Blind separation of speech mixtures

Vaninirappuputhenpurayil Gopalan Reju

2009

Vaninirappuputhenpurayil, G. R. (2009). Blind separation of speech mixtures. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/20784>

<https://doi.org/10.32657/10356/20784>

Blind Separation of Speech Mixtures

Vaninirappuputhenpurayil Gopalan Reju

School of Electrical & Electronic Engineering

**A thesis submitted to the Nanyang Technological University
in fulfilment of the requirement for the degree of
Doctor of Philosophy**

2009

Acknowledgments

I would like to express my deepest appreciation to my supervisor Professor Koh Soo Ngee who has given me an excellent opportunity to work with him; provided continual support, advice and guidance throughout my research. In addition, I would like to thank Associate Professor Soon Ing Yann for his kind support at all times and advice during this period. I would also like to extend my gratitude to Nanyang Technological University for the award of the research scholarship during my candidature.

I am grateful to my friends for their invaluable help and relief during tea breaks. I thank my mother and other family members for their support. Special thanks to my wife, Rashmi, for her love, understanding and support which enabled me to complete my thesis. Many thanks to my loving kids, Neha and Nitin who missed many of their weekends as I was in the laboratory working on my thesis.

Summary

This thesis addresses three well-known problems in blind source separation of speech signals, namely permutation problem in the frequency domain blind source separation (BSS), underdetermined instantaneous BSS and underdetermined convolutive BSS.

For solving the permutation problem in the frequency domain for determined mixtures, an algorithm named partial separation method is proposed. The algorithm uses a multistage approach. In the first stage, the mixed signals are partially (roughly) separated using a computationally efficient time domain method. In the second stage, the output from the time domain stage is further separated using the frequency domain BSS algorithm. For the frequency domain stage, the permutation problem is solved using the correlations between the magnitude envelopes of the DFT coefficients of the partially separated signals and those of the fully separated signals from the frequency domain stage. To solve the permutation problem for the case of underdetermined BSS, the k -means clustering approach is used. In this approach, the masks estimated for the separation of the sources by a Time-Frequency (TF) masking approach are clustered by using k -means clustering of small groups of nearby masks with overlap.

For the estimation of the mixing matrix in a two stage approach for the separation of the sources from their underdetermined mixtures, the algorithm first detects the single source points of the mixed signals in the TF domain. In this thesis it is shown that to check whether a point in the TF domain is a single source point, it only needs to compare the directions of the real and imaginary parts of the mixture sample vector at that point. If the directions are the same, then the point is a single source point. Subsequently, the mixing matrix can be estimated by clustering the detected single

source points. The proposed algorithm for the detection of the single source points is simpler than the algorithms previously reported.

Finally, for the separation of the sources from their underdetermined convolutive mixtures a TF masking approach is developed under the assumption that the sources are W -disjoint orthogonal in the TF domain. The main task in the TF masking approach is the estimation of the masks which are to be applied to the mixed signals in the TF domain. For the estimation of the masks the concept of angles in the complex vector space is used. Unlike the previously reported methods, the proposed algorithm does not require any estimation of the mixing matrix or the source position information for the mask estimation. The sample vectors of the mixture in the TF domain are clustered based on the Hermitian angles between the sample vectors and a randomly selected reference vector using the well known k -means or fuzzy c -means clustering algorithm. The membership functions so obtained from the clustering algorithm are directly used as the masks. An algorithm for automatic detection of the number of sources present in the mixtures is also proposed. The algorithm detects the number of sources by clustering the Hermitian angles calculated in a frequency bin. The validity of the proposed algorithm is evaluated for both collinear and non-collinear source configurations in a real room environment.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Scope of the Thesis	3
1.3	Contributions	5
2	Background of Blind Source Separation for Speech Signals	7
2.1	Brief introduction to BSS	7
2.2	Approaches for BSS of speech signals	10
2.2.1	Statistical independence	11
	Information theoretic	11
	Non-Gaussianity	12
	Nonlinear cross moments	15
2.2.2	Temporal structure of speech	19
2.2.3	Non-stationarity of speech	19
2.3	Convolutional BSS	23
2.4	Underdetermined BSS	31
3	Partial Separation Method for Solving the Permutation Problem	41
3.1	Introduction	41
3.2	Drawbacks of the existing methods	43
3.2.1	Direction Of Arrival approach	44
3.2.2	Correlation approach	46
3.2.3	Combined approach	46

3.3	Proposed method	47
3.3.1	Parallel configuration	48
3.3.2	Cascade configuration	51
3.4	Experimental results	53
3.4.1	Performance evaluation for collinear and non-collinear sources . .	56
3.4.2	Performance evaluation under different reverberation times	61
3.4.3	Performance evaluation using the measured real room impulse re- sponse	64
3.4.4	Robustness test for short speech utterances	70
3.4.5	Effect of combination order in cascade configuration	71
3.5	Summary	76
4	Mixing Matrix Estimation In Underdetermined Instantaneous Blind Source Separation	78
4.1	Introduction	78
4.2	Proposed method	84
4.2.1	Single-source-point identification	84
4.2.2	Mixing matrix estimation	90
4.3	Experimental Results	92
4.3.1	Comparison with other algorithms	99
	Determined case	99
	Underdetermined case	102
4.4	Summary	104
5	Underdetermined Convolutive Blind Source Separation via Time-Frequency Masking	106

5.1	Introduction	106
5.2	Proposed method	111
5.2.1	Basic idea	111
5.2.2	Clustering of mixture samples and mask estimation	115
	k -means clustering	116
	Fuzzy c -means clustering	117
5.2.3	Automatic detection of the number of sources	119
5.2.4	Permutation problem	121
5.2.5	Construction of the output signals	129
5.3	Experimental results	130
5.3.1	Experiments using real room impulse responses	131
5.3.2	Detection of the number of sources	133
5.3.3	Separation performance	134
5.3.4	Microphone spacing and selection of microphone output to apply mask.	137
5.3.5	Effect on the number of microphones	145
5.4	Summary	147
6	Conclusion and Recommendations	148
6.1	Conclusion	148
6.2	Recommendations for further research	151
Appendix		
A	Convolution Using Discrete Sine and Cosine Transforms	154
A.1	Introduction	154

A.2 Convolution in DTT domain	155
B Single Source Point Identification in DTT Domain	163
C Proof: Hermitian angle between two complex vectors will remain the same even if they are multiplied by complex scalars	169
Author's Publications	171
References	172

List of Figures

2.1	Illustration of blind source separation problem.	8
2.2	Diagrammatic representation of the convolutive mixing and unmixing process for the case of two sources and two sensors.	9
2.3	Flow of frequency domain blind source separation. In the frequency bins, signals corresponding to first, second and third separated sources are shown by dash-dot, dotted and dashed lines respectively.	26
2.4	Correlation between adjacent bins.	30
2.5	Solving permutation problem using dyadic sorting.	30
2.6	Overlapped Time-Frequency windows.	36
3.1	Directivity pattern of the two sources at two different frequencies. The actual directions of the sources are -30° and 20°	44
3.2	Block diagram for the proposed partial separation method for solving the permutation problem in frequency domain BSS (parallel configuration). . .	48
3.3	The two correlation matrices. (a) No column or row between the highest elements and hence the permutation problem can be solved with confidence (b) The highest elements are in the same column and hence the permutation problem cannot be solved with confidence.	51
3.4	Block diagram for the proposed method (cascade configuration).	52
3.5	Female speech utterances used for the experiments. F_n and M_n in Fig.3.6 together constitute one set, where $n \in \{1, 2, \dots, 10\}$. The audio files are available in the accompanying CD	54
3.6	Male speech utterances used for the experiments. F_n in Fig.3.5 and M_n together constitute one set, where $n \in \{1, 2, \dots, 10\}$. The audio files are available in the accompanying CD	55
3.7	The source-microphone configuration for the room impulse responses simulation.	56

3.8	Separation performance of the proposed method (reverberation time TR_{60} = 86ms): Partial Separation approach PS, Correlation approach C1 and the combined approaches PS+C1, PS+C2+C1 and PS+C2+Ha +C1	57
3.9	NRR at different frequencies for the 4 th set of speech utterances in Fig.3.8.	58
3.10	Room impulse response for different values of surface absorption: 0.3 (TR_{60} = 235ms), 0.5(TR_{60} = 130ms), 0.7(TR_{60} = 86ms) and 0.9(TR_{60} = 63ms). Only the impulse responses from Source 1 to Microphone 1 are shown.	63
3.11	Performance comparison of PS method alone with DOA method alone as a function of room surface absorption.	64
3.12	Performance comparison of PS method alone without confidence check with PS method after confidence check followed by the methods which utilizes the correlation between adjacent and harmonic bins, for parallel and cascade configurations. The DOA method after confidence check followed by correlation methods are also shown.	65
3.13	The source-microphone configuration for the measurement of real room impulse responses	65
3.14	Measured impulse responses of the room (Reverberation time TR_{60} = 187ms)	66
3.15	NRR for various algorithms using real room impulse responses. PS - Partial Separation method with confidence check, C1 - Correlation between the adjacent bins without confidence check, C2 - Correlation between adjacent bins with confidence check, Ha - Correlation between the harmonic components with confidence check, PS1 - Partial separation method alone without confidence check.	67
3.16	Waveform of the clean, mixed and separated signals. Permutation problem is solved by PS+C2+Ha+C1, NRR=14.68.	68
3.17	Separation result for 20 pairs of speech utterances with different methods for solving permutation problem. (The time domain stage is present in all the cases)	69

3.18	NRR for different lengths of speech utterances when the NRR of the partially separated signals used for solving permutation problem are of different levels.	70
3.19	Performance variation for various filter lengths of the time domain stage. The sampling frequency of the signals is 16kHz.	71
3.20	Performance of the frequency domain stage followed by time domain stage configuration for different lengths of filter taps as well as for different lengths of the data for learning.	72
3.21	Effect of permutation in the frequency bins, for time domain separation. NRR for the mixture due to the permutation of clean signals is indicated by “clean permuted” and that of the mixture due to the permutation of the mixed signals is indicated by “mixture permuted” . For example if multiple = 8, the permuted bins are 8, 16, 24,...,4096; similarly for other multiples.	73
4.1	Speech utterances used to plot the graph shown in Fig.4.2. Speech utterances s_n and s_{n+1} together constitute one pair, where $n \in \{1, 2, \dots, 15\}$. $s_1, s_2, \dots, s_{16}$ are obtained by concatenating the sentences taken from TIMIT database. The audio files are available in the accompanying CD	87
4.2	Percentage of samples which are below the magnitude of the difference between the ratios of the real and imaginary parts of the DFT coefficient of the signals.	88
4.3	Illustration of hierarchical clustering: (a) scatter diagram of the two dimensional data to be clustered (b) dendrogram generated for the data taking $1 - \cos(\theta) $ as the distance measure, where θ is the angle between the vectors constituted by the sample and the origin.	93

4.4	Scatter diagram of the mixtures taking samples from 40 frequency bins; $P = 2$; $Q = 6$; and $\Delta\theta = 0.8^\circ$ (a) all the DFT coefficients (b) samples at SSPs obtained by comparing the direction of $R\{\mathbf{X}(k, t)\}$ with that of $I\{\mathbf{X}(k, t)\}$ (c) samples at SSPs obtained after elimination of the outliers.	94
4.5	Mixing matrix estimation error before (dotted lines) and after (solid lines) elimination of the outliers from the initial estimated samples at SSPs for various values of $\Delta\theta$; $P = 2$ and $Q = 6$	96
4.6	Mixing matrix estimation error before and after re-clustering the outlier-free samples for various values of $\Delta\theta$; $P = 2$ and $Q = 6$	97
4.7	Comparison of mixing matrix estimation error when samples at SSPs from $R\{\mathbf{X}(k, t)\}$ alone is used with that when samples at SSPs from both $R\{\mathbf{X}(k, t)\}$ and $I\{\mathbf{X}(k, t)\}$ are used, for various values of $\Delta\theta$; $P = 2$ and $Q = 6$	98
4.8	Scatter diagram of the mixtures taking samples from 40 frequency bins; $P = 3$; $Q = 6$; and $\Delta\theta = 0.8^\circ$ (a) all the DFT coefficients (b) samples at SSPs after elimination of the outliers.	100
4.9	Comparison of NMSE on estimation of the mixing matrix using all the DFT coefficients in the TF plane with that using the estimated SSPs; $P = 3$; $Q = 6$; and $\Delta\theta = 0.8^\circ$	101
4.10	Comparison of the proposed algorithm with classical algorithms for determined case, $P = Q = 2$	103
4.11	Comparison of the proposed algorithm with that proposed in [1]	104
5.1	Masks generated by k -means clustering algorithm. (a) the plot of Hermitian angles $\Theta_H^{(k)}(t)$ (b) membership function (c) magnitude envelopes of the DFT coefficients in the k^{th} frequency bins of the signals picked up by the microphones (d) magnitude envelopes of the DFT coefficients in the k^{th} frequency bins of the separated signals.	116

5.2	Masks generated by FCM clustering algorithm. (a) the plot of Hermitian angles $\Theta_H^{(k)}(t)$ (b) membership function (c) magnitude envelopes of the DFT coefficients in the k^{th} frequency bins of the signals picked up by the microphones (d) magnitude envelopes of the DFT coefficients in the k^{th} frequency bins of the separated signals.	118
5.3	Correlation matrices (a) $\mathbf{C}_{\hat{S}_1\hat{S}_2}^{\text{mag}}$, correlation between the bin-wise magnitude envelopes of the clean signals picked up by the microphones (b) $\mathbf{C}_{\hat{S}_1\hat{S}_2}^{\text{Pratio}}$, correlation between the bin-wise power ratios of the clean signals picked up by the microphones (c) $\mathbf{C}_{Y_1Y_2}^{\text{Pratio}}$, Correlation between the bin-wise power ratios of the separated signals (d) $\mathbf{C}_{M_1M_2}^{\text{KM}}$, correlation between the masks estimated using k -means clustering algorithm; in both (c) and (d) the permutation problem is solved based on the correlation between the bin-wise power ratios of the separated signals and that of the clean signals picked up by the microphone on which masks are applied (e) $\mathbf{C}_{M_1M_2}^{\text{KM}}$, correlation between the masks estimated using k -means clustering (f) $\mathbf{C}_{M_1M_2}^{\text{FCM}}$, correlation between the masks estimated using fuzzy c -means clustering; in both (e) and (f) the permutation problem is solved by the proposed algorithm based on k -means clustering.	125
5.4	The source-microphone configuration for the measurement of real room impulse responses	131
5.5	Measured real room impulse response from source s_3 to the first microphone.	132
5.6	(a), (b) and (c) Mean histogram of the ‘estimated number of clusters (or sources)’ for the first 60 frequency bins. (d), (e) and (f) Total number of frequency bins used versus ‘estimated number of clusters (or sources)’; the estimation result will be more reliable with higher number of frequency bins used. In the figures, at some points, the ‘number of clusters estimated’ are not integers because it is the mean performance of 50 sets of speech utterances. All the source positions are with reference to Fig.5.4.	133

5.7	Waveform of clean speech (s_1 and s_3), individual signals picked up by the first microphone ($h_{11} * s_1$ and $h_{13} * s_3$), mixed signals (x_1 and x_2) and separated signals, separated by k -means (y_1^{KM} and y_3^{KM}) and FCM (y_1^{FCM} and y_3^{FCM}) algorithms, for the case of non-collinear sources. The notations are with reference to Fig.5.4. The audio files are available in the accompanying CD	138
5.8	Waveform of clean speech (s_1 and s_2), individual signals picked up by the first microphone ($h_{11} * s_1$ and $h_{12} * s_2$), mixed signals (x_1 and x_2) and separated signals, separated by k -means (y_1^{KM} and y_2^{KM}) and FCM (y_1^{FCM} and y_2^{FCM}) algorithms, for the case of collinear sources. The notations are with reference to Fig.5.4. The audio files are available in the accompanying CD	139
5.9	Waveform of individual signals picked up by the first microphone ($h_{11} * s_1$, $h_{12} * s_2$ and $h_{13} * s_3$), mixed signals (x_1 and x_2) and separated signals, separated by k -means algorithm (y_1^{KM} , y_2^{KM} and y_3^{KM}), for the underdetermined case. The notations are with reference to Fig.5.4. The audio files are available in the accompanying CD	140
5.10	Waveform of individual signals picked up by the first microphone ($h_{11} * s_1$, $h_{12} * s_2$ and $h_{13} * s_3$), mixed signals (x_1 and x_2) and separated signals, separated by FCM algorithm (y_1^{FCM} , y_2^{FCM} and y_3^{FCM}), for the underdetermined case. The notations are with reference to Fig.5.4. The audio files are available in the accompanying CD	141
5.11	The source-microphone configuration for the simulated room impulse responses	142

5.12	SDR/SIR/SAR versus index of the microphone output on which mask is applied, for different microphone spacings. Dotted lines are for the cases where the permutation problem is solved by finding the correlation between the bin-wise power ratios of the separated signals and that of clean signals picked up by the microphones. Solid lines are for the cases where the permutation problem is solved by the proposed method based on the k -means clustering algorithm. The mean input SDR, SIR and SAR are -0.09dB, 0dB and 20.82dB respectively.	143
5.13	Variation in angle between the column vectors $\mathbf{H}_q(k)$, $q = 1, 2$ versus microphone spacing. Dotted lines show the angles for different source combinations, as marked in the figure, and solid line shows the mean angle. . . .	144
5.14	Performance versus number of microphones. (a) output SDR (b) output SIR (c) output SAR (d) SDR improvement (e) SIR improvement (f) SAR improvement.	146
A.1	Generation of $\check{C}_1(k)$, $\check{S}_1(k)$, $\check{C}_2(k)$ and $\check{S}_2(k)$ from $C_1(k)$, $S_1(k)$, $C_2(k)$ and $S_2(k)$ respectively after decimation and symmetric or antisymmetric extension. The black squares represent the appended zeros to make the length of the sequences to $N + 1$ for element-wise operation.	155
B.1	dDCT2e and dDST2e coefficients of two speech utterances, s_1 and s_2	166
B.2	Performance comparison of the algorithm using $\mathbf{X}(k, t)$ and $\hat{\mathbf{X}}(k, t)$	168

List of Tables

2.1	The non-quadratic functions proposed in [2]	17
3.1	Experimental Conditions	53
3.2	NRR for the time domain method and DOA method for different microphone spacings (Room surface absorption = 0.5)	61
4.1	Algorithm for the detection of the single-source-points	90
4.2	Matlab code for the clustering algorithm	92
4.3	Experimental Conditions	95
5.1	Illustration of mask assignment to different clusters	124
5.2	Performance comparison of the proposed algorithm using k -means and FCM clustering.	135
5.3	Algorithm execution time	136
5.4	Experimental Conditions	137
A.1	Computational cost comparison	161

List of Symbols and Abbreviations

Π	Estimated permutation matrix
H	$= [h_1, \dots, h_Q]$, mixing matrix in time domain
R_{XY}	Correlation matrix calculated between X and Y
S	$= [S_1, \dots, S_Q]^T$, source signals in frequency domain
s	$= [s_1, \dots, s_Q]^T$, source signals in time domain
V	Whitening matrix
W	$= [w_1, \dots, w_P]^T$, unmixing matrix in time domain
X	$= [X_1, \dots, X_P]^T$, sensor outputs in frequency domain
x	$= [x_1, \dots, x_P]^T$, sensor outputs in time domain
Y	$= [Y_1, \dots, Y_Q]^T$, separated signals in frequency domain
y	$= [y_1, \dots, y_Q]^T$, separated signals in time domain
Γ	Actual permutation matrix
$\hat{H}$	Estimated H
$\mathcal{C}_i$	i^{th} cluster
μ	Adaptation step size
Ψ	$= [\psi_1, \dots, \psi_Q]^T$, column vector of the centroids while clustering $\Theta_H^{(k)}$ for cluster validation
ψ_i	Centroid of the i^{th} cluster while clustering $\Theta_H^{(k)}$ for cluster validation
θ_H	Hermitian angle
C_q	Centroid of the q^{th} cluster while clustering the masks
$I\{x\}$	Imaginary part of x
K	DFT length
k	Index of the frequency bin
$kurt(y)$	Kurtosis of y
L	Length of the unmixing filters
M_q	Mask for q^{th} source

P	Number of mixtures
Q	Number of sources
$R\{x\}$	Real Part of x
S_q	q^{th} source signal in frequency domain
s_q	q^{th} source signal in time domain
v_r^f	Magnitude envelope of the DFT coefficients of the r^{th} signal at frequency f
X_p	p^{th} sensor output in frequency domain
x_p	p^{th} sensor output in time domain
Y_q	q^{th} separated signal in frequency domain
y_q	q^{th} separated signal in time domain
$\mathbf{H}_q(k)$	q^{th} column of $\mathcal{H}(k)$
$\mathbf{h}_q$	q^{th} column of $\mathbf{H}$
$\mathbf{w}_p$	p^{th} column of $\mathbf{W}^T$
$\det \mathbf{W}$	Determinant of $\mathbf{W}$
$\mathcal{T}$	Total number of DFT coefficients in one frequency bin
$\mathcal{H}(k)$	$= [\mathbf{H}_1(k), \dots, \mathbf{H}_Q(k)]$, mixing matrix in frequency domain at k^{th} frequency bin
$\text{cor}(x, y)$	Correlation between x and y
bdiagA	Block diagonal operation on matrix $\mathbf{A}$ which will set the off-diagonal elements of $\mathbf{A}$ to zero
$\text{bf}\Theta_H^{(k)}$	$= [\theta_{H_1}, \dots, \theta_{H_T}]^T$, vector of Hermitian angles at k^{th} frequency bin
BSS	Blind Source Separation
DCT	Discrete Cosine Transform
DCT1e	DCT of type I even
DCT2e	DCT of type II even
DFT	Discrete Fourier Transform
DOA	Direction Of Arrival
DST	Discrete Sine Transform

DST1e	DST of type I even
DST2e	DST of type II even
DTT	Discrete Trigonometric Transform
DUET	Degenerate Unmixing Estimation Technique
ESPRIT	Estimation of Signal Parameters via Rotational Invariance Techniques
FCM	Fuzzy c -means
FIR	Finite Impulse Response
ICA	Independent Component Analysis
KL	Kullback-Leibler
KM	k -means
MSP	Multi Source Point
NMSE	Normalized Mean Square Error
NRR	Noise Reduction Rate
PS	Partial Separation
SAR	Signal to Artifact Ratio
SCA	Sparse Component Analysis
SDR	Signal to Distortion Ratio
SIR	Signal to Interference Ratio
SSP	Single Source Point
STFT	Short Time Fourier Transform
TF	Time-Frequency
TIFROM	Time Frequency Ratio Of Mixtures

Chapter 1

Introduction

1.1 Motivation

Blind source separation (BSS) is the technique for separating sources from their mixtures without any prior knowledge of either the sources or the mixing process. Since the introduction of the BSS concept in 1986 by J. Herault and C. Jutten [3], motivated by its wide range of applications from engineering to neuroscience, many algorithms have been developed for the separation of signals from their simple instantaneous mixtures to complex convolutive nonlinear time variant mixtures. However, there are still many challenges to overcome to make these algorithms suitable for real complex mixing environments. The main challenges are: unequal number of sources and sensors, noisy environment, orientation of sources and microphones, moving sources and nonlinear mixing. Numerous papers [4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18] are available which address these problems using different approaches.

The blind source separation algorithms for convolutive mixing can be broadly classified into two groups, namely, frequency domain and time domain methods. Compared to time domain methods, frequency domain methods are computationally efficient and separation performance is good when the mixing filters are longer. However, these methods generally have the disadvantage of inconsistent permutation in the Discrete Fourier Transform (DFT) bins of the separated signals. This is popularly known as permutation problem [16, 19, 20]. Many algorithms have been suggested for solving this problem; for example, methods utilizing the direction of arrival of the sources, correlation between the adjacent frequency bins and correlation

between harmonic bins. However, these methods are not robust. For example, a direction of arrival (DOA) based method fails under highly reverberant environments and when the sources are close in terms of the angle between them [21], assuming the microphone array center as the origin. For the case of the adjacent correlation method, as the permutation problem in one bin is solved based on that in the previous bins, a mistake in one bin may lead to complete misalignment in the following bins [21, 20]. The algorithms proposed in the literature to improve the robustness use the information from the separated signals alone. Instead of using the separated signals in the frequency bins alone, if another reference is used, which is independent of permutations in the separated signals, the robustness of the separation algorithm can further be improved [22, 21]. This motivated the development of a new algorithm called Partial Separation (PS) method for solving the permutation problem.

Signals from their instantaneous mixtures can be separated almost perfectly when the number of sources is smaller than or equal to the number of mixtures. However, when the number of mixtures is smaller than the number of sources, i.e., an underdetermined case, the problem is challenging. For underdetermined BSS sparsity of the signals in their Time-Frequency (TF) domain is commonly utilized [23, 24, 1, 25, 26]. The algorithms proposed in the literature for underdetermined BSS are generally computationally complex and the Single-Source-Points (SSPs) of the sources in their mixtures must have a number of other adjacent SSPs. To overcome these limitations a computationally very efficient algorithm is developed for the estimation of the SSPs and hence for the mixing matrix estimation. For the proposed algorithm, the SSPs need not be the adjacent points in the TF domain.

Separation of the sources from their underdetermined convolutive mixtures has very high practical importance as in many real environments the mixing is convolutive and the sources are more than the number of sensors. Motivated by the practical importance of the problem, many researchers proposed different algorithms to solve the problem [27, 28, 29, 25]. Because of the simplicity of the concept, the algorithm

based on TF masking attracted the attention of many researchers, where, under the assumption that the sources are W-disjoint in their TF domain, the estimated masks could be applied to the mixture in their TF domain and thus leading to the separation of the sources. In this technique, the main challenge is the estimation of the masks. The algorithms reported in the literature utilize source directions, estimated channel responses (assuming sparsity of the sources in the time domain) and estimated mixing vectors in the frequency domain which are obtained assuming that the total number of dominating sources is smaller than the number of microphones. This shows a real need for an algorithm for the estimation of the binary mask directly from the mixture without any assumption other than the W-disjoint orthogonality. Motivated by this, an algorithm for the estimation of the binary mask is developed utilizing the concept of angles in the complex vector space.

In many practical situations the number of sources present in the mixed signals may be unknown. This motivates the need for an algorithm for automatic detection of the number of sources before the mask estimation within the source separation. Motivated by this, a simple technique is incorporated into the above algorithm for the automatic detection of the number of sources.

1.2 Scope of the Thesis

This thesis mainly addresses two problems in BSS; the first one is the permutation problem and the second is BSS of underdetermined mixtures. The permutation problem is the major disadvantage of the frequency domain method for convolutive BSS. In this thesis, two algorithms are proposed to solve this problem, one is suitable only for the determined case and the other can be used for both determined and underdetermined cases. The first algorithm uses a partially separated signal, separated in the time domain, to solve the permutation problem. The algorithm is not only robust but will also improve the overall separation quality because of its cascading effect. The

performance of the second algorithm which is based on the clustering of the binary masks estimated for BSS, depends mainly on the quality of the separated signals, like many other correlation based methods; however, the algorithm is suitable for underdetermined cases also.

The separation of sources from their mixtures when the number of mixtures is smaller than the number of sources has high practical importance. In this thesis, two algorithms are proposed, one for instantaneous mixing and another for convolutive mixing. Both the algorithms are computationally very efficient and theoretically there is no limitation on the number of sources or sensors. In addition, an algorithm for the automatic detection of the number of sources is also incorporated into the algorithm for underdetermined convolutive BSS.

This thesis is organized as follows: in Chapter 2, the key BSS techniques are reviewed. The proposed algorithm based on partial separation method for solving the permutation problem, with some real room experimental results, is described in Chapter 3. In Chapter 4, it is shown that there is a simple method to detect the SSPs in the TF domain of their instantaneous mixtures and these SSPs can be used for the estimation of the mixing matrix. Then a clustering algorithm is proposed to cluster these SSPs and hence to estimate the mixing matrix. The superiority of the proposed algorithm is then demonstrated by comparing it with many classical algorithms for the determined case and a recently reported algorithm for the underdetermined case. The BSS algorithm based on the binary masking for separation of the sources from their overdetermined/underdetermined/determined convolutive mixtures using the concept of angles in complex vector space is described in Chapter 5. The experimental evaluation results of the algorithm for different source-microphone configurations, for both real and simulated room environments are provided. An algorithm for the automatic detection of the number of sources is also incorporated into the algorithm for underdetermined convolutive BSS. The proposed algorithm for solving the permutation problem for overdetermined/underdetermined/determined convolutive

mixtures by clustering the masks, estimated for BSS based on masking, is also described in the same chapter. Chapter 6 concludes the thesis with recommendations for future research.

1.3 Contributions

The major contributions of this dissertation are summarized as follows:

- 1) A robust algorithm for solving the permutation problem in frequency domain BSS of speech signals is proposed. The method uses correlation between the magnitude envelopes of the DFT coefficients in the corresponding frequency bins taken from two signals in the frequency domain. One of the signals is the partially separated signal, obtained using a time domain BSS method, and the other is the fully separated signal, obtained using a frequency domain technique. Unlike other correlation methods which utilize the inter-frequency correlation of the signals, the reliability of the proposed method is high as it utilizes the correlation with partially separated signals. The algorithm can optimally utilize its computational load for the time domain partial separation stage by cascading it with the frequency domain stage where the overall performance will be higher than that of the frequency domain stage alone.
- 2) An equation for circular convolution using discrete sine and cosine transforms is derived.
- 3) A simple and computationally efficient algorithm for single-source-point identification in the TF plane of the mixture signals for the estimation of mixing matrix in underdetermined BSS is developed. The algorithm can be used for the mixtures where the spectra of the sources overlap and the SSPs occur only at a small number of locations. Unlike in many other algorithms, these points need not be in adjacent locations in the TF plane.
- 4) An algorithm is proposed for separation of sources from their overdetermined/

underdetermined/ determined convolutive mixtures via TF masking. The advantages of the proposed algorithm for the design of the masks compared to the previously reported algorithms are: 1) it does not require the geometrical information or the channel parameters; 2) since the final data for clustering and hence for the estimation of the masks is a simple vector of the Hermitian angles, $\Theta_H^{(k)}$, irrespective of the number of microphones, the well-known clustering algorithms can be easily applied to $\Theta_H^{(k)}$ and the membership function so obtained can directly be used as the mask; and 3) the algorithm does not have the well-known scaling problem.

- 5) An algorithm for the automatic detection of the number of sources is proposed for the above convolutive underdetermined BSS.
- 6) Finally, an algorithm for solving the permutation problem by clustering the masks estimated for BSS based on TF masking is proposed. The advantages of the algorithm are: 1) the direct use of masks, estimated for source separation, reduces the additional computation of the power ratios of the separated signals; and 2) the well-known k -means algorithm could be used for clustering.

Chapter 2

Background of Blind Source Separation for Speech Signals

2.1 Brief introduction to BSS

The problem of extraction of the unobserved sources from their observed mixtures without any prior knowledge about the mixing process or the source signals is known as blind source separation (BSS). For example, consider the case where two people are talking and the mixed signals were picked up by two microphones placed at two different positions as illustrated in Fig.2.1. Here the objective of the BSS algorithm is to separate the speech signals from the mixed signals obtained from the microphone outputs without any prior knowledge about the source signals, positions of the microphones or the mixing process, i.e., separation of the sources from the mixtures blindly. In this particular example the data were acoustically mixed speech signals and the problem is commonly known as the cocktail party problem. It is not necessary for the signals to be confined to speech. It can be image or any other signals and the mixing can be instantaneous, convolutive, linear or nonlinear. A detailed discussion about the types of mixing are given in the following sections. Since J. Herault and C. Jutten [3] introduced the concept of BSS, many algorithms have been developed for the separation of signals from their simple instantaneous mixtures or their complex convolutive nonlinear time variant mixtures [30, 31].

The problem of BSS can be described as follows: suppose there are Q sources mixed convolutively to obtain P mixtures, then assuming that there is no additive

Fig. 2.1: Illustration of blind source separation problem.

noise, the P mixtures at the sensor output can be written as

$$\mathbf{x}(t) = \mathbf{H} * \mathbf{s}(t) \quad (2.1)$$

where $\mathbf{x}(t) = [x_1(t), x_2(t), \dots, x_P(t)]^T$ are the sensor outputs, $\mathbf{s}(t) = [s_1(t), s_2(t), \dots, s_Q(t)]^T$ are the sources and t is the time index. The superscript T represents the matrix transpose operator and $*$ denotes the convolution operator. The matrix $\mathbf{H}$ is the mixing matrix of order $P \times Q$ whose $(p, q)^{\text{th}}$ element, $h_{pq}(l)$, is the impulse response from the q^{th} source to the p^{th} sensor, so that $x_p(t) = \sum_{q=1}^Q \sum_{l=0}^{\infty} h_{pq}(l) s_q(t-l)$, for $p = 1, \dots, P$. Even though in a real acoustic environment the lengths of the mixing filters are extremely long (of the order of hundreds of milliseconds), the filter coefficients will decay to negligibly small values after a certain time duration, typically less than one second in a normal living room and few seconds in a big auditorium. Hence the length of the unmixing filter can be limited to a finite value.

The objective of BSS is to separate the sources $\mathbf{s}(t)$ from the mixtures $\mathbf{x}(t)$ in such a way that they are the scaled or filtered version of the original sources [8]. Hence, the

separated signals $y_r(t)$ ($r = 1, \dots, Q$) will be [20]

$$y_r(t) = \sum_l \alpha_r(l) s_{\Pi(r)}(t-l) \quad (2.2)$$

where $\alpha_r(l)$ is the l^{th} coefficient of the filter α_r , Π is a permutation matrix and $s_{\Pi(r)}(t)$ represents the r^{th} source with permutation, i.e., $[s_{\Pi(1)}, s_{\Pi(2)}, \dots, s_{\Pi(Q)}]^T = \Pi s$. This means that the order of the output signals (separated signals) need not be the same as that of the input signals (source signals) and in addition, the output signals will be the filtered version of the input signals. Since the mixing filters can be considered as finite length filters, the sources can be separated using the unmixing filters of finite length L and hence the output will be

$$\mathbf{y}(t) = \mathbf{W} * \mathbf{x}(t) \quad (2.3)$$

where $\mathbf{y}(t) = [y_1, y_2, \dots, y_Q]^T$ are the separated signals and $\mathbf{W}$ is a $Q \times P$ matrix of FIR filters with elements $w_{rp}(l)$, $r = 1, \dots, Q$, $p = 1, \dots, P$ and $l = 0, 1, \dots, L-1$. This mixing and separation model is depicted in Fig.2.2, for the case of two sources and two sensors.

When the length of the mixing filter is equal to one, the mixing is called instantaneous mixing and the separation of the sources from their mixtures could be easily achieved provided that the mixing matrix is full rank and the mixing is noise free [30, 31]. However there exists two types of ambiguities in blind source

Fig. 2.2: Diagrammatic representation of the convolutive mixing and unmixing process for the case of two sources and two sensors.

separation: namely, scaling and permutation. Since any multiplication of the sources by a constant will be absorbed by the mixing matrix, there is no way to separate the original signals from their mixtures with the same amplitudes, as long as any prior knowledge about the source amplitudes are not available. This problem is called the scaling problem [30, 31]. When the mixing is convolutive, this problem corresponds to an arbitrary filtering. Similarly, the order of the separated signals does not affect their independency. It is therefore not possible to always separate the sources from their mixtures in the order that they exist before mixing. This problem is known as the permutation problem [30, 31]. Hence the separated signals for instantaneous mixing can be written as

$$\mathbf{y} = \Gamma \mathbf{D} \mathbf{W} \mathbf{x} \quad (2.4)$$

where Γ is the permutation matrix (a matrix with only one element, which is 1, in any row or column) and $\mathbf{D}$ is a diagonal matrix whose $(i, i)^{\text{th}}$ element corresponds to the scaling factor of the i^{th} separated signal.

2.2 Approaches for BSS of speech signals

Most of the work during the initial period of BSS research was for instantaneous mixtures and subsequently for convolutive mixtures under the assumption that the mixing is overdetermined/determined (no. of sources $\leq$ no. of mixtures), which resulted in many excellent algorithms for the separation of overdetermined and determined mixtures. The BSS algorithm development under this class basically consists of two steps: *i*) selection of an appropriate cost function and *ii*) algorithm development for minimization or maximization of the selected cost function. The statistical properties such as robustness and consistency of the algorithms depend on the objective function selected. Whereas, the algorithmic properties such as convergence speed, numerical stability and memory requirement depend on the optimization algorithm selected. Ideally, these two categories of algorithmic properties are independent and

they can be selected according to the requirements.

To find the cost function, certain characteristics of the source signals are normally utilized. For speech signals, the typical characteristics utilized for finding the cost function are: 1) speech signals originating from different sources are independent, 2) over a short period of time, speech signals have unique temporal characteristics and 3) speech signals are non-stationary for a long time duration and quasi-stationary for a short time duration. These characteristics are briefly reviewed in the following sections.

2.2.1 Statistical independence

One of the most widely used assumptions in BSS is the same as that used in independent component analysis (ICA) [31], i.e., the original signals in the mixtures are independent. As speech signals from different sources may be assumed independent, the idea that is used in ICA can be used for BSS also. Normally higher order statistics [32, 33] are used for estimating the independent components from their mixtures where the separation is based on minimization of the fourth-order cumulants because of their covariant linear properties and relation to entropy [34]. In these types of algorithms, for successful separation, there should not be more than one Gaussian source as the Gaussian sources have zero higher order moments. A detailed discussion is given in [31]. The commonly used higher order (higher than second order) statistical methods are outlined below.

Information theoretic

The basic idea in the information theoretic approach is that the joint probability density of the independent sources will be the same as that of the product of their marginal distribution. i.e., $p(\mathbf{y}) = \prod_i p_i(y_i)$. This means that the sources in $\mathbf{y}$ do not carry any mutual information. Bell and Sejnowski's [10] well-known Infomax algorithm is based on this idea. The maximum likelihood method can also be used to

derive the Infomax algorithm [35]. This can also be done by maximizing the entropy in each separated source; the signals will be independent when the sum of the entropies of the signals is the same as that of the joint entropy. Hence, when the signals are independent there will not be any mutual information between them. This can also be interpreted as the Kullback-Leibler (KL) divergence between the densities of the signal. The signals can therefore be made independent by maximizing the KL divergence [36]. In information theoretic methods, the probability densities of the sources are assumed and approximated using some nonlinear functions except in a few cases [37], where the density is estimated from the available data, and the method is called non-parametric. Otherwise it is called parametric. In a parametric method, the performance of the algorithm depends on the selected nonlinearity and hence adaptive nonlinearities are also reported, instead of a fixed nonlinearity, for better accuracy [38]. The nonlinear function for complex valued ICA in the frequency domain is given in [39], which is based on polar coordinates and the algorithm is shown to have better convergence. The ideal form of nonlinearity is the cumulative distribution of the independent sources [40]. The commonly used nonlinearities for some typical source densities are listed in [30] (Table 6.1). In non-parametric methods, the estimation of the density requires more samples and the computational cost will generally be more compared to the parametric methods. However, recently a few methods are proposed which are computationally more efficient [5, 41]. The BSS problem can also be expressed in Bayesian formulation; the advantage of the method is that more sources than the sensors can be estimated [42, 43, 44, 45]. An Hidden Markov model can also be used for BSS [46, 4, 47]; however, because of its prior training requirement and higher computational cost, the method is not very popular.

Non-Gaussianity

According to the central limit theorem, when two or more signals are mixed together, the mixture will be more Gaussian than their individual components. In algorithms

which utilize the non-Gaussianity, the cost function of the algorithm will be optimized in such a way that the separated signals are as non-Gaussian as possible which in turn makes them as independent as possible. Hence this technique requires methods to measure the non-Gaussianity of signals after each iteration. The commonly used non-Gaussianity measures are kurtosis and negentropy.

Kurtosis: Kurtosis is the name given to the fourth order cumulant of a real random variable,

$$kurt(y) = E\{y^4\} - 3(E\{y^2\})^2 \quad (2.5)$$

If the random variable is normalized, i.e., variance $E\{y^2\} = 1$, then

$$kurt(y) = E\{y^4\} - 3 \quad (2.6)$$

Hence, kurtosis is simply the normalized version of 4th order moment. When the signal is Gaussian, its 4th moment is equal to $3(E\{y^2\})^2$ and hence kurtosis will be zero. Practically, for other signals kurtosis will be nonzero. Kurtosis can be negative or positive; for a super-Gaussian source it will be positive and for a sub-Gaussian source it will be negative. Hence by maximizing the absolute value of kurtosis of the separated signals with respect to the separating parameters (a simple unmixing matrix for instantaneous mixture, and unmixing filters for convolutive mixtures), the independent signals can be estimated. This can be achieved by a gradient method or by fast methods such as Newton's method. A fast and efficient version of the gradient method called natural gradient (NG) method is derived in [48]. An algorithm similar to the NG method is developed independently by J.-F. Cardoso and B. H. Laheld in [49]. An example of the algorithm based on Newton's method is the popular FastICA algorithm [2, 12]. The major problem with the natural gradient method lies in the tuning of its parameters. Because of the need to tune the parameters, it is difficult to optimize the performance of the natural gradient algorithm, especially when the

number of sources is higher. Also, the convergence speed is low. This problem is effectively solved by introducing the scaled natural gradient method [50], which is not very sensitive to the tuning parameters.

Negentropy: The main problem with the kurtosis based methods is that they are very sensitive to outliers. Hence, another measure for non-Gaussianity called negentropy is used, which is more robust than the kurtosis method. The negentropy of a signal is defined as follows. First, the entropy of a random signal y with a probability density function $p(y)$ is defined as

$$H(y) = - \int p(y) \log p(y) dy \quad (2.7)$$

Among the signals having zero mean and unit variance the entropy of a Gaussian signal is the highest, and for other signals it will be smaller. To obtain a value of zero for a Gaussian signal and a non-negative value for other signals, a differential entropy, $J(y) = H(y_{\text{gauss}}) - H(y)$, is defined [2], where y_{gauss} is a Gaussian random vector with the same correlation and covariance matrix as those of y . This differential entropy $J(y)$ is called the negentropy. The main problem with the negentropy is that the estimation using the definition is computationally expensive and hence approximate methods are generally used [2], in which by proper selection of some nonlinear function, the negentropy is calculated approximately. In [2] it is shown that the robustness and the separation performance of the ICA algorithms based on this approximate nonlinear function are far better than those based on kurtosis. In addition, by using this approximate nonlinear function, a fast algorithm namely FastICA [2, 12] is developed, which is one of the most widely accepted algorithms for ICA because of its convergence speed and absence of tuning parameters. Subsequently, the FastICA algorithm has been analyzed by many researchers [51, 52] and a more accurate algorithm called efficient FastICA (eFastICA) has been further proposed [53].

Nonlinear cross moments

The independent components are not correlated though the reverse need not always be true. However, it can be shown that by proper selection of the nonlinear functions, $f(\cdot)$ and $g(\cdot)$, the signals y_i and y_j can be made independent by nonlinear decorrelation, i.e., $E\{f(y_i) \cdot g(y_j)\} = 0$. Expanding the nonlinear function in the above equation using Taylor series, it can be seen that for proper selection of $f(\cdot)$ and $g(\cdot)$, higher order moments could be optimized and hence they can be used for blind source separation [54, 8, 55]. The relationship between the nonlinear principal component analysis (NPCA) and other well-known criteria for blind source separation are shown in [7]. Adaptive nonlinear PCA algorithms for BSS of un-whitened¹ observations were developed in [56], which are also suitable for online adaptation. A fast algorithm for NPCA is proposed in [57], which is faster than the LMS and RLS type NPCA algorithms developed previously both for online and offline adaptation.

It can be seen that the higher order statistical methods are inter connected in one way or another. For example, let $\mathbf{y} = \mathbf{W}\mathbf{x}$ be the separated signals from their mixtures $\mathbf{x}$. Then the mutual information between the output signals is given by [31]

$$I(y_1, y_2, \dots, y_Q) = \sum_i H(y_i) - H(\mathbf{x}) - \log |\det \mathbf{W}| \quad (2.8)$$

In the above equation, the last term in the right hand side (RHS) is constant and the second term is independent of $\mathbf{W}$. Hence the mutual information will be minimum when the entropy is minimum. The entropy is maximum for Gaussian signals and hence minimization of the entropy is equivalent to maximization of the non-Gaussianity of the estimated signals. As mentioned before, to obtain the measure of the non-Gaussianity which is zero for a Gaussian signal and nonnegative for other signals, a normalized version of the differential entropy called negentropy (which is

¹Whitening/sphering is a pre-processing technique normally applied to the zero mean observation vectors so as to make the observation vectors uncorrelated with unit variance

scale-invariant, i.e., multiplication of the variable by a constant does not change its negentropy) is generally used which is defined as $J(\mathbf{y}) = H(\mathbf{y}_{\text{gauss}}) - H(\mathbf{y})$, where $\mathbf{y}_{\text{gauss}}$ is a Gaussian random vector with the same correlation and covariance matrix as that of $\mathbf{y}$. In practice, the calculation of $H(\mathbf{y})$ will be difficult because it involves probability density estimation of $\mathbf{y}$, which is error prone and computationally complicated. Hence, approximation methods are normally used. After approximating the negentropy using polynomial density expansion, $J(\mathbf{y})$ can be written as [31]

$$J(\mathbf{y}) \approx \frac{1}{12}E\{\mathbf{y}^3\}^2 + \frac{1}{48}\text{kurt}(\mathbf{y})^2 \quad (2.9)$$

where $\mathbf{y}$ is assumed to have zero mean and unit variance. Assuming that the random variable $\mathbf{y}$ is of symmetric distribution, the first term in the above equation for approximate negentropy will be zero and the approximation will lead to kurtosis. Hence, maximization of negentropy or minimization of mutual information is equivalent to maximization of the square of its kurtosis or maximization of the absolute value of the kurtosis. However, as mentioned previously, maximization of the kurtosis will lead to the problem of robustness when outliers are present. To solve this problem, in [2], sophisticated approximations of negentropy were developed. The approach is to replace the higher-order cumulant approximations with expectations of general non-quadratic functions g_i , possibly more than two functions. Based on the maximum entropy principle, a simple approximation is developed in [58], which is of the form

$$J(y_i) \approx c[E\{g(y_i)\} - E\{g(v)\}]^2 \quad (2.10)$$

where g is the non-quadratic function, c is a constant and v is a Gaussian variable of zero mean and unit variance. Some choices of the function g are proposed in [2], and they are listed in Table 2.1.

To see the relation between cumulant and kurtosis, for a real zero mean signal, the fourth cumulant of y is given by $E\{y^4\} - 3[E\{y^2\}]^2$, which is the same as kurtosis.

Table 2.1: The non-quadratic functions proposed in [2]

Function $g(y)$	$g' = \frac{\partial g(y)}{\partial y}$	Remarks
$\frac{1}{a_1} \log \cosh(a_1 y)$ $1 \leq a_1 \leq 2$	$\tanh(a_1 y)$	General-purpose
$\frac{1}{a_2} \exp(-a_2 y^2/2)$ $a_2 \approx 1$	$y \exp(-a_2 y^2/2)$	Used for for highly super-Gaussian independent components or when robustness is very important
$\frac{1}{4} y^4$	y^3	For sub-Gaussian independent components without outliers

The relation between the Infomax method with the maximum likelihood method can also be established as follows [31, 59]. From the linear transformation $\mathbf{y} = \mathbf{W}\mathbf{x}$, $p_x(\mathbf{x})$ can be written as [31]

$$p_x(\mathbf{x}) = |\det \mathbf{W}| \prod_{i=1}^Q p_i(\mathbf{w}_i^T \mathbf{x}) \quad (2.11)$$

where $\mathbf{w}_i$ is the column vector such that $\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_i, \dots, \mathbf{w}_Q]^T$. The likelihood can then be written as the product of this density evaluated at T points, given by [31]

$$L(\mathbf{W}) = \prod_{t=1}^T \prod_{i=1}^Q p_i(\mathbf{w}_i^T \mathbf{x}(t)) |\det \mathbf{W}| \quad (2.12)$$

where $\mathbf{x}(1), \mathbf{x}(2), \dots, \mathbf{x}(T)$ are the T observations of $\mathbf{x}$. To simplify the algebraic manipulation, take the natural logarithm of the above relation. Then,

$$\log L(\mathbf{W}) = \sum_{t=1}^T \sum_{i=1}^Q \log p_i(\mathbf{w}_i^T \mathbf{x}(t)) + T \log |\det \mathbf{W}| \quad (2.13)$$

which can be written as

$$\frac{1}{T} \log L(\mathbf{W}) \approx E \left\{ \sum_{i=1}^Q \log p_i(\mathbf{w}_i^T \mathbf{x}) \right\} + \log |\det \mathbf{W}| \quad (2.14)$$

If the output is constrained to unit variance, the second term in the above equation will be a constant and it can be seen that the above relation is similar to that of Infomax.

It can be easily shown that, for proper section of the nonlinear function, the non-linear PCA algorithm will be equivalent to other ICA algorithms such as Infomax and kurtosis based [60]. For example, consider the cost function based on the nonlinear PCA for a whitened observation for which the mixing matrix is the transpose of the unmixing matrix, i.e., $\mathbf{H} = \mathbf{W}^T$,

$$\begin{aligned} J(\mathbf{W}) &= E \left\{ \left\| \mathbf{x} - \mathbf{W}^T g(\mathbf{W}\mathbf{x}) \right\|^2 \right\} \\ &= E \left\{ \mathbf{x}^T \mathbf{x} - g^T(\mathbf{y}) \mathbf{y} - \mathbf{y}^T g(\mathbf{y}) + g^T(\mathbf{y}) g(\mathbf{y}) \right\} \end{aligned} \quad (2.15)$$

where $\mathbf{y} = \mathbf{W}\mathbf{x}$. Now assume a simple nonlinearity $g(y) = y^3$. By substituting this nonlinearity on the above equation, it can be shown that

$$J(\mathbf{W}) = Q - 2E \left\{ \sum_{i=1}^Q y_i^4 \right\} + E \left\{ \sum_{i=1}^Q y_i^6 \right\} \quad (2.16)$$

If y is within ± 1 , $y_i^4 \gg y_i^6$, then

$$\begin{aligned} J(\mathbf{W}) &= Q - 2E \left\{ \sum_{i=1}^Q y_i^4 \right\} \\ &= Q - 6 - 2 \sum_{i=1}^Q kurt(y_i) \end{aligned} \quad (2.17)$$

where $kurt(y) = E \{y^4\} - 3$ is the kurtosis of y . Hence, minimization of $J(\mathbf{W})$ is the same as maximization of the last term in the above equation, which is the kurtosis. If the source kurtoses are negative, the nonlinearity to be used will be $g(y) = -y^3$.

2.2.2 Temporal structure of speech

If the signals have a unique temporal structure, second order statistics can be used. In this case the higher order statistics are not required and the signals to be separated need not be non-Gaussian. The algorithm based on second order statistics diagonalizes the output correlation simultaneously for different time lags. The main advantage of the second order statistics based system is that they are less sensitive to noise and outliers [61]. In [62], the temporal predictability of the signals is used for blind source separation.

2.2.3 Non-stationarity of speech

For stationary sources higher order statistics are necessary for successful separation, unless sources are temporarily correlated [11]. The non-stationarity of the signals, e.g., speech signals, can also be utilized for successful separation of the sources from their mixtures [63, 64, 65, 66, 67, 68, 19], where the signals are divided into blocks and a multichannel decorrelation matrix can then be computed for each block which will be different from block-to-block due to non-stationarity. The source separation is then achieved by simultaneous diagonalization of these correlation matrices. For better performance, the non-Gaussianity, non-whiteness and non-stationarity properties can be combined [69].

The algorithm used for minimization or maximization of the cost function can be the simple gradient method or its efficient versions such as natural gradient method or Newton's method. Even with a large number of available ICA algorithms, the most widely used algorithms are natural gradient or information maximization algorithm [10] and FastICA [2, 12]. Natural gradient ICA algorithms are generally used for online processing and FastICA for block processing. The fundamental equation for the natural gradient method is

$$\mathbf{W} \leftarrow \mathbf{W} + \mu [\mathbf{I} - f(\mathbf{y}) \mathbf{y}^T] \mathbf{W} \quad (2.18)$$

where μ is the adaptation step size, $\mathbf{W}$ is the unmixing matrix and $f(\mathbf{y}) = [f_1(y_1), \dots, f_P(y_Q)]^T$ is a nonlinear vector function. For the ideal case, the i^{th} component of $f(\mathbf{y})$ is [30]

$$f_i(y_i) = -\frac{d \log p_i(y_i)}{dy_i} \quad (2.19)$$

where $p_i(y_i)$ are the approximate models of the pdf of the source signals. Here, in a BSS problem, the probability distributions of the sources are not available and hence an approximation is usually used, depending on the type of the signals. For example, for super-Gaussian signals like speech, $f_i(y_i) = \tanh(y_i/\sigma_{y_i}^2)$ is a good nonlinear function [30]. A list of typical pdf of the sources and the corresponding nonlinear functions are given in [30]. When the number of sources is large, say more than 10, it is very difficult to find a constant step-size, μ , and an initialization matrix $\mathbf{W}(0)$. Moreover, the number of iterations required is also high. To overcome these limitations, S.C. Douglas et al. [50, 70] proposed an algorithm called scaled natural gradient algorithm. The algorithm is fast in convergence (typically less than 100 iterations) and independent of the initialization matrix $\mathbf{W}(0)$. Moreover, the algorithm is not very sensitive to the adaptation step-size, μ .

The FastICA algorithm [2, 12], also called the fixed point algorithm, is developed by minimizing the mutual information between the components of the separated signals. This is achieved by maximizing the sum of the approximate negentropies (equation (2.10)) of the separated signal components with respect to the unmixing matrix under the constraint of decorrelation between the separated signals, i.e.,

$$\text{maximize } \sum_{i=1}^Q J(\mathbf{w}_i) \text{ with respect to } \mathbf{w}_i, i = 1, \dots, Q \quad (2.20)$$

$$\text{under the constraints } E \{ (\mathbf{w}_k^T \mathbf{x}) (\mathbf{w}_j^T \mathbf{x}) \} = \delta_{jk}. \quad (2.21)$$

The basic fixed-point algorithm requires sphering or whitening the mixture signals, though a non-sphered version of the algorithm is also available. Whitening is the process of linear transformation of the observed data, say $\mathbf{z}$ using the transformation matrix $\mathbf{V}$ such that correlation matrix of $\mathbf{x} = \mathbf{V}\mathbf{z}$ is unity, i.e., $E\{\mathbf{x}\mathbf{x}^T\} = \mathbf{I}$. The whitening matrix $\mathbf{V}$ is calculated as

$$\mathbf{V} = \mathbf{E}\mathbf{D}^{-1/2}\mathbf{E}^T \quad (2.22)$$

where $\mathbf{E}$ is the orthogonal matrix of the eigen-vectors of $E\{\mathbf{x}\mathbf{x}^T\}$ and $\mathbf{D}$ is the diagonal matrix of its eigen-values. The basic one unit fixed-point algorithm, i.e., for the estimation of the unmixing matrix corresponding to one of the source signals, is given by

$$\mathbf{w}_{\text{temp}} = E\{\mathbf{x}g(\mathbf{w}^T\mathbf{x})\} - E\{g'(\mathbf{w}^T\mathbf{x})\}\mathbf{w} \quad (2.23)$$

$$\mathbf{w} = \mathbf{w}_{\text{temp}}/\|\mathbf{w}_{\text{temp}}\| \quad (2.24)$$

where $\mathbf{w}$ is normalized after every iteration to incorporate the constraint, $E\{\|\mathbf{w}^T\mathbf{x}\|^2\} = \|\mathbf{w}\|^2 = 1$, into the algorithm.

The algorithm (2.23) is based on Newton's method and hence the convergence is uncertain. The algorithm (2.23) is therefore modified by adding a step size to obtain a stabilized fixed point algorithm [2], i.e.,

$$\mathbf{w}_{\text{temp}} = \mathbf{w} - \mu [E\{\mathbf{x}g(\mathbf{w}^T\mathbf{x})\} - \beta\mathbf{w}] / [E\{g'(\mathbf{w}^T\mathbf{x})\} - \beta] \quad (2.25)$$

$$\mathbf{w} = \mathbf{w}_{\text{temp}}/\|\mathbf{w}_{\text{temp}}\| \quad (2.26)$$

where $\beta = E\{\mathbf{w}^T\mathbf{x}g(\mathbf{w}^T\mathbf{x})\}$ and μ is the step size parameter. In the case where the

mixture is not whitened, the algorithm in (2.23) is to be modified as

$$\mathbf{w}_{\text{temp}} = \mathbf{C}^{-1} E \{ \mathbf{x} g(\mathbf{w}^T \mathbf{x}) \} - E \{ g'(\mathbf{w}^T \mathbf{x}) \} \mathbf{w} \quad (2.27)$$

$$\mathbf{w} = \mathbf{w}_{\text{temp}} / \sqrt{\mathbf{w}_{\text{temp}}^T \mathbf{C} \mathbf{w}_{\text{temp}}} \quad (2.28)$$

where $\mathbf{C} = E \{ \mathbf{x} \mathbf{x}^T \}$ is the covariance matrix of the mixture. Similarly (2.25) may be modified as

$$\mathbf{w}_{\text{temp}} = \mathbf{w} - \mu [\mathbf{C}^{-1} E \{ \mathbf{x} g(\mathbf{w}^T \mathbf{x}) \} - \beta \mathbf{w}] / [E \{ g'(\mathbf{w}^T \mathbf{x}) \} - \beta] \quad (2.29)$$

$$\mathbf{w} = \mathbf{w}_{\text{temp}} / \sqrt{\mathbf{w}_{\text{temp}}^T \mathbf{C} \mathbf{w}_{\text{temp}}} \quad (2.30)$$

The whole unmixing matrix, $\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_P]^T$, can be estimated by the multi-unit constraint function (2.20) by applying the decorrelation constraint after every iteration. If $\mathbf{x}$ is the whitened mixture, in whitened space, the un-correlatedness of the separated signals is the same as orthogonalization of the unmixing matrix as $E \{ (\mathbf{w}_i^T \mathbf{x}) (\mathbf{w}_j^T \mathbf{x})^T \} = \mathbf{w}_i^T \mathbf{x} \mathbf{x}^T \mathbf{w}_j = \mathbf{w}_i^T \mathbf{w}_j$. Hence for estimating several independent sources, the one-unit algorithm is to be run several times (possibly using several units) with the unmixing vectors $\mathbf{w}_1, \dots, \mathbf{w}_Q$. After every iteration, the unmixing matrix $\mathbf{W}$ is to be orthogonalized to prevent the algorithm from converging to the same maximum.

The orthogonalization of the separated signals can be done by orthogonalizing the unmixing matrix, which can be achieved in different ways. In the deflation approach, where each independent component is separated one-by-one, the orthogonalization is achieved as follows. After estimation of the q unmixing vectors, $\mathbf{w}_1, \dots, \mathbf{w}_q$, during the estimation of the $(q+1)^{\text{th}}$ unmixing vector $\mathbf{w}_{q+1}$, after every iteration, the projections $(\mathbf{w}_{q+1}^T \mathbf{C} \mathbf{w}_j) \mathbf{w}_j, j = 1, \dots, q$ of the previously estimated unmixing vectors are subtracted

from $\mathbf{w}_{q+1}$ which is then normalized, i.e.,

$$\begin{aligned} \text{Step 1. } \mathbf{w}_{q+1} &\leftarrow \mathbf{w}_{q+1} - \sum_{j=1}^q (\mathbf{w}_{q+1}^T \mathbf{C} \mathbf{w}_j) \mathbf{w}_j \\ \text{Step 2. } \mathbf{w}_{q+1} &\leftarrow \mathbf{w}_{q+1} / \sqrt{\mathbf{w}_{q+1}^T \mathbf{C} \mathbf{w}_{q+1}} \end{aligned}$$

In cases where all the sources are to be separated without any special privilege for any of the sources, symmetric decorrelation of the unmixing matrix is required. This can be achieved in two ways. In the first method,

$$\mathbf{W} \leftarrow (\mathbf{W} \mathbf{C} \mathbf{W}^T)^{-1/2} \mathbf{W} \quad (2.31)$$

where $(\mathbf{W} \mathbf{C} \mathbf{W}^T)^{-1/2}$ can be obtained by eigen decomposition of $(\mathbf{W} \mathbf{C} \mathbf{W}^T)^T = \mathbf{E} \mathbf{D} \mathbf{E}^T$ as $(\mathbf{W} \mathbf{C} \mathbf{W}^T)^{-1/2} = \mathbf{E} \mathbf{D}^{-1/2} \mathbf{E}^T$. The second method is an iterative method as given below:

$$\begin{aligned} \text{Step 1. } \mathbf{W} &\leftarrow \mathbf{W} / \sqrt{\|\mathbf{W} \mathbf{C} \mathbf{W}^T\|} \\ \text{Step 2. } \mathbf{W} &\leftarrow \frac{3}{2} \mathbf{W} - \frac{1}{2} \mathbf{W} \mathbf{C} \mathbf{W}^T \mathbf{W} \\ &\text{Repeat Step 2 until convergence.} \end{aligned}$$

2.3 Convolutional BSS

The BSS task is easy when the mixing is linear instantaneous and the mixing matrix is full rank, where the mixing and hence the unmixing matrices will be a simple two dimensional matrices. However, in practice, instantaneous mixing rarely occurs. In practical environments mixing is generally convolutional, where the simple mixing matrices in the case of instantaneous mixing will be replaced by a matrix of mixing filters as shown in Fig.2.2. Hence to separate the signals, it is necessary to estimate the unmixing filters by extending any one of the principles mentioned above. There are two main approaches for the BSS of convolutional mixtures. The first one is the time domain approach [69, 71, 72, 73, 74, 75, 15, 50, 70] and the second is the

frequency domain approach [16, 76, 77, 78, 79]. In the frequency domain approach, the fact that circular convolution in the time domain is equivalent to multiplication in the frequency domain, i.e., $y = w * x \Leftrightarrow \mathbf{Y}(f) = \mathbf{W}(f)\mathbf{X}(f)$ is utilized so that the ICA algorithms developed for the complex instantaneous ICA [31, 30, 12] can be directly applied to each frequency bin. Since the complex instantaneous ICA algorithm is used in the following chapters for convolutive BSS in the frequency domain, the algorithm is briefly explained below.

The fixed point FastICA algorithm for complex numbers is developed by maximizing the contrast function

$$J_G(\mathbf{w}) = E \left\{ G \left(|\mathbf{w}^H \mathbf{x}|^2 \right) \right\} \quad (2.32)$$

This contrast function embeds the higher order statistics into the algorithm by the use of nonlinear function G . Here, $\mathbf{w}$ is the complex weight vector, which is estimated by the optimization of the following problem:

$$\text{minimize } \sum_{q=1}^Q J_G(\mathbf{w}_q) \quad \text{w.r.t } \mathbf{w}_q \quad (2.33)$$

$$\text{subject to the constraint } E \left\{ (\mathbf{w}_k^H \mathbf{x}) (\mathbf{w}_j^H \mathbf{x})^* \right\} = \delta_{jk} \quad (2.34)$$

To make the contrast function robust against outliers, a function which grows slowly as its argument increases is preferred. In [12], the following different nonlinear functions are proposed:

$$G_1(y) = \sqrt{a_1 + y}, \quad g_1(y) = \frac{1}{2\sqrt{a_1 + y}} \quad (2.35)$$

$$G_2(y) = \log(a_2 + y), \quad g_2(y) = \frac{1}{a_2 + y} \quad (2.36)$$

$$G_3(y) = \frac{1}{2}y^2, \quad g_3(y) = y \quad (2.37)$$

where g_n is the first derivative of G_n , and a_1 and a_2 are two arbitrary constants. With the notations defined above, the one unit fixed point ICA algorithm for complex data which searches for the extreme of the cost function $E \left\{ G \left(|\mathbf{w}^H \mathbf{x}|^2 \right) \right\}$ is given by

$$\mathbf{w}_{\text{temp}} = E \left\{ \mathbf{x} (\mathbf{w}^H \mathbf{x})^* g \left(|\mathbf{w}^H \mathbf{x}|^2 \right) \right\} - E \left\{ g \left(|\mathbf{w}^H \mathbf{x}|^2 \right) + |\mathbf{w}^H \mathbf{x}|^2 g' \left(|\mathbf{w}^H \mathbf{x}|^2 \right) \right\} \mathbf{w} \quad (2.38)$$

$$\mathbf{w} = \frac{\mathbf{w}_{\text{temp}}}{\|\mathbf{w}_{\text{temp}}\|} \quad (2.39)$$

The one unit algorithm can be extended for the estimation of all the independent components. To prevent the estimation of the same components which have already been estimated, outputs $\mathbf{w}_1^H \mathbf{x}, \dots, \mathbf{w}_q^H \mathbf{x}$ are to be decorrelated after every iteration. This can be achieved by using the Gram-Schmidt-like decorrelation approach which is explained in the previous section, i.e., after estimation of q vectors $\mathbf{w}_1, \dots, \mathbf{w}_q$, while running the $(q+1)^{\text{th}}$ one unit algorithm for $\mathbf{w}_{q+1}$, subtract the projections of the previously estimated q vectors from $\mathbf{w}_{q+1}$, after every iteration, i.e.,

$$\mathbf{w}_{q+1} \leftarrow \mathbf{w}_{q+1} - \sum_{j=1}^q \mathbf{w}_j \mathbf{w}_j^H \mathbf{w}_{q+1} \quad (2.40)$$

$$\mathbf{w}_{q+1} \leftarrow \frac{\mathbf{w}_{q+1}}{\|\mathbf{w}_{q+1}\|} \quad (2.41)$$

In situations where all the components are to be estimated simultaneously, this can be accomplished by symmetric decorrelation, i.e.,

$$\mathbf{W} \leftarrow \mathbf{W} (\mathbf{W}^H \mathbf{W})^{1/2} \quad (2.42)$$

where $\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_Q]$ is the unmixing matrix.

In addition to the time domain and frequency domain algorithms, there is a combination of these two, where the computational complexity of the time domain method

is reduced by implementing the time domain convolution operation in the frequency domain [80, 81].

For time domain separation, the second order statistical methods can be successfully utilized for blind source separation of convolutive mixtures [82, 83, 17], where the non-stationarity and non-whiteness properties of speech signals are utilized. Reference [82] is the generalization of the BSS method based on the second order statistics and hence it shows that the algorithms reported in [84, 85, 86] are the special cases. An efficient version of the algorithm proposed in [82] is available in [87]. A non-parametric BSS method is proposed in [88], which is based on the mutual information minimization method proposed in [89].

In Chapter 3, for partial separation of the mixture, the efficient version [87] of the algorithm proposed in [82] is used. The computational cost of the algorithm is very low but at the expense of separation performance. The algorithm is briefly explained below.

Fig. 2.3: Flow of frequency domain blind source separation. In the frequency bins, signals corresponding to first, second and third separated sources are shown by dash-dot, dotted and dashed lines respectively.

The algorithm is based on the second order statistics which utilize the non-stationarity and non-whiteness properties of the speech signals. The cost function for the algorithm as defined in [90, 81, 82, 87] is

$$J(m) = \sum_{i=0}^m \beta(i, m) \{ \log(\det(\mathbf{bdiag}(\mathbf{Y}^T(i) \mathbf{Y}(i)))) - \log(\det(\mathbf{Y}^T(i) \mathbf{Y}(i))) \} \quad (2.43)$$

where β is the weighing function, m is the block index (the speech signal is divided into different blocks) and “bdiag” represents the block diagonal operation. $\mathbf{Y}(m) = [\mathbf{Y}_1(m), \dots, \mathbf{Y}_Q(m)]$ is the block output signal matrix, the columns of $\mathbf{Y}_r(m)$ contain blocks of the r^{th} output signal, $y_r(t)$, of length N samples and each column is delayed by one sample, i.e.,

$$\mathbf{Y}_r(m) = \begin{bmatrix} y_r(mL_t) & \cdots & y_r(mL_t - L_t + 1) \\ y_r(mL_t + 1) & \cdots & y_r(mL_t - L_t + 2) \\ \vdots & \ddots & \vdots \\ y_r(mL_t + N - 1) & \cdots & y_r(mL_t - L_t + N) \end{bmatrix} \quad (2.44)$$

where N is the length of the output block and L_t is the length of the time domain unmixing filter. Hence the size of the matrix $\mathbf{Y}(m)$ will be of the order of $N \times L_t Q$. The natural gradient of (2.43) with respect to the unmixing filter in the time domain, $\mathbf{W}_t$, gives [3, 35]:

$$\nabla_{\mathbf{W}_t}^{NG} J(m) = 2 \sum_{i=0}^m \beta(i, m) \mathbf{W}_t(i) \{ \mathbf{R}_{\mathbf{Y}\mathbf{Y}}(i) - \mathbf{bdiag}(\mathbf{R}_{\mathbf{Y}\mathbf{Y}}(i)) \} \mathbf{bdiag}^{-1}(\mathbf{R}_{\mathbf{Y}\mathbf{Y}}(i)) \quad (2.45)$$

where the $L_t Q \times L_t Q$ correlation matrix $\mathbf{R}_{\mathbf{Y}\mathbf{Y}}$ consists of the correlation matrices $\mathbf{R}_{\mathbf{y}_p \mathbf{y}_q}(m) = \mathbf{Y}_p^T(m) \mathbf{Y}_q(m)$ of size $L_t \times L_t$. When the output signals are mutually independent $\mathbf{R}_{\mathbf{y}_p \mathbf{y}_q}(m) = 0$, for $p \neq q$. Since the temporal correlation of a speech signal is also taken into consideration, the algorithm is free from the whitening problem. The direct

computation of (2.45) is complex but an efficient and approximate version of the algorithm, which is given in [87], is very fast.

The main disadvantage of the time domain method is that it is computationally intensive and convergence speed is low for long filters, whereas the frequency domain method is computationally efficient as the convolution in the time domain becomes simple element-wise multiplication in the frequency domain. Hence the complex valued ICA algorithm can be applied to each DFT bin. Another advantage of the frequency domain method compared to the time domain method is that, in the frequency domain the coefficients can be more super-Gaussian than the time domain speech samples. For source separation the higher the super-Gaussianity of the signals, the better will be the performance [78]. However, the method has the disadvantage of inconsistent permutation in the DFT bins after separation by ICA. This is the popularly known permutation problem of frequency-domain BSS, which is depicted in Fig.2.3. The algorithms used for solving the permutation problem will align the permutation in the DFT bins so that the separated signals in the time domain will contain frequency components from the same source signals.

Various methods have been proposed to solve the permutation problem in frequency-domain BSS. L. Parra et al. [15] suggested a method which constrains the length of the filter, but this is not suitable for real acoustic environments where the length of the separation filter is of the order of thousands. Smoothing of the separation matrix is another method [16, 76]. The property that the adjacent bands are highly correlated for speech signals is utilized in [63] to solve the permutation problem. In a correlation based method, the magnitude envelope of the DFT coefficient in each frequency bin is first calculated. Then the correlations, r_{pq} , between the magnitude envelopes of the k^{th} and the $(k+1)^{\text{th}}$ bins are calculated as shown in Fig.2.4. The permutation between the sources in the $(k+1)^{\text{th}}$ bin are then solved in such a way that the sum of the correlation between the magnitude envelopes, $|y_i(k, t)|$, in the k^{th} and the $(k+1)^{\text{th}}$ bins of the same sources is maximum, i.e., for a two source case shown in Fig 2.4, if

$r_{11} + r_{22} < r_{12} + r_{21}$, the DFT coefficients in the $(k + 1)^{\text{th}}$ bins are interchanged between the source; otherwise, the same permutation is kept. Later in [91], it is shown that instead of magnitude envelopes of the DFT coefficients, by using the power ratios between the signals, the performance can be improved. The power ratio of the i^{th} separated signal in the k^{th} bin and t^{th} time frame is given by $P_{\text{ratio}_{Y_i}} = \frac{\|Y_i(k,t)\|^2}{\sum_{j=1}^Q \|Y_j(k,t)\|^2}$.

The main disadvantage of the correlation method is its lack of robustness. Since the permutation in one bin is solved based on the permutation in the previous bins, if the method fails in one bin, the remaining bins will follow the same pattern and the method will fail completely. In [92] the reliability of the correlation method is improved with the dyadic permutation sorting algorithm, as explained below.

For the purpose of explanation, assume that there are only 8 frequency bins, $k_0, k_1, \dots, k_7$. To solve the permutation problem, (see Fig.2.5), in the first level (Level 0), solve the permutation between the bins k_0 and k_1 , k_2 and k_3 , k_4 and k_5 and finally between k_6 and k_7 . After solving permutation in Level 0, in Level 1, the correlation is calculated between the mean magnitude envelopes $\frac{1}{2}(|Y(k_0, t)| + |Y(k_1, t)|)$ and $\frac{1}{2}(|Y(k_2, t)| + |Y(k_3, t)|)$. Using the calculated correlations, solve the permutation between the groups $\{k_0, k_1\}$ and $\{k_2, k_3\}$ such that the permutation for the bins k_0, k_1, k_2 and k_3 are the same. Similarly, solve the permutation between the bins k_4, k_5, k_6 and k_7 . In Level 2, correlation between the mean of the envelopes of the data of the bins k_0, k_1, k_2 and k_3 with that of k_4, k_5, k_6 and k_7 is calculated and accordingly the permutation is solved for the bins $k_0, k_1, \dots, k_7$. The same approach can be extended for K bins with total levels equal to $\log_2(K)$.

Direction of arrival (DOA) [93] is another popular method, where the directivity pattern formed by the separation matrix in the frequency domain is used to estimate the directions of the sources and hence the permutation problem is solved [93, 94, 68, 19]. However, the DOA method cannot solve the permutation problem in all the DFT bins especially for lower frequency bins. H. Sawada et al. [20] combined the DOA method with correlation method to form a robust method for solving the permutation problem

Fig. 2.4: Correlation between adjacent bins.

Fig. 2.5: Solving permutation problem using dyadic sorting.

which is suitable for all the cases except when the sources are very close to each other or collinear². Another approach is the combination of these two approaches, namely, time-frequency algorithm [78]. The algorithms defined in the time domain typically will not suffer any permutation problem even if they are implemented in the frequency domain, and frequency domain implementation will improve the speed of computation. It is also shown that filter bank based BSS will improve the performance and solving the permutation problem between the filter banks will also be easy when compared to the frequency domain methods [85].

The convergence of the frequency domain algorithm greatly depends on the initial values of the unmixing matrices. In [95] it is shown that the SNR improvement obtained by beamforming initialization is much higher than that obtained with center spike initialization. Hence, the combination of the beamforming technique with blind source separation is widely used [96, 97, 94, 84, 98, 99].

²In this thesis 'collinear' means the sources and the center of gravity of the microphone array are in the same line.

2.4 Underdetermined BSS

For the case of overdetermined/determined mixing, BSS algorithms based on ICA can give very good performance. However, when the mixing is underdetermined, the performance of the algorithms based on ICA deteriorates and other approaches such as sparse component analysis (SCA) could be used. In SCA, sparsity of signals is utilized to separate the signals from their mixtures. A signal is said to be sparse if the signal amplitude is zero during most of the time period. However, signals like speech are not very sparse in the time domain. P. Bofill, et al. [23] showed that speech signals are more sparse in the frequency domain than in the time domain and hence by transforming the time domain signal into the frequency domain, the sparsity can be utilized to separate the signals from their mixtures.

For underdetermined instantaneous mixtures, utilizing the sparsity of the sources, different algorithms have been reported. Some algorithms are based on a two-stage approach, where the mixing matrix is estimated in the first stage and in the second stage, the sources are separated using the estimated mixing matrix [100]. In some other algorithms, both the mixing process and sources are estimated concurrently by selecting an appropriate cost function and defining the problem as an optimization problem. For example the Euclidean distance between $\mathbf{x}$ and $\hat{\mathbf{H}}\mathbf{y}$, $\|\mathbf{x} - \hat{\mathbf{H}}\mathbf{y}\|$, can be taken as the cost function. Based on this cost function, in [101] Lee and Seung proposed a negative matrix factorization algorithm where, while optimizing $\mathbf{y}$, $\hat{\mathbf{H}}$ is fixed and the multiplicative update rule is used to update $\mathbf{y}$ and vice versa, i.e.,

$$(\mathbf{y})_{ij} \leftarrow (\mathbf{y})_{ij} \frac{(\hat{\mathbf{H}}^T \mathbf{x})_{ij}}{(\hat{\mathbf{H}}^T \hat{\mathbf{H}} \mathbf{y})_{ij}} \quad (2.46)$$

$$(\hat{\mathbf{H}})_{ij} \leftarrow (\hat{\mathbf{H}})_{ij} \frac{(\mathbf{x} \mathbf{y}^T)_{ij}}{(\hat{\mathbf{H}} \mathbf{y} \mathbf{y}^T)_{ij}} \quad (2.47)$$

where $(\cdot)_{ij}$ represents the $(i, j)^{\text{th}}$ element of $(\cdot)$.

Again, in some other algorithms, the source components in their transformed domain (e.g. DFT and wavelet) are assigned to different cluster corresponding to the sources and finally transforming the components in each cluster back to the time domain to obtain the separated signals [18, 102]. The basic idea of estimation of the mixing matrix or source component assignment to different clusters can be explained as follows:

Consider the following under-determined mixing process:

$$\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \end{bmatrix} \begin{bmatrix} s_1(t) \\ s_2(t) \\ s_3(t) \end{bmatrix} \quad (2.48)$$

If all the sources, except s_1 , are zero at any time, i.e., $s_2(t_1) = s_3(t_1) = 0$ and $s_1(t_1) \neq 0$, then (2.48) becomes

$$\begin{bmatrix} x_1(t_1) \\ x_2(t_1) \end{bmatrix} = \begin{bmatrix} h_{11} \\ h_{21} \end{bmatrix} s_1(t_1) \quad (2.49)$$

Equation (2.49) shows that, at $t = t_1$, the direction of $\mathbf{x}(t_1) = [x_1(t_1) \ x_2(t_1)]^T$ will be the same as that of the mixing vector $\mathbf{h}_1 = [h_{11} \ h_{21}]^T$. Similarly when $s_1(t_2) = s_3(t_2) = 0$ and $s_2(t_2) \neq 0$, the direction of $\mathbf{x}(t_2)$ will be the same as that of $\mathbf{h}_2 = [h_{12} \ h_{22}]^T$ and when $s_1(t_3) = s_2(t_3) = 0$ and $s_3(t_3) \neq 0$, the direction of $\mathbf{x}(t_3)$ will be the same as that of $\mathbf{h}_3 = [h_{13} \ h_{23}]^T$. Hence if the sources are sparse, the scatter plot of the mixtures will have a clear orientation in the directions of the column vectors of the mixing matrix and therefore the mixing matrix can be estimated from the scatter diagram of the mixtures, up to a scaling factor. For the estimation of the mixing vectors, many algorithms are available in the literature. Zibulevsky et al. [100] estimated the mixing matrix by first normalizing all the mixture points and mapping them to the unit hemisphere (to avoid the cluster centroid to fall on or be very close to the origin), and the samples are then clustered using the fuzzy c -means clustering

algorithm. The centroid of the clusters so formed is taken as the column vector of the mixing matrix. Another clustering approach where the topographic maps are used is reported in [103]. O'Grady and Pearlmutter [104] proposed yet another method based on modified k -means [105] clustering for the identification of the mixing matrix.

For determined instantaneous BSS, after estimation of the mixing matrix, $\hat{\mathbf{H}}$, the source signals can be calculated by the linear transformation $\mathbf{y} = \hat{\mathbf{H}}^{-1}\mathbf{x}$ up to permutation and scaling. For the case of under-determined BSS, as $\mathbf{x}(t) = \hat{\mathbf{H}}\mathbf{y}(t)$ has more elements in $\mathbf{y}$ than in $\mathbf{x}$, the relation is non-invertible and hence $\mathbf{y}$ cannot be calculated by linear transformation. Consequently, non-linear transformation is to be used for the estimation of $\mathbf{y}$. One approach for this non-linear transformation is hard partitioning [106, 107, 108], where the samples in $\mathbf{x}$ which are close to the column vectors of the mixing matrix are assigned to the source corresponding to that column vector. This idea will work well if the sources are perfectly sparse. In cases where the sources are not perfectly sparse, a logical extension of the above idea is partial assignment of the data to different columns of the estimated mixing matrix. This is generally done by the L_1 norm minimization method [109], also known as the shortest-path [26] or the basis pursuit [110] method. The L_1 norm minimization is accomplished by formulating the problem as a linear programming problem, i.e.,

$$\text{minimize } \|\mathbf{y}(t)\|_1 \text{ subject to } \mathbf{x}(t) = \hat{\mathbf{H}}\mathbf{y}(t) \quad (2.50)$$

In the DFT domain, where the samples are complex, the real and imaginary parts are generally treated separately.

In the time-frequency domain, if the spectra of the sources are not overlapped (also called the W-disjoint orthogonality condition [18, 102]), the sources can be estimated by the time-frequency masking method proposed in [18, 102]. In the time-frequency masking method, the mixing matrix is not estimated explicitly but a binary mask for each source is estimated in such a way that the mask will have a value one

corresponding to the point in the TF plane where the source (corresponding to the mask) component is present and zero where the source component is absent. The estimated masks, $M_i(k, t)$, are then applied to the mixtures in the TF domain to obtain the separated signals, i.e., $Y_i(k, t) = M_i(k, t) X_j(k, t)$, $i = 1, \dots, Q$, $j \in \{1, \dots, P\}$. The mask can be applied on any one of the mixtures in the TF domain. The resultant so obtained is transformed into the time domain to obtain the separated signals. One well cited paper based on the time-frequency masking method is [18], where an algorithm called Degenerate Unmixing Estimation Technique (DUET) using only two mixtures is presented. The DUET was originally reported in [102]. The reported experimental results show that the algorithm can separate both the anechoic and echoic mixtures. However, the separation quality is not very good for the latter. The basic idea of the DUET algorithm is briefly explained below.

If the signals received at the sensors from Q sources are

$$x_p(t) = \sum_{q=1}^Q h_{pq} s_q(t - \delta_{pq}), \quad p = 1, 2 \quad (2.51)$$

where h_{pq} and δ_{pq} are the attenuation coefficients and time delays associated with the path from q^{th} source to p^{th} sensor. Now by taking one of the sensors, say the first sensor, as the reference sensor such that the attenuation to that sensor is one and the delay is zero, i.e., $h_{1q} = 1$ and $\delta_{1q} = 0$ for $q = 1, \dots, Q$, the mixing equation (2.51) in the TF domain can be written as

$$\begin{bmatrix} X_1(k, t) \\ X_2(k, t) \end{bmatrix} = \begin{bmatrix} 1 & \dots & 1 \\ h_{21}e^{-jk\delta_{21}} & \dots & h_{2Q}e^{-jk\delta_{2Q}} \end{bmatrix} \begin{bmatrix} S_1(k, t) \\ \vdots \\ S_Q(k, t) \end{bmatrix} \quad (2.52)$$

Now from the ratios $R_{21}(k, t) = \frac{X_2(k, t)}{X_1(k, t)} = h_{2q}e^{-jk\delta_{2q}}, \quad \forall k, t$ (assuming the sources are W-disjoint orthogonal in the TF domain), it can be seen that $|R_{21}(k, t)| = h_{2q}$ and $-(1/k) \angle R_{21}(k, t) = \delta_{2q}$ where $\angle R_{21}(k, t)$ denotes the phase of $R_{21}(k, t)$ (note that here k

is the angular frequency in rads/sec, not simply a number). Now labeling the points, (k, t) , in the TF plane with the pairs $(|R_{21}(k, t)|, -(1/k) \angle R_{21}(k, t))$ will give Q groups of points each corresponding to one of the sources in the TF domain. The separated time domain signals can now be obtained by transforming these coefficients in each group into the time domain. In [18, 102], the grouping or labeling of the points is done by the histogram method.

The DUET algorithm is quite restrictive as it requires the sources to be W-disjoint orthogonal in the TF domain. In [111, 112] it is shown that, for speech signals, approximate W-disjoint orthogonality is sufficient to separate the signals. However if the sources are not W-disjoint orthogonal, the separated signals will be distorted depending on the overlap of their spectrum in the TF domain. To overcome this problem, algorithms are reported [113], which only require the sources to occur in a tiny set of adjacent TF windows while several sources may overlap everywhere else in the TF domain. The algorithm proposed in [113], called Time-Frequency Ratio Of Mixtures (TIFROM), will automatically detect the single source points and a “canceling coefficient” will be derived from the detected single-source-points. This canceling coefficient can be used for the removal of the source corresponding to the single source points from all the observations. The basic idea for the estimation of these canceling coefficients is as follows:

Let

$$x_1(t) = h_{11}s_1(t) + h_{12}s_2(t) \quad (2.53)$$

$$x_2(t) = h_{21}s_1(t) + h_{22}s_2(t) \quad (2.54)$$

Now the separated signals can be calculated as

$$y_i(t) = x_1(t) - c_i x_2(t) \quad (2.55)$$

where $c_i = \frac{h_{1i}}{h_{2i}}$ is called the canceling coefficient. The canceling coefficients can be

Fig. 2.6: Overlapped Time-Frequency windows.

obtained by finding the time t_n at which only one of the sources is present. For example, at t_n if only source s_1 is present and s_2 is absent, i.e., $s_1(t_n) \neq 0$ and $s_2(t_n) = 0$, then

$$x_1(t_n) = h_{11}s_1(t_n) \quad (2.56)$$

$$x_2(t_n) = h_{21}s_1(t_n) \quad (2.57)$$

So that

$$\frac{x_1(t_n)}{x_2(t_n)} = \frac{h_{11}}{h_{21}} = c_1 \quad (2.58)$$

Similarly at some other time instants, if $s_1(t_m) = 0$ and $s_2(t_m) \neq 0$, taking the ratio of the mixtures will give the second canceling coefficient, i.e., $\frac{x_1(t_2)}{x_2(t_2)} = \frac{h_{12}}{h_{22}} = c_2$. Knowing the canceling coefficients, the unmixing matrix is simply given by:

$$\mathbf{W} = \begin{bmatrix} 1 & 1 \\ 1/c_1 & 1/c_2 \end{bmatrix}^{-1} \quad (2.59)$$

As sparsity of the signals is more in the frequency domain, for the estimation of the canceling coefficient, the mixture is first converted to its TF domain. In the TF domain, overlapped windows as shown in Fig.2.6 are taken and for each window the mean and variance of the ratios

$$\alpha(k, t) = \frac{h_{11}S_1(k, t) + h_{12}S_2(k, t)}{h_{21}S_1(k, t) + h_{22}S_2(k, t)} \quad (2.60)$$

are calculated. For a general case with Q sources and two mixtures the ratio will be

$$\alpha(k, t) = \frac{\sum_{q=1}^Q h_{1q} S_q(k, t)}{\sum_{q=1}^Q h_{2q} S_q(k, t)} \quad (2.61)$$

For any window, if only one of the source contributions (say that of s_q) is present, then the ratios, $\alpha(k, t) = \frac{h_{1q} S_q(k, t)}{h_{2q} S_q(k, t)}$, for all the points in that window will be constant. Hence the variance of these ratios will be a constant and the mean will be the same as the canceling coefficient c_q . If the number of observations is more than two, pairs of observations are taken to estimate the canceling coefficients. After calculating the canceling coefficients, the sources can be estimated by global matrix inversion or successive source cancellation [113].

In [114], another algorithm is proposed where the W-disjoint orthogonality condition is relaxed allowing the sources to be nondisjoint in the TF domain. However, the restriction of the method is that the number of sources present at any point in the TF plane of the mixtures must be strictly less than the number of sensors. For the two sensor case the condition reduces to the W-disjoint orthogonality condition. Under this assumption, the sources can be separated by subspace projection.

In the two stage approach proposed in [1], a method which is the extension of the DUET and TIFROM methods is used for the estimation of the mixing matrix. The algorithm is based on the ratio of the mixtures in the TF domain under the assumption that there will be points in the TF plane where only one source has a nonzero value. In the second stage, a standard linear programming algorithm is used for the source estimation.

Like DUET and TIFROM algorithms in [1], the ratio matrix is first constructed

using the transformation coefficient of the matrix as

$$\begin{bmatrix} \frac{X_1(1)}{X_p(1)} & \cdots & \frac{X_1(K)}{X_p(K)} \\ \vdots & \ddots & \vdots \\ \frac{X_n(1)}{X_p(1)} & \cdots & \frac{X_n(K)}{X_p(K)} \end{bmatrix} \quad (2.62)$$

where $X_p(k)$ is the k^{th} coefficient of the p^{th} mixture in the TF domain. Then several sub matrices are detected from the ratio matrix such that the entries in the rows of each sub matrices are almost the same. For example, at points $i_1, \dots, i_L$, if only the contribution of source s_1 is present (the points $i_1, \dots, i_L$ need not be adjacent points), then for

$$[\mathbf{X}(i_1), \dots, \mathbf{X}(i_L)] = \mathbf{H} [\mathbf{S}(i_1), \dots, \mathbf{S}(i_L)] = [\mathbf{h}_1 S_1(i_1), \dots, \mathbf{h}_1 S_1(i_L)] \quad (2.63)$$

the sub matrix will be

$$\begin{bmatrix} \frac{X_1(i_1)}{X_p(i_1)} & \cdots & \frac{X_1(i_L)}{X_p(i_L)} \\ \vdots & \ddots & \vdots \\ \frac{X_n(i_1)}{X_p(i_1)} & \cdots & \frac{X_n(i_L)}{X_p(i_L)} \end{bmatrix} = \begin{bmatrix} \frac{h_{11}}{h_{p1}} & \cdots & \frac{h_{11}}{h_{p1}} \\ \vdots & \ddots & \vdots \\ \frac{h_{n1}}{h_{p1}} & \cdots & \frac{h_{n1}}{h_{p1}} \end{bmatrix} \quad (2.64)$$

Then each column in (2.64) is $\mathbf{h}_1$ up to a scale. By plotting the values of entries in (2.64) vs the column index, it is possible to get n horizontal lines, each one corresponding to one of the rows of the matrix in (2.64). The detailed algorithm for the estimation of the sub-matrices and hence the columns of the mixing matrices are given in [1]. There are many other algorithms for instantaneous under-determined BSS: for example, algorithms based on Laplacian mixture models [115], correlation [116] and source non-stationarity [117]. Another algorithm proposed in [24] assumed that the maximum number of active sources at any instant is less than the number of mixtures [24]. The algorithm proposed in [118] is suitable not only for instantaneous mixing but also for anechoic mixing with delay and attenuation.

The technique used for the estimation of the mixing matrix for an instantaneous mixing case cannot be directly used for the case of convolutive mixing because of the complex nature of the DFT coefficients of the mixing filter. On the contrary, the coefficients are real for instantaneous mixing as the mixing filter simplifies to a single pulse. The commonly used approach for the separation of signals from their under-determined convolutive mixtures is the binary masking method. For the estimation of the mask, the single source points are first estimated. The single source points are normally estimated based on the information available from the DOA [28], estimated channel parameters [29] or approximate mixing filters estimated based on the assumption that the total number of dominant sources are less than the microphones [27]. Detailed discussions on these techniques are given in Chapter 5. Many other techniques are also used for the separation of the sources from their under-determined convolutive mixtures; for example in [119], a two stage method based on a general maximum a posteriori (MAP) approach is proposed. In the first stage, assuming that the sources are sparse in their TF domain, the mixing matrix is estimated based on applying the hierarchical clustering directly on the complex valued data. In the second stage, using the estimated mixing matrix and mixtures, the sources are estimated by l_1 norm minimization.

T. Melia and S. Rickard [120] extended the DUET algorithm by combining it with the direction of arrival estimation algorithm (Estimation of Signal Parameters via Rotational Invariance Techniques (ESPRIT)) to obtain a new algorithm called DUET-ESPRIT (DESPRIT) which can separate signals from their convolutive mixtures.

Inspired by the well-known FastICA [2, 12] and the time domain fast fixed-point algorithm for determined and overdetermined convolutive mixtures [121], a new fast fixed point algorithm is proposed in [122]. In [25], an algorithm based on normalization and clustering of the level ratios and phase difference between the multiple observations are proposed.

In this chapter a general review on BSS is given. It paves the way for further

and more in-depth discussions on issues and solutions pertaining to BSS. Detailed reviews relevant to the contributions of this thesis are given in the following chapters.

Chapter 3

Partial Separation Method for Solving the Permutation Problem

3.1 Introduction

The separation of the source signals from their convolutive mixtures is addressed mainly by two different approaches, namely time domain and frequency domain approaches. As described in Section.2.3, the permutation problem in BSS cannot be avoided. The problem, called global permutation when it is viewed in the time domain is not a serious issue because it does not affect the quality of the separated signals. The global permutation will change only the order in which the separated signals appear at the output of the BSS system. In frequency domain BSS, since the separation algorithm is applied to each frequency bin independently, the permutation may be different for different frequency bins. This type of permutation, called local permutation, will degrade the overall separation performance of the algorithm. (The permutation problem in the frequency domain BSS is depicted in Fig.2.3). Hence the permutation in each of the DFT bins has to be aligned in such a way that the separated signals in the time domain will contain the frequency components of the same source signals. The convolutive mixing and separation process in the time domain can be mathematically expressed as

$$x_p(t) = \sum_{q=1}^Q \sum_{l=0}^{\infty} h_{pq}(l) s_q(t-l), \quad p = 1, \dots, P \quad (3.1)$$

and

$$y_r(t) = \sum_{p=1}^P \sum_{l=0}^{L-1} w_{rp}(l) x_p(t-l), \quad r = 1, \dots, Q \quad (3.2)$$

respectively, where $x_p(t)$ is the p^{th} sensor output, $s_q(t)$ is the q^{th} source, $y_r(t)$ is the r^{th} separated signal, $h_{pq}(l)$ is the l^{th} coefficient of the impulse response from the q^{th} source to the p^{th} sensor, $w_{rp}(l)$ is the l^{th} coefficient of the unmixing filter between the r^{th} separated signal and the p^{th} sensor, Q is the total number of sources and P is the total number of sensors. Here even though the length of the mixing filter is shown as infinity, practically the coefficients of the mixing filters will be negligibly small after a certain time period depending on the reverberation time. Using the convolution multiplication property of DFT, (3.1) and (3.2) can be written as (note that the mixing filter coefficients will be negligibly small after a certain length)

$$\mathbf{X}(f, t) = \mathbf{H}(f) \mathbf{S}(f, t) \quad (3.3)$$

and

$$\mathbf{Y}(f, t) = \mathbf{W}(f) \mathbf{X}(f, t) \quad (3.4)$$

respectively, where $\mathbf{X}(f, t)$, $\mathbf{S}(f, t)$ and $\mathbf{Y}(f, t)$ are the discrete time-frequency transformation vectors of $\mathbf{x} = [x_1, \dots, x_P]^T$, $\mathbf{s} = [s_1, \dots, s_Q]^T$ and $\mathbf{y} = [y_1, \dots, y_Q]^T$, respectively. $\mathbf{H}(f)$ and $\mathbf{W}(f)$ are respectively the mixing and unmixing filters in the frequency domain (It is assumed that the mixing filters remain unaltered during the whole time period). After applying independent component analysis (ICA) algorithms to each frequency bin of the mixture, $\mathbf{X}$, because of the scaling and permutation problem, the unmixing matrix obtained, $\mathbf{W}(f)$, need not be equal to $\mathbf{H}(f)^{-1}$, instead it will be a scaled and permuted (rows of $\mathbf{W}(f)$ interchanged) version of $\mathbf{H}(f)^{-1}$, i.e.,

$$\mathbf{W}(f) \mathbf{H}(f) = \Gamma(f) \mathbf{D}(f) \quad (3.5)$$

where $\Gamma(f)$ is the permutation matrix and $\mathbf{D}(f)$ is the scaling diagonal matrix.

The scaling problem is relatively easy to avoid [63], whereas the permutation problem is challenging. Many algorithms have been proposed for solving the permutation problem [15, 16, 76, 63, 92, 93, 94, 20, 91, 68, 19, 78] which are briefly reviewed in Section.2.3. Section.3.2 summarizes the drawbacks of the two popular methods, out of the many existing methods, for solving the permutation problem. After the brief summary of the drawbacks of the existing algorithms, a new algorithm called *partial separation method* is proposed in Section.3.3. The proposed algorithm can be used for the separation of the collinear sources also provided a time domain BSS method is used to separate the signals at least partially so that the spectra of the partially separated signals will be closer to those of the clean sources. The proposed method uses correlation between two signals in each DFT bin to solve the permutation problem. One of the signals is partially separated by a time domain blind source separation method and the other is obtained by the frequency domain blind source separation method. Two different ways of configuring the time and frequency domain blocks, i.e., in parallel or cascade, have been studied. The cascade configuration not only achieves a better separation performance but also reduces the computational cost as compared with the parallel configuration. To validate the applicability of the proposed algorithm, numerous experiments are done using both simulated and real room impulse responses, and the experimental results are summarized in Section.3.4.

3.2 Drawbacks of the existing methods

The most commonly used approaches for solving the permutation problem are the DOA approach and the correlation approach. For convenience, the basic principles and pitfalls of these methods are summarized below.

3.2.1 Direction Of Arrival approach

By neglecting the room reverberation, the Fourier transform of the impulse response from source $s_q(t)$ at direction θ_q to the p^{th} sensor, $h_{pq}(t)$ can be approximated as [93, 94, 20]

$$H_{pq}(f) = e^{j2\pi f \tau_{pq}}, \quad \tau_{pq} = c^{-1} d_p \sin \theta_q \quad (3.6)$$

where τ_{pq} is the time lag for the source with respect to the p^{th} sensor placed at position d_p (assuming that the sensors are placed in a linear array and the direction orthogonal to the array is 0°) and c is the velocity of the signal. Using (3.3) and (3.4), the frequency response from the q^{th} source to the r^{th} separated signal can be written as [93, 94, 20]

Fig. 3.1: Directivity pattern of the two sources at two different frequencies. The actual directions of the sources are -30° and 20° .

$$\begin{aligned}
U_{rq}(f) &= \sum_{p=1}^P W_{rp}(f) H_{pq}(f) \\
&= \sum_{p=1}^P W_{rp}(f) e^{j2\pi f c^{-1} d_p \sin \theta_q}
\end{aligned} \tag{3.7}$$

By considering the direction θ_q as a variable, say θ , the above equation becomes

$$U_r(f, \theta) = \sum_{p=1}^P W_{rp}(f) e^{j2\pi f c^{-1} d_p \sin \theta} \tag{3.8}$$

The gain $|U_r(f, \theta)|$ will vary according to the direction of the angle θ and hence it is called the directivity pattern. The value of the gain $|U_r(f, \theta)|$ will be high when θ is equal to the direction of the source corresponding to the separated signal y_r and low in the directions of the other sources. Hence the directions of all the sources can be estimated by the rows of the unmixing filter, $W_{rp}(f)$, $r = 1, \dots, Q$, for all the frequency bins. The permutation problem in each bin can be fixed by proper clustering of the above estimated directions. However, as shown in Fig.3.1, for lower frequencies, it is unable to estimate the directions of the sources from the directivity pattern and therefore the permutation problem for these bins cannot be fixed. The reason for this failure at lower frequency bins are 1) the lower spacing between the microphones and hence the smaller phase differences between the signals picked up by the microphones which will lead to higher measurement errors and 2) the relation 3.6 is approximated for plane wavefront under anechoic condition which is not true in a real room environment. Also, under heavy reverberant conditions, the algorithm may not be able to solve the permutation problem even for the higher frequencies, and hence, the permutations in these bins have to be fixed by some other methods. Another major disadvantage of using DOA method is that, when the sources are collinear or separated at a very small angle, the method will fail. In addition to that, to apply DOA method, the spacing between the microphones must be less than half the

wavelength of the highest frequency component of the signals to avoid spatial aliasing.

3.2.2 Correlation approach

For speech signals there is a strong correlation between the adjacent frequency bins. This inter-frequency correlation is used to solve the permutation problem in correlation methods (see [63] for more details). In [20], the algorithm proposed in [63] is simplified so that the envelope $v_r^f(t) = |S_{\Pi(r)}(f, t)|$ is used to measure the correlation between the neighboring bins. The correlation between the envelopes in the neighboring frequency bins will be high, if the separated signal belongs to the same source signal. Hence the permutation at frequency bin f , corresponding to $\Gamma^{-1}(f)$, can be estimated by maximizing the sum of the correlation between the neighboring frequencies within the frequency distance δ [20], i.e.,

$$\Pi_f = \operatorname{argmax}_{\Pi} \sum_{|g-f| \leq \delta} \sum_{r=1}^Q \operatorname{cor}(v_{\Pi(r)}^f, v_{\Pi_g(r)}^g) \quad (3.9)$$

where Π_g is the permutation at frequency g . The correlation between two signals $x(t)$ and $y(t)$ is defined as [20]

$$\operatorname{cor}(x, y) = \frac{E(xy) - E(x)E(y)}{\sqrt{E(x^2) - E^2(x)} \sqrt{E(y^2) - E^2(y)}} \quad (3.10)$$

The main disadvantage of this method is that a mistake in one frequency bin will lead to complete misalignment beyond that frequency bin.

3.2.3 Combined approach

In the work of H. Sawada et al. [20], the DOA and correlation approaches were combined so that the permutation is fixed for certain frequency bins where the confidence of the DOA method was sufficiently high. For the remaining bins the permutation was decided based on the correlation between the neighboring or harmonic frequency

bins without changing the permutation fixed by the DOA method. The harmonic correlation method utilizes the fact that, for speech signals, the envelope $v_r^f(t)$ at frequency f has strong correlation to that at its harmonic frequencies $2f$, $3f$ and so forth.

Hence, if the permutation is not fixed for frequency f , but fixed for its harmonic components, then the permutation at f , Π_f , can be fixed by maximizing the correlation between $v_r^f(t)$ at frequency f and its harmonic frequency bins [20], i.e.,

$$\Pi_f = \underset{\Pi}{\operatorname{argmax}} \sum_{g \in (\text{Harmonics of } f)} \sum_{r=1}^Q \operatorname{cor}(v_{\Pi(r)}^f, v_{\Pi_g(r)}^g) \quad (3.11)$$

The reason for using the harmonic correlation method in addition to the neighboring bands correlation method is that, the DOA approach does not provide sufficient confidence to fix the permutation continuously for a certain range of frequencies which are at the lower frequency end. In such a case, the use of the neighboring correlation method alone may not be able solve the permutation problem in these frequency bins. However, since the harmonic correlations at lower frequency bins are high, they can be used to solve the permutation problem.

3.3 Proposed method

The proposed system basically consists of two blocks, namely, the time domain and frequency domain blocks. The time domain block partially separates the signal mixture using the time domain algorithm for BSS, whereas the frequency domain block separates the signals using the frequency domain BSS algorithm. The permutation problem in the DFT bins of the frequency domain BSS block is solved by using the correlation between the envelope, $|\hat{S}_q(f, t)|$, in the DFT bins of the partially separated signal (partially separated signal is converted to the DFT domain time series signals, $\hat{S}_q(f, t)$) and that of the frequency domain BSS block.

The input to the frequency domain stage can either be the output from the microphones or the partially separated signals from the time domain stage. Correspondingly there are two configurations for the proposed system, namely the parallel configuration [123] and the cascade configuration [22]. Detailed explanations for each of these configurations are given in the following sections.

The time domain algorithm used in this chapter is the computationally efficient implementation [90] of the algorithm proposed in [81, 82]. The algorithm is based on the second order statistics which is briefly explained in Section.2.3.

3.3.1 Parallel configuration

The block diagram of the proposed method for the parallel configuration with two sources is shown in Fig.3.2. For simplicity, it is assumed that the number of sources is equal to the number of mixed signals, i.e., $P = Q$. The time domain signals at the

Fig. 3.2: Block diagram for the proposed partial separation method for solving the permutation problem in frequency domain BSS (parallel configuration).

microphone outputs are first converted to the frequency domain time series signals, $X_p(f, t)$, $p = 1, \dots, P$, using K point FFTs. The complex valued ICA [12] algorithm for linear instantaneous mixtures (see Section.2.3) is then applied to all the K frequency bins so as to obtain the separated signals in each bin. The separated signals in different frequency bins may have different permutations that have to be solved before they are combined by the inverse fast Fourier transform (IFFT) to obtain the separated signals, $y_r(t)$, $r = 1, \dots, Q$. To solve this permutation problem, the partially separated signals, $\hat{s}_q(t)$, $q = 1, \dots, Q$, separated in the time domain as shown in Fig.3.2 are used. The listening quality of the partially separated signal need not be very good but the spectra of the separated signals must be closer to the corresponding sources.

Each partially separated signal in the time domain, $\hat{s}_q(t)$, is then transformed to the frequency domain using FFT of the same length as that for the microphone output, i.e., K . Now the magnitude envelope of the k^{th} bin of the q^{th} partially separated signal, $|\hat{S}_q(f_k, t)|$, will be more correlated to the corresponding bin of the fully separated signal $|S_{\Pi(q)}(f_k, t)|$. So the magnitude envelopes among $|S_{\Pi(r)}(f_k, t)|$, $r = 1, \dots, Q$, having the highest correlation with $|\hat{S}_q(f_k, t)|$ are identified, which is $|S_{\Pi(r)}(f_k, t)|$, for $r = q$, and is assigned to $Y_q(f_k, t)$. Since the adjacent bins of a speech signal can be highly correlated, instead of taking a single bin from the partially separated signal, the average of the k^{th} and its adjacent bins defined by Δk is used. Hence the permutation, $\Pi_{f_k}(q)$, for the k^{th} bin is

$$\Pi_{f_k}(q) = \underset{\Pi(r), r=1, \dots, Q}{\operatorname{argmax}} \operatorname{cor} \left(\frac{1}{2\Delta k + 1} \sum_{b=k-\Delta k}^{k+\Delta k} |\hat{S}_q(f_b, t)|, |S_{\Pi(r)}(f_k, t)| \right) \quad (3.12)$$

The same procedure is repeated for all $\hat{S}_q(f_k, t)$, $q = 1, \dots, Q$, and $k = 1, \dots, K$. Subsequently, these assignments $Y_q(f_k, t)$ have to be converted back to the time domain signal $y_q(t)$ using IFFT. One problem with this method is that the algorithm may not be able to solve the permutations for all the bins, with full confidence, which can be explained as follows.

Let $c_{qr,k}$, $q = 1, \dots, Q$, $r = 1, \dots, Q$ be the correlation between the k^{th} bin of the q^{th} partially separated signal and that of the r^{th} fully separated signal. For Q sources there will be $Q \times Q$ such correlations for each bin. Considering $c_{qr,k}$ is an element in the q^{th} row and r^{th} column of a $Q \times Q$ matrix and if the highest Q values are distributed in that matrix in such a way that there is no column or row common to each other then it can be said that each partially separated signal has one and only one correlation with the fully separated signal. In that case $S_{\Pi(r)}(f_k, t)$ is assigned to $Y_q(f_k, t)$ with confidence. If not, leave that bin as it is and use the correlation between neighboring or harmonic bins to solve the permutation problem. Also for these bins, permutations are fixed after checking the confidence. To check the confidence, the same procedure that has just been explained is used. For the remaining bins, the neighboring correlation approach [20], which is shown in (3.9), is used to solve the permutation problem.

For clarification the confidence checking procedure can be illustrated with examples as follows. Let the number of sources be $Q = 2$ and the correlations at bin k be $c_{11,k} = 0.6$, $c_{22,k} = 0.7$, $c_{12,k} = 0.3$ and $c_{21,k} = 0.2$ as shown in Fig.3.3. Here, the highest two correlations are $c_{11,k}$ and $c_{22,k}$, which are the diagonal elements. By considering the elements of a matrix as the correlations, it is clear that there is no common row and column having the highest elements. In this case, the permutation problem can be solved with confidence. Instead, if $c_{11,k} = 0.8$, $c_{22,k} = 0.1$, $c_{12,k} = 0.3$ and $c_{21,k} = 0.4$, then the k^{th} bin will not be altered during the first round of solving the permutation (the round which uses confidence check), because the highest two correlations, $c_{11,k}$ and $c_{21,k}$, lie in the same column of the matrix. The permutation problem in the frequency bins where the algorithm failed to solve in the first round will be solved in the second round according to (3.9), i.e., without checking the confidence. During the second round, the algorithm will only check whether $c_{11,k} + c_{22,k} \geq c_{12,k} + c_{21,k}$ or $c_{11,k} + c_{22,k} < c_{12,k} + c_{21,k}$. Hence the proposed method consists of the following three steps:

Fig. 3.3: The two correlation matrices. (a) No column or row between the highest elements and hence the permutation problem can be solved with confidence (b) The highest elements are in the same column and hence the permutation problem cannot be solved with confidence.

- Fix the permutation using the correlation between the partially separated signal and the fully separated signal for the bins where it can be fixed with confidence.
- Fix the permutation either using the adjacent bins correlation or the harmonic correlation method with confidence, without changing the permutation fixed in the previous step.
- For the remaining bins, fix the permutation using (3.9).

3.3.2 Cascade configuration

Certain time domain algorithms will distort the spectrum of the separated signals. For example, the algorithm which does not consider the temporal correlation of the speech signal (e.g. [98]) may whiten the spectrum of the separated signal. For such time domain algorithms the parallel configuration will be better [123]; otherwise, the output of the frequency domain stage also will remain distorted because of its distorted input. On the other hand, if the algorithm is not distorting the spectrum of the separated signals, the cascade configuration will be the best option as it not only

improves the computational efficiency but also the overall separation performance. The block diagram for the cascade configuration is shown in Fig.3.4. By comparing the two block diagrams, parallel and cascade, it can be seen that the number of fast Fourier transform (FFT) blocks for the cascade configuration is only 2, whereas for the parallel configuration is 4. Hence the computational cost is reduced. Since the input signal to the frequency domain stage in the case of the cascade configuration is the partially separated signals, which is already separated to a certain extent, the overall performance is higher than that of the parallel configuration. In [124, 125] it is shown that the cascade configuration of the time domain and the frequency domain stages will improve the separation performance. Unlike in [124, 125], where the DOA method is used to solve the permutation problem in the frequency domain stage, the proposed algorithm uses the signals from the time domain stage to solve the permutation problem and hence the computational cost is optimally utilized.

Fig. 3.4: Block diagram for the proposed method (cascade configuration).

Table 3.1: Experimental Conditions

Source signals	Speech of 15sec, except in Section 3.4.4 (obtained by concatenating the sentences from TIMIT database)
Direction of sources	As shown in the respective figures
Distance between two microphones	20cm for PS and 4cm for DOA
Sampling rate f_s	16kHz
DFT size	$K = 4096$
Frequency resolution	$\Delta f = f_s / K = 3.90625\text{Hz}$
Distance Δk	$18\Delta f$
Distance δ	$9\Delta f$
Number of filter taps for time domain algorithm	512 (unless otherwise specified)
Room temperature	25°
Humidity of air	40% (for simulation)
Wall reflections	10^{th} order (for simulation)
Window function	Hanning window

3.4 Experimental results

For performance analysis of the proposed algorithms, both simulated and measured room impulse responses are used. For the simulation of the room impulse response, a freely available “shoebox” room simulation toolbox namely “Room MATLAB toolbox” is used [126]. Simulated room impulse responses are used in Sections 3.4.1 and 3.4.2. The wall reflections up to 10^{th} order are taken and humidity, temperature and absorption of sound due to air are considered for simulating the impulse responses [126]. Whereas, in Sections 3.4.3, 3.4.4 and 3.4.5, the measured real room impulse response is used. For all the experiments, the average performance of 10 sets of speech utterances is used to evaluate the performance. The speech data are obtained by concatenating the sentences taken from TIMIT database. For each set the combination of one female and one male speech is used. The speech data used in these experiments are shown in Fig.3.5 and 3.6. The algorithms are also tested with real recorded speech mixtures.

Fig. 3.5: Female speech utterances used for the experiments. F_n and M_n in Fig.3.6 together constitute one set, where $n \in \{1, 2, \dots, 10\}$. The audio files are available in the accompanying CD

Fig. 3.6: Male speech utterances used for the experiments. F_n in Fig.3.5 and M_n together constitute one set, where $n \in \{1, 2, \dots, 10\}$. The audio files are available in the accompanying CD

Fig. 3.7: The source-microphone configuration for the room impulse responses simulation.

3.4.1 Performance evaluation for collinear and non-collinear sources

The major disadvantages of the time domain BSS method for convolutive mixtures are the statistical interdependency among the filter coefficients, which hinders the convergence, and the heavy computational cost involved for long filter taps [127]. Furthermore, good separation of convolutive mixtures using a time domain method when the sources are collinear is a formidable task [128]. To show that the proposed algorithm can solve the permutation problem even when the quality of the time domain partially separated signal is very poor, the simulated impulse responses between the sources and the microphones for the configurations shown in Fig.3.7 are used. The sources are placed in two different configurations: *i*) collinear (Sources 2 and 2'), for which the time domain separation is very poor *ii*) non-collinear (Sources 3 and 3'), for which the time domain separation is good. The mixed signals are generated by convolving the impulse responses obtained for the above configurations with the speech signals so that the performance can be evaluated by calculating the noise

Fig. 3.8: Separation performance of the proposed method (reverberation time $TR_{60} = 86\text{ms}$): Partial Separation approach PS, Correlation approach C1 and the combined approaches PS+C1, PS+C2+C1 and PS+C2+Ha +C1

Fig. 3.9: NRR at different frequencies for the 4th set of speech utterances in Fig.3.8.

reduction rate (NRR¹) defined as follows [124]:

$$\text{NRR} = \frac{1}{Q} \sum_{q=1}^Q \left(\text{SNR}_q^{(O)} - \text{SNR}_q^{(I)} \right) \quad (3.13)$$

$$\text{SNR}_q^{(O)} = 10 \log_{10} \frac{\sum_f |A_{qq}(f) S_q(f)|^2}{\sum_f |A_{qp}(f) S_p(f)|^2} \quad (3.14)$$

$$\text{SNR}_q^{(I)} = 10 \log_{10} \frac{\sum_f |H_{qq}(f) S_q(f)|^2}{\sum_f |H_{qp}(f) S_p(f)|^2} \quad (3.15)$$

where $\text{SNR}_q^{(O)}$ and $\text{SNR}_q^{(I)}$ are the output SNR and the input SNR respectively and $q \neq p$. $A_{qp}(f)$ is the q^{th} row and the p^{th} column of the matrix $\mathbf{A}(f)$, which is given below:

$$\mathbf{A}(f) = \mathbf{W}(f) \mathbf{H}(f) \quad (3.16)$$

For each configuration of the source positions shown, both parallel and cascade configurations of the proposed methods are used to solve the permutation problem. In each case, the permutation is solved in five different ways:

- Partial Separation method (PS). Frequency bins whose permutation could not be solved with confidence are left unaltered.
- Correlation between adjacent frequency bins (C1) using (3.9)
- Combination of PS and C1 (PS+C1), i.e., PS followed by C1.
- PS followed by C2, that solves permutation by using correlation between adjacent frequency bins with confidence, and finally followed by C1 (i.e., PS+C2+C1)
- Combination of PS and C2, followed by Ha which solves permutation using correlation between the harmonic components with confidence, and finally followed by C1 (i.e., PS+C2+Ha+C1).

Fig.3.8 shows the performance of the proposed method for both collinear and non-

¹It may be noted that in this thesis the noisy mixture is not considered. Here, in NRR, the noise refers to the interfering signals.

collinear sources. For clarity, only the average performance of the 10 sets of speech utterances are shown, except for the collinear sources in the parallel configuration where the performances of all the 10 sets of speech utterances are shown. For the parallel configuration, since the time domain NRR is not directly observable at the output, it is shown with 'dash-dot' line. As reported in [20] and from Fig.3.8, it can be seen that C1 can solve the permutation problem in most of the cases but it is not stable and sometimes it results in very poor performance. The performance of PS alone is not good but stable as PS solves the permutation for the frequency bins where the confidence is sufficiently high. However, for PS+C1, the performance is improved in terms of NRR as compared with PS and stability as compared with C1 alone. After solving permutation by PS followed by C2 and before C1, the performance is again improved. Also it can be seen that PS+C2+Ha+C1 offers almost the same performance as that without Ha. This is because Ha is used to solve the permutation problem at lower frequencies where the problem is already solved by the PS method. The NRR measured at each frequency bin for a pair of speech utterances (set 4) using different methods are shown in Fig.3.9. It shows the effectiveness of the proposed method in solving the permutation misalignment problem. There are large regions of permutation misalignment (frequency bins in the range 800Hz to 7000Hz) when C1 alone is used. Since PS can provide correct permutation for certain frequencies in these regions, the problem is solved when PS is combined with C1. Unlike the DOA approach, where it is very hard to estimate the direction of arrival at lower frequencies and hence difficult to solve the permutation with confidence, PS solves the permutation problem almost uniformly irrespective of the frequencies as shown in Fig.3.9. Note that the case shown in Fig.3.9 is for the worst case, where the NRR for the time domain method is only 3.98dB and also it is for the parallel configuration. From Fig.3.8, it can be seen that even for collinear sources with poor separation in the time domain, the performance will be significantly better in the cascade configuration. In Fig.3.8 for the cascade configurations, the performance shown for the correlation

method alone (C1) is better than the all other methods, because C1 was successful for all the 10 sets of speech utterances. Whereas for the parallel configuration, the method failed for some of the sets. This is not because of the configuration but because of the reason that the correlation method alone is highly unreliable, as explained in Section 3.2.2.

3.4.2 Performance evaluation under different reverberation times

For further analysis, the performance of the algorithm is compared with the DOA method. Since DOA utilizes the direction of arrival of the signal for solving the permutation problem, the distance between microphones must be less than $c/2f$, where c is the velocity and f is the frequency of the signal, to avoid spatial aliasing. However, when the microphones are very close, the problem reduces to the single channel mixing case. For the time domain method used in this chapter more spacing between the microphones is required to achieve separation. Table 3.2 shows the performance of the time domain and DOA method for different microphone spacings, while keeping all other parameters such as the position of the sources, central point of the microphones, room surface absorption and size of the room the same as that in other experiments. Since the performance of the PS method depends on the quality of the partially separated signal, the NRR of the time domain method is shown instead of the NRR of the PS method. Also while calculating the NRR for the DOA method,

Table 3.2: NRR for the time domain method and DOA method for different microphone spacings (Room surface absorption = 0.5)

Spacing (cm)	Time domain (dB)	DOA (dB)
2	1.98	6.38
4	2.80	6.84
8	4.22	6.24
12	6.24	5.29
16	7.89	4.33
20	8.21	3.65

the frequency bins where the DOA method could not solve the permutation with confidence are left unaltered. This is because the robustness of the DOA method followed by correlation methods depends on the success of the DOA method alone in solving the permutation problem. From the table it can be seen that 4cm is the optimum distance for the DOA method. For the PS method 20cm is taken. Note that this comparison is done only to find out the optimum microphone distance for the DOA method. Since the distances between the microphones are not the same, the following comparisons are not exact but they will give an approximate result.

For the experiments in this section, room impulse response is simulated [126] for different values of room surface absorption. The impulse responses thus obtained are shown in Fig.3.10. The other experimental conditions are shown in Table. 3.1. The positions of the sources are shown in Fig.3.7 (Sources 1 and 1').

The performance of the two methods, PS and DOA, when used alone for different values of reverberation time are shown in Fig.3.11. For both the cases of 'DOA only' and 'PS only with confidence check' the bins whose permutation could not be fixed with confidence are left unaltered.

Unlike the correlation approach discussed in Section 3.2.2, which utilizes the inter-frequency correlation of the signal for permutation alignment [63], the PS method utilizes the correlation with the partially separated signal for solving permutation. So, even if the permutation fixed in one bin is wrong, the permutation alignment in the remaining bins will not be affected by the wrongly aligned one. Hence the reliability of the method is high and the permutation in all the bins can be fixed without confidence check. This is shown as 'PS only without confidence check' in Fig.3.11, where the permutation of all the bins are fixed without checking the confidence as

$$\Pi_{f_k} = \underset{\Pi}{\operatorname{argmax}} \sum_{q=1}^Q \operatorname{cor} \left(\frac{1}{2\Delta k + 1} \sum_{b=k-\Delta k}^{k+\Delta k} \left| \hat{S}_q(f_b, t) \right|, \left| S_{\Pi(q)}(f_k, t) \right| \right) \quad (3.17)$$

for all the bins, where $2\Delta k + 1$ is the total number of adjacent bins of the partially

Fig. 3.10: Room impulse response for different values of surface absorption: 0.3 ($TR_{60} = 235\text{ms}$), $0.5 (TR_{60} = 130\text{ms})$, $0.7 (TR_{60} = 86\text{ms})$ and $0.9 (TR_{60} = 63\text{ms})$. Only the impulse responses from Source 1 to Microphone 1 are shown.

Fig. 3.11: Performance comparison of PS method alone with DOA method alone as a function of room surface absorption.

separated signal taken to obtain the average. From Fig.3.12, it can be seen that the difference in performance of PS alone without confidence check ('PS only') and PS with confidence check followed by the methods utilizing the correlation between adjacent and harmonic bins ('PS followed by others') are very small. Therefore, the PS method alone can be used to solve the permutation problem which reduces the computational cost at the expense of very small performance reduction. In Fig.3.12, the performance of the DOA method followed by other correlation methods ('DOA followed by others') are also shown for comparisons. The reasons for the huge difference in performance of the time domain method, for different room reverberation times, as shown in Fig.3.11 and 3.12 will be explained in Section 3.4.5.

3.4.3 Performance evaluation using the measured real room impulse response

The performance of the PS method is also evaluated using the measured impulse response of a real furnished room. The reverberation time of the room (TR_{60}) is 187ms

Fig. 3.12: Performance comparison of PS method alone without confidence check with PS method after confidence check followed by the methods which utilizes the correlation between adjacent and harmonic bins, for parallel and cascade configurations. The DOA method after confidence check followed by correlation methods are also shown.

Fig. 3.13: The source-microphone configuration for the measurement of real room impulse responses

and is measured with the help of an acoustic impulse response measuring software “Sample Champion” [129]. The microphone and loud speaker transfer function are neglected in the measurement. The positions of the sources and sensors are shown in Fig.3.13 and the corresponding measured impulse responses are shown in Fig.3.14. The other experimental conditions are the same as those used in the previous sections. The measured performances are shown in Fig.3.15, from where it can be seen

Fig. 3.14: Measured impulse responses of the room (Reverberation time $TR_{60} = 187\text{ms}$)

Fig. 3.15: NRR for various algorithms using real room impulse responses. PS - Partial Separation method with confidence check, C1 - Correlation between the adjacent bins without confidence check, C2 - Correlation between adjacent bins with confidence check, Ha - Correlation between the harmonic components with confidence check, PS1 - Partial separation method alone without confidence check.

that the performance of PS alone without confidence check is very close to that of PS+C2+Ha+C1. Therefore the PS method alone, which is reliable and independent of the positions of the sources and better than the DOA approach, can be used to solve the permutation problem in frequency domain BSS of speech signals. Note that the impulse responses for the PS method and DOA methods are not exactly the same because of the difference in microphone spacing and hence, for C1, the NRRs are different. The waveforms of clean, mixed and separated signals are also shown in Fig.3.16 where PS+C2+Ha+C1 is used to solve the permutation problem. For clarity, only 5 seconds of the 15 seconds waveforms are shown.

The robustness of the PS method is further illustrated in Fig.3.17. In Fig.3.17, using 20 sets of speech utterances (obtained by concatenating speech utterances taken from TIMIT database, shown in Fig.3.5 and 3.6) and measured room impulse responses, and for each set of speech utterances the three methods, namely 1) partial separation correlation method alone, 2) adjacent bins correlation method alone and 3)

Fig. 3.16: Waveform of the clean, mixed and separated signals. Permutation problem is solved by PS+C2+Ha+C1, NRR=14.68.

Fig. 3.17: Separation result for 20 pairs of speech utterances with different methods for solving permutation problem. (The time domain stage is present in all the cases)

partial separation correlation method with confidence check followed by adjacent bins correlation method, are applied. From Fig.3.17 it can be seen that the performance of the adjacent bins correlation method is good in most of the cases and fails in some cases because of the reason mentioned in Section.3.2.2. Hence the adjacent bins correlation method alone is not reliable. When the partial separation correlation method alone is used, even though the performance is slightly below that of the adjacent bins correlation method, the reliability is very high as the permutation fixed in any bin is independent of the permutation of any other bins of that signal and depends only on the spectrum of the partially separated signals. When the methods are combined, i.e., PS method with confidence check followed by the adjacent bins correlation method for the bins where the PS method failed to solve the permutation

Fig. 3.18: NRR for different lengths of speech utterances when the NRR of the partially separated signals used for solving permutation problem are of different levels.

with confidence, the performance is improved and the robustness is unaffected, as shown in Fig.3.17.

3.4.4 Robustness test for short speech utterances

The accuracy of the correlation method for solving the permutation problem depends on the length of the speech utterances [94, 93] because as the speech duration becomes shorter, the speech utterances tend to resemble each other. Since the proposed method is based on the correlation between the partially separated and fully separated signals in the frequency bins, the robustness of the algorithm for shorter speech segments is tested in this experiment. The algorithm is tested for 1, 2, 3, 4, 5 and 10 seconds speech utterances using different levels of partial separation (2, 4, 6 and 8 dB). Fig.3.18 shows the test results and it can be seen that the algorithm works even for shorter speech utterances and 3 sec speech is sufficient to obtain a good separation. For these experiments the length of the data in the frequency bins is fixed at 500 for all speech durations, by adjusting the overlapping of the sliding window,

Fig. 3.19: Performance variation for various filter lengths of the time domain stage. The sampling frequency of the signals is 16kHz.

while converting into the frequency domain. The unmixing filter length of the time domain stage is 512 in all the cases and the number of iterations are adjusted to obtain different levels of partial separation.

3.4.5 Effect of combination order in cascade configuration

In [124, 125] the time domain and frequency domain stages were cascaded for improving the NRR. Unlike the proposed method, in [124, 125], the output of the frequency domain stage is given as the input to the time domain stage. Even though the objectives of [124, 125] were not to solve the permutation problem, in this thesis the performance difference between the order of the time domain and frequency domain stages when they are swapped are studied. To find out the optimum filter length for the time domain stage in the proposed cascade configuration, experiments for different values of filter lengths are conducted. The results are shown in Fig.3.19. To model the complete reflections in the room, the length of the unmixing filter must be higher than or equal to the length of the mixing filter [82]. From Fig.3.19

Fig. 3.20: Performance of the frequency domain stage followed by time domain stage configuration for different lengths of filter taps as well as for different lengths of the data for learning.

Fig. 3.21: Effect of permutation in the frequency bins, for time domain separation. NRR for the mixture due to the permutation of clean signals is indicated by “clean permuted” and that of the mixture due to the permutation of the mixed signals is indicated by “mixture permuted”. For example if multiple = 8, the permuted bins are 8, 16, 24,...,4096; similarly for other multiples.

it can be seen that the smaller the filter length, the poorer the performance. This is because of the reason that the length of the unmixing filter is less than the required. However, as the unmixing filter length increases, due to the interdependency of the filter coefficients [127], the convergence will become slower, which is evident from Fig.3.19. (Since the filter coefficients of all the unmixing filters are to be adjusted simultaneously to maximize or minimize the cost function for the source separation, the filter coefficients are highly interdependent). As the filter length increases from 64 to 4096, the NRR increases and then drops. Here, out of the different filter lengths used, the best performance is achieved at 512. Hence for the remaining parts of the experiments 512 taps are used for the time domain stage, unless otherwise specified.

The NRR obtained by using different filter lengths for the time domain stage in the cascade configuration proposed in [124, 125], i.e., frequency domain stage followed by the time domain stage, is shown in Fig.3.20. Fig.3.20 is for the 100% data learning case (i.e., all the samples in the frequency bins are used for training to estimate the

mixing matrix) and the other cases will be explained later. For the frequency domain stage the permutation problem is solved using the proposed parallel configuration with only the difference that, instead of using the partially separated signals, the clean signals are used to solve the permutation problem in the best possible way, which is the ideal case. The results show that for the cascade configuration with the frequency domain stage followed by the time domain stage, the NRR is poorer than that of the frequency domain stage alone. This is due to: *i*) For further separation, after the frequency domain separation, fine tuning of the unmixing filter coefficients with longer filter taps is needed. But as discussed above, when the number of filter taps increases, because of the interdependency of the filter coefficients, the convergence is poor. *ii*) In the frequency domain stage even if the signals are perfectly separated in all the frequency bins, the algorithm for solving the permutation problem may fail to solve the permutation problem perfectly. Hence the resultant signals from the frequency domain stage after converting them into time domain will be too complex for the time domain algorithm to separate. The condition will be even worse if the signals in the frequency bins remain mixed in addition to the permutation. This is clear from the simulation results shown in Fig.3.21, where in one case the clean signals are first converted into frequency domain and every B^{th} bin is permuted, where $B = 2^b$, $b = 2, 3, \dots, 11$, so that the signals in all the bins are fully separated. However, after converting them into the time domain, because of the permutation problem, the signals in the time domain are mixed. In the second case, instead of clean signals, the mixed (convolutive mixing using real room impulse response) signals are converted into the frequency domain and every B^{th} frequency bin is permuted as in the first case before they are converted to the time domain. When the time domain algorithm is applied to these two sets of signals, it is found that even for the case where the mixing was only due to the permutation, the separation is very difficult. Also, when the mixing is a result of convolutive plus permutation, the result is much worse. In Fig.3.21, for smaller values of B , the NRR for the “mixture permuted” (permuted using

mixed signals, i.e., second case) signals is higher than that of the “clean permuted” (permuted using clean signals, i.e., first case) signals. This is because, at the k^{th} frequency bin, $X_1(f_k, t)$ and $X_2(f_k, t)$ are more correlated than $S_1(f_k, t)$ and $S_2(f_k, t)$ [130], where $X_n(f_k, t)$ and $S_n(f_k, t)$ are the data in the k^{th} frequency bin of the n^{th} mixed and clean signals respectively. Hence the mixing effect due to the permutation of $S_1(f_k, t)$ and $S_2(f_k, t)$ is more than that of permuting $X_1(f_k, t)$ and $X_2(f_k, t)$. To make this point clear, consider an example: Let $X_1(f_k, t) = 0.6S_1(f_k, t) + 0.4S_2(f_k, t)$ and $X_2(f_k, t) = 0.4S_1(f_k, t) + 0.6S_2(f_k, t)$. Now the source contributions from s_1 and s_2 are 60% and 40% respectively for one group. Let this be the group from which the source s_1 will be reconstructed. Similarly, the source contributions from s_1 and s_2 are 40% and 60% respectively for the second group. Let this be the group from which the source s_2 will be reconstructed. Before permutation, the first group contains 60% of the required signal component which is s_1 , whereas the second group contains 60% of the required signal component which is s_2 . After permutation the first group will have 40% of the required signal whereas the second group will have 40% of the required signal component. The difference in source contributions in the two groups, because of the permutation, is only 20%. Now consider the case of permuting the clean signals, i.e., $X_1(f_k, t) = S_1(f_k, t)$ and $X_2(f_k, t) = S_2(f_k, t)$. In this case the difference in source contributions because of the permutation is 100%. Now assume that the permutation in every alternate bin are changed and after that they are converted into the time domain. Then it is obvious from the above discussion that the signal obtained by permuting the clean signals will have more mixing effect than that obtained by permuting the mixed signals. In the cascaded frequency domain followed by time domain configuration, even if the permutation is fully solved but the signals are not completely separated in the frequency bins, the time domain stage requires fine tuning with long filter taps, which is a difficult task because of the reasons mentioned in Section 3.4.1. Moreover, if the separation in each bin is not perfect, the chance of getting wrong permutations is high and this will further worsen the result.

The length of the data in the frequency bin will affect the separation at the frequency domain stage. This is because as the data length decreases the independence assumption will collapse [130] and hence the separation will be poor. Fig.3.20 also shows the variation in NRR for different data lengths. The maximum length of the data in each bin is 117 (15 sec speech, 16 kHz sampling frequency, 4098 point FFT and 50 % overlap). Since the mixing filter is constant over the full length of the speech signals, only a fraction of the data (20%, 40%, 60% and 80%) in the frequency bins is used for learning, ie., for estimating the unmixing filter. The full length data is then separated using the estimated filter. As expected [130], the smaller the data length for learning, the poorer the separation, which is shown in Fig.3.20. It can also be seen that the time domain stage could not improve the separation further, for any of these cases. Hence it can be concluded that it is better to place the time domain stage in front of the frequency domain stage due to the following reasons: *i*) The time domain block used to generate the partially separated signals for solving permutation problem can be fully utilized; otherwise, a separate time domain block will be required only for solving the permutation problem. *ii*) Efficient and approximate algorithms like [3] can be used with a smaller number of taps, as only partial separation required.

3.5 Summary

In this chapter, a method for solving the permutation problem in frequency domain BSS of speech signals is proposed. The method uses correlation between the signals in each frequency bin, with one of the stages being partially separated by a time domain BSS method and the other by a frequency domain BSS method. The algorithm does not require the knowledge of the directions of arrival of the sources. Hence, if the time domain BSS algorithm can partially separate the signals so as to make the spectra of the resulting signals closer to their corresponding sources, the proposed method can be used to separate collinear sources also. Unlike the correlation method which

utilizes the inter-frequency correlation of the signals, the reliability of the proposed method is high as it utilizes the correlation with partially separated signals. The additional computational cost of the proposed method in the cascade configuration is due to the time domain stage. Since it is cascaded with the frequency domain stage, the performance improvement is due to the correctly fixed permutation as well as the multistage configuration. Hence, the additional computational cost is optimally utilized.

Chapter 4

Mixing Matrix Estimation In Underdetermined Instantaneous Blind Source Separation

4.1 Introduction

This chapter addresses the problem of estimation of the mixing matrix from their underdetermined instantaneous mixtures. In two stage approaches for instantaneous BSS, this matrix can be used for the estimation of the sources from their mixtures. One of the properties of the signals which is widely utilized for underdetermined BSS is their sparsity in the frequency domain. Here, a simple algorithm is proposed for the detection of points in the Time-Frequency (TF) plane of the instantaneous mixtures where only single source contributions occur. The proposed algorithm identifies the single-source-points (SSPs) by comparing the absolute directions of the real and imaginary parts of the Fourier transform coefficient vectors of the mixtures. Then the hierarchical clustering technique is applied to these samples for the estimation of the mixing matrix. The proposed idea for the SSP identification is simpler than the previously reported algorithms.

The instantaneous noise-free mixing process can be mathematically expressed as:

$$\mathbf{x}(t) = \mathbf{H}\mathbf{s}(t) \quad (4.1)$$

where $\mathbf{x}(t) = [x_1(t), \dots, x_P(t)]^T$ are the P mixed signals, $\mathbf{H}$ is the real mixing matrix of order $P \times Q$ with h_{pq} as its $(p, q)^{\text{th}}$ element, $\mathbf{s}(t) = [s_1(t), \dots, s_Q(t)]^T$ are the Q sources, t is the time instant and T is the transpose operator.

A general review on underdetermined BSS is already given in Section.2.4. Hence, in this chapter the techniques reported mainly for the estimation of the mixing matrix are reviewed. Generally, the algorithms for the underdetermined case are also suitable for determined and overdetermined cases. This is true for the proposed algorithm also.

The idea of BSS based on TF representation was first reported by Belouchrani and Amin in [13]. The algorithm is for the separation of nonstationary sources in the overdetermined case (number of observations $>$ number of sources) based on joint-diagonalization of a set of Spatial Time Frequency Distributions (STFDs) of the whitened observations at selected TF locations. The algorithm is further extended in [131] such that it is also suitable for underdetermined cases, under the assumption that the sources are W-disjoint orthogonal in the TF domain. The idea has been further extended in [114] and [117]. In [132], the algorithm proposed in [13] is extended for the case of stochastic sources and a criterion is proposed for the selection of the points in the TF plane where the spatial matrices should be jointly diagonalized.

By utilizing sparsity in the TF domain, many algorithms have been proposed for blind source separation of underdetermined mixtures [18, 30, 23, 102, 113, 114, 115, 118, 117, 24, 1, 116, 133, 134, 131, 135, 136, 132]. In [135], the fact that, at SSPs, the directions of the modulus of the mixture vectors in the TF domain are the same as those of the column vectors of the mixing matrix is utilized to develop an algorithm called search-and-average-based method, which relaxes the degree of sparsity needed. The search-and-average-based algorithm for time domain signals is also proposed by the same authors in [136], where for the estimation of the mixing matrix, the algorithm removes the samples which are not in the same or opposite direction of the columns of the mixing matrix.

In [102], it is assumed that the signals are W-disjoint orthogonal in the TF plane,

i.e., only one source will occur in each TF window, which is quite restrictive. Later, it is shown that approximate W-disjoint orthogonality is sufficient to separate most speech signals and an algorithm called *Degenerate Unmixing Estimation Technique* (DUET) is proposed in [18]. Aïssa-El-Bey, et al. [114] relax the disjoint orthogonality constraint but assume that at any time the number of active sources in the TF plane is strictly less than the number of mixtures. The algorithm proposed in [24] also assumes that the maximum number of active sources at any instant is less than the number of mixtures. In [113, 116], these constraints are again relaxed with the only requirement that each source occurs alone in a *tiny set of adjacent TF windows* while several sources may co-exist everywhere else in the TF plane. This method can therefore be used even when the sources overlap in most of the areas in the TF plane. The algorithm proposed in [113] is based on the complex ratio of the mixtures in the TF domain and it is called the *Time Frequency Ratio Of Mixtures* (TIFROM)¹ method. In the TF domain if only one source occurs in several adjacent windows, then the complex ratio of the mixtures in those windows will remain constant and it will take different values only if more than one source occurs. Hence identifying the area where this ratio remains constant is equivalent to identifying the SSPs. The constant complex ratios of the mixtures at the SSPs are called canceling coefficients and these canceling coefficients can be used for the estimation of the sources from their mixtures. The TIFROM algorithm is further improved in [137].

One of the problems with the TIFROM method is its performance degradation because of the inconsistent estimation of the mixing system. This inconsistency is due to the fact that the TIFROM algorithm uses a series of minimum variances of the ratios of the mixed signals in the TF domain taken over the selected windows for the estimation of the column vectors of the mixing matrix. The absolute values of these variances monotonically increase with the increase in the mean of the corresponding ratios or the corresponding columns of the mixing matrix. Since the TIFROM algorithm looks

¹A detailed review on DUET and TIFROM methods is available in Section 2.4

for the mean corresponding to the minimum variance, in cases where the column matrix and hence the ratios and the corresponding variances are high, the algorithm will end up with a wrong result as it will take the mean of the ratios corresponding to the smaller variance as the column of the mixing matrix. This problem is solved in [133] by normalizing the variances. Even though the normalization of the variances created uniformity, if the TF windows used for estimating one of the column of the mixing matrix is sparser than the TF windows used for estimating another column of the mixing matrix, the variance corresponding to the first case will be smaller than that of the second case [133]. This difference in variance may lead to mixing matrix estimation error. To solve this problem an algorithm based on k-means clustering is proposed in [133].

The restriction of the TIFROM algorithm, i.e., the requirement of single-source-zone, is further relaxed in [134] where it requires only two adjacent points in the same frequency bin with single source contributions for the estimation of the SSPs. In [134], the fact that at SSPs the mixture vectors in the TF domain are proportional in magnitude to one of the columns of the mixing matrix is used, i.e., $|\mathbf{X}(k, t)| \simeq \mathbf{h}_j |S_j(k, t)|$. Hence, the scatter diagram using the magnitude of the observed data in the TF domain will have a clear orientation towards the directions of the column vectors of the mixing matrix, if the sources are sufficiently sparse. In situation where the sources are not sufficiently sparse, the orientation of the scatter diagram will not be very clear. Under such a situation, the estimation of the directions of the columns of the mixing vectors will be difficult. Now, at points (k, t) and $(k, t + 1)$ in the TF plane of the mixtures, if more than one source component is present, the directions of the mixture vectors $\mathbf{X}(k, t)$ and $\mathbf{X}(k, t + 1)$ will be the same only if the amplitudes of all the sources remain the same at both the points (k, t) and $(k, t + 1)$, i.e., at two consecutive time frames. Since this condition is very unlikely to happen, the mixture vectors $\mathbf{X}(k, t)$ and $\mathbf{X}(k, t + 1)$, $\forall t$, which keep the directions the same can be considered as SSPs. Utilizing this fact, in [134] the points which satisfy the condition (4.2), i.e.,

$$|\angle |\mathbf{X}(k, t)| - \angle |\mathbf{X}(k, t + 1)|| < \theta_{\text{th}} \quad (4.2)$$

where $\angle \mathbf{z}$ represents the angle of the vector $\mathbf{z}$ and θ_{th} is the threshold angle, are selected as the SSPs. Then the histogram method is applied to the estimated SSPs for the estimation of the mixing matrix.

In [1], a two stage approach consisting of mixing matrix estimation followed by source estimation is proposed. The restriction of the necessary condition for TIFROM algorithm, i.e., the requirement of some adjacent TF regions where only one of the sources occurs, is relaxed in [1]. The mixing matrix estimation procedure proposed in [1] is an extension of the DUET [18] and TIFROM [113] methods. Since the constraint of the proposed algorithm is the same as that of [1], in Section.4.3.1, the performance of the proposed algorithm and that in [1] are compared. Hence, for readers' convenience, the algorithm proposed in [1] is briefly explained below.

- Step 1: Convert the time domain mixed signals, $\mathbf{x}$, into TF domain. Store the mixture vectors at every point in the TF domain into $\mathbf{X}$ as column vectors.
- Step 2: From $\mathbf{X}$ remove all the column vectors whose norms are smaller than a predetermined constant, ζ_1 , to obtain $\tilde{\mathbf{X}}$. This step is to reduce the influence of noise and computational cost.
- Step 3: For $p_1 = 1$ to P , where P is the number of sensors, repeat Steps 3.1 and 3.2.
- Step 3.1: If the absolute value of any entry in the p_1^{th} row is less than the predetermined constant ζ_2 , remove that column. Now, suppose there are K_1 columns left in $\tilde{\mathbf{X}}$ and denoting the column indices as $n_1, \dots, n_{K_1}$. Next construct a new

ratio matrix using these K_1 columns of $\tilde{\mathbf{X}}$ as

$$\tilde{\mathbf{X}} = \begin{bmatrix} \frac{\tilde{\mathbf{X}}_1(n_1)}{\tilde{\mathbf{X}}_{p_1}(n_1)} & \cdots & \frac{\tilde{\mathbf{X}}_1(n_{K_1})}{\tilde{\mathbf{X}}_{p_1}(n_{K_1})} \\ \vdots & \vdots & \vdots \\ \frac{\tilde{\mathbf{X}}_P(n_1)}{\tilde{\mathbf{X}}_{p_1}(n_1)} & \cdots & \frac{\tilde{\mathbf{X}}_P(n_{K_1})}{\tilde{\mathbf{X}}_{p_1}(n_{K_1})} \end{bmatrix} \quad (4.3)$$

- Step 3.2: For $p_2 = 1$ to P , $p_1 \neq p_2$, repeat Steps 3.2.1, 3.2.2 and 3.2.3.
- Step. 3.2.1: Find the minimum, $\tilde{r}_{p_2}$, and maximum, $\tilde{R}_{p_2}$, of the p_2^{th} row of $\tilde{\mathbf{X}}$. Then divide the range $\tilde{r}_{p_2}$ to $\tilde{R}_{p_2}$ into M_0 equal intervals (bins), where M_0 is a predetermined positive integer. The matrix $\tilde{\mathbf{X}}$ is then divided into M_0 sub-matrices, denoted as $\tilde{\mathbf{X}}_1, \dots, \tilde{\mathbf{X}}_{M_0}$ such that all the entries in the p_2^{th} row of $\tilde{\mathbf{X}}_k$ are from the k^{th} bin, $k = 1, \dots, M_0$.
- Step 3.2.2: From the set of sub-matrices, remove the sub-matrices with the number of columns smaller than J_1 , where J_1 is a chosen positive integer, to obtain new sub-matrices $\tilde{\mathbf{X}}_j$, $j = 1, \dots, N_1$.
- Step 3.2.3: For $p_3 = 1$ to P , $p_3 \neq p_1$ and $p_3 \neq p_2$, repeat steps a1, a2 and a3 for every matrix $\tilde{\mathbf{X}}_j$, $j = 1, \dots, N_1$.
- Step a1: For sub-matrix $\tilde{\mathbf{X}}_j$, perform the step similar to Step 3.2.1, where p_2 is to be replaced by p_3 and $\tilde{\mathbf{X}}$ by $\tilde{\mathbf{X}}_j$. Let the M_0 sub-matrices so obtained be $\tilde{\mathbf{X}}_i^j$, $i = 1, \dots, M_0$.
- Step a2: From the set of sub-matrices, $\tilde{\mathbf{X}}_i^j$, $i = 1, \dots, M_0$, remove the sub-matrices with the number of columns smaller than J_2 , which is a chosen positive integer. From the new set of sub-matrices, select a matrix, $\tilde{\mathbf{X}}_p^j$, in such a way that the sum of the variances of the P rows is the smallest.
- Step a3: Calculate the mean of the column vectors of $\tilde{\mathbf{X}}_p^j$ to obtain an estimated column vector, $\mathbf{e}_i$, of the mixing matrix $\mathbf{H}$.
- Step 4: After the completion of all the above loops, let the array of the estimated column vectors of the mixing matrix be $\mathbf{E} = [\mathbf{e}_1, \dots, \mathbf{e}_{N_0}]$. Since there are several loops above, $\mathbf{E}$ may contain more columns than in $\mathbf{H}$, i.e, some of the columns in

E will be equal. To remove the duplication of columns in E, the direction of each column vector in E is calculated. If there are multiple column vectors which are almost parallel in direction, those vectors are replaced by their mean direction vector followed by normalization. Finally, the matrix obtained is taken as the estimate of the mixing matrix.

It can be seen that the main objective in all these algorithms is the detection of the points in the TF domain where only one source occurs at a time. In this chapter, a simple algorithm is proposed to identify these points and use them for the estimation of the mixing matrix using the hierarchical clustering algorithm which is well-known because of its versatility [138]. The proposed algorithm can be used for the mixtures where the sources are overlapped in the TF plane, except for some points. Unlike in [113] and [134], these SSPs need not be adjacent points in the TF domain and the proposed algorithm is simpler than that in [1], which requires many tuning parameters and a long procedure as explained above. The algorithms proposed in [113, 1] can be directly used for source estimation, either from the identified SSPs [113] or the estimated mixing matrix [113, 1].

This chapter is structured as follows. The proposed algorithm is derived in Section 4.2; in Section 4.3, some experimental results are given and finally the contributions are summarized in Section 4.4.

4.2 Proposed method

4.2.1 Single-source-point identification

The instantaneous noise-free mixing model in (4.1) can be expressed in the TF domain using short time Fourier transform (STFT) as:

$$\begin{aligned}\mathbf{X}(k, t) &= \mathbf{H}\mathbf{S}(k, t) \\ &= \sum_{q=1}^Q \mathbf{h}_q S_q(k, t)\end{aligned}\tag{4.4}$$

where $\mathbf{X}(k, t) = [X_1(k, t), \dots, X_P(k, t)]^T$ and $\mathbf{S}(k, t) = [S_1(k, t), \dots, S_Q(k, t)]^T$ are respectively the STFT coefficients of the mixtures and sources in the k^{th} frequency bin at time t and $\mathbf{h}_q = [h_{1q}, \dots, h_{Pq}]^T$ is the q^{th} column of the mixing matrix $\mathbf{H}$. For ease of explanation, assume that there are only two sources, i.e., $Q = 2$, and the number of mixtures is P . Now at any point in the TF plane, say (k_1, t_1) , if the source component from only one of the sources, say that of s_1 , is present, i.e., $S_1(k_1, t_1) \neq 0$ and $S_2(k_1, t_1) = 0$, equation (4.4) can then be written as

$$\mathbf{X}(k_1, t_1) = \mathbf{h}_1 S_1(k_1, t_1)\tag{4.5}$$

Now, from (4.5), the real and imaginary parts of $\mathbf{X}(k_1, t_1)$ can be written as

$$R\{\mathbf{X}(k_1, t_1)\} = \mathbf{h}_1 R\{S_1(k_1, t_1)\}\tag{4.6}$$

$$I\{\mathbf{X}(k_1, t_1)\} = \mathbf{h}_1 I\{S_1(k_1, t_1)\}\tag{4.7}$$

where $R\{x\}$ and $I\{x\}$ respectively represent the real and imaginary parts of x . From (4.6) and (4.7) it can be seen that the absolute directions² of $R\{\mathbf{X}(k_1, t_1)\}$ and $I\{\mathbf{X}(k_1, t_1)\}$ are the same as that of $\mathbf{h}_1$. Similarly, at another point, say (k_2, t_2) , if only the contribution from source s_2 is present, i.e., $S_1(k_2, t_2) = 0$ and $S_2(k_2, t_2) \neq 0$, then from (4.4)

$$R\{\mathbf{X}(k_2, t_2)\} = \mathbf{h}_2 R\{S_2(k_2, t_2)\}\tag{4.8}$$

²For finding the direction, the elements of the column vector can be consider as the terminal point and the initial point can always be taken as the origin. For example, for a vector $\mathbf{a} = [a_1 \ a_2]^T$, the initial point is $(0, 0)$ and the terminal point is (a_1, a_2) .

$$I \{ \mathbf{X}(k_2, t_2) \} = \mathbf{h}_2 I \{ S_2(k_2, t_2) \} \quad (4.9)$$

Hence at (k_2, t_2) the absolute direction of $R \{ \mathbf{X}(k_2, t_2) \}$ and $I \{ \mathbf{X}(k_2, t_2) \}$ are the same as that of $\mathbf{h}_2$. Now consider another point (k_3, t_3) where the contributions from both the sources are present. Then at (k_3, t_3) , the directions of $R \{ \mathbf{X}(k_3, t_3) \}$ and $I \{ \mathbf{X}(k_3, t_3) \}$ will be

$$R \{ \mathbf{X}(k_3, t_3) \} = \mathbf{h}_1 R \{ S_1(k_3, t_3) \} + \mathbf{h}_2 R \{ S_2(k_3, t_3) \} \quad (4.10)$$

$$I \{ \mathbf{X}(k_3, t_3) \} = \mathbf{h}_1 I \{ S_1(k_3, t_3) \} + \mathbf{h}_2 I \{ S_2(k_3, t_3) \} \quad (4.11)$$

Form (4.10) and (4.11) it can be seen that the absolute direction of $R \{ \mathbf{X}(k_3, t_3) \}$ will be the same as that of $I \{ \mathbf{X}(k_3, t_3) \}$ only if

$$\frac{R \{ S_1(k_3, t_3) \}}{I \{ S_1(k_3, t_3) \}} = \frac{R \{ S_2(k_3, t_3) \}}{I \{ S_2(k_3, t_3) \}} \quad (4.12)$$

However, in practice, the probability that the above condition is satisfied is very low. This fact is experimentally verified in Fig.4.2, where the mean of the percentage of the points in the TF plane which are below the *absolute value of difference between the ratios*, i.e.,

$$\left| \frac{R \{ S_1(k, t) \}}{I \{ S_1(k, t) \}} - \frac{R \{ S_2(k, t) \}}{I \{ S_2(k, t) \}} \right| \quad (4.13)$$

calculated for 15 pairs of speech utterances of length 10 s each is shown. For example, from Fig.4.2, there are only 0.3% of the total multi-source-points (MSPs) (i.e., the point in the TF plane of the mixture where more than one source occur) in the TF plane with difference between the ratios of less than 0.01, i.e., $\left| \frac{R \{ S_1(k, t) \}}{I \{ S_1(k, t) \}} - \frac{R \{ S_2(k, t) \}}{I \{ S_2(k, t) \}} \right| < 0.01$. It can also be seen from Fig.4.2 that the probability that the condition in (4.12) is satisfied

Fig. 4.1: Speech utterances used to plot the graph shown in Fig.4.2. Speech utterances s_n and s_{n+1} together constitute one pair, where $n \in \{1, 2, \dots, 15\}$. $s_1, s_2, \dots, s_{16}$ are obtained by concatenating the sentences taken from TIMIT database. The audio files are available in the accompanying CD

Fig. 4.2: Percentage of samples which are below the magnitude of the difference between the ratios of the real and imaginary parts of the DFT coefficient of the signals.

is almost zero ³. Hence, it can be concluded that, in practice, a point (k, t) in the TF plane of the mixture will be a SSP if the absolute direction of $R\{X(k, t)\}$ is the same as that of $I\{X(k, t)\}$; otherwise, it will be a MSP.

For a general case of P mixtures and Q sources, at a MSP (k, t) , the real and imaginary parts of $X(k, t)$ can be written as:

$$R\{X(k, t)\} = \sum_{q=1}^Q h_q R\{S_q(k, t)\} \quad (4.14)$$

$$I\{X(k, t)\} = \sum_{q=1}^Q h_q I\{S_q(k, t)\} \quad (4.15)$$

³This fact will be clearer by expressing the relation (4.4) in terms of Discrete Cosine Transform (DCT) and Discrete Sine Transform (DST); expressing the convolution operation using DCT and DST, and then particularizing it for instantaneous mixing. This is given in an Appendix; in Appendix A, the relation for circular convolution in terms of DCT and DST is derived and in Appendix B, SSPs estimation method in Discrete Trigonometric Transform (DTT) domain is explained.

Now, the angle between (4.14) and (4.15) is given by;

$$\begin{aligned}\theta &= \cos^{-1} \left(\frac{R\{\mathbf{X}(k, t)\}^T I\{\mathbf{X}(k, t)\}}{\sqrt{R\{\mathbf{X}(k, t)\}^T R\{\mathbf{X}(k, t)\}} \sqrt{I\{\mathbf{X}(k, t)\}^T I\{\mathbf{X}(k, t)\}}} \right) \\ &= \cos^{-1} \left(\frac{\sum_{p=1}^P \left(\left(\sum_{q=1}^Q h_{pq} R\{S_q(k, t)\} \right) \left(\sum_{q=1}^Q h_{pq} I\{S_q(k, t)\} \right) \right)}{\sqrt{\sum_{p=1}^P \left(\sum_{q=1}^Q h_{pq} R\{S_q(k, t)\} \right)^2} \sqrt{\sum_{p=1}^P \left(\sum_{q=1}^Q h_{pq} I\{S_q(k, t)\} \right)^2}} \right)\end{aligned}\quad (4.16)$$

In the above equation, θ will become 0° or 180° if

$$\frac{R\{S_1(k, t)\}}{I\{S_1(k, t)\}} = \dots = \frac{R\{S_q(k, t)\}}{I\{S_q(k, t)\}} = \dots = \frac{R\{S_Q(k, t)\}}{I\{S_Q(k, t)\}} \quad (4.17)$$

Hence, for the absolute directions of $R\{\mathbf{X}(k, t)\}$ and $I\{\mathbf{X}(k, t)\}$ to be the same, at any point (k, t) in the TF plane, either the point must be a SSP or the ratios between the real and imaginary parts of the Fourier transform coefficients of all the signals at that points must be the same. However, as shown previously, the probability for the second case is extremely low and this probability will decrease as the number of sources increases. Hence, it can be concluded that SSPs in the TF plane are the points where the absolute direction of $R\{\mathbf{X}(k, t)\}$ is the same as that of $I\{\mathbf{X}(k, t)\}$.

The probability of getting SSPs where the amplitudes of all the source contributions except one are exactly equal to zero is very low in a practical situation, particularly in noise. Hence, the condition for SSP is relaxed as the point in the TF plane where the component of one of the sources is significantly higher than that of the remaining sources. As a result, the point in the TF plane where the difference between the absolute directions of $R\{\mathbf{X}(k, t)\}$ and $I\{\mathbf{X}(k, t)\}$ is less than $\Delta\theta$ is taken as SSP, i.e., SSPs are the points in the TF plane where the following condition is satisfied:

Table 4.1: Algorithm for the detection of the single-source-points

Step 1:	Convert $\mathbf{x}$ in the time domain to the TF domain to get $\mathbf{X}$.
Step 2:	Check the condition in (4.18).
Step 3:	If the condition in (4.18) is satisfied, then $\mathbf{X}(k, t)$ is a sample at the SSP, and this sample is kept for mixing matrix estimation; otherwise, discard the point.
Step 4:	Repeat Steps 2 to 3 for all the points in the TF plane or until sufficient number of SSPs are obtained.

$$\left| \frac{R\{\mathbf{X}(k, t)\}^T I\{\mathbf{X}(k, t)\}}{\|R\{\mathbf{X}(k, t)\}\| \|I\{\mathbf{X}(k, t)\}\|} \right| > \cos(\Delta\theta) \quad (4.18)$$

where $|\cdot|$ represent the absolute value and $\|\mathbf{y}\| = \sqrt{\mathbf{y}^T \mathbf{y}}$. Samples at these SSPs are used for the clustering algorithm in Section 4.2.2. The algorithm to locate the SSPs in the TF plane is summarized in Table.4.1.

4.2.2 Mixing matrix estimation

After identifying the SSPs in the TF plane, the next stage is the estimation of the mixing matrix. Here the hierarchical clustering technique [138, 139] is used for the estimation of the mixing matrix⁴. The main contribution of this chapter is the efficient algorithm proposed in Section.4.2.1 for the detection of SSPs. For the mixing matrix estimation, by clustering, the real and imaginary parts of $\mathbf{X}(k, t)$ at the SSPs in the TF plane are stacked into an array, $\tilde{\mathbf{X}}$, and this array is used as the input for clustering. It can be seen that either the real or imaginary parts of the sample vectors at the SSPs are sufficient for clustering as the absolute directions of $R\{\mathbf{X}(k, t)\}$ and $I\{\mathbf{X}(k, t)\}$ are the same, except for a difference of maximum $\Delta\theta$. See Section 4.3 for more explanation.

For hierarchical clustering, $1 - |\cos(\theta)|$ is used as the distance measure, where

⁴It may be noted that this may not be the best algorithm to cluster the samples as other clustering algorithms are also available [117]. A detailed review on clustering algorithm can be found in [139] and the references therein.

$\cos(\theta) = \tilde{\mathbf{X}}_m^T \tilde{\mathbf{X}}_n / (||\tilde{\mathbf{X}}_m|| ||\tilde{\mathbf{X}}_n||)$ is the cosine of the angle between the m^{th} and n^{th} sample vectors (column vectors) $\tilde{\mathbf{X}}_m$ and $\tilde{\mathbf{X}}_n$ respectively in $\tilde{\mathbf{X}}$. This clustering is illustrated with a simple example in Fig.4.3, where the scatter diagram of the data and its dendrogram are shown. To get a clear idea about the clustering algorithm used, Matlab code (only up to the hierarchical tree generation) is also provided in Table.4.2. In hierarchical clustering, data are partitioned into different clusters by cutting the dendrogram at suitable distance, as shown in Fig.4.3. If the data contain outliers, the selection of the distance (equivalently the selection of the number of clusters) is important. For example in Fig.4.3, division of the dendrogram into two clusters will give wrong result as one of the cluster will contain only one point (point 15), which is the outlier, and the remaining points will be in the second cluster. In this particular case, the dendrogram has to be divided into three clusters and the cluster with the least number of samples has to be discarded so that the outliers will not be present. Automatic selection of the number of clusters without any knowledge about the data is difficult⁵. Hence, it is assumed here that out of the valid clusters (if there are Q sources, there must be Q valid clusters), the cluster with the minimum number of samples will contain at least 5% of the average number of samples in the remaining valid clusters. It is also assumed that the maximum number of outliers is less than 5% of the total number of samples in the valid clusters. Hence in the algorithm for cutting the dendrogram to form clusters, the dendrogram is first cut at a suitable height to form Q clusters and if the clusters do not satisfy the above conditions, the dendrogram will be cut at another height to form $Q + 1$ clusters. This process is repeated until the above condition is satisfied or when the maximum number of clusters is equal to twice the number of sources. In any of the experiments in this chapter, the total number of clusters never exceeded $2Q$.

Since $\tilde{\mathbf{X}}$ contains only the samples at SSPs, the scatter plot will have a clear ori-

⁵It may be noted that there are some advanced techniques for the automatic estimation of the number of sources, e.g.,[117]

Table 4.2: Matlab code for the clustering algorithm

```

Y = pdist(Xtilde, 'cosine');
Y = 1-abs(1-Y);
Z = linkage(Y, 'average');
T = cluster(Z, 'maxclust', C);

```

entation towards the directions of the column vectors in the mixing matrix, as shown in Fig.4.4, and hence the points in $\tilde{\mathbf{X}}$ will cluster into Q groups. After clustering, the column vectors of the mixing matrix are determined by calculating the centroid of each cluster. The points lying in the left hand side of the vertical axis in the scatter diagram (for two mixture case) are mapped to the right hand side (by changing their sign) before calculating the centroid; otherwise, very small value or zero will result.

To reduce further the mixing matrix estimation error, the points which are away from the mean direction of the cluster by $\epsilon\sigma_{\phi_q}$ are removed, where ϵ is a constant and σ_{ϕ_q} is the standard deviation of the directions of the samples in the q^{th} cluster. In other words, the i^{th} sample in the q^{th} cluster is removed if $|\phi_q(i) - \mu_{\phi_q}| > \epsilon\sigma_{\phi_q}$, where $\phi_q(i)$ is the absolute direction of the i^{th} sample in the q^{th} cluster and μ_{ϕ_q} is the mean of the absolute direction of the samples in the q^{th} cluster. This is illustrated in Fig.4.4.

4.3 Experimental Results

In all the experiments in this chapter except for the cluster diagram and Section 4.3.1, the average of the performances obtained for 100 randomly selected combinations of speech utterances (from the set of first 11 speech utterances, s_1 to s_{11} , shown in Fig.4.1) which are not sparse in the time domain is used. The other experimental conditions are: sampling frequency 16 kHz, STFT size 1024, Hanning window as the weighting function and $\epsilon = 0.5$.

To show that the proposed algorithm is effective in identifying the SSPs and hence in estimating the mixing matrix, six speech utterances are mixed using the mixing

Fig. 4.3: Illustration of hierarchical clustering: (a) scatter diagram of the two dimensional data to be clustered (b) dendrogram generated for the data taking $1 - |\cos(\theta)|$ as the distance measure, where θ is the angle between the vectors constituted by the sample and the origin.

Fig. 4.4: Scatter diagram of the mixtures taking samples from 40 frequency bins; $P = 2$; $Q = 6$; and $\Delta\theta = 0.8^\circ$ (a) all the DFT coefficients (b) samples at SSPs obtained by comparing the direction of $I\{\mathbf{X}(k, t)\}$ with that of $I\{\mathbf{X}(k, t)\}$ (c) samples at SSPs obtained after elimination of the outliers.

Table 4.3: Experimental Conditions

Source signals	Speech of 10sec (obtained by concatenating the sentences from TIMIT database)
Sampling rate f_s	16kHz
DFT size	$K = 1024$
Window function	Hanning window
ϵ	0.5

matrix

$$\mathbf{H} = \begin{bmatrix} 0.0872 & 0.3420 & 0.7071 & 0.9848 & 0.8660 & 0.5000 \\ 0.9962 & -0.9397 & -0.7071 & -0.1736 & 0.5000 & 0.8660 \end{bmatrix} \quad (4.19)$$

The scatter diagram in Fig.4.4 clearly shows the effectiveness of the proposed method for selecting the SSPs, which are in the direction of the column vectors of the mixing matrix, and rejecting the other points. The mixing matrix estimation error obtained is:

$$\mathbf{H} - \hat{\mathbf{H}} = \begin{bmatrix} -0.0020 & 0.0049 & 0.0032 & -0.0005 & 0.0007 & 0.0056 \\ 0.0002 & 0.0018 & 0.0032 & -0.0029 & -0.0012 & -0.0032 \end{bmatrix}$$

where $\hat{\mathbf{H}}$ is the estimated mixing matrix, which corresponds to -47.61dB normalized mean square error (NMSE). The NMSE in dB is defined as:

$$\text{NMSE} = 10 \log_{10} \left(\frac{\sum_{p,q} (\hat{h}_{pq} - h_{pq})^2}{\sum_{p,q} (h_{pq})^2} \right) \quad (4.20)$$

where $\hat{h}_{pq}$ is the $(p,q)^{\text{th}}$ element of the estimated matrix $\hat{\mathbf{H}}$. Since the number of samples to be used for clustering and estimating the mixing matrix is significantly reduced compared to using all the DFT coefficients, the computational time and memory requirement for the clustering algorithm are also reduced. For hierarchical clustering the computational complexity is $O(N^2)$, where N is the number of samples to be clustered [139]. Generally, in the TF domain, the samples having very small values dominate and these samples can be removed without much impact on the

Fig. 4.5: Mixing matrix estimation error before (dotted lines) and after (solid lines) elimination of the outliers from the initial estimated samples at SSPs for various values of $\Delta\theta$; $P = 2$ and $Q = 6$.

mixing matrix estimation error. In all the experiments, except where it is mentioned, samples in the TF domain having magnitude below 0.25 (i.e., $\|R\{\mathbf{X}(k, t)\}\| < 0.25$) are removed.

The advantage of elimination of the outliers from the samples at SSPs estimated by comparing the absolute directions of $R\{\mathbf{X}(k, t)\}$ and $I\{\mathbf{X}(k, t)\}$ is illustrated in Fig.4.5, where the q^{th} column vector of the mixing matrix $\mathbf{H}$ is $[\cos(\theta_q), \sin(\theta_q)]^T$, with $\theta_q = \left(\frac{-\pi}{2.4} + \frac{(q-1)\pi}{6}\right)$ and $q = 1, 2, \dots, 6$. In Fig.4.5, the mixing matrix estimation error when the initial samples at SSPs obtained by comparing the absolute directions of $R\{\mathbf{X}(k, t)\}$ and $I\{\mathbf{X}(k, t)\}$ is shown with dotted lines and those obtained after eliminating the outliers as explained in Section.4.2.2 and recalculating the centroid is shown with

Fig. 4.6: Mixing matrix estimation error before and after re-clustering the outlier-free samples for various values of $\Delta\theta$; $P = 2$ and $Q = 6$.

solid lines.

In Fig.4.6, the mixing matrix estimation error obtained by recalculating the centroids of the clusters after eliminating the outliers is compared with that obtained by re-clustering the outlier-free samples. It can be seen from the figure that there is no advantage in re-clustering samples after eliminating the outliers. Since the SSPs are identified by comparing the absolute direction of $R\{\mathbf{X}(k, t)\}$ with that of $I\{\mathbf{X}(k, t)\}$, at SSPs the maximum difference in direction between the two vectors will only be $\Delta\theta$. Hence there will not be much difference in performance even if $R\{\mathbf{X}(k, t)\}$ or $I\{\mathbf{X}(k, t)\}$ alone is used instead of both. This is illustrated in Fig.4.7 where the variation of mixing matrix estimation error for different values of $\Delta\theta$, when $R\{\mathbf{X}(k, t)\}$ alone (solid lines) and $R\{\mathbf{X}(k, t)\}$ together with $I\{\mathbf{X}(k, t)\}$ (dotted lines) are used as the data for

Fig. 4.7: Comparison of mixing matrix estimation error when samples at SSPs from $R\{\mathbf{X}(k, t)\}$ alone is used with that when samples at SSPs from both $R\{\mathbf{X}(k, t)\}$ and $I\{\mathbf{X}(k, t)\}$ are used, for various values of $\Delta\theta$; $P = 2$ and $Q = 6$.

clustering, as a function of the total number of frequency bins taken, is shown.

In all the experiments in this chapter, the frequency bins corresponding to one mixture, $\mathbf{x}_1$, were sorted in the descending order of their variance and the order of the frequency bins of other mixtures were modified according to that of $\mathbf{x}_1$ before starting the SSP detection. This is because most of the energy will be concentrated in nearly 10% of the frequency bins [118] and by sorting, the unnecessary computation in the frequency bins where the energy is low can be avoided. From Figs.4.5, 4.6, 4.7 and 4.9, it is clear that with a properly selected $\Delta\theta$, only 2 to 4% of the frequency bins are sufficient to obtain an accurate estimate of the mixing matrix. When the number of sources are fewer, the first few bins will be sufficient to obtain an accurate estimate

of the mixing matrix because the number of SSPs will increase as the number of sources decreases [114].

The case of $P = 3$, $Q = 6$ with a randomly selected mixing matrix

$$\mathbf{H} = \begin{bmatrix} 0.6330 & 0.7650 & 0.0612 & -0.7455 & -0.1988 & -0.6284 \\ 0.5179 & -0.2892 & -0.8156 & 0.3364 & -0.8156 & -0.5201 \\ 0.5754 & 0.6843 & 0.5621 & 0.4994 & -0.7589 & -0.5804 \end{bmatrix}$$

is illustrated in Fig.4.8 and the performance is shown in Fig.4.9. In Fig.4.9 the error in mixing matrix estimation obtained when all the samples in $R\{\mathbf{X}\}$ are used is also shown, where the same procedure described in Section.4.2.2 is used for the mixing matrix estimation.

4.3.1 Comparison with other algorithms

Determined case

The performance of the proposed algorithm and those of several classical algorithms using the *ICALAB Ver 3* toolbox available at [140] are compared. The proposed algorithm is compared with the following algorithms:

- AMUSE – Algorithm for Multiple Unknown Source Extraction based on EVD [141, 142, 30].
- EVD2 – Second order statistics BSS algorithm based on symmetric Eigen Value Decomposition [143, 144].
- SOBI – Second Order Blind Identification [145, 11, 146, 147].
- SOBI-RO – Robust SOBI with Robust Orthogonalization [148, 30].
- SOBI-BPF – Robust SOBI with bank of Band-Pass Filters [149, 150, 151].
- SONS – Second Order Nonstationary Source Separation [152, 153].
- JADE-OP – Robust Joint Approximate Diagonalization of Eigen matrices (with optimized numerical procedures) [144, 154].
- JADE-TD – HOS Joint Approximate Diagonalization of Eigen matrices with Time Delays [144, 155, 30]

Fig. 4.8: Scatter diagram of the mixtures taking samples from 40 frequency bins; $P = 3$; $Q = 6$; and $\Delta\theta = 0.8^\circ$ (a) all the DFT coefficients (b) samples at SSPs after elimination of the outliers.

Fig. 4.9: Comparison of NMSE on estimation of the mixing matrix using all the DFT coefficients in the TF plane with that using the estimated SSPs; $P = 3$; $Q = 6$; and $\Delta\theta = 0.8^\circ$.

- FPICA - Fixed-Point ICA [156, 157, 2].
- SANG - Self Adaptive Natural Gradient algorithm with nonholonomic constraints [158, 159, 31].
- NG-FICA -Natural Gradient Flexible ICA [160, 38].
- THIN-ICA - ThinICA Algorithm [156, 161, 162, 11, 84].
- ERICA - Equivariant Robust ICA - based on Cumulants [163, 164].
- SIMBEC - SIMultaneous Blind Extraction using Cumulants [165, 166, 167, 168].
- UNICA - Unbiased quasi Newton algorithm for ICA [169].

In this experiment the separation performance of each of the algorithms is obtained for five pairs of speech utterances (from the set of the first 6 speech utterances,

s_1 to s_6 , shown in Fig.4.1 and each has a length of 10 s) as well as for each pair, the performance is obtained for 100 randomly selected 2×2 real mixing matrices. The mean Signal-to-Interference Ratios (SIR) in dB, so obtained for the different algorithms, are shown in Fig.4.10. From the figure, it can be seen that the proposed algorithm outperforms the other classical algorithms. Here, the proposed algorithm is compared with other classical algorithms developed for determined case because for determined case the mixing matrix is square. Since the mixing matrix is square, the separated signals can be calculated by multiplying the mixed signals by the inverse of the mixing matrix. Hence, the separation performance is determined by the estimated mixing matrix only. For the underdetermined case, since the unmixing matrix cannot be estimated by calculating the inverse of the mixing matrix, the error introduced by the signal reconstruction stage will influence the final separation performance. It may also be noted that most of the algorithms developed for the determined case can be applied directly on the mixture in the time domain and the separation performance above say 20dB is practically not required. However, these algorithms cannot be used for the underdetermined case. Moreover, the performance of most of these algorithms will deteriorate as the number of sources increases and therefore the additional computational cost of the proposed algorithm is justified.

Underdetermined case

For the underdetermined case, the proposed algorithm is compared with one of the recently reported algorithms. The algorithm presented in [1] is an extension of the DUET and TIFROM algorithms. Unlike the case of the DUET method, the spectra of the sources can overlap in the TF domain, i.e, the W-disjoint orthogonality condition need not be met. Furthermore, unlike the case of the TIFROM algorithm, the 'single source region' is also not needed. This is true for the proposed algorithm also. Hence the proposed algorithm is compared with the algorithm reported in [1]. Here 12 experiments (3-sensor, 4-sensor and 5-sensor cases each for 4 to 7 sources) are

Fig. 4.10: Comparison of the proposed algorithm with classical algorithms for determined case, $P = Q = 2$

conducted as shown in Fig.4.11. Each experiment is repeated with 100 different randomly generated mixing matrices and the mean NMSEs obtained are shown in Fig.4.11. From the figure, it can be seen that the proposed algorithm outperforms that in [1] in all the cases.

Both the algorithms are implemented in the STFT domain and the number of frequency bins used for both the algorithms are the same. To decide the number of frequency bins to be used, for each experiment, the number of frequency bins is increased until the proposed algorithm detects a minimum of 1000 SSPs.

For the proposed algorithm, the magnitudes of the real parts of the mixture vectors at any point (k, t) which are less than 5% of the maximum magnitude of all the vectors in the TF plane, i.e., the points with $\|R\{\mathbf{X}(k, t)\}\| < 0.05 \max(\|R\{\mathbf{X}\}\|)$, are discarded.

Fig. 4.11: Comparison of the proposed algorithm with that proposed in [1]

For comparisons, the proposed hierarchical clustering algorithm is used to cluster the estimated column vectors stored in matrix E (please refer [1]) according to their directions. The other parameters used are the same as those used in [1], i.e., number of sub-matrices $M0 = 400$, minimum number of columns in the sub-matrices $J1 = J2 = 100$ (please refer to [1] for more details). In cases where the algorithm fails to identify sufficient number of sub-matrices, $M0$, $J1$ and $J2$ are divided by two to obtain their new values and the experiment is repeated using the new values.

4.4 Summary

In this chapter, a simple and effective algorithm for single-source-point identification in the TF plane of the mixture signals for the estimation of mixing matrix in un-

derdetermined blind source separation is developed. The algorithm can be used for the mixtures where the spectra of the sources overlap and the single-source-points occur only at a small number of locations. The proposed algorithm does not have any restriction on the numbers of sources and mixtures, and the single-source-points need not be in adjacent locations in the TF plane. Since only the samples at single-source-points are used for the clustering algorithm for the estimation of the mixing matrix, the estimation error, computation time and memory requirement are reduced as compared to using all the samples in the TF plane.

Chapter 5

Underdetermined Convolutional Blind Source Separation via Time-Frequency Masking

5.1 Introduction

In Chapter 4, an algorithm for the estimation of the mixing matrix for the separation of the sources from their underdetermined instantaneous mixtures is proposed. In this chapter, the problem of separation of unknown number of sources from their underdetermined convolutional mixtures via time-frequency (TF) masking is considered.

The problem of underdetermined convolutional blind source separation has been addressed by many researchers [27, 28, 29, 25]. Convolutional mixing of the signals can be mathematically expressed as

$$x_p(n) = \sum_{q=1}^Q \sum_{l=0}^{L-1} h_{pq}(l) s_q(n-l) \quad (5.1)$$

where $p = 1, \dots, P$, $q = 1, \dots, Q$, P is the number of mixtures, Q is the number of sources, L is the length of the mixing filters, $\mathbf{x} = [x_1, x_2, \dots, x_P]^T$ are the P sensor outputs, T is the transpose operator, $x_p = [x_p(0), \dots, x_p(N-1)]^T$ are the mixture samples at the p^{th} sensor output, N is the total number of samples, $\mathbf{s} = [s_1, s_2, \dots, s_Q]^T$ are the sources, $s_q = [s_q(0), \dots, s_q(N-1)]^T$ are the samples of the q^{th} source and $h_{pq}(l)$, $l = 0, \dots, L-1$ is the impulse response from the q^{th} source position to the p^{th} sensor.

Using the convolution-multiplication property, the mixing process can be expressed in the TF domain as

$$\mathbf{X}(k, t) = \mathcal{H}(k) \mathbf{S}(k, t) = \sum_{q=1}^Q \mathbf{H}_q(k) S_q(k, t) \quad (5.2)$$

where $\mathbf{X}(k, t) = [X_1(k, t), \dots, X_P(k, t)]^T$ is a column vector of the short time Fourier transform (STFT) [170] coefficients of the P mixed signals in the k^{th} frequency bin at time frame t , $\mathbf{S}(k, t) = [S_1(k, t), \dots, S_Q(k, t)]^T$, is the column vector of the STFT coefficients of the Q source signals, $\mathbf{H}_q(k) = [H_{1q}(k), \dots, H_{Pq}(k)]^T$ is the q^{th} column vector of the mixing matrix at the k^{th} frequency bin, i.e., $\mathcal{H}(k) = [\mathbf{H}_1(k), \dots, \mathbf{H}_Q(k)]$ is the mixing matrix at the k^{th} frequency bin and $H_{pq}(k)$ is the k^{th} DFT coefficient of the impulse response (or mixing filter) from the q^{th} source to the p^{th} sensor. Here, it is assumed that the impulse responses remain the same for all t . In this chapter, all the signals in the time domain are represented by small letters whereas signals in the frequency domain are represented by capital letters.

For underdetermined BSS of speech signals, the most widely used assumption is its disjoint orthogonality property in the TF domain [18]. Two speech signals s_1 and s_2 with supports Ω_1 and Ω_2 in the TF plane are said to be TF-disjoint if $\Omega_1 \cap \Omega_2 = \emptyset$. However, in practice the signals may not be perfectly disjoint. In [18], it is shown that for practical purposes an approximate disjoint orthogonality is sufficient for the separation of speech signals from their mixtures. The disjoint orthogonality property of the speech signals has been successfully utilized for the generation of binary masks which can be applied to the mixtures in the TF domain for the separation of the sources from their underdetermined convolutive mixtures [27, 28, 25, 29]. The techniques used for the estimation of masks in some of the recent papers are reviewed below.

The direction of arrival (DOA) information is utilized in [28] for the estimation of the binary masks and the signals are separated from their mixtures in two stages. For

the case of three sources and two mixtures demonstrated in [28], in the first stage, one of the sources is removed from the mixtures in the TF domain by locating the single-source-points (single-source-points are the points in the TF domain where only the component of one of the sources is present). In the second stage, the 2x2 ICA algorithm is applied to each of the frequency bins to separate the remaining sources from their mixtures. For the estimation of the binary masks utilizing omnidirectional microphones, the phase difference between the observations $X_1(k, t)$ and $X_2(k, t)$ is calculated as $\phi(k, t) = \angle \frac{X_1(k, t)}{X_2(k, t)}$. The DOA is then estimated at each time-frequency point by calculating $\theta_{DOA}(k, t) = \cos^{-1} \frac{\phi(k, t)c}{kd}$ where c is the velocity of sound in air and d is the spacing between the microphones. An histogram is then plotted using the DOAs, $\theta_{DOA}(k, t)$, $\forall t$, and the three peaks obtained from the histogram are taken as the DOAs of the three sources at that frequency. If these peaks are at θ_{DOA_1} , θ_{DOA_2} and θ_{DOA_3} , corresponding to the DOA of signals s_1 , s_2 and s_3 respectively, then the q^{th} signal can be extracted using the binary masks

$$M_q(k, t) = \begin{cases} 1 & \theta_{DOA_q} - \Delta \leq \theta_{DOA}(k, t) \leq \theta_{DOA_q} + \Delta \\ 0 & \text{otherwise} \end{cases} \quad (5.3)$$

i.e., $Y_q(k, t) = M_q(k, t)X_p(k, t)$, where $q = 1, 2, 3$; $p = 1$ or 2 and Δ is the extraction range parameter.

In [27], a two stage algorithm for the extraction of the dominant sources from their mixtures is proposed. The main assumption is that the total number of dominant sources is smaller than the number of microphones, but the number of dominant sources plus the interfering sources can be greater than the number of microphones. Thus, in the first stage, the frequency domain ICA algorithm is applied to the output of the microphones under the assumption that the number of independent components are equal to the number of microphones and in the second stage, time-frequency masking is used to improve the performance as the components separated by the ICA algorithm will contain some residuals caused by the interfering sources, when the

total number of sources is more than the number of microphones. After solving the permutation problem and estimating the number of sources in the first stage, binary masks are obtained based on the angles between the mixture sample vectors $\mathbf{X}(k, t)$ and the Fourier transform of the estimated mixing filters $\hat{\mathcal{H}}(k)$.

For the estimation of the binary masks, in [29], the impulse responses of the channels (i.e., the mixing filters) are estimated first. For the estimation of the mixing filters, it is assumed that the sources are sparse in the time domain so that the time interval during which only one of the sources is effectively present is estimated; then, for each estimated time interval the cross-correlation technique [171, 172] for the blind single input multiple output (SIMO) channel identification is applied. Since the single source intervals for the same source can exist at many different time slots, after estimation of the mixing filters, they are clustered into Q clusters using the k -means clustering algorithm. The centroids of the clusters are then taken as the estimated channel parameters. Under the assumption that the sources in their TF domain are disjoint, the spatial direction vectors, $\mathbf{v}(k, t) = \frac{\mathbf{X}(k, t)}{\|\mathbf{X}(k, t)\|}$, of the mixture at each point in the k^{th} frequency bin (after forcing the first entry of the spatial vector to be real and positive) are clustered into Q clusters by minimizing the criterion

$$\mathbf{v}(k, t) \in \mathcal{C}_i \Leftrightarrow i = \arg \min_q \left\| \mathbf{v}(k, t) - \frac{\hat{\mathbf{H}}_q(k) e^{-j\angle H_{q1}(k)}}{\|\hat{\mathbf{H}}_q(k)\|} \right\| \quad (5.4)$$

where $\mathcal{C}_i$ is the i^{th} cluster and $\hat{\mathbf{H}}_q(k)$ is the Fourier transform of the q^{th} channel vector estimate. The samples in each cluster are then taken as the samples corresponding to one source.

The main shortcoming with the algorithm proposed in [28] is that it requires the DOA of the sources. The accurate estimation of DOA is very difficult in a reverberant environment and when the sources are very close or collinear with the microphone array. For the algorithms in both [27] and [29], the approximate mixing parameters are to be estimated first. In [27], this is done using the ICA algorithm and hence

it cannot be used when the number of dominant (or the required) sources is more than the number of microphones. The channel estimation algorithm in [29] uses the assumption that the sources are sparse enough in the time domain for effective channel estimation.

Utilizing the concept of angles in complex vector space [173], a simple algorithm for the design of the separation masks which are used to separate the sources from their underdetermined convolutive mixtures under the assumption that the sources are sufficiently disjoint (sparse) in the TF domain is proposed in this chapter. Unlike the previously reported methods, the algorithm does not require any estimation of the mixing matrix or the source positions for mask estimation. The algorithm clusters the mixture samples in the TF domain based on the Hermitian angle between the sample vector and a reference vector using the well-known k -means or fuzzy c -means clustering algorithms. The membership functions so obtained from the clustering algorithms are directly used as the masks. In the TF masking approach, the proposed algorithm does not have the well-known scaling problem. However, it may be noted that the amplitudes of the separated signals may not be exactly equal to those of the original signals. Instead, they will be equal to those picked up by the microphones. Another advantage is that well-known clustering algorithms can be directly used and the membership function obtained from the clustering algorithms can be used as the mask. Also, the additional computational complexity in estimating the masks due to the increase in the number of microphones is very low. In addition to the TF masking method for the separation of the signals, an algorithm to solve the well-known permutation problem is also proposed. The algorithm is based on k -means clustering, where the estimated masks will be clustered to solve the permutation problem. Since the already available masks are used to solve the permutation problem, instead of using magnitude envelopes or power ratios of the separated signals, some computation time can be saved. A similar approach for solving the permutation problem is previously reported in [174], see Section 5.2.4 for a brief discussion on the difference between the

proposed algorithm and that in [174]. Unlike the conventional DOA based algorithms [93, 94, 68], the proposed algorithms for solving the permutation problem do not require any geometrical information of the source positions and hence can be used even when the sources are very close or collinear. The effectiveness of the algorithm in separating the sources, including collinear sources, from their underdetermined convolutive mixtures obtained in a real room environment, is demonstrated.

This chapter is organized as follows. In the next section the proposed algorithms for estimation of the masks and automatic detection of the number of sources, followed by the algorithm for solving the permutation problem, are described. The experimental results are given in Section 5.3. Finally, Section 5.4 summaries the chapter.

5.2 Proposed method

5.2.1 Basic idea

For ease of explanation, first consider the case of instantaneous mixing. For instantaneous mixing, the impulse responses will be single pulses of amplitude h_{pq} , where h_{pq} is the $(p, q)^{\text{th}}$ element of the mixing matrix. If the impulse response is a single pulse, the imaginary part of $H_{pq}(k)$ will be zero and the real part will be the same as h_{pq} , i.e., $I\{H_{pq}(k)\} = 0$ and $R\{H_{pq}(k)\} = h_{pq}$, $\forall k$. Hence $\mathbf{H}_q(k) = \mathbf{h}_q = [h_{1q}, \dots, h_{Pq}]^T$, $\forall k$, where $\mathbf{h}_q$ is the q^{th} column of the mixing matrix in the time domain and $\mathbf{H}_q(k)$ is the q^{th} column of the mixing matrix in the frequency domain at the k^{th} frequency bin. For ease of explanation assume that $P = Q = 2$. Now consider a point (k_1, t_1) in the TF plane where only the components of source s_1 is present. Then from (5.2)

$$\mathbf{X}(k_1, t_1) = \mathbf{H}_1(k_1)S_1(k_1, t_1) \quad (5.5)$$

This can be written as:

$$R\{\mathbf{X}(k_1, t_1)\} + jI\{\mathbf{X}(k_1, t_1)\} = \mathbf{H}_1(k_1) (R\{S_1(k_1, t_1)\} + jI\{S_1(k_1, t_1)\}) \quad (5.6)$$

Since $R\{S_1(t_1, k_1)\}$ and $I\{S_1(t_1, k_1)\}$ are real, comparing real and imaginary parts of (5.6), it can be seen that the direction of the column vectors $R\{\mathbf{X}(k_1, t_1)\}$ and $I\{\mathbf{X}(k_1, t_1)\}$ are the same and it is also the same as that of $\mathbf{H}_1(k_1)$, which is the same as that of the first column vector of the mixing matrix $\mathbf{h}_1$. Similarly, at another instant (k_2, t_2) , if only source s_2 is present, then

$$R\{\mathbf{X}(k_2, t_2)\} + jI\{\mathbf{X}(k_2, t_2)\} = \mathbf{H}_2(k_2) (R\{S_2(k_2, t_2)\} + jI\{S_2(k_2, t_2)\}) \quad (5.7)$$

Here the directions of both $R\{\mathbf{X}(k_2, t_2)\}$ and $I\{\mathbf{X}(k_2, t_2)\}$ are the same as that of $\mathbf{H}_2(k_2)$, which is the same as that of the second column vector of the mixing matrix $\mathbf{h}_2$. Hence if the sources are sparse in the TF domain, the scatter plot of both $R\{\mathbf{X}(k, t)\}$ and $I\{\mathbf{X}(k, t)\}$ will show a clear orientation towards the directions of the column vectors of the mixing matrix and once the directions are known, the mixing matrix can be determined and hence the sources can be estimated up to a scaling factor with permutation.

When the mixing is convolutive, the column vectors $\mathbf{H}_q(k)$ in (5.2) will be a complex column vector and multiplication of this complex vector by a complex scalar, $S_q(k, t)$, will change the complex-valued angle of the vectors. Hence the above approach, used for instantaneous mixing, cannot be directly applied for convolutive mixing. Now consider two complex vectors $\mathbf{u}_1$ and $\mathbf{u}_2$. The cosine of the complex-valued angle between $\mathbf{u}_1$ and $\mathbf{u}_2$ is defined as [173]

$$\cos(\theta_C) = \frac{\mathbf{u}_1^H \mathbf{u}_2}{\|\mathbf{u}_1\| \|\mathbf{u}_2\|} \quad (5.8)$$

where $\|\mathbf{u}\| = \sqrt{\mathbf{u}^H \mathbf{u}}$ and H represents the complex conjugate transpose operation.

$\cos(\theta_C)$ in (5.8) can be expressed as

$$\cos(\theta_C) = \rho e^{j\varphi} \quad (5.9)$$

where $\rho \leq 1$ [173]

$$\rho = \cos(\theta_H) = |\cos(\theta_C)| \quad (5.10)$$

In addition, $0 \leq \theta_H \leq \pi/2$ and $-\pi \leq \varphi \leq \pi$ are called the Hermitian and pseudo angle respectively between the vectors $\mathbf{u}_1$ and $\mathbf{u}_2$ [173]. The Hermitian angle between the complex vectors $\mathbf{u}_1$ and $\mathbf{u}_2$ will remain the same even if the vectors are multiplied by any complex scalars, whereas φ will change (see Appendix.C for proof). This fact can be used for the design of masks for the BSS of underdetermined convolutive mixtures as follows. Since multiplication of a complex vector by a complex scalar does not affect the Hermitian angle between the vector and another vector (reference vector), a P element vector $\mathbf{r}$, with all the elements equal to $1 + j1$ can be taken as the reference vector. The Hermitian angle between the reference vector $\mathbf{r}$ and $\mathbf{H}_q(k)$ will remain the same even if $\mathbf{H}_q(k)$ is multiplied by any complex scalar $S_q(k, t)$. If the signals s_q , $q = 1, \dots, Q$ are sparse in the TF domain, at any point in the TF plane only one of the source components will be present and the Hermitian angle between the reference vector and the mixture vectors $\mathbf{X}(k, t)$ at that point will be the same as that between $\mathbf{H}_q(k)$ corresponding to the source component $S_q(k, t)$ present at that point and the reference vector $\mathbf{r}$. Hence the mixture samples in each frequency bin, k , will form Q clusters with a clear orientation with respect to the reference vector and all the samples in one cluster will belong to the same source. It is not necessary to make all the elements of the reference vector equal to $1 + j1$. In fact, any random vector can be used. The only difference is that, for different reference vectors, the Hermitian angles between the reference vectors and $H_q(k)$, $q = 1, \dots, Q$ will be different whereas those between the column vectors $H_q(k)$, $q = 1, \dots, Q$ will remain the same, for a particular frequency bin. Finding the clusters is equalant to finding the samples which belong

to the sources corresponding to those particular clusters. In the following section this idea is illustrated with two sources and two sensors, i.e., $P = Q = 2$.

Assume that at point (k_1, t_1) only the contribution of source s_1 is present, i.e., $S_1(k_1, t_1) \neq 0$ and $S_2(k_1, t_1) = 0$. Let the reference vector be $\mathbf{r} = [1 + j1, 1 + j1]^T$. At point (k_1, t_1) the Hermitian angle $\Theta_H^{(k_1)}(t_1)$ between the reference vector $\mathbf{r}$ and the mixture vector $\mathbf{X}(k_1, t_1) = [X_1(k_1, t_1), X_2(k_1, t_1)]^T$ will be the same as that between $\mathbf{r}$ and $\mathbf{H}_1(k_1, t_1) = [H_{11}(k_1), H_{21}(k_1)]^T$. This angle, $\Theta_H^{(k_1)}(t_1)$, will be the same for all the points in the frequency bin k_1 , where only the component of the source s_1 is present. Similarly at another point, (k_1, t_2) , if $S_1(k_1, t_2) = 0$ and $S_2(k_1, t_2) \neq 0$, the Hermitian angle $\Theta_H^{(k_1)}(t_2)$ between $\mathbf{r}$ and $\mathbf{X}(k_1, t_2)$ will be the same as that between $\mathbf{r}$ and $\mathbf{H}_2(k_1) = [H_{12}(k_1), H_{22}(k_1)]^T$ and this will remain the same for all the points in the frequency bin k_1 where only the component of source s_2 is present. Hence among the calculated Hermitian angles between $\mathbf{r}$ and $\mathbf{X}(k_1, t)$, $\forall t$, depending on presence or absence of the components of the sources, there will be a clear grouping of the mixture vectors according to the Hermitian angles between the reference vector and the mixture vectors. This is demonstrated in Fig.5.1(a) where the Hermitian angle between the reference vector $\mathbf{r}$ and $\mathbf{H}_1(k)$ is 14.96° and that between $\mathbf{r}$ and $\mathbf{H}_2(k)$ is 29.40° for $k = 54$. In practice the signals in the TF domain may not be fully sparse, i.e., there may be instants where both the components of sources s_1 and s_2 are present. However, as demonstrated in [18] for the case of instantaneous mixing, for speech signals, approximate sparsity or disjoint orthogonality is sufficient for the separation of sources from their mixtures via binary masking.

For a general case of P mixtures and Q sources, the Hermitian angle between the reference vector $\mathbf{r}$ having P elements (say each element is $1 + j1$), and each of the mixture vectors in the k_1^{th} frequency bin, $\mathbf{X}(k_1, t)$, $\forall t$ is calculated, to obtain a vector of Hermitian angles, $\Theta_H^{(k_1)}$, where the value of $\Theta_H^{(k_1)}$ at t_1 is given by

$$\Theta_H^{(k_1)}(t_1) = \cos^{-1}(|\cos(\theta_C(k_1, t_1))|) \quad (5.11)$$

$$\cos(\theta_C(k_1, t_1)) = \frac{\mathbf{X}(k_1, t_1)^H \mathbf{r}}{\|\mathbf{X}(k_1, t_1)\| \|\mathbf{r}\|} \quad (5.12)$$

The Hermitian angle vector, $\Theta_H^{(k)}$, calculated for the frequency bin k is used for partitioning the mixture samples in the k^{th} frequency bin. The membership functions for the partitioning of the samples so obtained from the clustering algorithm are used as the mask, $M_q(k, t)$, $\forall t$, which will be multiplied by the mixture in the TF domain, $X_p(k, t)$, $\forall t$, to obtain the separated signal $Y_q(k, t)$, $\forall t$ in the TF domain, i.e.,

$$Y_q(k, t) = M_q(k, t)X_p(k, t), \forall t, q = 1, \dots, Q \quad (5.13)$$

where $p \in \{1, \dots, P\}$ is the index of the microphone output to which the mask is applied.

5.2.2 Clustering of mixture samples and mask estimation

The partitioning of the values of $\Theta_H^{(k)}$ and hence the corresponding mixture samples in the TF domain into different groups can be done using the well established data clustering algorithms [138, 139]. In this thesis, the use of two well-known clustering algorithms namely, k -means [139] and fuzzy c -means (FCM) [175] clustering algorithms for the partitioning of samples in $\Theta_H^{(k)}$, is examined. The k -means algorithm is a hard partitioning technique, which means that any sample in the data vector to be clustered will be fully assigned to any one of the clusters, i.e., the membership function will be binary (0 or 1). Hence if the membership function obtained from the k -means algorithm is taken as the mask, it will be a binary mask. On the other hand the FCM algorithm is a soft partitioning technique and hence the mask generated by FCM will be a smooth one compared to that from the k -means algorithm. In the following section the clustering and the mask estimation procedures using the k -means and fuzzy c -means algorithms are explained in detail.

Fig. 5.1: Masks generated by k -means clustering algorithm. (a) the plot of Hermitian angles $\Theta_H^{(k)}(t)$ (b) membership function (c) magnitude envelopes of the DFT coefficients in the k^{th} frequency bins of the signals picked up by the microphones (d) magnitude envelopes of the DFT coefficients in the k^{th} frequency bins of the separated signals.

k -means clustering

If the samples in the TF domain are perfectly sparse, the Hermitian angles $\Theta_H^{(k)}$ will contain only Q different values, each corresponding to a particular source and hence the samples can be partitioned perfectly without any ambiguity. However, in a real situation this may not be the case. Hence a clustering algorithm is to be used for partitioning the samples into different clusters. The Hermitian angles in degree, calculated for $k = 54$, $P = Q = 2$ are shown in Fig.5.1(a). From the figure, it is clear that most of the samples in $\Theta_H^{(k)}$, $\Theta_H^{(k)}(t)$, are either close to 14.96° or to 29.40° , which are the actual directions of the mixing vectors $\mathbf{H}_1(k)$ and $\mathbf{H}_2(k)$

respectively with respect to the reference vector $\mathbf{r}$. Using the k -means algorithm, the samples in $\Theta_H^{(k)}$ can be partitioned into 2 clusters. Since the k -means algorithm is a hard partitioning technique, each sample will belong to either one of the clusters and the membership function obtained will be binary (0 or 1). The direction of the estimated mixing matrix is the centroid of the angles corresponding to that particular cluster. Since the estimation of the signals are achieved by masking, the main interest here is on the estimated membership function which will be used as the mask. The membership functions obtained from k -means clustering are purely binary. To make them smoother, the samples away from the mean direction or centroid by $\Delta\phi$ are given the membership value $\cos(\Delta\phi)$. The membership functions so obtained are used as the mask, as shown in Fig.5.1(b), which are multiplied with the mixture samples obtained from one of the microphone outputs in the TF plane. Fig.5.1(c) is the magnitude envelope of the DFT coefficients of the clean signals picked up by the microphone on which the mask is applied. Fig.5.1(d) is the magnitude envelope of the estimated signals obtained by applying the mask on the mixture samples in the TF domain.

It is a well-known fact that the starting centroid of the k -means clustering algorithm will have an impact on the final centroid of the clusters [176]. Hence the k -means algorithm is initialized with the result obtained from the histogram method on $\Theta_H^{(k)}$, i.e., the k -means algorithm is initialized with the bin centers of the highest Q bins in the histogram. The algorithm starts with $\max(10, Q)$ bins and if any one of the highest Q bins are empty (this happens when the angle between the column vectors $\mathbf{H}_q(k)$, $q = 1, \dots, Q$ are very small), the number of bins are doubled to reduce the bin width and the histogram estimation is repeated. This process is repeated until none of the Q bins is empty.

Fuzzy c -means clustering

The k -means algorithm described in Section 5.2.2 is a hard partitioning method, and as a result of which the estimated signal will contain abrupt changes in their

Fig. 5.2: Masks generated by FCM clustering algorithm. (a) the plot of Hermitian angles $\Theta_H^{(k)}(t)$ (b) membership function (c) magnitude envelopes of the DFT coefficients in the k^{th} frequency bins of the signals picked up by the microphones (d) magnitude envelopes of the DFT coefficients in the k^{th} frequency bins of the separated signals.

amplitude as shown in Fig.5.1(d). These abrupt changes in the amplitude will introduce artifacts in the reconstructed signals in the time domain. To avoid this problem the use of the FCM clustering algorithm is examined. The FCM clustering partitions the samples into clusters with membership values which are inversely related to the distance of $\Theta_H^{(k)}(t)$ to the centroids of the clusters. For example, if a sample is equidistant from the estimated centroids of the clusters, the k -means clustering algorithm will assign that sample to one of the clusters, with membership value equal to 1 with respect to the cluster into which the sample is assigned and zero for the other clusters, i.e., the membership function will be binary. In the case of the FCM algorithm, for the same condition, the sample will be assigned to all the clusters

with equal membership values of $1/Q$, where Q is the number of clusters. The FCM algorithm when applied to the same frequency bin as that used in Section 5.2.2 is shown in Fig.5.2. From the figure it can be seen that the mask, which is the same as the membership function obtained from the FCM algorithm, is smooth and hence the magnitude envelope of the DFT coefficients of the estimated signals is also smooth. Consequently, it will reduce the artifacts in the reconstructed speech signals in the time domain. However, as shown in Section 5.3.1, the reduction in artifacts is at the cost of a reduction in signal-to-interference ratio (SIR).

5.2.3 Automatic detection of the number of sources

In the previous section, it is assumed that the total number of sources is known in advance. However, in a practical situation this may not be the case. Hence it is necessary to estimate the number of sources present in the mixture before clustering $\Theta_H^{(k)}$ for the mask estimation, i.e, the number of clusters in $\Theta_H^{(k)}$ is to be estimated. Many algorithms are available in the literature for the estimation of the number of clusters [177, 178, 179, 180]. One commonly used technique is the cluster validation technique. This technique require some knowledge about the possible maximum number of clusters. Then the data are clustered for different number of clusters, c , $c = 2, \dots, c_{\max}$, where $c_{\max}$ is the possible maximum number of clusters. The clusters so obtained for different values of c are validated using the cluster validation technique [177, 178, 179] and the number of clusters in the best cluster is taken as the actual number of clusters. In this thesis a recently reported cluster validation technique [178] for the estimation of the number of clusters is used. Since the data to be clustered are one dimensional, the validation index proposed in [178] for multidimensional data can be simplified as

$$\text{Validation index } V(U, \Psi, c) = \text{Scat}(c) + \frac{\text{Sep}(c)}{\text{Sep}(c_{\max})} \quad (5.14)$$

where the different column vectors of $U \in \mathbb{R}^{T \times c}$ contain the membership values of the data to different clusters, $\Psi = [\psi_1, \dots, \psi_c]^T$, ψ_i is the centroid of the i^{th} cluster, c is the total number of clusters, T is the total number of samples in $\Theta_H^{(k)}$. Here $Scat(c)$ represents the compactness of the obtained cluster when the number of clusters is c

$$Scat(c) = \frac{\frac{1}{c} \sum_{i=1}^c \sigma_{\psi_i}}{\sigma_{\Theta_H^{(k)}}} \quad (5.15)$$

$$\sigma_{\Theta_H^{(k)}} = \frac{1}{T} \sum_{t=1}^T \left(\Theta_H^{(k)}(t) - \bar{\Theta}_H^{(k)} \right)^2 \quad (5.16)$$

$$\sigma_{\psi_i} = \frac{1}{T} \sum_{t=1}^T u_{ti} \left(\Theta_H^{(k)}(t) - \psi_i \right)^2 \quad (5.17)$$

$$\bar{\Theta}_H^{(k)} = \frac{1}{T} \sum_{t=1}^T \Theta_H^{(k)}(t) \quad (5.18)$$

The range of $Scat(c)$ is between 0 and 1. For compact clustering $Scat(c)$ will be smaller. The term $Sep(c)$ represents the separation between the clusters, which is given by

$$Sep(c) = \frac{d_{\max}^2}{d_{\min}^2} \sum_{i=1}^c \left(\sum_{j=1}^c (\psi_i - \psi_j)^2 \right)^{-1} \quad (5.19)$$

$$d_{\min} = \min_{i \neq j} |\psi_i - \psi_j| \quad (5.20)$$

and

$$d_{\max} = \max_{i \neq j} |\psi_i - \psi_j| \quad (5.21)$$

The value of $Sep(c)$ will be smaller when the cluster centers are well distributed and larger for irregular cluster centers. Hence the best clustering is the one which minimizes $V(U, \Psi, c)$.

The source contribution from different sources will be different in each frequency bin and in some bins the contribution from some of the sources may be very weak.

Hence the number of clusters (or sources) estimated from a single frequency bin will not be reliable. To make the estimation more robust, the cluster validation technique is applied to many frequency bins and the number which is most frequently detected over these frequency bins is taken as the actual number, Q , of sources present.

5.2.4 Permutation problem

The main weaknesses with frequency domain blind source separation are the scaling and the permutation problems. Since the masks are applied directly to the mixture in the TF domain without any other stage in front of it, the well-known scaling problem is avoided. In general, this is true for all TF masking approaches. Therefore only the permutation problem need to be solved. However, it may be noted that the amplitudes of the separated signals may not be exactly equal to those of the original signals. Instead, they are equal to those picked up by the microphone. In the literature many algorithms have been reported for solving the permutation problems [63, 93, 94, 68, 19, 91, 21, 22]. The DOA based algorithms [93, 94, 68, 19] are not effective in highly reverberant environments or when the sources are collinear or very close to one another [21]. In [63] it is shown that for speech signals, the magnitude envelopes of the adjacent frequency bin in the TF domain are highly correlated and this property can be used to solve the permutation problem. Later in [91] it is shown that the correlation between the power ratios are more suitable than those between the magnitude envelopes. This fact is further verified in Fig.5.3, where in Fig.5.3(a), the correlation matrix whose entries are the correlations between the bin wise magnitude envelopes of the STFT coefficients of the two clean signals $\hat{s}_1$ and $\hat{s}_2$ picked up by the microphones are shown. In the figure, the magnitudes of the entries in the correlation matrix are shown by gray levels. The above correlation

matrix $\mathbf{C}_{\hat{S}_1 \hat{S}_2}^{\text{mag}} \in \mathbb{R}^{2K' \times 2K'}$ is calculated as:

$$\mathbf{C}_{\hat{S}_1 \hat{S}_2}^{\text{mag}} = \begin{bmatrix} \mathbf{R}_{\tilde{S}_1 \tilde{S}_1} & \mathbf{R}_{\tilde{S}_1 \tilde{S}_2} \\ \mathbf{R}_{\tilde{S}_2 \tilde{S}_1} & \mathbf{R}_{\tilde{S}_2 \tilde{S}_2} \end{bmatrix} \quad (5.22)$$

where $\mathbf{R}_{\tilde{S}_i \tilde{S}_j} \in \mathbb{R}^{K' \times K'}$, $i, j \in \{1, 2\}$ is the correlation matrix whose $(m, n)^{\text{th}}$ element, $\left(\mathbf{R}_{\tilde{S}_i \tilde{S}_j}\right)_{mn}$, is the Pearson correlation coefficient between the m^{th} and n^{th} rows of $\tilde{S}_i \in \mathbb{R}^{K' \times T}$ and $\tilde{S}_j \in \mathbb{R}^{K' \times T}$ respectively, $K' = \frac{K}{2} + 1$ if the DFT length K is even; otherwise $K' = \frac{K+1}{2}$ and T is the total number of samples in each frequency bin. Because of the conjugate symmetry property of the DFT coefficients, only the first K' bins are taken. The $(k, t)^{\text{th}}$ element of $\tilde{S}_q$, $q \in \{1, 2\}$, is given by

$$\tilde{S}_q(k, t) = \left| \hat{S}_q(k, t) \right| \quad (5.23)$$

Here, $\hat{S}_q(k, t)$ are the STFT coefficients of $\hat{s}_q = h_{pq} * s_q$, which is the clean signal picked up by the p^{th} microphone to which the mask is applied.

The correlations between the bin-wise power ratios of the STFT coefficients of the signals are shown in Fig.5.3(b). The correlation matrix is defined as:

$$\mathbf{C}_{\hat{S}_1 \hat{S}_2}^{\text{Pratio}} = \begin{bmatrix} \mathbf{R}_{P_{\hat{S}_1}^{\text{Pratio}} P_{\hat{S}_1}^{\text{Pratio}}} & \mathbf{R}_{P_{\hat{S}_1}^{\text{Pratio}} P_{\hat{S}_2}^{\text{Pratio}}} \\ \mathbf{R}_{P_{\hat{S}_2}^{\text{Pratio}} P_{\hat{S}_1}^{\text{Pratio}}} & \mathbf{R}_{P_{\hat{S}_2}^{\text{Pratio}} P_{\hat{S}_2}^{\text{Pratio}}} \end{bmatrix} \quad (5.24)$$

where $P_{\hat{S}_q}^{\text{Pratio}}(k, t) = \frac{||\hat{S}_q(k, t)||^2}{||\hat{S}_1(k, t)||^2 + ||\hat{S}_2(k, t)||^2}$, $q = 1, 2$, $k = 1, \dots, K'$, $\forall t$ and the correlation matrix $\mathbf{R}_{P_{\hat{S}_i}^{\text{Pratio}} P_{\hat{S}_j}^{\text{Pratio}}} \in \mathbb{R}^{K' \times K'}$, $i, j \in \{1, 2\}$ is defined in a similar way as that in (5.22). (The size of all the correlation matrices shown in Fig.5.3 are the same as that of $\mathbf{C}_{\hat{S}_1 \hat{S}_2}^{\text{mag}}$).

Comparing Fig.5.3(a) and (b), it can be seen that the correlation between the power ratios is the better choice than that between the magnitude envelopes for solving the permutation problem. The reasons for the improvement in performance are [91] as follows: 1) The values of power ratios are clearly bounded between 0 and 1. 2)

Because of the sparseness of the signals, most of the time, the power ratios will be closer to either 0 or 1. 3) The power ratios of different sources are exclusive to each other, i.e., for a two source case, if $P_{\hat{S}_1}^{\text{ratio}}(k, t)$ is close to 1 then $P_{\hat{S}_2}^{\text{ratio}}(k, t)$ will be close to 0. This shows that the binary mask or the membership functions obtained from the clustering algorithms in Section 5.2.2 are the ideal candidates to replace the power ratios in solving the permutation problem as their values are also close to either 1 or 0. This approach has another advantage that the power ratio calculation can be avoided; instead, the already available masks/membership functions can be used which will save some computation time. The correlations calculated between the power ratios of the STFT coefficients in each frequency bin of the separated signals, $C_{Y_1 Y_2}^{\text{Pratio}}$, and that between the masks, $\mathbf{C}_{M_1 M_2}$, are shown respectively in Fig.5.3(c) and (d). (In cases where it is necessary to specify the algorithm used to estimate the masks, the name of the clustering algorithm will be added as superscript to $\mathbf{C}_{M_1 M_2}$ and M_q . For example, the correlation matrix and the masks estimated by the k -means algorithm will be represented as $\mathbf{C}_{M_1 M_2}^{\text{KM}}$ and M_q^{KM} respectively whereas those by the FCM algorithm will be represented as $\mathbf{C}_{M_1 M_2}^{\text{FCM}}$ and M_q^{FCM} respectively). The correlation matrix $C_{Y_1 Y_2}^{\text{Pratio}}$ is defined similarly to $\mathbf{C}_{\hat{S}_1 \hat{S}_2}^{\text{Pratio}}$, except that $\hat{S}_1$ and $\hat{S}_2$ are replaced by Y_1 and Y_2 respectively. The correlation matrix $\mathbf{C}_{M_1 M_2}$ is calculated as

$$\mathbf{C}_{M_1 M_2} = \begin{bmatrix} \mathbf{R}_{M_1 M_1} & \mathbf{R}_{M_1 M_2} \\ \mathbf{R}_{M_2 M_1} & \mathbf{R}_{M_2 M_2} \end{bmatrix} \quad (5.25)$$

where $M_1 \in \mathbb{R}^{K' \times T}$ and $M_2 \in \mathbb{R}^{K' \times T}$ are the arrays of the first K' masks corresponding to the first and second sources respectively. The correlation matrix $\mathbf{R}_{M_i M_j}$, $i, j \in \{1, 2\}$ is defined in a similar way as that in (5.22). For both Fig.5.3(c) and (d) the permutation problem is solved based on the correlation between the bin-wise power ratios of the separated signals and that of the clean signals picked up by the microphone on which the masks are applied. From the figures it is clear that both the methods will give

Table 5.1: Illustration of mask assignment to different clusters

Freq. bin	No. of masks assigned by k -means algorithm to different clusters					
	$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{C}_3$	$\mathcal{C}_4$	$\mathcal{C}_5$	$\mathcal{C}_6$
k	1	2	1	1	1	0
$k+1$	1	1	0	1	3	0
$k+2$	1	0	0	1	3	1
$k+3$	0	1	1	1	2	1
$k+4$	1	1	1	1	1	1
$k+5$	0	4	1	0	0	1
$k+6$	0	2	2	0	1	1
$k+7$	1	1	1	1	1	1
$k+8$	1	0	1	1	2	1
$k+9$	1	1	1	1	1	1
$k+10$	1	1	2	1	0	1
$k+11$	1	1	1	1	1	1
$k+12$	3	1	1	0	0	1
$k+13$	1	2	1	1	1	0
$k+14$	1	1	1	1	1	1
$k+15$	1	1	1	1	1	1

almost the same performance. A quantitative comparison is given in Section 5.3.1.

The main disadvantage of the correlation based method in solving the permutation problem is that, as the permutation in one frequency bin is solved based on the permutation of the previous frequency bins, failure in one frequency bin will lead to a complete misalignment beyond that frequency bin. Many algorithms have been proposed to circumvent this problem [20, 22, 21]. Sawada et al. [20] combined the DOA and correlation based approaches to improve the robustness of the algorithm. However, the algorithm cannot be used when the sources are collinear [21]. The partial separation method [22, 21] improved the robustness of the correlation method by incorporating a time domain stage in front of the frequency domain stage. To reduce the computational cost, the time domain stage is normally implemented using computationally efficient algorithms [90] with a small number of unmixing filter taps so as to obtain the partially separated signals. The partially separated signal is then input to the frequency domain stage where it is fully separated. Then the permu-

Fig. 5.3: Correlation matrices (a) $\mathbf{C}_{\hat{S}_1 \hat{S}_2}^{\text{mag}}$, correlation between the bin-wise magnitude envelopes of the clean signals picked up by the microphones $\hat{S}_1$ and $\hat{S}_2$; (b) $\mathbf{C}_{\hat{S}_1 \hat{S}_2}^{\text{pratio}}$, correlation between the bin-wise power ratios of the separated signals $\hat{S}_1$ and $\hat{S}_2$; (c) $\mathbf{C}_{Y_1 Y_2}^{\text{pratio}}$, correlation between the bin-wise power ratios of the separated signals Y_1 and Y_2 ; (d) $\mathbf{C}_{M_1 M_2}^{\text{KM}}$, correlation between the masks estimated using k -means clustering M_1 and M_2 ; (e) $\mathbf{C}_{M_1 M_2}^{\text{KM}}$, correlation between the masks estimated using fuzzy c -means clustering M_1 and M_2 ; (f) $\mathbf{C}_{M_1 M_2}^{\text{FCM}}$, correlation between the masks estimated using fuzzy c -means clustering M_1 and M_2 .

tation problem in each frequency bin is solved based on the bin wise correlation between the magnitude envelopes of the DFT coefficients of the fully separated and the partially separated signals. Though the partial separation method can be used with an additional time domain stage in front of the masking stage, the separation of the signals using a time domain ICA algorithm will be very poor when the mixtures are underdetermined and hence this approach could not be used. In this thesis, an algorithm based on k -means clustering is proposed to solve the permutation problem, where the masks are clustered into Q clusters, $\mathcal{C}_q$, $q = 1, \dots, Q$, in such a way that the sum of the distances D_q , $q = 1, \dots, Q$, is minimum. D_q is the total distance between the masks within the q^{th} cluster to its cluster centroid, i.e.,

$$\text{minimize } \mathcal{D} = \sum_{q=1}^Q \sum_{\substack{M_i^{(k)} \in \mathcal{C}_q \\ i=1, \dots, Q \\ k=k_{\text{st}}, \dots, k_{\text{end}}}} \left(1 - r_{M_i^{(k)} C_q} \right) \quad (5.26)$$

where $M_i^{(k)}$ is the i^{th} mask in the k^{th} frequency bin, C_q is the centroid of the q^{th} cluster $\mathcal{C}_q$, $r_{M_i^{(k)} C_q}$ is the Pearson correlation between $M_i^{(k)}$ and the cluster centroid C_q , k_{st} and k_{end} are the indices of the starting and ending frequency bins of the group of adjacent frequency bins used for clustering, i.e., the total number of frequency bins used is $k_{\text{end}} - k_{\text{st}} + 1$. Here $1 - r_{M_i^{(k)} C_q}$ is used as the distance measure so that masks which are highly correlated (smaller distance) will form one cluster. Since there are Q sources, Q clusters are formed using the k -means algorithm. In an ideal case, each cluster must contain one and only one mask from each frequency bin after clustering. But in practice this may not be the case, especially when the number of sources is large. Under such situations, the bins in each cluster are to be identified where the permutation could not be solved perfectly. This can be done as follows:

In an ideal case, after clustering, each cluster will contain masks corresponding to one and only one source and hence the number of masks in each frequency bin will be exactly one. Hence, after clustering, if the number of masks in a particular

frequency bin in any cluster is different from one, it is assumed that the k -means clustering algorithm has failed to solve the permutation problem in that particular frequency bin between those clusters. A typical example for the case of six sources (hence six clusters) is shown in Table 5.1, where the masks from 16 adjacent bins are clustered. In Table 5.1, entries other than '1' indicate that the algorithm fails to solve the permutation problem for that cluster at that particular frequency bin. For example at the k^{th} frequency bin, the algorithm fails in clusters $\mathcal{C}_2$ and $\mathcal{C}_6$. For frequency bins where the k -means clustering algorithm fails to solve the permutation problem, the correlation between the cluster centroids of the failed clusters and the masks in those clusters are used to solve the permutation problem. This is done by reassigning the masks in the failed clusters in such a way that the sum of the correlations between the centroids of the clusters and the masks is maximum, i.e., the permutation matrix Π_k for the k^{th} frequency bin among the failed clusters is calculated as

$$\Pi_k = \arg \max_{\Pi} \sum_i^F \sum_j^F (\Pi \bullet \mathbf{R}_{\text{CM}})_{ij} \quad (5.27)$$

where $\bullet$ represents element wise multiplication between the matrices, F is the number of failed clusters, Π is the permutation matrix with one and only one element, which is 1, in any row or column, $\mathbf{R}_{\text{CM}} \in \mathbb{R}^{F \times F}$ is the correlation matrix, $(\mathbf{R}_{\text{CM}})_{ij}$ is the Pearson correlation between the i^{th} and j^{th} rows of $\mathbf{C}$ and $\mathbf{M}$ respectively, $\mathbf{C} = [\dots, C_q^T, \dots]^T$, $C_q \in \mathbb{R}^{1 \times T}$ is the centroid of the q^{th} cluster, $q \in \{\text{indices of failed clusters}\}$, $\mathbf{M} = [\dots, M_q^T, \dots]^T$, $M_q \in \mathbb{R}^{1 \times T}$ are the masks in the failed clusters at the k^{th} frequency bin. Then the matrix of *permutation solved masks* at frequency bin k will be $\Pi_k \mathbf{M}$.

For example, for the $(k+1)^{\text{th}}$ frequency bin in Table 5.1 three masks are assigned to cluster $\mathcal{C}_5$ whereas none are assigned to clusters $\mathcal{C}_3$ and $\mathcal{C}_6$. Hence for the $(k+1)^{\text{th}}$ frequency bin, the permutation problem is to be solved among the clusters $\mathcal{C}_3$, $\mathcal{C}_5$ and $\mathcal{C}_6$ by calculating the correlations between the centroids of the clusters, $\mathbf{C} = [C_3^T, C_5^T, C_6^T]^T$, and the masks assigned to $\mathcal{C}_5$. The masks assigned to clusters $\mathcal{C}_1$, $\mathcal{C}_2$

and $\mathcal{C}_4$ are not altered.

For speech signals in the TF domain, when the frequency bins are far apart, the correlation between them will decrease [91]. To overcome this problem, instead of taking all the masks to form the clusters, only a few adjacent frequency bins are taken at a time with overlap (for example 16 bins with 75% overlap in all the experiments in this chapter) and cluster them using the k -means clustering algorithm as explained previously. In the k -means algorithm, since the initialization vector used (as the initial centroids) has an impact on the final clustering [139, 176], the centroids of the current clusters are used as the starting centroids (initializing vector) for the next group of masks for clustering. For the starting group of masks (i.e., for bins $k = 1$ to 16 in the experiments in this chapter) the centroids of the clusters obtained by applying the k -means algorithm on the masks in the frequency range of 500Hz to 1000Hz are used as the initialization vectors. The advantages of taking small groups of adjacent masks, overlapping and initializing with the centroids of the previous clusters are: 1) The correlations between the masks corresponding to the same source will be high if the masks belong to the nearby frequency bins and hence there will be a clear separation between the clusters. 2) The centroids of the current clusters will be close to those of the clusters formed by the next group of masks, if both groups are overlapped. This will decrease the convergence time of the k -means clustering algorithm. 3) When initialized with the centroids of the previous group, because of the overlap, the starting centroids will be close to the actual centroids. Hence the permutation of the present group will be the same as that of the previous group of masks.

A similar approach for solving the permutation problem using the masks is reported in [174]. In addition, the correlation between the masks is used as the distance measure. The main difference between the proposed method and that in [174] is that, in the proposed method, the well-known k -means algorithm is used for clustering the masks. There are many improved versions for the basic k -means clustering algorithm (see [139] and the references therein) and any of these algorithms can be used.

Moreover, the proposed method uses small groups of adjacent frequency bins with overlap and each group is initialized with the cluster centroids of the previous group. As explained above and shown in [176], this kind of initialization will increase the convergence speed and significantly reduce the computation time. However, there may be some frequency bins where the k -means algorithm fails to solve fully the permutation problem. The permutation of these bins could be solved by maximizing the sum of the correlations between the centroids of the failed clusters and the masks in those clusters, using (5.27). Whereas in [174], first all the frequency bins are globally aligned by maximizing the sum of the correlations between the cluster centroids and the masks in each frequency bin in such a way that, from each frequency bin, one and only one mask is assigned to one cluster. This is similar to the k -means algorithm applied to the mask taken from all the frequency bins with the constraint that at any frequency bin one and only one mask is assigned to one cluster. Then for fine local optimization at frequency bin k , the sum of the correlations between the masks in the k^{th} bin and the masks from a set of other frequency bins are maximized. The set of frequency bins typically consists of the adjacent and the harmonically related frequency bins of the k^{th} bin. This is repeated for all the bins until no improvement is found for any of the frequency bins.

5.2.5 Construction of the output signals

Using the separated signals Y_q obtained by applying the masks to one of the microphone outputs in the TF domain, i.e., $Y_q(k, t) = M_q(k, t)X_p(k, t)$, $q = 1, \dots, Q$, $p \in \{1, \dots, P\}$, the separated signals in the time domain are constructed by taking inverse STFT followed by the overlap add method [170]. The masks can be applied to any one of the microphone outputs. However, the performance will be slightly affected by the microphone position. Readers are referred to Section.5.3.4 for more explanation.

5.3 Experimental results

For performance evaluation of the proposed algorithm, both real room and simulated impulse responses are used. In Section 5.3.1 the impulse response of a real furnished room is used whereas for the remaining experiments, to have a fine control on the position of the microphones and sources as well as on the acoustic environment, simulated impulse responses are used [126]. In all the experiments discussed in this chapter, average performances of 50 combinations of speech utterances, selected randomly from 16 speech utterances shown in Fig.4.1, are used. For the same number of sources, in all the experiments, the combination of speech utterances used are the same. For experiments in Sections 5.3.4 and 5.3.5, the wall reflections up to 29th order is taken and humidity, temperature and absorption of sound due to air are considered while calculating the impulse responses. The reverberation time, TR_{60} , of the simulated room is 115ms.

During the separation process, the signals may be distorted especially when the sources are overlapped in their TF domain. Hence it is necessary to measure the distortion and the artifacts introduced by the algorithm to assess the quality of separation. The quality of separation of the algorithm is measured using the method proposed in [181, 182], where the separated (estimated) signals are first decomposed into three components as

$$y_q = y_{q_{\text{target}}} + e_{q_{\text{interf}}} + e_{q_{\text{artif}}} \quad (5.28)$$

where $y_{q_{\text{target}}}$ is the target source with allowed deformation such as filtering or gain, $e_{q_{\text{interf}}}$ accounts for the interference due to unwanted sources and $e_{q_{\text{artif}}}$ corresponds to the artifacts introduced by the separation algorithm. Then the source-to-distortion ratio (SDR), source-to-interference ratio (SIR) and source-to-artifacts ratio in dB are calculated as

$$\text{SDR} = 10 \log_{10} \frac{\|y_{q_{\text{target}}}\|^2}{\|e_{q_{\text{interf}}} + e_{q_{\text{artif}}}\|^2} \quad (5.29)$$

Fig. 5.4: The source-microphone configuration for the measurement of real room impulse responses

$$\text{SIR} = 10 \log_{10} \frac{\|y_{q_{\text{target}}}\|^2}{\|e_{q_{\text{interf}}}\|^2} \quad (5.30)$$

$$\text{SAR} = 10 \log_{10} \frac{\|y_{q_{\text{target}}} + e_{q_{\text{interf}}}\|^2}{\|e_{q_{\text{artif}}}\|^2} \quad (5.31)$$

In the proposed algorithm, since the mask is applied to one of the microphone outputs in the TF domain, the target signal is taken as the signal picked up by the microphone to which the mask is applied. Here the target source is $y_{q_{\text{target}}} = h_{pq} * s_q$ where h_{pq} is the impulse response from the q^{th} source to the p^{th} microphone, if the mask is applied to the p^{th} microphone output. The other experimental conditions are: length of speech utterances are 5 seconds, speech sampling frequency is 16 kHz, DFT frame size $K=2048$ and the window function used is Hanning window.

5.3.1 Experiments using real room impulse responses

In this experiment the impulse responses measured in a real furnished room is used. The reverberation time of the room (TR_{60}) is 187 ms and the impulse response is measured with the help of an acoustic impulse response measuring software ‘Sample Champion’ [129]. The microphone and loud speaker transfer function are neglected in the measurements. The position of the microphones and sources are shown in Fig.5.4. One of the impulse responses (from source s_3 to the first microphone) is shown in Fig.5.5. The sources s_1 and s_2 are collinear. The separation of the sources when

they are collinear is a challenging task using algorithms like independent component analysis. For example, using the computationally efficient implementation [90] of the time domain convolutive BSS algorithm proposed in [81, 82], with an unmixing filter length of 512, the SIR obtained is 10.9dB for noncollinear sources (s_1 and s_3) and only 3.8dB for collinear sources (s_1 and s_2). Here the unmixing filter of length 512 is taken because, as discussed in [21], if the filter length is longer, the interdependency of the unmixing filter coefficients will cause the convergence to be poor. On the other hand, an unmixing filter with a shorter filter length will not be able to achieve any significant unmixing effect.

Fig. 5.5: Measured real room impulse response from source s_3 to the first microphone.

Fig. 5.6: (a), (b) and (c) Mean histogram of the ‘estimated number of clusters (or sources)’ for the first 60 frequency bins. (d), (e) and (f) Total number of frequency bins used versus ‘estimated number of clusters (or sources) ’; the estimation result will be more reliable with higher number of frequency bins used. In the figures, at some points, the ‘number of clusters estimated’ are not integers because it is the mean performance of 50 sets of speech utterances. All the source positions are with reference to Fig.5.4.

5.3.2 Detection of the number of sources

For the detection of the number of sources present in the mixture, the cluster validation technique explained in Section.5.2.3 is applied to $\Theta_H^{(k)}$ for the three different cases shown in Fig.5.4. The first case involves non-collinear sources (s_1 and s_3), the second case collinear sources (s_1 and s_2) and finally the third case all the three sources (s_1 , s_2 and s_3). The mean performance obtained for 50 combinations of speech utterances is shown in Fig.5.6. Fig.5.6(a), (b) and (c) show the mean histogram of the estimated number of clusters (or sources) over the first 60 frequency bins for three cases of s_1

and s_3 , s_1 and s_2 and s_1 , s_2 and s_3 respectively. From the figure it can be seen that the algorithm successfully estimated the number of sources in all the three cases. Fig.5.6(d), (e) and (f) show the total number of frequency bins used versus the estimated number of sources. The figures clearly show that it is not necessary to apply the cluster validation technique to all the frequency bins; instead a fraction of the total frequency bins is sufficient for the successful estimation of the number of sources. Since the Hermitian angle calculated at any instant depends on the relative amplitude of the source, the variations in the calculated Hermitian angles will be high during the period where the unvoiced parts of the sources overlap. For example in Figs.5.1 and 5.2, during the time frame $t = 80$ to 120 the magnitude envelopes of the sources are small in amplitude and the variation in Hermitian angles are high. In contrast, during the periods where the magnitude envelopes are high in amplitude, the variations in Hermitian angles are low. Considering this fact in all the experiments in this chapter, $\Theta_H^{(k)}(t)$ at any point where $\|\mathbf{X}(k, t)\| < 0.1 \frac{1}{T} \sum_{t=1}^T \|\mathbf{X}(k, t)\|$ are removed from $\Theta_H^{(k)}$ before they are clustered for the estimation of the number of sources. This will reduce not only the estimation error but also the computation time.

5.3.3 Separation performance

The separation performance obtained using the proposed algorithm for the three cases namely collinear, non-collinear and underdetermined with collinear sources are shown in Table 5.2. The corresponding waveforms are shown in Fig.5.7, 5.8, 5.9 and 5.10. In the table, the performances of the algorithm when k -means and fuzzy c -means clustering are used for the design of masks are shown for the cases where the permutation problem is solved by: 1) comparing the correlation between power ratios of the separated signals with that of the clean signals picked up by the microphones, and 2) using the proposed k -means clustering approach. Here the correlation between the power ratios of the clean signals and the separated signals for solving the permutation problem is used as the bench mark to evaluate the

Table 5.2: Performance comparison of the proposed algorithm using k -means and FCM clustering.

Active sources	Performance measure	Input (dB)	Permutation solved using clean signals				Permutation solved by k -means clustering			
			k -means		FCM		k -means		FCM	
			Output (dB)	Improvement (dB)	Output (dB)	Improvement (dB)	Output (dB)	Improvement (dB)	Output (dB)	Improvement (dB)
s_1 and s_3 (Non-collinear)	SDR	-0.2	6.1	6.4	6.5	6.8	6.5	6.8	6.8	7.1
	SIR	0.0	18.2	18.2	16.8	16.8	18.9	18.9	17.3	17.3
	SAR	16.1	6.6	-9.5	7.2	-8.9	6.9	-9.1	7.4	-8.6
s_1 and s_2 (Collinear)	SDR	-0.3	4.7	5.0	5.1	5.3	5.4	5.7	5.7	5.9
	SIR	-0.0	15.6	15.6	14.5	14.5	16.9	16.9	15.6	15.6
	SAR	16.2	5.4	-10.8	5.9	-10.3	6.0	-10.2	6.4	-9.7
s_1, s_2 and s_3 (Underdetermined with collinear)	SDR	-3.4	1.8	5.2	2.0	5.4	0.5	3.9	1.0	4.4
	SIR	-3.2	11.9	15.1	10.4	13.6	10.0	13.2	9.1	12.3
	SAR	16.0	2.6	-13.4	3.2	-12.8	1.7	-14.3	2.5	-13.5

Table 5.3: Algorithm execution time

Mask estimation method	No. of sources (Each of 5 sec length)	Time to solve the permutation problem alone (using the proposed algorithm based on K-means clustering) (seconds)	Total time to separate the sources from their mixtures. (seconds)
<i>k</i> -means	2	2.33	5.79
	3	3.22	9.60
FCM	2	2.30	5.05
	3	3.33	10.50

proposed *k*-means clustering algorithm for solving the permutation problem because it is very robust, independent of the quality of separation in each bin and in the ideal case where the separation is perfect, the permutation can be solved perfectly. The permutation matrix estimation procedure can be mathematically expressed as follows:

$$\mathbf{\Pi}_k = \arg \max_{\mathbf{\Pi}} \sum_i^Q \sum_j^Q \left(\mathbf{\Pi} \bullet \mathbf{R}_{\mathbf{P}_Y^{\text{ratio}} \mathbf{P}_S^{\text{ratio}}} \right)_{ij} \quad (5.32)$$

where $\mathbf{R}_{\mathbf{P}_Y^{\text{ratio}} \mathbf{P}_S^{\text{ratio}}}$ is the correlation matrix, $\left(\mathbf{R}_{\mathbf{P}_Y^{\text{ratio}} \mathbf{P}_S^{\text{ratio}}} \right)_{ij}$ is the Pearson correlation between i^{th} and j^{th} rows of $\mathbf{P}_Y^{\text{ratio}}$ and $\mathbf{P}_S^{\text{ratio}}$ respectively, $\mathbf{P}_Y^{\text{ratio}}$ is the matrix of power ratios of the separated signals in the k^{th} frequency bin whose t^{th} column is given by

$$\mathbf{P}_Y^{\text{ratio}}(t) = \left[\frac{\|Y_1(k, t)\|^2}{\sum_{q=1}^Q \|Y_q(k, t)\|^2}, \dots, \frac{\|Y_Q(k, t)\|^2}{\sum_{q=1}^Q \|Y_q(k, t)\|^2} \right]^T. \quad (5.33)$$

Similarly, $\mathbf{P}_S^{\text{ratio}}$ is the matrix of power ratios of the signal picked up by the p^{th} microphone at the k^{th} frequency bin whose column vectors are given by

$$\mathbf{P}_S^{\text{ratio}}(t) = \left[\frac{\|H_{p1}(k) S_1(k, t)\|^2}{\sum_{q=1}^Q \|H_{pq}(k) S_q(k, t)\|^2}, \dots, \frac{\|H_{pQ}(k) S_Q(k, t)\|^2}{\sum_{q=1}^Q \|H_{pq}(k) S_q(k, t)\|^2} \right]^T. \quad (5.34)$$

where $p \in \{1, \dots, P\}$ is the index of the microphone to which the mask is applied. From table 5.2, it can be seen that the SIR improvement is higher when *k*-means

Table 5.4: Experimental Conditions

Source signals	Speech of 5sec (obtained by concatenating the sentences from TIMIT database)
Direction of sources	As shown in the respective figures
Distance between two microphones	As mentioned in the respective experiments
Sampling rate f_s	16kHz
DFT size	$K = 2048$
Room temperature	25°
Humidity of air	40% (for simulation)
Wall reflections	29 th order (for simulation)
Window function	Hanning window

clustering is used compared to FCM clustering. However, the improvement in artifacts and distortion are higher when the FCM clustering algorithm is used. It can also be seen from the table that the proposed method based on k -means clustering for solving the permutation problem is as good as solving the permutation problem by comparing the separated signals with the clean signals.

The time taken to execute the proposed algorithm when coded in Matlab (version 7.4.0.287 (R2007a)) and run in a PC with Intel Core 2 Duo 2.66 GHz CPU, 2 GB of RAM is shown in Table 5.3. Note that the k -means algorithm for the mask estimation is initialized with the result obtained from the histogram method on $\Theta_H^{(k)}$, whereas the FCM algorithm was initialized with randomly selected samples from $\Theta_H^{(k)}$.

5.3.4 Microphone spacing and selection of microphone output to apply mask.

The estimated mask can be applied to the mixture in the TF domain obtained from one of the microphone outputs. This experiment examines the output of the microphone on which the mask is to be applied to obtain the best performance. It is logical to apply the masks to the output of the center microphone which is proven experimentally and shown in Figs.5.12 to be the best choice.

In the experiments, the simulated impulse responses obtained for the source mi-

Fig. 5.7: Waveform of clean speech (s_1 and s_3), individual signals picked up by the first microphone ($h_{11} * s_1$ and $h_{13} * s_3$), mixed signals (x_1 and x_2) and separated signals, separated by k -means (y_1^{KM} and y_3^{KM}) and FCM (y_1^{FCM} and y_3^{FCM}) algorithms, for the case of non-collinear sources. The notations are with reference to Fig.5.4. The audio files are available in the accompanying CD

Fig. 5.8: Waveform of clean speech (s_1 and s_2), individual signals picked up by the first microphone ($h_{11} * s_1$ and $h_{12} * s_2$), mixed signals (x_1 and x_2) and separated signals, separated by k -means (y_1^{KM} and y_2^{KM}) and FCM (y_1^{FCM} and y_2^{FCM}) algorithms, for the case of collinear sources. The notations are with reference to Fig.5.4. The audio files are available in the accompanying CD

Fig. 5.9: Waveform of individual signals picked up by the first microphone ($h_{11} * s_1$, $h_{12} * s_2$ and $h_{13} * s_3$), mixed signals (x_1 and x_2) and separated signals, separated by k -means algorithm (y_1^{KM} , y_2^{KM} and y_3^{KM}), for the underdetermined case. The notations are with reference to Fig.5.4. The audio files are available in the accompanying CD

Fig. 5.10: Waveform of individual signals picked up by the first microphone ($h_{11} * s_1$, $h_{12} * s_2$ and $h_{13} * s_3$), mixed signals (x_1 and x_2) and separated signals, separated by FCM algorithm (y_1^{FCM} , y_2^{FCM} and y_3^{FCM}), for the underdetermined case. The notations are with reference to Fig.5.4. The audio files are available in the accompanying CD

crophone configuration shown in Fig.5.11 are used. Out of the total six sources, only two sources are active at any time and hence there are a total of $\frac{6!}{2!(6-2)!} = 15$ combinations of source positions. For each combination of source positions the experiment is repeated for 50 sets of utterances. The performances shown in Figs.5.12 are the mean performances of these 750 experiments. To study the effect of microphone spacing, these 750 experiments are repeated for different microphone spacing. For this purpose microphone arrays consisting of five microphones with different spacings (2cm, 5cm, 10cm and 20cm) are used. For all the microphone spacings the center of the array is kept at the same point. The experimental results show that the performance improves as the spacing between the microphones increases, and after a certain distance this improvement begins to drop. The reason for the variation in performance because of the variation in spacing between the microphones can be explained as follows.

When the microphones are very close, the difference between the impulse responses of any one source and the microphones is small. For example, the impulse response between source s_1 and microphone Mic.1 will be almost the same as that between s_1 and microphone Mic.2 when both microphones are very close to one

Fig. 5.11: The source-microphone configuration for the simulated room impulse responses

Fig. 5.12: SDR/SIR/SAR versus index of the microphone output on which mask is applied, for different microphone spacings. Dotted lines are for the cases where the permutation problem is solved by finding the correlation between the bin-wise power ratios of the separated signals and that of clean signals picked up by the microphones. Solid lines are for the cases where the permutation problem is solved by the proposed method based on the k -means clustering algorithm. The mean input SDR, SIR and SAR are -0.09dB, 0dB and 20.82dB respectively.

Fig. 5.13: Variation in angle between the column vectors $\mathbf{H}_q(k)$, $q = 1, 2$ versus microphone spacing. Dotted lines show the angles for different source combinations, as marked in the figure, and solid line shows the mean angle.

another. Hence in the frequency domain, the column vectors $\mathbf{H}_q(k)$, $q = 1, \dots, Q$ will be very close to one another and as a result the angles between them will be small. When the angles between the mixing vectors are very small, partitioning of the samples will be difficult and the separation performance will also be poor. However, as the maximum value of the Hermitian angle is $\pi/2$, the angle between the column vectors $\mathbf{H}_q(k)$, $q = 1, \dots, Q$ will not increase in proportion to the spacing between the microphones. Hence, after a certain spacing the performance improvement due to the increase in spacing will not be significant. This fact is illustrated in Fig.5.13 where the average angle between the column vectors $\mathbf{H}_q(k)$, $q = 1, \dots, Q$, over the first 100 bins as a function of spacing between the microphones is shown.

It may be noted that, in Fig.5.12, for 2 cm microphone spacing the performance is lower when the proposed k -means clustering algorithm is used for solving the permutation problem than when the correlation between the power ratios of the

separated and clean signals are used. This is because when the spacing is small, the clustering of $\Theta_H^{(k)}$ will be difficult which will lead to errors in mask estimation. For the proposed algorithm for solving the permutation problem, the robustness of the cluster formation depends on the quality of the estimated masks. If the mask quality is poor, the permutation problem will not be solved perfectly which will result in poor separation in the time domain. On the other hand, if the correlation between the clean signals and the separated signals is used for solving the permutation problem, the robustness will be very high and the drop in performance will be mainly due to the imperfect separation in each frequency bin, and that due to the error in solving the permutation problem will be minimum.

5.3.5 Effect on the number of microphones

Generally in BSS, the larger the number of microphones, the better the performance. Here also this observation holds. The SDR, SIR and SAR improvements for different combinations of number of sources and microphones are shown in Fig.5.14 where the masks are generated using k -means clustering. The source microphone positions are the same as those in Fig.5.11. The spacings between the microphones are fixed at 10cm for all the experiments. For the case of odd number of microphones, the masks are applied to the output of the centre microphone. When the numbers of microphones are 2 and 4, masks are applied to the first and the second microphone outputs respectively. As explained in Section 5.3.4, for two sources, because of the 15 combinations of source position, 750 simulations were done. Similarly, 1000, 750, 300 and 50 simulations were done for 3, 4, 5 and 6 sources respectively and the mean performances so obtained are shown in Fig.5.14. From Figs.5.12 and 5.14, it can be seen that the binary masking method for the separation of the sources from their mixtures will introduce artifacts due to nonlinear distortions. This cannot be avoided and it will increase as the overlapping of the sources increases. To mitigate this problem, some post processing techniques have to be used [183].

Fig. 5.14: Performance versus number of microphones. (a) output SDR (b) output SIR (c) output SAR (d) SDR improvement (e) SIR improvement (f) SAR improvement.

5.4 Summary

In this chapter, an algorithm for separation of an unknown number of sources from their underdetermined convolutive mixtures via TF masking and a method for solving the permutation problem by clustering the masks using k -means clustering is proposed. The algorithm uses the membership functions from the clustering algorithm as the masks. The separation performance of the algorithm is evaluated for the two popular clustering algorithms, namely k -means and fuzzy c -means. The crisp nature of the membership functions generated by the k -means algorithm resulted in more artifacts in the separated signals compared to those by fuzzy c -means algorithm, which is a soft partitioning technique. For the automatic detection of the number of sources, the optimum number of clusters formed by the Hermitian angles in different frequency bins are estimated and the number that estimated most frequently is taken as the number of sources present in the mixture. In this chapter, the cluster validation technique is used for the estimation of the number of cluster; however, other techniques can also be used. In TF masking methods for BSS, in general, the scaling problem does not exist and this is true for the proposed algorithm also. However, the well-known permutation problem still exists but could be solved by clustering. The validity of the proposed algorithms are demonstrated for both real room and simulated speech mixtures.

In all the three stages of underdetermined convolutive BSS investigated in this chapter (detection of the number of sources, mask estimation and solution of the permutation problem), clustering techniques are used, and the separation performance depends mainly on the clustering techniques used.

Chapter 6

Conclusion and Recommendations

6.1 Conclusion

In this thesis two methods for solving the well-known permutation problem of frequency domain BSS are proposed. The first method, partial separation method, is based on the correlations between the magnitude envelopes of the DFT coefficients at two different frequency bins of the same frequency. One of the frequency bins is obtained from a time domain stage where the signals are partially separated using a time domain BSS algorithm and the separated signals are then converted to the TF domain. The other bin is from the frequency domain stage where the permutation problem is to be solved. The algorithm does not require any information about the source positions, instead it needs a time domain convolutive BSS algorithm which can partially separate the mixed signals. Since it requires only a partial separation, computationally efficient versions of the available algorithms can be used. The computation time can further be reduced by decreasing the number of filter taps of the unmixing filters. Since the only requirement to successfully solve the permutation problem is that the spectra of the partially separated signals should be close to the clean signals, the algorithm can be used to solve the permutation problem even when the sources are collinear, provided an algorithm is available for the partial separation of the mixed signals. Unlike the other correlation based approaches, since the permutation at one bin is solved independently of the permutations in the previous bins, the algorithm is more robust. For the proposed cascade configuration, the computational cost of the time domain stage for the partial separation is optimally utilized and the overall

performance is better than that of the scheme using frequency domain stage alone.

The second method based on the mask clustering approach proposed for solving the permutation problem can also be used for the underdetermined BSS cases. The algorithm solves the permutation problem by clustering the masks which can then be used subsequently for source separation. The use of the mask has all the advantages of the scheme based on power ratios of the DFT coefficients of the separated signals in the frequency bins. Moreover, the computational time for the calculation of the power ratios is saved.

In a two stage approach for separation of the sources from their underdetermined mixtures, estimation of the single source points in the TF plane of their mixtures is an important task especially when the overlapping of the source spectra is very high. The algorithms proposed in literature are either complex or require some single source “zones” instead of points. The algorithm proposed in this thesis for the detection of the SSP is very simple and the SSPs do not require any other adjacent SSPs. Unlike some previously reported algorithms, the number of mixtures is not restricted to two; instead it can be any number. The mixing matrix is then estimated by clustering the identified SSPs.

Separation of sources from their underdetermined convolutive mixtures is a challenging task in BSS. In this thesis, this problem is addressed using the TF masking approach. For the estimation of the masks by clustering the Hermitian angles calculated in each frequency bin, the suitability of two well-known clustering algorithms, namely k -means and fuzzy c -means algorithms, are examined. For both the approaches, the algorithm gave good separation performance. The crisp nature of the membership functions generated by the k -means algorithms introduced more artifacts in the separated signals than when the membership functions from the fuzzy c -means algorithm is used. However, the reduction in artifacts is at the cost of reduction in signal-to-interference ratio (SIR). In addition to the source separation, an algorithm for the automatic detection of the number of sources present in the mixed

signals is also proposed. The estimation of the number of sources is by estimating the optimum number of clusters formed by the Hermitian angles in different frequency bins. Since the contribution of sources in different frequency bins may be different and in some bins the contributions from some of the sources may be very weak, to improve the robustness of the estimation, the number that is estimated most frequently for different frequency bins is taken as the actual number of sources. For the estimation of the number of clusters the cluster validation technique is used. The other techniques for the estimation of the number of sources can also be used. However, in practice, the disjoint orthogonality condition of the source signals is not fully satisfied. This will result in overlapping between the clusters formed by the Hermitian angles. Hence it is recommended to use algorithms which are suitable for clusters with overlap. In TF masking methods for BSS, in general, the scaling problem does not exist and this is true for the proposed algorithm also. In summary, the main advantages of the proposed algorithms are as follows:

The partial separation method for solving the permutation problem is very robust and in the cascade configuration, the overall separation performance is not only due to the frequency domain stage but also the time domain stage. The partial separation method alone could achieve 6.5dB higher improvement in NRR compared to that by the DOA method alone. When the other methods such as adjacent bands correlation and harmonic correlation methods are combined with both partial separation and DOA, the improvement that the proposed scheme could achieve compared to the DOA method is 3dB higher. The algorithm for solving the permutation problem based on k -means clustering is suitable for determined as well as the underdetermined cases. The proposed algorithm for the detection of single source points in the TF domain of the instantaneous mixtures is computationally much faster than the previously reported algorithms in the literature and the constraint on the mixing is very much relaxed, i.e., the SSPs do not require any other adjacent SSPs; also, the performance of the algorithm is better than the conventional algorithms. For example, for the case

of two sources and two mixtures, the separation performance obtained is 61dB which is 13dB higher than that of the best performing algorithm used for comparisons. For the underdetermined case, with six sources and two mixtures, the mixing matrix estimation error (NMSE) obtained is -43dB. In the algorithm developed for underdetermined convolutive blind source separation via time-frequency masking, well known clustering algorithms are used for the mask estimation where the masks are estimated by clustering the one dimensional vector of Hermitian angles. Since the data to be clustered are always one dimensional irrespective of the number of microphones, the increase in computational cost because of the increase in number of microphones is not significant. In addition to the computational efficiency of the algorithm, the separation performance obtained is also good. For example, separation performance (SIR) of 15.1dB is obtained for the underdetermined case (three sources and two microphones) in a real room environment with reverberation time $TR_{60} = 187\text{ms}$, where two of the sources are collinear. Also, the formation of clear clusters by the vector of Hermitian angles leads to a simple algorithm for the automatic detection of the number of sources present in the mixtures.

6.2 Recommendations for further research

In the two stage approach for separation of the sources from their underdetermined instantaneous mixtures, the algorithm proposed in Chapter 4 for the estimation of the SSPs is extremely simple. However, the already available algorithms for the source estimation using the estimated mixing matrices are complex and imperfect especially when the number of sources is very large compared to the number of sensors. Hence, research in this direction is envisaged to have some potential.

Another idea is the incorporation of an algorithm similar to that in Chapter 5 for automatic detection of the number of sources. Since the SSPs have already been estimated, the estimation of the number of sources will be more accurate than directly

applying the clustering based algorithms for automatic detection of the number of sources on all the samples.

The underdetermined convolutive BSS technique proposed in Chapter 5 seems highly promising and the idea can further be exploited for better separation. In the TF masking approach for BSS, the separation performance of the algorithm depends on the quality of the estimated masks. At MSPs in the TF plane, the Hermitian angle between the resultant mixture vector and the reference vector will be different from that of the angle between the reference vector and the column of the mixing matrix. This variation depends on the relative amplitudes of the source contributions at that point. Since the Hermitian angles calculated at MSPs are independent of the actual magnitude of the sources, the influence of the Hermitian angles calculated from the MSPs with smaller amplitude can be avoided if the magnitude of the samples can also be included along with the Hermitian angles in the cost function of the clustering algorithm for mask estimation.

For the proposed underdetermined convolutive BSS algorithm, it is assumed that the sources are disjoint in the TF domain and some overlap is allowed. However, the greater the spectral overlap, the more the artifacts introduced in the separated signals. If the idea used for the estimation of the masks can be further extended for mixing matrix estimation, then the algorithm can be used for the separation of the signals with overlapped spectra. Hence, further research in this direction seems to be very promising.

Another idea to improve the separation performance and the automatic detection of the number of sources is to cluster the samples using Hermitian angles calculated using more than one reference vectors, preferably with P mutually orthogonal vectors, where P is the number of sensors. Though the use of more than one reference vectors will increase the computational cost, the separation performance can be further improved. This improvement is possible because in multidimensional space the angle between the single reference vector and another vector cannot give the full

information about the position of the second vector. Instead, by the use of more reference vectors, the position of the vector can be estimated more accurately. With P mutually orthogonal vectors, the position of any vector of dimension P can be exactly determined. This will result in accurate estimation of the clusters and hence the masks. Another recommendation is the more extensive evaluation of the algorithms using noisy data as all the algorithms proposed in this thesis are evaluated using only noise free data.

Appendix A

Convolution Using Discrete Sine and Cosine Transforms

A.1 Introduction

The convolution multiplication property of the discrete Fourier transform (DFT) is well-known. For discrete cosine and sine transforms (DCTs & DSTs), called discrete trigonometric transform (DTT), such a nice property does not exist. S.A.Martucci [184, 185] derived the convolution multiplication properties of all the families of discrete sine and cosine transforms, in which the convolution is a special type called symmetric convolution. For symmetric convolution the sequences to be convolved must be either symmetric or antisymmetric. The general form of the equation for symmetric convolution in the DTT domain is $s(n) * h(n) = T_c^{-1} \{T_a \{s(n)\} \bullet T_b \{h(n)\}\}$, where $s(n)$ and $h(n)$ are the input sequences, $\bullet$ represents the element-wise multiplication operation and $*$ represents the convolution operation. The type of the transforms T_a , T_b and T_c to be used depends on the type of the symmetry of the sequences to be convolved (see [184, 185] for more details). In [184, 185] and [186] it is also showed that by proper zero-padding of the sequences, symmetric convolution can be used to perform linear convolution.

Here a relation for circular convolution in the DTT domain is derived. The advantage of this new relation is that the input sequences to be convolved need not be symmetrical or antisymmetrical and the computational time is less than that of the symmetric convolution method.

Fig. A.1: Generation of $\check{C}_1(k)$, $\check{S}_1(k)$, $\check{C}_2(k)$ and $\check{S}_2(k)$ from $C_1(k)$, $S_1(k)$, $C_2(k)$ and $S_2(k)$ respectively after decimation and symmetric or antisymmetric extension. The black squares represent the appended zeros to make the length of the sequences to $N + 1$ for element-wise operation.

A.2 Convolution in DTT domain

The discrete sine and cosine transforms used here are the same as those used in [184] and [185], which are given below.

$$S_{C_1}(k) = 2 \sum_{n=0}^N \zeta_n s(n) \cos\left(\frac{\pi k n}{N}\right) \quad k = 0, 1, \dots, N \quad (\text{A.1})$$

$$S_{C_2}(k) = 2 \sum_{n=0}^{N-1} s(n) \cos\left(\frac{\pi k (2n+1)}{2N}\right) \quad k = 0, 1, \dots, N-1 \quad (\text{A.2})$$

$$S_{S_1}(k) = 2 \sum_{n=1}^{N-1} s(n) \sin\left(\frac{\pi k n}{N}\right) \quad k = 1, 2, \dots, N-1 \quad (\text{A.3})$$

$$S_{S_2}(k) = 2 \sum_{n=0}^{N-1} s(n) \sin \left(\frac{\pi k(2n+1)}{2N} \right) \quad k = 1, 2, \dots, N \quad (\text{A.4})$$

$$\zeta_n = \begin{cases} \frac{1}{2} & n = 0 \quad \text{or} \quad N \\ 1 & n = 1, 2, \dots, N-1 \end{cases}$$

where $S_{C_1}(k)$, $S_{C_2}(k)$, $S_{S_1}(k)$, $S_{S_2}(k)$ denote the type I even DCT (DCT1e) coefficients, type II even DCT (DCT2e) coefficients, type I even DST (DST1e) coefficients and type II even DST (DST2e) coefficients, respectively of the sequence $s(n)$.

Let the sequences to be convolved be $s(n)$ and $h(n)$ of length N so that the convolved signal is $s(n) \otimes h(n)$, where $\otimes$ represents the circular convolution operation. The DFT of $s(n)$ is given by [187]

$$S(k) = \sum_{n=0}^{N-1} s(n) e^{-j2\pi kn/N} \quad k = 0, 1, \dots, N-1 \quad (\text{A.5})$$

Multiplying (A.5) by $2e^{-j\pi k/N}$ it becomes

$$2e^{-j\pi k/N} S(k) = 2 \sum_{n=0}^{N-1} s(n) \left(\cos \left(\frac{\pi k(2n+1)}{N} \right) - j \sin \left(\frac{\pi k(2n+1)}{N} \right) \right) \quad (\text{A.6})$$

Comparing (A.2) and first term of (A.6), it can be observed that $2 \sum_{n=0}^{N-1} s(n) \cos \left(\frac{\pi k(2n+1)}{N} \right)$ is the decimated and antisymmetrically extended version of (A.2) with index $k = 0, 1, \dots, N-1$ (It may be noted that, for convenience, the index range in equations will be represented using the notation “:”. For example $0, 1, \dots, N$ will be represented as $1 : N$). Similarly, comparing (A.4) and second term of (A.6) it can be observed that $2 \sum_{n=0}^{N-1} s(n) \sin \left(\frac{\pi k(2n+1)}{N} \right)$ is the decimated and symmetrically extended version of (A.4) with index $k = 1, 2, \dots, N$. For convenient element-wise operation in the following equations, append 0 at $k = N$ to the resulting sequence of the first term, and at $k = 0$ to the resulting sequence of the second term so as to obtain the sequences $\check{S}_{C_2}(k)$ and $\check{S}_{S_2}(k)$ respectively of length $N+1$. Hence (A.6) becomes

$$2e^{-\frac{j\pi k}{N}} S(k) = \check{S}_{C_2}(k) - j\check{S}_{S_2}(k) \quad (\text{A.7})$$

A similar equation can be written for $h(n)$ as

$$2e^{-\frac{j\pi k}{N}} H(k) = \check{H}_{C_2}(k) - j\check{H}_{S_2}(k) \quad (\text{A.8})$$

Element-wise multiplication of (A.7) and (A.8) gives

$$S(k)H(k) = \frac{1}{4}e^{\frac{j2\pi k}{N}} \left\{ \left(\check{S}_{C_2}(k)\check{H}_{C_2}(k) - \check{S}_{S_2}(k)\check{H}_{S_2}(k) \right) - j \left(\check{S}_{S_2}(k)\check{H}_{C_2}(k) + \check{S}_{C_2}(k)\check{H}_{S_2}(k) \right) \right\} \quad (\text{A.9})$$

Taking the real part of the inverse discrete Fourier transform of (A.9), i.e.,

$$\begin{aligned} \text{real} \left(\frac{1}{N} \sum_{k=0}^{N-1} S(k)H(k)e^{\frac{j2\pi kn}{N}} \right) &= \frac{1}{4N} \sum_{k=0}^N \underbrace{\left(\check{S}_{C_2}(k)\check{H}_{C_2}(k) - \check{S}_{S_2}(k)\check{H}_{S_2}(k) \right)}_{T_1(k)} \cos \left(\frac{2\pi k(n+1)}{N} \right) \\ &\quad + \frac{1}{4N} \sum_{k=1}^{N-1} \underbrace{\left(\check{S}_{S_2}(k)\check{H}_{C_2}(k) + \check{S}_{C_2}(k)\check{H}_{S_2}(k) \right)}_{T_2(k)} \sin \left(\frac{2\pi k(n+1)}{N} \right) \end{aligned} \quad (\text{A.10})$$

Since $\check{S}_{C_2}(N) = \check{H}_{C_2}(N) = \check{S}_{S_2}(N) = \check{H}_{S_2}(N) = \check{S}_{S_2}(0) = \check{H}_{S_2}(0) = 0$, the summation range of the first term in (A.10) is changed from $k = 0, 1, \dots, N-1$ to $k = 0, 1, \dots, N$ and that of the second term to $k = 1, 2, \dots, N-1$.

Comparing (A.1), (A.3) and (A.10) it can be observed that without the scaling factor $\frac{1}{4N}$, the first term in (A.10) is the decimated and symmetrically extended version of DCT1e coefficients, $C_1\{T_1\}$, and the second term is the decimated and antisymmetrically extended version of the DST1e coefficients, $S_1\{T_2\}$, except for the shift in the resulting sequences by one sample and the absence of the constants ζ_n and 2. Considering these constants and using the fact that inverse of DCT1e is the same as

DCT1e and inverse of DST1e is the same as DST1e, except for a scaling factor $2N$ [184, 185], the above equation can be rewritten as

$$s(n) \otimes h(n) = \frac{1}{4} \left(\check{C}_1^{-1} \left\{ \xi_k \left(\check{C}_2 \{s\} \bullet \check{C}_2 \{h\} - \check{S}_2 \{s\} \bullet \check{S}_2 \{h\} \right) \right\} + \check{S}_1^{-1} \left\{ \check{S}_2 \{s\} \bullet \check{C}_2 \{h\} + \check{C}_2 \{s\} \bullet \check{S}_2 \{h\} \right\} \right) \quad (\text{A.11})$$

where

$$\xi_k = \begin{cases} 2 & k = 0 \quad \text{or} \quad N \\ 1 & k = 1, 2, \dots, N-1 \end{cases}$$

The steps for computing (A.11) can be explained as follows.

- Compute $\check{C}_2 \{s\}$ and $\check{S}_2 \{s\}$ as

$$[\check{C}_2]_{k,n} = 2 \cos \left(\frac{\pi k(2n+1)}{N} \right) \quad k, n = 0, 1, \dots, N-1$$

$$[\check{S}_2]_{k,n} = 2 \sin \left(\frac{\pi k(2n+1)}{N} \right) \quad \begin{matrix} k = 1, 2, \dots, N \\ n = 0, 1, \dots, N-1 \end{matrix}$$

$$\begin{aligned} \check{C}_2 \{s\} &= [\check{S}_{C_2}]_{(N+1) \times 1} = \begin{bmatrix} & & & & \\ & \check{C}_2 & & & \\ 0 & 0 & \dots & 0 & 0 \end{bmatrix}_{(N+1) \times N} \begin{bmatrix} s \\ \end{bmatrix}_{N \times 1} \\ \check{S}_2 \{s\} &= [\check{S}_{S_2}]_{(N+1) \times 1} = \begin{bmatrix} 0 & 0 & \dots & 0 & 0 \\ & \check{S}_2 & & & \\ & & & & \end{bmatrix}_{(N+1) \times N} \begin{bmatrix} s \\ \end{bmatrix}_{N \times 1} \end{aligned}$$

Alternatively $\check{C}_2 \{s\}$ and $\check{S}_2 \{s\}$ can be found from the sequences $C_2 \{s\}$ and $S_2 \{s\}$ respectively after decimating and extending them antisymmetrically and symmetrically as shown in Fig.A.1. The square markings in the figure show the appended zeros.

- Similarly, compute $\check{C}_2 \{h\}$ and $\check{S}_2 \{h\}$.
- Compute $T_1(k)$ and $T_2(k)$ as

$$[T_1]_{(N+1) \times 1} = [\check{S}_{C_2}] \bullet [\check{H}_{C_2}] - [\check{S}_{S_2}] \bullet [\check{H}_{S_2}]$$

$$[T_2]_{(N+1) \times 1} = [\check{S}_{S_2}] \bullet [\check{H}_{C_2}] + [\check{S}_{C_2}] \bullet [\check{H}_{S_2}]$$

- Multiply $T_1(0)$ and $T_1(N)$ by $\xi_k = 2$ and keep all other elements the same to obtain the new sequence $T'_1(k)$ of length $N + 1$.
- Discard $T_2(0)$ and $T_2(N)$ to obtain the new sequence $T'_2(k)$ of length $N - 1$.
- Compute $\check{C}_1^{-1} \{T'_1\}$ and $\check{S}_1^{-1} \{T'_2\}$ as

$$[\check{C}_1]_{k,n} = 2\zeta_n \cos\left(\frac{2\pi kn}{N}\right) \quad k, n = 0, 1, \dots, N$$

$$[\check{S}_1]_{k,n} = 2\sin\left(\frac{2\pi kn}{N}\right) \quad k, n = 1, 2, \dots, N - 1$$

$$\check{C}_1^{-1} \{T'_1\} = [\check{T}'_{1C_1^{-1}}]_{(N+1) \times 1} = \frac{1}{2N} [\check{C}_1]_{(N+1) \times (N+1)} [T'_1]_{(N+1) \times 1}$$

$$\check{S}_1^{-1} \{T'_2\} = [\check{T}'_{2S_1^{-1}}]_{(N-1) \times 1} = \frac{1}{2N} [\check{S}_1]_{(N-1) \times (N-1)} [T'_2]_{(N-1) \times 1}$$

Alternatively $\check{C}_1^{-1} \{T'_1\}$ and $\check{S}_1^{-1} \{T'_2\}$ can be found from the sequences $C_1 \{T'_1\}$ and $S_1 \{T'_2\}$ after scaling, decimating and extending them symmetrically and antisymmetrically as shown in Fig.A.1.

- Discard the first element of $\check{T}'_{1C_1^{-1}}$, append one zero at the end of $\check{T}'_{2S_1^{-1}}$, add the resulting sequences together and scale them with the scaling factor $\frac{1}{4}$ to obtain

the convolved signal as

$$s(n) \circledast h(n) = \frac{1}{4} \left(\check{T}'_{1C_1^{-1}}(1 : N) + \left[\check{T}'_{2S_1^{-1}}; 0 \right] \right)$$

It is interesting to note that in symmetric convolution [184, 185], the time sequences are symmetric or antisymmetric whereas in (A.11), the DTT coefficients are symmetric or antisymmetric except for the appended zeros in the sequences $\check{C}_2(k)$ and $\check{S}_2(k)$. Utilizing the fact that any signal can be split into symmetric and antisymmetric sequences, in [184, 185, 186] and [188], it was shown that symmetric convolution can be used for linear convolution. For example, if a long sequence $x(n)$ is to be convolved with filter coefficients $h(n)$ of length $\mathcal{Q}$, then segment the signal $x(n)$ into blocks of length $\mathcal{M}$ with overlap $2\mathcal{Q} - 1$. Let $x_b(n)$ be the b^{th} block and $h'(n)$ be the filter coefficients of length $\mathcal{M}$ after appending $\mathcal{M} - \mathcal{Q}$ zeros, then calculate $w_b(n) = C_1^{-1} \{T_c - T_s\}$, where $T_c(0 : \mathcal{M} - 1) = C_2 \{x_b\} \bullet C_2 \{h'\}$, $T_c(\mathcal{M}) = 0$, $T_s(1 : \mathcal{M}) = S_2 \{x_b\} \bullet S_2 \{h'\}$ and $T_s(0) = 0$. The $\mathcal{P} = \mathcal{M} - 2\mathcal{Q} + 1$ samples of $w_b(n)$ after removing $\mathcal{Q}$ samples from both sides of $w_b(n)$ will be the valid linear convolution coefficients. Hence, symmetric convolution can be used for linear convolution. However, it can be seen that, since the block length of the input sequence is $\mathcal{M}$, the length of the DTTs to be calculated are also of length $\mathcal{M}$ or $\mathcal{M} + 1$ ($\mathcal{M}$ for C_2 and S_2 , $\mathcal{M} + 1$ for C_1^{-1}) and the valid outputs will be of length $\mathcal{P} = \mathcal{M} - 2\mathcal{Q} + 1$.

Since (A.11) is for circular convolution, similar to DFT, by proper zero padding, it can be used for linear convolution also. For example, as in the previous case, to filter a long sequence $x(n)$ with filter coefficients $h(n)$ of length $\mathcal{Q}$, segment the signal $x(n)$ into blocks of length $\mathcal{P}$ and append each block with $\mathcal{Q} - 1$ zeros to get blocks of length $\mathcal{R} = \mathcal{P} + \mathcal{Q} - 1$. Similarly, append $\mathcal{P} - 1$ zeros to the filter coefficients to make its length equal to $\mathcal{R}$. Then apply (A.11), overlap and add the resulting output blocks to get the filtered signal. While computing, because of the symmetry of the DTT coefficients in (A.11), it is sufficient to calculate only half of the total number of

Table A.1: Computational cost comparison

Method used	Number of DTT coefficients to be calculated				$\times$	$+/-$
	DCT1e	DST1e	DCT2e	DST2e		
Symmetric convolution	$\mathcal{M} + 1$	0	$2\mathcal{M}$	$2\mathcal{M}$	$2\mathcal{M}$	$\mathcal{M} - 1$
Proposed	$\lfloor \frac{\mathcal{R}}{2} \rfloor + 1$	$\lfloor \frac{\mathcal{R}-1}{2} \rfloor$	$2 \lceil \frac{\mathcal{R}}{2} \rceil$	$2 \lfloor \frac{\mathcal{R}}{2} \rfloor$	$3 \lceil \frac{\mathcal{R}}{2} \rceil + \lfloor \frac{\mathcal{R}}{2} \rfloor - 2$	$\lfloor \frac{\mathcal{R}-1}{2} \rfloor + 2 \lceil \frac{\mathcal{R}}{2} \rceil - 2$

$\lfloor y \rfloor$ and $\lceil y \rceil$ round y to the nearest integer towards minus infinity and plus infinity respectively.

Filter length = $\mathcal{Q}$, valid output samples per block = $\mathcal{P}$, $\mathcal{M} = \mathcal{P} + 2\mathcal{Q} - 1$ and $\mathcal{R} = \mathcal{P} + \mathcal{Q} - 1$.

coefficients. The remaining half is the symmetrically extended version of the first half. Also, for the second part of (A.11), the same DTT coefficients $\check{S}_{C_2}$, $\check{H}_{C_2}$, $\check{S}_{S_2}$ and $\check{H}_{S_2}$ that were used for the first part can be used. Similarly, for the element-wise multiplication, because of the symmetry of the DTT coefficients, only half of the coefficients need to be multiplied. The other half will be the same as the first half with or without the sign changes. Likewise, for the addition and subtraction operations only half of the elements need to be added or subtracted. Moreover, unlike the symmetric convolution method, the length of the DTTs to be calculated here are $\mathcal{R} + 1$, $\mathcal{R}$ or $\mathcal{R} - 1$ ($\mathcal{R} + 1$ for $\check{C}_1$, $\mathcal{R}$ for $\check{C}_2$ and $\check{S}_2$, $\mathcal{R} - 1$ for $\check{S}_1$), which is smaller than that for the symmetric convolution method. The computational cost per DTT coefficient will decrease as DTT length decreases. Hence, the computational time of (A.11) is less than that of the symmetric convolution method. Table A.1 summarizes the computational cost for the two methods in filtering application, neglecting the cost involved for sign change, symmetric or antisymmetric extension of the DTT coefficients and multiplication by the scaling factors.

Appendix B

Single Source Point Identification in DTT Domain

Let $s(n)$ and $h(n)$ be the two sequences of length N , $n = 0, 1, \dots, N-1$, and $x(n) = s(n) \circledast h(n)$ is the convolved signal, where $\circledast$ represents the circular convolution operation. Using (A.9), this operation can also be expressed as (since this section deals with only DCT2e and DST2e, for convenience, the subscripts '2' in C_2 and S_2 are dropped. For example H_{C_2} and H_{S_2} will be represented as H_C and H_S respectively):

$$\begin{aligned}\hat{X}(k) &= 4e^{\theta_k} H(k) S(k) \\ &= \left(\check{H}_C(k) \check{S}_C(k) - \check{H}_S(k) \check{S}_S(k) \right) - j \left(\check{H}_C(k) \check{S}_S(k) + \check{H}_S(k) \check{S}_C(k) \right)\end{aligned}\tag{B.1}$$

where $k = 0, 1, \dots, N$, $\theta_k = -j2\pi k/N$, $\hat{X}(k) = 4e^{\theta_k} X(k)$, $X(k)$ is the DFT coefficients of the sequence $x(n)$ after appending zero at $k = N$, $\check{S}_C$ is the decimated and anti-symmetrically extended version of the discrete cosine type II even (DCT2e) transform coefficients of the sequence $s(n)$ after appending zero at $k = N$, and $\check{S}_S$ is the decimated and symmetrically extended version of the discrete sine type II even (DST2e) transform coefficients of the sequence $s(n)$ after appending zeros at $k = 0$, i.e.,

$$\check{S}_C(k) = \begin{cases} 2 \sum_{n=0}^{N-1} s(n) \cos\left(\frac{\pi k(2n+1)}{N}\right) & k = 0, 1, \dots, N-1 \\ 0 & k = N \end{cases} \quad (\text{B.2})$$

$$\check{S}_S(k) = \begin{cases} 0 & k = 0 \\ 2 \sum_{n=0}^{N-1} s(n) \sin\left(\frac{\pi k(2n+1)}{N}\right) & k = 1, 2, \dots, N \end{cases} \quad (\text{B.3})$$

The DCT2e and DST2e coefficients of a sequence are defined in (A.2) and (A.4) respectively. Now equating the real and imaginary parts of (B.1),

$$R\{\hat{X}(k)\} = \check{H}_C(k) \check{S}_C(k) - \check{H}_S(k) \check{S}_S(k) \quad (\text{B.4})$$

$$I\{\hat{X}(k)\} = \check{H}_C(k) \check{S}_S(k) + \check{H}_S(k) \check{S}_C(k) \quad (\text{B.5})$$

where $R\{\cdot\}$ and $I\{\cdot\}$ represent the real and imaginary part operations respectively. In the TF domain, (B.4) and (B.5) can be extended for the convolutive mixing of Q sources to obtain P mixtures as

$$\begin{aligned} R\{\hat{\mathbf{X}}(k, t)\} &= \check{\mathbf{H}}_C(k) \check{\mathbf{S}}_C(k, t) - \check{\mathbf{H}}_S(k) \check{\mathbf{S}}_S(k, t) \\ &= \sum_{q=1}^Q \left(\check{\mathbf{H}}_{qC}(k) \check{S}_{qC}(k, t) - \check{\mathbf{H}}_{qS}(k) \check{S}_{qS}(k, t) \right) \end{aligned} \quad (\text{B.6})$$

$$\begin{aligned} I\{\hat{\mathbf{X}}(k, t)\} &= \check{\mathbf{H}}_C(k) \check{\mathbf{S}}_S(k, t) + \check{\mathbf{H}}_S(k) \check{\mathbf{S}}_C(k, t) \\ &= \sum_{q=1}^Q \left(\check{\mathbf{H}}_{qC}(k) \check{S}_{qS}(k, t) + \check{\mathbf{H}}_{qS}(k) \check{S}_{qC}(k, t) \right) \end{aligned} \quad (\text{B.7})$$

where $\hat{\mathbf{X}}(k, t) = [\hat{\mathbf{X}}_1(k, t), \dots, \hat{\mathbf{X}}_P(k, t)]^T$, $\check{\mathbf{H}}_{qC}(k) = [\check{H}_{1qC}(k), \dots, \check{H}_{PqC}(k)]^T$ and $\check{\mathbf{H}}_{qS}(k) = [\check{H}_{1qS}(k), \dots, \check{H}_{PqS}(k)]^T$ are the q^{th} column vectors of the matrices $\check{\mathbf{H}}_C$ and $\check{\mathbf{H}}_S$ respectively. In the case of instantaneous mixing $x_p(n) = \sum_{q=1}^Q h_{pq}s_q(n)$, $p = 1, 2, \dots, P$, where $\mathbf{x}_p = [x_p(0), \dots, x_p(N-1)]$ is the p^{th} mixture, $\mathbf{s}_q = [s_q(0), \dots, s_q(N-1)]$ is the q^{th} source, h_{pq} is the $(p, q)^{th}$ element of the mixing matrix and N is the total number of samples)

as the mixing filters are simple pulses of amplitudes h_{pq} , (B.2) and (B.3) will lead to $\check{H}_{pq_C}(k) = 2h_{pq} \cos(\pi k/N)$ and $\check{H}_{pq_S}(k) = 2h_{pq} \sin(\pi k/N)$. Then in (B.6) and (B.7), $\check{\mathbf{H}}_{q_C}(k) = 2\mathbf{h}_q \cos(\pi k/N)$ and $\check{\mathbf{H}}_{q_S}(k) = 2\mathbf{h}_q \sin(\pi k/N)$, where $\mathbf{h}_q = [h_{1q}, \dots, h_{Pq}]^T$. Hence the absolute direction of the vector $\check{\mathbf{H}}_{q_C}(k)$ in the mixture space will be the same as that of $\check{\mathbf{H}}_{q_S}(k)$ for all the frequencies, which is the absolute direction of the q^{th} column vector of the mixing matrix. Hence $\check{\mathbf{H}}_{q_S}(k)$ can be written as

$$\check{\mathbf{H}}_{q_S}(k) = \lambda(k)\check{\mathbf{H}}_{q_C}(k) \quad (\text{B.8})$$

where $\lambda(k) = \tan(\pi k/N)$, which is the same for all the samples in the same frequency bin. Substituting (B.8) in (B.6) and (B.7) will give:

$$R\{\hat{\mathbf{X}}(k, t)\} = \sum_{q=1}^Q \check{\mathbf{H}}_{q_C}(k) \left(\check{S}_{q_C}(k, t) - \lambda(k)\check{S}_{q_S}(k, t) \right) \quad (\text{B.9})$$

$$I\{\hat{\mathbf{X}}(k, t)\} = \sum_{q=1}^Q \check{\mathbf{H}}_{q_C}(k) \left(\check{S}_{q_S}(k, t) + \lambda(k)\check{S}_{q_C}(k, t) \right) \quad (\text{B.10})$$

The DCT2e coefficients and DST2e coefficients of a signal are not the same in magnitude or sign but in most of the frequency bins they occur concurrently. Hence the decimated and symmetrically extended DCT2e (dDCT2e) and DST2e (dDST2e) coefficients also occur concurrently. This can be seen in Fig.B.1, where dDCT2e and dDST2e coefficients of two speech signals in a randomly selected frequency bin are shown.

For ease of explanation, assume $P = Q = 2$. Now, if only one of the sources is present at one point in the TF plane, then the absolute direction of the vector $R\{\hat{\mathbf{X}}(k, t)\}$ (or slope of the line passing through $R\{\hat{\mathbf{X}}(k, t)\}$ and the origin) will be the same as that of $I\{\hat{\mathbf{X}}(k, t)\}$. For example, if only the contribution from source $\mathbf{s}_1$ is present in the TF plane at (k_1, t_1) , i.e., $\check{S}_{1_C}(k_1, t_1) \neq 0$, $\check{S}_{1_S}(k_1, t_1) \neq 0$, $\check{S}_{2_C}(k_1, t_1) = 0$ and $\check{S}_{2_S}(k_1, t_1) = 0$, the absolute direction of $R\{\hat{\mathbf{X}}(k_1, t_1)\}$ and $I\{\hat{\mathbf{X}}(k_1, t_1)\}$ will then be the

Fig. B.1: dDCT2e and dDST2e coefficients of two speech utterances, s_1 and s_2 .

same as that of $\check{\mathbf{H}}_{1_C}(k_1)$. Similarly when only $\mathbf{s}_2$ is present, say at (k_2, t_2) , the absolute direction of $R\{\hat{\mathbf{X}}(k_2, t_2)\}$ and $I\{\hat{\mathbf{X}}(k_2, t_2)\}$ will be the same as that of $\check{\mathbf{H}}_{2_C}(k_2)$. Now assume another instant, (k_3, t_3) , in the TF plane where both the sources are present. The probability for $R\{\hat{\mathbf{X}}(k_3, t_3)\}$ and $I\{\hat{\mathbf{X}}(k_3, t_3)\}$ to have the same absolute direction is very low as the absolute direction of $R\{\hat{\mathbf{X}}(k_3, t_3)\}$ is the absolute direction of the vector, which is the sum of the mixing vectors $\check{\mathbf{H}}_{1_C}(k_3)$ and $\check{\mathbf{H}}_{2_C}(k_3)$ after multiplying them with $(\check{S}_{1_C}(k_3, t_3) - \lambda(k)\check{S}_{1_S}(k_3, t_3))$ and $(\check{S}_{2_C}(k_3, t_3) - \lambda(k)\check{S}_{2_S}(k_3, t_3))$ respectively, whereas in (B.10), the multiplication factors for the mixing vectors $\check{\mathbf{H}}_{1_C}(k_3)$ and $\check{\mathbf{H}}_{2_C}(k_3)$ are $(\check{S}_{1_S}(k_3, t_3) + \lambda(k)\check{S}_{1_C}(k_3, t_3))$ and $(\check{S}_{2_S}(k_3, t_3) + \lambda(k)\check{S}_{2_C}(k_3, t_3))$ respectively. Hence in the TF plane, at the point (k, t) , where the absolute direction of the vector $R\{\hat{\mathbf{X}}(k, t)\}$ is the same as that of $I\{\hat{\mathbf{X}}(k, t)\}$, is a single-source-point. This idea can also be easily extended for the case of P mixtures of Q sources.

Multiplication of $\mathbf{X}(k, t)$ by a complex number will not affect the angle between $R\{\mathbf{X}(k, t)\}$ and $I\{\mathbf{X}(k, t)\}$, if they are in the same direction. Hence instead of using $\hat{\mathbf{X}}(k, t)$, $\mathbf{X}(k, t)$ can be used for the detection of the SSPs, i.e., the SSPs are the points in the TF plane where the absolute direction of $R\{\mathbf{X}(k, t)\}$ is the same as that of $I\{\mathbf{X}(k, t)\}$. This fact is illustrated in Fig.B.2, where both $\mathbf{X}(k, t)$ and $\hat{\mathbf{X}}(k, t)$ are used for the detection of the SSPs. From the figure it can be seen that the performance for both the cases are almost the same. The slight difference in performance is due to the fact that the points in the TF plane are taken as SSPs in such a way that the angle between $R\{\mathbf{X}(k, t)\}$ and $I\{\mathbf{X}(k, t)\}$ (similarly between $R\{\hat{\mathbf{X}}(k, t)\}$ and $I\{\hat{\mathbf{X}}(k, t)\}$) is less than $\Delta\theta$ instead of zero. If the angle between $R\{\mathbf{X}(k, t)\}$ and $I\{\mathbf{X}(k, t)\}$ are not the same, the angle between $R\{\hat{\mathbf{X}}(k, t)\}$ and $I\{\hat{\mathbf{X}}(k, t)\}$ may not be equal to that between $R\{\mathbf{X}(k, t)\}$ and $I\{\mathbf{X}(k, t)\}$. However, this difference, i.e., $(\angle R\{\mathbf{X}(k, t)\} - \angle I\{\mathbf{X}(k, t)\}) - (\angle R\{\hat{\mathbf{X}}(k, t)\} - \angle I\{\hat{\mathbf{X}}(k, t)\})$, will be very small as $\Delta\theta$ is very small.

Fig. B.2: Performance comparison of the algorithm using $\mathbf{X}(k, t)$ and $\hat{\mathbf{X}}(k, t)$

Appendix C

Proof: Hermitian angle between two complex vectors will remain the same even if they are multiplied by complex scalars

If $\mathbf{u}_1$ and $\mathbf{u}_2$ are multiplied by the complex scalars a and b respectively, then (5.8) will become

$$\begin{aligned}\cos(\theta_C) &= \frac{(a\mathbf{u}_1)^H(b\mathbf{u}_2)}{\sqrt{(a\mathbf{u}_1)^H(a\mathbf{u}_1)}\sqrt{(b\mathbf{u}_2)^H(b\mathbf{u}_2)}} \\ &= \frac{\sum_i a^* u_{i1}^* b u_{i2}}{\sqrt{\sum_i a^* u_{i1}^* a u_{i1}} \sqrt{\sum_i b^* u_{i2}^* b u_{i2}}}\end{aligned}\tag{C.1}$$

where u_{iq} is the i^{th} element of the column vector $\mathbf{u}_q$ and $*$ represents the complex conjugate operation. Let

$$a = Ae^{j\theta_A}\tag{C.2}$$

$$b = Be^{j\theta_B}\tag{C.3}$$

$$u_{i1} = U_{i1}e^{j\phi_i}\tag{C.4}$$

$$u_{i2} = U_{i2}e^{j\varphi_i}\tag{C.5}$$

then

$$\begin{aligned}
 \cos(\theta_C) &= \frac{\sum_i A e^{-j\theta_A} U_{i1} e^{-j\phi_i} B e^{j\theta_B} U_{i2} e^{j\varphi_i}}{\sqrt{\sum_i A e^{-j\theta_A} U_{i1} e^{-j\phi_i} A e^{j\theta_A} U_{i1} e^{j\phi_i}} \sqrt{\sum_i B e^{-j\theta_B} U_{i2} e^{-j\varphi_i} B e^{j\theta_B} U_{i2} e^{j\varphi_i}}} \\
 &= \frac{A B e^{j(\theta_B - \theta_A)} \sum_i U_{i1} U_{i2} e^{j(\varphi_i - \phi_i)}}{A \sqrt{\sum_i U_{i1}^2} B \sqrt{\sum_i U_{i2}^2}} \\
 &= \frac{e^{j(\theta_B - \theta_A)} \sum_i U_{i1} U_{i2} e^{j(\varphi_i - \phi_i)}}{\sqrt{\sum_i U_{i1}^2} \sqrt{\sum_i U_{i2}^2}}
 \end{aligned} \tag{C.6}$$

and

$$\begin{aligned}
 \cos(\theta_H) &= |\cos(\theta_C)| \\
 &= \frac{|e^{j(\theta_B - \theta_A)} \sum_i U_{i1} U_{i2} e^{j(\varphi_i - \phi_i)}|}{\sqrt{\sum_i U_{i1}^2} \sqrt{\sum_i U_{i2}^2}} \\
 &= \frac{|e^{j(\theta_B - \theta_A)}| |\sum_i U_{i1} U_{i2} e^{j(\varphi_i - \phi_i)}|}{\sqrt{\sum_i U_{i1}^2} \sqrt{\sum_i U_{i2}^2}} \\
 &= \frac{|\sum_i U_{i1} U_{i2} e^{j(\varphi_i - \phi_i)}|}{\sqrt{\sum_i U_{i1}^2} \sqrt{\sum_i U_{i2}^2}}
 \end{aligned} \tag{C.7}$$

which is independent of a and b .

Author's Publications

Journal papers

1. V. G. Reju, S. N. Koh and I. Y. Soon, "Underdetermined Convolutional Blind Source Separation via Time-Frequency Masking," *IEEE Transactions on Audio, Speech and Language Processing*, Vol. 18, NO. 1, Jan. 2010, pp. 101–116.
2. V. G. Reju, S. N. Koh and I. Y. Soon, "An algorithm for mixing matrix estimation in instantaneous blind source separation," *Signal Processing*, Vol. 89, Issue 9, September 2009, pp. 1762–1773.
3. V. G. Reju, S. N. Koh and I. Y. Soon, "Partial separation method for solving permutation problem in frequency domain blind source separation of speech signals," *Neurocomputing*, Vol. 71, NO. 10–12, June 2008, pp. 2098–2112.
4. V. G. Reju, S. N. Koh and I. Y. Soon, "Convolution Using Discrete Sine and Cosine Transforms," *IEEE Signal Processing Letters*, Vol. 14, NO. 7, July 2007, pp. 445–448.

Conference papers

1. V. G. Reju, S. N. Koh and I. Y. Soon, "A Robust Correlation Method for Solving Permutation Problem in Frequency Domain Blind Source Separation of Speech Signals," *In Proc. of the IEEE Asia Pacific Conference on Circuits and Systems*, pp. 1891–1894, Dec. 2006.
2. V. G. Reju, S. N. Koh, I. Y. Soon and X. Zhang, "Solving permutation problem in blind source separation of speech signals: A method applicable for collinear sources," *In Proc. of the Fifth International Conference on Information, Communications and Signal Processing*, pp. 1461–1465, Dec. 2005.

References

- [1] Y. Li, S. Amari, A. Cichocki, D. W. C. Ho, and S. Xie, "Underdetermined blind source separation based on sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, p. 423–437, Feb. 2006.
- [2] A. Hyvarinen, "Fast and robust fixed-point algorithms for independent component analysis," *IEEE Transactions on Neural Networks*, vol. 10, p. 626–634, May 1999.
- [3] J. Herault and C. Jutten, "Space or time adaptive signal processing by neural network models," in *Proceedings of the American Institute for Physics Conference*, (New York), Aug. 1986.
- [4] K. A. Meraim, W. Qiu, and Y. Hua, "Blind system identification," *Proceedings of the IEEE*, vol. 85, p. 1310–1322, Aug. 1997.
- [5] D. T. Pham, "Fast algorithms for mutual information based independent component analysis," *IEEE Transactions on Signal Processing*, vol. 52, p. 2690–2700, Oct. 2004.
- [6] P. Comon and L. Rota, "Blind separation of independent sources from convolutive mixtures," *IEICE Transactions on Fundamentals*, vol. E86–A, no. 3.
- [7] J. Karhunen, P. Pajunen, and E. Oja, "The nonlinear PCA criterion in blind source separation: Relation with other approaches," *Neurocomputing*, vol. 22, p. 5–20, Nov. 1998.
- [8] E. Oja, "From neural learning to independent components," *Neurocomputing*, vol. 1–3, p. 187–199, Nov. 1998.
- [9] J. F. Cardoso, "Infomax and maximum likelihood for blind source separation," *IEEE Signal Processing Letters*, vol. 4, p. 112–114, Apr. 1997.
- [10] A. J. Bell and T. J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," *Neural Computation*, vol. 7, no. 6, p. 1129–1159, 1995.

- [11] A. Belouchrani, K. A. Meraim, J. F. Cardoso, and E. Moulines, "A blind source separation technique using second-order statistics," *IEEE Transactions on Signal Processing*, vol. 45, p. 434–444, Feb. 1997.
- [12] E. Bingham and A. Hyvarinen, "A fast and fixed-point algorithm for independent component analysis of complex valued signals," *International Journal on Neural Systems*, vol. 10, p. 1–8, Feb. 2000.
- [13] A. Belouchrani and M. G. Amin, "Blind source separation based on time-frequency signal representation," *IEEE Transactions on Signal Processing*, vol. 46, p. 2888–2897, Nov. 1998.
- [14] J. V. Stone, "Blind deconvolution using temporal predictability," *Neurocomputing*, vol. 49, p. 79–86, 2002.
- [15] L. Parra and C. Spence, "Convolutional blind separation of non-stationary sources," *IEEE Transactions on Speech and Audio Processing*, vol. 8, p. 320–327, May 2000.
- [16] P. Smaragdis, "Blind separation of convolved mixtures in the frequency domain," *Neurocomputing*, vol. 22, p. 21–34, Nov. 1998.
- [17] W. Wang, S. Sanei, and J. A. Chambers, "Penalty function-based joint diagonalization approach for convolutional blind separation of nonstationary sources," *IEEE Transactions on Signal Processing*, vol. 53, p. 1654–1669, May 2005.
- [18] O. Yilmaz and S. Rickard, "Blind separation of speech mixtures via time-frequency masking," *IEEE Transactions on Signal Processing*, vol. 52, p. 1830–1847, July 2004.
- [19] M. Z. Ikram and D. R. Morgan, "Permutation inconsistency in blind speech separation: Investigation and solutions," *IEEE Transactions Speech Audio Processing*, vol. 13, p. 1–13, Jan. 2005.
- [20] H. Sawada, R. Mukai, S. Araki, and S. Makino, "A robust and precise method for solving the permutation problem of frequency domain blind source separation," *IEEE Transactions on Speech and Audio Processing*, vol. 12, p. 530–538, Sept.

2004.

- [21] V. G. Reju, S. N. Koh, and I. Y. Soon, "Partial separation method for solving permutation problem in frequency domain blind source separation of speech signals," *Neurocomputing*, vol. 71, p. 2098–2112, June 2008.
- [22] V. G. Reju, S. N. Koh, and I. Y. Soon, "A robust correlation method for solving permutation problem in frequency domain blind source separation of speech signals," in *Proceedings of the APCCAS*, p. 1893–1896, 2006.
- [23] P. Bofill and M. Zibulevsky, "Underdetermined blind source separation using sparse representation," *Signal Processing*, vol. 81, p. 2353–2362, Nov. 2001.
- [24] P. Georgiev, F. Theis, and A. Cichocki, "Sparse component analysis and blind source separation of underdetermined mixtures," *IEEE Transactions on Neural Networks*, vol. 16, p. 992–996, July 2005.
- [25] S. Arakia, H. Sawadaa, R. Mukaia, and S. Makino, "Underdetermined blind sparse source separation for arbitrarily arranged multiple sensors," *Signal Processing*, vol. 87, p. 1833–1847, Aug. 2007.
- [26] P. Bofill and M. Zibulevsky, "Blind separation of more sources than mixtures using the sparsity of the short-time fourier transform," in *2nd Int. Workshop on Independent Component Analysis and Blind Signal Separation*, p. 87–92, June 2000.
- [27] H. Sawada, S. Araki, R. Mukai, and S. Makino, "Blind extraction of dominant target sources using ICA and Time–Frequency masking," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, p. 2165–2173, Nov. 2006.
- [28] S. Araki, S. Makino, A. Blin, R. Mukai, and H. Sawada, "Underdetermined blind separation for speech in real environments with sparseness and ICA," in *Proceedings of the ICASSP*, p. iii–881–884, May 2004.
- [29] A. Aissa–El–Bey, K. Abed–Meraim, and Y. Grenier, "Blind separation of underdetermined convolutive mixtures using their time–frequency representation," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, p. 1540–1550,

July 2007.

- [30] A. Cichocki and S. Amari, *Adaptive Blind Signal and Image Processing Learning Algorithms and Applications*. John Wiley & Sons Ltd, New York, 2002.
- [31] A. Hyvarinen, J. Karhunen, and E. Oja, *Independent Component Analysis*. John Wiley & Sons Ltd, New York, 2001.
- [32] J. M. Mendel, "Tutorial on higher-order statistics (spectra) in signal processing and system theory: Theoretical results and some applications," *Proceedings of the IEEE*, vol. 79, p. 278–305, Mar. 1991.
- [33] J. F. Cardoso, "Higher-order contrasts for independent component analysis," *Neural Computation*, vol. 11, no. 1, p. 157–192, 1999.
- [34] Y. Li, P. Wen, and D. Powers, "Methods for the blind signal separation problem," in *Proceedings of the ICNNSP*, p. 1386–1389, 2003.
- [35] B. A. Pearlmutter and L. C. Parra, "Maximum likelihood blind source separation: A context-sensitive generalization of ICA," in *Proceedings of the NIPS'97*, p. 613–619, 1997.
- [36] J. F. Cardoso, "Blind signal separation: Statistical principles," *Proceedings of the IEEE*, vol. 86, p. 2009–2025, Oct. 1998.
- [37] Z. He, L. Yang, J. Liu, Z. Lu, C. He, and Y. Shi, "Blind source separation using clustering-based multivariate density estimation algorithm," *IEEE Transactions on Signal Processing*, vol. 48, p. 575–579, Feb. 2000.
- [38] S. Choi, A. Cichocki, and S. Amari, "Flexible independent component analysis," *Journal of VLSI Signal Processing*, vol. 26, no. 1–2, p. 25–38, 2000.
- [39] H. Sawada, R. Mukai, S. Araki, and S. Makino, "Polar coordinate based nonlinear function for frequency-domain blind source separation," *IEICE Trans. Fundamentals*, vol. E86–A, p. 1–7, Mar. 2003.
- [40] T. W. Lee and A. J. Bell, "Blind source separation of real world signals," in *Proceedings of the ICNN*, vol. 4, p. 2129–2134, 1997.
- [41] D. T. Pham, "Blind separation of instantaneous mixture of sources via an

- independent component analysis,” *IEEE Tran. on Signal Processing*, vol. 44, p. 2768–2779, Nov. 1996.
- [42] H. Attias, “New EM algorithms for source separation and deconvolution with a microphone array,” in *Proceedings of ICASSP’03*, vol. V, p. 297–300, Apr. 2003.
 - [43] C. Andrieu and S. Godsill, “A particle filter for model based audio source separation,” in *Proceedings of the Int. Symposium on Independent Component Analysis and Blind Signal Separation*, p. 381–386, 2000.
 - [44] J. R. Hopgood, “Bayesian blind MIMO deconvolution of nonstationary subband autoregressive sources mixed through subband all-pole channels,” in *Proceedings of the SSP’03*, p. 422–425, 2003.
 - [45] S. J. Godsill and C. Andrieu, “Bayesian separation and recovery of convolatively mixed autoregressive sources,” in *Proceedings of the ICASSP’99*, vol. 3, p. 1733–1736, 1999.
 - [46] H. Attias, “Source separation with a sensor array using graphical models and subband filtering,” in *Proceedings of the NIPS’02*, vol. 15, p. 1229–1236, 2002.
 - [47] S. Sanei, W. Wang, and J. A. Chambers, “A coupled HMM for solving the permutation problem in frequency domain BSS,” in *Proceedings of the ICASSP’04*, vol. 5, p. 565–568, 2004.
 - [48] A. Amari, “Natural gradient works efficiently in learning,” *Neural Computation*, vol. 10, no. 2, p. 251–276, 1998.
 - [49] J. F. Cardoso and B. H. Laheld, “Equivariant adaptive source separation,” *IEEE Transactions on Signal Processing*, vol. 44, p. 3017–3030, Dec. 1996.
 - [50] S. C. Douglas and M. Gupta, “Scaled natural gradient algorithm for instantaneous and convolutive blind source separation,” in *Proceedings of the ICASSP’07*, vol. 2, p. 637–640, 2007.
 - [51] P. Tichavský, Z. Koldovský, and E. Oja, “Performance analysis of the FastICA algorithm and Cramér–Rao bounds for linear independent component analysis,” *IEEE Transactions on Signal Processing*, vol. 54, p. 1189–1203, Apr. 2006.

- [52] E. Oja and Z. Yuan, "The FastICA algorithm revisited: Convergence analysis," *IEEE Transactions on Neural Networks*, vol. 17, p. 1370–1381, Nov. 2006.
- [53] Z. Koldovský, P. Tichavský, and E. Oja, "Efficient variant of algorithm FastICA for independent component analysis attaining the Cramer–Rao lower bound," *IEEE Transactions on Neural Networks*, vol. 17, p. 1265–1277, Sept. 2006.
- [54] J. Karhunen and J. Joutsensalo, "Representation and separation of signals using nonlinear PCA type learning," *Neural Networks*, vol. 7, no. 1, p. 113–127, 1994.
- [55] E. Oja, "The nonlinear PCA learning rule in independent component analysis," *Neurocomputing*, vol. 17, p. 25–45, Sept. 1997.
- [56] X. L. Zhu, X. D. Zhang, Z. Z. Ding, and Y. Jia, "Adaptive nonlinear PCA algorithms for blind source separation without prewhitening," *IEEE Transactions on Circuits and Systems –I*, vol. 53, p. 745–753, Mar. 2006.
- [57] X. Zhu, X. Zhang, and Y. Su, "A fast NPCA algorithm for online blind source separation," *Neurocomputing*, vol. 69, p. 964–968, Mar. 2006.
- [58] A. Hyvarinen, "New approximations of differential entropy for independent component analysis and projection pursuit," in *Proceedings of the Advances Neural Information Processing Systems*, p. 273–279, 1998.
- [59] J. Cardoso, "Infomax and maximum likelihood for blind source separation," *IEEE Signal Processing Letters*, vol. 4, p. 112–114, Apr. 1997.
- [60] M. Girolami and C. Fyfe, "Stochastic ICA contrast maximisation using Oja's nonlinear PCA algorithm," *International Journal of Neural Systems*, vol. 8, p. 661–678, Oct. 1997.
- [61] M. S. Pedersen, J. Larsen, U. Kjems, and L. C. Parra, *Springer Handbook of Speech Processing*. Springer Press, 2007.
- [62] J. V. Stone, "Blind source separation using temporal predictability," *Neural Computation*, vol. 13, no. 7, p. 1559–1574, 2001.
- [63] N. Murata, S. Ikeda, and A. Ziehe, "An approach to blind source separation

- based on temporal structure of speech signals,” *Neurocomputing*, vol. 41, p. 1–24, Oct. 2001.
- [64] B. Yin and P. Sommen, “Adaptive blind signal separation using a new simplified mixing model,” in *Proceedings of the RISC’99*, p. 601–606, 1999.
- [65] E. Visser and T. W. Lee, “Speech enhancement using blind source separation and two-channel energy based speaker detection,” in *Proceedings of the ICASSP*, vol. 1, p. 884–887, 2003.
- [66] E. Visser, K. Chan, S. Kim, and T. W. Lee, “A comparison of simultaneous 3-channel blind source separation to selective separation on channel pairs using 2-channel BSS,” in *Proceedings of the ICLSP’04*, vol. 4, p. 2869–2872, 2004.
- [67] M. Z. Ikram and D. R. Morgan, “A multiresolution approach to blind separation of speech signals in a reverberant environment,” in *Proceedings of the ICASSP’01*, vol. 5, p. 2757–2760, 2001.
- [68] M. Ikram and D. Morgan, “A beamforming approach to permutation alignment for multichannel frequency-domain blind speech separation,” in *Proceedings of the ICASSP’02*, p. 881–884, 2002.
- [69] S. C. Douglas, H. Sawada, and S. Makino, “A spatio-temporal fastICA algorithm for separating convolutive mixtures,” in *Proceedings of the ICASSP*, vol. 5, p. 165–168, 2005.
- [70] S. C. Douglas, M. Gupta, H. Sawada, and S. Makino, “A spatio-temporal fastica algorithm for blind separation of convolutive mixtures,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, p. 1511–1520, July 2007.
- [71] L. Zhang, A. Cichocki, and S. Amari, “Multichannel blind deconvolution of nonminimum-phase system using filter decomposition,” *IEEE Transactions on signal Processing*, vol. 52, p. 1430–1442, May 2004.
- [72] H. Buchner, R. Aichner, and W. Kellermann, “Blind source separation for convolutive mixtures exploiting nongaussianity, nonwhiteness and nonstationarity,”

- in *Proceedings of the International Workshop on Acoustic Echo and noise Control*, p. 275–278, 2003.
- [73] X. Sun and S. C. Douglas, “A natural gradient convolutive blind source separation algorithm for speech mixtures,” in *Proceedings of the Int. Symposium on Independent Component Analysis and Blind Signal Separation*, p. 59–64, 2001.
- [74] K. Kokkinakis and A. K. Nandi, “Multichannel blind deconvolution for source separation in convolutive mixtures of speech,” *IEEE transactions on Audio, Speech and Language Processing*, vol. 14, p. 200–212, Jan. 2006.
- [75] S. C. Douglas, H. Sawada, and S. Makino, “Natural gradient multichannel blind deconvolution and speech separation using causal FIR filters,” *IEEE Transactions on Speech and Audio Processing*, vol. 13, p. 92–104, Jan. 2005.
- [76] F. Asano, S. Ikeda, M. Ogawa, H. Asoh, and N. Kitawaki, “A combined approach of array processing and independent component analysis for blind separation of acoustic signals,” in *Proceedings of the ICASSP*, p. 2729–2732, 2001.
- [77] S. Makino, H. Sawada, R. Mukai, and S. Araki, “Blind source separation of convolutive mixtures of speech in frequency domain,” *IEICE Transactions Fundamentals*, vol. E88–A, p. 1640–1655, July 2005.
- [78] N. Mitianoudis and M. E. Davies, “Audio source separation of convolutive mixtures,” *IEEE Transactions on Speech and Audio Processing*, vol. 11, p. 489–497, Sept. 2003.
- [79] S. Ding, J. Huang, D. Wei, and A. Cichocki, “A near real-time approach for convolutive blind source separation,” *IEEE Transactions on Circuits and Systems I*, vol. 53, p. 114–128, Jan. 2006.
- [80] M. Joho and P. Schniter, “Frequency domain realization of a multichannel blind deconvolution algorithm based on the natural gradient,” in *Proceedings of the Int. Symposium on Independent Component Analysis and Blind Signal Separation*, p. 543–548, 2003.

- [81] H. Buchner, R. Aichner, and W. Kellermann, "A generalization of a class of blind source separation algorithms for convolutive mixtures," in *Proceedings of the Int. Symposium on Independent Component Analysis and Blind Signal Separation*, p. 945–950, 2003.
- [82] H. Buchner, R. Aichner, and W. Kellermann, "A generalization of blind source separation algorithms for convolutive mixtures based on second-order statistics," *IEEE Transactions on Speech and Audio Processing*, vol. 13, p. 120–134, Jan. 2005.
- [83] M. Joho, "Blind signal separation of convolutive mixtures: A time–domain joint–diagonalization approach," in *Proceedings of the Int. Symposium on Independent Component Analysis and Blind Signal Separation*, p. 578–585, 2004.
- [84] J. F. Cardoso and A. Souloumiac, "Blind beamforming for non–gaussian signals," *IEE Proceedings F*, vol. 140, p. 362–370, Dec. 1993.
- [85] M. Kawamoto, K. Matsuoka, and N. Ohnishi, "A method of blind separation for convolved nonstationary signals," *Neurocomputing*, vol. 22, p. 157–171, Nov. 1998.
- [86] T. Nishikawa, H. Saruwatari, and K. Shikano, "Comparison of time–domain ICA, frequency–domain ICA and multistage ICA for blind source separation," in *Proceedings of the EUSIPCO*, vol. 2, p. 15–18, 2002.
- [87] R. Aichner, H. Buchner, F. Yan, and W. Kellermann, "A real–time blind source separation scheme and its application to reverberant and noisy acoustic environments," *Signal Processing*, vol. 86, p. 1260–1277, June 2006.
- [88] K. E. Hild, D. Pinto, D. Erdogmus, and J. C. Principe, "Convolutive blind source separation by minimizing mutual information between segments of signals," *IEEE Transactions on Circuits and Systems –I*, vol. 52, p. 2188–2196, Oct. 2005.
- [89] D. Pham, "Mutual information approach to blind separation of stationary sources," *IEEE Transactions on Information Theory*, vol. 48, p. 1935–1946, July 2002.

- [90] R. Aichner, H. Buchner, F. Yan, and W. Kellermann, "Real-time convolutive blind source separation based on a broadband approach," in *Proceedings of the Int. Symposium on Independent Component Analysis and Blind Signal Separation*, p. 840–848, 2004.
- [91] H. Sawada, S. Araki, and S. Makino, "Measuring dependence of bin-wise separated signals for permutation alignment in frequency-domain BSS," in *IEEE Int. Symp. on Circuits and Systems*, p. 3247–3250, May 2007.
- [92] K. Rahbar and J. P. Reilly, "A frequency domain method for blind source separation of convolutive audio mixtures," *IEEE Transactions on Speech and Audio Processing*, vol. 13, p. 832–844, Sept. 2005.
- [93] S. Kurita, H. Saruwatari, S. Kajita, K. Takeda, and F. Itakura, "Evaluation of blind signal separation method using directivity pattern under reverberant conditions," in *Proceedings of the ICASSP*, p. 3140–3143, 2000.
- [94] H. Saruwatari, S. Kurita, K. Takeda, F. Itakura, T. Nishikawa, and K. Shikano, "Blind source separation combining independent component analysis and beamforming," *EURASIP Journal on Applied Signal Processing*, no. 11, p. 1135–1146, 2003.
- [95] S. Makino, H. Sawada, T. W. Lee, S. Araki, and W. Kellermann, *Blind Speech Separation*. Springer, 2007.
- [96] L. C. Parra and C. V. Alvino, "Geometric source separation: Merging convolutive source separation with geometric beamforming," *IEEE Transactions on Speech and Audio Processing*, vol. 10, p. 352–362, Sept. 2002.
- [97] H. Saruwatari, T. Kawamura, and T. Nishikawa, "Blind source separation based on a fast-convergence algorithm combining ICA and beamforming," *IEEE transactions on Audio, Speech and Language Processing*, vol. 14, p. 666–678, Mar. 2006.
- [98] R. Aichner, S. Araki, S. Makino, T. Nishikawa, and H. Saruwatari, "Time domain blind source separation of non-stationary convolved signals by utilizing

- geometric beamforming,” in *Proceedings of the Workshop on Neural Networks for Signal Processing*, p. 445–454, 2002.
- [99] S. Araki, S. Makino, Y. Hinamoto, R. Mukai, Y. Nishikawa, and H. Saruwatari, “Equivalence between frequency-domain blind source separation and frequency-domain adaptive beamforming for convolutive mixtures,” *EURASIP Journal on Applied Signal Processing*, no. 11, p. 1157–1166, 2003.
 - [100] M. Zibulevsky, Y. Y. Zeevi, P. Kisilev, and B. Pearlmutter, “Blind source separation via multinode sparse representation,” in *Advances in Neural Information Processing Systems 12*, p. 1049–1056, MIT Press, 2001.
 - [101] D. D. Lee and H. S. Seung, “Algorithms for non-negative matrix factorization,” *Advances in neural information processing systems*, vol. 13, p. 556–562, 2001.
 - [102] A. Jourjine, S. Rickard, and O. Yilmaz, “Blind separation of disjoint orthogonal signals: demixing N sources from 2 mixtures,” in *Proceedings of the ICASSP*, p. 2986–2988, June 2000.
 - [103] M. V. Hulle, “Clustering approach to square and non-square blind source separation,” in *IEEE Workshop on Neural Networks for Signal Processing*, p. 315–323, 1999.
 - [104] P. D. O’Grady and B. A. Pearlmutter, “Hard-LOST: Modified k-means for oriented lines,” in *Proc Irish Signals and Systems Conf*, p. 247–252, July 2004.
 - [105] J. MacQueen, “Some methods for classification and analysis of multivariate observations,” in *Proceedings of the 5th Berkeley Symp.*, vol. 1, p. 281–297, 1967.
 - [106] L. Vielva, D. Erdogmus, C. Pantaleon, I. Santamaria, J. Pereda, and J. Principe, “Underdetermined blind source separation in a time-varying environment,” in *Proceedings of the ICASSP*, vol. 3, p. 3049–3052, 2002.
 - [107] L. Vielva, D. Erdogmus, and J. Principe, “Underdetermined blind source separation using a probabilistic source sparsity model,” in *2nd Int. Workshop on Independent Component Analysis and Blind Signal Separation*, p. 675–679,

June 2000.

- [108] J. K. Lin, D. G. Grier, , and J. D. Cowan, "Feature extraction approach to blind source separation," in *IEEE Workshop on Neural Networks for Signal Processing*, p. 398–405, 1997.
- [109] I. Takigawa, M. Kudo, A. Nakamura, and J. Toyama, "On the minimum l_1 -norm signal recovery in underdetermined source separation," in *Fifth Int. Conf on Independent Component Analysis*, p. 193–200, Sept. 2004.
- [110] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, no. 1, p. 33–61, 1999.
- [111] R. Balan and J. Rosca, "Statistical properties of STFT ratios for two channel systems and applications to blind source separation," in *International Workshop on ICA and BSS*, p. 429–434, 2000.
- [112] S. Rickard and O. Yilmaz, "On the approximate W-disjoint orthogonality of speech," in *Proceedings of the ICASSP*, p. 13–17, May 2002.
- [113] F. Abrard and Y. Deville, "A time–frequency blind signal separation method applicable to underdetermined mixtures of dependent sources," *Signal Processing*, vol. 85, p. 1389–1403, July 2005.
- [114] A. Aissa-El-Bey, N. L. Trung, K. A. Meraim, A. Belouchrani, and Y. Grenier, "Underdetermined blind separation of nondisjoint sources in the time–frequency domain," *IEEE Transactions on Signal Processing*, vol. 55, p. 897–907, Mar. 2007.
- [115] N. Mitianoudis and T. Stathaki, "Batch and online underdetermined source separation using laplacian mixture models," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, p. 1818–1832, Aug. 2007.
- [116] Y. Deville and M. Puigt, "Temporal and time–frequency correlation-based blind source separation methods. Part I: Determined and underdetermined linear instantaneous mixtures," *Signal Processing*, vol. 87, p. 374–407, Mar. 2007.
- [117] Y. Luo, W. Wang, J. A. Chambers, S. Lambotharan, and I. Proudler, "Exploita-

- p>tion of source nonstationarity in underdetermined blind source separation with advanced clustering techniques,”
- IEEE Transactions on Signal Processing*
- , vol. 54, p. 2198–2212, June 2006.
- [118] R. Saab, O. Yilmaz, M. J. McKeown, and R. Abugharbieh, “Underdetermined anechoic blind source separation via l^q -Basis-Pursuit with $q < 1$,” *IEEE Transactions on Signal Processing*, vol. 55, p. 4004–4017, Aug. 2007.
 - [119] S. Winter, W. Kellermann, H. Sawada, and S. Makino, “Map-based underdetermined blind source separation of convolutive mixtures by hierarchical clustering and l_1 -norm minimization,” *EURASIP Journal on Advances in Signal Processing*, vol. 2007, p. 81–81, Jan. 2007.
 - [120] T. Melia and S. Rickard, “Underdetermined blind source separation in echoic environments using DESPRIT,” *EURASIP Journal on Applied Signal Processing*, vol. 2007, p. 2007, Jan. 2007.
 - [121] J. Thomas, Y. Deville, and S. Hosseini, “Time-domain fast fixed point algorithms for convolutive ICA,” *IEEE Signal Processing Letters*, vol. 13, p. 228–231, Apr. 2006.
 - [122] J. Thomas, Y. Deville, and S. Hosseini, “Differential fast fixed-point algorithms for underdetermined instantaneous and convolutive partial blind source separation,” *IEEE Transactions on Signal Processing*, vol. 55, p. 3717–3729, July 2007.
 - [123] V. Reju, S. N. Koh, I. Y. Soon, and X. Zhang, “Solving permutation problem in blind source separation of speech signals: A method applicable for collinear sources,” in *Proceedings of the ICICS*, p. 1461–1465, 2005.
 - [124] T. Nishikawa, *Blind source separation based on multistage independent component analysis*. PhD thesis, Nara Institute of Science and Technology, 2005.
 - [125] T. Nishikawa, H. Saruwatari, and K. Shikano, “Blind source separation of acoustic signal based on multistage ICA combining frequency-domain ICA and time-domain ICA,” *IEICE Transactions Fundamentals*, vol. E86–A, no. 4,

p. 846–858, 2003.

- [126] D. R. Campbell, “Roomsim user guide,” May 2004.
<http://bass-db.gforge.inria.fr/BASS-dB/?show=browse&id=filters>.
- [127] P. Smaragdis, “Efficient blind separation of convolved sound mixtures,” in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 1997.
- [128] J. R. Hopgood, P. J. W. Rayner, and P. W. T. Yuen, “The effect of sensor placement in blind source separation,” in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, (New Paltz, NY), p. 95–98, Oct. 2001.
- [129] <http://www.purebits.com/>.
- [130] S. Araki, R. Mukai, S. Makino, T. Nishikawa, and H. Saruwatari, “The fundamental limitation of frequency domain blind source separation for convolutive mixtures of speech,” *IEEE Transactions Speech Audio Process.*, vol. 11, p. 109–116, Mar. 2003.
- [131] L. T. Nguyen, A. Belouchrani, K. A. Meraim, and B. Boashash, “Separating more sources than sensors using time–frequency distributions,” in *International Symposium on Signal Processing and its Applications*, p. 583–586, Aug. 2001.
- [132] C. Févotte and C. Doncarli, “Two contributions to blind source separation using Time–Frequency distributions,” *IEEE Signal Processing Letters*, vol. 11, p. 386–389, Mar. 2004.
- [133] D. Smith, J. Lukasiak, and I. S. Burnett, “A two channel, block–adaptive audio separation technique based upon time–frequency information,” in *In Proc. of the 12th European Signal Processing Conf.*, p. 393–396, 2004.
- [134] P. Bofill, “Identifying single source data for mixing matrix estimation in instantaneous blind source separation,” in *ICANN (1)*, p. 759–767, 2008.
- [135] M. Xiao, S. Xie, and Y. Fu, “A novel approach for underdetermined blind sources separation in frequency domain,” *Advances in Neural Networks –ISNN 2005*, vol. 2005, p. 484–489, May 2005.

- [136] X. Ming, X. ShengLi, and F. YuLi, "Searching-and-averaging method of under-determined blind speech signal separation in time domain," *Science in China Series F: Information Sciences*, vol. 50, p. 771–782, Oct. 2007.
- [137] Y. Deville, M. Puigt, and B. Albouy, "Time-frequency blind signal separation : extended methods, performance evaluation for speech sources," in *Proceedings of the International Joint Conference on Neural Networks*, p. 255–260, 2004.
- [138] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Computing Surveys*, vol. 31, p. 264–323, Sept. 1999.
- [139] R. Xu and Donald Wunsch II, "Survey of clustering algorithms," *IEEE Transactions on Neural Networks*, vol. 16, p. 645–678, May 2005.
- [140] A. Cichocki, S. Amari, K. Siwek, T. Tanaka, and A. H. P. et al, "ICALAB Toolboxes." <http://www.bsp.brain.riken.jp/ICALAB>.
- [141] L. Tong, V. Soon, Y. F. Huang, and R. Liu, "Indeterminacy and identifiability of blind identification," *IEEE Transactions on Circuits and Systems*, vol. 38, p. 499–509, Mar. 1991.
- [142] L. Tong, Y. Inouye, and R. Liu, "Waveform-preserving blind estimation of multiple independent sources," *IEEE Transactions on Signal Processing*, vol. 41, p. 2461–2470, July 1993.
- [143] P. Georgiev and A. Cichocki, "Blind source separation via symmetric eigenvalue decomposition," in *in Proceedings of Sixth International Symposium on Signal Processing and its Applications*, p. 17–20, Aug. 2001.
- [144] P. Georgiev and A. Cichocki, "Robust blind source separation utilizing second and fourth order statistics," in *in Proceedings of International Conference on Artificial Neural Networks*, p. 1162–1167, Aug. 2002.
- [145] A. Belouchrani, K. Abed-Meraim, J. F. Cardoso, and E. Moulines, "Second-order blind separation of temporally correlated sources," in *in Proceedings of International Conference on Digital Signal Processing*, p. 346–351, 1993.
- [146] L. Molgedey and G. Schuster, "Separation of a mixture of independent signals

- using time delayed correlations,” *Physical Review Letters*, vol. 72, no. 23, p. 3634–3637, 1994.
- [147] A. Ziehe and K. Muller, “TDSEP –an efficient algorithm for blind separation using time structure,” in *Proceedings of ICANN’98*, p. 675–680, 1998.
- [148] A. Belouchrani and A. Cichocki, “Robust whitening procedure in blind source separation context,” *Electronics Letters*, vol. 36, p. 2050–2051, Nov. 2000.
- [149] A. Cichocki and A. Belouchrani, “Sources separation of temporally correlated sources from noisy data using a bank of band-pass filters,” in *Proceedings of Third International Conference on Independent Component Analysis and Signal Separation*, p. 173–178, Dec. 2001.
- [150] A. Cichocki, T. Rutkowski, and K. Siwek, “Blind signal extraction of signals with specified frequency band,” in *Proceedings of the 12th workshop on Neural Networks for Signal Processing*, p. 515–524, 2002.
- [151] R. R. Gharieb and A. Cichocki, “Second order statistics based blind source separation using a bank of subband filters,” *Digital Signal Processing*, vol. 13, p. 252–274, Apr. 2003.
- [152] S. Choi and A. Cichocki, “Blind separation of nonstationary sources in noisy mixtures,” *Electronics Letters*, vol. 36, p. 848–849, Apr. 2000.
- [153] S. Choi and A. Cichocki, “Blind separation of nonstationary and temporally correlated sources from noisy mixtures,” in *Proceedings of the IEEE Workshop on Neural Networks for Signal Processing*, p. 405–414, Dec. 2000.
- [154] J. Cardoso and A. Souloumiac, “Jacobi angles for simultaneous diagonalization,” *SIAM Journal of Matrix Analysis and Applications*, vol. 17, p. 161–164, Jan. 1996.
- [155] P. Georgiev and A. Cichocki, “Robust independent component analysis via time-delayed cumulant functions,” *IEICE Transactions on Fundamentals*, vol. E86–A, p. 573–579, Mar. 2003.
- [156] A. Hyvarinen and E. Oja, “A fast fixed-point algorithm for independent compo-

- nent analysis,” *Neural Computation*, vol. 9, p. 1483–1492, Oct. 1997.
- [157] A. Hyvarinen and E. Oja, “Independent component analysis: Algorithms and applications,” *Neural Networks*, vol. 13, p. 411–430, June 2000.
 - [158] S. Amari, T. Chen, and A. Cichocki, “Non-holonomic constraints in learning blind source separation,” in *Proceedings of the ICONIP*, vol. 1, p. 633–636, 1997.
 - [159] S. Amari, T. Chen, and A. Cichocki, “Nonholonomic orthogonal learning algorithms for blind source separation,” *Neural Computation*, vol. 12, no. 6, p. 1463–1484, 2000.
 - [160] S. Choi and A. Cichocki, “Flexible independent component analysis,” in *Proceedings of the IEEE Workshop on NNSP*, p. 83–92, 1998.
 - [161] S. Cruces and A. Cichocki, “Combining blind source extraction with joint approximate diagonalization: Thin algorithms for ICA,” in *Proceedings of the Fourth Symposium on Independent Component Analysis and Blind Signal Separation*, p. 463–469, 2003.
 - [162] L. D. Lathauwer, P. Comon, B. De-Moor, and J. Vandewalle, “Higher-order power method –application in independent component analysis,” in *Proceedings of the International Symposium on Nonlinear Theory and its Applications*, p. 91–96, 1995.
 - [163] S. Cruces, L. Castedo, and A. Cichocki, “Robust blind source separation algorithms using cumulants,” *Neurocomputing*, vol. 49, p. 87–117, Dec. 2002.
 - [164] S. Cruces, L. Castedo, and A. Cichocki, “Novel blind source separation algorithms using cumulants,” in *Proceedings of the ICASSP*, vol. V, p. 3152–3155, 2000.
 - [165] S. Amari, “Natural gradient learning for over-and under-complete bases in ICA,” *Neural Computation*, vol. 11, p. 1875–1883, Nov. 1999.
 - [166] S. Cruces, A. Cichocki, and S. Amari, “Criteria for the simultaneous blind extraction of arbitrary groups of sources,” in *Proceedings of the 3rd international conference on Independent Component Analysis and Blind Signal Separation*,

p. 740–745, 2001.

- [167] S. Cruces, A. Cichocki, and S. Amari, “The minimum entropy and cumulant based contrast functions for blind source extraction,” in *Lecture Notes in Computer Science, Springer-Verlag, IWANN’2001*, vol. II, p. 786–793, 2001.
- [168] S. Cruces, A. Cichocki, and S. Amari, “On a new blind signal extraction algorithm: different criteria and stability analysis,” *IEEE Signal Processing Letters*, vol. 9, p. 233–236, Aug. 2002.
- [169] S. Cruces, A. Cichocki, and L. Castedo, “Blind source extraction in Gaussian noise,” in *Proceedings of the 2nd International Workshop on Independent Component Analysis and Blind Signal Separation*, p. 63–68, 2000.
- [170] A. V. Oppenheim, R. W. Schaffer, and J. R. Buck, *Discrete-Time Signal Processing*. Prentice Hall, 2003.
- [171] G. Xu, H. Liu, L. Tong, and T. Kailath, “A least-squares approach to blind channel identification,” *IEEE Transactions on Signal Processing*, vol. 43, p. 2982–2993, Dec. 1995.
- [172] A. Aissa-El-Bey, M. Grebici, K. Abed-Meraim, and A. Belouchrani, “Blind system identification using cross-relation methods: Further results and developments,” in *Proceedings of the Int. Symp. Signal Process. Applicat.*, p. 649–652, July 2003.
- [173] K. Scharnhorst, “Angles in complex vector spaces,” *Acta Applicandae Mathematicae*, vol. 69, p. 95–103, Nov. 2001.
- [174] H. Sawada, S. Araki, and S. Makino, “A two-stage frequency-domain blind source separation method for underdetermined convolutive mixtures,” in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, p. 139–142, Oct. 2007.
- [175] J. C. Bezdek, *Pattern Recognition With Fuzzy Objective Function Algorithms*. New York: Plenum Press, 1981.
- [176] D. Arthur and S. Vassilvitskii, “ k -means++: The advantages of careful seeding,”

- in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, p. 1027–1035, 2007.
- [177] Y. Zhang, W. Wang, X. Zhang, and Y. Li, “A cluster validity index for fuzzy clustering,” *Information Sciences*, vol. 178, p. 1205–1218, Feb. 2008.
 - [178] H. Sun, S. Wang, and Q. Jiang, “FCM-based model selection algorithms for determining the number of clusters,” *Pattern Recognition*, vol. 37, p. 2027–2037, Oct. 2004.
 - [179] M. K. Pakhira, S. Bandyopadhyay, and U. Maulik, “Validity index for crisp and fuzzy clusters,” *Pattern Recognition*, vol. 37, p. 487–501, Mar. 2004.
 - [180] P. Guo, C. L. P. Chen, and M. R. Lyu, “Cluster number selection for a small set of samples using the bayesian Ying–Yang model,” *IEEE Transactions on neural networks*, vol. 13, p. 757–763, May 2002.
 - [181] E. Vincent, R. Gribonval, and C. Fevotte, “Performance measurement in blind audio source separation,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, p. 1462–1469, July 2006.
 - [182] C. Fevotte, R. Gribonval, and E. Vincent, “BSS EVAL toolbox user guide, IRISA technical report 1706,” tech. rep., Rennes, France, Apr. 2005. http://www.irisa.fr/metiss/bss_eval/.
 - [183] J. Rosca, T. Gerkmann, and D. C. Balcan, “Statistical inference of missing speech data in the ICA domain,” in *Proceedings of the ICASSP*, vol. 5, p. 617–620, May 2006.
 - [184] S. A. Martucci, “Symmetric convolution and the discrete sine and cosine transforms,” *IEEE Transactions on Signal Processing*, vol. 42, p. 1038–1051, May 1994.
 - [185] S. A. Martucci, *Symmetric convolution and the discrete sine and cosine transforms: Principles and applications*. PhD thesis, Georgia Institute of Technology, Atlanta, 1993.
 - [186] X. Zou, S. Muramatsu, and H. Kiya, “Generalized overlap–add and overlap–save

methods using discrete sine and cosine transforms for FIR filtering,” in *Proc. ICSP*, vol. 1, p. 91–94, 1996.

- [187] Proakis and G. John, *Digital signal processing : principles, algorithms, and applications*. Prentice–Hall, 1996.
- [188] S. A. Martucci, “Digital filtering of images using the discrete sine or cosine transform,” in *Proc. SPIE*, vol. 2308, p. 1322–1333, 1994.