BLIND SIGNAL PROCESSING (BSP) OF TWO-INPUT TWO-OUTPUT LINEAR SYSTEM FOR SEPARATING AUDIO SIGNALS USING INDEPENDENT COMPONENT ANALYSIS APPLIED IN NATURAL GRADIENT ALGORITHM

JAMES PAUL CHIBOLE

MASTER OF SCIENCE

(Electrical & Electronic (Telecommunication) Engineering)

JOMO KENYATTA UNIVERSITY OF AGRICULTURE AND TECHNOLOGY

2019

Blind Signal Processing (BSP) of Two-Input Two-Output Linear System for separating Audio Signals using Independent Component Analysis applied in Natural Gradient Algorithm

James Paul Chibole

A thesis submitted in partial fulfillment for the degree of Master of Science in Electrical and Electronic (Telecommunication) Engineering in the Jomo Kenyatta University of Agriculture and Technology

2019

DECLARATION

This thesis is my original work and has not been presented for a degree in any other University.

Signature: ……. Date: ……...…….

James Paul Chibole

This thesis has been submitted for examination with the approval of the University supervisor.

Signature: …….. Date: ……...……..

Prof. Heywood Ouma, Ph.D

UON, Kenya

Signature: ……..… Date: ……...……..

Dr. Edward Ndungu, Ph.D

JKUAT, Kenya

ACKNOWLEGMENT

First, I acknowledge the Almighty God for life and good health He gave me throughout this journey. Second, this work would have not been possible without the great input of my supervisors: Prof. Heywood Ouma Absaloms, and Dr. Edward Ndungu. Their guidance and probing questions in every aspect during my study and work on this project contributed immensely to the success of the thesis. I would also like to appreciate all my lecturers and masters' colleagues in Electrical and Telecommunication departments for their service, encouragement and support throughout this journey.

DEDICATION

I dedicate the thesis to my father for his encouragement to pursue further education, my mother, my wife and my two sons.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

NOTATION

In this thesis:

1. Lowercase Greek letters (α, β, etc) are used for scalars.
2. Lowercase (Roman) letters (a, b, etc) are used for vectors
3. Uppercase (Roman) letters (A, B, etc) are used to denote matrices

Exceptions include the letters i, j, k, l, m and n which are typically used to denote integers. Typically, if a given uppercase letter is used for a matrix, then its corresponding lowercase letter is used for its columns (which can be thought of as vectors), and the corresponding Greek letter for the elements of the matrix. In some cases the elements of a matrix or vector are denoted by roman lower case letter with subscripts such as a_{ij} where i and j represents the position of the element in rows and columns, respectively, within a matrix or a vector.

ABBREVIATIONS

BCE	Blind Channel Equalization
BSP	Blind Signal Processing
BSS	Blind Source Separation
BSS/E	Blind Source Separation and Extraction
CDF	Cumulative Density Function
CMA	Constant Modules Algorithm
EEG	ElectroEncephaloGram
FFT	First Fourier Transform
FAF	Fibonacci Activation Function
FDMA	Frequency Division Multiple Access
FIR	Finite Impulse-response Filter
FSF	Frequency-Selective Fading
HOS	Higher – Order Statistics
i.i.d	Independent and Identically Distributed
ICA	Independent Component Analysis
ISI	Inter-Symbol Interference
JADE	Joint Approximate Diagnalization of Eigen-matrices
jPDF	Joint Probability Density Function
KL	Kullback-Leibler divergence
MBD	Multichannel Blind Deconvolution
MI	Minimum Likelihood
MIMO	Multiple-Input Multiple-Output
ML	Maximum Likelihood
mPDF	Marginal Probability Density Function
MRI	Magnetic Resonance Imaging
MSE	Mean Square Error
NGA	Natural Gradient Algorithm
NS	Nonstationarity
PDF	Probability Density Function

SAF	Sigmoid Activation Function
SOBI	Second-Order Blind Identification
SONS	Second-order Non-stationary source Separation
SOS	Second-Order Statistics
STF	Space-Time Frequency
TDMA	Time Division Multiple Access

ABSTRACT

In Blind Source Separation (BSS) the challenge is to recover the source signals from the observed mixed signals. Blindness means that neither the sources nor the mixing system are known. Separation can be based on the theoretically limiting but practically feasible assumption that the sources are statistically independent. The statistical independence of source signals assumption connects BSS and Independent Component Analysis (ICA). The main aim of this research is to solve the separation problem for source signals and mixing system that are not known by comparing two activation functions. The research uses the Natural Gradient Algorithm (NGA) to separate pairs of sub-Gaussian (music), super-Gaussian (speech) and sub-super-Gaussian mixed signals into their original components using Independent Component Analysis (ICA) assumption of statistical independence of the source signals. Two activation functions are used within the NGA for each of the pairs before separation comparison is made. The NGA is formulated using instantaneous Blind Signal Processing where time delay is not factored in the computation of the independent signals. The design uses a 2 x 2 Multiple Input Multiple Output (MIMO) system to accept the pairs of blind audio signals, mix them and separate them to retain their original form or their filtered version. The Fibonacci activation function and the Sigmoid activation functions are used in iterating the coefficients of the NGA up to a hundrend iterations where convergence is realized. Comparing the output (estimated) to the input signals is by waveforms, frequency spectra, and the measure of the Magnitude-Squared Coherence. The results show that the NGA algorithm with Fibonacci and Sigmoid activation function for speech signals pairs yield high performance when compared to other pairs.

CHAPTER ONE

INTRODUCTION

1.1 Background

In the field of signal processing, multiple signals are often mixed and later separated in various applications. In most of these applications that include multiple biomedical, multiple antennas, multiple microphones, sensors have to be used to get the correct data (Amari & Cichocki, 1998). Such systems are best described using Multiple-Input Multiple-Output (MIMO) system models. In a number of MIMO cases, several signals from the source are present at the same time but there is difficulty in knowing their properties and how they contribute to each sensor output. In cases where source signals and the properties of the system are not known but only the sensor outputs are observed, the methods used in signal processing to recover the original signals are termed as blind.

In Blind Source Separation (BSS) the challenge is to recover the source signals from the observed mixed signals (Lecumberri, Gómez & Carlosena, 2005). Blindness means that neither the sources nor the mixing system are known. Separation can be based on the theoretically limiting but practically feasible assumption that the sources are statistically independent. The statistical independence of source signals assumption connects BSS and Independent Component Analysis (ICA).

The main aim of this research is to solve the separation problem for source signals and mixing system that are not known by comparing two activation functions. There exist several methods to separate sources without knowing the source distributions or even the characteristics of source distributions. The natural gradient algorithm is chosen to model the problem and provide a solution to the ICA assumptions of input-output MIMO system.

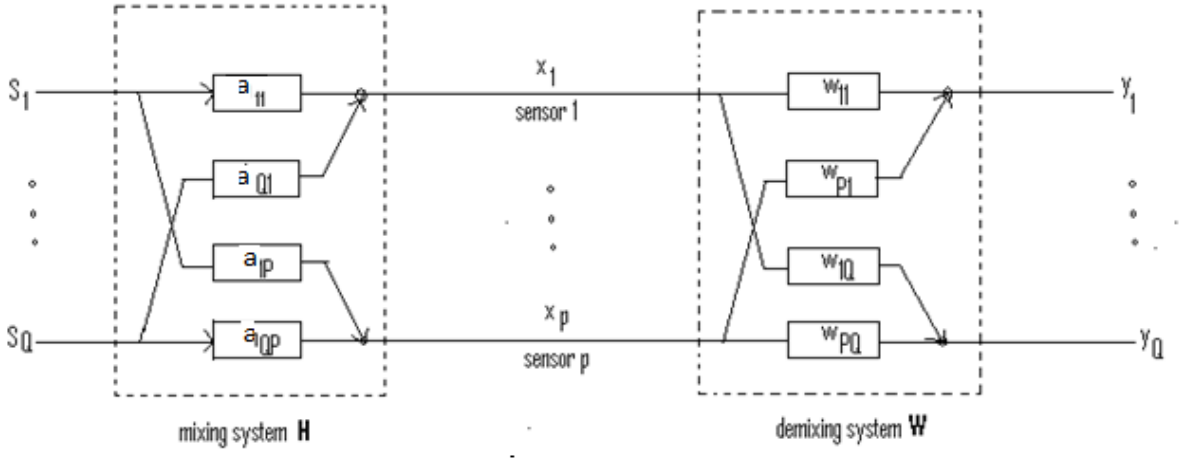


Figure 1.1 General Set up for MIMO Signal Processing System

1.2 Problem Statement

An archtypical problem in engineering and science starts with a vector function f that goes from $\mathbb{R}^n$ to $\mathbb{R}^m$ (Myers, van de Geijn, & van de Geijn, 2015). In other words, in goes one vector, out comes another vector.

$$f: \mathbb{R}^n \rightarrow \mathbb{R}^m \tag{1.1}$$

There are three possible category problems that may be asked regarding the linear transformation by the function depicted in equation (1.1) (Myers, van de Geijn, & van de Geijn, 2015):

1. One action that may be done is to take a vector plug it into$f$ and evaluate it. This is relatively straightforward.
2. A slightly more complex problem is when given a vector $y \in \mathbb{R}^n$, you are asked to figure out what x you can plug into f so that

$$f(x) = y \tag{1.2}$$

In this set of problems, f and y are given and the problem is actually in finding out the value of x.

3. Another very important problem, which is known as an eigen-value problem, is where you are given a function f, and the problem is knowing when

$$f(x) = \lambda x \tag{1.3}$$

where λ is some scalar quantity.

The problems are actually very difficult for general vector function f. However, it turns out that these set of problems can be solved easily if f is a linear transformation. So the linear transformation is a subset of vector functions for which these problems are simpler to solve (Myers, van de Geijn, & van de Geijn, 2015). Moreover, the general f, which is often called a non-linear problem, is often solved by actually making it into a linear function (Myers, van de Geijn, & van de Geijn, 2015). In other words, f function is a transformation and in this thesis it is a matrix.

The thesis' problem is more complex that the three equations – Equations (1.1), (1.2) and (1.3) depicted above. In this research, f and x are not known, only the output *y* is known. The main parameters used to describe the problem are:

s: unknown input signals.

A: unknown mixing matrix

x: is a vector realized after mixing the unknown input signals s with the mixing matrix W: demixing matrix

y: known output signals

f: transformation that does the mixing and demixing processes

Two unknown independent input signals are mixed with an unknown $2\ x\ 2$ mixing matrix. The mixture is passed through an unknown demixing matrix to get back the original signals.

$$\text{s}\rightarrow \underbrace{A = x \rightarrow W}_{f} = y \tag{1.4}$$

Equation (1.4) can be read as s is passed through A to get x, in turn x is passed through W to get y. In an ideal situation, A should be the inverse of W if $s = y$. Unfortunately A and W are unknown neither is s, which falls under blind signal processing (BSS) and the basis of this research problem. It turns out that the problem can be solved if the input signals s are independent of each other, bringing in the concept of Independent Component Analysis (ICA). The ICA requirement for the input signals means that the signals must be non-Gaussian.

Most researches that have taken up the problem of BSS-ICA handle the comparison of different algorithms to know which performs better. Rarely is there a comparison of activation function for the BSS-ICA problem using a single algorithm. In most literature non-Gausianity is treated as a single entity yet in reality there are two forms of non-Gaussian signals; super-Gaussian and sub-Gaussian signals, an aspect that is handled in this research. The problem of this research can be summed up as the use of natural gradient algorithm (NGA) to compare two activations functions, Fibonacci and Sigmoid, in separating a pair of super-Gaussian (speech) signals, sub-Gaussian (music) signals and a mixture of super- sub – Gaussian (speech - music) signals.

1.3 Justification of the Study

Signal separation that utilize few components is an active area of research and continue to receive increased attention (Saduf, 2013). This, coupled with the desire to separate blind signals remains a challenge in engineering and other applicable fields. In an increasing number of applications, the separation of signals using signal properties like frequency is not attainable. Not only are the sources unknown in a number of cases, there are inherent problems associated with signals that have almost common characteristics and where separation is not attainable using the known filtering methods. Independent Component Analysis has proved effective in blind source separation and where the only known characteristic of the signal is that they are non-Gaussian in nature. The Natural Gradient Algorithm is used because its slope of the cost function varies from small changes in the parameters. NGA modifies the slope by taking into considerations the real structure of the optimization space and thus provides

better approximations to the steepest descent direction (Amari & Cichocki, 1998). The choice of a good algorithm is not enough. A good algorithm must be accompanied by a good activation function. While the theory and literature supports the choice of NGA algorithms, it is important to compare the working of this algorithm by various activation functions. The choices of activation functions for this research are the Fibonacci and Sigmoid.

The use of ICA means the sources must be non-Gaussian. Although a number of publications have exhaustively handled this problem, their solutions have only dealt with either sub-Gaussian signals or super-Gaussian signals, with no attempt to offer solutions for a mixture that contains both sub-Gaussian and super-Gaussian signals. As we shall see from the literature, there are indeed those that have attempted to separate a mixture of sub-Gaussian and super-Gaussian mixtures to satisfactory results. However, they have increased another complexity of adapting the activation function first before carrying out the separation itself. The increased computations takes further the ICA from real-time applications. Some publications simply combine various activations functions, such as periodic polynomial, Gaussian and sigmoid (Saduf, 2013). This research intends to use two activation functions with the chosen algorithm to separate super-Gaussian and sub-Gaussian signals.

1.4 Objectives

1.4.1 Main Objective

To design a system for separating two blind audio signals that conform to the ICA assumptions using the Natural Gradient Algorithm.

1.4.2 Specific Objectives

1. To formulate the NGA algorithm for mixing and separating blind audio signals.
2. To determine the effectiveness of the Fibonacci and Sigmoid activation functions for the algorithm on super-Gaussian, sub-Gaussian and super-sub-Gaussian pairs of audio signals.

3. To compare the performance of Fibonacci activation function and the Sigmoid activation function with NGA in separating non-Gaussian signals.

1.5 Overview of the Chapters

Chapter One provides the fundamental principles underlying BSS and an overview of MIMO systems in a broader perspective. Abbreviations, justification of the research, the problem statement, together with research objectives are formulated in this chapter. **Chapter Two** centers on literature review on works done in BSS, ICA, the link between BSS and ICA, formulation of the NGA, super-Gaussian and sub-Gaussian signals characteristics. Further, the chapter looks at the activations functions, and the measure of magnitude squared coherence. **Chapter Three** handles the methodology of the thesis. This chapter gives the systematic process used in formulating the experimental setup and procedures. It further gives the details of the data preparation and the inherent assumption on the input signals. The waveforms and frequency spectra figures of the input and mixed signals, respectively, are displayed in this chapter. In **Chapter Four,** the objective is to output the results and offer a discussion. Three forms of outputs are displayed here and include waveforms, frequency spectra and the measure of the magnitude squared coherence. A discussion on the results is then made. **Chapter Five** gives the results' conclusions, linking the objectives to the outcomes of the research. Limitations, as well as, future research that may be linked to this thesis are then given. The last sections cover **REFERENCES** in IEEE citation style and the **APPENDIX**. Finally, a journal publication from this thesis is:

CHIBOLE, James; HEYWOOD, A.; NDUNGU, Edward. Performance Analysis of the Sigmoid and Fibonacci Activation Functions in NGA Architecture for a Generalized Independent Component Analysis. *IOSR Journal of VLSI and Signal Processing (IOSR- JVSP) Volume 7, Issue 1, Ver. I (Jan. - Feb. 2017), PP 00-00 e-ISSN: 2319 – 4200, p-ISSN No.: 2319 – 4197 www.iosrjournals.org*

CHAPTER TWO

LITERATURE REVIEW

2.1 Background

Blind Signal Processing (BSP) has become one of the exciting and hot topics in the field of signal processing, advanced statistics and neural computation with strong theoretical foundations and with many potential applications. In fact, BSP has developed to become an important topic for development with intensive research in a number of areas including speech enhancing, medical imaging, remote sensing, biomedical engineering, data mining, communication systems, geophysics and exploration seismology (Cichocki & Amari, 2002). The three main areas of BSP are Multichannel Blind Deconvolution (MBD) and Equalization, Independent Component Analysis (ICA), and Blind Signal Separation and Extraction (BSS/E) (Cichocki & Amari, 2002).

2.2 Blind Source Separation

Blind Source Separation is a problem that deals with the separation of blind signals. It is a simple specialty of the general BSP concept. Blind Source Separation (BSS) has received a lot of research attention in communication, neural networks, and signal processing in the recent years. Wireless communication systems have come of age in the use of blind signal processing in combating frequency-selective fading and Inter-Symbol Interference (ISI), Blind channel estimation, blind equalization and blind detections are widely used for communication applications (Lecumberri, Gómez, & Carlosena, 2005). Blind channel estimation can be based on very little or no training information to acquire the channel state information from received signal. On the other hand, blind deconvolution can serve as a means to extract the continuously transmitted signal directly from the received signal. Blind detection can recover the transmitted information symbol without the explicit *a priori* knowledge of the received data. While classical estimation/equalization/detection methods for communication systems

require the transmission of known signals (training sequences), untrained or blind approaches reach beyond this severe limitation and are, therefore, preferred from an efficiency (throughput) point of view. In addition, Independent Component Analysis (ICA), Blind Source Separation, Constant Modules Algorithm (CMA) and Higher – Order Statistics (HOS) are some of the promising tools for the blind signal processing which can be applied for antenna arrays, beam former and MIMO systems (Amari & Cichocki, 1998).

Blind Signal Separation (BSS) is a very promising direction of signal processing. This technique was first applied to the solution of the well-known "cocktail party problem", where the problem is related with the extraction of one human voice from the recorded mixture of voices (Lecumberri, Gómez, & Carlosena, 2005). However, it is worth emphasing that BSS is also often applied in various disciplines, such as medical signal processing, image denoising, face recognition, time series analysis, among others. In this research the focus is on audio signal processing.

2.3 Background of BSS

Suppose that there are N sources $s_{1..N}$ and receivers, which get the mixed combination of the input signals. At first, it is assumed that each mixed signal x_j is the linear combination of the inputs with unknown coefficients A_{ij}:

$$x_j = \sum_{i=1}^{N} A_{ij} s_i \tag{2.1}$$

Further discussion it is considered that the simplest situation for the problem is that formulated by 2 separate sources and 2 mixed outputs MIMO system, so that the problem of equation (2.1) simplifies to equation (2.2):

$$\begin{aligned} x_1 &= a_{11} s_1 + a_{12} s_2 \\ x_2 &= a_{21} s_1 + a_{22} s_2 \end{aligned} \tag{2.2}$$

From equation (2.2), we can see that if the coefficients a are known, then the problem is simplified to solving of linear algebraic equations. However, in our case, these coefficients are unknown. Further, to make matters complex, nothing is known about the input signals. Such unknown signals are called *blind signals*. For example, if there is any information about the spectra of two input signals and these spectra do not overlap, the separation is a trivial task, which may be performed via Fourier analysis. To find a solution to the foregoing problem, it is important to consider the formulation of BSS problem and analyze the main conditions that should hold in order to efficiently separate the two signals. Equation (2.2) can be written in the matrix/vector form

$$x = As \tag{2.3}$$

The mixing scheme is shown in Figure 2.1

$$\begin{bmatrix} s_1 \\ s_2 \end{bmatrix} \rightarrow \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \rightarrow \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Figure 2.1 Mixing of two input signals.

In this problem, the first condition is that the signals are assumed to be statistically independent and mixing is instantaneous, so that the obtained mixtures are the weighted sums of the input signals without any time delays.

The problem of two signals' separation can be formulated as the estimation of the de-mixing matrix. An unknown mixing matrix A is applied to the input signal s to obtain the matrix (actually a vector) of Equation (2.3). The aim is then to find the best estimates of y which are as close as possible to the original signal s. In order to obtain the needed signals s, the de-mixing matrix W is computed.

$$y = Wx \tag{2.4}$$

Theoretically, $Wx = WAs$. What is means is that if A is a transformation from point $\theta_1$to θ_2, then W should be the reverse - a translation from θ_2 to θ_1. In other words,

$W = A^{-1}$ for perfect separation to occur. Since A is unknown in our case, the algorithm to be applied to approximate as closely as possible the inverse matrix A^{-1}. The scheme of the above described principle is shown in Figure 2.2.

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \rightarrow \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix} \rightarrow \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$$

Figure 2.2 Scheme of de-mixing

2.4 BSS Methods and Approaches

Although there are many BSS algorithms, their principle of operation can be classified into four approaches as shown in Figure 2.3.

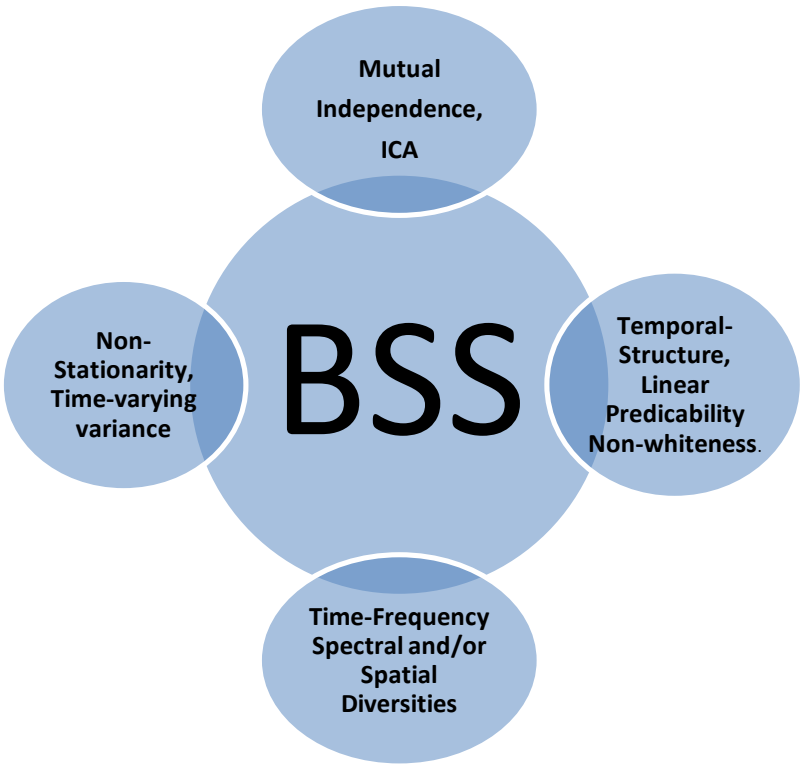


Figure 2.3 Approaches used to solve the BSS problem

The most common approach uses the measure of cost function of signals' independence, sparseness, or non-Gaussinaity. The original source signals are termed as statistically independent and have no temporal structures. Temporal structures are the filtered, delayed and reverberated versions of the individual signals because of multipath effects (Scott, 2002). Such problems are solved using ICA's higher-order statistics (HOS) methods. A common consideration is that not more than one source signal should be Gaussian. Research articles that have utilized these methods include (Matteson & Tsay, 2016) (Lahat, Cardoso, & Messer, 2012) (Lee, Shen, Troung, Lewis, & Huang, 2011).

If the sources contain temporal structures, then it is obvious that all the sources have non-vanishing temporal correlation. This means that much less restrictions than statistical independence methods can be employed to solve the problem. This case is solved using second-order statistics (SOS) methods for estimating the source and the mixing matrix. The SOS methods cannot separate sources signals that have identical power spectra shapes or independent and identically distributed (i.i.d) sources. Some journal articles in this area include (Lee, 2001) (Chan, Ma, Chi, & Wang, 2008).

The third approach uses nonstationarity (NS) properties and SOS. The main concern of the second-order nonstationarity is that the variances of the source signals vary with time. Unlike other approaches, the nonstationarity information based methods make it possible for the separation of colored Gaussian signals that have identical power spectra shapes. However, the approach is not able to separate sources that have identical nonstationarity properties. Some journal articles that make use of signals that are nonstationarity include (de-Frein & Richard, 2011) (Choi, Cichocki, & Beloucharni, 2001) (Hosseini, Daville, & Saylani, 2009).

The fourth approach relies on the various diversities of signals. The diversities include time, frequency ("time coherence" or spectra) and and/or frequency-time diversities, on more generally, joint space-time frequency (STF) diversity. Space-time-frequency concept is commonly used in wireless communication systems. Signals can be separated if they do not overlap in the frequency, time, or the time-frequency domain

(Cichocki & Amari, 2002). When the signals do not overlap in the time domain, it means one signal is silent (stops), before the other begins and are separated using Time Division Multiple Access (TDMA). If the source signals do not overlap in frequency domain, then they can be separated using band-pass filters in a method called Frequency Division Multiple Access (FDMA). Research articles in this category include (Barros & Cichocki, 2001) (Zhang, Cai, & Ding, 2014) (Moon & Hong, 2014). The meaning of temporal structures and their importance is well outlined in Moon and Hong (2014).

2.5 BSS Models

Three applications of BSS include array processing in wireless communication, signal enhancement in medicine and speech separation in acoustics (Scott, 2002). In the present wireless networks, a number of wireless devices attempt to communicate through a base station. The base station uses multiple antennas for receiving and directing the various users' signals within the spectral bandwidth of the communication channel. When the device's transmitted signals overlap in frequency and time, FDMA and TDMA respectively, separation is no longer possible. BSS can be applied to the antenna measurements to separate and enhance the various transmitted signals without knowing the devices' positions relative to the base station and without knowing the exact forms of the transmitted signals (Paulraj & Papadias, 1997). In the second application, a number of noninvasive medical technologies such as MRI and ElectroEncephaloGram (EEG) are able to characterize body processes through multichannel recording (Scott, 2002). Because of the complex propagation properties of human tissues, these multichannel recordings are often difficult to understand. However, through BSS, there is potential of extraction of identifiable and coherent signal features that can be more easily tired to specific body ailments or functions (McKeown, et al., 1998). In speech separation, speech signals collected by distant microphones in room environments can be difficult to understand especially if there are multiple conversations (Scott, 2002). However, BSS provides a hand in

separating individual speech signals from the microphone's signals, making them more listenable and intelligible (Torkkola, 2000).

The above three applications of BSS differ in the model of mixing generated by the measurement process. Two types of mixing models in BSS are possible: (1) instantaneous or spatial mixing and (2) convolutive or spatio-temporal mixing.

2.5.1 Instantaneous BSS Model

In instantaneous or spatial mixing, the mixtures are weighted sum of the individual source signals without dispersion or time delay. The example on narrowband array processing in wireless networks involves spatial mixing conditions (Scott, 2002). Figure 2.6 shows the structure of BSS for instantaneously mixed sources.

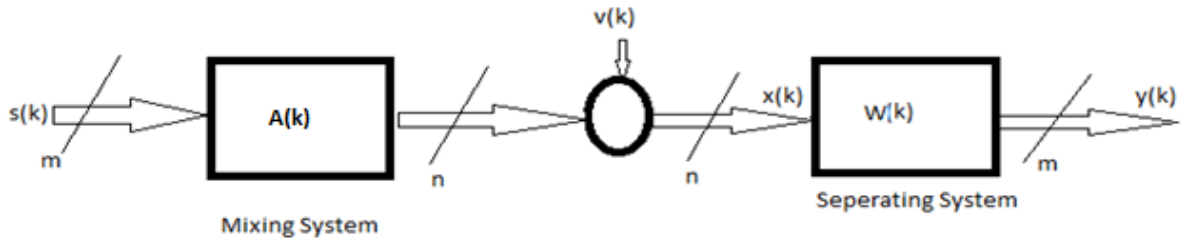


Figure 2.4 Instantaneous BSS task

At the left is the unknown source signal vector at time k given by

$$\boldsymbol{s}(k) = [s_1(k)\ s_2(k)\ \dots\ s_m(k)]^T \tag{2.5}$$

where $s_i(k)$ is the *i*th source signal. The m source signals are linearly mixed by $(n\ x\ m)$ unknown mixing matrix A with entries a_{ij}, yielding the n-dimensional measured or observed signal vector $x(k)$ as

$$\boldsymbol{x}(k) = As(k) + v(k) \tag{2.6}$$

where $v(k)$ is an *n*-dimensional noise vector sequence that is unrelated to the source signal sequence $\boldsymbol{s}(k)$ (Scott, 2002). In typical applications, the source signal contains

useful but unknown information and the mixing model describes the undesirable smoothing and propagating effect inherent on some physical measurement process (Scott, 2002). For this discussion, the assumption that $\boldsymbol{v}(k)$ is made up of uncorrelated and jointly Gaussian-distributed elements which are independent of the elements of $\boldsymbol{s}(k)$, a common assumption in many signal processing formulations (Scott, 2002). In this case, $m = n$, the number of independent sources is equals the number of measurement sensors. In practical applications, however, it is unreasonable to assume that the number of sources is always equal to the number of sensors. In array processing for wireless communications, for example, sources can "come and go" depending on the current network state and transmitter use (Scott, 2002).

Application of instantaneous mixtures model include the study of brain science. BSS plays a great role in the identification of underlying components of brain activity from recording of brain activity as given by an EEG (Cichocki & Amari, 2002). In image processing, BSS instantaneous mixture model is used in quality improvement and extraction of independent features in images. Music and speech signals can also be separated using instantaneous BSS so long as the signals are not in real time and time delay is not a factor of consideration. BSS using the instantaneous mixtures of *m* binary antipodal sources, which are linearly combined by an unknown system, is used in (Kofidisa, Margarisa, Diamantarasb, & Roumeliotis, 2008). Unfortunately, BSS instantaneous mixtures cannot handle the delay aspects of the signals, often occurring in real-time situations.

2.5.2 Convolutive BSS Model

In convolutive or spatio-temporal mixing, the mixtures contain filtered, delayed and reverberated versions of the individual signals because of multipath effects. The multi-microphone speech separation involves spatio-temporal (Scott, 2002). The instantaneous mixture can be extended by putting into consideration the time delay that results from sound propagation over space and in some instances, the multipath generated by reflected sound off different objects, mostly in large rooms and enclosed

settings. The convolutive mixing system is implemented in finite impulse response filters.

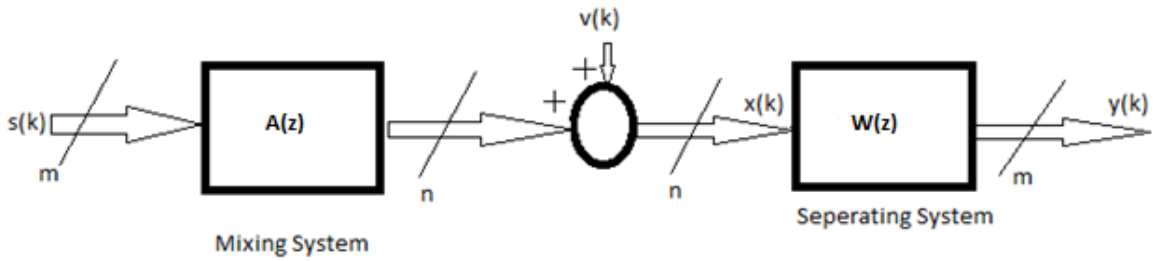


Figure 2.5 Convolutive BSS task

Figure 2.5 shows the structure of blind deconvolution task. An unknown signal given by $s(k)$ passes through an unknown linear time-invariant filter with impulse response a_i, $-\infty < i < \infty$. The resulting noisy received signal $x(k)$ is given by

$$x(k) = \sum_{i=\infty}^{\infty} a_i s(k-i) + v(k) \tag{2.7}$$

where $v(k)$ is a zero-mean Gaussian random process that is independent of the source signal $s(k)$. In typical applications, the source signal contains useful but unknown information and the unknown filter describes the smoothing effect of the physical system. The measurement noise $v(k)$ models any sensor and channel noise inherent in the measurement process.

The goal of blind deconvolution task is to extract an estimate of the source signal sequence $s(k)$ from the measured sequence $x(k)$ using a linear filter of the form

$$y(k) = \sum_{l=0}^{L} w_l(k) x(k-l) \tag{2.8}$$

where $w(k)$ for $0 \leq l \leq L$ are the coefficients of the system and L is a filter length parameter. In this case, the assumption of a causal finite-impulse-response (FIR) filter

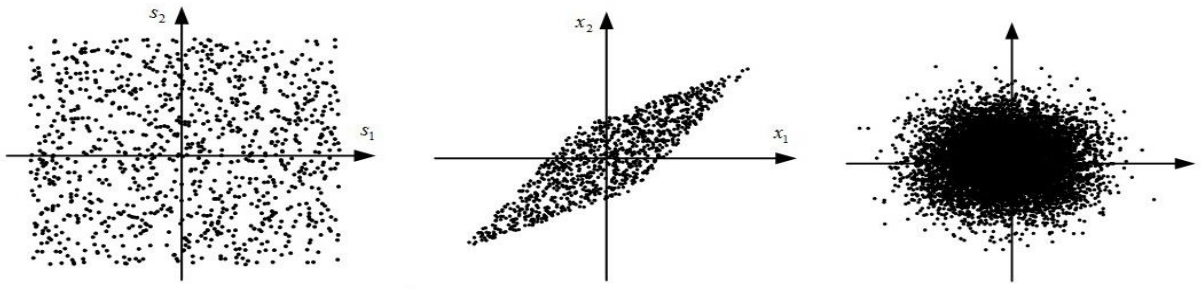
is made for the deconvolution model. FIR models are best suited for adaptive filters because they are bounded-input-bounded-output stable for bounded coefficients and are computationally simple (Scott, 2002). The above problem formulation is for a single channel blind deconvolution and can easily be extended to multi-channel blind deconvolution. Non-Gaussianity and convolution BSS model is used in the measure of blind image deconvolution of ICA using genetic algorithm with robust results realized (Hujun & Hussain, 2008).

Interestingly, some authors in their publications such as Scott (Scott, 2002) and Douglas and Haykin (2000), synonymously use the words blind signal separation to mean instantaneous blind separation. Comparing Figure 2.4 and Figure 2.5 show that blind signal separation (instantaneous models) and blind deconvolution entail similar tasks. Both revolve around blindly estimating the inverse of the linear system from the measurement of the system's output. It means that blind signal separation methods can be converted to handle blind deconvolution problems (Matteson & Tsay, 2016). The technique is beyond the scope of this research and will not be considered. However, the technique for performing this conversion can be found in (Douglas & Haykin, 2000). From here henceforth, the thesis will dwell in handling instantaneous model techniques and algorithms.

2.6 Independent Component Analysis

Independent component analysis (ICA) is a set of methods that use the assumption of the statistical independence of the input signals. In simple words, two variables y_1, y_2 are statistically independent when the value of y_1 does not depend on y_2 and vice-versa. An example of joint PDF from two uniform distributions is shown in Figure 2.6 (a) (Aapo & Erkki, 2000). This joint PDF is uniformly distributed to form a square, which follows the basic definition of the independency. The joint PDF after application of the mixing matrix A with elements $a_{11} = 2, a_{12} = 3, a_{21} = 4, a_{22} = 2$, in Figure 2.6 (b) (Aapo & Erkki, 2000). The matrix A is

$$A = \begin{bmatrix} 2 & 3 \\ 4 & 2 \end{bmatrix} \tag{2.9}$$



(a) (b) (c)

Figure 2.6 Joint density probabilities

Figure 2.6(a) shows the densities before mixing for uniform distributions, Figure 2.6(b), shows the densities after mixing for uniform distribution, while Figure 2.6(c) (Aapo & Erkki, 2000) shows a normal distribution before mixing. It can be seen from Figure 2.6(b) that the values of x_2 and x_1 depend on each other. The goal of the ICA is to attain the independence, which means the transformation from joint PDF on Figure 2.6(b) to joint PDF on Figure 2.6(a). We can see that such transformation can be observed as scaling and rotation of joint distribution.

2.6.1 Why Gaussian variables are forbidden

The fundamental requirement in ICA is that the independent components must be non-Gaussian. To understand why Gaussian signals do not work with ICA is to assume that input signals $\boldsymbol{s_i}$ are Gaussian and the mixing matrix is orthogonal. Then $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$ are equally Gaussian based on the central limit theorem, they are also of unit variance and are uncorrelated. The joint distribution of these mixed Gaussian variables is given by:

$$p(x_1, x_2) = \frac{1}{2\pi} e^{\frac{x_1^2 + x_2^2}{2}} \tag{2.10}$$

The simulated Figure 2.6(c) shows that the joint density has a symmetric shape, revealing that ICA does not hold for Gaussian signals. Mathematically, as Equation 2.10 reveals, the joint PDF of multivariate normal distribution is circular, thus the rotation of the joint PDF cannot be estimated. Meaning that it does not contain any information on the direction of the columns of the mixing matrix **A**. It is the reason why Gaussian signals do not work based on the fact that it is not possible to estimate **A**. In addition, the distribution of any orthogonal transformation of the Gaussian $(\boldsymbol{x_1},\boldsymbol{x_2})$ has the same distribution as $(\boldsymbol{x_1},\boldsymbol{x_2})$. Therefore, for Gaussian variables, it is only possible to estimate the model of ICA up to an orthogonal transform. Hence, the ICA model cannot estimate Gaussian independent components (Tharwat, 2018).

2.6.2 Preprocessing: Centering and whitening (sphering)

Before the application of the ICA some preprocessing steps need to be performed to ease the separation. These steps include centering and whitening (sphering) of the data. Such data manipulations make two mixed signals uncorrelated, that is, the covariance matrix is made diagonal. In addition, the whitening step includes the normalization of the variances to one. This explains why this step is called "whitening", because the covariance matrix of white noise is diagonal with unit variances. In most cases, the mixture to be separated contains a combination of the input signals with unit variance (Vrins & Verleysen, 2005).

Centering and whitening are described by Equations (2.11) and (2.12) respectively

$$x_i = x_i - E(x_i) \tag{2.11}$$

$$X = XE(XX^T)^{-1/2} \tag{2.12}$$

The preprocessing steps simplify the ICA problem to the estimation of a single parameter –rotation angle of the joint PDF. The joint PDF of the uniform distributed data before and after preprocessing is shown in Figure 2.7 (Aapo & Erkki, 2000).

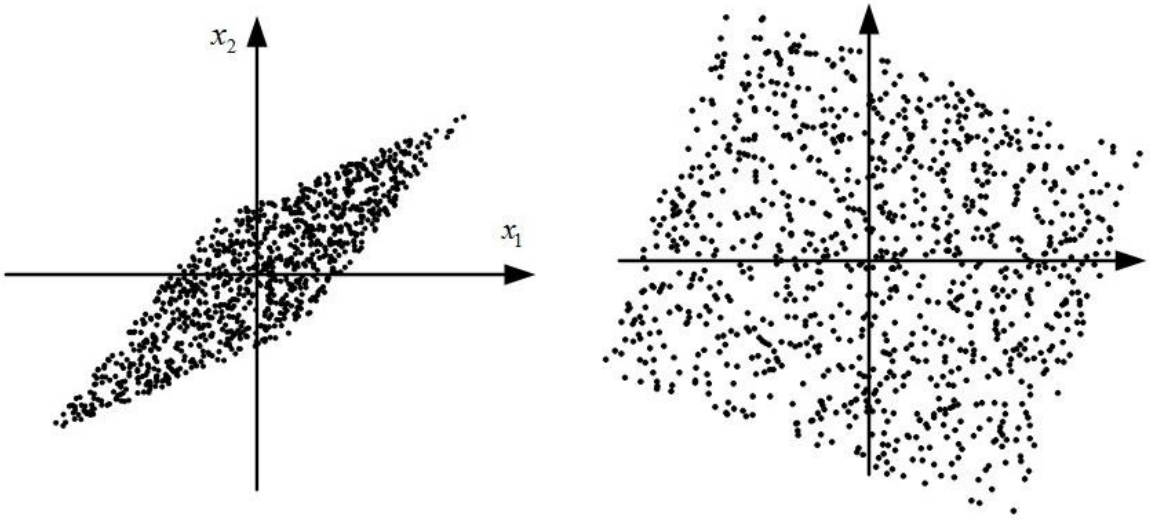


Figure 2.7 Illustration of centering and whitening steps

The aim of ICA techniques is to estimate the unknown rotation angle of the joint distribution. There are several criteria used for such estimation. Two main criteria, which are often used independently, are minimization of the mutual information and maximization on non-Gaussianity. Both general groups of methods use some objective function and evaluate its values that are to the extreme, which corresponds to the minimization of the mutual information or maximization of non-Gaussianity respectively, which are enough conditions for independence to be realised.

ICA has two important components; the objective function and the optimization algorithm. The objective function is responsible for determining the statistical independency of y. The robustness and effectiveness of ICA depends on the selection of the objective function. The purpose of the optimization algorithm is to control the objective function until it attains stability. The speed of convergence and stability of ICA depends on the algorithm.

2.6.3 ICA Assumptions

One of the advantages of ICA over other approaches in signal separation is its requirement of relatively few assumptions on the mixing matrix and the source signals.

Assumption I: *The source signals are statistically independent*

This assumption is considered critical and fundamental to ICA. Statistical independence is an important consideration that makes it possible for the estimation of independent components $\hat{s}_j(t)$ from the observed mixtures $x_i(t)$.

Assumption II: *The independent components have non-Gaussian distribution*

This is another important assumption because of the connection between independence and Gaussianity. It is not possible to separate Gaussian source signals using the framework of ICA because the sum of variables that are Gaussian is itself Gaussian (see Equation 2.10, and Figure 2.6(c), above). It means that the sum of signals that are Gaussian is the same as that of a single Gaussian in the framework of ICA. It is for this reason that source signals that are Gaussian cannot be separated using ICA and its algorithms.

Assumption III: *The mixing matrix is invertible*

If the mixing matrix is not invertible then clearly the de-mixing matrix to be estimated is not even available. It is only after the three assumptions are satisfied that the independent components can be estimated. It is important to note that the three assumptions are not restrictive and, therefore, extremely little information about the mixing matrix and the source signals is required.

2.6.4 ICA Ambiguity

There are two major ICA ambiguities: (i) magnitude and scaling and (ii) permutation.

Magnitude and Scaling ambiguity : The correct variance of independent components cannot be determined (Tharwat, 2018). This can be illustrated mathematically as:

$$x = \sum_{j=1}^{N} a_j s_j \tag{2.13}$$

where a_j represents the jth column of the mixing matrix A. Both the coefficients a_j of mixing matrix and independent components s_j are unknown. Equation (2.13) can be transformed into:

$$x = \sum_{j=1}^{N} (1/\propto_j s_j)\,(\propto_j s_j) \tag{2.14}$$

In most applications, however, this ambiguity is insignificant. The known means to overcome this ambiguity has been the use of the assumptions that each of the sources has unit variance. Further still, the signs of the sources can also not be determined.

Permutation ambiguity: The order of the estimated independent components is unspecified. During the rotation, the update of the demixing matrix is done iteratively, therefore the source signals are extracted, but in no specific order (Tharwat, 2018). By introducing a permutation matrix P and its inverse into the mixing process of equation (2.1), we get:

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

$$= A's' \tag{2.15}$$

The elements of Ps are actually the original sources, only that they follow a totally different order and $A' = AP^{-1}$ is another unknown mixing matrix. In ICA, Equation (2.15) is the same as Equation (2.1) or its vector form Equation (2.3) and only conveys the information that permutation is a common feature of BSS. In separating signals, however, the aim is not always to have restrictions on the order of separated signals, but to simply separate them. This makes the permutation of source signals irrelevant.

2.7 Comparison of BSS and ICA

In a number of literatures, the terms ICA and BSS are interchanged bringing confusion. It is true that ICA and BSS refer to the same models and their problems are solved by almost the same algorithms with the assumption of the mutual independence of the source signals holding. However, from the operational point of view, the aim of BSS is quite different from that of ICA. The purpose of BSS is to estimate the original source signals even if the assumption of mutually independence of the source signals is not completely attainable. On the other hand, ICA's key purpose is to determine a transformation and the assumptions of the mutually independent source signals hold all the time.

While ICA methods depend on high-order statistics (HOS) for many of their applications, BSS methods use only the second-order statistics (SOS). Higher order methods assumptions is on their mutual independence, while second order methods assumptions is on the sources having some form of temporal structure (Cichocki & Amari, 2002). A key observation to note is that second order statistics methods cannot perform independent component analysis tasks. Another difference is that higher-order statistics methods cannot work with signals that are Gaussian, while second order methods have no problem with Gaussian signals. BSS research using Gaussian signals have been used in a number of applications. A model using Gaussian signals has been used to yield robust and satisfactory results on high-resolution satellite data (Jalobeanu, Feraud, & Zerubia, 2004). The same results cannot be realized using ICA because the mutual independence is not observed by Gaussian signals. Therefore, BSS methods do not in any way replace ICA and vice versa, for the simple fact that each approach uses different assumptions, criteria and have different objectives (Cichocki & Amari, 2002).

2.7.1 Instantaneous BSS and ICA

Here we expand the single equation model of instantaneous model of section 2.5.1, by looking at it from the multiple signal points of view and which helps in connecting

BSS to ICA. This is important because the research deals with multiple input and output signals – two to be specific.

Assume that the source signals are stationary zero-mean processes and are mutually statistically independent. Let $s(k) = (s_1(k), \ldots\ s_m(k)\ )^T$ be the unknown independent source vector and $x(k) = (x_1(k), \ldots\ x_n(k)\ )^T$ be a sensor vector, which is a linear instantaneous mixture source by

$$x(k) = As(k) + \cdots v(k) \tag{2.16}$$

where $A \in R^{nxm}$is unknown mixing matrix of full rank and $v(k)$ is the vector of Gaussian noises (Zhang, Cichocki, & Amari, 2004). The BSS problem is to recover original source signals from observations $\boldsymbol{x}(k)$ without prior knowledge on the source signals and the mixing matrix. The demixing model used here is a linear transformation of the form

$$y(k) = Wx(k) \tag{2.17}$$

where $\boldsymbol{y}(k) = (y_1(k), \ldots\ y_m(k)\ )^T$, $W \in R^{nxn}$ is a demixing matrix to be determined during training. The assumption of $m = n$ is maintained that the source signals equals the sensor signals. The general solution to the BSS problem is to find a matrix W such that

$$WA = \Lambda P \tag{2.18}$$

where $\Lambda = \begin{bmatrix} \mathrm{A_o} \\ 0 \end{bmatrix}$, $\mathrm{A_o}$ is a diagonal matrix and P is a permutation (Zhang, Cichocki, & Amari, 2004). The algorithm applied trains the de-mixing model W such as m components are designed to recover m source signals.

The aim of BSS is to adapt the de-mixing model such that its output signals are mutually independent. Remember that ICA main operational point is the independence of signal providing a link between BSS and ICA. Two things are unknown in this

problem: one is the number of active sources and, two, is the probability density functions (pdfs) in the BSS framework. The traditional approach is to estimate the number of active sources before training the de-mixing matrix model, which may fail if the sensor signals are noisy or the source signals are very weak (Zhang, Cichocki, & Amari, 2004). This research does not intent to estimate the number of active sources before training the de-mixing matrix. The estimation of pdfs is computationally demanding and its convergence is extremely slow for ordinary gradient- descent methods (Zhang, Cichocki, & Amari, 2004). This research uses the natural gradient algorithm that shows improved results. The pdfs are not directly estimated in natural gradient algorithm (NGA), which is used to train the parameters of the activation function and which shorten the convergence rate. The pdf determines whether the signal is sub-Gaussian, Gaussian or super-Gaussian for appropriate activation function to be used.

This independence of the components is measured using criteria such as smoothness, sparseness, linear predictability or by an information-theoretic cost function such as the Kullback-Leibler (KL) divergence (Cichocki & Amari, 2002). The optimal generated weight then corresponds to the statistical independence of the output signals. The KL divergence plays the all-important role in estimating the pdfs of the signals in NGA.

2.8 ICA Algorithms

ICA has many methods that use both Second order and Higher order statistics for separating sources. The most common ICA algorithms are FastICA, Second Order Blind Identification (SOBI), Joint Approximate Diagnalization of Eigen-matrices (JADE) and the Second Order Non-stationary source Separation (SONS) (Naik, 2012). These algorithms have been successfully applied in ICA for various applications in different fields of data mining mostly in audio signal processing (Dubnov, Tabrikian, & Targan, 2006). However, in most of the applications of these ICA algorithms, the application has been on signals that are super-Gaussian in nature. This thesis introduces the NGA and analyzes its working on both sub- and super-Gaussian signals.

Unlike the other ICA algorithms mentioned, NGA can work for both instantaneous and convolutive models. However, in this research handles and derives the NGA applicable in spatial models only.

2.8.1 Natural Gradient Algorithm

One disadvantage of stochastic gradient optimization methods is that they suffer from slow convergence because of the inherent statistical correlation found in most real-world signals in the parameter updates. This traditional natural gradient algorithm adopts fixed-step-size; the choice of step size directly affects the convergence speed and steady state performance (Peng & Yang, 2010). Gauss-Newton methods can be used to counter the performance drawbacks of these schemes (Amari, Douglas, Cichocki, & Howard, 1997), but they are expensive when it comes to computation and they often suffer from numerical problems if their implementation is poorly done. It is important to have an algorithm that is not only simply and robust of the stochastic gradient but which achieves convergence performance independent of any statistical dependencies (Amari, Douglas, Cichocki, & Howard, 1997). To elaborate and as we shall see, the NGA does not necessarily depend on the mixing matrix, other than the initial conditions.

The non-Gaussianity of the sources is because of certain identifiability conditions that need to be satisfied for any BSS-ICA formulation to work properly. BSS methods that use this formulation depend on some knowledge of the lower-order or higher-order amplitude statistics of the source signal to perform the separation (Scott, 2002). Algorithms for BSS-ICA that use spatial independence are classified into two: (i) those that use density matching of the sources, and (ii) those that use contrast function optimization. This thesis explores the Natural Gradient Algorithm that uses density matching of the sources.

2.8.2 Density Matching BSS-ICA using Natural Gradient Adaptation

Density matching BSS-ICA methods rely heavily on the concept of information theory. Information theory has applications in a number of fields including economics, communication, physics and neuroscience (Cover & Thomas, 1991). In BSS and ICA information theory is useful because it helps in characterising the amount of shared information in a set of signals. In fact, the concept of information theory is the basis for the use of the word "blindness" in signal processing. The concept makes it possible to separate signals when no common information can be found between any two-output signal sets.

The natural gradient algorithm (NGA) is an improvement on the shortcoming of the stochastic gradient algorithm (SGA), which demands the specifying of the initial condition. In both SGA and the modified NGA, the aim is to adjust the coefficient of the demixing matrix. Adjusting $W(k)$ means that the joint Probability Density Function (jPDF) of the separated signal $y(k)$ is made as close as possible to the assumed distribution $\bar{p}_y(y)$ (Scott, 2002) in each iteration of the algorithm. The measure has been successfully done by Cardoso using the divergence measure of Kullback-Leibler (Cardoso, 1998).

$$KL(p_y \parallel \bar{p}_y) = \int p_y(y) \log\left(\frac{p_y(y)}{\bar{p}_y(y)}\right) \mathrm{dy} \tag{2.19}$$

where $p_y(y)$ is the actual distribution and $\bar{p}_y(y)$ is the assumed distribution of the estimated signal vector. In an ideal situation $p_y = \bar{p}_y$ giving a KL of zero. In fact, Equation 2.19 can correctly be referred to as the objective function (Chibole, 2014).

The tricky part is in choosing $\bar{p}_y(y)$, which makes Equation (2.19) unreliable in many applications. A cost function that is instantaneous and which has the same expected value as that of the $KL(p_y \parallel \bar{p}_y)$ is given by

$$\check{J}(W) = -log\left(\prod_{i=1}^{m} p_s(y_i(k))\,|detW|\right) \tag{2.20}$$

where $detW$ indicates the determinant W. The cost function (Equation 2.20) can be computed further to yield the stochastic gradient descent given as

$$W(k+1) = W(k) + \mu(k)[W^{-T}(k) - \varphi(y(k)x^T(k))] \tag{2.21}$$

here $\varphi(y) = [\varphi(y_1), \dots, \varphi(y_m)]^T$, $\varphi(y) = -\delta log p_s(y/dy)$, is the cdf or algorithmically the activation function. $\mu(k)$ is the step size used by the algorithm to move from one iteration to the next. According to Scott (2002), Equation (2.21) is able to perform Blind Signal Processing (BSS) for any simple choice of $\varphi(y)$. The Stochastic gradient descent of Equation (2.21) is limited in terms of application because of its slow convergence. A modification to the stochastic gradient descent to remove the mixing matrix A's ill-condition is the natural gradient algorithm by Amari (Amari S. , 1998) or the relative gradient by Gardoso (Cardoso, 1998). In NGA, the initial condition is not put into consideration because the algorithm aims at the overall system matrix $C(k)$. The full derivation of the NGA is found in (Scott, 2002) with its final equation given as:

$$C(k+1) = C(k) + \mu(k)[I - \varphi(C(k)s(k))s^T(k)C^T(k)]C(k) \tag{2.22}$$

From Equation (2.22) it is evident that the mixing matrix A does not play any role in the separation process as it only used in the initial condition $\mathrm{C}(0) = \mathrm{W}(0)\mathrm{A}$. Therefore even if the mixing matrix is poorly chosen, it will not affect the quality of separated signal. The NGA works even better if the inputs $\mathrm{s}(\mathrm{k})$ adhere to the ICA's independence requirement. Despite the modifications, both SGA and NGA require a good choice of the activation function $\varphi(.)$ as shown in Equation (2.22).

2.9 Activation Functions

Input signals are often non-linear in nature. However, during the mixing process, such inputs are combined linearly. In fact, inputs to algorithms are simply a linear transformation of the $\boldsymbol{x}$ vector of mixed signals. $\boldsymbol{x}$ on the other hand is combination of the $\boldsymbol{A}$ mixing matrix and $\boldsymbol{s}$ the vector of the input signals. To retain back the original signals s, a reverse transformation through a de-mixing matrix $\boldsymbol{W}$ is necessary but often not sufficient. The de-mixing process will have to be guided by an activation function even if the inputs are linear. Because the input signals are non-linear in most cases, for $\boldsymbol{y}$ to resemble $\boldsymbol{s}$, the transformation process will have to be passed through a non-linear activation function. Otherwise, it is not possible to retain the non-linearity of the input signals. *Non-linearity* means that it is not possible to reproduce the output when the inputs are combined linearly.

While the input takes values in the range $[-\infty, \infty]$, the output controlled by the activation function gives bounded values within the intervals [1, 0] or [-1, 1]. Without an activation function, it is not possible to map the wide inputs to relatively small output parameters. One essential characteristic of an activation function is that it must be differentiable within its bounded interval. Differentiation helps to give the direction of adjusting the weights. When the derivative value is significant, it means the corresponding weights will equally have large weight adjustments. Calculus rules are that a significant derivative value indicates the minima is still far. For each iteration, the weights of the algorithm are adjusted to correspond to the direction of the steepest descent on the surface of the cost function defined by the total error. Computing the error is by subtracting the expected from the observed values. Then each weight within the matrices of weights is adjusted according to the gradient error calculated. In a more general sense, the derivation is done along the activation function curve as the expected value using optimization techniques such as the natural gradient in finding the minima of the objective function.

The first derivative determines the first point on the curve by ensuring that there is a tangent with a slope of zero on the line tangent. A slope of zero suggests the location

of the *minima* and it can be local or global for the function. However, the first derivative also suggests something significant: it informs the algorithm if it is headed in the correct direction that will take it to the minimum of the function. The derivative value, that is, the slope at that point decreases gradually. What this means is that for a minimized function, its derivative must be calculated and that the value must be decreasing if the algorithm is following the correct direction. It is for this reason that the activation function must be differentiable within its range, revealing it critical role in algorithms. The better the choice of the activation function, the better the algorithm.

2.9.1 Sigmoid and Fibonacci Activation Functions

The research will compare the commonly used activation function - the Sigmoid - to the less explored in the literature - the Fibonacci. In fact, the SAF is so common that there is an Artificial Neural Network (ANN) named after it, the Sigmoid Neurones. Sigmoid neurones are modified version of the perceptions where small changes in the bias and weights are reflected as small variations in the output (Nielsen, 2015). Equation (2.23) shows the FAF

$$\varphi(y) = \frac{\sqrt{5}-1}{2} y^3 + \frac{3-\sqrt{5}}{2} y^5 \tag{2.23}$$

while the SAF has the Equation (2.24)

$$\varphi(y) = \frac{1}{1+ e^{-y}} \tag{2.24}$$

Activation functions are simply the cumulative density functions (cdf) and they have a relation to Gaussian functions as explained below. Assuming there is a random variable s, having probability density $P_s(S)$. The cumulative density function (cdf) is given as:

$$F(s) = P(S \leq s) \tag{2.25}$$

Suppose, this is the Gaussian density then

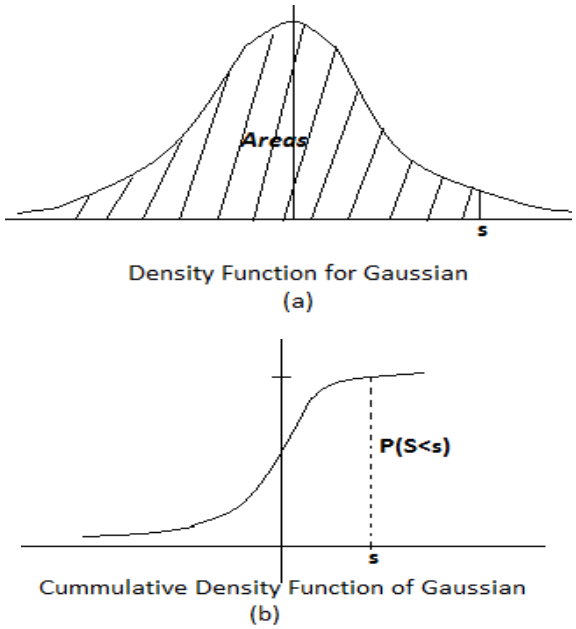


Figure 2.8 Density function (a) and its cdf (b)

The height of the function s (Figure 2.8 (b)) is equal to the area of the Gaussian density (Figure 2.8 (a)), up to the period s. Going further (Figure 2.8 (b)) the function (Figure 2.8 (a)) becomes Gaussian. Equation 2.26 is another form of writing the cdf of Equation 2.25.

$$F(s) = \int_{-\infty}^{s} P_s(t)dt \tag{2.26}$$

Suppose there is a random variable s and the aim is to model the distribution of this variable. Two option exist: the first option is by specifying the density P_s(s) or the second option is to specify the cdf $F(S)$. Equation (2.26) is used to relate the two options. Most literature prefers the cdf option because of the easy in the calculation and because it is continuous over the specified space. Continuous here means it is differentiable over its range. In signal processing and other applications, the cdf is truly referred to as the activation function. It is, therefore, easy to recovery the density P_s(s) by taking the cdf and computing its derivative

$$P_s(S) = F'(S) \tag{2.27}$$

There is always a step in algorithms when the random variable for s is assumed by either specifying the density P_s or the cdf. In the real sense, the assumptions involve the choice of an activation function that can model the output of the signals to be separated. The cdf or the activation function has to be some function increasing from 0 to 1 or some other specified range.

The ideal form for $\varphi(y)$ is the commulative density function (cdf) of the distribution of the independent sources. In practice, however, if $\varphi(y)$ is a sigmoid function, the learning rule reduces to that proposed in (Bell & Sejnowski, 1995). The algorithm is then limited to separating sources with super-Gaussian distribution. For a mixture of sub- super-Gaussian mixtures, Zhang, Cichoki and Amari (2004) and Zhang, Amari, and Cichocki (2001) proposed the idea of AFs adaptations, in steady of estimating the pdfs of the sources. This is motivated by the idea that there is need for the exact score functions for successful separation to be achieved by ICA. However, the method greatly slows down the convergence process of the algorithm. Super-Gaussianity and Sub-Gaussianity is handled next.

2.10 Super- and Sub-Gaussian Signals

There are two types of non-Gaussian signals. These are super-Gaussian and sub-Gaussian or “planty kurtotic” and “lepto kurtotic” respectively (Naik, 2012).

2.10.1 Sub-Gaussian Sources

For sub-Gaussian density, a symmetrical form is adopted as follows

$$p(x) = \frac{1}{2}\left(N(\overline{x}, \sigma^2) + N(-\overline{x}, \sigma^2)\right) \tag{2.28}$$

where $N(\overline{x}, \sigma^2)$ is the normal density with mean $\overline{x}$ and variance σ^2 (Naik, 2012)

$$N(\overline{x}, \sigma^2) = p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\overline{x})^2}{2\sigma^2}} \tag{2.29}$$

Note that the normal distribution is associated with the Gaussian distribution. Signals with sub-Gaussian PDF have a wide distributed function as illustrated in Figure 2.9 (Naik, 2012).

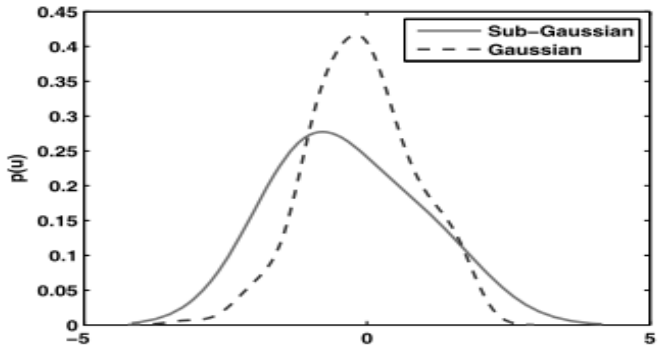


Figure 2.9 PDF of a sub-Gaussian signal

A saw-tooth signal, music signal and white noise signal are typically sub-Gaussian sources (Cristescu, Ristaniemi, Joutsensalo, & Karhunem, 2000). The sub-Gaussian signals have PDF that are not as peaky when compared to Gaussian signals.

2.10.2 Super-Gaussian Sources

For a super-Gaussian density of speech signal, the Laplacian density is used and represented as:

$$p(x) = \frac{1}{\sigma\sqrt{2}} e^{-\frac{\sqrt{2}|x-\bar{x}|}{\sigma}} \tag{2.30}$$

The values of PDF for a super-Gaussian signal are clustered around zero. A speech signal is a typical example for a super-Gaussian source (Naik, 2012). Figure 2.10 is a representation of a typical super-Gaussian (speech) signal (Naik, 2012).

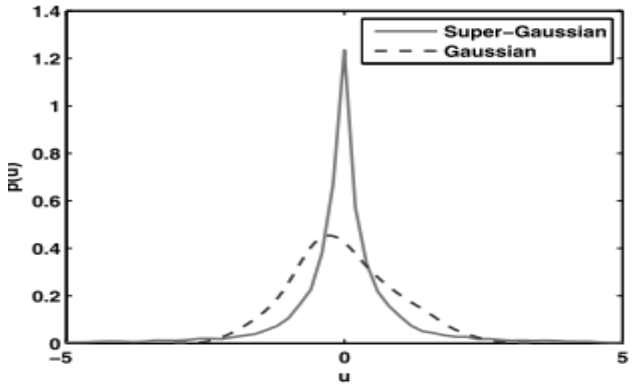


Figure 2.10 PDF of a super-Gaussian (speech) signal

From Figure 2.10, it is also clear that the super-Gaussian signals contain PDFs that are peakier than that of Gaussian signals. Kurtosis and Entropy are two measures used to calculate whether the signal is super-Gaussian or sub-Gaussian. Intensive research has shown that speech signals are super-Gaussian. Research dedicated to testing whether the non-Gaussian signals are super- or sub-Gaussian can be found at (Naik, 2012) (Rai & Yogesh, 2004).

2.11 Frequency Spectra of Signals

Fourier analysis has application for the synthesis and reproduction of sound. Most musical instruments contain the fundamental and subsequent harmonics, as superposition of pure waves. Fourier analysis is the decomposition of musical instrument's sound into its individual components used to form it (Daqarta for DOS, n.d.). Sound waves can best be characterized into amplitudes of the component sine-sound waves summed to produce it. It is this set of numbers that give information regarding the harmonic content of sound produced and it is often termed as the harmonic spectrum of such a sound wave. The quality or timbre of a musical note is determined by its harmonic content.

After gathering the harmonic content of a musical sound note from Fourier analysis, it is possible to synthesize the specific sound from a number of pure tone producers by

simply changing their phases and amplitudes and then summing them together. This process is the Fourier synthesis. Fourier transform can be easily used to show that a square wave has harmonics that are odd numbered with decreasing values as amplitudes. The nth harmonic has $^1/_n$ times the fundamental amplitude. This is shown in Figure 2.11 (Daqarta for DOS, n.d.).

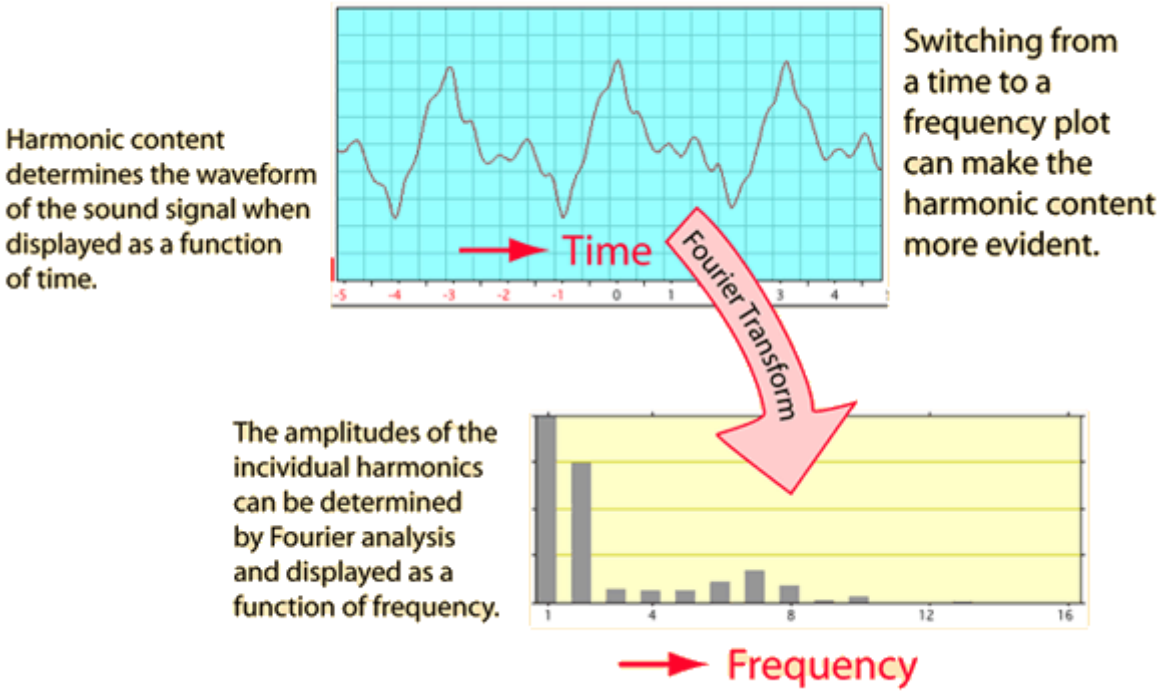


Figure 2.11 Fourier Transform

2.12 Measure of Signal Separation Quality

In most ICA methods, the quality of source separation is measured using the well-known traditional methods that include Signal to Noise Ratio (SNR), Signal to Distortion Ratio (SDR), Signal to Artifacts Ratio (SAR) and Signal to Interference Ratio (SIR) (Naik, 2012). In a number of audio applications and bio signals, SIR is the most popular measure of the quality of separation (Cichocki & Amari, 2002). Signal to Interference Ratio is the ratio of the power of the wanted signal to the total residue power of the unwanted signal (Naik, 2012). However, the mentioned methods to

measure the separation quality gives a general summary figure of the separation. Each audio signal is made up of several frequencies. It is important to know at which frequency is the separation effective and at which it is not. The thesis seeks an alternative method that measures the separation of signal at each frequency. The solution is the measure of the signal coherence at each frequency. The method gives the output that spans the entire waveform revealing where there is complete separation, partial separation or no separation. Further investigation can be done to understand why the separation is not achieved at specific frequencies, a realization that cannot be achieved using the traditional methods mentioned in (Naik, 2012). This research introduces the magnitude-squared coherence in MATLAB in comparing the estimated signals to their corresponding original input signals after the separation is complete.

2.12.1 Magnitude-Squared Coherence

The magnitude-squared coherence between two signals $s(n)$ and $y(n)$ is measured using Welch's averaged modified periodogram method. According to Mathsworks (2014), the magnitude-squared coherence estimate is a function of the frequency and the quotient output is a real value between 0 and 1, which indicate how close y resembles s at each frequency ω. The magnitude-squared coherence is a function of the power spectral densities $P_{ss}(\omega)$ and $P_{yy}(\omega)$, of s and y, and the cross power spectral density $P_{sy}(\omega)$ of s and y. The magnitude-squared coherence equation is shown in Equation (2.31).

$$C_{sy}\omega = \frac{|P_{sy}(\omega)|^2}{P_{ss}(\omega)P_{yy}(\omega)} \tag{2.31}$$

where s and y must be of the same length (Mathsworks, 2014). For real s and y, the MATLAB function for measuring magnitude-squared coherence, *mscohere* returns a one-sided coherence estimate. For complex s and y, it returns a two-sided estimate. An example of a MATLAB generated magnitude-squared coherence between an input signal s and and its corresponding output y is shown in Figure 2.11 (Mathsworks, 2014):

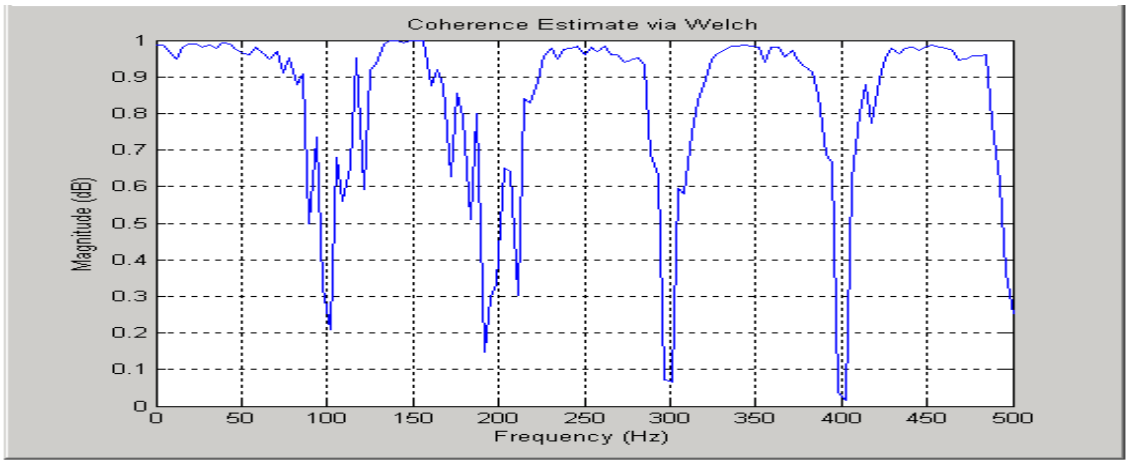


Figure 2.12 Measure of magnitude-squared coherence

Full coherence is realized when the same signal is used for both inputs results in a straight line graph revealing that the signals are similar.

CHAPTER THREE

METHODOLOGY

3.1 Background

The problem of separating audio mixtures with varying time series is predominant in many applications of signal processing; a common example is the well-known cocktail party problem. In the cocktail party problem, the objective is to recover the speech signals of multiple speakers who are talking simultaneously in the room (Figure 3.1). In a normal situation, the room is reverberant because of the reflection on the walls of the signals. What this means is that the original source signals in the separation problem are filtered by an input and output system before the sensors pick them up. In the problem of Blind Source Separation (BSS), the interest is in finding the corresponding demixing system. Two speakers are used as sensors in the simulated experiment.

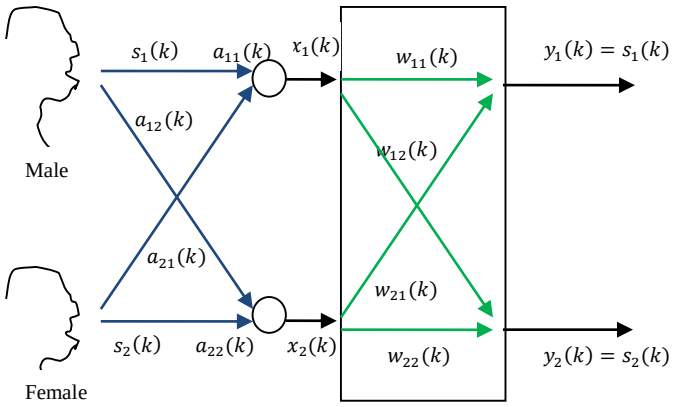


Figure 3.1 A 2 by 2 system used for mixing and separating a pair of signals

3.2 Method Employed

In practice, the experiment is set up to reflect what is depicted in Figure 3.1. The experimental setup tries to emulate the famous "the cocktail party problem" discussed

in (Douglas & Gupta, 2007). The research implemented the setup with some modifications. First, the input signals are recorded and stored in the computer system, instead of generating them in real time. Second, time delay and echoes, as would happen in cocktail party procedures are not factored, therefore, there is no filtering done on the input signals before receiving them by the sensors. This is deviation from the real life situation of a reverberant room; however, the results obtained prove that the NGA improves the output signals when applied to ICA, and can be extended to real experiment without loss of their effectiveness. The procedure involve the use of music and human signals with the intention of understanding how effective the algorithm works with the two activation functions - Sigmoid and Fibonacci.

There are two key assumptions made on the input signals. The assumptions are confirmed on the input signals giving way for them to be mixed using the ICA-NGA algorithm in MATLAB. The mixture is then preprocessed before separation is done. The experiment is carried out to evaluate the performance of NGA on two pairs of sub-Gaussian, super-Gaussian and sub-super-Gaussian signals. The procedure entails recording and storing of waveforms, frequency spectra and sounds.

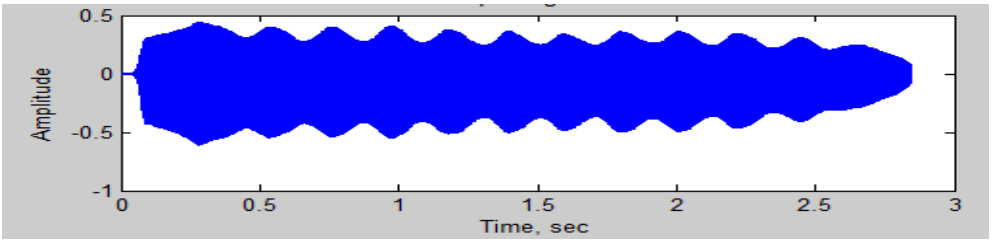
3.3 The Input Stage

The input signals used in the research are sub-Gaussian and super-Gaussian signals. The sub-Gaussian signals are all music instrument of a saxophone and violin sounds. For this purpose, the music (sub-Gaussian) signals were downloaded and stored as .wav files. On the other hand, the speech (super-Gaussian) are human speeches recorded using the internal microphones of a computer. The first speech recordings were a male voice speaking the words *I study MATLAB*. The second recording is of a 12-year-old female speaking the words *I am Junior*. The distance between the speaker and the computer microphone was kept close enough to prevent multi-paths for each emitted source signal. The recordings used in this research are summarized in Table 3.1.

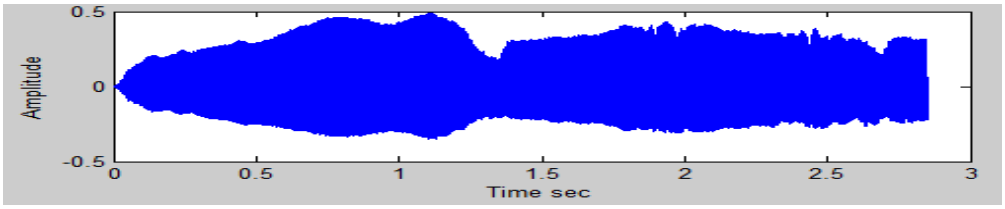
Table 3.1 The input signals used in the experiment.

Non-Gaussian Signal	Audio Signal
Sub-Gaussian	Saxophone
	Violin-2
Super-Gaussian	Male Speech
	Female Speech

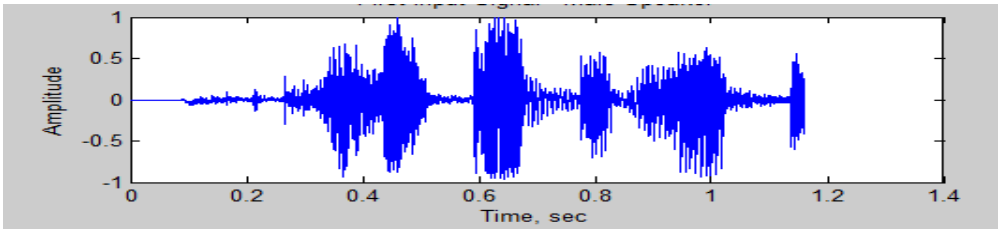
The input signals are then displayed in MATLAB as seen in Figure 3.2.



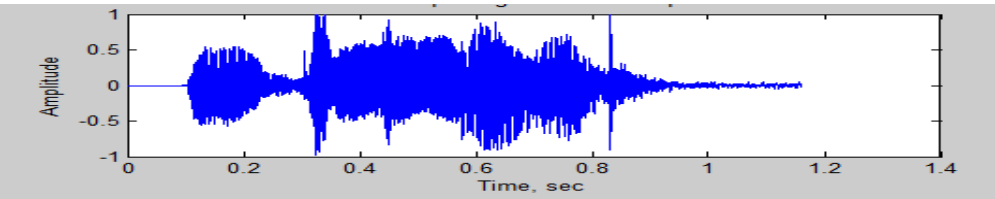
Saxophone



Violin-2



Male Speech



Female Speech

Figure 3.2 Waveforms of the input signals

Notice that from Figure 3.2 the input signals are of different lengths and amplitudes. Before the input signals shown in Figure 3.2 were used, they were equated to the shortest length in each pair to enable mixing and later separation. The amplitudes in each pair were also normalized. Table 3.2 shows the pairing done.

Table 3.2 The input signals in their pairs.

	Input-1	**Input-2**
Pair-1	Saxophone	Violin-2
Pair-2	Male	Female
Pair-3	Female	Violin-2

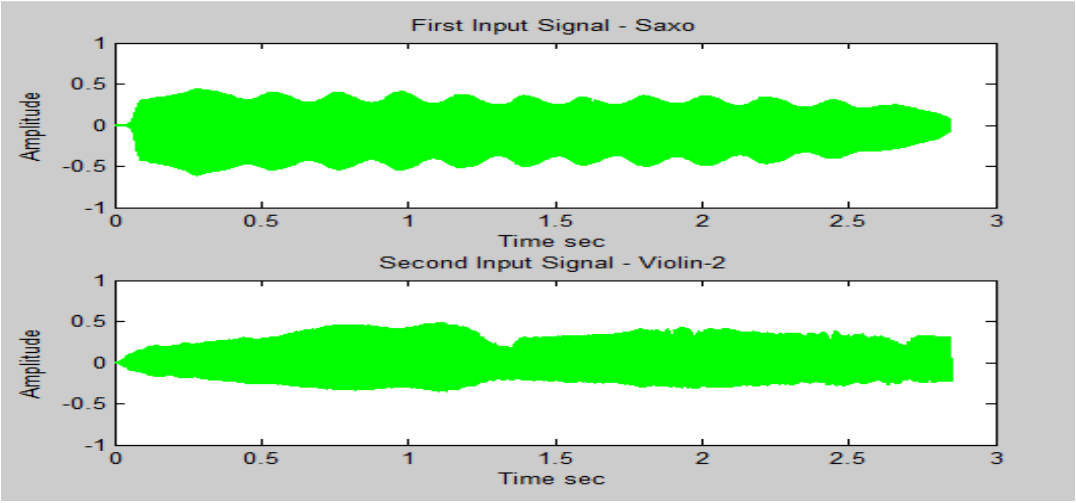
Table 3.3 is information generated in MATLAB showing the characteristics of each of the input signals before pairing and after pairing and equating.

Table 3.3 The input signals in their pairs before and after equating

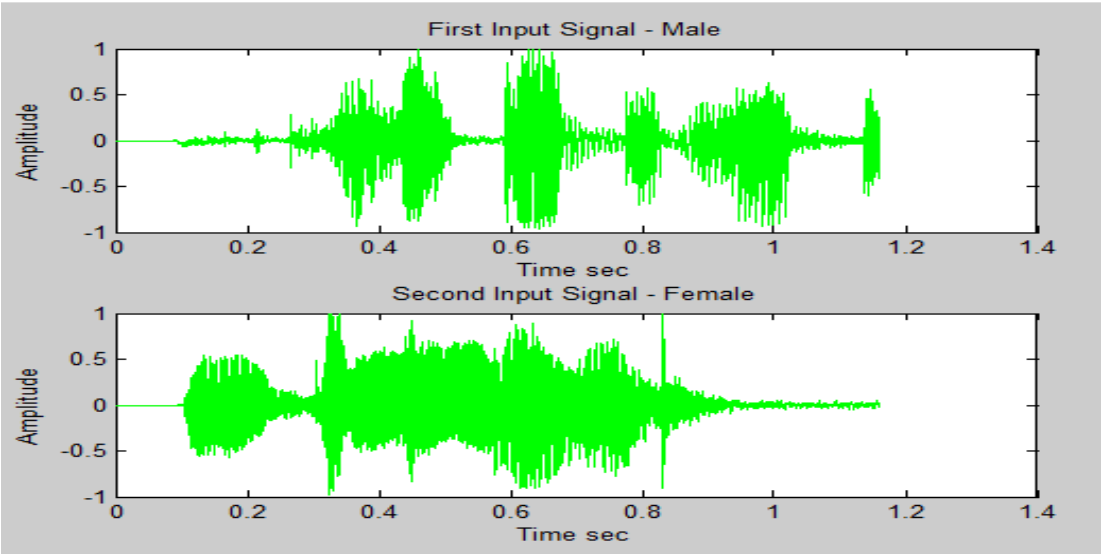
	Before Pairing	**After Pairing and Equating**
Pair-1	Information on the Saxophone sound file before equating	Information on the Saxophone sound file after equating
	Duration = 125632 in samples	Duration = 125632 in samples
	Duration = 2.8488 in seconds	Duration = 2.8488 in seconds
	Bit resolution = 16 bits/sample	Bit resolution = 16 bits/sample
	Sampling rate = 44100 Samples/second	Sampling rate = 44100 Samples/second
	Information on the Violin-2 sound file before equating	Information on the Violin-2 sound file after equating
	Duration = 295470 samples	Duration = 125632 samples
	Duration = 6.7 in seconds	Duration = 2.8488 in seconds
	Bit resolution = 16 bits/sample	Bit resolution = 16 bits/sample
	Sampling rate = 44100 Samples/second	Sampling rate = 44100 Samples/second
Pair-2	Information on the Male sound file before equating	Information on the Male sound file after equating
	Duration = 77824 in samples	Duration = 51200 in samples
	Duration = 1.76472 in seconds	Duration = 1.161 in seconds
	Bit resolution = 16 bits/sample	Bit resolution = 16 bits/sample
	Sampling rate = 44100 Samples/second	Sampling rate = 44100 Samples/second

	Information on the Female sound file before equating	Information on the Female sound file after equating
	"Duration = 51200 samples	"Duration = 51200 samples
	Duration = 1.161 in seconds	Duration = 1.161 in seconds
	Bit resolution = 16 bits/sample	Bit resolution = 16 bits/sample
	Sampling rate = 44100 Samples/second	Sampling rate = 44100 Samples/second
Pair-3	Information on the Female sound file before equating	Information on the Female sound file after equating
	Duration = 51200 in samples	Duration = 51200 in samples
	Duration = 1.161 in seconds	Duration = 1.161 in seconds
	Bit resolution = 16 bits/sample	Bit resolution = 16 bits/sample
	Sampling rate = 44100 Samples/second	Sampling rate = 44100 Samples/second
	Information on the Violin-2 sound file before equating	Information on the Violin-2 sound file after equating
	Duration = 295470 samples	Duration = 51200 samples
	Duration = 6.7 in seconds	Duration = 1.161 in seconds
	Bit resolution = 16 bits/sample	Bit resolution = 16 bits/sample
	Sampling rate = 44100 Samples/second	Sampling rate = 44100 Samples/second

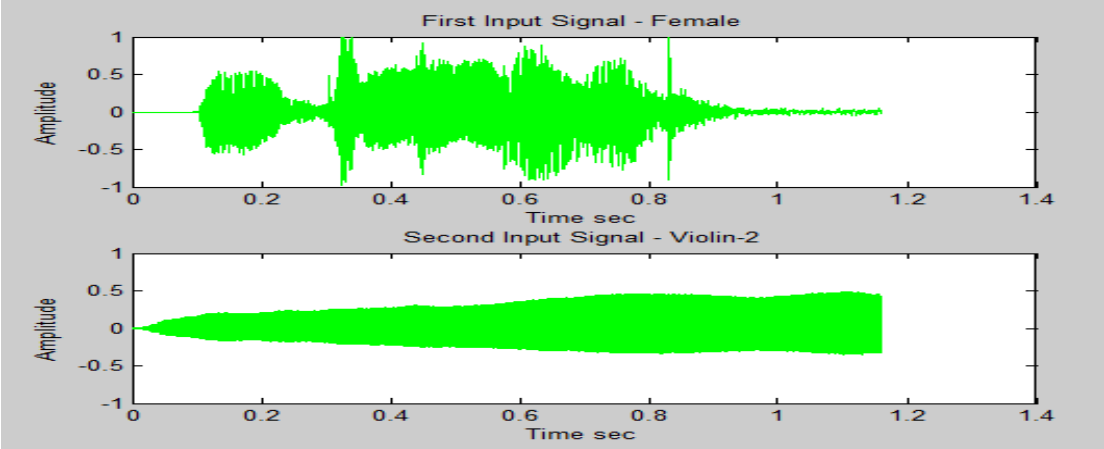
The paired inputs are represented as s_1 and s_2 and their waveforms are shown in Figure 3.3.



(a) Pair-I



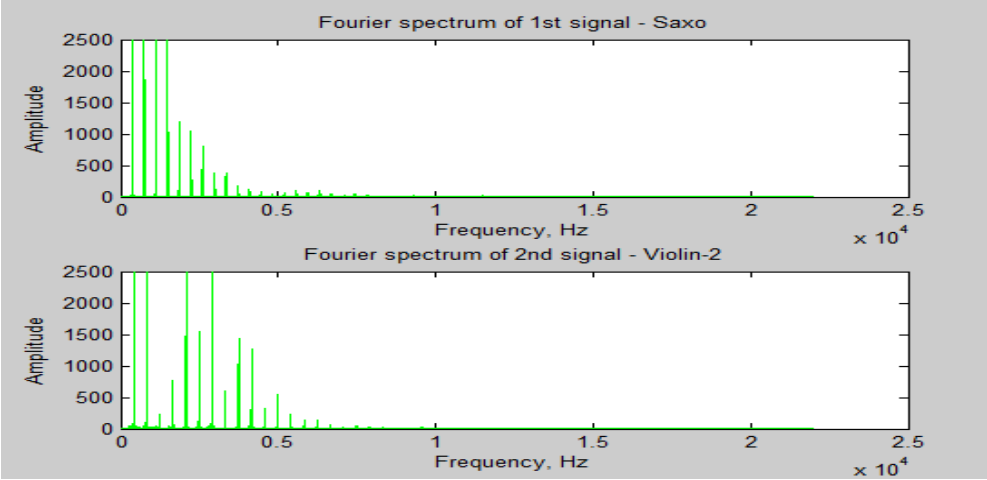
(b) Pair-II



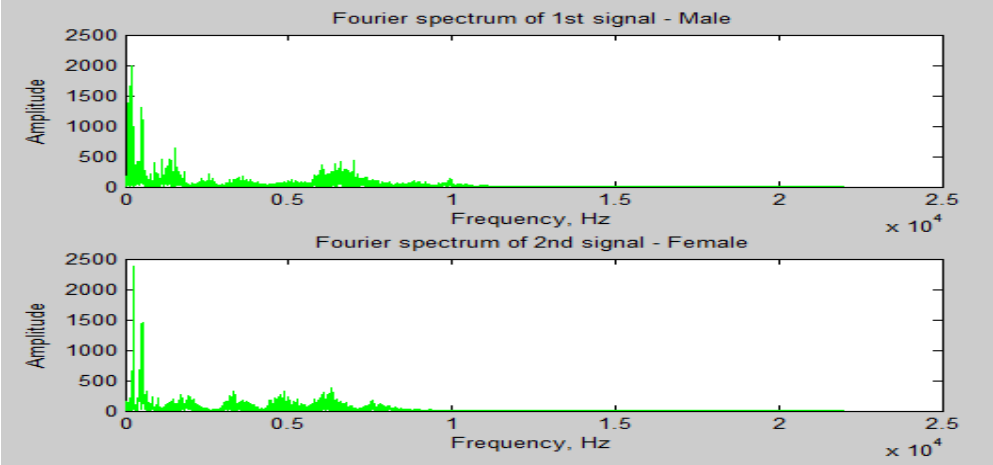
(c) Pair-III

Figure 3.3 Waveforms of input signals

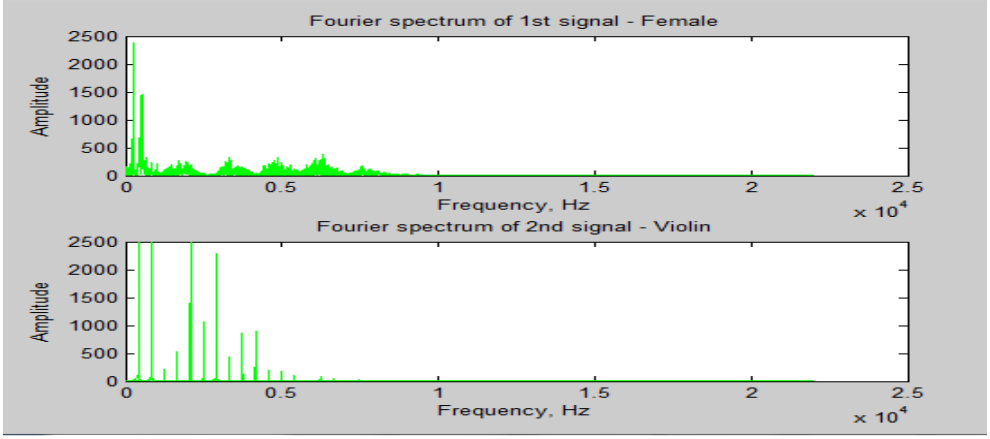
The frequency spectra of the input signals for each pair were generated and shown in Figure 3.4.



(a) Pair-I



(b) Pair-II



(c) Pair-III

Figure 3.4 Frequency spectra of the input signals

3.3.1 Signal Preparation at the Input Stage

In order to perform efficient separation of two source signals, it is important that some preparations are performed on the input signals. These include checking to confirm that the two source signals are independent and they exhibit non-Gaussianity characteristics. These conditions on the input sources must be fulfilled for the signal to be separated using any ICA algorithm.

i. Independence

Statistical independence is a fundamental concept in the separation of signals using ICA algorithms, including NGA. Consider two unique random signals as s_1 and s_2. For the separation to work well, the random signal s_1 must be independent of s_2. Independence here means that any information on s_1 cannot be used to get information on s_2 and vice versa. In this case, s_1 and s_2 are assumed to be random variables originating from two unique physical processes, which have no relationship to each other.

Many researchers have exhaustively dealt with the independence of signals and the literature review has also shown that the input signals are independent. Therefore, the methodology of this research will not concern itself with proving the independence of the input signals. A number of publications on speech signals and independence include (Naik, 2012) (Kumar & Naik, 2011) (Aapo & Erkki, 2000) (Parameswaran, Finitha, & Sama, 2010). On the same note, publications done on music instruments to show that they are independent exist at (Naik, 2012) (Kumar & Naik, 2011) (Aapo & Erkki, 2000).

ii. Non Gaussianity

A second and equally important condition to be fulfilled by the input signals for them to work with ICA is that they must be non-Gaussian. The literature review in *Chapter Two* has exhaustively looked at this ICA condition. Again, this section will not endeavor itself to prove that the input signals fulfill this condition as part of its experimental procedure. From the existing theory, the input signals used in this research have overwhelming literature to support that they exhibit the condition of non-Gaussianity. Publications and research on non-Gaussianity of speech signals include (Naik, 2012) (Kumar & Naik, 2011) (Aapo & Erkki, 2000) (Parameswaran, Finitha, & Sama, 2010). While those on music signals include (Naik, 2012) (Cristescu, Ristaniemi, Joutsensalo, & Karhunem, 2000) and (Aapo & Erkki, 2000). The research

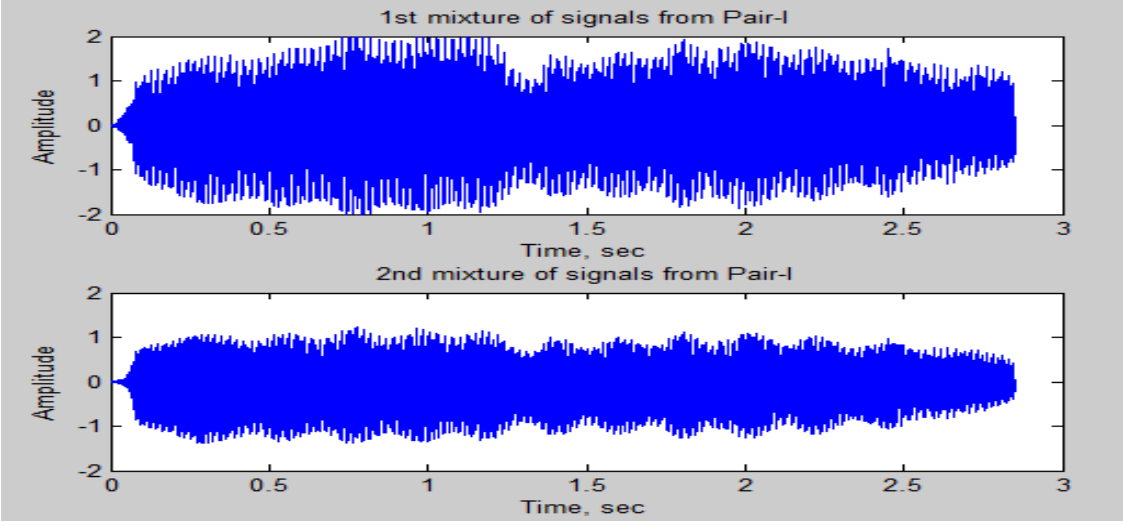
will also not concern itself in proving that all the input signals exhibit non-Gaussianity but the requirement is assumed.

3.4 The Mixing Stage

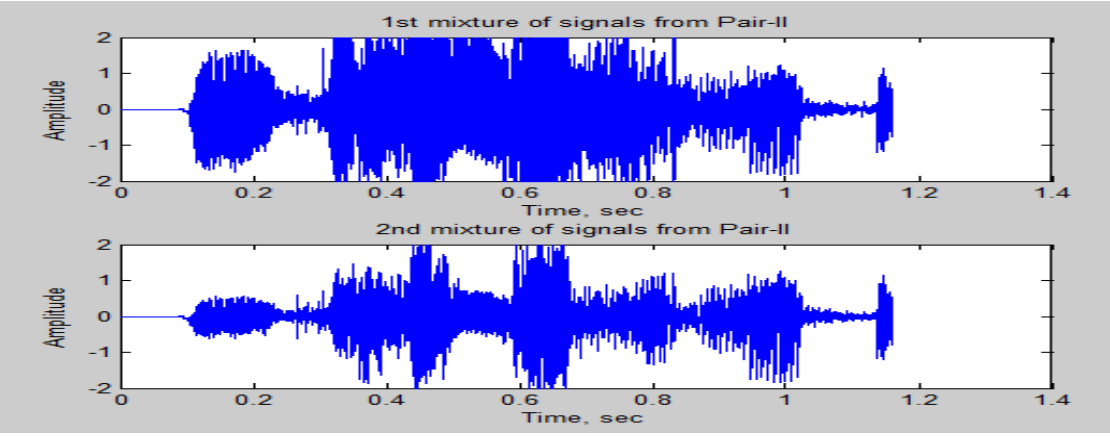
The input signal are mixed by applying the "unknown" mixing matrix in each case. The mixing matrix A is:

$$A = \begin{bmatrix} 2 & 3 \\ 2 & 1 \end{bmatrix} \tag{3.1}$$

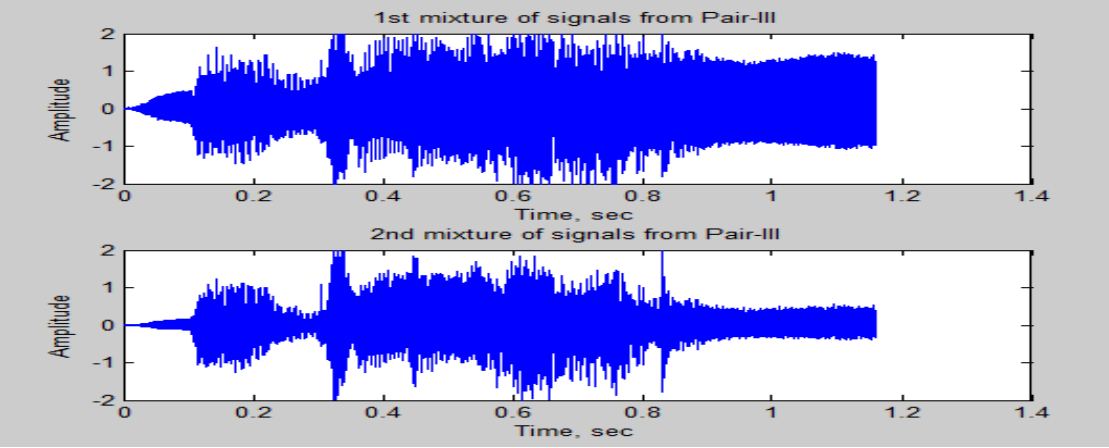
The input mixing matrix of equation (3.1) is same as used in (Aapo & Erkki, 2000). Since there are two input signals, the mixing process is designed to use sensors to take up the two signals. Each of the pairs of the input signal is fed into the input as shown Figure 3.1. The outputs of the mixed signals are represented as $x_1$and x_2. $x_1$and x_2 are dependent on each other unlike $s_1$and s_2 which are independent (*see* Appendix I). The outputs are then stored in three formats – the waveform, frequency spectrum and sound. The mixed signal waveforms and frequency spectra are displayed in Figure 3.5 and Figure 3.6, respectively. The sound from these waveforms are stored as .wav files.



(a) Pair-I



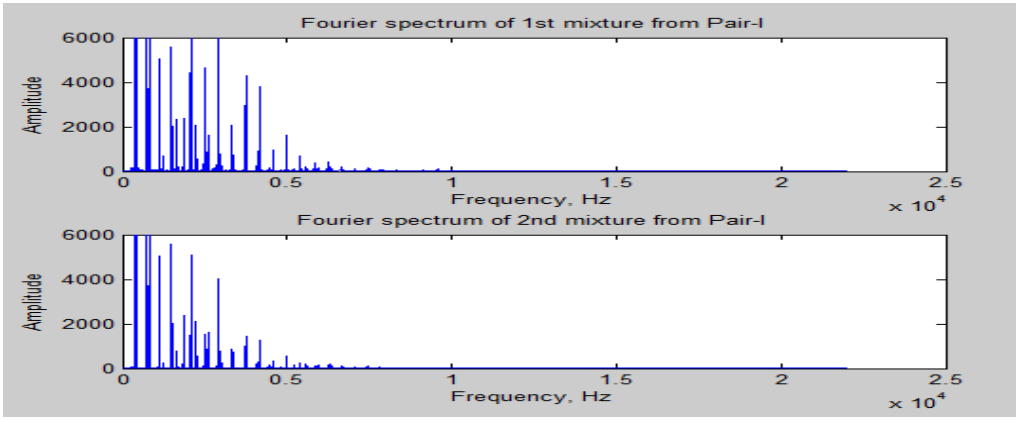
(b) Pair-II



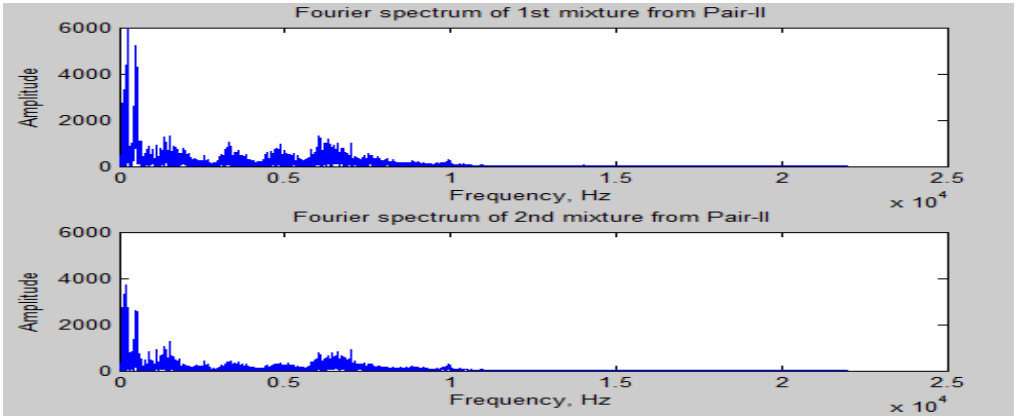
(c) Pair-III

Figure 3.5 Waveforms of mixtures of input signals

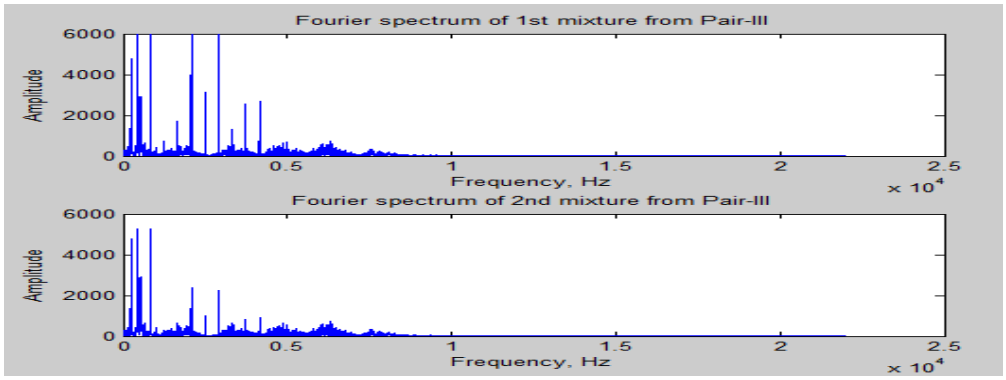
while Figure 3.6 displays their frequency spectra.



(a) Pair-I



(b) Pair-II



(c) Pair-III

Figure 3.6 Frequency spectra of mixtures of input signals

3.4.1 Preparation of Mixed Signals before Separating

Pre-processing for separation involves whitening and centering. Most ICA algorithms work better when whitening and centering is done on the single mixtures (Kumar & Naik, 2011). In this research, the preprocessing stage of the mixtures was done for NGA in MATLAB code. First, the centering procedure was done before carrying the whitening procedure. The mathematical concept of whitening and centering is given in **Appendix II**. The whitening procedure helps to reduce the number of elements need to reach optimization level for the algorithm.

3.5 Separating Stage

A key issue at this stage is that the separating system is only aware that there is a two signal mixture to be separated, but it does not know how the signals were mixed, the original signals nor the ratio of their mixtures. In other words, the separating system only knows that it needs to separate the mixtures into two by way of grouping components that resemble each other. Noise in the mixture is Gaussian and is evenly distributed. Another important consideration is that separation of the mixtures x_1 and x_2 depends on the assumption that the signals are independent and non-Gaussian in nature.

The separation process uses the Fibonacci Activation Function (FAF) and Sigmoid Activation Function (SAF) that continuously changes the entropy of each of the mixtures. Equation (2.23) for the Fibonacci activation functions used in the separation is indicated below:

$$\varphi(y) = \frac{\sqrt{5}-1}{2}y^3 + \frac{3-\sqrt{5}}{2}y^5$$

while Equation (2.24) of the Sigmoid activation function is shown as and is also used in the separation:

$$\varphi(y) = \frac{1}{1 + e^{-y}}$$

The MATLAB generated cdf plots for the two activation functions are shown in Figure 3.7:

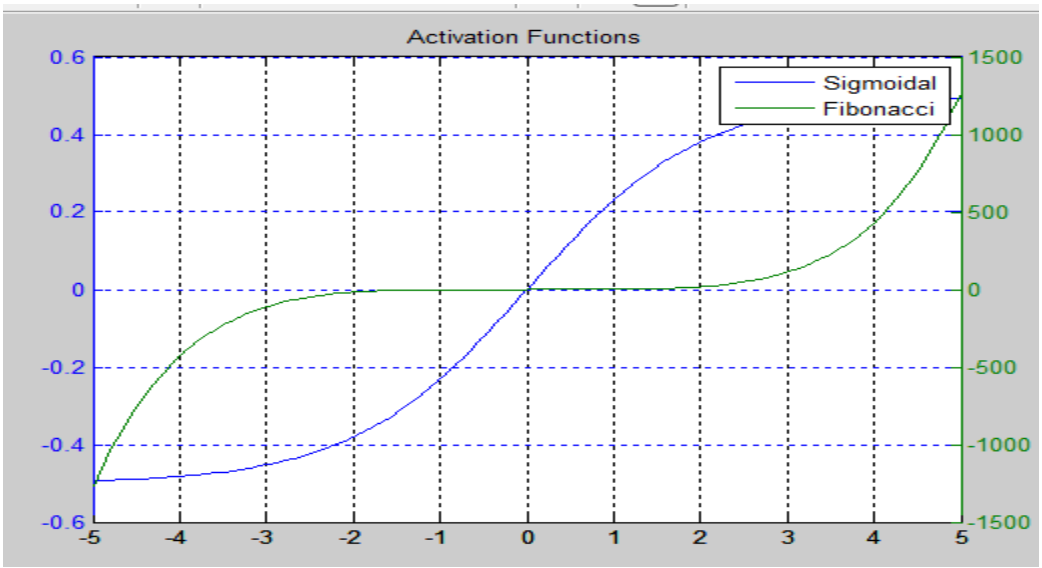


Figure 3.7 Cdf plots for the two activation function

A hundred iterations were used to attain the separation because higher numbers resulted in the same output and were rather too slow for the algorithm to converge. The learning rate was set at 0.001. The outputs y1 and y2 do not necessarily mean that the signals correspond to their s1 and s2, respectively, because of the ambiguity constraint in ICA. It can be the other way round, where y1 corresponds to s2 and y2 corresponds to s1. By listening to the separated signals and the input signals, and comparing their waveforms, it is only when it was possible to link the separated signal to its corresponding input counterpart. After the comparison is confirmed, the input and its corresponding output signals are quantified using the measure of Magnitude Squared Coherence. The convergence rates for the two activation functions were also compared with the results displayed in graphs. The outputs of this section form part of the results and are displayed in the next chapter.

3.6 Implementation methodology

1. A pair of input signals were prepared and stored in computer memory.
2. The two input signals were applied to the mixing matrix A. The mixing and separation was a $2x2$ MIMO system and the task was to recover the individual sources having only the record of their mixtures (the mixing process of the source is unknown).
3. To recover the original signals, the NGA code was first used to normalize and then the mixtures were whitened to make the separation easier. This was followed by applying the mixtures to a separating matrix. The separation was carried out to one hundred iterations using both the Fibonacci and Sigmoid activation function at learning rate of 0.001. The estimated signals were then stored for analysis.
4. The final stage was the comparison of the input signal and the estimated signal using Magnitude Squared Coherence function and the convergence rate elements in MATLAB. The methodology is summarized in Figure 3.8.

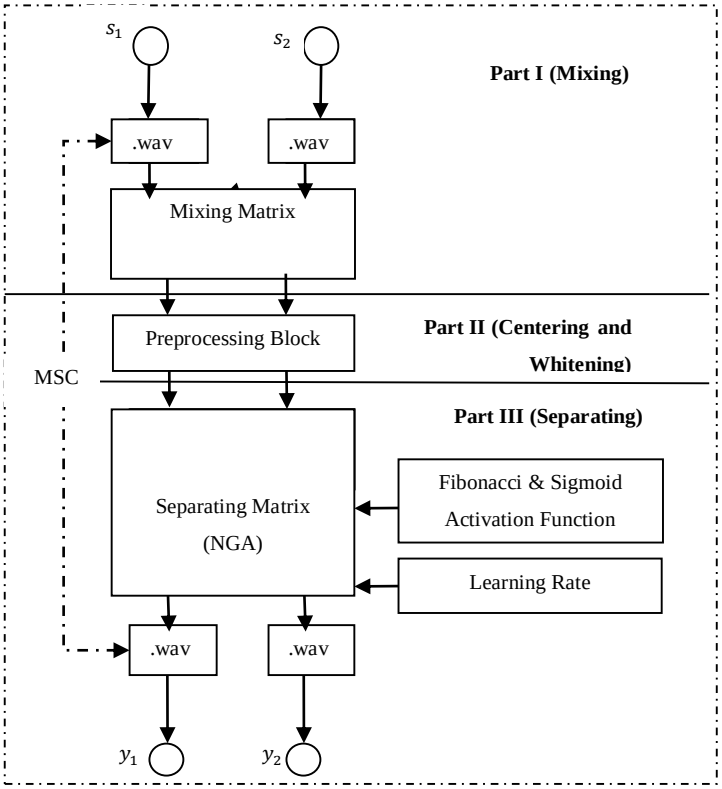


Figure 3.8 Schematic Flow Diagram of the Methodology

CHAPTER FOUR

RESULTS AND DATA ANALYSIS

4.1 Results

The results displayed here are for waveforms, frequency spectra and measure of magnitude-squared coherence. The estimated waveforms are displayed for each of the pairs of the input signals. Just like the waveforms of the inputs signals, the waveforms of the estimated signals are displayed with Amplitude in the y-axis and Time in seconds in the x-axis for easy comparison.

4.1.1 Results for waveforms and frequency spectra using Fibonacci AF

The estimated (output) signal waveforms and frequency spectra are displayed in red color in Figure 4.1, Figure 4.2, Figure 4.3 and Figure 4.4, Figure 4.5, Figure 4.6, respectively, for each of the three pairs using Fibonacci activation function.

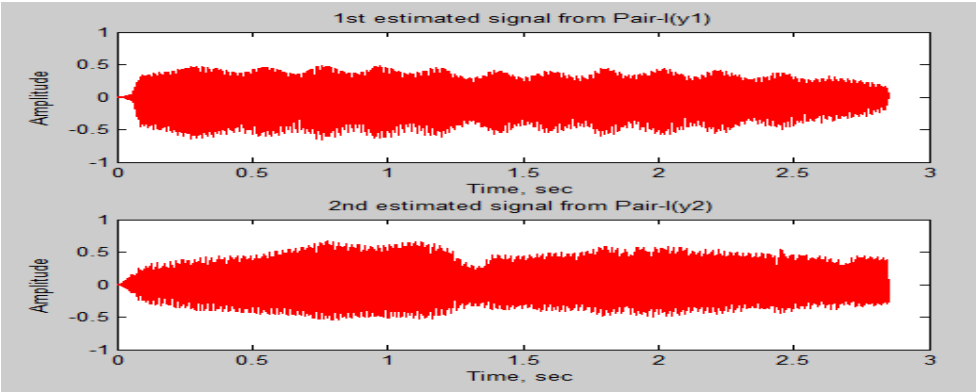


Figure 4.1 Waveforms of estimated signals from Pair-I using FAF

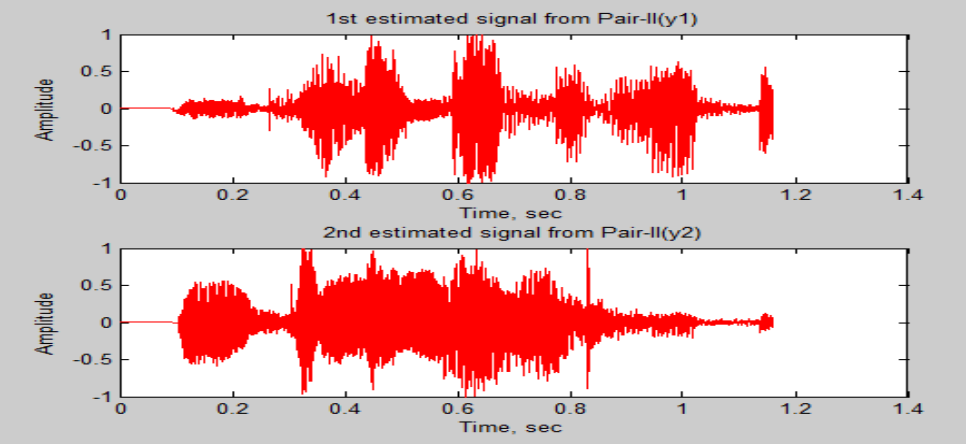


Figure.4.2 Waveforms of estimated signals from Pair-II using FAF

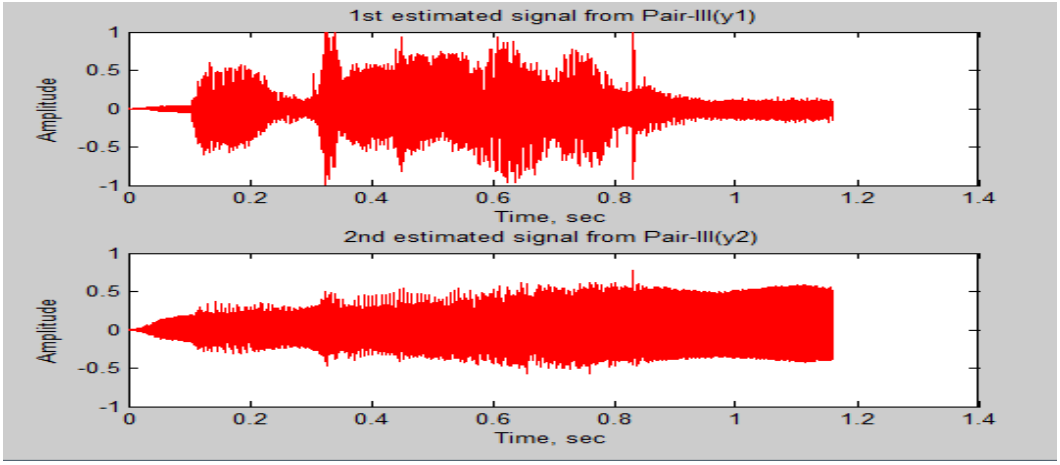


Figure 4.3 Waveforms of estimated signals from Pair-III using FAF

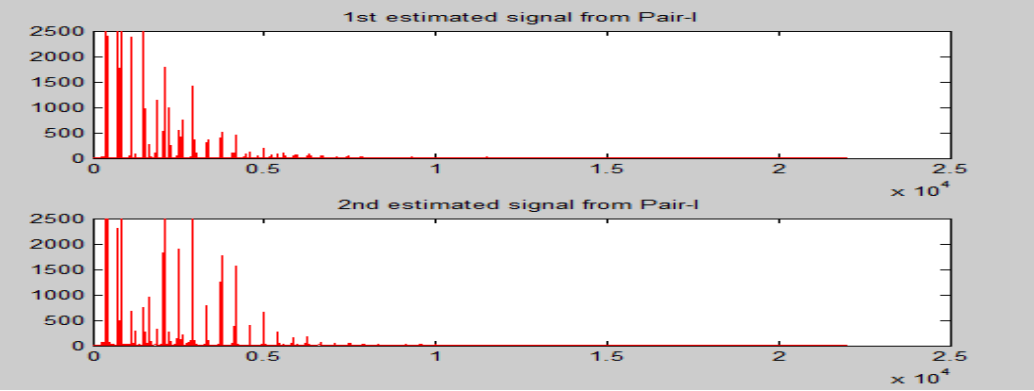


Figure 4.4 Frequency spectra of estimated signals from Pair-I using FAF

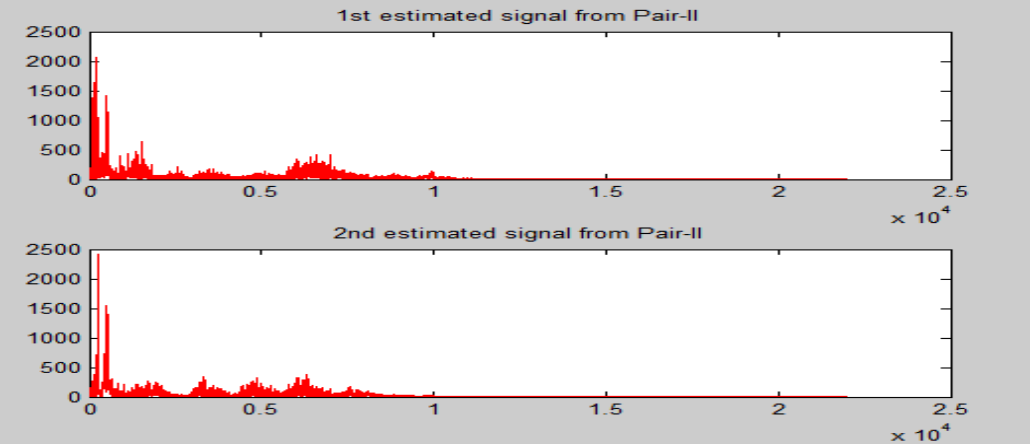


Figure 4.5 Frequency spectra of estimated signals from Pair-II using FAF

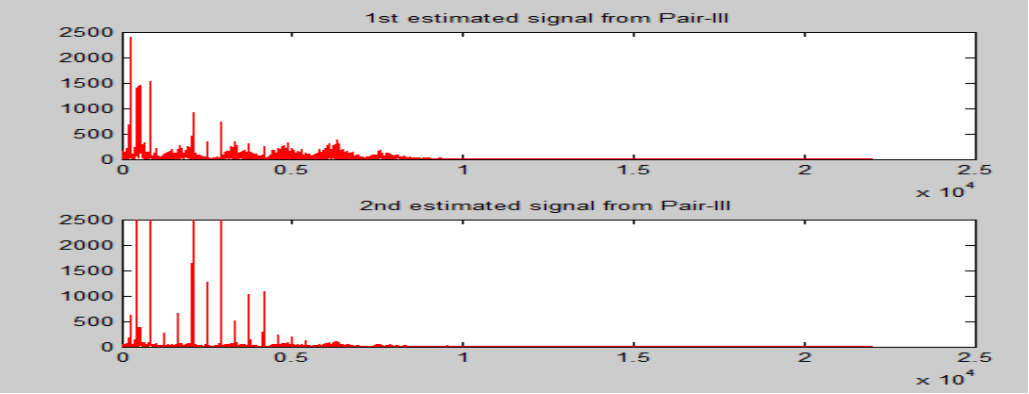


Figure 4.6 Frequency spectra of the estimated signals from Pair-III using FAF

4.1.2 Results for waveforms and frequency spectra using Sigmoid AF

The estimated (output) signal waveforms and frequency spectra are displayed in Figure 4.7, Figure 4.8, Figure 4.9 and Figure 4.10, Figure 4.11, Figure 4.12, respectively, for each of the three pairs using Sigmoid activation function.

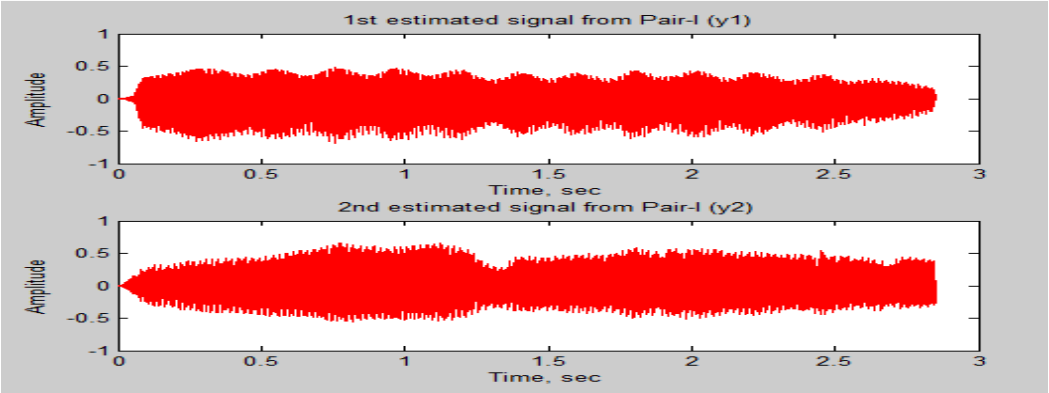


Figure 4.7 Waveforms of estimated signals from Pair-I using SAF

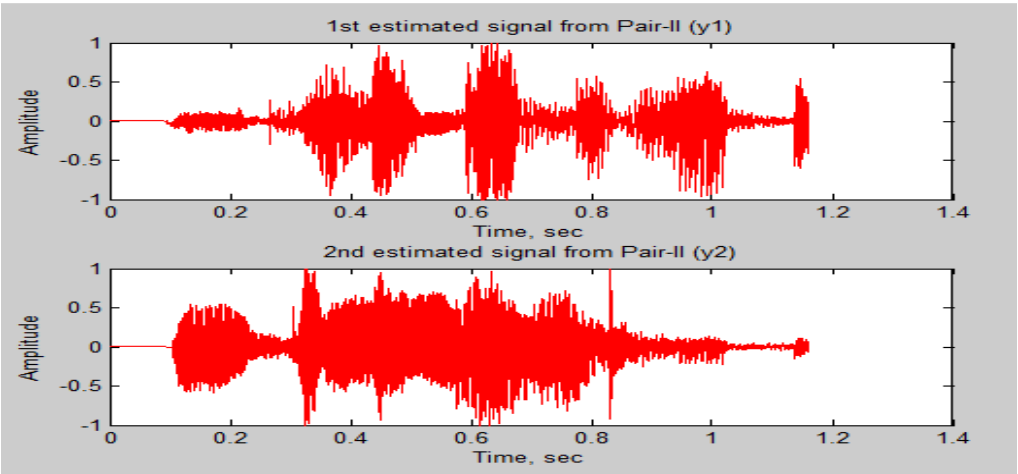


Figure 4.8 Waveforms of estimated signals from Pair-II using SAF

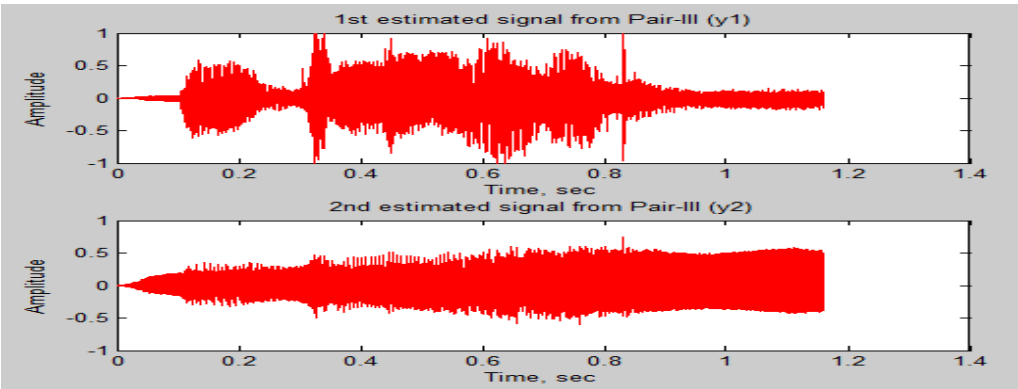


Figure 4.9 Waveforms of estimated signals from Pair-III using SAF

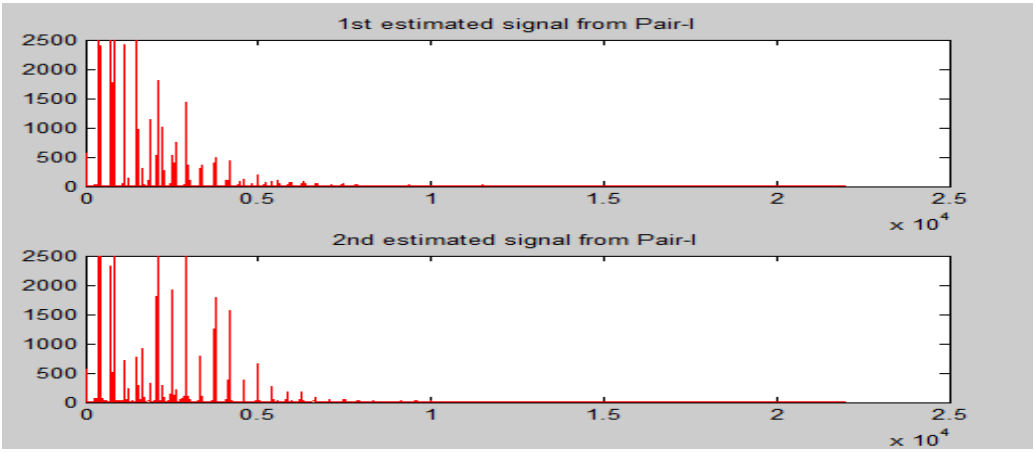


Figure 4.10 Frequency spectra of the estimated signals from Pair-I using SAF

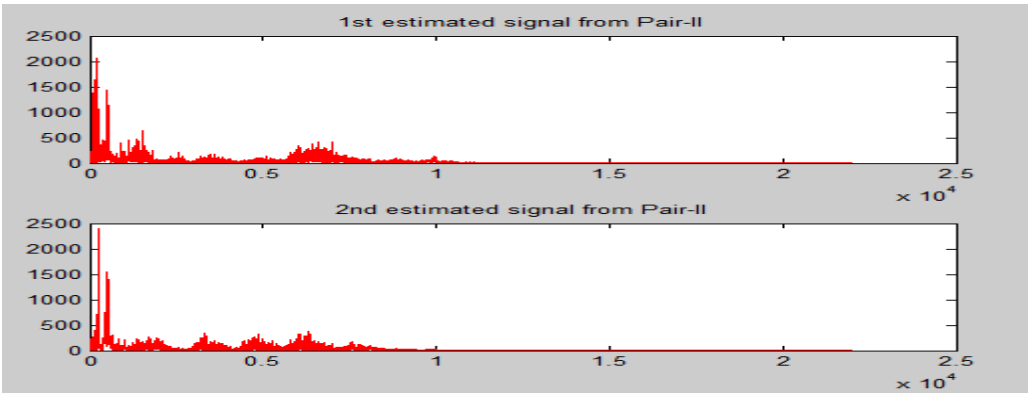


Figure 4.11 Frequency spectra of the estimated signals from Pair-II using SAF

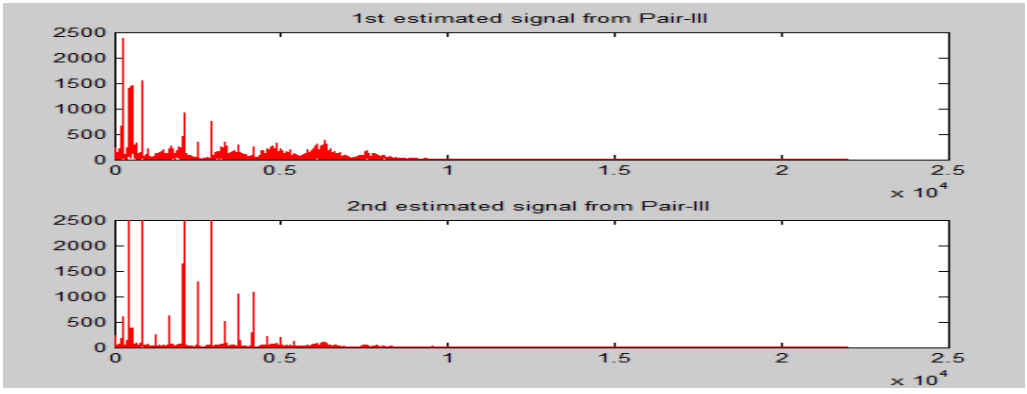


Figure 4.12 Frequency spectra of the estimated signals from Pair-III using SAF

4.2 Comparison of Input-Output Waveforms and Frequency Spectra

For easy comparison, analysis and discussion, the input and output waveforms and frequency spectra are placed side by side. First, the thesis compares the waveforms and then the frequency spectra.

4.2.1 Waveform Comparison using Fibonacci AF

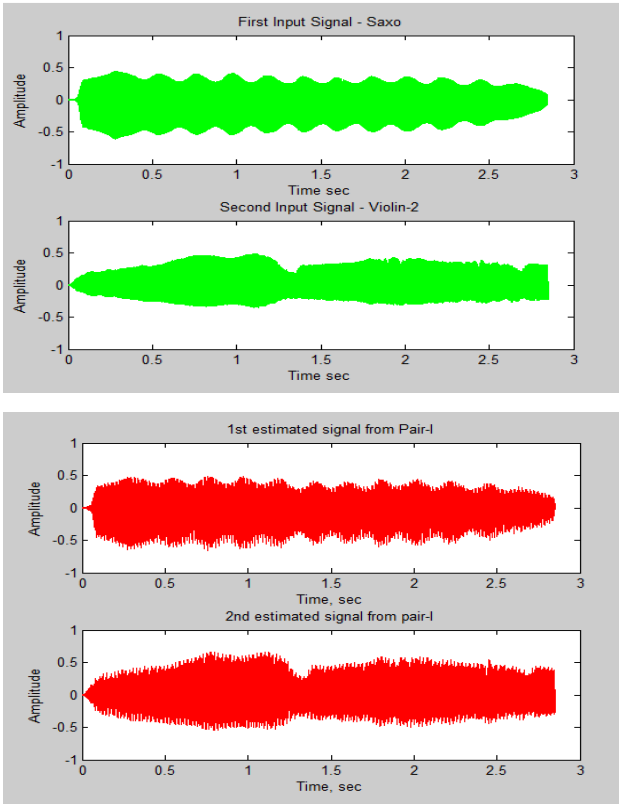


Figure 4.13 Waveform Input-Output comparison from Pair-I

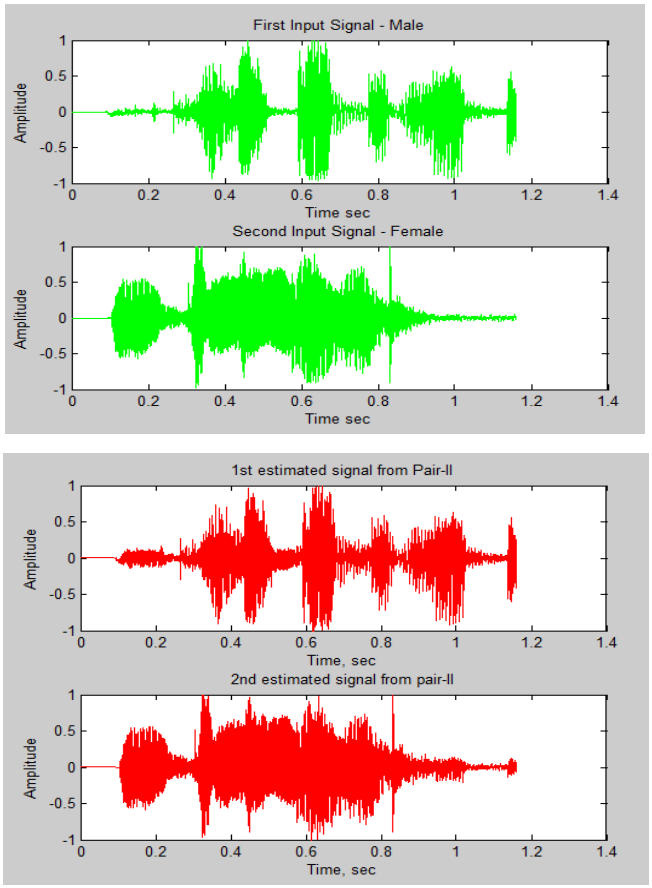


Figure 4.14 Waveform Input-Output comparison from Pair-II

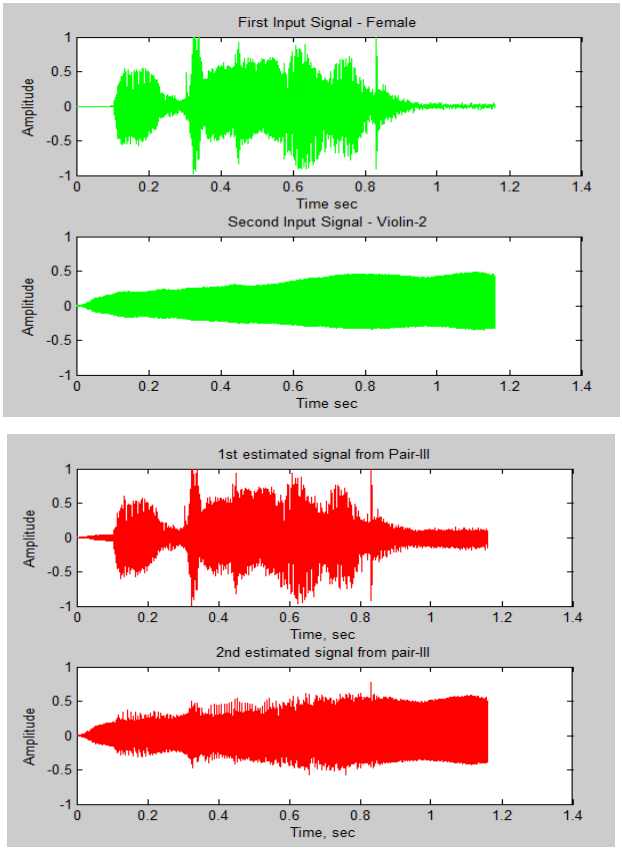


Figure 4.15 Waveform Input-Output comparison from Pair-III

4.2.2 Waveform Comparison using Sigmoid AF

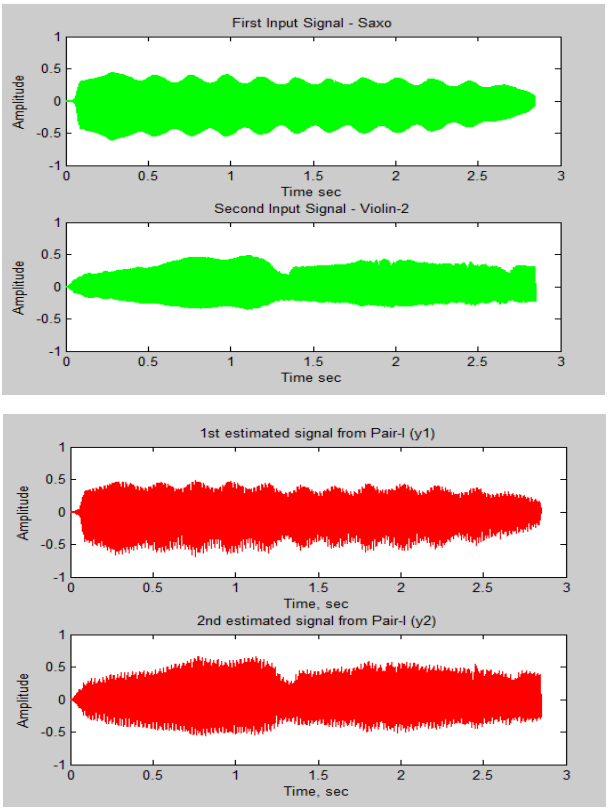


Figure 4.16 Waveform Input-Output comparison from Pair-I

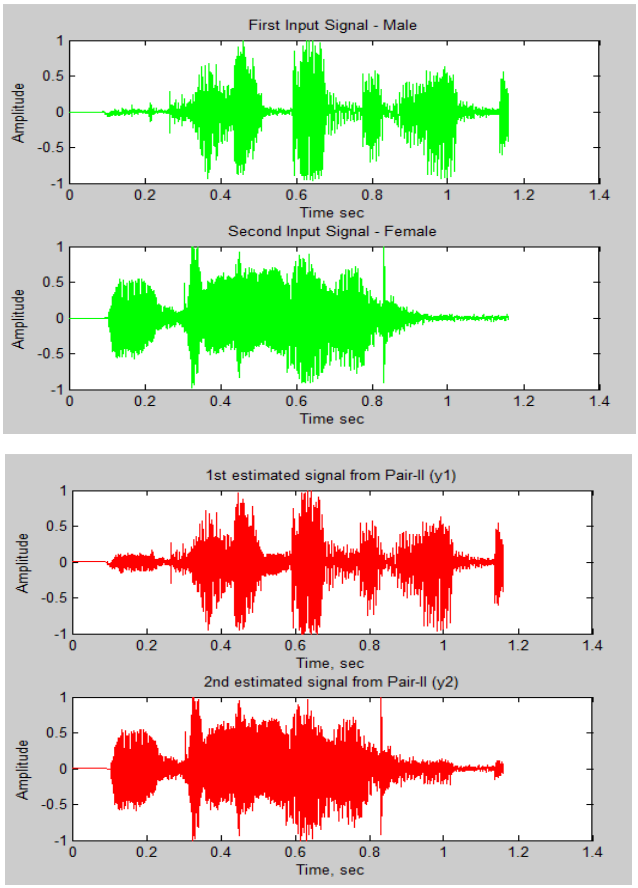


Figure 4.17 Waveform Input-Output comparison from Pair-II

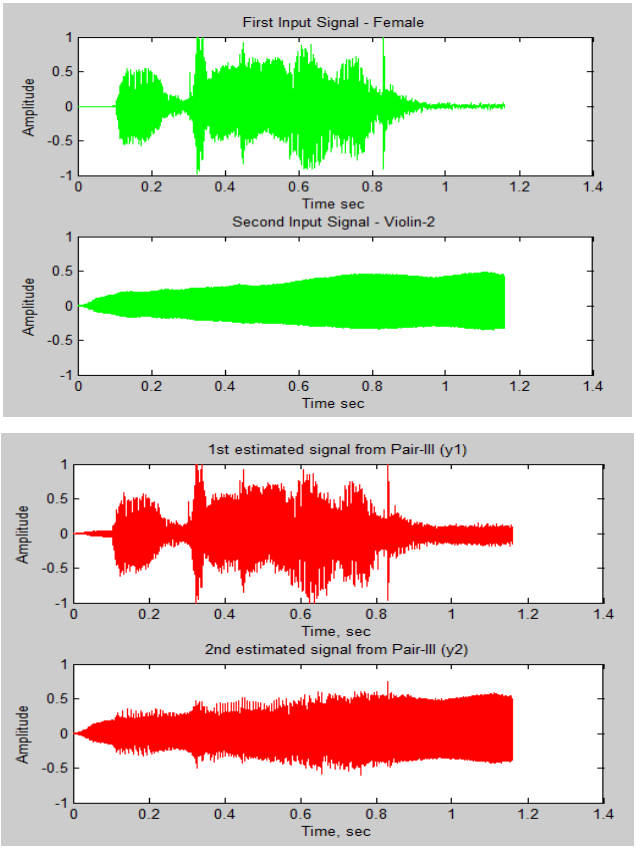


Figure 4.18 Waveform Input-Output comparison from Pair-III

As can be observed from the comparisons for the waveforms using both FAF and SAF, the separated signals resemble their input counterparts. However, a closer observation show that the output signals have more sparks, unlike their input counterparts that have smooth edges, explaining why the output signals are only the filtered versions of their original counterparts.

From the observation of the waveforms, it is not easy to distinguish the effectiveness of the chosen algorithm and the activation actions. The shapes of the waveforms show that FAF works in a similar manner just like the well-known SAF.

4.2.3 Frequency Spectrum comparison using FAF

Further comparison is extended to compare the frequency spectra shown in Figure 4.19, Figure 4.20 and Figure 4.21.

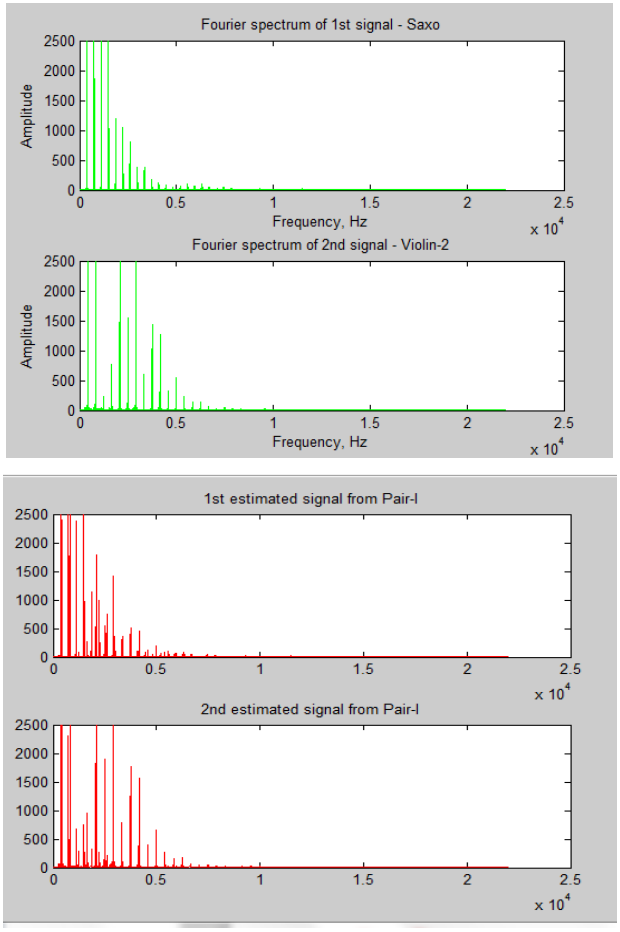


Figure 4.19 Input-Output frequency spectra comparison from Pair-I using FAF

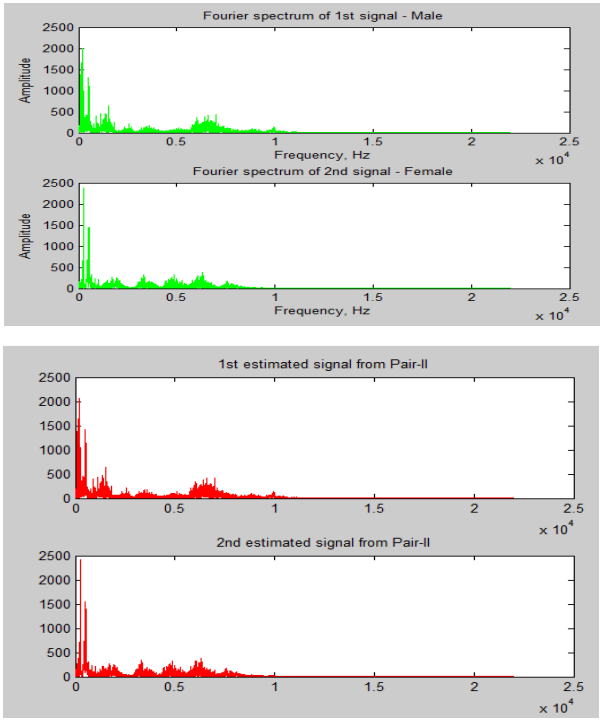


Figure 4.20 Input-Output frequency-spectra comparison (Pair-II) using FAF

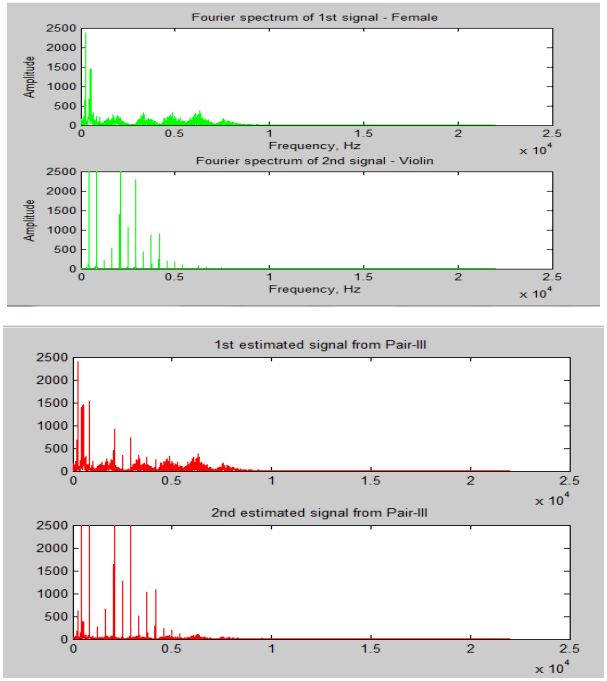


Figure 4.21 Input-Output frequency spectra comparison (Pair-III) using FAF

As can be observed from Figure 4.19, Figure 4.20 and Figure 4.21., most of the frequencies from the input signals are retained in the output signals. However, the corresponding frequency spectra in the Pair-II shows the closest resemblance to their input signals, followed by Pair-II signal frequencies then Pair-I come last. It can be said that the super-Gaussian signals are the most efficiently separated and their frequencies are the most retained, followed by a pair containing the sub-Gaussian signal and super-Gaussian signal. The least separated are those with both inputs as sub-Gaussian signals.

4.2.4 Frequency Spectrum comparison using SAF

Further comparison is extended to compare the frequency spectra shown in Figure 4.22, Figure 4.23 and Figure 4.24.

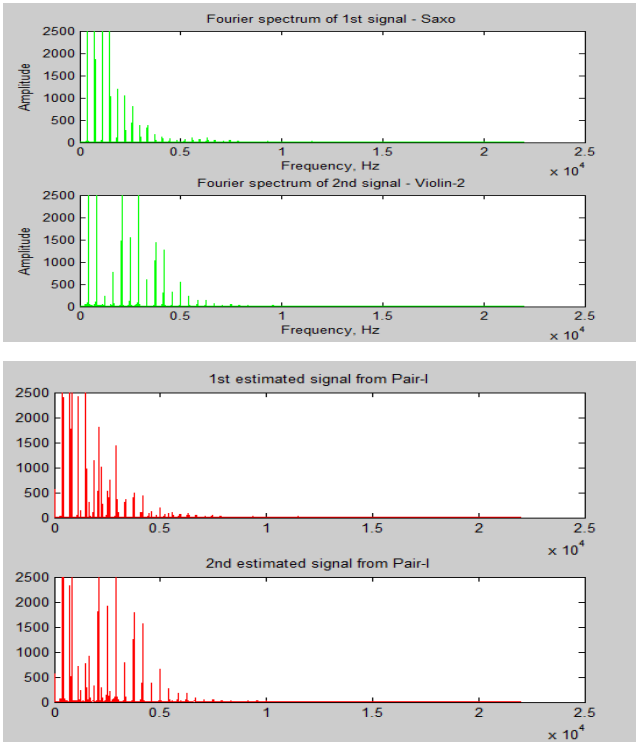


Figure 4.22 Input-Output frequency spectra comparison (Pair-I) using SAF

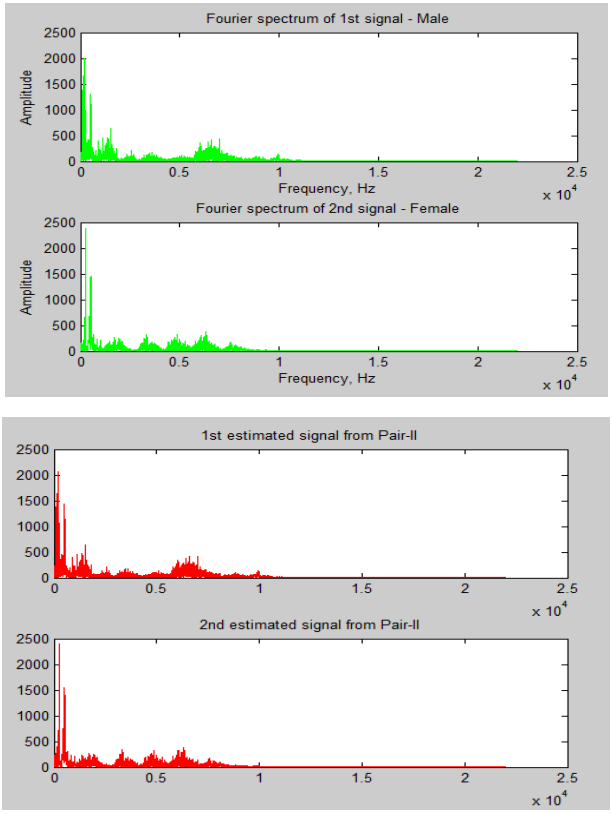


Figure 4.23 Input-Output frequency spectra comparison (Pair-II) using SAF

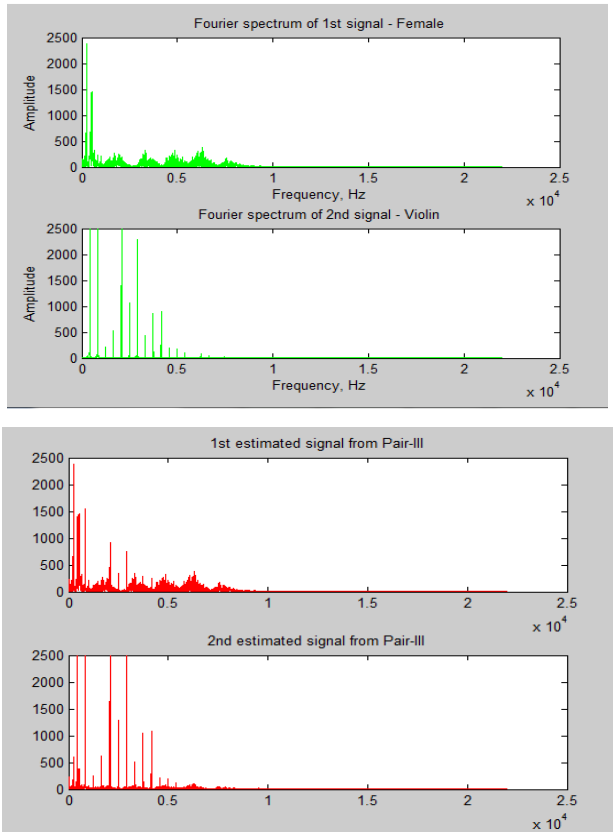


Figure 4.24 Input-Output frequency spectra comparison (Pair-III) using SAF

4.3 Analysis of the Results

We first look at the performance of input signals that are sub-Gaussian. The Pair-I represents all input signals as sub-Gaussian (Saxophone and Violin). The results obtained from this pair reveal a relatively poor resemblance of their original signals on listening to their audio files. From the comparisons of the waveforms, frequency spectra, the working of the NGA algorithm and the FAF show the poorest separation when compared to other pairs. The second least performance was on Pair-III, which is made up of a super-Gaussian and sub-Gaussian signals (Female and Violin). Violin sound is sub-Gaussian signal while the Female speech is a super-Gaussian signal and the purpose was to find out how the two activations functions fair with this kind of mixed signals. In both FAF and SAF, the waveform and frequency spectra of the separated Female voice had the closed resemblance to the original signals when

compared to the Violin sound. Pair-II shows the best separation among the three pairs used. It, therefore, means that the two activation functions are meant for signals that are super-Gaussian in nature when used inside the NGA algorithm. The findings show that the Fibonacci and Sigmoid activation functions works best for speech signals than for music signals.

4.3.1 Magnitude Squared Coherence Measure

With the exception of listening to the sounds of input and output signals and making a comparison, the waveforms and frequency spectrum do not prove clearly the analysis that FAF and SAF work effectively for super-Gaussian signals. The results on waveforms and frequency spectra help in comparison through observation, which is often biased as a result of different interpretations. It is important to quantify the quality of separation in order to understand how the algorithm performs the separation at specific point in time on the entire signal. In most cases, the comparison is done using Signal-to-Noise Ratio by showing the percentage or ratio of separation for the entire audio signal. However, it is important to understand how the comparison works at each frequency making the use of MATLAB's *mscoherence* measure an ideal choice for this situation. The inputs to the *mscoherence* functions are the input signal and its corresponding output signal in each pair.

Because of the permutation ambiguity of ICA, the signals are simply separated in two groups, without knowing which one corresponds to which input signals. That means that the output signals y_1 and y_2 do not necessarily correspond to input signals s_1 and s_2, respectively. To know which output corresponds to which input, we had to listen to the outputs first and then observe the waveforms and frequency spectra. The Magnitude Squared Coherence was computed using Equation (2.31) indicated below:

$$C_{sy}\omega = \frac{\left|P_{sy}(\omega)\right|^2}{P_{ss}(\omega)P_{yy}(\omega)}$$

and it was used to display the comparison over a range of frequencies. First, a comparison of the same signal as input-one and input-two is used without mixing and separating it to show an ideal situation for the male speech as shown in Figure 4.25.

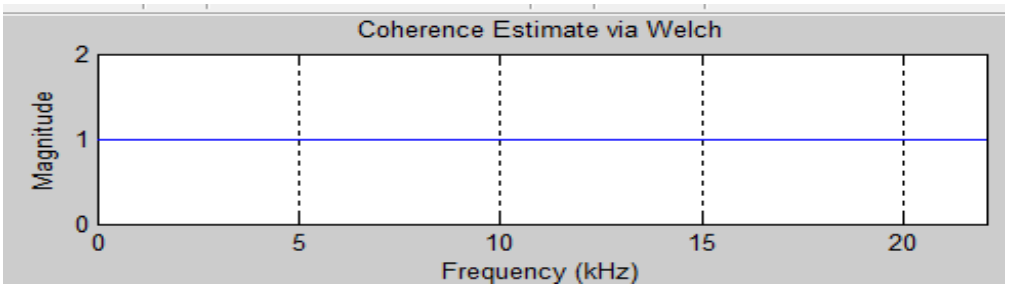
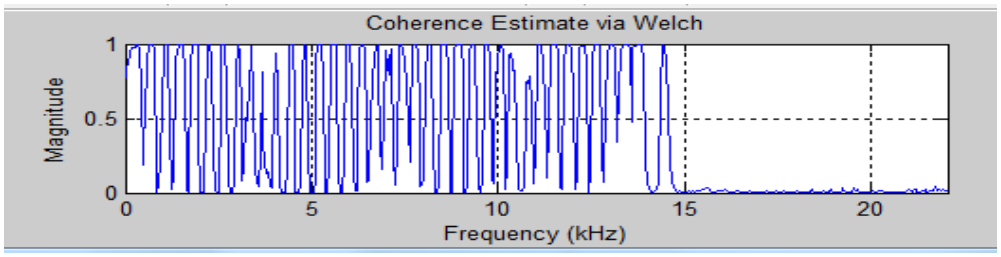


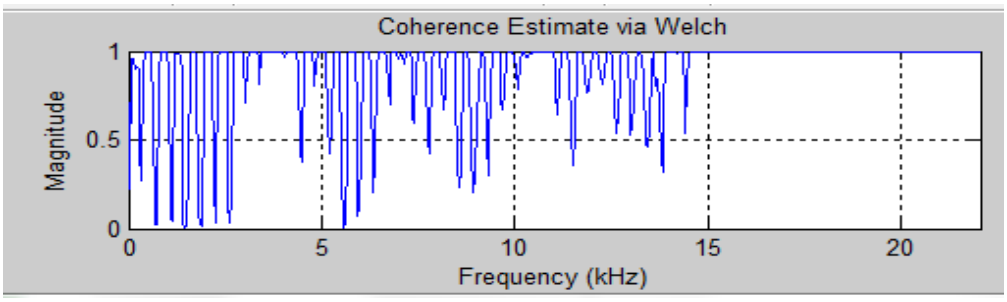
Figure 4.25 Input-Output frequency spectra comparison from Pair-III

The research's measure of Magnitude Squared Coherence is shown in Figure 4.26 and Figure 4.27 for Fibonacci and Sigmoid AF, respectively.

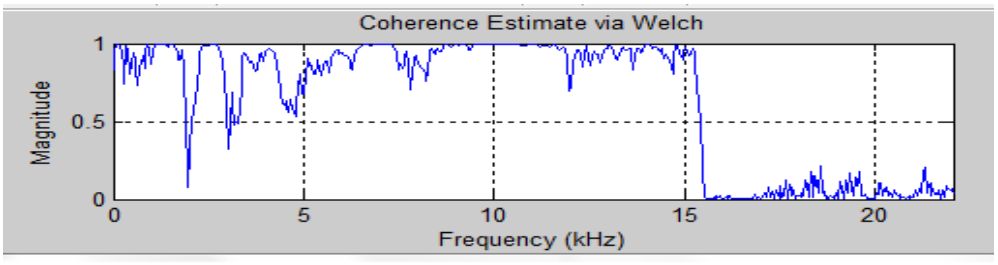
(i) MSC Measure using Fibonacci Activation Function



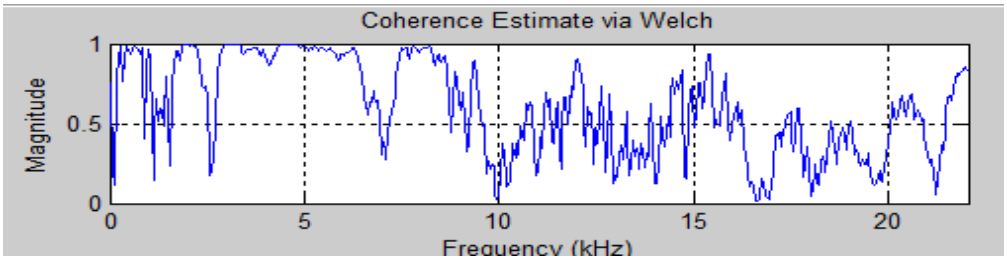
(a) Saxophone from Pair-I



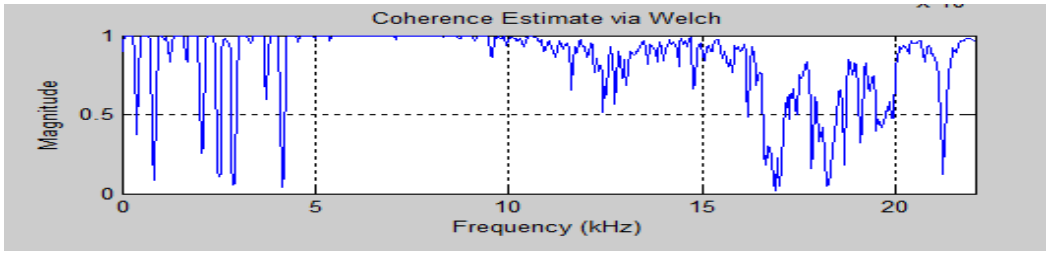
(b) Violin-2 from Pair-I



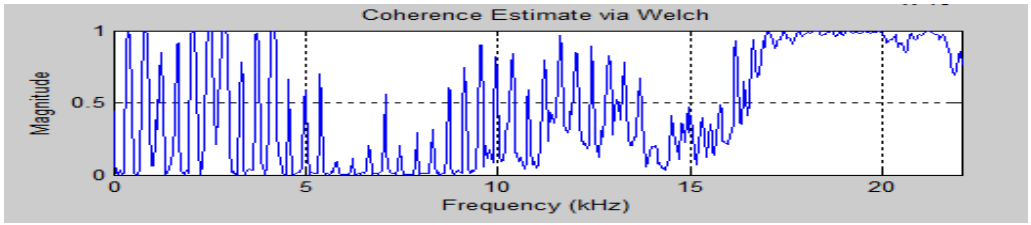
(c) Male from Pair-II



(d) Female from Pair-II



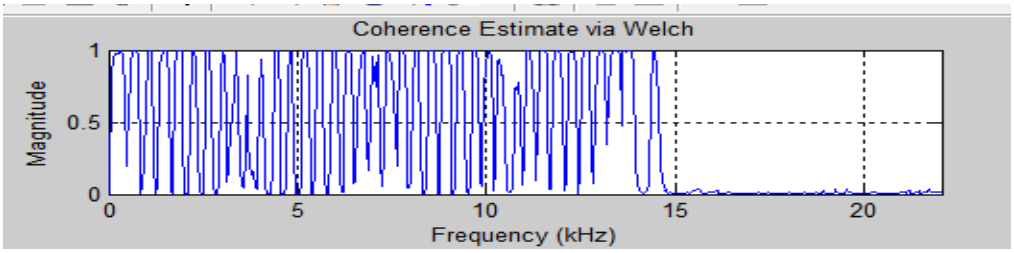
(e) Female from Pair-III



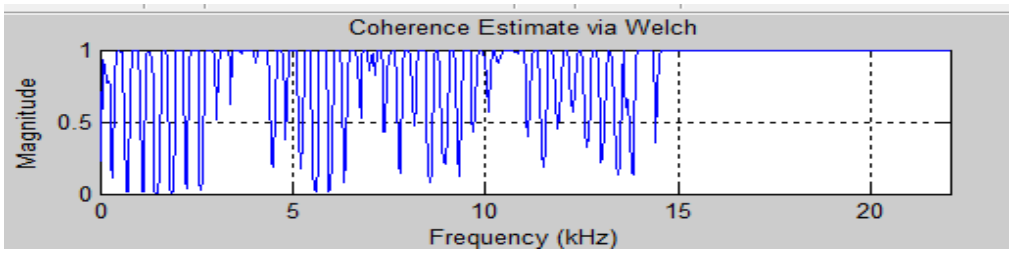
(f) Violin-2 from Pair-III

Figure 4.26 Results of the measure of magnitude squared coherence using FAF

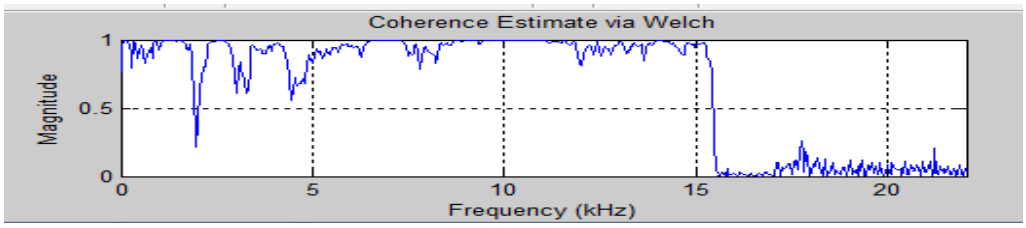
(ii) **MSC Measure using Sigmoid Activation Function**



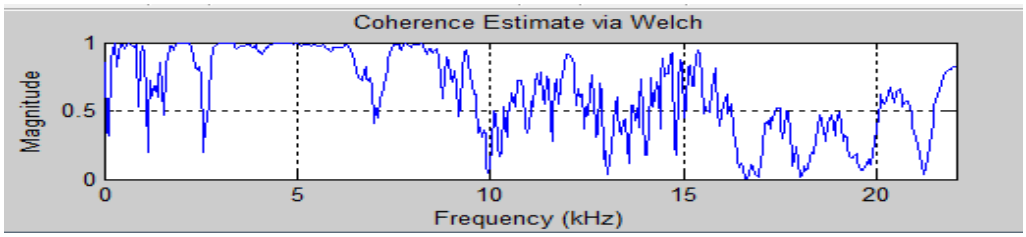
(a) Saxo from Pair-I



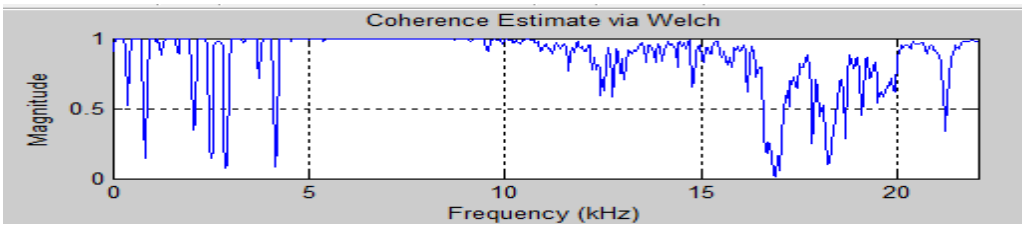
(b) Violin-2 from Pair-I



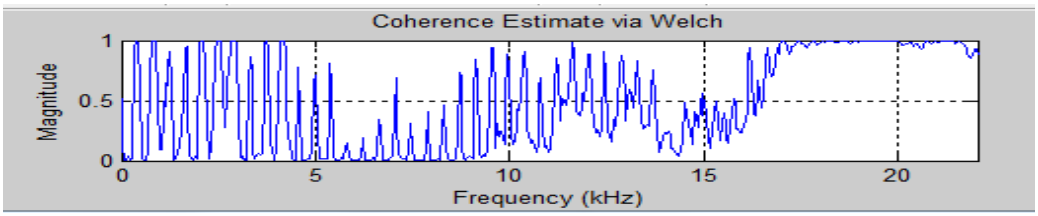
(c) Male from Pair-II



(d) Female from Pair-II



(e) **Female from Pair-III**



(f) Violin-2 from Pair-III

Figure 4.27 The measure of magnitude squared coherence using SAF

From Figure 4.26, and Figure 4.27, the magnitude squared coherence measure of the male and female signals in Pair-II give the best performance of all the pairs under investigations. This is followed by the Pair-III, containing the human (female) and music signal in both Figure 4.26 and Figure 4.27, and within this pair, the human separation (Figure 4.26(e) and Figure 4.27(e)) is much better than that its music counterpart of (Figure 4.26(f) and Figure 4.27(e)). The Pair-I represents the least separation and it contains all music signals (Figure 4.26(a) and (b) and Figure 4.27(a) and (b)).

4.3.2 Quantitative Comparison of Corresponding Input-Output Signals on the Activation Functions

Here, the Magnitude Squared Coherence (MSC) at each frequency is compared for both Sigmoid and Fibonacci Activation Functions on the same graph. In the first graph (a), the corresponding input and output signal are placed side by side. To get an average of the comparisons, the coherence measure of (a) in each pair is passed through a moving average filter of size 50 to generate a smoothened version of the MSC measure

shown in (b) for each comparison of the two activation functions. Further (b) is plotted for its semi-log likelihood and the results shown in (c). The results of the graphs are shown in Figures 4.28 to Figure 4.33, below:

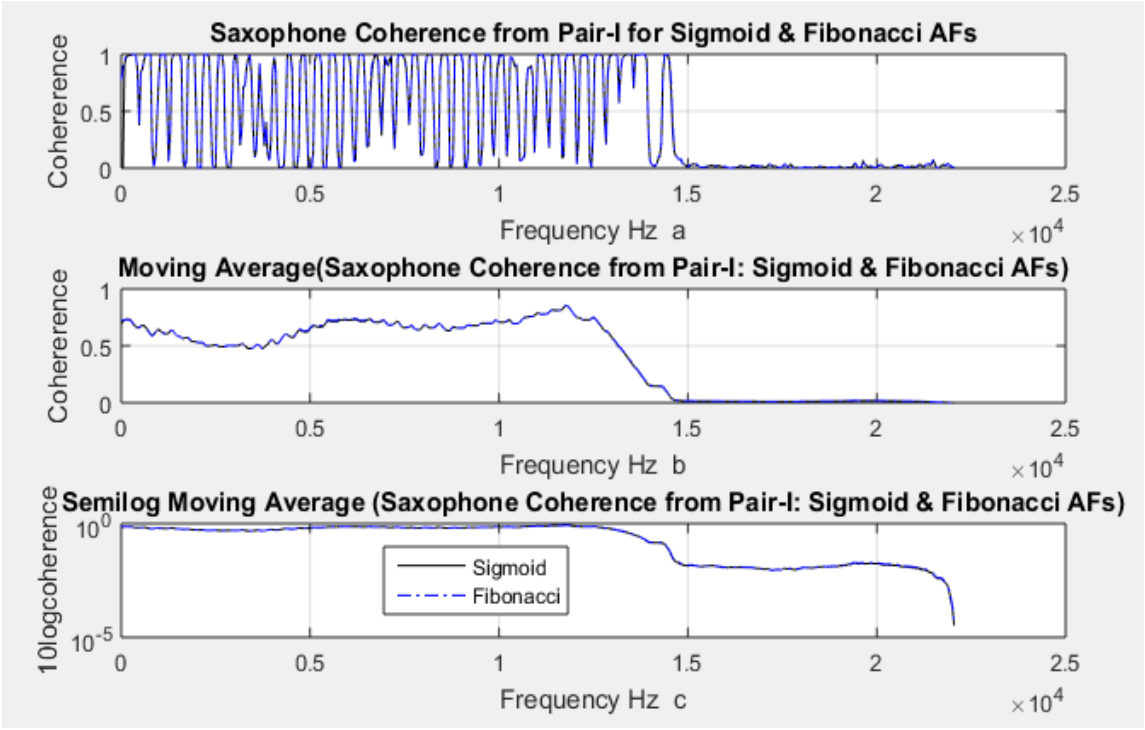


Figure 4.28 Saxophone coherence measure comparison of separated signals using Sigmoid and Fibonacci AFs from Pair-I: (a) coherence; (b) moving average coherence; (c) semi-log moving average coherence.

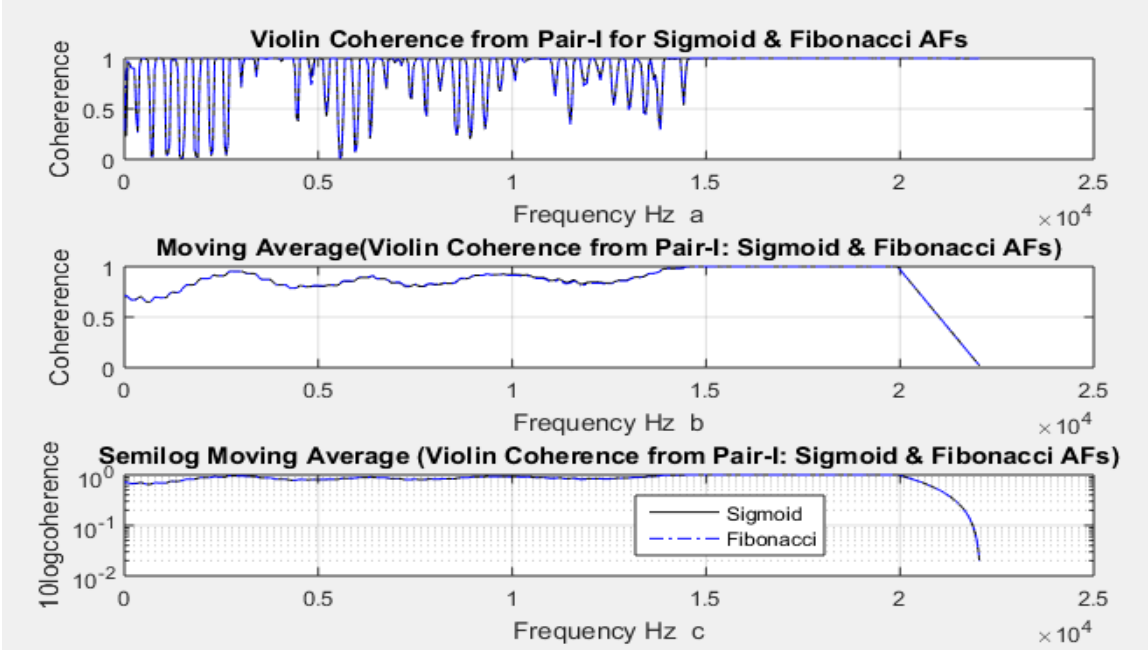


Figure 4.29 Violin coherence measure comparison of separated signals using Sigmoid and Fibonacci AFs from Pair-I: (a) coherence; (b) moving average coherence; (c) semi-log moving average coherence.

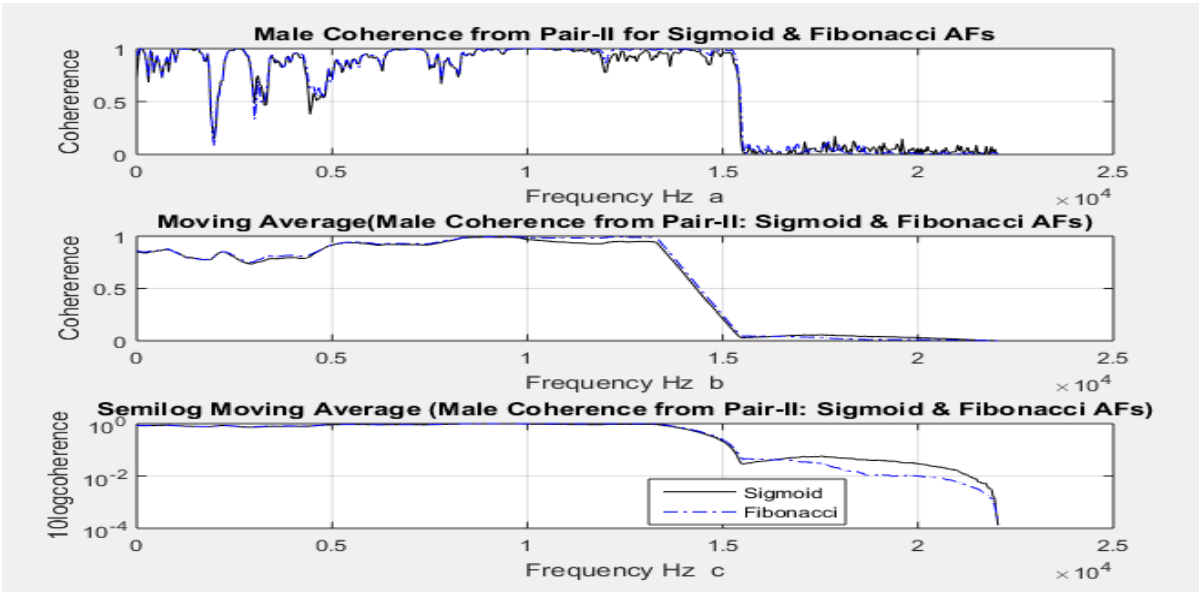


Figure 4.30 Male coherence measure comparison of separated signals using Sigmoid and Fibonacci AFs from Pair-II: (a) coherence; (b) moving average coherence; (c) semi-log moving average coherence.

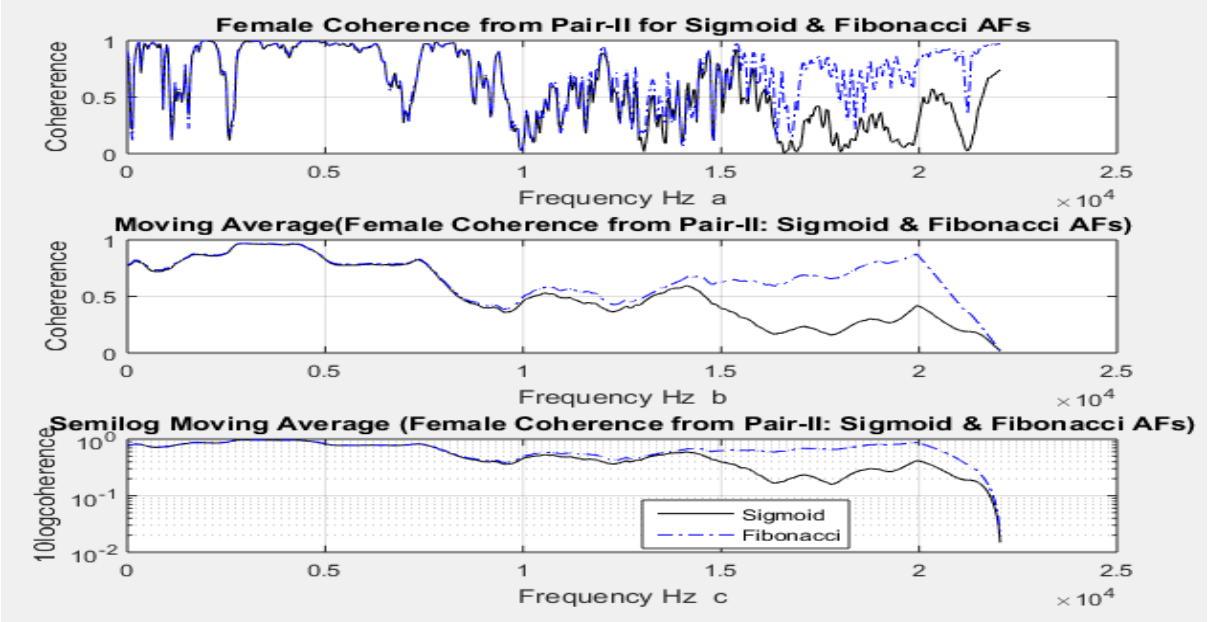


Figure 4.31 Female coherence measure comparison of separated signals using Sigmoid and Fibonacci AFs from Pair-II: (a) coherence; (b) moving average coherence; (c) semi-log moving average coherence.

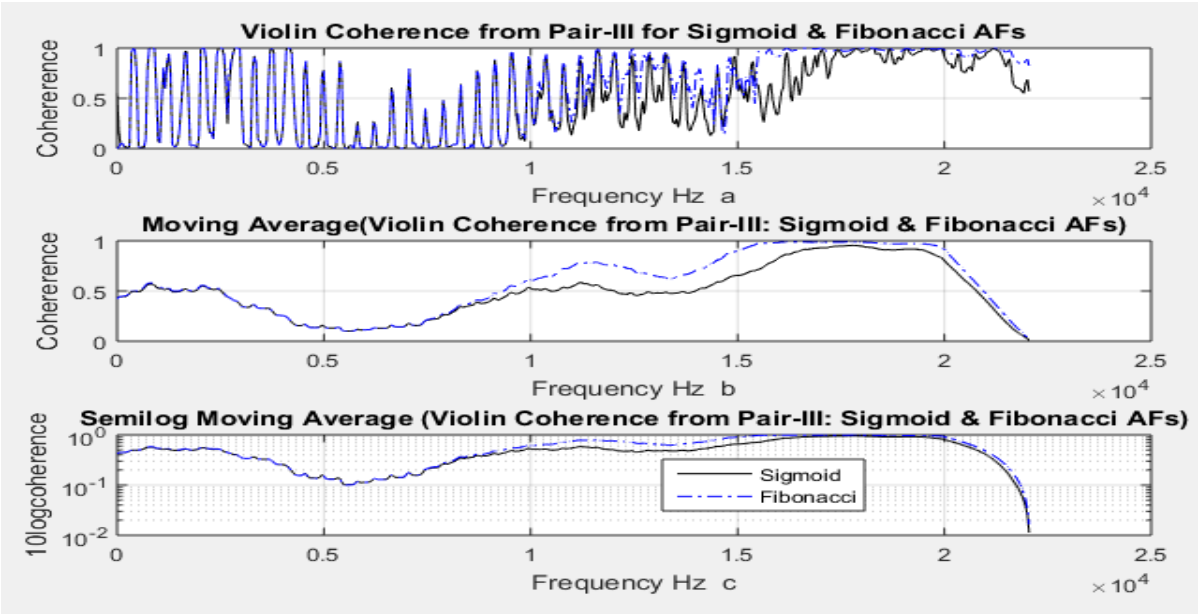


Figure 4.32 Violin coherence measure comparison of separated signals using Sigmoid and Fibonacci AFs from Pair-III: (a) coherence; (b) moving average coherence; (c) semi-log moving average coherence.

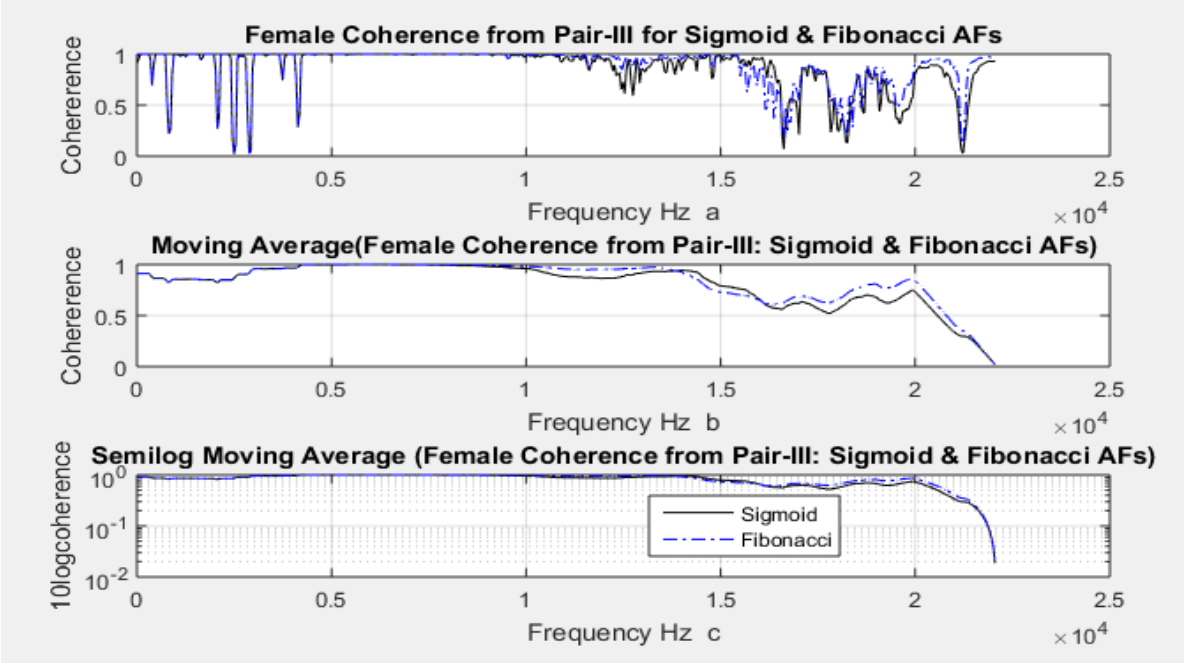


Figure 4.33 Violin coherence measure comparison of separated signals using Sigmoid and Fibonacci AFs from Pair-III: (a) coherence; (b) moving average coherence; (c) semi-log moving average coherence.

It is evident that Fibonacci and Sigmoid Activation Functions are meant for speech signals because the best separation is from Pair II, that is, speech signals. Further that Fibonacci AF works better than Sigmoid AF because it has a higher Magnitude Coherence Measure.

4.3.3 Correlation Coefficient Analysis

The analysis of the two activation functions is further quantified through the measure of correlation-coefficient. Correlation coefficient is calculated using Equation 4.1:

$$Correcoef(S,Y) = \frac{Covariance(S,Y)}{\sqrt{Covariance(S,S)Covariance(Y,Y)}} \tag{4.1}$$

where S is the input signal, and Y is the estimated/output signal.

The result for the correlation coefficient are shown in Table 4.1 below:

Table 4.1 Correlation coefficient of input to its corresponding output signal

	Input-Output Signals	**Correlation Coefficient using Sigmoid AF**	**Correlation Coefficient using Fibonacci AF**
Pair-I	Saxophone	0.7132	0.7327
	Violin	0.7237	0.7581
Pair-II	Male	0.8156	0.8584
	Female	0.8103	0.8512
Pair-III	Violin	0.7541	0.7556
	Female	0.7924	0.8013

It is evident from the correlation coefficients that the signals separated with Fibonacci Activation Functions had higher values in each pair when compared to the pairs separated with Sigmoid Activation Functions. Further that both Sigmoid and Fibonacci are good for super-Gaussian Signals (speech signals of Pair II had the highest correlation coefficients) with Fibonacci showing slightly better result than Sigmoid.

CHAPTER FIVE

CONCLUSION

The residing theme here is to relate the results obtained to the objectives of the thesis and explain any justifications for the outcomes. In addition, the chapter looks at the achievements of the research, limitations and recommendation for future research areas that may be pursued.

5.1 Attainment of Objectives

5.1.1 Main Objective

The thesis' main aim was to create a system for separating two input audio signals that obey the assumptions of ICA for separating two pairs of audio signal. The objective has been achieved as all the three pairs of signals could be separated using the NGA modeled in MATLAB for non-Gaussian signals. The fact that the NGA algorithm was able to take two input audio signals, mix them, and separate them into two output just like the inputs, reveals the design of the system to accomplish this main objective has been accomplished.

5.1.2 Specific Objective One

The specific objective one aims of formulating NGA algorithms for mixing and separating blind signals formulated using ICA was also achieved. However, it was not just about separating non-Gaussian signals, the specific objectives three and four were to understand the effectiveness of the designed system in terms of quality with a key focus on two activation functions – the Sigmoid and the Fibonacci activation functions.

5.1.3 Specific Objective Two

For specific objective two, the waveforms, frequency spectrums and the coherence measure all showed that Fibonacci and Sigmoid are classified not just as non-Gaussian activation function, but as super-Gaussian activation functions. This is because the two

activation functions showed the highest level of similarity between input and output signals when the speech signals are used. From theory, speech signals are classified as super-Gaussian signals.

The research realized that FAF and SAF are super-Gaussian activation functions and affirmed the conclusion that it is not simple enough to classify an algorithm but its effectiveness is determined by the choice of the AF. The effectiveness of the algorithm is shown in Table 5.1 below with the top pair as the best separation realized and the bottom the least realized for both FAF and SAF.

Table 5.1 Performance of NGA and FAF and SAF on the three pairs

Pair	**Signals**
Pair-II	Male, Female
Pair-III	Female, Violin-2
Pair-I	Saxo, Violin-2

From Table 5.1, the top pair realized the best results while the bottom pair was the least. As evidenced from the separation of Table 5.1, the algorithm works better for a pair of super-Gaussian signals, and least for a pair of sub-Gaussian signals when both Fibonacci and Sigmoid Activation functions are used.

5.1.4 Specific Objective Three

The objective was achieved because the correlation coefficients of the two activation functions in separating the pairs show that the Fibonacci Activation function is better than Sigmoid Activation Function (Table 4.1). The table shows that the quality of separation was high in super-Gaussian (speech) signals when compared to the pairs from sub-Gaussian (music) signals, and super-sub-Gaussian signal mixture separation. However, algorithms and by extension, activation function are measured on two fronts: quality and speed. While the quality is achieved through magnitude squared coherence (Figures 4.28 - 4.33) and correlation coefficient (Table 4.1) measures, the speed and

specifically, the convergence rate was not effectively achieved for the reasons outlined below.

Next is to analyze the speed, through the measure of convergence rates. Knowing that the two activation functions are good for speech signals (Pair-II) and that Fibonacci realizes better quality results than Sigmoid, the measure of convergence rate was also performed. The title of this thesis looks at linear systems for separating audio signals. Speech signals are non-stationary in nature; bring in the element of nonlinearity. In fact, from theory speech signals are only stationary in the range of 20ms to 35ms. Convergence measure of the two activation functions on speech signals would be the measure of convergence rates of the algorithm. Unfortunately, this measure is not possible for speech signals with the current title in mind. Linear transforms, as in this research use First Fourier Transforms and eigenvalues to approximate linearity. For nonlinear transforms and non-stationarity in speech signals, this will mean the use of wavelets in place of First Fourier Transforms and lyapunov exponents in place of eigenvalues. In mathematics, the Lyapunov exponets often within a dynamical system are the quantities that determine the separation rate characteristic of infinitesimally close trajectories. Lyaponuv exponents can be used to analyze the stability of any steady state behavior (Übeyl & Guler, 2007). The title would have to be changed to a nonlinear system, which is beyond the scope of this research if this is to be achieved.

To illustrate this, comparison of the two activation functions on simple artificially-generated stationary sine, cosine, square and saw tooth input wave signals on convergence and number of elements needed to reach optimization is shown below.

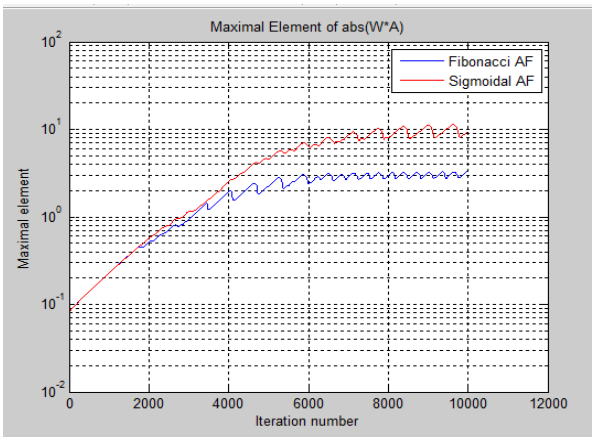


Figure 5.1 Comparison of convergence and maximal elements for stationary signals

Figure 5.1 is a plot of maximal element of matrix P, calculated as $P = Wk * A$ (from the algorithm). In optimal case, $P = eye(4)$ for four input signals and $P = eye(2)$ for two input signals and maximal element is 1. Figure 5.1 is used here to see the ideal situation of the convergence rates of the two activation functions on stationary signals. However, this is not possible from the current setup because of the use of speech signals that do not exhibit characteristics of stationarity. An attempt to use it directly in this research achieves the following:

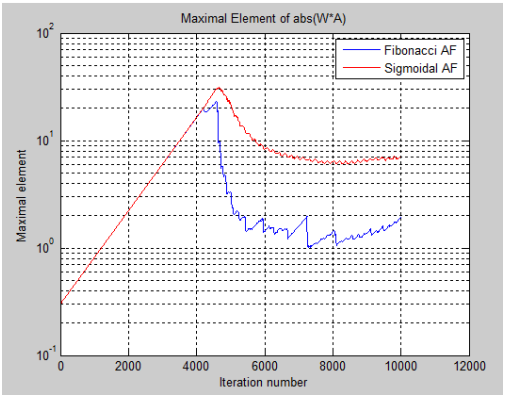


Figure 5.2 Convergence and maximal elements for non-stationary signals

It is clear from Figure 5.2 that the desired output is not achieved. This is attributed to the non-stationary of the speech signals that will require advanced concepts that are,

unfortunately, beyond the scope of this research, and is therefore considered as a future research consideration.

5.2 Thesis Contribution

5.2.1 Summary of the Contribution

Algorithms are made up of three main components: the algorithm itself at the top; the cost functions within the algorithm used to achieve convergence and optimization, at the middle; and the adjusting parameters within the cost function at the bottom. Most studies compare performance of different algorithms: this research uses a single algorithm. ICA algorithms work with non-Gaussian signals, and algorithms are compared at this levels. In reality, however, non-Gaussian signals are further divided into super-Gaussian and sub-Gaussian signals. Hardly are there studies dedicted to understanding the performance of non-Gaussian ICA algorithm at the much lower levels of super-Gaussian and sub-Gaussian levels and this study offers an insight in the performance of the algorithm at this deeper level. Therefore, the research goes deeper in non-Gaussian signals to understanding the working of the algorithm as the specified parameters on super-Gaussian and sub-Gaussian signals, that it, the thesis compares the performance of the NGA at these inner levels, and by analyizing the performance of two activation functions: Singmoid Activation Functions, which is commonly used, and the new and rarely used Fibonacci Activation Function. Therefore, while most studies are concerned with algorithm performance, the thesis is concerned with activation function within algorithms. The study offers insights of the performance of the NGA with the conclusion that the two activation functions are best classified as super-Gaussian activation functions , because the best seperation is of speech signals. In literature, the algorithm alongside activation function is simply classified as a non-Gaussian algorithm, yet this thesis has proved that the agorithm when used with the two activation functions should be classifies as a super-Gaussian algorithm.

5.2.2 Detailed Contribution

The ICA concept used in BSS problems has been used in research extensively. The outcome has been the coming up of many algorithms in trying to solve the different variant of the problem. Because of the many algorithms in place, recent studies have focused on finding out which among the existing algorithms is best for problems such as speech signals, biological/brain signals, music signals, among others. Studies that compare different algorithms include (Jianying, Cheng, Tianshu, & Bineng, 2016) (Mohanaprasad & Arulmozhivarman, 2013). In an effort to find the best algorithms, the studies have found out that beneath an algorithm there are cost functions that play a critical role in optimization and convergence for any given problem.

The well-known activation functions include maximum mutual information, maximum kurtosis, maximum negentropy, and maximum likelihood. For example, (Jianying, Cheng, Tianshu, & Bineng, 2016) compared cost functions including maximum mutual information, maximum kurtosis, maximum negentropy, and maximum likelihood with various algorithms to find out which of these cost functions combines well with which algorithm for which particular set of problem. In a 2018 thesis (Zhenyi , 2018), the researcher reviewed the best cost function for efficient estimation using Fast Fixed-Point Independent Component Analysis (FastICA) algorithm. While many studies compare algorithms, others are the middle level by comparing cost functions, this study goes deeper at the third layer to compare the activation function parameters. The efficiency of cost functions to optimize and converge properly depends, among other things, the following system parameters: the initial condition, the step-size value, and the estimation function, better known as the activation functions. So far among all the algorithms proposed for BSS-ICA problem, the one that does not bother or put into consideration the initial condition is the Natural Gradient Algorithm (NGA) (Amari, 1998). NGA is therefore chosen for this research because its cost function is not be affected with the initial condition.

Activations functions work beneath cost functions, and their analysis and performance has not been extensively studied as compared to algorithms. One critical, but often

ignored system parameter is the selection of a proper activation function (Basirat & Roth, 2018). A number of studies compare activations functions (Karlik & Olgac, 2011) (Dushkoff & Ptucha, 2016) (Hou, et al., 2016). For this study, the chosen activation function is Sigmoid Activation function, because it is widely used and most common in research (Ide & Saito, 2006) (Zhang L. , 2017), and specifically the use of sigmoid activation function in neural networks (Guliyev & Isailov, 2016) (Costarelli & Vinti, 2016) (Costarelli & Spigler, 2016) Then the sigmoid is compared to the new and promising Fibonacci Activation Function (Kyurkchiev & Iliev , 2017).

Therefore, the main aim of the research was to compare two activation functions: Sigmoid and Fibonacci within the natural gradient. The closest paper to this research uses Independent Component Analysis to separate three speech signals using fastICA algorithm and the activation function used is tanh (Parameswaran, Finitha, & Sama, 2010). Yet NGA is a supervisor algorithms for speech separation than fastICA, and sigmoid and Fibonacci are better activation functions for speech separation than tanh activation functions. Although the waveform output results from (Parameswaran, Finitha, & Sama, 2010), are of same quality as those realized from this research, this thesis goes further by comparing the frequency spectra, measuring the magnitude-squared coherence, and the correlation coefficient, besides comparing two activation functions pair of six signals that are both speech and music instrument signals. The conclusion from (Parameswaran, Finitha, & Sama, 2010) is that the algorithm works better for non-Gaussian signals, without elaborating further that the algorithms is actually meant for super-Gaussian signald; this research handles this shortcoming. alf as well. It is also clear from the research that the comparison of algorithms is not enough, because activation functions play a key role in algorithm. That the classification of signals should not be left at Gaussian and non-Gaussian level, rather non-Gaussianity should go farther to include super-Gaussian and sub-Gaussian signals in classifying activation functions and algorithms.

5.3 Limitations to the Research

Noise was a major limitation for this research. Throughout the research, it was assumed that noise was negligible. This is a wrong assumption as the noise may have entered the system during the recording stage of the speech signals. The already recorded signals may perhaps also have noise. The noise may have contributed to the inefficiencies of the quality of the separated signals when compared to the original signals. Although the quality of separation increases with the number of iterations used, the result of such higher number of iterations required to separate limits the application of Natural Gradient Algorithm in real life situations. The reason why the iterations were pegged at 100 is because 1000 or even 10,000 was too slow and in cases developing infinite numbers after the huge multiplied numbers during an iterate bring a number larger than what MATLAB can store. Further, the results after the hundredth iteration were the same, meaning convergence had been realized. The outcome of this is that its application in real time application still requires further research.

5.3.1 Recommendations for Future Research

Filtering of the noise to reduce or eliminate its presence in the generated outputs is an area for future research. Unlike the present research, that uses one level separation scheme, future research may need to incorporate two levels of signal separations. The first level will need to incorporate a filter to eliminate noise in the separated signals or just after recording them. The second level will do what this research has done in the blind source separation. The expanding use of genetic algorithm with a less complex activation function than Fibonacci are promising areas in Independent Component Analysis to cater for the real-time application bottlenecks exhibited in Natural Gradient Algorithm and the employed activation functions. Real-time processing of the signals using microphones and the use of convolved mixtures as opposed to instantaneous mixtures where time delay is not a factor are important future options for this research. For effective comparison of Fibonacci and Sigmoid activation functions for speech signals, the use of advanced concepts like wavelets and lyapunov

exponents is required. The future is towards having adaptive activations functions that will work for any type of signals (Dushkoff & Ptucha, 2016).

REFERENCES

Aapo, H., & Erkki, O. (2000). Independent Component Analysis: Algorithms and Applications . *Neural Networks, 13*(4-5), 411-430.

Amari, A., & Cichocki, A. (1998). Adaptive Blind Signal Processing – Neural network Approaches. *Proceeding of the IEEE, 86*(10).

Amari, S. (1998). Natural gradient works efficiently in learning. *Neural Computation, 10*, 251-276.

Amari, S.-i., Douglas, S. C., Cichocki, A., & Howard, Y. H. (1997). Multichannel Blind Deconvolution and Equalization using the Natural Gradient. *First IEEE Signal Processing Workshop on Signal Processing Advances in Wireless Communications.* Paris.

Barros, A. K., & Cichocki, A. (2001). Extraction of specific signals with temporal structures. *Neural Computation, 13*(9), 1995-2003.

Basirat, M., & Roth, P. M. (2018). The quest for the golden activation function. *Neural and Evolutionary Computing*. Retrieved from https://arxiv.org/pdf/1808.00783.pdf

Bell, A., & Sejnowski, T. (1995). An Information Maximization Approach to Blind Separation and Blind Deconvolution. *Neural Computation, 7*, 1129-1159.

Cardoso, J. F. (1998). Blind signal separation: statistical principles. *Proceedings of the IEEE.* Cambridge, MA.

Chan, T. H., Ma, W. K., Chi, C. Y., & Wang, Y. (2008). A covex analysis framework for blind sepration of non-negative sources. *IEEE Transitional on Signal Processing, 56*, 5120-5134.

Chibole, J. P. (2014). Blind separation of two human speech signals using natural gradient algorithm by employing the assumptions of independent component analysis. *Proceedings of the 2014 International Annual Conference on Sustainable Research and Innovation.* Nairobi.

Choi, S., Cichocki, A., & Beloucharni, A. (2001). Second order nonstationary source seperation . *Journal of VLSI Signal Processing*, 1-13.

Cichocki, A., & Amari, S.-i. (2002). *Adaptive Blind Signal and Image Processing: learning Algorithms and Applications.* New York: John Wiley & Sons.

Costarelli, D., & Spigler, R. (2016). Solving numerically nonlinear systems of balance laws by multivariate sigmoidal functions approximation. *Computational and Applied Mathematics* . doi:10.1007/s40314-016-0334-8

Costarelli, D., & Vinti, G. (2016). Pointwise and uniform approximation by multivariate neural network operators of the max-product type Neural Networks . *Neural Networks*. doi:10.1016/j.neunet.2016.06.002

Cover, T. M., & Thomas, J. A. (1991). , *Elements of Information Theory, .* New York: JohnWiley & Sons .

Cristescu, R., Ristaniemi, T., Joutsensalo, J., & Karhunem, J. (2000). Cdma delay estimation using fast ica algorithm. *Procceedings of IEEE Inteernational Symposium on Personl, Indoor and Mobile Communication, 2*, 1117-1120. California.

Daqarta for DOS. (n.d.). *Fourier Transform Theory*. Retrieved from http://www.daqarta.com/0t0ifft1.htm

de-Frein, R., & Richard, S. (2011). The sychronised short-time-Fourier-transform: properties and definitions for multichannel source seperation. *IEEE Transitional on Signal Processing, 59*, 91-103.

Douglas, S. C., & Gupta, M. (2007). *Convolutive blind source separation for audio signals.* (S. Maino, P. Okuyayi, J. Tawwoi, N. OscN, n. JiouN, p. Juz, & K. Yusufkeukeu, Eds.) New York: New York Springer.

Douglas, S. C., & Haykin, S. (2000). Relationships between blind deconvolution and blind source separation. In S. Haykin (Ed.), *Unsupervised Adaptive Filtering, Vol. II: Blind Deconvolution* (pp. 113–145). New York: JohnWiley & Sons.

Dubnov, S., Tabrikian, J., & Targan, A. M. (2006). Speech source separation in convolutive environments using space-time-frequency analysis. *EURASIP Journal of Advances in Signal Processing, 2006*(1).

Dushkoff, M., & Ptucha, R. (2016). Adaptive activation functions for deep networks. *Electronic Imaging, 2016*(19), 1-5.

Guliyev, N., & Isailov, V. (2016). A single hidden layer feedforward network with only one neuron in the hidden layer san approximate any univariate function. *Neural Computation, 28*, 1289-1304.

Hosseini, S., Daville, Y., & Saylani, H. (2009). Blind seperation of linear instantaneous mixtures of non-stationary signals in the frequency domain. *Signal Processing, 89*(5), 8119-8130.

Hou, L., Samaras, D., Kurc , T. M., Tahsin, M., Gao, Y., & Saltz, J. H. (2016). Neural networks with smooth adaptive activation functions for regression. *arXiv preprintarXiv :1608.06557*.

Hujun, Y., & Hussain, I. (2008). Independent Component Analysis and non gaussianity for blind image deconvolution and bluring. *Integrated Computer-Aided Engineering, 15*, 219-228.

Ide, A. N., & Saito, J. H. (2006). Chapter 7 Field Programable Gate Array (FPGA). In *FPGA Implementations of Neural Network* (pp. 197-222). New York: Springer.

Jalobeanu, A., Feraud, B. L., & Zerubia, J. (2004). An adaptive Gaussian model for satellite image deblurring. *IEEE Trans. Image Processing, 13*, 613-621.

Jianying, W., Cheng, W., Tianshu, Z., & Bineng, Z. (2016). Comparison of different Independent Component Analysis Algorithms for output-only modal analysis. *Shock and Vibrations*, 1-25. doi:10.1155/2016/6309084

Karlik, B., & Olgac, V. A. (2011). Performance analysis of various activation functions in generalized MLP architectures of neural networks. *International Journal of Artificial Intelligence and Expert Systems (IJAE), 1*(4), 111-122.

Kofidisa, N., Margarisa, A., Diamantarasb, K., & Roumeliotis, M. (2008). Blind system identification: instantaneous mixtures of n sources. *International Journal of Computer Mathematics, 85*(9), 1333–1340.

Kumar, D. K., & Naik, G. R. (2011). An overview of Independent Component Analysis and its Application. *Informatica, 35*, 63-81.

Kyurkchiev , N., & Iliev , A. (2017). A Note on the New Fibonacci Hyperbolic Tangent Activation Function. *International Journal of Innovative Science, Engineering & Technology, 4*(5).

Lahat, D., Cardoso, J.-F., & Messer, H. (2012). Second-order multidimensional ICA: Performance Analysis. *IEEE Transactions on Signal Processing, 60*(9), 4598-4610.

Lecumberri, P., Gómez, M., & Carlosena, A. (2005). Multichannel Blind Deconvolution of Impulsive Signals. *Proceedings of the 13 European Signal Processing Conference.* Antalya, Turkey.

Lecumberri, P., Gómez, M., & Carlosena, A. (2005). Multichannel Blind Deconvolution of Impulsive Signals . *Proceedings of the 13 European Signal Processing Conference.* Antalya.

Lee, S. H. (2001). Algorithms for non-negative matrix factorization. *Advanced Neural Information Processing System, 13*, 556-562.

Lee, S., Shen, H., Troung, Y., Lewis, M., & Huang, X. (2011). Independent Component Analysis involving autocorrelated sources with an application to functional magnetic resonance imaging. *Journal of the American Statistical Association, 106*(495), 1009-1024.

Mathsworks. (2014). *Spectral Analysis*. (MATLAB) Retrieved July 10, 2014, from http://www.mathworks.com/help/signal/ug/spectral-analysis.html

Matteson, D. S., & Tsay, R. S. (2016). Independent Component Analysis via Distance covariance. *Journal of the American Statistical Association*, 1-36.

McKeown, M. J., Makeig, S., Brown, G. G., Jung, T. P., Kindermann, S. S., Bell, A. J., & Sejnowski, T. J. (1998). Analysis of fMRI data by blind separation into independent spatial components . *Human Brain Mapping, 6*, 160–188.

Meyer, C. D. (2000). *Matrix Analysis and Applied Linear Algebra.* Cambridge, UK: Cambridge University Press.

Mohanaprasad , K., & Arulmozhivarman, P. (2013). Comparison of Fast ICA and Gradient Algorithms of Independent Component Analysis for Separation of Speech Signals. *International Journal of Engineering and Technology (IJET), 5*(4), 3196-3202.

Moon, J., & Hong, H. S. (2014). What is temporal fine structure and why it is important? *Korean Journal of Audiology, 18*(1), 1-7.

Myers, M. E., van de Geijn, P. M., & van de Geijn, R. A. (2015). *Linear Algebra: Foundations to Frontiers.* New York: LAFF Group.

Naik, G. R. (2012). Measure of Quality of Sound Seperation for Sub and Super Gaussian Audio Mixtures. *INFORMATICA, 23*(4), 581-599.

Nielsen, M. (2015). *Neural Networks and Deep learning.* New York: Determination Press. Retrieved from http://neuralnetworksanddeeplearning.com/chap1.html

Parameswaran, S., Finitha, K. C., & Sama, A. A. (2010). Blind Source Separation for Speech Signals. *IIST*, 4 - 9.

Paulraj, A. J., & Papadias, C. B. (1997). Space-time processing for wireless communicationsl. *IEEE Signal Processing Magazine, 14*(6), 49–83.

Peng, J. Y., & Yang, Y. (2010). Blind Source Separation Based on Improved Natural Gradient Algorithm . *Proceedings of the 8th World Congress on Intelligent Control and Automation, July 6-9.* Jinan, China.

Rai, C. S., & Yogesh, S. (2004). Source distribution models for blind source seperation. *Neurocomputing, 57*, 501-502. doi:10.1016/j.neucom.2004.01.003

Saduf, M. A. (2013). Comparative Study of Back Propagation Learning Algorithms for Neural Networks. *International Journal of Advanced Research in Computer Science and Software Engineering, 3*(12), 1151-1156.

Scott, D. C. (2002).Blind Signal Seperation and Blind Deconvolution. New York: CRC Press LLC.

Tharwat, A. (2018). Independent component analysis: An introduction. *Applied Computing and Informatics*, 1-15. doi:10.1016/j.aci.2018.08.006

Torkkola, K. (2000). Blind separation of delayed and convolved sources. In S. Haykin (Ed.), *Unsupervised Adaptive Filtering, Vol. I: Blind Signal Separation* (pp. 321-375). New York: JohnWiley & Sons.

Übeyl, E. D., & Guler, I. (2007). Statistics over Lyapunov Exponents for Feature Extraction: Electroencephalographic Changes Detection Case . *International Journal of Medical, Health, Biomedical, Bioengineering and Pharmaceutical Engineering , 1*(2), 134-137.

Vrins, F., & Verleysen, M. (2005). On the entropy minimization of a linear mixture of variables for source separation. *UCL, Electrical Engineering Department, Microelectronics Laboratory. Signal Processing , 85*, 1029 -1044.

Zhang, H., Cai, G.-W. P., & Ding, S. Z. (2014). A fast blind source seperation algorithm based on the temporal structure of signals. *Neurocomputing*, 261-271.

Zhang, L. (2017). Implementation of Fixed-point Neuron Models with Threshold, Ramp and Sigmoid Activation Functions. *4th International Conference on Mechanics and Mechatronics Research (ICMMR 2017): IOP Conference Series: Materials Science and Engineering.* Ottawa. doi:10.1088/1757-899X/224/1/012054

Zhang, L. Q., Cichoki, A., & Amari, S. (2004). Self-adaptive blind source separation based on activation functions adaptation. *IEEE Trans. Neural Networks, 15*(2), 233-244.

Zhang, L., Amari, S., & Cichocki, A. (2001). Equi-convergence Algorithm for blind separation of sources with arbitrary distributions. *6th Int'l Work Conf. on Artificial and Natural Neural Networks*, 826-833.

Zhang, L., Cichocki, A., & Amari, S. (2004, March). Self-Adaptive Blind Source Separation Based on Activation Function Adaptation. *IEEE Transactionon Neural Networks, 15*(2), 1-12.

Zhenyi, A. (2018). *Different Estimation Methods for the Basic Independent Component Analysis Model.* Washington: St. Louis: Washington University: Arts & Sciences Electronic Theses and Dissertations.

APPENDICES

7.1 Appendix I: Independence and Non-Gaussianity

Mathematically, independence is a statistical concept that has direct bearing on probability density functions of the two signals. Joint probability density function (pdf) of the two signals s_1 and s_2 is defined mathematically as $p(s_1, s_2)$. On the other hand, the marginal pdf of the same signals s_1 and s_2 are defined as $p_1(s_1)$ and $p_2(s_2)$ respectively. Therefore, s_1 and s_2 can only qualify to be independent if and only if their joint and marginal pdfs are expressed as:

$$p(s_1, s_2) = p_1(s_1)\, p_2(s_2) \tag{7.1}$$

It is important to note that the same independence can be defined in terms of respective cumulative distributive function by substituting the cumulative distributive function in place of pdfs as:

$$E\{p(s_1)p(s_2)\} = E\{g_1(s_1)\}E\{g_1(s_2)\} \tag{7.2}$$

where $E\{.\}$ is the expectation operator (Kumar & Naik, 2011).

Signal independence can be measured using the mathematical properties of indices. Using matrices, it is possible to check for the linear independency and dependency of signals using global matrices (Permutation matrix *P*) (Kumar & Naik, 2011).

Rank of the Matrix

For linear dependency, the rank of the matrix is usually less than the size of the matrix, while for linear independency, the rank is the size of the matrix (Meyer, 2000). However, this could not be assured because of the involved noise in the signal. This means that in order to determine the original number of sources, we only need to compute the determinant of the matrix.

Determinant of the Matrix

In many applications, the value of Determinant is always more than zero (and always close to one) for linear independency and its value should be zero for linear dependency (Meyer, 2000).

*Cross-correlation and Independenc*e

The two signal variables s_1 and s_2 are termed as uncorrelated if they have a covariance $C(s_1, s_1)$ equal to zero.

$$C(s_1, s_1) = E\{(s_1 - m_{s^1})(s_2 - m_{2^2})\}$$

$$= E\{(s_1 s_2 - s_1 m_{s^2} - s_2 m_{s^1} + m_{s^1} m_{s^2})\}$$

$$= E\{s_1 s_2\} - E\{s_1\}E\{s_2\} \quad (7.3)$$

in this case m_{s^1} and m_{s^2} represent the mean of s_1 ands_2 signals, respectively. From Equations (7.2) and (7.3), it is clearly shown that the signals are identical for independent variables taking $g_1(s_1) = s_1$. This mathematical test shows that independent signals are always uncorrelated. However, it is also important to mention that the opposite of the above statement is not always correct. At the preprocessing stage of signals, and as evidenced from the above analysis, signal independence is of much stronger and of more value than uncorrelatedness in the separation process of two or more signals. This is clearly the reason why independence is given more predominance in ICA source estimation techniques than correlation. However, it should be noted that uncorrelatedness plays a crucial role in computing the mixing matrix in ICA.

7.1.1 Non-Gaussianity and Independence

According to the mathematical theory of central limit theory, under some conditions, the distribution of a sum of independent signals and those of arbitrary distribution are always inclined towards a Gaussian distribution. What this means is that given, for example, two independent signals, their distribution is towards Gaussian than the two

original signal distributions. This analysis leads to the conclusion that a Gaussian signal can be considered as a linear combination of several independent signals (Kumar & Naik, 2011). This statement is the basis of separation possibilities of independent signals from a Gaussian mixture. It simply means that the separation of independent signals from their mixtures can be accomplished through transformation of the linear signals to make them non-Gaussian, as much as possible.

ICA estimation and separation algorithms including NGA depend entirely on non-Gaussianity as a key principle. In order to use non-Gaussianity in ICA estimation, it is important that quantitative measure of non-Gaussianity of a signal be achieved. However, before employing any measure of non-Gaussianity, the signal must be normalized. Normalization is a preprocessing technique that will be discussed later in this section. Non-Gaussianity of a signal is determined in two ways; through the measure of kurtosis or through entropy measure.

Kurtosis

For many years, non-Gaussianity of signals was measured using Kurtosis. In the preprocessing procedure, data is made to have unit variance, and then kurtosis, which is usually equivalent to the fourth moment of the data, is done. Signal(s) Kurtosis, denoted as kurt(s), is defined as:

$$kurt(s) = E\{s^4\} - 3(E\{s^4\})^2 \tag{7.4}$$

Equation (7.4) is the most basic definition of kurtosis using higher order (fourth order) cumulant. The above simplification of kurtosis is based on the assumption that the signal has zero mean. To make equation (3.6) simpler, it can further be assumed that (s) has been normalized so that its variance is equal to one, that is: $E\{s^2\} = 1$. Therefore, equation (7.4) can be simplified to:

$$kurt(s) = E\{s^4\} - 3 \tag{7.5}$$

Equation (7.5) shows that kurtosis is a normalized form of the fourth moment $E\{s^4\} = 1$. For signals that are Gaussian, $E\{s^4\} - 3(E\{s^4\})^2$ and hence its kurtosis is zero. For signals that are non-Gaussian, the kurtosis is always nonzero. There is positive kurtosis or negative kurtosis. Random variables having positive kurtosis are termed as *super Gaussian* or *platykurtotic* (Kumar & Naik, 2011). Therefore, non-Gaussianity is measured using the absolute values of kurtosis or the square of kurtosis. Kurtosis has been extensively used to measure Non-Gaussianity in ICA and other fields of science and engineering because it is a simple theory to compute. Mathematically, it has a linearity property such that:

$$kurt(s_1 \pm s_2) = kurt(s_1) \pm kurt(s_2) \tag{7.6}$$

and

$$kurt(\alpha s_1) = \alpha^4 kurt(s_1) \tag{7.7}$$

where α is a constant. Kurtosis can be computed using the fourth moment of the sample data by ensuring that the variance of the signal is constant.

In simpler terms, kurtosis measures the "spikiness" of a distribution or in other words, the size of the tails. Kurtosis is simple to calculate, however, its key disadvantage is that it is extremely sensitive to outliers in the data set (Kumar & Naik, 2011). It is regarded as having poor statistical significance, because its calculated outcomes may be based on a few values in the tails. Kurtosis is, therefore, not robust enough for ICA. A better measure of non-Gaussianity than kurtosis in ICA algorithms is entropy.

Entropy

Entropy is used to measure the uniformity of a distribution of a bounded set of values (Kumar & Naik, 2011). This means that when a distribution has complete uniformity, then it is regarded as having maximum entropy. From the information theory point of view, entropy is taken to mean the measure of signal randomness. Entropy H of discrete-valued signal S is defined as

$$H(S) = -\sum P(s = a_i)\, logP(s = a_i) \tag{7.8}$$

This definition of entropy can be generalized for a continuous-valued signal (s) called differential entropy, and is defined as:

$$H(S) = -\int p(s) logp(s) ds \tag{7.9}$$

Among all types of signals that have unit variance, it is the Gaussian signals that have the largest entropy. Entropy is usually small among signals that have a distribution associated to certain values, or those with extremely "spiky" pdf (Kumar & Naik, 2011). This is the main reason why entropy stands out to be a good measure of non-Gaussianity.

In order to make the computation of separating relatively simply, non-Gaussian and Gaussian signals must have a measure of nonzero and zero respectively. Random vector's code length has a close link to entropy (Kumar & Naik, 2011). Normalization of entropy is also possible and is done using a new measure called Negentropy J, expressed mathematically as:

$$J(S) = H(s_{gauss}) - H(s) \tag{7.10}$$

Where s_{gauss} is the Gaussian signal with an equivalent covariance matrix as s. From equation (7.10), Negentropy takes a positive value and it is only zero in case of a signal that is purely Gaussian. Although Negentropy is always stable, it is actually extremely difficult to calculate. This is the reason why entropy is only computed using approximated values.

7.2 Appendix II: Centering and Whitening

7.2.1 Centering

The preprocessing step carried out to "centre" the observation signal vector x is accomplished by simply subtracting its mean vector $m = E\{x\}$. The observed and centered vector x_c is shown as:

$$x_c = x - m \tag{7.11}$$

The purpose of centering is to ensure that the ICA algorithm work is simplified by assuming a zero mean (Kumar & Naik, 2011). After the de-mixing matrix is estimated using the centered values, it is possible to get back to the actual estimates of the independent components in the following manner:

$$\hat{s}(t) = A^{-1}(x_c + m) \tag{7.12}$$

From this point onwards, all observed vectors are assumed to be centered. However, the mixing matrix remains the same before and after the preprocessing stage as centering of the observations does not affect the estimated mixing matrix (Kumar & Naik, 2011).

7.2.2 Whitening

Another preprocessing step done on the mixture is to whiten the observed vector x. This process entails transforming the observation vector linearly to make sure that each component does not correlate to each other by having a unit variance (Meyer, 2000). Let x_w denote the whitened observed vector, then it will satisfy the equation:

$$E\{x_w x_w^T\} = I \tag{7.13}$$

where $E\{x_w x_w^T\}$ is the covariance matrix of x_w. Since the ICA framework is not affected by the variances of the independent components, it is possible to assume without loss of generality that the source vector, s is white, that is $E\{ss^T\} = I$. The

simplest way to carry out the whitening transformation is to use of eigenvalue decomposition (EVD) (Meyer, 2000) of x. This is done by the decomposition of the covariance matrix of x as follows:

$$E\{xx^T\} = \mathrm{VDV^T} \tag{7.14}$$

where V is the matrix of the eigenvector of $E\{xx^T\}$, and D is the diagonal matrix of the eigenvalues, that is, $D = diag\{\lambda_1, \lambda_2, \dots, \lambda_n\}$. The observed vector can, therefore, be whitened by the following transformation:

$$x_w = VD^{-1/2}V^T x \tag{7.15}$$

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

$$x_w = VD^{-1/2}V^T As = A_w s \tag{7.16}$$

hence,

$$\begin{aligned} E\{x_w x_w^T\} &= A_w E\{ss^T\}A_w^T \\ &= A_w A_w^T \\ &= I \end{aligned} \tag{7.17}$$

The purpose of whitening is to reduce the number of parameters to be estimated. Instead of having to estimate the n^2 elements of the original matrix, A, whitening reduces it to only the estimation of the new orthogonal mixing matrix. An orthogonal matrix has $n(n-1)/2$ degree freedom. Through whitening, half of the ICA problem is solved as it greatly reduces the complex computations within ICA (Kumar & Naik, 2011). MATLAB has functions that carry out whitening and centering of mixtures.