

1-1-2012

Under-Determined Blind Source Separation

Mehrdad Kafaiezadtehrani,
Ryerson University

Follow this and additional works at: <http://digitalcommons.ryerson.ca/dissertations>



Part of the [Electrical and Computer Engineering Commons](#)

Recommended Citation

Kafaiezadtehrani,, Mehrdad, "Under-Determined Blind Source Separation" (2012). *Theses and dissertations*. Paper 1500.

This Thesis is brought to you for free and open access by Digital Commons @ Ryerson. It has been accepted for inclusion in Theses and dissertations by an authorized administrator of Digital Commons @ Ryerson. For more information, please contact bcameron@ryerson.ca.

UNDER-DETERMINED BLIND SOURCE SEPARATION

by

Mehrdad Kafaiezadtehrani, B.A.Sc, McMaster University, 2010

A thesis presented to Ryerson University
in partial fulfillment of the
requirements for the degree of
Master of Applied Science
in the Program of
Electrical and Computer Engineering

Toronto, Ontario, Canada, 2012

©(Mehrdad Kafaiezadtehrani) 2012

Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my thesis may be made electronically available to the public.

Abstract

UNDER-DETERMINED BLIND SOURCE SEPARATION

Mehrdad Kafaiezadtehrani

Ryerson University

Master of Applied Science, 2012

Electrical and Computer Engineering

The Under-determined Blind Source Separation problem aims at estimating N source signals, with only a given set of M known mixtures, where $M < N$. The problem is solved by a two-stage approach.

The first stage is the estimation of the unknown mixing matrix. The contributions made unravel a more precise and accurate tool which directly relates to the initialization of the clustering algorithm. Different schemes such as segmentation, correlation and least square curve fitting are used to take advantage of the sparsity of the sources. A significant addition involves applying linear transforms to produce a higher sparse domain.

Further, the second stage is the sparse source recovery using a Matching Pursuit algorithm. The contributions involve a Matching Pursuit algorithm with different optimization criteria that use a redundant dictionary. Atom extraction criteria are proposed and the significance of sparsity is emphasized by showing higher sparsity in the transform domain.

Acknowledgment

I wish to express my gratitude to my supervisor; Dr. Kaamran Raahermifar from the Faculty of Electrical and Computer Engineering, Ryerson University, for his continuous support and supervision during the years of my study. It is his brilliant ideas and expertise that led this study to its successful outcome.

Special thanks to the staff of Electrical and Computer Engineering department for providing me with the essential tools needed to complete my research. Not forgetting one excellent Professor of the Mathematics department, Dr. Sebastian Ferrando for his help, effort and interest in my research.

My appreciation goes to everyone for their care, thoughtfulness, and devoted involvement in this study. Continuous support gave me the motivation to complete this study especially during the hard times.

Dedication

To Mom and Dad for all of their unending support

Contents

1	Introduction	1
1.1	Motivation of the Thesis	1
1.2	Assumptions	2
1.3	Aim of the Thesis	3
1.4	Outline of the Thesis	5
2	Literature Review	6
2.1	Summary	14
3	Unknown Mixing Matrix Identification	16
3.1	Clustering Method	18
3.1.1	The Distance of a point to a Hyperline	18
3.1.2	Basics of the K-Hyperline Clustering Algorithm	20
3.1.3	Steps of K-Hyperline Clustering Algorithm	22
3.2	Robust K-Hyperline Clustering Algorithm	24
3.2.1	Proposed ε Neighborhood Initialization	25

3.2.2	Basis Partitioning using Correlation	26
3.2.3	Steps of Partitioning using Correlation	27
3.2.4	Proposed Initialization using Polar Coordinates & Least Square Curve Fitting	29
3.2.5	Least Squares Curve Fitting	31
3.2.6	Steps of Proposed Polar Coordinates & Least Square Curve Fitting .	31
3.2.7	Proposed Sparsity Factor of the Sources	34
3.3	Summary	36
4	Separation of Sparse Sources	37
4.1	Dictionary	38
4.1.1	Gabor dictionary	38
4.2	Matching Pursuit Algorithm	40
4.2.1	Steps of Matching Pursuit Algorithm	41
4.2.2	Implementation of Gabor Atom	42
4.2.3	Optimal Atom Definition	43
4.2.4	Single Channel Atom Extraction	44
4.3	Multi-channel Matching Pursuit algorithm and Cumulative Atom Selection .	49
4.3.1	Steps of Multi-Channel Matching Pursuit algorithm using Cumulative Atoms	54
4.3.2	The Proposed Multi-channel Matching Pursuit algorithm & Maximum Residue Atom Selection	57

4.3.3	Steps of Proposed Multi-Channel Matching Pursuit algorithm using Maximum Residue Atom	59
4.4	Binary Masking	60
4.4.1	Recovering Sparse Sources via Binary Masking	61
4.5	Proposed High Sparsity via Transform Domain	63
4.6	Summary	67
5	Simulation and Results	68
5.1	Estimation of the Unknown Mixing Matrix	68
5.2	Separation of Sparse Sources	79
5.3	Summary	91
6	Conclusion	92
6.1	Future Work	94

List of Figures

1.1	Blind Source Separation Scenario	1
1.2	Block Diagram of Blind Source Separation	5
3.1	Disjoint Temporal Support,(Gribonval & Zibulevsky,[13])	17
3.2	Distance from Point P to line L	19
3.3	Scatter Plot (Mixtures 1 & 2)	30
3.4	Scatter Plot (Mixtures 1 & 2) Without Linear Transform	35
3.5	Scatter Plot (Mixtures 1 & 2) After applying Linear Transform	35
4.1	Gabor Function (Blue line represents the real Gabor function and the red line is the Gaussian envelope)	39
4.2	Matching Pursuit algorithm Using Real Gabor Dictionary: On the left (a) there exist an original signal and on the right (b) the recovered approximation of the signal produced by the Matching Pursuit algorithm.	47
4.3	Projection Coefficients: 397 Coefficients used to recover the signal in Fig. 4.2(b) show a density function of a super Gaussian distribution	48

4.4	Error & Residue: Error in the left decreases as more of the signal is decomposed and after 397 iterations the residue has become nearly zero shown on the right	48
4.5	Source Signals: 4 Audio signals of 800 sample points with full temporal support	55
4.6	Mixture Signals: 3 Mixtures are produced by a random mixing matrix	56
4.7	4 Recovered Source Signals by multi-channel Matching Pursuit algorithm using Cumulative Atom Selection	57
4.8	4 Recovered Source Signals by multi-channel Matching Pursuit algorithm using Maximum Residue Atom Selection	60
4.9	Recovered Source Signals by multi-channel Matching Pursuit Algorithm using Transformed Domain	66
5.1	Confidence Plot of Correlation Method	69
5.2	Scatter Plot of 3 Mixtures	70
5.3	Least Square Linear Curve Fitting	71
5.4	Confidence Plot of Proposed Method	72
5.5	Confidence Plot of Correlation Method	74
5.6	Confidence Plot of Proposed Method	75
5.7	Scatter Plot of Mixtures with Non-Disjoint Temporal Support	76
5.8	Scatter Plot in Transform Domain	77
5.9	Confidence Plot of Correlation Method	77
5.10	Confidence Plot of Proposed Method	78
5.11	Original Sources	80

5.12 Mixtures	81
5.13 Recovered Sources by Cumulative Atom Selection	82
5.14 Recovered Sources by Maximum Residue atom Selection	83
5.15 Recovered Sources by Maximum Residue atom Selection Using FFT	85
5.16 Original Sources	86
5.17 Mixtures	86
5.18 Recovered Sources by Maximum Residue atom Selection	87
5.19 Mixtures	88
5.20 Original Sources	89
5.21 Recovered Sources by Maximum Residue atom Selection	90

List of Tables

5.1	SIR Performance (dB) of Matrix Identification Algorithm (Example 1)	. . .	70
5.2	SIR Performance (dB) of Matrix Identification Algorithm (Example 1)	. . .	72
5.3	SIR Performance (dB) of Matrix Identification Algorithms (Example 2)	. . .	75
5.4	SIR Performance (dB) of Matrix Identification Algorithms (Example 3)	. . .	79
5.5	SIR Performance (dB) of Sparse Source Recovery (Example 1)	84
5.6	SIR Performance (dB) of Sparse Source Recovery (Example 1)	85
5.7	SIR Performance (dB) of Sparse Source Recovery (Example 2)	88
5.8	SIR Performance (dB) of Sparse Source Recovery (Example 3)	91

List of Flowcharts

3.1 Hyperline Clustering Algorithm	23
3.2 Initialization Using Correlation	28
3.3 Initialization Using Polar Clustering and Least Square Linear Curve Fitting .	33
4.1 Real Gabor Dictionary	46
4.2 Multi-channel Matching Pursuit Algorithm	62

Chapter 1

Introduction

1.1 Motivation of the Thesis

The motivation of this work is to extract the underlying sources from mixtures without strong additional information about the individual sources or any constraints on the mixing procedure. One can picture a simple scenario shown in Fig. 1.1 as the problem of Blind Source Separation, where the system implemented must estimate the mixing matrix and recover the sources with only having knowledge of the mixtures presented by the sensors.

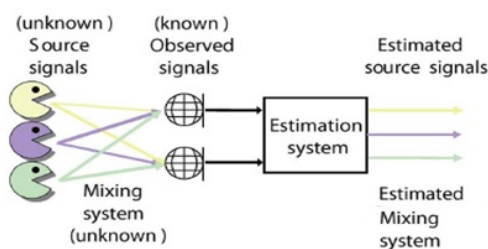


Figure 1.1: Blind Source Separation Scenario

Blind Source Separation is an attractive subject in the signal processing domain and has had many applications during the recent years such as biomedical engineering, wireless communication and data mining. Due to the unknown parameters of the mixtures and

an insufficient number of sensors, the Under-determined Blind Source Separation problem cannot be solved by using a standard approach such as Independent Component Analysis (ICA). Thus, due to existence of infinite solutions, a great deal of attention has been paid to the Sparse Component Analysis approach. Sparsity of the sources is an important property where the probability of two or more sources being active at once is low. In addition, the proposed estimation methods in this research also take advantage of these appropriate sparsity conditions. A desirable approach in sparse component analysis is a two stage approach where the first stage involves a matrix estimation and the second stage involves the source recovery. The ability of representing a signal in its most sparse form will lead to an increased number of practical signal processing algorithms in which sparse representation plays a key role.

1.2 Assumptions

The environmental assumptions regarding the surrounding in which the sensors record the samples increase the complexity of the problem. For analytical purposes the proposed algorithm makes less realistic assumptions about the environment to make the problem more tractable. The case of interest is the instantaneous case where all the sources arrive at the sensors at one time each having different intensity.

It is assumed that the source signals are sparse. Sparsity plays an important role within this frame of work. The advantage of a sparse signal representation is that the probability of two or more sources being simultaneously active is low. Therefore, the sparse representation accepts good separability because most of the energy in a basis coefficient belongs to one source. Such signals have the probability density function of a sharp peak at zero and fat tails that resemble the Gaussian distribution.

Assumption 1) Instantaneous mixing, where the source signals arrive at the sensors at one time with different intensities.

Assumption 2) The mixing matrix is $\mathbf{A} \in \mathbb{R}^{M \times N}$ where $M < N$ and has the property of

non-singularity that any $M \times M$ submatrix would have.

Assumption 3) The sources are sufficiently sparse, where each column of the source matrix \mathbf{S} has a single active source.

Assumption 4) Input signals and dictionary signals are finite in extent and do not exist beyond the boundaries of an interval.

1.3 Aim of the Thesis

In this research the problem of linear representation of a data set \mathbf{X} is given in the form of:

$$\mathbf{X} = \mathbf{A}\mathbf{S} \tag{1.1}$$

where $\mathbf{A} \in \mathbb{R}^{M \times N}$ and $\mathbf{S} \in \mathbb{R}^{N \times T}$. Here N is the number of source signals and, M is the number of observations with T as the duration of signals.

These linear systems have significant applications in systems such as decomposition of the objects into their elementary natural components or acknowledging the segmentations of the objects and its properties. Moreover, applications such as redundancy and dimensionality reduction, micro-array data mining, enhancement of speech and images in signal processing or nuclear medicine have become more popular in the recent years.

In this sequel, the problem of Blind Source Separation is solved by a two stage approach. In the first stage of the algorithm the mixing matrix is estimated. In order to estimate the mixing matrix, different clustering algorithms were studied and the Hyperline Clustering Algorithm was chosen. Since the sources are represented in the sparse form, the clustering algorithm uses the observed data points collected by the sensors to detect the true direction of the lines that are formed by the data. The matrix estimation by the conventional approach is achieved by overestimation of the randomized basis function. The contributions made within this particular stage involve the initialization of the system, which plays an important role in the detection of the true solution and the convergence of the system. Particularly, the proposed schemes such as:

- Initialization of the basis matrix using polar segmentation along with Least Square Curve Fitting.
- Employing linear transforms for more robust and efficient sparsity.

Next, the second stage of the Blind Source Separation problem becomes the separation of the sources from the mixtures with the estimated mixing matrix obtained from the first stage. The problem cannot be solved with a simple matrix inversion since it is one of the under-determined cases. Therefore, different approaches which allow the signals to be represented in their sparse form are looked at more in depth. The chosen algorithm which has a greedy but robust nature is the Matching Pursuit algorithm. In the single channel case the Matching Pursuit algorithm is able to represent the signal in its sparse form which is a series of unitary atoms along with appropriate coefficients. The dictionary containing these atoms is defined as a redundant Gabor Dictionary. The previous multi-channel Matching Pursuit algorithms use optimization criteria that extract one optimal atom for M mixture signals. The proposed multi-channel Matching Pursuit algorithm uses real Gabor atoms that can represent the sources in their most sparse form. The proposed schemes involve:

- Multi-channel Matching Pursuit algorithm using real Gabor Dictionary and maximum Residue atom Selection along with Binary Masking for Source recovery.
- Higher sparse Multi-channel Matching Pursuit algorithm using Fourier Transform and maximum Residue atom Selection along with Binary Masking for Source recovery.

The proposed Binary Masking scheme uses a least square method to detect an active source at an instance. By using such scheme it is possible to recover N sources from M mixtures successfully. Also, a significant addition to the contribution involves a higher sparse representation of data in the transform domain. In addition, both of the stages combined will work together to solve the problem of Blind Source Separation, an example is provided in Fig. 1.2.

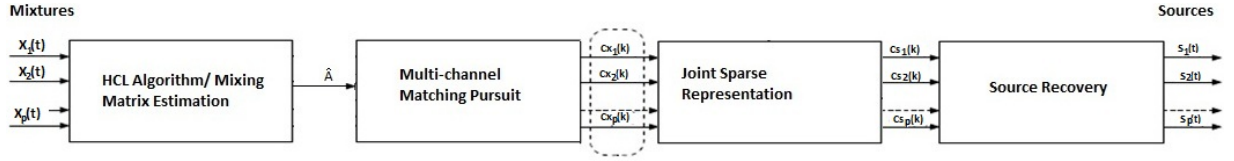


Figure 1.2: Block Diagram of Blind Source Separation

1.4 Outline of the Thesis

The thesis structure is detailed in Chapter 2 along with a literature review outlining most of the work related to Blind Source Separation. Chapter 3 explains in detail the estimation of the unknown mixing matrix using the K-Hyperline Clustering method and most important the contributions made on the initialization of the system. These contributions involve the initialization of the basis matrix using polar segmentation and least square curve fitting. Chapter 4 involves the sparse separation of the sources. In this section the appropriate multi-channel Matching Pursuit algorithm along with appropriate multi-atom Gabor dictionary are introduced. Chapter 5 shows the simulations and results and gives accurate comparisons for the latter and proposed topologies. In addition, Chapter 6 states the Conclusion and the Future Work that is suggested for this research area.

Chapter 2

Literature Review

Blind Source Separation has been studied for almost two decades. One of the earliest works traces back to 1986 by Herault and Jutten [16] where instantaneous linear combination of even-determined mixtures were presented and non-Gaussian sources were extracted under the assumption of independent source signals. This developed to a vast research area using different utilities and mathematical techniques that continues today with faster and more efficient topologies, which not only governs the fundamental and applications of electrical engineering but almost every field. During recent years a lot of interest, dedication and effort have been paid to this field. The works that are relevant to this topic and are worth mentioning, examine the mixing matrix parameters in the first stage and recover the source in the second stage. A variety of research has been done in both areas and ultimately there are different ways and techniques to combine for the purpose of the Blind Source Separation problem.

When a set of observations are presented from sensors such as microphones, the process of extracting the underlying sources is called Blind Source Separation. Generally the problem states that given M linear mixtures of N sources mixed by an unknown $M \times N$ mixing matrix \mathbf{A} , estimate the sources from the set of mixtures given. When $M \geq N$ will lead to constructing an un-mixing matrix \mathbf{W} , where $\mathbf{W} = \mathbf{A}^{-1}$. This work developed early on in 1953 and is famous for Darmois' Theorem and it was in later years used by Theis

[23] which states that the sources must be non-Gaussian and statistically independent. The dimensionality of the mixing matrix is also important to the complexity of the source separation. For example, when $M = N$ then the mixing matrix \mathbf{A} which is defined as a square mixing matrix and non-singular, can be estimated using linear transformation. However, when $M > N$ the mixing matrix \mathbf{A} is an over-determined matrix that is full rank where the estimation is done by least-square optimization or linear transformation involving pseudo inverse. Next, in the case of $M < N$ the mixing matrix is under-determined and source separation is done by non-linear techniques.

Further, there are environmental assumptions made with each model about the surrounding in which the sensors record the mixtures. These assumptions also relate to the complexity of the problem. The cocktail party problem was introduced and an adaptive blind separation was conducted by Cichocki [8] where the separation of sources using convolutive mixtures in an uncontrolled acoustic environment with propagation delays, multiple echoes and reverberation took place. Generally, the Blind Source Separation topologies used are different from the difficult real world scenarios and make less realistic assumptions about the environment. One of the assumptions being that the sources arrive instantly at the sensors but with different intensity belongs to the instantaneous mixture model. In addition, the anechoic model is the same as the instantaneous mixture model however there are arrival delays between sensors. The anechoic model is extended further to having multiple paths between sources and sensors which is the echoic case or called the convolution mixing.

There are also assumptions made about the sources such as the statistical properties of independence and being stationary. A signal is sparse when it contains zeros or is nearly zero more than expected from its variance. The probability density function of these signals has a sharp peak and heavy tails which looks like a Gaussian distribution. The sparse representation allows for good separability because most of the energy in a basis coefficient at any time instant belongs to a single source. Sparsity model is most effective when there are more sources than mixtures. This was studied in detail by Georgiev [12] where it could be seen that sparse representation of information is a breakthrough that also happens in the natural world, such as brain activity where neurons encode data in a sparse way. Neurons

encode data in a sparse matter if their firing pattern is characterized by long period of inactivity. During the recent years the solution for under-determined case was proposed by Belouchrani and Cardoso [4] which presented a maximum posteriori (MAP) probability approach for the discrete QAM sources. Later, an approach for sparse sources representation was proposed by Jourine [15] for an anechoic environment using the DUET algorithm, which introduced the first algorithm for moving sources.

The work done by Rickard [22] [27] involved a novel method for Blind Source Separation of N number of sources using $M = 2$ mixtures. The assumption made in this field was the disjoint orthogonality of the sources, where the support for the Fourier Transform is disjoint. This method estimated the mixing matrix by the clustering ratios of time-frequency representation of the mixtures of delayed sources. Next, the estimate of the mixing matrix was used to partition one of the data to recover the sources. Rickard [22] implemented a number of new results on the demixing degenerate mixtures where it relied on the assumption that the speech signals must satisfy the disjoint orthogonality condition. A downfall of such algorithm is the selection of appropriate window or the relationship between the statistical orthogonality condition and the disjoint orthogonality condition. Similarly, this work was also investigated by Aissa [2] where a general framework for the signal representation was proposed by two methods with respect to the time-frequency (TF) domain. In this work an assumption is made by allowing the sources to be non-disjoint to a certain extent. The separation can still be achieved due to subspace projection that allows identifying the sources present. In addition, Kisilev [17] also examined sparsity in other transform domains. The specific work studied demonstrated that the sparsity of the mixtures and original sources projected onto a proper space by applying a transform technique improves the quality of the separation.

The first set of algorithms which involve the estimation of the mixing matrix examine the process as determining the column of the matrix individually which allows a solution to be found for the under-determined case. A famous algorithm used is to recognize the line in a scatter plot using a clustering method. This was done by Zibulevsky [6] where the clustering was performed by normalizing the observation matrix and later mapping them

to a unit hemisphere. This reveals the line orientation of each source in both hemispheres, which ultimately produced two clusters for each source. The mixing matrix was estimated either by external optimization or by a clustering algorithm. Later given the mixing matrix, a minimal l_1 norm representation of the sources were obtained by solving a low dimensional linear programming problem for each data point. Within the work of Zibulevsky [6] it was illustrated that for the case of $M = 2$, both the number of the sources and the mixing matrix are estimated by the maxima of the potential function along the circle of a unit radius. Next, the minimal l_1 norm representation of the data by a linear combination of the pair of basis vectors that enclosed it are obtained. Some of the work that they did not consider were the case of $M > 2$ and a deeper study of the distribution of the sources and other sparse representation domains which would lead to improving sparsity.

A considerable amount of work has been done on the sparsity assumption along with representation of the signals in the sparse case. Rickard and Yilmaz [22] have proved that source signals are higher in sparsity in the time-frequency domain. They employed binary time-frequency masks for the separation of the sources from a single mixture. The assumption made was based on the time-frequency representation of the sources which do not overlap. The results demonstrated an ideal binary time-frequency mask that separated several speech signals from one mixture based on the anechoic model. The approach used was the maximum likelihood mixing parameter estimation and a power weighted histogram constructed from the ratio of the mixtures. The histogram is used to create time-frequency masks that partition one of the mixtures into the original source. The same concept was used by Bofill [5] where a histogram procedure was looked at for improving the potential function method used to estimate the mixing matrix in an instantaneous blind source separation model. The method describes how the idea of detecting single source data is implemented by selecting only the data which remain for two consecutive frames in the same spatial signature. Therefore, these data sets are most likely to belong to a single source and are used to accurately identify the spatial directions of the sources and finally the mixing matrix. The framework within their research strictly relies on the disjoint condition of the sources which is significant to source separation. Therefore, under different conditions there needs to be

modifications made to the algorithm. However, this has brought a lot of research interests into the field in terms of creating new algorithms with different transform domains.

Later work by Fabian and Lang [24] considered the under-determined blind separation where k means clustering was applied to estimate the mixing matrix based on the assumption that the sources are highly sparse. They extended this work by using median-based clustering which is another algorithm that recovered the mixing matrix using polar coordinate and the mixing angles for sparse cases. However the assumption that was made insisted that the given data must be of a delta distribution. In addition, by having this restriction on the data it performed well however without applying any transform domain the source recovery was difficult.

Another interesting method to find the direction of the line orientation is by performing eigenvector decomposition on the covariance matrix of the data. This method was conducted as early as 1901 and was used in the recent years by Raychaudhuri [21] which is a highly popular approach for the dimensionality reduction applications. The eigenvector with the largest eigenvalue indicates the direction of the line. This is an extension of the principal component analysis (PCA) which is useful in the clustering algorithms for the estimation of the mixing matrix. The same technique was employed by Gribonval [3] for the estimation of the number of audio sources and the mixing matrix of linear instantaneous mixtures. This approach uses multi-scale short time Fourier Transform and relies on the assumption that the time-frequency points are in the neighborhood of some scale where only one source contributes to the mixture. This assumption lead to a particular clustering algorithm called the Demix and one of Gribonval's [3] contributions. The work done by Gribonval challenges the scheme that uses the DUET algorithm or other similar sparsity-based algorithms, which rely on global scatter plots. The modified algorithm exploits a local confidence measure that weights the influence of each time-frequency point in the estimated matrix.

A similar approach for the estimation of the mixing matrix is the k -means algorithm. The algorithm is randomly initialized and each point is hard assigned to a closest line orientation. Each line orientation has a stochastic gradient algorithm associated which is

updated using the new point. The stochastic gradient algorithm is essentially an online PCA method. There are different forms of this algorithm existing today; one that was modified by O’Grady and Pearlmutter [20] which included hard data assignment and a batch mode of operation where the stochastic gradient was replaced with the Eigenvalue Decomposition of the covariance matrix. The approach is called an Expectation Maximization algorithm (EM).

Further, a significant breakthrough was studied by Aharon and Elad [1] where sparse representation for signals were studied more in depth. Using an over-complete dictionary that contained signal atoms allowed for sparse linear combination of signals. The concentration within this work involved mainly introducing the K-SVD algorithm as a generalization of the K-Means clustering process and demonstrated the behavior on both synthetic tests and in applications on real data. This work influenced more research areas on the K-EVD Clustering algorithms for sparse component analysis. Mainly, the work done by Cichocki and He [14] involving the estimation of the number of sources and mixing matrix by over-estimation of the basis matrix and iterative algorithm that used Eigenvalue Decomposition as a significant tool.

It is important to mention that the performance of these methods decrease when there is noise, outliers, non orthogonality or low sparsity. The algorithms used for the mixing model identification problem have good approximations of the true solution with minor errors. The contributions over the years have made a significant progress in the field, however there is still much more work to be done. Most of the research proposed still make several assumptions for analytical purposes, however these assumptions are not usually met in real world situations.

After estimating the mixing matrix the separation of the underlying sources can be performed. The complexity of the separation process is influenced by the mixing model used and the relative number of sources and sensors. In the case of under-determined source recovery which is usually restricted to sparse methods, a linear transformation is not possible. Therefore, some of the non-linear techniques that are looked at involve assigning

observed data $x(t)$ to the columns of the mixing matrix that characterize each source. The most basic technique is to hard assign each data point to only one source based on some measure of clusters to the columns of the mixing matrix, which was demonstrated by the work of Vielva [25]. First the mixing matrix is found by tracking its variation in the non-parametric maximum likelihood approach based on Parzen windowing and then the linear problem is presented by an estimator that chooses the best de-mixing matrix in a sample by sample basis taking advantage of some previous knowledge of the statistics of the sources. For this stage they have developed a MAP estimator that chooses the best inversion matrix. However, some statistical knowledge about the sources is needed. This is one of the down falls of their research.

Another way of separation is the partial assignment of each data point to multiple sources. This can be looked at by minimization of the l_1 norm or called the shortest-path algorithm. Bofill and Zibulevsky [6] have worked in this field and used the shortest path method to recover the sources. The underlying work by Bofill [6] assumes that the input distribution is sparse and the mixing matrix can be estimated either by external optimization or by clustering methods. The applied algorithm then involves a minimal l_1 norm representation of the sources which can be obtained by solving a low-dimensional linear programming problem for each of the data points. The work that they have demonstrated involves the estimation using $M = 2$ mixtures or in other words a $2 \times N$ mixing matrix. This is a 2 dimensional problem where a sparser representation is obtained in the transformed domain. Therefore, the mixing matrix is estimated first by the maxima of a potential function along the circle of unit radius. The drawback of using the mentioned algorithm is that l_1 norm minimization brings good separation if and only if the sources are disjoint or almost disjoint, regardless of whether they are Laplacian distribution or not. But when sources overlap the shortest path separation criterion, although statistically optimal, is unable to give the credit to the sources actually involved. The same work was later looked at by Donoho [9] where sparse signals were represented by l_1 norm minimization method. The underlying work involved representation of a dictionary for a given signal. The work compared l_0 norm minimization to l_1 norm minimization method where the latter approach

produced higher sparsity representation.

Also a different approach by Chen [7] was studied where the Basis pursuit (BP) algorithm applied a principle for decomposing a signal into an optimal superposition of dictionary elements. The optimal dictionary element is chosen by having the smallest l_1 norm of coefficients among all such decompositions. Greedy algorithms such as Basis pursuit along with l_1 norm minimization to recover sparse sources with over-complete dictionaries which correspond to the problems often encountered in under-determined source separation of multi-channel signal processing are of interest. A corresponding identifiable structure is set using a generalized version of Tropp's Babel function. Signals supported on this multichannel structure can be recovered exactly by both the Basis Pursuit algorithm and any other multi-channel greedy algorithm. In later years, Georgiev [11] improved the work done by Chen [7] where a new algorithm for the estimation of the mixing matrix as well as an algorithm for sparse component analysis was given.

One of the earliest works of greedy algorithms was the work by Mallat and Zhang [19] where they worked on Matching Pursuit algorithms with Time-Frequency Dictionaries. This involved a method that decomposed any signal to a linear expansion of waveforms that are selected from a redundant dictionary of functions. These waves were chosen in order to best match the signal structure and an adaptive signal representation was created. The Gabor functions were used which define adaptive time-frequency transforms. The matching pursuit algorithm developed for a single channel was extremely flexible in terms of signal representation since the dictionary was redundant. The time-frequency dictionary yields an adaptive decomposition tool where the signal structures are represented by atoms that match their time-frequency signatures. In later years, Ferrando [10] worked on a flexible implementation of the Matching Pursuit algorithm using Gabor dictionaries where the algorithm by Mallat was discussed in the discretized domain and the dictionaries implemented ultimately were a more accurate version. The goal of Ferrando's work was to provide a good synthesis tool for applications such as de-noising and super resolutions. A similar approach with a two stage method for the mixing matrix estimation and the source recovery were performed by Wang [26]. Wang used a clustering algorithm for the first stage and for the second stage of

source recovery the greedy Matching Pursuit algorithm was used.

2.1 Summary

The performances of these proposed algorithms are accurate based on the assumptions that they make. Most of these methods produce similar results and try to solve the same under-determined problem. However, in each case there are some disadvantages which require improvement. By studying these methods it is evident that in the near future an ultimate adaptive topology which uses a combination of most of the different schemes stated will be looked at more in depth. However, in order to get to this stage more incremental research needs to be done and within this sequel there are a lot of contributions which consider a more robust technique. It is important to mention that the performance of these methods decrease when there is noise, outliers, non orthogonality or low sparsity. Similar issues have been found on the topic of matrix estimation amongst several authors. The assumptions made by authors share the same environmental aspects which enable more analytical scenarios for both of the stages of the problem. Assumptions such as degree of sparsity or fixed number of mixtures have been analyzed by these authors. The goal here is to produce a method which not only can estimate the mixing matrix accurately but to be able to recover the sources by a non-linear method. Some of the challenges that are looked at more in depth are solving the system when there are many outliers, ill-conditioned matrix or specifically a new clustering algorithm which can detect the number of the sources and ultimately the correct mixing matrix in the first stage. In the second stage an appropriate non-linear adaptive algorithm must be chosen with the appropriate optimization criteria to recover the correct sparse sources. There are different approaches that several authors have examined in detail. Algorithms such as Basis Pursuit, Matching Pursuit or l_1 norm minimization provide good separability results. A non-linear method is ultimately looked at more in depth for the second stage. The two stages that are of interest relate to the algorithms that in the first stage use Eigenvalue Decomposition or Singular Value Decomposition to estimate the mixing matrix and in the second stage a greedy algorithm that with appropriate optimization

criteria recovers the sources iteratively with the aid of a dictionary containing elementary functions.

Chapter 3

Unknown Mixing Matrix Identification

Here is presented a clustering algorithm for the mixing matrix estimation where all the appropriate variables within the system are denoted in the following manner. Let $\mathbf{x}(\mathbf{t})$ be an M dimensional column vector representing the output of M sensors at any time instant t . Also define \mathbf{X} as an $M \times T$ matrix corresponding to the data collected by M sensors of duration time $t = 1, \dots, T$ such that $\mathbf{X} = [\mathbf{x}(\mathbf{1}), \dots, \mathbf{x}(\mathbf{T})] \in \mathbb{R}^{M \times T}$, where $T \gg M$. Let $\mathbf{S} \in \mathbb{R}^{N \times T}$ be a matrix representing the source signals and denote $\mathbf{A} = [\mathbf{a}_1, \dots, \mathbf{a}_n] \in \mathbb{R}^{M \times N}$ as the full rank mixing matrix consisting of \mathbf{a}_n columns. The problem of the Blind Source Separation raised involves solving the following system in the noiseless case:

$$\mathbf{X} = \mathbf{A}\mathbf{S} \tag{3.1}$$

where \mathbf{A} and \mathbf{S} are both unknown.

When the sources are sufficiently sparse, it then becomes necessary for estimation methods to consider the disjoint orthogonality condition. Several clustering algorithms such as, K-means, fuzzy-C clustering or median base clustering methods can be applied to estimate the mixing matrix. However, the performance of these methods usually decay with noise, outliers, non orthogonality or low sparsity and even with an ill-condition basis matrix \mathbf{A} .

Further, most test cases have revealed poor quality in estimating the number of unknown sources which in return is one of the dimensions of the matrix estimated, ultimately causing unsatisfactory results.

Therefore, the proposed method to overcome such issues is a robust K-Hyperline Clustering algorithm known as K-HCL, which relies on the existing K-SVD method for Hyperline clustering. This research conducted by Aharon and Elad [1] relates to the K-SVD estimation of the basis matrix using line orientation clustering methods. When the sources are sufficiently sparse it is necessary to consider the disjoint temporal support condition. Based on the theory explained by Gribonval and Zibulevsky [13] for this purpose, the set of points where the source with index \mathbf{n} is most active than others can be denoted by $\Lambda_n \subset \{1, \dots, \mathbf{T}\}$ representing the temporal support of source n . By this assumption of sparsity there exists a relationship where for all $t \in \Lambda_n$ one source is most active meaning that $|S_n(t)| \gg |S_m(t)|$. In addition, when the mixtures of the sources are plotted against each other, due to their disjoint temporal support it could be easily observed that the plots show alignments along the columns of the mixing matrix. The set of points are aligned along a straight line passing through the origin and directed by the columns of the mixing matrix denoted by \mathbf{a}_n . Therefore, one uses this intuitive observation combined with a more accurate clustering algorithm to estimate a precise mixing matrix. This property is observed in Fig. 3.1 where the top three signals denoted by $s_1(t)$, $s_2(t)$, $s_3(t)$ represent the sources and the following two plots denoted by $x_1(t)$, $x_2(t)$ are the corresponding mixtures.

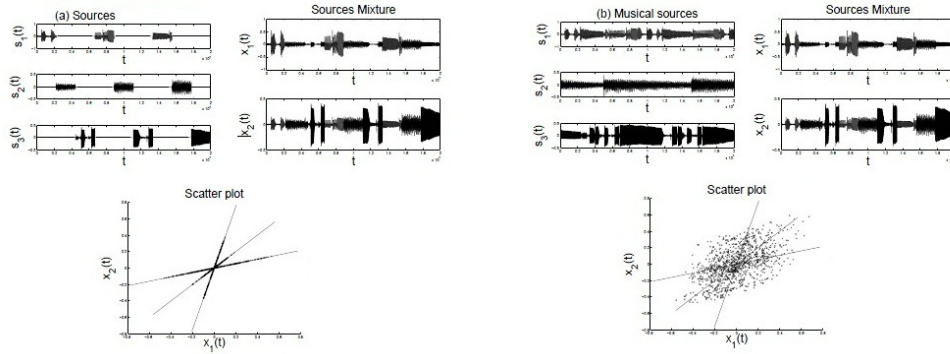


Figure 3.1: Disjoint Temporal Support,(Gribonval & Zibulevsky,[13])

It is shown in Fig. 3.1 how sparsity relates to the disjoint temporal support property where the scatter plot unravels the alignments along the direction of the columns of the mixing matrix. Further, it is displayed that when the sources do not comprise of this property, an appropriate transform can be applied to take advantage of the disjoint property once again. For example, in the musical case the scatter plots do not reveal any alignments towards the columns of the mixing matrix. This will be looked at closely in section 3.2.7. The Blind Source Separation is designed in a two-step algorithm in this context, and both require high sparsity for high temporal support.

3.1 Clustering Method

The K-HCL method overestimates the number of sources and in the subsequent stages it removes all the spurious basis hyperlines. The algorithm attempts to group a set of points from a given set of M dimensional observed data denoted by \mathbf{X}_t , where $t = 1, \dots, T$. The procedure considers a set of K hyperlines and logically initializes the lines as a basis matrix and thereafter groups the set of points from the given data to the associated hyperlines. The weight function that decides which set of points are associated to a specific hyperline was investigated by He and Aharon [1] [14]. They have simply considered the weight function as the minimal distance of a point to hyperlines. A distance function is calculated more in depth in section 3.1.1. To summarize, the objective becomes the detection and estimation of the direction of hyperlines given sparse data.

3.1.1 The Distance of a point to a Hyperline

If a point is considered as $P(p_1, \dots, p_m)$ in an m -dimensional domain, the minimal distance from a hyperline within the same domain can be calculated by formulating the distance between P and an arbitrary point $z = (z_1, \dots, z_m)$ which lies on the hyperline. Let's denote the line as L where the direction vector is $l = (l_1, \dots, l_m)$. This will allow for the problem to be written as an optimization criterion given by:

$$\begin{cases} \min_z d(z) = \|p - z\|^2 \\ \text{subject to : } \frac{z_1}{l_1} = \dots = \frac{z_m}{l_m} \end{cases} \quad (3.2)$$

where $\min_z d(z)$ is the minimum distance between point P and Z_* in Fig. 3.2, and Z_* is the optimal solution to the optimization problem stated in Eq. 3.2.

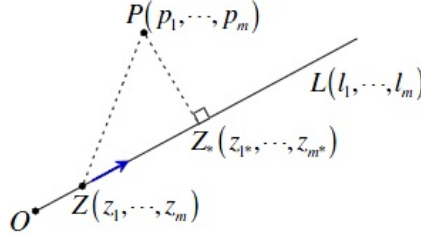


Figure 3.2: Distance from Point P to line L

Let's denote the subject condition in Eq. 3.2 as $\frac{z_1}{l_1} = \dots = \frac{z_m}{l_m} = v$. By doing so, it can be concluded that:

$$z = (z_1, \dots, z_m) = v \times (l_1, \dots, l_m) = vl \quad (3.3)$$

Using Eq. 3.2, the problem can be simplified such as:

$$\begin{aligned} \min_z d(z) &= \|p - vl\|^2 = \langle p - vl, p - vl \rangle \\ &= \{\langle p, p \rangle - \langle p, vl \rangle - \langle vl, p \rangle + \langle vl, vl \rangle\} \\ &= \{\langle p, p \rangle - v\langle p, l \rangle - v^*\langle l, p \rangle + vv^*\langle l, l \rangle\} \end{aligned} \quad (3.4)$$

where v^* is the conjugate value of v and $\langle \cdot, \cdot \rangle$ denotes the inner (dot) product of the two vectors. Next, taking the derivatives with respect to the conjugate z^* , gives:

$$\frac{\partial f(z)}{\partial R(z)} + i \frac{\partial f(z)}{\partial I(z)} = 2 \frac{\partial f(z)}{\partial z^*} \quad (3.5)$$

where $R(z)$ and $I(z)$ represent the real and imaginary parts.

In addition, using this property which ignores the non-conjugate occurrences of z , the derivative of the cost function represented by Eq. 3.3 with respect to v^* gives:

$$\frac{\partial d(.)}{\partial v^*} = -\langle l, p \rangle + v \langle l, l \rangle \quad (3.6)$$

In order to minimize the function, the derivative is set to zero by $\frac{\partial d(.)}{\partial v^*} = 0$ and the root is found as:

$$v = \frac{\langle l, p \rangle}{\langle l, l \rangle} \quad (3.7)$$

Next, by substitution of Eq. 3.7 and Eq. 3.3 into Eq. 3.2 the distance function will be:

$$d(p, l) = \langle p, p \rangle - \frac{\langle l, p \rangle \langle p, l \rangle}{\langle l, l \rangle} \quad (3.8)$$

3.1.2 Basics of the K-Hyperline Clustering Algorithm

Similar to Aharon & Elad's [1] approach the initial phase is to set a preliminary K clusters representing the vectors A_k , where $k = 1, \dots, K$. The number of clusters is usually greater than the number of the given data points, in order to cover most of the range and have a better initialization. Therefore, the initialized basis matrix is overestimated largely and iteratively the spurious hyperlines are neglected. Since the given data is of size $M \times T$, the initialized size of the mixing matrix is $M \times K$, where in theory $K \gg M$ initially.

Since the underlying work mostly involves sparse signals, it is necessary to have the data evaluated only at the significant coefficients. This means that the complete range of the data is compared to a small threshold ξ , and the ones smaller than this threshold are not of interest and are removed. Doing so will affect the mixture data such that the computational steps will decrease and also the entries that are unequally effected by the outliers are removed.

A simple way of accomplishing this is to evaluate the norm of each mixture column and removing the ones smaller than the threshold ξ . Let's denote the number of extracted columns from the mixture data as \hat{T} where $\hat{T} \leq T$. Therefore, the sub-matrix extracted is $\hat{\mathbf{X}} = [\mathbf{x}(\mathbf{1}), \dots, \mathbf{x}(\hat{\mathbf{T}})]$. Next, in order to block away the outliers, each of the columns of the observation mixture $\hat{\mathbf{X}}$ are normalized such that each column $\|\mathbf{x}(\hat{\mathbf{t}})\|_2 = \mathbf{1}$ for $t = 1, \dots, \hat{T}$.

After manipulating all the outliers and normalizing the data vectors, each vector is associated to a cluster. This evaluation and arrangement is illustrated by the distance relationship of a point and a hyperline examined previously in section 3.1.1.

By creating clusters as $\phi(\hat{A}_j)$ where $j = 1, \dots, K$, one can easily assign data points from the observation mixture into K clusters. According to Eq. 3.8 the distance from the observation point $x(t)$ to a cluster line of \hat{A}_j is calculated as $d(x(t), \hat{A}_j)$. Next the condition of arrangement between the data points and the clusters is evaluated by their minimal distance. Therefore, the observation points are all assigned to a specific cluster where all satisfy $d(x(t), \hat{A}_j) = \min \{d(x(t), \hat{A}_j), j = 1, \dots, K\}$. After completion of the assigning task, the result will reveal sub-matrices such that points are assigned to specific clusters.

Each cluster is updated by renewing the line orientation employing the Eigenvalue Decomposition (EVD). Let's assume that there exists \hat{T}_I points in each cluster $\phi(\hat{A}_j)$. Further, there are K sub-matrices to be updated along with their confidences. Noting that the size of each sub-matrices differs with associated observed data mixtures, then the Eigenvalue Decomposition is applied in the form of $\frac{1}{T_I}(\mathbf{X}^i)^H(\mathbf{X}^i) = \mathbf{V}\mathbf{D}\mathbf{V}^H$, where $i = 1, \dots, K$ is the index number of each cluster, T_I is the number of entries within each cluster and H denotes the transpose. Also $\mathbf{V}_k = [\mathbf{V}_1, \dots, \mathbf{V}_k]$ are a set of eigenvectors and $\mathbf{D}_k = \text{diag}[\mathbf{D}_1, \dots, \mathbf{D}_k]$ are the corresponding set of eigenvalues.

After decomposing the clusters using EVD, the line orientations are updated according to the vector of the largest eigenvalue. The largest diagonal value of the matrix \mathbf{D} is chosen along with the corresponding eigenvector for the updating purposes. The eigenvectors are sorted such that, $\mathbf{V}_1, \dots, \mathbf{V}_k$ correspond to $\mathbf{D}_1 \geq \dots \geq \mathbf{D}_k$ respectively. Therefore, the line orientations of the clusters $\phi(\hat{A}_j)$ are updated by \mathbf{V}_{1k} . Next the confidences are updated by the largest eigenvalues. This is simply shown as a confidence function of \mathbf{f}_k which is updated by $\mathbf{f}_k \leftarrow \mathbf{D}_{1k}$.

Ultimately this procedure is executed iteratively until appropriate convergence limits are satisfied. Since the cost function of this optimization problem decreases monotonically with number of iterations the convergence limit is set as $\|\mathbf{A}^{(\text{iter})} - \mathbf{A}^{(\text{iter}-1)}\| < \epsilon$. The advantage of this algorithm over others is in the size of the matrices that are passed to the EVD or

SVD as inputs. Most of the methods take an $M \times T$ matrix \mathbf{X} and perform EVD directly. However, the method described here only performs EVD operation on smaller matrices. It does so by passing a symmetric matrix $\mathbf{X}\mathbf{X}^H$ into the EVD algorithm producing a computationally efficient routine.

3.1.3 Steps of K-Hyperline Clustering Algorithm

Here the steps along with the flowchart of the clustering algorithm are mentioned which helps to understand and analyze the matrix identification algorithm much easier.

Step 1) Initialization: Initialize the basis matrix $\mathbf{A}^{\text{iter}} = [\mathbf{A}_1^{\text{iter}}, \dots, \mathbf{A}_K^{\text{iter}}]$, $K \gg N$, Load the observed mixture matrix composed of column vectors, $\mathbf{X} = [\mathbf{x}(1), \dots, \mathbf{x}(T)]$.

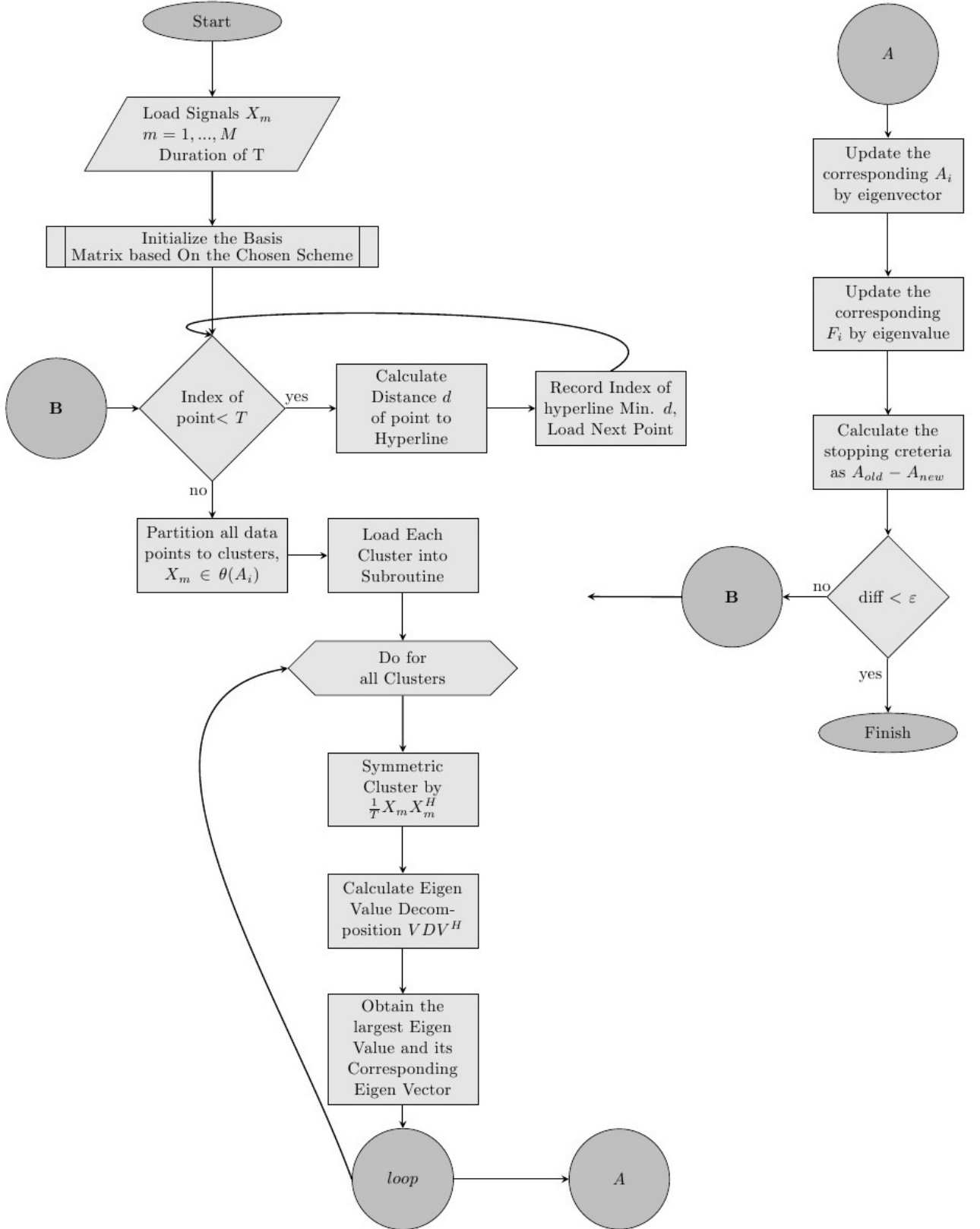
Step 2) Partition Clusters: Assign the observed points $x(t)$, $t = 1, \dots, \hat{T}$ into K different clusters set by $\phi(\hat{A}_j)$, $j = 1, \dots, K$. Based on the distance Eq. 3.8 the minimal distance is calculated for each observed data point to the hyperlines of the basis matrix by $\min \{d(x(t), \hat{A}_j)\}$ and the points are associated to the clusters if their distance is minimal.

Step 3) Update the Direction Matrix: Update the basis matrix \mathbf{A} and the confidence indices f_k . Apply the EVD to the clusters, by making the cluster symmetrical $\frac{1}{T}(\mathbf{X}^i)^H(\mathbf{X}^i) = \mathbf{V}\mathbf{D}\mathbf{V}^H$, where $i = 1, \dots, K$ and each \mathbf{X}^i contains T_i elements, where $\mathbf{V}_k = [\mathbf{V}_1, \dots, \mathbf{V}_k]$ are a set of eigenvectors and $\mathbf{D}_k = \text{diag}[\mathbf{D}_1, \dots, \mathbf{D}_k]$ are the corresponding eigenvalues. If $\mathbf{V}_1, \dots, \mathbf{V}_k$ correspond to $\mathbf{D}_1 \geq \dots \geq \mathbf{D}_k$ respectively then: $\mathbf{f}_k \leftarrow \mathbf{D}_1$ and $\mathbf{A}_k \leftarrow \mathbf{V}_1$

Step 4) Return to Step (2) until $\|\mathbf{A}^{(\text{iter})} - \mathbf{A}^{(\text{iter}-1)}\| < \epsilon$.

Step 5) Output and extract N significant hyperlines based on the largest confidence values in the gap, \mathbf{f}_j , $j = 1, \dots, K$.

In Flowchart 3.1 the basic HCL routine is presented which makes it easier to understand and allows for a clear picture of how the clustering algorithm operates, where it refers to another subroutine for the initialization scheme explained in section 3.2.



Flowchart 3.1: Hyperline Clustering Algorithm

3.2 Robust K-Hyperline Clustering Algorithm

The previously described algorithm has some flaws and disadvantages that need to be looked at closely for the estimation of mixing matrix. Even though, sparsity of the signals is known, the algorithm does not have a perfect rate of convergence in terms of detecting the number of hyperlines. This approach in return will actually overestimate or underestimate the columns of the mixing matrix and will produce inaccurate results. The contributions and the development of different schemes for appropriate initialization are much more significant due to the aid that they provide in escaping the local minima and the accurate convergence of the overall algorithm. Some of these issues are looked at closely and different methods are implemented and tested in order to create a more robust routine.

A significant way to estimate the number of hyperlines is by looking at the confidence function of K entries. When the function is plotted after the routine has converged, it is noticed that there are only a certain number of values that are significant and the rest are of really small values. In fact, this gap is noticeable and distinguishes between the significant confidence values and the spurious ones. Therefore, by initially overestimating the size of the mixing matrix and next, by looking at the confidence function, the size can be reduced by removing all the spurious vectors that correspond to the spurious confidence values.

The new robust algorithm essentially uses the basic HCL algorithm described previously with modifications to overcome the disadvantages. The search for the gap in the confidence function is a tool to approximate the true size of the mixing matrix and number of the hyperlines. An appropriate initialization becomes important since the clustering algorithm is tested to work and converge for cases that initially are close to the true solution. Let's denote this strict neighborhood close to the true solution as the ε neighborhood.

The K-HCL algorithm decreases monotonically as mentioned before and converges when the desired limit is met. Therefore, by having an initialized matrix that falls into the neighborhood of ε , the algorithm is able to find the optimal minima which corresponds to the true hyperlines. The true number of hyperlines are the exact number of sources

$k = 1, \dots, N$. In order to cover all the hyperlines the initialized basis of the clusters $\phi(\hat{A}_j)$ is set to a large dimension, such as $K = T$ and the values of the vectors are normalized. By normalizing the initial hyperline vectors the ε neighborhood range becomes really dense and therefore the N true directions are bound to be covered.

Therefore, the most important step in creating a robust and efficient algorithm becomes the initialization of the system. Having a robust initialization algorithm which falls into the ε neighborhood becomes more of interest. Some of the simple schemes for such task that have been tested are mentioned. For example a simple method is to extract K column vectors from the observed mixture \mathbf{X} . Typically to cover most of the true hyperlines a large number of K vectors are extracted. Another simple way of doing this is by initializing the hyperlines randomly within the range values of $[-1, 1]$, creating a dense neighborhood which will most likely cover the true hyperline directions. These schemes for their simplicity work well only in some cases and their performance decrease as the dimensions of the mixing matrix increases or when the observed data becomes large.

In addition, the initialization is considered the true direction scheme to an optimal convergence point. As mentioned by He [14], one can employ an existing clustering algorithm to estimate the initial direction of the hyperlines and set it as an initial basis.

3.2.1 Proposed ε Neighborhood Initialization

The goal of the contributions made are explained in this section which initialize the hyperlines as close as possible to the optimal solution knowing that sparsity exists for the source signals. In order to do so, some of the fundamentals explained previously are recalled here. Based on the sparsity of the source signals the existence of disjoint temporal support in the observation mixtures contribute an initialization that is in the ε neighborhood. Meaning that the true solution is in the direction of the mixing matrix columns when there exists high temporal support. Moreover, by looking at Fig. 3.1 the direction of the observation mixture scatter plots show this alignment. An intuitive solution to the problem can be obtained by creating hyperlines that are in the ε neighborhood. For example, for the case in Fig. 3.1

where there are three sparse source signals and two mixtures, one can initialize the system by creating hyperlines that have direction vectors along the scatter plot alignments.

Further this can be achieved by different ways. The best solution will ultimately have less computational weight, a better set of initial values as close as possible to the true solution and will converge in a reasonable negligible time.

3.2.2 Basis Partitioning using Correlation

The conventional method previously mentioned for initializing the system created a matrix that was overestimated and was reduced in size by removing the spurious vectors. Here a better approximation is obtained with segmentation of the overestimated initial hyperlines. Let's denote G as an arbitrary number, where $G \ll K$. Therefore, the original $M \times K$ pre-initialized direction matrix is segmented into K/G lower matrices each of size $M \times G$, let's denote this number of sub-matrices as scalar $P = K/G$.

Next, the hyperlines of each sub-matrix that is most correlated with the observation samples are selected as the representative for that particular range. This will produce an initial matrix that is of size $M \times P$ including the most correlated hyperlines with the data. Let's denote the first initialized hyperlines as $\mathbf{A}^{(0)} = [\mathbf{A}^{(0)}(\mathbf{1}), \dots, \mathbf{A}^{(0)}(\mathbf{K})]$. After segmentation there will be P numbers of sub-matrices that have hyperlines shown as $\mathbf{A}_{(i)}^{(0)} = [\mathbf{A}_{(i)}^{(0)}(\mathbf{1}), \dots, \mathbf{A}_{(i)}^{(0)}(\mathbf{G})]$ where $i = 1, \dots, P$. Next, the correlation of the samples $\mathbf{X}(\hat{\mathbf{t}})$ and each sub-matrix $\mathbf{A}_{(i)}^{(0)}$ for $i = 1, \dots, P$ are calculated. In addition, the set of hyperlines are chosen if and only if they satisfy $g = \arg \max_i |C(\mathbf{X}(\hat{\mathbf{t}}), \mathbf{A}_{(i)}^{(0)})|$. Here the maximum correlated hyperline is chosen and the same procedure is repeated until there are P numbers of hyperlines. This output of $M \times P$ initial direction matrix can be inputted to the K-HLC algorithm described in section 3.1.3. Applying such segmentation to the overestimated basis matrix and with the aid of correlation, a basis matrix is initialized that is close to the hyperlines of the true mixing matrix. Since the mixtures align along the columns of the mixing matrix, this approach will initialize the basis matrix with the highest correlated data points.

In order to have a robust and improved algorithm, at each stage the hyperlines along with their confidence indices are updated and when the convergence limit is met the spurious ones

are removed. The spurious hyperlines are removed by searching the gap of the confidence plot. The designed algorithm works accurate and efficient in the case where the number of sources is unknown and sparsity is a desired property of the signals.

3.2.3 Steps of Partitioning using Correlation

In order to make the Hyperline Clustering algorithm explained in section 3.2.2 easier to understand, the steps along with the flowchart are mentioned. One can analyze the overall algorithm step by step through reading this section.

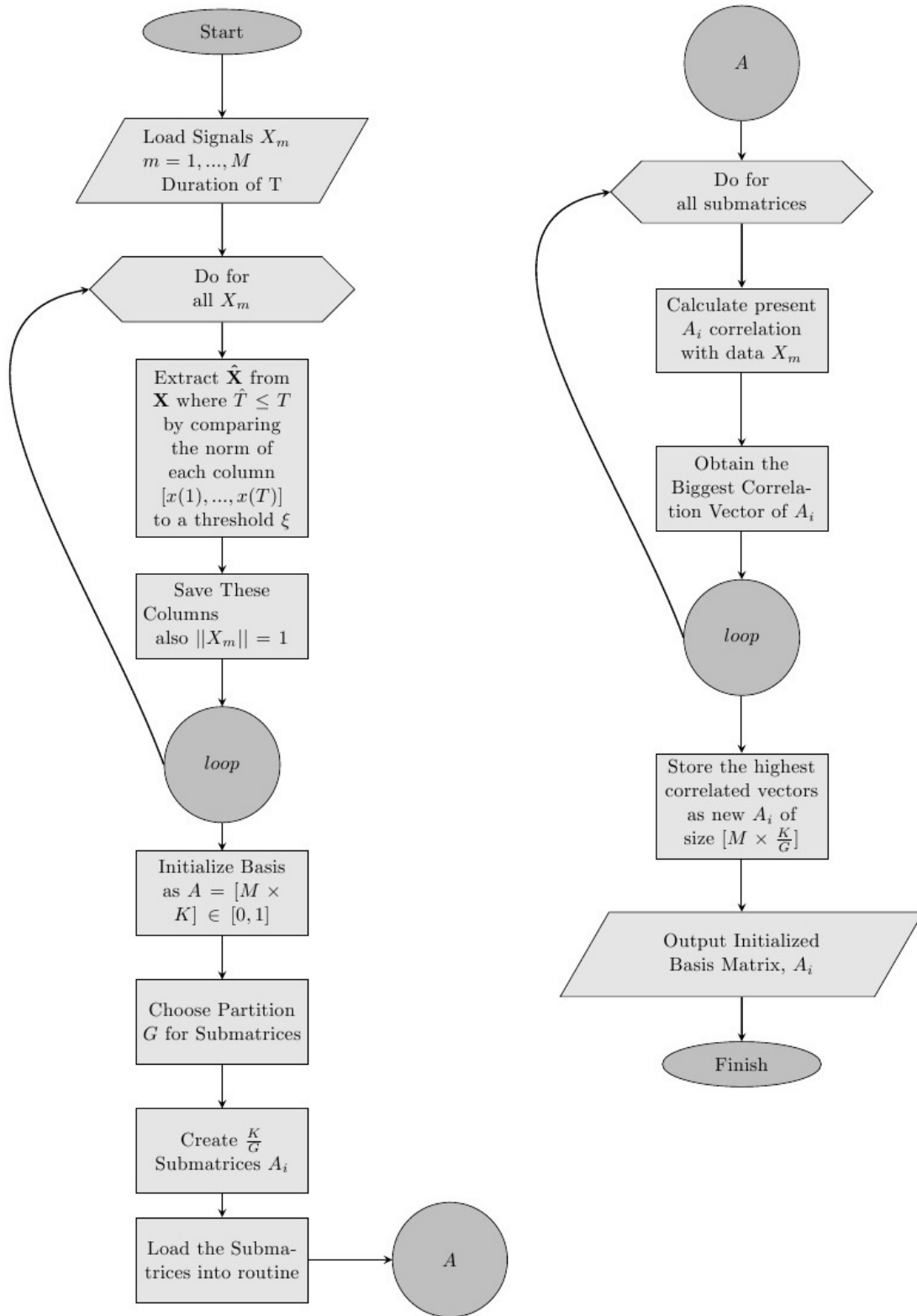
Step 1) *Partitioning*: Initialize the basis matrix $\mathbf{A}^{\text{iter}} = [\mathbf{A}_1^{\text{iter}}, \dots, \mathbf{A}_K^{\text{iter}}]$, $K \approx T \gg N$, then partition the $M \times K$ basis to $\frac{K}{G}$ segments where $G \ll K$ producing sub-matrices of size $M \times G$.

Step 2) Normalize and remove outliers by extracting submatrix $\hat{\mathbf{X}}$ from \mathbf{X} where $\hat{T} \leq T$ by comparing the norm of each column $[x(1), \dots, x(T)]$ to a threshold ξ , and setting the norm as $\|x(t)\|_2 = 1$ for $t = 1, \dots, \hat{T}$.

Step 3) Calculate the correlation coefficient of each sub-matrix and the observed data, and extract the basis vector from A_i^{iter} , $i = 1, \dots, \frac{K}{G}$ which has the maximum correlation coefficient. Extract an optimal direction vector from each sub-matrix and store in a new final basis matrix.

Step 4) Input the initialized basis matrix into the Hyperline Clustering Algorithm

In Flowchart 3.2 the initialization of the HCL algorithm is presented which uses correlation between the observed data and an overestimated randomized mixing matrix in a dense range. Since the data aligns along the direction of the columns of the true mixing matrix, the correlation employed sets the initial basis matrix in a neighborhood close to the true solution.



Flowchart 3.2: Initialization Using Correlation

3.2.4 Proposed Initialization using Polar Coordinates & Least Square Curve Fitting

The initialization method mentioned before uses correlation as a factor to forecast the correct direction of the scatter plot lines and generates random data as the basis vectors. This approach has its flaws in terms of impurities of inadequate sparsity in the observed data which ultimately can lead to under or overestimation of the basis matrix. Therefore, here a method is explained thoroughly which will improve the initialization criteria and will use less of a random process. The importance of correct initialization becomes one of the important tasks and contributions in this research, along with an appropriate iterative algorithm which employs the right tools for mixing matrix estimation. Therefore, the theory by Gribonval and Zibulevsky [13] which explains disjoint temporal support as a significant breakthrough for the matrix estimation is looked at closely. The theory explained in the previous chapter emphasizes that if only one of the sources is different from zero at each time instance, then all the observed mixtures appear to be proportional to the columns of the mixing matrix. Since high sparsity within sources means that there are more likely smaller coefficients to exist, and if at one time instance one is more active and significant, then the remaining are highly probable to be close to zero. Therefore, by looking at Fig. 3.1 it can be seen that the density of the mixture space shows a clear tendency to cluster along the direction of the columns of the mixing matrix. Therefore, finding the mixing matrix becomes the task of initializing the clustering algorithm as close as possible to the true solution.

The approach to estimating the correct ε neighborhood initialized basis vectors consists of partitioning the data into individual cluster lines and analyzing them separately. Here the case for $M = 2$ observed mixtures and $N = 4$ sparse sources is looked at closely. To separate each cluster line individually the polar coordinates are used as:

$$\begin{aligned} r(t) &= \sqrt{((x_1(t))^2 + (x_2(t))^2)} \\ \theta(t) &= \tan^{-1} \left(\frac{x_2(t)}{x_1(t)} \right) \end{aligned} \quad (3.9)$$

Since the cluster lines continue in both positive and negative quadrant of the plots, only one semi-plane of θ_k is discretized by taking K samples in the potential field and using equally

spaced grids such as:

$$\theta_k = \frac{\pi}{2K} + \frac{k\pi}{K} \quad (3.10)$$

where $k = 1, \dots, K$.

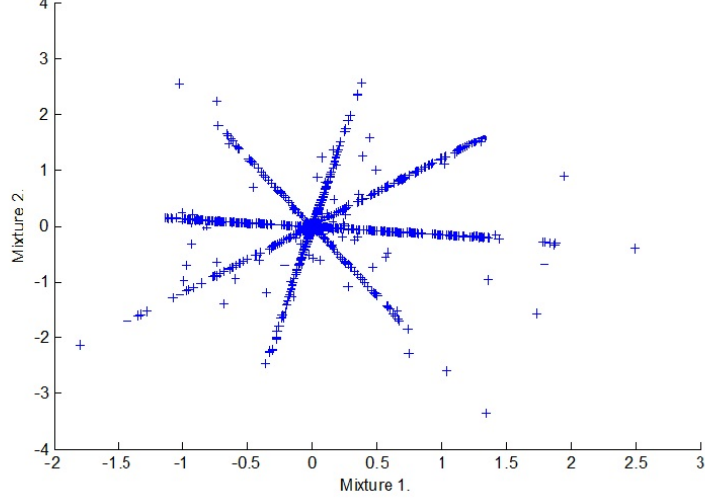


Figure 3.3: Scatter Plot (Mixtures 1 & 2)

Fig. 3.3 belongs to four sparse sources that have been mixed using a random 2×4 mixing matrix, the two following observed mixtures were then plotted against each other in a scatter plot. It is observed that the data point align along the direction of the columns of the mixing matrix and by just taking the top semi-plane, the four scatter lines can be presented in four different clusters. Therefore, by using the clusters defined in Eq. 3.10, the data points can be partitioned into these clusters. Further, it is more likely that the data which are along one direction vector will share the same cluster. By doing so, each cluster line at the top semi-plane can be distinguished and looked at separately. This is significant because the equation of each cluster line can be represented by a least squares curve fitting method which uses different degrees of polynomials. In addition, to keep the goal of ε neighborhood initialization in mind, the task becomes a simple procedure when the cluster lines can be separated and analyzed by a curve fitting method. The advantage of such scheme is that there is no need to overestimate the basis matrix hoping to cover the hyperlines. This particular scheme will initialized the correct amount of hyperlines according to the number

of cluster lines that contain most of the data. The line equations will be significant in the sense that one can initialize the system almost on the hyperline direction vectors that belong to the true solution.

3.2.5 Least Squares Curve Fitting

Let's consider the general form of the polynomial of order M :

$$f(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + \dots + a_Mx^M = a_0 + \sum_{i=1}^M a_ix^i \quad (3.11)$$

The goal of least squares becomes picking the coefficients of the equation in a manner where it best fits the curve to the data and one that gives the minimum error between the data and the fit. Therefore, the general expression for the error using the least square approach becomes:

$$\begin{aligned} error &= (y_1 - f(x_1))^2 + (y_2 - f(x_2))^2 + \dots + (y_K - f(x_K))^2 \\ error &= \sum_{k=1}^K (y_k - f(x_k))^2 \end{aligned} \quad (3.12)$$

where there are K data point presented in the top semiplane. Using the least square linear curve the equations of each cluster can be obtained and the basis matrix can be initialized perfectly.

3.2.6 Steps of Proposed Polar Coordinates & Least Square Curve Fitting

Step 1) Normalize and remove outliers by extracting submatrix $\hat{\mathbf{X}}$ from \mathbf{X} where $\hat{T} \leq T$ by comparing the norm of each column $[x(1), \dots, x(T)]$ to a threshold ξ , and setting the norm as $\|x(t)\|_2 = 1$ for $t = 1, \dots, \hat{T}$.

Step 2) *Partitioning*: Partition the polar plane into P number of cluster slots by $\theta_p = \frac{\pi}{2P} + \frac{p\pi}{P}$, with a unity radius, where P can be either the number of observed data points

or an arbitrary number based on the difference of angles between each cluster slots. For example $\theta_p - \theta_{p-1} \approx 0.5 \text{ degrees}$ would lead to a choice of $P = 360$.

Step 3) Calculate the polar coordinate distance given the unity radius of the cluster slot and its polar angle as θ_P with the given observed data, $x_1(t)$ & $x_2(t)$, where polar coordinates are calculated by Eq 3.9. The distance is given by:

$$d = \sqrt{r_p^2 + r_1^2 - 2r_p r_1 \cos(\theta_p - \theta_1)} \quad (3.13)$$

Step 4) Assign each observed data pair to a cluster if the distance is minimized as:

$$\theta_p = \operatorname{argmin} \sqrt{r_p^2 + r_1^2 - 2r_p r_1 \cos(\theta_p - \theta_1)} \quad (3.14)$$

where $p = 1, \dots, P$

Step 5) Now there exist a max P number of clusters with a total of T data points. Extract the clusters containing at least $\frac{T}{P}$ point.

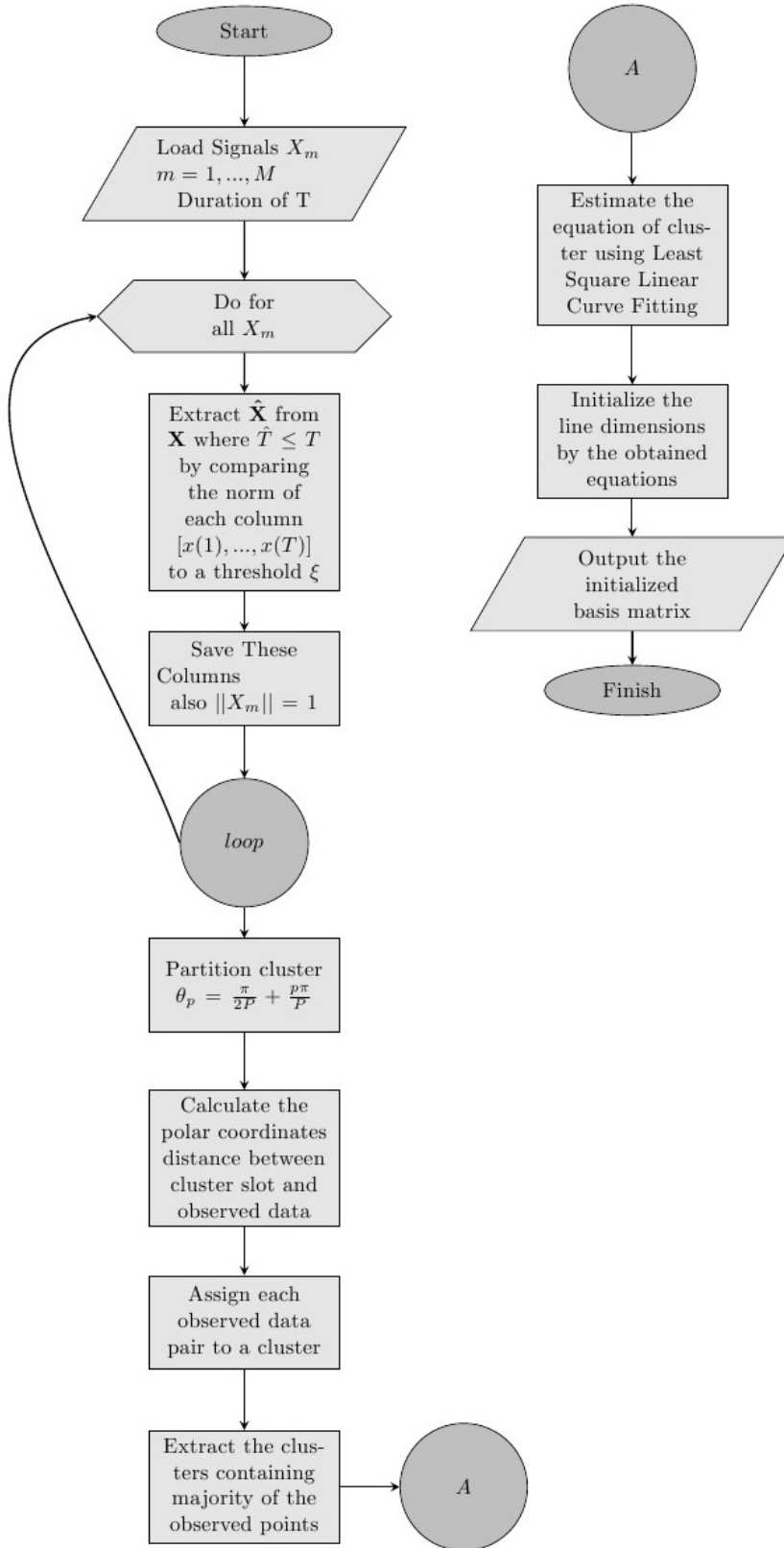
Step 6) For the extracted clusters use the Least Square Linear Curve Fitting to estimate the equation of the cluster lines given by:

$$X = A^{-1} * B$$

$$\begin{bmatrix} b \\ a \end{bmatrix} = \begin{bmatrix} n & \sum x_k \\ \sum x_k & \sum x_k^2 \end{bmatrix}^{-1} * \begin{bmatrix} \sum y_k \\ \sum (y_k x_k) \end{bmatrix} \quad (3.15)$$

Step 7) Initialize the basis matrix with hyperlines belonging to each cluster

Flowchart 3.3 reveals the proposed algorithm which is simple and efficient for the detection of hyperlines and initialization of the mixing matrix. Within this topology there is no need for overestimation of the basis matrix in the initialization stage. The sparsity of the sources align the mixtures along the columns of the true mixing matrix, and by polar partitioning and simple curve fitting tools a basis matrix is initialized in the ε neighborhood.



Flowchart 3.3: Initialization Using Polar Clustering and Least Square Linear Curve Fitting

3.2.7 Proposed Sparsity Factor of the Sources

Sparsity plays an important role in representation of the observed mixtures in a form that facilitate the estimation of the mixing matrix correctly, such as Fig. 3.3. Sometimes the data in the time domain which is our frame of work within this research does not satisfy the requirement of sparsity needed for the matrix estimation. In situations like this, it is good to look at the linear transform domain where the data will be higher in sparsity. This transformation is linear so the model can be re-written as:

$$T(\mathbf{X}) = \mathbf{A}T(\mathbf{S}) \quad (3.16)$$

Next, the clustering algorithm will be performed in the transformed domain. This approach is also tested for the under-determined case where $M < N$ and the observed mixtures are not as sparse in the time domain. Therefore, the observed mixtures are presented in a different domain using a linear transform such as Short Term Fourier Transform. In addition, this is accomplished by turning the observed mixture data into frames of length L and multiplying them by a window such as a Hanning Window. Next, a hop of distance d was used between the starting points of the successive frames. Further, each frame was transformed with a standard Fast Fourier Transform and the successive frames were concatenated in a single vector which was inputted into the clustering algorithm. The scatter plots in Fig. 3.4 show this relationship where the data with less sparsity has a big cloud of scatters where the direction of the mixing matrix is not distinguishable. However, after transforming the data using the STFT in Fig. 3.5, the scatters align along the columns of the mixing matrix, revealing their direction and presenting a higher sparsity domain.

It can be viewed that sparsity has increased when the linear transform is applied to the observed data. Here by taking the STFT, the observed scatter points from a big cloud in Fig. 3.4 have all lined up in the direction of the columns of the mixing matrix in Fig. 3.5. It is worthy to mention such scheme exist when the underlying frame of study does not consider sparsity of the sources and still tries to apply a clustering algorithm. With such a tool it will be much more reliable to apply such clustering algorithm and still obtain accurate results.

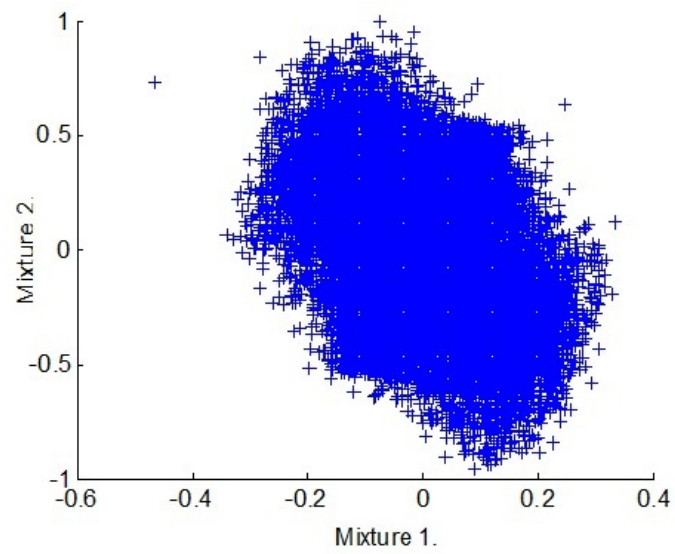


Figure 3.4: Scatter Plot (Mixtures 1 & 2) Without Linear Transform

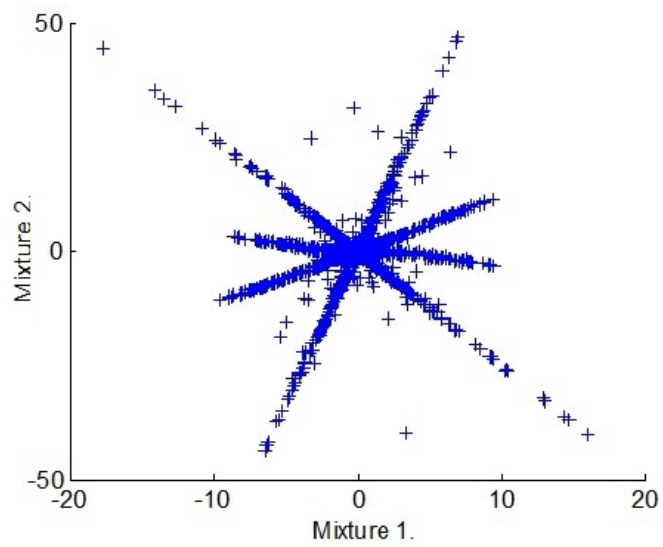


Figure 3.5: Scatter Plot (Mixtures 1 & 2) After applying Linear Transform

3.3 Summary

In Chapter 3 the first stage of the problem is addressed by discussing the appropriate schemes to estimate the mixing model. In addition, a clustering algorithm which assigns the observed data points to clusters based on the distance function is introduced. The hyperline clustering algorithm uses the Eigenvalue Decomposition tool to update each cluster and find the true direction of the mixing matrix. Next, the initialization scheme of the clustering algorithm becomes more significant and is discussed throughout the chapter. This initialization scheme determines the convergence rate, accuracy and performance of the algorithm. There are different initialization methods mentioned and the importance of the ε Neighborhood initialization scheme is emphasized. In order to initialize the hyperlines as close as possible to the true solution, the correlation method is discussed in depth. This method overestimates the initial basis matrix and creates a dense range that covers the direction of the columns of the true mixing matrix. Further, the proposed method looks at polar partitioning the observed data and separating each hyperline individually and analyzing them using least square curve fitting model. The proposed approach initializes the basis matrix as close as possible to the true mixing model. The essentials of these models rely on the disjoint temporal support which explains the density of the observed mixtures aligning along the columns of the true mixing model. In addition, a linear transform model is proposed to show that higher sparsity is gained when there exists no disjoint temporal support for the sources in the analyzed domain.

Chapter 4

Separation of Sparse Sources

The principle of sparse source separation consists of representing the sources by a linear combination of a few elementary atoms with unit energy denoted here by φ_k . Therefore, the sources can be approximated by these atoms in the form of:

$$s(t) = \sum_{k=1}^K C_s(k) \varphi_k(t) \quad (4.1)$$

where the scalar vector $C_s(k)$ represents the coefficients for the atoms. Most of these coefficients are near zero and only a few are significant, which makes the probability density function of the coefficients resemble a super-Gaussian distribution. In this context the signals are represented by their atoms where they are taken from a dictionary. Further, the model in Eq. 3.1 can be analyzed by its sparse representation such as:

$$x(t) = \sum_{k=1}^K C_x(k) \varphi_k(t) = \sum_{k=1}^K AC_s(k) \varphi_k(t) \quad (4.2)$$

The chosen scheme uses a redundant dictionary where the signals can be approximated in the form of Eq. 4.2. There is an infinite number of possible coefficient sets for reconstruction of each source since the dictionary is redundant. Selecting an appropriate joint sparse approximation scheme for the observed mixture depends on the algorithm's ability to provide a representation as $C_x \approx AC_s$.

4.1 Dictionary

Within a redundant dictionary there exist an infinite number of possible coefficient sets to choose from which allows for the reconstruction of a particular signal. A redundant dictionary is a set denoted by Φ that can be examined by a $K \times T$ matrix, where K rows are the atoms represented by $\varphi_k(t)$ such that $1 \leq t \leq T$ and $k = 1, \dots, K$. In addition, for a redundant dictionary there is an infinite number of possible coefficient sets for atoms $K > T$ that can be used to recover a particular signal. Choosing an appropriate type of dictionary is very crucial to the separation procedure. Since a sparse signal representation is chosen by the atoms within the dictionary, the particular nature of the signal determines the dictionary's relevance. In recent years, the study of developing different types of dictionaries that consider the nature of the analyzed signal has become more popular. Since the dictionary is the core of the signal decomposition algorithm, it should be designed in a manner where it contains all the possible signal representations. Therefore, designing a robust dictionary for a particular application becomes significant based on the nature of the presented data. With a poor dictionary the signals would be represented by more atoms and the time of convergence would be larger than employing a dictionary that is more complete of different atoms.

4.1.1 Gabor dictionary

Here the dictionary chosen uses the Gabor atoms. The Gabor dictionary implemented is built with a Gaussian window function denoted by $g(\cdot)$. The dictionary is scaled, translated and modulated as stated:

$$g_{(s,u,v,w)}(t) = \frac{K_{(s,u,v,w)}}{\sqrt{s}} g\left(\frac{t-u}{s}\right) \cos(vt + w) \quad (4.3)$$

where $K_{(s,u,v,w)}$ is the normalizing coefficient and (s, u, v, w) and t are continuous parameters. Next the function is defined for a discrete period of N , comparing to the signal $f = (f[0], f[1], \dots, f[N-1])$ where it is sampled at $t = i$ and $i = 1, \dots, N-1$. Therefore, the Gabor function in Eq. 4.3 is also sampled at time $t = i$ and made periodic.

The parameters chosen for this particular dictionary are $(s, u, v) = (a^j, pa^j \triangle u, ka^{-j} \triangle v)$ where $\triangle u = \frac{1}{a}$, $\triangle v = \pi$, $0 < j < \log_2(N)$, $0 \leq p < Na^{-j+1}$ and $0 \leq k < a^{j+1}$. Fig. 4.1 shows the Gaussian window envelope with Gabor function isolating.

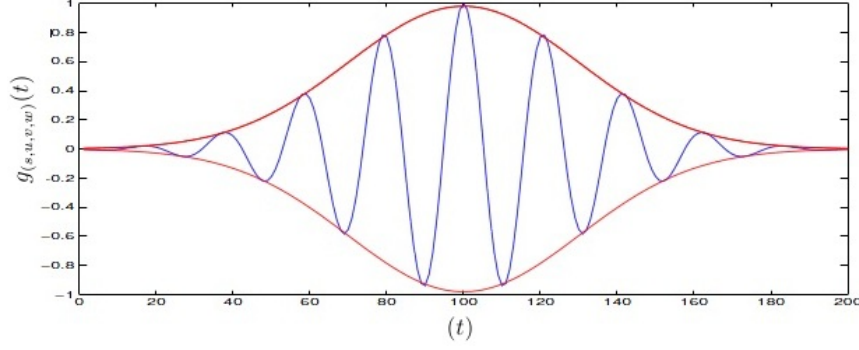


Figure 4.1: Gabor Function (Blue line represents the real Gabor function and the red line is the Gaussian envelope)

An efficient way of creating the atoms is to use complex arithmetic for the phase parameter w , which reduces the order of complexity of the overall dictionary. This means that since the atoms are used to match a particular signal, the phase parameter of the atom can be calculated to best suit the signal based on the optimization criteria. This will reduce the computations when building the dictionary because the phase parameter is specifically calculated after all the other parameters such as (s, u, v) have been set. Here the input signals and the dictionary are finite and do not exceed the interval. By looking at the initialized parameters used in the implementation of the dictionary, the phase of the atom is calculated based on the optimization criteria. However, using the manipulation of phase is just one of the ways that the speed is increased. The Gabor dictionary is a redundant collection of atoms where they are selected as bases to represent the given signal. This type of dictionary is explained and employed particularly within the greedy Matching Pursuit algorithm described in section 4.2.

4.2 Matching Pursuit Algorithm

The procedure where an adaptive method can approximate the signal as a superposition of basic waveforms which are chosen from a defined set in a manner that they best suit the signal is called the Matching pursuit algorithm. This algorithm estimates the sparse representation of a signal denoted by f , not through the global optimization of a criterion but using an iterative greedy algorithm. The goal becomes clear by looking at Eq. 4.2 when the dictionary Φ is redundant. The joint sparse representation of C_x depends on the chosen algorithm that can satisfy the condition $C_x \approx AC_S$ that is a significant criterion for sparse separation. The algorithm finds a linear expansion of a signal f by selecting vectors from Φ in a way that the signal can be represented in its most sparse form. Using the Matching Pursuit algorithm, the successive approximation of the signal is done by producing an orthogonal projection on the atom given by $\varphi_{\gamma_0} \in \Phi$. Therefore, the signal f can be explained as:

$$f = \langle f, \varphi_{\gamma_0} \rangle \varphi_{\gamma_0} + R_f \quad (4.4)$$

where R_f is the residue vector. R_f is the original signal without the approximated signal in the direction of the chosen atom. The chosen atom φ_{γ_0} is orthogonal to the residue R_f and satisfies:

$$\|f\|^2 = |\langle f, \varphi_{\gamma_0} \rangle|^2 + \|R_f\|^2 \quad (4.5)$$

The objective of this algorithm is to minimize the residue signal in each iteration. Therefore, in order to minimize the residue R_f the projection term explained as $|\langle f, \varphi_{\gamma_0} \rangle|$ must be maximized over the range of $\varphi_{\gamma_0} \in \Phi$. Such optimization becomes the goal of finding an optimal atom such that:

$$|\langle f, \varphi_{\gamma_0} \rangle| = \max_{\gamma \in \Gamma_\alpha} |\langle f, \varphi_\gamma \rangle| \geq \alpha \sup_{\gamma \in \Gamma} |\langle f, \varphi_\gamma \rangle| \quad (4.6)$$

where $\Gamma_\alpha \subseteq \Gamma$ and α is a factor having a range of $0 < \alpha < 1$. The continuous parameter Γ determines the index of the dictionary containing a discrete grid of the form of Γ_α which can be used to create a sub-dictionary. Next, each step of the algorithm consists of finding the best atom within the dictionary where when added to the set of already found atoms,

the residue approximation would tend toward zero and the error would decrease to a large extent. The iterative algorithm is initialized by $R_f = f$, and after m iterations R_f^m is obtained. Therefore, the atom element of φ_{γ_m} is chosen such that:

$$R_f^m = \langle R_f^m, \varphi_{\gamma_m} \rangle \varphi_{\gamma_m} + R_f^{m+1} \quad (4.7)$$

where R_f^{m+1} is the residue of the next iteration $m + 1$.

After repeating the decomposition M times the signal can be written as:

$$f = \sum_{m=1}^{M-1} \langle R_f^m, \varphi_{\gamma_m} \rangle \varphi_{\gamma_m} + R_f^M \quad (4.8)$$

One of the theorems that can also be proven by the notations explained previously is the work of Mallat and Zhang [19] that analyzes the representation of the signal in the form of the atoms and its projections as an infinite sum given by:

$$f = \sum_{k=0}^{\infty} \langle R_f^k, \varphi_{\gamma_k} \rangle \varphi_{\gamma_k} \quad (4.9)$$

This also explains the nature of the dictionary as being redundant. The signal can have an infinite number of atoms. However, in the sparse representation only a few are significant and most of them tend towards zero.

4.2.1 Steps of Matching Pursuit Algorithm

A simple Matching Pursuit algorithm is stated step by step to allow for a better understanding for practical analysis and data manipulations.

Step 1) Initialization: Initialize the residue as $R_f^m = f$, the iteration number $m = 1$ and the signal coefficients $C_f^m = 0$

Step 2) Compute the inner product between the signal residue R_f^m and each atom φ_{γ_m}

Step 3) Selection of the atom such that $k_m = \arg \max_m |\langle f, \varphi_{\gamma_m} \rangle|$

Step 4) Update the residue by $R_f^{m+1} = R_f^m - \langle R_f^m, \varphi_{\gamma_m} \rangle \varphi_{\gamma_m}$

Step 5) Update the coefficients by $C_f^m = \langle R_f^m, \varphi_{\gamma_m} \rangle$

Step 6) If the stopping criterion has not reached, $m = m + 1$ and go back to step (2)

4.2.2 Implementation of Gabor Atom

In order to build the dictionary, the system should be analyzed by continuous Gabor functions windowed by trigonometric functions that have infinite decreasing exponential tails. The notations here are compatible with the dictionary used in [19]. The window function is a Gaussian function described as:

$$g(t) = 2^{\frac{1}{4}} e^{-\pi t^2} \quad (4.10)$$

Therefore, the Gabor function φ_γ for $\gamma = (s, u, v)$ is given as:

$$\varphi_\gamma = \frac{1}{\sqrt{s}} g\left(\frac{t-u}{s}\right) e^{ivt} \quad (4.11)$$

The parameters are the normalization factor of $\frac{1}{\sqrt{s}}$, where s is the scaling factor, u belongs to the translation and v is the frequency modulation. By looking at the function closely it is observed that φ_γ is centered at u and most of the energy is concentrated in this area of amplitude s . In addition, since the signal f is real, the corresponding dictionary must be designed using time-frequency functions. Therefore, $\gamma = (s, u, v)$ will take any values for $s > 0$ and the phase interval is $w \in [0, 2\pi]$. The function is then rewritten as:

$$\varphi_{(\gamma, w)} = \frac{K_{(\gamma, w)}}{2} \left(e^{(iw)} g_{(s, u, v)}(t) + e^{(-iw)} g_{(s, u, -v)}(t) \right) = \frac{K_{(\gamma, w)}}{\sqrt{s}} g\left(\frac{t-u}{s}\right) \cos(vt + w) \quad (4.12)$$

where the normalization factor of $K_{(\gamma, w)}$ is chosen as $\|\varphi_{(\gamma, w)}\|_2 = 1$. Further, it can be viewed that the hidden phase in the complex coefficient is now in the parameter of the real Gabor function. Therefore, the dictionary is a real time-frequency vector with parameters of (γ, w) . Let's denote the dictionary parameters as $\beta = (\gamma, w)$.

The Matching Pursuit algorithm can be rewritten in terms of the real dictionary:

$$f(t) = \sum_{m=0}^{\infty} \langle R_f^m, \varphi_{\beta_m} \rangle \varphi_{\beta_m}(t) \quad (4.13)$$

The objective here is to find the best indices β_n which maximizes $|\langle R_f^m, \varphi_{\beta_m} \rangle|$.

4.2.3 Optimal Atom Definition

Given that an input signal has been sampled with N equally intervals at times $t = 0, \dots, N-1$, then:

$$f = [f(0), f(1), \dots, f(N-1)] \quad (4.14)$$

The same sampling procedure has taken place on the dictionary atoms and the discrete partitions are chosen. The normalization term from the Eq. 4.3 is dropped and the norm of the total function is calculated such as stated below.

$$\varphi_{(s,u,v,w)}(t) = \frac{g\left(\frac{t-u}{s}\right)\cos(vt+w)}{\|g\left(\frac{t-u}{s}\right)\cos(vt+w)\|} \quad (4.15)$$

By using simple cosine trigonometric identities the cosine term can be broken down into:

$$\cos(vt+w) = \cos(vt)\cos(w) - \sin(vt)\sin(w) \quad (4.16)$$

Further, having the parameters set by the notation $\gamma = (s, u, v)$ and the Gaussian expression as g given in Eq. 4.10, it can be re-arranged to obtain the functions such as:

$$D_\gamma(t) = g\left(\frac{t-u}{s}\right)\cos(vt) \quad (4.17)$$

and,

$$O_\gamma(t) = g\left(\frac{t-u}{s}\right)\sin(vt) \quad (4.18)$$

Therefore, by using these functions and the previous notation the dictionary atoms can be written as:

$$\varphi_{(\gamma,w)} = \frac{D_\gamma\cos(w) - O_\gamma\sin(w)}{\|D_\gamma\cos(w) - O_\gamma\sin(w)\|} \quad (4.19)$$

and consequently the inner product of the input signal in the Matching Pursuit algorithm can be calculated by taking the dot product of the above dictionary atom and the signal f such as:

$$\langle f, \varphi_{(\gamma,w)} \rangle = \frac{\langle f, D_\gamma \rangle \cos(w) - \langle f, O_\gamma \rangle \sin(w)}{\|D_\gamma \cos(w) - O_\gamma \sin(w)\|} \quad (4.20)$$

The goal of the Matching Pursuit algorithm is to find the atom that in each iteration best matches the properties of the signal. This is viewed by an optimization criterion explained

by $\max_{\gamma, w} |\langle R_f, \varphi_{(\gamma, w)} \rangle|$. The fundamentals of the dictionary implementation here has taken the size into consideration where the phase w is uniquely determined, making the size only dependent on the first three parameters. Such dictionary is computationally more efficient and faster. In section 4.2.4 there are detailed derivations that explain the extraction of an optimal atom from a redundant dictionary given a real signal.

4.2.4 Single Channel Atom Extraction

Given an input signal $f = [f(0), f(1), \dots, f(N-1)]$ and $\gamma = (s, u, v)$, let's denote the following terms:

$$a = \langle f, D \rangle, b = \langle f, O \rangle, a_1 = a\|O\|^2 - b\langle D, O \rangle, b_1 = b\|D\|^2 - a\langle D, O \rangle \quad (4.21)$$

By closely examining the Eq. 4.20, it can be rewritten using the defined parameters. Let's denote $p(x) = \frac{a-bx}{\|D-Ox\|}$ where $x = \tan(w)$.

The optimization problem with the only dependent variable x becomes the maximization of $p(x)^2$. By taking the derivative of $p(x)^2$ with respect to x and using the chain rule, the derivative can be calculated simply as:

$$\frac{\partial}{\partial x} \left(\frac{(a-bx)^2}{(D-Ox)^2} \right) = \frac{2(a-bx)(aO-bD)}{(D-Ox)^3} \quad (4.22)$$

By substitution of Eq. 4.21 into Eq. 4.22 will lead to a simpler form in terms of the defined parameters such as:

$$\frac{\partial}{\partial x} \left(\frac{(a-bx)^2}{(D-Ox)^2} \right) = \frac{-2(a-bx)}{(D-Ox)^4} (a_1x + b_1) \quad (4.23)$$

By equating Eq. 4.23 to zero, one can find the value of x maximizing $p(x)^2$ given by:

$$-2(a-bx)(a_1x + b_1) = 0 \quad (4.24)$$

$$(-ba_1)x^2 + (aa_1 - bb_1)x + (ab_1) = 0 \quad (4.25)$$

Using the quadratic formula the roots are found and stated as:

$$x = \frac{-b_1}{a_1}, x = \frac{a}{b} \quad (4.26)$$

However the second root belongs to a minimum and is discarded and the maximum is achieved at $x = -\frac{b_1}{a_1}$. The condition on a_1 when it is zero becomes the maximization of x which approaches an asymptotically infinity meaning that $w_0 = \frac{\pi}{2}$. Also when a_1 is not at a value of zero, then the maximum happens when $x = -\frac{b_1}{a_1}$.

By looking at Eq. 4.20, one can obtain the different cases of optimal atom based on the value of w_0 . These cases are explained here covering the interval of $w \in [0, 2\pi]$. Let's Denote $\langle f, \varphi_{(\gamma, w)} \rangle$ as $\langle f, \varphi \rangle$ for simplicity for the following cases.

Case (1) Let $v = 0$, maximum is achieved when:

$$\langle f, \varphi \rangle = \frac{a}{||D||} \quad (4.27)$$

and $w_0 = 0$

Case (2) Let $v \neq 0$ and $a_1 = 0$, maximum is achieved when:

$$\langle f, \varphi \rangle = -\frac{b}{||O||} \quad (4.28)$$

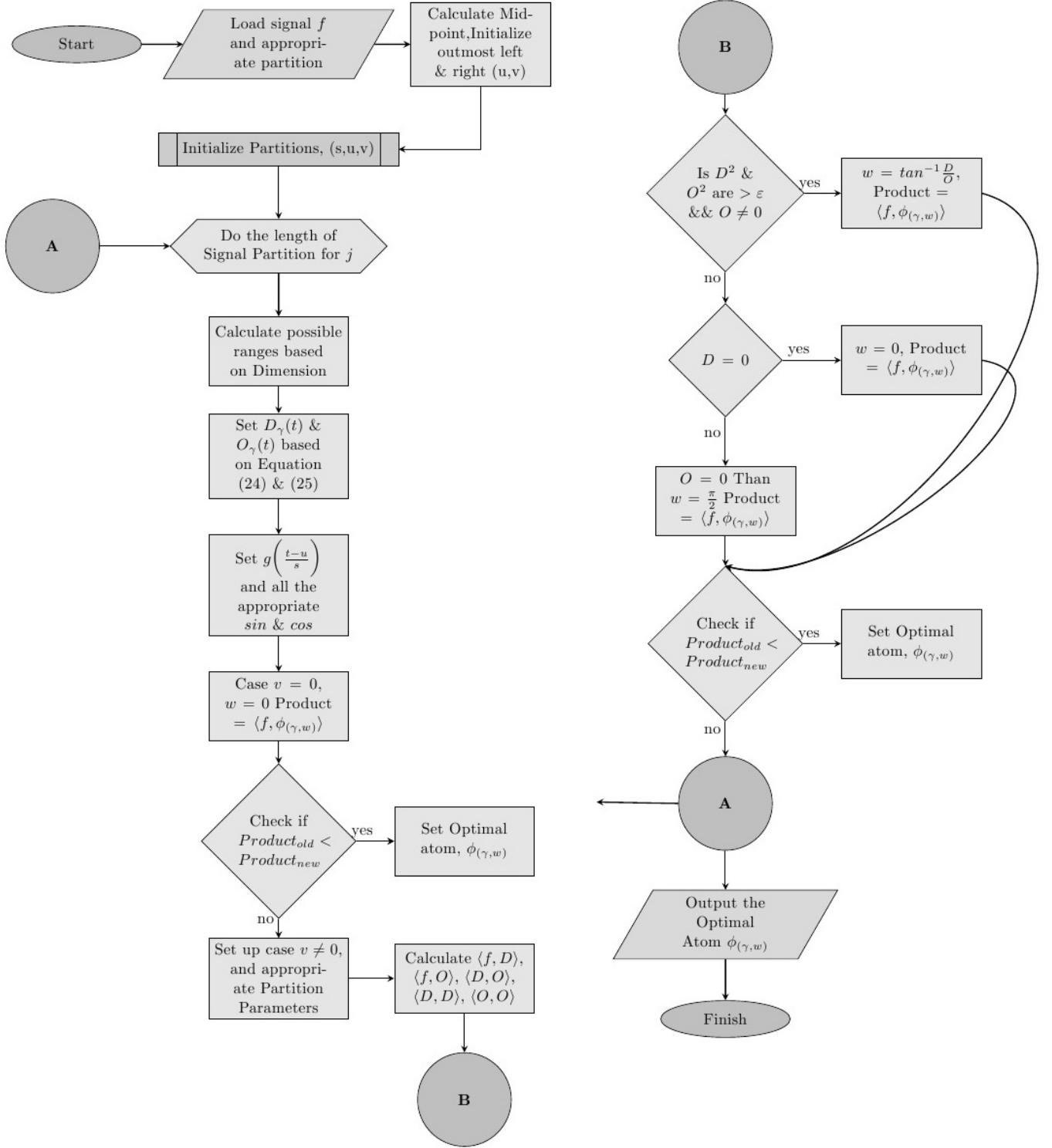
and $w_0 = \frac{\pi}{2}$

Case (3) Let $v \neq 0$ and $a_1 \neq 0$, maximum is achieved when:

$$\langle f, \varphi \rangle = \frac{aa_1 + bb_1}{||Da_1 + Ob_1||} \quad (4.29)$$

and $\tan(w_0) = -\frac{b_1}{a_1} = \frac{D}{O}$

The dictionary implementation is presented in Flowchart 4.1 where it shows how the atoms are selected based on the Matching Pursuit algorithm.



Flowchart 4.1: Real Gabor Dictionary

The signal's nature will ultimately define the properties of the dictionary. For example an audio signal can be decomposed into its elementary signals using the Gabor dictionary defined and the Matching pursuit algorithm stated. To show a simple test case for Matching Pursuit algorithm, a signal of $44.1KHz$ of 500 samples was tested. The signal and its approximated version are both shown in Fig. 4.2.

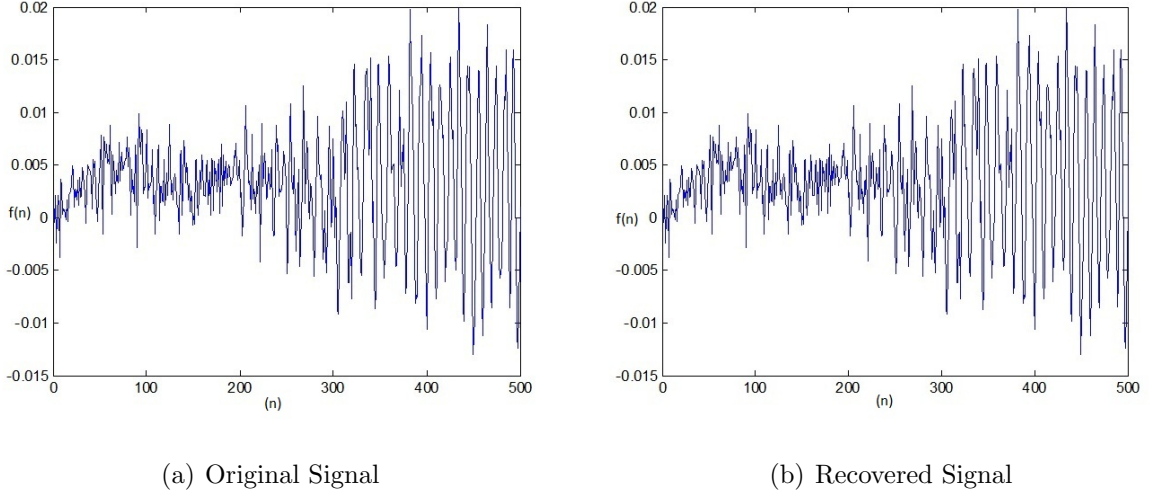


Figure 4.2: Matching Pursuit algorithm Using Real Gabor Dictionary: On the left (a) there exist an original signal and on the right (b) the recovered approximation of the signal produced by the Matching Pursuit algorithm.

The steps of recovering the source signal is given in section 4.2.1. The signal was recovered by Eq. 4.13 where the iteration took about 397 appropriate atoms chosen by the criterion stated in Eq. 4.6. Further, the coefficients were calculated by Eq. 4.20, where the squared sum of these coefficients is equal to the original signal energy according to Eq. 4.5. The signal energy is 0.212 and the coefficient squared sum is 0.2119. Ultimately, the coefficients resemble the density plot of a super Gaussian function. This is shown in Fig. 4.3 where the coefficients on the positive half, contain a sharp peak at the origin and heavy tail as the number of the coefficients increase. This emphasizes the sparse representation of the signal, where there are only a few significant coefficients that exist and the rest are near zero.

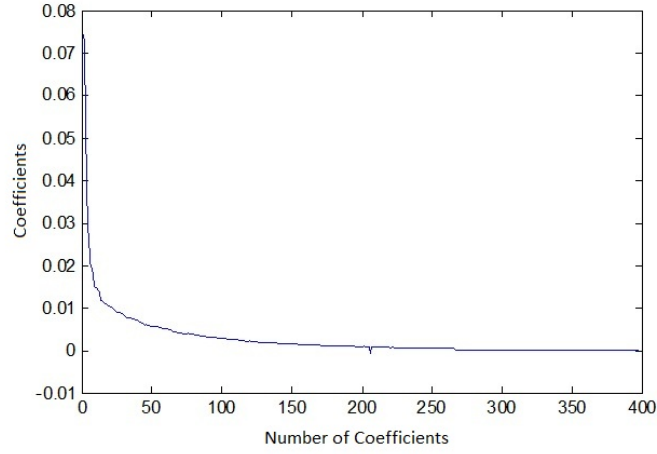


Figure 4.3: Projection Coefficients: 397 Coefficients used to recover the signal in Fig. 4.2(b) show a density function of a super Gaussian distribution

Since the dictionary is redundant then the signal can be represented by an infinite number of possible coefficient sets, also stated by Eq. 4.13. Therefore, there needs to be a desired threshold or a stopping criterion. The stopping criterion of the procedure was set to an error of 10^{-10} or a max iteration = 2000. Further, the signal decomposition is according to Eq. 4.8 where there is a residue function and the error calculated by the square sum of the residue signal shown in Fig. 4.4 with 397 iterations with the desired error satisfied.

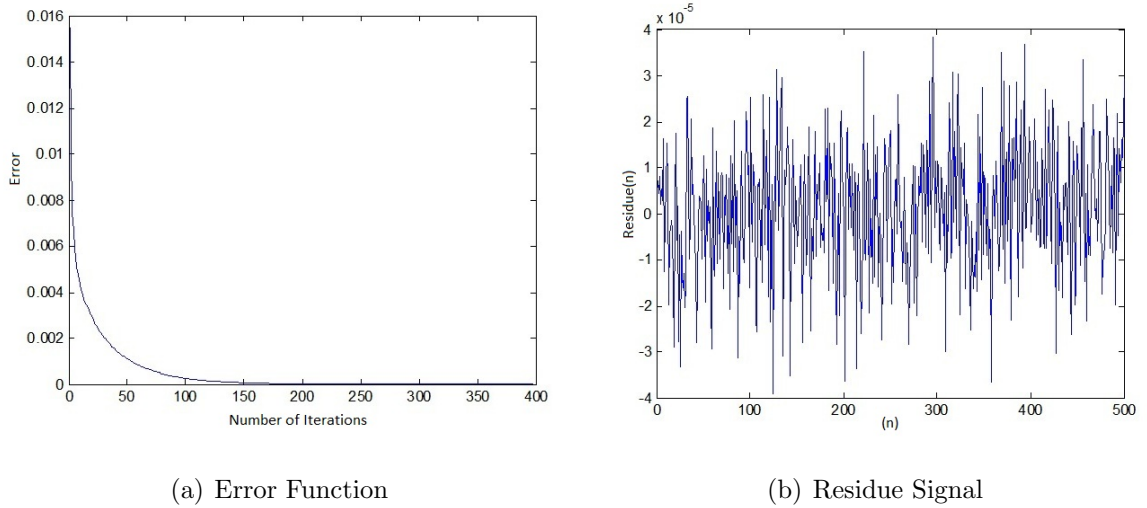


Figure 4.4: Error & Residue: Error in the left decreases as more of the signal is decomposed and after 397 iterations the residue has become nearly zero shown on the right

4.3 Multi-channel Matching Pursuit algorithm and Cumulative Atom Selection

The second stage of the problem which consists of separation of the sources becomes more interesting where an adaptive decomposition tool is used to represent the signals in the sparsest form. The only knowledge of the sources given is the mixture recorded by the sensors. Therefore, the initial phase is to be able to represent these mixture observations in the sparse form such that the mixture \mathbf{X} is decomposed by $\mathbf{X} \approx \mathbf{C}_x \Phi$.

$$C_x = \begin{bmatrix} C_{x_1} \\ \cdot \\ \cdot \\ \cdot \\ C_{x_M} \end{bmatrix} = [C_x(1) \dots C_x(K)] \quad (4.30)$$

In addition, the coefficient matrix is of size $M \times K$ where there are M channels each of K coefficients. By looking at Eq. 4.2 the separation problem becomes not an optimization problem of a global sparse model however an iterative greedy algorithm to estimate the joint sparse model of the mixtures $X = AC_s \Phi$. At this stage C_s must be solved given that $C_x \approx AC_s$. This optimization problem aims at solving the system where C_s is sparse and of size $N \times K$, where each row is a sparse vector of K coefficients of one of the N sources.

$$C_s = \begin{bmatrix} C_{s_1} \\ \cdot \\ \cdot \\ \cdot \\ C_{s_N} \end{bmatrix} = [C_s(1) \dots C_s(K)] \quad (4.31)$$

Further, if the sources are sparse and only one active at one time satisfying the sparsity assumptions, then they possess the disjoint support property and the mixture should enable the separation. However, where the dictionary Φ is redundant the representation of

C_x depends on the algorithm that is chosen to decompose the signal. Therefore, selecting the appropriate algorithm which is able to provide an approximation that best satisfies $C_x \approx AC_s$ is significant to source separation where the representation C_s is known to have disjoint support.

The Gabor dictionary is used in the algorithm for a single-channel processing. However, in the case of multi-channel the convergence is different. The multi-channel Matching Pursuit algorithm is used to recover N sources from M mixtures. The observation mixtures are the only given input for the algorithm and one must find a logical way of using the sparse coefficients in recovering the sources. The procedure can be analyzed as updating the residues by orthogonal projection of the given signal on the linear span of all the selected atoms. The dictionary still remains the same containing Gabor atoms explained by Eq. 4.15 and the atom functions are simply given as:

$$\varphi_{(\gamma,w)} = \frac{D_\gamma \cos(w) - O_\gamma \sin(w)}{||D_\gamma \cos(w) - O_\gamma \sin(w)||} \quad (4.32)$$

However the multi-channel case differs from a single channel because there exists more input signals and a process must be implemented where all of the signals can be decomposed at once. Therefore, the importance of multi-channel Matching Pursuit algorithm will be in the atom selection stage. The goal of the implementation is to find the best suited atom for all of the signals $f_i = [f_i(0), f_i(1), \dots, f_i(N-1)]$, where $i = 1, \dots, M$. Meaning that the atom which is selected must satisfy the following condition:

$$K_m = \arg \max_{\gamma,w} \sum_{i=1}^M |\langle R_{f_i}, \varphi_{(\gamma,w)} \rangle|^2 \quad (4.33)$$

where each residue signals R_{f_i} are initially set to the signals f_i correspondingly and the parameters D_γ and O_γ are as explained previously by Eq. 4.17 and Eq. 4.18 respectively. Therefore, the inner product of the input signals in the multi-channel Matching Pursuit algorithm can be calculated by taking the dot product of the dictionary defined and each signal in the form of:

$$\langle f_i, \varphi_{(\gamma,w)} \rangle = \frac{\langle f_i, D_\gamma \rangle \cos(w) - \langle f_i, O_\gamma \rangle \sin(w)}{||D_\gamma \cos(w) - O_\gamma \sin(w)||} \quad (4.34)$$

where $i = 1, \dots, M$.

Therefore, the selection of the atom in Eq. 4.34 can be used where the best atom in the dictionary is chosen for all of the input signals based on the condition in Eq. 4.33 such as:

$$K_m = \underset{\gamma, w}{argmax} \sum_{i=1}^M \left| \frac{\langle f_i, D_\gamma \rangle \cos(w) - \langle f_i, O_\gamma \rangle \sin(w)}{\|D_\gamma \cos(w) - O_\gamma \sin(w)\|} \right|^2 \quad (4.35)$$

Eq. 4.32 can be rewritten using the following defined terms:

$$a_{fi} = \langle f_i, D \rangle, b_{fi} = \langle f_i, O \rangle \quad (4.36)$$

where $i = 1, \dots, M$, M is the number of observation signals.

The optimal atom can be analyzed based on derivation in Eq. 4.34 by denoting $x = \tan(w)$ which gives:

$$p_i(x) = \frac{a_{fi} - b_{fi}x}{\|D - Ox\|} \quad (4.37)$$

The optimization problem with the only dependent variable x becomes the maximization of $p_i(x)^2$. When examining the derivation in Eq. 4.35, it can be observed that the maximization has taken over all the inner products and the sum is evaluated. Therefore, the optimization problem stretches over the range of all signals in each iteration. This can be viewed as taking the derivative of the following function:

$$\frac{\partial}{\partial x} \left(\frac{(a_{f1} - b_{f1}x)^2}{(\|D - Ox\|)^2} + \dots + \frac{(a_{fM} - b_{fM}x)^2}{(\|D - Ox\|)^2} \right) \quad (4.38)$$

$$= \left(\frac{-2(a_{f1} - b_{f1}x)}{(D - Ox)^3} (a_{f1}O - b_{f1}D) + \dots + \frac{-2(a_{fM} - b_{fM}x)}{(D - Ox)^3} (a_{fM}O + b_{fM}D) \right) \quad (4.39)$$

The solution to the optimization problem in Eq. 4.39 is achieved by maximizing the overall function which consists of individual residues of a total number of M . The goal of this approach is to consider the convergence time of the algorithm since one atom is chosen for all the signals in each iteration. This is looked at more in depth in this section and the complications with such algorithm are also stated. The maximum points can be found by

finding the roots of the Eq. 4.39. A hidden root is simply found by multiplying the equation by the term in the denominator and setting it to zero. This will follow the same derivation as previously explained which has an advantage of a simpler final answer. The multi-case can be written as:

$$(a_{f1} - b_{f1}x)(a_{f1}O - b_{f1}D)(D - Ox) + (a_{f2} - b_{f2}x)(a_{f2}O - b_{f2}D)(D - Ox) + \dots + (a_{fM} - b_{fM}x)(a_{fM}O - b_{fM}D)(D - Ox) = 0 \quad (4.40)$$

Eq. 4.40 is essentially a simple root finding problem and can be expanded in terms of its coefficients. However, when the number of residue signals increases the complexity of the coefficients also increase. The number of observation sensors provided for this problem and the complexity of finding the root of Eq. 4.39 has a proportional relationship. As the number of given sensors increase the algorithm must find a root to a polynomial with larger coefficients which will increase the number of constraints. However, this can be simply done by appropriate programming and techniques.

The root of the function stated in Eq. 4.40 is calculated by simple manipulation and keeping the terms containing the respected variable x in mind. In each of the terms there is an expression of $(D - Ox)$ and $(a_{fM} - b_{fM}x)$ which respectively belong to a maximum and a minimum value of Eq. 4.37. The maximum value is given at:

$$x = \frac{D}{O} \quad (4.41)$$

It can be simply observed that when the limit of the parameter O approaches zero, the value of x goes to infinity. In addition, the value of w which is the tangent, approaches an asymptotically maximum of $w = \frac{\pi}{2}$. However, when the parameter $O \neq 0$, the value of x is given by Eq. 4.41. Also when the parameter of the basis atom denoted by v in Eq. 4.11 is at value of zero, the *cosine* parameter will greatly impact the overall function and the maximum will be achieved when $w = 0$.

Due to this optimization criterion, one atom can act greatly on a certain residue function and may not significantly impact the other residues. However, since the Matching Pursuit

algorithm has a greedy nature, it will iteratively choose as many atoms as needed to decompose each of the residue functions until they reach a small value of ϵ . The following cases are stated for the $\gamma = (s, u, v, w)$ selection, particularly the phase w . The summation of $\langle f_i, \varphi_{(\gamma, w)} \rangle$ are calculated which is the optimization criterion stated in Eq. 4.33:

Case (1) Let $v = 0$, maximum is achieved when:

$$\sum_{i=1}^M \langle f_i, \varphi \rangle = \sum_{i=1}^M \frac{a_{fi}}{\|D\|} \quad (4.42)$$

and $w_0 = 0$

Case (2) Let $v \neq 0$ and $a_1 = 0$, maximum is achieved when:

$$\sum_{i=1}^M \langle f_i, \varphi \rangle = - \sum_{i=1}^M \frac{b_{fi}}{\|O\|} \quad (4.43)$$

and $w_0 = \frac{\pi}{2}$

Case (3) Let $v \neq 0$ and $a_1 \neq 0$, maximum is achieved when:

$$\sum_{i=1}^M \langle f_i, \varphi \rangle = \sum_{i=1}^M \frac{a_{fi}a_{1fi} + b_{fi}b_{1fi}}{\|Da_{1fi} + Ob_{1fi}\|} = \sum_{i=1}^M \frac{a_{fi} - b_{fi}\tan(w)}{\|D - O\tan(w)\|} \quad (4.44)$$

and $\tan(w_0) = -\frac{b_{1fi}}{a_{1fi}} = \frac{D}{O}$

The outlined atom selection for the multi-channel case relies on an optimization criterion that selects the optimal atom from Φ that best suits all of the signals. This approach has its drawbacks where each signal is not utilized appropriately and full advantage is not taken. The atom selected will only be best correlated with M number of signals and the final residue will converge to a small value of ϵ in a longer time and with more coefficients. A better approach which will take full advantage of each signal presented is explained in section 4.3.2.

Further, the appropriate projection coefficients are calculated:

$$C_x^k = \langle R_{fi}^k, \varphi_{(\gamma, w)} \rangle \quad (4.45)$$

where k is the iteration number and i is the residue signal of $i = 1, \dots, M$. In each iteration there will be a set of coefficients produced of size $M \times 1$. This will produce a matrix of size $M \times K$ of coefficients after K iterations, which is stated in Eq. 4.30.

Further, after appropriate atom selection the residues are updated in such manner:

$$R_{fi}^{k+1} = R_{fi}^k - \langle R_{fi}^k, \varphi_{(\gamma,w)} \rangle \varphi_{(\gamma,w)} \quad (4.46)$$

The main parameter is the choice of stopping criterion. The two possible approaches is by setting a predefined number of desired iterations or setting the target approximation error $\|R_{fi}^k\|$ beforehand.

4.3.1 Steps of Multi-Channel Matching Pursuit algorithm using Cumulative Atoms

The steps of a multi-channel Matching pursuit algorithm which selects one atom for all of the signals is explained thoroughly in this section. Along with the flowchart, it is easier to understand the implementation of such an algorithm.

Step 1) Initialization: Initialize the residues as $R_{fi}^k = X_i$, the iteration number $k = 1$ and the signal coefficients $C_x^k = 0$

Step 2) Compute the inner product between the signal residues R_{fi}^k and atom $\varphi_{(\gamma,w)}$

Step 3) Selection of the atom such that $K_m = \arg \max_{\gamma,w} \sum_{i=1}^M |\langle R_{fi}^k, \varphi_{(\gamma,w)} \rangle|^2$

Step 4) Update the coefficients by $C_x^k = \langle R_{fi}^k, \varphi_{(\gamma,w)} \rangle$

Step 5) Update the residues by $R_{fi}^{k+1} = R_{fi}^k - \langle R_{fi}^k, \varphi_{(\gamma,w)} \rangle \varphi_{(\gamma,w)}$

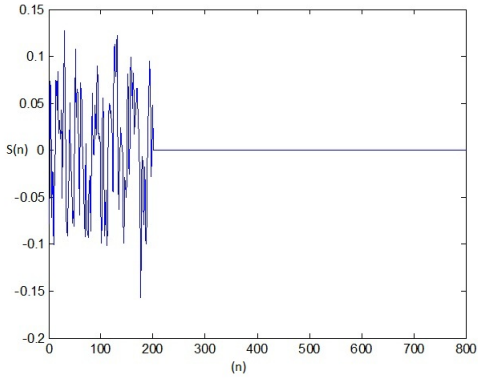
Step 6) Use Binary Masking Described in section 4.4 to recover the sources

Step 7) If the stopping criterion has not reached, $k = k + 1$ and go back to step (2)

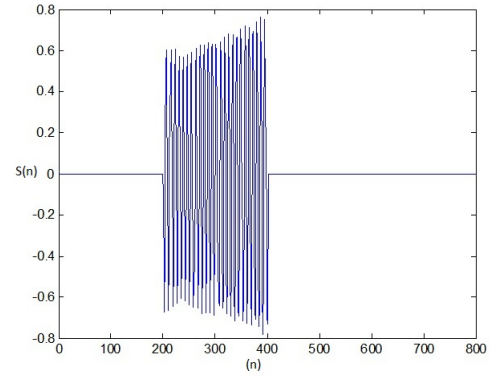
In each iteration the coefficients of the observed data C_x is approximated and the residues are updated and the process is repeated until the desired criterion is met. The sparse

representation algorithm here can estimate the source coefficients in each iteration by $C_x \approx AC_s$ which will be looked at in the section 4.4.

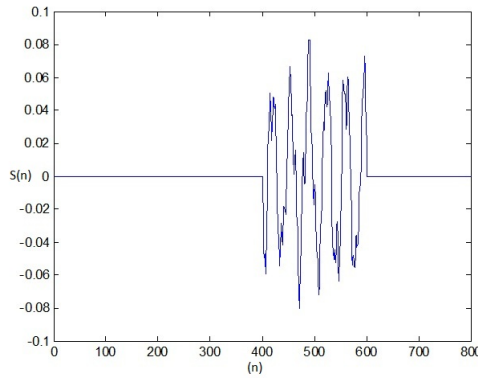
Here a simple example is given in order to show how the multi-channel Matching Pursuit algorithm recovers the sources. The performance of different test cases are explained and shown in section 5 denoted by "Simulation and Results". There are four signals of $16KHz$ that were chosen with high sparsity and mixed with a random 3×4 mixing matrix which produced 3 mixtures. These mixtures were passed onto the multi-channel Matching Pursuit algorithm and the sparse source signals were recovered. The algorithm converged with 5521 iteration and a chosen error of 10^{-9} . In Fig. 4.5 the original sources before the mixing stage involved with this algorithm are listed. The goal of the Matching pursuit algorithm is to recover these signals with only the given mixtures in Fig. 4.6.



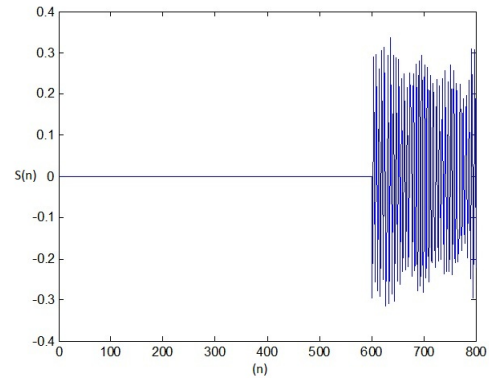
(a) Source Signal 1



(b) Source Signal 2



(c) Source Signal 3



(d) Source Signal 4

Figure 4.5: Source Signals: 4 Audio signals of 800 sample points with full temporal support

Further signals in Fig. 4.5 were mixed with the random mixing matrix producing mixtures shown in Fig. 4.6.

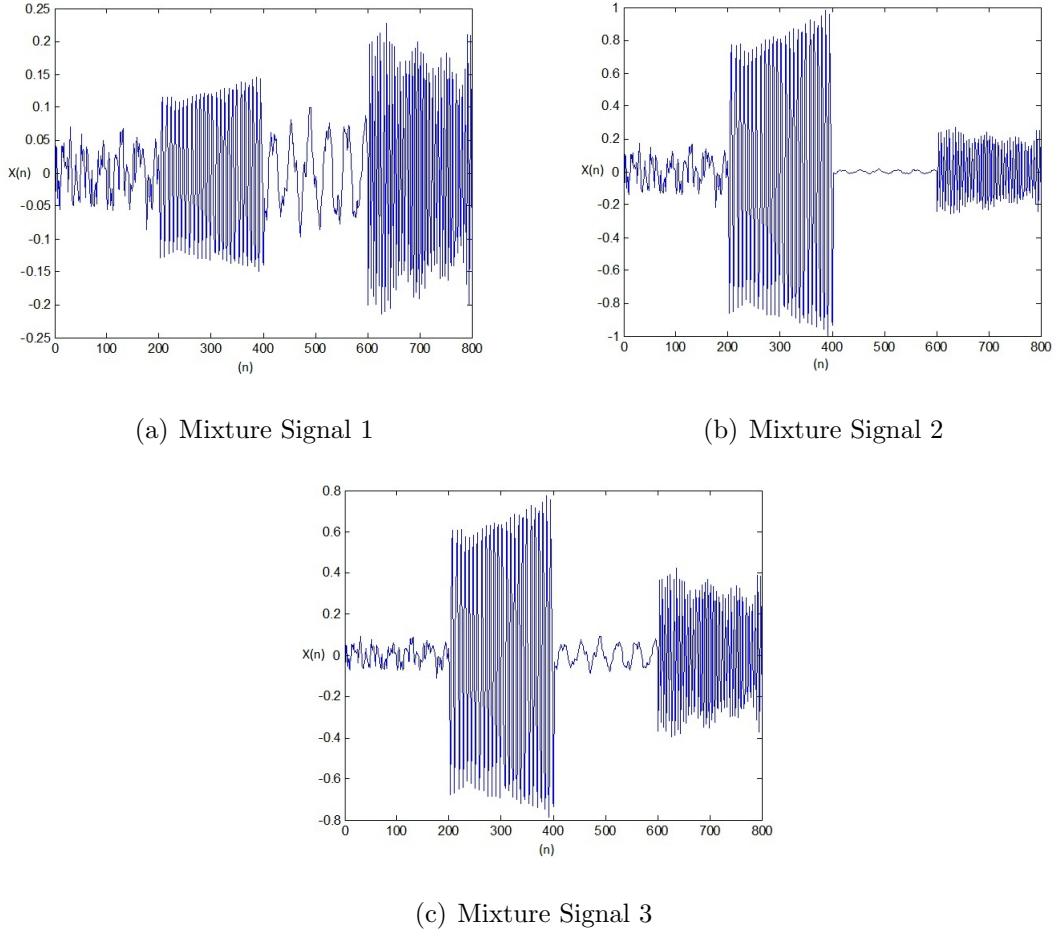
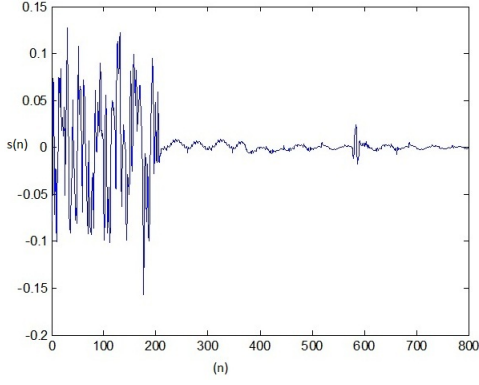
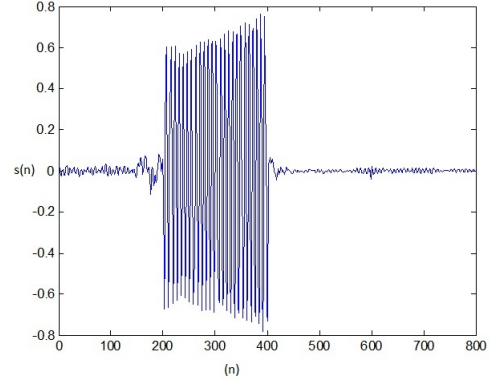


Figure 4.6: Mixture Signals: 3 Mixtures are produced by a random mixing matrix

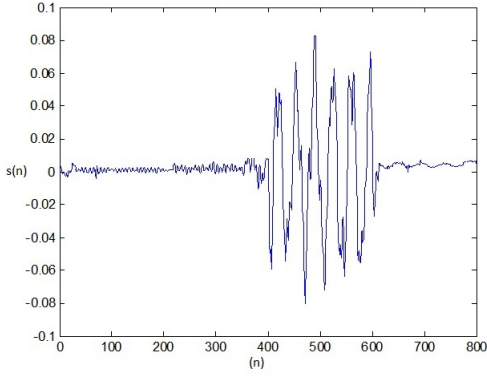
Next, the mixtures in Fig. 4.6 are the only inputs to the multi-channel algorithm. The algorithm chooses an atom that best matches the 3 mixtures based on the optimization criteria in Eq. 4.33. The optimal atom is used to update the projection coefficients along with the residue signals. This procedure is repeated until the set desired error and maximum iteration criterion is met. The sparse sources recovered are shown in Fig. 4.7. It can be viewed that the algorithm successfully recovered the sources, however there exist noise in the portion where the signals are nearly zero. This noise can be removed by a simple de-noising algorithm after the estimated sources are obtained.



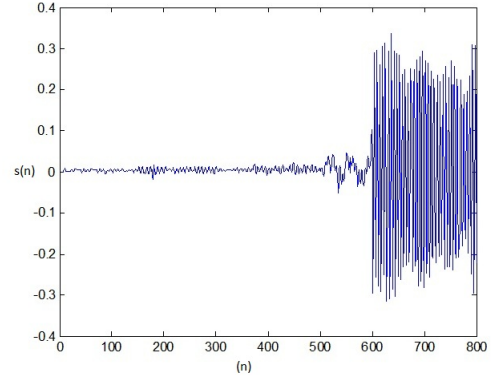
(a) Recovered Signal 1



(b) Recovered Signal 2



(c) Recovered Signal 3



(d) Recovered Signal 4

Figure 4.7: 4 Recovered Source Signals by multi-channel Matching Pursuit algorithm using Cumulative Atom Selection

4.3.2 The Proposed Multi-channel Matching Pursuit algorithm & Maximum Residue Atom Selection

A different approach to select an atom in each iteration is by analyzing each residue signal individually. If there are M residue signals then there will be M atoms calculated along with their projection coefficients on each signal. Next, the atom which belongs to a maximum residue projection will be chosen as the optimal atom. This takes advantage of an optimization criterion explained as:

$$\max_{\gamma, w} |\langle R_{fi}^k, \varphi_{(\gamma, w, i)} \rangle|^2 \quad (4.47)$$

where in each iteration there are $i = 1, \dots, M$ number of atoms selected for all residue signals. Next the projection criterion explained by Eq. 4.47 will ultimately choose the appropriate atom. This becomes significant due to the fact that the atom will greatly impact the signal residue in each iteration and the algorithm will converge with less iteration. The disadvantage of this approach is that instead of storing one atom in each iteration, there are M number of atoms that need to be calculated. However, the advantage is that in each iteration the atom will not only be correlated with all of the signals but more importantly it will be maximally correlated with at least one residue, which was not the case in the cumulative selection approach. The convergence time also depends on the trade-off between memory and speed. In the cumulative case a better result is gained with less memory needed and the speed becomes an important factor to look at and in the individual case more memory will be needed. Therefore, one can take advantage of these fundamental algorithms designed to produce a fast and efficient tool based on the speed of the processing unit available, memory and also the technique of appropriate programming.

In addition, after the selection of an atom i for each of the residue signals R_{fi}^n , there are M number of atoms obtained and maximum projection determines the optimal atom. Next, the appropriate projection coefficients are calculated as:

$$C_x^k = \langle R_{fi}^k, \varphi_{(\gamma,w)} \rangle \quad (4.48)$$

where k is the iteration number. In each iteration there will be a set of coefficients produced of size $M \times 1$. The coefficients are produced by a different atom and will form a matrix of size $M \times K$ after k iterations.

In addition, after appropriate atom selection the residues are updated such as:

$$R_{fi}^{k+1} = R_{fi}^k - \langle R_{fi}^k, \varphi_{(\gamma,w)} \rangle \varphi_{(\gamma,w)} \quad (4.49)$$

Further, the stopping criterion is set by the target approximation error $\|R_{fi}^n\|$ beforehand.

4.3.3 Steps of Proposed Multi-Channel Matching Pursuit algorithm using Maximum Residue Atom

This section allows for a better understanding of the overall algorithm step by step which makes it clearer to picture or even to implement.

Step 1) Initialization: Initialize the residues as $R_{fi}^k = X_i$, the iteration number $k = 1$ and the signal coefficients $C_x^k = 0$

Step 2) Select the best atom for each residue R_{fi}^k as $\varphi_{(\gamma,w,i)}$ and calculate their corresponding projection as $\langle R_{fi}^k, \varphi_{(\gamma,w,i)} \rangle$

Step 3) Select the optimal atom based on maximum projection coefficient from Step (2) as $\varphi_{(\gamma,w)}$ given by Eq. 4.47

Step 4) Compute the inner product between the signal residues R_{fi}^k and optimal atom $\varphi_{(\gamma,w)}$ by $C_x^k = \langle R_{fi}^k, \varphi_{(\gamma,w)} \rangle$

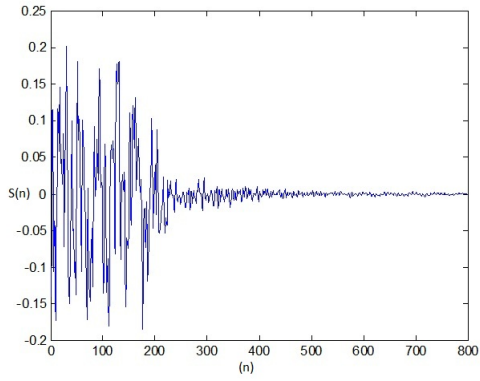
Step 5) Update the residues by $R_{fi}^{k+1} = R_{fi}^k - \langle R_{fi}^k, \varphi_{(\gamma,w)} \rangle \varphi_{(\gamma,w)}$

Step 6) Use Binary Masking explained in section 4.4 to recover the sources

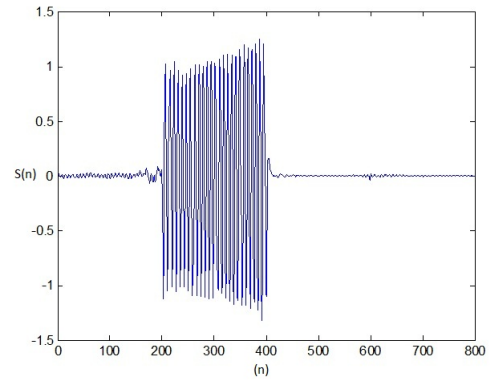
Step 7) If the stopping criterion has not reached, $k = k + 1$ and go back to step (2)

In each iteration the coefficients of the observed data C_x is approximated by the optimal atom belonging to a maximum correlated atom of the channel and the residues are updated. The sparse representation of the source coefficients are estimated in each iteration by $C_x \approx AC_s$.

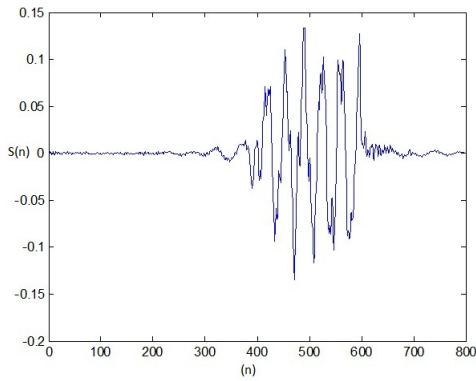
The same four source signals that were tested in section 4.3.1 along with the three mixtures produced by the same mixing matrix were tested with the proposed algorithm to make the nature of the algorithm clearer. The result obtained of the recovered signals is shown in Fig. 4.8. The routine converged with 2372 iteration which is almost less than half of what it took for the previous routine. More performance results will be discussed in section 5.



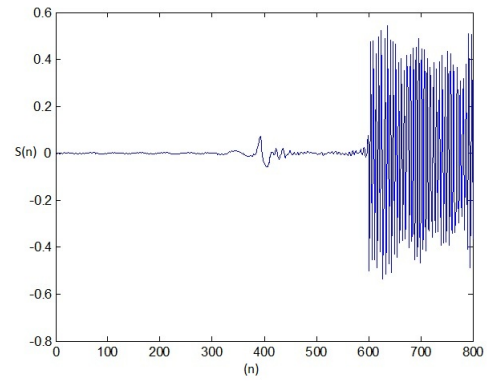
(a) Recovered Signal 1



(b) Recovered Signal 2



(c) Recovered Signal 3



(d) Recovered Signal 4

Figure 4.8: 4 Recovered Source Signals by multi-channel Matching Pursuit algorithm using Maximum Residue Atom Selection

4.4 Binary Masking

At this stage the problem becomes recovering the sparse sources with the known mixing matrix and the sparse representation of the mixtures C_x . The representation of the estimated sources is $\mathbf{s} = \mathbf{C}_s \Phi$ which assumes that $\mathbf{C}_x \approx \mathbf{A} \mathbf{C}_s$. If there exist high sparsity for the sources where there is only one active source, then in order to recover the sources Binary Masking is applied as a thresholding method to choose which coefficient of the sources are masked. Further, the degree of a source's activeness depends on the correlation between the corresponding estimated mixing matrix columns A_n and the component C_x of the mixtures.

The main assumption of sources having disjoint support still remains a crucial requirement for the source recovery effectively resulting in $\mathbf{C}_x \approx \mathbf{A}\mathbf{C}_s$.

4.4.1 Recovering Sparse Sources via Binary Masking

Knowing the mixing matrix with unit columns $\|a_i\|_2 = \sum_{i=1}^N a_i = 1$, and the optimal atom $\varphi_{(\gamma,w)}$, the atom will contribute to the source of index \hat{i}_s which possesses the most correlation of the column vector a_i and the coefficients of the mixture.

Let's denote I_p as indices of an active source in the coefficient mixtures C_x . Denoting A_I as the mixing matrix composed of A columns with index $k \in I$, where the indices of an active source is I , then the estimated source coefficients will be:

$$\begin{cases} C_s = A_I^\dagger C_x & \text{if : } k \in I \\ 0 & \text{if : } k \notin I \end{cases} \quad (4.50)$$

Next the set of I_p which belong to the active source is chosen in a way that it minimizes the mean square error of reconstruction of the mixture coefficients C_x , given by:

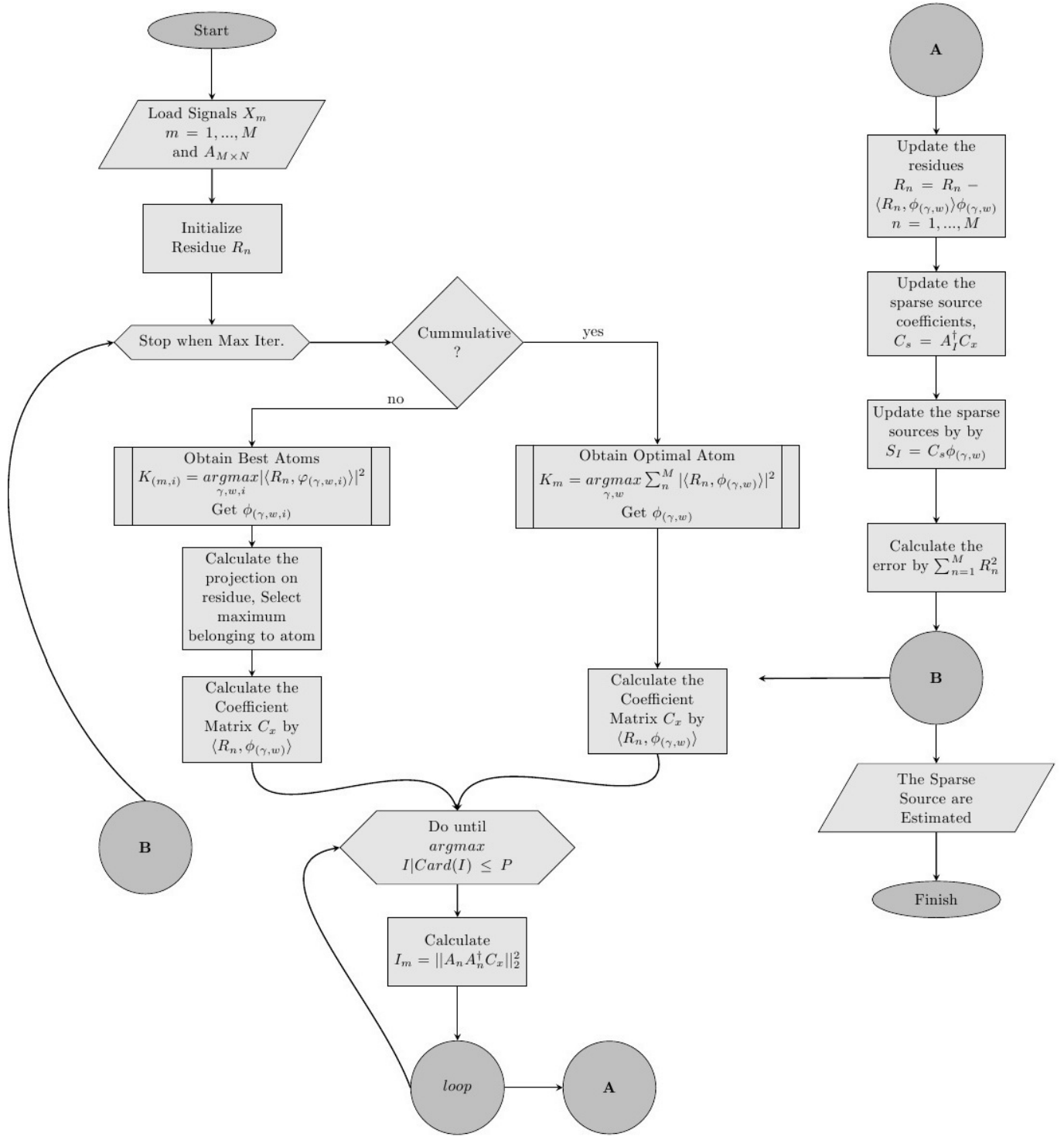
$$I_p = \arg \min_I \|C_x - A_I A_I^\dagger C_x\|_2^2 \quad (4.51)$$

$$I_p = \arg \max_I \|A_I A_I^\dagger C_x\|_2^2 \quad (4.52)$$

After the coefficients of the sources are calculated the source signals are recovered using the optimal atoms such as:

$$S_n = \sum_{n=p} C_s \varphi_{(\gamma,w)} \quad (4.53)$$

The matching pursuit algorithm routine is presented in Flowchart 4.2 where the algorithm recovers the sparse sources by an iterative greedy algorithm. The algorithm chooses the atom selection method based on the optimization criterion explained as cumulative selection in section 4.3.1 and maximum residue selection in section 4.3.3.



Flowchart 4.2: Multi-channel Matching Pursuit Algorithm

4.5 Proposed High Sparsity via Transform Domain

Higher sparsity is a requirement for separability of the sources in the case of under-determined blind separation. In this case the proposed approach is a linear transformation such that when applied to the signal a higher sparse representation is gained. The implementation of FFT will also help the calculation of the projection of a large signal on the optimal atoms. The correlation calculation will be much faster in the transform domain. Next, the blind source separation is performed in the transformed domain. The selected transform is the Fast Fourier Transform implementation of the Discrete Fourier Transform. The price to be paid is the increase number of manipulations but the overall gain in higher sparsity is more significant. This leads to data that will be represented with fewer coefficients. In addition, since the only given data in an under-determined blind source separation problem is the observation mixture, the transform domain starts with taking the Fast Fourier Transform (FFT) of the mixtures. Next, the mixtures have to be represented in their sparse form such as Eq. 4.2. The Matching Pursuit algorithm has to be applied with the corresponding optimal Gabor atom in the frequency domain. Therefore, by looking at the definition of the Discrete Fourier Transform of M data such as:

$$F_k = \sum_{j=0}^{M-1} f_j e^{(-2\pi i j k / M)} \quad (4.54)$$

it is observed that the above can be expanded in the form of :

$$F_k = \sum_{j=0}^{M-1} f_j \cos(2\pi j k / M) - i \sum_{j=0}^{M-1} f_j \sin(2\pi j k / M) \quad (4.55)$$

Further, by looking at Eq. 4.17 and Eq. 4.18 it can be simply observed that the dictionary can be partitioned using the above cosine and sine terms. The parameter of the extracted optimal atom in the transform domain is given by:

$$D_k(j) = g\left(\frac{j-u}{s}\right) \cos\left(\frac{2\pi j k}{N}\right) \quad (4.56)$$

$$O_k(j) = g\left(\frac{j-u}{s}\right) \sin\left(\frac{2\pi j k}{N}\right) \quad (4.57)$$

This problem can be analyzed by the structure of the dictionary implemented in the previous chapter. The special case will produce parameters in the following manner, with $t = j$ where

$\gamma = (s, u, v) = (s, u, \frac{2\pi k}{N})$, where $k = 1, \dots, N - 1$. Next the parameters needed to calculate the FFT and ultimately finding the appropriate optimal atom as $\langle f, \phi_\gamma \rangle$ are:

$$\langle f, D_k \rangle \quad \langle f, O_k \rangle \quad \langle D_k, D_k \rangle \quad \langle D_k, O_k \rangle \quad \langle O_k, O_k \rangle \quad (4.58)$$

The purpose of choosing these parameters is to take into account the real and imaginary parts of the Discrete Fourier Transform. Therefore, with the correct partition and input, there are few modifications performed on the existing Gabor dictionary. The allowable values for the parameter s are $s = 2^j$ for $j = 1, \dots, J$ and where J represents the power of 2. For a given value of s the frequency parameter is $v = \frac{2\pi k}{M}$, where $M = 8s$ and $k = 1, \dots, M - 1$. There are two cases to consider, since the value of $M = 8s$ and the FFT of the signal of length N is being taken. The first case is when $8s > N$ and the second is when $8s \leq N$.

Case 1) The assumption where $8s > N$ means the support is taken over the whole interval, where input $x_j = f(j)g\left(\frac{j-u}{s}\right)$ and where $M = N$, next for $k = 1, \dots, M - 1$,

$$\begin{aligned} \langle f, D_k \rangle &= \text{output.Real}[X_k] \\ \langle f, O_k \rangle &= -\text{output.Real}[X_k] \end{aligned} \quad (4.59)$$

By keeping the relation above in mind, the appropriate shifts are also calculated where $C = \text{input.real}[0]$,

$$\begin{aligned} \langle D_k, D_k \rangle &= \frac{1}{2}(C + \text{output.Real}[2X_k]) \\ \langle O_k, O_k \rangle &= \frac{1}{2}(C - \text{output.Real}[2X_k]) \\ \langle O_k, D_k \rangle &= -\frac{1}{2}\text{output.Imag}[2X_k] \end{aligned} \quad (4.60)$$

Case 2) The assumption where $8s \leq N$ means the support is smaller than the whole interval, where $M = 8s$ which represents fewer data points. Based on the appropriate shift parameter u there are three cases to consider. One where the shift u falls in the section of $0 \leq u \leq 4s$, which is similar to the procedure applied as before.

The next shift zone will fall into the range where the left hand point of the interval is no longer in the support of the Gaussian window. Therefore, to consider this the indices are

translated where the input $x_j = f(j+u-4s)g\left(\frac{j-4s}{s}\right)$ leads to a modulation in the frequency domain of $f_{mod} = 2\pi \frac{(u-4s)k}{M}$.

$$\langle f, D_k \rangle = \cos(f_{mod}) \text{ output.Real}[X_k] + \sin(f_{mod}) \text{ output.Imag}[X_k]$$

$$\langle f, O_k \rangle = \sin(f_{mod}) \text{ output.Real}[X_k] - \cos(f_{mod}) \text{ output.Imag}[X_k]$$

By keeping the relation above in mind, the appropriate shifts with $k = 0, 1, \dots, \frac{M}{2} - 1$ give:

$$\langle D_k, D_k \rangle = \frac{1}{2}(C + \cos(f_{2mod})\text{output.Real}[2X_k]) + \sin(f_{2mod}) \text{ output.Imag}[2X_k]$$

$$\langle O_k, O_k \rangle = \frac{1}{2}(C - \cos(f_{2mod})\text{output.Real}[2X_k]) - \sin(f_{2mod}) \text{ output.Imag}[2X_k]$$

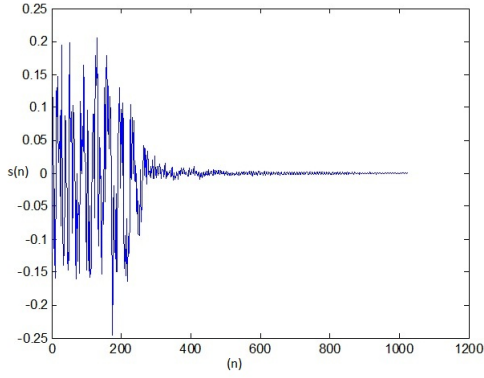
$$\langle O_k, D_k \rangle = \frac{1}{2}(\sin(f_{2mod}) \text{ output.Real}[2X_k] - \cos(f_{2mod})\text{output.Imag}[2X_k]) \quad (4.61)$$

Next when $k = \frac{M}{2}, \dots, M - 1$ the same procedure is repeated with k replaced. For the last case where the Gaussian window support is past the right hand endpoint of the interval, zero padding is used. For the case of $j = 0, 1, \dots, N - 1 - u + 4s$, x_j will be defined as previously stated, and if j is not at this value then $x_j = 0$.

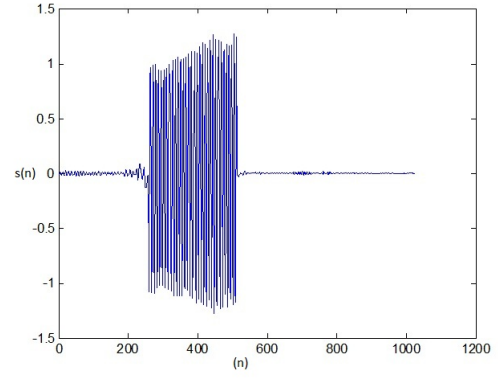
The framework of this research is based on sparse signals and their recovery given under-determined conditions. However, it is also significant to look at the case where the signal can be presented in a higher sparsity manner. The reason of using this transform domain is to gain better sparsity. Doing so the mixture signals will be represented with fewer coefficients which means it will increase the overall sparsity. This is an important factor that will be taken into consideration when working with sparse representations and recovering sources.

An example is given here to make it easier to understand and show how the mentioned procedure operates. The mixture signals used in this example are the same four signals used in the example in section 4.4.1 with $16KHz$ and 1024 samples points. These signals were chosen and they were mixed with the same random mixing matrix producing 3 mixtures. These mixtures were passed onto the multi-channel Matching Pursuit algorithm that uses

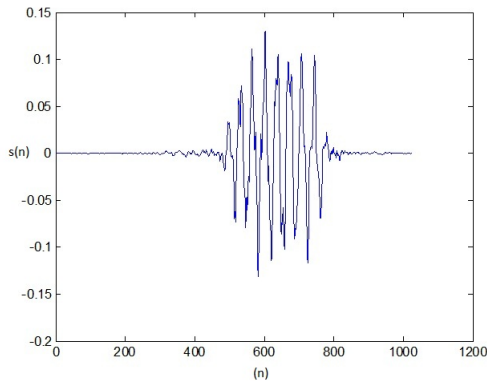
FFT implementation. After the algorithm converged the sparse source signals were recovered which are shown in Fig. 4.9. It is important to note that the algorithm converged with 2050 iterations and a chosen error of 10^{-9} .



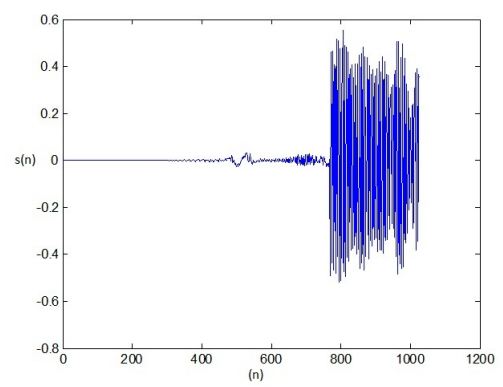
(a) Recovered Signal 1



(b) Recovered Signal 2



(c) Recovered Signal 3



(d) Recovered Signal 4

Figure 4.9: Recovered Source Signals by multi-channel Matching Pursuit Algorithm using Transformed Domain

More accurate examples and performance evaluations for the algorithms mentioned in this chapter are stated in Chapter 5.

4.6 Summary

The second stage of the problem is addressed through a non-linear approach where the sparse sources are recovered. This stage involves representing the source signals by a linear combination of a few elementary atoms from a dictionary. The dictionary is a redundant dictionary of infinite number of possible Gabor functions which allow the reconstruction of a particular signal. The importance of dictionary and how it relates to sparse source recovery is mentioned throughout the chapter. To decompose a particular signal into its most sparse elements, a Matching Pursuit algorithm that is a non-linear greedy method is introduced. In order to separate the sources, the atom of the dictionary is defined based on the optimization criterion that leads to a maximum projection on the signal. This is expanded to the multi-channel case based on an optimization criterion that involves choosing an atom for more than one signal in the Matching Pursuit algorithm. This type of atom selection is called the Cumulative Atom Selection. There are simple examples at the end of each section which shows how each scheme operates. The proposed method involves an optimization criterion which extracts one atom for a few signals. This optimal atom is maximally correlated with at least one of the signals. In addition, a binary masking scheme is mentioned for assigning each of the mixture coefficients to the active source at different instances. An addition to this chapter is the introduction of sparse signal representation using the transform domain. The dictionary is altered to handle the Fast Fourier Transform case where it shows improvement in the sparse representation of a particular signal.

Chapter 5

Simulation and Results

In this chapter the simulations are shown and the results are analyzed in detail to outline the performance of the proposed methods. In the first section of this chapter the results of the estimated mixing matrix via HLC algorithm with the proposed initialization scheme is shown. This section was implemented using Matlab R2012a, with an Intel processor of 3.30 GHz. In the following section the results of the source separation stage using the multi-channel Matching Pursuit algorithm is mentioned. The second stage was implemented using C++, Linux with a Pentium 4, 2.20 GHz.

5.1 Estimation of the Unknown Mixing Matrix

Here the HCL algorithm is demonstrated to identify the unknown basis matrix $\hat{\mathbf{A}}$ from the given observed data. The results of the proposed method is compared to the conventional method in cases which are different and produce more challenging benchmarks. These examples are challenging due to the following reasons:

- 1) There are many outliers in the system,
- 2) The basis matrix is very ill-conditioned,
- 3) The problem is relatively large scale.

In order to verify how well the basis matrix is estimated, the following Signal Interference Ratio as a performance index is introduced which takes into consideration the distance between the column vector of the mixing matrix and the corresponding estimated one.

$$SIR(\hat{A}, A) = -20 \log_{10} \left(\frac{1}{n} \sum_{i=1}^n \min_{j=1, \dots, n} \frac{\min\{\|a_i - \hat{a}_j\|_2, \|a_i + \hat{a}_j\|_2\}}{\|a_i\|_2} \right) (dB) \quad (5.1)$$

where $\hat{\mathbf{A}}$ is the estimated mixing matrix by the algorithms implemented and \mathbf{A} is the true mixing matrix used to create the mixtures from the sources.

Example 1) In the first example sparse sources were created using 4 different audio signals at 44.1 KHz, where sparsity is high based on the assumptions. The sources were mixed using a predefined 3×4 mixing matrix. Next the 3 mixtures were inputted into the clustering algorithm. The HCL algorithm with overestimated initialized basis matrix using the correlation method converged with the correct estimated matrix and a squared sum error of $1.4540 e^{-15}$. The 4 sources are observed to be the highest confidence values and the rest are spurious in Fig. 5.1.

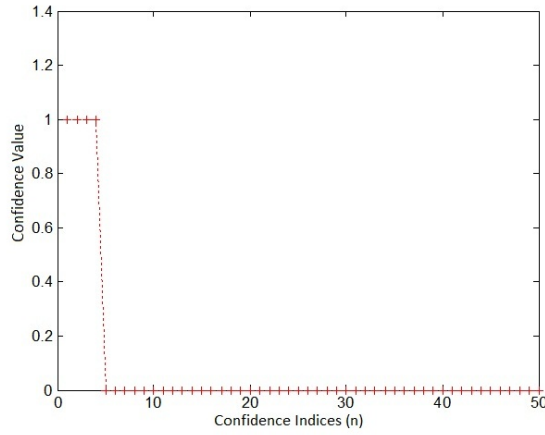


Figure 5.1: Confidence Plot of Correlation Method

Here it is seen that the 4 correct mixing matrix columns are estimated and separated from the spurious ones by a gap. The algorithm was tested 20 times and the results with minimum, maximum and average SIR values are shown in Table 5.1.

Table 5.1: SIR Performance (dB) of Matrix Identification Algorithm (Example 1)

Method	$SIR_{min}(dB)$	$SIR_{max}(dB)$	SIR Average of 20 runs (dB)
HCL using Correlation	305.21	309.85	307.46

The matrix \mathbf{A} below on the right is the true mixing matrix that was used to create the mixtures from the original sources and the estimated matrix $\hat{\mathbf{A}}$ on the left is the matrix that was obtained by the clustering algorithm with overestimated correlation scheme.

$$\hat{A} = \begin{bmatrix} -0.8256 & 0.3828 & -0.3994 & -0.7218 \\ 0.2008 & -0.7022 & 0.7950 & -0.5609 \\ 0.5273 & 0.6003 & 0.4566 & 0.40530 \end{bmatrix} \quad A = \begin{bmatrix} -0.8256 & 0.3828 & 0.3994 & 0.7218 \\ 0.2008 & -0.7022 & -0.7950 & 0.5609 \\ 0.5273 & 0.6003 & -0.4566 & -0.4053 \end{bmatrix}$$

In the above example the correlation was used to initialize the matrix where the basis matrix was overestimated to 50 hyperlines. This method is accurate and gives result however, in the next example the result of the proposed method is shown where the polar clustering is performed with a least square linear curve fitting to initialize the basis matrix.

The same inputs as the previous examples were used in the following simulation. The mixtures were plotted in a scatter plot to show the sparsity in Fig. 5.2.

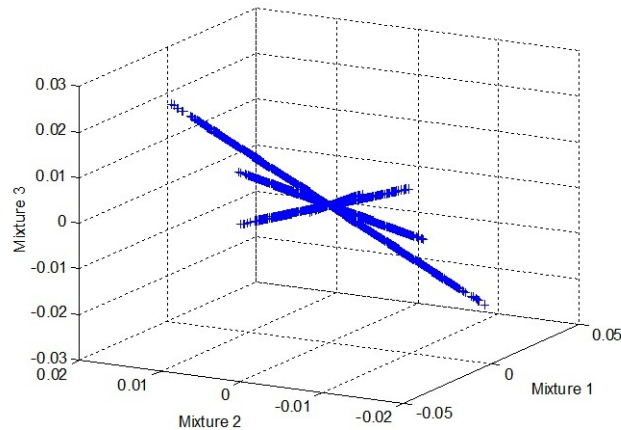


Figure 5.2: Scatter Plot of 3 Mixtures

It is seen that the sources have high sparsity and therefore, the cluster lines can be separated and using least square linear curve fitting where the equation can be obtained by looking at each 2 dimensional spaces such as in Fig. 5.3.

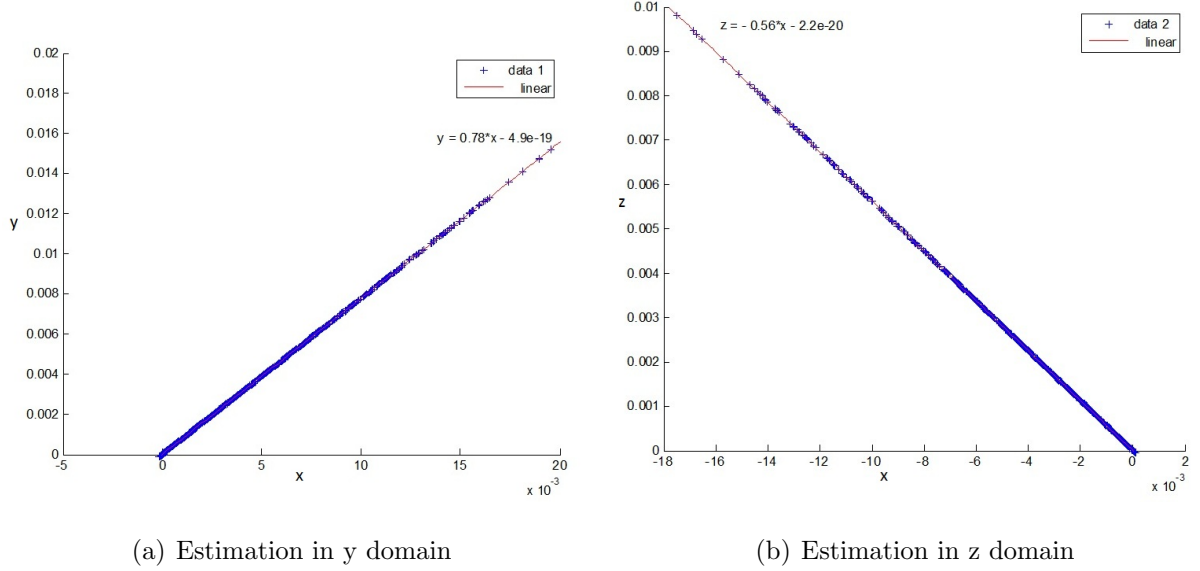


Figure 5.3: Least Square Linear Curve Fitting

As seen in Fig. 5.2 all of the 4 scatter lines can be approximated and the basis matrix can be initialized using only these 4 obtained hyperlines. These lines are clustered into 4 different sections and the equations are obtained using the linear curve fitting tool shown in Fig. 5.3. The most significant advantage of this initialization scheme is that the overestimation factor is not an issue anymore. Instead of 50 initialized hyperlines, now there exist only 4 hyperlines and the HCL algorithm will have less manipulations. In the previous scheme the basis matrix was initialized as large as $M \times T$ where T was the duration of the mixture signal and the data was partitioned to get 50 of the best correlated hyperlines. Therefore, the mixing matrix obtained was largely overestimated in the initial stage.

The reason that this overestimation was performed is because since the mixtures have disjoint temporal support the overestimated matrix would cover all the true directions of the mixing matrix. Next, by applying the Hyperline Clustering algorithm with the overestimated matrix, it would move along the columns of the mixing matrix and update the overestimated clusters entries with their corresponding Eigenvectors. This procedure is simple and works

however, the amount of calculation is large. Since the distance of each data point must be calculated to each hyperline and consecutive EVD calculations are needed for each cluster update, the proposed method will significantly reduce the number of steps within the algorithm. The proposed algorithm does not randomly initialize the basis matrix, however, it guides the clustering algorithm by passing an initialized basis matrix that is nearly in the direction of the true solution. Next, another advantage of the proposed technique is that the basis matrix entries are not overestimated to a large value but precisely almost the same number of the sources. This is an important contribution to this segment of the research and the obtained result and performance is looked at below.

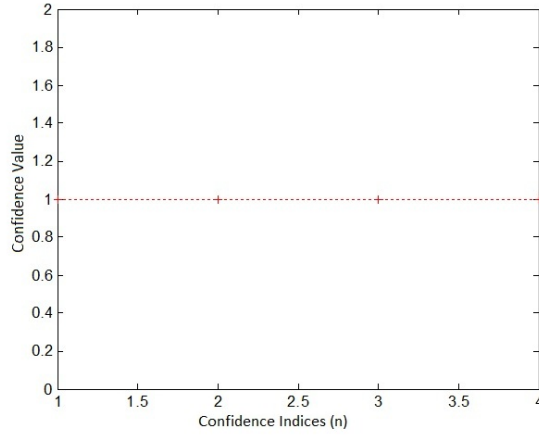


Figure 5.4: Confidence Plot of Proposed Method

As seen in Fig. 5.4 the confidence plot shows only the 4 true optimal hyperlines which are the same number of hyperlines initialized by the proposed technique. Therefore, there is no overestimation of the initialized basis matrix. The proposed method initializes the hyperlines nearly as close as possible to the true solution. In table 5.2 the performance of the proposed method is calculated by the SIR evaluation of the original mixing matrix and the estimated one by the proposed scheme.

Table 5.2: SIR Performance (dB) of Matrix Identification Algorithm (Example 1)

Method	$SIR_{min}(dB)$	$SIR_{max}(dB)$	SIR Average of 20 runs (dB)
Proposed HCL	305.96	310.02	308.15

The matrix A below on the right is the true mixing matrix that was used to create the mixtures from the original sources and the estimated matrix \hat{A} on the left is the matrix that was obtained by the proposed scheme.

$$\hat{A} = \begin{bmatrix} -0.8256 & 0.3828 & -0.3994 & -0.7218 \\ 0.2008 & -0.7022 & 0.7950 & -0.5609 \\ 0.5273 & 0.6003 & 0.4566 & 0.4053 \end{bmatrix} \quad A = \begin{bmatrix} -0.8256 & 0.3828 & 0.3994 & 0.7218 \\ 0.2008 & -0.7022 & -0.7950 & 0.5609 \\ 0.5273 & 0.6003 & -0.4566 & -0.4053 \end{bmatrix}$$

By comparing table 5.1 which belongs to the correlation scheme to table 5.2 belonging to the proposed scheme, it is observed that the proposed method produces results that are slightly better than the latter scheme. However, the advantage is in the number of calculations needed for the clustering algorithm to converge. Since the initialized basis matrix is not overestimated there is less computational weight. The method converges using less iteration and in less time since it does not need to overestimate the initialized basis matrix. The next examples are shown for different cases with larger mixing matrix. It is important to note that the first example provides a perfect matrix estimation scenario to illustrate that the algorithm implemented is able to estimate the matrix accurately in perfect conditions. However, in the next example there are more outliers, low sparsity and a larger data considered for the inputs.

Example 2) In this example there are 7 speech sources with a predefined 3×7 mixing matrix. The mixtures were passed onto the HCL algorithm and both of the implementations were tested and the results obtained are shown. Below is the mixing matrix that was generated to mix the sources.

$$A = \begin{bmatrix} 0.7930 & -0.7428 & 0.1404 & 0.9021 & 0.3884 & 0.3684 & 0.7283 \\ 0.1480 & -0.5901 & 0.7010 & -0.3691 & -0.7214 & -0.7262 & 0.5695 \\ -0.5910 & -0.3161 & -0.6992 & -0.2235 & 0.6036 & -0.4179 & -0.4533 \end{bmatrix}$$

The recovered mixing matrix by the first implementation which considers initialization using

correlation and overestimating the basis matrix, yields the following result. The gap of the confidence plot in Fig. 5.5 separates the optimal solution and the spurious hyperlines.

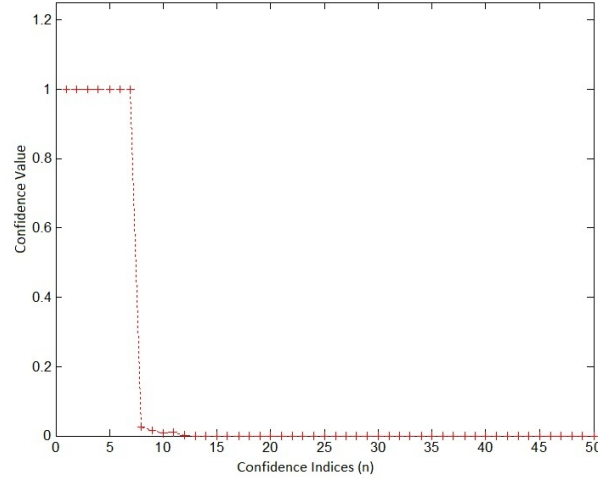


Figure 5.5: Confidence Plot of Correlation Method

The estimated matrix by the clustering algorithm with overestimated initial basis matrix is given by:

$$\hat{A} = \begin{bmatrix} -0.7930 & 0.7428 & -0.1404 & -0.9021 & 0.3817 & -0.3702 & -0.7193 \\ -0.1480 & 0.5901 & -0.7010 & 0.3691 & -0.7189 & 0.7334 & -0.5581 \\ 0.5910 & 0.3161 & 0.6992 & 0.2235 & 0.5991 & 0.4266 & 0.4482 \end{bmatrix}$$

The same input signals and mixing matrix is tested with the proposed algorithm. The estimated matrix converges in much less time since there are no overestimations of the basis matrix. The algorithm initialized 7 hyperlines for the basis matrix and the estimated mixing matrix obtained is given by:

$$\hat{A} = \begin{bmatrix} -0.7930 & 0.7428 & -0.1404 & -0.9021 & 0.3828 & -0.3695 & -0.7283 \\ -0.1480 & 0.5901 & -0.7010 & 0.3691 & -0.7214 & 0.7339 & -0.5670 \\ 0.5910 & 0.3161 & 0.6992 & 0.2235 & 0.6024 & 0.4267 & 0.4523 \end{bmatrix}$$

The estimated matrix is nearly identical to the true mixing matrix where the algorithm does not overestimate the basis matrix in the initialization stage. The initialized matrix by the proposed method contains 7 hyper lines which are nearly on the scatter lines, therefore producing accurate and fast converging results. The confidence plot is shown in Fig. 5.6.

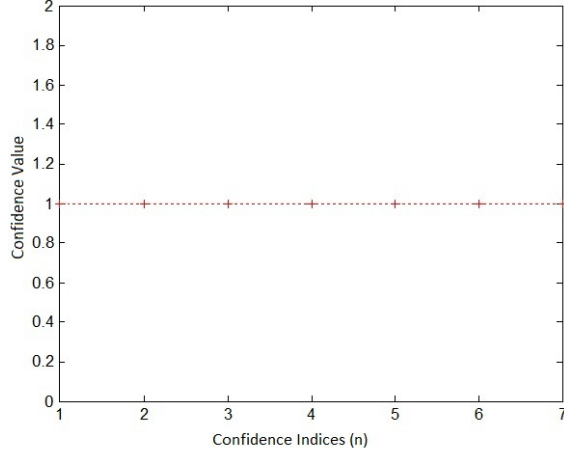


Figure 5.6: Confidence Plot of Proposed Method

Table 5.3 compares the performance of the previous correlation performance to the proposed scheme. The performance of the proposed algorithm is better than the conventional method because there are less initialized dimensions for the basis matrix.

Table 5.3: SIR Perfomance (dB) of Matrix Identification Algorithms (Example 2)

Method	$SIR_{min}(dB)$	$SIR_{max}(dB)$	SIR Average of 20 runs (dB)
Correlation used in HCL	39.21	45.38	43.15
Proposed Method used in HCL	40.42	48.02	45.56

Example 3) This example considers the case where the data is against the assumption made on sparsity. The reason of testing this case is to reveal the robustness of the algorithm and to underline the importance of the relation between sparsity and the linear transform domain. Here data is presented in the time domain which is not sparse. Therefore, in order to take advantage of the implementation of the clustering schemes proposed, the data

is transformed into the time-frequency domain using Fast Fourier Transform. When the mixtures are obtained, it can be seen that there exists no sparsity in the time domain. The scatter plot in Fig. 5.7 shows no alignment in the direction of the column of the mixing matrix.

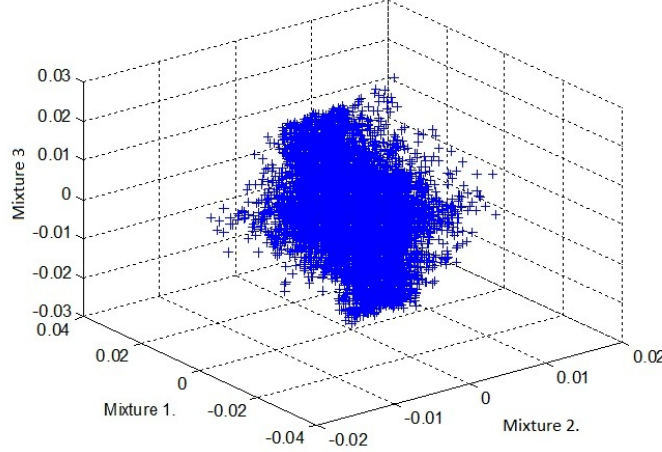


Figure 5.7: Scatter Plot of Mixtures with Non-Disjoint Temporal Support

Next, the mixtures are processed into frames of length L and multiplied by a Hanning window. Then a hop distance of d is used between the starting points of the successive frames. Each frame was transformed using a Fast Fourier Transform of length L , and concatenated to form the mixtures in the frequency domain. The scatter plots of the transformed domain in Fig. 5.8 shows the sparsity which reveals hyperlines aligning in the direction of the columns of the true mixing matrix \mathbf{A} given by:

$$\mathbf{A} = \begin{bmatrix} 0.3351 & 0.1164 & 0.7361 & -0.4133 \\ -0.8367 & 0.7810 & -0.3214 & -0.4917 \\ 0.4333 & 0.6136 & 0.6659 & 0.7665 \end{bmatrix}$$

After applying the STFT as mentioned above, the scatter plot shown in Fig. 5.8 reveals lines directed along the columns of \mathbf{A} .

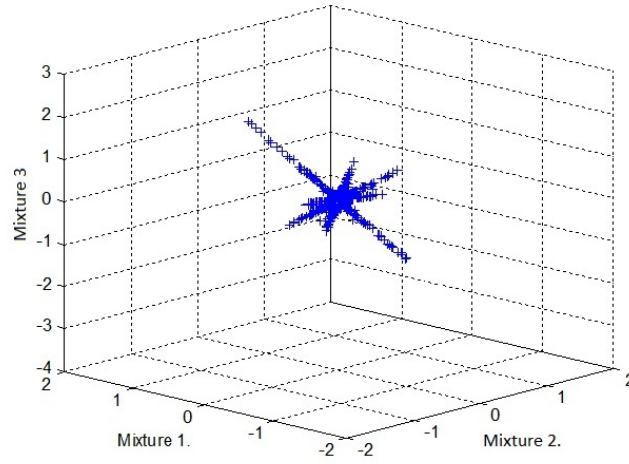


Figure 5.8: Scatter Plot in Transform Domain

Using high sparse data obtained in the transformed domain, the mixtures are passed into the clustering algorithms. The performance of the correlation scheme and the proposed method in terms of estimated mixing matrix, confidence plots and SIR values are shown next. The overestimated number of hyperlines in the initialized basis matrix by the correlation method converges with confidence plot shown in Fig. 5.9.

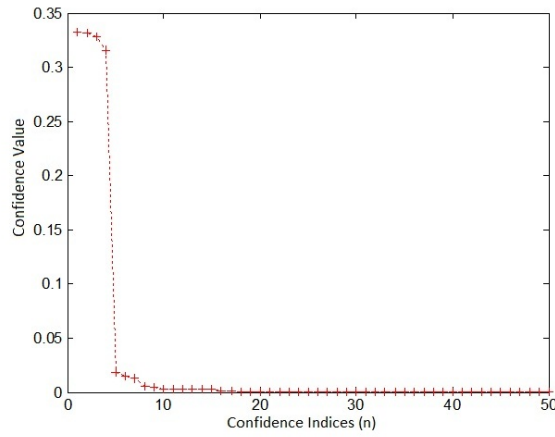


Figure 5.9: Confidence Plot of Correlation Method

The estimated basis matrix is separated by the gap from the spurious hyperlines. The estimated mixing matrix obtained by this scheme is:

$$\hat{A} = \begin{bmatrix} 0.3353 & 0.1157 & 0.7354 & -0.4131 \\ -0.8377 & 0.7807 & -0.3185 & -0.4917 \\ 0.4337 & 0.6141 & 0.6661 & 0.7666 \end{bmatrix}$$

Next the proposed method is used where the polar clusters separate each line and the least square linear curve fitting is used to initialize each cluster. The obtained results are shown below where the confidence plot in Fig. 5.10 shows that there are only 4 initialized hyperlines instead of 50 and the scheme converges using less manipulations and less time.

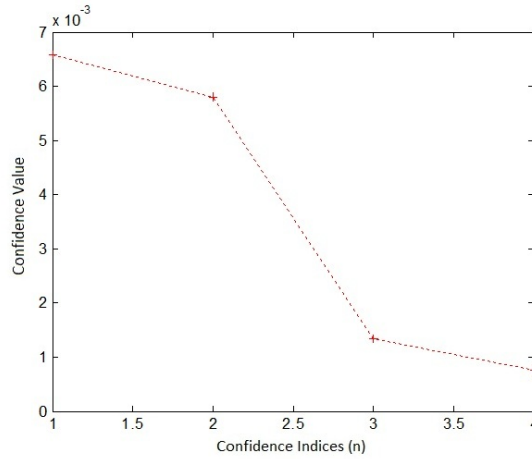


Figure 5.10: Confidence Plot of Proposed Method

The estimated mixing matrix obtained by using the proposed scheme is given by:

$$\hat{A} = \begin{bmatrix} 0.3356 & 0.1164 & 0.7360 & -0.4133 \\ -0.8370 & 0.7810 & -0.3117 & -0.4919 \\ 0.4331 & 0.6134 & 0.6660 & 0.7666 \end{bmatrix}$$

In the next stage, 20 iterations of the proposed scheme and the correlation algorithm are compared and the results are shown in Table 5.4.

Table 5.4: SIR Performance (dB) of Matrix Identification Algorithms (Example 3)

Method	$SIR_{min}(dB)$	$SIR_{max}(dB)$	SIR verage of 20 runs (dB)
Correlation used in HCL	45.78	52.54	46.95
Proposed Method used in HCL	46.31	54.55	50.46

The correlation scheme was tested with only 4 initialized hyperlines and the HCL algorithm failed to converge. This shows that the initialization scheme proposed can produce better results with less computational weight.

5.2 Separation of Sparse Sources

In this section observed mixtures produced by M sensors along with the estimated mixing matrix from the previous section are used to approximate N sparse sources. The separations of the sources from the mixtures are accomplished by using a non-linear approach. The multi-channel Matching Pursuit algorithm with real Gabor dictionary schemes explained in section 4 is used. The assumption of sparsity and temporal support for the mixtures still hold. Here the two methods that are used consist of the cumulative atom selection which is the conventional method and the proposed maximum residue atom selection. The number of iterations spent to achieve a set desired error for signal separation is compared along with the Signal Interference Ratio given by:

$$SIR(s, \hat{s}) = 10 \log_{10} \frac{\sum_{t=1}^T s^2(t)}{\sum_{t=1}^T (s(t) - \hat{s}(t))^2} (dB) \quad (5.2)$$

where s is the original sources and \hat{s} is the recovered sources by the multi-channel Matching Pursuit topologies.

Example 1) In the first example there are 4 audio sources of length 512 with full disjoint temporal support and frequencies of 44.1kHz mixed with a 3×4 random mixing matrix.

The mixing matrix is estimated by the previous section and used in the source recovery procedure. The original sources are shown in Fig. 5.11.

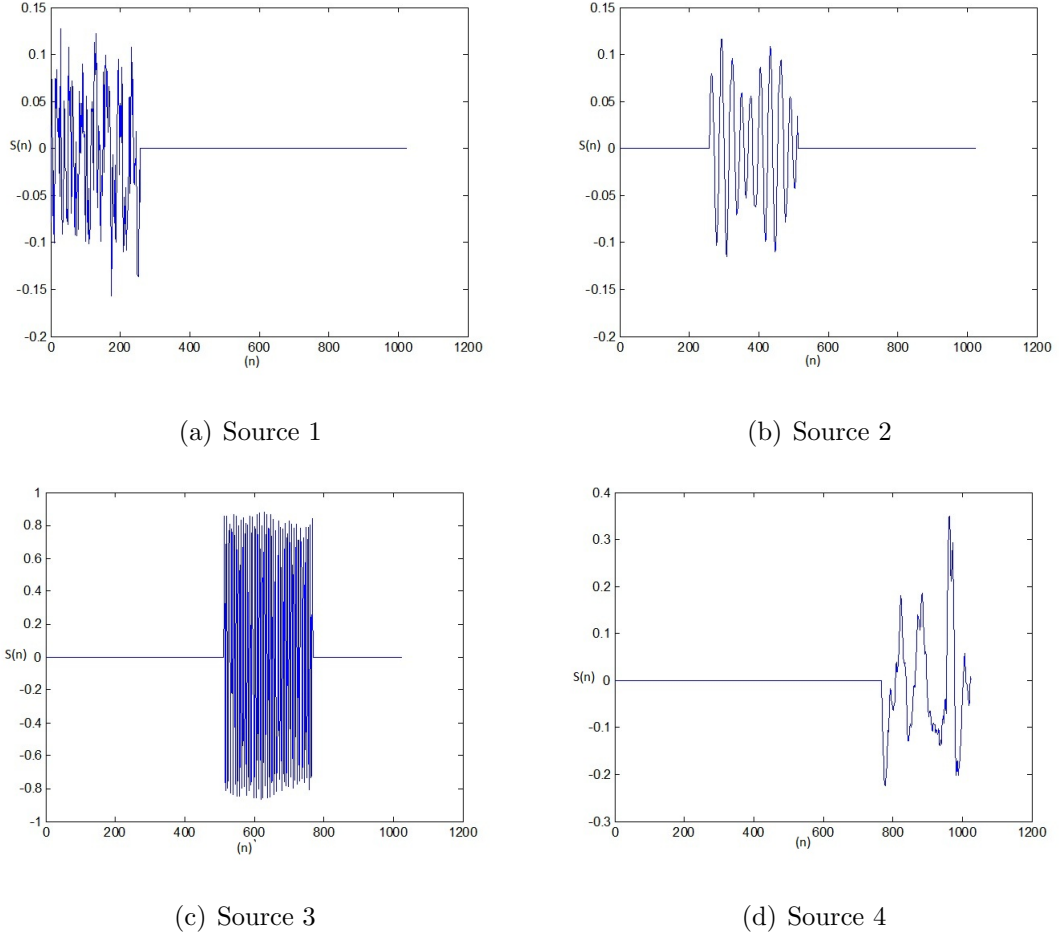
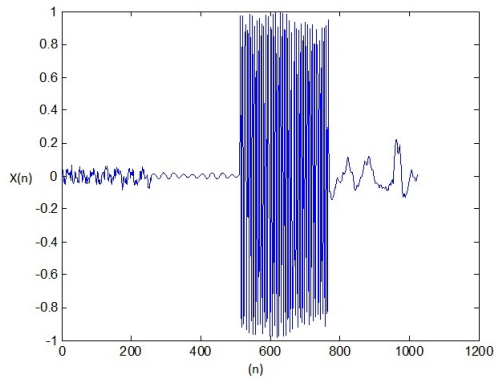
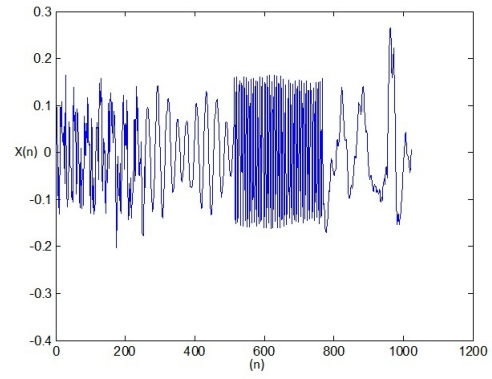


Figure 5.11: Original Sources

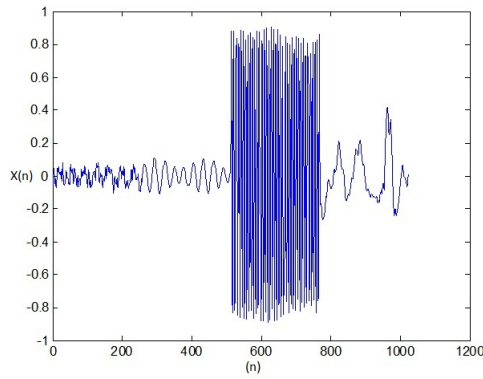
Next, the mixtures produced by the sources and the mixing matrix are passed into the multi-channel scheme. In the first evaluation the conventional method which uses the cumulative atom criterion is tested. The mixtures are shown in Fig. 5.12 where they are produced by 3 sensors. In each iteration the scheme selects one atom for the 3 mixtures. This optimal atom must be best correlated with all the residue signals. The coefficients are calculated by the projection of the atom on each residue signal. The residues are next updated by removing the projection from the signal. In this manner the signal is decomposed into a linear combination of elementary functions.



(a) Mixture 1



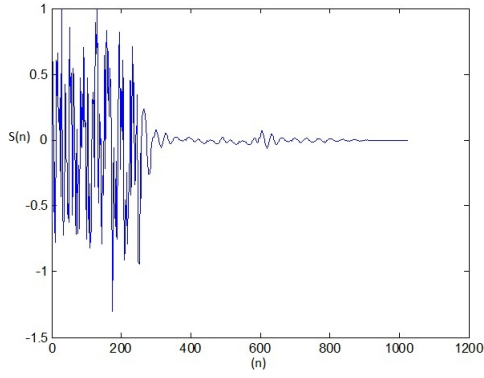
(b) Mixture 2



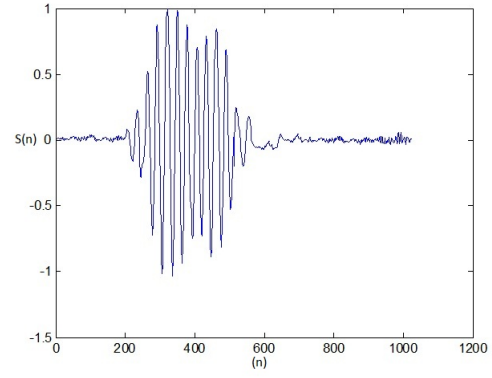
(c) Mixture 3

Figure 5.12: Mixtures

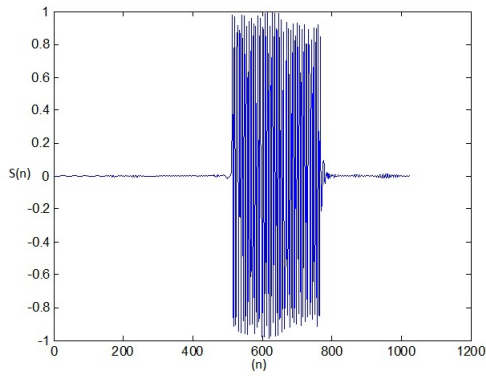
The recovered signals are shown in Fig. 5.13. First the joint sparse representation of the mixtures is obtained and then each source is assigned a coefficient. An optimal atom contributes to the coefficients and in each iteration the source that possesses maximum correlation with the column vector of the mixing matrix is assigned the estimated projection. This is achieved by the binary masking scheme which uses minimization of mean square error for the reconstruction of sparse sources. The recovered signals in this figure can be compared to the original signals in Fig. 5.11 and it is seen that the scheme perfectly recovers the signals.



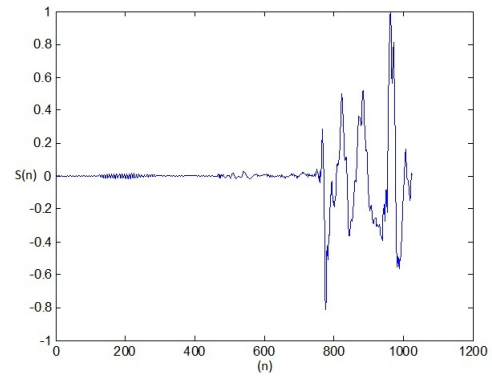
(a) RecoveredSource 1



(b) RecoveredSource 2



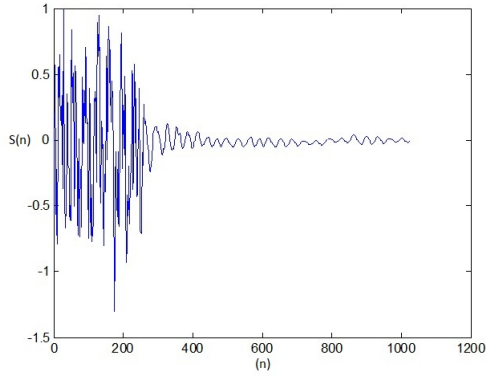
(c) RecoveredSource 3



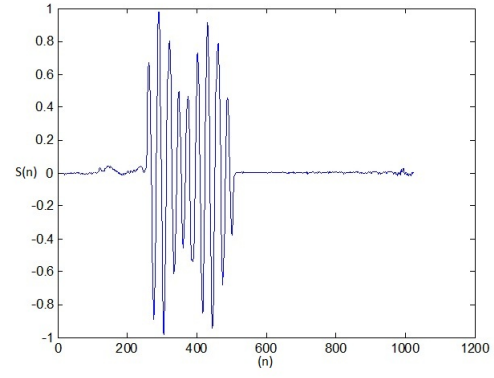
(d) RecoveredSource 4

Figure 5.13: Recovered Sources by Cumulative Atom Selection

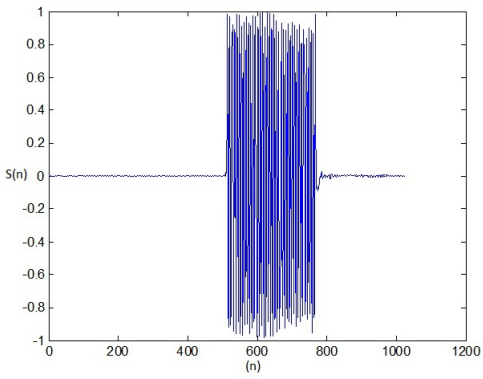
In the next stage the proposed method which uses a different optimization criterion is tested. In the proposed scheme one atom is chosen for the mixture signals. This optimal atom not only correlates with all the signals but is maximally correlated with at least one of the residue signals. The same mixture signals in Fig. 5.12 are used with the proposed method. The multi-channel Matching Pursuit algorithm with maximum residue atom extraction method recovered the signals accurately and is shown in Fig. 5.14.



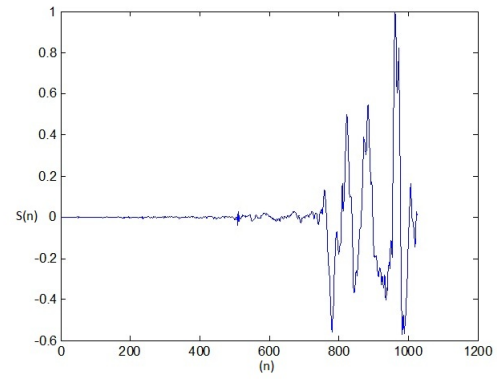
(a) RecoveredSource 1



(b) RecoveredSource 2



(c) RecoveredSource 3



(d) RecoveredSource 4

Figure 5.14: Recovered Sources by Maximum Residue atom Selection

To compare the performances of these schemes the Signal Interference Ratio of the original and recovered sources are calculated by Eq. 5.2. The estimated sparse sources may have arbitrary scale; therefore, it is rescaled to have the same energy as the original sources before computing SIRs. In Table 5.5 the conventional method that uses cumulative atom selection is compared with the proposed maximum residue atom selection and the SIR performances are stated. The table shows that the proposed algorithm has slightly better performance in recovering the sources. However, the gain in performance for the proposed method is in the number of iterations or atoms it uses to recover the signal. By examining both of the results, it is shown that the optimization criterion proposed uses less atoms to recover the signals from the mixtures in comparison to the conventional approach.

Table 5.5: SIR Performance (dB) of Sparse Source Recovery (Example 1)

Method	$SIR_{min}(dB)$	$SIR_{max}(dB)$	Average of 4 SIRs (dB)
Former Cumulative Atom selection			
selection	18.36	31.89	26.72
Proposed Maximum Residue Atom			
selection	20.87	32.62	28.95

It is important to keep in mind that the mixture signals do not change and the atoms selected in each stage for both of the schemes will be the same in each test run case. Therefore, there is no need for computing the signal recovery scheme more than once since the same atom is selected by the chosen scheme every time. The cumulative scheme spent 6627 iterations to converge with an error of 10^{-12} and the same desired error was achieved with the proposed maximum residue atom selection scheme with only 3459 iterations. This shows that the optimization criterion is much important in the convergence of multi-channel Matching Pursuit algorithm.

The next scheme takes advantage of Fourier Transform algorithm mentioned in section 4.5. The proposed algorithm uses the same scheme as maximal residue for its atom selection, however it is implemented by transforming the signal using Fourier Transform. The number of calculations within the algorithm increases but the overall manipulations in terms of the number of atoms selected for a desired error convergence greatly decreases. This is important in this research because the signal recovered must be represented in its most sparse form. When employing this scheme there are fewer atoms selected to recover the signals. Therefore, having such method increases sparse representation as oppose to other schemes and is considered faster, efficient and definitely higher in sparsity. The same mixtures were used with this scheme and it took only 3021 atoms or iterations for the source separation with an error of 10^{-12} . The results of this scheme are also shown in Fig. 5.15 with the recovered sources listed first and the SIR performance values in Table 5.6.

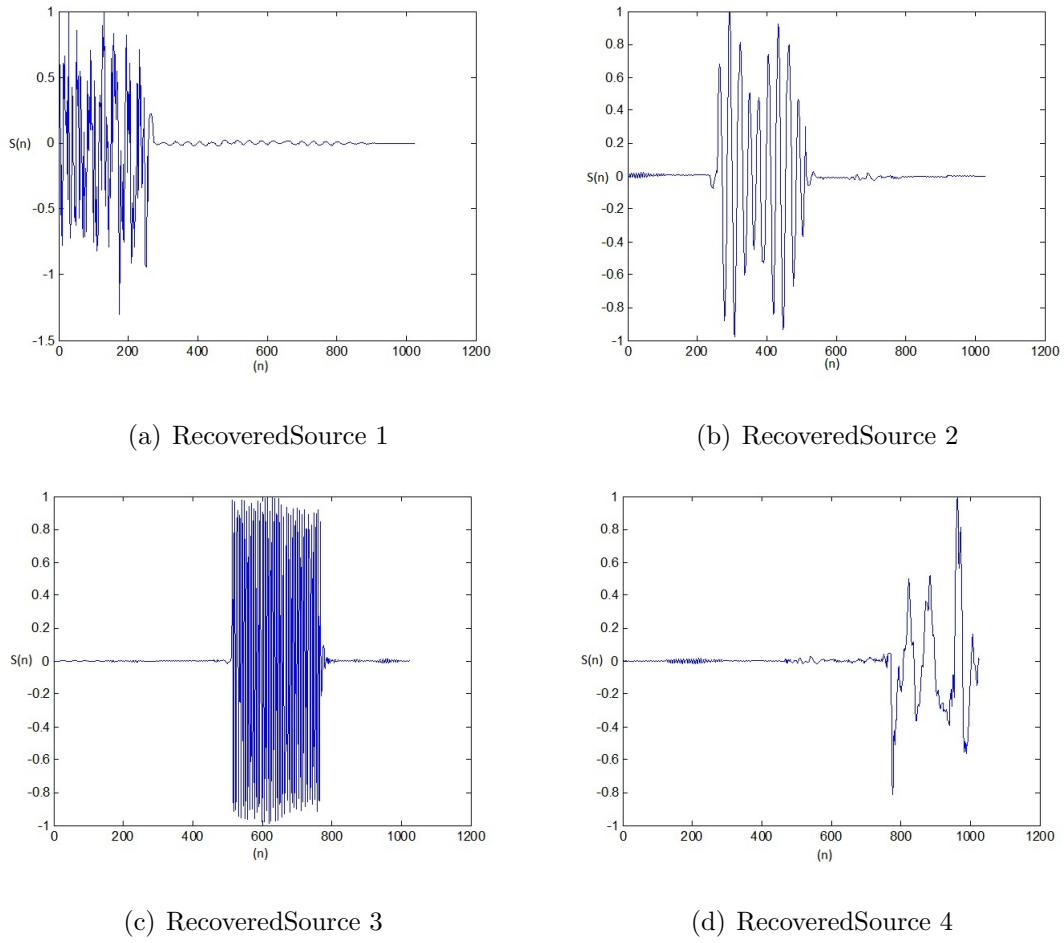
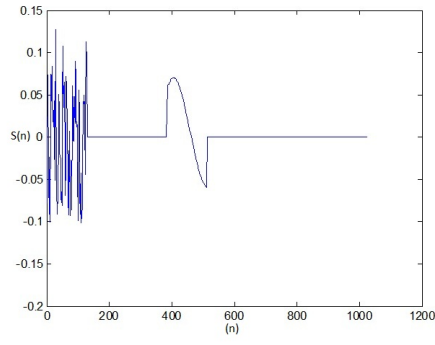


Figure 5.15: Recovered Sources by Maximum Residue atom Selection Using FFT

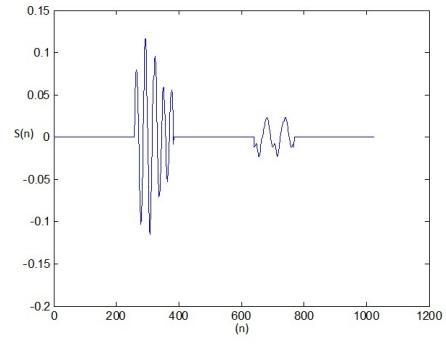
Table 5.6: SIR Performance (dB) of Sparse Source Recovery (Example 1)

Method	$SIR_{min}(dB)$	$SIR_{max}(dB)$	Average of 4 SIRs (dB)
Proposed Maximum Residue Atom			
selection using FFT	19.79	33.42	29.05

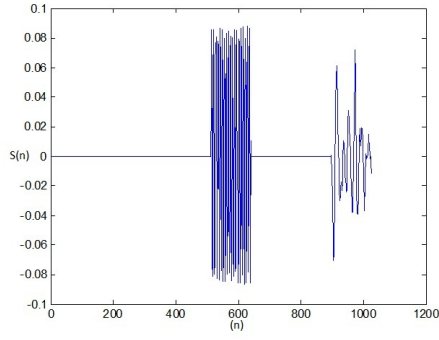
Example 2) In the next example 1024 samples of 4 audio sources at 44.1 KHz are used as original sparse sources. They are mixed using a 3×4 mixing matrix which was estimated by the proposed HCL algorithm. These mixtures are used with the multi-channel Matching Pursuit algorithms which were able to recover these sources. The sources and their mixtures are shown in Fig. 5.16 and Fig. 5.17 respectively.



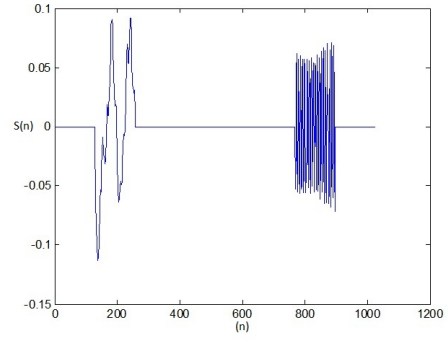
(a) Source 1



(b) Source 2

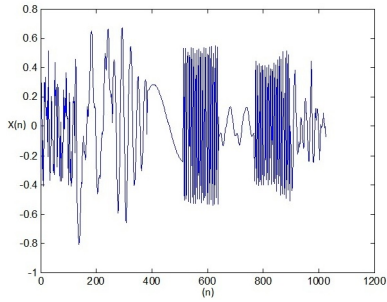


(c) Source 3

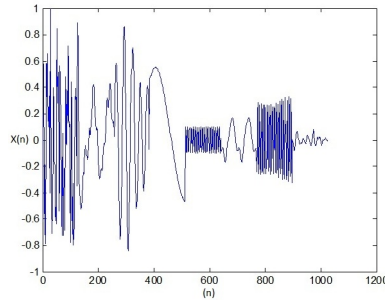


(d) Source 4

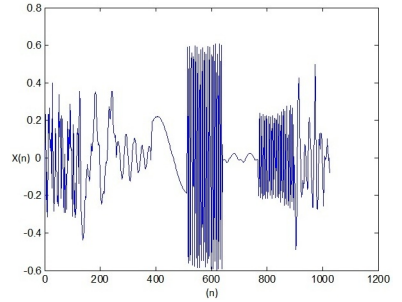
Figure 5.16: Original Sources



(a) Mixture 1



(b) Mixture 2



(c) Mixture 3

Figure 5.17: Mixtures

In order to show the recovered signals, the proposed scheme which uses maximum residue atom selection is applied and with 3982 iterations the scheme converges with an error of 10^{-12} . These recovered signals are shown in Fig. 5.18. Also the proposed maximum residue

atom selection which uses FFT implementation was employed and the algorithm converged with 3720 iterations.

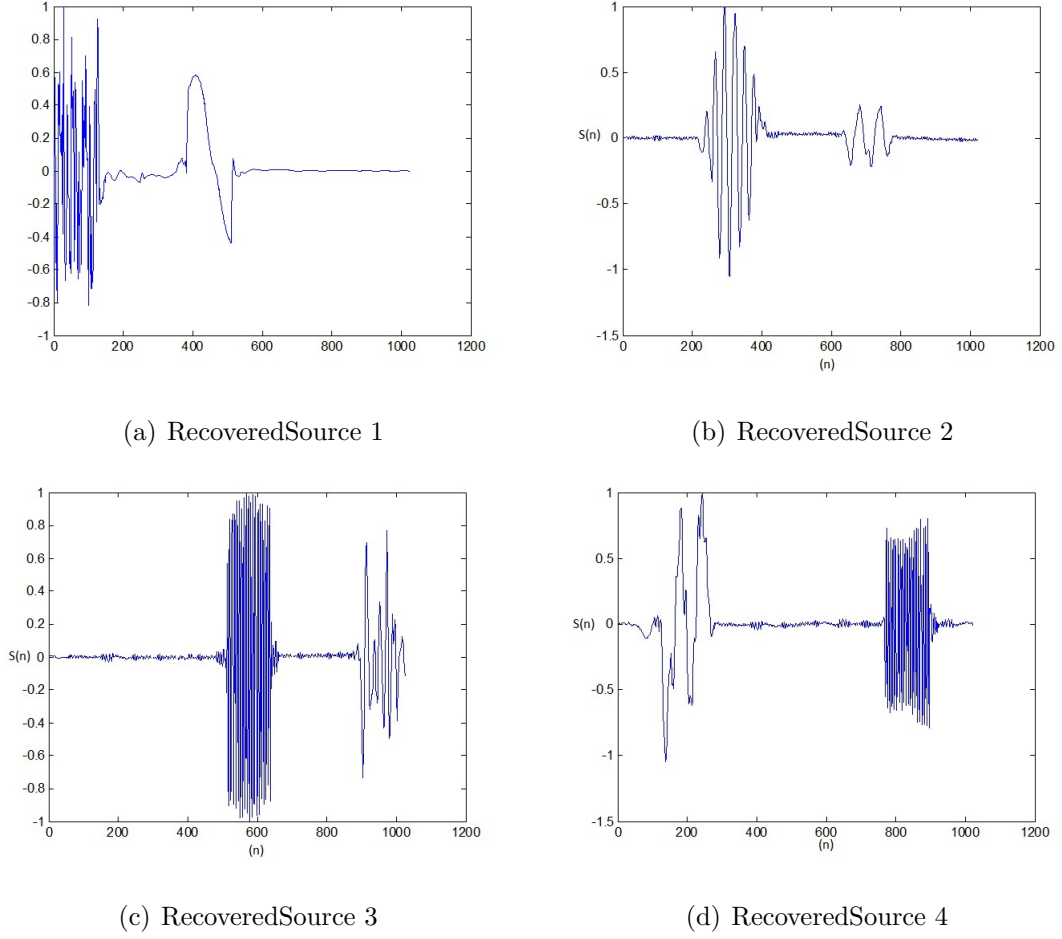


Figure 5.18: Recovered Sources by Maximum Residue atom Selection

Next, the conventional scheme which uses cumulative atom selection is compared with the two proposed algorithms. This scheme spent 6936 iterations to recover the same sources. Therefore, it can be observed that the proposed optimization criterion for atom selection improves the source recovery. The performance is measured by the minimum, maximum and average of the four recovered source's SIRs which are shown in Table 5.7.

Table 5.7: SIR Performance (dB) of Sparse Source Recovery (Example 2)

Method	$SIR_{min}(dB)$	$SIR_{max}(dB)$	Average of 4 SIRs (dB)
Former Cumulative Atom selection	27.42	42.17	38.23
Proposed Maximum Residue Atom selection	28.65	42.38	39.47
Proposed Maximum Residue Atom selection using FFT	28.52	42.97	39.28

Example 3) In the third example 1536 samples of 6 sparse sources show in Fig. 5.20 along with a 3×6 random mixing matrix produced 3 mixtures shown in Fig. 5.19. The recovered sources by the proposed multi-channel Matching Pursuit algorithm with maximum residue atom selection is chosen to recover the sources. The procedure converged successfully after 4462 iteration for an error of 10^{-12} . The recovered sources are shown in Fig. 5.21 and the SIR performance is shown in Table 5.8.

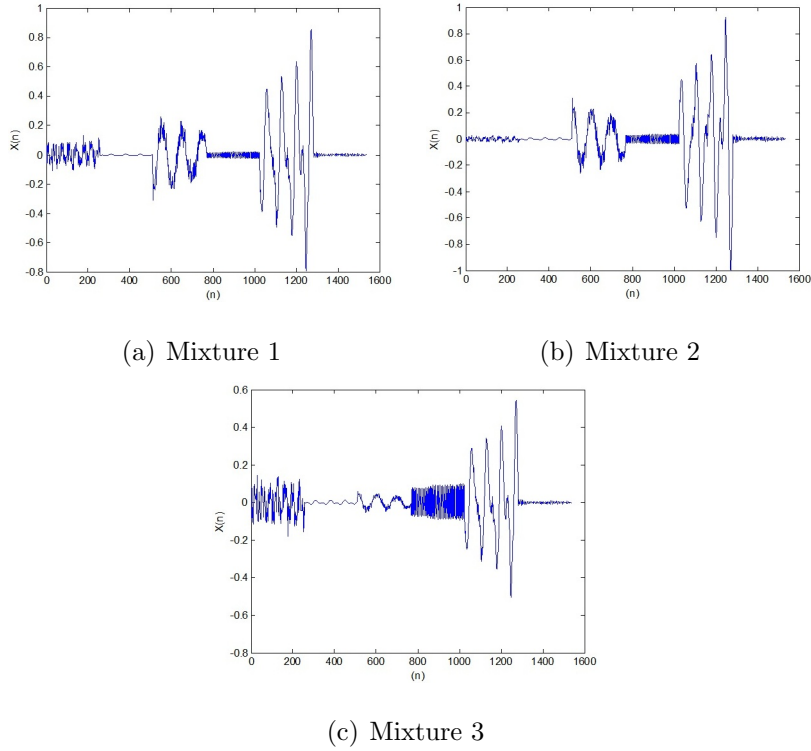
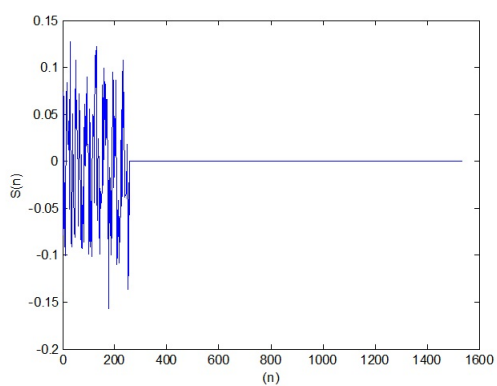
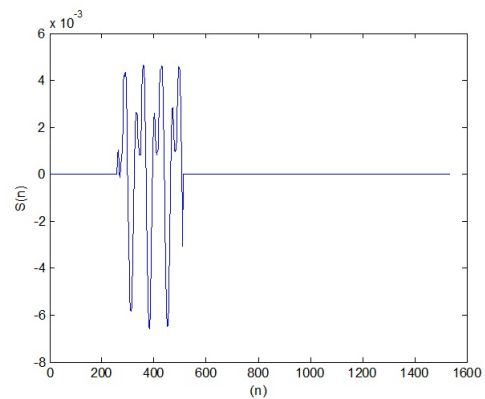


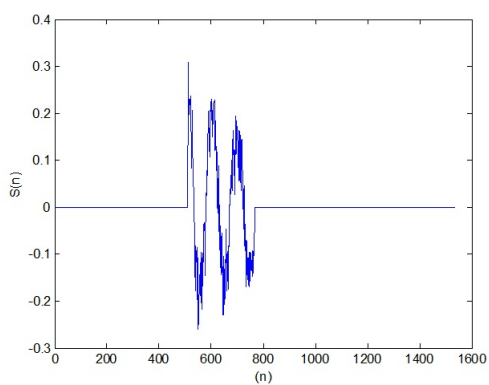
Figure 5.19: Mixtures



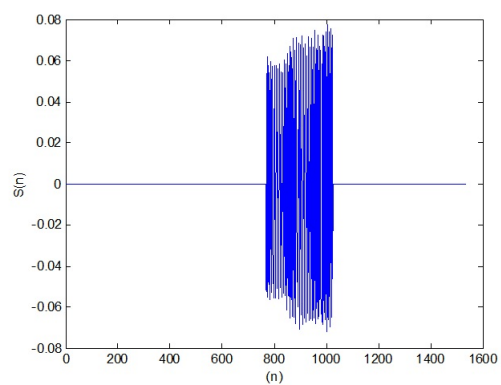
(a) Source 1



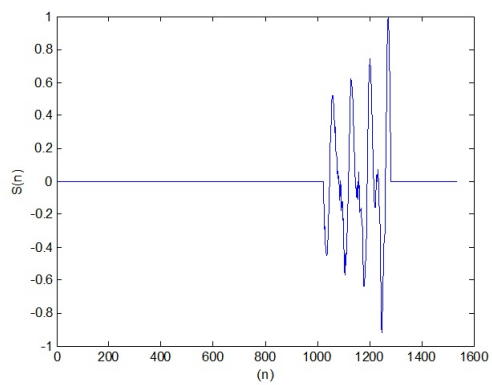
(b) Source 2



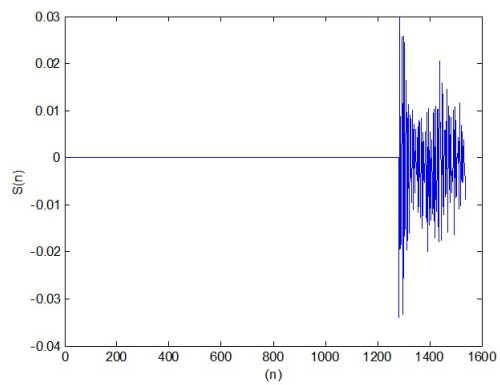
(c) Source 3



(d) Source 4



(e) Source 5



(f) Source 6

Figure 5.20: Original Sources

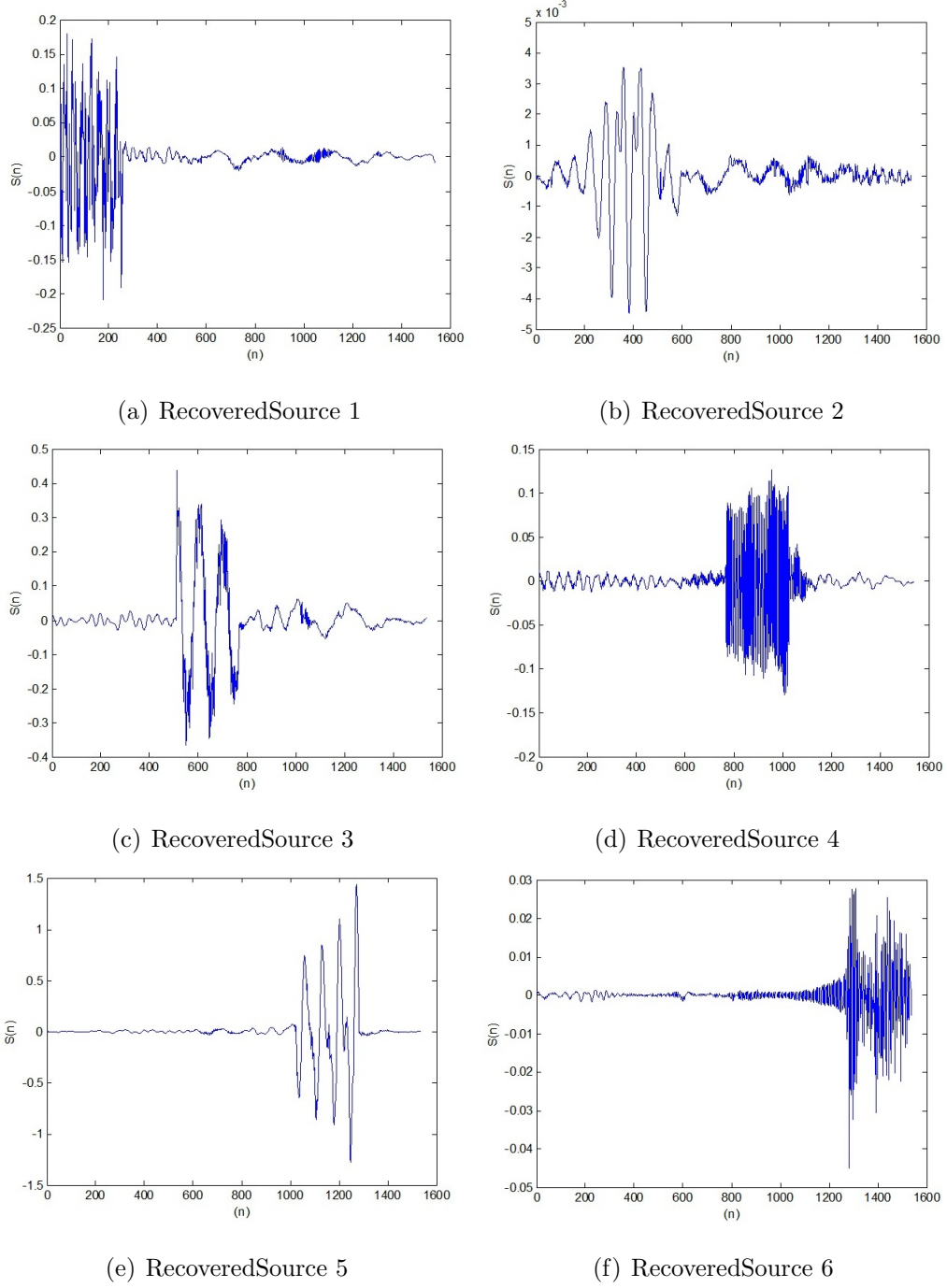


Figure 5.21: Recovered Sources by Maximum Residue atom Selection

Here the blind source recovery problem is using 3 mixtures to recover 6 sources, when $M < N$. The performance of the proposed algorithm is evaluated by minimum, maximum and average SIR values shown in Table 5.8.

Table 5.8: SIR Performance (dB) of Sparse Source Recovery (Example 3)

Method	$SIR_{min}(dB)$	$SIR_{max}(dB)$	Average of 4 SIRs (dB)
Proposed Maximum Residue Atom			
selection	18.38	41.17	31.96

5.3 Summary

Chapter 5 compares the performances of the conventional and proposed methods in different cases. In the first section the estimation of the mixing matrix which uses a clustering algorithm is mentioned. The conventional method overestimates the initial basis matrix and identifies the basis matrix based on the confidence gap. The proposed method which uses polar segmentation and least square curve fitting initializes the basis matrix to the correct dimension and as close as possible to the true direction. This method is able to estimate the mixing model accurately with less manipulations and computational weight. Also the performance of the generic model is shown to operate in the case where the mixtures are non-disjointly supported. In addition, the relationship of sparsity and transform domains is emphasized. In the second stage the sources are separated from the mixtures using the multi-channel Matching Pursuit algorithm. Different cases with mixtures are used and the conventional optimization criterion explained by cumulative atom selection is tested against the proposed optimization criterion. The proposed method uses the maximum residue atom selection and is able to perform much better. It is shown that the latter method is able to recover the source but with a larger number of iteration. However, the proposed method is able to recover the same signals with much less atoms. In addition, the sparse representation of sources through the transform domain is tested which shows improvement of sparsity. The examples are set in a manner which deals with most challenging sceneries that show the performance gap between the conventional method and the proposed approach.

Chapter 6

Conclusion

We have presented a solution to the Blind Source Separation problem which accurately estimates the mixing model in the first stage and recovers the sparse sources in the second stage. We have demonstrated that with appropriate initialization techniques the clustering algorithm helps to estimate the mixing matrix model very accurately. By studying the model of sparsity the alignment structures of the observed mixtures lead to an intuitive concept that was used to initialize the basis matrix. This concept was further developed by relevant tools which initialized the basis extremely close to the true hyperlines of the mixing model. We have introduced an efficient polar segmentation with a linear curve fitting tool to replace the overestimated randomized data for the initial basis. The proposed method reduced the overestimated dimensions greatly which lead to less computational weight and a much faster and simpler clustering algorithm. The overall performance of the proposed scheme for identifying the mixing model was extremely high, efficient and accurate. Further, it was shown that the model works well without the sparsity assumptions. However, further linear transforms had to be applied to the observed mixtures before the clustering algorithm was applied. The study of these linear transforms and their relationship to the sparsity model is the next step in this research. In addition, it was shown that applying efficient transforms such as STFT to the mixtures that do not possess the property of sparsity, will lead to a model which can gain full advantage of high sparsity once again. Therefore, this robust

generic model can be used in different applications with the appropriate settings.

After developing an efficient algorithm for identifying the mixing model, the problem of Blind Source Separation was targeted by a greedy multi-channel Matching Pursuit algorithm, which used a redundant real Gabor dictionary. The single channel case was investigated in detail in order to gain better knowledge of the implementation of real Gabor dictionary and Matching Pursuit algorithm. The same concept by Mallat [19] and Ferrando [10] were applied to represent a single channel signal in its most sparse form. The multi-channel case was analyzed by joint sparse representation of the mixtures. Further, the multi-channel optimization criterion mentioned by Lesage [18] was studied as a cumulative atom selection. This procedure represented the mixtures by coefficients that belonged to a single atom in each iteration. The joint sparse representations of mixtures were developed by one cumulative atom that was maximally correlated with all the observed mixtures. The mixtures in this model were decomposed at a much slower rate and with higher coefficients, which lead to a less sparse model. In addition, we have shown that the residue signals will decompose much faster when a maximum residue atom is selected. The experiments have shown that the same recovered signals obtained by the proposed methods have used less atoms and coefficients to recover the sources. This is very important where the sources can be represented in a much higher sparse form. Further, in order to gain better results the same proposed optimization criterion was used with FFT. We have demonstrated that the transform domain model increased sparsity and was also able to recover the sources. We have shown that fewer atoms are used for converging with a small error when the proposed scheme is used. The approach taken here to achieve an accurate solution for the BSS problem not only has made the existing algorithms more efficient but it has also introduced schemes that could be used in different applications such as image processing or compress sensing.

6.1 Future Work

We have introduced efficient algorithms for both the estimation of the mixing model and recovering the sparse sources using HCL clustering algorithm and a multi-channel Matching pursuit algorithm respectively. The models that these schemes were developed and tested based on work under the assumption of full sparse sources. However, when the sources are not fully sparse in the time domain more modifications need to take place before employing the proposed methods. This is due to the fact that the sparse sources create mixtures forming scatter lines, thereby distinguishing hyperlines which align along the columns of the mixing model. Therefore, the presented generic model works when the data is presented in its sparse domain. Further, we have presented non sparse data in time domain and shown that STFT model can transform the data into a much higher sparse frequency domain. It is suggested that further investigation needs to take place for other transform models such as STFT with different windows, Wavelet Cosine Transforms, Modified Discrete Cosine Transforms, as well as other models. Applying these transform models and their relationship with the sparsity model must be investigated in further research. Therefore, the proposed schemes will work more efficiently when the data are presented in their most sparse transform domain. Further, the implementation of an appropriate dictionary that contains multiple forms of atoms will be more efficient for source recovery. When the dictionary contains atoms represented with more functions there will then be more signals to choose from. These functions can include gabor, harmonic, chirp, constants, nyquist or dirac atoms that bring more variety for the Matching Pursuit algorithm. For future work, a Matching Pursuit algorithm that is used to partition the signal and represent each frame with the appropriate transform domain, windowing function, and best suited optimal atom will be suggested for an adaptive algorithm.

Bibliography

- [1] M. Aharon, M. Elad, and A. Bruckstein, *K-SVD and its Non-Negative Variant for Dictionary Design*. Proceedings of the SPIE conference wavelets, Vol. 5914, 2005.
- [2] A. Aissa, N. Trung, K. Meriam, A. Belouchrani, and Y. Grenier, *Underdetermined Blind Separation of Nondisjoint Sources in the Time-Frequency Domain*. IEEE Transactions on Signal Processing, Vol. 55 (2007), pp. 897-907.
- [3] S. Arberet, R. Gribonval, and F. Bimbot, *A robust method to count and locate audio sources in a multichannel underdetermined mixture*. IEEE Transactions on Signal Processing, Vol. 58, 2010.
- [4] A. Belouchrani and J.F. Cardoso, *Maximum likelihood source separation for discrete sources*. In Proceeding EUSIPCO, 1994, pp. 768-771.
- [5] P. Bofill, *Identifying Single Source Data for Mixing Matrix Estimation in Instantaneous Blind Source Separation*. Artificial Neural Networks, Vol 5163 (2008), pp. 759-767.
- [6] P. Bofill and M. Zibulevsky, *Underdetermined blind source separation using sparse representation*. Signal Processing, Vol.81 (2001), pp. 2353-2362.
- [7] S.S. Chen, D.L. Donoho, and M. A. Saunders, *Atomic Decomposition by Basis Pursuit*. Society for Industrial and Applied Mathematics, 2001.
- [8] S. Choi and A. Cichocki, *Adaptive blind separation of speech signals: Cocktail party problem*. ICSP 97, 1997, pp. 617-622.

- [9] D. L. Donoho, and M. Elad, *Maximal Sparsity Representation via l_1 Minimization*. Proceeding of the National Academy of Sciences, Vol. 100 (2003), pp. 2197-2202.
- [10] S.E. Ferrando, L.A. Kolasa, and N. Kovacevic, *A Flexible Implementation of Matching Pursuit for Gabor Dictionaries*. ACM Transactions Mathematics, Vol. 28 (2002), pp. 337-353.
- [11] P. Georgiev, and A. Cichocki, *Sparse Component Analysis of Over-complete Mixtures by Improved Basis Pursuit Method*. IEEE International Symposium on Circuits and Systems, Vol. 5 (2004), pp. v-37 - v-40.
- [12] P. G. Georgiev, F. J. Theis, and A. Cichocki, *Blind source separation and sparse component analysis for over-complete mixtures*. Proceeding International Conference Acoustic Speech Signal Processing, 2004, pp. 493-496.
- [13] R. Gribonval and M. Zibulevsky, *Sparse Component Analysis*. Handbook of Blind Source Separation: Independent Component Analysis and Applications, ELSEVIER 2010, pp.367-420
- [14] Z. He, A. Cichocki, J. Rosca, D. Erdogmus, J. Principe, and S. Haykin, *K-EVD Clustering and Its Applications to Sparse Component Analysis*. Independent Component Analysis and Blind Signal Separation, Vol. 3889 (2006), pp. 90-97.
- [15] A. Jourjine, S. Rickard, and O. Yilmaz, *Blind Separation of Disjoint Orthogonal Signals: Demixing N sources from 2 Mixtures*. Signal Processing, Vol. 81(2001), pp. 2353-2362.
- [16] C. Jutten and J. Herault, *Space or time adaptive signal processing by neural models*. In Proceeding AIP Conference on Neural Networks for Computing, 1986, pp. 206-211.
- [17] P. Kisilev, M. Zibulevsky, and Y. Zeevi, *A multiscale framework for blind separation of linearly mixed signals*. The Journal of Machine Learning Research, Vol. 4 (2003), pp. 1339-1363.
- [18] S. Lesage, S. Krstulovic, and R. Gribonval, *Under-Determined Source Separation: Comparison of Two Approaches Based on Sparse Decompositions*. Proceeding of the Inter-

- national Workshop on Independent Component Analysis and Blind Signal Separation (ICA), Vol. 4 (2006), pp. 633-640.
- [19] S. Mallat and Z. Zhang, *Matching pursuit with time-frequency dictionaries*. IEEE Transactions Signal Processing, Vol. 41 (1993), pp. 3397-3415.
 - [20] P.D. O’Grady and B.A. Pearlmutter, *Hard-Lost: Modified k-means for oriented lines*. In Proceeding Irish Signals and Systems Conference, 2004, pp. 247252.
 - [21] S. Raychaudhuri, J.M. Stuart, R.B. Altman *Principal Component Analysis to Summarize Micro-array experiments: Application to Sporulation Time Series*. Pac. Symp. Biocomput. 2000, pp. 455-466.
 - [22] S. Rickard and O. Yilmaz, *On the Approximate W-Disjoint Orthogonality of Speech*. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Vol. 1 (2002), pp. 529-532.
 - [23] F.J. Theis, *Multidimensional independent component analysis using characteristic functions*. Proc. EUSIPCO 2005, Antalya, Turkey, 2005
 - [24] F. J. Theis, C. G. Puntonet, and E. W. Lang, *Median-Based Clustering for Underdetermined Blind Signal Processing*. IEEE Transactions on Signal Processing, Vol. 13, 2006.
 - [25] L. Vielva, I. Santamaria, C. Pantaleon, J. Ibanez, D. Erdogmus, and J. C. Principe, *Estimation of the mixing matrix for underdetermined blind source separation using spectral estimation techniques*. Proceedings XI European Signal Processing Conference, Vol. 1 (2002), pp. 557-560.
 - [26] W. Wang, G. Liu, and W. Yu, *Novel Algorithm for Underdetermined Blind Source Separation Based On Matching Pursuit*. Ninth International Conference on Machine Learning and Cybernetics, Vol. 6 (2010), pp. 3107-3110.
 - [27] O. Yimaz and S. Rickard, *Blind Separation of speech mixtures via time-frequency masking*. IEEE Transactions on Signal Processing, Vol. 52 (2004), pp. 1830-1847.

Glossary

B

Basis Pursuit: Basis Pursuit is applied in cases where there is an underdetermined system of linear equations and must satisfy exactly the sparsest solution to the l_1 norm minimization.

Binary Masking: Thresholding procedure that sets the appropriate coefficient for the active source and masks the other sources at the same instance.

C

Clustering Algorithm: The task of assigning a set of data into groups (called clusters) so that the objects in the same cluster are more similar (in some sense or another) to each other than to those in other clusters. There are different criteria that could define which data gets assigned to which cluster.

Correlation: Mutual relation of two or more things, refers to any of a broad class of statistical relationships involving dependence.

Covariance Matrix: A covariance matrix is a matrix whose element in the i, j position is the covariance between the i^{th} and j^{th} elements of a random vector. Each element of the vector is a scalar random variable, either with a finite number of observed empirical values or with a finite or infinite number of potential values specified by a theoretical joint probability distribution of all the random variables.

D

Dimentionality Reduction: Dimension reduction is the process of reducing the number of random variables under consideration, and can be divided into feature selection and feature extraction.

E

Eigendecomposition: Eigendecomposition or sometimes spectral decomposition is the factorization of a matrix into a canonical form, whereby the matrix is represented in terms of its eigenvalues and eigenvectors.

Expectation Maximization Algorithm: Expectation maximization (EM) algorithm is an iterative method for finding maximum likelihood or maximum a posteriori (MAP) estimates of parameters in statistical models which depends on unobserved latent variables. The EM iteration alternates between performing an expectation (E) step that computes the expectation of the log-likelihood evaluated using the current estimate for the parameters, and a maximization (M) step, which computes parameters maximizing the expected log-likelihood found on the E step.

G

Gabor Dictionary: Dictionary with infinite Gabor functions which Cosine modulated and Gaussian windowed.

Gradient Algorithm: Gradient descent is a first-order optimization algorithm which finds a local minimum of a function using gradient descent, one takes steps proportional to the negative of the gradient of the function at the current point. If instead one takes steps proportional to the positive of the gradient, one approaches a local maximum of that function; the procedure is then known as gradient ascent.

I

Ill Conditioned Matrix: A matrix is ill-conditioned if the condition number is too large (and singular if it is infinite).

K

K-means Clustering: K-means clustering is a method of cluster analysis which aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean. This results into a partitioning of the data space into Voronoi cells.

M

Maximum Posteriori Estimation: Maximum posteriori probability (MAP) estimate is a mode of the posterior distribution that can be used to obtain a point estimate of an unobserved quantity on the basis of empirical data and employs an augmented optimization objective which incorporates a prior distribution over the quantity one wants to estimate.

Microarray data mining: Microarrays are one of the experimental molecular biology that allow monitoring the expression levels of tens of thousands of genes simultaneously.

Mixing Matrix: Mixing Matrix is a Matrix of size $M \times N$ which depicts the properties of the mixing model.

N

Non-Singularity: Nonsingular matrices are sometimes also called regular matrices. A square matrix is nonsingular iff its determinant is nonzero.

O

Optimization Criterion: Specifies the criterion for the optimization which could be maximization or minimization of certain function with or without constraints.

P

Parzen Windowing: Parzen-window density estimation is essentially a data-interpolation technique where given an instance of the random sample, \mathbf{x} , Parzen-windowing estimates the PDF $P(X)$ from which the sample was derived. It essentially superposes kernel functions placed at each observation or datum.

Principal Component Analysis: Principal component analysis (PCA) is a procedure that uses an orthogonal transformation to convert a set of observations of correlated variables into a set of linearly uncorrelated variables called principal components. The number of principal components is less than or equal to the number of original variables. This transformation has the largest possible variance (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it be orthogonal to the preceding components.

R

Redundant Dictionary: A dictionary which contains elementary functions which is infinite in size.

S

Segmentation: Partitioning a plane into different parts based on a given segmentation criterion.

Spurious: Not being what it purports to be; false which can be disregarded.

T

Transform Domain: It's a mathematical procedure which can be linear done in data that converts it from one domain (time, for example) to another (frequency).