REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

BLIND AUDIO SOURCE SEPARATION USING INDEPENDENT COMPONENT ANALYSIS AND INDEPENDENT VECTOR ANALYSIS METHODS

ALYAA MAHDI

MSc. THESIS
DEPARTMENT OF COMPUTER ENGINEERING
PROGRAM OF COMUTER ENGINEERING

ADVISER
PROF. DR. NIZAMETTIN AYDIN

İSTANBUL, 2017

REPUBLIC OF TURKEY

YILDIZ TECHNICAL UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

BLIND AUDIO SOURCE SEPARATION USING INDEPENDENT COMPONENT ANALYSIS AND INDEPENDENT VECTOR ANALYSIS METHODS

A thesis submitted by Alyaa MAHDI in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE** is approved by the committee on 25.5.2017 in Department of computer Engineering

Thesis Adviser

Prof.Dr. Nizamettin AYDIN

Yıldız Technical University

Approved By the Examining Committee

Prof. Dr.Nizamettin AYDIN

Yıldız Technical University

Assistant Prof. Dr.Gökhan Bilgin , Member

Yıldız Technical University

Prof. Dr. Atakan KURT , Member

İstanbul University

ACKNOWLEDGEMENTS

I would first like to express my deep gratitude to my adviser Prof. Dr. Nizamettin AYDIN of the Department of computer engineering at Yıldız Technical University for his assistant to finish this work.

Additionally, I must express my very profound gratitude to my father for his moral support and encouragement, to my mother Soul which I feel she encouraged me in each weakness moments, to my husband who without his efforts I would not be able to achieve my dream, for his sacrifice, patience, supporting and helping.

I would also like to thank my friends, my classmates who never hesitate to help me whenever I asked.

I would like to appreciate Iraq government to let me have this opportunity to study abroad and have my higher education

Finally, I would like to thank with deep appreciation my teacher Assoc. Prof. Dr. Fethullah KARABİBER of the Department of computer engineering at Yıldız Technical University for his help whenever I had question about my research or writing.

MAY, 2017

Alyaa MAHDI

TABLE OF CONTENTS

LIST OF SYMBOLS

s(t)	Source signals
x(t)	Observed signals
ŝ,y	Estimated signals
A	Mixing matrix
W	Un- mixing matrix
p$\mathcal{F}$	($\mathcal{F}$-correlation) construct function of Kernel-ICA
p	The joint probability density function
Πq	The product of probability density function
J	The construct function for fast-IVA algorithm
μ_j , $\sum_i$	The Gaussian Mixture Model parameters

LIST OF ABBREVIATIONS

BSS	Blind Source Separation
EM	Expectation Maximization
EER	Equal Error Rate
Fast-ICA	Fast –fixed-point - ICA
Fast-IVA	Fast –fixed-point - IVA
GMM	Gaussian Mixture Model
ICA	Independent Component Analysis
IVA	Independent Vector Analysis
I-vector	Intermediate Vector
Kernel-ICA	Kernel Independent Component Analysis
KCCA	Kernel Canonical Correlation Analysis
MFCC	Mel Frequency Cepstral Coefficients
MAP	Maximum A Posteriori
PCA	Principal Component Analysis
SAR	Source-to-Artifact Ratio
SDR	Source-to- Distortion Ratio
SNR	Source-to- Noise Ratio
SIR	Source-to- Interference Ratio
UBM	Universal Background Model

LIST OF FIGURES

Page

LIST OF TABLES

Page

ABSTRACT

BLIND AUDIO SOURCE SEPARATION USING INDEPENDENT COMPONENT ANALYSIS AND INDEPENDENT VECTOR ANALYSIS METHODS

Alyaa MAHDI

Department of Computer Engineering

MSc. Thesis

Adviser: Prof. Dr. Nizamettin AYDIN

Blind Source Separation (BSS) is one of the most challenging problems in the field of audio and speech processing. Many different methods have been proposed to solve BSS problem in the literature. In addition, speaker recognition systems have gained considerable interest from researchers for decades due to the breadth of their field of application.

In this study, we have compared the performance of three popular BSS methods implementations: Fast-ICA, Kernel-ICA and Fast-IVA which are based on Independent Component analysis (ICA) and Independent Vector Analysis (IVA) respectively. Initially, classical performance comparison metrics such as Source-to-Artifact Ratio, Source-to- Distortion Ratio, Source-to-Noise Ratio, are implemented for comparison.

For further investigation, speaker recognition system has been developed to examine the effect of speech separation on the performance of these recognition systems.

In our experiments, we used two data set the first one is in Arabic languge and contains voice records frome 13 speaker: 3 female , 10 male.the second data set is the ELSDSR data which in English languge and contains voice records from 22 speakers: 10 female, 12 male.

The performance of BSS methods is measured under four scenarios. The first three is composed to see the effect of noise. Therefore, we used the mixture of clean source signals, the mixture of source signals with additive Gaussian noise, adding Gaussian noise to clean source mixture. In the fourth scenario, we applied speaker recognition system based on Gaussian mixture models (GMMs) and I-vectors, the performance of the speaker recognition system is measured by Equal Error Ratio (EER), which is, the most reliable measurement in this field.

Experimental results show that the Fast-IVA has better performance than the Fast-ICA method according to performance metrics used in this study. In terms of EER, I-vector gives the better result than GMM for separated signals by IVA and ICA.

Key words: Blind source separation, Independent component analysis, Independent vector analysis, kernel Independent component analysis.

ÖZET

BAĞIMSIZ BİLEŞEN ANALİZİ VE BAĞIMSIZ VEKTÖR ANALİZİ KULLARAK SES SİNYALLERİNDE KÖR KAYNAK AYRIŞTIRIMI

Alyaa MAHDI

Bilgisayar Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Tez Danışmanı: Prof. Dr. Nizamettin AYDIN

Kör kaynak ayrıştırma (KKA) ses ve konuşma işleme alanındaki önemli problemlerden birisidir. Bu problemi çözmek için literatürde farklı yöntemler önerilmiştir. Buna ek olarak konuşmacı tanıma da araştırmacılar için onyıllarca üzerinde çalışmaların yapıldığı bir çalışma alanıdır. Bu çalışmada kör kaynak ayrıştırma problemini çözmek için kullanılan Bağımsız bileşen analizi (BBA) ve bağımsız vektör analizi (BVA) yöntemlerinin uygulamaları olan Hızlı – BBA, Çekirdek - BBA ve Hızlı – BVA algoritmaları üzerinde karşılaştırmalar yapılmıştır. İlk olarak Source-to-Artifact oranı, Source-to- Distortion oranı, Source-to-Noise oranı karşılaştırma yapmak için uygulanmıştır. Daha sonraki araştırmalar için konuşmacı tanıma sistemi kaynak ayrıştırma işlemlerinin konuşmacı tanıma etkisini incelemek maksadıyla oluşturulmuştur. Çalışmada 10 kadından ve 12 erkekten oluşan 22 konuşmacıdan elde edilen ELSDSR veri seti kullanılmıştır. KKA sisteminin performansı dört farklı senaryo ile test edilmiştir.

Bunlardan ilk üç tanesi gürültünün sisteme etkisini görmek için yapılmıştır. Dolayısıyla bunu yapmak için gürültü içermeyen karıştırılmış sinyaller, karıştırılmış sinyallere Gauss gürültüsü eklenmiş yeni sinyal ve gürültü içermeyen sinyallere Gauss gürültüsü eklenmiş sinyaller kullanılmıştır. Çalışmadaki dördüncü senaryo ise konuşmacı tanıma sisteminin Gauss Karışım modeli ve Ivector yöntemlerinin konuşmacı tanıma sistemi performanslarının sıklıkla kullanılan Eşit Hata Oranı (EHO) ile karşılaştırılmasını içermektedir. Deneysel sonuçlar; EHO ölçütüne göre Hızlı-BVA algoritmasının Hızlı-BBA algoritmasının daha başarılı olduğunu göstermektedir. Ayrıca EHO ölçütü

açısından BVA ve BBA tarafından ayrılan sinyallerin I-Vektör yönteminin Gauss Karışım Modelinden daha başarılı sonuçlar vermiştir.

Anahtar Kelimeler: Kör Kaynak Ayrıştırma, Bağımsız Bileşen Analizi, bağımsız vektör analizi , Çekirdek Bağımsız Bileşen Analizi.

CHAPTER 1

INTRODUCTION

1.1 Literature Review

The Blind Source Separation (BSS) Problem has expansive attention for many decades. The simple description of this problem when two or more people were in a room and many conversations might happen simultaneously, so how can we accurately determine what a particular people talks among several people that are talking at the same time?". Humans can overcome this problem according to the remarkable abilities of their brain for sorting the mixture of auditory sources, but it is considered as complicated problem for digital signal processing. This problem is also known as "cocktail party problem" that was first proposed by Colin Cherry in 1953, many efforts have been dedicated to this problem in variety of science fields: physiology, neurobiology, psychophysiology, cognitive psychology, biophysics, computer science, and engineering [1].

Several techniques were proposed to solve BSS problem specifically the Blind Audio Source separation problem was firstly addressed by Herault and Jutten in 1985 [2]. In their work, the sound is directly transmitted to the microphones without any delay which is known as standard blind source separation. Then in 1995, Bell and Sejnowski developed the Independent Component Analysis (ICA) method to separate the sources when they are mixed simultaneously [3]. Also, some representational algorithms of ICA were proposed. In 1997 Aapo Hyvarien and Erkki Oja, proposed the Fast-Fixed Point ICA algorithm and could successfully prove that new algorithm is 10 to 100 times faster than gradient algorithm [4], the Joint Approximate Diagonalization of Eigen matrices (JADE-ICA) technique has been proposed by Jean-Fran¸cois Cardoso in 1999[5], the Extended Generalized Lambda Distribution EGLD-ICA approach was addressed by J.Eriksson, J.Karvanen, V.Koivunen, in 2000 [6]. Furthermore, the MS-ICA approach is

proposed to offer the ability to separate the nonlinear mixture sources in 1994[7], and the Kernel-ICA technique which was proposed by R. Bach, Michael I. Jordan in 2002[8].

According to the fact of sound wave reflection from the ground, the ceiling and all the furniture inside the room in real life, the sound waves take multiple paths before reaching the microphone. As a result, this problem becomes more complicated for real room environment and this speech propagation problem is called convolutive blind source separation (CBSS) [9].

Initially, the solutions were posed in the time domain. Due to the complicated calculation caused by convolution, Parra et al [10] suggested another method based on the frequency domain. In the frequency domain, the convolution is replaced with multiplication to have low cost in terms of execution time. However, this method still has scaling and permutation ambiguities. One of the proposed methods to solve the scaling ambiguities was matrix normalization [11, 12]. The permutation problem was more challenging to solve, in [13, 14] the authors solve the permutation ambiguities successfully by using the inverse of de-correlating metrics and the envelope of the sound signal depending on the theory that the speech signal is stationary in short period of time. Furthermore, more efforts to overcome these ambiguities have been submitted as shown in [15]. To overcome difficulties that ICA has been faced to separate Multivariate sources, an advanced method named independent vector analysis (IVA) was proposed by Kim et al [16]. The IVA method was developed later by I. Lee, T. Kim, T.-W. Lee to produce Fast – fixed point Independent vector analysis Fast-IVA algorithm [17, 18].

On the other hand, researchers show high interest in speaker recognition systems for more than five decades ago duo to the widespread of automatic speech recognition system application such as automatic call processing in telephone networks and query-based information systems that provide updated travel information, stock price quotations, weather reports, etc. Working in speaker recognition field has begun in 1960's when Bell Labs submitted the experiment that worked over dialed-up telephone lines [19]. The development of speaker recognition system that based on Hidden Markov Model (HMM) began in 1980's. In 1990's the score normalization and text induced methods were evolved. In 2000's the text independent speaker recognition systems were successfully developed [20].

Additionally, Blind Source Separation techniques have been used to improve the performance of speaker recognition system as shown in [21, 22] where the number of sources equal to the number of microphones. Furthermore, the Over-determined Blind Speech Separation (OBSS) case also was studied in [23], where the number of microphones was more than the number of sources. The extra number of microphones have a positive effect on the performance of the process and speaker recognition system at the same time. In the [24] novel solution to the (OBSS) problem was proposed to enhance the speaker recognition performance. In [25], Martin showed the effect of BSS methods of the Speaker Recognition System ARS performance by comparing the performance of the SRS before and after separation mixed speech signals for the teleconferences. He used the Diarization Error Rate (DER) for measure the performance of speaker recognition system. The DER value was 0% for SRS before separating and 66% after separating. He used IVA in his work for sources separation.

1.2 Objective of the Thesis

The objective of this thesis is to implement Blind Source Separation methods on the mixture of speech signals and apply speaker recognition system to the separating signals in order to find the better method that has high performance than the others.

In this thesis, three different proposed scenarios are followed by mixing and separating the speech signals. For each scenario, the speech signals are mixed in pairs. Fast-ICA, Kernel-ICA, and Fast-IVA algorithms are used for separating signals. In the fourth scenario, speaker recognition system including speaker identification and speaker verification is applied for all the separated signals that obtained from each three previous scenarios.

1.3 Hypothesis

Blind Source Separation (BSS) is one of the most challenging problems that has been attracted the attention of researchers in different fields of since. This problem describes the situation of focusing on one speaker in case of several persons talking simultaneously in the same room. To separate the mixed speech signals and obtain just a speech signal which belongs to a particular speaker is very challenging and complicated problem [1]. The challenge of this problem is the estimation of the sources without any prior knowledge about the original sources or mixing matrix. Three statistical methods were utilized in this work to separate the mixture speech signals on two databases such as Fast-ICA, Kernel-ICA, and Fast-IVA.

Due to the significant importance of the Speaker Recognition systems that have widespread applications in various fields, the effect of BSS methods of the speaker recognition system have been examined in this work. Figure 1.1 shows the main structure of this thesis work. The Speaker recognition System can be divided into two main tasks: speaker identification and speaker verification, this work deals with both of these tasks.

For speaker identification, the Gaussian Mixture Model (GMM) is used as a classification method which is, based on Mel-Frequency Cepstrum Coefficients (MFCC) for feature extraction. The i-vectors method is used for speaker verification.

Speaker Recognition is applied for all separated speech signals that have been separated by BSS methods which are used in this work.

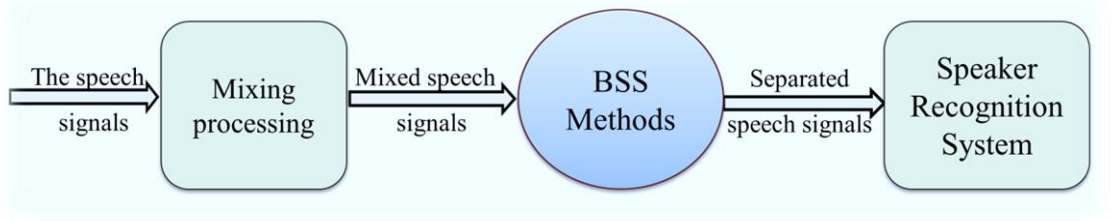


Figure 1.1 the main structure

1.4 Thesis Organization

This thesis was organized as follow: Chapter 1 includes the general introduction of this thesis wok. Chapter 2 gives the information with details about the Blind source Separation BSS methods and the Speaker Recognition System. Moreover, this chapter includes the explanation of the implementation of these methods. The proposed scenarios, the results of our experiments, the discussion of these results and the future work were presented in Chapter 3. Finally, the conclusion was included in Chapter 4.

CHAPTER 2

METHODS AND METARIALS

.

2.1 Blind Sources Separation

The blind source separation (BSS) is the process of separation mixed sources. When two or more people were talking in a room that has microphones placed in a different location, the observed records by one or more of these microphones would detect several conversations at once. The term “blind” refers to the ability to estimate the original sources from the observation signals by microphones array without knowing the characteristic of the transmission channel or how these sources have been mixed. The number of sources and microphones determine the BSS problem model. Thus, when the number of microphone is more than number of sources (N < M) this is known as Over-determined BSS [23, 24]. The Underdetermined BSS expression refers to the situation when the number of microphones is less than the number of sources (N>M)[26], and classical BSS when the number of sources and microphone are equal (N=M) as shown in figure 2.1.

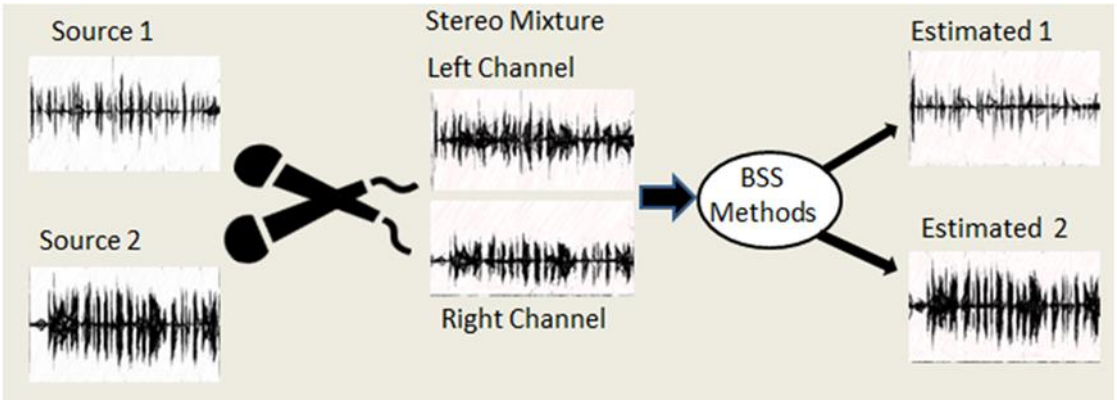


Figure 2.1 classical BSS

Additionally, each of BSS model can occur with two different mixture model.

Instantaneous Mature model: in this model, the sources signals reach the microphones at the same time without any delay in time.

Convolutive mixture model: This model refers to the mixing process that happened in a real room so, due to the reflection that caused by the room walls and the furniture of the room the source signal would not arrive at the microphone at the same time. Hence, this model is more complicated than the Instantaneous Mature model .

2.1.1 Independent Component Analysis

Independent Component Analysis (ICA) is one of the most popular BSS methods. ICA was used extensively for many applications in the various fields of science and engineering. ICA, which is a statistical computational method, is employed to find underlying hidden factors among a set of random vectors. The main aim of ICA method is to obtain the independent components (ICs), which are linearly independent or as independent as possible.

For deep analysis, let us explain the ICA Model in the time domain.

Mathmatically, we can expresses N number of differrent source signals with sources index i=1,2,…..N as a vector in this way :

$$\mathbf{s}(t) = (s_1(t), s_2(t), \dots\dots\dots s_N(t))^T \tag{2.1}$$

Also we can define the observed signals observed signal $\mathbf{x}(t)$ and the noise signal ***n(t)*** with microphone index j=1,2,……M as a vector as follow :

$$\mathbf{x}(t) = (x_1(t), x_2(t), \dots\dots\dots x_M(t))^T \tag{2.2}$$

$$\mathbf{n}(t) = (n_1(t), n_2(t), \dots\dots\dots n_M(t))^T \tag{2.3}$$

Where M refers to the number of microphones and *t* is the time index.

Thus, for the Instantaneous Mixture Model we can define each observed signal as:

$$x_j(t) = \sum_{i=1}^{N} a_{ji}.s_i(t) + n_j(t) \tag{2.4}$$

Where, a_{ji} represents the weighted vector parameter that depends on the distance between the source and the microphone, i is the sources index and j is the observed signals index.

By using vector-matrix notation instead of summations like in equation (2.4), we obtain the following expression:

$$\mathbf{x}=\mathbf{As}+\mathbf{n} \tag{2.5}$$

For noise-free mixture model we can rewrite the equation (2.5) as following way:

$$\mathbf{x}=\mathbf{As} \tag{2.6}$$

Where **A** is the mixing matrix, **x** is a vector of the observed signals and **s** is a vector of the source signals.

For the classical BSS model when the number of sources and microphones are equal, The ICA method achieves its aim by finding the un-mixing matrix **W** which is, the inverse of mixing matrix **A.** This can be written as:

$$\mathbf{W}=\mathbf{A}^{-1} \tag{2.7}$$

As a result, ICs denoted by **y** is obtained simply by:

$$\mathbf{y}=\mathbf{W}\,\mathbf{x} \tag{2.8}$$

To simplify the complicated calculation in convolution BSS [9], ICA in the frequency domain are used. The convolution in the time domain is the multiplication in the frequency domain. So that, for free- noise model and by applying Fourier transform we can rewrite the Equation (2.4) in the following:

$$X_i(\omega) = \sum_k A_{ik}(\omega) s_{ik}(\omega) \tag{2.9}$$

We can rewrite the previous equation in vector-matrix notation as follow:

$$\boldsymbol{X}(\omega) = \boldsymbol{A}(\omega) . \boldsymbol{S}(\omega) \tag{2.10}$$

Thus, the estimation of the sources can be occurred by finding the un-mixing matrix W(ω) for each frequency $\omega = 2\pi f$. Where the un-mixing matrix W(ω) is equal to the inverse of $A(\omega)$:

$$\mathbf{W}(\omega) = \boldsymbol{A}(\omega)^{-1} \tag{2.11}$$

Hence, we can obtain the estimated signals Y(ω) as follow:

$$\begin{aligned} \mathrm{Y}(\omega) &= \mathbf{W}(\omega). \boldsymbol{X}(\omega) \\ &= \boldsymbol{A}(\omega)^{-1} . \boldsymbol{X}(\omega) \end{aligned} \tag{2.12}$$

As we know the speech signal is not- stationary signal and under the assumption that the non-stationary signal transformed to the stationary signal in short blocks. So we need to apply the short-time Fourier transform (STFT), windowing and a discrete Fourier transform (DFT). Thus we can describe each observed signal **x**(*f*, i) in the time-frequency domain for each frequency bin as below:

$$\mathbf{x}^f = [\, x_1^f , x_2^f , \ldots\ldots x_M^f \,]^T \tag{2.13}$$

Here, *i* is the time index refers to the *i*-th block and *f* is the index of the frequency

Thus, we can refer to the estimated source Y in each block for each frequency bin as below:

$$\begin{aligned} \mathrm{Y}^f &= \mathrm{W}^f . \mathrm{x}^f \\ &= (A^f)^{-1} . \mathrm{x}^f \end{aligned} \tag{2.14}$$

ICA assumptions

In order to make ICA model work properly, a few assumptions must be made.

The first assumption: The original sources should be statistically independent [27]. Statistical independence is defined in terms of probability density function (PDF) of the sources signals. Thus, the joint probability density function (PDF) of N different original sources (s_i) can be expressed as:

$$p_N(s_1, s_2, \dots\dots, s_N) = p_1(s_1).p_2(s_2)p_2 \dots\dots\dots . p_N(s_N) \quad (2.15)$$

Similarly, independence could be defined by replacing the pdf by the respective cumulative distributive functions as:

$$E\{p_1(s_1)\, p_2(s_2), \dots\dots, p_N(s_N)\} = E\{g_1(s_1)\}\, E\{g_1(s_1)\} \dots\dots E\{g_N(s_N)\} \quad (2.16)$$

Where E{.} is the expectation operator.

The Second assumption: The sources (s_i) have non-Gaussian distribution.

Theoretically, a Gaussian distribution signal can be considered as a linear combination of many independent signals, thus, a Gaussian signal cannot be distinguished by ICA method as a single source, so that ICA method assumed that the sources signals have non-Gaussian distribution in order to recognize them effectively. Several measurements methods such as kurtosis and entropy methods are used to measure the non-Gaussianity distribution of the sources.

Kurtosis is the statistical method that used for measuring the non-Gaussianity. The basic definition of Kurtosis for signal (*s*) that has zero mean can be expressed by:

$$Kurt(s) = E\{s^4\} - 3(E\{s^4\})^2 \quad (2.17)$$

In another word, kurtosis method measures the skewness of the distribution. Its value gives the description of the distribution tails. So, this method is known with its sensitivity to the outliers and statically kurtosis is not robust for ICA method.

Entropy is the measurement of any disorder system. Theoretically, it can be used to measure the randomness of the signal. The entropy H of the signal (S) can be defined as:

$$H(S) = -\int P(S)\ log\ P(S)\ ds \tag{2.18}$$

According to the information theory, the signal that has Gaussian- distribution it has largest entropy value and vice versa. Thus the entropy can be considered as the non-Gaussian measurement.

For simplicity, the entropy measurement has been normalized to produce a new measurement that called Negentropy. We can define Negentropy measurement J as follow:

$$J(S) = H(S_{Gaussian}) - H(S) \tag{2.19}$$

According to equation (2.13), the Negentropy value would be positive or zero with a pure Gaussian signal.

The third assumption: The unknown mixing matrix **A** is assumed to be invertible or pseudo-invertible that makes it possible to invert or pseudo-invert the missing matrix **W** and estimate the source components using the equation (2.8).

ICA Ambiguities

ICA method has two ambiguities: the magnitude and scaling ambiguity and the permutation ambiguity.

- **Magnitude and scaling ambiguity**

This ambiguity occurred because of the disability to determine the true variance of the sources. For more illustration, we can rewrite the mixing in equation (2.6) as follow:

$$= \sum_{j=1}^{N} a_j \, s_j \tag{2.20}$$

Where, a_j refers to the *j*-th column of the mixing matrix A. Since neither the coefficients of the mixing matrix nor the original sources s_j are unknown, we can rewrite the Equation (2.20) in this way:

$$\mathrm{x} = \sum_{j=1}^{N} (1/\alpha_j \; a_j \,)(\alpha_j \, s_j \,) \tag{2.21}$$

- **Permutation ambiguity**

Since the ICA method separates the mixed of independent sources blindly without any information about the original sources. Thus, the order of the sources is unknown and that led to estimate the sources probably in different order. The Equation bellow expresses that ambiguity in this way:

$$x = AP^{-1} \, Ps$$

$$x = A's' \tag{2.22}$$

The elements of *Ps* are the original sources, but in a different order, and $A' = AP^{-1}$ is the unknown mixing matrix. The expression in Equation (2.22) cannot be distinguished within the ICA framework. This problem is insignificant in time domain since it only causes that the estimated signal will be in different order than the original sources, but imagine this problem in the frequency domain when the separation process is done in each frequency bin.

Preprocessing

Some preprocessing steps can be performed to improve the performance of ICA methods. Useful preprocessing techniques are discussed below.

1. Centering

In this step the observed vector x is centered by subtracting its mean vector m=E{x} to obtain zero mean. The center vector x_c, can be defined as follows:

$$\mathbf{x_c = x - m} \tag{2.23}$$

Thus, the un-mixing matrix will be estimated using the centered data. The ICs are estimated using the following equation:

$$Y = A^{-1}(x_c + m) \tag{2.24}$$

After performing this preprocessing without affecting the estimation of the mixing matrix, all the observing vectors can be considered as centered.

2. Whitening

Whitening of the observation vector *x* is a useful and important step for ICA algorithm. It includes linearly transforming the observation vector *x* such that its components are uncorrelated and have unit variance. The eigenvalue decomposition (EVD) is used to perform the whitening transformation in a simple way [28]. Thus, decomposition of the covariance matrix of *x* is calculated as follows:

$$E\{xx^T\} = VDV^T \tag{2.25}$$

Where $E\{xx^T\}$ is the covariance matrix of x, V denotes the matrix of eigenvectors of $E\{xx^T\}$,and D is the diagonal matrix of eigenvalues, i.e. D = diag$\{\lambda_1, \lambda_2, \ldots, \lambda_n\}$.

The following transformation is employed to whiten the observation vector.

$$x_w = VD^{-1/2}\, V^T x \tag{2.26}$$

Where the matrix $D^{-1/2}$is obtained by a simple component-wise operation as $D^{-1/2}$ = diag $\{\lambda_1{}^{-1/2}, \lambda_2{}^{-1/2}, \ldots\ldots \lambda_n{}^{-1/2}\}$. Whitening transforms the mixing matrix into a new orthogonal matrix:

$$x_w = VD^{-1/2}\, V^T\, A_s = A_w s \tag{2.27}$$

Whitening reduces the numbers of parameters to be estimated. The new orthogonal mixing matrix is needed to be estimated. As a result, whitening can solve a half of ICA problem.

Fast Fixed-point ICA (Fast-ICA)

One of the algorithms that based on ICA method is the fast fixed-point ICA algorithm (Fast-ICA), which transforms the neural network rule into a fixed-point iteration. So it is known for its simplicity and speed when it is compared with the gradient based algorithms. The (Fast-ICA) is used for separating sources and extracting features [4]. Fast-ICA algorithm has two estimation approaches : deflation approach to estimate ICs one by one and symmetric approach to estimate ICs simultaneously.

1. The default estimation approach for Fast -ICA algorithm:

Under the assumption that we obtain the whitening vector $\boldsymbol{x_w}$ of the observed signal after applying the whitening pre-processing as we explained previously and due to Eq.(2.27) and (2.17). Fast fixed-point Algorithm estimates the sources one by one by estimating the un-mixing matrix *w* as shown in the following steps:

1- Initialize with a random vector value w (0) of norm 1 let p=1.
2- Let $w(p) = E\, x(w(p-1)^T x)^3 - 3w(p-1)$, Estimate the expectation using a large sample of x vectors.
3- Divide $w(p)$ by its norm.
4- If $|(w(p)^T w(p-1)|$ is not close enough to 1, consider $p = p + 1$ and go back to step 2 , otherwise return the vector $w(p)$ as output.

Where, the output vector $w(p)$ is equal to one of orthogonal matrix column A_w . Thus in terms of BSS problem that mean $w(p)$ separate one of non-Gaussian sources signal

2. Symmetric approach to estimate ICs simultaneously:

Typically, we obtain several estimated sources by several running of the algorithm but we need to be sure that they are different. So an orthogonalizing projection is added to the step 3 of the algorithm

3- Let $w(p) = w(p) - \bar{A}\bar{A}^T w(p)$.Divide $w(p)$ by its norm.

Where, the columns of $\bar{A}$ matrix are previously found the columns of the orthogonal matrix A_w .

Kernel Independent Component Analysis

The Kernel Independent Component Analysis (Kernel-ICA) is a different version of ICA model that based on the minimization of a contrast function based on kernel ideas [8]. Kernel _ICA model has two different algorithms The Kernel ICA-KCCA algorithm and the Kernel ICA-KGV algorithm.

Let us first define the contrast function that measures the dependence of random variables. In a case of two variable x_1 , x_2 and Ƒ is the vector space of functions from R to R so we can describe the Ƒ- correlation PƑ as follow:

$$P\text{Ƒ} = \max_{f_1, f_2 \in F} corr\,(f_1(x_1), f_2(x_2)) = \max_{f_1, f_2 \in F} \frac{cov(f_1(x_1), f_2(x_2))}{(var f_1(x_1))^{½} (var f_1(x_1))^{½}} \qquad (2.28)$$

PF is the maximal correlation between$f_1\ (x_1\ )and\ f_2(x_2)$,where f_1 and f_2 ranges are over ℱ

In order to implement the *F*-correlation, the reproducing kernel Hilbert space (RKHS) ideas are used. With considering that ℱ is an RKHS on R, and K(x; y) is the associated kernel, and $\phi(x)$ = K(.; x) is the feature map, where K(.; x) is a function in ℱ for each *x*. We then have the well-known reproducing property:

$f(x)$ =$\langle\phi(x), f\rangle\ \forall\ f\ \in$ ℱ, x $\in$ R:

This implies:

$$corr\ (f_1\ (x_1\ ), f_2(x_2))\ =\ corr\ (\langle\phi(x_1), f_1\rangle, \langle\phi(x_2), f_2\rangle) \tag{2.29}$$

The maximal correlation between one dimension projection $\phi(x_1)$ and $\phi(x_2)$ is denoted by the F-correlation. Thus, depending on that definition that similar to the definition of the first canonical correlation between $\phi(x_1)$ and$\phi(x_2)$. Hence, the canonical correlation in function space can be computed based on the ICA contrast function. Canonical correlation analysis (CCA) is a statistical technique which is similar to principal component analysis (PCA) technique, but the (CCA) works with a pair of random vectors and maximizes correlation between sets of projections and leads to generalized eigenvector problem (instead of work on one random vector as in (PCA). Thus, the (CCA) is carried out to compute the contrast function for ICA by using the kernel trick.

2.1.2 Independent Vector Analysis

Due to the problems that ICA method had faced such as the promotion problem in the frequency domain and separating the multivariate sources, the Independent Vector Analysis (IVA) which is one of the most advanced methods that shows better performance in the field of BSS [16] was proposed. It is designed according to an assumption that all the elements of one source vector over all the frequency bins are dependent, but the elements of different sources vectors within one frequency bin are independent. Thus, we can represent each source vector as $s = [s^1, s^2, \ldots\ldots, s^F]$ and each mixture vector as $\mathrm{x} = [\mathrm{x}^1, \mathrm{x}^2, \ldots\ldots, \mathrm{x}^F]$ where *F* is the number of the frequency

bins [17]. Fig. 2.2 shows the instruments mixture model of IVA for two sources and two microphones.

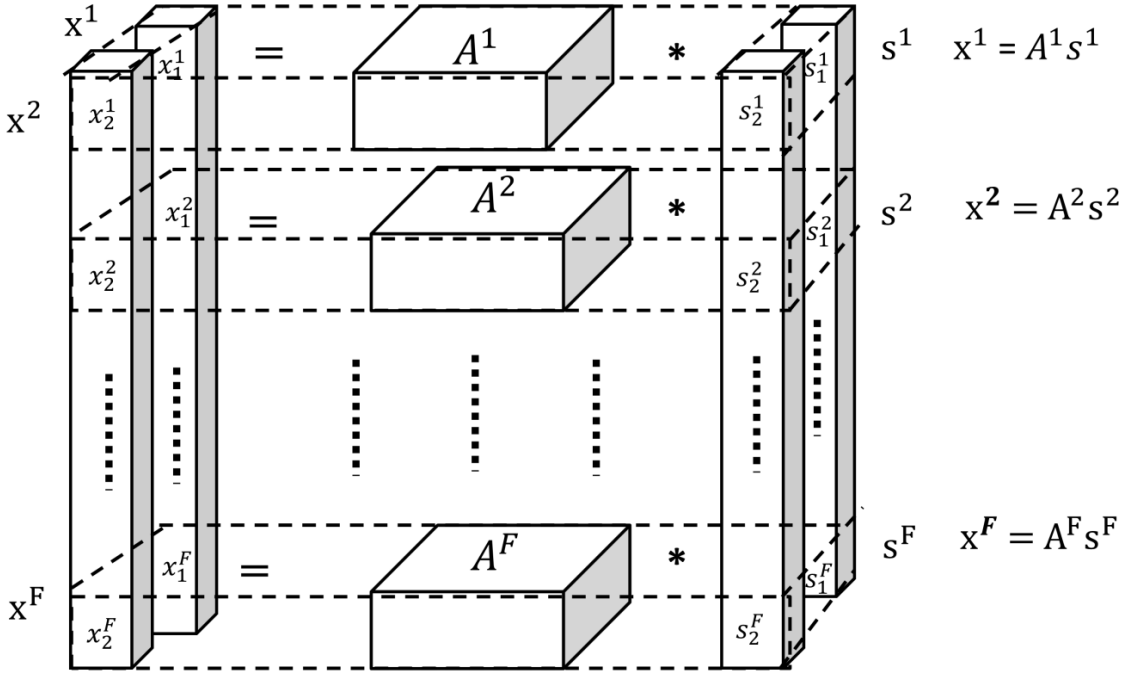


Figure 2.2 The instantaneous mixture model of IVA for 2 sources and 2 microphones the sources vector and the observed vector are represented as vertical pillars

An objective function is defined to separate multivariate sources from multivariate observations, Kullback-Leibler divergence between two functions as the measure of dependence is employed in IVA. The kullback-Leibler divergence between the joint probability density function $p = \hat{S}_1, \ldots \ldots \hat{S}_n$ and the product of probability density functions of the individual source vectors $\prod q(\hat{S}_i)$) can be defined as follow:

$$J = kL(p(\hat{s}_1, \ldots \ldots, \hat{s}_n) \parallel \prod q(\hat{s}_i))$$
$$= const - \sum_{k=1}^{K} \log|\det(W(k))| - \sum_{i=1}^{n} E[logq(\hat{s}_i)] \quad (2.30)$$

We can keep the dependency among the components of each vector, and remove the dependency between the source vectors if the cost function is minimized [18].

In literature, there is a different version of IVA such as NG-IVA, Fast-IVA and Aux-IVA [18]. In this study, Fast-IVA algorithm is used for BSS.

Fast fixed-point Independent Vector Analysis (Fast-IVA)

This algorithm utilizes Newton's method for updating the original IVA method, which converges quadratically and it's free from selecting an efficient learning rate. The quadratic Taylor series polynomial approximation is introduced in the notations of complex variables. Thus Newton's method can be applied in the update rules. A quadratic Taylor series polynomial can be used for a contrast function of complex-valued variables [18]. The contrast function used by Fast IVA is as follows:

$$\mathrm{J} = \sum_{i=1}^{n} \left(\mathrm{E}\left[\mathrm{G} \sum_{k=1}^{K} |\hat{\mathrm{s}}_i^{(K)}|^2 \right] - \sum_{k=1}^{K} \lambda_{\mathrm{i}}^{(\mathrm{K})} (\mathrm{w}_i^{(K)} (\mathrm{w}_{\mathrm{i}}^{(\mathrm{K})})^{\mathrm{H}} - 1 \right) \tag{2.31}$$

Where, λ_i is the *i*th Lagrange multiplier, and $\mathrm{w_i}^{(K)}$ denotes the ith row of the unmixing matrix *W*, *G* (·) is the nonlinearity function, which can take on several different forms as discussed in [18]. With normalization, the learning rule is:

$$(\mathrm{w}_{\mathrm{i}}^{(\mathrm{K})})^{\mathrm{H}} \leftarrow E\left[\acute{\mathrm{G}}\left(\sum_K |\hat{\mathrm{s}}_i^{(K)}|^2 \right) + \sum_K |\hat{\mathrm{s}}_i^{(K)}|^2 G''\left(\sum_K |\hat{\mathrm{s}}_i^{(K)}|^2 \right) \right]$$

$$\mathrm{x}\ (\mathrm{w}_{\mathrm{i}}^{(\mathrm{K})})^{\mathrm{H}} - \mathrm{E}\left[\left(\hat{\mathrm{s}}_i^{(K)} \right) * \acute{\mathrm{G}}\left(\sum_K \left| \hat{\mathrm{s}}_i^{(K)} \right|^2 \right) x^K \right] \tag{2.32}$$

Where $\mathrm{G}'(\cdot)$ and $\mathrm{G}''(\cdot)$ denote the derivative and second derivative of $\mathrm{G}(\cdot)$, respectively. If this is used for all sources, an un-mixing matrix W (k) can be constructed which must be de-correlated with

$$W^{(K)} \leftarrow (W^{(K)} \left(W^{(K)} \right)^H)^{-1/2} W^{(K)} \tag{2.33}$$

In this study, we implemented Fast-IVA in the frequency domain. Figure (2.8) shows the flowchart of this algorithm.

2.2 Speaker Recognition System

Speech is not only words or messages being spoken, speech carry information about language that is being said and also specific information about the speaker. Thus, this information used to achieve the goal of speech and speaker recognition systems. Speaker recognition includes several fields [29] that shown in fig 2.3.

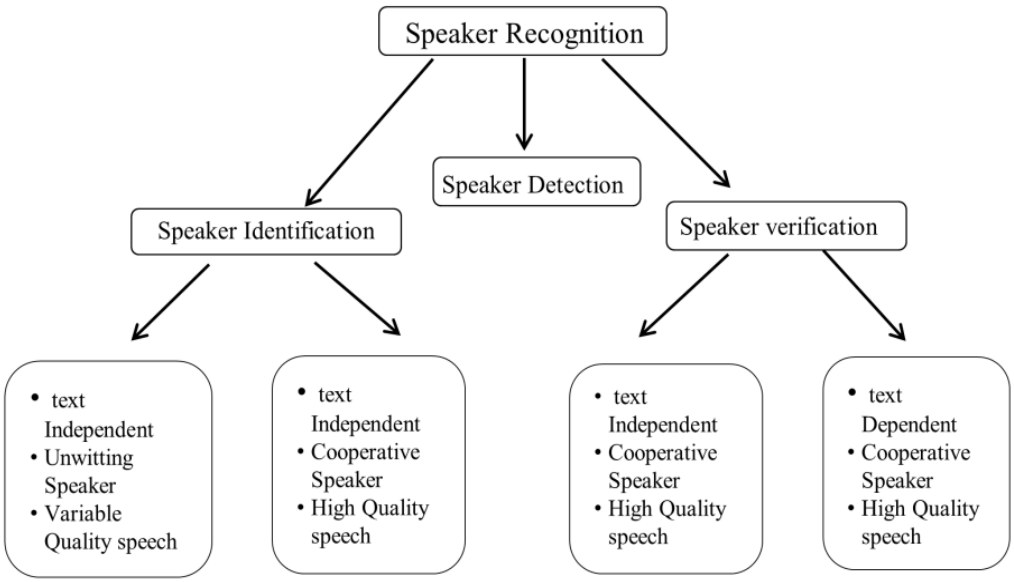


Figure 2.3 Speaker Recognition

This thesis work deals with speaker identification and speaker verification. In speaker identification, the goal is to specify the identity of the input speaker voice by finding which one of the known speaker sound group is best matches with input speaker sound sample.

In speaker verification, the goal is to determine from a voice sample if a person is whom he or she claims.

Typically, the standard speaker recognition system consists of two main processing: extracting features and classification, which is also known as pattern recognition.

2.2.1 Feature Extraction

In order to reduce the amount of the data of speech signal that used in speaker recogntion system and obtain the vocal characteristics the feature extraction process were utilized for both training and test data. Several approaches addressed the problem of feature extraction such as Linear Predictive Coding (LPC), Local Discriminant Bases

(LDB) [30]. In our work, we used Mel-frequency cepstral coefficients (MFCC) [31] which is the most popular and efficient in speech and audio processing. Fig 2.3 describes the diagram of (MFCC).

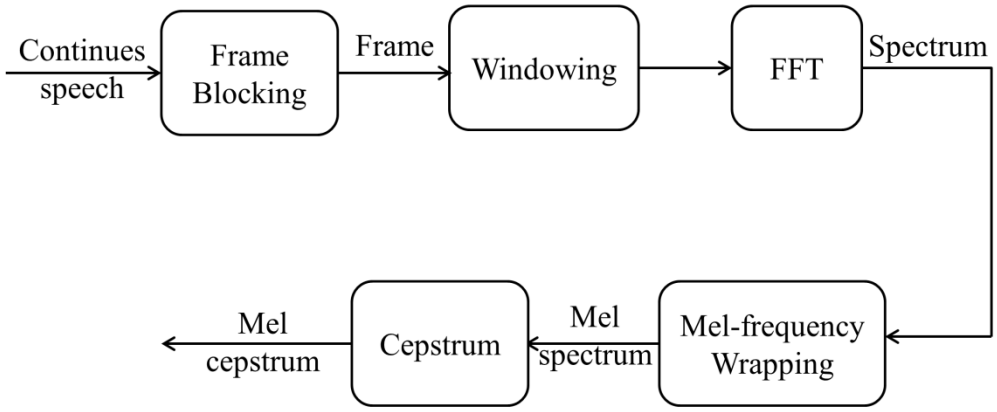


Figure 2.4 The structure of an MFCC processor

In order to obtain MFCCs we need several steps:

1. Frame Booking:

Theoretically, the speech signal is a non-stationary signal, so it is broken down into short frames. As a result, it seems stationary in each frame. This process is carryed out by blocking the speech signal into frames. Each frame size between 30 to 100-millisecond. Fig 2.5 shows the first frame with N samples and the adjusted frame with M samples. The overlapping is occurred by N-M . Overlapping is used to smooth the transition from one frame to the other.

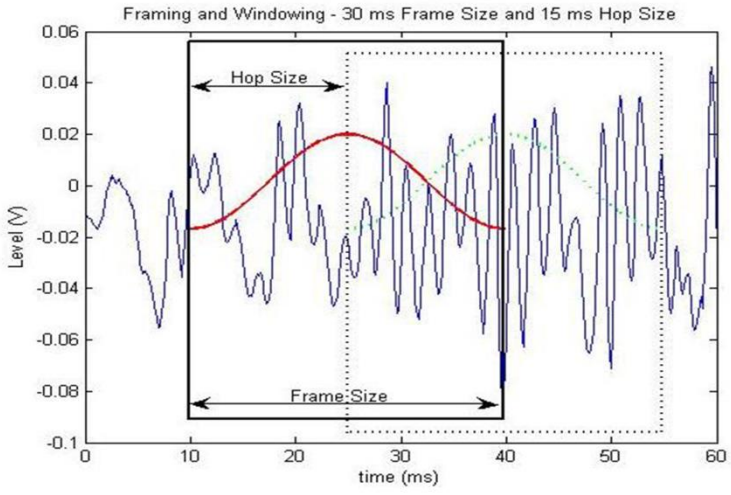


Figure 2.5 Frame blocking and overlapping

2. Windowing:

In this step, we minimize the spectral distortion and the discontinuities by using the window. So the signal will be zero at the beginning and ending of each frame.

3. Fast Fourier Transform FFT:

We need to apply the Fast Fourier transform (FFT) for each frame in order to transform the signal from time domain into frequency domain. As we know FFT is the fast algorithm of discrete Fourier transform (DFT) that can be define as follow:

$$X_k = \sum_{n=0}^{N-1} x_n \, e^{-j2\pi kn/N} , \qquad k = 0,1,2 \dots \dots N-1 \tag{2.34}$$

4. Mel- Frequency Wrapping :

Warpping step containing filtering the signal energy by triangular-band filters called (mel_filters), these filters makes our features match more closely to the human auditory system. Fig 2.6 show an example of the mel –flters bank

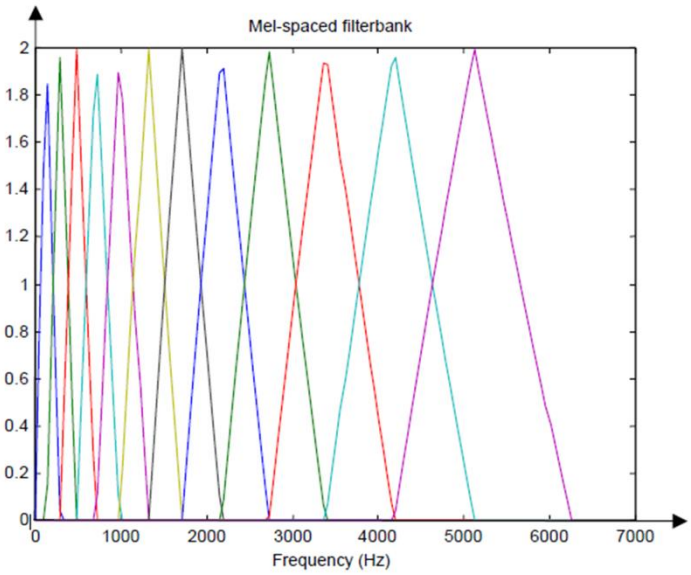


Figure 2.6 Example of mel-spaced frequency bank

5. Cepstrum:

Since these filters are applied in the frequency domain, discrete cosine transform (DCT) is used to convert the log Mel spectrum back to the time domain in the final step Cepstrum, so we can get the MFCCs as follow:

$$c_n = \sum_{K=1}^{K} (log\ S_K)\ cos\left[n\left(K - \frac{1}{2}\right)\frac{\pi}{K}\right] \quad n = 0,1 \ldots\ldots, K-1 \tag{2.35}$$

Where K represents the number of Mel spectrum coefficients and S_K as an output of mel_filters. Finally, we can obtain *n* number of MFCCs as features for each speech frame. These features can be represented as the feature vector, thus the feature vectors for all frames can be represented as feature matrix.

2.2.2 Classification

In classification the feature matrix is compared with calculated speaker models, thus recognize the speech sample is best fits with which model. In this work, for identification task, we chose Gaussian Mixture Model (GMM) and the I-vector technique for verification task.

The likelihood ratio is frequently used to explain the variability of speech when biometric data is used for identity verification [32]. The speech utterance recorded from a sensor (source), and the records stored in the database (references) are described with the same type of features to form a dataset. The main goal is to determine whether the source and reference are derived from the same speaker or not. An ideal system should be able to provide reliable verification results without being affected by speech length and quality.

A source record X is transformed into the data vector X = [x1, x2, ..., xd] after passing through the feature extraction process. The speaker, the source of the speech vector X, is modeled mathematically by the set of mean vector μ and covariance matrix Σ from the Gaussian distribution of all the speech utterances stored in the dataset that belongs to the speaker, and this set is denoted by λ0. The likelihood ratio (LR) of X belonging to the target speaker is expressed as follows, with the target speaker model λ0 and the

alternative model λA (Universal Background Model, UBM) which is generated by the other registered speaker's utterances.

$$\frac{P(X|\lambda_0)}{P(X|\lambda_A)} \tag{2.36}$$

For scaling purposes, the logarithm of likelihood ratio is often used:

$$\Lambda(X) = \log P(X|\lambda_0) - \log P(X|\lambda_A) \tag{2.37}$$

In verification systems, the general consensus is to verify the source X as an authentic speech sample from the target speaker if the calculated LR is greater than some threshold. Determining an appropriate threshold is; however, a difficult task. Since our main problem is not speaker verification, for the purpose of simplification we used maximum likelihood ratio criteria.

Gaussian Mixture Model

This method has strong classification tools in pattern recognition, especially in speech recognition. Furthermore, (GMM) has better performance than Hidden Markov models (HMM) in text independent speaker recognition [32]. Moreover, (GMM) has based on well-understood statistical models.

The other reason for using Gaussian mixture densities for speaker identification is the capability of Gaussian basis functions for modeling a large class of sample distributions. A GMM can form smooth approximations to arbitrarily shaped densities. Thus, for these reasons that we mentioned above, we choose Gaussian Mixture Model (GMM) in our work to calculat speaker models. (GMM) is the weighted sum of N components Gaussian densities [32] as the equation follow:

$$p(x|\lambda) = \sum_{i=1}^{N} w_i \, p_i(x) \tag{2.38}$$

Where x is the extraction features vector and w_i representing the mixture weights.

Each component Gaussian densities are computed as follow:

$$p_i(x) = \frac{1}{(2\pi)^{D/2}|\sum_i|^{1/2}} \exp\left\{-\frac{1}{2}(x-\mu_i)'\sum_i^{-1}(x-\mu_i)\right\} \quad (2.39)$$

Where μ_i is the mean vector and $\sum_i$ is the covariance matrix and they considered as GMM parameters. By considering the log of Gaussian distribution and after the derivative is carried out [33], the linear super-position of Gaussians can be defined as follow:

$$P(x) = \sum_{k=1}^{K} \pi_k p_k(x) \quad (2.40)$$

Where K is the number of mixture Gaussians and π_k is the Mixing coefficient

The Expectation Maximization algorithm (EM) was used to compute the GMM parameters. Typically, EM algorithm is an iterative method that has two basic steps

1. **Estimation step**: in this step the parameter values are estimated by computing the latent variableγ_j.

$$\gamma_j(x) = \frac{\pi_k p_k(x)}{\sum_{j=1}^{K} \pi_j p_j(x)} \quad (2.41)$$

2. Maximization step: updates the value of GMM parameters depending on the latent variable γ_j .

$$\mu_j = \frac{\sum_{n=1}^{N} \gamma_j(x_n) x_n}{\sum_{n=1}^{N} \gamma_j(x_n)} \qquad \sum_j = \frac{\sum_{n=1}^{N} \gamma_j(x_n - \mu_j)(x_n - \mu_j)^T}{\sum_{n=1}^{N} \gamma_j(x_n)}$$

$$\pi_j = \frac{1}{N}\sum_{n=1}^{N} \gamma_j(x_n) \quad (2.42)$$

I-vector

I-vectors are widely used in speaker recognition field. The i-vector extraction can be considered as a compressed process that reduces the dimensionality of speech-session. Each patterned speech (speech signal component) consists of speakers and channel dependent components as given in Equation (2.43). These are speaker-independent

Universal Background Model (UBM) components (m), speaker components (Vy), channel components (Ux), and residual components (Dz).

$$S = m + V\gamma + Ux + Dz \tag{2.43}$$

The i-vector approach assumes that $V\gamma + Ux + Dz$ component resides on a lower-dimensional subspace of the Eq.(2.43). According to this approach, a speech utterance is modeled as in Eq.(2.44) where M denotes the super-vector connected to the utterance, m_UBM denotes the super-vector of the UBM, T denotes the matrix of eigen-voices (total variation matrix) and x is the i-vector to be extracted [34], [35].

$$M = m_{UBM} + Tx \tag{2.44}$$

Unlike GMM, which uses UBM and MAP to be used in modeling a speaker for verification, the i-vector method passes the entire data set through the same i-vector extraction algorithm. By adapting the T matrix via EM, a speaker model is constructed for all target speakers.

For a mixture c of the UBM, Baum-Welch Null (Nc) and 1st degree (Fc) statistics are assumed to summarize an uncompleted observation of each utterance super-vector. As shown in Eq.(2.45) and Eq.(2.46), statistical values are calculated for $\gamma_t(c) = p(c|O_t, \lambda_{ubm})$ where O_t is the t'th observation.

$$N_c(S) = \sum_{t \in S} \gamma_t(C) \tag{2.45}$$

$$F_c(s) = \sum_{t \in S} \gamma_t(c) Y_t - m_c N_c \tag{2.46}$$

Hence, the i-vector x is extracted using Eq.(2.47) and Eq.(2.48).

$$\mathrm{Cov}(x, x) = (I + \sum_{c} N_c T_c^* T_c)^{-1} \tag{2.47}$$

$$x = \mathrm{Cov}(x, x) \sum_{c} T_c^* F_c \tag{2.48}$$

In this multidimensional model, dimensionality reduction is performed with GPLDA [35] to extract non-speaker-specific dimensions.

2.3 The Performance Evaluation

In order to measure the performance of algorithms that we used in our work, we utilized SAR, SDR, SIR, and SNR as performance metrics in sources separation field, and EER measurement in speaker recognition field.

2.3.1 Evaluation for Sources Separation

There are several performance measurement metrics to evaluate the quality of estimated signals obtained by BSS methods. The performance of BSS algorithms is measured by comparing each estimated source *ŝj* to a given true source *sj*. The measurement processing includes two successive steps [36]. The first step involves decomposing *ŝj* as:

$$\hat{s}_j = s_{target} + e_{interf} + e_{noise} + e_{artif} \tag{2.50}$$

Where $s_{target} = f(s_j)$ denotes the version of s_j modified by an allowed distortion, s_{interf} , s_{noise} and s_{artif} denotes the interferences, noise, and artifacts error terms, respectively.

The second step involves computing the energy ratios in order to estimate the relative amount of each of these four terms either on the local frames of the signal or the whole signal duration. The way of how to decompose into four terms are given in [36] in detail. Relevant energy ratios between these terms are defined.

After the decomposition of $\hat{s}_j$ following the procedures given in [36], numerical performance criteria was defined by computing energy ratios expressed in decibels. Definition of source-to-distortion ratio (SDR), the source- to- interference ratio (SIR),

source to artifact ratio SAR and source to noise ratio (SNR) are given below, respectively

$$\text{SDR} := 10\log_{10}\frac{||s_{\text{target}}||^2}{||e_{\text{interf}}+e_{\text{noise}}+e_{\text{artif}}||^2} \tag{2.51}$$

$$\text{SIR} := 10\log_{10}\frac{||s_{\text{target}}||^2}{||e_{\text{interf}}||^2} \tag{2.52}$$

$$\text{SAR} := 10\log_{10}\frac{||s_{\text{target}}+e_{\text{interf}}+e_{\text{noise}}||^2}{||e_{\text{artif}}||^2} \tag{2.53}$$

$$\text{SNR} := 10log_{10}\frac{||s_{target}+e_{interf}||^2}{||e_{noise}||^2} \tag{2.54}$$

2.3.2 Evaluation for Speaker Recognition

The equal error rate (EER) is a measure to evaluate the speaker recognition system performance. Whenever the value of EER is lower, the system performance is better [37]. Typically in order to obtain the EER value, two values have to be found:**False Positive Rate FPR** representing the value of false acceptance of impostor patterns divided by all the number of all impostor patterns; **False Negative Rate FNR** representing the value of false rejection of client pattern divided by the total number of client patterns. Thus the intersect point of FPR and FNR which is the same for both of them represent the EER value.

2.4 The Methods Implementation

2.4.1 Fast-Fixed Independent Compenent Analysis algorithm Implementation

Fast-ICA is implemented in MATLAB. This algorithm uses the fixed-point algorithm developed by Aapo Hyvarinen [38]. The flowchart in figure 2.7 illustrate the steps of Fast-ICAalgorithm [35].

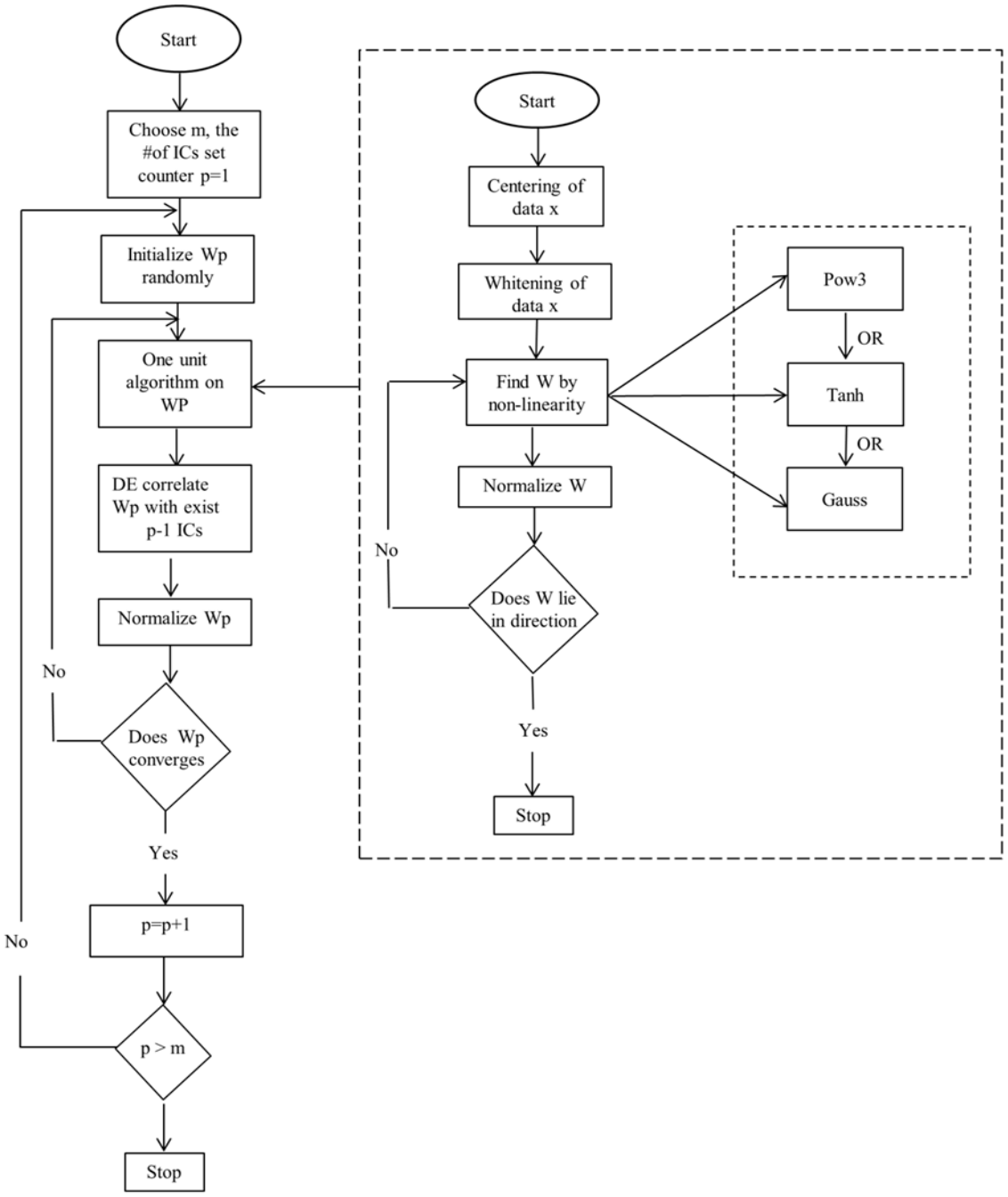


Figure 2.7 Fast –ICA algorithm flowchart

As we mentioned before the fast fixed point algorithm has two approaches: the symmetric and the default approaches. The Fig 2.7 illustrates the default approach. The Fast-ICA algorithm starts with setting the counter (*p*) with the same number (*m*) of the original sources.

Considering a random values for the first un-mixing matrix W*p*, for each iteration, this algorithm will estimate one source by the following steps:

Centering: In this step we center the observed data vector ***x*.** Thus, the product of this step is a vector with zero mean.

Whitening: In whitening step the mixing matrix is transformed into a new orthogonal matrix. Hence, the problem dimension is reduced as we explain in whitening preprocessing section.

Find the w by non-linearity: Finding the un-mixing matrix by using one of non-linearity (pow3, than, Gauss) according to the inputs of the function (*fpica* function).

Normalize **W:** Normalizing the un–mixing matrix (divided it by its norm). If the matrix values achieve the function condition, the algorithm estimated the W matrix for the first source.

The algorithm will start again to estimate the un-mixing matrix for the next source if the counter value *p* is not equal of the number of the sources.

2.4.2 Kernel Independent Component Analysis algorithm Implementation

Implementing the kernel-ICA algorithm on the given mixtures returns a un-mixing matrix W. Thus, the estimated sources are obtained by multiplying the un-mixing matrix by the mixture signal.

The Kernel- ICA algorithm consists of the following steps

Centering and scaling: These processes simplify the work of the algorithm by reducing the dimension of the problem.

SVD Technique: The second step is to apply the singular value decomposition (SVD). This linear algebra technique supplies a method for dividing the matrix into several simpler parts. This technique is used in this algorithm to divide the matrix (from the previous step) into three simpler matrixes. They are simpler because each of them has fewer parameters to infer, so it will be simple also to invert them. Thus, we can get the initial value of the un-mixing matrix.

Applying the Steepest descent method: Applying this method in order to find the minima in the Stiefel manifold of orthogonal matrices. After finding the global minimization of the contract function, in this work, we use 'kcca' the default contrast function. Hence, the output of this step is the un- mixing matrix and by multiplying it with the observed data vector $\boldsymbol{x}$ we will get the independent components.

2.4.3 Fast-Fixed-Point Independent Vector Analysis Algorithm Implementation

Fast –IVA algorithm was implemented in MATLAB as written by Taesu Kim, 2005 [17].

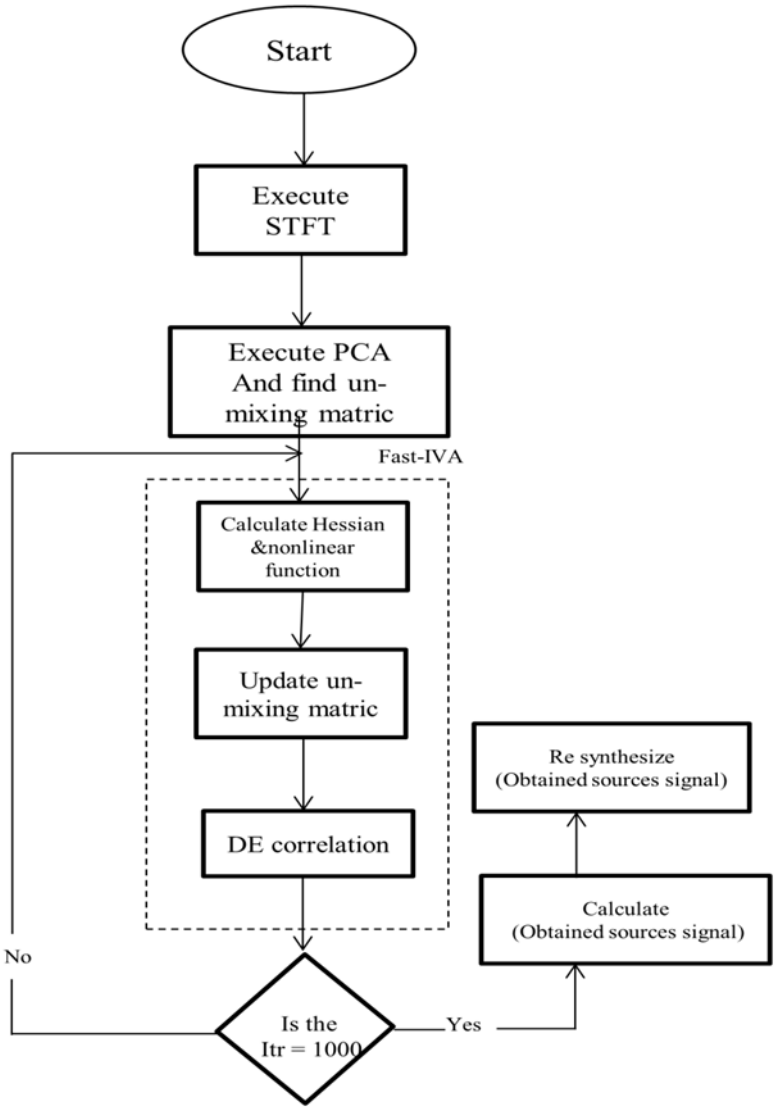


Figure 2.8 Fast –IVA flowchart

As flowchart in figure 2.8 shows, the Fast-IVA algorithm consists of these steps:

STFT: Since the speech mixture signal is un-stationary signal a short-time Fourier transform is used, in order to get short blocks that are stationary. The sampling rate of 16 kHz and a window size of 1024 samples have been used.

PCA Method Implementation: The PCA method is applied for each frequency bin in order to simplify the problem by reducing the dimension and find the principle components. In this step, we initialize the value of the un-mixing matrix.

Calculate Hessian and nonlinear function: As we mention previously, the fast –IVA algorithm uses Newton's method and a quadratic Taylor series polynomial as a contrast function of complex-valued variables. Thus, in order to calculate the quadratic approximation we need first find the Hessian matrix of it.

ISTFT: The inverse short – time Fourier transform is applied to transform the signal back to the time domain after the separation is done. Thus, the separated signals could be heard

2.4.4 Speaker Recognition System

The speaker recognition system is implemented in this thesis by using the MSR Identity Toolkit v1.0 and voicebox which was downloaded from [39]. As mentioned before in 2.2, this system workes by creating GMM based on MFCCs for feature extracting. All speech signals that entered to this system have the frequency of 16 KH. The universal background model UBM is the GMM based model and we have used the MAP (Maximum a posteriori) method for the adaptation.

CHAPTER 3

RESULTS AND DISCUSSION

3.1 Proposed Scenarios

In order to examine the performance of the BSS methods that we used in this work (the Fast-ICA, the Kernel-ICA, and Fast –IVA) we proposed three different scenarios. In each scenario we mixed the sources signals as pairs by using random valuse. Thus, obtaining the mixed signal X as shown in Eq(3.1) , Eq (3.2) :

$$M_1 = y * S_1 + z * S_2$$

$$M_2 = z * S_1 + y * S_2 \quad (3.1)$$

$$X = M_1 + M_2 \quad (3.2)$$

Where y, z are represent the random values added to the S_1 , S_2 the orginal sources. M_1 , M_2 are the left and right channel respectively of the stereo mixed signal X .The first proposed scenario includes measuring and comparing the performance of these BSS methods for separating mixing speech signals without noise, as shown in Figure 3.1. Figure 3.2 illustrates the second scenario which shows the performance of these methods for separating mixed speech signals with the Additive white Gaussian noise (AWGN) added to signals before mixing. In the third scenario, we add the Additive white Gaussian noise (AWGN) [40] to the signals after mixing as shown in Figure 3.3. Since Gaussian noise is added to the sources or mixtures in second and third scenarios, Savitzky-Golay smoothing filter [41] is performed to enhance the signals before the separation.

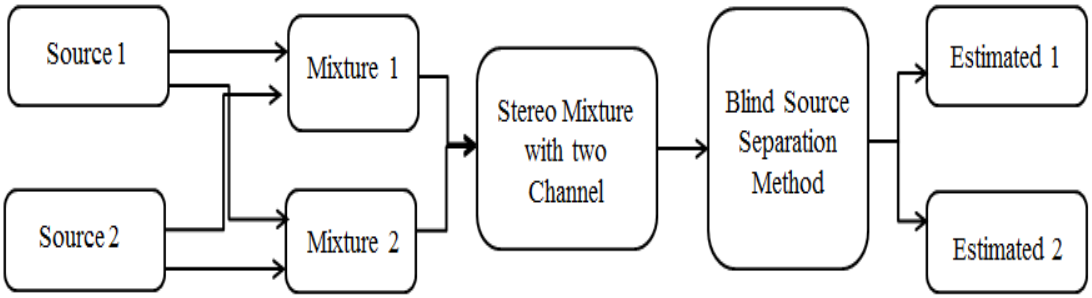


Figure 3.1 Illustration of the first scenario

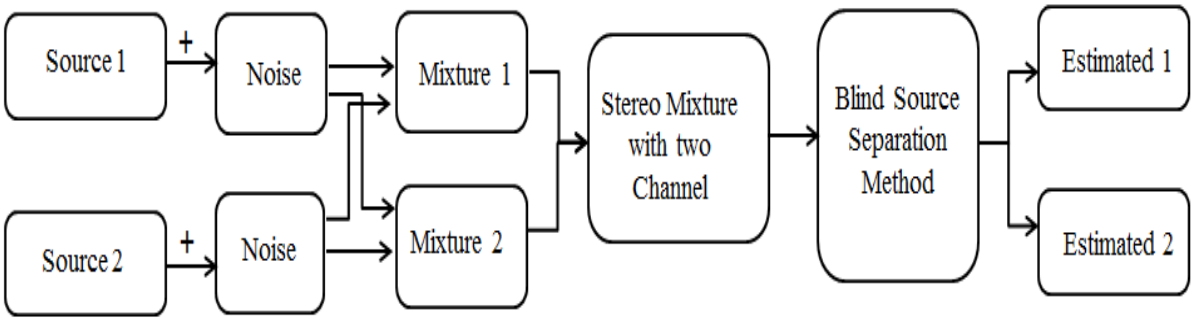


Figure 3.2 Illustration of the second scenario

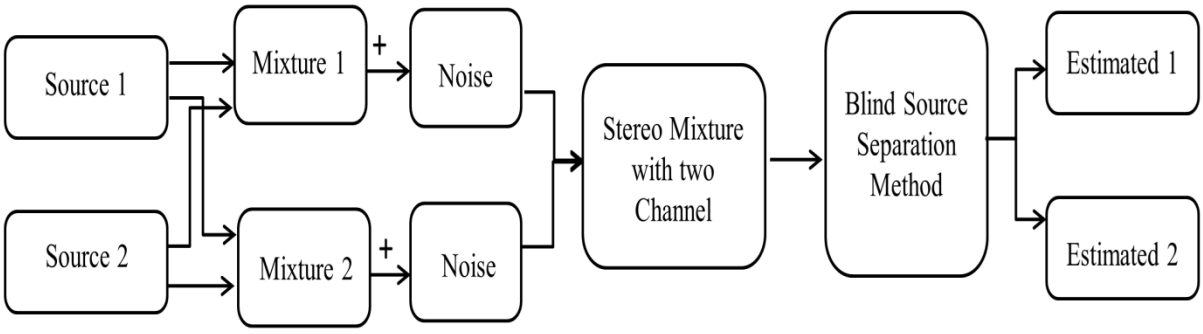


Figure 3.3 Illustration of the third scenario

3.2 Additive white Gaussian noise

It is a simple model of noise. We use the term "**additive"** because we add this noise to the signal instead of multiplying it with the signal.

$$y(t) = x(t) + n(t) \tag{3.3}$$

Where x(t) is the clean signal and n(t) is the noise.

It is **Gaussian** because it has 0 mean and its variance value depends on its power. Of course, it is not deterministic otherwise we can subtract it from y (t) signal.

In the frequency domain, this type of noise has the same power for all frequencies so that it is called white. It has a flat level in every frequency.

3.3 Savitzky -Golay Filters:

These filters are known for its ability to provide quick and easy smoothing for the data and it can also determine the derivatives at each point in a set of data that equally spaced in the abscissa. Savitzky – Golay filters use the polynomial model that fitting the subset of the data. Moreover, the coefficients of the polynomial model are used to determine the calculated derivatives. Some factors can affect the determination of the filters, the order of the polynomial, the number of data points that used in the fit and the time at wich the smooth value and the derivatives are discovered. In another word with considering that the input data is set of *y*-values, the Savitzky – Golay filter is a vector with size equal to the number of sequences. *y*-values to be used in the determination and whose dot product with those *y*-values provides the desired derivative.

3.4 The Datasets

In this thesis, two data sets are used.

The first data is in the Arabic language. We collected this data by recording the voice of 13 speakers: 3 female, and 10 male, in a real room, each record being length of 10 sec in with 16 kHz.

The second data is the ELSDSER [42] data that contains voice records from 22 speakers: 10 female, 12 male, each record being length of 10sec in the English language with 16 kHz. The ages are covered from 24 to 63. The test set of ELSDSER data consists of 2 records for each speaker, (2*22) 44 records. The training set of ELSDSER data consists of 7 records for each speaker, (7*22) 154 records.

3.5 Mixing Approach

The speech signals are mixed in pairs by using different parameters (each speaker record is mixed with the rest of speakers records of the data).So that, as shown in table 3.1 the Arabic data 77 mixture signals are created from 13 original signals. Thus, we get 154 separated signals. For ELSDSR data there were 231 different mixture signals from 22 original sources. So we obtain 462 separating signals.

Table 3.1 mixing approach

The data set	**Number of speakers**	**The mixing probabilities**
The Arabic data set	13	**(1,2)**; (1,3); (1,4); (1,5); (1,6);(1,7);(1,8);(1,9);(1,10);(1,11); (1,12); (1,13);**(2,3);** (2,4); (2,5); (2,6); (2,7); (2,8); (2,9); (2,10);(2,11); (2,12); (2,13); **(3,4);** (3,5); (3,6); (3,7); (3,8); (3,9); (3,10);(3,11); (3,12); (3,13);**(4,5);** (4,6); (4,7); (4,8); (3,9); (4,10);(4,11); (4,12); (4,13); **(5,6);** (5,7); (5,8); (5,9); (5,10);(5,11); (5,12); (5,13); **(6,7);**(6,8); (6,9); (6,10);(6,11); (6,12); (6,13); **(7,8);** (7,9); (7,10);(7,11); (7,12); (7,13); **(8,9);** (8,10);(8,11); (8,12); (8,13); **(9,10);**(9,11); (9,12); (9,13);**(10,11);** (10,12); (10,13); **(11,12);** (11,13);(12,13).

Table 3.2 Mixing Approach (cont'd)

The data set	Number of speakers	The mixing probabilities
The ELSDSR data	22	(1,2); (1,3); (1,4); (1,5);(1,6);(1,7);(1,8);(1,9);(1,10);(1,11); (1,12); (1,13); (1,14); (1,15); (1,16); (1,17); (1,18);(1,19);(1,20);(1,21);(1,22); (2,3); (2,4); (2,5); (2,6);(2,7);(2,8);(2,9);(2,10);(2,11); (2,12); (2,13); (2,14); (2,15); (2,16); (2,17); (2,18);(2,19);(2,20);(2,21);(2,22); (3,4); (3,5); (3,6);(3,7);(3,8);(3,9);(3,10);(3,11); (3,12); (3,13); (3,14); (3,15); (3,16); (3,17); (3,18);(3,19);(3,20);(3,21);(3,22); (4,5); (4,6);(4,7);(4,8);(4,9);(4,10);(4,11); (4,12); (4,13); (4,14); (4,15); (4,16); (4,17); (4,18);(4,19);(4,20);(4,21);(4,22); (5,6);(5,7);(5,8);(5,9);(5,10);(5,11); (5,12); (5,13); (5,14); (5,15); (5,16); (5,17); (5,18);(5,19);(5,20);(5,21);(5,22); (6,7); (6,8);(6,9);(6,10);(6,11); (6,12); (6,13); (6,14); (6,15); (6,16); (6,17); (6,18);(6,19);(6,20);(6,21);(6,22); (7,8); (7,9);(7,10);(7,11); (7,12); (7,13); (7,14); (7,15); (7,16); (7,17); (7,18);(7,19);(7,20);(7,21);(7,22); (8,9); (8,10);(8,11); (8,12); (8,13); (8,14); (8,15); (8,16); (8,17); (8,18);(8,19);(8,20);(8,21);(8,22); (9,10); (9,11); (9,12); (9,13); (9,14); (9,15); (9,16); (9,17); (9,18);(9,19);(9,20);(9,21);(9,22); (10,11); (10,12); (10,13); (10,14); (10,15); (10,16); (10,17); (10,18);(10,19);(10,20);(10,21);(10,22); (11,12); (11,13); (11,14); (11,15); (11,16); (11,17); (11,18);(11,19);(11,20);(11,21);(11,22); (12,13); (12,14); (12,15); (12,16); (12,17); (12,18);(12,19);(12,20);(12,21);(12,22); (13,14); (13,15); (13,16); (13,17); (13,18);(13,19);(13,20);(13,21);(13,22); (14,15); (14,16); (14,17); (14,18);(14,19);(14,20);(14,21);(14,22); (15,16); (15,17); (15,18);(15,19);(15,20);(15,21);(15,22); (16,17); (16,18);(16,19);(16,20);(16,21);(16,22); (17,18); (17,19);(17,20);(17,21);(17,22); (18,19);(18,20);(18,21);(18,22); (19,20);(19,21);(19,22); (20,21);(20,22);(21,22)

3.6 Results of Experiment 1

In this experiment, the three proposed scenarios that we mentioned previously are applied on Arabic data. The performance of Fast-ICA, Kernel-ICA and Fast-IVA methods for separating mixing speech signals was compared. The Fast-IVA has better performance than the Fast- ICA based methods according to performance metrics of Source-to-Artifact Ratio, Source-to-Distortion Ratio, and Source-to-Noise Ratio. But Fast-ICA methods give better results than Fast-IVA according to the Source-to-Interference Ratio as shown in Table 3.2, Table 3.3, and Table 3.4.

Table 3.3 Comparison Results for the first scenario for Arabic data

Algorithm	Criteria	SAR(dB)	SDR(dB)	SIR(dB)	SNR(dB)
Fast-ICA	**Average**	6.114	6.087	29.671	-13.47
	STdev	1.084	1.087	2.3746	3.0666
Kernel-ICA	**Average**	6.1374	6.1130	29.9469	-13.328
	STdev	1.0889	1.0904	2.03176	3.14367
Fast-IVA	**Average**	21.651	18.873	24.181	8.6396
	STdev	7.6494	8.278	9.4262	8.7521

Table 3.3 Comparison Results for the second scenario for Arabic data

Algorithm	Criteria	SAR(dB)	SDR(dB)	SIR(dB)	SNR(dB)
Fast-ICA	**Average**	5.0701	5.0289	27.705	-13.660
	STdev	1.8008	1.7975	3.0743	2.8805
Kernel-ICA	**Average**	4.792	4.822	28.657	-13.611
	STdev	2.678	2.976	4.987	2.954
Fast-IVA	**Average**	8.8899	6.4275	15.2160	2.0444
	STdev	3.2868	5.5514	10.2810	1.8588

Table 3.4 Comparison Results for the third scenario for Arabic data

Algorithm	Criteria	SAR(dB)	SDR(dB)	SIR(dB)	SNR(dB)
Fast-ICA	**Average**	5.0037	4.9645	27.8722	-13.745
	STdev	1.8200	1.8168	3.1256	2.7637
Kernel-ICA	**Average**	4.9694	4.9435	28.9186	-14.035
	STdev	1.8174	1.8199	2.61689	2.6280
Fast-IVA	**Average**	8.8166	6.3595	15.0365	2.0093
	STdev	3.3440	5.4436	9.9977	1.9585

3.7 Results of Experiment 2

In this experiment, we focus only on Fast-ICA and Fast-IVA methods. We applied these methods to ELSDSR data set. The performance of Fast-ICA and Fast-IVA methods for separating mixing speech signals was compared. For each three scenarios, again the Fast-IVA has better performance than the Fast-ICA method according to performance metrics of Source-to-Artifact Ratio, Source-to-Distortion Ratio, and Source-to-Noise Ratio. But Fast-ICA method give better results than Fast-IVA according to the Source-to-Interference Ratio as shown in Table 3.5, Table 3.6, and Table 3.7.

Table 3.5 Comparison Results for the first scenario for ELSDSR data

Algorithm	Criteria	SAR(dB)	SDR(dB)	SIR(dB)	SNR(dB)
Fast-ICA	**Average**	7.8727	7.7271	25.8788	-20.75
	STdev	0.7774	0.8123	4.55227	3.141
Fast- IVA	**Average**	14.063	10.079	13.7579	6.7344
	STdev	5.8238	7.0582	8.19038	5.4886

Table 3.6 Comparison Results for the second scenario for ELSDSR data

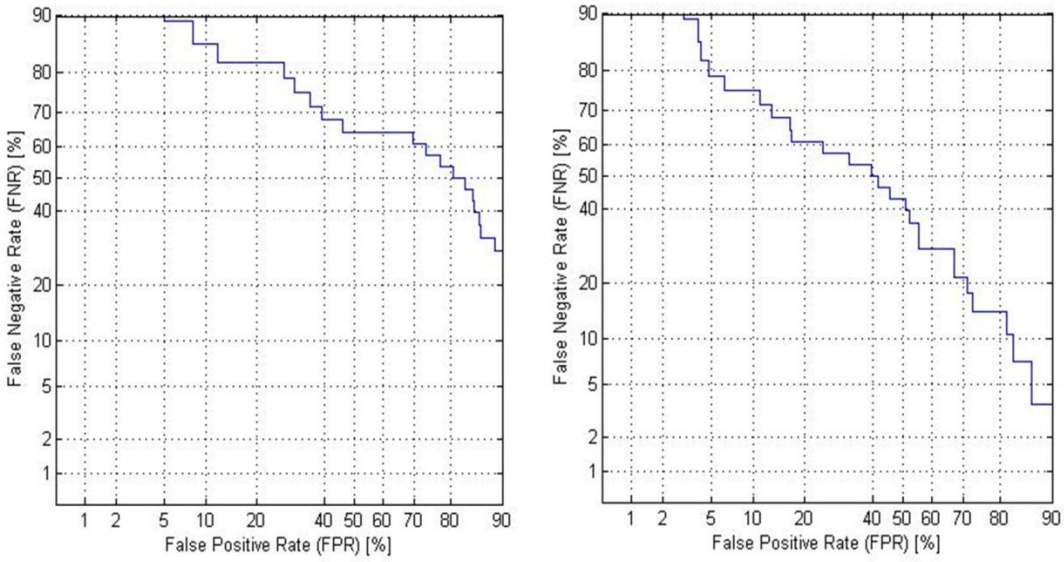
Algorithm	Criteria	SAR(dB)	SDR(dB)	SIR(dB)	SNR(dB)
Fast-ICA	**Average**	7.17496	6.9337	24.3245	-20.839
	STdev	1.0280	1.5377	5.0973	2.47148
Fast-IVA	**Average**	8.8185	4.9718	9.8079	2.81872
	STdev	3.2309	5.2678	7.9537	2.39165

Table 3.7 Comparison Results for the third scenario for ELSDSR data

Algorithm	Criteria	SAR(dB)	SDR(dB)	SIR(dB)	SNR(dB)
Fast-ICA	**Average**	7.120913	6.909553	24.39968	-20.8539
	STdev	1.018439	1.235233	4.612443	2.463391
Fast-IVA	**Average**	8.835028	5.151567	9.95961	2.966636
	STdev	2.922471	4.998545	7.744064	2.292565

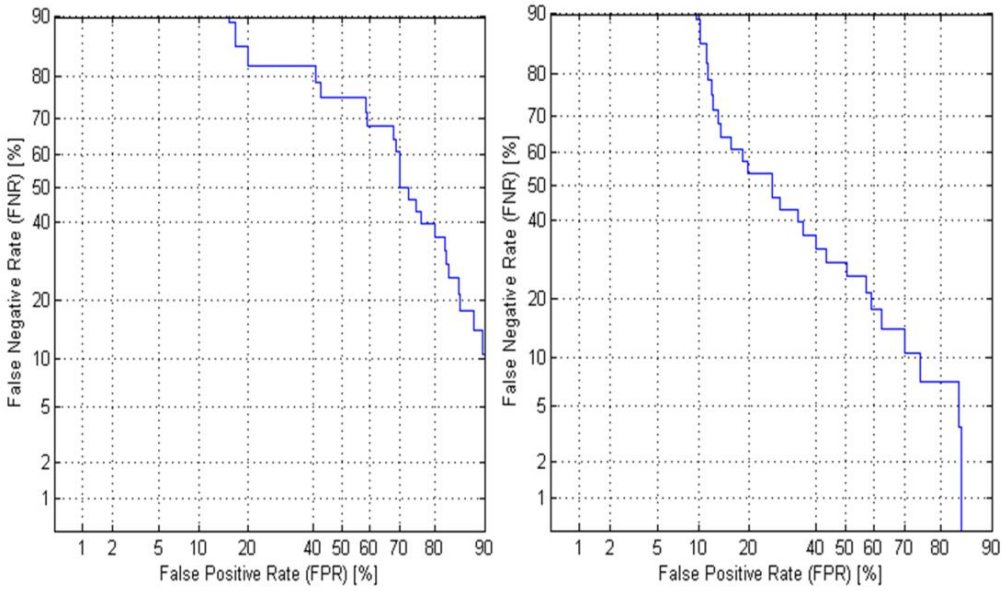
3.8 Results of Experiment 3

Speaker recognition system including Identification and Verification tasks were applied on the separated signals that we obtained from the second experiment. According to the results of the Identification task that shown in figures 3.4, 3.5 and 3.6, the Fast-IVA has higher performance than Fast –ICA. The same result is also clear in table 3.8 for the Verification task.



EER for Fast-ICA=64.2857 EER for Fast-IVA=42.8571

Figure 3.4 Results of the ID-recognition for the first scenario



EER for Fast-ICA=67.8571 EER for Fast-IVA=35.7143

Figure 3.5 Results of the ID-recognition for the second scenario

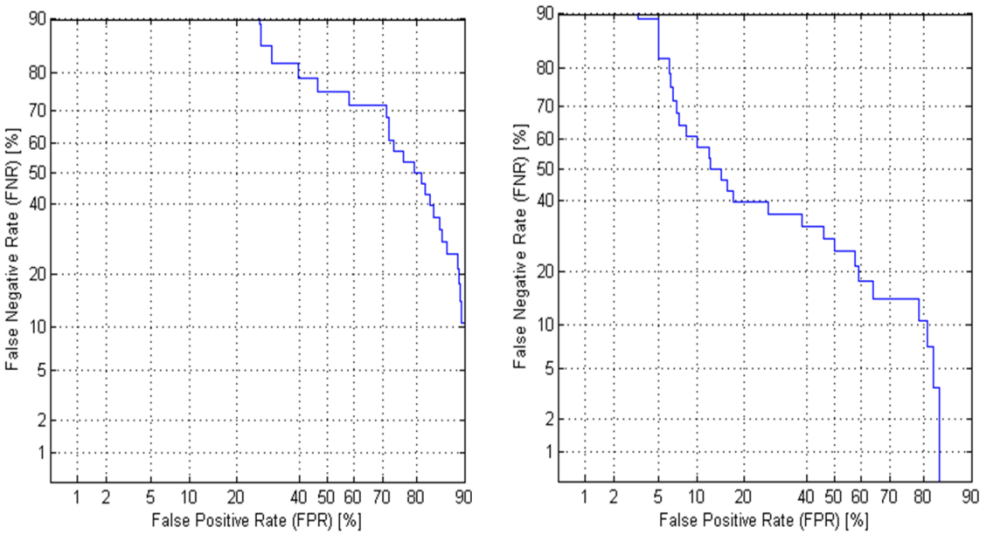


Figure 3.6 Results of the ID-recognition for the Third scenario

Table 3.8. EER values of Speaker Recognition system

Scenario number	Algorithm	EER value I-Vector	EER value GMM
First scenario	Fast-ICA	15.806 %	64. 285 %
	Fast-IVA	10.210 %	42.857 %
Second scenario	Fast-ICA	35.708 %	67.857 %
	Fast-IVA	34.898 %	35.714%
Third scenario	Fast-ICA	38.169 %	71.152 %
	Fast-IVA	34.472 %	35.714 %

3.9 Discussion and Future work

In this study, we make comparison among the most three popular BSS methods (Fast-ICA, Kernel-ICA, and Fast-IVA). As a major result of our investigation that Fast – IVA method has better performance than Fast-ICA and Kernel-ICA. We can minutely discuss our results of our experiments as follow:

The Results of the first experiment on Arabic data: We used the performance metrics of Source-to-Artifact Ratio (SAR), Source-to-Distortion Ratio (SDR), Source-to-Interference Ratio (SIR) and Source-to-Noise Ratio (SNR). Since the Artifact, Distortion, Interference and Noise value of the signal represents the denominator of Equations (34, 35, 36, 37) that we use to calculate the value of those metrics. Therefore,

whenever the values of these metrics are highest, that mean the BSS methods have better performance. So according to Table 3.2 which show the result for the first proposed scenario, the Kernel-ICA method has better performance than Fast - ICA method. But, the Fast-IVA method has better performance than the Kernel-ICA method and Fast - ICA method. In terms of Table 3.3 and Table 3.4 that show the results for the second and third scenarios respectively, the Fast - ICA method has better performance than Kernel-ICA method. But, the Fast-IVA method has better performance than the Kernel-ICA method and Fast - ICA method. On the other hand, the ICA methods give better results than Fast-IVA according to the Source-to-Interference Ratio (SIR) that was the unexpected result but we may get this result according to applying the Fast – IVA in the frequency domain.

The Results of the Second experiment on ELSDSR data: According to the Table 3.5, 3.6 and Table 3.7 that show the results for our three proposed scenarios respectively, the Fast-IVA method has better performance than the Fast -ICA method and Kernel - ICA method. But, the Fast -ICA method give better results than Fast-IVA according to the Source-to-Interference Ratio (SIR) for the first scenario and according to both Source-to-Interference Ratio (SIR) and Source-to-Distortion Ratio (SDR). Getting this result may be because of adding the additive white gaussian noise (awgn) in the second and third scenarios.

The Results of the third experiment on ELSDSR data for speaker recognition system: Speaker recognition system faced difficulties for recognition with mixture

speech signal [25], so that when BSS methods and the recognition system are connected that will improve the performance of the recognition system.

In this thesis we evaluate the performance of BSS algorithms in speaker recognition system. The figures 3.4, 3.5, 3.6 and Table 3.8 show the results of this experiment for speaker Identification and speaker verification. We use (EER) measurement for measuring the speaker recognition system's performance. As we mention before whenever the EER has a lower value, that means the speaker recognition system has better performance. Hence, our results confirm that the Fast-IVA method gives better results than Fast –ICA method.

It should be noted that I-vector gives different EER values for each different run. EER is the intersection point of FAR and FRR. The respective evaluation functions in i-vector toolbox used in this study, these rates are calculated using log likelihood ratios. LLR depends very highly on feature values, and since feature values are obtained using a randomized matrix, the differences may occur. In such non-deterministic procedures, one of the common practices is to run the experiment several times, and take the average. In this study, I-vector is run 10 times and then the average of EER values are used as a result.

For the future work, we will attempt to apply different BSS methods such as the Axul-IVA method, the ICA method in the frequncy domain or the Jade-ICA method etc.

Also, we can test our proposed scenario on the other data set that is well- known. Additionally, we attempt to work on meeting problem by using these BSS methods to separate the speech signals of the meeting and applying the speaker recognition system to recognize who was talking and what he said in a specific period of time.

Moreover, we can apply for this work in real time, so it will be more effective in our real life.

CHAPTER 4

CONCLUSION

Blind source separation (BSS) is one of the most interesting and challenging problems for the researchers in audio and speech processing fields. In this study, we implemented and compared three popular BSS methods, which are Fast-ICA, Kernel-ICA, and Fast-IVA. Three different scenarios were proposed to test the performance of BSS methods extensively. The first scenario includes mixing and separating the clear speech signal. We add the additive wait Gaussians noise to the speech signal before mixing the signal in the second scenario, and adding this type of noise after mixing in the third scenario in order to test the effect of the noise on the BSS methods. We used four different commonly performance metrics (SAR), (SDR), (SIR) and (SNR) to evaluate the performance of the BSS methods. Two data set have been used; the Arabic data set that we calculated by recording the voice messages for 13 students of Yildiz Technical University and the second data is the ELSDSR data which consist of 22 speakers. According to experimental results, Fast-IVA method has high performance than Fast-ICA method and Kernel-ICA method. Additionally, we evaluated the performance of BSS algorithms in speaker recognition system. According to EER values, Fast-IVA again has high performance than Fast-ICA.

REFERENCES

[1] Haykin, S., and Chen, Z. (2005). "The cocktail party problem". Neural computation, 17(9): 1875-1902.

[2] J .Herault, Jutten C, B Ans, "Detection of primitive magnitudes in a Message composite by a neural computing architecture unsupervised learning", in Proc. gretsı 8(Nice, France, 1985): 1017-1022.

[3] Bell, A. J., and Sejnowski, T. J. (1995). "An information-maximization approach to blind separation and blind deconvolution". Neural computation, 7(6): 1129-1159.

[4] Oja, E., and Hyvarinen, A. (1997). "A fast fixed-point algorithm for independent component analysis". Neural computation, 9(7): 1483-1492.

[5] Cardoso, J. F. (1999). "High-order contrasts for independent component analysis. Neural computation", 11(1): 157-192.

[6] Eriksson, J., Karvanen, J., and Koivunen, V. (2000, June). "Source distribution adaptive maximum likelihood estimation of ICA model". In Proceedings of the 2nd International Conference on ICA and BSS (227-232).

[7] Molgedey, L., and Schuster, H. G. (1994). "Separation of a mixture of independent signals using time delayed correlations". Physical review letters, 72(23): 3634.

[8] Bach, F. R., and Jordan, M. I. (2002). "Kernel independent component analysis". Journal of machine learning research, 3(Jul): 1-48.

[9] Pedersen, M. S., Larsen, J., Kjems, U., and Parra, L. C. (2007). "A survey of convolutive blind source separation methods". Multichannel Speech Processing Handbook.

[10] Pedersen, M. S., Larsen, J., Kjems, U., and Parra, L. C. (2008). "Convolutive blind source separation methods". In Springer Handbook of Speech Processing (1065-1094). Springer Berlin Heidelberg.

[11] Comon, P., and Jutten, C. (Eds.). (2010). Handbook of Blind Source Separation: Independent component analysis and applications. Academic press.

[12] Rivet, B., Girin, L., and Jutten, C. (2007). "Mixing audiovisual speech processing and blind source separation for the extraction of speech signals from convolutive mixtures". IEEE transactions on audio, speech, and language processing, 15(1): 96-108.

[13] Mahajan, S. M., and Betrabet, S. K. (2015). “Blind Source Separation using ICA for Additive Mixing in Time and Frequency domain”. International Journal of Scientific Engineering and Technology: 4(4).

[14] Murata, N., Ikeda, S., and Ziehe, A. (2001). An approach to blind source separation based on temporal structure of speech signals. Neurocomputing, 41(1): 1-24.

[15] Parra, L. C., and Alvino, C. V. (2002). Geometric source separation: Merging convolutive source separation with geometric beamforming. IEEE Transactions on Speech and Audio Processing, 10(6): 352-362.

[16] Kim, T., Eltoft, T., and Lee, T. W. (2006, March). “Independent vector analysis: An extension of ICA to multivariate components”. In International Conference on Independent Component Analysis and Signal Separation (165-172). Springer Berlin Heidelberg.

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

[18] Liang, Y., Harris, J., Naqvi, S. M., Chen, G., and Chambers, J. A. (2014). “Independent vector analysis with a generalized multivariate Gaussian source prior for frequency domain blind source separation”. Signal Processing, 105: 175-184.

[19] Furui, S. (2005). 50 years of progress in speech and speaker recognition research. ECTI Transactions on Computer and Information Technology (ECTI-CIT), 1(2): 64-74.

[20] Adikane, A., Moon, M., Dehankar, P., Borkar, S., and Desai, S. Speaker Recognition Using MFCC and GMM with EM.

[21] Koutras, A., Dermatas, E., and Kokkinakis, G. (2000). “Blind separation of speakers in noisy reverberant environments”: A neural network approach. Neural Network World, 10(4): 619-630.

[22] Yen, K. C., and Zhao, Y. (1997, April). “Co-channel speech separation for robust automatic speech recognition: stability and efficiency”. In Acoustics, Speech, and Signal Processing, 1997. ICASSP-97:(859-862). IEEE.

[23] Westner, A., and Bove, V. M. (1999, January). “Blind separation of real world audio signals using overdetermined mixtures”. In Proceedings of ICA 99: 11-15.

[24] Koutras, A., Dermatas, E., and Kokkinakis, G. K. (2001, September). “Improving simultaneous speech recognition in real room environments using overdetermined blind source separation”. In INTERSPEECH. 1009-1012.

[25] Michael U, Martin R, Klaus D. March 25, 2014. “Blind Source Separation for Speaker Recognition Systems”, Technische Universität München, LDV. March 25, 2014.

[26] Li, Y., Amari, S. I., Cichocki, A., Ho, D. W., and Xie, S. (2006). Underdetermined blind source separation based on sparse representation. IEEE Transactions on signal processing, 54(2): 423-437.

[27] Naik, G. R., and Kumar, D. K. (2011). An overview of independent component analysis and its applications. Informatica: An International Journal of Computing and Informatics, 35(1): 63-81.

[28] C. D. Meyer, Matrix Analysis and Applied Linear Algebra. Cambridge, UK, 2000.

[29] Campbell, J. P. (1997). Speaker recognition: A tutorial. Proceedings of the IEEE, 85(9): 1437-1462.

[30] Tiwari, V. (2010). MFCC and its applications in speaker recognition. International journal on emerging technologies, 1(1): 19-22.

[31] TO, A. S. Y. C. A. (2014). voıce ıdentıfıcatıon and recognıtıon system.

[32] Reynolds, D. A., Quatieri, T. F., and Dunn, R. B. (2000). Speaker verification using adapted Gaussian mixture models. Digital signal processing, 10(1-3): 19-41.

[33] Bishop, C. M. (2006). Pattern recognition. Machine Learning, 128: 1-58.

[34] Kanagasundaram, A., Vogt, R., Dean, D. B., Sridharan, S., and Mason, M. W. (2011, August). “I-vector based speaker recognition on short utterances”. In Proceedings of the 12th Annual Conference of the International Speech Communication Association (2341-2344). International Speech Communication Association (ISCA).

[35] Kenny, P., Ouellet, P., Dehak, N., Gupta, V., and Dumouchel, P. (2008). A study of interspeaker variability in speaker verification. IEEE Transactions on Audio, Speech, and Language Processing, 16(5): 980-988.

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

[37] SYRIS Technology Corp 2004. About FAR, FRR and EER

[38] G, Hugo http://research.ics.aalto.fi/ica/fastica/code/dlcode.shtml 1 April 1998.

[39] MSR Identity https://www.microsoft.com/en-us/download/details.aspx?id=52279 17 October 2017

[40] McClaning, Kevin, and Tom Vito. "Radio Receiver Design.(The Book End)." Microwave Journal 45.2 (2002): 188-189.

[41] DeSerio, R. (2008). Savitsky-Golay Filters.

[42] NOVA online, WGBH Science Unit, 1997http://www.pbs.org/wgbh/nova/pyramid/

CURRICULUM VITAE

PERSONAL INFORMATION

Name Surname : Alyaa Abdul Hussain MAHDI

Date of birth and place : 3/7/1983

Foreign Languages : English

E-mail : alyaaamahdi@gmail.com

EDUCATION

Degree	**Department**	**University**	**Date of Graduation**
Master			
Undergraduate	Computer Science	Almustansiriya University	2004-2005
High School	ALKhadraa for girls		2000-2001

WORK EXPERIENCE

Year	**Corporation/Institute**	**Enrollment**
2006	The Ministry of Education, Iraq	employee

PUBLISHMENTS

Papers

1. Mahdi. A, Elbir. A, Karabiber. F “Blind Audio Source Separation Using Independent component analysis and Independent Vector Analysis” International Journal of Applied Mathematics Electronics and Computers (IJAMEC), 4, 174-177, 2016.