Blind Separation of Convolved Sources Using the Independent Component Analysis and Information Maximization Approach

Md. Hasanuzzaman

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Master of Applied Science in Electrical Engineering at

Concordia University

Montreal, Quebec, Canada

July 2005

Your file Votre référence
ISBN: 0-494-10235-7
Our file Notre référence
ISBN: 0-494-10235-7

Abstract

Blind Separation of Convolved Sources Using the Independent Component Analysis and Information Maximization Approach

Md. Hasanuzzaman

Independent Component Analysis (ICA) is very closely related to the method called blind source separation (BSS) or blind signal separation. In Independent Component Analysis (ICA) components are assumed statistically independent which we call independent source signal. In our thesis we have considered only noiseless ICA case. In a number of real-world signal processing applications, signals from various independent sources may get distorted by environmental factors that can be represented as convolutive mixtures of original signals received at the sensors. In this thesis, the effects of environmental factors and modeling assumptions on the performance capabilities of independent component analysis-based techniques are investigated. The so-called blind source separation feedback network architecture that is capable of coping with convolutive mixtures of sources is derived using Bell and Sejnowski's information maximization principle. We developed ideal solutions for separation of independent source signals from the convolutive mixtures that is applicable to an arbitrary $N \times N$ feedback network architecture. A number of simulation case studies corresponding to various types of environment filters are presented using synthetically generated data. Different kinds of filter structures have been taken into accounts and the adaptation rules for those

filters have been derived based on information maximization principle. Also how the distribution of filters as well as the order of the mixing environmental filters affect the quality of the recovered signals have also been investigated. The effect of noise at each sensor and also the effect of SNR while generating the synthetic data have been taken into consideration. The location of poles and zeros of the mixing filters and the initial values of demixing filters have significant effect on the stability of the whole system. Measures have been taken to keep the whole system stable and workable. Constant values of adaptation rate have been used in different epochs.

Acknowledgements

This work would not have been possible without the help of several individuals and institution. I am especially thankful to my supervisor, Professor K. Khorasani, for his advice, encouragement, guideline and support while I pursued this thesis. Professor K. Khorasani provided much of the initial motivation for pursuing this investigation and also provided invaluable feedback that has improved this work in nearly every respect. Financial support for this research was provided by my Professor K. Korasani. I would like to convey my deepest gratitude to him for his research grants. I would also like to thank the Department of Electrical and Computer Engineering, Concordia University for using its lab without which I would not be able to perform my work.

I also owe especially my parents and family members a great deal of gratitude for their strong support of my higher education in Canada.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Representing the observation data in suitable way by using a suitable transformation is the central problem in neural network research, statistics and signal processing. The data is represented in a manner which facilitates the subsequent analysis of the data [1]. There are many applications of this analysis of this data, e.g., in data compression, pattern recognition, de-noising visualization or some other areas.

Let us assume $\mathbf{x}$ to be a random variable of dimension m and $\mathbf{s}$ to be a random variable of dimension n. Then our basic goal is to find a function $\mathbf{f}$ such that

$$\mathbf{s} = \mathbf{f}(\mathbf{x}) \tag{1.1}$$

where the transformed vector $\mathbf{s} = (s_1, s_2, s_3, ..., s_n)^T$ of dimension n will have some desirable properties. Our aim is to represent $\mathbf{s}$ as a linear transform of the observed variables $\mathbf{x}$ which is as follows:

$$\mathbf{s} = \mathbf{W}\mathbf{x} \tag{1.2}$$

where the matrix $\mathbf{W}$ needs to be determined. Finding a suitable linear trasnformation is necessary because using linear transformations makes the problem computationally

simpler. It also makes the problem conceptually easier and facilitates the interpretation of the results. There are several methods developed to find a suitable linear transformation, e.g., principal component analysis, factor analysis, projection pursuit, independent component analysis and some others [1]. The optimality of the linear transformation can be defined by some indexes, e.g., by the optimal dimension reduction $\mathbf{W}$, statistical "interestingness" of the estimated components s_i, simplicity and computational complexity of the transformation $\mathbf{W}$ or may be some other criteria.

Independent component analysis (ICA) is one of the most popular methods for finding a linear transformation. There are many applications of independent component analysis and it has drawn wide-spread attention for its application in different areas [1]. In ICA the main objective is to find a transformation such that the components s_i are made statistically as independent from each other as possible. Some major applications of ICA are the blind source separation, feature extraction, redundency reduction, exploratory data analysis as well as projection pursuit. In blind source separation, the components $s_i(t)$ are called source signals or original signals. In fact, these signals are the uncorrupted signals which are unknown. The only known variables are the observed values of $\mathbf{x}$ which is a discrete-time signal $\mathbf{x}(t)$, $t = 1, 2, ...$ of dimension m. We, then, try to recover the original source signals from the linear mixtures of the observed variables x_i by using a transformation so that the transformed signals are statistically as independent from each other as possible. Feature extraction is another major application of ICA which is widely used in neurosciences, in which s_i is the coefficient of the i-th feature in the observed vector $\mathbf{x}$. We shall discuss some second-order and higher-order methods in brief in the next section [1].

1.1 Classical linear transformations

There are varieties of use of classical linear transformations in statistics, neuroscience, and statistical and geoseismic signal processing. The basic goal in these linear transformations is to find a suitable linear representation of a random variable. There are some

methods which have been developed to find this linear transformation. In this section, we shall discuss some second-order and higher-order classical methods for finding the linear trasnformations. We shall use centered variables in all the methods discussed in this section . The centered variables are obtained by subtracting the means of the random vector from its original values. Let us denote by $\mathbf{x_0}$ as the original non-centered variable. Then the centered variable $\mathbf{x}$ can be computed as $\mathbf{x} = \mathbf{x_0} - E\{\mathbf{x_0}\}$.

1.1.1 Second-order methods

Because of the simplicity of the computation, second-order methods are the most popular methods for finding a linear transform as in equation (1.2). Very often it requires only classical matrix manipulations. As the name implies only second-order information is required to find the linear transformation in this classical methods. The distribution of a random variable, $\mathbf{x}$, is fully determined by the information contained in the covariance of the variable $\mathbf{x}$ if the variable, $\mathbf{x}$, has a normal, or Gaussian distribution.

In most of the higher-order methods [2–5] the aim is to find a meaningful representation of the data. But the second-order methods try to find a faithful representation of the data, which is unlike higher-order methods. In this section we will discuss two classical second-order methods which are principal component analysis and factor analysis [6–8].

Principal component analysis

There are numerous applications of principal component analysis, or PCA (see [7, 8]), in statistics, signal processing, and neural computing for its simplicity of computation. In PCA usually the dimension of the observation data is taken higher than that of original source signals, $m >> n$. Then the aim in PCA is to reduce the dimension of the data in the mean-square sense [8]. There are several advantages of this optimal dimensional reuction technique. First, the computation is reduced in the subsequent processing stages. Second, since the last $m - n$ components are mostly due to the

noise, therefore, noise reduction is also achieved in this method. Third, sometimes it is useful to find the projection into a very small dimensional subspace to visualize the data.

We can define PCA by using a recursive formula. According to this formula, $\mathbf{w}_1$, the direction of the first principal component is given by:

$$\mathbf{w}_1 = \arg \max_{||\mathbf{w}||=1} E\{(\mathbf{w}^T\mathbf{w})^2\} \tag{1.3}$$

Infact, $\mathbf{w}_1$ gives the projection on the direction in which the variance of the projection is maximized. The k-th principal component is computed as the principal component of the residual as follows after determining the first $k-1$ principal components:

$$\mathbf{w}_1 = \arg \max_{||\mathbf{w}||=1} E\{[\mathbf{w}^T(\mathbf{x} - \sum_{i=1}^{k-1} \mathbf{w}_i\mathbf{w}_i^T\mathbf{x})]^2\} \tag{1.4}$$

Having determined $\mathbf{w_i}$, the principal components are computed according to the formulla $s_i = \mathbf{w}_i^T\mathbf{x}$. But in practice principal components are computed from the eigen vectors, $\mathbf{w_i}$, of the covariance matrix $\mathbf{C}$, where $\mathbf{C} = E\{\mathbf{x}\mathbf{x}^T\}$. In this case, $\mathbf{w_i}$ corresponds to the n largest eigenvalues of the covariance matrix $\mathbf{C}$.

Factor analysis

Factor analysis [6, 7] and PCA are closely related in the sense that both of these methods are used as dimension reduction techniques. The basic difference has been made by adding a noise vector n in the following generative model:

$$\mathbf{x} = \mathbf{A}\mathbf{s} + \mathbf{n} \tag{1.5}$$

where both the noise vector $\mathbf{n}$ and the vector of observed variable $\mathbf{x}$ have the same dimension that is m. $\mathbf{s}$ is the vector of the latent variables, i.e., variables that can not be

observed or unknown. In order to apply this method both the latent variables of $\mathbf{s}$ and noise vector $\mathbf{n}$ must be Gaussian. Because all of the information of Gaussian variables are contained in the covariance matrix which is the second order central moment. Also the dimension of $\mathbf{s}$ should be lower than the dimension of $\mathbf{x}$, otherwise dimension reduction will not be possible.

There are two cases we can consider, the first one is when the noise covarince matrix of the noise vector is known and the second one is when the noise covarince matrix is unknown. For the first case, usually there are two methods to perfrom the factor analysis [7]. The method of principal factors is the first method where PCA is applied on the data $\mathbf{x}$ so that the noise effect is considered. In this method it is assumed that the covariance matrix of noise vector, $\sum$, is known where $\sum = E\{\mathbf{n}\mathbf{n}^T\}$. At first the covariance matrix, $\mathbf{C}$, of the observed vector, $\mathbf{x}$, is computed and $\sum$ is subtracted from it to give out the factors. The second method is based on the maximum likelihood estimation in which the principal components are estimated by finding the modified covariance matrix, i.e., $\mathbf{C} - \sum$. In the second case when the noise covarince matrix is unknown, some other methods [6, 7] have been used to estimate the components.

1.1.2 Higher-order methods

PCA and other second-order statistical methods use the covarince matrix to estimate the data model because in these techniques data is assumed to be Gaussian and the whole information of Gaussian variables are contained in their covariance matrix. But if the data is non-Gaussian, determining their covariance matrices will not provide the full information about the observed data. For an instance, PCA does not consider the independence of the data which is not same as the uncorrelatedness for non-Gaussian data. To consider such aspects of non-Gaussian data, e.g., clustering and independence of components higher-order statistics is needed. In this section we will give an overview of three widely used higher-order statistical methods for estimating the model of the non-Gaussian data. These are projection pursuit, redundancy reduction, and blind deconvolution:

Projection pursuit

In projection pursuit [2, 9–13] the main objective is to reduce the dimension in such a way that some of the "interesting" features of the data are preserved which is in contrast to PCA where the objective is to reduce the dimension so that the representation is as faithful as possible in the mean-square sense.

In statistics projection pursuit has been developed for finding "interesting" projections of multidimensional data. This projections has the use in the optimal visualization of the clustering structure of the data, density estimation and regression. It can also be used for the dimension reduction if the objective is the visualizations of the data.

According to Huber [10] and Jones and Sibson [11] the least Gaussian distribution show the most interesting direction of $\mathbf{w}$ such that the projection of the data in that direction, $\mathbf{w}^T\mathbf{x}$, has an "interesting" distribution, i.e., displays some structure and the Gaussian distribution is the least interesting one.

In projection pursuit the "interestingness" of a direction is defined by the projection pursuit index which is some measure of non-Gaussianity. Differential entropy [10, 11] can be the most natural choice to define the projection pursuit index. Let us denote by $\mathbf{y}$ a random vector of density $f(.)$ and fixed covariance. Then the differential entropy H of $\mathbf{y}$ is defined as:

$$H(\mathbf{y}) = -\int f(\mathbf{y}) \log f(\mathbf{y}) d\mathbf{y} \tag{1.6}$$

If f is a Gaussian density then the differential entropy $H(\mathbf{y})$ is maximized with respect to $\mathbf{w}$. Differential entropy is very small if $\mathbf{y}$ has different distribution. In that case the directions of projection pursuit is found by minimizing $\mathbf{w}^T\mathbf{x}$ with respect to $\mathbf{w}$, constraining the variance of $\mathbf{w}^T\mathbf{x}$ to be constant.

From the definition the estimation of entropy requires the estimation of the density of $\mathbf{w}^T\mathbf{x}$, which is the problem with differential entropy. Theoratically and practically this is difficult. To solve this problem other measures of non-normality based on weighted L^2 distances between the density of x and the multivariate Gaussian density

have been proposed [2, 12]. Two other methods are cumulant-based approximations of differential entropy [11]. Also approximations of negentropy based on the maximum entropy principle has also been used in [55].

Redundancy reduction

Redundancy reduction is an important characteristic of sensory processing in the brain which has been described by Barlow [15–17, 19, 20] and some other authors [21–24]. The basic goal in redundancy reduction is to represent the input data in terms of components or features such that the components are made independent of each other as possible. For later processing stages this kind of representation of data is very useful. According to theory, the activities of the neurons represent the values of the components. In this method, the sum of the weight vectors of the neurons, which are weighted by their activations, are represented by the observed vector $\mathbf{x}$.

There are two methods for performing the redundency reduction. The first method is sparse coding and the second one is predictibility minimization. In sparse coding [16, 20, 22] the data $\mathbf{x}$ is represented using a set of neurons and everytime only a small number of neurons is activated. In another word, only a single neuron is activated very rarely. The data is said to be "sparse" if it has certain statistical properties and redundancy reduction will be obtained roughly by this kind of coding [22]. Predictability minimization [23] is the second method for redundancy reduction where if two random variables are independent of each other, the information obtained from one variable can not be used to predict the other variable.

Blind deconvolution

Blind deconvolution is an important application of Independent Component Analysis, which has wide variety of application in signal processing. Here, we shall discuss about this topic in brief since our work in this thesis is based on this method.

In blind deconvolution, only the observed signal $x(t)$ is available which is obtained by the convolution of original source signal $s(t)$, which is unknown [25–31].

Then using the convolved mixture $x(t)$ we try to find the separating filter h so that $s(t) = h(t) * x(t)$.

The filter $h(t)$ may be FIR (Finite Impulse Response) or IIR (Infinite Impulse Response) type. To ignore the truncation effects one can assume $h(t)$ to be a FIR (Finite Impulse Response) filter of sufficient length. If we assume that at two different points of time the values of the signal $s(t)$ are statistically independent then only by whitening the signal $x(t)$ with some assumptions we can solve the problem. In that case, the signal $s(t)$ should be assumed to be non-Gaussian, and we shall have to use the higher-order statistics [27, 29] to estimate the model.

1.2 Literature review

There are many papers till now on blind source separation and independent component analysis. In these papers different aspects of blind source separation and independent component analysis have been discussed. Also different issues have been investigated and different algorithms have been proposed and used to solve those problems.

In [32] the complete details of a procedure for determining the average steady-state mean square error of a blind source separation algorithm have been given. The procedure has been applied to nine existing blind source separation criteria.

In [33] a simple LMS (Least Mean Square) learning algorithm for a fully recurrent convolutive blind source separation with transmission delay constraint has been proposed. In the mixing process some assumptions are imposed on transmission delay time, which are practically acceptable. A cost function and the learning algorithm have been derived based on this assumption.

In [34] a new low-cost design and implementation of an improved BSS algorithm for audio signals based on ICA technique has been proposed. It is performed by implementing non-causal filters instead of causal filters within the feedback network of

the ICA based BSS method. As a result, the required length of the unmixing filters has been reduced considerabley. The new design provides better results and faster convergence compared to the case with conventional causal filters. System level approach to the design of FPGA (Field Programmble Gate Array) prototype is adopted.

Blind source separation has also been done in frequency domain. A solution to the classical cocktail-party problem can be the blind source separation of acoustic mixtures aims. The inherent delays and convolutions in microphone recordings, adds a modification in the Independent Component Analysis. The separation of the source signal has been achieved by making the assumption of statistical independence of the linearly combined source signals. In [35] by shifting the domain of the ICA to Time-Frequency domain and applying ICA to each of the frequency components individually the proposed algorithm has provided a solution for the blind source separation problem .

In [36] a new fast-convergence algorithm for blind source separation of real convolutive mixture has been evaluated in which independent component analysis and beamforming are combined to resolve the low-convergence problem through optimization in ICA. The proposed method consists of the following three parts: (1) frequency-domain ICA with direction-of-arrival (DOA) estimation, (2) null beamforming based on the estimated DOA, and (3) integration of (1) and (2) based on the algorithm diversity in both iteration and frequency domain. The matrix based on null beamforming through iterative optimization has temporally substituted the inverse of the mixing matrix obtained by ICA. The temporal alteration between ICA and beamforming can realize fast and high convergence optimization.

In [37] a new and quickly converging algorithm which uses an alternating least-square (ALS) optimization method has been used to perform the frequency domain joint diagonalization. It is first shown that joint diagonalization of the cross power spectral density matrices of the signals at the output of the mixing system is sufficient to identify the mixing system at each frequency bin up to a scale and permutation ambi-

guity. The effect of the unknown scaling ambiguities is partially resolved using a novel initialization procedure for the ALS algorithm. The frequency dependent permutation ambiguities has been resolved by using an efficient dyadic algorithm.

Some papers have investigated the case of moving targets. A robust real-time blind source separation (BSS) method for moving speech signals in a room has been described in [38]. In the first stage, the frequency domain independent component analysis (ICA) is employed using blockwise batch algorithm. In the second stage, the separated signals are refined by post processing using crosstalk component estimation and non-stationary spectral subtraction. When sources are fixed, the blockwise batch algorithm achieves better performance than an online algorithm, and the performance degradation caused by source movement has been compensated by the postprocessing of the separated signals.

For the growing multimedia systems, under a noisy environment, more efficient signal separation methods are required to preserve the quality of voice or music recording. In [39] an improved method using differential information has been proposed. Since a noise component is usually independent on the other signals some of signal separation methods are based on minimizing the dependent measure among input signals to separate a noise component. A new genetic algorithm (GA) which directly minimizes the Kullback-Leibler divergence is proposed to separate independent signal components.

In our thesis we have used mutual information for linear independent component analysis. But some papers have also worked on nonlinear independent component analysis because this domain of independent component analysis are wide open for research. In [40] a single network with a specialized structure, trained with a single objective function is used for both the extraction of independent components and the estimation of their distributions simultaneously.

A fast algorithm for mutual information based independent component analysis has been developed in [41]. The binning technique and the use of cardinal splines allow the fast computation of the density estimator over a regular rid. The criterion can evaluated quickly together with its gradient using a discretized form of the entropy and this can be expressed in terms of the score functions. Both offline and online separation algorithms have been developed.

Typically in multichannel blind deconvolution and convolutive blind source separation system practical gradient-based adaptive algorithms employ FIR (Finite Impulse Response) filters for the separation of signals. The use of the FIR (Finite Impulse Response) filters cause the inadequate use of the signal truncation within algorithms which introduce steady-state biases into their converged solutions that lead to degraded separation and deconvolution performances. To mitigate these effects a natural gradient multichannel blind deconvolution and source algorithm has been developed in [42] The algorithm functions in a reasonable manner even when the filter lengths chosen are much shorter than would be required for an accurate channel inverse.

In most of the practical cases environment between source signals and sensors are not fixed, i. e., environment changes with time. These can be represented by time-varying FIR (Finite Impulse Response) system. A new adaptive blind separation scheme for sources mixed by a multiple-input multiple-output (MIMO) linearly time-varying (LTV) FIR (Finite Impulse Response) system is proposed in [43]. In the first stage, measured samples have been divided into a series of short segments. Then time-varying coefficients of the mixing system are approximated by polynomials in time over each segment. In the second stage, a two-step BSS scheme is presented. In the first step, the conventional input/output system indentification scheme is used to estimate the time variation and convolution effects of the mixing system, and reduce the LTV-FIR (Finite Impulse Response) mixing system to a linearly time-invariant (LTI) instantenous system. In the second step the mutual independence knowledge of the sources is used to further separate the sources from the LTI instantenous system.

There are enormous factors which have been considered in blind source separation. Blind source separation has also been done using time-delayed signals [44]. A modified version of AMUSE,, called dAMUSE, has been proposed in [44]. The dimension of the data vectors is increased by joining delayed versions of the observed mixed signals. The new data is used to compute a matrix pencil and its generalized eigendecomposition is performed as in AMUSE.

The performance of blind source separation depends on various factors. On of the factors is contrast function. The are different kinds of contrast functions that have been used in different papers. In [45] a sinusoidal contrast function for the blind separation of statistically independent sources has been used. In the two-dimensional (2-D) case, one can prove that, under the whitness constraint, the fourth-order moment-based approximation of the marginal entropy (ME) cost function yields a sinusoidal objective function. Therefore, minimization of the new objective function is possible by estimating only its phase.

In a most recent paper [46] a general broadband approach to blind source separation (BSS) for convolutive mixtures based on second-order statistics has been presented. The concept is applicable to offline, online, and block-online algorithms by introducing a general weighting function allowing for tracking of time-varying real acoustic environments. The new framework simultaneously exploits the nonwhiteness property and nonstationary property of the source signals which is in contrast to traditional narrow-band approaches. Constraints are obtained based on the broadband approach time-domain. These constraints provide a deeper understanding of the internal permutation problem in traditional narrowband frequency-domain BSS. Links between the time-domain and the frequency-domain algorithms can be established using the so-called generalized coherence. The cost function leads to an update equation with an inherent normalization ensuring a robust adaptation behavior.

1.3 Contribution of this thesis

In this thesis we have presented the ideal solution for $N \times$ N feedback network architecture. Also how the distribution of filters and order of the mixing environmental filters affect the quality of recovered independent sources have been investigated. Effects of pole-zero location of mixing filters, noise at each sensor and SNR while generating the synthetic data have all been considered in this thesis. We have used different FIR (Finite Impulse Response) and IIR (Infinite Impulse Response) filter architectures to investigate how their variation impacts recovery of the signal. The derivation of ideal solution for $N \times N$ network has been done with a certain constraint when direct mixing and demixing filters have unity gain. The adaptation rules derived for IIR (Infinite Impulse Response) architecture and ideal solution for $N \times N$ feedback network architecture are new in this thesis which were not done before.

1.4 Outline of this thesis

In chapter one we introduce different methods of independent component analysis which include second-order methods and higher-order methods.

In chapter two we introduce some basic definitions of some terminologies needed to understand the derivations of the ideal solution network and adaptation rules for $N \times N$ feedback network architectures.

Objective functions for ICA, analysis of estimators and choice of contrast functions and definitions of gradients and adaptation rules are included in chapter three.

The derivation of ideal solution for $N \times N$ feedback network architectures will be shown in chapter four.

In chapter five we derive the adaptation rules for different kinds of demixing filter architectures.

Simulation results and discussion are introduced in chapter six.

In chapter seven the future works and conclusion are described.

1.5 Conclusions

In this chapter different methods of classical linear transformations i.e., second-order and higher-order methods of independent component analysis have been introduced. There are different kinds of second-order methods i.e., principal component analysis, factor analysis, projection pursuit and redundancy reduction. One of the higher-order methods is blind deconvolution. We have a brief overview of all of these methods in this chapter.

Chapter 2

Independent Component Analysis

To begin with, we shall recall some basic definitions needed [47].

2.1 Some basic definitions

2.1.1 Distribution of a random variable

Let us denote x as a random variable. Then the *cumulative distribution function* of x at point $x = x_0$ and of the event $x \leq x_0$ can be defined as :

$$F_x(x_0) = P(x \leq x_0) \tag{2.1}$$

where, $x \leq x_0$ means x can take any value less or equal to x_0. The cdf is a nondecreasing continuous function that usually increases monotonically and it is also negetive. The value of cdf ranges from 0 to 1 and it is obtained when x_0 is changed from -∞ to ∞. Thus we can write, $0 \leq F_x(x) \leq 1$, i.e., $F_x(-\infty) = 0$, and $F_x(+\infty) = 1$.

By taking the derivative of the cumulative distribution function of x we can find its *probability density function* which is as follows:

$$p_x(x_0) = \frac{dF_x(x)}{dx}|_{x=x_0} \tag{2.2}$$

It is necessary to compute the *probability density function* because usually the density function rather than the cdf of a random variable is used to define its probability distribution. The usual practice is to compute the cdf is by integrating the known pdf :

$$F_x(x_0) = \int_{-\infty}^{x_0} p_x(\xi)\, d\xi \tag{2.3}$$

Very often the subscript is omitted to denote $F_x(x)$ and $p_x(x)$ by $F(x)$ and by $p(x)$ respectively for the simplicity of the expression.

2.1.2 Distribution of a random vector

Let us denote x as a random vector of dimension n and it has the components $x_1, x_2, ..., x_n$ which are continuous random variables.

$$\mathbf{x} = (x_1, x_2, ..., x_n)^T \tag{2.4}$$

where T means the transpose because all vectors used in this thesis are column vectors. We can define the cumulative distribution function of x as follows:

$$F_{\mathbf{x}}(\mathbf{x}_0) = P(\mathbf{x} \leq \mathbf{x}_0) \tag{2.5}$$

where, P denotes the probability of the event that $\mathbf{x} \leq \mathbf{x}_0$ which means components of vector x are less than or equal to the respective components of the vector $\mathbf{x}_0$. The cdf defined in equation (2.3) is multivariate and it is an increasing function. The value of each function of each component ranges from 0 to 1, i.e., $0 \leq F_x(x) \leq 1$. $F_{\mathbf{x}}(\mathbf{x}) = 1$ when all the components of x approach infinity and $F_{\mathbf{x}}(\mathbf{x}) = 0$ when any component $x_i \rightarrow -\infty$. Now, if $p_{\mathbf{x}}(\mathbf{x})$ of x denotes the multivariate probability density function of x and $F_{\mathbf{x}}(\mathbf{x})$ denotes the cumulative distribution function of x then $p_{\mathbf{x}}(\mathbf{x})$ of x at the point $\mathbf{x}_0$ can be derived as the derivative of $F_{\mathbf{x}}(\mathbf{x})$ with respect to all components of the random vector x at the point $\mathbf{x}_0$:

$$p_{\mathbf{x}}(\mathbf{x}_0) = \frac{\partial}{\partial x_1}\frac{\partial}{\partial x_2}...\frac{\partial}{\partial x_n}F_{\mathbf{x}}(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_0} \tag{2.6}$$

The cumulative distribution function can be defined as :

$$F_{\mathbf{x}}(\mathbf{x}_0) = \int_{-\infty}^{\mathbf{x}_0} p_{\mathbf{x}}(\mathbf{x})\,d\mathbf{x} = \int_{-\infty}^{x_{0,1}}\int_{-\infty}^{x_{0,2}}...\int_{-\infty}^{x_{0,n}} p_{\mathbf{x}}(\mathbf{x})\,dx_n...dx_2\,dx_1 \tag{2.7}$$

where, $x_{0,i}$ is the ith component of the vector $\mathbf{x}_0$. For the range of $\mathbf{x}$ from $-\infty$ to ∞

$$\int_{-\infty}^{+\infty} p_{\mathbf{x}}(\mathbf{x})\,d\mathbf{x} = 1 \tag{2.8}$$

2.1.3 Joint and marginal distributions

Let us assume another random vaector $\mathbf{y}$ of dimension m and form a supervector $\mathbf{z}^T = (\mathbf{x}^T, \mathbf{y}^T)$ of dimension $m+n$ by concatenating two random vectors $\mathbf{x}$ and $\mathbf{y}$. Then the *joint distribution function* of $\mathbf{x}$ and $\mathbf{y}$ is infact the cdf of $\mathbf{x}$ and $\mathbf{y}$ and is given by the following formulla:

$$F_{\mathbf{x},\mathbf{y}}(\mathbf{x}_0, \mathbf{y}_0) = P(\mathbf{x} \leq \mathbf{x}_0, \mathbf{y} \leq \mathbf{y}_0) \tag{2.9}$$

where the definition used in equation (2.9) is the joint probability of the event $\mathbf{x} \leq \mathbf{x}_0$ and $\mathbf{y} \leq \mathbf{y}_0$. $\mathbf{x}_0$ and $\mathbf{y}_0$ are two constant vectors of the same dimension as of $\mathbf{x}$ and $\mathbf{y}$, respectively.

Let us denote by $p_{\mathbf{x},\mathbf{y}}(\mathbf{x},\mathbf{y})$ as the *joint density function* of $\mathbf{x}$ and $\mathbf{y}$ which can be computed by differentiating the known joint distribution function $F_{\mathbf{x},\mathbf{y}}(\mathbf{x},\mathbf{y})$ with respect to components of the random vectors $\mathbf{x}$ and $\mathbf{y}$. Therefore, from the inverse relationship we can write:

$$F_{\mathbf{x},\mathbf{y}}(\mathbf{x}_0, \mathbf{y}_0) = \int_{-\infty}^{\mathbf{x}_0} \int_{-\infty}^{\mathbf{y}_0} p_{\mathbf{x},\mathbf{y}}(\xi, \eta) d\eta d\xi \tag{2.10}$$

when both $\mathbf{x}_0 \to \infty$ and $\mathbf{y}_0 \to \infty$, the value of $F_{\mathbf{x},\mathbf{y}}(\mathbf{x}_0, \mathbf{y}_0)$ approaches unity.

The *marginal densitiy* $p_{\mathbf{x}}(\mathbf{x})$ of $\mathbf{x}$ is computed by integrating $p_{\mathbf{x},\mathbf{y}}(\mathbf{x}, \mathbf{y})$ over $\mathbf{y}$ and the *marginal densitiy* of $p_{\mathbf{y}}(\mathbf{y})$ of $\mathbf{y}$ is obtained by integrating $p_{\mathbf{x},\mathbf{y}}(\mathbf{x}, \mathbf{y})$ over $\mathbf{x}$:

$$p_{\mathbf{x}}(\mathbf{x}) = \int_{-\infty}^{\infty} p_{\mathbf{x},\mathbf{y}}(\mathbf{x}, \eta) d\eta \tag{2.11}$$

$$p_{\mathbf{y}}(\mathbf{y}) = \int_{-\infty}^{\infty} p_{\mathbf{x},\mathbf{y}}(\xi, \mathbf{y}) d\xi \tag{2.12}$$

2.1.4 Mean vector and correlation matrix

If $\mathbf{x}$ is a random vector, then the first moment of it, denoted by $\mathbf{m}_{\mathbf{x}}$, is called the mean vector of $\mathbf{x}$. This mean vector is expressed as the expectations of $\mathbf{x}$:

$$\mathbf{m}_{\mathbf{x}} = E\{\mathbf{x}\} = \int_{-\infty}^{\infty} \mathbf{x} p_{\mathbf{x}}(\mathbf{x}) d\mathbf{x} \tag{2.13}$$

where $\mathbf{m}_{\mathbf{x}}$ has the dimension n and every component of it, m_{x_i} can be written as:

$$m_{x_i} = E\{x_i\} = \int_{-\infty}^{\infty} x_i p_{\mathbf{x}}(\mathbf{x}) d\mathbf{x} = \int_{-\infty}^{\infty} x_i p_{x_i}(x_i) dx_i \tag{2.14}$$

where x_i is the ith component of $\mathbf{x}$ and $p_{x_i}(x_i)$ is the marginal density of the x_i.

The second moment of $\mathbf{x}$ is called the correlation, r_{ij}, which is the correlation between the ith and jth components of $\mathbf{x}$ and defined as follows:

$$r_{ij} = E\{x_i x_j\} = \int_{-\infty}^{\infty} x_i p_{\mathbf{x}}(\mathbf{x}) d\mathbf{x} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x_i x_j p_{x_i,x_j}(x_i, x_j) dx_j dx_i \tag{2.15}$$

The correlation matrix of the vector $\mathbf{x}$ is defined as:

$$\mathbf{R}_{\mathbf{x}} = E\{\mathbf{x}\mathbf{x}^T\} \tag{2.16}$$

The matrix $\mathbf{R}_{\mathbf{x}}$ has the dimension $n \times n$ and each element of it in row i and column j is r_{ij}. It should be noted that r_{ij} can take both positive or negative value.

2.1.5 Covariance and joint moments

In this section we will derive the formullas for the correlation and convariance matrices for the same and two different random vectors as well as for a single random variable. Correlations and covariances use second-order statistics to measure the dependence between the random variables. If $\mathbf{x}$ is a random vector then the covariance matrix of $\mathbf{x}$ is defined as:

$$\mathbf{C}_{\mathbf{x}} = E\{(\mathbf{x} - \mathbf{m}_{\mathbf{x}})(\mathbf{x} - \mathbf{m}_{\mathbf{x}})^T\} \tag{2.17}$$

where $\mathbf{C}_{\mathbf{x}}$ is the notation used for the covariance matrix and it has the dimension $n \times n$. The covariance matrix $\mathbf{C}_{\mathbf{x}}$ corresponds to the quantity correlation matrix $\mathbf{R}_{\mathbf{x}}$ of $\mathbf{x}$. The i, jth element of the covariance matrix can be written as:

$$c_{ij} = E\{(x_i - m_i)(x_j - m_j)\} \tag{2.18}$$

where c_{ij} is an element in the ith row and the jth column of $\mathbf{C}_{\mathbf{x}}$. The covarinace is the central moments corresponding to the correlations r_{ij} defined in equation (2.15). Both the covariance matrix $\mathbf{C}_{\mathbf{x}}$ and the correlation matrix $\mathbf{R}_{\mathbf{x}}$ have the same properties.

From the properties of the expectation operator we can write

$$\mathbf{R}_{\mathbf{x}} = \mathbf{C}_{\mathbf{x}} + \mathbf{m}_{\mathbf{x}}\mathbf{m}_{\mathbf{x}}^T \tag{2.19}$$

From equation (2.19) we can see that the correlation matrix becomes the covariance matrix if the mean vector $\mathbf{m_x} = 0$. Usually, in independent component analysis for the preprocessing of the observation data the estimated mean vector is subtracted from the data vectors to make them zero-mean.

If we have a single random variable x instead of a random vector $\mathbf{x}$, then the mean value of the variable will be, $m_x = E\{x\}$. The correlation matrix to the second moment for this variable will become $E\{x^2\}$ and the covariance matrix to the variance of x will be reduced to:

$$\sigma^2{}_x = E\{(x - m_x)^2\} \tag{2.20}$$

Then equation (2.19) will be simplified to $E\{x^2\} = \sigma^2{}_x + m_x{}^2$.

For two different random vectors $\mathbf{x}$ and $\mathbf{y}$ the expectation for the functions $\mathbf{g}(\mathbf{x}, \mathbf{y})$ can expressed in terms of their joint density:

$$E\{\mathbf{g}(\mathbf{x}, \mathbf{y})\} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mathbf{g}(\mathbf{x}, \mathbf{y}) p_{\mathbf{x},\mathbf{y}}(\mathbf{x}, \mathbf{y}) d\mathbf{y}\, d\mathbf{x} \tag{2.21}$$

Again, the cross-correlation matrix of random vectors $\mathbf{x}$ and $\mathbf{y}$ is defined by the following formulla:

$$\mathbf{R_{xy}} = E\{\mathbf{x}\mathbf{y}^T\} \tag{2.22}$$

the cross-covariance matrix becomes

$$\mathbf{C_{xy}} = E\{(\mathbf{x} - \mathbf{m_x})(\mathbf{y} - \mathbf{m_y})^T\} \tag{2.23}$$

If the vectors $\mathbf{x}$ and $\mathbf{y}$ are of the same dimensions then the cross-correlation and covariance matrices will be square matrices but if not these two matrices will not be squre. Being square is not necessary for these two matrices. Another important property of these matrices are generally they are not symmetric. We can write from their definitions

$$\mathbf{R}_{\mathbf{xy}} = \mathbf{R}^T{}_{\mathbf{yx}}, \quad \mathbf{C}_{\mathbf{xy}} = \mathbf{C}^T{}_{\mathbf{yx}} \tag{2.24}$$

In the case of zero mean vectors of $\mathbf{x}$ and $\mathbf{y}$ the cross-corrleation and cross-covariance matrices will be the same. Some times we need to compute the covariance matrix of the sum of two random vectors $\mathbf{x}$ and $\mathbf{y}$ that have the same dimension. For this covariance matrix we can write that

$$\mathbf{C}_{\mathbf{x}+\mathbf{x}} = \mathbf{C}_{\mathbf{x}} + \mathbf{C}_{\mathbf{xy}} + \mathbf{C}_{\mathbf{yx}} + \mathbf{C}_{\mathbf{y}} \tag{2.25}$$

2.1.6 Estimation of expectations

Formally, the density function is used to define the expectation of a function of a random variable. But it is not possible to know the exact probability density function of a vector or a scalar valued random variable. So, we can use expectations instead of probability function which can be computed directly from the observation data. Because that is the only available information we have.

Let us denote by $\mathbf{g}(\mathbf{x})$ a quantity of the random vector $\mathbf{x}$ which may be either a scalar, vector, or a matrix. The notation used to denote the expection of $\mathbf{g}(\mathbf{x})$ is $E\{\mathbf{g}(\mathbf{x})\}$. The expection of $\mathbf{g}(\mathbf{x})$ is defined as follows:

$$E\{\mathbf{g}(\mathbf{x})\} = \int_{-\infty}^{\infty} \mathbf{g}(\mathbf{x}) p_{\mathbf{x}}(\mathbf{x}) d\mathbf{x} \tag{2.26}$$

The result of the above integration is a vector of matrix which has the same size as of $\mathbf{g(x)}$. This integration is performed on each component of the vector of the matrix individually. Let us consider the simplest case, $\mathbf{g(x)} = \mathbf{x}$. Since, we have a set of K samples $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, ..., \mathbf{x}_K$ available from $\mathbf{x}$, using the formula [48] we can estimate the expectations defined in equation (2.26) as follows:

$$E\{\mathbf{g(x)}\} \approx \frac{1}{K} \sum_{itj=1}^{K} \mathbf{g}(\mathbf{x}_j) \tag{2.27}$$

From equation (2.27) the sample mean $\hat{\mathbf{m}}_\mathbf{x}$ for the mean vector $\mathbf{m}_\mathbf{x}$ of $\mathbf{x}$ can be written as follows:

$$\hat{\mathbf{m}}_\mathbf{x} = \frac{1}{K} \sum\nolimits_{itj=1}^{K} \mathbf{x}_j \tag{2.28}$$

where $\hat{\mathbf{m}}_\mathbf{x}$ is the standard notation for an estimator of a quantity.

Similarly, if we have K sample pairs $(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), ..., (\mathbf{x}_K, \mathbf{y}_K)$ available, the expectation equation (2.21) can be estimated instead of the joint density $p_{\mathbf{x},\mathbf{y}}(\mathbf{x}, \mathbf{y})$ of the random vectors $\mathbf{x}$ and $\mathbf{y}$ and is given by the following equation:

$$E\{\mathbf{g(x,x)}\} \approx \frac{1}{K} \sum\nolimits_{itj=1}^{K} \mathbf{g}(\mathbf{x}_j, \mathbf{x}_j) \tag{2.29}$$

The estimation formula for the cross-correlation matrix can be written as:

$$\hat{\mathbf{R}}_{\mathbf{xy}} = \frac{1}{K} \sum\nolimits_{itj=1}^{K} \mathbf{x}_j \mathbf{y}_j{}^T \tag{2.30}$$

2.1.7 Uncorreletadness and whiteness

If the cross-covariance matrix, $\mathbf{C_{xy}}$ of two random vecotrs $\mathbf{x}$ and $\mathbf{y}$ is a zero matrix then these vectors are said to be uncorrelated. The cross-covariance matrix, $\mathbf{C_{xy}}$ of these vectors are given by the following equation:

$$\mathbf{C_{xy}} = E\{(\mathbf{x} - \mathbf{m_x})(\mathbf{y} - \mathbf{m_y})^T\} = 0 \tag{2.31}$$

which means,

$$\hat{\mathbf{R}}_{\mathbf{xy}} = E\{\mathbf{x}\mathbf{y}^T\} = E\{\mathbf{x}\}E\{\mathbf{y}^T\} = \mathbf{m_x}\mathbf{m_x}^T \tag{2.32}$$

For two scalar random variables x and y if their covariance C_{xy} is zero then they are also uncorrelated:

$$c_{xy} = E\{(x - m_x)(y - m_y)^T\} = 0 \tag{2.33}$$

In another way,

$$r_{xy} = E\{xy\} = E\{x\}E\{y\} = m_x m_y \tag{2.34}$$

We can say that zero covariance means zero correlation if variables have zero-mean.

Again the different components of a random vector $\mathbf{x}$ are said to be uncorrelated if the following condition of uncorrelatedness is satisfied:

$$\mathbf{C_x} = E\{(\mathbf{x} - \mathbf{m_x})(\mathbf{x} - \mathbf{m_x})^T\} = \mathbf{D} \tag{2.35}$$

where $\mathbf{D}$ is a diagonal matrix of dimension $n \times n$.

$$\mathbf{D} = diag(c_{11}, c_{22}, ..., c_{nn}) = diag(\sigma^2{}_{x_1}, \sigma^2{}_{x_2}, ..., \sigma^2{}_{x_n}) \tag{2.36}$$

where $\sigma^2{}_{x_i} = E\{(x_i - m_{x_i})\} = c_{ii}$ are the variances of the components x_i of $\mathbf{x}$ and these are the n diagonal elements $\mathbf{D}$.

The zero-mean random vectors are said to be white if they have unit covariance as well as unit correlation matrix. Therefore, the conditions for the white random vectors are:

$$\mathbf{m_x} = 0, \mathbf{R_x} = \mathbf{C_x} = \mathbf{I} \tag{2.37}$$

where $\mathbf{I}$ is the identity matrix of dimension $n \times n$.

An $n \times n$ matrix $\mathbf{T}$ is said to be an orthogonal matrix if it rotates the coordinate axes of a random vector x in the n-dimensional space when the matrix $\mathbf{T}$ is applied to it while preserve the norms and distances. This kind of transformation is called orthogonal transformation and is given by:

$$\mathbf{y} = \mathbf{Tx}, \quad \text{where} \quad \mathbf{T}^T\mathbf{T} = \mathbf{T}\mathbf{T}^T = \mathbf{I} \tag{2.38}$$

If the random vector $\mathbf{x}$ is white then it will satisfy the following condition:

$$\mathbf{m_y} = E\{\mathbf{Tx}\} = \mathbf{T}E\{\mathbf{x}\} = \mathbf{Tm_x} = 0 \tag{2.39}$$

and

$$\mathbf{C_y} = \mathbf{R_y} = E\{\mathbf{Tx}(\mathbf{Tx})^T\} = \mathbf{T}E\{\mathbf{x}\mathbf{x}^T\}\mathbf{T}^T = \mathbf{T}\mathbf{R_x}\mathbf{T}^T = \mathbf{T}\mathbf{T}^T = \mathbf{I} \tag{2.40}$$

It is clear that there also exists infinitely many ways to decorrelate the original data, because whiteness is a special case of the uncorrelatedness property.

2.1.8 Statistical independence

The basic idea on which the foundation of independent component is based on is the statistical independence. Let us consider two random variables x and y. "If knowing the value of y does not provide any information on the value of x then we can say that the random variable x is independent of y."

If the joint density $p_{x,y}(x, y)$ of x and y can be expressed as the product of their marginal densities $p_x(x)$ and $p_y(y)$ then variables x and y are said to be independent of each other.

$$p_{x,y}(x, y) = p_x(x)p_y(y) \tag{2.41}$$

Statistical independence can also be defined in terms of the cumulative function. In that case, the probability density function in equation (2.41) needed to be replaced by their respective cumulative distribution functions. The basic property of two independent random variables is:

$$E\{g(x)h(y)\} = E\{g(x)\}E\{h(y)\} \tag{2.42}$$

The proof of the above property is as follows:

$$\begin{aligned} E\{g(x)h(y)\} &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x)h(y)p_{x,y}(x, y)\,dy\,dx \\ &= \int_{-\infty}^{\infty} g(x)p_x(x)\,dx \int_{-\infty}^{\infty} h(y)p_y(y)\,dy \\ &= E\{g(x)\}E\{h(y)\} \end{aligned} \tag{2.43}$$

From equation (2.42) we see that in order to compute $E\{g(x)h(y)\}$ both $g(x)$ and $h(y)$ must be integrable functions of x and y, respectively. If we consider only second-order

statistics, i.e., correlations and covariances and assume that the the random variables have Gaussian distributions then the definition of uncorrelatedness defined in equation (2.34) becomes the special case of the definition of the statistical independence as in equation (2.42). Statistical independence becomes uncorrelatedness when the random variables have the Gaussian distribution.

The definition of statistical independence in equation (2.41) can be generalized for any N number of random variables as well as for random vectors. An N number of random vectors $\mathbf{x}, \mathbf{y}, \mathbf{z}, ...$, are said to be statistically independent if and only if

$$p_{\mathbf{x},\mathbf{y},\mathbf{z},...}(\mathbf{x}, \mathbf{y}, \mathbf{z}, ...) = p_{\mathbf{x}}(\mathbf{x})p_{\mathbf{y}}(\mathbf{y})p_{\mathbf{z}}(\mathbf{z}) \tag{2.44}$$

The generalization of the basic property in equation (2.42) is:

$$E\{\mathbf{g}_{\mathbf{x}}(\mathbf{x})\mathbf{g}_{\mathbf{y}}(\mathbf{y})\mathbf{g}_{\mathbf{z}}(\mathbf{z})\}..... = E\{\mathbf{g}_{\mathbf{x}}(\mathbf{x})\}E\{\mathbf{g}_{\mathbf{y}}(\mathbf{y})\}E\{\mathbf{g}_{\mathbf{z}}(\mathbf{z})\}..... \tag{2.45}$$

"where $\mathbf{g}_{\mathbf{x}}(\mathbf{x})$, $\mathbf{g}_{\mathbf{y}}(\mathbf{y})$, and $\mathbf{g}_{\mathbf{z}}(\mathbf{z})$ are arbitrary functions of the random variables $\mathbf{x}$,$\mathbf{y}$, and $\mathbf{z}$ for which the expectations in equation (2.45) exist."

2.1.9 Ordinary Entropy

Let us assume that X is a discrete-valued random variable. Then the entropy H of X is defined as:

$$H(X) = -\sum_i P(X = a_i) \log P(X = a_i) \tag{2.46}$$

where $P(X = a_i)$ is the probability that X has the values a_i. Entropy has different kind of units which is based on the base of the logarithm. The typical unit is bit when the base of the logarithm is 2. Let us assume that $P(X = a_i) = p$, then the equation (2.46) can be rewritten as

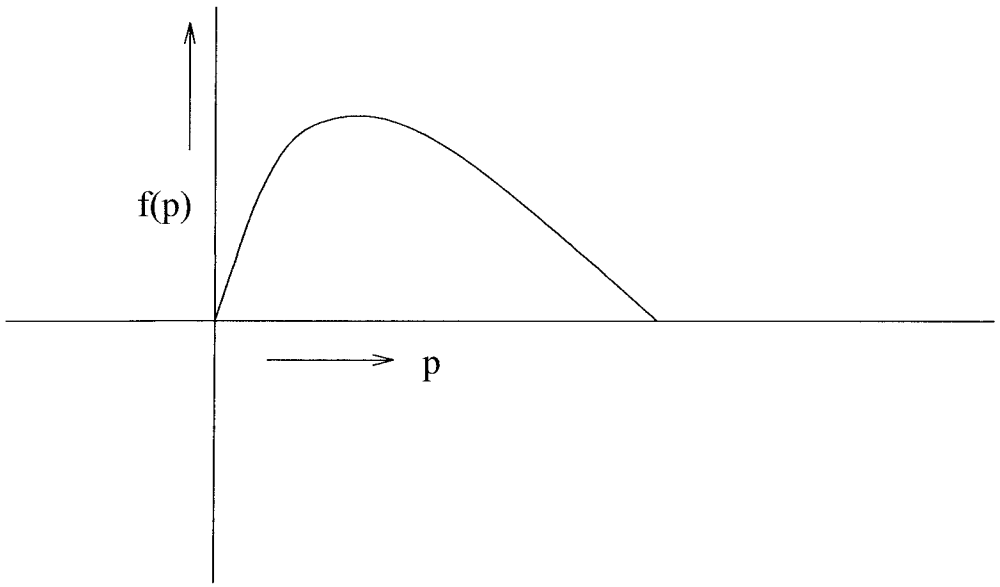

Figure 2.1: The function f in equation 2.47, plotted on the interval [0,1]

$$f(p) = -p \log p, for \;\; 0 \leq p \leq 1 \tag{2.47}$$

where the function $f(p)$ is always nonnegative. It is positive for the range of the values of p between 0 and 1 and zero for $p = 0$ and for $p = 1$ which is shown in Fig. (2.1). Equation (2.46) can be rewritten as

$$H(X) = -\sum_i f(P(X = a_i)) \tag{2.48}$$

From the Fig. (2.1) we can see that entropy is large if p has the values in between 0 and 1 and entropy is small if p is 0 or 1.

Basically entropy of a random variable provides the degree of information about that variable. The larger the entropy of a random variable the more random or unpredictable that variable is. On the contrary we can say smaller entropy means there is little randomness in the variable.

2.1.10 Differential Entropy

The difference between the ordinary entropy and the differential entropy is that the later one is computed when the random variable is continuous-valued instead of discrete-valued.

If x is a continuous-valued random variable with density $p_x(.)$ then its differential entropy H is defined as :

$$H(x) = -\int p_x(\xi) \log p_x(\xi) d\xi = \int f(p_x(\xi)) d\xi \tag{2.49}$$

Like the ordinary entropy, differential entropy is also a measure of randomness. But unlike the ordinary entropy differential entropy can take negetive values and it may have large absolute value.

2.1.11 Maximality property of the Gaussian distribution

Let us consider the set of zero-mean random variables that have unit variance and can take all the real values. The distribution of this kind of variables has the following form:

$$p_0(\xi) = Aexp(a_1\xi^2 + a_2\xi) \tag{2.50}$$

Both the maximum entropy distribution of this kind of variables and the probability densities which have the form as in equation (2.50) have Gaussian distribution.

Of all kind of distributions Gaussian distribution shows the most randomness and entropy is small when the variable is clustered. Distributions of this kind of variables are concentrated on certain values and it has very spiky shaped pdf. Thus we can draw the conclusion that "*a Gaussian variable has the largest entropy among all random variables of unit variance.*"

2.1.12 Entropy of transformation

Let us consider a random vector $\mathbf{x}$ that has the following invertible transformation:

$$\mathbf{y} = \mathbf{f}(\mathbf{x}) \tag{2.51}$$

Our objective here is to show how the entropy of $\mathbf{x}$ is related to that of $\mathbf{y}$, in another way we shall try to find the function $\mathbf{f}$.

The Jacobian matrix of the function $\mathbf{f}$ is the matrix of the partial derivative of $\mathbf{f}$ at point ξ and is denoted by $J\mathbf{f}(\xi)$. If the density of $\mathbf{y}$ is p_y and that of $\mathbf{x}$ is p_x then the relation between p_y and p_x can be written by the following formulla:

$$p_y(\eta) = p_x(\mathbf{f}^{-1}(\eta))|detJ\mathbf{f}(\mathbf{f}^{-1}(\eta))|^{-1} \tag{2.52}$$

The entropy of $\mathbf{y}$ can be expressed as:

$$H(\mathbf{y}) = -E\{\log p_y(\mathbf{y})\} \tag{2.53}$$

Again,

$$\begin{aligned} E\{\log p_y(\mathbf{y})\} &= E\{\log[p_x(\mathbf{f}^{-1}(\mathbf{y}))|detJ\mathbf{f}(\mathbf{f}^{-1}(\mathbf{y}))|^{-1}]\} \\ &= E\{\log[p_x(\mathbf{x})|detJ\mathbf{f}(\mathbf{x})|^{-1}]\} \\ &= E\{\log p_x(\mathbf{x})\} - E\{\log|detJ\mathbf{f}(\mathbf{x})|\} \end{aligned} \tag{2.54}$$

Thus the relation between the entropies of trasformed vector $\mathbf{y}$ and the input vector $\mathbf{y}$ becomes:

$$H(\mathbf{y}) = H(\mathbf{x}) + E\{\log|detJ\mathbf{f}(\mathbf{x})|\} \tag{2.55}$$

We can see that after the transformation the entropy has been increased by the quantity $E\{\log|detJ\mathbf{f}(\mathbf{x})|\}$.

Let us consider the following transformation:

$$\mathbf{y} = \mathbf{M}\mathbf{x} \tag{2.56}$$

After transformation we obtain:

$$H(\mathbf{y}) = H(\mathbf{x}) + \log|det\mathbf{M}| \tag{2.57}$$

which means differential entropy is scale-variant. The equation in (2.57) shows that if we multiply a random variable x by a scalar constant, α then the differential entropy will be:

$$H(\alpha x) = H(x) + \log|\alpha| \tag{2.58}$$

2.1.13 Negentropy

Negentropy is the normalized version of the ordinary entropy which is denoted by J and can be defined as:

$$J(\mathbf{x}) = H(\mathbf{x}_{gauss}) - H(\mathbf{x}) \tag{2.59}$$

where $\mathbf{x}_{gauss}$ is a Gaussian random vector that has the same covariance and the correlation matrix $\sum$ as of $\mathbf{x}$. The entropy of $\mathbf{x}_{gauss}$ can be computed as

$$H(\mathbf{x}_{gauss}) = \frac{1}{2}\log|det\sum| + \frac{n}{2}[1 + \log 2\pi] \tag{2.60}$$

where $\mathbf{x}$ has the dimension n.

Negentropy is zero if and only if $\mathbf{x}$ has a Gaussian distribution and always non-negetive because of the maximality property of the Gaussian distribution. Another interesting property of negentropy is it is invariant in case of invertible linear transformation. The reason is if $\mathbf{y} = \mathbf{Mx}$ then we can write $E\{\mathbf{yy}^T\} = \mathbf{M}\sum\mathbf{M}^T$. Negentropy of the transformed vector $\mathbf{y}$ can be computed as follows using the formulla in equation (2.60)

$$\begin{aligned} J\mathbf{Mx} &= \frac{1}{2}\log|\det(\mathbf{M}\sum\mathbf{M}^T)| + \frac{n}{2}[1 + \log 2\pi] - (H(\mathbf{x}) + \log|\mathbf{M}|) \qquad (2.61)\\ &= \frac{1}{2}\log|\det\sum| + 2\frac{1}{2}\log|\det\mathbf{M}| + \frac{n}{2}[1 + \log 2\pi] - H(\mathbf{x}) - \log|\det\mathbf{M}| \\ &= \frac{1}{2}\log|det\sum| + \frac{n}{2}[1 + \log 2\pi] - H(\mathbf{x}) = H(\mathbf{x}_{gauss}) - H(\mathbf{x}) = J(\mathbf{x}) \end{aligned}$$

If a random variable is multiplied by a constant then its negentropy ramains unchanged, i.e., negentropy is scale-invariant.

2.1.14 Mutual information

Let us consider a set of n (scalar) random variables x_i where $i = 1,...,n$. Then the mutual information I between these variables can be defined in terms of their entropies as follows:

$$I(x_1, x_2, ..., x_n) = \sum_{i=1}^{n} H(x_i) - H(\mathbf{x}) \tag{2.62}$$

The n components x_i of $\mathbf{x}$ will not give any information on each other if they are independent of each other. Thus we can say that "Mutual information is a measure of the information that members of a set of random variables have on the other random variables in the set."

2.1.15 Mutual information as a measure of independence

PCA and all other related methods take account only covariance but the mutual information considers the whole dependence structure of the variables. Another important property of mutual information is that it is always non-negative. Also for statistically independent variables mutual information is always zero. For all of these reasons this criterion can be used to estimate the independent components of the ICA model. If $\mathbf{x}$ is the observation vector then the ICA of it can be defined as the following invertible transformation:

$$\mathbf{s} = \mathbf{A}\mathbf{x} \tag{2.63}$$

where $\mathbf{A}$ is the matrix that needs to be estimated such that mutual information of the transformed components s_i is minimized.

2.1.16 Mutual information and nongaussianity

Let us consider the invertible linear transformation $\mathbf{y} = \mathbf{A}\mathbf{x}$. Using the formulla in equations (2.55) and (2.62) we can write the following formulla for the mutual information between the components of the trasnformed vector $\mathbf{y}$:

$$I(y_1, y_2, ..., y_n) = \sum_{i=1}^{n} H(y_i) - H(\mathbf{x}) - \log|\det \mathbf{A}| \tag{2.64}$$

We need to satisfy the condition $E\{\mathbf{y}\mathbf{y}^T\} = \mathbf{A}E\{\mathbf{x}\mathbf{x}^T\}\mathbf{A}^T = \mathbf{I}$ to constrain y_i to be of unit variance and be uncorrelated . Now,

$$\det\mathbf{I} = 1 = \det(\mathbf{A}E\{\mathbf{x}\mathbf{x}^T\}\mathbf{A}^T) = (\det\mathbf{A})(\det E\{\mathbf{x}\mathbf{x}^T\})(\det\mathbf{A}^T) \tag{2.65}$$

To make the right hand side of equation (2.65) constant $\det\mathbf{A}$ must be constant because $\det E\{\mathbf{x}\mathbf{x}^T\}$ is the function of x only. Now, to satisfy the condition of y_i to be of unit variance we can write the relation between the mutual information among independent components by using the the definition of negentropy:

$$I(y_1, y_2, ..., y_n) = cons. - \sum_{i=1}^{n} J(y_i) \tag{2.66}$$

where the constant term is independent of $\mathbf{A}$.

From equation (2.66) we see that by minimizing the mutual information we can find an invertible linear transformation $\mathbf{A}$. Because the direction in which the mutual information is minimized is approximately equivalent to finding the directions in which the negentropy is maximized. Thus we can conclude that "ICA estimation by minimization of mutual information is equivalent to maximizing the sum of nongaussianities of the estimates of the independent components, when the estimates are constrained to be uncorrelated."

Therefore for the estimation of ICA model minimization of mutual information is more justified than using the concept of finding maximally nongaussian directions.

But, still these two criteria have some important differences:

1. By using negentropy and other measures of nongaussianity we can find the maxima of nongaussianity of a single projection $\mathbf{b}^T\mathbf{x}$. So in this method we can estimate he independent components by using the deflationary or one-by-one scheme which is not possible in mutual information.

2. We can reduce the optimization space by using mutual information mutual information approach instead of using nongaussianity. The reason is the estimation of the independent components are forced to be uncorrelated if we use nongaussianity. But this is not necessary if someone use mutual information approach.

2.2 Definition of linear independent component analysis

The definition of independent component analysis given here will be only for linear case where as the non linear ICA also exists. In [3, 4] we find at least three basic definitions for linear ICA. Since ICA is a new research topic, most of the research has concentrated on the simplest one of these definitions. Let us denote by $\mathbf{x} = (x_1, ..., x_m)^T$ as the observed m dimensional random vector.

Following is the most general definition since no assumptions on the data are made:

"**Definition 1**: (*General definition*) *ICA of the random vector* $\mathbf{x}$ *consists of finding a linear transform* $\mathbf{s} = \mathbf{W}\mathbf{x}$ *so that the components* s_i *are as independent as possible, in the sense of maximizing some function* $F(s_1, ..., s_m)$ *that measures independence.*"

The next two definitions are no longer general definition because some assumptions on the data have been made. The more estimation-theoretically oriented definition, where noise has been taken into account in the ICA model is as follows:

"**Definition 2**: (*Noisy ICA model*) *ICA of a random vector* $\mathbf{x}$ *consists of estimating the following generative model for the data*:

$$\mathbf{x} = \mathbf{A}\mathbf{s} + \mathbf{n} \tag{2.67}$$

where the latent variables (*components*) s_i *in the vector* $\mathbf{s} = (s_1, ..., s_n)^T$ *are assumed independent. The matrix* $\mathbf{A}$ *is a constant* $m \times n$ *mixing matrix, and* $\mathbf{n}$ *is a* $\mathbf{m}$ *dimensional random noise vector.*"

Thus the ICA problem has been reduced to ordinary estimation of a latent vari-

able model yet it is not very simple to estimate the model. Thus, most of the research on ICA has focused on the following definition where noise vector has been omitted:

"**Definition 3** (*Noise – free ICA model*) *ICA of a random vector* $\mathbf{x}$ *consists of estimating the following generative model for the data*:

$$\mathbf{x} = \mathbf{A}\mathbf{s} \tag{2.68}$$

where $\mathbf{A}$ *and* $\mathbf{s}$ *are as in Definition 2*."

This was the earliest explicit formulation (probably) of ICA by Jutten in his PhD [49] (see also [50]). The same model has been used by Jutten and Herault in their seminal thesis [3].

2.3 Identifiability of the ICA model

In [4] the conditions for the ICA model has been imposed only for the noise-less case. The basic assumption fot the ICA model is the statistical independence. But only this assumption does not assure the identifiability of the model. There are also some other restrictions that need to be imposed to identify the ICA model. These are as follows:

1. "All the independent components s_i, with the possible exception of one component, must be non-Gaussian.
2. The number of observed linear mixtures m must be at least as large as the number of independent components n, i.e., $m \geq n$.
3. The matrix $\mathbf{A}$ must be full column rank."

If $\mathbf{x}$ and $\mathbf{s}$ are not simple random variables but also interpreted as the stochastic process, then one must at least assume that this stochastic process is stationary. We also

need to put some other restriction of ergodicity [51]. If the process is i.i.d. over time all of these assumptions will be fulfilled. We can consider the stochastic process as a random variable after imposing all of these restrictions.

In independent component analysis, the column of $\mathbf{A}$ and the independent components can be estimated upto a multiplicative constant only. But, this indeterminacy is insignificant since one can cancell the multiplicative constant of an independent component in equation (2.68) by dividing the corresponding column of the mixing matrix $\mathbf{A}$ by the same constant. The independent components are made to be unique by defining the independent components s_i to have unit variances. However, these variances for different independent components can be upto different multiplicative constants for different independent components [4].

In PCA there is ordering of the principle components in which they are estimated. But in ICA there is no such order in which the ICAs are estimated. There are two ways by which we can introduce an order between the independent components. The first way to order the ICAs is to use the norm of the columns of the mixing matrix $\mathbf{A}$ in the descending order because the independent components s_i contribute to their variances x_i by these norm. The second way is to use the projection pursuit index or the contrast function which are two measures of non-Gaussianity.

Now we shall discuss the three restrictions we introduced to identify the ICA model.

To identify the ICA model [4] the first restriction of non-Gaussianity is necessary. Only by the decorrerlation we can find the independent components if the random variables are Gaussian. Because for Gaussian random variables uncorrelatedness means independence. Still one can identify the non-Gaussian independent components if the ICA model consists of more than one Gaussian independent components s_i.

The second restriction that has been imposed to identify the ICA model is not completely necessary. In the literature this case has been said to be under complete case, $m \geq n$. Although there is no rigorous proofs, the mixing matrix $\mathbf{A}$ is still identifiable [52] even when $m < n$ which is called over complete case [52–54, 56, 57]. In the over complete case, the mixing matrix $\mathbf{A}$ is non invertible and as a result it is not possible to

identify the realizations of the independent components. In our thesis, it is necessary to impose the second assumption because most of the existing theory for ICA is not valid over complete case.

It is also necessary to have some rank restriction on the mixing matrix that we can say about the mixing matrix.

If we assume that the noise is independent from the components s_i [58–60] then the noise variables can be treated as some independent components. In that case, the noisy ICA model can be considered as a special case of the noise-free ICA model with $m < n$ and the same three restrictions can be introduced to identify the noisy-ICA model partially. Both the mixing matrix $\mathbf{A}$ and the noise covariance matrix are also identifiable [60]. But it will not guarantee the complete separation of the independent components from noise because the realization of s_i can not be identified.

2.4 Conclusions

In this chapter we have discussed the definitions of some important terminologies in brief related to Independent Component Analysis (ICA). These terminologies have been used throughout the whole thesis and also to derive the adaptation rules and the ideal solutions for $N \times N$ feedback network architecture. We have also defined what is the linear independent component analysis in this chapter. The identifiability of the ICA model has been discussed here too.

Chapter 3

Objective (contrast) functions for ICA

3.1 Introduction

The function which is used to estimate the data model of independent component analysis is called the objective function. For some certain class of objective functions some authors use the term contrast functions [4]. Some times this function is called loss function or cost function. In our thesis, we shall use the terms contrast function or objective function because these are the most widely used conventional terms. The estimation of the independent components is performed by minimizing or maximizing the contrast function. There are different types of algorithms which are used to optimize these contrast functions. But there is a distinction between the contrast functions and the algorithms used to optmize these. In the following sections we will discuss the differences between the contrast function and the algorithms, use of one unit contrast functions, use of negentropy in contrast function, choice of contrast function based on the estimators and the stochastic gradient rule.

3.2 Difference between objective functions and algorithms

The performance of the ICA method depends on both the objective function and the optimization algorithm. So, the ICA method can be formulated by the following equation:

ICA method = Objective function + Optimization algorithm.

Sometimes, it is difficult to separate the objective function and the optimization algorithm. But, if objective functions are explicitly formulated, any of the classical methods such as Newton-like methods, etc can be used. The performance of the ICA method depends on two major properties:

- The statistical properties: By choosing the proper objective function we can optimize any of these propeties. The examples of such properties are, e.g., consistency, asymptotic variance and robustness.
- The algorithm properties: These properties depend on the choice of the suitable optimization algorithm. The convergence speed, memory requirements and numerical stability falls in this category of properties.

To optimize a single objective function one can use different optimization algorithms or a single optimization algorithm can be used to optimize different objective functions.

There are two types of contrast functions, one unit contrast function and the multi-unit contrast function. In the following section we will describe only one unit contrast function.

3.3 One unit contrast functions

The contrast function which is used to estimate only one independent component each time, is called one unit contrast function [1]. In this case we at first estimate only one vector $\mathbf{w}$, and the newly found $\mathbf{w}$ is combined linearly with $\mathbf{x}$, i.e., $\mathbf{w}^T\mathbf{x}$, to estimate the first independent component. All the independent components are estimated by using the same method.

The one-unit contrast functions have the following usefulness:

- Using one unit contrast function enables to estimate independent components one after another. So one does not need to know the number of independent components for the estmation of whole ICA model.

- By optimizing the contrast function of each neuron it is possible to construct a neural network which is a very simple solution to construct a neural network from the computational point of view. So, we can say that there is a direct connection of using the unit approach to neural networks.

- There is also a direct connection of using the one unit method to projection pursuit.

- Using one unit approach in projection pursuit reduces the dimension of the data if the input data has very large dimension. In projection pursuit one does not need to find the independent components in order because the most interesting component is found at first which is the least Gaussian one [1]. The independent components are found in the descending order of non-Gaussianity if the one unit contrast functions are optimized globally. In that case, only some of the independent components are needed to estimate. Finding all of the independent components are not necessary, which also reduces the computational complexity.

According to the definition of ICA, all independent components are mutually uncorrelated. After estimating the first independent component the rest of the components can be estimated by using the method of decorrelation. The second independent component is found by maximizing the one-unit contrast function under the constrain of decorrelation that the first component is found already. One can repeat the same procedure to find the rest of the components. Another principle is to use the symmetric (parallel) decorrelation [61–64].

3.4 Contrast functions through approximations of negentropy

The objective functions, used to estimate the ICA model are conventionally cumulant-based [4, 11, 65]. But a more accurate method of the approximations of the objective functions have been developed which is based on the maximum entropy principle [14]. It has been found this new approximations are more accurate than the conventional cumulant-based approximations [14]. The newly developed approximations have the following form:

$$J(y_i) \approx c[E\{G(y_i)\} - E\{G(v)\}]^2 \tag{3.1}$$

where, y_i is the random variable of zero mean and unit variance, v is a Gaussian variable of zero mean and unit variance, c is an irrelevant constant and G is any non-quadratic function. If the variables are symmetric the generalization of the cumulant-based approximation [4] can be obtained by assuming, $G(y_i) = {y_i}^4$.

The new objective function given by equation (3.1) can be used to estimate the ICA model. At first, by maximizing the function J_G given by the following equation we can find the first independent component or the projection pursuit direction defined by, $y_i = \mathbf{w}^T\mathbf{x}$:

$$J_G(\mathbf{w}) = [E\{G(\mathbf{w}^T\mathbf{x})\} - E\{G(v)\}]^2 \tag{3.2}$$

where, $\mathbf{w}$ is the weight vector of dimension m with the constrain that $E\{(\mathbf{w}^T\mathbf{x})^2\} = 1$. After that by using the deflation scheme [1] we can find the rest of the independent components one-by-one.

To estimate all of the components of the matrix $\mathbf{W}$ in equation (1.2) we can extend the contrast functions for one-unit approach given by equation (3.2) by using the principle of minimizing mutual information . According to equation (2.62)) by maximizing

the sum of the negentropies of the components the mutual information can be minimized under the constraint of decorrelation. Thus we have the following optimization problem from the maximization of the sum of n one-unit contrast functions with the condition of the constraint of decorrelation:

$$\text{maximize} \sum_{i=1}^{n} J_G(\mathbf{w}_i) \text{ wrt. } \mathbf{w}_i, i = 1, ..., n \tag{3.3}$$

$$\text{subject to } E\{(\mathbf{w}^T{}_k\mathbf{x})E\{(\mathbf{w}^T{}_j\mathbf{x}) = \delta_{jk} \tag{3.4}$$

where $\mathbf{w}_i, i = 1, ..., n$ is one of the rows of the matrix $\mathbf{W}$. Then the independent components are obtained from the transformation equation, $\mathbf{s} = \mathbf{W}\mathbf{x}$. In the next section some algorithms have been presented as the solution of the optimization problem and we have analyzed the properties of estimators and discussed the selection criteria of G.

3.5 Analysis of estimators and choice of contrast function

3.5.1 Behavior under the ICA data model

There are certain properties of the estimators derived in the preceeding section which will be analyzed in this section by assuming the mixing matrix $\mathbf{A}$ in equation (1.2) as a square matrix. Here, we shall consider the estimation of only a single independent component. Let us assume that $\hat{\mathbf{w}}$ is a vector which has been obtained by maximizing J_G in equation (3.2). Then we can say that $\hat{\mathbf{w}}$ is an estimator of a row of the matrix $\mathbf{A}^{-1}$

Consistency

By using the following theorem [64] we shall prove that in the ICA data model $\hat{\mathbf{w}}$ is a locally consistent estimator for a single independent component :

"**Theorem 1** *Assume that the input data follows the ICA data model and that is a sufficiently smooth even function. Then the set of local maxima of* $J_G(\mathbf{w})$ *under the constraint* $E\{(\mathbf{w}^T\mathbf{x})^2\} = 1$, *includes the i-th row of the inverse of the mixing matrix A such that the corresponding independent component* s_i fulfils

$$E\{s_i g(s_i) - \acute{g}(s_i)\}[E\{G(s_i)\} - E\{G(\nu)\}] > 0 \tag{3.5}$$

where $g(.)$ *is the derivative of* $G(.)$, *and* ν *is a standardized Gaussian variable.*"

According to Theorem 1, the condition happens to be true for most distributions of the s_i and selections of G. For a particular value of $G(u)$, e.g., $G(u) = u^4$, the condition is fulfilled for any distribution of kurtosis that has the non-zero value. But there will be many spurious optima which has been proved in [66].

Asypmtotic variance

Asymptotic variance is the second statistical property which can be used to choose the function G. The mean-square error of the two estimators can be obtained by comparing the traces of the asymptotic covariance matrices of those estimators. It is possible to evaluate the asymptotic variances which has been shown in [67] for some contrast functions which are close to each other. The theorem based on the asymptotic variance is as follows [64]:

"**Theorem 2** *The trace of the asymptotic (co) variance of* $\hat{\mathbf{w}}$ *is minimized when G is of the form*

$$G_{opt}(u) = k_1 \log f_i(u) + k_2 u^2 + k_3 \tag{3.6}$$

where $f_i(.)$ *is the density function of* s_i, *and* k_1, k_2, k_3 *are arbitrary constants.*"
According to this theorem a good choice of G can be $G_{opt}(u) = \log f_i(u)$

Robustness

When the single, highly erroneous observations can influence the result much then we can say that the estimator is not robust against outliers [68]. In that case one should choose an estimator which is robust enough against outliers. Usually those estimators are robust that have bounded influence functions. It is not possible to have the estimators which have a completely bounded influence function. But we can obtain a simple form of robustness which is called B-robustness [68]. The theorem related to the robustness of the estimators is as follows [64]:

"**Theorem 3** *Assume that the data* $\mathbf{x}$ *is whitened (sphered) in a robust manner. Then the influence function of the estimator* $\hat{\mathbf{w}}$ *is never bounded for all* $\mathbf{x}$. *However, if* $h(u) = ug(u)$ *is bounded, the influence function is bounded in sets of the form* $\{\mathbf{x} \mid \hat{\mathbf{w}}^T\mathbf{x}/||\mathbf{w}|| > \epsilon\}$ *for every* $\epsilon > 0$, *where* g *is the derivative of* G."

Usually if "*a function* $G(u)$ *that is bounded*", h is also bounded. Some times it may not be possible to obtain an estimators with bounded influence function. In that case, we should choose a function $G(u)$ that does not grow very fast with $\mid u \mid$.

3.5.2 Practical choice of contrast function

Performance in the exponential power family

The theoretical results derived in the preceding section can be used to determine the performance in the exponential power family of density functions shown below [64]:

$$f_\alpha(s) = k_1 \exp(k_2 \mid s \mid^\alpha + k_3 \tag{3.7}$$

where f_α is a probability density, k_1, k_2 are normalization constants and α is a positive parameter. Two constants k_1, k_2 ensure that f_α has the unit variance. The shapes of the family of the densities vary depending on different values of alpha. We will consider

here three ranges of alpha. First, if $0 < \alpha < 2$, the density has positive kurtosis and it gives sparse shape which implies this is a super-Gaussian density. Second, if $\alpha = 2$, the distribution is Gaussian, and lastly, if $\alpha > 2$, the density has negative kurtosis and it is sub-Gaussian. T

Recall from the definition of asymptotic variance from Theorem 2, to estimate an independent component the required optimal contrast function which has the density function of the form f_α is defined by the following equation:

$$G_{opt}(u) = |u|^\alpha \tag{3.8}$$

Equation 3.8 shows that the optimal contrast function grows faster than quadratically for sub-Gaussian densities and slower than quadratically for super-Gaussian densities. Again from Theorem 3 of robustness, the contrast function becomes very non-robust against outliers if $G(u)$ increases fast with $\mid u \mid$. Since, practially most of the independent components are super-Gaussian [69, 70], we can choose the following function as the contrast function G for estimating the super-Gaussian independent components:

$$G_{opt}(u) = |u|^\alpha, \text{where } \alpha < 2 \tag{3.9}$$

The contrast function shown in equation (3.9) are not differentiable for the value of alpha 0 when $\alpha \leq 1$. The problem with these contrast functions can be solved by using approximating differentiable functions that shows the qualitative behavior in the same way. When $\alpha = 1$, we can use the function $G_1(u) = \log\cosh a_1 u$ where $a_1 \geq 1$ is a constant. In that case, the derivative of G_1 will be the function $\tanh$ if $a_1 = 1$. The independent components are highly super-Gaussian when $\alpha < 1$, i.e., , we can approximate $G_{opt}(u)$ for large value of u by using the Gaussian function $G_2(u) = -\exp(-a_2 u^2/2)$, where a_2 is a constant. The derivative of this function is 0 for larger values and similar to a sigmoid for small values. Being within the framework of the

type of the estimators shown in equations (3.3) and (3.4) this contrast function exhibits the behaviour of a good estimator which fulfils the condition in Theorem 3 as well as is robust enough. Experimentally, it has been found that when the values of the constant a_1 lie between 1 and 2, i.e., $1 \leq a_1 \leq 2$ and also $a_2 = 1$, the contrast function shows good approximations.

Choosing the Contrast Function in Practice

There are two important criteria we need to consider before using any contrast function G.

First, the computational complexity and second, the order in which the independent components are estimated.

The contrast function should be computationally less complex, preferably simple and fast. Both polynomial and non-polynomial functions have certain advantages and disadvantages over each other. Usually most of the polynomial functions are computationally faster than the non-polynomial functions, e.g., the hyperbolic tangent. Non-polynomial functions are slower to compute but these have certain advantages which are not possible to obtain using the polynomial functions. To avoid the computational complexity one can use piece wise linear approximations in place of non-polynomial functions.

If one uses the approach of one unit contrast function then the order in which the independent components are computed should be considered. In this approach there is no particular method of determining this order because this is highly application-dependent. There is a relation between the contrast function to the distribution of certain independent components. Which means it is possible to influence the order by a suitable choice of the contrast functions. Another reason is that the sizes of the basins of the attractions of the maxima of the contrast functions are different. Usually, the ordinary optimization methods try to find that maxima first which has large basins of attraction.

Recall from the discussion given above about two points, we may have the following

contrast functions and their derivatives as our choices of use:

$$G_1(u) = \frac{1}{4}u^4, \quad g_1(u) = u^3 \tag{3.10}$$

$$G_2(u) = -\frac{1}{a_1}\exp(-a_2u^2/2), \quad g_2(u) = u\exp(-a_2u^2/2) \tag{3.11}$$

$$G_3(u) = \frac{1}{a_1}\text{logcosh}\, a_1u, \quad g_3(u) = \tanh(a_1u) \tag{3.12}$$

where $1 \leq a_1 \leq 2$, $a_2 \approx 1$. Using different contrast functions in equations (3.10), (3.11) and (3.12) have following benefits:

- one can use G_1 or kurtosis is justified for the estimation if the independent components are sub-Gaussian and when there are no outliers.
- one should use the piecewise linear approximations of G_2 and G_3 to reduce the computational complexity.
- for the highly super-Gaussian independent components robustness is a very important factor. In that case, G_2 is a very good choice.
- as a general-purpose contrast function G_3 is a good choice.

In the conclusion we can say that choice of the contrast function is necessary only for the optimization of the performance of the method.

3.6 Gradient

3.6.1 Vector gradient

Let us denote by g a scalar valued function of m variables and let us assume that the function g is differentiable

$$g = g(w_1, ..., w_m) = g(\mathbf{w}) \tag{3.13}$$

where $\mathbf{w} = (w_1, ..., w_m)^T$, is a column vector. The vector gradient of g with respect to $\mathbf{w}$ is defined as:

$$\frac{\partial g}{\partial \mathbf{w}} = \begin{pmatrix} \frac{\partial g}{\partial w_1} \\ \frac{\partial g}{\partial w_2} \\ \vdots \\ \frac{\partial g}{\partial w_m} \end{pmatrix} \tag{3.14}$$

where, $\frac{\partial g}{\partial \mathbf{w}}$ is the gradient that is a m-dimensional column vector. Two other notations are ∇g or $\nabla_{\mathbf{w}} g$.

The second-order gradient of a function g with respect to $\mathbf{w}$ can be defined as:

$$\frac{\partial^2 g}{\partial \mathbf{w}^2} = \begin{pmatrix} \frac{\partial^2 g}{\partial {w_1}^2} & ..... & ..... & \frac{\partial^2 g}{\partial w_1 w_m} \\ ..... & ..... & ..... & ..... \\ \frac{\partial^2 g}{\partial w_m w_1} & ..... & ..... & \frac{\partial^2 g}{\partial {w^2}_m} \end{pmatrix} \tag{3.15}$$

The matrix defined above is called *Hessian matrix* of the function $g(\mathbf{w})$ and it has the dimension $m \times m$. As we can see, each element of the *Hessian matrix* is a second order partial derivative. One important property of the *Hessian matrix* is that it is always symmetric.

For an n-element *vector-valued* functions we can write:

$$\mathbf{g}(\mathbf{w}) = \begin{pmatrix} g_1(\mathbf{w}) \\ \vdots \\ g_n(\mathbf{w}) \end{pmatrix} \tag{3.16}$$

the elements $g_i(\mathbf{w})$ of the matrix defined above are functions of $\mathbf{w}$.

The *Jacobian matrix* of $\mathbf{g}$ with respect to $\mathbf{w}$ is the matrix of the partial derivatives of the elements of the function $\mathbf{g}$ with respect to the elements of $\mathbf{w}$ and written as follows:

$$\frac{\partial \mathbf{g}}{\partial \mathbf{w}} = \begin{pmatrix} \frac{\partial g_1}{\partial w_1} & ..... & ..... & \frac{\partial g_n}{\partial w_1} \\ ..... & ..... & ..... & ..... \\ \frac{\partial g_1}{\partial w_m} & ..... & ..... & \frac{\partial g_n}{\partial w_m} \end{pmatrix} \tag{3.17}$$

Sometimes $J\mathbf{g}$ is used as the notation for the Jacobian matrix.

3.6.2 Matrix gradient

Now, let us define g in a new way and that is g is a scalar valued functions of the matrix $\mathbf{W}$ where $\mathbf{W} = (w_{ij})$ the matrix of dimesnion $m \times n$. Therefore, we can write the following equation for newly defined g:

$$g = g(\mathbf{W}) = g(w_{11}, ..., w_{ij}, ..., w_{mn}) \tag{3.18}$$

The matrix gradient can be defined in the same way as the vector gradient has been defined. The ijth element of the matrix gradient is the partial derivative of g with respect to w_{ij}. The matrix gradient has the dimension $m \times n$ which is same as of matrix $\mathbf{W}$ and it can be written as follows:

$$\frac{\partial g}{\partial \mathbf{W}} = \begin{pmatrix} \frac{\partial g}{\partial w_{11}} & ..... & ..... & \frac{\partial g}{\partial w_{1n}} \\ ..... & ..... & ..... & ..... \\ \frac{\partial g}{\partial w_{m1}} & ..... & ..... & \frac{\partial g}{\partial w_{mn}} \end{pmatrix}$$

where, $\frac{\partial g}{\partial \mathbf{W}}$ is used as the notation for the matrix gradient.

3.7 Learning rules

3.7.1 Gradient descent

Most of the ICA methods aim to estimate the independent components by minimizing a cost function $\jmath(\mathbf{W})$ with respect to a parameter matrix $\mathbf{W}$. We have two kinds of solution for solving this problem, first, the solutions without any kind of constraints and second, the solutions under some constraints. In the later case, the number of possible solutions is restricted by the constraints. The most common type of constraints require the solution vector to have a bounded norm or the solution matrix to have orthonormal columns.

The steepest descent or gradient descent is the most classic approach of minimizing a multivariate function for the unconstrained case. In this section we shall discuss the steepest descent or gradient descent when the solution is a vector $\mathbf{w}$.

In steepest descent, our objective is to minimize a function $\jmath(\mathbf{w})$. To do so, we start from an initial point $\mathbf{w}(0)$ and at this point we estimate the gradient of $\jmath(\mathbf{w})$. After, by choosing a suitable distance we move in the steepest descent or the negative gradient direction. When we reach the new point, the whole procedure is followed again in an iterative fashion and we find the next point. Thus the adaptive or update rule for the consecutive values of t, say, $t = 1, 2, ...$, becomes:

$$\mathbf{w}(t) = \mathbf{w}(t-1) - \alpha(t)\frac{\partial \jmath(\mathbf{w})}{\partial \mathbf{w}}|_{\mathbf{w}=\mathbf{w}(t-1)} \tag{3.19}$$

where the value of $\mathbf{w}(t)$ is computed at the point $\mathbf{w}(t-1)$. The parameter $\alpha(t)$ is called the *learning rate* or *step size* which defines the length of the step in the negative gradient direction. The adaptation rule in equation (3.19) is repeated again and again; and when the Euclidean distance between two consequtive values of $\mathbf{w}(t)$, i.e., $||\mathbf{w}(t) - \mathbf{w}(t-1)||$ reaches a tolerance level, we can say the algorithm has converged.

Let us denote the difference between two consequent solutions by

$$\mathbf{w}(t) - \mathbf{w}(t-1) = \Delta\mathbf{w} \tag{3.20}$$

Using the notation written above we can write the update rules as follows:

$$\Delta\mathbf{w} = -\alpha\frac{\partial\jmath(\mathbf{w})}{\partial\mathbf{w}}$$

or,

$$\Delta\mathbf{w} \propto -\frac{\partial\jmath(\mathbf{w})}{\partial\mathbf{w}}$$

where $\propto$ is the proportionality constant and the gradient vector on the right-hand side and the vector on the left-hand side $\Delta\mathbf{w}$ of equation written above have the same directions. In terms of the programming languages the update rules can be written as:

$$\mathbf{w} \leftarrow \mathbf{w} - \alpha\frac{\partial\jmath(\mathbf{w})}{\partial\mathbf{w}}$$

where the symbol $\leftarrow$ means substitution, i.e., the value of the right-hand side is computed and substituted in $\mathbf{w}$.

In the gradient descent algorithm we always move in the steepest downward direction which has a very big disadvantage. This rule may reach to the global minimum if the function $\frac{\partial\jmath(\mathbf{w})}{\partial\mathbf{w}}$ is very smooth and simple otherwise it may converge to the closest local minimum. The cost functions with non-quadrectic expressions may have many local maxima and minima. For this kind of cost functions there is no escape from the local minima once the algorithm converges there. One possible solutions is to choose a proper initial values.

Another important factor we need to consider is the speed of convergence of the algorithm. Since the gradient becomes zero at the minimum point, the algorithm becomes very slow here. We can call the point, where the algorithm converges, as the

local or global minimum point and let us denote the values of $\mathbf{w}$ at that point by $\mathbf{w}^*$. Substituting this value in equation (3.19) we get have

$$\mathbf{w}(t) - \mathbf{w}^* = \mathbf{w}(t-1) - \mathbf{w}^* - \alpha(t)\frac{\partial \jmath(\mathbf{w})}{\partial \mathbf{w}}|_{\mathbf{w}=\mathbf{w}(t-1)} \tag{3.21}$$

By using the Taylor series expansion we can expand the gradient vector $\frac{\partial \jmath(\mathbf{w})}{\partial \mathbf{w}}$ around the point $\mathbf{w}^*$. Taking only the zeroth- and first-order terms the ith element can be written as:

$$\frac{\partial \jmath(\mathbf{w})}{\partial w_i}|_{\mathbf{w}=\mathbf{w}(t-1)} = \frac{\partial \jmath(\mathbf{w})}{\partial w_i}|_{\mathbf{w}=\mathbf{w}^*} + \sum_{j=1}^{m} \frac{\partial^2 \jmath(\mathbf{w})}{\partial w_i w_j}|_{\mathbf{w}=\mathbf{w}^*}[w_j(t-1) - w_j{}^*] + \ldots \tag{3.22}$$

In terms of the Hessian matrix:

$$\frac{\partial \jmath(\mathbf{w})}{\partial \mathbf{w}}|_{\mathbf{w}=\mathbf{w}(t-1)} = \mathbf{H}(\mathbf{w}^*)[\mathbf{w}(t-1) - \mathbf{w}^*]$$

where $\mathbf{H}(\mathbf{w}^*)$ is the Hessian matrix computed at the point $\mathbf{w} = \mathbf{w}^*$. Using this value in equation (3.21) we get

$$\mathbf{w}(t) - \mathbf{w}^* \approx [\mathbf{I} - \alpha(t)\mathbf{H}(\mathbf{w}^*)][\mathbf{w}(t-1) - \mathbf{w}^*]$$

From the above equation we can see that both the size of the Hessian matrix and the learning rate determines the speed of the convergence. The Hessian will be small if the cost function $\jmath(\mathbf{w})$ is very flat and the second partial derivatives is very small at the minimum. The result is the slow convergence with the constraint that $\alpha(t)$ is fixed. If the cost function is fixed, then it is impossible to change the shape of it. In that case the only option is to choose a proper value of the step size $\alpha(t)$.

But it is also very difficult to choose the proper value of $\alpha(t)$ which is very essential although. Too small values of $\alpha(t)$ will result in slow convergence and too large values of it will cause overshooting and instability. One possible solution to this problem is so called momentum method where a two-step iteration is used. There are also some adaptive rules for step size selection but that is not our objective to discuss here.

3.7.2 Stochastic gradient descent

Stochastic gradient descent is specific form of gradient descent rule where a specific kind of cost function $\jmath(\mathbf{W})$ or $\jmath(\mathbf{w})$ is considered.

The estimation of the ICA model requires the observation data. There is no solution in this method without any available data. Thus ICA is totally data dependent technique. Usually, in this method following is the form of the cost functions:

$$\jmath(\mathbf{w}) = \mathbf{E}\{g(\mathbf{w}, \mathbf{x})\} \tag{3.23}$$

where $\mathbf{x}$ is the available observation vector of density $f(\mathbf{x})$ which is not known. $\mathbf{E}\{.\}$ is the expectation operator with respect to this density. For the estimation of the ICA model we must have the samples $\mathbf{x}(1), \mathbf{x}(2), ...,$ available.

Before proceeding to the discussion about the types of algorithms used in this steepest descent algorithm we need to know how the gradient and the Hessian can be computed from the cost function shown in equation (3.23). The gradient is the first derivative and the Hessian is the second derivative of the cost function with respect to the elements of vector $\mathbf{w}$. In both of the cases, the derivatives can be taken inside the integral operator. Thus the gradient is defined as:

$$\frac{\partial}{\partial \mathbf{w}} \mathbf{E}\{g(\mathbf{w}, \mathbf{x})\} = \frac{\partial}{\partial \mathbf{w}} \int g(\mathbf{w}, \xi) f(\xi) d\xi \tag{3.24}$$

$$= \int [\frac{\partial}{\partial \mathbf{w}} g(\mathbf{w}, \xi)] f(\xi) d\xi \tag{3.25}$$

Similarly we can find the Hessian by taking the second derivatives of equation (3.23). To find both the gradient and the Hessian of the function $g(\mathbf{w}, \mathbf{x})$ it must be differentiable.

In general there are two different kinds of learning algoriths followed in the most of the ICA methods. The first one is called *batch learning* and the second one is the *on-line learning*. In both of the cases, the sample data of the observation vector $\mathbf{x}$ must be available. Batch learning is the simplest case and it is applicable when the observations that keep on coming do not change with time. For, batch learning the steepest descent rule has the following form:

$$\mathbf{w}(t) = \mathbf{w}(t-1) - \alpha(t)\frac{\partial}{\partial \mathbf{w}}\mathbf{E}\{g(\mathbf{w}, \mathbf{x}(t))\}|_{\mathbf{w}=\mathbf{w}(t-1)} \tag{3.26}$$

As we can see from equation (3.26), everytime the new values of $\mathbf{w}(t)$ is estimated from the previous values of $\mathbf{w}(t)$ which is $\mathbf{w}(t-1)$. At every step of the iteration the whole observation data is used to calculate the expectation of the cost function over the sample $\mathbf{x}(1), ..., \mathbf{x}(T)$. In fact, this mean value is used as the expection of the cost function used in equation (3.26).

This batch algorithm will not give the true information in case of the slowly changing observation data. Because, the statistics will be changing with every new oncoming observation data. To be able to track the changing statistics one should use the latest observation vector $\mathbf{x}(t)$ at every step of the iteration instead of the mean value of the whole observation vector in batch. In that case the equation (3.26) is changed to the on-line learning rule where the expectation operator has been dropped:

$$\mathbf{w}(t) = \mathbf{w}(t-1) - \alpha(t)\frac{\partial}{\partial \mathbf{w}}g(\mathbf{w}, \mathbf{x})|_{\mathbf{w}=\mathbf{w}(t-1)} \tag{3.27}$$

Because of the changing statistics of the every new on coming observation data the directions of the instantenous gradients change in every subsequent iteration. But the average direction in which the algorithm proceeds is quite same as the direction in

which steepest descent rule moves. Usually the sample vectors $\mathbf{x}(t)$ are chosen by the random choice or in cycle. The better choice is the shuffling or random selection. It is possible to obtain a good accuracy of the result after running the online learning rule over the training set many times. If the training set is fixed it should be used several times. That is why it requires many more steps to converge the on-line algorithm. If we compare the speed, the stochastic algorithm converges much slower than the steepest descent algorithm. But it has a great advantage over the steepest descent rule which is computationally it is less complex than the steepest algorithm. Computational cost is also reduced T times because if we have T number of sample vectors then the function $\frac{\partial}{\partial \mathbf{w}} \mathrm{E}\{g(\mathbf{w}, \mathbf{x}\}$ is needed to compute T times and these values are summed up and divided by T to obtain the average value. But in online-learning the function $\frac{\partial}{\partial \mathbf{w}} g(\mathbf{w}, \mathbf{x})$ is needed to compute only for once for each iteration.

3.8 Conclusions

In this chapter we have described the contrast functions needed for estimation of the data model of independent component analysis. The importance of contrast functions, the classifications of contrast functions, analysis of estimators and choice of contrast functions have also been included here. The definitions of gradient, vector gradient, matrix gradient, stochastic gradient and introduction to adaptation rules have also been illustrated in this chapter.

Chapter 4

Ideal solution for an $N \times N$ network

Kari Torokkola [71] presented the ideal solution for the separation of source signals from observed mixtures for two sources and two sensors case. Here, we have extended the solution for an arbitrary $N \times N$ network utilizing the same feedback architecture but with certain additional conditions and assumptions. Before presenting our proposed solution, the methodology developed in [71] is briefly reviewed.

Consider for simplicity the case of two sources expressed in the discrete-time domain (these results will subsequently be generalized to an arbitrary number of sources):

$$\begin{aligned} X_1(z) &= A_{11}(z)S_1(z) + A_{12}(z)S_2(z) \\ X_2(z) &= A_{21}(z)S_1(z) + A_{22}(z)S_2(z) \end{aligned} \tag{4.1}$$

where $A_{ij}(z)$ is the z-transform of the corresponding filter. The original sources S may be obtained using the mixtures X according to the following expressions:

$$\begin{aligned} S_1(z) &= (A_{22}(z)X_1(z) - A_{12}(z)X_2(z))/G(z) \\ S_2(z) &= (-A_{21}(z)X_1(z) + A_{11}(z)X_2(z))/G(z) \end{aligned} \tag{4.2}$$

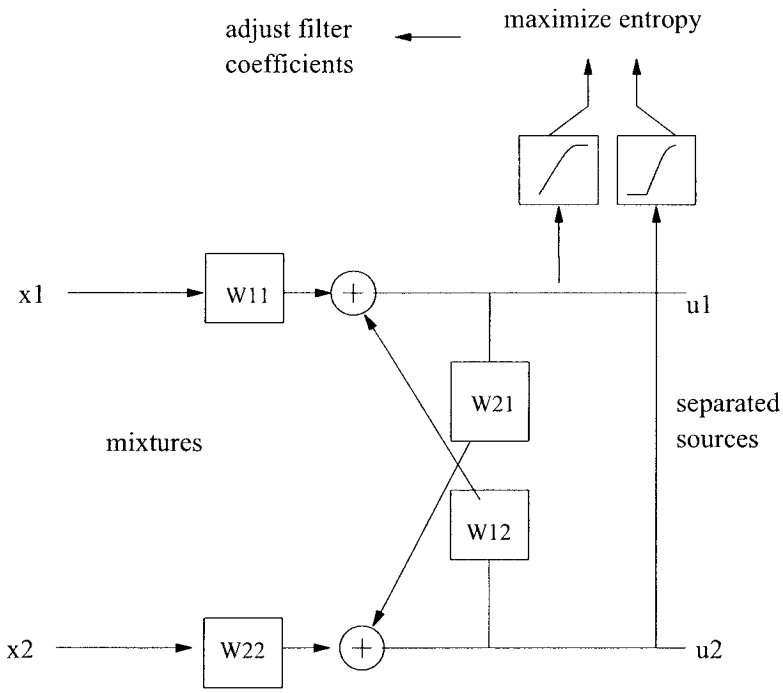

Figure 4.1: A feedback network architecture with adaptive filters for the separation of convolved sources.

where, $G(z) = A_{11}(z)A_{22}(z) - A_{12}(z)A_{21}(z)$. By referring to Fig. (4.1) the two output signals can be derived as

$$\begin{aligned} U_1(z) &= W_{11}(z)X_1(z) + W_{12}(z)U_2(z) \\ U_2(z) &= W_{21}(z)U_1(z) + W_{22}(z)X_2(z) \end{aligned} \tag{4.3}$$

Rearranging above equations we get

$$\begin{aligned} U_1(z) - W_{12}(z)U_2(z) &= W_{11}(z)X_1(z) \\ -W_{21}(z)U_1(z) + U_2(z) &= W_{22}(z)X_2(z) \end{aligned} \tag{4.4}$$

We can solve two equations to get $U_1(z)$ and $U_2(z)$ in terms of $X_1(z)$ and $X_2(z)$. Those equations are

$$U_1(z) = \frac{W_{11}(z)}{1 - W_{12}(z)\,W_{21}(z)} X_1(z) + \frac{W_{12}(z)\,W_{22}(z)}{1 - W_{12}(z)\,W_{21}(z)} X_2(z)$$
$$-U_2(z) = \frac{W_{11}(z)\,W_{21}(z)}{1 - W_{12}(z)\,W_{21}(z)} X_1(z) + \frac{W_{22}(z)}{1 - W_{12}(z)\,W_{21}(z)} X_2(z) \tag{4.5}$$

Ideally $U_1(z)$ should be identical to $S_1(z)$ and $U_2(z)$ should be identical to $S_2(z)$. By equating the coefficients of $X_1(z)$ and $X_2(z)$ we get the following four equations:

$$\frac{A_{22}(z)}{A_{11}(z)A_{22}(z) - A_{12}(z)A_{21}(z)} = \frac{W_{11}(z)}{1 - W_{12}(z)\,W_{21}(z)} \tag{4.6}$$
$$\frac{A_{11}(z)}{A_{11}(z)A_{22}(z) - A_{12}(z)A_{21}(z)} = \frac{W_{22}(z)}{1 - W_{12}(z)\,W_{21}(z)} \tag{4.7}$$
$$\frac{-A_{12}(z)}{A_{11}(z)A_{22}(z) - A_{12}(z)A_{21}(z)} = \frac{W_{12}(z)\,W_{22}(z)}{1 - W_{12}(z)\,W_{21}(z)} \tag{4.8}$$
$$\frac{-A_{21}(z)}{A_{11}(z)A_{22}(z) - A_{12}(z)A_{21}(z)} = \frac{W_{11}(z)\,W_{21}(z)}{1 - W_{12}(z)\,W_{21}(z)} \tag{4.9}$$

Dividing equation (4.9) by equation (4.6) and equation (4.8) by equation (4.7) we get the expressions of $W_{12}(z)$ and $W_{21}(z)$ and putting those expressions in equations (4.6) and (4.7) we obtain the expressions for $W_{11}(z)$ and $W_{22}(z)$ in terms of the four demixing filters $A_{11}(z)$, $A_{22}(z)$, $A_{12}(z)$ and $A_{21}(z)$, namely

$$W_{11}(z) = A_{11}(z)^{-1} \quad W_{12}(z) = -A_{12}(z)A_{11}(z)^{-1}$$
$$W_{22}(z) = A_{22}(z)^{-1} \quad W_{21}(z) = -A_{21}(z)A_{22}(z)^{-1} \tag{4.10}$$

However by maximizing the entropy at the output nodes will result in $W_{11}(z)$ and $W_{22}(z)$ that are not only inverting $A_{11}(z)$ and $A_{22}(z)$, but also whitening the sources as well. This can be avoided by constraining the characterization of $W_{11}(z)$ and $W_{22}(z)$ to simply constant gains. In the ideal case, $W_{12}(z)$ and $W_{21}(z)$ will have the following solution:

$$W_{11}(z) = 1,\ W_{12}(z) = -A_{12}(z)A_{22}(z)^{-1}$$
$$W_{22}(z) = 1,\ W_{21}(z) = -A_{21}(z)A_{11}(z)^{-1} \tag{4.11}$$

Using the above ideal solution in equation (4.11) one obtains $U_1(z) = S_1(z)A_{11}(z)$ and $U_2(z) = S_2(z)A_{22}(z)$, which represent what each sensor would have observed in the absence of interfering sources and no distorting effects. The existence of this solution requires that $A_{11}(z)$ and $A_{22}(z)$ have stable inverses, in addition to $G(z)$ having also a stable inverse. Therefore, if these conditions are not satisfied, the network is unable to achieve the required source separation objective. Note that when both $A_{11}(z)$ and $A_{22}(z)$ are set to 1, the only stable inverse requirement would be for $G(z)$.

4.1 Ideal solution of three sources and three sensors case

Let us look at three sources in the z-transform domain for simplicity; this can be generalized to any number of sources.

$$X_1(z) = A_{11}(z)S_1(z) + A_{12}(z)S_2(z) + A_{13}(z)S_3(z)$$
$$X_2(z) = A_{21}(z)S_1(z) + A_{22}(z)S_2(z) + A_{23}(z)S_3(z)$$
$$X_3(z) = A_{31}(z)S_1(z) + A_{32}(z)S_2(z) + A_{33}(z)S_3(z) \tag{4.12}$$

where A_{ij} are the z-transforms of any kind of filters as in the two sources and two sensor cases. By applying the delta rule we can solve the sources S in terms of the mixtures X :

$$
\begin{aligned}
S_1(z) &= \frac{(A_{22}(z)A_{33}(z) - A_{32}(z)A_{23}(z))}{G_{m3}(z)}X_1(z) + \frac{(A_{13}(z)A_{32}(z) - A_{12}(z)A_{33}(z))}{G_{m3}(z)}X_2(z) \\
&\quad + \frac{(A_{12}(z)A_{23}(z) - A_{22}(z)A_{13}(z))}{G_{m3}(z)}X_3(z) \\
S_2(z) &= \frac{(A_{31}(z)A_{23}(z) - A_{21}(z)A_{33}(z))}{G_{m3}(z)}X_1(z) + \frac{(A_{11}(z)A_{33}(z) - A_{31}(z)A_{13}(z))}{G_{m3}(z)}X_2(z) \\
&\quad + \frac{(A_{21}(z)A_{13}(z) - A_{11}(z)A_{23}(z))}{G_{m3}(z)}X_3(z) \\
S_3(z) &= \frac{(A_{21}(z)A_{32}(z) - A_{31}(z)A_{22}(z))}{G_{m3}(z)}X_1(z) + \frac{(A_{31}(z)A_{12}(z) - A_{11}(z)A_{32}(z))}{G_{m3}(z)}X_2(z) \\
&\quad + \frac{(A_{11}(z)A_{22}(z) - A_{21}(z)A_{12}(z))}{G_{m3}(z)}X_3(z) \qquad (4.13)
\end{aligned}
$$

Now from feedback architecture of 3-sources and 3-sensors case we can write the equations of recovered signals in z-domain

$$
\begin{aligned}
U_1(z) &= W_{11}(z)X_1(z) + W_{12}(z)U_2(z) + W_{13}(z)U_3(z) \\
U_2(z) &= W_{21}(z)U_1(z) + W_{22}(z)X_2(z) + W_{23}(z)U_3(z) \\
U_3(z) &= W_{31}(z)U_1(z) + W_{32}(z)U_2(z) + W_{33}(z)X_3(z) \qquad (4.14)
\end{aligned}
$$

Assuming direct demixing filters $W_{11}(z)$, $W_{22}(z)$ and $W_{33}(z)$ as unity filters and applying delta rule to solve the equations we get for $U_1(z)$, $U_2(z)$ and $U_3(z)$:

$$
\begin{aligned}
U_1(z) &= \frac{(1 - W_{23}(z)W_{32}(z))}{G_{d3}(z)}X_1(z) + \frac{(W_{12}(z) + W_{13}(z)w_{32}(z))}{G_{d3}(z)}X_2(z) + \\
&\quad \frac{(W_{13}(z) + W_{12}(z)W_{23}(z))}{G_{d3}(z)}X_3(z) \\
U_2(z) &= \frac{(W_{21}(z) + W_{23}(z)W_{31}(z))}{G_{d3}(z)}X_1(z) + \frac{(1 - W_{13}(z)W_{31}(z))}{G_{d3}(z)}X_2(z) + \\
&\quad \frac{(W_{23}(z) + W_{13}(z)W_{21}(z))}{G_{d3}(z)}X_3(z) \\
U_3(z) &= \frac{(W_{31}(z) + W_{21}(z)W_{32}(z))}{G_{d3}(z)}X_1(z) + \frac{(W_{32}(z) + W_{31}(z)W_{12}(z))}{G_{d3}(z)}X_2(z) \\
&\quad + \frac{(1 - W_{21}(z)W_{12}(z))}{G_{d3}(z)}X_3(z) \qquad (4.15)
\end{aligned}
$$

Our target is to separate the signals so that they become statistically as independent as possible, ideally each recovered signal should be the same as its respective original uncorrupted signal, i.e.,

$$U_1(z) = S_1(z)$$
$$U_2(z) = S_2(z)$$
$$U_3(z) = S_3(z)$$

Equating the coefficients of $X_1(z)$, $X_2(z)$ and $X_3(z)$ we get

$$\begin{aligned}
\frac{(1 - W_{23}(z)\,W_{32}(z))}{G_{d3}(z)} &= \frac{(A_{22}(z)A_{33}(z) - A_{32}(z)A_{23}(z))}{G_{m3}(z)} \\
\frac{(W_{21}(z) + W_{23}(z)\,W_{31}(z))}{G_{d3}(z)} &= \frac{(A_{31}(z)A_{23}(z) - A_{21}(z)A_{33}(z))}{G_{m3}(z)} \\
\frac{(W_{31}(z) + W_{21}(z)\,W_{32}(z))}{G_{d3}(z)} &= \frac{(A_{21}(z)A_{32}(z) - A_{31}(z)A_{22}(z))}{G_{m3}(z)} \\
\frac{(W_{12}(z) + W_{13}(z)\,W_{32}(z))}{G_{d3}(z)} &= \frac{(A_{13}(z)A_{32}(z) - A_{12}(z)A_{33}(z))}{G_{m3}(z)} \\
\frac{(1 - W_{13}(z)\,W_{31}(z))}{G_{d3}(z)} &= \frac{(A_{11}(z)A_{33}(z) - A_{31}(z)A_{13}(z))}{G_{m3}(z)} \\
\frac{(W_{32}(z) + W_{31}(z)\,W_{12}(z))}{G_{d3}(z)} &= \frac{(A_{31}(z)A_{12}(z) - A_{11}(z)A_{32}(z))}{G_{m3}(z)} \\
\frac{(W_{13}(z) + W_{12}(z)\,W_{23}(z))}{G_{d3}(z)} &= \frac{(A_{12}(z)A_{23}(z) - A_{22}(z)A_{13}(z))}{G_{m3}(z)} \\
\frac{(W_{23}(z) + W_{13}(z)\,W_{21}(z))}{G_{d3}(z)} &= \frac{(A_{11}(z)A_{23}(z) - A_{21}(z)A_{13}(z))}{G_{m3}(z)} \\
\frac{(1 - W_{21}(z)\,W_{12}(z)}{G_{d3}(z)} &= \frac{(A_{11}(z)A_{22}(z) - A_{21}(z)A_{12}(z))}{G_{m3}(z)}
\end{aligned} \tag{4.16}$$

Let us denote the right side of the above equations as below:

$$\frac{(A_{22}(z)A_{33}(z) - A_{32}(z)A_{23}(z))}{G_{m3}(z)} = B$$
$$\frac{(A_{31}(z)A_{23}(z) - A_{21}(z)A_{33}(z))}{G_{m3}(z)} = B1$$
$$\frac{(A_{21}(z)A_{32}(z) - A_{31}(z)A_{22}(z))}{G_{m3}(z)} = B2$$
$$\frac{(A_{13}(z)A_{32}(z) - A_{12}(z)A_{33}(z))}{G_{m3}(z)} = C1$$
$$\frac{(A_{11}(z)A_{33}(z)) - (A_{31}(z)A_{13}(z))}{G_{m3}(z)} = C$$
$$\frac{(A_{31}(z)A_{12}(z) - A_{11}(z)A_{32}(z))}{G_{m3}(z)} = C2$$
$$\frac{(A_{12}(z)A_{23}(z) - A_{22}(z)A_{13}(z))}{G_{m3}(z)} = A1$$
$$\frac{(A_{11}(z)A_{23}(z) - A_{21}(z)A_{13}(z))}{G_{m3}(z)} = A2$$
$$\frac{(A_{11}(z)A_{22}(z) - A_{21}(z)A_{12}(z))}{G_{m3}(z)} = A$$

Then the equations become

$$\frac{(1 - W_{23}(z)\,W_{32}(z))}{G_{d3}(z)} = B \tag{4.17}$$

$$\frac{(W_{21}(z) + W_{23}(z)\,W_{31}(z))}{G_{d3}(z)} = B1 \tag{4.18}$$

$$\frac{(W_{31}(z) + W_{21}(z)\,W_{32}(z))}{G_{d3}(z)} = B2 \tag{4.19}$$

$$\frac{(W_{12}(z) + W_{13}(z)\,W_{32}(z))}{G_{d3}(z)} = C1 \tag{4.20}$$

$$\frac{(1 - W_{13}(z)\,W_{31}(z))}{G_{d3}(z)} = C \tag{4.21}$$

$$\frac{(W_{32}(z) + W_{31}(z)\,W_{12}(z))}{G_{d3}(z)} = C2 \tag{4.22}$$

$$\frac{(W_{13}(z) + W_{12}(z)\,W_{23}(z))}{G_{d3}(z)} = A1 \tag{4.23}$$

$$\frac{(W_{23}(z) + W_{13}(z)\,W_{21}(z))}{G_{d3}(z)} = A2 \tag{4.24}$$

$$\frac{(1 - W_{21}(z)\,W_{12}(z))}{G_{d3}(z)} = A \tag{4.25}$$

Dividing equation (4.23) by equation (4.25)

$$\frac{W_{13}(z) + W_{12}(z)\,W_{23}(z)}{1 - W_{21}(z)\,W_{12}(z)} = \frac{A1}{A}$$

After cross multiplication and rearranging we get

$$W_{12}(z) = \frac{A1 - A\,W_{13}(z)}{A\,W_{23}(z) + A1\,W_{21}(z)} \tag{4.26}$$

Dividing equation (4.23) by equation (4.24)

$$\frac{W_{13}(z) + W_{12}(z)\,W_{23}(z)}{W_{23}(z) + W_{13}(z)\,W_{21}(z)} = \frac{A1}{A2}$$

Cross multiplying and rearranging we get

$$W_{13}(z) = \frac{A1\,W_{23}(z) - A2\,W_{12}(z)\,W_{23}(z)}{A2 - A1\,W_{21}(z)} \tag{4.27}$$

Putting the value of $W_{13}(z)$ in equation (4.26) yields

$$W_{12}(z) = \frac{A1 - \frac{AA1\,W_{23}(z) - AA2\,W_{12}(z)\,W_{23}(z)}{A2 - A1\,W_{21}(z)}}{A\,W_{23}(z) + A1\,W_{21}(z)}$$

Cross multiplying we get

$$\begin{aligned} &AA2\,W_{12}(z)\,W_{23}(z) - AA1\,W_{12}(z)\,W_{21}(z)\,W_{23}(z) \\ &+A1A2\,W_{12}(z)\,W_{21}(z) - A1^2\,W_{12}(z)\,W_{21}(z)^2 \\ &= A1A2 - A1^2\,W_{21}(z) - AA1\,W_{23}(z) + AA2\,W_{12}(z)\,W_{23}(z) \end{aligned}$$

Cancelling the common terms from both sides and rearranging results in

$$\begin{aligned} &W_{23}(z)(-AA1\,W_{12}(z)\,W_{21}(z) + AA1) \\ &= A1A2 - A1A2\,W_{12}(z)\,W_{21}(z) - A1^2\,W_{21}(z) + A1^2\,W_{12}(z)\,W_{21}(z)^2 \end{aligned}$$

After factorization we can write the above equation in product form as below

$$W_{23}(z)AA1(1 - W_{12}(z)\,W_{21}(z)) = A1(1 - W_{12}(z)\,W_{21}(z))(A2 - A1\,W_{21}(z))$$

Dividing both sides of the equation by the common factor $1 - W_{12}(z)\,W_{21}(z)$ we get the value of $W_{23}(z)$ as

$$W_{23}(z) = \frac{A2 - A1\,W_{21}(z)}{A} \tag{4.28}$$

Putting the value of $W_{23}(z)$ in equation (4.27) We get the value of $W_{13}(z)$

$$W_{13}(z) = \frac{A1 - A2\,W_{12}(z)}{A} \tag{4.29}$$

Dividing equation (4.18) by equation (4.25)

$$\frac{W_{21}(z) + W_{23}(z)\,W_{31}(z)}{1 - W_{21}(z)\,W_{12}(z)} = \frac{B1}{A}$$

Putting the value of $W_{23}(z)$ from equation (4.28) in above equation and cross multiplying and rearranging we get

$$\begin{aligned}(A2 - A1\,W_{21}(z))\,W_{31}(z) &= B1(1 - W_{12}(z)\,W_{21}(z)) - A\,W_{21}(z) \\ &= B1 - W_{21}(z)(A + B1\,W_{12}(z))\end{aligned}$$

Rearranging we get the equation for $W_{31}(z)$

$$W_{31}(z) = \frac{W_{21}(z)(A + B1\,W_{12}(z)) - B1}{A1\,W_{21}(z) - A2} \tag{4.30}$$

Dividing equation (4.17) by equation (4.25) we get

$$\frac{1 - W_{23}(z)\,W_{32}(z)}{1 - W_{12}(z)\,W_{21}(z)} = \frac{B}{A}$$

Putting the value of $W_{23}(z)$ from equation (4.28) in the above equation and cross multiplying and rearranging we get the expression of $W_{32}(z)$

$$W_{32}(z) = \frac{B(1 - W_{12}(z)\,W_{21}(z)) - A}{A1\,W_{21}(z) - A2} \tag{4.31}$$

Dividing equation (4.19) by equation (4.25) and using the expressions of $W_{31}(z)$ and $W_{32}(z)$ from equations (4.30) and (4.31) respectively, we have

$$\begin{aligned}
\frac{\frac{-B1+W_{21}(z)(A+B1\,W_{12}(z))}{-A2+A1\,W_{21}(z)}}{1-W_{12}(z)\,W_{21}(z)} + \\
\frac{\frac{B\,W_{21}(z)(1-W_{12}(z)\,W_{21}(z))-A\,W_{21}(z)}{A1\,W_{21}(z)-A2}}{1-W_{12}(z)\,W_{21}(z)} &= \frac{B2}{A} \\
\frac{A\,W_{21}(z)+B1\,W_{12}(z)\,W_{21}(z)-B1}{(A1\,W_{21}(z)-A2)(1-W_{12}(z)\,W_{21}(z))} \\
+\frac{B\,W_{21}(z)-B\,W_{12}(z)\,W_{21}(z)^2-A\,W_{21}(z)}{(A1\,W_{21}(z)-A2)(1-W_{12}(z)\,W_{21}(z))} &= \frac{B2}{A} \\
\frac{(B\,W_{21}(z)-B1)(1-W_{12}(z)\,W_{21}(z))}{(A1\,W_{21}(z)-A2)(1-W_{12}(z)\,W_{21}(z))} &= \frac{B2}{A} \\
\frac{B\,W_{21}(z)-B1}{A1\,W_{21}(z)-A2} &= \frac{B2}{A} \\
AB\,W_{21}(z)-AB1 &= \\
A1B2\,W_{21}(z)-A2B2
\end{aligned}$$

$$W_{21}(z) = \frac{AB1 - A2B2}{AB - A1B2} \tag{4.32}$$

Putting the value of $W_{21}(z)$ from equation (4.32) in equation (4.28) we get the value of $W_{23}(z)$

$$\begin{aligned}
W_{23}(z) &= \frac{A2 - A1\,W_{21}(z)}{A} \\
&= \frac{A2 - A1\frac{(AB1-A2B2)}{AB-A1B2}}{A} \\
&= \frac{AA2B - A1A2B2 - AA1B1 + A1A2B2}{A(AB - A1B2)} \\
&= \frac{A(A2B - A1B1)}{AAB - A1B2} \\
W_{23}(z) &= \frac{A2B - A1B1}{AB - A1B2}
\end{aligned} \tag{4.33}$$

Dividing equation (4.21) by equation (4.25) we have

$$\frac{1 - W_{13}(z)\,W_{31}(z)}{1 - W_{12}(z)\,W_{21}(z)} = \frac{C}{A}$$

$$\frac{1 - \frac{A1 - A2\,W_{12}(z)}{A}\,\frac{B1 - W_{21}(z)(A + B1\,W_{12}(z))}{A2 - A1\,W_{21}(z)}}{1 - W_{12}(z)\,W_{21}(z)} = \frac{C}{A}$$

After simplifying

$$\frac{AA2 - AA1\,W_{21}(z) - \{A1B1 - A1\,W_{21}(z)(A + B1\,W_{12}(z) - A2B1\,W_{12}(z)}{A(A2 - A1\,W_{21}(z))(1 - W_{12}(z)\,W_{21}(z))} +$$
$$\frac{A2B1\,W_{12}(z) + A2\,W_{12}(z)\,W_{21}(z)(A + B1\,W_{12}(z))\}}{A(A2 - A1\,W_{21}(z))(1 - W_{12}(z)\,W_{21}(z))} = \frac{C}{A}$$

Cancelling the common terms from numerator and also from the denominator of the left side and the right side of the equation we get

$$\frac{AA2 - A1B1 + A1B1\,W_{12}(z)\,W_{21}(z) + A2B1\,W_{12}(z)}{(1 - W_{12}(z)\,W_{21}(z))(A2A1\,W_{21}(z))}$$
$$- \frac{AA2\,W_{12}(z)\,W_{21}(z) - B1A2\,W_{12}(z)^2\,W_{21}(z)}{(1 - W_{12}(z)\,W_{21}(z))(A2A1\,W_{21}(z))} = C$$

Factorizing the numerator in product form we get

$$\frac{AA2(1 - (1 - W_{12}(z)\,W_{21}(z)))A1B1(1 - (1 - W_{12}(z)\,W_{21}(z)))}{(A2A1\,W_{21}(z))(1 - W_{12}(z)\,W_{21}(z))}$$
$$+ \frac{A2B1\,W_{12}(z)(1 - W_{12}(z)\,W_{21}(z)))}{(A2A1\,W_{21}(z))(1 - W_{12}(z)\,W_{21}(z))} = C$$
$$\frac{(AA2A1B1 + A2B1\,W_{12}(z))(1 - W_{12}(z)\,W_{21}(z))}{(A2A1\,W_{21}(z))(1 - W_{12}(z)\,W_{21}(z))} = C$$

Cancelling the common terms from numerator and denominator of the left side of the equation and putting the value of $W_{21}(z)$ we get

$$A2B1\,W_{12}(z) + AA2A1B1 = C\{A2 - A1(\frac{AB1 - A2B2}{AB - A1B2})\}$$
$$= \frac{AC(A2B - A1B1)}{AB - A1B2}$$
$$W_{21}(z) = \frac{AC}{A2B1}(\frac{A2B - A1B1}{AB - A1B2}) - \frac{A}{B1} + \frac{A1}{A2}$$
$$W_{12}(z) = \frac{A}{B1}(\frac{C}{A2}W_{23}(z) - 1) + \frac{A1}{A2} \tag{4.34}$$

Using the expression of $W_{12}(z)$ from equation (4.34) we can find the expression of $W_{13}(z)$ in terms of $W_{23}(z)$ from equation (4.29)

$$\begin{aligned} W_{13}(z) &= \frac{A1 - A2\,W_{12}(z)}{A} \\ &= \frac{A1 - \frac{AA2}{B1}(\frac{C}{A2}W_{23}(z)) - 1) - A1}{A} \\ &= \frac{A2}{B1}(1 - \frac{C}{A2}W_{23}(z)) \\ W_{13}(z) &= \frac{1}{B1}(A2 - C\,W_{23}(z)) \end{aligned} \tag{4.35}$$

Now using the expressions of $W_{12}(z)$ and $W_{21}(z)$ from equations (4.34) and (4.29) respectively we can find the expression of $W_{31}(z)$ from equation (4.30)

$$\begin{aligned} W_{31}(z) &= \frac{W_{21}(z)(A + B1\,W_{12}(z))B1}{A1\,W_{21}(z) - A2} \\ A + B1\,W_{12}(z) &= A + A(\frac{C}{A2}W_{23}(z) - 1) + \frac{A1B1}{A2} \\ &= A1 + \frac{C}{A2}W_{23}(z) - 1 + \frac{A1B1}{A2} \\ &= \frac{AC\,W_{23}(z)}{A2} + \frac{A1B1}{A2} \\ &= \frac{AC\,W_{23}(z) + A1B1}{A2} \end{aligned} \tag{4.36}$$

Now,

$$
\begin{aligned}
W_{21}(z)(A + B1\,W_{12}(z)) - B1 &= \\
&\frac{AB1 - A2B2}{AB - A1B2}\left(\frac{AC\,W_{23}(z) + A1B1}{A2}\right) - B1 \\
&= \frac{(AB1 - A2B2)(AC\,W_{23}(z) + A1B1)}{A2(AB - A1B2)} \\
&\quad - \frac{B1A2(AB - A1B2)}{A2(AB - A1B2)} \\
W_{31}(z) &= \frac{(AB1 - A2B2)(AC\,W_{23}(z) + A1B1)}{AA2(A1B1A2B)} \\
&\quad - \frac{B1A2(AB - A1B2)}{AA2(A1B1A2B)} \\
&= \frac{AC\,W_{23}(z)(AB1 - A2B2)}{AA2(A1B1 - A2B)} \\
&\quad + \frac{AA1B1^2 - A1A2B1B2 - AA2BB1}{AA2(A1B1 - A2B)} \\
&\quad + \frac{A1A2B1B2}{AA2(A1B1 - A2B)} \\
&= \frac{AC\,W_{23}(z)(AB1 - A2B2) + AB1(A1B1 - A2B)}{AA2(A1B - A2B)} \\
&= W_{23}(z)C\frac{(AB1 - A2B2)}{A1B1 - A2B} + \frac{B1}{A2} \\
&= \frac{B1 - C\,W_{23}(z)\frac{W_{21}(z)}{W_{23}(z)}}{A2}
\end{aligned}
$$

$$
W_{31}(z) = \frac{B1 - C\,W_{21}(z)}{A2} \tag{4.37}
$$

In the similar way we can find the value of $W_{32}(z)$

$$
W_{12}(z)\,W_{21}(z) = \frac{AB1 - A2B2}{AB - A1B2}\left\{\frac{A}{B1}\left(\frac{C}{A2}\,W_{23}(z) - 1\right) + \frac{A1}{A2}\right\}
$$

$$B(1 - W_{12}(z)\,W_{21}(z)) - A = B[1 - \frac{AB1 - A2B2}{AB - A1B2}\{\frac{A}{B1}(\frac{C}{A2}\,W_{23}(z) - 1) + \frac{A1}{A2}\}] - A$$

$$\frac{[(AB - A1B2) - (AB1 - A2B2)\{\frac{A}{B1}(\frac{C}{A2}\,W_{23}(z) - 1) + \frac{A1}{A2}\}] - A(AB - A1B2)}{AB - A1B2}$$

$$= \frac{(AB - A1B2)(B - A) - B(AB1 - A2B2)\{\frac{A}{B1}(\frac{C}{A2}\,W_{23}(z) - 1) + \frac{A1}{A2}\}}{AB - A1B2}$$

Now,

$$W_{32}(z) = \frac{B(1 - W_{12}(z)\,W_{21}(z)) - A}{A1\,W_{21}(z) - A2}$$

$$= \frac{(AB - A1B2)(B - A) - B(AB1 - A2B2)\{\frac{A}{B1}(\frac{C}{A2}\,W_{23}(z) - 1) - \frac{A1}{A2}\}}{A(A1B1 - A2B)}$$

Putting the value of $W_{23}(z)$ in above equation we get

$$W_{32}(z) = \frac{AA2B^2B1 - A^2A2BB1 - A1A2BB1B2 + AA1A2B1B2}{AA2B1(A1B1 - A2B)}$$

$$+\frac{A^2A2BB1 - AA2^2BB2}{AA2B1(A1B1 - A2B)}$$

$$\frac{-AA1BB1^2 + A1A2BB1B2 - ACB(AB1 - A2B2)\,W_{23}(z)}{AA2B1(A1B1 - A2B)}$$

$$= \frac{ABB1(A2B - A1B1) + AA2B2(A1B1 - A2B) - ACB(AB1 - A2B2)\,W_{23}(z)}{AA2B1(A1B1 - A2B)}$$

$$= \frac{A(A1B1 - A2B)(A2B2 - BB1)}{AA2B1(A1B1 - A2B)}$$

$$-\frac{ACB(AB1 - A2B2\,W_{23}(z))}{AA2B1(A1B1 - A2B)}$$

$$= \frac{A2B2 - BB1}{A2B1} - \frac{BC}{A2B1}\frac{(AB1 - A2B2)}{A1B1 - A2B}\,W_{23}(z)$$

$$= \frac{B2}{B1} - \frac{B}{A2} + \frac{BC}{A2B1}\frac{W_{21}(z)}{W_{23}(z)}\,W_{23}(z)$$

$$= \frac{A2B2 - BB1 + BC\,W_{21}(z)}{A2B1}$$

$$= \frac{A2B2 + B(C\,W_{21}(z) - B1)}{A2B1}$$

$$= \frac{B2}{B1} + \frac{AB}{AB1A2(C\,W_{21}(z) - B1)}$$

$$W_{32}(z) = \frac{1}{B1}[B2 - B\,W_{31}(z)] \tag{4.38}$$

Thus we have values for all the demixing coefficients

$$\begin{aligned}
W_{21}(z) &= \frac{AB1A2B2}{ABA1B2} \\
W_{23}(z) &= \frac{A2BA1B1}{ABA1B2} \\
W_{13}(z) &= \frac{1}{B1}(A2 - C\,W_{23}(z)) \\
W_{12}(z) &= \frac{A}{B1}(\frac{C}{A2}\,W_{23}(z) - 1) + \frac{A1}{A2} \\
W_{31}(z) &= \frac{B1 - C\,W_{21}(z)}{A2} \\
W_{32}(z) &= \frac{1}{B1}[B2 - B\,W_{31}(z)]
\end{aligned}$$

The values we have found of all demixing coefficients are in terms of A, $A1$, $A2$, B, $B1$, $B2$, C, $C1$, $C2$. Taking the direct mixing filters $A_{11}(z)$, $A_{22}(z)$ and $A_{33}(z)$ as the unity filters we will find the expressions of six demixing filters in terms of mixing filters $A_{12}(z)$, $A_{13}(z)$, $A_{21}(z)$, $A_{23}(z)$, $A_{31}(z)$ and $A_{32}(z)$. We do not need to find the expressions for direct demixing filters $W_{11}(z)$, $W_{22}(z)$ and $W_{33}(z)$ because in the derivation of adaptation rules for demixing coefficients we have assumed those as unity filters.

$$
\begin{aligned}
AB1 - A2B2 &= \frac{(1 - A_{12}(z)A_{21}(z))(A_{23}(z)A_{31}(z) - A_{21}(z))}{G_{m3}(z)^2} \\
&\quad - \frac{(A_{21}(z)A_{13}(z) - A_{23}(z))(A_{32}(z)A_{21}(z) - A_{31}(z))}{G_{m3}(z)^2} \\
&= \frac{A_{23}(z)A_{31}(z) - A_{21}(z) - A_{12}(z)A_{21}(z)A_{23}(z)A_{31}(z) + A_{12}(z)A_{21}(z)^2}{G_{m3}(z)^2} \\
&\quad \frac{-A_{13}(z)A_{32}(z)A_{21}(z)^2 + A_{21}(z)A_{13}(z)A_{31}(z) + A_{21}(z)A_{23}(z)A_{32}(z)}{G_{m3}(z)^2} \\
&\quad - \frac{A_{23}(z)A_{31}(z)}{G_{m3}(z)^2}
\end{aligned} \tag{4.39}
$$

$$
\begin{aligned}
AB - A1B2 &= \frac{(A_{11}(z)A_{22}(z) - A_{21}(z)A_{12}(z))(A_{22}(z)A_{33}(z) - A_{32}(z)A_{23}(z))}{G_{m3}(z)^2} \\
&\quad \frac{-(A_{12}(z)A_{23}(z) - A_{22}(z)A_{13}(z))(A_{21}(z)A_{32}(z) - A_{31}(z)A_{22}(z))}{G_{m3}(z)^2} \\
&= \frac{(1 - A_{21}(z)A_{12}(z))(1 - A_{23}(z)A_{32}(z))}{G_{m3}(z)^2} \\
&\quad \frac{(A_{12}(z)A_{23}(z) - A_{13}(z))(A_{21}(z)A_{32}(z) - A_{31}(z))}{G_{m3}(z)^2} \\
&= \frac{1 - A_{12}(z)A_{21}(z) - A_{13}(z)A_{31}(z) - A_{23}(z)A_{32}(z)}{G_{m3}(z)^2} \\
&\quad + \frac{A_{12}(z)A_{23}(z)A_{31}(z) + A_{13}(z)A_{32}(z)A_{21}(z)}{G_{m3}(z)^2}
\end{aligned} \tag{4.40}
$$

Dividing equation (4.39) by equation (4.40) we get the expression for $W_{21}(z)$

$$
\begin{aligned}
W_{21}(z) &= \frac{AB1 - A2B2}{AB - A1B2} \\
&= -A_{21}(z)
\end{aligned} \tag{4.41}
$$

Now,

$$
\begin{aligned}
A2B - A1B1 &= \frac{(A_{21}(z)A_{13}(z) - A_{23}(z))(1 - A_{23}(z)A_{32}(z))}{G_{m3}(z)^2} \\
&\quad \frac{-(A_{12}(z)A_{23}(z) - A_{13}(z))(A_{23}(z)A_{31}(z) - A_{21}(z))}{G_{m3}(z)^2} \\
&= \frac{A_{21}(z)A_{13}(z) - A_{13}(z)A_{23}(z)A_{21}(z)A_{32}(z) - A_{23}(z) + A_{23}(z)^2 A_{32}(z)}{G_{m3}(z)^2} \\
&\quad \frac{-A_{12}(z)A_{23}(z)^2 A_{31}(z) + A_{12}(z)A_{21}(z)A_{23}(z) + A_{13}(z)A_{31}(z)A_{23}(z)}{G_{m3}(z)^2} \\
&\quad \frac{-A_{13}(z)A_{21}(z)}{G_{m3}(z)^2} \\
&= \frac{-A_{23}(z)(1 - A_{23}(z)A_{32}(z) - A_{12}(z)A_{21}(z) - A_{13}(z)A_{31}(z)}{G_{m3}(z)^2} + \\
&\quad \frac{A_{13}(z)A_{21}(z)A_{32}(z) + A_{12}(z)A_{23}(z)A_{31}(z))}{G_{m3}(z)^2}
\end{aligned}
$$

Therefore,

$$
W_{23}(z) = \frac{A2B - A1B1}{AB - A1B2} = -A_{23}(z) \tag{4.42}
$$

$$
\begin{aligned}
W_{13}(z) &= \frac{A2 - C\,W_{23}(z)}{B1} \\
&= \frac{A_{21}(z)A_{13}(z) - A_{23}(z) + A_{23}(z)(1 - A_{13}(z)A_{31}(z))}{A_{23}(z)A_{31}(z) - A_{21}(z)} \\
&= \frac{-A_{13}(z)(A_{23}(z)A_{31}(z) - A_{21}(z))}{(A_{23}(z)A_{31}(z) - A_{21}(z))} \\
&= \frac{-A_{13}(z)(A_{23}(z)A_{31}(z) - A_{21}(z))}{(A_{23}(z)A_{31}(z) - A_{21}(z))} \\
W_{13}(z) &= -A_{13}(z)
\end{aligned} \tag{4.43}
$$

$$
\begin{aligned}
W_{12}(z) &= \frac{A}{B1}(-\frac{C}{A2}A_{23}(z)-1)+\frac{A1}{A2} \\
&= \frac{(1-A_{12}(z)A_{21}(z))}{(A_{23}(z)A_{31}(z)-A_{21}(z))}\{-A_{23}(z)\frac{(1-A_{13}(z)A_{31}(z))}{(A_{21}(z)A_{13}(z)-A_{23}(z))}-1\} \\
&\quad +\frac{(A_{12}(z)A_{23}(z)-A_{13}(z))}{A_{21}(z)A_{13}(z)-A_{23}(z)} \\
&= \frac{(1-A_{12}(z)A_{21}(z))}{(A_{23}(z)A_{31}(z)-A_{21}(z))}\{\frac{-A_{23}(z)+A_{23}(z)A_{13}(z)A_{31}(z)}{A_{21}(z)A_{13}(z)-A_{23}(z)}\} \\
&\quad +\frac{(1-A_{12}(z)A_{21}(z))}{(A_{23}(z)A_{31}(z)-A_{21}(z))}\{\frac{-A_{21}(z)A_{13}(z)+A_{23}(z)}{A_{21}(z)A_{13}(z)-A_{23}(z)}\} \\
&\quad +\frac{A_{12}(z)A_{23}(z)-A_{13}(z)}{A_{21}(z)A_{13}(z)-A_{23}(z)} \\
&= \frac{(1-A_{12}(z)A_{21}(z))}{(A_{23}(z)A_{31}(z)-A_{21}(z))}A_{13}(z)\frac{((A_{23}(z)A_{31}(z)-A_{21}(z))}{(A_{21}(z)A_{13}(z)-A_{23}(z))} \\
&\quad +\frac{A_{12}(z)A_{23}(z)-A_{13}(z)}{A_{21}(z)A_{13}(z)-A_{23}(z)} \\
&= \frac{A_{13}(z)-A_{13}(z)A_{12}(z)A_{21}(z)+A_{12}(z)A_{23}(z)-A_{13}(z)}{A_{21}(z)A_{13}(z)-A_{23}(z)} \\
&= \frac{-A_{12}(z)(A_{13}(z)}{A_{21}(z)-A_{23}(z))}(A_{13}(z)A_{21}(z)-A_{23}(z))
\end{aligned}
$$

$$
W_{12}(z) = -A_{12}(z) \tag{4.44}
$$

$$
\begin{aligned}
W_{31}(z) &= \frac{B1-C\,W_{31}(z)}{A2} \\
&= \frac{A_{32}(z)A_{31}(z)-A_{21}(z)+A_{21}(z)(1-A_{13}(z)A_{31}(z))}{A_{21}(z)A_{13}(z)-A_{23}(z)} \\
&= \frac{A_{23}(z)A_{31}(z)-A_{21}(z)+A_{21}(z)-A_{21}(z)A_{13}(z)A_{31}(z)}{A_{21}(z)A_{13}(z)-A_{23}(z)} \\
&= \frac{-A_{31}(z)(A_{21}(z)A_{13}(z)-A_{23}(z))}{(A_{21}(z)A_{13}(z)-A_{23}(z))}
\end{aligned}
$$

$$
W_{31}(z) = -A_{31}(z) \tag{4.45}
$$

$$
\begin{aligned}
W_{32}(z) &= \frac{B2 - B\,W_{31}(z)}{B1} \\
&= \frac{A_{32}(z)A_{21}(z) - A_{31}(z) + A_{31}(z)(1 - A_{23}(z)A_{32}(z))}{A_{23}(z)A_{31}(z) - A_{21}(z)} \\
&= -A_{32}(z)\frac{(A_{23}(z)A_{31}(z) - A_{21}(z))}{(A_{23}(z)A_{31}(z) - A_{21}(z))}
\end{aligned}
\tag{4.46}
$$

$$
W_{32}(z) = -A_{32}(z) \tag{4.47}
$$

The above results are the same as the results in the two sources and two sensor case which were

$$
\begin{aligned}
W_{12}(z) &= -A_{12}(z) \\
W_{21}(z) &= -A_{21}(z) \\
W_{11}(z) &= 1 \\
\text{and } W_{22}(z) &= 1
\end{aligned}
\tag{4.48}
$$

Thus, generally for any NXN network with feedback architecture and assuming direct filters have the unity gain we can write the relationship between mixing and demixing filters as

$$
W_{ij}(z) = -A_{ij}(z) \tag{4.49}
$$

which is the ideal solution when direct mixing and demixing filters have unity gain.

4.2 Conclusions

In this chapter we have derived the ideal solution for $N \times N$ feedback network architecture with a certain constraint that the direct mixing environmental and demixing filters have unity gains. The relations found between mixing and demixing filters are essential to verify if the whole system is stable or unstable. At first we have introduced Kari Torokkola's [71] feedback network architecture and his derivation of ideal solution for two sources and two sensors case and afterwards we have extended the solution for $N \times N$ case.

Chapter 5

Derivation of adaptation rules

5.1 The one input and one output case

If a single input x is passed through a transforming function $g(x)$ to get an output variable y, both the output entropy $H(y)$ and mutual information between the input variable x and the output variable y are maximized when the high density parts of the probability desnity function (pdf) $f_x(x)$ of x is aligned with the highly sloping parts of the function $g(x)$. According to [69] "this is the idea of matching a neuron's input-output function to the expected distribution of signals." When $g(x)$ has a unique inverse i.e., it increases or decreases monotonically, the pdf, $f_y(y)$, of the output variable, y, can be expressed in terms of the pdf, $f_x(x)$, of the input variable, x, [69]:

$$f_y(y) = \frac{f_x(x)}{|\partial y/\partial x|} \tag{5.1}$$

where $|\partial y/\partial x|$ means the absolute value of $\partial y/\partial x$. The entropy of the transformed variable, $H(y)$, is expressed by the following formulla:

$$H(y) = -E[\ln f_y(y)] = -\int_{-\infty}^{\infty} f_y(y) \ln f_y(y) dy \tag{5.2}$$

where $E[.]$ is the expectation operator. Substituting equation (5.1) into equation (5.2) gives

$$H(y) = -E\left[\ln\left|\frac{\partial y}{\partial x}\right|\right] - E[\ln f_x(x)] \tag{5.3}$$

The term $E[\ln f_x(x)]$ on the right hand side of the equation (5.3) depends only on the input variable x because this quantity is the entropy of x only. Therefore, a change in the parameter w will not affect the term $E[\ln f_x(x)]$. So, we need to maximize the term $E\left[\ln\left|\frac{\partial y}{\partial x}\right|\right]$ with respect to the parameter w in order to maximize the output entropy $H(y)$. We can use the so called "online", stochastic gradient ascent learning rule for this purpose:

$$\Delta w \propto \frac{\partial H}{\partial w} = \frac{\partial}{\partial w}\left(\ln\left|\frac{\partial y}{\partial x}\right|\right) = \left(\frac{\partial y}{\partial x}\right)^{-1}\frac{\partial}{\partial w}\left(\frac{\partial y}{\partial x}\right) \tag{5.4}$$

Multiplying the input by a weight w and adding a bias-weight w_0 to it, i.e., $u = wx + w_0$ we can derive the adaptation rule for the following logistic transfer function:

$$y = \frac{1}{(1 + e^{-u})} \tag{5.5}$$

The term $\frac{\partial y}{\partial x}$ is derived as:

$$\frac{\partial y}{\partial x} = wy(1 - y) \tag{5.6}$$

$$\frac{\partial}{\partial w}\left(\frac{\partial y}{\partial x}\right) = y(1 - y)(1 + wx(1 - 2y)) \tag{5.7}$$

Thus the learning rule for the given logistic function is obtained by dividing equation (5.6) by equation (5.7) and calculated from the general rule of equation (5.4):

$$\Delta w \propto \frac{1}{w} + x(1-2y) \tag{5.8}$$

The learning rule for the bias-weight w_0 can be found in the same way:

$$\Delta w_0 \propto (1-2y) \tag{5.9}$$

5.2 The general N X N Network case

Referring to Fig. 4.1 we first derive the adaptation equations in the time domain using mixtures of two sources where g is the logistic function $y = g(u) = \frac{1}{(1+e^{-u})}$. Following Bell and Sejnowski [69], we can minimize the mutual information between outputs y_1 and y_2 by maximizing the entropy at the output, which is equal to maximizing $E[\ln|J|]$.

5.2.1 Two sources and two sensors case

There are two cases we can consider.

First Case

Assuming causal FIR (Finite Impulse Response)-filters for W_{ij}, the network carries out the following in the time domain

$$u_1(t) = \sum_{k=0}^{L_{11}} w_{1k1} x_1(t-k) + \sum_{k=1}^{L_{12}} w_{1k2} u_2(t-k) \tag{5.10}$$

$$u_2(t) = \sum_{k=0}^{L_{22}} w_{2k2} x_2(t-k) + \sum_{k=1}^{L_{21}} w_{2k1} u_1(t-k) \tag{5.11}$$

Since the direct mixing filters $W_{11}(z)$ and $W_{22}(z)$ are assumed to have unity gain for information maximization criterion, we don't need to learn the update rules for these two demixing filters and we will learn only cross demixing filters $W_{12}(z)$ and $W_{21}(z)$. In that case equations (5.10) and (5.11) can be rewritten as follows:

$$u_1(t) = x_1(t) + \sum_{k=1}^{L_{12}} w_{1k2} u_2(t-k)$$
$$u_2(t) = x_2(t) + \sum_{k=1}^{L_{21}} w_{2k1} u_1(t-k)$$

The multivariate probabilty density function of $\mathbf{y}$ can be written [69]:

$$f_{\mathbf{y}}(\mathbf{y}) = \frac{f_{\mathbf{x}}(\mathbf{x})}{|J|} \tag{5.12}$$

where $|J|$ is the absolute value of the Jacobian of the transformation. The Jacobian is the determinant of the matrix of partial derivatives:

$$J = \det \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \cdots\cdots & \cdots\cdots & \frac{\partial y_1}{\partial x_n} \\ \frac{\partial y_2}{\partial x_1} & \cdots\cdots & \cdots\cdots & \frac{\partial y_2}{\partial x_n} \\ \cdots\cdots & \cdots\cdots & \cdots\cdots & \cdots\cdots \\ \cdots\cdots & \cdots\cdots & \cdots\cdots & \cdots\cdots \\ \frac{\partial y_n}{\partial x_1} & \cdots\cdots & \cdots\cdots & \frac{\partial y_n}{\partial x_2} \end{bmatrix} \tag{5.13}$$

in our present case we assume n is equal to 2. In that case $|J|$ becomes: The derivation proceeds as in the previous section instead of maximizing $\ln |\partial y / \partial x|$, we maximize $\ln |J|$. Now,

$$y_1(t) = g(u_1) = \frac{1}{(1+e^{-u_1})} \tag{5.14}$$

$$y_2(t) = g(u_1) = \frac{1}{(1 + e^{-u_2})} \tag{5.15}$$

The determinant of Jacobian matrix for above two equations is

$$J = \det \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_n} \end{bmatrix} \tag{5.16}$$

$$|J| = \frac{\partial y_1}{\partial x_1}\frac{\partial y_2}{\partial x_2} - \frac{\partial y_1}{\partial x_2}\frac{\partial y_1}{\partial x_2} \tag{5.17}$$

where,

$$\frac{\partial y_1}{\partial x_1} = \frac{\partial y_1}{\partial u_1}\frac{\partial u_1}{\partial x_1} \tag{5.18}$$

$$\frac{\partial y_1}{\partial x_2} = \frac{\partial y_1}{\partial u_1}\frac{\partial u_1}{\partial x_2} \tag{5.19}$$

$$\frac{\partial y_2}{\partial x_1} = \frac{\partial y_2}{\partial u_2}\frac{\partial u_2}{\partial x_1} \tag{5.20}$$

$$\frac{\partial y_2}{\partial x_2} = \frac{\partial y_2}{\partial u_2}\frac{\partial u_2}{\partial x_2} \tag{5.21}$$

From equation (5.10) and equation (5.11) we get for $k = 1$

$$u_1(t) = x_1(t) + w_{112}u_2(t-1) \tag{5.22}$$

$$u_2(t) = x_2(t) + w_{211}u_1(t-1) \tag{5.23}$$

From equation (5.22) and (5.23) we get

$$\frac{\partial u_1}{\partial x_1} = 1, \frac{\partial u_1}{\partial x_2} = 0 \tag{5.24}$$

$$\frac{\partial u_2}{\partial x_2} = 1, \frac{\partial u_2}{\partial x_1} = 0 \tag{5.25}$$

where,

$$\frac{\partial y_1}{\partial u_1} = y_1{}' \text{ and } \frac{\partial y_2}{\partial u_2} = y_2{}'$$

Putting the values from equations (5.24) and (5.25) in equation (5.17) we get following equation:

$$|J| = y_1{}' y_2{}' \tag{5.26}$$

Taking logarithm on both sides of equation (5.26) we have

$$\ln |J| = \ln y_1{}' + \ln y_2{}' \tag{5.27}$$

Now, taking partial derivatives of $\ln |J|$ with respect to w_{112} and w_{211} we get

$$\Delta w_{112} \propto \frac{\partial \ln |J|}{\partial w_{112}} = \frac{1}{y_1{}'} \frac{\partial y_1{}'}{\partial w_{112}} + \frac{1}{y_2{}'} \frac{\partial y_2{}'}{\partial w_{112}} \tag{5.28}$$

$$\Delta w_{211} \propto \frac{\partial \ln |J|}{\partial w_{211}} = \frac{1}{y_1{}'} \frac{\partial y_1{}'}{\partial w_{211}} + \frac{1}{y_2{}'} \frac{\partial y_2{}'}{\partial w_{211}} \tag{5.29}$$

For the logistic function $\frac{\partial y_i{}'}{\partial y_i} = 1 - 2y_i$. Thus we can write for the partial derivatives for w_{112} and w_{211} :

$$\begin{aligned} \frac{\partial y_1{}'}{\partial w_{112}} = \frac{\partial y_1{}'}{\partial y_1} \frac{\partial y_1}{\partial u_1} \frac{\partial u_1}{\partial w_{112}} &= (1 - 2y_1) y_1{}' u_2(t-1), \\ \frac{\partial y_2{}'}{\partial w_{112}} = \frac{\partial y_2{}'}{\partial y_2} \frac{\partial y_2}{\partial u_2} \frac{\partial u_2}{\partial w_{112}} &= 0, \\ \frac{\partial y_1{}'}{\partial w_{211}} = \frac{\partial y_1{}'}{\partial y_1} \frac{\partial y_1}{\partial u_1} \frac{\partial u_1}{\partial w_{211}} &= 0, \end{aligned} \tag{5.30}$$

$$\frac{\partial y_2{}'}{\partial w_{211}} = \frac{\partial y_2{}'}{\partial y_2} \frac{\partial y_2}{\partial u_2} \frac{\partial u_2}{\partial w_{211}} = (1 - 2y_2) y_2{}' u_1(t-1), \tag{5.31}$$

$$\tag{5.32}$$

Thus the adaptation rules for two demixing filter coefficients for the first delay become

$$\Delta w_{112} \propto (1 - 2y_1)u_2(t-1)$$
$$\Delta w_{211} \propto (1 - 2y_2)u_1(t-1) \tag{5.33}$$

In general we can write the adaptation rule for demixing filter coefficients as follows:

$$\Delta w_{ikj} \propto (1 - 2y_i)u_j(t-k) \tag{5.34}$$

Second Case

Assuming IIR (Infinite Impulse Response)-filters for W_{ij}, for the network of Fig. 4.1 we can write the following equations in z-domain

$$U_1(z) = U_{X1}(z) + U_{U2}(z) \tag{5.35}$$
$$U_2(z) = U_{X2}(z) + U_{U1}(z) \tag{5.36}$$

In terms of demixing coefficients we can write

$$U_1(z) = W_{11}(z)X_1(z) + W_{12}(z)U_2(z) \tag{5.37}$$
$$U_2(z) = W_{22}(z)X_2(z) + W_{21}(z)U_1(z) \tag{5.38}$$

where $W_{11}(z)$ and $W_{22}(z)$ have unity gain and $W_{12}(z)$ and $W_{21}(z)$ have the following IIR (Infinite Impulse Response) structure

$$W_{12}(z) = \frac{\sum_{k=0}^{L_{12}} b_{1k2}z^{-k}}{1 - \sum_{k=1}^{L_{12}} a_{1k2}z^{-k}} \tag{5.39}$$
$$W_{21}(z) = \frac{\sum_{k=0}^{L_{21}} b_{2k1}z^{-k}}{1 - \sum_{k=1}^{L_{21}} a_{2k1}z^{-k}} \tag{5.40}$$

Using the above two values of $W_{12}(z)$ and $W_{21}(z)$ we can write $U_{U2}(z)$ and $U_{U1}(z)$ as shown below

$$U_{U2}(z) = \frac{\sum_{k=0}^{L_{12}} b_{1k2} z^{-k}}{1 - \sum_{k=1}^{L_{12}} a_{1k2} z^{-k}} U_2(z) \tag{5.41}$$

$$U_{U1}(z) = \frac{\sum_{k=0}^{L_{21}} b_{2k1} z^{-k}}{1 - \sum_{k=1}^{L_{21}} a_{2k1} z^{-k}} U_1(z) \tag{5.42}$$

In the time domain we can write

$$u_{u2}(t) = \sum_{k=1}^{L_{12}} a_{1k2} u_{u2}(t-k) + \sum_{k=0}^{L_{12}} b_{1k2} u_2(t-k) \tag{5.43}$$

$$u_{u1}(t) = \sum_{k=1}^{L_{21}} a_{2k1} u_{u1}(t-k) + \sum_{k=0}^{L_{12}} b_{2k1} u_1(t-k) \tag{5.44}$$

Now, in time domain we can write equations for $u_1(t)$ and $u_2(t)$ as

$$u_1(t) = x_1(t) + \sum_{k=1}^{L_{12}} a_{1k2} u_{u2}(t-k) + \sum_{k=0}^{L_{12}} b_{1k2} u_2(t-k) \tag{5.45}$$

$$u_2(t) = x_2(t) + \sum_{k=1}^{L_{21}} a_{2k1} u_{u1}(t-k) + \sum_{k=0}^{L_{12}} b_{2k1} u_1(t-k) \tag{5.46}$$

Following the steps used in the first case (FIR (Finite Impulse Response) case) we can derive the adaptation rules for demixing coefficients a_{1k2}, a_{2k1}, b_{1k2} and b_{1k2}. As we have done before the determinant of the Jacobian is the determinant of the matrix of partial derivatives:

$$\det J = \det \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \cdots & \cdots & \frac{\partial y_1}{\partial x_n} \\ \frac{\partial y_2}{\partial x_1} & \cdots & \cdots & \frac{\partial y_2}{\partial x_n} \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial y_n}{\partial x_1} & \cdots & \cdots & \frac{\partial y_n}{\partial x_n} \end{bmatrix} \tag{5.47}$$

For simplicity we assume n is equal to 2. Now,

$$y_1(t) = g(u_1) = \frac{1}{(1+e^{-u_1})}$$
$$y_2(t) = g(u_1) = \frac{1}{(1+e^{-u_2})} \tag{5.48}$$

The determinant of the Jacobian matrix for the above two equations is

$$\det J = \det \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} \end{bmatrix} \tag{5.49}$$

$$|J| = \frac{\partial y_1}{\partial x_1}\frac{\partial y_2}{\partial x_2} - \frac{\partial y_1}{\partial x_2}\frac{\partial y_1}{\partial x_2} \tag{5.50}$$

when,

$$\frac{\partial y_1}{\partial x_1} = \frac{\partial y_1}{\partial u_1}\frac{\partial u_1}{\partial x_1} \tag{5.51}$$
$$\frac{\partial y_1}{\partial x_2} = \frac{\partial y_1}{\partial u_1}\frac{\partial u_1}{\partial x_2} \tag{5.52}$$
$$\frac{\partial y_2}{\partial x_1} = \frac{\partial y_2}{\partial u_2}\frac{\partial u_2}{\partial x_1} \tag{5.53}$$
$$\frac{\partial y_2}{\partial x_2} = \frac{\partial y_2}{\partial u_2}\frac{\partial u_2}{\partial x_2} \tag{5.54}$$

There is no zero delay coefficients for a_{1k2}, a_{2k1}. When $k = 1$ equation (5.45) and (5.46) become

$$u_1(t) = x_1(t) + a_{112}u_{u2}(t-1) + b_{112}u_2(t-1) \tag{5.55}$$
$$u_2(t) = x_2(t) + a_{211}u_{u2}(t-1) + b_{211}u_2(t-1) \tag{5.56}$$

From equations (5.24) and (5.25) we get

$$\frac{\partial u_1}{\partial x_1} = 1, \frac{\partial u_1}{\partial x_2} = 0 \tag{5.57}$$

$$\frac{\partial u_2}{\partial x_2} = 1, \frac{\partial u_2}{\partial x_1} = 0 \tag{5.58}$$

Let, $\frac{\partial y_1}{\partial u_1} = {y_1}'$ and $\frac{\partial y_2}{\partial u_2} = {y_2}'$

Then the equations become

$$\frac{\partial y_1}{\partial x_1} = {y_1}' \tag{5.59}$$

$$\frac{\partial y_1}{\partial x_2} = 0 \tag{5.60}$$

$$\frac{\partial y_2}{\partial x_1} = 0 \tag{5.61}$$

$$\frac{\partial y_2}{\partial x_2} = {y_2}' \tag{5.62}$$

Putting the above two values we get the new equation for determinant of Jacobian matrix which is

$$|J| = {y_1}'{y_2}' \tag{5.63}$$

Taking the logarithm on both sides

$$\ln |J| = \ln {y_1}' + \ln {y_2}' \tag{5.64}$$

Taking partial derivative of $\ln |J|$ with respect to a_{112} we get

$$\frac{\partial \ln |J|}{\partial a_{112}} = \frac{1}{{y_1}'}\frac{\partial {y_1}'}{\partial a_{112}} + \frac{1}{{y_2}'}\frac{\partial {y_2}'}{\partial a_{112}} \tag{5.65}$$

Like as in the FIR (Finite Impulse Response) case for the logistic function we have $\frac{\partial y_i{}'}{\partial y_i} = 1 - 2y_i$. Thus we can write for the partial derivatives for a_{112}:

$$\frac{\partial y_1{}'}{\partial a_{112}} = \frac{\partial y_1{}'}{\partial y_1}\frac{\partial y_1}{\partial u_1}\frac{\partial u_1}{\partial a_{112}} \tag{5.66}$$

$$= (1 - 2y_1)y_1{}'u_{u2}(t-1), \tag{5.67}$$

$$\frac{\partial y_2{}'}{\partial a_{112}} = \frac{\partial y_2{}'}{\partial y_2}\frac{\partial y_2}{\partial u_2}\frac{\partial u_2}{\partial a_{112}} \tag{5.68}$$

$$= (1 - 2y_2)y_2{}'u_{u2}.0 = 0, \tag{5.69}$$

The adaptation rule for a_{112} becomes

$$\Delta a_{112} \propto \frac{\partial \ln |J|}{\partial a_{112}} = \frac{1}{y_1{}'}\frac{\partial y_1{}'}{\partial a_{112}} \tag{5.70}$$

$$= \frac{1}{y_1{}'}\frac{\partial y_1{}'}{\partial y_1}\frac{\partial y_1}{\partial u_1}\frac{\partial u_1}{\partial a_{112}}$$

$$= \frac{1}{y_1{}'}(1 - 2y_1)y_1{}'u_{u2}(t-1)$$

$$= (1 - 2y_1)u_{u2}(t-1) \tag{5.71}$$

Using the same steps as for a_{112} we can derive the adaptation rule for a_{211}. Taking partial derivative of $\ln |J|$ with respect to a_{211} we have

$$\frac{\partial \ln |J|}{\partial a_{211}} = \frac{1}{y_1{}'}\frac{\partial y_1{}'}{\partial a_{211}} + \frac{1}{y_2{}'}\frac{\partial y_2{}'}{\partial a_{211}} \tag{5.72}$$

Thus we can write for the partial derivatives for a_{211}:

$$\frac{\partial y_1{}'}{\partial a_{211}} = \frac{\partial y_1{}'}{\partial y_1}\frac{\partial y_1}{\partial u_1}\frac{\partial u_1}{\partial a_{211}} = (1 - 2y_1)y_1{}'.0 = 0 \tag{5.73}$$

$$\frac{\partial y_2{}'}{\partial a_{211}} = \frac{\partial y_2{}'}{\partial y_2}\frac{\partial y_2}{\partial u_2}\frac{\partial u_2}{\partial a_{211}} = (1 - 2y_2)y_2{}'u_{u1}(t-1), \tag{5.74}$$

The adaptation rule for a_{211} becomes

$$\Delta a_{211} \propto \frac{\partial \ln |J|}{\partial a_{211}} = \frac{1}{y_2'} \frac{\partial y_2'}{\partial a_{211}} \tag{5.75}$$
$$= \frac{1}{y_2'} \frac{\partial y_2'}{\partial y_2} \frac{\partial y_2}{\partial u_2} \frac{\partial u_2}{\partial a_{211}}$$
$$= (1 - 2y_2) u_{u1}(t-1)$$

Using the steps used for a_{112} and a_{211} we can prove that the adaptation rules for a_{122} and a_{221}, i.e., when the delay index $k = 2$ are as follows:

$$\Delta a_{122} \propto (1 - 2y_1) u_{u2}(t-2) \tag{5.76}$$
$$\Delta a_{221} \propto (1 - 2y_2) u_{u1}(t-2) \tag{5.77}$$

In general for any delay k the adaptation rules for a_{ikj} will be

$$\Delta a_{ikj} \propto (1 - 2y_i) u_{uj}(t-k) \tag{5.78}$$

where, $i = 1, ..., N$ and where $j = 1, ..., N$. N is the number of sources or the number of sensors assuming that the number of sources is equal to the number of sensors.

5.2.2 Three sources and three sensors case

First case

Assuming causal FIR (Finite Impulse Response)-filters for the W_{ij} of the network yields the following dynamical relationships

$$\begin{aligned} u_1(t) &= x_1(t) + \sum_{k=1}^{L_{12}} w_{1k2} u_2(t-k) + \sum_{k=1}^{L_{13}} w_{1k3} u_3(t-k) \\ u_2(t) &= x_2(t) + \sum_{k=1}^{L_{21}} w_{2k1} u_1(t-k) + \sum_{k=1}^{L_{23}} w_{2k3} u_3(t-k) \\ u_3(t) &= x_3(t) + \sum_{k=1}^{L_{31}} w_{3k1} u_1(t-k) + \sum_{k=1}^{L_{32}} w_{3k2} u_2(t-k) \end{aligned} \tag{5.79}$$

The multivariate probabilty density function of y can be written as

$$f_{\mathbf{y}}(\mathbf{y}) = \frac{f_{\mathbf{x}}(\mathbf{x})}{|J|} \tag{5.80}$$

where $|J|$ is the absolute value of the Jacobian of the transformation. Now,

$$\begin{aligned} y_1(t) &= g(u_1) = \frac{1}{(1+e^{-u_1})} \\ y_2(t) &= g(u_2) = \frac{1}{(1+e^{-u_2})} \\ y_3(t) &= g(u_3) = \frac{1}{(1+e^{-u_3})} \end{aligned} \tag{5.81}$$

By taking the logarithm of the determinant of the Jacobian matrix and then partial derivatives of it with respect to the demixing coefficients, the adaptation laws for w_{112}, w_{113}, w_{211}, w_{213}, w_{311} and w_{312} may be developed as follows:

$$\begin{aligned} &\Delta w_{112} \propto (1-2y_1)u_2(t-1), \Delta w_{113} \propto (1-2y_1)u_3(t-1) \\ &\Delta w_{211} \propto (1-2y_2)u_1(t-1), \Delta w_{213} \propto (1-2y_2)u_3(t-1) \\ &\Delta w_{311} \propto (1-2y_3)u_1(t-1), \Delta w_{312} \propto (1-2y_3)u_2(t-1) \end{aligned} \tag{5.82}$$

In general, the adaptation law for the cross filter coefficients is given by

$$\Delta w_{ikj} \propto (1-2y_i)u_j(t-k) \tag{5.83}$$

Second case

Assuming IIR (Infinite Impulse Response)-filters for the W_{ij}'s, the network shown in Figure 4.1 can now be represented formally according to the following equations corresponding to three sources

$$\begin{aligned} U_1(z) &= U_{X1}(z) + U_{U12}(z) + U_{U13}(z) \\ U_2(z) &= U_{X2}(z) + U_{U21}(z) + U_{U23}(z) \\ U_3(z) &= U_{X3}(z) + U_{U31}(z) + U_{U32}(z) \end{aligned} \tag{5.84}$$

where in terms of the demixing coefficients it can be written as

$$\begin{aligned} U_1(z) &= W_{11}(z)X_1(z) + W_{12}(z)U_{12}(z) + W_{13}(z)U_{13}(z) \\ U_2(z) &= W_{21}(z)U_{21}(z) + W_{22}(z)X_2(z) + W_{23}(z)U_{23}(z) \\ U_3(z) &= W_{31}(z)U_{31}(z) + W_{32}(z)U_{32}(z) + W_{33}(z)X_3(z) \end{aligned} \tag{5.85}$$

The filters $W_{11}(z)$, $W_{22}(z)$ and $W_{33}(z)$ have unity gains and $W_{12}(z)$, $W_{13}(z)$, $W_{21}(z)$, $W_{23}(z)$, $W_{31}(z)$ and $W_{32}(z)$ have following IIR (Infinite Impulse Response) structure

$$\begin{aligned} W_{1j}(z) &= \frac{\sum_{k=0}^{L_{1j}} b_{1kj}z^{-k}}{1-\sum_{k=1}^{L_{1j}} a_{1kj}z^{-k}}, \quad j=2,3 \\ W_{2j}(z) &= \frac{\sum_{k=0}^{L_{2j}} b_{2kj}z^{-k}}{1-\sum_{k=1}^{L_{2j}} a_{2kj}z^{-k}}, \quad j=1,3 \\ W_{3j}(z) &= \frac{\sum_{k=0}^{L_{3j}} b_{3kj}z^{-k}}{1-\sum_{k=1}^{L_{3j}} a_{3kj}z^{-k}}, \quad j=1,2 \end{aligned} \tag{5.86}$$

Using equation (5.86) we can obtain equations for $u_1(t)$, $u_2(t)$ and $u_3(t)$ in the time domain as

$$\begin{aligned} u_1(t) = x_1(t) &+ \sum_{k=1}^{L_{12}} a_{1k2}u_{u12}(t-k) + \sum_{k=0}^{L_{13}} b_{1k2}u_2(t-k) \\ &+ \sum_{k=1}^{L_{13}} a_{1k3}u_{u13}(t-k) + \sum_{k=0}^{L_{13}} b_{1k3}u_3(t-k) \\ u_2(t) = x_2(t) &+ \sum_{k=1}^{L_{21}} a_{2k1}u_{u21}(t-k) + \sum_{k=0}^{L_{21}} b_{2k1}u_1(t-k) \\ &+ \sum_{k=1}^{L_{23}} a_{2k3}u_{u23}(t-k) + \sum_{k=0}^{L_{23}} b_{2k3}u_3(t-k) \\ u_3(t) = x_3(t) &+ \sum_{k=1}^{L_{31}} a_{3k1}u_{u31}(t-k) + \sum_{k=0}^{L_{31}} b_{3k1}u_1(t-k) \\ &+ \sum_{k=1}^{L_{32}} a_{3k2}u_{u32}(t-k) + \sum_{k=0}^{L_{32}} b_{3k2}u_2(t-k) \end{aligned} \tag{5.87}$$

Following the steps used in case the FIR (Finite Impulse Response) case we can derive the adaptation laws for the demixing coefficients a_{ikj} and b_{ikj}. Since there are no zero delay coefficients for a_{ikj} we can begin with $k = 1$. Under this condition, the equation (5.87) becomes

$$\begin{aligned} u_1(t) = \; & x_1(t) + a_{112}u_{u12}(t-1) + b_{112}u_2(t-1) + \\ & a_{113}u_{u13}(t-1) + b_{113}u_3(t-1) \\ u_2(t) = \; & x_2(t) + a_{211}u_{u21}(t-1) + b_{211}u_1(t-1) + \\ & a_{213}u_{u23}(t-1) + b_{213}u_3(t-1) \\ u_3(t) = \; & x_3(t) + a_{311}u_{u31}(t-1) + b_{311}u_1(t-1) + \\ & a_{312}u_{u32}(t-1) + b_{312}u_2(t-1) \end{aligned} \tag{5.88}$$

The adaptation rules for a_{112}, a_{211},a_{213}, a_{312}, a_{113} and a_{311} can be shown to be governed by

$$\begin{aligned} \Delta a_{112} &\propto \frac{\partial \ln |J|}{\partial a_{112}} = (1-2y_1)u_{u12}(t-1) \\ \Delta a_{211} &\propto \frac{\partial \ln |J|}{\partial a_{211}} = (1-2y_2)u_{u21}(t-1) \\ \Delta a_{213} &\propto \frac{\partial \ln |J|}{\partial a_{213}} = (1-2y_2)u_{u23}(t-1) \\ \Delta a_{312} &\propto \frac{\partial \ln |J|}{\partial a_{312}} = (1-2y_3)u_{u32}(t-1) \\ \Delta a_{113} &\propto \frac{\partial \ln |J|}{\partial a_{113}} = (1-2y_1)u_{u13}(t-1) \\ \Delta a_{311} &\propto \frac{\partial \ln |J|}{\partial a_{311}} = (1-2y_3)u_{u31}(t-1) \end{aligned} \tag{5.89}$$

In general, for the k^{th} delay term the adaptation rules for a_{ikj} becomes

$$\Delta a_{ikj} \propto (1-2y_i)u_{u_{ij}}(t-k) \tag{5.90}$$

where, $i = 1, ..., N$, $j = 1, ..., N$, and N is the number of sources or sensors (assuming that the number of sources is equal to the number of sensors). The derivation of the adaptation rules for b_{ikj} follows along the same line and are given by

$$\Delta b_{ikj} \propto (1 - 2y_i)u_j(t - k) \tag{5.91}$$

5.3 Conclusions

In this chapter we have presented the adaptation rules for the demixing coefficients of $N \times N$ feedback network architecture shown in chapter 4. At first we derived the adaptive rules for two sources and two sensors case and afterwards we have generalized the adaptive rules for $N \times N$ architecture. In both cases we have used FIR (Finite Impulse Response) and IIR (Infinite Impulse Response) architectures for demixing filters.

Chapter 6

Simulation results and discussion

In this chapter, we present our simulation results for the two cases that were considered in the previous section.

6.1 Two sources and two sensors case

Two sinusoidal signals each composed of two different frequencies are synthetically generated. The frequencies in the first signal are 50 Hz and 75 Hz; and the frequencies contained in the second signal are 50 Hz and 100 Hz. The Signal to Noise Ratio (SNR) is set to 15 dB. These signals are then convolved with mixing filters that represent the environmental factors. Our objective is to determine the effects of variations in the length of the mixing filters on the quality of the recovered signals. A total of 4000 data points are generated from the two signals and used in the mini-batch learning system to learn the demixing filter coefficients. In every batch we passed 50 data points each time and the total number of windows used was 79 for the complete data set. The results presented below are obtained after 100 epochs through the entire data set and correspond to what was obtained from the last window.

The simulation results are conducted for two specific scenarios, namely

(i) W_{12} and W_{21} are FIR (Finite Impulse Response) filters so that from equation (4.11)

we can make $W_{12}(z)$ FIR (Finite Impulse Response) filters in three ways:

1. $A_{11}(z)$ and $A_{22}(z)$ have unit gains and $A_{12}(z)$ and $A_{21}(z)$ are FIR (Finite Impulse Response) filters,

2. $A_{11}(z)$ and $A_{22}(z)$ are IIR (Infinite Impulse Response) filters and $A_{12}(z)$ and $A_{21}(z)$ have unit gains, and

3. $A_{11}(z)$ and $A_{22}(z)$ are IIR (Infinite Impulse Response) filters and $A_{12}(z)$ and $A_{21}(z)$ are FIR (Finite Impulse Response) filters.

(ii) W_{12} and W_{21} are IIR (Infinite Impulse Response) filters. The results obtained are discussed below.

(i1): As shown in Table 6.1, and as expected, with the increase in the order of the mixing filters the correlation factor decreases and the mean square error increases except the first one which was not desired. Clearly the higher the filter order the more the source signals suffer from the environmental distortions. The correlation is highest and the mean square is the minimum when the filter length of the mixing filter is set to 6. The original source signals, convolved signals at the sensors, the separated signals, the mixing filters and, the learned coefficients are shown in Figures 6.2, 6.1 and 6.3.

Type of demixing Filter	Order of mixing filter				Correlation		Mean Square Error	
	A11	A22	A12	A21	S1, U1	S2, U2	S1, U1	S2, U2
Case 1	Unity gain	Unity gain	2	2	0.81	0.97	0.4	0.41
	Unity gain	Unity gain	2	4	0.96	0.99	0.1	0.07
	Unity gain	Unity gain	4	4	0.84	0.94	0.37	0.29
	Unity gain	Unity gain	6	4	0.69	0.92	0.67	0.5
Case 2	2	2	Unity gain	Unity gain	0.94	0.93	0.48	0.45
	2	4	Unity gain	Unity gain	0.94	0.95	0.0001	0
	4	4	Unity gain	Unity gain	0.83	0.66	0.0003	0.0001
Case 3	2	2	2	2	0.51	0.64	1.63	1.0

Table 6.1: The summary of the results for the cases (i1)-(i3) when W_{12} and W_{21} are FIR (Finite Impulse Response) filters.

(i2): In the second case we varied the filter length in the same way as we did in the FIR (Finite Impulse Response)st case. From Table 6.1 we can see that with the increase in the order of the two mixing filters, the correlation between the original source signals and the recovered signals decreases. In this case, the first result is much better than that in the first case. The highest correlations are 0.94 and 0.95. There is no significant difference when the total filter length was increased from 4 to 6 in contrast to what has happened in the first case. The correlation between the first source and the first recovered signal was changed drastically from 0.81 to 0.96. The correlation is decreasing considerably in both cases when the combined filter length is exceeding 6. Also, us-

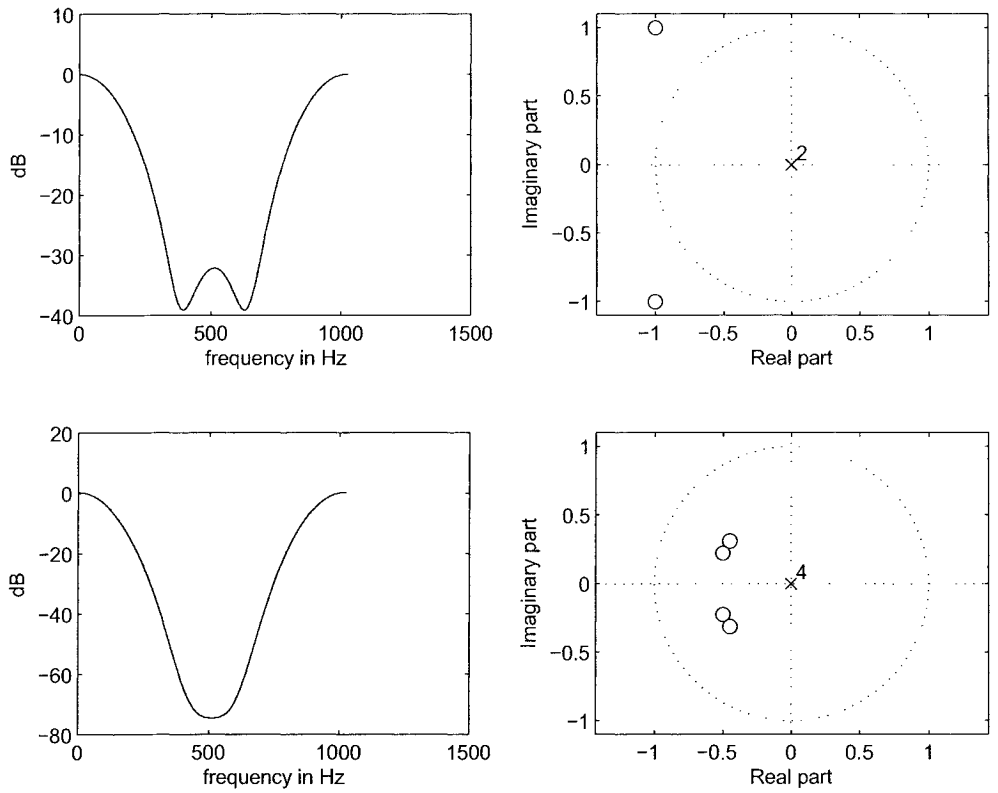

Figure 6.1: Pole-zero locations and magnitude response of A_{12} and A_{21}

.

ing the same total filter length we don't obtain the same result in two cases implying that the distribution of mixing filters have an effect on the quality of the recovered signals. We used unity gain in the direct filters in case (i1) and in cross filters in case (i2). Therefore, different kind of distribution of mixing filters are producing different results having the same total mixing filter lengths. The separated signals, the mixing filters and, the learned coefficients are shown in Figures 6.5, 6.7 and 6.6.

(i3): In this case no filter was used with unity gain. We have used two FIR (Finite Impulse Response) and IIR (Infinite Impulse Response) filters implying that the environmental distortion is significant. Our proposed method was not able to separate the signals completely and as a result the recovered signals are weakly correlated to the source signals. Therefore, the distribution of the mixing filters used to convolve

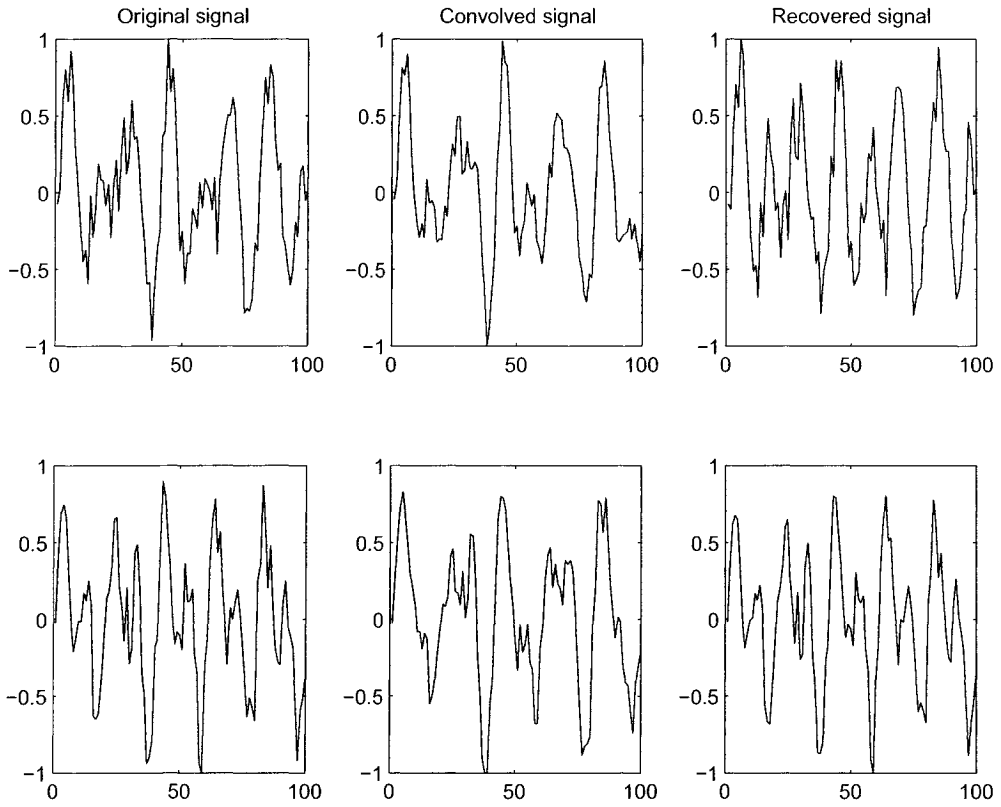

Figure 6.2: The separated signals corresponding to case (i1), second scenario.

the source signals is of utmost importance in the performance of the source separation problem using the proposed methodology.

(ii): The demixing coefficients are adjusted using the adaptation laws given by equations (5.78). In this case the recovered signals have correlations of 0.91 and 0.85. The recovered signals, the pole-zero locations of the four filters and the learned coefficients are shown in Figures 6.8, 6.7 and 6.9. The four filters are selected to be IIR (Infinite Impulse Response). The complexity of the system implementation was increased considerably by using this structure. The combined mixing filter length is set to 11. We can state that with this higher filter length the obtained results are better, although the second signal is less correlated to its respective source signal implying that it is not separated completely.

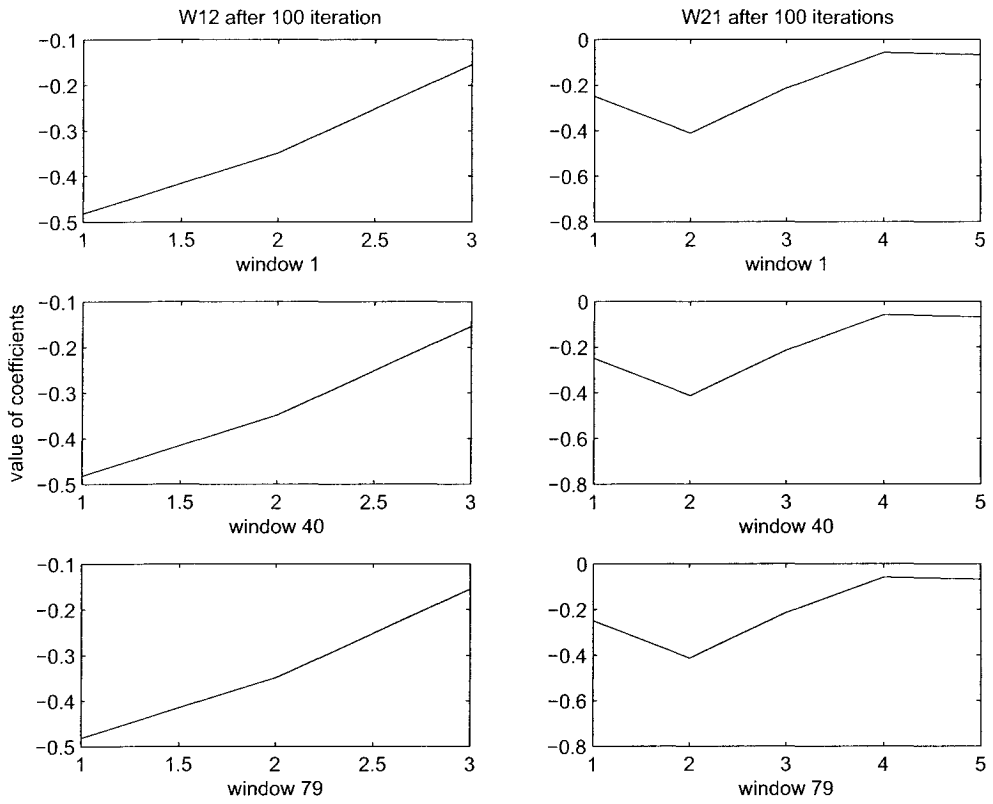

Figure 6.3: W_{12} and W_{21} from different windows after 100 iteration

6.2 Three sources and three sensors

The same information maximization approach was also utilized for three sources and three sensors case. We synthetically generated three sinusoidal signals each composed of two different frequencies. The frequencies in the first signal are 50 Hz and 150 Hz; and the frequencies contained in the second signal are 50 Hz and 120 Hz and the third signal contains 50 Hz and 75 Hz frequencies. The Signal to Noise Ratio (SNR) is set to 15 dB as in previous case. We used nine mixing environmental filters for convolution of signals. For each cross mixing filter the constant dc gain was maintained below 0 dB. Only the direct mixing filters have unity gain. It has been done to ensure that the signal strength coming through the direct mixing filter at each sensor is higher than the combined signal strength coming through the other two cross mixing filters at the same sensors. We generated a total of 4000 data points from the three signals and used then

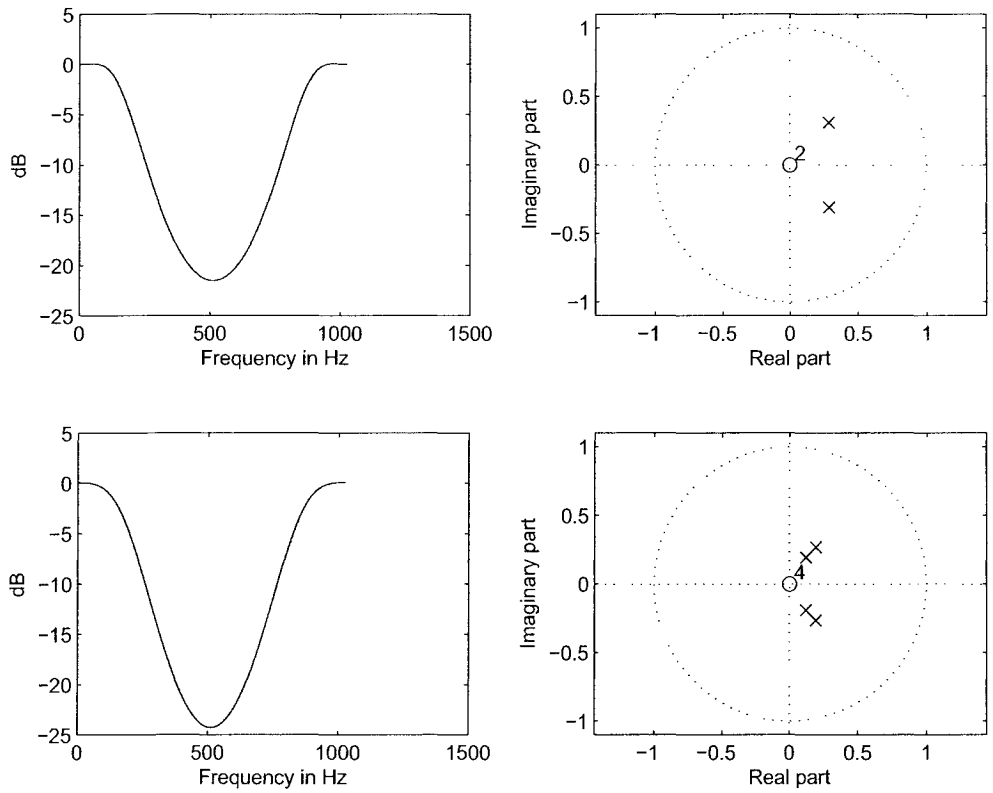

Figure 6.4: Pole-zero location and magnitude response of A_{11} and A_{22}.

the mini-batch learning system to learn the demixing filter coefficients as in the the two sources and two sensors case. In every batch we passed 50 data points each time and the total number of windows used was 79 for the complete data set. The results presented below are obtained after 100 epochs through the entire data set and correspond to what was obtained from the last window.

Since the ideal solution for 3 sources and 3 sensors has been generalized for any arbitrary number N (see equation (4.49)) the special case which is direct mixing filters which have always unity gain, the simulation results are conducted for only one specific scenarios.

(i) W_{12}, W_{13}, W_{21}, W_{23}, W_{31} and W_{32} are FIR (Finite Impulse Response) filters and $A_{11}(z)$, $A_{22}(z)$ and $A_{33}(z)$ have unit gains and $A_{12}(z)$, $A_{13}(z)$, $A_{21}(z)$, $A_{23}(z)$, $A_{31}(z)$ and $A_{32}(z)$ are FIR (Finite Impulse Response) filters.

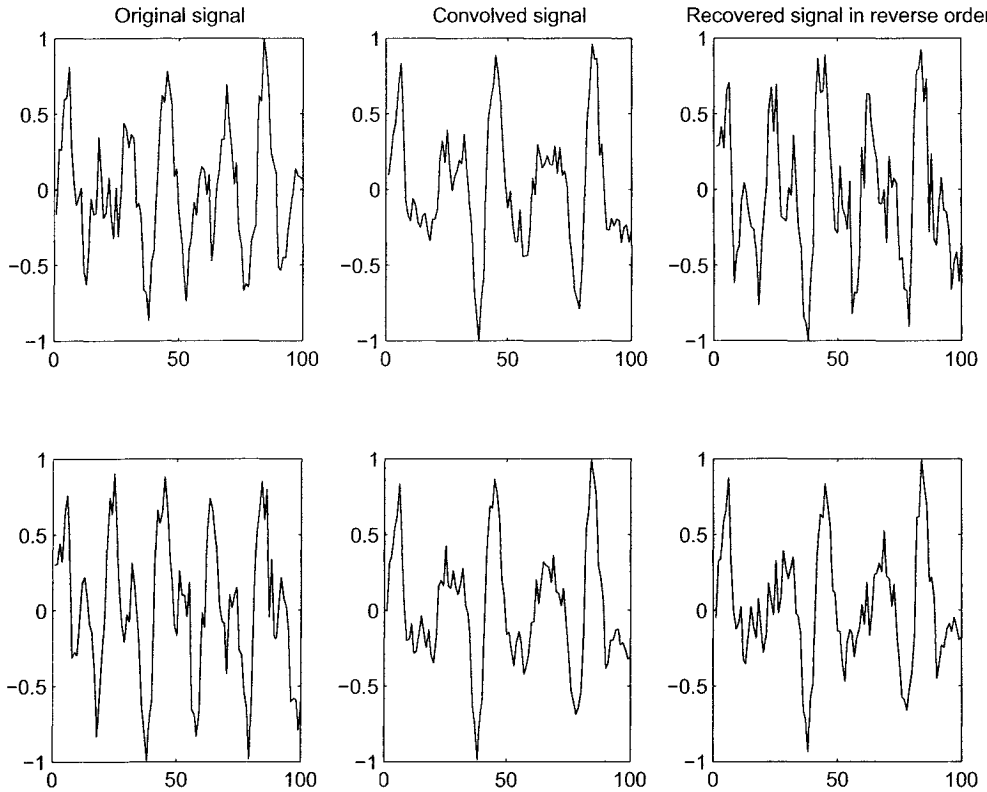

Figure 6.5: The separated signals corresponding to case (i2), second scenario.

We have started from filter length 1 and increased up to the length 9. There is no combined filter length here since according to ideal solution demixing filters depend only on that corresponding mixing filters. Each time we used all mixing filters of the same order. How the variation of the mixing filter length affect the quality of the recovered signals has been presented in tabular form in Table 6.2

In two sources and two sensors case the variations in the quality of signals and their correlation factor with original signals was noticable with the variation of the combined mixing filter length with same mixing filter distribution. But from Table 6.2 we can see the correlation factor didn't decrease gradually as the filter length increases. The reason is that for initialization process of demixing filters we started the initial values quite close to the ideal values and gradually increased the radius (the distance between the ideal value and initial value) of the circle in which the initial value lies. We have found

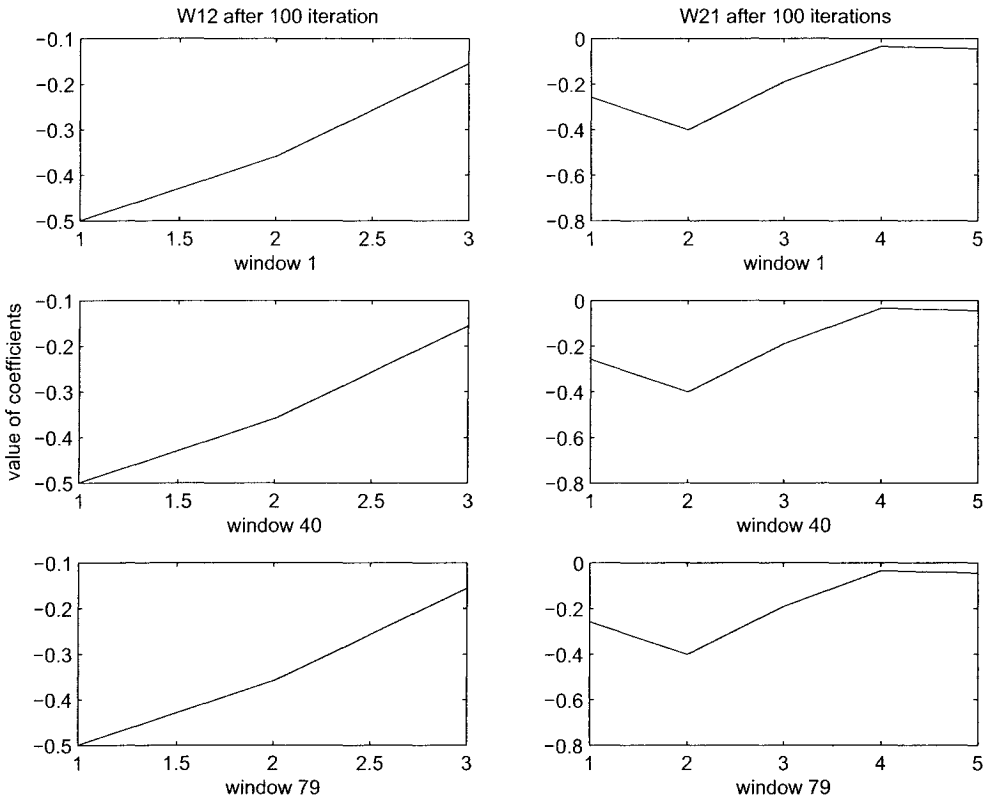

Figure 6.6: W_{12} and W_{21} from different windows after 100 iteration

all initial values for same order of mixing filters didn't give the same result. As the radius increases the correlation factor decreases and means square increases. Finally we have used the initial value which was giving approximately the same result as of the one closest to the ideal solution. The highest correlation we obtained here is 0.98 and the lowest one is 0.88. For the first reccovered signal the correlation factor has been decreased slightly. But for the other two recovered signals some times it is increasing and some times this is decreasing which is not desirable. For all the filter lengths from 1 to 8 we have used the same procedure which was not taken in the two sources and two sensors case. The separated signals, pole zero locations of six mixing filters and learned coefficients are shown in Figure 6.10, 6.11 and 6.12 when the mixing filter length is 4.

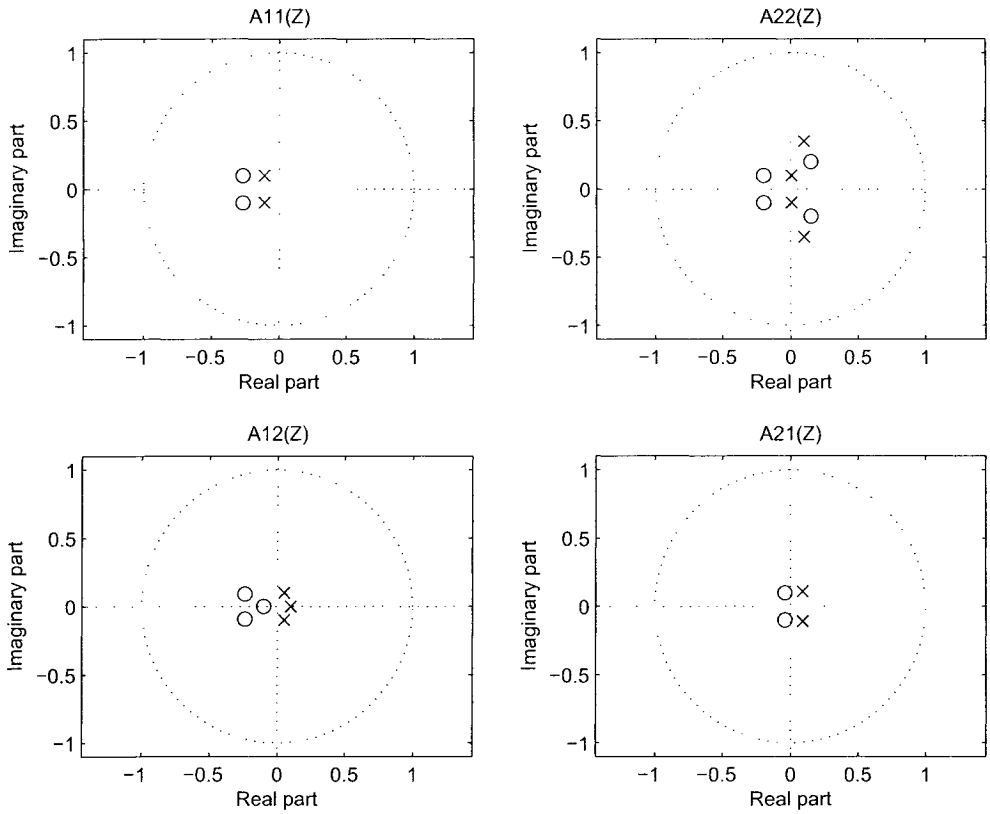

Figure 6.7: Pole-zero locations of A_{11}, A_{22}, A_{12}, and A_{21}.

In closing, we would like to point out some important attributes that seem to affect the performance of the source separation system as follows:

Adaptation rate:

1. Two sources and two sensors case: Although there are a number of adaptive schemes for adjusting the learning rates we didn't get very good results using these methods. Instead a constant adaptation rate was used in each iteration. In most of the cases during the first iteration the learning rate was around 0.00007 and the next 9 iterations it was changed to 0.000001 and for the last 90 iterations it was set to 0.0000001. Using a higher learning rate generally made the system unstable. Our best result (case (i1), second scenario) was obtained using the learning rate of 0.000067 in the first iteration, 0.000001 in the next 9 iterations

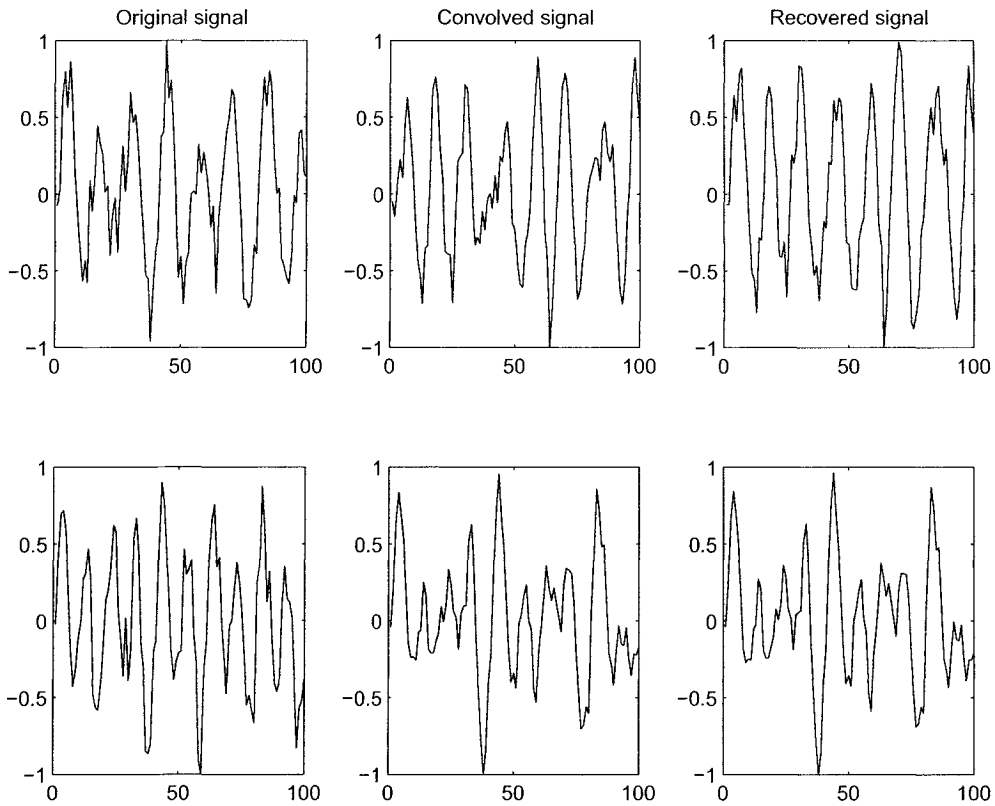

Figure 6.8: The separated signals corresponding to case (ii) when W_{12} and W_{21} are IIR (Infinite Impulse Response) filters.

and for the last 90 iterations it was set to 0.0000001.

2. Three sources and three sensors case: We learned the demixing filter coefficients using the learning rate of 0.00001 for the first 4 iterations, 0.000001 for the next 6 iterations and for the last 90 iterations it was set to 0.0000001.

Locations of poles and zeros: Before applying the learning procedure we need to ensure that the resulting mixing filters will yield stable system. This can be verified easily by using equation (4.11) corresponding to the ideal solution of the general $N \times N$ case. Also, choosing the proper initial values for the demixing filters are important as one has to make sure that the first initial values of the demixing filters also produce stable output. One of the major drawbacks of this approach is that if the initial value of demixing filters are very far from the ideal values (specified according to the ideal

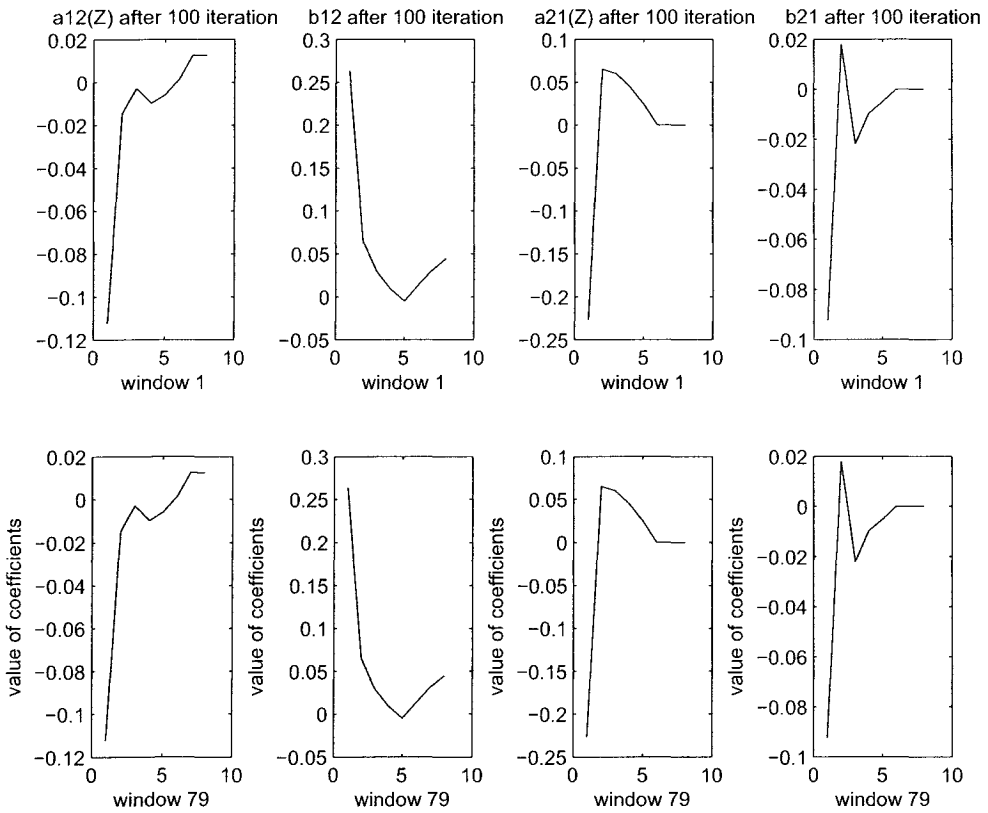

Figure 6.9: learned coefficients of $a_{12}(z)$, $_{a21}(z)$, $b_{12}(z)$ and $b_{21}(z)$

solution), this algorithm may not be able to converge to the right values. Its major advantage and utility is its fast and easy implementation and that one does not need to pre-whiten the observed data.

Effect of SNR:

1. For the generation of synthetic data in all of the cases we set the value of the SNR to 15 dB. By changing the value from 5 dB to 30 dB with an increment by 5 dB each time we verified our results to determine the sensitivity of the results if this 15 dB value is crucial or not. We did not find significant differences in the perfromance of the system. For three sources and three sensors case a change in SNR from 5 dB to 30 dB caused less than 0.5 percentage of change of correlation factor when the filter length was 4.

Order of mixing filter									Correlation			Mean square error		
A11	A22	A33	A12	A13	A21	A23	A31	A32	U1, S1	U2, S2	U3, S3	U1, S1	U2, S2	U3, S3
Unity gain	Unity gain	Unity gain	1	1	1	1	1	1	0.97	0.97	0.91	0.18	0.22	0.51
Unity gain	Unity gain	Unity gain	2	2	2	2	2	2	0.98	0.92	0.93	0.15	0.43	0.5
Unity gain	Unity gain	Unity gain	4	4	4	4	4	4	0.95	0.95	0.95	0.27	0.28	0.31
Unity gain	Unity gain	Unity gain	6	6	6	6	6	6	0.91	0.96	0.90	0.6	0.25	0.58
Unity gain	Unity gain	Unity gain	8	8	8	8	8	8	0.88	0.96	0.92	0.87	0.25	0.44

Table 6.2: The summary of the results when W_{12}, W_{13}, W_{21}, W_{23}, W_{31}, W_{32} are FIR (Finite Impulse Response) filters.

2. At the sensors we always maintained higher signal to noise ratio which is very crucial for the separtion of signals. In our case, for these specific synthetically genrated data this signal to noise ratio must be at least 4.46 dB for successful separation of the signals. Here the signals which are coming through the cross mixing filters are defined as the noises at the sensors and the signal which is coming through the direct mixing filter is defined as signal. If the combined noise strength is higher than the signal strength then it is quite impossible to separate the signals independently.

6.3 Conclusions

In this chapter simulation results have been presented in the tabular form and have been using different pole-zero diagrams and separated signals. In most of the cases we have obtained desired results. There are also some exceptational cases in two sources and two sensors case which were not desired. The correlation factors especially in three sources and three sensors case have high percentages in most of the filter orders because special precaution have been taken in this case. The performance of the system is almost unaffected by the SNR variation. But the signal to noise ratio at the sensors have severe effect on the stability as well as the performance of the whole system.

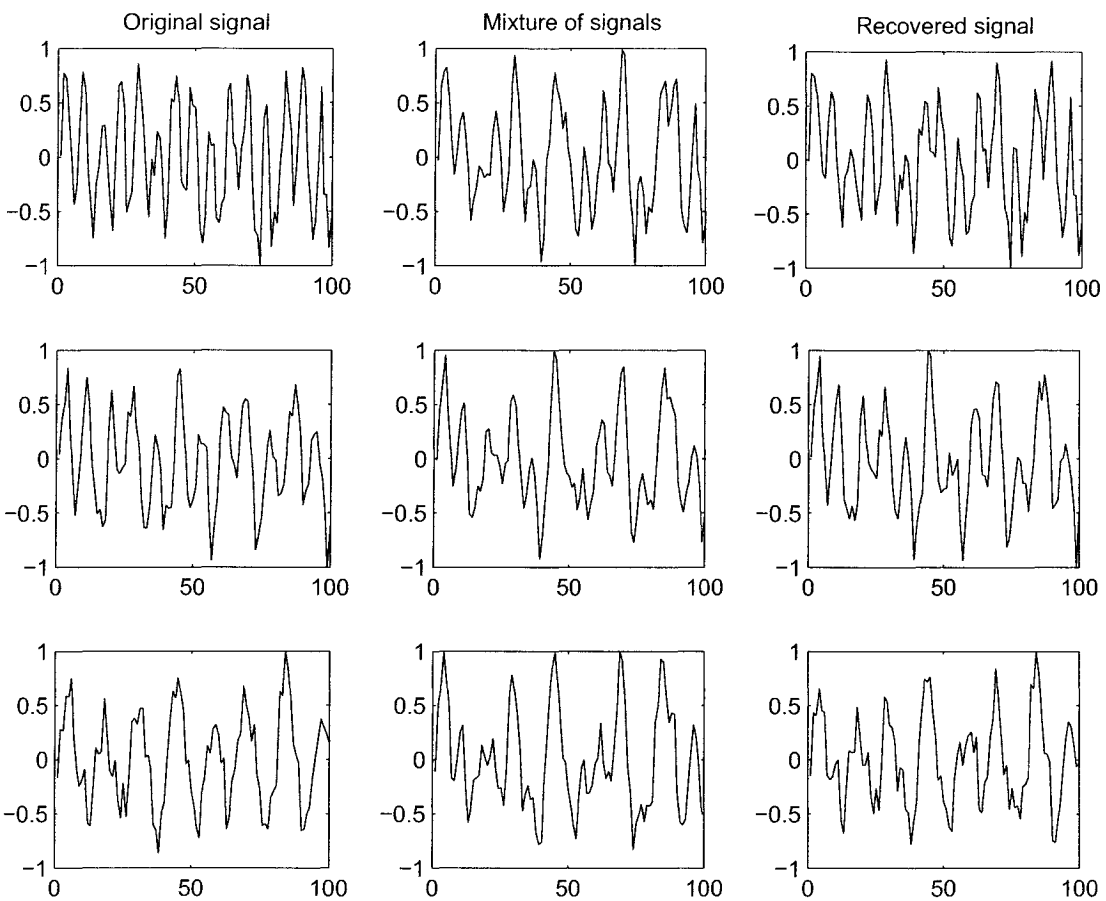

Figure 6.10: Separated signals for three sources and three sensors case when demixing filters are FIR (Finite Impulse Response).

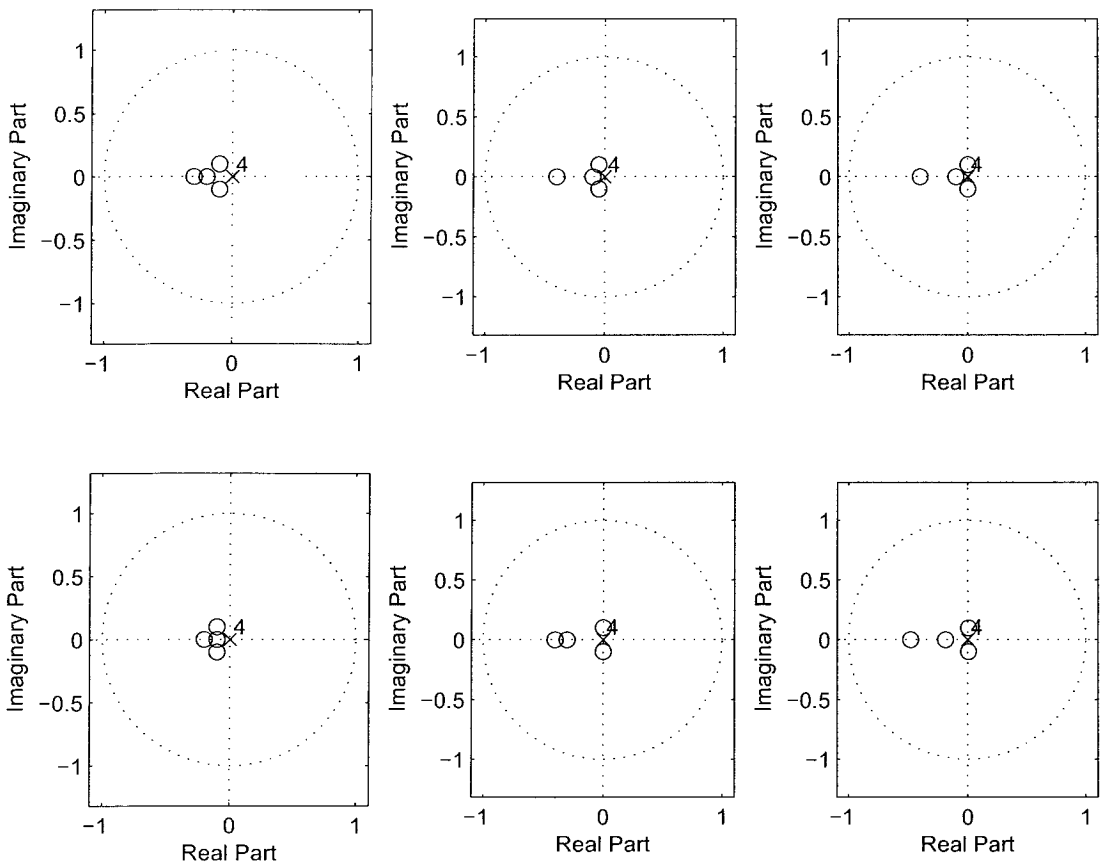

Figure 6.11: Pole-zero locations of $A_{12}(z)$, $A_{21}(z)$, $A_{31}(z)$, $A_{13}(z)$, $A_{23}(z)$, and $A_{32}(z)$ (clock wise).

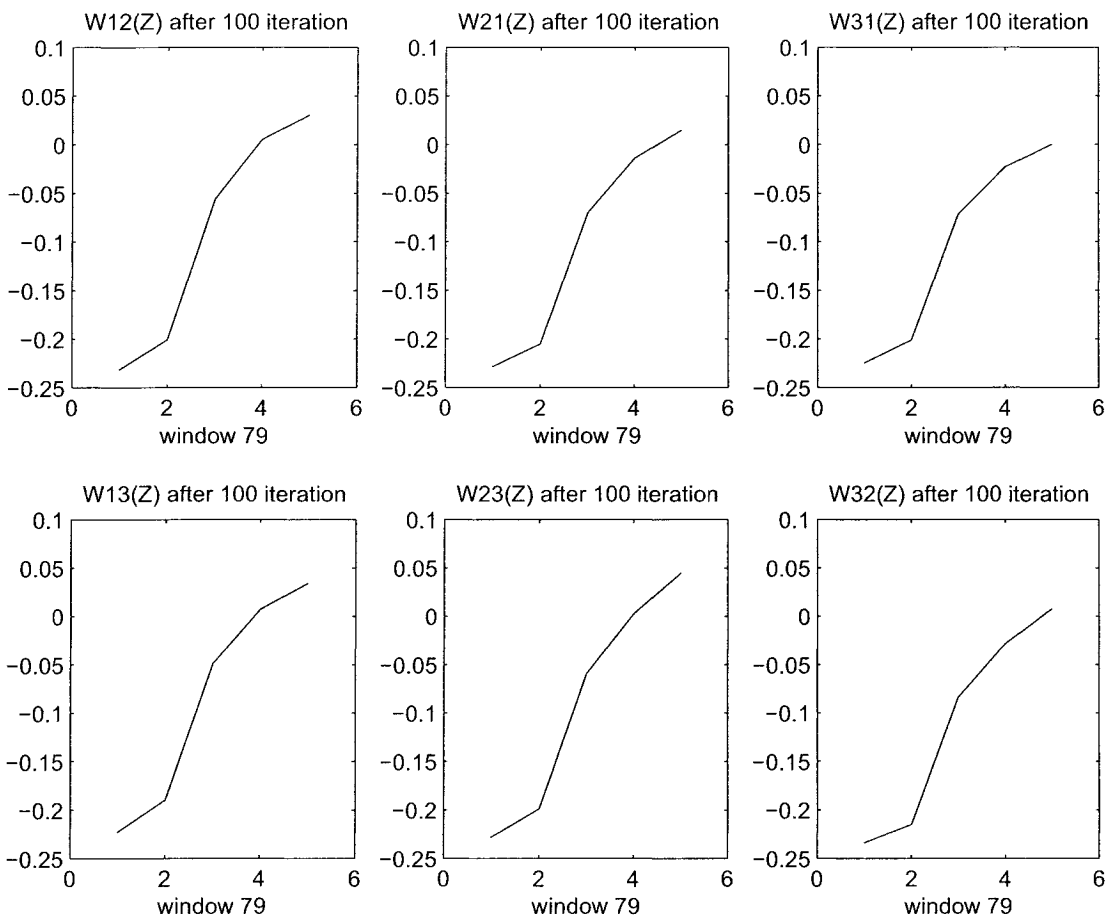

Figure 6.12: The learned coefficients of $W_{12}(z)$, $_{W13}(z)$, $W_{21}(z)$, $W_{23}(z)$, $W_{31}(z)$ and $W_{32}(z)$.

Chapter 7

Conclusions and future work

7.1 Thesis contribution

In this thesis, our goal was to determine how the quality of separated independent source components is affected by the variations in the environmental characteristics as modeled by convolving filter lengths and distribution of the mixing filters. It was determined that corresponding to a specific type of filter distribution the quality of the separated signals deteriorate with the increase in total filter length. We have also found some exceptional cases which yield results that are undesirable. As the filter length of both FIR (Finite Inpulse Response) and IIR (Infinite Inpulse Response)are increased it becomes more difficult to separate the signals, implying that this method can not always separate the mixed signals. This is one of the drawbacks of the information maximization approach. The ideal solution for an $N \times N$ feedback network architecture and the corresponding adaptive rules are derived for both of the FIR (Finite Impulse Response) and IIR (Infinite Impulse Response) demixing filter architectures. The results of FIR (Finite Impulse Response) and IIR (Infinite Impulse Response) scenario of two sources and two sensors case have been presented in the literature. We have also included the results of the three sources and three sensors case. But we have considered only FIR (Finite Impulse Response) case. The IIR (Infinite Impulse Response) case has not been shown here because of the insufficiency of time. The major contributions of this thesis are the newly

derived ideal solution for $N \times N$ feedback network architecture and adaptation rules for IIR (Infinite Impulse Response) demixing filter architecture. From simulation results chapter we have found that signal separation is more difficult with IIR (Infinite Impulse Response) filter architecture because it increases the system complexity. Also the number of demixing filter coefficients increase in this case. It will be more complex if the number of sources and the number of sensors are higher than three. The ideal solution derived in chapter four are very essential to verify if the mixing environmental filters to generate synthetic data can make the system stable or not. Special measures have been taken to prevent the system to become unstable. Two important measures were to control the adaptaion rates and initial values of the demixing filters. In all of the cases we have started with larger values of adaptation rates in first epoch and we have decreased those to lower values in rest of the epochs. But in each epoch we have used the constant value of adaptation rate. One of the major advantage of this information maximization approach is that it is easy to implement and does not require to prewhiten the data but it does not always guarantee the stable system. Measures must be taken to make the system stable.

7.2 Future work

In our thesis we have considered only two cases:

1. Two sources and two sensors case and,

2. Three sources and three sensors case.

For three sources and three sensors case we have presented results only for FIR (Finite Impulse Response) filter architectures. The results of IIR (Infinite Impulse Response) filter architectures have not been simulated because of the lack of the time. We hope to include this result in our future work. Also the maximum order of the filter we have taken in our thesis was eleven (11). We also hope to increase the filter order and extend the network to higher number of the sources and the sensors. The synthetically

generated datas were only for narrow band of frequencies which we hope to increase in future. We shall emphasize working on the moving targets in which case the added delay will be considered. Because in most of the practical cases there is a time delay between original source signal when it is generated and the time when it is received at the sensors. Since the statistics of the data changes continuously as the targets move, we can not use the steepest descent algorithm in this case. In chapter 3 we have discussed that stochastic gradient algorithm is applicable to the case when the statistics of the data changes continously. Computationally it is easier because we do not need to compute the mean of the data.

Bibliography

[1] A. Hyvärinen,"Survey on Independent Component Analysis.", *Neural Computing Surveys*, 2:94–128, 1999.

[2] J. H. Friedman., "Exploratory projection pursuit.", *J. of the American Statistical Association.*, 82(397):249-266, 1987.

[3] C. Jutten and J. Herault., "Blind separation of sources, partI: An adaptive algorithm based on neuromimetic architecture.", *Signal Processing.*, 24:1-10, 1991.

[4] P. Common., "Independent Component Analysis - a new concept?.", *Signal Processing.*, 36:287-314, 1994.

[5] J. Karhunen, A. Hyvärinen, R. Vigario, J. Hurri, and E. Oja.,"Applications of neural blind separation to signal and image processing.", In *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP'97).*, pages 131-134, Munich, Germany, 1997.

[6] H. H. Harman., *Modern Factor Analysis.*, University of Chicago Press, 2nd edition, 1967.

[7] M.Kendall., *Multivariate Analysis.*,Charles Griffin & Co., 1975.

[8] I.T. Jolliffe., *Principal Component Analysis.*, Springer-Verlag, 1986.

[9] J. H. Friedman and J. W. Tukey., "A projection pursuit algorithm for exploratory data analysis.", *IEEE Trans. of Computers*, c-23(9):881-890, 1974.

[10] P. J. Huber., "Projection pursuit.", *The annals of statistics*, 13(2):435-475, 1985.

[11] M.C. Jones and R. Sibson., "What is projection pursuit ?.", *J. of the Royal Statistical Society, ser. A*, 150:1-36, 1987.

[12] D.Cook, A. Buja, and J. Cabera., "Projection pursuit inexes based on orthonormal funcion expansions.", *J. of Computational and Graphical Statistics*, 2(3):225-250,1993.

[13] J. Sun., "Some practical aspects of exploratory projection pursuit.", *SIAM J. of Sci. Comput.,*, 14:68-80, 1993.

[14] A. Hyvärinen., "New approximations of differential entropy for independent component analysis and projection pursuit.", In *Advances in Neural Information Processing Systems 10*, pages 273-279. MIT Press, 1998.

[15] H. B. Barlow, "Possible principles underlying the transformations of sensory messages .", In W. A. Rosenblith, editor, *Sensory Communication*, pages 217-234, MIT Press, 1961.

[16] H. B. Barlow., "Single units and sensation: A neuron docrine for perceptual psychology?.", *Perception*, 1:371-394, 1972.

[17] H. B. Barlow., "Unsupervised learning.", *Neural Computation*, 1:295-311, 1989.

[18] J. F. Cardoso., "Source separation using higher order moments.", In *Proc. ICASSP'89*, pages 2109-2112, 1989.

[19] H. B. Barlow, T. P. Kaushal, and G. J. Mitchison, "Finding minimum entropy codes.", *Neural Computation*, 1:412-423, 1989.

[20] H. B. Barlow., "What is the computational goal of the neocortex?.",In C. Koch and J. L. Davis, editors, *Large-scale neuronal theories of the brain*, MIT Press, Cambridge, MA, 1994.

[21] J. J. Atick, "Entropy minimization: A design principle for sensory perception?.", *International Journal of Neural Systems*, 3:81-90, 1992.

[22] D. J. Field, "What is the goal of sensory coding?.", *Neural Computation*, 6:559-601,1994.

[23] J. Schmidhuber, M. Eldracher, and B. Flotin., "Semilinear predictibility minimization produces well-known feature detectors.", *Neural Computation*, 8:773-786, 1996.

[24] G. Deco and D. Obradovic., "Linear redundency reduction learning", *Neural Networks*, 8(5):751-755, 1995.

[25] Y. Sato., "A method for self-recovering equalization for multilevel amplitude-modulation system.", *IEEE Trans. on Communications*, 23:679-682, 1975.

[26] D. Donoho., "On minimum entropy deconvolution.", In *Applied Time Series Analysis II*, pages 565-608, Academic Press, 1981.

[27] O. Shalvi and E. Weinstein., "New criteria for blind deconvolution of nonminimum phase systems (channels).", *IEEE Trans. on Information Theory*, 36(2):312-321, 1990.

[28] O. Shalvi and E. Weinstein., "Super-exponential methods for blind deconvolution.", *IEEE Trans. on Information Theory*, 39(2):504-519, 1993.

[29] S. Haykin,"editor.", *Blind Deconvolution.*, Prentice-Hall, 1994.

[30] S. Haykin., *Adptive Filter Theory.*, Prentice-Hall International, 3rd edition, 1996.

[31] R. H. Lambert., *Multichannel Blind Deconvolution: FIR (Finite Impulse Response) Matrix Algebra and Separation of Multipath Mixtures.*, PhD thesis, Univ. of Southern California, 1996.

[32] Xiaoan Sun; Douglas, S.C., "Mean square error analyses of adaptive blind source separation algorithms.", *Neural Networks for Signal Processing XI, 2001. Proceedings of the 2001 IEEE Signal Processing Society Workshop*, Page(s):333 - 342, 10-12 Sept. 2001.

[33] Nakayama K., Hirano A., Horita A., "A learning algorithm for convolutive blind source separation with transmission delay constraint.", *Proceedings of the 2002 International Joint Conference on Neural Networks, 2002. IJCNN '02*, Volume 2, Page(s):1287 - 1292, 12-17 May, 2002.

[34] Sattar F., Charayaphan C., "Low-cost design and implementation of an ICA-based blind source separation algorithm.", *ASIC/SOC Conference, 2002. 15th Annual IEEE International*, Page(s):15 - 19, 25-28 Sept. 2002.

[35] Jayaraman, S.; Sitaraman, G.; Seshadri, R., "Blind source separation of acoustic mixtures using time-frequency domain Independent Component Analysis.", *Proceedings of the 9th International Conference on Neural Information Processing, 2002. ICONIP '02.*, Volume 3, Page(s):1383 - 1387, 18-22 Nov. 2002.

[36] Saruwatari H., Kawamura, T., Sawai K., Shikano, K., Kaminuma A., Sakata M., "Evaluation of fast convergence algorithm for blind source separation of real convolutive mixture.", *6th International Conference on Signal Processing, 2002*, Volume 1, Page(s):346 - 349, 26-30 Aug. 2002.

[37] Rahbar K., Reilly, J.P., "A new fast-converging method for blind source separation of speech signals in acoustic environments.", *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 2003* , Page(s):21 - 24, 19-22 Oct. 2003.

[38] Mukai, R.; Sawada, H.; Araki, S.; Makino, S., "Robust real-time blind source separation for moving speakers in a room.",*IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03).*, Volume 5, 6-10, Page(s):V - 469-72, April 2003.

[39] Yoshioka, M.; Omatu, S., "Independent component analysis using time delayed sampling.", *International Joint Conference on Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS*, Volume 4, Page(s):75 - 78, 24-27 July 2000.

[40] Almeida, L.B., "Linear and nonlinear ICA based on mutual information.", *The IEEE 2000 Symposium on Adaptive Systems for Signal Processing, Communications, and Control AS-SPCC.*, Page(s):117 - 122, 1-4 Oct. 2000.

[41] Dinh-Tuan Pham., "Fast algorithms for mutual information based independent component analysis.",*IEEE Transactions on Acoustics, Speech, and Signal Processing*, Volume 52, Issue 10, Oct. 2004.

[42] Douglas S.C., Sawada H., Makino S., "Natural gradient multichannel blind deconvolution and source separation using causal FIR (Finite Impulse Response) filters.",*IEEE International Conference on Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04)*, Volume 5, Page(s):477-80, 17-21 May, 2004.

[43] Dai, X., "Adaptive blind source separation of multiple-input multiple-output linearly time-varying FIR (Finite Impulse Response) system.",*Vision, Image and Signal Processing, IEE Proceedings*, Volume 151, Issue 4, Page(s):279 - 286, 30 Aug. 2004.

[44] Tome A.M., Teixeira A.R., Lang E.W., Stadlthanner K., Rocha A.P., Almeida R., "Blind source separation using time-delayed signals.",*IEEE International Joint Conference on Neural Networks, 2004. Proceedings, 2004*, Volume 3, Page(s):2187 - 2191, 25-29 July 2004.

[45] Murillo-Fuentes, J.J. Gonzalez-Serrano, F.J. ., "A sinusoidal contrast function for the blind separation of statistically independent sources.",*IEEE Transactions on Acoustics, Speech, and Signal Processing*, Volume 52, Issue 12, Page(s):3459 - 3463, Dec. 2004.

[46] Buchner H., Aichner R., Kellermann W., "A generalization of blind source separation algorithms for convolutive mixtures based on second-order statistics.",*IEEE Transactions on Speech and Audio Processing*, Volume 13, Issue 1, Page(s):120 - 134, Jan. 2005.

[47] A. Hyvärinen, Juha Karhunen, and E. Oja.,In *Independent Component Analysis.*, A Wiley-Interscience Publication, John Wiley & Sons, 2001.

[48] C. Therrien, *Discrete Random Signals and Statistical Signal Processing*, Prentice-Hall, 1992.

[49] C. Jutten., *Calcul neuromimétique et traitement du signal, analyse es composantes indépendentes.*, PhD thesis, INPG, Univ. Grenoble, 1987.

[50] Y. Bar-Ness,"Bootstrapping adaptive interference cancellers: Some practical limitations.", In *The Glovecom Conf.*, pages 1251-1255. Miami, 1982. Paper F3.7.

[51] Athanasios Papoulis., *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill, 3rd edition, 1991.

[52] J. F. Cardoso., "Super-symmetric decomposition of the fourth-order cumulant tensor. blind identification of more sources than sensors.", In *Proc. ICASSP'91*, pages 3109-3112, 1991.

[53] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by V1.", *Vision Research*, 37:3311 - 3325, 1997.

[54] M. Lewicki and B. Olshausen, "Inferring soarse, overcomplete image codes using an efficient coding framework.", In *Advances in Neural Information Processing 10 (Proc. NIPS*97)*, pages 815-821, MIT Press, 1998.

[55] A. Hyvärinen, "New approximations of differential entropy for independent componenet analysis and projection pursuit.", *In Advances in Neural Information Processing Systems 10*, pages 273-279.MIT Press, 1998.

[56] M. Lewicki and T. J. Sejnowski, "Learning overcomplete representations.", In *Advances in Neural Information Processing 10 (Proc. NIPS*97)*, pages 556-562, MIT Press, 1998.

[57] A. Hyvärinen, R. Cristescu, and E. Oja., "A fast algorithm for estimating overcomplete ICA bases for image windows.", In *Proc. Int. Joint Conf. on Neural Netweorks*, Washington, D.C., 1999.

[58] P. Common., "Blind identification in presence of noise.", In *Signal Processing VI: Theories and Application (Proc. EUSIPCO).*, pages 835-838. Elsevier, 1992.

[59] L. De Lathauwer, B. De Moor, and J. Vandewalle., "A technique for higher-order-only blind source separation.", In*Proc. ICONIP*, Hong Kong, 1996.

[60] E. Moulines, J.-F. Cardoso, and E. Gassiat, "Maximum likelihood for blind separaion and deconvolution of noisy signals using mixture models.", In *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP'97)*, pages 3617-3620, Munich, Germany, 1997.

[61] A. Hyvärinen and E. Oja., "Simple neuron models for independent component analysis.", *Int. Journal of Neural Systems*, 7(6):671-687, 1996.

[62] A. Hyvärinen., "A family of fixed-point algorithms for independent component analysis.", In *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing(ICASSP'97)*, pages 3917-3920, Munich, Germany, 1997.

[63] A. Hyvärinen, E. Oja, L. Wang, R. Vigario, and J. Joutsensalo.", "A class of neural networks for independent component analysis.", *IEEE Trans. on Neural Networks*, 8(3):486-504, 1997.

[64] A. Hyvärinen.,"Fast and robust fixed-point algorithms for independent component analysis.", *IEEE Trans. on Neural Networks*, 1999.

[65] S. Amari, A. Cichocki, and H. H. Yang. "A new learning algorithm for blind signal separation.", In *Advances in Neural Information Processing Systems 8.*, MIT Press, 1996.

[66] N. Delfosse and P. Loubaton, "Adaptive blind separation of independent sources: a deflation approach.", *Signal Processing*, 45:59-83, 1995.

[67] A. Hyvärinen, "One-unit contrast functions for independent component analysis: A statistical analysis.", In *Neural Networks for Signal Processing VII (Proc. IEEE Workshop on Neural Networks for Signal Processing)*, pages 388-397, Amelia Island, Florida, 1997.

[68] F. R. Hampel, E. M. Ronchetti, P. J. Roussew, and W. A. Stahel, "Robust Statistics.", *Robust Statistics*, Wiley, 1986.

[69] A.J. Bell and T. J. Sejnowski. "An information-maximisation approach to blind separation and blind deconvolution.", *Neural Computation,*, 7(6): 1129-1159, 1995. *Probability, random variables and stochastic processes, 2nd edition,*, McGraw-Hill, New York.

[70] A. Hyvärinen, E. Oja, P. Hoyer, and J. Hurri., "Image feature extraction by sparse coding and independent component analysis.", In *Proc. Int. Conf. on Pattern Recognition (ICPR'98)*, pages 1268-1273, Brisbane, Australia, 1998.

[71] Kari Torkkola., "Blind separation of convolved sources based on information maximization", *IEEE workshop on Neural Networks for Signal Processing,* Kyoto, Japan, Sept 4-6 1996.