2) in Fig. 5.1(b) where the sources are emitting from similar distances and the DOA difference is of 65°, for source 1, the IIVA shows +2 dB in terms of SIR improvement compared to the one of CIVA, with similar SD, while in the case of the extraction of source 2, the CIVA outperforms the proposed method. Nevertheless, it is important to notice that, while



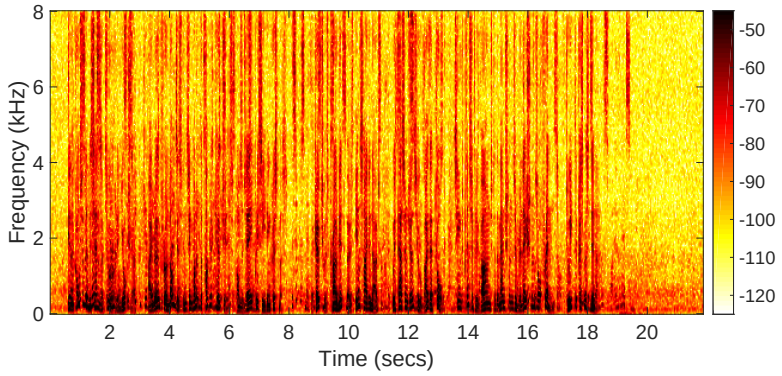

(a) Mixture




(b) Noiseless source




(c) CIVA

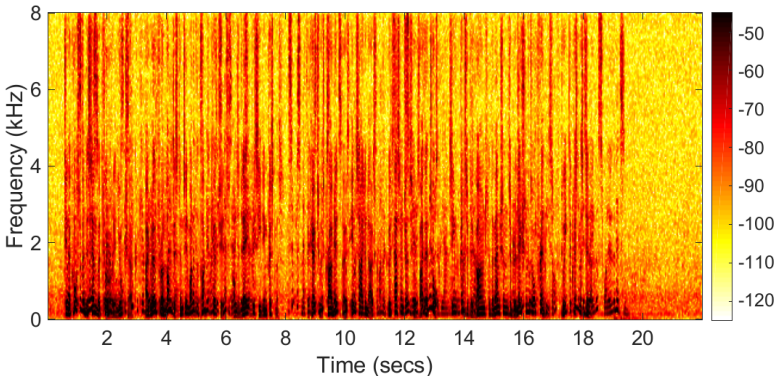



(d) IIVA

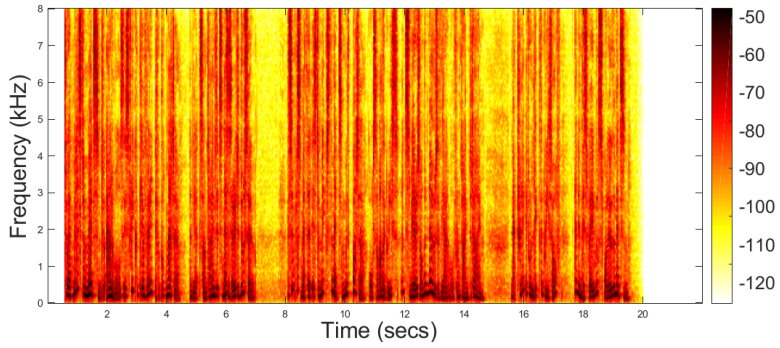
**Figure 5.6:** Spectrogram of source 1 extracted using IIVA (d) and using CIVA (d) for configuration 2. We can see that in the case of CIVA in (c) there is an unwanted attenuation around 1.8 KHz which was not present in the noiseless signal in (b).

the CIVA allows us to reach a better performance for the latter case, analyzing the spectrogram of the outputs when the desired source is source 1 in configuration 2 in 5.6(c) it is possible to see that in the case of the output of the CIVA algorithm there is a region of the spectrum, near 1.9 kHz, in which there is a clear attenuation visible as a straight yellow line which is present through most of the time (indicated by the x-axis on the spectrograms) in which the algorithm was working. This behavior is probably due to the influence of the introduction of the steering vector in (3.32) which is substituted to the first filter in the demixing matrix $\mathbf{G}$ so that the update rule of that specific filter is penalized by the penalty term in (3.34).

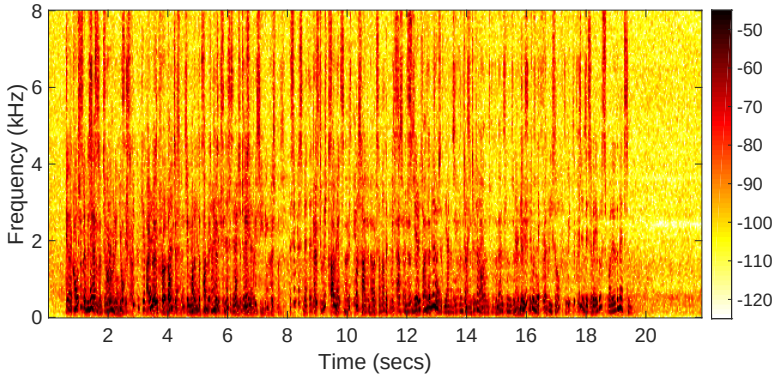
If we look at the spectrogram of the same configuration when the desired source is source 2, we can notice that at second 7, in which the source to be estimated should be silent as we can see from the spectrogram of the noiseless desired source to be estimated in Fig. 5.7(b), in the case of CIVA Fig. 5.7(c) there is much more frequency content with respect to that present in the case of the spectrogram of the output of IIVA in Fig. 5.7(d): frequency content coming from source 1 leaked through and this is probably due do the fact that the initialization of the desired source is done by a steering vector which is pointing to the direction of the desired source which in this configuration is similar to the undesired one.



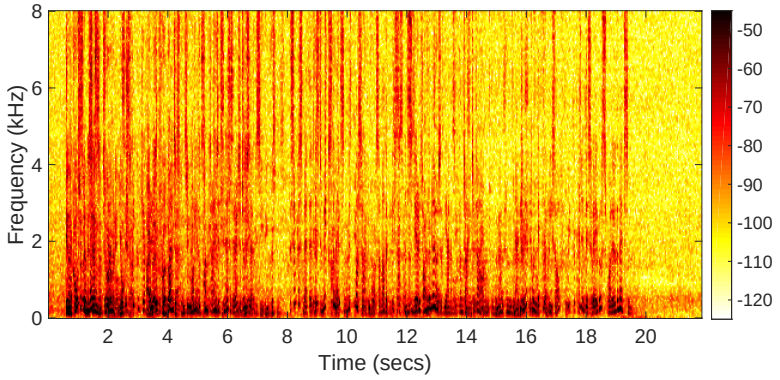

(a) Mixture




(b) Noiseless source




(c) CIVA




(d) IIVA

**Figure 5.7:** Spectrogram of source 2 extracted using IIVA (d) and using CIVA (d) for configuration 2. We can see that in the case of CIVA in (c) there is a considerable leakage of interference in correspondence of second 8 in which the source should not have much frequency content as we can see in (b).

# 6

# CONCLUSIONS AND FUTURE WORK

In this thesis we investigated the Supervised Independent Vector Analysis and proposed a new approach, that we called Informed Independent Vector Analysis, to exploit the algorithm for blind source extraction by the use of a CNN-based DOA estimator [17] which is used to reveal the activity of a desired source in a certain Region-of-Interest. The term informed comes from the fact that our algorithm uses some prior information: we assume to approximately know the Direction of Arrival, given as an angular Region-of-Interest, of the desired source that we aim to extract.

The algorithm, similarly to the well-known Independent Component Analysis [20] algorithms, relies only on statistical hypothesis about the sources to be extracted from the mixture signal and the main assumption, which is often verified in real-world, is that of the independence of the sources to be extracted [10]. The local permutation problems described in section 2.2.1 were solved thanks to the fact that the source pdf is modeled as a dependent multivariate super-Gaussian distribution which allows us to arrive to an expression of the gradient update rule which captures inter-frequency dependencies in the analyzed data [30].

The global permutation problem has been solved thanks to the pilot components, which make use of statistical information about the source signals driving the algorithm in a limited space solution, so that the first output of the algorithm is the desired signal. The statical information about the signals have been provided by revealing the activity of the desired source and adding a contribution coming from the observation set: when a desired source is revealed as active, the other sources are considered as inactive, and this condition usually hold for a time-frequency unit [36], so that the observations are

likely to contain signals coming from the desired source, enforcing the algorithm to converge to a proper solution.

At first, we developed an oracle version to verify that the additive statistical information provided by the pilot components improved the performances over the standard IVA. The first oracle version, described in section 4.1.1, made use of the source signal which would be received at the microphone if no interference was present, revealing the activity of the source by evaluating at each time-frame the power of the signal and adding the clean source signal components to the normalization term of the standard IVA so that the algorithm is able to converge in a restricted solution space. Once we assessed that the additive information effectively improved the algorithm convergence and separation properties, we developed another oracle version, which we named noisy-oracle version in section 4.1.2, that still uses the power of the desired source to reveal the activity of the source but the component added to the normalization term comes from the noisy mixture signal captured by the microphone array.

After the validation of the oracle-algorithms we developed the CNN-based version, which makes use of a multichannel CNN-based localizer [17] in order to track the activity of a desired source given the angular region in which the source is located as explained in 4.2.1. For computing the posterior probabilities, which are used to decide the activation of the pilot components, we investigated two models, proposed in section 4.2.2.1: a hard activation and a soft activation model; furthermore, in section 4.2.2.2, we proposed two models for the additive information provided by the pilots: the single-microphone model, which provides information coming only from the reference microphone, and the all-microphones model, which provided information using the whole observations set. To understand the difference in terms of performance given by the different models, we conducted some experiments in section 5.3.1 and section 5.3.1 obtaining similar performances: based on the results of the experiments, we decided to opt for the use of the hard (or binary) activation model with the all-microphone model for the additive information. We investigated in 5.3 the CNN-based version, that we call IIVA, with various experiments regarding all the parameters which can be tuned and the effect that the variation of these parameters provides to the performances of the algorithm.

We discovered that the smoothing parameter $\beta_{\text{SIVA}}$ helps to improve the performance of the IIVA as shown in section 5.3.4, with a high boost in terms of SIR improvement in the cases in which there is a relevant difference between the power of the interferences and the power of the desired source to be extracted (due to different distance with respect to the ULA). Furthermore, in section 5.3.5 we verified

that the choice of a proper learning rate is crucial to obtain good performances while keeping low the required time for the optimization: we assessed that the proposed algorithm is suitable for real-time implementations.

In section 5.4, we compared the performances of the geometrically Constrained IVA (CIVA) [27], which is a known Blind Source Extraction IVA algorithm in the literature, and the IIVA: the performances of both algorithms are similar and in most cases the IIVA performs slightly better than the CIVA. However, from a perceptual point of view, the IIVA also tends to capture with more fidelity all the frequency components related to the desired source that we aim to extract while the CIVA tends to achieve good performances, and sometimes better than those of IIVA, but with some unwanted attenuations at certain frequencies and with more leakages of signals components from other sources coming from similar DOAs as clearly visible from their time-frequency representation in section 5.4. Furthermore, the IIVA seems to be robust to mismatches of the powers of the desired sources: we showed in our experiments that when one of the sources is far, so that its signal is attenuated with respect to the interfering source in the mixture captured by the microphones, the algorithm is still capable to achieve good performances and an overall good separation.

The advantage of the IIVA with respect to the CIVA is that of not modifying directly the demixing matrix: in the IIVA the performances are increased by providing some additional information coming from the pilots, which is added in the update rule of the standard IVA gradient update, and this is a wanted property since in the CIVA, the penalization term introduced for the utilization of the steering vector, penalizes the algorithm also bringing some unwanted losses in the frequency content of the estimated sources.

The proposed algorithm has shown promising results and this let us envision future developments. It would be interesting to extend the concept of the pilot and to build, for each source, different pilots which could gather and add information of different type such as visual information, e. g. using a face recognition system which detects the movement of the lips to decide whether a source is active or not, or to investigate the behavior of the algorithm in other kinds of applications, for example it would be possible to model different kind of noises, e. g. engine noise, wind noise, etc., as interferences to be suppressed. Furthermore, during the experiments it has been assessed that the normalization used to avoid the divergence of the natural gradient update influences the performances of the algorithm itself: it would be interesting to understand how different normaliza-

tions influence the behavior and the convergence properties of the algorithm.

# BIBLIOGRAPHY


[1] J Allen. "Applications of the short time Fourier transform to speech processing and spectral analysis." In: *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'82.* Vol. 7. IEEE. 1982, pp. 1012–1015 (cit. on p. 19).

[2] Jonathan Allen. "Short term spectral analysis, synthesis, and modification by discrete Fourier transform." In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 25.3 (1977), pp. 235–238 (cit. on p. 19).

[3] Jont B Allen and Lawrence R Rabiner. "A unified approach to short-time Fourier analysis and synthesis." In: *Proceedings of the IEEE* 65.11 (1977), pp. 1558–1564 (cit. on p. 19).

[4] Shun-Ichi Amari, Tian-Ping Chen, and Andrzej Cichocki. "Non-holonomic orthogonal learning algorithms for blind source separation." In: *Neural computation* 12.6 (2000), pp. 1463–1484 (cit. on p. 32).

[5] Shun-ichi Amari and Andrzej Cichocki. "Adaptive blind signal processing-neural network approaches." In: *Proceedings of the IEEE* 86.10 (1998), pp. 2026–2048 (cit. on p. 11).

[6] Shun-ichi Amari, Andrzej Cichocki, and Howard Hua Yang. "A new learning algorithm for blind signal separation." In: *Advances in neural information processing systems*. 1996, pp. 757–763 (cit. on p. 27).

[7] Anthony J Bell and Terrence J Sejnowski. "An information-maximization approach to blind separation and blind deconvolution." In: *Neural computation* 7.6 (1995), pp. 1129–1159 (cit. on p. 18).

[8] Anthony J Bell and Terrence J Sejnowski. "The "independent components" of natural scenes are edge filters." In: *Vision research* 37.23 (1997), pp. 3327–3338 (cit. on p. 16).

[9] Adel Belouchrani et al. "A blind source separation technique using second-order statistics." In: *IEEE Transactions on signal processing* 45.2 (1997), pp. 434–444 (cit. on pp. 12, 13).

[10] J. Benesty, M.M. Sondh, and Y.A. Huang. *Springer Handbook of Speech Processing*. Springer-Verlag New York, Inc, 1st ed. 2007. ISBN: 3540491252 (cit. on pp. 19, 65).

[11] David T Blackstock. *Fundamentals of physical acoustics*. John Wiley & Sons, 2000 (cit. on p. 39).

[12] Michael Brandstein and Darren Ward. *Microphone arrays: signal processing techniques and applications*. Springer Science & Business Media, 2013 (cit. on p. 48).

[13] DH Brandwood. "A complex gradient operator and its application in adaptive array theory." In: *IEE Proceedings F-Communications, Radar and Signal Processing*. Vol. 130. 1. IET. 1983, pp. 11–16 (cit. on p. 36).

[14] Herbert Buchner, Robert Aichner, and Walter Kellermann. "A generalization of blind source separation algorithms for convolutive mixtures based on second-order statistics." In: *IEEE transactions on speech and audio processing* 13.1 (2005), pp. 120–134 (cit. on pp. 12, 13).

[15] J-F Cardoso. "Source separation using higher order moments." In: *Acoustics, Speech, and Signal Processing, 1989. ICASSP-89., 1989 International Conference on*. IEEE. 1989, pp. 2109–2112 (cit. on pp. 12, 13).

[16] Jean-François Cardoso and Antoine Souloumiac. "Blind beamforming for non-Gaussian signals." In: *IEE proceedings F (radar and signal processing)*. Vol. 140. 6. IET. 1993, pp. 362–370 (cit. on p. 16).

[17] Soumitro Chakrabarty and Emanuël AP Habets. "Multi-Speaker Localization Using Convolutional Neural Network Trained with Noise." In: *arXiv preprint arXiv:1712.04276* (2017) (cit. on pp. 3, 39, 41, 42, 51, 65, 66).

[18] E Colin Cherry. "Some experiments on the recognition of speech, with one and with two ears." In: *The Journal of the acoustical society of America* 25.5 (1953), pp. 975–979 (cit. on p. 8).

[19] Seungjin Choi, Andrzej Cichocki, and Shunichi Amari. "Equivariant nonstationary source separation." In: *Neural Networks* 15.1 (2002), pp. 121–130 (cit. on p. 11).

[20] Pierre Comon. *Independent component analysis*. 1992 (cit. on pp. 2, 7, 65).

[21] Arnaud Delorme, Terrence Sejnowski, and Scott Makeig. "Enhanced detection of artifacts in EEG data using higher-order statistics and independent component analysis." In: *Neuroimage* 34.4 (2007), pp. 1443–1449 (cit. on pp. 12, 13).

[22] Emanuel AP Habets. "Room impulse response generator." In: *Technische Universiteit Eindhoven, Tech. Rep* 2.2.4 (2006), p. 1 (cit. on p. 48).

[23] Yiteng Huang, Jacob Benesty, and Jingdong Chen. *Acoustic MIMO signal processing*. Springer Science & Business Media, 2006 (cit. on p. 14).

[24] Aapo Hyvarinen. "Fast ICA for noisy data using Gaussian moments." In: *Circuits and Systems, 1999. ISCAS'99. Proceedings of the 1999 IEEE International Symposium on*. Vol. 5. IEEE. 1999, pp. 57–61 (cit. on p. 16).

[25] Aapo Hyvärinen and Erkki Oja. "Independent component analysis: algorithms and applications." In: *Neural networks* 13.4-5 (2000), pp. 411–430 (cit. on p. 15).

[26] Ian T Jolliffe. "Principal component analysis and factor analysis." In: *Principal component analysis*. Springer, 1986, pp. 115–128 (cit. on pp. 2, 7).

[27] A.H. Khan, M. Taseska, and E.A.P. Habets. "A Geometrically Constrained Independent Vector Analysis Algorithm for Online Source Extraction." In: *Vincent E., Yeredor A., Koldovský Z., Tichavský P. (eds) Latent Variable Analysis and Signal Separation. LVA/ICA 2015. Lecture Notes in Computer Science* 9237 (2015), pp. 396–403 (cit. on pp. 3, 5, 6, 23, 33, 35, 45, 47, 61, 67).

[28] T. Kim. "Real-time independent vector analysis for convolutive blind source separation." In: *IEEE Transactions on Circuits and Systems I: Regular Papers* 57.15 (July 2010), pp. 1431–1438 (cit. on pp. 31, 48, 61).

[29] T. Kim, T. Eltoft, and T.W. Lee. "Independent vector analysis: An extension of ICA to multivariate components." In: *IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING* 3889 (Mar.2006), pp. 165–172 (cit. on pp. 23, 30).

[30] T. Kim et al. "Blind Source Separation Exploiting Higher-Order Frequency Dependencies." In: *International Conference on Independent Component Analysis and Signal Separation* 15.1 (Jan.2007), pp. 70–79 (cit. on pp. 2, 3, 8, 23–25, 29, 65).

[31] Daniel D Lee and H Sebastian Seung. "Learning the parts of objects by non-negative matrix factorization." In: *Nature* 401.6755 (1999), p. 788 (cit. on pp. 2, 8).

[32] Intae Lee, Taesu Kim, and Te-Won Lee. "Fast fixed-point independent vector analysis algorithms for convolutive blind source separation." In: *Signal Processing* 87.8 (2007), pp. 1859–1871 (cit. on p. 32).

[33] Kiyotoshi Matsuoka. "Minimal distortion principle for blind source separation." In: 4 (2002), pp. 2138–2143 (cit. on pp. 17, 20, 30, 33).

[34] F. Nesta and Z. Koldovskj. "Supervised Independent Vector Analysis through Pilot Dependent Components." In: *ICASSP2017* (March 2017), pp. 536–540 (cit. on pp. 3, 5, 23, 33, 36, 37, 39, 50, 56).

[35] Nobutaka Ono and Shigeki Miyabe. "Auxiliary-function-based independent component analysis for super-Gaussian sources." In: *International Conference on Latent Variable Analysis and Signal Separation*. Springer. 2010, pp. 165–172 (cit. on p. 34).

[36] Scott Rickard and Ozgiir Yilmaz. "On the approximate W-disjoint orthogonality of speech." In: *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*. Vol. 1. IEEE. 2002, pp. I–529 (cit. on pp. 42, 65).

[37] Richard Roy and Thomas Kailath. "ESPRIT-estimation of signal parameters via rotational invariance techniques." In: *IEEE Transactions on acoustics, speech, and signal processing* 37.7 (1989), pp. 984–995 (cit. on p. 2).

[38] Paris Smaragdis. "Blind separation of convolved mixtures in the frequency domain." In: *Neurocomputing* 22.1-3 (1998), pp. 21–34 (cit. on p. 3).

[39] Petre Stoica and Arye Nehorai. "MUSIC, maximum likelihood, and Cramer-Rao bound." In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 37.5 (1989), pp. 720–741 (cit. on p. 2).

[40] T. Taniguchi et al. "An auxiliary-function approach to online independent vector analysis for real-time blind source separation." In: (2014), pp. 107–111 (cit. on pp. 3, 5, 23, 33, 34).

[41] Felix Weninger et al. "Discriminatively trained recurrent neural networks for single-channel speech separation." In: (2014), pp. 577–581 (cit. on p. 37).

[42] Peter H Westfall. "Kurtosis as peakedness, 1905–2014. RIP." In: *The American Statistician* 68.3 (2014), pp. 191–195 (cit. on p. 14).

[43] Donald J Wheeler. "Problems with Skewness and Kurtosis, Part Two." In: *American Statistical Association and the American Society for Quality* (2011) (cit. on p. 14).