Institutionen för systemteknik
Department of Electrical Engineering
Examensarbete

STUDY OF ASA ALGORITHMS

MASTER THESIS

BY
NAGARAJU ARDAM

LITH-ISY-EX--10/4334--SE

Linköping, 2010.

TEKNISKA HÖGSKOLAN
LINKÖPINGS UNIVERSITET
INSTITUTIONEN FÖR SYSTEMTEKNIK

STUDY OF ASA ALGORITHMS

EXAMENSARBETE UTFÖRT I ELEKTRONIKSYSTEM
VID TEKNISKA HÖGSKOLAN I LINKÖPING

AV

NAGARAJU ARDAM

(e-mail-nagar975@student.liu.se)

LITH-isy-ex--10/4334--SE

Linköping, May 24, 2011

Handledare: J Jacob Wikner

Examinator: J Jacob Wikner

ISY, Linköpings universitet

ISY, Linköpings Universitet
Study of ASA algorithms

Abstract
Hearing aid devices are used to help people with hearing impairment. The number of people that requires hearing aid devices are possibly constant over the years, however the number of people that now have access to hearing aid devices increasing rapidly. The hearing aid devices must be small, consume very little power, and be fairly accurate. Even though it is normally more important for the user that hearing impairment look good (are discrete). Once the hearing aid device prescribed to the user, she/he needs to train and adjust the device to compensate for the individual impairment.

We are within the framework of this project researching on hearing aid devices that can be trained by the hearing impaired person her-/himself. This project is about finding suitable noise cancellation algorithm for the hearing-aid device. We consider several types of algorithms like, microphone array signal processing, Independent Component Analysis (ICA) based on double microphone called Blind Source Separation (BSS) and DRNPE algorithm.

We run this current and most sophisticated and robust algorithms in certain noise backgrounds like Cocktail noise, street, public places, train, babble situations to test the efficiency. The BSS algorithm was well in some situation and gave average results in some situations. Where one microphone gave steady results in all situations. The output is good enough to listen targeted audio.

The functionality and performance of the proposed algorithm is evaluated with different non-stationary noise backgrounds. From the performance results it can be concluded that, by using the proposed algorithm we are able to reduce the noise to certain level. SNR, system delay, minimum error and audio perception are the vital parameters considered to evaluate the performance of algorithms. Based on these parameters an algorithm is suggested for hearing-aid.

Nyckelord (Key words)
Auditory scene analysis (ASA), Noise suppression, ICA, Blind source suppression, Hearing aid.
ABSTRACT

Hearing aid devices are used to help people with hearing impairment. The number of people that requires hearing aid devices are possibly constant over the years, however the number of people that now have access to hearing aid devices increasing rapidly. The hearing aid devices must be small, consume very little power, and be fairly accurate. Even though it is normally more important for the user that hearing impairment look good (are discrete). Once the hearing aid device prescribed to the user, she/he needs to train and adjust the device to compensate for the individual impairment.

We are within the framework of this project researching on hearing aid devices that can be trained by the hearing impaired person her-/himself. This project is about finding suitable noise cancellation algorithm for the hearing-aid device. We consider several types of algorithms like, microphone array signal processing, Independent Component Analysis (ICA) based on double microphone called Blind Source Separation (BSS) and DRNPE algorithm.

The idea behind microphone array signal processing is to extract the required information from acoustic waves received by multiple microphones located at several places of targeted area. Developing an algorithm for microphone array processing is difficult because it has to be re-developed based on the target areas. It contains lot of tricky mathematical calculations and also required to write hundreds of lines of code for implementation according to the targeted areas which would lead to overhead on portable hearing-aid.

The main idea behind the ICA is “Retrieving unobserved signals or sources from observed linearly mixed signals based on the assumption that these signals are mutually independent”. The limitation of ICA is that the sources should be statistically independent. This works very fine in all areas, but it contains lot of mathematics. According to the study, the BSS algorithm does not perform well in all targeted areas and it is not giving steady results in all noisy backgrounds.

The aim of single-microphone noise suppression algorithms is to reduce the noise as much as possible. Unfortunately it is impossible to completely remove all the noise. One reason is that with one microphone you cannot distinguish between speech and noise on the basis of where the sounds come from, for example, while some extent this is possible when more microphones are available.

Basically what the algorithm does is chop up the microphone signal in short intervals (called 'frames') and look at the frequency content in each frame. The part of the signal that has a fairly slowly changing spectrum over time is assumed to be the noise, while speech is assumed to change more rapidly. This is used to estimate the average noise spectrum.

In the noise suppression step, the frequencies with lots of noise are attenuated. So the idea is to apply relatively more attenuation to the frequencies with low signal-to-noise ratio. Unfortunately this means that if there is some speech present at those frequencies it will be attenuated as well. If you want to completely remove the noise, you will get a lot of speech distortion, and the algorithm has to balance this by letting some of the noise pass through.

Next, We run this algorithm in certain noise backgrounds like Cocktail noise, street, public places, train, babble situations to test the efficiency. The BSS algorithm was well in some situation and gave average results in some situations. Where one microphone gave steady results in all situations. The output is good enough to listen targeted audio.

The functionality and performance of the proposed algorithm is evaluated with different non-stationary noise backgrounds. From the performance results it can be concluded that, by using the proposed algorithm we are able to reduce the noise to certain level. SNR, system delay, minimum error and audio perception are the vital parameters considered to evaluate the performance of algorithms. Based on these parameters an algorithm is suggested for hearing-aid.
ACKNOWLEDGEMENT

I am heartily thankful to my supervisor Dr. Jacob Wikner for his constant valuable support, encouragement and guidance since the initial to final level which allowed me to understand the subject. He has made his support available in a number of ways, Thanks for being friendly with me. This thesis would not have been possible without his support.

I also would like to thank Manildev for his support during the project and documentation. Lastly, I offer my regards to my family for their affection and support and to my friends also those who supported me in any respect during the completion of the project.
TABLE OF CONTENTS

1. Introduction .................................................................................................................................................. 11
   1.1. Background ........................................................................................................................................... 11
   1.2. Microphone array processing for noise extraction .............................................................................. 12
   1.3. Independent Component Analysis (ICA) ............................................................................................. 16
   1.4. Single microphone for non-stationary noise .......................................................................................... 16
   1.5. Thesis scope ......................................................................................................................................... 16
   1.6. Methodology ....................................................................................................................................... 18
   1.7. Organization ....................................................................................................................................... 18
   1.8. Definitions .......................................................................................................................................... 20

2. Independent Component Analysis (ICA) theory .......................................................................................... 21
   2.1. Methodology ....................................................................................................................................... 21
   2.2. ICA by equation ................................................................................................................................... 26
   2.3. Equivocalness of ICA .......................................................................................................................... 28
   2.4. ICA properties ..................................................................................................................................... 30
   2.5. Pre-processing ..................................................................................................................................... 32
      2.5.1. Centering ....................................................................................................................................... 32
      2.5.2. Whitening ..................................................................................................................................... 32
      2.5.3. Further preprocessing .................................................................................................................. 36

3. Simulation and performance of CoBlISS ................................................................................................. 37
   3.1. Introduction ....................................................................................................................................... 37
   3.2. BSS algorithm ..................................................................................................................................... 37
      3.2.1. Notations ...................................................................................................................................... 37
      3.2.2. Optimization criterion .................................................................................................................. 38
      3.2.3. Frequency domain approach ....................................................................................................... 42
      3.2.4. Convolution constraint ................................................................................................................. 44
      3.2.5. Time-frequency compliance ....................................................................................................... 46
      3.2.6. Exact weight update ..................................................................................................................... 46
      3.2.7. Approximated weight update ....................................................................................................... 48
      3.2.8. Normalization ............................................................................................................................... 48
   3.3. CoBlISS algorithm flow ....................................................................................................................... 50
   3.4. Simulation and results .......................................................................................................................... 52

4. Source separation from single source ......................................................................................................... 57
   4.1. One microphone sources separation technique ....................................................................................... 57
   4.2. Spectral modeling ............................................................................................................................... 58
   4.3. Amplitude estimation .......................................................................................................................... 58
      4.3.1. Minima Statistics (MS) method ....................................................................................................... 58
      4.3.2. Improved Minima Controlled Recursive Averaging (IMCRA) method .......................................... 60
      4.3.3. SKS method (Sugiyama. Kato. Serizawa) ..................................................................................... 60
   4.4. MMSE estimation of noise power to reduce speech leakage .................................................................. 62
      4.4.1. Speech probability estimation ...................................................................................................... 64
      4.4.2. Prior SNR estimation ..................................................................................................................... 64
      4.4.3. Safety net ...................................................................................................................................... 66
      4.4.4. Gain function for noise estimation ............................................................................................... 68
   4.5. Simulation and results ........................................................................................................................... 72

5. Comparison of performance ....................................................................................................................... 77
   5.1. Compiled results under various circumstances .................................................................................... 104

6. Suggested future work .................................................................................................................................. 107
INDEX OF TABLES

Table 1: List of Acronyms ................................................................................................................................................. 10
Table 3.1: Results after applying the BSS algorithm on noisy source .................................................................................. 56
Table 4.1: Results obtained after applying the single-microphone algorithm on noisy source .............................................. 74
Table 5.1: Results after applying the BSS algorithm on speech contaminated by a music ...................................................... 84
Table 5.2: Results obtained after applying BSS on cross talk environment ............................................................................. 90
Table 5.3: Results obtained after applying DRNPE based on single microphone algorithm on speech contaminated by music ........................................................................................................ 98
Table 5.4: Results obtained after applying the DRNPE based on single microphone on cross talk ...................................... 100
Table 5.5: Results after applying the DRNPE based on single microphone with train noise including speech ......................................................... 102
Table 5.6: Test results with babble noise using DRNPE based on single microphone ............................................................. 102
Table 5.7: Test results with exhibition noise using DRNPE based on single microphone ....................................................... 102
Table 5.8: Test results with street noise using DRNPE based on single microphone algorithm .................................................. 102
Table 5.9: SNR results for different background noise sources .................................................................................................. 104
Table 5.10: Minimum Error with different noise sources .......................................................................................................... 104
Table 5.11: System delay results with various noisy sources .................................................................................................. 106
### LIST OF ACRONYMS

<table>
<thead>
<tr>
<th>Abb.</th>
<th>Explanation</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASA</td>
<td>Auditory Scene Analysis</td>
<td>Process to organize sound in to meaningful elements.</td>
</tr>
<tr>
<td>BSS</td>
<td>Blind Source Separation</td>
<td>Recovering independent sources from sensor observations that are linearly independent source signals. The name Blind indicates that the way the source signals are mixed together is unknown.</td>
</tr>
<tr>
<td>CoBliss</td>
<td>Convolutive Blind source separation</td>
<td>Combination of blind source separation and acoustic echo canceling.</td>
</tr>
<tr>
<td>Cocktail-party</td>
<td>Type of noise</td>
<td>Mixture of required speech with music and other speech.</td>
</tr>
<tr>
<td>DFT</td>
<td>Discrete Fourier Transform</td>
<td>Transforming a input function in to frequency domain, input should be finite.</td>
</tr>
<tr>
<td>DRNPE</td>
<td>Data Driven Recursive Noise Power Estimation</td>
<td>A method to implement an algorithm based on single microphone</td>
</tr>
<tr>
<td>EEG</td>
<td>Electroencephalogram</td>
<td>Measurement of electrical activity in the brain.</td>
</tr>
<tr>
<td>FIR</td>
<td>Finite Impulse Response</td>
<td>A filter of a kind whose impulse response reaches to zero in a finite duration.</td>
</tr>
<tr>
<td>HOS</td>
<td>Higher order statistics</td>
<td></td>
</tr>
<tr>
<td>ICA</td>
<td>Independent Component Analysis</td>
<td>Method to implement BSS algorithm to suppress the noise during the auditory scene analysis.</td>
</tr>
<tr>
<td>IDEA</td>
<td>International Dialect for English Archive</td>
<td>Free, online archive of primary source dialect and accent recordings.</td>
</tr>
<tr>
<td>IMCRA</td>
<td>Improved Minima Controlled Recursive Averaging</td>
<td>The idea is to estimate the noise variance $\lambda_D$ by recursive smoothing of the noisy power.</td>
</tr>
<tr>
<td>MMSE</td>
<td>Minimum mean square error</td>
<td>An estimator which manages in minimizing the error.</td>
</tr>
<tr>
<td>MS</td>
<td>Minimum Statistics</td>
<td>The minima statistics method uses the minima of the smoothed periodogram of the noisy speech to estimate the noisy level on each frequency bin.</td>
</tr>
<tr>
<td>PDF</td>
<td>Probability Density Function</td>
<td>A function to estimate the chance of random variable to occur at a given point.</td>
</tr>
<tr>
<td>SKS method</td>
<td>Sugiyama.Kato.Serizawa</td>
<td>The noisy power $R(k,m)$ is weighted by factor $W(k,m)$ that depends on the posterior SNR $\left(\zeta(k,m)\right)$.</td>
</tr>
<tr>
<td>SOS</td>
<td>Second order statistics</td>
<td></td>
</tr>
<tr>
<td>SNR</td>
<td>Signal-to-Noise Ratio</td>
<td>Measurement used to estimate how much a signal is corrupted by noise.</td>
</tr>
<tr>
<td>VAD</td>
<td>Voice Activity Detector</td>
<td>Detects a voice over signal.</td>
</tr>
</tbody>
</table>

Table 1: List of Acronyms
1. INTRODUCTION

Auditory scene analysis is a process by human auditory system where it recovers individual
descriptions of individual sounds from a mixture of sounds and the purpose of auditory scene algorithm
is to reduce or remove all the unwanted surrounded noise of a speech to make the speech in to
meaningful elements. The purpose of implementing the ASA algorithm is for the people whose hearing
is poor. The motivation behind this thesis to suggest a suitable algorithm to implement a hearing aid
device. This chapter briefly describes the background and the number of methods to implement an ASA
algorithms.

1.1. Background

Hearing aid devices are used to help people with hearing impairment. The number of people that
requires hearing aid devices are possibly constant over the years, however the number of people that
now have access to hearing aid devices increasing rapidly. The hearing aid devices must be small,
consume very little power, and be fairly accurate. We are within the framework of this project
researching on hearing aid devices that can be trained by the hearing impaired person by
herself/himself. This must be achieved at the same time as we can guarantee low power and low
area/weight.

Imagine that you are in a party and you are surrounded by a of people speaking loudly and also a nice
heavy music going on. But you are only trying to focus on conversation with your friend, your auditory
system in side your ear trying to filter out unwanted sounds. Our auditory system can manage to focus
on required sound. Imagine one more situation that you are in railway station or airport and waiting for
an announcement for your train or plane departure gate number and you are surrounded by non
stationary noise. If the announcement starts, our hearing system can give attention to a required sound
irrespective of surrounded noisy sounds. Now that is the analyzing of auditory scene as for the
requirements by our hearing system, so called Auditory Scene Analysis (ASA). It is a process by a
human auditory system which organizes the mixture of sounds in to descriptive individual sounds.

In the above situations, One do not required any hearing aid if his/her auditory system works perfectly
or at least good, what if one can not hear properly?. This can be solved by using hearing aid, which
contains same environment like auditory scene analysis. This thesis work explains the methods of
noise filter algorithm and suggests the suitable algorithm for portable hearing-aid.

Since decades research is going on ASA algorithms where separating the required speech from
unwanted sounds. Many algorithms were proposed and only few of them can survive in any
environment while giving steady results. There are several ways to deal with this issue like Microphone
Array, BSS by Independent Component Analysis (ICA), DRNPE based on single microphone. Our main
intention is on single microphone method and the reasons to choose this one was discussed in further
chapters in depth. During the thesis work over ASA algorithms few algorithms are giving steady results
based on the category, this thesis work explains some of them.

This report contains the details regarding how the noise cancellation is done by using above methods.
This thesis work will deal with them thoroughly to suggest a best one in them as per requirements.
Figure 1.1 shows the block diagram of auditory scene analysis for cock-tail party noise problem, where
you are trying to focus on some speaker under the heavy noise circumstances like music, people’s conversation.

1.2. Microphone array processing for noise extraction

Microphone array processing is a method to extract the required information from acoustic waves received by multiple microphones located at several places of targeted area. Figure 1.2 demonstrates the microphone array processing in a room, where multiple microphones are located in a noisy background. Due to the non-stationary, random and broad band speech and also circumstances of targeted area like play ground, auditorium, room, forest makes microphone array signal processing more complicated. Developing an algorithm for microphone array processing is difficult because it has to be re-developed based on the target areas like above stated.
To solve the problem demonstrated in Figure 1.2 and to get the required speech we have to solve the following, noise reduction, echo reduction, cocktail party, estimation of number of sources, localization of multiple sources, location of a single source, source separation and de-reverberation. Many of the above issues are solved by passing \( Z(a) \) through some filters which have to be optimized according to the problem. The advantage of microphone array processing is, it does not distort the original speech signal. The problem behind the algorithm is, as we are interested in implementation of algorithm for hearing aid which should be portable in size, as user can not carry multiple microphones. We are not interested in this method. If the reader want to know further details author suggests to follow the book dedicated for microphone array processing [1], which gives detailed information regarding microphone array processing.

Figure 1.2: Example scenario for microphone array signal processing.
1.3. Independent Component Analysis (ICA)

Independent Component Analysis (ICA) is a method based on double microphone. Based on double microphone several algorithms are proposed according to targeted areas. BSS is a method which comes under ICA category. Recently ICA got attention because of its characteristics and can applicable to not only in signal processing in context of speech recognition and also in telecommunication, Image processing and neural networks. The main idea behind the ICA is “Retrieving unobserved signals or sources from observed linearly mixed signals based on the assumption that these signals are mutually independent” [2]. The limitation of ICA is that the sources should be statistically independent.

1.4. Single microphone for non-stationary noise

This algorithm helps to retrieve the speech signal from non stationary noisy source and also can track the noise variance accurately up to noise power level of 10 dB/s. Algorithm estimates the noise variance and updates recursively with the minimum mean square error of the current noise power, for each time frame and for each frequency bin. In addition a smoothing parameter, which is time and frequency dependent, is used and it varies according to the estimation of speech presence probability. A spectral gain function by using an iterative data-driven training method is included to estimate the noise power. The algorithm is tested under many circumstances i.e., stationary and non stationary noise sources and various signal-to-noise ratios, for speech enhancement system, improvements in segmental signal to noise ratio is more than 1 dB is achievable under most non stationary noise sources.

The three methods that are explained above is to solve the same problem, but we need to choose one optimal algorithm which suits for hearing aid device. DRNPE is the best suitable for hearing aid as it only requires one microphone and also as we increase the noise profiles in algorithm it can solve many more circumstances as per the user requirement. Further in this thesis we will review the blind source Separation method based on ICA and compare with the single microphone source separation method. This thesis attempts to give solution for the following problems,

- Finding a optimal algorithm for hearing aid.
- Performance evaluation of blind source separation method based on ICA.
- Performance evaluation of one microphone source separation method.
- Conclusion for an efficient method for auditory scene analysis.
- Future work.

The above mentioned problems are the goal of this thesis, In further chapters these problems are discussed in brief hence proposed a suitable solution based on present noise extracting algorithms.

1.5. Thesis scope

The thesis will give a best suitable algorithm for hearing-aid based on single microphone. The proposed algorithm will sustain in a any kind of noise backgrounds especially non-stationary backgrounds like market, shopping malls where the noise changes abruptly. The proposed algorithm was tested under non-stationary backgrounds and showed good results. The following chapters will discuss why to use DRNPE and why not others and results.
• **Study of the current noise suppression algorithms:**

This thesis report deals with three kinds of algorithms i.e., Microphone Array, ICA, DRNPE noise cancellation.

• **Study of blind source separation by ICA:**

Origin of method, method and efficiency of ICA will be discussed in detail.

• **Implementation and performance evaluation of BSS:**

Algorithm Flow, testing of algorithm using some audio files acquired from International Dialect by showing the results by using graphs also Study of the advantages and limitations of ICA.

• **Study one microphone source separation technique:**

Description about how one microphone algorithm actually works and also what improvements made in comparison of present once microphone algorithms.

• **Implementation and performance:**

Brief explanation of algorithm, advantages, evaluation of one microphone source separation technique, Graphs showing the performance and some results.

• **Comparison of ICA and one microphone source separation:**

Comparison is done by the results acquired in the previous chapters and choosing a optimized algorithm for hearing aid device.

The best effort has been put to make this document user friendly and also understandable to all kind of readers and also those who are not familiar to technical things like signal processing. It is suggested to read the books mentioned in bibliography for further understanding.

1.6. **Methodology**

A literature study was done to investigate different methods for auditory scene analysis problem. This thesis work choose three types of algorithm to deal with ASA. These three methods will be discussed briefly to find a optimal algorithm for hearing aid. ICA and DRNPE methods will be implemented and the performance is evaluated with respective algorithms. Tests are conducted by using the some audio files and comparison will be done using results. Conclusion will be made to choose optimal algorithm. At the end future work will be mentioned.

1.7. **Organization**

The chapters are organized in perspective of reader, so that any user with any background can understand the purpose of the document in detail. Each chapter discusses the things in detail and also took some real time examples to involve the user in document.

2. **Independent Component Analysis (ICA) theory**

Explains the origin of ICA briefly, method of implementation, assumptions made while implementing the algorithm, properties of ICA methodology, advantages of ICA. The flow of algorithm is explained briefly in this chapter by using appropriate equations.

3. **Simulation and performance of CoBliSS**
This chapter shows simulation results and performance evaluation based on the results by using the non-stationary and stationary noisy audio sources. Required graphs are drawn to show the performance.

4. Source separation from single source

Introduction to DRNPE source separation, improvements made compared to the present algorithms, assumptions made for the implementation of algorithm, required mathematical equations will explained briefly.

5. Comparison of performance

DRNPE and double microphone based algorithms will go under test using audio sources contaminated by stationary and non-stationary noise, results are shown in the graph and performance will be evaluated based on the results. Optimal algorithm will be chosen.

6. Suggested future work

Suggested future work is mentioned in respective of improvements needed for chosen algorithm and hardware requirements.

7. Conclusions

Gives a conclusion of the thesis regarding the work made. Discusses the further possible improvements that could be made to this thesis work.

Bibliography

Provides the links to the references that are used during the thesis work are mentioned.

The report is divided into total of eight chapters and discusses the ASA algorithms and demonstrates the results and suggests a robust and optimal algorithm for implementing a portable hearing-aid.

1.8. Definitions

The following definitions describes the notations took during the document. The author suggests to read these carefully to understand these to get the full knowledge of the document.

- Bold lower case letters indicate vectors and bold upper case letters denote matrices.
- Gaussian distribution: Continuous distribution of data that varies near the mean.
- System delay units are samples
2. INDEPENDENT COMPONENT ANALYSIS (ICA) THEORY

The aim of blind source separation is about recovering independent sources from sensor observations that are linearly independent source signals. The name blind indicates that the way the source signals are mixed together is unknown. ICA is a solution to this problem (BSS). ICA tries to find out the coordinate system so that the recovered signals are independent statistically and linearly. In the context of correlation based transformations ICA not only tries to de-correlate the sources but also tries to decrease the higher order statistical dependencies. In total we can express the ICA by “a method for tracing the non-orthogonal co-ordinate system determined by second and higher order statistics of the original data sources. The aim is to perform a linear transformation the resulting variables to be statistically independent from each other as much as possible”. The source of this following theory is from [2], dedicated to the BSS method.

2.1. Methodology

Considering the cocktail-party problem, imagine that you are in party. Where you are involved in a conversation with your friend and also nice heavy music is going on. Two microphones were recording from different locations. We will get two recorded time signals, denote them as \(x_1(t)\) and \(x_2(t)\) where \(x_1\) and \(x_2\) are amplitudes and 't' is a time index. These recorded signals contains some speech by the speakers, denote them as a \(S_1(t)\) and \(S_2(t)\).

Put them in to a linear equation,

\[
x_1(t) = a_{11}s_1 + a_{12}s_2
\]

\[
x_2(t) = a_{21}s_1 + a_{22}s_2
\]

where \(a_{ij}\) are some parameters depends on the various factors like distance between microphone and speakers. The problem will be solved if we could estimate the \(S_1(t)\) and \(S_2(t)\) in the above equations by using only the recorded signals \(x_1(t)\) and \(x_2(t)\). We could neglect the time delays or any other factors in the problem. The spectrum in time domain are demonstrated in the Figure 2.1Figure 2.2. Figure 2.1 indicates the original speech signals by two speakers and mixed signals look like in the Figure 2.2. The problem is to recover the speech signals in Figure 2.1 from Figure 2.2. The problem will be solved if we know the \(a_{ij}\) parameters, as we can solve the above equations (1) and (2). But we are unaware of these values; here comes ICA to solve this issue.

One way of solving this problem is to use the statistical properties of signals \(S_1(t)\) and \(S_2(t)\) to estimate \(a_{ij}\). Basically it is enough to assume that these speech signals are statistically independent at each time instant 't'. If this is true or even need not to be in real time. ICA can be used in this situation to estimate the \(a_{ij}\) based on the property that they are independent and we are done with the problem by using the \(a_{ij}\) values we can solve \(S_1(t)\) and \(S_2(t)\). Figure 2.3 gives the two speech signals recovered using ICA. They are almost similar to the Figure 2.1 (original speech signals). ICA is developed to solve the problems that are more related to the cocktail-party problem issues. Moreover this method (ICA) solves several other issues like Image Processing, electroencephalogram (EEG), etc.
Figure 2.1: Time-domain of original speech signals.
Figure 2.2: Time-domain of observed two mixture signals using two microphones.
2.2. ICA by equation

Consider a situation of \( n \) linear mixtures \( x_1(t), \ldots, x_n(t) \), of \( n \) independent components. This can be written as (3). In this equation we are not considering the timing issue as before (example: \( x_1(t) \)) rather we assume each mixture \( x_j \) and each independent speech signal \( S_k \) as a random variable instead of timing signal.

\[
X_j = a_{j1}s_1 + a_{j2}s_2 + \ldots + a_{jn}s_n \quad \text{for all} \quad j
\]  

(3)

The observed values of \( x_j(t) \) are samples of this random variable. The mean of mixture variables and speech variables (\( S_k \)) have zero mean. If this condition fails then the observed variables of \( x_j \) can always be centered by subtracting the sample mean, which makes the model zero-mean.
To deal this issue more easy way (in fact this is what we are looking for!)

By using vector-matrix notation instead of the equations like (1), (2) and (3). Consider \( X \) as random vector consisting of mixture elements \( x_1(t), \ldots, x_n(t) \), \( S \) as a random vector consisting of elements \( S_1, S_2, \ldots, S_n \), and \( A \) with elements \( a_{ij} \). All vectors are expressed as a column vectors. Now we can express our mixing model as,

\[
x = As
\]

\[
X = \sum a_{i}S_i \text{ where } i=1\ldots n
\]

The model in the equation (4) is independent component analysis or ICA model. This generative model, i.e., defines the observed data \( X \) consisting of mixed components \( S_i \). The independent components are latent variables, meaning that they cannot be observed directly and the mixing matrix \( X \) is unknown. Our aim is to estimate the \( A \) and \( S \) by using the observed vector \( X \).

First we assume that the components in \( S_i \) are statistically independent and the independent components are assumed to be non-Gaussian distributions. But presently in this simplified version we assume that we are unaware of what kind of data is it. To make the things simple we assume that unknown matrices are square matrices. After calculating the \( A \), we can easily calculate the inverse of it, say \( W \) and get the rest matrices by using the equation (6).

\[
s = Wx
\]

In the name of blind source separation (BSS), the source is the speaker in a cocktail-party problem and the blind means we know a very little about mixing matrix and make some assumption on source signals to estimate \( S \). ICA is a method to perform blind source separation.

### 2.3. Equivocality of ICA

After all there is still uncertainty regarding some things in ICA but they are insignificant most of the times.

- Difficulty in specifying the variances of independent components.
- Difficulty in specifying the order of independent components.

The reason behind the uncertainty in specifying the variance is being unaware of \( A \) and \( S \), from equation (2).

We can conclude that any scalar multiplier in the sources \( S_i \) can be canceled by dividing the corresponding the column \( a_i \) of \( A \) by the same scalar. Apparently we may fix the magnitudes of the independent components as they are random variables. Most easy way to do this is by assuming each has a unit variance i.e., \( E(s_i^2) = 1 \).

Then the matrix \( A \) will be considered in ICA to take in to account this restriction. But still there is uncertainty in context of sign. We can solve this issue by multiplying \(-1\) to independent components without affecting the model.
When it comes to ambiguity regarding order, we can change the order of the sum in equation (5) and make any of the independent components as first one. A permutation matrix $Q$ and its inverse $Q^T$ in the equation (7) to solve $X$.

$$X = A (Q^{-1}) \cdot QS$$  \hspace{1cm} (7)

The elements in $QS$ are independent variables $S_i$ but with different order. The matrix $A Q^{-1}$ is just same as $A$ but with different order and still need to be calculated by ICA algorithm.

### 2.4. ICA properties

Until now we briefly learned about how ICA works, now we are going to learn about some of ICA properties that are compiled below which puts ICA different from the other algorithms that do the same work.

- **What does it mean by components are independent?**

  Assume that $r_1$ and $r_2$ are two scalar valued random variables. These two are said to be independent when information on $r_1$ does not give any information regarding $r_2$ and viceversa and if the joint PDF (Probability Density Function) is factorisable.

  $$P(r_1, r_2) = P(r_1) P(r_2)$$  \hspace{1cm} (8)

- **Uncorrelated?**

  If two variables are independent that means that they are uncorrelated. Two random variables $r_1$ and $r_2$ are said to uncorrelated if their covariance is zero.

  $$E(r_1 r_2) - E(r_1) E(r_2) = 0$$  \hspace{1cm} (9)

  If two random variables are uncorrelated it does not mean that they are independent. The advantage of independent components is, if two random variables are said to be independent that means they are uncorrelated. The ICA method always tightens up the estimation so that every estimated independent component is uncorrelated. This simplifies the problem.

- **What if independent components are Gaussian values ?**

  The basic restriction of the ICA. To apply ICA the independent components should be non-Gaussian. To illustrate the problem assume that mixing matrix is orthogonal and $S_i$ are Gaussian. Then $y_1$, $y_2$ are Gaussian, uncorrelated and unit variance. Their joint density can be expressed as,

  $$P(y_1, y_2) = \frac{1}{2\pi \exp \left( \frac{-(y_1^2 + y_2^2)}{2} \right)}$$  \hspace{1cm} (10)

  The distribution in coordinate is illustrated in the below Figure 2.4. The distribution is completely symmetric. So, it will not contain any information in the direction of column for the required mixing matrix $A$. So $A$ cannot be defined.
If the independent components are Gaussian variables we can only estimate up to an orthogonal transformation. So the required matrix $A$ cannot be calculated if independent components are Gaussian variables.

2.5. Pre-processing

Before applying the algorithm to data, there is some things should be done before. Next we are going to discuss some pre-processing techniques to solve the problem more conveniently.

2.5.1. Centering

In the process of pre-processing first step is to center $X$. That means we subtract its mean vector to make $X$ as a zero mean variable leads to make $S$ as a zero mean variable too (from the equation (1)). This step is done just to simplify the algorithm. After estimating the mixing matrix $A$ we can add this mean vector of $S$ back to the cantered estimated of $S$ to complete the estimation.

2.5.2. Whitening

Another step in pre-processing is whitening. Before applying ICA of course after the centering, we apply whitening on observed vector $X$ so that we get a new vector $\tilde{X}$ which is white. Components in white vector matrix are uncorrelated and their variance equals to unity and the covariance is equal to Identity matrix.

$$E[\tilde{x}\tilde{x}^T]=I$$

Whitening reduces the difficulty of algorithm by reducing the number of parameters to be calculated. Instead of estimating the $n^2$ parameters of $A$, we estimate the $\hat{A}$. In a two dimension matrix the
orthogonal parameters to be determined is 1, but if we consider higher order orthogonal matrix it has only half of the original arbitrary matrix, i.e., whitening solves the problem by half. Whitening is applied to the data in Figure 2.5 and Figure 2.4 illustrates how the data will be changed after whitening.

Figure 2.5: The distribution of mixed signals.

Figure 2.6: Distribution of mixture matrix after applying whitening.
2.5.3. Further preprocessing

The success of ICA on some type of data is depends crucially on performing application dependent preprocessing steps. If we consider a situation where if the data consists of time signals, band pass filtering may be useful. This can be explained with an example,

Consider the matrix $X$, contains observations of $X(1)$, $X(2)$, $X(3)$,......$X(n)$ as its columns and same for $S$. The ICA model for this is

$$X = AS$$

Time filtering for $X$ corresponds to multiplying $X$ from the right by a matrix, call the multiplying matrix as $Z$. Which gives,

$$X^* = XM = ASM = AS^*$$

Which shows the ICA model is still valid.
3. SIMULATION AND PERFORMANCE OF COBLISS

3.1. Introduction

Blind Signal Separation is a method to recovering the independent signals using only observed mixtures of these. To deal with acoustical applications, These observed mixtures are signals of multiple microphones. For this a convolutive algorithm is used, called multichannel finite impulse response to filter these signals. Several algorithms are proposed for convolutive separation [3][4][5]. Some authors says that Second Order Statistics (SOS) won't work out with BSS and most of the authors used Higher Order Statistics (HOS). The HOS algorithms contains non-linear elements that can be tuned to the data in order to get the good results. We are only going to discuss about BSS which is only depends on SOS and it never required to be tuned with any parameters. The advantage of CoBliss algorithm is that no assumptions are made about the probability density function or any other properties of signals. Experiments are with real recordings in a living room, shows the algorithm performance. As we discussed in earlier BSS algorithms no parameters needs to be tuned. The algorithm proposed by [6],does not need any parameters to be tuned and it based only on SOS.

The optimization criteria behind this algorithm is based on minimizing the cross correlation among the outputs of the multichannel separating filter. Considering the expensiveness of this algorithm and also to get a fast convergence of algorithm the criterion transformed in to frequency domain. This is explained in further sections. The filter coefficients are calculated in a way that the cross correlations are equal to zero. It does not impose any restrictions, but to ensure that the filter coefficients correspond to real filters of a given length in a time domain. A remedy is discussed in further sections so that the cross correlations are non zero. As we are dealing with two domains, We have two sets of constraints an iterative method is proposed in which the weights are adjusted iteratively in alternatively one and other domain. The main idea is to obtain a good performance in terms of separation and convergence is to find suitable adoption in the frequency domain which does not intact the time domain as much as possible, the time- frequency domain compliance is discussed in further sections. A normalization must be applied in order to prevent the whitening of signals by algorithm. This newly discussed blocks above compared to the chapter 2. builds the new algorithm Convolutive Blind Signal Separation. This algorithm is tested with people that are speaking in a recorded room. The experimental results are discussed at the end of this chapter. The author suggests to read the book [7], for more detailed explanation regarding BSS.

3.2. BSS algorithm

The following sections explains the notations required during the implementation, optimization criteria and frequency and time domain approach for the problem also explains the how the noise extraction is done in this method. How optimization is done to retrieve required noise free speech.

3.2.1. Notations

Before discussing about algorithm, first we should learn some prerequisites to understand the algorithm in detail. In this whole section time domain signals will be denoted by lower case and frequency signals
will be denoted by upper case characters. Vectors are denoted by underline (example $\underline{V}$), Subscript denotes the vector or matrix dimensions. A matrix with one sub script will be square matrix. Then, $A^*, A^T, A^{-1}$ denotes complex conjugate, matrix transpose and inverse matrix respectively and $j^2 = -1$. $\otimes$ denotes the element wise multiplications. The expectation operator is denoted by $E\{\}$.

The $N \times N$ identity matrix will be denoted by $I_N$ and $K \times L$ zero matrix will be denoted by $0_{K,L}$. The $M \times M$ Fourier matrix $F_M$ is defined as $(F_M)_{ij} = e^{-2\pi j i M}$ and diagonal $\{\}$ converts the diagonal matrix elements into a vector.

### 3.2.2. Optimization criterion

An algorithm based on blind source separation controls the MC-FIR filter is also minimizes the cross correlations among the outputs of this filter. The notation in accordance to below figure describes the mixing/unmixing system. The independent sources $s_1,...,s_J$ are mixed by the mixing system $H$ which gives the sensor signals $x_1,...,x_J$. Totally the independent sources and sensor signals are same count equal to $J$.

![Figure 3.1: Mixing and unmixing system](image)

Time indexes are not mentioned explicitly in all formulas, the separation filters transfer function from the $i^{th}$ input to the $m^{th}$ output is denoted by $w_{im}^N$. $y_m$, the separation filters $m^{th}$ output is calculated from the observations $x_i^N$.

$$y_m[n] = \sum_{i=1}^{J} (W_{im}^N) x_i^N[n]$$

with $J$ number of microphones and with the filter length $N$ (by assuming all filters have the same length for simplicity).

$$W_{im}^N = \begin{pmatrix} W_{im}[N-1] \\ \vdots \\ W_{im}[0] \end{pmatrix} \quad x_i^N = \begin{pmatrix} x_i[n-N+1] \\ \vdots \\ x_i[n] \end{pmatrix}$$

The cross correlation within the outputs can be denoted as (13)
\[ r_{X,Y[l]} = E[y[n]y[n+l]] = \]

\[
\sum_{i=1}^{J} \sum_{j=1}^{J} \sum_{c=0}^{N-1} \sum_{d=0}^{N-1} W_w[b] W_{y'[d]} r_{x,x}[l+b-d]
\]

(13)

with \( r_{x,x}[l] = E[x[n]x[n+l]] \)

Filters \( w_{ia} \) can be optimized by using the above expression, which can be done by using the cross correlations of the observed data which is an added advantage. These cross correlations are does not depend on the separation filters, so they never need to be updated every time the separation filter is updated. Cost function can be derived from (13), for example, sum of squares of cross correlation coefficients. Due to large number of filter coefficients, the straight forward minimization of cost function is not possible. This can be explained by an example, Consider that 2 sources and 2 microphones are used. In this case four FIR filters need to be calculated and each filter having several hundreds to thousands of coefficients. Moreover these coefficients are dependent on each other which makes this problem even worst. We need some solution which can give filter subset coefficients which are independent to each other as possible. This is the reason we transform (13) in to frequency domain.

For all lags the cross correlations are stacked in a vector and they are considered as \( l_1 = l_2 = \ldots = l_2 \), \( \sum_{i=1}^{J} \sum_{j=1}^{J} r_{x,x}[l_1] \ldots r_{x,x}[l_2] \) and

\[
A_{\mu}^{N,L} = \begin{bmatrix}
W_x[0] & 0 & \cdots & 0 \\
0 & \ddots & \ddots & \vdots \\
\vdots & \ddots & \ddots & \ddots \\
0 & \cdots & W_x[0] & \ddots \\
\vdots & \cdots & \ddots & \ddots \\
0 & \cdots & 0 & W_x[N-1]
\end{bmatrix}
\]

(15)

\[
R_{\mu}^{N,L} = \begin{bmatrix}
r_{x,x}[l_1+N-1] & \cdots & r_{x,x}[l_1-N+1] \\
\vdots & \ddots & \vdots \\
r_{x,x}[l_2+N-1] & \cdots & r_{x,x}[l_2-N+1]
\end{bmatrix}
\]

(14)

If \( l_1 \leq -N+1 \), \( l_2 \geq N-1 \) satisfies then there is a guarantee that the solutions for the MC-FIR to be non-ambiguous. In the sequel \( l_1 = -N+1 \) and \( l_2 = N-1 \) then (14) can be rewritten as,

\[
R_{\mu}^{L} = \sum_{a=1}^{J} \sum_{c=1}^{J} (I^T 0^{L-M,M}) R_{\mu}^{M \times M} A_{\mu}^{M \times M} W_{ia}
\]

(16)

Where \( M = L = 2N-1 \).
\( \hat{W}_M = \left[ 0^{W_M/M-N \times N} \right] \) and \( \hat{A}_M \) is formed by extending \( A_M^{N \times N} \) on the right such that it becomes circulant

\[
\hat{A}_M = \begin{pmatrix}
W_M[0] & 0 & \cdots & 0 \\
W_M[N-1] & W_M[0] & \cdots & 0 \\
0 & \ddots & \ddots & \ddots \\
0 & \ddots & \ddots & \ddots \\
W_M[N-1] & \cdots & \cdots & W_M[0]
\end{pmatrix}
\]

Next in the equation (16), \( R_{ac}^M \) - cross correlation matrix is approximated by it is circulant variant \( \hat{R}_{ac}^M = E \left[ \hat{X}_M^c (\hat{X}_c)^T \right] \), with \( \hat{X}_M \) the circulant data matrix

\[
\hat{X}_M[\eta B] = \begin{pmatrix}
x_i[\eta B-M+1] & x_i[\eta B] & \cdots & x_i[\eta B-M+2] \\
\vdots & \ddots & \ddots & \vdots \\
x_i[\eta B] & \cdots & \cdots & x_i[\eta B-M+1]
\end{pmatrix}
\]

3.2.3. Frequency domain approach

As we decided to work on frequency domain to reduce the complexity issues in context of hardware, to do that first we transform the cross correlation expression in equation (16) to frequency domain. The cross correlation matrix \( R_{ac}^M \) in equation (16) is replaced by it is circulant approximation makes it possible to diagonalize the matrices in equation (16) using FFT’s. We can do this by inserting the identity matrix \( (F^M)^{-1} \) \( F^M \) in between all matrices, resulting

\[
\mathcal{L}_{\gamma_i,\gamma_j}^L = \left( I^L 0^{L-L_M} \right) \sum_{a=1}^{J} \sum_{c=1}^{J} (F^M)^{-1} (\hat{R}_{ac}^M \otimes \hat{W}_{ac}^M \otimes (\hat{W}_{ac}^M)^*) \otimes (\hat{L}_{\gamma_i,\gamma_j}^M)^{-1} \tag{17}
\]

where

\[
\hat{R}_{ac}^M = \text{diag} \left[ F^M \hat{R}_{ac} (F^M)^{-1} \right] \\
\hat{W}_{ac}^M = F^M \left[ \begin{array}{c}
J^N \\
0^{W_M/M-N}
\end{array} \right] \\
\hat{L}_{\gamma_i,\gamma_j}^M = \left( 1, e^{\frac{2\pi i}{M}}, \ldots, e^{\frac{2\pi i (M-1)}{M}} \right)^T
\]

and with \( N \times N \) and the mirror matrix \( J^N \), having ones on its anti diagonal and zero’s elsewhere. Compensations has done by \( \hat{L}_{\gamma_i}^M \) and the complex conjugate in (17) for the fact that \( W_M^N \) is not flipped upside down in \( \hat{W}_{ac}^M \) as opposed to \( \hat{W}_{ac}^M \).

Separating the signal is possible when all cross correlations among the outputs equal to zero,

\[ \forall i \neq j: \mathcal{L}_{\gamma_i,\gamma_j}^L = 0^L. \]

Using (17), uncorrelated outputs \( \forall i \neq j \),
From the equation (18) we can see that, from now on frequency domain filter coefficients are no longer depend on the window \(I^L \ 0^{L-M}\) and the expression is reduced to a set of scalar equations, we solved little bit of our problem. Next, we find an approach to solve these scalar equations individually. For that, we put \(p^{th}\) elements of matrices \(\ddot{W}_{ij}^r, \ddot{R}_{ij}^r\) are shifted to matrix \(\forall i, j\).

\[
\ddot{W}_p^r = \begin{pmatrix} (\ddot{W}_{11}^r)_p & \cdots & (\ddot{W}_{1J}^r)_p \\ \vdots & \ddots & \vdots \\ (\ddot{W}_{J1}^r)_p & \cdots & (\ddot{W}_{JJ}^r)_p \end{pmatrix}
\]

\[
\ddot{R}_p^r = \begin{pmatrix} (\ddot{R}_{11}^r)_p & \cdots & (\ddot{R}_{1J}^r)_p \\ \vdots & \ddots & \vdots \\ (\ddot{R}_{J1}^r)_p & \cdots & (\ddot{R}_{JJ}^r)_p \end{pmatrix}
\]

During the practical situations the number of linearly independent columns or rows of a matrix of \(\ddot{R}_{ij}^r\) is full, So the equation (18) can be rewritten as,

\[
\forall p \quad (\ddot{W}_p^r)^* \ddot{R}_p^r (\ddot{W}_p^r)^T = \Lambda_p^j
\]

\[
\Rightarrow (\ddot{W}_p^j)^* \Lambda_p^{j-1} (\ddot{W}_p^j)^T = (\ddot{R}_p^j)^{-1}
\]

(19)

Where \(\Lambda_p^j\) stands for diagonal matrix. The off- diagonal elements are zeros because of equation (18) and in frequency bin \(p\) determines the auto correlation of the outputs of the BSS, due to \(\Lambda_p^j\) is real by definition and its inverse is also diagonal, from weight matrices it can be absorbed. The consequences are discussed in the following sections. By definition \(\ddot{R}_p^j\) is symmetrical and also the circulant cross correlation matrices \(\ddot{R}_{ij}\) are symmetrical that gives a relation, \(\ddot{R}_{ij} = \ddot{R}_{ji}\). The symmetric matrix \(\ddot{R}_p^j\) inverse is also symmetric, then in the equation (19), the right hand side can be decomposed in various ways. To obtain a right solution these, in general the matrix decomposition should be different \(\forall p\). During the practice these values are unknown, for this reason initially all \(\ddot{R}_p^j\) are decomposed in the same manner.

### 3.2.4. Convolution constraint

Equation (19) is solved independently for \(\forall p\), So that frequency domain filter coefficients are no longer related to the window. In other way there is no relation between the frequency domain filter to the real time domain filters of length \(N\). The truth is that time domain filters should be real is not a problem, because both frequency domain cross correlation vector and frequency domain filters have the same symmetric properties, but the time domain filters must be a length of \(N\) and it will be achieved by doing \(\forall j, c\).

\[
(F^M)^{-1} \ddot{W}_{jc}^r = \begin{pmatrix} W_N^c \\ 0^{M-N} \end{pmatrix}
\]

(20)
$$\tilde{W}_p^M := F^M \begin{pmatrix} I^N & 0^{N,M-N} \\ 0^{M-N} & 0^{M-N} \end{pmatrix} (F^M)^{-1} \tilde{W}_p^M$$

(21)

The above equation concludes that, in time domain the filter coefficients which must be set to zero are set to zero. This destroys the whole idea of solution by equation (19), So there needs a compliance between frequency domain solution and convolution constraint and it is discussed in following section.

3.2.5. Time-frequency compliance

There is no common solution in a closed form for the equations (19) and (20). An iterative approach is followed. Weight matrices are initialized so that $$(\tilde{w}_p^T)^* \tilde{w}_p^* = \tilde{R}_p^{-1}$$, after this following two steps must be initialized

1)According to equation (21), filters are constraint in the same domain

2) $\tilde{R}_p$, cross correlation matrices are updated

The main idea is to hold the equation (19) again by finding a way to adapt the weight matrices slightly. Until convergence is achieved this weight adoption and step (21) is performed, as discussed in the previous section this corresponds to finding the individual decompositions for $\tilde{R}_p$. Weight update is done by following two methods, One is by Exact and other is by approximate. Approximation is advantage because it reduces the computational complexity.

3.2.6. Exact weight update

As we discussed earlier section regarding weight update, The following derivation explains how to do that. All matrices mentioned in the derivation are $J \times J$ size and the respective super scripts will be omitted. Filters are having a restricted design as in equation (20), that gives weight matrix product $\tilde{W}_p^T \tilde{W}_p^* = B_p$ with $B_p = \tilde{R}_p^{-1}$ and the cross correlation matrices are updated $\tilde{R}_p = \tilde{R}_p$. The idea is to find a matrix $C_p$, so that it satisfies $\tilde{W}_p^T \tilde{W}_p^* = \tilde{R}_p^{-1}$ with $\tilde{W}_p = \tilde{W}_p C_p$. When $B_p$ is near to the $\tilde{R}_p^{-1}$, $C_p$ should be near to the matrix identity. By following this idea, the previous solution is unchanged as much as possible and also fast convergence is guaranteed. To compute the transform matrix by using decomposed matrices

$$D_p = \text{sqrtn}(B_p)^* \Leftrightarrow D_p^T D_p^* = B_p$$

$$D_p = \text{sqrtn}(R_p)^{-1} \Leftrightarrow D_p^T D_p^* = \tilde{R}_p^{-1}$$

(22)

with $\text{sqrtn}(.)$ the matrix square root, i.e., $A = \text{sqrtn}(B) \Leftrightarrow A^H A = B$, where $A^H = A$ and $B$ is a complex symmetric matrix, i.e., $B^H = B$. $C_p$, the transform matrix can be derived from

$$B_p = D_p^T (D_p^*)^{-1} \tilde{R}_p^{-1} (D_p^*)^{-1} D_p^*$$

$$\Leftrightarrow \tilde{W}_p^T \tilde{W}_p^* = D_p^T (D_p^*)^{-1} \tilde{W}_p^T \tilde{W}_p^* (D_p^*)^{-1} D_p^*$$

$$\Leftrightarrow \tilde{W}_p^* = \tilde{W}_p (D_p^*)^{-1} D_p^*$$

$$\Leftrightarrow \tilde{W}_p^* = \tilde{W}_p (D_p^*)^{-1} D_p$$

(23)
Then, \( C_p = (\sqrtm(B_p)^* \sqrtm(R_p^{-1})^*) \). In offline implementation part of the algorithm the cross correlations will be estimated first and \( (R_p^{-1})^* \) will only be calculated once. When it comes to an online implementation, the cross correlations change with time and also it is required to be compute \( C_p \) after every update. The \( \sqrtm \) involved in the above equations is used because of its necessity when there are many signals to be separated (large \( J \)). In the next section we are going to discuss about the weight update by using the method of approximation which requires less computational complexity.

### 3.2.7. Approximated weight update

To reduce the computational complexity, we are not using the square root in this method which updates the weight faster than the exact weight update. However the cross correlation matrices change only slowly in time. From the previous section after the time domain constraint is applied \( (21) \), \( \tilde{W}_p^T \tilde{W}_p^* = \tilde{R}_p^{-1} \). \( \tilde{R}_p \) changes to \( \tilde{R}_p^{-1} \) when cross correlation matrices are updated. \( \epsilon_p \) Must be found so that,

\[
\tilde{W}_p^T \tilde{W}_p^* = \tilde{R}_p^{-1} \quad \text{with} \quad \tilde{W}_p = (I + \epsilon_p) \tilde{W}_p
\]  

(24)

Next, weight matrices should be derived in way that their product becomes equal to the inverse of the updated cross correlation matrices. Refer that \( \Delta \tilde{R}_p = \tilde{R}_p' - \tilde{R}_p \) so that,

\[
\tilde{W}_p^H (I + \epsilon_p)^H (I + \epsilon_p) \tilde{W}_p = (\tilde{R}_p + \Delta \tilde{R}_p)^{-1}
\]

\[
\tilde{W}_p^H \tilde{W}_p + \tilde{W}_p^H (\epsilon_p^H + \epsilon_p) \tilde{W}_p^H \approx \tilde{R}_p^{-1} - \tilde{R}_p^{-1} \Delta \tilde{R}_p \tilde{R}_p^{-1}
\]

\[
\Rightarrow \tilde{W}_p^H (\epsilon_p^H + \epsilon_p) \tilde{W}_p \approx - \Delta \tilde{R}_p \tilde{R}_p^{-1}
\]

(25)

In equation (25), The approximation representing to neglecting of higher order terms of \( \Delta \tilde{R}_p \) in the expansion of the series \( (I + \Delta \tilde{R}_p)^{-1} \). Next \( \epsilon_p \) must be chosen according to equation (25), so that the changes to \( \tilde{W}_p \) are small and the fast convergence is achieved for sure. Both sides of the equation (25) are symmetrical by definition. From the triangle inequality the \( \epsilon_p \) wit the smallest \( L_{2} \) norm satisfying equation (25) is

\[
\epsilon_p = \epsilon_p^H = - \frac{1}{2} \tilde{W}_p \Delta \tilde{R}_p \tilde{W}_p^H
\]

According to equation (24), the weight update becomes

\[
\tilde{W}_p' = (I - \frac{1}{2} \tilde{W}_p \Delta \tilde{R}_p \tilde{W}_p^H) \tilde{W}_p
\]

\[
= \tilde{W}_p (I - \frac{1}{2} \tilde{R}_p \tilde{W}_p^H \tilde{W}_p)
\]  

(26)

### 3.2.8. Normalization

In this section we are going to discuss about the impact by setting equal of both constraint matrices \( \Lambda_p \) to the matrix identity in frequency domain approach section. Diagonal elements of the constraint matrices \( \Lambda_p \) prescribe the power of the outputs of the separation filter at the corresponding frequency. At first we discuss about the effect of choosing the constraint matrix equal to the matrix identity for sources that have equal energy distributions as a function of frequency. However, In real time situations
signals like speech for higher frequencies the energy decays significantly. When the BSS algorithm is
drove forcibly to yield the outputs having equal energy for all possible frequencies this will result
energy boosting for the signals that are weak and also energy will be reduced for the signal that are
strong. This may cause some problem, for example unwanted signal equalization’s where the higher
frequencies are boosted and lower frequencies are lowered which results artificial sounding recovered
from speech. We can not solve this directly because the ideal constraint matrices, depends on the
unknown original sources. There is another approach which can solve this issue, first the $\hat{W}_p$ are
calculated from equation (19) by using the $\Lambda_p^J=I^J$, next weight matrices are normalized by using

$$\tilde{W}_p := \frac{W_p}{\|W_p\|}$$ (27)

$l_1$ norm gives the good performance and it can be used. The truth is that all filter coefficients that have
having the same order of magnitude after this normalization is applied, advantage is that the timbre of
the speech signals is unaffected which was produced by a all pass filter. Next problem will be powers of
the source signals do not evolve by a function of frequency which results unwanted equalization still
occurs despite the scalar normalization. If this is the situation, then more sophisticated procedure could
be followed where $\Lambda_p^J$ will be estimated by using the separated signals. However, this is is out of topic
for us.

### 3.3. CoBliSS algorithm flow

In this section we are going to discuss the flow of CoBliSS algorithm which consist of building blocks as
discussed in earlier sections. The procedure consists of following steps:

1) Transform the input data blocks in to frequency domain $\forall a$ :

$$\tilde{X}_a^M = F^M \begin{pmatrix} x_a[nB-M+1] \\ \vdots \\ x_a[nB] \end{pmatrix}$$

where each block is of length $M$ and overlapping and per each block only $B$ new samples are used.

2) Efficiently updating the cross correlations in the frequency domain $\forall a, c$ ,

$$\tilde{K}_p := \alpha \tilde{K}_p + (1-\alpha)((\tilde{X}_a^M)^* \otimes \tilde{X}_c^M)$$

Where forgetting factor $\alpha$ , can be vary between 0 to 1 based on the application. However it is chosen
to near 1 (for example $\alpha = 0.99$ ).

3)After updating the cross correlation matrices several times, the weights are initialized by decomposing equation (19) by using the matrix square root,

$$\forall p: W_p^J = sqrtm(\tilde{K}_p^J)^{-1}$$

Note that: $(\tilde{K}_p^J)_{a,c} = (\tilde{K}_p^M)_{a,c}$ .

4) To hold the equation (19), the weights are changed where all matrices are of size $J \times J$,
∀ \, p: \tilde{W}_p := \tilde{W}_p C_p

C_p = \left( \text{sqrtm} \left( \tilde{W}_p \tilde{W}_p^T \right) \right)^{-1} \text{sqrtm} \left( \tilde{R}_p^{-1} \right)

Note: this step can be omitted at initialization.

5) Normalization of weight matrices by using

\tilde{W}_p := \frac{\tilde{W}_p}{\| \tilde{W}_p \|}

6) According to equation (20), the weight matrices are constraint ∀ \, p :

\tilde{W}_{jc}^M := F^M \left( \begin{array}{cc} I^N & 0^{N, M-N} \\ 0^{M-N, N} & 0^{M-N} \end{array} \right) (F^M)^{-1} \tilde{W}_{jc}^M

Note: \left( \tilde{W}_{p_{a, c}}^J \right)(a,c) = \left( \tilde{W}_{a, c}^M \right)(p).

7) Using overlapping method in equation (20), the filtering is performed efficiently in the frequency domain to obtain the separated outputs from the noisy sources.

8) Except for the initialization in item 3 all steps are repeated.

\tilde{Y}_j = \left( \begin{array}{c} 0^{B, M-B} \\ I^B \end{array} \right) (F^M)^{-1} \sum_{a=1}^J \left( \sum_a \tilde{W}_{ja}^M \right)

The updating and filtering are done independently. To reduce the computational complexity and cost of slower convergence we can reduce the rate of update.

3.4. Simulation and results

In this section we are going to present some of the simulation results done using some audio samples which have certain noisy background. Below graphs shows the simulation results. As this thesis is about finding a best suitable algorithm, we used same audio file to test the efficiency of different algorithms. In this section we only took one noisy audio file to present the results. In the Comparison of performance chapter, we will elaborate the simulation results of the both algorithms using several types of noisy files. The following link [8], presents a results regarding the CoBliss algorithm which was invented by Schobben.

To test the algorithm we used two microphones, where two people (male and female) standing in a room and reading a small note which was written in English which runs for 10 seconds. We placed the microphones each one near to the speaker, so that the microphone can record both voices. The recorded audio files are 16 bit and frequency of 24 kHz. The separation filters controlled by this algorithm are of length 512 and our goal here is to separate the two voices. Figure 3.2 shows the audio files before processing and Figure 3.3 shows the processed audio using the ICA algorithm. Updating is done for every 2560 samples and also with in 0.25 seconds algorithm converges a good solution. The real advantage of this algorithm is it also cancels the echo [9]. The algorithm also tested with the music background and it works good and able to separate the music files.
Figure 3.2: Time-domain of a speech source contaminated by noise.

Figure 3.3: Time-domain of a noise free audio file after applying the ECoBliss algorithm
SNR  |  5.5147 dB  
--- | ---
Minimum error  |  137.15  
Delay  |  0.000143 samples  

Table 3.1: Results after applying the BSS algorithm on noisy source.

Even if these results looks good, We still not sure whether this audio satisfies in context of listening to general people. Probably some audio engineers or some other people working in the same area would satisfy with the results. The real problem is “perception”. Perception differs from person to person. We just can not decide that this algorithm performs well with some mathematical results that are obtained in lab by running some algorithms. To make sure which one is better, Few friends of mine decided by listening to the audio files before and after processing using the algorithms.
4. SOURCE SEPARATION FROM SINGLE SOURCE

Single channel source separation algorithms based on DFT (Discrete Fourier Transform) are famous because of their low complexity and good performance. One such algorithm is based on the IEEE paper published “Tracking of Non stationary Noise based on data-driven recursive noise power estimation” in 2008 by Jan S. Erkelens and Richard Heusdens [10]. Estimating a non stationary noise is been always difficult part in ASA (Auditory Scene Analysis). This algorithm helps to retrieve the speech signal from non stationary noise based source and also can track the noise variance accurately up to noise power level of 10 dB/s. Algorithm estimates the noise variance and updates recursively with the minimum mean square error of the current noise power, for each time frame and for each frequency bin. In addition a smoothing parameter, which is time and frequency dependent, is used and it varies according to the estimation of speech presence probability. A spectral gain function by using an iterative data-driven training method is included to estimate the noise power. The algorithm is tested under many circumstances i.e., stationary and non stationary noise sources and various signal-to-noise ratios, for speech enhancement system, improvements in segmental signal to noise ratio is more than 1 dB is achievable under most non stationary noise sources.

4.1. One microphone sources separation technique

The technique behind the algorithm is estimation of noise power spectral density as in most of the single source noise suppression algorithms. The difference comes at one point i.e., up to how much percentage can we estimate the noise power spectral density?. For most of the stationary noise sources we can use Voice Activity Detector (VAD) to find the speech pauses for estimation of noise power spectral density. Though they are useful in high signal to noise ratio sources, when it comes to low SNR this method isn’t reliable especially when it is non stationary noise based source. Here comes the tough task. Best way is to find a method to estimate reliable noise spectrum estimates also during the speech, it is hard to control the speech power leakage in to noise spectrum. Several methods were proposed for this problem. Best one is to track the minima of the smoothed noisy spectrum. This kind of method is used in Kalman filtering and subspace decompositions. The Sugiyama method is not based on minima statistics (MS), works fine by using the method “Weighted noise estimation”. The squared noisy amplitude’s are down weighted according to the estimated SNR values. This method works fine even during the speech to estimate noise power by reducing the problem of speech leakage in to noise spectrum.

We will look in to several methods proposed by various authors to solve the same situation and compare them. To start with, minimum statistics (MS) method by Martin, the method of Sugiyama, and the improved minima controlled recursive averaging (IMCRA) method by Cohen. To reduce the speech leakage, MS and IMCRA methods need a long time to track the minimum; this limits the algorithm to follow up the rapid increase in noise level. In that case the minimum racking is lag behind by the window length. We discuss the details in further. One more method minimum mean square error (MMSE), to estimate the noise power with reduced risk of the speech leakage in to noise power spectrum. The MMSE estimates are gained by multiplying the noisy powers with spectral gain function. The spectral gain function for noisy power estimation is obtained by iterative data-driven method. This method removes the most of the speech contribution from noisy spectrum. The three methods have their own importance in one or other way, we will discuss these methods in detail and compare them by
tracking and overall performance. To understand further, first we need to get an idea about some assumptions and definitions.

4.2. Spectral modeling

Consider an additive-noise signal model equation,

\[ X(k, m) = S(k, m) + N(k, m) \]  \hspace{1cm} (28)

Where \( X(k, m) \) is noisy speech, \( S(k, m) \) is clean speech and \( N(k, m) \) is noise are complex random variables representing the short period \( X(k, m) \) of DFT coefficients obtained at certain frequency index \( K \) and in a signal frame \( m \) from noisy speech. We assume that these three values are statistically independent from time and frequency and also each other. We make some notations here to skip the confusion by drop the pin over time/frequency index. The noisy speech amplitude \( R = \left| X \right| \), clean speech amplitude \( A = \left| S \right| \) and the noise amplitude \( D = \left| N \right| \). We assume that the DFT coefficient of noise follows the Gaussian distribution with variance \( \lambda_D \) and \( D^2 \) is the noise power, its expectation is \( \lambda_D \). Similarly, the speech spectral variance \( \lambda_S \), \( A^2 \) is the speech variance, its expectation is \( \lambda_S \).

The prior SNR \( \xi \), the post SNR \( \varsigma \) is

\[ \xi(k, m) = \frac{\lambda_S(k, m)}{\lambda_D(k, m)}, \quad \varsigma(k, m) = \frac{R^2(k, m)}{\lambda_D(k, m)} \] \hspace{1cm} (29)

4.3. Amplitude estimation

According to [11], speech amplitudes (\( A \)) estimation could be done by multiplying the noisy amplitudes \( R \) by spectral gain function. Usually, any power \( A^\beta \) of the speech amplitude can be estimated by applying the specific \( \beta \) order gain function.

\[ \hat{A}^\beta = G_{A^\beta}(\xi, \varsigma) R^\beta \] \hspace{1cm} (30)

In above equation \( G_{A^\beta} \) depends on the assumed statistical models for the speech, the noise and the model we are optimized for. After this we will find the noise power \( D^2 \) by using the gain function \( G_{D^2} \).

4.3.1. Minima Statistics (MS) method

The minima statistics method uses the minima of the smoothed periodogram of the noisy speech to estimate the noisy level on each frequency bin. The speech energy is zero during the pauses and syllables also in some frequency bins the speech power is much smaller than the noise power. This method can be useful to estimate the noise floor by the minima of the smoothed periodogram in a finite window that is quite enough to figure out the high power speech segments. The window size can be in order of ones. The method uses the time varying smoothing parameter to estimate the degree of stationary of the noise signal. The minimum value is expected to be smaller than the mean power level, a bias correction procedure is implemented.

This method has two main drawbacks. First, since the minimum value window is been used, in case of increasing noise power the estimates of the noise variance is lag behind by a window length. This may affect the performance of the method for very non stationary noise sources. Then the second, it is difficult to find the correct bias compensation factor. The compensation factor is calculated based on the window contains only noise. In general, a large fraction of the window contains noisy speech. The
periodogram values which contain speech power likely to be less than the values that contain only noisy power. So, the minimum is minimum number of a fraction of periodogram values. To estimate the bias compensation factor, as estimation of number of effective periodogram values should be made.

### 4.3.2. Improved Minima Controlled Recursive Averaging (IMCRA) method

This method was proposed by Cohen and discussed in “Noise spectrum estimation in adverse environments: Improved minima controlled recursive averaging”. The idea is to estimate the noise variance \( \hat{\lambda}_D \) by recursive smoothing of the noisy power.

\[
\hat{\lambda}_D(k, m) = \alpha_s(k, m) \hat{\lambda}_D(k, m-1) + (1 - \alpha_s(k, m)) R^2(k, m)
\]  

(31)

The smoothing parameter \( \alpha_s(k, m) \), depends on the speech presence probability \( \hat{p}(k, m) \).

\[
\alpha_s(k, m) = \alpha_d + (1 - \alpha_d) \hat{p}(k, m)
\]  

(32)

The smoothing parameter \( \alpha_s \) value is in between 0-1. From the above equation we can conclude that \( \alpha_s \) value will always lies between \( \alpha_d \) to 1. If the speech presence probability \( p \) is near to 1, so is \( \alpha_s \). From the equation (31), noise estimate is kept close to its previous value, preventing the speech power leakage in to the noise power estimate. The lower the probability of speech presence, faster the noise variance is updated. To prevent the leakage we need to have a perfect value of \( p(k, m) \). In the current method, the speech presence probability \( p \) is controlled by the minima values of smoothed power spectrum of the noisy signal. Usually estimation is done by the exponential smoothing is the time direction, apart from that some averaging over neighboring frequencies is performed taking in to account that the strong correlation of speech presence probability in neighboring frequency bins of consecutive frames. A fixed bias compensation factor is used. The IMCRA method uses two iterations of smoothing and tracking, in order to make the minimum tracking over speech activity more robust. Comparing IMCRA with MS, It reacts very slowly to increase in the noisy level. It also depends on the accuracy of bias compensation factor

### 4.3.3. SKS method (Sugiyama. Kato. Serizawa)

The MS and IMCRA methods were based on the minima statistics principle and the SKS method is based on different principle. The noisy power \( R^2(k, m) \) is weighted by factor \( W(k, m) \) that depends on the posterior SNR \( \hat{\zeta}(k, m) \).

\[
W(k, m) = \begin{cases} 1 & : \hat{\zeta}(k, m) \leq \zeta_1 \\ \frac{(\zeta_2 - \hat{\zeta}(k, m))}{(\zeta_2 - \zeta_1)} & : \zeta_1 < \hat{\zeta}(k, m) < \Theta_z \\ 0 & : \hat{\zeta}(k, m) \leq \Theta_z \end{cases}
\]  

(33)

where \( \zeta_1 < \Theta < \zeta_2 \).

The estimated posterior SNR \( \hat{\zeta}(k, m) \) is obtained from equation (29), by substituting an estimate of \( \hat{\lambda}_D(k, m) \) of noise spectral variance.
The estimated values of noise variance $\hat{\kappa}_D(k, m)$ are average of last non zero values of $W(k, m)$, $R^2(k, m)$ . Greater values of $R^2(k, m)$ are down weighted to prevent the speech power leakage. The weighting component $W(k, m)$ plays a vital role about detection speech presence.

When $\hat{\zeta}(k, m) \geq \Theta_2$, we can assume the speech presence by $W(k, m) = 0$ and the noise power is not used to update the noise variance estimates. This method works well for non stationary sources because of its updating possibility even during the speech periods.

Although it looks fine but in practical it has some drawbacks. First, if there is sudden increase in noise source then those values will be down weighted, as we discussed earlier greater values of noise power are down weighted and not used for updating the noise variance estimates. This results a slow response of algorithm for sudden increase in noise source. This problem can be solved by using safety net. It is discussed in further chapter. Second, the noise variance estimates calculated using this method are biased low in noise only regions because $W$ is always less than or equal to 1, as we can see from the equation (33). Luckily this problem can be solved by a bias correction factor $C > 1$ . Third drawback is the weighting function $W$ is heuristic, so the noise variance estimator is not optimal.

### 4.4. MMSE estimation of noise power to reduce speech leakage

To prevent the speech power leakage in to noise power estimates, MS and IMCRA methods uses the minimum values in a window of considerable length. This results a slow response to sudden increase in noise levels. To overcome this problem, we avoid using the noisy power $R^2$ directly by removing as much as possible speech contribution from it, by using the smoothing equation like (31). By replacing the $R^2(k, m)$ in equation (31) by an estimate of the noise power $D^2(k, m)$.

$$
\hat{\kappa}_D(k, m) = \alpha_{\hat{\zeta}}(k, m) \hat{\kappa}_D(k, m-1) + (1 - \alpha_{\hat{\zeta}}(k, m)) \hat{D}^2(k, m).
$$

(35)

This idea allows reducing the speech power leakage in to the noise variance estimate and the noise variance estimate can updated reliably during the speech activity. Also, the speech presence probability does not need to be accurate, the smoothing parameter needs to be close to 1 less frequently and fast tracking can be achieved. This idea is resemble to SKS method and differs in some ways, i.e., in this method we use time varying smoothing parameter controlled by estimates of speech presence probability and we use $D^2$ the mmse estimator of the noise power instead of applying the heuristic weighting of equation (33). The optimal gain function $G_\alpha$ is calculated by iterative data driven method. From equation (32) The smoothing parameter $\alpha_{\hat{\zeta}}$ depends on estimate of speech presence probability, instead we use simplified estimation procedure $p$ is that allows for faster tracking and these are discussed in further sections.

The problem of estimating $D^2$ is reverse of that estimating of $A^2$ in some way. It is easy to calculate speech characteristics at high SNR and the opposite is true for the noise. Now we are only interested in finding the noise spectral variance. A Gaussian model is used, while the super Gaussian model gives the better results as it in equations (32), (33), and (36). In general speech characteristics change slowly and have many short pauses. Here we assume noise properties change more slowly than speech characteristics. This allows us to use of exponential smother in (31), (35) which responds slowly to changes in noise level than the estimated prior SNR does to changes in speech variance. Usage of exponential smother gives reduction in variance, still there are some limitations like how accurate can we estimate the noise variance?. However, it depends on how reliable we can detect the speech presence. One more limit is how fast we can track the noise. If we react to the unexpected changes in
noise spectrum, we start to track the speech, resulting the over estimation of the noise variance which is speech leakage problem and too much suppression in an enhancement setting.

4.4.1. Speech probability estimation

To reduce the speech leakage we update the noise variance with an estimate of the noise power instead of with noisy power, so that the errors in speech probability estimates have less severe consequences and we can use a simpler speech presence estimator. To take into account of strong correlation of speech presence in neighboring frequency bins, the posterior SNR is smoothed over neighboring frequency bins.

\[
\hat{\zeta}(k, m) = \sum_{i=-w}^{w} b(i) \zeta(k-i, m), \text{with} \sum_{i=-w}^{w} b(i) = 1. 
\]  

(36)

For speech presence,

\[
\hat{\zeta}(k, m) > T(k, m) \\
I(k, m) = 1 \text{ speech present} \\
\text{else} \\
I(k, m) = 0 \text{ speech absent} 
\]  

(37)

The speech presence probability estimate is updated with a first order recursion.

\[
\hat{p}(k, m) = \alpha_p \hat{p}(k, m-1) + (1-\alpha_p) I(k, m) 
\]  

(38)

\( \alpha_p \) lies between 0 and 1. This estimate is used in equation (32) to calculate the smoothing parameter in (35). This method is similar to the method in (37). In equation (37), the ratio of the smoothed noise spectrum and its local minimum is compared with a threshold. The local minimum in equation (37) is tracked by the method \[12\] [equation (40)], with an adaptation time of the minimum tracking about 0.5 s for non-stationary noise. Since, we only use posterior noise of the current frame in equation (37) in this method, we can react almost instantaneously to abrupt changes in noise levels. The parameter \( T \) in equation (37) controls the tradeoff between the tracking speed and the amount of speech leakage. The higher value gives faster tracking speed, but there is a high risk of speech leakage.

4.4.2. Prior SNR estimation

The gain function takes the prior and posterior SNR estimations as arguments. The prior and posterior SNRs to be calculated. From earlier discussions, we noted that noise tracking performance depends on the prior SNR estimator. From equation (29), decision-directed estimator is more suitable for speech spectral amplitude estimation, but we found that a modified estimator improves the noise tracking performance. So, we use a different estimator for the speech estimation and speech tracking.

\[
\hat{\xi}(k, m) = \max \left[ \alpha \frac{\hat{A}^2(k, m-1)}{\hat{\lambda}_0(k, m)} + (1-\alpha) \frac{R^2(k, m)}{\hat{\lambda}_0(k, m)} - 1, \hat{\xi}_{\text{min}} \right] 
\]  

(39)

Where \( \hat{\xi}_{\text{min}} \) is a small value but larger than zero. This estimator leads to less musical noise. Due to the recursive nature of the estimator, \( \hat{A}^2(k, m-1) \) depends on all previous noisy amplitudes
\( R(k, m-j), j > 0 \). \( A^2(k, m-1) \) summarizes the knowledge about current speech spectral variance \( \lambda_s(k, m) \) from previous noisy amplitudes. The equation (39), an estimator makes use of the correlation between the spectral amplitude of the consecutive frames. However, speech spectral amplitudes assumes to be independent of noise spectral amplitudes, So the equation (39) may not be much useful to track the noise where we are interested in estimation of \( \lambda_D(k, m) \). Now we have more reasons to looking for a modified prior SNR estimator for noise tracking.

Before using the modified prior SNR estimator there are some things to be remember, at first equation (39) is delayed at speech offsets and onsets. \( \xi \) in equation (39) is too small and the SNR is underestimated leads the speech power leakage in to noise variance estimates. Same case in offsets, noise variance will be underestimated. Also, any error in the noise variance estimate will affect the following prior and posterior SNR estimates.

The individual values in equation (39) will affect the whole idea. Consider, if \( \lambda_D \) is underestimated (overestimated), \( \xi \) and \( \zeta \) will be overestimated (underestimated), so the gain function \( G_D(\xi, \zeta) \) will be a too little value, leads the errors in \( \lambda_D \) and \( A^2 \) to be negatively correlated. These error of \( \lambda_D \) will tend to amplify itself in the first term of equation (39). It also affects the second term, because \( R^2/\lambda_D - 1 \) is equals to \( (R^2 - \lambda_D)/\lambda_D \), therefore the errors in numerator and denominator are also negatively correlated in this term. So to overcome these effects we use the below equation as input to \( G_D \) which is less sensitive to errors in \( \lambda_D \):

\[
\xi_{NT}(k, m) = \max \left[ \alpha_{NT} \frac{R^2(k, m-1)}{\lambda_D(k, m)} + (1-\alpha_{NT}) \frac{R^2(k, m)}{\lambda_D(k, m)}, \xi_{\min} \right] \tag{40}
\]

For equation (40), we use latest available estimates of the noise variance \( \lambda_D(k, m) \). This estimator is a smoothed version of posterior SNR because it uses the latest available noise variance estimates. In this algorithm we track signals that varies more slowly than those of speech, so we must use more smoothing in equation (40).

### 4.4.3. Safety net

Safety Net ensures that abrupt changes in noise power will not affect the performance of the noise variance estimation. This is the same problem in SKS method, i.e., if the noise level increases much faster than 10 dB/s and it continues stay at the same level, \( \zeta \) in equation (36) will be calculated on the basis of noise variance estimate is too low. Therefore it becomes more likely in equation (37), the algorithm react slowly. To overcome this issue we introduce a Safety Net giving an assurance that any abrupt changes in noise power level will not lead in to negative performance of the algorithm, while greatly improving the performance for unexpected increase in noise level.

The idea behind the safety net is to push noise variance estimate in a right direction when we detect that its value is much too low. For reference we use the minima \( P_{\min}(k, m) \) of the smoothed values \( P(k, m) \) of the noise power \( R^2(k, m) \) in a short window of length \( W_{\min} \),

\[
P(k, m) = P(k, m-1) + (1-\eta) R^2(k, m) \tag{41}
\]

Where \( \eta \) is a small smoothing parameter. By checking the following condition after updating the \( \lambda_D \),

\[
B.P_{\min}(k, m) < \lambda_D(k, m) \tag{42}
\]
Where $B > 1$ is a Correction Factor. If in the case of sudden increase in noise level that the algorithm cannot follow $B, P_{\text{min}}(k, m)$ will be larger than $\lambda_D(k, m)$ after a time of the order of the window length. In that case, we reset the $\lambda_D(k, m)$ values that violated the equation (42) to $\max \{B, P_{\text{min}}(k, m), D^2(k, m)\}$, and the corresponding $\hat{\lambda}(k, m)$ to zero. The correction factor $B$ is taken larger than 1, but very smaller than the bias correction factor that would apply if the window $w_{\text{min}}$ contains only noise. This gives assurance that safety net will not come in to action unintentionally when speech energy leaks in to the $P_{\text{min}}$. In addition to that we use only small values of $\lambda_D$ (small $\eta$) to calculate the $P_{\text{min}}$, so that it gives facility to keep the window $w_{\text{min}}$ short. For getting good results, the values of $B$ and the window length are not critical but still the minimum of 0.5 sec of window length is required.

Safety net is based on the simplified minimum statistics principle, but in our method it uses very small value’s of smoothing parameter ($\eta$ close to zero). It works in our method because we are not trying to estimate the exact noise variance from $P_{\text{min}}$. If we need to be so accurate then the values of $\eta$ (and $B$) must be larger. According to R. Martin (Spectral subtraction based on minimum statistics), if $\eta \geq 0.9$ window length must be $w_{\text{min}} \geq 0.8 S$. In our case we only use $P_{\text{min}}$ to speed up the convergence of the algorithm in case of abrupt change in noise level. Even at high SNRs this method does not leads to increase in speech leakage.

4.4.4. Gain function for noise estimation

For noise amplitude $D^2$, we use the mmse estimator because this is an unbiased estimator of $\lambda_D$ with low variance. Due to some reasons the optimal gain function is hard to derive, i.e., probability density function of speech DFT coefficients differs from a Gaussian model. This may cause complications when we are calculating the mmse estimator of $D^2$. Next reason is analytical gain functions are always derived under the assumption that speech and variance are known, but in general these are estimated, affecting the optimality of gain function. We use data driven method to find the gain function, but in an iterative fashion. The advantage of this method is, it does not make any explicit assumptions about the speech statistics and also can influence the inaccuracies in the estimated speech and noise variances.

Iterative data-driven gain optimization: The method in [13] makes use of large training database of speech material, contaminated with various levels stationary white Gaussian noise of known SNR. The prior and posterior SNRs are calculated for every frequency index and every time frame and each of these values are independent and separated in a grid typically by 1 dB step. Each pair of $(\xi, \zeta)$ having a corresponding pair of $(D^2, R^2)$ values associated with it. Statistics are taken for all training data, then one scalar gain value $G$ value is calculated for each grid cell such that the mean square error between $D^2$ and $(\hat{D}^2)$ is minimized. Which gives a two dimensional table containing optimal gain values of prior SNR and posterior SNR. Each grid in our method covers the range [-19 dB, 40 dB] for $\xi$ and [-30 dB, 40 dB] for $\zeta$ ( step of 1 dB for both), 25% of the speech training data consists of TIMIT-TRAIN database. White noise will be added to each file at the range of -12.5 to 27.5 dB with a step of 5 dB. For optimization of gain function frames with clean energy having more than 40 dB below the maximum clean frame energy of a speech sentence and also noise only frames are not taken in to account.

The data driven method is not going to work for noise estimation problem we are dealing with but, during the training noise variance $\lambda_D$ is known. When we use gain table $G_{(\xi)}$ with the noise tracking method, the noise variance is unknown but estimated by using $G_{(\xi)}$. The gain functions input parameters are depends on the quantity of $\lambda_D(k, m)$, which can be calculated using the same gain function.

A non linear recursion is introduced which was ignored during the training. Luckily, we can still optimize the gain function while taking in to account that the recursion by mean of an iterative scheme proposed next.
In further equation’s if the value of any quantity $i^{th}$ iteration will be denoted by a subscript $i$. For example the gain function for $D^2$ estimation with $i^{th}$ iteration will be written as $G_{D,i}$. To break the recursion $G_{D,i}$ is only used to compute the next iteration. The input parameters for the $G_{D,i}, \xi_{NT}, i(k,m), \hat{\xi}_i(k,m)$ will be depend on the data computed in the previous iteration. The below shows the optimization procedure,

a) Initialization ($i=0$).

\[ \hat{D}_i^2(k,m) = D^2(k,m), \hat{\lambda}_{D,i}(k,m) = \lambda_D(k,m) \]  (43)

b) Compute $\hat{\xi}_i, \hat{\xi}_{NT}, i$.

\[ \hat{\xi}_i(k,m) = R^2(k,m) / \hat{\lambda}_{D,i}(k,m), \hat{\xi}_{NT,i}(k,m) = \max \left\{ \alpha_{NT} \left( R^2(k,m-1) / \hat{\lambda}_{D,i}(k,m) \right) + \left( 1 - \alpha_{NT} \right) \left( R^2(k,m) / \hat{\lambda}_{D,i}(k,m) \right), \xi_{min} \right\} \]

collect ($D^2, R^2$) statistics per grid cell.

update $\alpha_{x,i}(k,m)$ according to (32),(36) and (38) and $\hat{\lambda}_{D,i}$:

\[ \hat{\lambda}_{D,i}(k,m+1) = \alpha_{x,i}(k,m) \hat{\lambda}_{D,i}(k,m) + (1 - \alpha_{x,i}(k,m)) \hat{D}_i^2(k,m) \]

\[ m := m + 1; \]

compute step (a) for all training data;

c) Minimize the mean square error in $\hat{D}^2$ for each grid cell,

\[ \Rightarrow G_{D,i+1} (\hat{\xi}_{NT}, \hat{\xi}) \]

d) Compute the data for next iterations,

\[ \hat{D}^2_{i+1}(k,m) = G_{D,i+1} (\hat{\xi}_{NT,i}(k,m), \hat{\xi}_i(k,m)) R^2(k,m) \]

e) $i := i + 1$

Go to step (a) if $i < i_{max}$.

This scheme typically converges in less than $i_{max} = 7$ iterations. In step 1) we are not applying the safety net because it is unnecessary as there is no abrupt change in noise level occurred during the training.

The $D^2$ are initialized with the true noise powers $D^2$ (noise initialization), then alternatively they can be initialized with noise power $R^2$ (noise initialization) or more clearly with speech power $A^2$ (speech initialization).

The normalized Mean Square Error is been computed with the following equation
\[ \text{mse} = \frac{\sum_u \sum_v \sum_w (D^2(u,v,w) - \hat{D}^2(u,v,w))^2}{\sum_u \sum_v \sum_w \hat{D}^4(u,v,w)} \] (15)

The equation above, variables \( u, v \) will run over all \( (\xi,\eta) \) cells, and the index variable \( w \) will run over all the data collected in each cell. The MSE increases on each step of iteration in the case of noise and noise initialization while in the case of speech initialization the MSE decreases. In all cases does the MSE converge to the same value of 0.082? After each iteration the estimated noise power becomes worst in case of noise and noisy initialization, instead when speech initialization is used it becomes better. After all the gain function and MSE converge to the same result.

The reason behind how this iterative formula is succeeds in optimizing for the practical case when there are recursions. The inputs for the next time frame are calculated from the output of the gain function in the current time frame. In our case, convergence means that \( G(D^2,u),i \) changes less and less from one iterations to the next when \( i \) increases. Eventually, the difference between \( \hat{D}^{i+1}_r \) and \( \hat{D}^i_r \) will be smaller and smaller. We will be in a same situation with recursion when \( \hat{D}^{i+1}_r \) and \( \hat{D}^i_r \) becomes almost equal. However we can not prove convergence, we are sure that we can find lowest possible MSE. By using the noise initialization, the MSE after step 2) has to increase, because in practice our noise power estimates are always contaminated by some speech power. This is different in case of using the speech initialization, after step 2) the MSE will decrease because our noise power estimates are not totally contaminated by some speech power.

From the figure 1(a), we can see that noise initialization and speech initialization are converge to the same MSE (there is no gap), which concludes that we cannot get more contamination in case of noise initialization and also not less contamination in case of speech initialization. In fig(b), we used optimized gain function \( G(D^2,i) \) on the training data recursively, the MSE becomes lower for all different initializations and converges at the same value like in fig (a) i.e., 0.082. We can write it as, iterative optimization finds a gain function and corresponding data \( D^2 \) with the lowest possible amount of contamination.

The gain function from the original iterative data-Driven scheme called \( G(D^2,0) \), is optimized using the true noise level. When data is applied recursively, it achieves MSE of 0.23. which confirms that out iterative optimization gives a better gain function, because the obtained mean square error by 65% using \( G(D^2,0) \).

In the next section we will discuss the improvement in the performance of noise tracking by updating the gain iteratively for various noise sources. The new noise tracking method \( G(D^2,i\text{max}) \) will be compared with MS and SKS method.

### 4.5. Simulation and results

The DRNPE based on single microphone algorithms tries to reduce the noise as much as possible. Unfortunately it is impossible to remove the noise completely. The main reason is that the single-microphone cannot distinguish between noise and speech, on the basis of where the sounds come from, for example, while this is to some extent possible when more microphones are available. That is why humans and animals have two ears!

Basically what the algorithm does is chop up the microphone signal in short intervals (called 'frames') and look at the frequency content in each frame. The part of the signal that has a fairly slowly changing spectrum over time is assumed to be the noise, while speech is assumed to change more rapidly. This is used to estimate the average noise spectrum.
n the noise suppression step, the frequencies with lots of noise are attenuated. So the idea is to apply relatively more attenuation to the frequencies with low signal-to-noise ratio. Unfortunately this means that if there is some speech present at those frequencies it will be attenuated as well. If you want to completely remove the noise, you will get a lot of speech distortion, and the algorithm has to balance this by letting some of the noise pass through.

In this section we are going to test the algorithm by using some non stationary noisy audio file which was recorded using a single microphone. To test the efficiency of algorithm, in this section we only used one audio file. In the next chapter we are going to elaborate the early discussed methods by using some more noisy files. The sample audio file was recorded with a cocktail-party noise and a male voice counting the numbers from one to ten and having a frequency of 16 kHz.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SNR</td>
<td>3.75 dB</td>
</tr>
<tr>
<td>Minimum error</td>
<td>943.82</td>
</tr>
<tr>
<td>System delay</td>
<td>0.000135 samples</td>
</tr>
</tbody>
</table>

Table 4.1: Results obtained after applying the single-microphone algorithm on noisy source.

Figure 4.1: Time-domain of a non-stationary noisy audio file recorded using DRNPE based on single microphone.
On average the system is having about 0.000130. We can expect different results using different methods because they interpolate between the samples differently. Resulting errors are approximately one thousandth of a sample. These results are good, in most cases people would be happy with an error of 0.1 samples, that concludes the efficiency of algorithm without having much delay. After all these results may look good to see but how efficiently a normal person can perceive the required sounds. As we discussed earlier, perception differs from from person to person. It is always tricky to say whether the algorithm is efficient or not. After listening to the both processed audio files using the single microphone and double microphone based algorithms, The DRNPE based on single microphone algorithm works better.

Figure 4.2: Time-domain of a noisy free audio file after applying the noise tracking algorithm
5. COMPARISON OF PERFORMANCE

In this chapter we are going to compare the two algorithms used to extract the noise from audio files. To compare the methods what kind of parameters would be better? Are these parameters would really evaluate the performance of the algorithm? Even if we get the good results does it really satisfies the end user? Can we decide the a better algorithm by perceiving the output of these algorithms? To conclude these questions, We took some parameters to evaluate the performance. In general these kind of algorithms are evaluated by using some parameters like SNR, System Delay, Minimum error and also perception. In our case we are using the same parameters to evaluate the performance of these algorithm.

During the testing I have observed something with the algorithm, there are some specific areas where this algorithm does perform so well, like the situation we are performing now, where there is no existence non stationary noise surrounded probably there is some echo due to the closed room. One more advantage with this algorithm is it can suppress the echo during the processing. In case if there is any non-stationary noise i.e., when we are not in any room probably in busy street or during the traffic or may be in a party, This algorithm does not perform well. The real problem with this algorithm is it depends on the placement of microphone always. My target is to find a suitable algorithm for hearing-aid device which will perform better in any kind of surrounded noise. The end user never know that what kind of circumstances he will be, So we should find an algorithm which can take any kind of noise.

To resolve the above discussed issue we found an algorithm which performs well in any kind of surrounded noise. To prove this we took some examples, which are recorded in surrounded room and in a public place and compared with several parameters and also I tried it with my friends to decide manually by listening to them to decide which performs well. According to them DRNPE based on single microphone does well in all situations.

To test the algorithm with several types of noisy background, I downloaded the noisy files from internet database. These two are music files. In the following diagrams, I’ll show you how these two original music files look like and also how will these look like when they are recorded by two microphones located in room with certain distance between them, while these music is running. Next two figures will demonstrates how these music files looks after running the algorithm. Then we will calculate the required parameters to estimate how far this algorithm does the job for us.
Figure 5.1: Time-domain of a original music file-1 in time domain

Figure 5.2: Time-domain of a original music file-2 in time domain
Figure 5.3: Time-domain of a microphone signal-1 in time domain

Figure 5.4: Time-domain of a original microphone signal-2 in time domain
Figure 5.5: Time-domain of a separated signal-1 in time domain

Figure 5.6: Time-domain of a separated signal-2 in time domain
In the above test, the first two observed microphone signals are original music signals and then the next two are recorded by two microphone separated by certain distance in a room. These signals are 16 bits and having a 22 kHz frequency with the play format (.wav). The signal separation is done by using FIR filters. At first second order statistics of the data are estimated, then by using this filters are calculated in frequency domain. The results shown below are after running the algorithm. The system delay is very high as compared to the DRNPE based on single microphone algorithm. The DRNPE based on single microphone results are shown further with the same audio input. The output is clear and as we discussed earlier this algorithm does so well in closed room.

<table>
<thead>
<tr>
<th>SNR</th>
<th>1.40 dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum error</td>
<td>5.47E+003</td>
</tr>
<tr>
<td>Delay</td>
<td>0.14</td>
</tr>
</tbody>
</table>

Table 5.1: Results after applying the BSS algorithm on speech contaminated by a music.

Figure 5.7: Auditory scene where speech is corrupted by background music
Figure 5.8: Time-domain of a first microphone recording (two persons speaking 4 sentences loudly)

Figure 5.9: Time-domain of a BSS - second microphone recording (two persons speaking loudly)
Figure 5.10: Time-domain of a BSS- one person's speech after applying the BSS algorithm

Figure 5.11: Time-domain of a BSS- other persons speech after applying algorithm
The above experiment is done with audio recorded in a real acoustical environment. This was recorded in a room, two persons read 4 sentences aloud. The loudspeaker is kept silent for this experiment. The resulting sound was recorded by two microphones which were spaced apart. The two recordings are 16 bit and at 24 kHz frequency. The separation filters are of length 512 and controlled by BSS algorithm.

<table>
<thead>
<tr>
<th>Category</th>
<th>Separated output-1</th>
<th>Separated output-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNR</td>
<td>1.68</td>
<td>0.54</td>
</tr>
<tr>
<td>Minimum error</td>
<td>5.35E+004</td>
<td>5.59E+004</td>
</tr>
<tr>
<td>System delay</td>
<td>0.15</td>
<td>0.39</td>
</tr>
</tbody>
</table>

Table 5.2: Results obtained after applying BSS on cross talk environment

The previous experiment is repeated again with a radio news on a small loud speaker. Two microphones were used to record the sound, the recorded sounds are 16 bit and at 24 kHz frequencies. The separating filters are of length 512, the BSS algorithm converges fast. To reduce the computational complexity only one update is done for every 2560 samples. With in 0.25 seconds the algorithm converges a good solution. This approach proved that, algorithm can do echo cancellation and double talk was never in consideration.
Figure 5.13: Time-domain of a BSS- first microphone recording of three people speaking

Figure 5.14: Time-domain of a BSS- second microphone recording of three people speaking
Figure 5.15: Time-domain of a BSS- separated signal-1 of the three speakers speaking simultaneously

Figure 5.16: Time-domain of a BSS- separated signal-2 of the three people speaking simultaneously
Figure 5.17: Time-domain of a noisy audio recorded using DRNPE based on single microphone.

Figure 5.18: Time-domain of a noisy free audio after applying a DRNPE based on single microphone algorithm.
The above experiment was done with same audio, that was used to test the BSS algorithm. The result shown in below are bit better than the BSS algorithm.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SNR</strong></td>
<td>3.76</td>
</tr>
<tr>
<td><strong>Minimum error</strong></td>
<td>943.82</td>
</tr>
<tr>
<td><strong>System delay</strong></td>
<td>0.43</td>
</tr>
</tbody>
</table>

Table 5.3: Results obtained after applying DRNPE based on single microphone algorithm on speech contaminated by music

![Figure 5.19: Time-domain of audio input used to test the BSS algorithm](image-url)
Table 5.4: Results obtained after applying the DRNPE based on single microphone on cross talk.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SNR</strong></td>
<td>2.7 dB</td>
</tr>
<tr>
<td><strong>Minimum error</strong></td>
<td>2.34E+003</td>
</tr>
<tr>
<td><strong>Delay</strong></td>
<td>0.35</td>
</tr>
</tbody>
</table>

Figure 5.20: Time-domain of a enhanced audio output of the DRNPE based on single microphone algorithm

Figure 5.21: Time-domain of a train noise including female speech recorded using DRNPE based on single microphone
Table 5.5: Results after applying the DRNPE based on single microphone with train noise including speech

<table>
<thead>
<tr>
<th>SNR</th>
<th>3.08</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum error</td>
<td>1.34E+003</td>
</tr>
<tr>
<td>Delay</td>
<td>0.000245</td>
</tr>
</tbody>
</table>

Table 5.6: Test results with babble noise using DRNPE based on single microphone.

<table>
<thead>
<tr>
<th>SNR</th>
<th>2.23</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min error</td>
<td>2.72097e+003</td>
</tr>
<tr>
<td>Delay</td>
<td>0.000255</td>
</tr>
</tbody>
</table>

Table 5.7: Test results with exhibition noise using DRNPE based on single microphone.

<table>
<thead>
<tr>
<th>SNR</th>
<th>1.26</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min error</td>
<td>3.40E+003</td>
</tr>
<tr>
<td>Delay</td>
<td>0.000133</td>
</tr>
</tbody>
</table>

Table 5.8: Test results with street noise using DRNPE based on single microphone algorithm.
5.1. Compiled results under various circumstances

The following results are obtained when the BSS and single-microphone algorithms are gone under test with various noise sources. Both algorithms are tested with the same noise sources to estimate which is better. As we can see the single-microphone algorithm giving constant results under various circumstances and the BSS varies according to the noise sources. These results would not decide completely whether which algorithm is better. Audio perception is crucial factor in deciding the performance of the algorithm. Totally seven cases are considered to compare the performance of the algorithms.

- Music background: The audio consists a speaker and a heavy noisy background music.
- Cross talk: Two persons are speaking loudly simultaneously, algorithm recovers single audio.
- Cross talk: Three persons speaking simultaneously.
- Train: Train noise and a clear speech are mixed up together in this audio.
- Babble: A baby speech consisting a meaningless confusion of words or sounds specking with very low or high voice.
- Exhibition: The audio source is from exhibition where noise changes abruptly.
- Street: The audio source is from a busy traffic street consists vehicle noise, beep sounds, loud speakers.

The reason behind these noise audio source is these are the places probably where person with disability are can not concentrate or hear on required speech. The BSS algorithm is giving competitive results compare to the DRNPE based on single microphone. There are several algorithms based on BSS which can perform better than single microphone but those require tricky mathematics. In this comparison the DRNPE algorithm performs much better in cases of SNR, Minimum error, system delay.

<table>
<thead>
<tr>
<th>Noise source</th>
<th>BSS algorithm</th>
<th>DRNPE based on single microphone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Music Background</td>
<td>2.4</td>
<td>3.7584</td>
</tr>
<tr>
<td>Cross Talk (two persons)</td>
<td>2.68</td>
<td>2.7006</td>
</tr>
<tr>
<td>Cross(three persons)</td>
<td>2.22</td>
<td>3.0780</td>
</tr>
<tr>
<td>Train</td>
<td>1.57</td>
<td>2.98</td>
</tr>
<tr>
<td>Babble</td>
<td>0.99</td>
<td>2.2343</td>
</tr>
<tr>
<td>Exhibition</td>
<td>0.53</td>
<td>1.2595</td>
</tr>
<tr>
<td>Street</td>
<td>1.37</td>
<td>2.1398</td>
</tr>
</tbody>
</table>

Table 5.9: SNR results for different background noise sources in dB
### Table 5.10: Minimum Error with different noise sources

<table>
<thead>
<tr>
<th>Noise source</th>
<th>BSS algorithm</th>
<th>DRNPE based on single microphone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Music Background</td>
<td>4.47E+003</td>
<td>3.44E+002</td>
</tr>
<tr>
<td>Cross Talk (two persons)</td>
<td>3.95E+003</td>
<td>2.34E+002</td>
</tr>
<tr>
<td>Train</td>
<td>4.59E+004</td>
<td>2.72E+002</td>
</tr>
<tr>
<td>Babble</td>
<td>4.9326e+003</td>
<td>3.40E+002</td>
</tr>
<tr>
<td>Exhibition</td>
<td>3.5602e+003</td>
<td>2.81E+002</td>
</tr>
<tr>
<td>Street</td>
<td>3.83E+003</td>
<td>2.60E+002</td>
</tr>
</tbody>
</table>

### Table 5.11: System delay results with various noisy sources in samples.

<table>
<thead>
<tr>
<th>Noise source</th>
<th>BSS algorithm</th>
<th>DRNPE based on single microphone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Music Background</td>
<td>0.14</td>
<td>0.000122</td>
</tr>
<tr>
<td>Cross Talk (two persons)</td>
<td>0.04</td>
<td>0.000177</td>
</tr>
<tr>
<td>Train</td>
<td>0.15</td>
<td>0.000245</td>
</tr>
<tr>
<td>Babble</td>
<td>0.39</td>
<td>0.000255</td>
</tr>
<tr>
<td>Exhibition</td>
<td>0.042612</td>
<td>0.000133</td>
</tr>
<tr>
<td>Street</td>
<td>0.022612</td>
<td>0.000197</td>
</tr>
</tbody>
</table>
6. **SUGGESTED FUTURE WORK**

The goal of thesis work is fulfilled successfully by suggesting a suitable algorithm. The following areas are still need to be explored to further optimization of algorithm and make it suitable under any kind of noisy source. This can be done by optimizing the following parameters used in algorithm.

- alpha NT: smoothing parameter in prior SNR parameter for noise tracking
- alphas E: smoothing parameter in decision-directed prior SNR estimator
- G_A : Gain function for clean speech amplitude estimation
- G_A2: Gain function for clean speech power estimation (used in the decision-directed prior SNR estimator)
- G_D2: Gain function for noise power estimation(used in the noise tracker).
- Safetynet : Safetynet ensures to track the sudden change of noise level. The effectiveness of safetynet depends on the smoothing parameter $\eta$.

These parameters optimized further so that the algorithm can sustain under various noise sources.

- Implementation in HDL language: This thesis work was done in MATLAB and tested in MATLAB. The proposed algorithm is implemented in MATLAB script. Algorithm needs to be tested with hardware after implementing the algorithm in any HDL language. A part of the project is already implemented where a sound signal from a CD player is modified by using a FPGA board i.e., Altera Cyclone II 2C35 FPGA. After implementing the algorithm in HDL language just mount it in FPGA to test it in real time hardware.

The above suggested future work will make the algorithm much more sophisticated and robust for various kinds of noisy environments. This helps the hearing-aid device to work better as it can filter out the unwanted noise from the desired speech.
7. CONCLUSIONS

This thesis work was carried out on a purpose of finding a robust, sustainable to all kind of noisy environments and low mathematical complexity algorithm for hearing aid and it fulfilled the purpose by suggesting a DRNPE based on single microphone algorithm. This conclusion is made upon by the comparison results achieved between DRNPE based on single microphone and BSS.

In a process to find the suitable algorithm, came across several algorithms which are quite competitive and showing significantly good results in their specialized kind of noise sources i.e., not able to applicable to all kind of noisy environments. The BSS algorithms based on the ICA property are showing for all kind of noisy environments but with a large mathematical complexity which can not be afforded according to the thesis goals. CoBliss [9] based on ICA property and DRNPE based on single microphone invented by [10] are chosen.

BSS algorithm is based on the property of ICA and there are several algorithms are proposed based on this property. Some of them are BSS, CoBliss, EcoBliss. In this report these three explained briefly and chose CoBliss algorithm for it is low mathematical complexity algorithm for comparison between DRNPE based on single microphone and BSS. CoBliss algorithm is extension of blind source separation. The advantage of CoBliss is, it can perform joint acoustic echo cancelling and also blind source separation at low mathematical complexity. The performance is evaluated by using noisy audio sources including echo and demonstrated in this report. The mathematical complexity is significantly low as compare to normal ICA based algorithm. Moreover the advantage of Co Bliss algorithm is not only helps for acoustic echo cancelling and also it can applicable in double talk situations without any complications. Which is helpful in teleconferencing and hands free telephony.

The DRNPE based on single microphone algorithm is developed especially for speech contaminated with non stationary noise and the algorithm is capable of fast noise tracking. The principle behind the algorithm is “recursive averaging of the mmse estimates of the noise power”. The speech probability estimation will control the smoothing parameter. The main problem in DRNPE based on single microphone algorithm is it is hard to distinguish between noise and speech. The DRNPE based on single microphone algorithm manages this problem by using estimating the noise power instead of noisy power which strongly reduces the speech leakage in to noise variance estimates. This gives the simpler noise probability estimator which reacts instantaneously to abrupt changes in noise level.

At the time of suggesting the optimal algorithm for hearing aid, it was not easy as the BSS algorithm is giving competitive results to DRNPE based on single microphone. The DRNPE based on single microphone is suggested not only based on the results but also audio perception which is crucial. Audio perception is also crucial in suggesting the algorithm.

During the thesis work, It was really hard to study several algorithms and eliminating to make a final list was not easy. Some of the audio samples were took from International Dialect for English Archive (IDEA) and noise audio sources are collected from internet database.

Algorithms based on multiple microphones can perform better than algorithms for one microphone because the former have some information about the direction of the sound sources available. Further research was not done on Blind Source Separation based on Independent Component Analysis, but there are much more computations involved than for the DRNPE based on single microphone case.
BIBLIOGRAPHY

1. Benesty, Jacob, Chen, Jingdong, Huang, Yiteng, Microphone Array Signal Processing, 2008,
4. R.H. Lambart, Multichannel Blind Deconvolution FIR Matrix Algebra and Separation of Multipath Mixtures, 1996,
8. Daniel Schobben, Daniel Schobben's Blind Signal Separation page, 1999,
12. G. Doblinger, Computationally efficient speech enhancement by spectral minima tracking in sub bands, 1995, 1513–1516