DEVELOPMENT OF NOVEL INDEPENDENT COMPONENT ANALYSIS TECHNIQUES AND THEIR APPLICATIONS

Dissertation submitted in partial fulfillment of the requirements for the award of
Doctor of Philosophy

by

DEBIPRASAD PRIYABRATA ACHARYA

Under the supervision of

Prof. (Dr.) G. Panda
Dr. Y. V. S. Lakshmi

ELECTRONICS AND COMMUNICATION ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY
ROURKELA
INDIA

Dedicated to my Grand Parents "**Maa o Jeji**"

ACKNOWLEDGEMENTS

I take the opportunity to express my reverence to my supervisors Prof. G.Panda and Dr. Y.V.S.Lakshmi for their guidance, inspiration and innovative technical discussions all during the course of this work. I find words inadequate to thank them for enabling me to complete this work in spite of all obstacles.

I thank my colleagues Jaison, Raju at Center for Development of Telematics, Bangalore for their friendly support. I am indebted to Wg. Cdr. H.R.Pathasarathy and Devadas of C-DOT for their encouragements.

I am also thankful to all faculty members and research students of ECE Department, NIT Rourkela. Special thanks go to Dr. K.K.Mohapatra, Dr. S.K.Patra and Dr. S.Meher for their constructive criticism and motivation during the course of my work.

I am grateful to Prof. P.K.Dash for the seed of motivation sown by him for this research. I express my sincere regards to Dr. S.Mishra of IIT Delhi for his endless patience to clarify my doubts and a brotherly affection without which this work would not have taken this shape.

I am thankful to Dr. D.P.Das for his moral support at times of need. It’s my pleasure to show my indebtedness to my friends at NIT like Sunil, Saroj, Manas, Babita, Shusant, Ayeskant, Jitendrasir, Debi, Nithin and Ajit for their help during the course of this work. Special thanks to Prakash for infallible motivation.

I can’t represent my words to speak of the inspirations from my father and also father-in-law. Love and support from Chhuabou, Babapa, Bou, Badababapa and Jeji in disguise has been instrumental in completing this task. Wishes from Kanha, Baidhana, Baimana, Gunthu, Tuba, Laltu, Pultu, Babloo, Tutuna, Tuna, Lulu, Gagy, Bapunu, Kuna, Muna, Manu, Julu and Tulu is the key to my steps towards success.

I find words inadequate to express any form of acknowledgement to my grand mother who has been an emblem of blessings in all my steps of life.

(Debiprasad Priyabrata Acharya)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

ADC	Analog-to-Digital Converter
AWGN	Additive White Gaussian Noise
ASIC	Application Specific Integrated Circuit
BER	Bit Error Rate
BFO	Bacterial Foraging Optimization
BFOICA	Bacterial Foraging Optimization based Independent Component Analysis
BPSK	Binary Phase Shift Keying
BSS	Blind Source Separation
CDMA	Code Division Multiple Access
CGAICA	Constrained Genetic Algorithm based Independent Component Analysis
CT	Cross Talk
DFT	Discrete Fourier Transform
DS-SS	Direct Sequence Spread Spectrum
ECG	Electro Cardiogram
EVD	Eigenvalue Decomposition
FPGA	Field Programmable Gate Array
GA	Genetic Algorithm
IC	Independent Component
IC	Integrated Circuit
ICA	Independent Component Analysis
IDFT	Inverse Discrete Fourier Transform
ISI	Inter Symbol Interference
JADE	Joint Approximate Diagonalization of Eigenmatrices
LFSR	Linear Feedback Shift Register
LMS	Least Mean Square
MEG	Magneto Encephalography

ML	Maximum Likelihood
MSE	Mean Square Error
MUD	Multi User Detection
MUI	Multi User Interference
PCA	Principal Component Analysis
PN	Pseudorandom Noise
PSO	Particle Swarm Optimization
SJR	Signal to Jammer Ratio
SNR	Signal to Noise Ratio
VHDL	Very High Speed Integrated Circuits Hardware Description Language
VLSI	Very Large Scale Integrated Circuits

Abstract

Independent Component Analysis (ICA) is a statistical signal processing technique having emerging new practical application areas, such as blind separation of mixed voices or images, analysis of several types of data or feature extraction. This thesis develops linear and nonlinear independent component analysis methods which exploit the computational intelligence of evolutionary techniques. A fixed-point error estimation of the linear ICA techniques for their Very Large Scale Integrated Circuits (VLSI) implementation is carried out. The linear ICA technique is applied to interference cancellation in direct sequence spread spectrum communications.

Novel methods for ICA using evolutionary computation techniques are developed. The use of evolutionary computation based optimizations in ICA resolves the permutation ambiguity to a large extent. This also ensures the convergence of the algorithm to a global optimum.

Most of the real world mixtures are nonlinear in nature. Separation of signals from their nonlinear mixtures is performed by nonlinear ICA technique. Though several algorithms exist for linear ICA, very less work has been carried out on nonlinear ICA algorithms. A computationally intelligent technique called bacterial foraging based optimization is used for ICA algorithms for separation of signals from their post nonlinear mixtures.

ICA techniques being computationally demanding, give rise to errors when implemented by using fixed-point arithmetic. Hence an error analysis becomes inevitable prior to their implementation by using any VLSI architecture. An analysis on the effects of finite register length on different ICA algorithms for their VLSI implementation is carried out. The most popular fixed-point fast ICA algorithm is implemented in FPGA.

The spread spectrum technology forms the backbone of the third generation wireless communication systems and possible future ones. ICA algorithms can be conveniently applied in wireless communication systems such as Direct Sequence Spread Spectrum receivers (DS-SS) based receivers. Different ICA techniques developed in this thesis are applied to mitigate the interference in Direct Sequence Spread Spectrum receivers. When ICA is used for separation of interference and the DS-SS signal, two antenna array diversity is used. However at the output of the ICA block, the recognition of the signal from the two separated outputs becomes very much necessary. This is due to the inherent permutation ambiguity of ICA techniques and is generally accomplished in many cases by correlation with a training signal. But in some cases the correlation techniques do not yield satisfactory result, and also it becomes advantageous to avoid the processing overhead of signal classification. In such cases the evolutionary ICA techniques seem to be beneficial. This issue has been taken up in the present work.

Chapter

I

INTRODUCTION

1.1 Motivation

Real world problems very often provide minimum information regarding their causes. This is mainly due to the system complexities and noninvasive techniques employed by scientists and engineers to study such systems. Signal and image processing techniques used for analyzing such systems essentially tend to be blind. Earlier, training signal based techniques were used extensively for such analyses. But many times either these training signals are not practicable to be availed by the analyzer or become burden on the system itself. Hence blind signal/image processing techniques are becoming predominant in modern real time systems. In fact, blind signal processing has become a very important topic of research and development in many areas, especially biomedical engineering, medical imaging, speech enhancement, remote sensing, communication systems, exploration seismology, geophysics, econometrics, data mining, sensor networks etc. Blind Signal Processing has three major areas: Blind Signal Separation and Extraction, Independent Component Analysis (ICA) and Multichannel Blind Deconvolution and Equalization. ICA technique has also been typically applied to the other two areas mentioned above. Hence ICA research with its wide range of applications is quite interesting and has been taken up as the central domain of the present work.

Though several efficient ICA algorithms have been reported in literature, they suffer from the following problems.

Permutation and Scaling Ambiguity:

Almost all of the ICA algorithms suffer from the problem that the recovered independent components are not in a proper order always. With different initializations their order of appearance varies. Also the components are estimated only up to multiplicative scalar.

Convergence to different Solutions:

Many (not all) gradient ICA algorithms converge to different solutions for different initializations.

Nonlinear ICA:

Most of the real world mixtures are nonlinear in nature. Separation of signals from their nonlinear mixtures is performed by nonlinear ICA technique.Though several algorithms exist for linear ICA, very less work has been carried out on nonlinear ICA algorithms.

Apart from these problems, there has been no analysis in literature for error introduced due to fixed-point hardware implementation of any of the ICA algorithms.

The spread spectrum technology forms the backbone of the third generation systems and possible future ones. ICA algorithms can be conveniently applied in wireless communication systems such as DS-SS based receivers.

1.2 Scope of the Thesis

This thesis was born out of the need to develop linear and nonlinear independent component analysis methods which exploit the computational intelligence of evolutionary techniques, carry out a fixed-point error estimation of the linear ICA techniques for their Very Large Scale Integrated Circuits (VLSI) implementation and apply linear ICA to interference cancellation in direct sequence spread spectrum communications. Hence, the following are outlined as the scope of this thesis:

1. To develop novel methods for ICA using evolutionary computation techniques. The use of evolutionary computation based optimizations in ICA

resolves the permutation ambiguity to a large extent. This also ensures the convergence of the algorithm to a global optimum.

2. To apply the computationally intelligent techniques for ICA algorithms for separation of signals from their post nonlinear mixtures.

3. To carry out an analysis on the effects of finite register length on different ICA algorithms for their VLSI implementation and to implement the most popular fast ICA algorithms in FPGA. ICA techniques being computationally demanding, give rise to errors when implemented by using fixed-point arithmetic. Hence an error analysis becomes inevitable prior to their implementation by using any VLSI architecture. The scope also covers a fixed-point implementation of fast ICA algorithm for prototype demonstration to supplement the validity of error analysis.

4. To apply different ICA techniques to interference suppression in DS-SS receivers. When ICA is used for separation of interference and the DS-SS signal, two antenna array diversity is used. However at the output of the ICA block, the recognition of the signal from the two separated outputs becomes very much necessary. This is due to the inherent permutation ambiguity of ICA techniques and is generally accomplished in many cases by correlation with a training signal. But in some cases the correlation techniques do not yield satisfactory result, and also it becomes advantageous to avoid the processing overhead of signal classification. In such cases the evolutionary ICA techniques seem to be beneficial. This issue has been taken up in the present work.

The structure of the proposed thesis is detailed below.

1.3 Structure and Chapter wise Contribution of the Thesis

Chapter-I

Introduction

The motivation behind the Independent Component Analysis problem is introduced. The reported work on this topic is reviewed in this chapter. The summary of the contributions is also outlined.

Chapter-II

Review of Independent Component Analysis and its applications

In this chapter the basic principle behind the independent component analysis technique is discussed. The contrast functions for different routes to independence are clearly depicted. Different existing algorithms for ICA are briefly illustrated and are critically examined with special reference to their algorithmic properties. The ambiguities present in these algorithms are also presented. Finally the application domains of this novel technique are presented. Some of the futuristic works on ICA technique which need further investigation are development of nonlinear ICA algorithms, design of low complexity ICA algorithms and use of evolutionary computing optimization tools for developing ICA and finally alleviation of permutation and scaling ambiguities existing in present ICA.

Chapter-III

Development of Novel Constrained Genetic Algorithm Based ICA Algorithm

The Genetic Algorithm based optimization is used in a constrained manner to estimate the independent components from their observed mixtures. This scheme is tested using several examples including the speech signals for instantaneous mixing cases. From the simulation results it is very clear that in the CGAICA algorithm the MSE of the estimated ICs decreases with increase in the number of chromosomes. Also for a fixed number of chromosomes the number of generations affects the MSE. The CGAICA algorithm yields almost comparable MSE as obtained by the fast ICA algorithm. The permutation ambiguity present in ICA techniques is resolved by use of the CGAICA algorithm if we have relative knowledge of the statistical characteristics of the signals to be estimated. But this doesn't resolve the uncertainty if the signals have nearly equal value of cost function.

Chapter-IV

Development of a New ICA Algorithm Based on Constrained Bacterial Foraging Optimization Technique

In this Chapter the bacteria foraging based optimization is used in a constrained manner to estimate the independent components from their observed mixtures. This scheme is tested using several examples including the speech signals for instantaneous mixing cases. From the simulation results it is very clear that the BFOICA algorithm has faster convergence and better mean square error performance than as compared to the CGAICA algorithm. In comparison to the fast ICA algorithm it has good mean square error. The permutation ambiguity present in ICA techniques is resolved by use of the BFOICA algorithm if relative knowledge of the statistical characteristics of the signals to be estimated is known. The computational complexity of the BFOICA is obviously higher as compared to the gradient based algorithms. But as the number of independent components increases the BFOICA algorithm performs better than the available gradient based methods.

Chapter-V

Development of Novel Nonlinear ICA Technique based on Bacteria Foraging Optimization

A Bacteria Foraging Optimization based nonlinear ICA approach is proposed which is applied to post nonlinear mixtures of signals. BFO is used for estimating the nonlinear functions blindly and then the most popular fast ICA algorithm is hybridized with the algorithm to evaluate the linear demixing matrix. The developed technique is compared with the GA based nonlinear ICA technique and found to have better convergence.

Chapter-VI

On Effects of Finite Register Length on Different ICA Algorithms for VLSI Implementation

The present paper studies the effect of finite register length on the accuracy of two different evolutionary computation based ICA algorithms BFOICA and CGAICA and also the most popular fast ICA algorithm. Extensive simulation studies reveal that kurtosis based CGAICA yields higher MSE compared to kurtosis based BFOICA. Further for the same bit length, the fixed-point BFOICA offers substantially low MSE compared to the CGAICA (kurtosis based) and fixed-point fast ICA offers better performance than both evolutionary computation based ICA algorithms. The separation ability of fixed-point ICA depends on the number of bits used. Fixed-point BFOICA performs superior to the fixed-point CGAICA. Performance of fixed-point fast ICA in comparison to both fixed-point BFOICA and CGAICA algorithms is better due to less computational complexity involved in fast ICA algorithm. The performance of the proposed architecture for fast ICA is evaluated based on trial implementations with FPGA. It is also inferred that with speed optimization more resources are required for implementation.

Chapter-VII

Blind Jammer Suppression in DS-SS Systems using ICA Techniques

This Chapter studies the jammer mitigation in Direct Sequence Spread Spectrum communications system by using fast ICA technique and proposes a new BFOICA and CGAICA based scheme. It is shown that both fast ICA CGAICA and BFOICA assisted DS-SS model yield better BER performance even at very high jammer power values. However selection of desired spread spectrum signal in case of fast ICA based scheme adds to the computational complexity. Our proposed CGAICA/BFOICA assisted jammer suppression in DS-SS communication systems has almost comparable performance as the fast ICA based jammer suppression. The additional signal selection block needed in case of fast ICA and other gradient based ICA techniques for jammer mitigation in DS-SS systems, is avoided by our proposed scheme.

Chapter-VIII

Conclusions and Future Work

The overall contributions of the thesis are listed with reference to their limitations. The scope for future research activities is outlined.

1.4 List of Publications Related to Thesis

Published/Accepted:

Conferences:

[P 1] D.P.Acharya, G.Panda, "Effect of Finite Register Lengths on Fast ICA Algorithm", *NUCONE* , Ahmedabad, India 2006

[P 2] D.P.Acharya, G.Panda and Y.V.S.Lakshmi, "FPGA Implementation Of A Novel Optimized Fixed-Point VLSI Architecture For Fast ICA Algorithm", *(IEEE) International Conference on Signal and Image Processing*, Hubli, Karnataka, 2006.

[P 3] D.P.Acharya, G.Panda and Y.V.S.Lakshmi, "Fixed-point Error Evaluation of Fast ICA and Algebraic ICA Algorithms", *IEEE International Conference on Industrial Technology*, Mumbai, India, Dec. 2006.

[P 4] D.P.Acharya, G.Panda and Y.V.S.Lakshmi, "Constrained Genetic Algorithm Based Independent Component Analysis", *IEEE Congress on Evolutionary Computation*, Singapore, Sept. 2007.

[P 5] D.P.Acharya, G.Panda, S.Mishra and Y.V.S.Lakshmi, "Bacteria Foraging Based Independent Component Analysis", *IEEE International Conference on Computational Intelligence and Multimedia Applications*, Sivakasi, India , Dec. 2007.

[P 6] D.P.Acharya, G.Panda and Y.V.S.Lakshmi, "Issues and Performance Study of Jammer Suppression in Direct Sequence Spread Spectrum Communications using Fast ICA", *National Conference on Emerging Trends in Information Technology*, Pune, India September 2007.

[P 7] D.P.Acharya, G.Panda and Y.V.S.Lakshmi, "Effect of Finite Register Length on Bacterial Foraging Optimization based ICA and Constrained Genetic Algorithm based ICA Algorithm" *International Conference on Signal Processing and Communication Networks*, Chennai, India Jan 2008.

[P 8] D.P.Acharya, G.Panda and Y.V.S.Lakshmi, "Blind Suppression of Jammer Interference in Spread Spectrum Communications using Bacterial Foraging Optimization based Independent Component Analysis", Accepted for presentation in *International Conference in Sensors, Signal Processing, Communication, Control and Instrumentation* (SSPCCIN), Pune, India , Jan. 2008.

Journals

[P 9] D.P.Acharya and G.Panda, "Blind Suppression of Jammer Interference in Spread Spectrum Communications using Bacterial Foraging Optimization based Independent Component Analysis", Accepted in *International Journal of Computer Science and Network Security*.

[P 10] D.P.Acharya, G.Panda, S.Mishra and Y.V.S.Lakshmi, "Bacteria Foraging Based Independent Component Analysis", *International Journal of Systemics, Cybernetics and Informatics* (In Press).

Communicated:

[C 1] D.P.Acharya, G.Panda and Y.V.S.Lakshmi, "Effects of Finite Register Length on Fast ICA, Bacterial Foraging Optimization based ICA and Constrained Genetic Algorithm based ICA Algorithm", Communicated to Elsevier, Digital Signal Processing, Manuscript No:DSP-S-0700225[1].

[C 2] D.P.Acharya, G.Panda and Y.V.S.Lakshmi, "Independent Component Analysis Based on Constrained Genetic Algorithm Optimization Technique", Communicated to International Journal of Computational Intelligence and Applications.

[C 3] D.P.Acharya and G.Panda, "Independent Component Analysis and its applications- A Review", Communicated to IETE Technical Review.

Under Preparation:

[1] Jammer Mitigation in Direct Sequence Spread Spectrum Communications using Evolutionary Computation based ICA Techniques, Signal Processing, Elsevier.

[2] Nonlinear Independent Component Analysis using Bacteria Foraging Optimization for IEEE Transactions on Evolutionary Computation.

1.5 Summary of Publications Related to Thesis

[P 1] This paper evaluates the error performance of fast ICA algorithm for its fixed-point implementations. Simulation study is carried on both fixed- and floating-point fast ICA. It is observed that the word length greatly influences the separation performance. The performance analysis is compared for fast ICA algorithm with three different optimization functions. Out of the three kurtosis based and exponential contrast function based fast ICA offers superior performance than tanhy based fast ICA when the same word length is used in all the cases.

[P 2] This work proposes Very Large Scale Integrated Circuits (VLSI) architecture for fixed-point implementation of fast ICA algorithm using kurtosis as optimization function. Simulation analysis is carried out for fixed-point number representation of the fast ICA algorithm with the proposed architecture. With an optimized register length, the proposed architecture is described in VHDL for implementation through Field Programmable Gate Array (FPGA) and Altera Quartus II tool is used for synthesis and fitting. An Altera Stratix II FPGA is used as the target device. Results of FPGA simulation and synthesis are reported.

[P 3] This investigation evaluates the error performance of fast ICA and algebraic ICA algorithms for their fixed-point implementations. Algebraic ICA is the fastest ICA but as the number of sources to be separated increases the complexity increases. Simulation study is carried on both fixed and floating point ICAs. Out of the two ICAs studied the algebraic ICA offers superior performance when the same word length is used in both the cases.

[P 4] This paper proposes a constrained genetic algorithm optimization based independent component analysis assuming a noise free Independent Component Analysis (ICA) model. It investigates on the application and performance of GA, a popular evolutionary computation technique, in independent component analysis problem. It is observed that the proposed constrained genetic algorithm optimization based ICA overcomes the long standing permutation ambiguity and recovers the independent components in a fixed order which is dependent on the statistical characteristics of the signals to be estimated. The constrained GA based ICA has also been compared with the most popular fast ICA algorithm.

[P 5] The present paper proposes a bacteria foraging optimization based independent component analysis (BFOICA) algorithm assuming a linear noise free model. It is observed that the proposed BFOICA algorithm overcomes the long standing permutation ambiguity and recovers the independent components(IC) in a fixed order which depends on the statistical characteristics of the signals to be estimated. The paper compares the performance of BFOICA algorithm with the constrained genetic algorithm based ICA (CGAICA) and most popular fast ICA algorithm. The proposed algorithm offers comparable or even better performance compared to fast ICA algorithm and faster convergence and better mean square error performance compared to CGAICA.

[P 6] The present work studies and analyses the issues involved in interference rejection in direct sequence spread spectrum communication systems based on Independent Component Analysis technique. The ICA technique tries to separate the unwanted interfering signal from the desired signal so that contamination of the desired spread spectrum signal is minimized. The effect of the inherent ambiguities of ICA on this model is also analyzed. Results of the simulation study carried out on the fast ICA assisted jammer mitigation for spread spectrum communication is presented in this work.

[P 7] This work evaluates the error performance of BFOICA and CGAICA algorithm for its fixed-point implementation. Simulation study is carried on both fixed and floating point ICA algorithms. It is observed that the word length used greatly influences the separation performance. A comparison of fixed-point error performance of both the algorithms is also carried out in this work.

[P 8] The present work proposes the interference rejection in direct sequence spread spectrum (DS-SS) communication systems based on BFOICA technique.

The BFOICA technique separates the unwanted interfering signal from the desired signal so that contamination of the desired spread spectrum signal is minimized. The effect of the inherent ambiguities of fast ICA on ICA assisted jammer suppression in case of DS-SS system model is also analyzed. Results of the simulation study carried out on the BFOICA assisted interference mitigation for direct sequence spread spectrum communication is compared with fast ICA assisted jammer mitigation DS-SS communication in this work. A continuous wave sinusoidal jammer is considered here.

[P 9] This paper uses the proposed BFOICA technique for jammer mitigation in direct sequence spread spectrum (DS-SS) communication receivers. The BFOICA technique separates the unwanted interfering signal from the desired signal so that contamination of the desired spread spectrum signal is minimized. The effect of the inherent ambiguities of fast ICA on ICA assisted jammer suppression in case of DS-SS system model is also analyzed. BFOICA is reported to have over come the permutation ambiguity in many cases. Results of computer simulation carried out on the BFOICA assisted jammer mitigation for direct sequence spread spectrum communication is compared with fast ICA assisted jammer mitigation DS-SS communication in this work. The jammers considered here are continuous wave sinusoidal and random in nature.

[P 10] The bacteria foraging based optimization is used in a constrained manner to estimate the independent components from their observed mixtures. This scheme is tested using several examples including the speech signals for instantaneous mixing cases. From the simulation results it is very clear that the BFOICA algorithm has faster convergence and better mean square error performance than compared to the CGAICA algorithm. In comparison to the fast ICA algorithm it has good mean square error. The permutation ambiguity present in ICA techniques has been resolved by use of the BFOICA algorithm if we have relative knowledge of the statistical characteristics of the signals to be estimated.

Chapter

II

REVIEW OF INDEPENDENT COMPONENT ANALYSIS AND ITS APPLICATIONS

2.1 Introduction to ICA

Recently, there has been an increasing interest in statistical models for learning data representations. A very popular method for this task is independent component analysis (ICA), the concept of which was initially proposed by Comon [2.1]. The ICA algorithm was initially proposed to solve the blind source separation (BSS) problem i.e. given only mixtures of a set of underlying sources, the task is to separate the mixed signals and retrieve the original sources [2.2, 2.3]. Neither the mixing process nor the distribution of sources is known in the process. A simple mathematical representation of ICA model is as follows.

Consider a simple linear model which consists of N sources of T samples i.e. $\mathbf{s}_{\mathrm{i}} = [\mathrm{s}_{\mathrm{i}}(1)\ldots\mathrm{s}_{\mathrm{i}}(\mathrm{t})\ldots\mathrm{s}_{\mathrm{i}}(\mathrm{T})]$. The symbol t here represents time but it may represent some other parameter like space. M weighted mixtures of the sources are observed as $\mathbf{X}$, where $\mathbf{X}_{\mathrm{i}} = [\mathrm{X}_{\mathrm{i}}(1)\ldots\mathrm{X}_{\mathrm{i}}(\mathrm{t})\ldots\mathrm{X}_{\mathrm{i}}(\mathrm{T})]$. This can be represented as

$$\mathbf{X} = \mathrm{A}\,\mathbf{S} + \mathbf{n} \tag{2.1}$$

where

$$\mathbf{X} = \begin{bmatrix} \mathrm{X}_1 \\ \mathrm{X}_2 \\ \cdots \\ \mathrm{X}_\mathrm{M} \end{bmatrix}, \ \mathbf{S} = \begin{bmatrix} \mathrm{S}_1 \\ \mathrm{S}_2 \\ \cdots \\ \mathrm{S}_\mathrm{N} \end{bmatrix} \text{ and } \mathbf{n} = \begin{bmatrix} n_1 \\ n_2 \\ \cdots \\ n_K \end{bmatrix} \tag{2.2}$$

and n represents the additive white Gaussian noise (AWGN). It is assumed that there are at least as many observations as sources i.e. $\mathrm{M} \geq \mathrm{N}$. The $\mathrm{M} \times \mathrm{N}$ matrix A is represented as

$$\mathrm{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \cdots & \cdots & \cdots & \cdots \\ a_{M1} & a_{M2} & \cdots & a_{MN} \end{bmatrix} \tag{2.3}$$

relates X and S. A is called the mixing matrix. The estimation of the matrix S with knowledge of X is the linear source separation problem. This is schematically shown in Fig 2.1. Fig 2.2 shows two original signals and their mixtures.

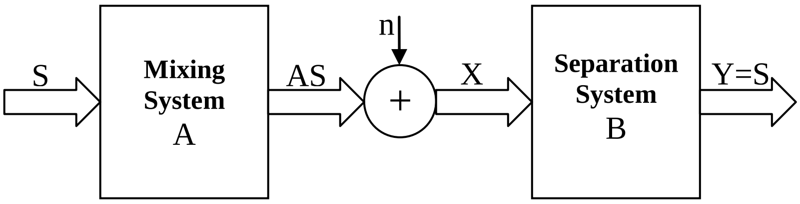

Fig 2.1 Illustration of mixing and separation system.

A is the mixing matrix and B is the unmixing matrix.

The source separation problem cannot be solved if there is no knowledge of either A or $\mathbf{S}$ apart from the observed mixed data $\mathbf{X}$. If the mixing matrix A is known and the additive noise n is negligible then the original sources can be estimated by evaluating the pseudo inverse of the matrix A which is known as the unmixing matrix B such that

$$\mathrm{B}\mathbf{X} = \mathrm{BA}\mathbf{S} = \mathbf{S} \tag{2.4}$$

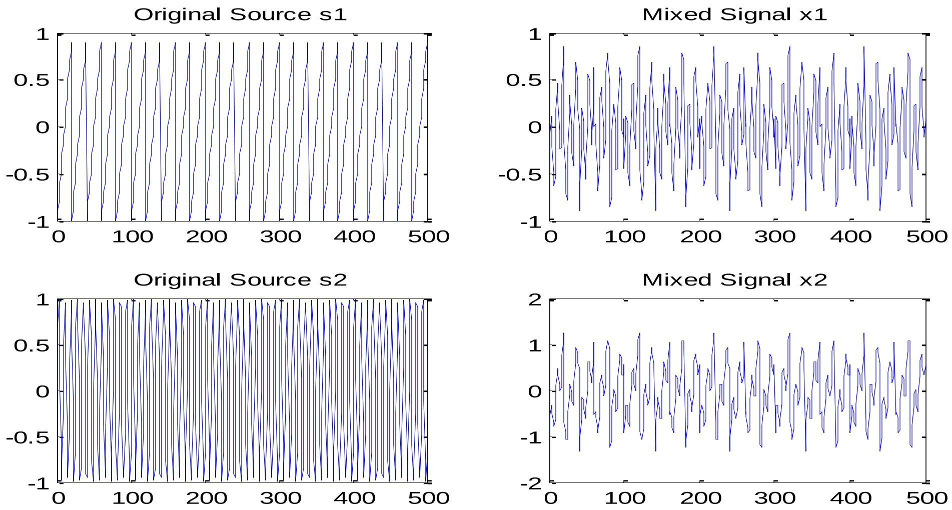

Fig 2.2 Effect of mixing. The original sources s_1 and s_2 are shown in left plot, and the mixed signals x_1 and x_2 are shown in the right plot.

For cases where the number of observations M equals number of sources N (i.e. $\mathrm{M} = \mathrm{N}$), the mixing matrix A is a square matrix with full rank and $\mathrm{B} = \mathrm{A}^{-1}$.

The necessary and sufficient condition for the pseudo-inverse of A to exist is that it should be of full rank. When there are more observations than the sources (i.e. $\mathrm{M} > \mathrm{N}$), there exist many matrices B which satisfy the condition $\mathrm{BA} = \mathrm{I}$. Here the choice B depends on the components of S that we are interested in. When the number of observations is less than the number of sources (i.e. $\mathrm{M} < \mathrm{N}$), a solution does not exist unless further assumptions are made.

On the other side of the problem, if there is no prior knowledge of the mixing matrix A then the estimation of both A and $\mathbf{S}$ is known as the blind source separation (BSS) problem. The problem defined in (2.1) under the assumption of negligible Gaussian noise $\mathbf{n}$ is solvable with the following restrictions.

- The sources (i.e. the components of $\mathbf{S}$) are statistically independent.
- At most one of the sources is Gaussian distributed

- The mixing matrix is of full rank.

A very popular technique for solution of the BSS problem is independent component analysis [2.4]. Estimation of the underlying independent sources is the primary objective of the BSS problem. Above discussion leads to following definitions of ICA.

Definition1: ICA is a linear transformation $\mathbf{S} = \mathrm{W}\mathbf{X}$ of a multivariate signal $\mathbf{X}$, such that the components of $\mathbf{S}$ are as independent as possible in the sense of maximizing some objective function $f(s_1,\dots,s_N)$, which is a measure of statistical independence.

Definition2: ICA can be defined as computationally efficient statistical signal processing technique for separating a multivariate signal into its components assuming that all of these components are statistically independent.

2.2. Statistical Independence

The above discussions make it clear that statistical independence is the key foundation of independent component analysis. For the case of two different random variables x and y, x is independent of the value of y if knowing the value of y does not give any information on the value of x. Statistical independence is defined mathematically in terms of the probability densities as: the random variables x and y are said to be independent if and only if

$$p_{x,y}(x,y) = p_x(x)p_y(y) \tag{2.5}$$

where $p_{x,y}(x,y)$ is the joint density of x and y, $p_x(x)$ and $p_y(y)$ are marginal probability densities of x and y respectively. Marginal probability density function of x is defined as

$$p_x(x) = \int p_{x,y}(x,y)\,dy \tag{2.6}$$

Generalizing this for a random vector $s = [s_1, \ldots, s_N]^T$ with multivariate density $p(s)$ has statistically independent components if the density can be factorized as

$$p(s) = \prod_{i=1}^{N} p_i(s_i) \tag{2.7}$$

Otherway stated, the density of s_1 is unaffected by s_2 when two variables s_1 and s_2 are independent. Statistical independence is a much stronger property than uncorrelatedness which takes into account the second order statistics only. If the variables are independent, they are uncorrelated but the converse is not true.

2.3. Central Limit Theorem

The central limit theorem is the most popular theorem in statistical theory and plays a predominant role in ICA. According to it let

$$x_k = \sum_{i=1}^{K} z_i \tag{2.8}$$

be a partial sum of sequence $\{z_i\}$ of independent and identically distributed random variables z_i. Since the mean and variance of x_k can grow without bound as $k \to \infty$, consider the standardized variables y_k instead of x_k,

$$y_k = \frac{x_k - m_{x_k}}{\sigma_{x_k}} \tag{2.9}$$

where m_{x_k} and σ_{x_k} are mean and variance of x_k. The distribution of y_k converges to a Gaussian distribution with zero mean and unit variance when $k \to \infty$.

This theorem has implicit consequences in ICA and BSS. A typical mixture or component of the data vector $\mathbf{x}$ is of the form

$$x_i = \sum_{j=1}^{M} a_{ij} s_j \tag{2.10}$$

where $a_{ij},\ j = 1,\ldots,M$ are constant mixing coefficients and $s_j,\ j = 1,\ldots,M$ are the M unknown source signals. Even for a fairly small number of sources the distribution of the mixture x_k is usually close to Gaussian. In a very simple way, the central limit theorem can be stated as the sum of even two independent identically distributed random variables is more Gaussian than the original random variables. This implies that independent random variables are more non-gaussian than their mixtures. Hence non-gaussianity is independence. This is one of the bases of independent component analysis.

2.4. Contrast Functions for ICA

The data model for independent component analysis is estimated by formulating an objective function and then minimizing or maximizing it. Such a function is often called a contrast function or cost function or objective function. The optimization of the contrast function enables the estimation of the independent components. The ICA method combines the choice of an objective function and an optimization algorithm. The statistical properties like consistency, asymptotic variance, and robustness of the ICA technique depend on the choice of the objective function and the algorithmic properties like convergence speed, memory requirements, and numerical stability depend on the optimization algorithm. The contrast function in some way or other is a measure of independence. In this section different measures of independence is discussed which is frequently used as contrast functions for ICA.

2.4.1 Measuring Nongaussianity

Kurtosis:

Central limit theorem discussed above provides a good intuitive idea that nongaussian is independent. So nongaussianity is a strong measure of independence.

The first quantitative measure of nongaussianity is kurtosis which is the fourth order moment of random data. Given some random data $\mathbf{y}$ the kurtosis of $\mathbf{y}$ denoted by $\mathrm{kurt(y)}$ is defined as

$$\mathrm{kurt(y)} = \mathrm{E}\{\mathrm{y}^4\} - 3\mathrm{E}\{\mathrm{y}^2\} \qquad (2.11)$$

where $E\{.\}$ is the statistical expectation operator. For simplicity if we assume y to be normalized so that the variance is equal to unity i.e. $\mathrm{E}\{\mathrm{y}^2\} = 1$, then $\mathrm{kurt(y)} = \mathrm{E}\{\mathrm{y}^4\} - 3$. This indicates kurtosis is simply the normalized version of the fourth moment $\mathrm{E}\{\mathrm{y}^4\}$.

For a Gaussian $\mathbf{y}$, the fourth moment equals to $3(\mathrm{E}\{\mathrm{y}^2\})^2$. So for Gaussian random variables the kurtosis value is zero and for nongaussian random variables kurtosis value is non-zero. Particularly when kurtosis value is positive the random variables are called *supergaussian* or *leptokurtic* and when negative called *subgaussian* or *platykurtic*. Supergaussian random variables have a ‘spiky’ probability density function with heavy tails and subgaussian random variables have a flat probability density function. However kurtosis is very sensitive to outliers in data set and this is a limitation of kurtosis as the contrast function.

Negentropy:

A second optimal quantitative measure of nongaussianity is negentropy which is based on the information theoretic differential entropy. The entropy of data is related to the information that is observed. The more random and unpredictable the data is the larger entropy it will have. The entropy S of a random variable y with a density of $\mathrm{p_y(\eta)}$ is

$$\mathrm{S(y)} = -\int \mathrm{p_y(\eta) log p_y(\eta) dp_y(\eta)} \qquad (2.12)$$

Of all the observed random variables with unit variance a Gaussian variable has the largest entropy value. Differential entropy normalized with respect to Gaussian variables result in negentropy. Negentropy of $\mathbf{y}$ denoted by $\mathrm{H(y)}$ is defined as

$$\mathrm{H(y)=S(y_{gausss})-S(y)} \qquad (2.13)$$

where $\mathbf{y_{gauss}}$ is a Gaussian random variable with the same correlation and covariance as $\mathbf{y}$. Since the negentropy is normalized, it is always nonnegative and zero if $\mathbf{y}$ is Gaussian distributed. Negentropy has the additional interesting property that it is invariant for invertible linear transformations.

Approximations to negentropy:

However negentropy is practically difficult to compute and require complex computation. Hence the approximations to negentropy in terms of expectation of certain nonlinear function G is expressed as

$$\mathrm{H(y)=K[E\{G(y_i)\}-E\{G(\nu)\}]^2} \qquad (2.14)$$

where K is a constant and ν is a Gaussian variable of zero mean and unit variance. Wise choice of G makes a good contrast function $\mathrm{H(y)}$ for optimization in ICA. Particularly if G is chosen such that it does not grow too fast then more robust estimators are obtained. The frequent choices of G that have proved useful are

$$G_1(y)=\frac{1}{a_1}\log\cosh(a_1y) \qquad (2.15)$$

$$\mathrm{G_2(y)=-\frac{1}{a_2}\exp(-a_2y^2/2)} \qquad (2.16)$$

$$\mathrm{G_3(y)=\frac{1}{4}y^4} \qquad (2.17)$$

where $\mathrm{a_1}$ and $\mathrm{a_2}$ are constants.

Under the approximation

$$E\{(w_i^T z)(w_j^T)\} = \delta_{ij} \tag{2.18}$$

where δ_{ij} is known as Kronecker delta function.

$\mathrm{H(y)}$ expression in equation simplifies to

$$\mathrm{H(w) = E\{G(w^T z)\}} \tag{2.19}$$

which is a good contrast function for optimization in ICA problems.

2.4.2 Mutual Information

Mutual information is natural measure of dependency between random variables i.e. it is a measure of the information that member of a set of random variables have on the other random variable in the set.

If y is a n-dimensional random variable and $\mathrm{p_y(\eta)}$ its probability density function then vector y has mutually independent components if and only if

$$\mathrm{p_y(\eta) = p_{y_1}(\eta_1).p_{y_2}(\eta_2)..p_{y_n}(\eta_n)} \tag{2.20}$$

A natural way of checking whether $\mathbf{y}$ has ICs is to measure a distance between both sides of above equation

$$\mathrm{I(p_y) = \delta(p_y, \prod p_{y_i})} \tag{2.21}$$

Average mutual information of $\mathbf{y}$ as given by Comon [2.5] is given by

$$\mathrm{I(p_y) = \int p_y(\eta)\log(\frac{p_y(\eta)}{\prod p_y(\eta)})d\eta} \tag{2.22}$$

Average mutual information vanishes if and only if the variables are mutually independent and strictly positive otherwise. In terms of negentropy mutual information is written as

$$I(y_1, y_2, ..., y_n) = H(y) - \sum H(y_i) \tag{2.23}$$

But the contrast functions based on mutual information discussed above require the estimation of the density function and this has severely restricted the use of these contrast functions.

Before these optimization functions are used in ICA optimization algorithm the observed data is processed as described in the following section.

2.5. Preprocessing of Data for ICA

Generally ICA is performed on multidimensional data. This data may be corrupted by noise, and several original dimensions of data may contain only noise. So if ICA is performed on a high dimensional data then this may lead to poor results due to the fact that such data contain very few latent components. Hence, reduction of the dimensionality of the data is a preprocessing technique that is carried prior to ICA. Thus finding a principal subspace where the data exist reduces the noise. Besides this when number of parameters is larger compared to the number of data points, the estimation of those parameters becomes very difficult and often lead to over-learning. Over learning in ICA typically produces estimates of the independent components that have a single spike or bump and are practically zero everywhere else [2.5]. This is because in the space of source signals of unit variance, nongaussianity is more or less maximized by such spike / bump signals.

Apart from reducing the dimension the observed signals are centered and decorrelated. The observed signal **X** is centered by subtracting its mean:

$$\mathbf{X} \leftarrow \mathbf{X} - E\{\mathbf{X}\} \tag{2.24}$$

Second-order dependences are removed by decorrelation which is achieved by the principal component analysis (PCA) [2.6, 2.7]. ICA problem is greatly simplified if the observed mixture vectors are first whitened. A zero-mean random vector $\mathbf{z} = (z_i \dots z_j)^T$ is said to be white if its elements z_i are uncorrelated and have unit variances $E\{z_i z_j\} = \delta_{i,j}$

In terms of Covariance matrix, above equation can be restated as,

$$E\{zz^T\} = I \quad (2.25)$$

where I is the identity matrix. A synonymous term for white is sphered. If the density of the vector **z** is radially symmetric and suitably scaled, then it is sphered, but converse is not always true. Because whitening is essentially decorrelation followed by scaling for which the PCA technique can be used. The problem of whitening now: Given a random vector x with n elements, we have to have a linear transformation V into another vector z such that

$$\mathrm{z} = \mathrm{V\,x} \quad (2.26)$$

is white or sphered.

Suppose $\mathrm{E} = [\mathrm{e}_{1.........}\mathrm{e}_{\mathrm{n}}]$ is the matrix whose columns are the unit-norm eigenvectors of the covariance matrix $\mathrm{C}_{\mathrm{x}} = \mathrm{E}\{\mathrm{xx}^{\mathrm{T}}\}$ and $\mathrm{D} = \mathrm{diag}[\mathrm{d}_1........\mathrm{d}_{\mathrm{n}}]$ is the diagonal matrix of the eigenvalues of C_{x} then $\mathrm{C}_{\mathrm{x}} = \mathrm{E\,D\,E}^{\mathrm{T}}$.This is called the eigenvectors decomposition of the covariance matrix. The linear whitening transform is expressed as

$$\mathrm{V} = \mathrm{D}^{-\frac{1}{2}}\,\mathrm{E}^{\mathrm{T}} \quad (2.27)$$

Hence $\mathrm{V} = \mathrm{D}^{-\frac{1}{2}}\,\mathrm{E}^{\mathrm{T}}\,\mathrm{x}$ (2.28)

Now ICA estimation is performed on the whitened data z instead of original data x. For whitened data it is sufficient to find an orthogonal separation matrix if the independent components are assumed white.

Dimensionality reduction by PCA is carried on by projecting the N dimensional data to a lower dimensional space spanned by m $(\mathrm{m} < \mathrm{N})$ dominant eigenvectors (i.e. eigenvectors corresponding to large eigenvectors) of the correlation matrix C_x.The eigenvectors matrix E and the diagonal matrix of eigenvectors D are of dimension $\mathrm{N} \times \mathrm{m}$ and $\mathrm{m} \times \mathrm{m}$ respectively. Practically it is a non-trivial task to identify the lower dimensional subspace properly. For noise free data, a subspace corresponding to the non-zero eigenvalues is required to be found.

In most of the scenario data are corrupted by noise and are not contained exactly within the subspace. In this case the eigenvectors corresponding to the largest eigenvalues should describe the data well; however, in general "weak" independent components may be lost in dimension reduction process. This involves a hit and trial process.

Dimensionality reduction can also be accomplished by methods other than PCA. These methods include local PCA [2.8] and random projection. For noise reduction another popular technique called principal factor analysis [2.9] is used.

The unmixing matrix B in Fig.2.1 can be regarded as two step process i.e. whitening and rotation. Hence

$$\mathrm{B} = \mathrm{W}^{\mathrm{T}}\,\mathrm{V} \tag{2.29}$$

The whitening matrix $\mathrm{V} = \mathrm{D}^{-1/2}\,\mathrm{E}^{\mathrm{T}}$ is estimated by PCA and rotation matrix W is found by one of the ICA techniques described in the following section. This process can be depicted in figure 3 below.

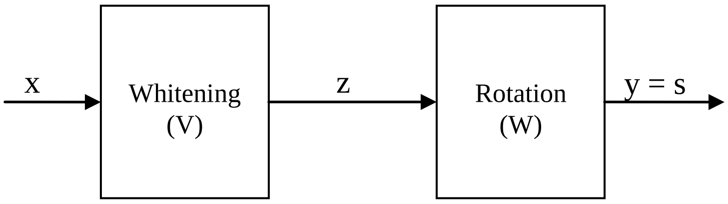

Fig. 2.3 Schematic of separation: whitening and rotation. The unmixing matrix B in figure 1 can be regarded as a concatenation of the whitening matrix V and the (orthogonal) rotation matrix W

2.6. Algorithms for ICA

Some of the ICA algorithms require a preprocessing of data X and some may not. Algorithms those need no preprocessing (centering and whitening), often converge better with whitened data. However in certain cases if it is necessary then sphered data Z is used otherwise no mention of sphering is done for cases where whitened is not required.

2.6.1 Non-linear cross correlation based Algorithm:

Principle of cancellation of non-linear cross correlation is used to estimate independent components in [2.10, 2.11]. Non-linear cross correlations are of the form $E\{g_1(y_i)g_2(y_j)\}$, where g_1 and g_2 are some suitably chosen nonlinearities. If y_i and y_j are independent, then these cross correlations are zero for y_i and y_j having symmetric densities. The objective function in such cases is formulated implicitly and exact objective function may not even exist. Jutten and Herault in [2.5] used this principle to update the nondiagonal terms of the matrix W according to

$$\Delta W_{ij} \propto g_1(y_i)\, g_2(y_j) \text{ for } i \neq j \tag{2.30}$$

Here y_i are computed at every iteration as $y = (I + W)^{-1} z$ and the diagonal terms W_{ii} are set to zero. After convergence y_i give the estimates of the independent components. However the algorithm converges only under severe restrictions [2.12].

2.6.2 Nonlinear Decorrelation Algorithm

To reduce the computational overhead by avoiding matrix inversions in Jutten-Herault algorithm and improve stability some algorithm has been proposed in [2.13]. In those the following algorithm has been proposed

$$\Delta W \propto (I - g_1(y)\, g_2(y^T))\, W \tag{2.31}$$

where $y = W x$, the nonlinearities $g_1(.)$and $g_2(.)$are applied separately on every component of the vector y, and the identity matrix can be replaced by any positive definite diagonal matrix. EASI algorithm has been proposed in [2.14, 2.15].

According to EASI,

$$\Delta W \propto (I - y\, y^T - g(y)\, y^T + y\, g(y^T))\, W \tag{2.32}$$

The choice of the nonlinearities used in above rules is generally provided by the maximum likelihood (or infomax) approach.

2.6.3 Infomax Estimation or Maximum Likelihood Algorithm

Maximum likelihood (ML) estimation is based on the assumption that the unknown parameters to be estimated θ are constants or no prior information is available on them. When the number of samples is large ML estimator becomes a desirable choice due to its asymptotic optimality properties. This ML estimation can be simply interpreted as those parameters having highest probability for the observations act as the estimates. The simplest algorithm for maximizing the likelihood (also log-likelihood) is given by Bell and Sejnowski [2.16] by using stochastic gradient methods. The algorithm for ML estimation derived by Bell and Sejnowski in [2.16] is

$$\Delta W \propto [W^T]^{-1} + E\{g(Wx)\, x^T\} \tag{2.33}$$

Here the nonlinearity g is very often chosen as $\tanh$ function because it is the derivative of log density of the logistic distribution. This function works for estimation of most super-gaussian independent components; however other functions should be used for subgaussian independent components. The convergence of the algorithm described by above equation is very slow especially due to the inversion of the matrix W that is needed at every step. The convergence of the algorithm can be improved by whitening the data, and especially by using the natural gradient.

The natural (or relative) gradient method simplifies the maximization of the likelihood and makes it better conditioned. The natural gradient principle is based on the geometrical structure of parameter space. This is related to the relative gradient principle, which uses the Lie group structure of the ICA problem. In the case of basic ICA, both these principles amount to multiplying the right side of above equation by $W^T W$. This gives

$$\Delta W \propto (I + E\{g(y)\, y^T\})\, W \tag{2.34}$$

where $y = Wx$. After this modification the algorithm needs no sphering. This algorithm can be interpreted as a special case of nonlinear decorrelation algorithm described in pervious section.

A Newton method for maximizing the likelihood has been introduced in [2.26]. Though it converges with less iteration but suffers from the problem that a matrix inversion is needed in each iteration.

Infomax principle [2.16] is a very closely related maximum likelihood estimation principle for ICA [2.17]. This is based on maximizing the output entropy or information flow of a neural network with nonlinear outputs. Hence it is named as infomax.

2.6.4 Nonlinear PCA Algorithm

Another approach to ICA that is related to PCA is the so called non-linear representation is sought for the input data that minimizes a least mean square error criterion. For linear case principal components are obtained and in some cases the nonlinear PCA approach gives independent components instead. In [2.10] the following version of a hierarchical PCA learning rule is introduced.

$$\Delta w \propto g(y_i)x - g(y_i)\sum_{j=1}^{i} g(y_i)w_j \tag{2.35}$$

where g is a suitable non-linear scalar function. The introduction of non-linearities means that the learning rule uses higher order information in the learning. In [2.18] it is proven that for well chosen non-linearities, the learning rule in above equation does indeed perform ICA, if the data is whitened. Algorithms for exactly maximizing the nonlinear PCA criteria are introduced in [2.11].

2.6.5 One-unit neural learning rules:

Simple algorithms from the one-unit contrast functions can be derived using the principle of stochastic gradient descent. Considering whitened data Hebbian like learning rule [2.19, 2.20] is obtained by taking instantaneous gradient of contrast function with respect to w

$$\Delta w \propto [E\{G(w^T x)\} - E\{G(\nu)\}] x g(w^T x) \tag{2.36}$$

Such one unit algorithms were first introduced is [2.21] using kurtosis. For estimation of several independent components of system of several units is needed.

2.6.6 Tensor based ICA Algorithm

Another approach for the estimation of independent components consists of using higher-order cumulant tensors. Tensors are generalizations of matrices, or linear operators. Cumulant tensors are then generalizations of the covariance matrix C_x. The covariance matrix is the second order cumulant tensor, and the fourth order tensor is defined by fourth-order cumulants as $\mathrm{cum}(x_i, x_j, x_k, x_l)$. Eigenvalue decomposition (EVD) is used to whiten the data. By whitening, the data is transformed so that its second-order correlations are zero. This principle can be generalized so that the off-diagonal elements of the fourth-order cumulant tensor can be minimized. This kind of (approximate) higher-order decorrelation results in a class of methods for ICA estimation.

Joint approximate diagonalization of eigenmatrices (JADE) proposed by Cardoso [2.6] is based on the principle of computing several cumulant tensors $F(M_i)$, where F represents the cumulant tensor and M_i represents the eigenmatrices. These tensors are diagonalized jointly as well as possible. If a matrix W diagonalizes $F(M)$ for any M then $W\,F(M)\,W^T$ is diagonal since the matrix F is a linear combination of the terms $w_i w_i^T$, assuming that the ICA model in equation (1) holds. A measure of the diagonality of $Q = WF(M_i)W^T$ is the sum of squares of the off-diagonal elements $\sum_{k\neq l} q_{kl}^2$. In other words, since the matrix W is orthogonal and it does not change the total sum of squares of a matrix, minimization of the sum of squares of the off-diagonal elements is equivalent to the maximization of the squares of the diagonal elements. Thus, the following function can be a good measure of the joint diagonalization process.

$$J_{JADE}(W) = \sum_i \left\| \mathrm{diag}\left(WF(M_i)W^T\right) \right\|^2 \qquad (2.37)$$

This represents the sum of squares of all the diagonal elements of all the diagonalized cumulant tensors.

M_i are chosen as the eigenmatrices of the cumulant tensor because the n eigenmatrices span the same subspace as the cumulant tensor, and hence they contain all the relevant information on the cumulants. With this choice the contrast function expressed in above equation can be restated as

$$J_{JADE}(W) = \sum_{ijkl\neq iikl} \left(\mathrm{cum}(y_i, y_j, y_k, y_l)\right)^2 \qquad (2.38)$$

where y is the estimate of the independent sources obtained as $y = Wx$. Above equation means that by minimizing J_{JADE}, sum of squared cross cumulants of

y_i is also minimized. But JADE is restricted to small dimensions mostly due to the computational complexity of explicit tensor EVD. Its statistical properties are inferior to methods using likelihood or non-polynomial cumulants [2.22]. However, with low dimensional data, JADE is a competitive alternative to most popular FastICA algorithm described in the next section.

A similar approach that uses the EVD is the fourth-order blind identification (FOBI) method [2.7] which is simpler, and deals with the EVD of the weighted correlation matrix. It is of reasonable complexity, and is probably the most efficient of all the ICA methods. However, it fails to separate the sources when they have identical kurtosis. Other approaches include maximization of squared cumulants [2.23], and fourth-order cumulant based methods as described in [2.24, 2.25].

2.6.7 Fast ICA Algorithm

One of the most popular solutions for linear ICA/BSS problem is Fast ICA [2.26] due to its simplicity and fast convergence. The basic algorithm involves the preprocessing and a fixed-point iteration scheme for one unit.

Fixed-point Iteration for one unit:

The fast ICA algorithm for one unit estimates one row of the demixing matrix W as a vector w^{T} that is an extremum of contrast functions. FastICA [2.26, 2.19] is an iterative fixed point algorithm, derived from a general objective or a contrast function. Assume x is the whitened data vector and w^{T} is one of the rows of the rotation/separating matrix W. Estimation of w proceeds iteratively with following steps until a convergence as stated below is achieved.

1) Choose an initial random vector w of unit norm.

2) $W \leftarrow E\{zg(w^T z)\} - E\{g'(w^T z)\}w$ (2.39)

where $\mathrm{g}_1(\mathrm{y}) = \mathrm{y}^3$ (derivative of kurtosis),

$$\mathrm{g}_2(\mathrm{y}) = \tanh(\mathrm{ay}), 1 \le \mathrm{a} \le 2$$

and $\mathrm{g}'(\mathrm{y})$ are the corresponding derivatives.

3) $w \leftarrow w / \| w \|$ where ||w|| is the norm of w.

4) if $\left| \mathrm{w}_{\mathrm{old}} - \mathrm{w}_{\mathrm{new}} \right| \leq \varepsilon$ is not satisfied then go back to step 2 where ε is a convergence parameter (~10^{-4}) and $\mathrm{w}_{\mathrm{old}}$ is the value of w before it's replacement by the newly calculated value $\mathrm{w}_{\mathrm{new}}$.

Fixed-point Iteration for Several units:

The independent components (ICs) can be estimated one by one — deflationary approach — or can be estimated simultaneously — symmetric approach. In the deflationary approach, it must be ensured that the rows w_{j} of the separating matrix W are orthogonal. This can be done after every iteration step by subtracting from the current estimate w_{p} the projections of all previously estimated $\mathrm{p}-1$ vectors before normalization.

$$\mathrm{w}_{\mathrm{p}} \leftarrow \mathrm{w}_{\mathrm{p}} - \sum_{\mathrm{j}=1}^{\mathrm{p}-1} (\mathrm{w}_{\mathrm{p}}^{\mathrm{T}} \mathrm{w}_{\mathrm{j}}) \mathrm{w}_{\mathrm{j}} \tag{2.40}$$

In the symmetric approach the iteration step is computed for all w_p and after the matrix W is orthogonalized as

$$\mathrm{W} \leftarrow (\mathrm{W}\mathrm{W}^{\mathrm{T}})^{-1/2}\, \mathrm{W} \tag{2.41}$$

The convergence properties of the FastICA algorithm are discussed in [2.26, 2.27]. The asymptotic convergence of the algorithm is at least quadratic and usually cubic when the ICA model (1) holds. This rate is much faster than that of gradient-based optimization algorithms. With a kurtosis based contrast function, FastICA can be shown to converge globally to the independent components [2.19].

2.6.8 Algebraic ICA Algorithm

An algebraic solution to ICA is proposed by Taro Yamaguchi et al. in [2.28]. This is a non-iterative algorithm but becomes extremely complex to compute when the number of sources goes more than two. For two sources separation it works very fast .Two observed signals x_1 and x_2 are given by linear mixture of two independent original signals s_1 and s_2 as:

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 & \alpha \\ \beta & 1 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} \tag{2.42}$$

where α and β are unknown mixing rates.

The algebraic solution to α and β are given by

$$\beta = \frac{\alpha C_2 - C_3}{\alpha C_3 - C_1} \tag{2.43}$$

$$\begin{aligned} &(C_2C_{10} - C_{11}C_3)\alpha^4 + (3C_9C_3 - 3C_8C_2 - C_3C_{10} + C_1C_{11})\alpha^3 \\ &+ (3C_6C_2 + 3C_8C_3 - 3C_9C_1 - 3C_7C_3)\alpha^2 \\ &+ (C_5C_3 + 3C_7C_1 - 3C_6C_3 - 3C_2C_4)\alpha \\ &+ (C_3C_4 - C_1C_5) = 0 \end{aligned} \tag{2.44}$$

where

$$C_1 = E[x_1^2] - \{E[x_1]\}^2$$

$$C_2 = E[x_2^2] - \{E[x_2]\}^2$$

$$C_3 = E[x_1x_2] - E[x_1]E[x_2]$$

$$C_4 = E[x_1^4] - E[x_1^3]E[x_1]$$

$$C_5 = E[x_1^3x_2] - E[x_1^3]E[x_2]$$

$$C_6 = E[x_1^3x_2] - E[x_1^2x_2]E[x_1]$$

$$C_7 = E[x_1^2x_2^2] - E[x_1^2x_2]E[x_2]$$

$$C_8 = E[x_1^2x_2^2] - E[x_1x_2^2]E[x_1]$$

$$C_9 = E[x_1x_2^3] - E[x_1x_2^2]E[x_2]$$

$$C_{10} = E[x_1x_2^3] - E[x_1]E[x_2^3]$$

$$C_{11} = E[x_2^4] - E[x_2^3]E[x_2] \tag{2.45}$$

Where $E[.]$ denotes the expectation operation.

α and β are obtained by solving the equations (2.43, 2.44, 2.45) with the Ferrari method. Excluding the solutions having non-zero imaginary parts and negative sizes the proper solution is selected. Original independent signals are computed from equation (2.42) by solving value of α and β.

2.6.9 Evolutionary ICA Algorithm

Evolutionary computation techniques are very popular population search based optimization methods. Genetic Algorithms, Swarm intelligence are the most used evolutionary computation based optimization techniques. By evolutionary mechanism, GA can search for the optimal separating matrix that minimizes the dependence. Instead of updating the matrix by a fixed formula, GA transforms a population of individuals into a new population using genetic operators based on fitness function. However the success of GA relies on the definition of fitness function. The population based search methods like GA converge to a global optimum unlike the case of gradient based methods which gets trapped in local optima. GA has been used for nonlinear blind source separation in [2.29, 2.30] and for noise separation from electrocardiogram signals in [2.31]. Particle swarm optimization (PSO) is used in ICA technique in [2.32]. Currently, several biologically motivated optimization algorithms are also used in ICA method. However the price paid by evolutionary computation based ICA techniques is the heavy computational complexity of the methods. But with the advent of highly parallel processors these methods provide competitive solutions to the problems.

2.6.10 Some Extensions to ICA Algorithm

2.6.10.1 Noisy ICA Algorithm

The estimation of the noiseless model seems to be a challenging task in itself, and thus the noise is usually neglected in order to obtain tractable and simple results. Moreover, it may be unrealistic in many cases to assume that the data could be divided into signals and noise in any meaningful way. Perhaps the most promising approach to noisy ICA is given by bias removal techniques. This means that noise-free ICA methods are modified so that the bias due to noise is removed or at least removed. In [2.33] bias reduction is performed by modifying the natural gradient ascent for likelihood. The new concept of Gaussian moments is introduced in [2.34] to derive one-unit contrast functions and to obtain a version of the fast ICA algorithm that has no asymptotic bias i.e. is consistent even in the presence of noise. These techniques can even be used in large dimensions. In [2.35], J. Cao et al. have proposed a robust approach for independent component analysis (ICA) of signals that observations are contaminated with high-level additive noise and/or outliers.

2.6.10.2 Complex ICA Algorithm

Separation of complex valued signals is a frequently arising problem in signal processing. For example, separation of convolutively mixed source signals involves computations on complex valued signals. The FastICA algorithm can be extended to complex valued signals. In [2.36], it is assumed that the original, complex valued source signals are mutually statistically independent, and the problem is solved by the independent component analysis (ICA) model.

2.6.10.3 Nonlinear ICA Algorithms

In most of the practical cases the linear mixtures pass though a certain type of nonlinearity before being actually observed. Most often the observing sensor introduces the nonlinearity by itself. So ICA must perform the separation from these observed nonlinear mixtures. The case of ICA for post nonlinear mixtures has been an area of interest for researchers [2.29, 2.30].

2.7. Ambiguities of ICA

2.7.1 Permutation Ambiguity

The order of independent components can not be determined. The linear noise free version of ICA model can be represented as

$$X = \sum_{i=1}^{N} a_i s_i = AS \tag{2.46}$$

Now, both A and S being unknown, the order of the terms can be changed freely in above equation and any of the independent components can be called the first one. This implies that the correspondence between a physical signal and the estimated independent component is not one-to-one. This indeterminacy is particularly severe in many applications where identification of the estimated components is of very high importance. Formally, this means that the following relation between the mixing matrix A and the separation matrix B holds

$$AB = P \tag{2.47}$$

where P is a permutation matrix.

2.7.2 Scaling Ambiguity

The energy of the independent components can not be determined. Since both A and S are unknown, the effect of multiplication of one of the source estimates with a scalar constant k is canceled by dividing its corresponding column in the mixing matrix by k. This indeterminacy can be solved by ensuring that the random variables have unit variance i.e,

$$E\{s_i^2\} = 1 \tag{2.48}$$

This still leaves the ambiguity of sign. While this is insignificant in certain applications, care has to be taken in applications where sign plays a crucial role.

2.8. Applications of ICA

ICA being a blind statistical signal processing technique finds application in many emerging new application areas such as blind separation of mixed voices or images [2.37, 2.38], analysis of several types of data [2.5], feature extraction [2.12], speech and image recognition [2.39, 2.17], data communication [2.40], sensor signal processing [2.41, 2.14], system identification[2.42, 2.43], biomedical signal processing [2.44, 2. 45, 2.13, 2.29] and several others [2.30, 2.46].

2.8.1 Biomedical Signal Processing

Magnetoencephalography (MEG) is a noninvasive technique by which the activity or the cortical neurons can be measured with very good temporal resolution and moderate spatial resolution. When using a MEG record, as a research or clinical tool, the investigator may face a problem of extracting the essential features of the neuromagnetic signals in the presence of artifacts. The amplitude of the disturbances may be higher than that of the brain signals, and the artifacts may resemble pathological signals in shape. In [2.47], a new method to separate brain activity from artifacts using ICA has been introduced.

2.8.2 Telecommunications

Another emerging application area of great potential is telecommunications. An example of a real-world communications application where blind separation techniques are useful is the separation of the user’s own signal from the interfering other users’ signals in CDMA (Code-Division Multiple Access) mobile communications [2.48]. This problem is semi-blind in the sense that certain additional prior information is available on the CDMA data model. But the number of parameters to be estimated is often so high that suitable blind source separation techniques taking into account the available prior knowledge provide a clear performance improvement over more traditional estimation techniques.

2.8.3 Revealing Hidden Factors in Financial Data

It is a tempting alternative to try ICA on financial data. There are many situations in that application domain in which parallel time series are available, such as currency exchange rates or daily returns of stocks, that may have some common underlying factors. ICA might reveal some driving mechanisms that otherwise remain hidden. In a recent study of a stock portfolio [2.49], it was found that ICA is a complementary tool to PCA, allowing the underlying structure of the data to be more readily observed.

2.8.4 Natural Image Denoising

Bell and Sejnowski proposed a method to extract features from natural scenes by assuming a linear image synthesis model [2.50]. In such a model each patch of an image is a linear combination of several underlying basis functions. A set of digitized natural images were used. Denote the vector of pixel gray levels in an image window by $\mathbf{x}$. Note that, multivalued time series or images changing with time are not considered here; instead the elements of $\mathbf{x}$ are indexed by the location in the image window or patch. The sample windows were taken at random locations. The 2-D structure of the windows is of no significance here: row by row scanning was used to turn a square image window into a vector of pixel values. Each window corresponds to one of the columns $\mathbf{a}_i$ of the mixing matrix $\mathbf{A}$. Thus an observed image window is a superposition of these windows with independent coefficients [2.50].

Now, suppose a noisy image model holds:

$$z = x + n \tag{2.49}$$

where **n** is uncorrelated noise, with elements indexed in the image window in the same way as **x**, and **z** is the measured image window corrupted with noise. Let us further assume that **n** is Gaussian and **x** is non-Gaussian. There are many ways to clean the noise; one example is to make a transformation to spatial frequency space by DFT, do low-pass filtering, and return to the image space by IDFT. This is not very efficient, however. A better method is the recently introduced Wavelet Shrinkage method [2.51] in which a transform based on wavelets is used, or methods based on median filtering. None of these methods is explicitly taking advantage of the image statistics, however.

2.8.5 Feature Extraction

ICA is successfully used for face recognition and lipreading. The goal in face recognition is to train a system that can recognize and classify familiar faces given a different image of the trained face. The test images may show the faces in a different pose or under different lighting conditions. Traditional methods for face recognition have employed PCA-like methods. Bartlett and Sejnowski [2.52] compare the face recognition performance of PCA and ICA for two different tasks: (1) different pose and (2) different lighting condition. They show that for both tasks ICA outperforms PCA. The method is roughly as follows: The rows of the face images constitute the data matrix x. Performing ICA, a transformation W is learned so that u ($u = Wx$) represent the independent face images. Nearest neighbor classification is performed on the coefficients of u.

2.8.6 Sensor Signal Processing

A sensor network is a very recent, widely applicable and challenging field of research. As size and cost of sensors decrease, sensor networks are increasingly becoming an attractive method to collect information in a given area. Multi-sensor data often presents complementary information about the region surveyed and data fusion provides an effective method to enable comparison, interpretation and analysis of such data. Image and video fusion is a sub area of the more general topic of data fusion dealing with image and video data. Cvejic et. al. [2.41] have applied independent component analysis for improving the fusion of multimodal surveillance images in sensor networks. ICA is also used for robust automatic speech recognition [2.53].

2.9. Conclusions

In this chapter the basic principle behind the independent component analysis technique is discussed. The contrast functions for different routes to independence are clearly depicted. Different existing algorithms for ICA are briefly illustrated and are critically examined with special reference to their algorithmic properties. The ambiguities present in these algorithms are also presented. Finally the application domains of this novel technique are presented. Some of the futuristic works on ICA technique which need further investigation are development of nonlinear ICA algorithms, design of low complexity ICA algorithms and use of evolutionary computing optimization tools for developing ICA and finally alleviation of permutation and scaling ambiguities existing in present ICA.

References

[2.1] P. Comon, “Independent Component Analysis-A new concept?” *Signal Processing*, vol-36, pp.287-314, 1994.

[2.2] J.F.Cardoso, “Blind Signal Separation: Statistical Principles”, *Proc. of IEEE*, vol-9, No.10, pp. 2009-2025, 1998.

[2.3] S.Z.Li, et al., “Learning Multiview Face Subspaces and Facial Pose Estimation using Independent Component Analysis,” *IEEE Trans. Image Processing*, vol-14, No.6, pp.705-712, June 2005.

[2.4] Capizzi G., Coco S. and Laudani A., “A New Tool for the Identification and Localization of Electromagnetic Sources by Using Independent Component Analysis”, *IEEE Trans. On Magnetics,* vol.-43, Issue-4, pp.1625-1628, Apr. 2007.

[2.5] C.Jutten and J.Herault, “Blind separation of sources, part I: An adaptive algorithm based on neuromimetric architecture,” *Signal Processing,* 24:1-10, 1991.

[2.6] J.F. Cardoso and A. Souloumiac. Blind beamforming for non Gaussian signals. *IEE Proceedings-F,* 140(6), pp.362–370, 1993.

[2.7] J.F.Cardoso “Source separation using higher order moments”, *Proc. of the IEEE Int. Conf on Acoustics, Speech and Signal Processing (ICASSP 1989),* pp. 2109–2112, Glasgow, UK, May 1989.

[2.8] K. Fukunaga and D. R. Olsen. “An algorithm for finding intrinsic dimensionality of data” *IEEE Transactions on Computers,* 202:176–183, 1971.

[2.9] D.T.Pham, P.Garrat and C.Jutten, “Separation of a mixture of independent sources through a maximum likelihood approach”, *Proc. EUSIPCO,* pp.771-774, 1992.

[2.10] E.Oja, H.Ogawa, and J.Wangviwattana, “Learning in nonlinear constrained Hebbian networks”, T.Kohonen et al., editor, Artificial Neural Networks, *Proc. ICANN’91,* pp. 385-390, Espoo, Finland, 1991.

[2.11] E.Oja, “Nonlinear PCA criterion and maximum likelihood in independent component analysis”, *Proc. Int. Workshop on Independent Component Analysis and Signal separation (ICA’99),* pp.143-148, Aussois, France, 1999.

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

[2.13] A.Cichocki et al., “Robust Neural Networks with on line biasing for blind identification and blind source separation”, *IEEE Trans. On Circuits and Systems,* vol-43, no.11, pp.894-906, 1996.

[2.14] J.F.Cardoso and B.H.Laheld, “Equivariant adaptive source separation”, *IEEE Trans. On Signal Processing,* vol-44, No.12, pp.3017-3030, 1996.

[2.15] B. Laheld and J.F.Cardoso, “Adaptive source separation with uniform performance”, *Proc. of EUSIPCO,* pp.183-186, Edinburgh, 1994.

[2.16] A.J.Bell and T.J.Sejnowski, "An information-maximization approach to blind separation and blind decovolution", *Neural Computation*, 7, pp.1129-1159, 1995.

[2.17] J.F.Cardoso, "Infomax and maximum likelihood for sources separation," *IEEE Letters on Signal Processing,* 4:112-114, 1997.

[2.18] H. Hotelling. "Analysis of a complex of statistical variables into principal components", *Journal of Educational Psychology,* 24:417–441, 498–520, 1993.

[2.19] A. Hyvarinen and E. Oja "A fast fixed-point algorithm for independent component analysis", *Neural Computation*, 9, pp.1483–1492, 1997.

[2.20] A.Hyverinen and E.Oja, "Independent component analysis by general nonlinear Hebbian like learning rates", *Signal Processing*, vol-64, No.3, pp.301-303, 1998.

[2.21] T.Kohonen, "*Self Organizing Maps"*, Springer-Verlag, Berlin, Heidelberg, New York, 1995.

[2.22] A. Hyverinen., J.Kahrunen. and E. Oja, " *Independent Component Analysis*", John Wiley & Sons, 2001.

[2.23] F. Herrmann and A. K. Nandi, "Maximisation of squared cumulants for blind source separation" *Electronics Letters,* 36(19), pp.1664–1665, 1996.

[2.24] A. K. Nandi and F. Herrmann, "Fourth-order cumulant based estimator for independent component analysis", *Electronic Letters,* 37(7), pp.469–470, 2001.

[2.25] A. K. Nandi and V. Zarzoso, "Fourth-order cumulant based blind source separation", *IEEE Signal Processing Letters*, 3(12), pp.312–314, 1996.

[2.26] A.Hyverinen, "Fast and Robust Fixed-point Algorithm for Independent Component Analysis", *IEEE Trans. on Neural Networks,* vol-10, No.3, pp.626-634, 1999a.

[2.27] E. Oja. "Convergence of the symmetrical FastICA algorithm", *Proc. of the 9th Int. Conf. on Neural Information Processing (ICONIP 2002),* Singapore, November 2002.

[2.28] T.Yamaguchi, I. Kuzuyoshi., "An Algebraic Solution to Independent Component Analysis", *Optics Communications, Elsevier Science,* 178, pp.59-64, 2000.

[2.29] Y.Tan and J.Wang, "Nonlinear Blind Source Separation Using Higher Order Statistics and a Genetic Algorithm," *IEEE Trans. On Evolutionary Computation*, vol.5, No.6, pp.600-611, Dec. 2001.

[2.30] F.Rojas, C.G.Puntonet, M.Rodriguez-Alvarez, I.Rojas, and R.Martin-Clemente, "Blind Source separation in post-nonlinear mixtures using competitive learning, simulated annealing and a genetic algorithm," *IEEE Trans. On Systems, Man and Cybernatics –Part C: Applications and Reviews,* vol.34, no.4, pp.407-416, Nov. 2004.

[2.31] R.Palaniappan and C.N.Gupta, "Genetic Algorithm based independent component analysis to separate noise from Electrocardiogram signals," *Proc. IEEE,* 2006.

[2.32] D. J. Krusienski and W. K. Jenkins, "Nonparametric Density Estimation Based Independent Component Analysis via Particle Swarm Optimization", *Proc. IEEE*, pp. IV – 357- IV – 360, 2005.

[2.33] S.C.Douglas, A.Cichocki and S. Amari, "A bias removal technique for blind source separation with noisy measurements", *Electronics Letters*, 34, pp.1379-1380, 1998.

[2.34] A.Hyvarinen, "Fast Independent Component analysis with noisy data using Gaussian moments", *Proc. Int. Symp. On Circuits and Systems*, Orlando, Florida, pp.57-61, 1999.

[2.35] J. Cao, N. Murata, S. Amari, A. Cichocki, and T. Takeda "A Robust Approach to Independent Component Analysis of Signals With High-Level Noise Measurements", *IEEE Trans. on Neural Networks*, vol. 14, No. 3, pp.631-645,May 2003

[2.36] E.Bingham and A. Hyvarinen, "A Fast fixed-point algorithm for independent component analysis of complex valued signals", *International Journal of Neural Systems,* Vol. 10, No. 1, pp.1-8, February, 2000.

[2.37] I.Dagher, R.Nachar, "Face Recognition using IPCA-ICA Algorithm," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol-28, No.6, pp.996-1000, June 2006.

[2.38] Keun-Chang Kwak, Pedrycz,W, "Face Recognition using Enhanced Independent Component Analysis Approach", *IEEE Trans. on Neural Networks,* vol-18, No-4, pp.530-541, Mar. 2007.

[2.39] Keun-Chang Kwak and W.Pedrycz, "Face Recognition using an Enhanced Independent Component Analysis Approach," *IEEE Trans. On Neural Networks*, vol-18, Issue-2, pp.1625-1628, Apr. 2007.

[2.40] E.Oja, H.Ogawa, and J. Wangviwattana, "Learning in nonlinear constrained Hebbian networks," In T.Kohonen et al.,editor, *Artificial Neural Networks,Proc.ICANN'91*, pp.385-390, Espoo, Finland, 1991. North-Holland, Amsterdam.

[2.41] Cvejic N., Bull D. and Canagarajah N., "Improving Fusion of Surveilliance Images in Sensor Networks using Independent Component Analysis", *IEEE Trans. On Consumer Electronics*, vol.53, Issue 3, pp.1029-1035, Aug.2007.

[2.42] Jun-Mei Yang, Sakai, H., "A New Adaptive Filter Algorithm for System Identification using Independent Component Analysis", *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 2007, vol-3, pp.III-1341-III-1344, Apr. 2007

[2.43] Ou Shifeng, Zhao Xiaohui, Gao Ying, "Linear System Identification Employing Independent Component Analysis", *IEEE Int. Conf. on Automation and Logistics*, 2007, pp.1336-1340, Aug.2007.

[2.44] Van Dun B., Wouters J. and Moonen M., "Improving Auditory Steady-State Response Detection Using Independent Component Analysis on Multichannel EEG Data", *IEEE Trans. On Biomedical Engineering*, vol.54, Issue 7, pp.1220-1230, July 2007.

[2.45] Waldert S, et.al."Real-Time fetal heart Monitoring in Biomagnetic Measurements Using Adaptive Real –Time ICA", *IEEE Trans. On Biomedical Engineering*, vol.54, Issue 107, pp.1864-1874, Oct. 2007.

[2.46] Vitria J, Bressan M. and Radeva P., “Bayesian Classification of Cork Stoppers Using Class-Conditional Independent Component Analysis”, *IEEE Trans. On Systems, Man and Cybernatics –Part C: Applications and Reviews*, vol.37, Issue.1, pp.32-38, Jan. 2007.

[2.47] R.Vigário, , J. Särelä, , and Oja, E., “Independent component analysis in wave decomposition of auditory evoked fields”, *Proc. Int. Conf. on Artificial Neural Networks*, pp. 287–292, Skövde, Sweden, 1998.

[2.48] T. Ristaniemi, and J. Joutsensalo, , “On the performance of blind source separation in CDMA downlink”, *Proc. Int. Workshop on Independent Component Analysis and Signal Separation*, pp. 437–441, Aussois, France, 1999.

[2.49] A. D.Back, and A. S. Weigend, , “A first application of independent component analysis to extracting structure from stock returns”, *Int. J. on Neural Systems*, 8(4), pp.473–484, 1997.

[2.50] A.J.Bell and T.J.Sejniwski, “The ‘Independent Components’ of natural scenes are edge filters”, *Vision Research*, 37(23), pp.3327-3338, 1997.

[2.51] D.L. Donoho, I.M.Johnstone, G.Kerkyacharian, and D.Picard, “Wavelet shrinkage: asymptopia?” *Journal of the Royal Statistical Society, Ser. B*, 57, pp.301–337, 1995.

[2.52] M.Bartlett and T.J.Sejniwski, “Viewpoint invariant face recognition using independent component analysis and attractor networks”, *Advances in Neural Information Processing Systems 9*, pp. 817-823, MIT Press, 1997.

[2.53] L.Potanutis et al., “Independent Component Analysis applied to Feature Extraction for Robust Automatic Speech Recognition,” *Electronics Lett.*, vol-36, No.23, pp.1977-1978, Nov.2000.

Chapter

III

DEVELOPMENT OF NOVEL CONSTRAINED GENETIC ALGORITHM BASED ICA ALGORITHM

3.1. Introduction

All the existing ICA methods do not find a global optimum solution once the algorithm reaches a local optimum. It gets stuck in the valley of the contrast function and is unable to jump the surrounding hills. The non linear objective functions for ICA being multimodal, this problem becomes more prominent. Therefore, appropriate initial values are important in running these algorithms. In addition to the problem of getting trapped in a local optimum, these algorithms have the ambiguities like scaling and permutation. This means that the independent components (ICs) are estimated only up to a multiplicative scalar constant and the order in which the ICs are estimated are not known. In [3.1] attempts have been made to overcome the permutation ambiguity in the frequency domain only but the ambiguity remains still unresolved in the time domain.

The performance of available neural learning algorithms relies on the selection of activation functions. Usually different activation functions are used for sources with different statistics (sub-Gaussian or super-Gaussian) which are difficult to estimate in blind source separation. Also most of the algorithms are under the condition, though not mentioned directly, that the sources are of the same statistics. Sub-Gaussian and super-Gaussian signals have different sign of fourth order cumulants (*kurtosis*), and this is not considered in the derivation of many of the neural learning algorithms. Therefore at times it may so happen that the learning algorithm fails to converge to a consistent optimum when the sources have different signs of kurtosis.

ICA algorithms attempt to maximize the estimated entropy of the system. The preeminent ICA algorithms primarily use gradient based techniques to perform the entropy maximization. These gradient based techniques are often misguidedly deemed acceptable for several reasons. For one, because of the inherent complexity of many ICA algorithms, additional complexity is avoided by incorporating a reliable gradient based algorithm with provable (albeit often suboptimal) local convergence. Also, this suboptimal performance may occur less frequently than expected because many global optima of the performance surface exist as scaled and permuted versions of the unmixing matrix. However, for example, it is well-known and commonly overlooked that neural networks such as implemented in variations of Infomax (and in general) are highly nonlinear and produce multimodal performance surfaces that do not lend themselves well to gradient based techniques. Likewise other ICA techniques contain nonlinear entropy estimation functions. Again, gradient based optimization will inevitably lead to suboptimal solutions on such nonlinear performance functions, which requires multiple restarts to avoid a meaningless unmixing matrix. In addition, the number of local minima of the performance surface can dramatically increase when the unknown sources have multimodal distributions.

The evolutionary computation based optimization techniques like Genetic Algorithms (GA) have an inherent characteristic to converge to the global optimum. In a preliminary study done by Yoshioka *et al.* [3.2] GA is applied to separate original images from noise corrupted images by directly minimizing Kullback Leibler (KL) divergence. However prior probability estimation of the sources is a prerequisite of the method. Tan and Wang in [3.3] used GA to solve the nonlinear BSS problem using higher order statistics where the sources have been estimated regardless of the indeterminacies of permutation and scaling. Rajas et al. in [3.4] applied the GA based method for signal separation from their post nonlinear mixtures. The guided GA is very nicely applied to optimize the blind source separation on post-nonlinear mixtures by Gorriz *et. al.* in [3.5]. GA has been directly applied to ICA problem for first time in [3.6, 3.7] for denoising the electrocardiogram (ECG) signals where the method estimates only one independent component i.e. the ECG signal. However all the applications of GA to BSS problem still have the permutation indeterminacy. In practical applications where we need all the signals or independent components to be recovered, simple application of GA seems to be inadequate.

In this Chapter we propose a novel Independent Component Analysis algorithm using optimization based on the constrained GA (CGAICA) which recovers all the ICs using the popular deflation approach. Our method overcomes the long standing permutation ambiguity and recovers the ICs in a fixed order which is dependent on the statistical characteristics of the signals to be estimated. It offers almost comparable performance to the most popular fast ICA algorithm. The limitation of the technique is also discussed in this Chapter.

3.2. Constrained GA based ICA Algorithm

3.2.1 The Genetic Algorithm:

Concisely stated, a genetic algorithm (or GA for short) is a programming technique that mimics biological evolution as a problem-solving strategy. Given a specific problem to solve, the input to the GA is a set of potential solutions to that problem, encoded in some fashion, and a metric called a fitness function that allows each candidate to be quantitatively evaluated. These candidates may be solutions already known to work, with the aim of the GA being to improve them, but more often they are generated at random. The GA then evaluates each candidate according to the fitness function. In a pool of randomly generated candidates, of course, most will not work at all, and these will be deleted. However, purely by chance, a few may hold promise - they may show activity, even if only weak and imperfect activity, toward solving the problem. These promising candidates are kept and allowed to reproduce. Multiple copies are made of them, but the copies are not perfect; random changes are introduced during the copying process. These digital offspring then go on to the next generation, forming a new pool of candidate solutions, and are subjected to a second round of fitness evaluation. Those candidate solutions which were worsened, or made no better, by the changes to their code are again deleted; but again, purely by chance, the random variations introduced into the population may have improved some individuals, making them into better, more complete or more efficient solutions to the problem at hand. Again these winning individuals are selected and copied over into the next generation with random changes, and the process repeats. The expectation is that the average fitness of the population will increase each round, and so by repeating this process for hundreds or thousands of rounds, very good solutions to the problem can be obtained.

As astonishing and counterintuitive as it may seem to some, genetic algorithms have proven to be an enormously powerful and successful problem-solving strategy, dramatically demonstrating the power of evolutionary principles. Genetic algorithms have been used in a wide variety of fields to evolve solutions to problems as difficult as or more difficult than those faced by human designers. Moreover, the solutions they come up with are often more efficient, more elegant, or more complex than anything comparable a human engineer would produce. In some cases, genetic algorithms have come up with solutions that baffle the programmers who wrote the algorithms in the first place.

3.2.2 Methods of representation

Before a genetic algorithm can be put to work on any problem, a method is needed to encode potential solutions to that problem in a form that a computer can process. One common approach is to encode solutions as binary strings: sequences of 1's and 0's, where the digit at each position represents the value of some aspect of the solution. Another, similar approach is to encode solutions as arrays of integers or decimal numbers, with each position again representing some particular aspect of the solution. This approach allows for greater precision and complexity than the comparatively restricted method of using binary numbers only and very often is intuitively closer to the problem space [3.8].

This technique was used, for example, in the work of Steffen Schulze-Kremer, who wrote a genetic algorithm to predict the three-dimensional structure of a protein based on the sequence of amino acids that go into it [3.9]. Schulze-Kremer's GA used real-valued numbers to represent the so-called "torsion angles" between the peptide bonds that connect amino acids. (A protein is made up of a sequence of basic building blocks called amino acids, which are joined together like the links in a chain. Once all the amino acids are linked, the protein folds up into a complex three-dimensional shape based on which amino acids attract each other and which ones repel each other. The shape of a protein determines its function.) Genetic algorithms for training neural networks often use this method of encoding also.

A third approach is to represent individuals in a GA as strings of letters, where each letter again stands for a specific aspect of the solution. One example of this technique is Hiroaki Kitano's "grammatical encoding" approach, where a GA was put to the task of evolving a simple set of rules called a context-free grammar that was in turn used to generate neural networks for a variety of problems [3.9].

The virtue of all three of these methods is that they make it easy to define operators that cause the random changes in the selected candidates: flip a 0 to a 1 or vice versa, add or subtract from the value of a number by a randomly chosen amount, or change one letter to another. Another strategy, developed principally by John Koza of Stanford University and called genetic programming, represents programs as branching data structures called trees [3.10]. In this approach, random changes can be brought about by changing the operator or altering the value at a given node in the tree, or replacing one subtree with another as in Fig.3.1.

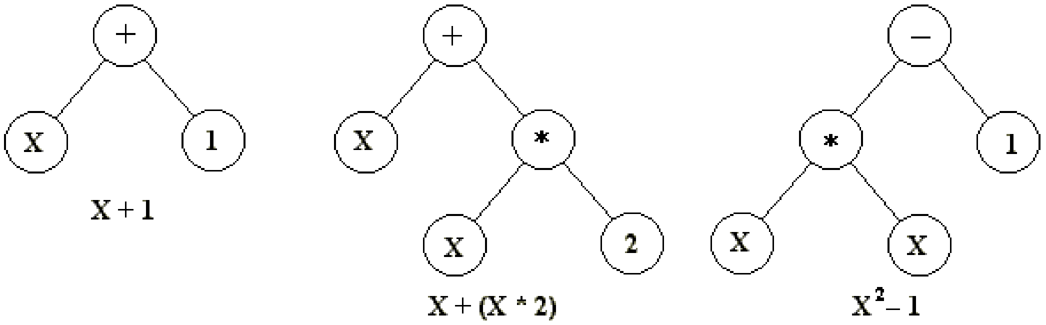

Fig.3.1 Three simple program trees of the kind normally used in genetic programming. The mathematical expression that each one represents is given underneath.

It is important to note that evolutionary algorithms do not need to represent candidate solutions as data strings of fixed length. Some do represent them in this way, but others do not; for example, Kitano's grammatical encoding mentioned above can be efficiently scaled to create large and complex neural networks, and Koza's genetic programming trees can grow arbitrarily large as necessary to solve whatever problem they are applied to.

3.2.3 Methods of selection

There are many different techniques which a genetic algorithm can use to select the individuals to be copied over into the next generation, but listed below are some of the most common methods. Some of these methods are mutually exclusive, but others can be and often are used in combination.

Elitist selection:

The most fit members of each generation are guaranteed to be selected. (Most GAs do not use pure elitism, but instead use a modified form where the single best, or a few of the best, individuals from each generation are copied into the next generation just in case nothing better turns up.)

Fitness-proportionate selection:

More fit individuals are more likely, but not certain, to be selected.

Roulette-wheel selection:

A form of fitness-proportionate selection in which the chance of an individual's being selected is proportional to the amount by which its fitness is greater or less than its competitors' fitness. (Conceptually, this can be represented as a game of roulette - each individual gets a slice of the wheel, but more fit ones get larger slices than less fit ones. The wheel is then spun, and whichever individual "owns" the section on which it lands each time is chosen.)

Scaling selection:

As the average fitness of the population increases, the strength of the selective pressure also increases and the fitness function becomes more discriminating. This method can be helpful in making the best selection later on when all individuals have relatively high fitness and only small differences in fitness distinguish one from another.

Tournament selection:

Subgroups of individuals are chosen from the larger population, and members of each subgroup compete against each other. Only one individual from each subgroup is chosen to reproduce.

Rank selection:

Each individual in the population is assigned a numerical rank based on fitness, and selection is based on these ranking rather than absolute differences in fitness. The advantage of this method is that it can prevent very fit individuals from gaining dominance early at the expense of less fit ones, which would reduce the population's genetic diversity and might hinder attempts to find an acceptable solution.

Generational selection:

The offspring of the individuals selected from each generation become the entire next generation. No individuals are retained between generations.

Steady-state selection:

The offspring of the individuals selected from each generation go back into the pre-existing gene pool, replacing some of the less fit members of the previous generation. Some individuals are retained between generations.

Hierarchical selection:

Individuals go through multiple rounds of selection each generation. Lower-level evaluations are faster and less discriminating, while those that survive to higher levels are evaluated more rigorously. The advantage of this method is that it reduces overall computation time by using faster, less selective evaluation to weed out the majority of individuals that show little or no promise, and only subjecting those who survive this initial test to more rigorous and more computationally expensive fitness evaluation.

3.2.4 Methods of change

Once selection has chosen fit individuals, they must be randomly altered in hopes of improving their fitness for the next generation. There are two basic strategies to accomplish this. The first and simplest is called *mutation*. Just as mutation in living things changes one gene to another, so mutation in a genetic algorithm causes small alterations at single points in an individual's code.

The second method is called *crossover*, and entails choosing two individuals to swap segments of their code, producing artificial "offspring" that are combinations of their parents. This process is intended to simulate the analogous process of recombination that occurs to chromosomes during sexual reproduction. Common forms of crossover include *single-point crossover*, in which a point of exchange is set at a random location in the two individuals' genomes, and one individual contributes all its code from before that point and the other contributes all its code from after that point to produce an offspring. The second form of crossover is *uniform crossover*, in which the value at any given location in the offspring's genome is either the value of one parent's genome at that location or the value of the other parent's genome at that location, chosen with 50/50 probability.

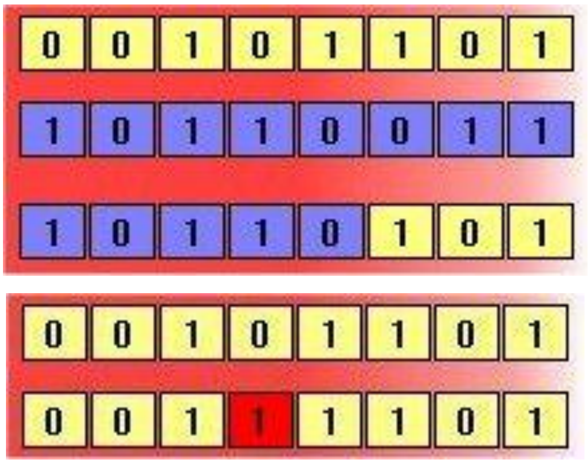

Fig.3.2 Crossover and mutation.

The above diagrams in Fig.3.2 illustrate the effect of each of these genetic operators on individuals in a population of 8-bit strings. The upper diagram shows two individuals undergoing single-point crossover; the point of exchange is set between the fifth and sixth positions in the genome, producing a new individual that is a hybrid of its progenitors. The second diagram shows an individual undergoing mutation at position 4, changing the 0 at that position in its genome to a 1.

3.3. The Proposed Constrained GA based ICA Algorithm

The operations like encoding of parameters, initialization of population, mate selection, crossover, mutation and population replacement constitute a canonical GA. These operations of a GA are described in the following manner

Encoding: The parameters of the independent component analysis system to be optimized are generally encoded into genes and chromosomes (also called individuals) as a string of binary digits using one’s complement representation. The parameters are assumed to be bounded in the region.

$$|\theta_k| \le b_k \text{ for } k = 1,\ldots\ldots,h$$

where h represents the number of parameters. The length of the gene and individuals can be computed as the length of the binary string B_k to be encoded θ_k based on b_k and the desired accuracy. Other encoding methods are also possible.

Initial population generation: The initial population is generated randomly in the range of each parameter. Therefore, at the beginning of the separating procedure, *N* individuals are generated as random binary string.

Evaluation of fitness: After the initial population generations, the fitness of each individual is determined. Fitness is a numeric index to measure the effectiveness of each individuals of the population as a solution, which is usually utilized to select members of the population for reproduction. For the ICA problem, we can define the fitness function based on cost functions discussed in Chapter 2. Most widely used *kurtosis* defined in equation (2.11) is considered as the fitness function for GA.

Selection Operation: A pair of individuals is selected from the current population for mating using tournament selection.

Crossover Operation: A multipoint crossover with probability is applied to the newly selected (parents) individuals to generate two offspring. Specifically, the number of crossover points in our application is equal to the number of the parameters to be optimized.

Mutation Operation: Random mutation operator is applied to the newly generated offspring to prevent from premature convergence. It randomly alters the gene from zero to one or from one to zero with a probability expressed by, , where is called mutation probability.

A random column vector w which is represented as two chromosomes is used to find the linear transformation $w^T z$. The following steps represent the constrained GA based ICA algorithm.

Step-1: Data Centering

The mean $\mathbf{X_m} = (x_{1m}, x_{2m} \ldots, x_{nm})^T$ of the observed mixed signal data $\mathbf{X} = (x_1, x_2 \ldots, x_n)^T$ is computed and the mean is subtracted from the observed data set to make it zero mean.

$$\mathbf{X_c} = \mathbf{X} - \mathbf{X}_m \tag{3.1}$$

Step-2: Whitening

The covariance matrix CovX of the centered data $\mathbf{X_c}$ is computed. The eigenvalue decomposition of CovX is performed. If D is the eigenvalue matrix and E is the eigenvector matrix then

$$\mathrm{Z} = \mathrm{D}^{-1/2}\mathrm{E}^{*}\,\mathrm{X_c} \tag{3.2}$$

Step -3: GA-based ICA Iteration

The following iterative procedure is followed to implement GA based ICA algorithm.

1) An initial population $\{\hat{\theta}_i\}\underset{i=1}{N}$ of size N is created from a random initial set of parameter. The encoding length of each parameter is 15 bits. By decoding the individual to get the parameter of the system, the fitness for each individual is evaluated.

2) Two mates are selected for reproduction with probabilities proportional to their using tournament selection.

3) The multipoint crossover operator with crossover probability P_c is applied to the two mates and a pair of offspring is generated.

4) The mutation operator with probability $\mathrm{P_m}$ is applied to the newly generated offspring.

5) The fitness value for the off spring are computed after they are decoded as the parameter sets of the parametric system.

6) GA steps from 2 to 5 are repeated until an entirely new population of individuals is generated.

7) The previous population is replaced with the new population with the addition of an elitist selection.

8) If the stopping criterion is satisfied, go to GA step 11.

9) If generation number is greater than a predetermined value go to GA step 2.

10) Reinitialize the population survival, go to GA step 2.

11) Output the individual with the best fitness value and terminate the iterative procedure.

Step-4: Evaluation of Second Independent Component

To estimate the other ICs, step 3 of the algorithm is repeated for getting weight vectors $w_2,\ldots,w_n$. To prevent different vectors from converging to the same optimum and hence the same IC, the weight vectors are decorrelated using Gram-Schmidt like orthogonaliztion. When p vectors $w_1,\ldots,w_p$ have been estimated, step 3 is run for w_{p+1} and after every iteration step the following iteration steps are performed.

$$w_{p+1} = w_{p+1} - \sum_{j=1}^{p} (w_{p+1}^T w_j) w_j \tag{3.3}$$

$$w_{p+1} = \frac{w_{p+1}}{\sqrt{w_{p+1}^T w_{p+1}}}$$

(3.4)

Above equations constrain the GA based optimization process.

3.4. Simulation Experiment

In the experimental studies for the verification of the validity and performance of the proposed constrained GA optimization based ICA algorithm, programs for separating the signals blindly from their observed mixtures are written. In the simulation environment, the two signals are mixed by a known matrix A and the mixed signals are the inputs to the CGAICA algorithm for separation. Two different examples are taken to verify the separation capability of the proposed algorithm. For a particular example, the parameters such as a number of chromosomes N_c, number of bits of binary coded GA N_b, number of generations N_{re}, probability of crossover P_c and probability of mutation P_m parameters are tuned, to get the proper separation. In this simulation for CGAICA we have considered the following typical values: $N_c = 8, p = 2, N_b = 15, N_{re} = 10, P_c = 0.85$ and $P_m = 0.01$.

The separation performance parameter, the mean square error is evaluated. For the first example, separation is performed by using contrast functions kurtosis.The minimum value of the reciprocal of fitness function J is plotted against the number of its evaluations.

3.5. Results & Discussions

The separation capability of CGAICA is verified through following three examples.

Example 1:

A random binary wave and a sine wave with 400 samples as shown in Fig.3.3 are mixed by the mixing matrix

$$A = \begin{pmatrix} 0.9121 & 0.2292 \\ 0.4763 & 0.7348 \end{pmatrix} \qquad (3.5)$$

Their mixtures are represented in Fig 3.4.

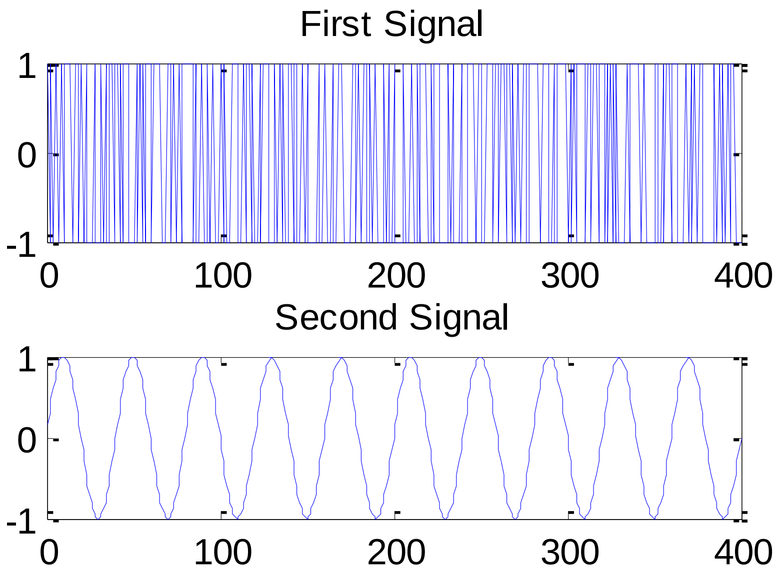

Fig.3.3 Original Signals

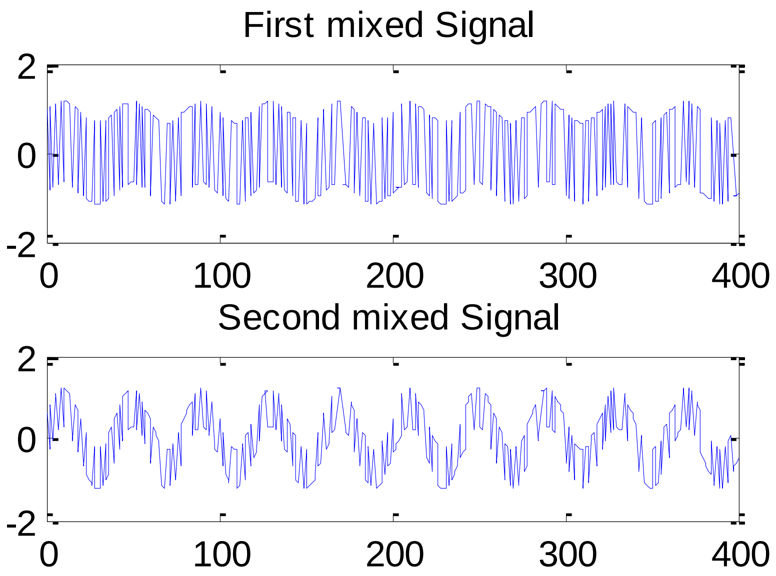

Fig.3.4 Mixer of Random Binary & Sine wave

Using CGAICA the two signals or ICs are recovered in the decreasing order of the value of their contrast function. For the case of kurtosis as contrast function the separated signals are depicted in Fig 3.5.

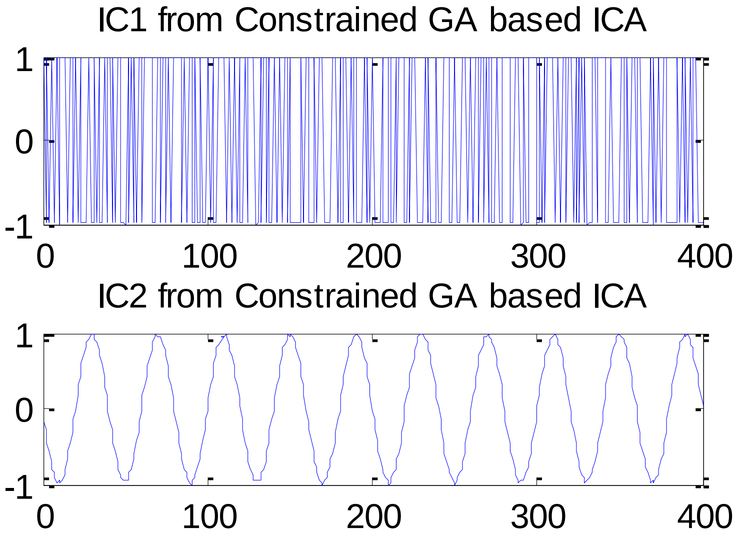

Fig.3.5 The Recovered Independent Components

Random binary signal has more kurtosis value than the sine wave used. So it is observed that the random binary is recovered first and the sine wave comes as the second IC. The same random binary signal and sine wave are considered for 400 samples each and mixed by mixing matrix A as in (3.5). Applying CGAICA algorithm the signals are separated clearly shown in Fig 3.5 in the case of 400 samples. Two sets of tuned values of the optimization parameters for CGAICA algorithm with kurtosis as the contrast function and the corresponding mean square error (MSE) for random binary signal is summarized in Table 3.1.

Table 3-1 Parameters of CGAICA to recover Sine wave and Random Binary wave from its mixtures

No of Chromosomes N_c	4	8	8	10	20
No of Generations N_{re}	5	10	20	20	20
No of Bits N_b	15	15	15	15	15
Probability of Crossover P_c	0.85	0.85	0.85	0.85	0.85
Probability of Mutation P_m	0.01	0.01	0.01	0.01	0.01
MSE for Random Binary IC	3.562×10^{-4}	4.588×10^{-5}	2.764×10^{-5}	1.226×10^{-6}	2.181×10^{-8}
MSE for Sine Wave IC	8.60×10^{-2}	8.61×10^{-2}	8.60×10^{-2}	8.58×10^{-2}	8.58×10^{-2}

Example 2:

In this example two speech signals (Fig. 3.6) with 50000 samples each are taken and instantaneously mixed by the artificial mixing matrix A as given in (3.5) which is shown in Fig 3.7.

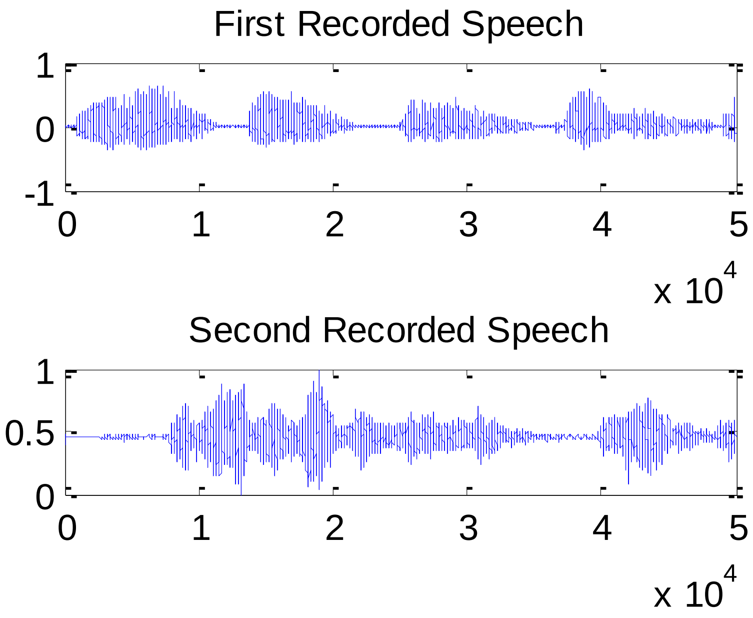

Fig.3.6 Recorded Speech Signals

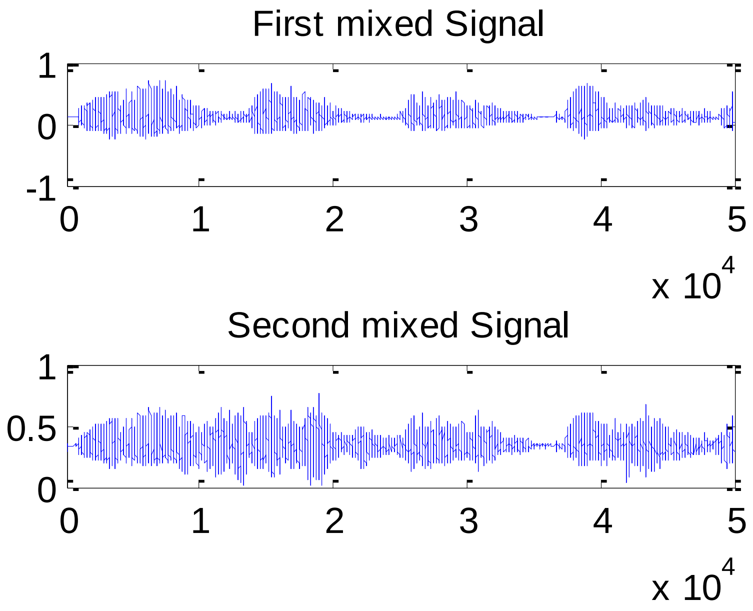

Fig.3.7 Speech Signals mixed with artificial mixing matrix

The mixed signals are applied to CGAICA algorithm for separation using kurtosis as the contrast function. The estimated speech signals show a clear separation from the mixtures which is depicted in Fig.3.8 first followed by the speech signal with lower value of kurtosis.

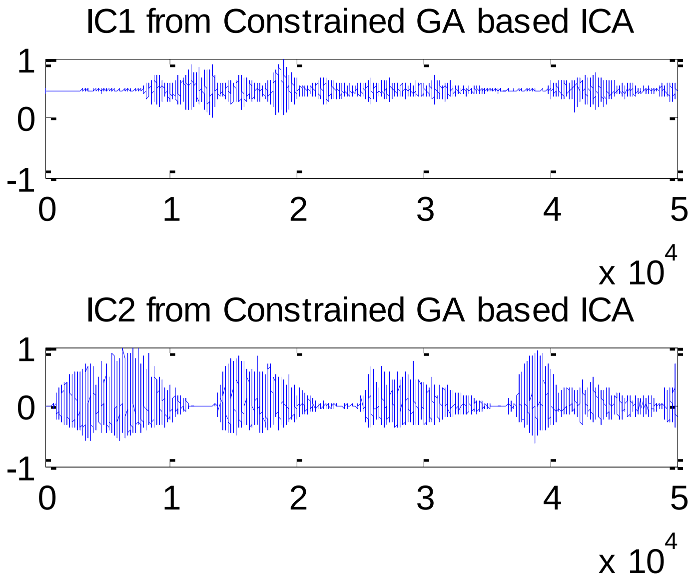

Fig.3.8 Speech Signals recovered by CGAICA

Example 3:

This example considers the case of a supergaussian (speech) signal and a subgaussian (random noise) signal shown in Fig 3.9. The mixtures of these signals are depicted in Fig.3.10. The signals recovered with CGAICA are shown in Fig.3.11.

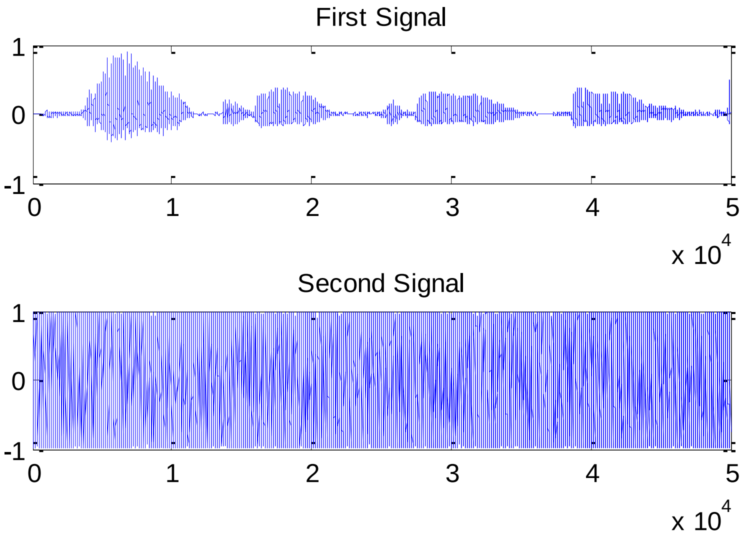

Fig.3.9 Original Speech Signal and random noise

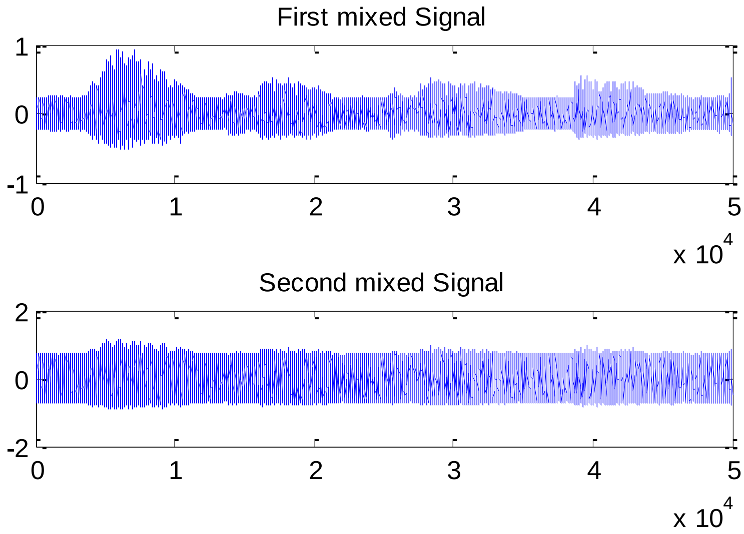

Fig.3.10 Mixture of Speech Signal and random noise

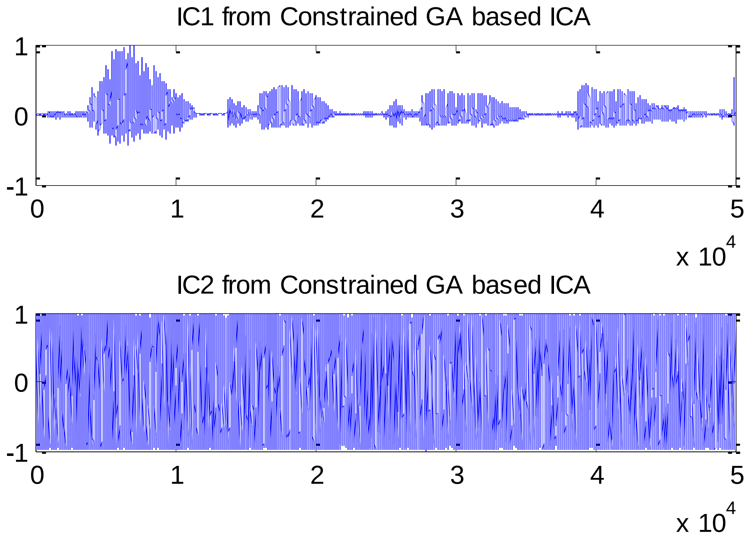

Fig.3.11 Speech Signal and random noise recovered by CGAICA

3.5.1 Comparison with Gradient Based Optimization Scheme

The signal separation as done in example 1 is also performed by the most popular fast ICA algorithm which is a gradient based scheme to have a comparison with our proposed CGAICA algorithm. Table 3.2 summarizes the typical MSE values estimated for random binary and sine wave components for fast ICA and

CGAICA algorithms. We have chosen the GA with 15 bits which is most commonly considered for binary GAs. If we increase the number of bits for binary coding then MSE performance improves significantly which is very much natural for binary GAs as with higher number of bits they tend towards real coded GA. It is clearly observed that CGAICA yields almost comparable performance as that of the most popular fast ICA algorithm. The MSE performance of fastICA algorithm varies with varying initialization.

Table 3-2 Comparison of fast ICA and CGAICA

Algorithm	MSE	
	Random Binary	Sine wave
Fast ICA	2.5004×10^{-9}	8.58×10^{-2}
CGAICA	2.1811×10^{-8}	8.58×10^{-2}

3.5.2 Convergence

In GA schemes, a *gene* is converged when 95% of the population shares the same value. To know the convergence of CGAICA, the variation of the best value of fitness function with the number of evaluations of the fitness function is studied. In the case of example 1 with same mixing matrix A, CGAICA is performed. Fig 3.12 shows the variation of the reciprocal of fitness function J values with the number of J evaluations for random binary component. Similar results are obtained by taking approximation to negentropy as a contrast function.

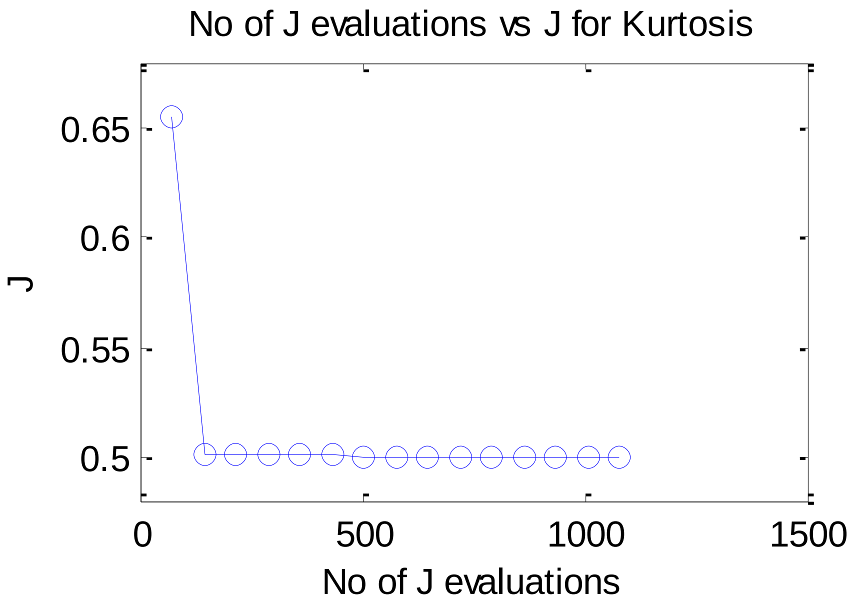

Fig. 3.12 Convergence characteristics of CGAICA

3.5.3 Permutation Ambiguity

From all the above two examples it is observed that using CGAICA independent components are recovered always in a fixed order in all runs of the simulation experiments. The IC for which the fitness function has a global maximum value appears first and then appears the IC with subsequent maximum value of the fitness function. So the order of the ICs can be predicted if the value of their statistical property like kurtosis is relatively known. Hence the permutation indeterminacy present in all ICA techniques seems to be dissolved by use of the proposed technique. This can be extended without loss of generalization to cases of ICA with any number of independent components.

But if the contrast function (kurtosis) values for both the components are almost equal then it becomes difficult to predict the order of recovered signal components and hence the permutation uncertainty of the independent components remains in such cases.

3.6. Conclusions

The Genetic Algorithm based optimization is used in a constrained manner to estimate the independent components from their observed mixtures. This scheme is tested using several examples including the speech signals for instantaneous mixing cases. From the simulation results it is very clear that in the CGAICA algorithm the MSE of the estimated ICs decreases with increase in the number of chromosomes. Also for a fixed number of chromosomes the number of generations affects the MSE. The CGAICA algorithm yields almost comparable MSE as obtained by the fast ICA algorithm. The permutation ambiguity present in ICA techniques is resolved by use of the CGAICA algorithm if we have relative knowledge of the statistical characteristics of the signals to be estimated. But this doesn’t resolve the uncertainty if the signals have nearly equal value of cost function. However care should be taken while adjusting the parameters for constrained Genetic Algorithm based optimization so that premature convergence to a local optimum does not occur. For this, the parameters should be initially varied over a large range to ensure the proper convergence to a global optimum.

References

[3.1]. H.Sawada, R.Mukai, S.Araki and S.Makino, “Solving Permutation and Circularity problem in Frequency-Domain Blind Source Separation,” *Proc. International Conf. on ICA*, Japan, 2004.

[3.2]. M.Yoshioka and S. Omatu, “Signal separation method using genetic algorithms,” *Proc. IEEE Int. Joint Conf. Neural Networks*, vol.2, pp.909-912, 1998.

[3.3]. Y.Tan and J.Wang, “Nonlinear Blind Source Separation Using Higher Order Statistics and a Genetic Algorithm,” *IEEE Trans. On Evolutionary Computation*, vol.5, No.6, pp.600-611, Dec. 2001.

[3.4]. F.Rojas, C.G.Puntonet, M.Rodriguez-Alvarez, I.Rojas, and R.Martin-Clemente, “Blind Source separation in post-nonlinear mixtures using competitive learning, simulated annealing and a genetic algorithm,” *IEEE Trans. On Systems, Man and Cybernatics –Part C: Applications and Reviews*, vol.34, no.4, pp.407-416, Nov.2004.

[3.5]. J.M.Gorriz, C.G.Puntonet, F.Rojas, R.Martin, S.Hornillo and E.W.Lang, “Optimizing blind source separation with guided genetic algorithms”, *Neurocomputing* , 69, pp.1442-1457, 2006.

[3.6]. R.Palaniappan and C.N.Gupta, “Genetic Algorithm based independent component analysis to separate noise from Electrocardiogram signals,” *Proc. IEEE* , 2006.

[3.7]. Xiang-Yan Zeng, Yen-Wei Chen, Zensho Nakao and Katsumi Yamashita, “Signal Separation by Independent Component Analysis based on a genetic algorithm”, *Proc. IEEE*, pp. 1688-1694, 2000.

[3.8]. P. Fleming and R.C. Purshouse. "Evolutionary algorithms in control systems engineering: a survey." *Control Engineering Practice*, vol.10, pp.1223-1241, 2002.

[3.9]. Mitchell, Melanie, “*An Introduction to Genetic Algorithms*”, MIT Press, 1996.

[3.10]. J.Koza, K.Martin and S.Maththew, “*Evolving Inventions*”, Scientific American, Feb. 2003.

Chapter

IV

DEVELOPMENT OF A NEW ICA ALGORITHM BASED ON CONSTRAINED BACTERIAL FORAGING OPTIMIZATION TECHNIQUE

4.1. Introduction

The optimization landscape of the contrast functions for ICA is very often multimodal. Secondly, the optimization complexity increases with increase in the number of independent components. In such a scenario one becomes tempted to use the population based search techniques as has been described in the previous Chapter with genetic algorithm based independent component analysis due to their global convergence capability.

The Bacterial Foraging Optimization (BFO) is a recently developed derivative free efficient evolutionary computational optimization technique which reveals the global optimum of the contrast function [4.1]. In this technique the foraging behavior of the *E.coli* bacteria present in our intestine has been mimicked. This novel scheme has been successfully used for several applications in [4.2, 4.3]. Since bacteria foraging optimization has been reported to have better performance than GA and GA has been applied successfully to ICA [4.4] it is quite motivating to study the application of bacteria foraging based optimization to ICA problem.

In the present work we propose a novel Independent Component Analysis algorithm using the evolutionary optimization based on the Bacterial Foraging and use the popular deflation approach to estimate all the independent components one by one. Using this algorithm in a constrained manner overcomes the long standing permutation ambiguity and recovers the ICs in a fixed order which is dependent on the statistical characteristics of the signals to be estimated. The Bacteria Foraging Optimization based ICA (BFOICA) has been shown to offer comparable or sometimes better performance as compared to the most popular fast ICA algorithm. The proposed BFOICA algorithm converges faster and yields better performance than the CGAICA algorithm described in previous Chapter. It is also successfully applied to separate signals from mixtures of supergaussian and subgaussian signals.

This Chapter is organized as follows. In Section 4.2, the bacteria foraging optimization based ICA algorithm is proposed. The simulation experiments have been carried out in Section 4.3 to show the effectiveness of the proposed ICA algorithm. In Section 4.4 the results of the investigation has been illustrated and discussed. Finally, in Section 4.5, the conclusion of the study has been outlined.

4.2. Bacterial Foraging Optimization Based ICA Algorithm

4.4.1 Bacterial Foraging Optimization

Bacterial foraging is a new evolutionary computational method proposed by Passino [4.1] where foraging can be modeled as an optimization process where an animal seeks to maximize energy obtained per unit time spent on foraging. Search strategies form the basic foundation for foraging decisions. Animals search for food and obtain nutrients in a way that maximizes E/T where E is energy obtained, and T is time spent on foraging (or they maximize long-term average rate of energy intake). Evolution optimizes foraging strategies since animals that have poor foraging performance do not survive.

Generally, a foraging strategy involves finding a "patch" of food (e.g., group of bushes with berries), deciding whether to enter it and search for food and when to leave the patch. There are predators and risks. energy required for travel. and physiological constraints (sensing, memory. cognitive capabilities). Foraging scenarios can be modeled and optimal policies can be found using, for instance dynamic programming. Search and optimal foraging decision-making of animals can be broken into three basic types: cruise (e.g.. tuna fish, hawks), saltatory (e.g.. birds. fish, lizards, and insects), and ambush (e.g., snakes, lions). In cruise search the animal searches the perimeter of a region, and in ambush it sits and waits. In saltatory search an animal typically moves in some direction, stops (or slows down), looks around, and then changes direction. It searches throughout a whole region.

Some animals forage as individuals and others forage as groups. While to perform social foraging an animal needs communication capabilities. It can gain advantages in that it can essentially exploit the sensing capabilities of the group, the group can "gang-up" on large prey, individuals can obtain protection from predators while in a group, and in a certain sense the group can forage with a type of collective intelligence. Social foragers include birds. bees. fish. ants. wildebeasts. and primates. Note that there is a type of "cognitive spectrum" where some foragers have little cognitive capability, and other higher life forms have significant capabilities (e.g.. compare the capabilities of a single ant with those of a human).

In this scheme, the foraging (methods for locating, handling and ingesting food) behavior of *E.coli*, which is a common type of bacteria present in our intestines is mimicked. The E. coli bacterium has a plasma embrane, cell wall, and capsule that contains the cytoplasm and nucleoid. The pili (singular, pilus) are used for a type of gene transfer to other E. coli bacteria, and flagella (singular, flagellum) are used for locomotion. The cell is about 1micron diameter and 2 micron in length. The E. coli cell only weighs about 1 picogram and is about 70% water. Salmonella typhimurium is a similar type of bacterium.

The E. coli bacterium is probably the best understood microorganism. Its entire genome has been sequenced; it contains 4,639,221 of the A, C, G, and T “letters”—adenosine, cytosine, guanine, and thymine—arranged into a total of 4,288 genes. Mutations in E. coli occur at a rate of about 10^{-7} per gene, per generation, and can affect its physiological aspects (e.g., reproductive efficiency at different temperatures). E. coli bacteria occasionally engage in a type of “sex” called "conjugation" where small gene sequences are unidirectionally transferred from one bacterium to another via an extended pilus. When E. coli grows, it gets longer, and then divides in the middle into two "daughters." Given sufficient food and held at the temperature of the human gut (one place where they live) of 37 ° C, E. coli can synthesize and replicate everything it needs to make a copy of itself in about 20 min; hence growth of a population of bacteria is exponential with a relatively short time to double. These bacteria undergo different stages such as chemotaxes, swarming, reproduction and elimination and dispersal.

Chemotaxes:

The motion patterns (called "*taxes*") that a bacteria generates in the presence of chemical attractants and repellants are called chemotaxes. This process is achieved through swimming and tumbling via flagella. An *E.coli* bacterium can move in two different ways; it can run (swim for a period of time) or it can tumble, and alternate between these two modes of operation in the entire lifetime. To represent a tumble, a unit length random direction, say $\phi(j)$, is generated; this will be used to define the direction of movement after a tumble.

In particular $\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i)\phi(j)$ (4.1)

Where $\theta^i(j,k,l)$ represent the ith bacterium jth chemotactic kth reproductive and lth elimination and dispersal step. $C(i)$ is the size of the step taken in the random direction specified by the tumble (run length unit).

Swarming:

When a group of *E.coli* cells is placed in the center of a semisolid agar with a single nutrient chemo-effecter (sensor), they move out from the center in a traveling ring of cells by moving up the nutrient gradient created by consumption of the nutrient by the group. Moreover, if high levels of succinate are used as the nutrient, then the cells release the attractant aspartate so that they congregate into group and hence, move as concentric patterns of groups with high bacterial density. The spatial order results from outward movement of the ring and the local releases of the attractant; the cells provide an attraction signal to each other so they swarm together.

Reproduction:

The least healthy bacteria die and the other healthier bacteria each split into two bacteria, which are placed in the same location. This makes the population of bacteria constant.

Elimination and Dispersal:

It is possible that the local environment, where a population of bacteria live changes either gradually (e.g. via consumption of nutrients) or suddenly due to some other influence. Events can occur such that all the bacteria in a region are killed or a group is dispersed into a new part of the environment .They have the effect of possibly destroying the chemotactic progress, but they also have the effect of assisting in chemotaxis, since dispersal may place bacteria near good food sources. From a broad perspective, elimination and dispersal are parts of the population-level long-distance motile behavior. The detailed mathematical treatment of this new concept is presented in [4.1].

4.4.2 The Proposed BFOICA Algorithm:

With Bacterial Foraging optimization algorithm we use kurtosis defined in (2.11) as the contrast (nutrient) functions to be maximized. The algorithm is made for minimization of the nutrient function J , so we consider

$$J = \frac{1}{Contrast\ Function} \tag{4.2}$$

Before presenting the observed mixed signal data for optimization the two preprocessing steps, centering and whitening, are performed on it.

Step-1: Preprocessing

The mean $X_m = (x_{1m}, x_{2m} \ldots, x_{nm})^T$ of the observed mixed signal data $X = (x_1, x_2 \ldots, x_n)^T$ is computed and the mean is subtracted from the observed data set to make it zero mean.

$$X_c = X - X_m \tag{4.3}$$

The covariance matrix $CovX_c$ of the centered data X_c is computed. The eigenvalue decomposition of $CovX_c$ is performed. If D is the eigenvalue matrix and E is the eigenvector matrix then

$$Z = D^{-1/2} E^* X_c \tag{4.4}$$

where $Z = (z_1, z_2, \ldots, z_n)^T$ represents the whitened observed mixed signal data. The random column vector w which is represented as the position P of the bacteria is used to find the linear transformation $w^T z$. The bacteria foraging technique finds the value of w at which $J(w^T z)$ is minimized.

Step-2: Initialization

The following parameters are initialized for the bacteria foraging optimization algorithm.

i) Number of parameters p to be optimized.

ii) Number of bacteria S to be used for searching the total region.

iii) Swimming length N_s after which tumbling of bacteria will be undertaken in a chemotatic loop.

iv) Number of iterations N_c to be under taken in a chemotatic loop $N_c > N_s$.

v) Maximum number of reproduction steps N_{re} to be undertaken.

vi)Maximum number of elimination and dispersal events N_{ed} to be imposed over the bacteria.

vii)Probability with which the elimination and dispersal P_{ed} will continue.

viii) The location of each bacterium $P(1-p,1-S,1)$ which is specified by random numbers on [0,1].

ix) The value of $C(i)$ which is assumed to be constant in our case for all the bacteria to simplify the design strategy.

Step-3 Iterative algorithm for optimization

This section models the bacterial population chemotaxis, reproduction, elimination and dispersal (initially, $j=k=l=0$). For the algorithm updating θ^i automatically results in updation of P

i) Elimination-dispersal loop: $l=l+1$

ii) Reproduction loop: $k=k+1$

iii) Chemotaxis loop: $j=j+1$

a) For $i=1,2,.......,S,$ calculate cost function value for each bacterium ***i*** as follows.

*Compute value of cost function $J(i,j,k,l)$.

*let $J_{last}=J(i,j,k,l,)$ to save his value since we may find a better cost via a run.

*End of for loop.

b) For $i=1,2,...S$ take the tumbling /swimming decision

*Tumble: Generate a random vector $\Delta(i)\in\Re^p$ with each element $\Delta_m(i)m=1,2,...p,$ a random number on [-1,1].

*Move: Let. $$\theta^i(j+1,k,l)=\theta^i(j,k,l)+C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \qquad (4.5)$$

Fixed step size in the direction of tumble for bacterium *i* is considered.

*compute $J(i,j+1,k,l)$

*Swim: i) Let $\mathbf{m}=\mathbf{0}$;(counter for swim length)

ii) while $m<N_s$ (have not climbed down too long)

*Let $m = m + 1$

*If $J(i, j+1, k, l) < J_{last}$ (if doing better),

let $J_{last} = J(i, j+1, k, l)$ and

let $\theta^{i}(j+1, k, l) = \theta^{i}(j, k, l) + C(i)\dfrac{\Delta(i)}{\sqrt{\Delta^{T}(i)\Delta(i)}}$ (4.6)

and use this $\theta^{i}(j+1, k, l)$ to compute the new $J(i, j+1, k, l)$.

*Else, let $m = N_s$.This is the end of the while statement.

c) Go to the next bacterium $(i+1)$ if $i \neq S$ (i.e go to b) to process the next bacterium.

iv) If $j < N_{c,}$ go to (iii).In this case, continue chemotaxis since the life of the bacteria is not over.

v) Reproduction:

a) For the given **k** and ***l***, and for each $i = 1, 2, \ldots S,$ let $J^{i}_{health} = \min\limits_{j \in \{1 \ldots N_s\}} \{ j\ (i, j, l) \}$ be the health of the bacterium ***I*** (a measure of how many nutrients it got over its life time and how successful it was at avoiding noxious substance).Sort bacteria in order of ascending cost J_{health} (higher cost means lower health).

b) The $S_r = S/2$ bacteria with highest J_{health} values die and other S_r bacteria with the best value split (and the copies that are made are placed at the same location as their parent)

vi) If $k < N_{re}$ go to (ii), in this case, we have not reached the number of specified reproduction steps, so we start the next generation in the chemotactic loop.

vii) Elimination & dispersal: $i = 1, 2, ..S$ with probability P_{ed} eliminate and disperse each bacterium (this keeps the number of bacteria the population constant) to a random location on the optimization domain.

The position of the bacteria w_1 at which global minimum value is obtained yields the first independent component.

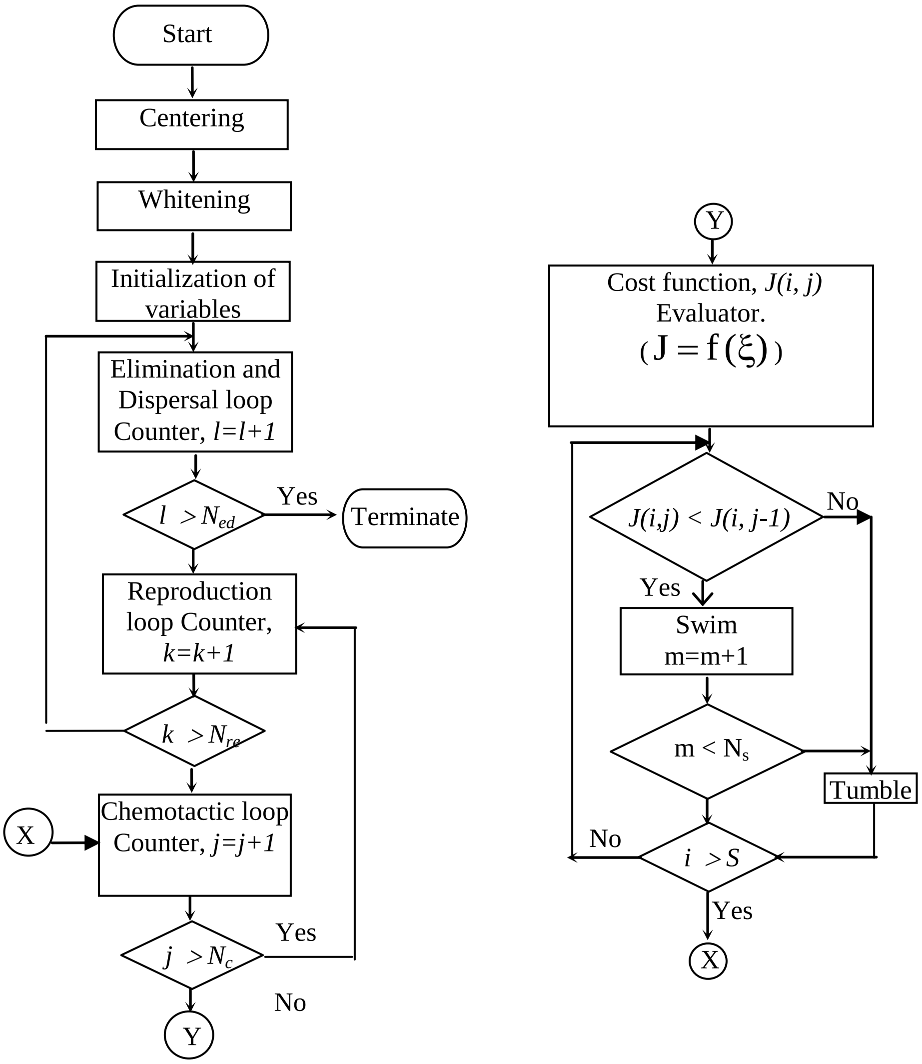

Fig. 4.1 Flowchart of the Bacteria Foraging based ICA Algorithm for estimation of one IC. This includes the preprocessing steps centering and whitening. The other independent components can be estimated by repeating the steps again in the orthogonal space.

Step-4: Evaluation of the Other Independent Components

To estimate the other ICs step 3 of the algorithm is repeated for getting weight vectors $w_2,...,w_n$. To prevent different vectors from converging to the same optimum and hence the same IC, the weight vectors are decorrelated using Gram-Schmidt like orthogonalization. When p vectors $w_1,...,w_p$ have been estimated, step 3 is run for w_{p+1} and after every iteration step the following iteration steps are performed.

$$\mathrm{w}_{p+1} = \mathrm{w}_{p+1} - \sum_{j=1}^{p}(\mathrm{w}_{p+1}^{T}\mathrm{w}_{j})\mathrm{w}_{j} \tag{4.7}$$

$$\mathrm{w}_{p+1} = \frac{\mathrm{w}_{p+1}}{\sqrt{\mathrm{w}_{p+1}^{T}\mathrm{w}_{p+1}}} \tag{4.8}$$

Above equations constrain the Bacteria Foraging Optimization process. The flowchart of the bacteria foraging based ICA algorithm is shown in Fig.1.

4.3. The Simulation Experiment

In the experimental studies for the verification of the validity and performance of the proposed bacteria foraging optimization based ICA algorithm, programs for separating the signals blindly from their observed mixtures are written in MATLAB. In the simulation environment, the two signals are mixed by a known matrix A and the mixed signals are the inputs to the BFOICA algorithm for separation. Several examples are taken to verify the separation capability of the proposed algorithm. For a particular example, the parameters such as a number of bacteria (S), number of chemotactic steps (N_c), number of elimination and dispersal events (N_{ed}), number of reproduction steps (N_{re}), probability of elemination and dispersal (P_{ed}) and runlengthunit parameters are tuned, to get the proper separation. In this simulation for BFOICA the following typical values:

$S = 8, p = 2, N_c = 8, N_s = 7, N_{re} = 4, N_{ed} = 4, P_{ed} = 0.25$ are used. The parameter tuning is not a rule based one in this case and is performed on a trial basis.

The separation performance parameters, the mean square error (MSE) and crosstalk (CT) are evaluated using (4.8) and (4.9).

$$\mathrm{MSE}_i = \frac{\sum_{t=1}^{N} \{s_i(t) - y_i(t)\}^2}{N} \tag{4.9}$$

$$\mathrm{CT}_i = 10\log\left(\frac{\sum_{t=1}^{N} \{s_i(t) - y_i(t)\}^2}{\sum_{t=1}^{N} \{s_i(t)\}^2}\right) \tag{4.10}$$

The MSE and crosstalk at the first glance appears to be very ambiguous performance indices in case of ICA where there are uncertainties due to scaling and permutation. However any standard index of performance has to encounter the difficulties due to these uncertainties. In the present study, the scaling uncertainty has been taken care of by normalization process and the permutation ambiguity has been overcomed. Hence MSE and crosstalk can serve as good performance indices for comparison. In all examples, separation is carried out by using contrast functions kurtosis and approximations to negentropy with function as in (2.11) and (2.14).The minimum value of the nutrient function J is plotted against the number of its evaluations. Besides these indices the estimation of ICs using BFOICA has been pictorially depicted to have clarity of estimation in all examples considered here.

As one more index for measurement, the demixing matrix W given by (4.11) is estimated by each of the ICA algorithms.

$$W = \begin{pmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{pmatrix} \tag{4.11}$$

4.4. Analysis of Results

The main thrust of the present work being the BFOICA algorithm, the comparison has been restricted to fast ICA (the fastest and efficient of all the Newtonian methods for ICA) and the CGAICA (ICA based on most popular evolutionary computation technique GA discussed in Chapter 3). The separation capability of BFOICA is verified by providing four different examples.

Example 1:

A random binary wave and a sine wave with 400 samples as shown in Fig. 4.2 are mixed by the mixing matrix

$$A = \begin{pmatrix} 0.9121 & 0.2292 \\ 0.4763 & 0.7348 \end{pmatrix} \tag{4.12}$$

Their mixtures are represented in Fig. 4.3.

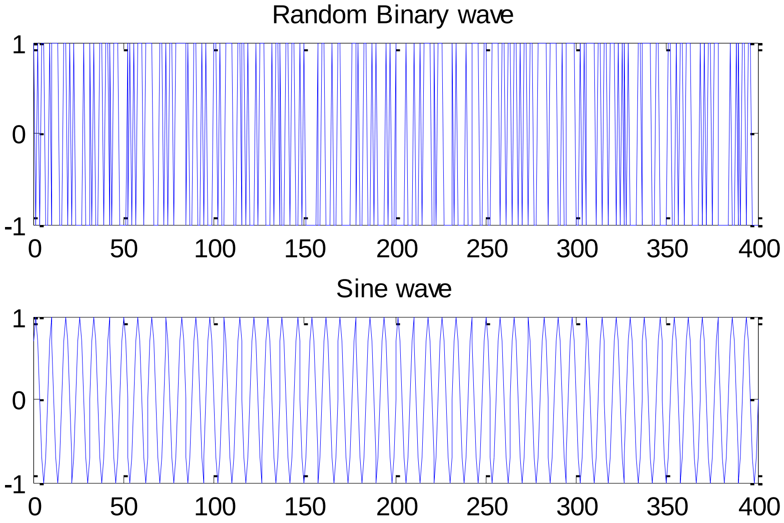

Fig. 4.2 Original Signals

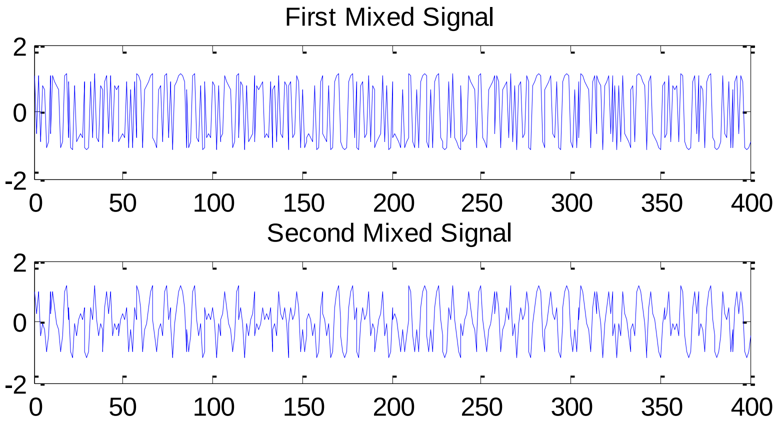

Fig. 4.3 Mixture of random binary and sine wave

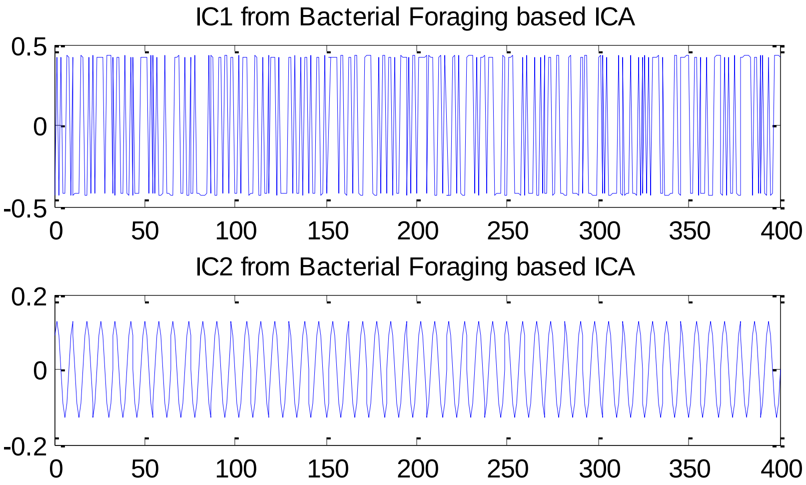

Fig. 4.4 The recovered independent components

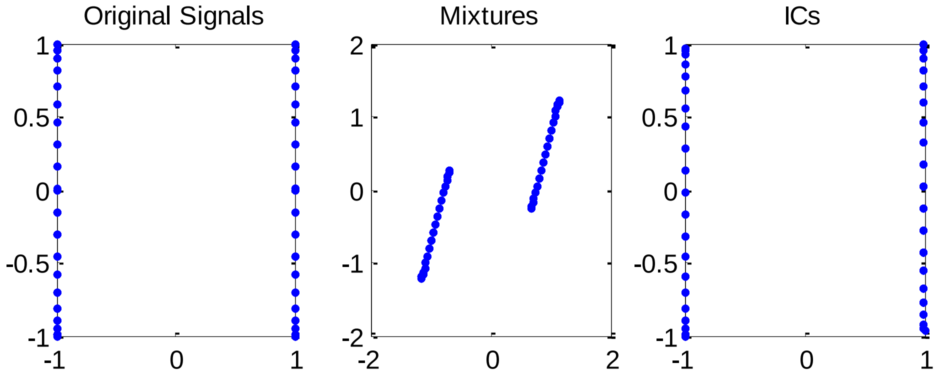

Fig. 4.5 Scatter Diagram of the Original Signals, Mixed Signals and Independent Components

Using BFOICA the two signals or ICs are recovered in the decreasing order of the value of their contrast function. For the case of kurtosis as contrast function the separated signals are depicted in Fig. 4. Scatter plot of the random binary signal and the sine wave, their mixtures and the separated independent components are shown in Fig.5. Random binary signal has more kurtosis value than the sine wave used. So it is observed that the random binary is recovered first and the sine wave comes as the second IC. Three sets of tuned values of the optimization parameters for BFOICA algorithm with kurtosis as the contrast function and the corresponding mean square error (MSE) for random binary signal is summarized in Table 4.1.The estimated values of W have been compared in Table 4.2. This type of examples is often used to demonstrate the fundamental separation ability of any ICA algorithm in the first step.

Table 4-1 Parameters of BFOICA to recover Random binary and sine wave with 400 samples

No of Bacteria S_b	6	12	40
No of Chemotactic Steps N_c	5	8	30
No of Reproduction Steps N_{re}	4	3	3
No of Elimination and Dispersion Events N_{ed}	4	4	4
Probability of Elimination & Dispersion p_{ed}	0.02	0.02	0.02
MSE for Random Binary Component	1.684×10^{-5}	1.196×10^{-6}	3.084×10^{-10}

Table 4-2 Demixing matrix Comparison for different ICA algorithms for Example 1

Algorithm	w_{11}	w_{12}	w_{21}	w_{22}
Fast ICA	-1.3188	0.4089	1.2132	-2.3031
CGAICA	-0.2615	0.6283	0.5240	0.2412
BFOICA	0.9045	-2.1393	0.0923	0.0391

Example 2:

We consider another example with a sine wave and its third harmonic and mix these two by the same mixing matrix A as given in (4.11). The original signals and their mixtures are shown in Fig. 4.6 and 4.7 respectively. Using BFOICA with kurtosis as the contrast function, the sine wave and its third harmonic are clearly separated which is shown in Fig 4.8. In this case also it is clearly observed that the signal with maximum kurtosis appears first. Hence there is no permutation ambiguity. This example has been considered to demonstrate the separation capability in case of mixtures of same type of signals with different frequencies.

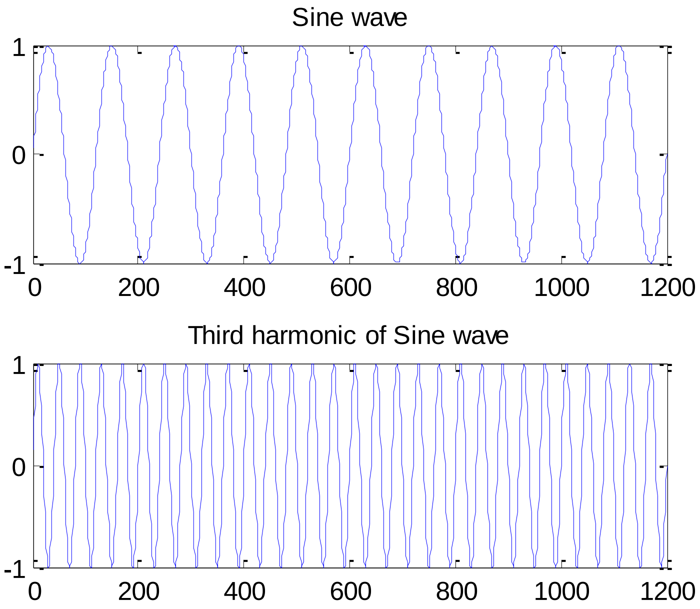

Fig. 4.6 Original sine wave and its third harmonic.

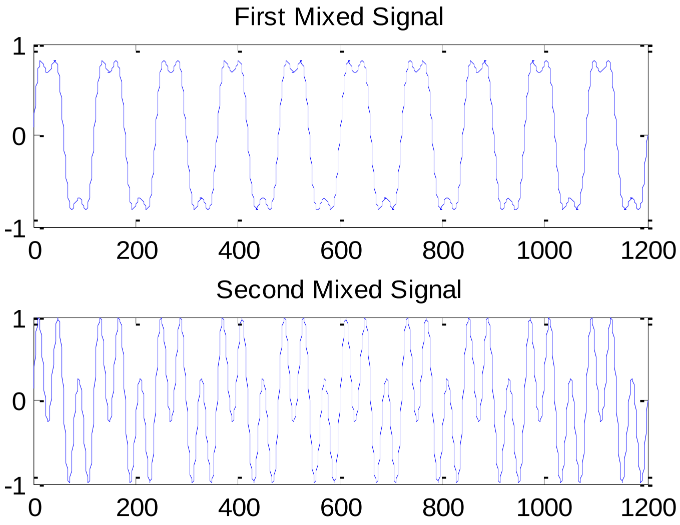

Fig. 4.7 Mixture of sine wave and its third harmonic

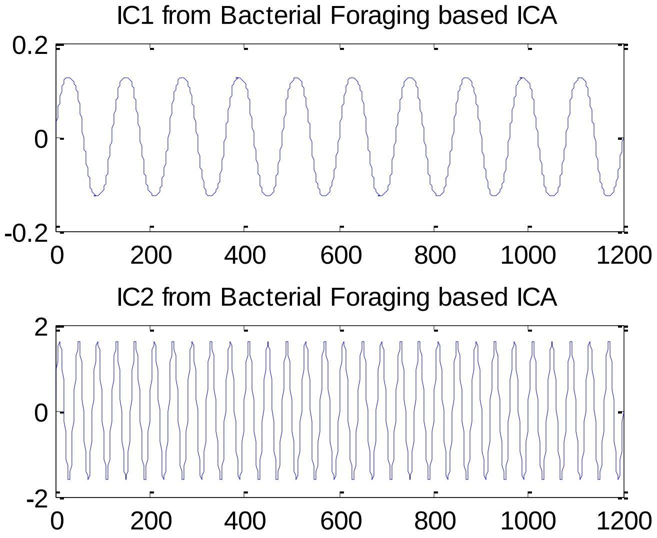

Fig. 4.8 Recovered Independent Components

Example 3:

In this example separation of speech signals from their instantaneous mixtures using the ICA algorithms is dealt. Two speech signals (sampled at 16kHz) with 120000 samples (Fig. 4.9) each are taken [4.6] and instantaneously mixed by the artificial random mixing matrix B as given in equation (4.12). The mixture signal is depicted in Fig.4.10.

$$B = \begin{pmatrix} 0.9121 & 0.2292 \\ 0.6763 & 0.7348 \end{pmatrix} \tag{4.13}$$

The mixed signals are applied to BFOICA algorithm for separation using kurtosis as the contrast function. The estimated speech signals show a clear separation from the mixtures which is shown in Fig.4. 11. It is also observed that the speech signal with maximum value of kurtosis is estimated first followed by the speech signal with lower value of kurtosis. The value of J defined in equation (4.1) for first speech component is 0.1448 and that for the second speech component is 0.3794. The separation was also observed for shorter lengths of speech data i.e. speech signal with less number of samples. The estimated value of W as given in equation (4.10) for the case considered here has been compared in Table 4.3.

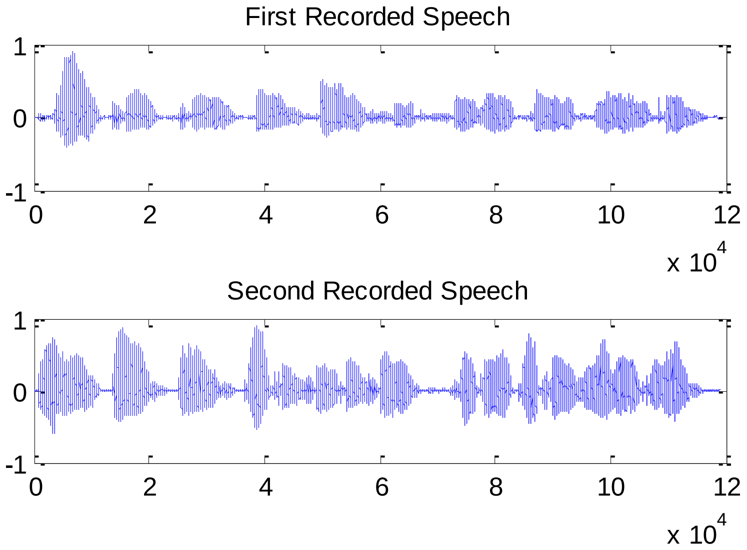

Fig. 4.9 Two recorded speech signals

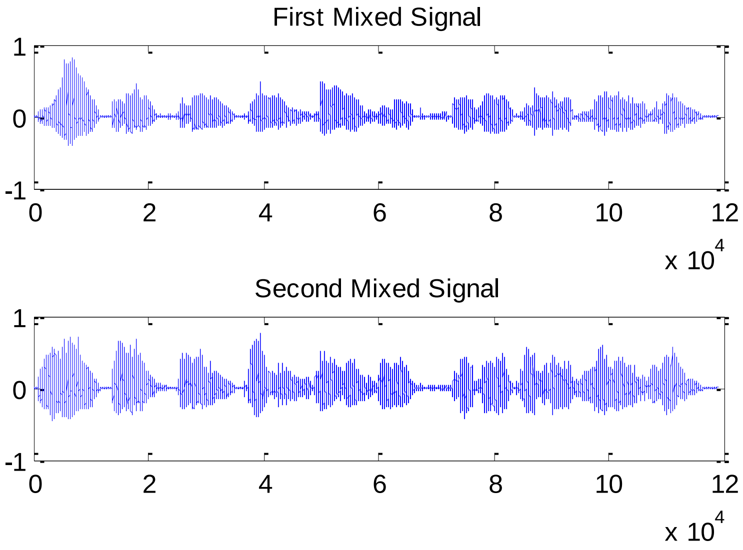

Fig. 4.10 Artificial mixture of two recorded speech signals

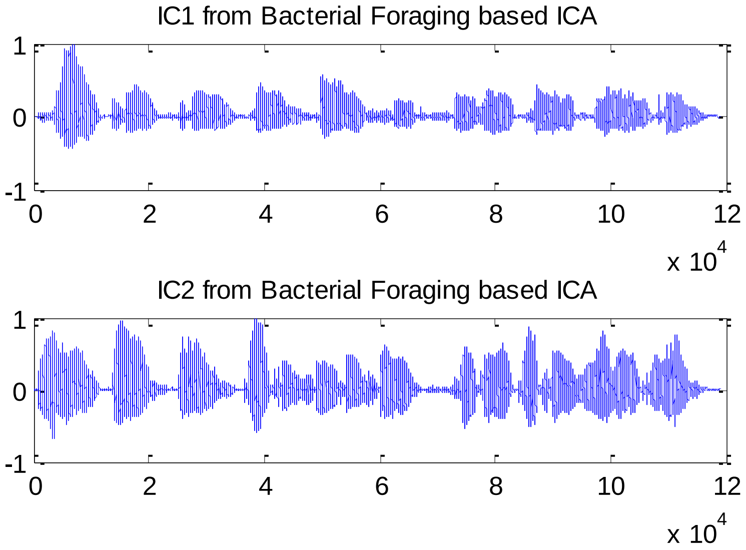

Fig. 4.11 Speech signals recovered by BFOICA

Table 4-3 Demixing matrix Comparison for different ICA algorithms for Speech mixture case in Example 3

Algorithm	w_{11}	w_{12}	w_{21}	w_{22}
Fast ICA	-0.9797	0.3184	-0.5337	0.7462
CGAICA	0.7283	-0.6386	0.0684	0.0684
BFOICA	-0.4343	0.4788	0.9597	0.8707

Example 4:

This example illustrates the capability of separation of the algorithm in the case of mixture of a supergaussian signal and a subgaussian signal. Speech is supergaussian and the random noise is subgaussian shown in Fig.4.12. Speech and random noise are instantaneously mixed by the mixing matrix A. The mixed signals are shown in Fig. 4.13 and the BFOICA separated signals are depicted in Fig 4.14. The scatter plots of the original signals, mixed signals and the recovered signals are given in Fig. 4.15.

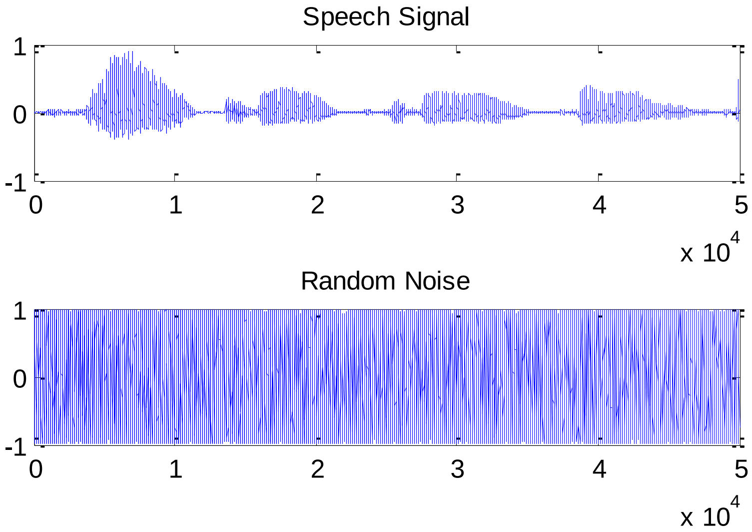

Fig. 4.12 The Speech & random binary Signals

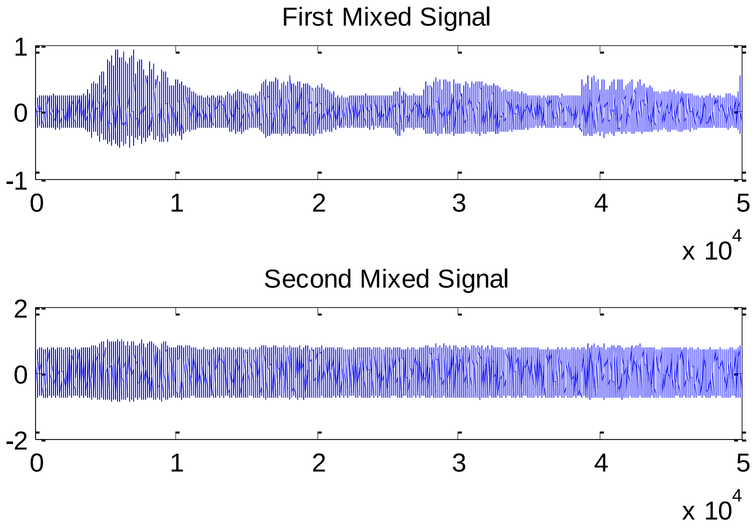

Fig. 4.13 The mixer of speech & random binary signals

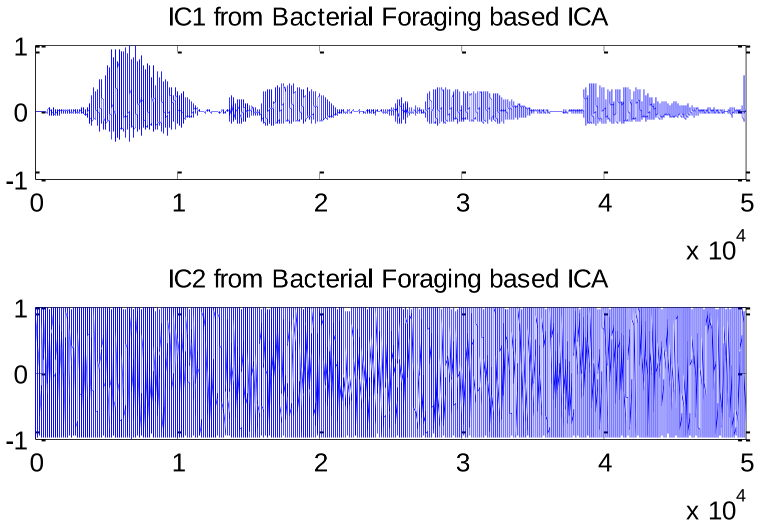

Fig. 4.14 Estimated Speech and random binary Components

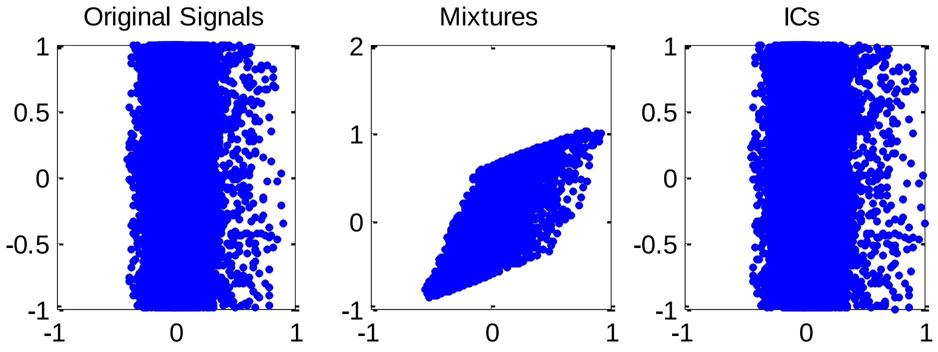

Fig. 4.15 Scatter Diagrams of the Original Signals, Mixed Signals and Independent Components

4.4.1 Comparison of Different Optimization Schemes

The signal separation carried out in Example 1 is also performed by fast ICA and CGAICA algorithms to have a comparison with our proposed BFOICA algorithm. Table 4.4 summarizes the typical MSE values estimated for random binary and sine wave components for fast ICA, CGAICA and BFOICA algorithms. We have chosen the GA with 15 bits which is most commonly considered for binary GAs. The number of chromosomes chosen for CGAICA is 20 with crossover and mutation probabilities 0.85 and 0.1 where as the number of bacteria chosen for BFOICA is 40 with 30 chemotactic steps, 3 reproductions and 4 elimination and dispersal events. It is clearly observed that BFOICA yields far better MSE performance than CGAICA and comparable or sometimes better performance than the most popular fast ICA algorithm. Apart from this the performance of neural learning algorithms relies on the selection of activation functions. Usually different activation functions are used for sources with different statistics (subgaussian or supergaussian) which are difficult to estimate in blind source separation. Also most of the algorithms are under the condition, though not mentioned directly, that the sources are of the same statistics. Subgaussian and supergaussian signals have different sign of fourth order cumulants (*Kurtosis*), and this is not considered in the derivation of the neural learning algorithm. Therefore it may so happen sometimes that the learning algorithm fails to converge to a consistent optimum when the sources have different signs of kurtosis. However from the fourth example it is clear that the BFOICA algorithm recovers the independent components from the mixture of supergaussian and subgaussian signals.

Table 4-4 MSE & Crosstalk Comparison for different ICA algorithms

Algorithm	MSE		Crosstalk (dB)	
	Random Binary	Sine wave	Random Binary	Sine wave
Fast ICA	2.5004×10^{-9}	8.58×10^{-2}	-198.0682	-17.6259
CGAICA	2.1811×10^{-8}	8.58×10^{-2}	-176.4085	-17.6259
BFOICA	3.084×10^{-10}	8.58×10^{-2}	-218.9962	-17.6259

4.4.2 Comparison of Convergence

To have a comparative convergence study the GA (a popular evolutionary algorithm) based approach is also simulated along with BFOICA and CGAICA based algorithms. In GA schemes, a *gene* is converged when 95% of the population shares the same value. To have a common ground of comparison with CGAICA, the variation of the best value of nutrient function with the number of times the nutrient function is evaluated, is studied. In case of example 1 with same mixing matrix A, CGAICA is performed. Fig. 4.16 shows the variation of the nutrient function J values with the number of J evaluations for random binary component. This clearly indicates that BFOICA algorithm has much faster convergence as compared to the GAICA algorithm. Similar results are also obtained by taking approximation to negentropy as contrast function. Needless to say that fast ICA has faster convergence than CGAICA or BFOICA algorithm. The analytical convergence properties of the proposed BFOICA are difficult to analyze due to highly complex nature of the underlying problem and is beyond the scope of the present work.

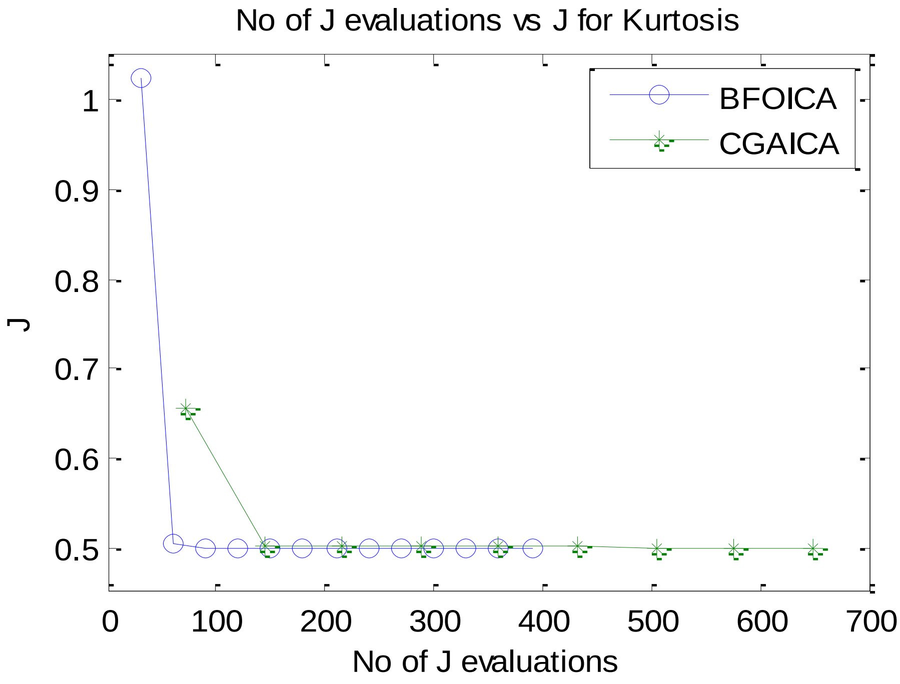

Fig. 4.16 Minimum value of the nutrient function vs number of its evaluations for BFOICA (for kurtosis) and CGAICA

4.4.3 Permutation Ambiguity

From all the above examples it is observed that using BFOICA independent components were recovered always in a fixed order in all runs of the simulation experiments. The IC for which the nutrient function has a global minimum value appeared first and then appeared the IC with subsequent minimum value of the nutrient function J. So the order of the ICs can be predicted the value of their statistical property like kurtosis is relatively known. However this is not the case with fast ICA or any other gradient based ICA. In deflation approach of conventional ICA techniques every different initialization of weight vectors leads to a different order of IC recovery. The permutation indeterminacy present in ICA techniques seems to be dissolved by use of our proposed technique. This can be extended without loss of generalization to cases of ICA with any number of independent components. This is very useful in applications where the knowledge of the order of ICs or any particular IC recovery is very important. Generally to recognize any particular IC a post processing (like some training based approach) is carried out in some applications like denoising the ECG signal. Using the proposed technique the required post processing can be eliminated as it is well known that ECG signal being supergaussian has more kurtosis than noise and hence it can be recovered first. In this case there is no need to estimate the second IC (noise).

4.5. Conclusions

The bacteria foraging based optimization is used in a constrained manner to estimate the independent components from their observed mixtures. This scheme is tested using several examples including the speech signals for instantaneous mixing cases. From the simulation results it is very clear that the BFOICA algorithm has faster convergence and better mean square error performance than as compared to the CGAICA algorithm. In comparison to the fast ICA algorithm it has good mean square error. The permutation ambiguity present in ICA techniques is resolved by use of the BFOICA algorithm if relative knowledge of the statistical characteristics of the signals to be estimated is known. However care should be taken while adjusting the parameters for bacteria foraging optimization so that premature convergence in a local optimum does not occur. For this the parameters should be initially varied over a large range to ensure the proper convergence to a global optimum. However the computational complexity of the BFOICA is obviously higher as compared to the gradient based algorithms. But as the number of independent components increases the BFOICA algorithm performs better than the available gradient based methods.

References

[4.1] K.M.Passino, “Biomimicry of Bacterial Foraging for distributed optimization and control,” *IEEE Control Syst. Mag.*, vol.22, no.3, pp52-67, Jun.2002.

[4.2] S.Mishra, “A hybrid least square-fuzzy bacteria foraging strategy for harmonic estimation,” *IEEE Trans. Evol. Comput.*, vol.9, no.1, pp.61-73, Feb.2005.

[4.3] S.Mishra and C.N.Bhende, “Bacterial Foraging Technique-Based Optimized Active Power Filter for Load Compensation,” *IEEE Trans. Power Del.*, vol.22, no.1, pp.457-465, Jan.2007.

[4.4] D.P.Acharya, G.Panda and Y.V.S. Lakshmi, “Constrained Genetic Algorithm based Independent Component Analysis,” *IEEE Congress on Evolutionary Computation*, Singapore, Sep. 2007.

[4.5] D.P.Acharya, G.Panda, S.Mishra and Y.V.S. Lakshmi, “Bacterial Foraging Optimization based Independent Component Analysis,” *IEEE International Conference on Computational Intelligence and Multimedia Applications*, Sivakasi, India, Dec. 2007.

[4.6] http://www.cnl.salk.edu/~tewon/Blind

Chapter

V

DEVELOPMENT OF NEW NONLINEAR ICA ALGORITHM BASED ON BACTERIAL FORAGING OPTIMIZATION TECHNIQUE

5.1. Introduction

The ICA problem addressed extensively so far pertains to linear mixing. As the ICA model performs poorly in many practical situations, a natural extension is to develop nonlinear models to achieve improved performance. In practice, the sources usually undergo a nonlinear mixing process. This is due to the fact that the mixing process of multiple sensors contains some nonlinear transformation such as the saturation distortion of sensors. Hence nonlinear ICA is more realistic in many cases. A fundamental characteristic of the nonlinear ICA problem is that in the general case, solutions always exist, and they are highly non-unique. One reason for this is that if x and y are two independent random variables, any of their functions f(x) and g(y) are also independent. An even more serious problem is that in the nonlinear case, x and y can be mixed and still be statistically independent [5.1, 5.2].

In spite of many difficulties in recovering the Independent Components from nonlinear mixtures, several models and solutions have recently been proposed. Deco [5.3] studied a volume-conserving nonlinear transform for nonlinear BSS. Pajunen et al. [5.4] used Kohonen’s self-organizing map (SOM) to extract sources from nonlinear mixtures. It is a model-free method but suffers from the exponential growth of network complexity and interpolation error in recovering continuous sources. In another publication Taleb et al. [5.5] proposed an entropy-based BSS algorithm in post nonlinear mixtures. Yang et al. in [5.6] have suggested an information back-propagation algorithm for interchannel nonlinear mixtures in the sense of entropy maximization and mutual information minimization, and have adopted a sigmoidal nonlinear transformation of the nonlinear model based on the work by Burel [5.7]. Very recently, Tan et al. have applied radial basis function networks for nonlinear BSS [5.8]. These newly developed models are established on the basis of parametric models because it is very important for nonlinear BSS to obtain unique separating results when only the independence of sources is known a priori. All of these methods are developed based on gradient-based optimization to avoid computing some unknown quantities in an unsupervised manner. Therefore, these methods are susceptible to the local minima problem during the learning process and are thus limited in many practical applications. In this Chapter a new nonlinear ICA algorithm is suggested using an efficient evolutionary computing based on bacteria foraging optimization (BFO). Its performance is evaluated through simulation study and is compared with that obtained from existing nonlinear ICA.

This Chapter is organized as follows. In Section 5.2, we present the Nonlinear ICA model. Section 5.3 proposes the new BFO based nonlinear ICA algorithm. The simulation study of the proposed method is carried out in Section 5.4 using an example of nonlinear mixing and its performance is evaluated. The GA based nonlinear model is also simulated and the results of both the methods are compared in Section 5.4. Finally the conclusion on the findings is included in Section 5.5.

5.2. Nonlinear ICA Model

The mixing process of sources can be described by different models. A post nonlinear mixture model has been used extremely in literature[5.9, 5.10]. In [5.9] learning rules for the nonlinear mixing models based on the information maximization criterion is proposed. In this model the mixing process is divided into a linear mixing part and a nonlinear transfer channel, in which the nonlinear functions are approximated by parametric sigmoidal or by higher order polynomials.

The task of ICA is that of estimating the unknown source components from sensor signals. This is described as

$$\mathbf{x}(t) = F[A.s(t)] \qquad (5.1)$$

where $x(t) = [x_1, x_2, ..., x_n]^T$ is observed $n \times 1$ sensor vector, $s(t) = [s_1, s_2, ..., s_n]^T$ is an $n \times 1$ unknown source vector having statistically independent non-Gaussian elements $s_i(t)$, A is an $n \times n$ unknown full-rank and nonsingular mixing matrix and $F = [f_1, f_2, ..., f_n]^T$ are the set of invertible nonlinear transfer functions. The nonlinear ICA problem consists of recovering the source vector $s(t)$ using only the observed data $x(t)$, the assumption of statistical independence between the entries of the input vector $s(t)$ and possibly some *a priori* information about the probability distribution of the inputs. If all of the function f_i are linear then (5.1) reduces to the linear mixing model. The dimensions of x and s, generally though not equal, in the present study it is assumed to be same.

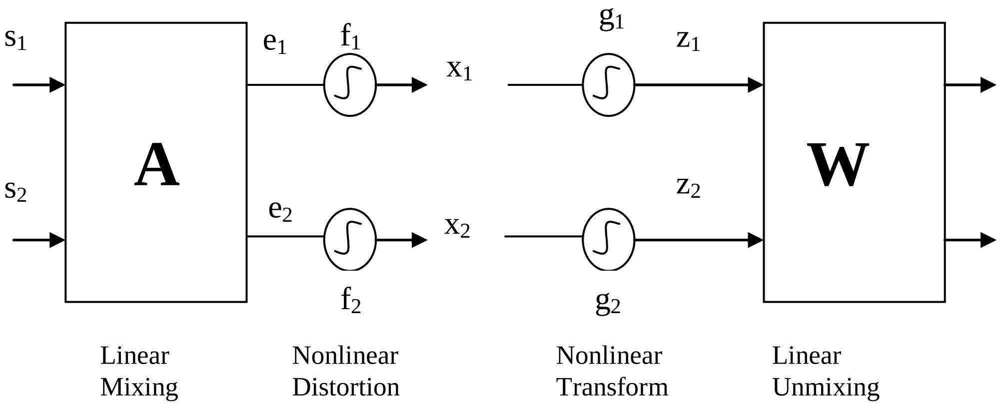

Fig 5.1 Nonlinear ICA model for Post Nonlinear Mixtures

Fig.5.1 shows the mixing system (known as a post nonlinear mixture) is divided into two different phases as proposed in [5.11]. A linear mixing is done first and then each channel *i* is passed through a non-linear transformation. In the reverse manner, for the separating system, first we need to approximate g_i, which is the inverse of the nonlinear function in each channel, and then separate the linear mixture by applying W to the output of the g_i nonlinear function

$$y_i(t) = \sum_{j=1}^{n} w_{ij} g_j(x_j(t)) \tag{5.2}$$

In many cases, the inverse function g_j is approximated by a sigmoidal transfer function, but because of certain situation in which the human expert is not given a prior knowledge about the mixing model, a more flexible nonlinear transfer function based on an Mth order odd polynomial as given in (5.3) is used.

$$p_j(x_j) = \sum_{k=1}^{M} p_{jk} x_j^{2k-1} \tag{5.3}$$

where $p_j = [p_{j1},\ldots,p_{jM}]$ is a parameter vector to be determined. Hence the output sources are calculated as

$$y_i = \sum_{j=1}^{n} w_{ij} \sum_{k=1}^{M} p_{jk} x_j^{2k-1} \tag{5.4}$$

5.3. The Proposed BFO based Nonlinear ICA Algorithm

The bacterial foraging based optimization technique presented in Chapter 4 is used to estimate the nonlinear parameters. The demixing matrix W is computed by a traditional gradient based technique. The nutrient function used in BFO for minimization is given by

$$\text{Nutrient function} = \frac{1}{I(y)} \tag{5.4}$$

where I(y) known as the mutual information and is given by

$$I(y) = -\log|W| - \sum_{i=1}^{n} E\left[\sum_{k=1}^{M} (2k-1) p_{ik} x_i^{2k-2}\right] + \sum_{i=1}^{n} H(y_i) \tag{5.5}$$

with

$$H(y_i) \approx \frac{\log(2\pi e)}{2} - \frac{(k_3^i)^2}{2.3!} - \frac{(k_4^i)^2}{2.4!} + \frac{3}{8}(k_3^i)^2 k_4^i + \frac{1}{16}(k_4^i)^3 \quad (5.6)$$

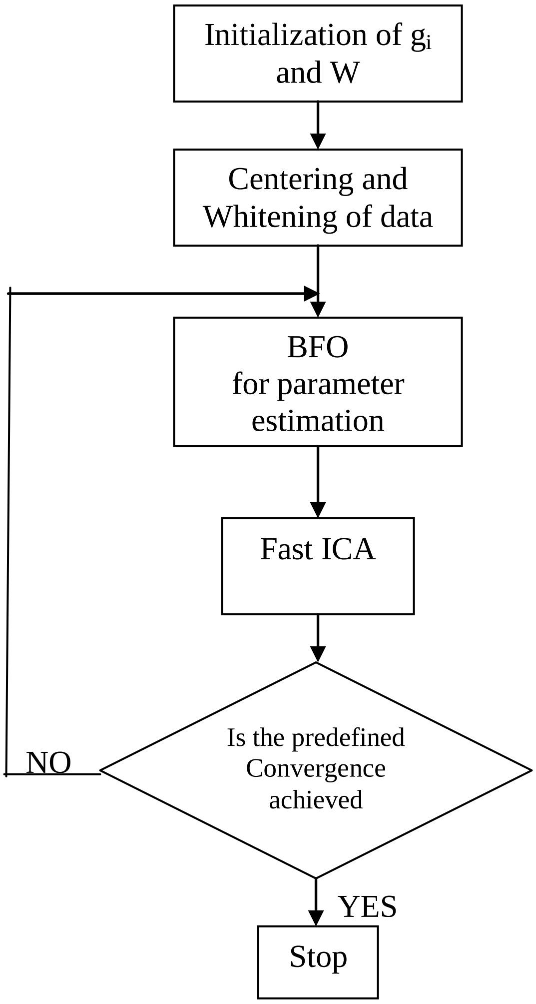

Fig 5.2 Flow chart of the proposed BFO based Nonlinear ICA Algorithm

In which, $k_3^i = m_3^i, k_4^i = m_4^i - 3$ for a centered and prewhitened data. The zero mutual information value means that the y_i is independent of each other and the calculation of entropy $H(y_i)$, needs to estimate the distribution density function of y_i firstly. The proposed algorithm is depicted by the flow chart of Fig 5.2.

5.4. The Simulation Results

To verify the validity of the proposed BFO based Nonlinear ICA algorithm, simulation study is carried out. In the simulation environment, the two signals are mixed by a random matrix A and then passed though following two nonlinearities respectively.

$$f_1(x) = \tanh(0.5x) \tag{5.7}$$

$$f_2(x) = \tanh(x) \tag{5.8}$$

The mixed signals are the inputs to the proposed algorithm for separation. The GA based nonlinear ICA is also applied to the same example for comparison purpose.

In the simulation a random binary wave and a random noise are considered as input signals as used in case of [5.10]. The original random binary and random noise signals and their nonlinear mixtures are depicted in Figs 5.3 and 5.4 respectively. The GA and BFO based nonlinear estimation of the independent components obtained by simulation are shown in Figs 5.5 and 5.6 respectively. In Fig. 5.5 the random binary signal is recovered first and then the random noise but in Fig. 5.6 the order observed is reversed. Here it may be noted that the permutation ambiguity is present as GA or BFO is used only for parameter estimation and the gradient based algorithm like fast ICA is used for linear ICA. Hence the order of appearance of the independent components is not always the same.

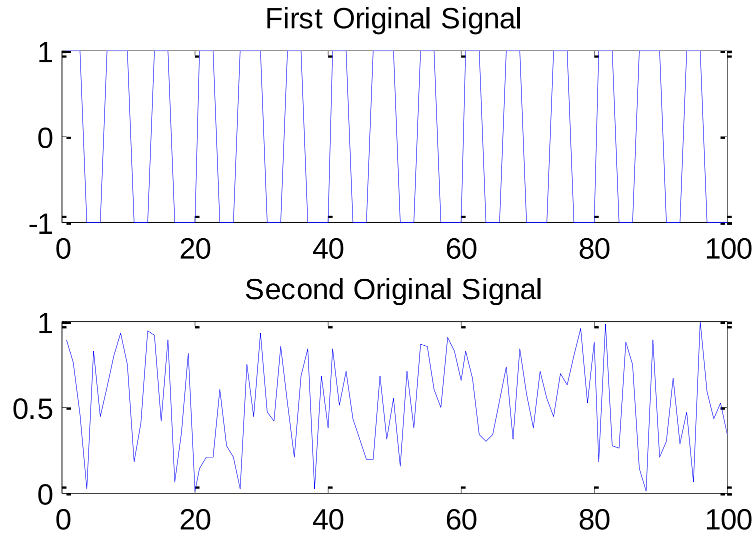

Fig 5.3 Original Random binary and random noise Signals

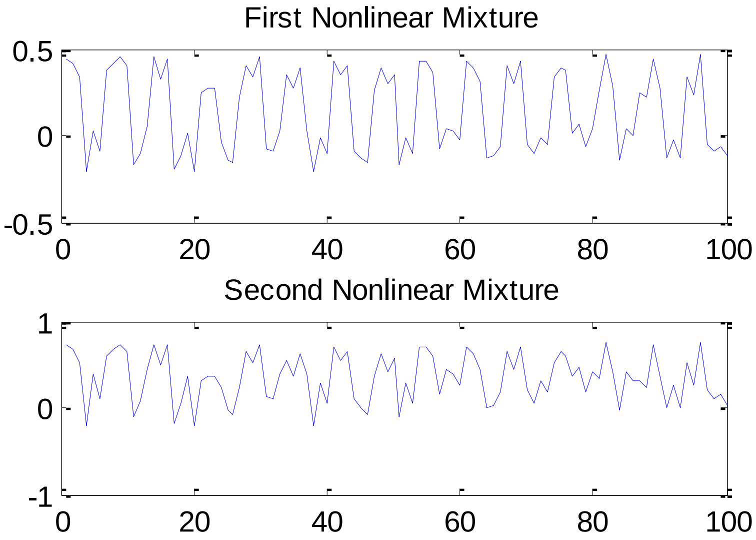

Fig 5.4 The nonlinear mixtures of Random Binary and Random Noise Signals

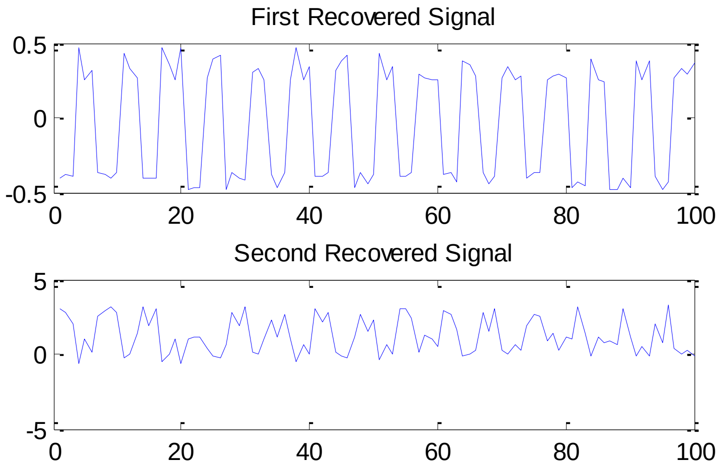

Fig 5.5 Signal separation using GA based Nonlinear ICA

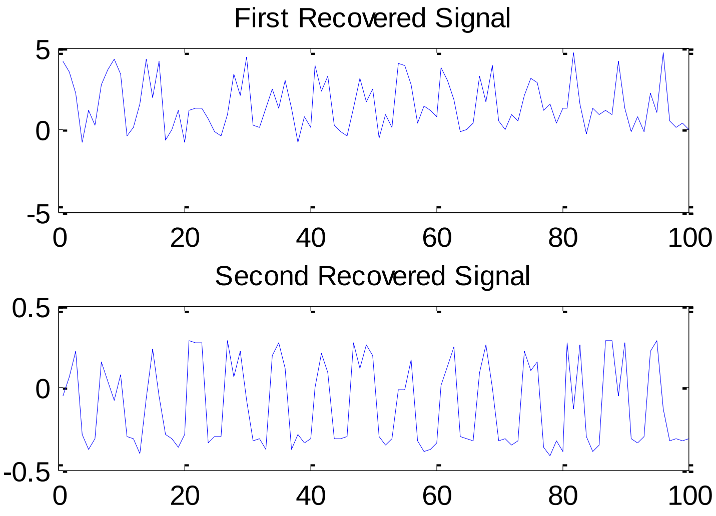

Fig 5.6 Signal separation using BFO based Nonlinear ICA

5.5. Conclusions

The bacteria foraging based optimization is used to estimate the independent components from their observed nonlinear mixtures. This scheme is tested using an example for instantaneous nonlinear mixing case. The BFO is reported to have faster convergence than its counterpart GA and this has been illustrated in Chapter 4. In this work we assess the potentiality of BFO along with the fast ICA algorithm for separation of sources from their nonlinear mixtures. The preliminary study reveals encouraging results. Further investigation is needed to apply the proposed technique to separate other input signals and using various nonlinear mixtures. The study can be extended to separate more than two signals using the proposed nonlinear ICA.

References

[5.1] A. Hyvarinen and P. Pajunen. “Nonlinear independent component analysis: Existence and uniqueness results”, *Neural Netw.*, 12(3):429{439, 1999.

[5.2] C. Jutten and J. Karhunen. Advances in nonlinear blind source separation. In *Proc. of the 4th Int. Symp. on Independent Component Analysis and Blind Signal Separation (ICA2003)*, pages 245{256, Nara, Japan, April 1-4 2003. Invited paper in the special session on nonlinear ICA and BSS.

[5.3] G. Deco and W. Brauer, “Nonlinear higher order statistical decorrelation by volume-conserving neural architectures,” *Neural Netw.*, vol. 8, pp. 525–535, 1995.

[5.4] P. Pajunen, A. Hyvarinen, and J. Karhunen, “Nonlinear blind source separation by self-organizing maps,” *Progress in Neural Information Processing: Proceedings of the International Conference on Neural Information Processing.* New York: Springer-Verlag, vol. 2, pp. 1207–1210, 1996.

[5.5] A. Taleb, C. Jutten, and S. Olympieff, “Source separation in post nonlinear mixtures: an entropy-based algorithm,” *Proc. Eur. Symp. Artificial Neural Networks*, pp. 2089–2092, 1998.

[5.6] H. H. Yang, S. Amari, and A. Cichocki, “Information back-propagation for blind separation of sources from nonlinear mixture,” *Proc. IEEE Int. Conf. Neural Networks*, pp. 2141–2146, 1997.

[5.7] G. Burel, “Blind separation of sources: a nonlinear neural algorithm,” *Neural Netw.*, vol. 5, pp. 937–947, 1992.

[5.8] Y. Tan, J. Wang, and J. M. Zurada, “Nonlinear blind source separation using a radial basis function network,” *IEEE Trans. Neural Networks*, vol. 12, pp. 134–144, Jan. 2001.

[5.9] F.Rojas, C.G.Puntonet, M. Rodriguez-Alvarez, I.Rojas and R.M.Clemente, “Blind Source Separation in Post-Nonlinear Mixtures Using Competitive Learning, Simulated Annealing and a Genetic Algorithm”, *IEEE Transactions on Systems, Man, and Cybernatics-Part C: Applications and Reviews*, vol.34, No.4, pp.407-416, Nov. 2004.

[5.10] Y. Tan and J. Wang, “Nonlinear Blind Source Separation Using Higher Order Statistics and a Genetic Algorithm”, *IEEE Trans. On Evol. Compt.*, vol.5, no.6, pp.600-611, Dec.2001.

[5.11] A. Taleb and C. Jutten, “Source Separation in post-nonlinear mixtures”, *IEEE Trans. Signal Processing*, vol.47, no.10, pp.2807-2820, 1999.

Chapter

VI

ON EFFECTS OF FINITE REGISTER LENGTH ON DIFFERENT ICA ALGORITHMS FOR VLSI IMPLEMENTATION

6.1 Introduction

Though several algorithms for ICA have been reported in literature very few attempts have been made for successful practical implementation of any of these algorithms [6.1, 6.2]. An algorithm can be implemented either by 'fixed point arithmetic' or by 'floating point arithmetic'. The floating-point method demands more computational overhead resulting in a large number of processing elements. Thus the system becomes speed limited with large chip area and consuming more power in the process. In practice, only fixed-point arithmetic is used for application-specific ICs because the hardware is much simpler and faster compared to the floating-point arithmetic.

The complex arithmetic, the iterative computation with slow convergence rate and large volumes of raw and processed data have caused ICA algorithms a time consuming process for implementation in software. The implementation in hardware provides potentially faster and real-time solutions with optimal parallelism. While software implementation is useful for investigating the capabilities of ICA algorithms and is sufficient for most applications, hardware implementation is essential to fully benefit from the parallel architecture and to facilitate high speed processing. The major difference between hardware and software implementations lies in the fact that hardware subroutines are executed by integrated circuits (ICs) instead of a series of microinstructions. Hardware implementation also solves the insufficient memory problem encountered by software for large data sets and high dimensionality.

While implementing using fixed-point arithmetic the sample values of signal are stored in registers and after any processing again stored in some other registers. In these cases every operation introduces an error (truncation/round up) due to finite register length. These errors propagate and appear as a noise at the output of the system. Such noise alters or degrades the performance of the desired system. The character of this noise depends on many factors – for example, type of non-linearity, structure of the algorithm adopted for implementation, type of arithmetic, representation of negative numbers, and properties of the input signal. Hence a detailed analysis of error and account for implementation noise introduced by the fixed-point operations becomes the essential before practical implementation of a system [6.3, 6.4, and 6.22]. The analysis also helps to know the accuracy that can be expected out of the existing system and to design with the minimum cost a new system to meet the required accuracy specifications [6.5]. Such an analysis with any of the ICA algorithms is not found in literature to our knowledge. An analysis with fast ICA and algebraic ICA algorithms has been carried out in one of our conference papers [6.6]. The fixed-point architecture and its trial implementation have been performed in [6.7].

Bacterial Foraging based Independent Component Analysis (BFOICA) [6.8] and Constrained Genetic Algorithm based Independent Component Analysis (CGAICA) [6.9] are two recently developed derivative free evolutionary computational ICA techniques which have been discussed in details in Chapters III and IV respectively. Both BFO and GA being population search based optimization techniques, they have several commonalities. However, BFOICA is reported to have faster convergence as compared to CGAICA. Both these evolutionary computation based algorithms resolve the permutation uncertainty associated with all gradient based ICA techniques like fast ICA. Therefore it is quite motivating to study the effect of finite register length implementation of both the proposed algorithms. The present work focuses on the fixed-point performance evaluation of BFOICA algorithm and CGAICA algorithm when implemented with different register lengths. The fixed-point performance of the most popular fast ICA algorithm is also presented in this work.

This Chapter is organized as follows. Section 2 presents a comprehensive overview of the VLSI solutions to different ICA algorithms. The fixed-point models of three ICA algorithms are described in Section 3. In Section 4 a detailed architecture for the most popular fast ICA algorithm is provided. The computer simulations for fixed-point performance for all the algorithms specified above are described in Section 5. The fixed-point trial implementation of the most popular fast ICA algorithm is also described in section 5. Section 6 analyses the results of simulation experiment and synthesis results of fast ICA implementation. Finally the conclusions are provided in Section 7.

6.2 VLSI Solutions to Independent Component Analysis

Advances in very large scale integrated (VLSI) circuit technologies allow designers to implement some ICA algorithms on fully analog CMOS circuits, analog–digital (AD) mixed-signal ICs, digital application-specific ICs (ASICs), and general field programmable gate arrays (FPGAs). Both analog CMOS circuits and mixed-signal ICs are fully customized by designers using either analog or AD mixed technologies, where the silicon is utilized in the most efficient manner but the development expense is incredibly high [6.12]. The digital nonprogrammable ASICs such as standard-height library and mask gate arrays are also full-custom VLSI and are used to implement designs at high circuit density by specifying interconnections during latter stages of the IC manufacturing process. In addition, the large amount of available standard libraries of basic logic cells makes the design expense much cheaper and the design process much faster. The FPGAs based on the reconfiguration technology are the most economic and efficient solutions to ICA algorithms since they allow end users to modify and configure their designs multiple times. Specifically, recent rapid increase in the density of FPGAs has made it possible to implement large ICA designs with a completely hardware-driven approach.

Many ICA algorithms are slow processes in signal/image processing applications due to the complicated arithmetics as well as the time-consuming iterative computation. VLSI being an ideal algorithm implementation carrier offers many features such as high processing speed, which is extremely desired in ICA implementations. The complex computation of ICA is one of the main barricades in ICA hardware implementation, especially in synthesis procedure. Therefore, hierarchy and modularity techniques in VLSI design are essential for most ICA implementations to overcome the complexity of ICA algorithms. The hierarchy, or the divide and conquer technique, involves dividing an ICA process into subprocessing modules until the complexity of the bottom level submodules becomes manageable. These submodules are independently developed, then integrated together and put into a design and development environment for performing tasks such as synthesis, optimization, placement, and routing. The use of modularity enables the parallelism of the design process and facilitates the development of generic modules in various designs. Presently, the concept of modularity has gained more attention because of a need to reduce the design cycle and the development cost. For example, Cho and Lee [6.13] implemented the InfoMax ICA on the analog CMOS circuit in 2001, and Cauwenberghs [6.14] implemented the same algorithm using the AD mixed-signal technology in 2003. Both implementations utilized modular architectures such that the designs could be easily expanded to larger chips and further integrated into multichip systems for a large number of input and output channels.

Another VLSI design challenge concerning the ICA implementation is the iterative computation that requires large amount of processing time and RAM to store intermediate results. The best solution to this problem is to reduce the ICA algorithm at the early preprocessing stage of the VLSI design process.

Computationally efficient ICA algorithms would contain less iteration and converge rapidly to the desired accuracy. The digital domain solutions are of importance in many respects and necessitate the fixed-point error analysis. The following two solutions provide the digital implementation of ICA algorithms.

6.2.1 ASIC Solutions

The analog CMOS and AD mixed signal technologies provide user full-custom solutions to ICA algorithms. End users are generally required to have sufficient knowledge and focus more on detailed analog physical problems and basic component designs. Therefore, the application domains are comparatively limited, and the development costs significant time with expenses. Fortunately, the fast-blooming digital VLSI technologies like ASIC and FPGA allow end users to concentrate on the algorithm implementation itself because IC vendors provide enormous standard libraries. ASICs and FPGAs are therefore called user semi custom solutions.

From the aspect of circuit density and efficiency, the nonprogrammable ASICs cover the lower end of analog CMOS and AD mixed-signal full-custom VLSI and the higher end of reprogrammable FPGAs. Compared to reprogrammable FPGAs, nonprogrammable ASICs retain the benefits of compact circuit design and low power consumption. Although the nonprogrammable feature increases the design expense and risk, ASICs that typically contain ten million logic gates or more are the appropriate solutions to very complex ICA designs. For example, the standard-height library cell is a design technique for nonprogrammable ASICs, where the vendors develop standard-height library cells for the implementation of large amount of functions. When implementing an ICA algorithm, the end users only need to select necessary cells that are logic level components with constant height on chip and then specify interconnections between layers such as poly, metal1, and metal2 according to their designs.

To pursue potential solutions to the InfoMax-based ICA algorithms on higher density digital ICs, the Computational Neuro Systems Laboratory, Korea Advanced

Institute of Science and Technology [6.15], is designing an ASIC chip for ICA for use it as a front end to control noise in speech recognition. All the analog, mixed-signal, and ASIC solutions depend on compact designs from end users at the beginning stage and fabrication of application-specific chips from hardware companies at the final stage. The chip size and I/O reflect the compactness of individual ICA designs, whereas the fabrication parameters and voltage reflect the trend of the VLSI technology development. Obviously, “low-voltage circuit,” which directly results in squared power conservation, and “small chip size,” which requires compact circuit design, is the current trend.

6.2.2 FPGA Solutions

Among all the VLSI technologies, FPGAs provide the most economic and efficient solutions to comparatively simple ICA algorithms and could provide lower cost substitute of nonprogrammable ASICs. Unlike nonprogrammable VLSI devices, FPGAs are standard and general-purpose products fabricated by hardware companies before end users implement specific ICA designs on them. FPGAs are developed based on reconfigurable technologies, in which end users are allowed to modify their designs for multiple times and program the interconnections in a few hours instead of waiting several weeks for the final fabrication and metalization. These savings in the development expense and the turnaround time of prototyping directly lead to time-to-market reduction and profit increase. Most FPGAs contain 2000 to 2000000 logic gates and use architectures that support a balance between logic resources and routing resources. Typical FPGAs are composed of a two-dimensional array of input/output blocks, interconnects, and configurable logic blocks (CLBs) that can be customized to implement logic functions. The programmable interconnects between these CLBs allow end users to implement the multilevel logic functions [6.16, 6.17]. FPGA vendors prefabricate rows of gates and programmable connections, whereas end users specify and interconnect the programmable CLBs to perform the desired ICA algorithms. In recent years, FPGAs have become the most popularly used devices for various VLSI implementations of

ICA algorithms. In 2001, Lim et al. [6.18] implemented two small size independent component neural network (ICNN) prototypes that were based on mutual information (between input and output) maximization and output divergence minimization. The implementation was on Xilinx Virtex XCV 812E, which contains 0.25 million logic gates [6.19]. All the variables in the network were represented as fixed-point numbers. The input signals were stored in 1-Mb RAMs and iteratively read by the ICNNs until the weight-updating process converged. Two 7-neuron ICNN prototypes were implemented and comparison was conducted in terms of cost and performance to evaluate which one was more suitable for hardware implementation. In [6.20], Nordin et al. proposed a pipelined ICA architecture for potential FPGA implementation. The InfoMax algorithm programmed in MATLAB from Tony Bell [6.21] was first broken down into four modules, each of which was translated into C, and then into HDL to implement on the four-stage pipelined FPGA array. Since each FPGA in this pipeline does not have data dependence with others, all blocks could be further implemented and executed in parallel.

6.3 Fixed-Point Models

A number in fixed-point format is represented as either in 2’s complement form or in sign magnitude form. In sign magnitude representation a (t+1)-bit binary fraction has the first bit as sign bit and other bits magnitude with binary point according to the convention adopted. Any algorithm performs various arithmetic operations such as additions, subtractions, multiplications and divisions. When any of these operations is carried out with a fixed number of bit’s’ then either truncation or rounding operation is performed. Hence for every operation an error signal is introduced at the location of that operation which appears at the output as noise. The fixed-point model introduces these truncation and rounding operations at each step of the algorithm so that one gets the fixed-point output of the system, which can be compared with the corresponding output with full precision value. For each fixed-point addition and multiplication truncation and rounding operations are carried out respectively. Such a choice has been reported to introduce less mean square error in actual implementation. In the following algorithm, $[.]_F$ denotes the fixed-point value of the parameter [.] and ‘t’ denotes the finite register length or number of bits without the sign bit. With sign bit the actual register length becomes t+1. For example fixed-point addition of two numbers A and B is performed as $[A+B]_F$=add(A,B,t), where add function represents both A and B in t-bits, adds them and then truncates the result after t-bits.

The quantization effect of the finite bit product

$$C=[A*B]_F \tag{6.1}$$

is modeled by an additive error or noise ‘n’

$$C=A*B+n \tag{6.2}$$

Similar is the case for addition. Certain complex fixed-point operations are done by special functions mentioned below as and when required. The analysis is carried out for a mixture of two signals for the sake of simplicity which can be easily extended for more than two signals.

6.3.1 The Fast ICA Algorithm

1.Centering the data matrix X:

$$[\text{meanx}_1]_F=((((x_{11}+x_{12})_F+x_{13})_F+\ldots+x_{1N})/N)_F \tag{6.3a}$$

$$[\text{meanx}_2]_F=((((x_{21}+x_{22})_F+x_{23})_F+\ldots+x_{2N})/N)_F \tag{6.3b}$$

where meanx_1, meanx_2 and denote the mean of first row of X, mean of second row of X and number of columns of X. Mean is subtracted from the data X to get the centered data $(cx_{1i};cx_{2i})^T$.

$$cx_{1i}=(x_{1i}-[\text{meanx}_1]_F)_F \tag{6.4a}$$

$$cx_{2i}=(x_{2i}-[\text{meanx}_2]_F)_F \tag{6.4b}$$

2. The centered data is whitened by finding out its covariance matrix and then by eigenvalue decomposition. Let covX is the covariance matrix of centered data.

$$\begin{aligned}[\mathrm{covX}_{11}]_F = [((((x_{11}.x_{11})_F + (x_{12}.x_{12})_F)_F + \\ (x_{13}.x_{13})_F)_F + \ldots + (x_{1N}.x_{1N})_F)_F / N]_F\end{aligned} \tag{6.5a}$$

$$\begin{aligned}[\mathrm{covX}_{12}]_F = [((((x_{11}.x_{21})_F + (x_{12}.x_{22})_F)_F + \\ (x_{13}.x_{23})_F)_F + \ldots + (x_{1N}.x_{2N})_F)_F / N]_F = [\mathrm{covX}_{21}]_F\end{aligned} \tag{6.5b}$$

$$\begin{aligned}[\mathrm{covX}_{22}]_F = [((((x_{21}.x_{21})_F + (x_{22}.x_{22})_F)_F + \\ (x_{23}.x_{23})_F)_F + \ldots + (x_{2N}.x_{2N})_F)_F / N]_F\end{aligned} \tag{6.5c}$$

$$[\sigma]_F = [([\mathrm{covX}_{22}]_F - [\mathrm{covX}_{11}]_F)_F / (2 * [\mathrm{covX}_{12}]_F)_F]_F \tag{6.6a}$$

$$[T]_F = [\mathrm{sign}(\sigma)_F / [\mathrm{abs}(\sigma)_F + [\sqrt{[1 + [(\sigma)_F.(\sigma)_F]_F]_F}\,]_F]_F]_F \tag{6.6b}$$

where $[\sqrt{y}]_F$ represents the fixed point output of √y.

$$[c]_F = [1/[\sqrt{1 + [(T)_F(T)_F]_F}\,]_F]_F \tag{6.7}$$

$$[s]_F = [[T]_F[c]_F]_F \tag{6.8}$$

$$[E]_F = \begin{bmatrix} [c]_F & [s]_F \\ -[s]_F & [c]_F \end{bmatrix} \tag{6.9}$$

$$[D]_F = [[E]_F^T * [[\mathrm{covX}]_F * [E]_F]_F]_F \tag{6.10}$$

Here the matrix multiplications are implemented by a function which does fixed point multiplication and additions required for t-bits.

$$\text{If}\,[\mathrm{D}]_{\mathrm{F}} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}_{\mathrm{F}} \ \text{then}\,[\mathrm{D}^{-1/2}]_{\mathrm{F}} = \begin{bmatrix} [\lambda_1^{-1/2}]_{\mathrm{F}} & 0 \\ 0 & [\lambda_2^{-1/2}]_{\mathrm{F}} \end{bmatrix} \tag{6.11}$$

with $[\lambda_1^{-1/2}]_F = [\frac{1}{[\sqrt{\lambda_1}]_F}]_F$ and $[\lambda_2^{-1/2}]_F = [\frac{1}{[\sqrt{\lambda_2}]_F}]_F$

$$[\text{whitened\_matrix}]_{\mathrm{F}} = [[\mathrm{D}^{-1/2}]_{\mathrm{F}} * [\mathrm{E}^{-1/2}]_{\mathrm{F}}]_{\mathrm{F}} \tag{6.12}$$

$$[\text{whitened\_X}]_{\mathrm{F}} = [[\text{whitened\_matrix}]_{\mathrm{F}} * [\mathrm{X}]_{\mathrm{F}}]_{\mathrm{F}} \tag{6.13}$$

$$[\mathrm{Z}]_{\mathrm{F}} = [\text{whitened\_X}]_{\mathrm{F}} \tag{6.14}$$

3. Initial random vector   is chosen and norm of w is calculated as

$$[\text{norm\_w}]_{\mathrm{F}} = [\sqrt{[\mathrm{w}_1 * \mathrm{w}_1]_{\mathrm{F}} + [\mathrm{w}_2 * \mathrm{w}_2]_{\mathrm{F}}}]_{\mathrm{F}} \tag{6.15}$$

$$[\mathrm{w}]_{\mathrm{F}} = ([\mathrm{w}_1 / [\text{norm\_w}]_{\mathrm{F}}]_{\mathrm{F}}, [\mathrm{w}_2 / [\text{norm\_w}]_{\mathrm{F}}]_{\mathrm{F}}) \tag{6.16}$$

The normalized fixed-point value of   is given in (6.17).

4. ICA iteration process in fixed point mode is given by

$$[\mathrm{u}]_{\mathrm{F}} = [[\mathrm{Z}]_{\mathrm{F}}^{\mathrm{T}} * [\mathrm{w}]_{\mathrm{F}}]_{\mathrm{F}} \tag{6.17}$$

$$[\mathrm{u}^2]_{\mathrm{F}} = [[\mathrm{u}]_{\mathrm{F}} * [\mathrm{u}]_{\mathrm{F}}]_{\mathrm{F}} \tag{6.18}$$

$$[\mathrm{u}^3]_{\mathrm{F}} = [[\mathrm{u}^2]_{\mathrm{F}} * [\mathrm{u}]_{\mathrm{F}}]_{\mathrm{F}} \tag{6.19}$$

In (6.18) and (6.19), fixed point multiplication is performed on element-by-element basis.

$$[Z_u]_F = [[Z]_F * [u^3]_F]_F \tag{6.20}$$

$$[A]_F = [[Z_u]_F / N]_F \tag{6.21}$$

$$[w]_F = [[A]_F - [3*[w]_F]_F]_F \tag{6.22}$$

$$[B_i]_F = [[B_i]_F * [B_i]_F^T]_F \tag{6.23}$$

for i = 1,2 and where

$$[B_1]_F = \begin{bmatrix} [w_1]_F & 0 \\ [w_2]_F & 0 \end{bmatrix}$$ for the first independent

component (IC)

and $$[B_2]_F = \begin{bmatrix} [w_1]_F & [w_1\_new]_F \\ [w_2]_F & [w_2\_new]_F \end{bmatrix}$$ for the second IC.

where $\begin{bmatrix} [w_1\_new]_F \\ [w_2\_new]_F \end{bmatrix}$ is the new random vector for second IC.

$$[B_w]_F = [[B_i]_F * [w]_F \tag{6.24}$$

$$[w]_F = [[w]_F - [B_w]_F]_F \tag{6.25}$$

Norm of $[w]_F$ is again found out as described in step 3.

6.3.2 The BFOICA Algorithm

The preprocessing steps are modeled as described in case of fast ICA algorithm. The BFOICA iteration process is modeled for fixed-point representation as follows. The direction for bacterial search algorithm is given by Delta. The fixed-point representation of Delta is

$$[Delta]_F = [[2*[round(rand(p,1))-1]_F]_F .* rand(p,1)]_F \quad (6.26)$$

where rand and round are the random and rounding operators respectively.

$$\mathrm{P} = [\mathrm{rand}(2,1)]_\mathrm{F} \ \text{ and } \ \mathrm{w} = [\mathrm{P}]_\mathrm{F}$$

$$[u]_F = [[w_1 / [norm\_w]_F]_F * y]_F \quad (6.27)$$

$[\mathrm{norm\_w}]_\mathrm{F}$ is norm of w and represented as in (6.15).

$$[\mathrm{M}]_\mathrm{F} = [\mathrm{mean}[\mathrm{u}^4]_\mathrm{F}, 2]_\mathrm{F} \quad (6.28)$$

The nutrient function in fixed point is represented as

$$[\mathrm{J}]_\mathrm{F} = [1/\mathrm{abs}[\mathrm{M}-3]_\mathrm{F}]_\mathrm{F} \quad (6.29)$$

Where abs is the absolute operator and M is represented in (6.28).

In the next iteration the column vector is incremented by a constant step of *runlengthunit (r)* as follows.

$$[\mathrm{w}]_\mathrm{F} = [\mathrm{w}]_\mathrm{F} + [[\mathrm{r}*[\mathrm{Delta}]_\mathrm{F}]_\mathrm{F} / [\sqrt{[[\mathrm{Delta}]_\mathrm{F}^{'} * [\mathrm{Delta}]_\mathrm{F}]_\mathrm{F}}]_\mathrm{F}]_\mathrm{F} \quad (6.30)$$

6.3.3 The CGAICA Algorithm

The fixed-point models for preprocessing steps like centering and whitening remain same as described above for BFOICA algorithm. The decoding of parent and children population is carried out as follows

$$\begin{aligned}&[\text{original decoded value}]_F = \\ &\qquad [\text{range min} + [[[\text{range max} - \text{range min}]_F / \\ &\qquad\quad [2^{[n_1/p]_F} - 1]_F]_F * \text{decoded value}]_F]_F \end{aligned} \tag{6.31}$$

In the above equation range min and range max are the minimum and maximum range for the GA optimization process.   and   represent the number of bits and number of parameters to be optimized in binary coded GA.

The fitness function using fixed point arithmetic is represented as

$$[J]_F = [\text{abs}[M - 3]_F]_F \tag{6.32}$$

where $[M]_F$ is represented in (6.28).

Since binary coded GA is used for optimization in CGAICA, number of bits in GA is also kept same as the finite register length for the fixed-point analysis.

6.4 Architectural Model for Fast ICA Algorithm:

1. Centering the data matrix X:

The incoming sampled data is fed to the centering block shown in Fig.6.1. In this block the mean values of both data sets X1 and X2 are found out and subsequently each data is subtracted from the mean of the corresponding data set to give the centered data.

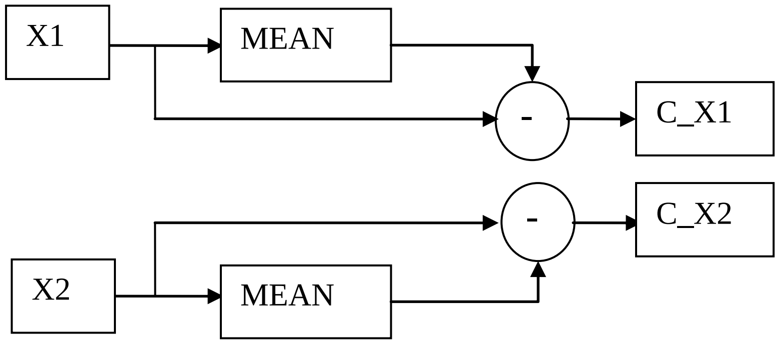

Fig 6.1 Architecture for Centering of the data

2. Whitening:

The centered data is whitened by first finding its covariance matrix and then by performing eigenvalue decomposition. This is carried out as depicted in Fig 6.2. Each incoming centered data samples of first mixed signal x_{1i} and centered data samples of second mixed signal x_{2i} are squared and cross multiplied. The means of each of the terms x_{1i}^2, x_{2i}^2 and $x_{1i}x_{2i}$ give the elements of the covariance matrix. The eigenvalue decomposition of this covariance matrix covX is carried out as proposed by Brent and Luk in [6.11] using equations (6.33) – (6.39) which is performed by the EVD block shown in Fig 6.2.

If $$\mathrm{cov}X = \begin{bmatrix} a & b \\ b & d \end{bmatrix} \tag{6.33}$$

then $$\sigma = \frac{d-a}{2b} \tag{6.34}$$

$$t = \frac{\mathrm{sign}(\sigma)}{\mathrm{mod}(\sigma) + \sqrt{1+\sigma^2}} \tag{6.35}$$

$$c = \frac{1}{\sqrt{1+t^2}} \tag{6.36}$$

$$s = c \times t \tag{6.37}$$

$$E = \begin{bmatrix} c & s \\ -s & c \end{bmatrix} \tag{6.38}$$

$$D = E^T(\mathrm{cov}X)E \tag{6.39}$$

where E is the eigenvector matrix and D is the eigen value matrix and these are outputs of EVD block. The D values are inputs to the INV_SQRT block and E values are inputs to MATMUL block which is a dedicated block for matrix multiplication.

If $D = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}$ then $D^{-1/2} = \begin{bmatrix} \lambda_1^{-1/2} & 0 \\ 0 & \lambda_2^{-1/2} \end{bmatrix}$

with $\lambda_1^{-1/2} = \dfrac{1}{\sqrt{\lambda_1}}$ and $\lambda_2^{-1/2} = \dfrac{1}{\sqrt{\lambda_2}}$ (6.40)

The INV_SQRT block performs the above computations and outputs to MATMUL block.

$$\mathrm{whitening\_matrix} = D^{-1/2}E^T \tag{6.41}$$

$$\mathrm{whitened\_X} = \mathrm{whitening\_matrix} \times X \tag{6.42}$$

$$\mathrm{whitened\_X} = WX \tag{6.43}$$

Initial random vector $w = [w_1 \quad w_2]^T$ is chosen

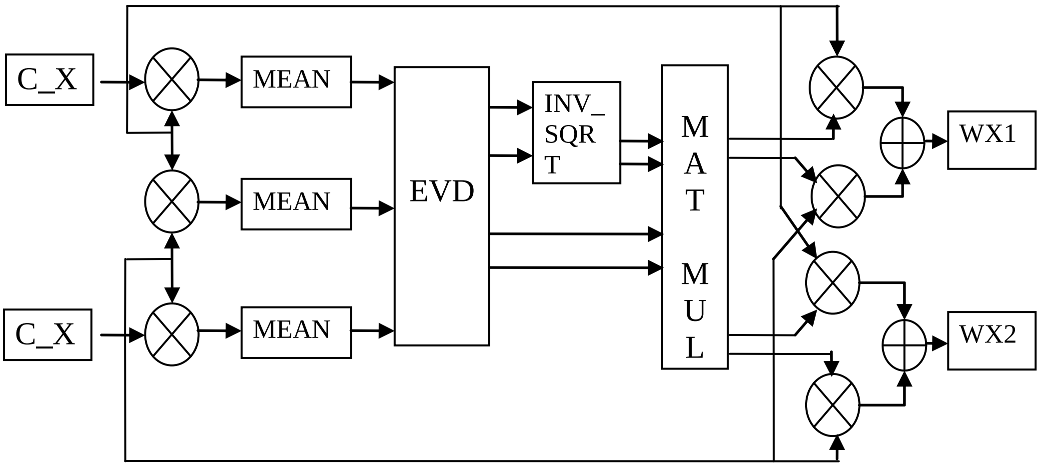

Fig 6.2 Data whitening architecture

3. ICA Iteration process:

Fig 6.3 shows the fixed-point ICA iteration for a single unit. The random vector w is multiplied with the whitened data to get $u = w_1.wx_1 + w_2.wx_2$.After the computation of $u^3.WX$ its mean is found out. This is then subtracted from ($3 \times w$). This result is normalized as described in step 5. The normalized value w_new is compared with the old value w_old and if the values do not match then w_new fed back to the input of the block through a multiplexer and also stored as w_old in a register for the purpose of comparison. When w_new = w_old then this value is given to the output as the converged vector w which gives one independent component.

For finding the other independent component a new random vector w is assumed and it is decorrelated with the earlier w and is again put to the iteration process for getting an optimized converged value.

Norm of w is calculated as

$$norm = \sqrt{w_1.w_1 + w_2.w_2}$$

Then w_{1n} and w_{2n} are computed as (6.44)

$$w_{1n} = \frac{w_1}{norm} \tag{6.45}$$

$$w_{2n} = \frac{w_2}{norm} \tag{6.46}$$

The normalized *w* is then found as

$$w_n = w_{1n} + w_{2n} \tag{6.47}$$

This is illustrated in Fig 6.4.

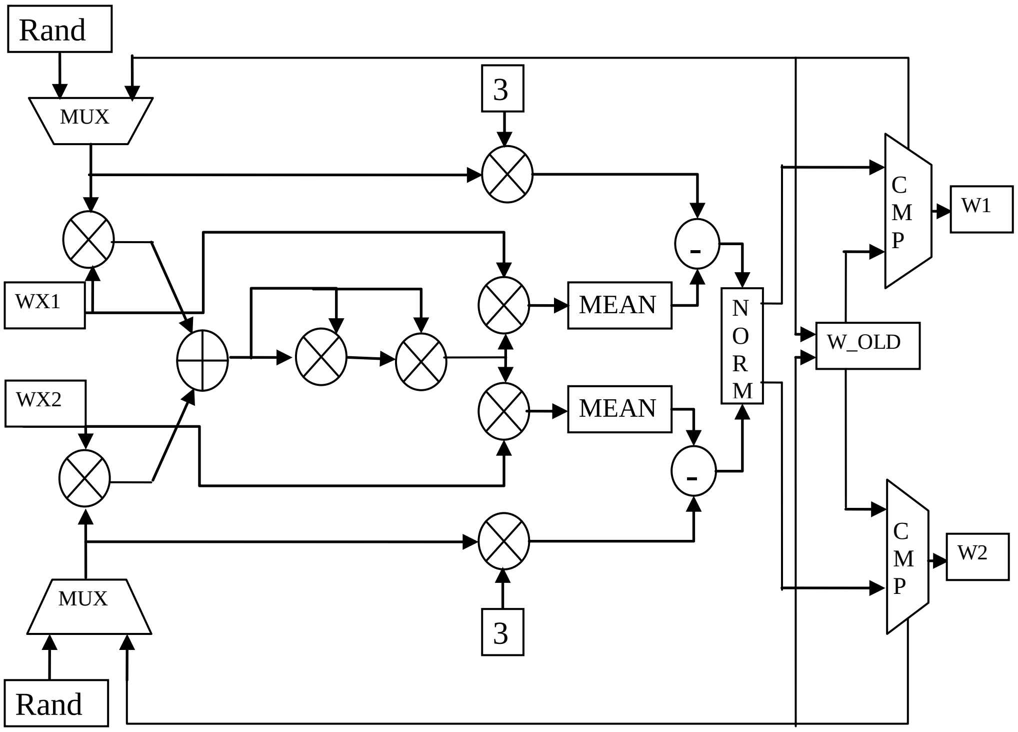

Fig 6.3 Iteration Scheme for fast ICA for one unit with Kurtosis optimization

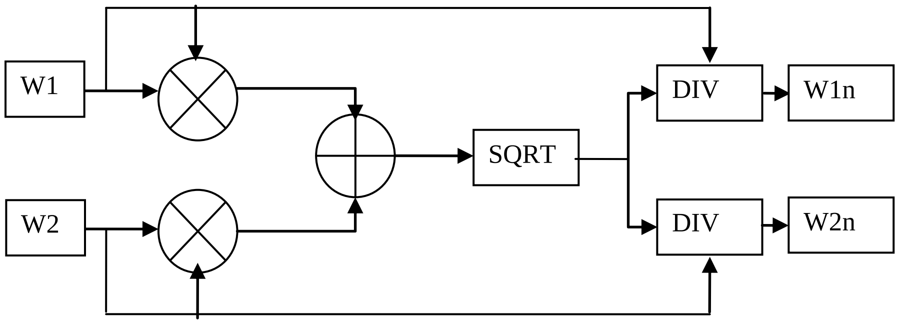

Fig 6.4 Norm Computation

6.5 Experimental Set up

For evaluating the performance of fixed-point ICAs simulation study of fast ICA, BFOICA and CGAICA algorithms are carried both in fixed and floating point arithmetic. In the simulation experiment study fixed-point as well as floating point programs of fast ICA, BFOICA and CGAICA algorithms are written. The fixed-point programs are equivalent to fixed-point machines capable of simulating the operations of any word size. With a known value of t, the fixed-point iteration for BFOICA algorithm with kurtosis as the optimization function is run and the mean square error (MSE) is computed.

$$\mathrm{MSE} = \frac{\sum_{t=1}^{n} (s(t) - y(t))^2}{n} \tag{6.47}$$

The value of t is varied from 7 to 32 and the corresponding MSEs are computed. The same procedure is followed for CGAICA algorithm with kurtosis as optimization function. From the MSEs and output signal powers the noise to signal ratio (NSR) in dB for each value of t is computed for both the algorithms.

$$NSR = 10 * \log\left(\frac{\sum_{t=1}^{n}(s(t) - y(t))^2}{\sum_{t=1}^{n}(s(t))^2}\right) \tag{6.48}$$

When t=11 the performance of various fixed-point ICAs is found to be acceptable and is considered for implementation. However due to fractional division there is an unavoidable overflow of bits which is also considered. Fixed-point analysis at various nodes of division is carried out in details. It is observed that additional 4 bits are necessary to accommodate for the extra bits due to this overflow before the binary point in the fixed-point representation. Since we follow a sign and magnitude representation of numbers throughout our implementation, an extra bit is allocated for the sign representation. Fig 5.5 shows the bit representation format.

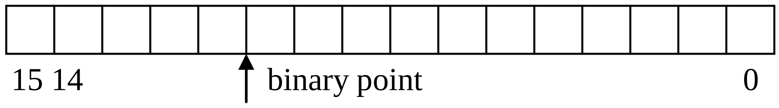

Fig. 6.5 Bit representation for Implementation

In the above format bit 15 is the sign bit, bits 14-11 represent the integral part, bits 10 to 0 represent the fractional part and binary point is after bit 11.

With 16 bit real fixed-point representation as shown in Fig 6.5, VHDL codes for the preprocessing steps of centering and whitening and fixed-point iteration of fast ICA algorithm with kurtosis as the optimization function are written. They are synthesized and fitted using Altera’s Quartus II tool [6.17]. Quartus II is ALTERA’s design and synthesis tool for ALTERA FPGA family. Stratix II device EP2S15F484C3 is used as the target FPGA for testing different performance parameters. The synthesis, placement and routing and timing analysis are carried out with settings in the Quartus II tool for speed optimization and optimization for balanced speed and area. In the first setting the tool makes efforts to optimize the speed of the implemented design without bothering for the area aspect. In the second setting the tool makes equal efforts to optimize both the speed and the area in the design.

6.6 Analysis of Results

The simulation studies, of fixed-point models as described in the previous section for fast ICA, BFOICA and CGAICA algorithms with kurtosis as contrast function are performed [6.10]. The mean square error (MSE) decreases with increase in the register length for all cases as depicted in Table 6.1. The noise to signal ratio (NSR) parameter in dB as a function of register length has been plotted. Fig 6.6 shows the NSR variation with the variation of bit length for BFOICA and CGAICA algorithms.

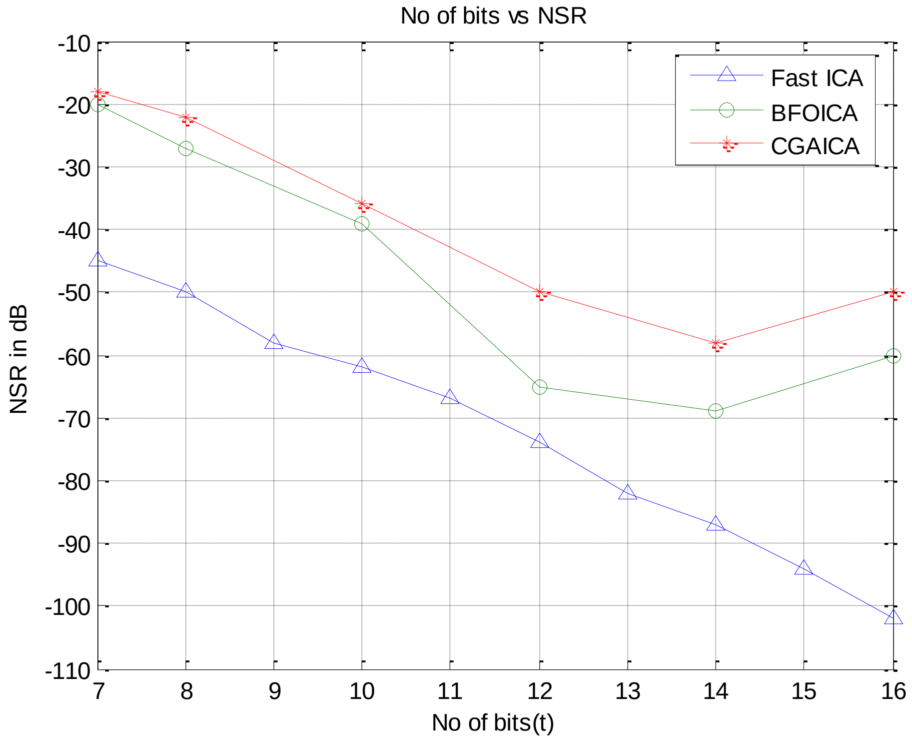

Fig 6.6 NSR variation with bit length for Fast ICA, BFOICA and CGAICA with kurtosis as contrast function

Table 6-1 Comparison of MSE of Different ICAs at Different Bit Lengths

No of Bits	MSE		
	Fast ICA	BFOICA	CGAICA
7	2.83×10^{-5}	9.3×10^{-3}	1.56×10^{-2}
8	8.16×10^{-6}	1.9×10^{-3}	5.2×10^{-3}
10	5.8×10^{-7}	1.0997×10^{-4}	2.2334×10^{-4}
12	3.66×10^{-8}	3.0983×10^{-7}	9.43×10^{-6}
14	1.45×10^{-9}	1.0714×10^{-7}	1.5372×10^{-6}
16	8.5×10^{-11}	8.4878×10^{-7}	8.4302×10^{-6}

The plots clearly indicate that NSR decreases with increase in t in both the cases. At register (bit) lengths (t<16) the BFOICA consistently performs better than the CGAICA. Also it is observed that at higher bit lengths CGAICA yields comparable or even better performance. This is because increasing the number of bits of binary coded GA tends towards a real coded GA. The NSR is observed to be minimum at register length 14 for both the algorithms. The NSR fluctuation at certain bit lengths for both the algorithms is due to the need of variations of the different tuning parameters that greatly influence the computational overhead. NSR performance for fixed-point fast ICA is better than both fixed-point BFOICA and fixed-point CGAICA algorithms. This is because BFOICA and CGAICA have more computational complexity in comparison to fast ICA. Hence due to more number of operations more quantization noise is generated at the system output due to more rounding and truncation operations. It is interesting to observe that at register length around 14 all the ICA algorithms have identical MSE values. This observation indicate that word length beyond 14 has minimal effect on the performance of different ICA algorithm studies. Fig 6.7 and Fig 6.8 show the original signals (a simple case of rectangular wave and noise has been considered) and the mixed signals respectively.

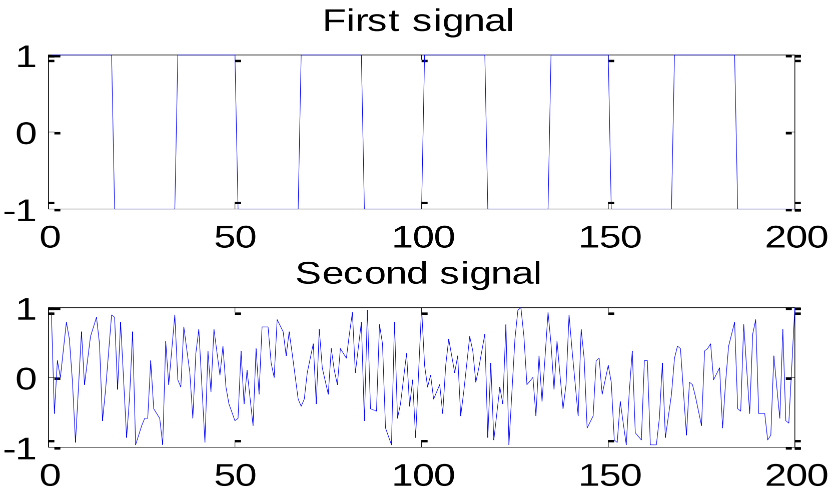

Fig.6.7 Original Rectangular wave and Random Noise Signals

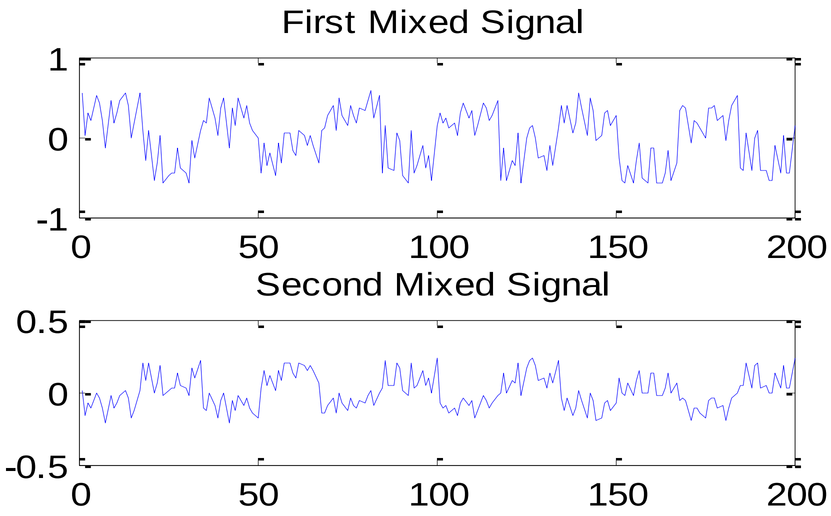

Fig.6.8 Linear Mixture of rectangular wave and random noise Signals

Figs 6.9 and 6.10 display the separation with register lengths 8 and 16 respectively by using Fast ICA algorithm.

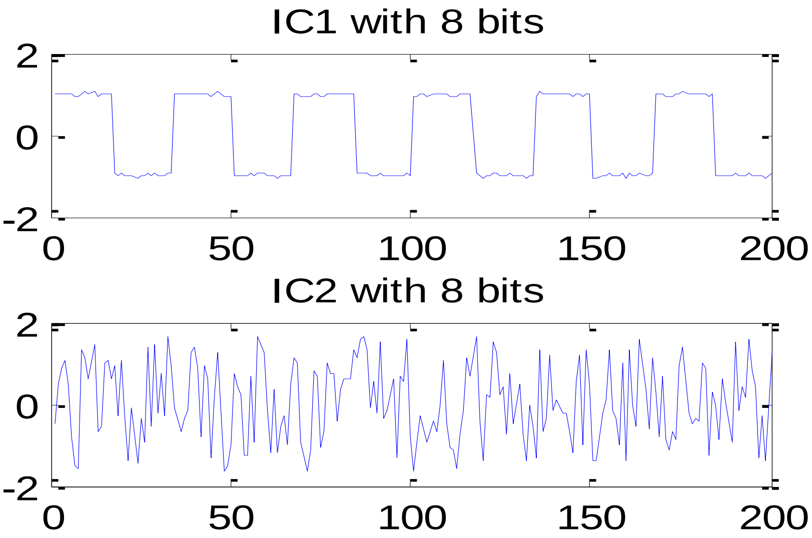

Fig.6.9 Separated Signals for Fast ICA with 8 bits

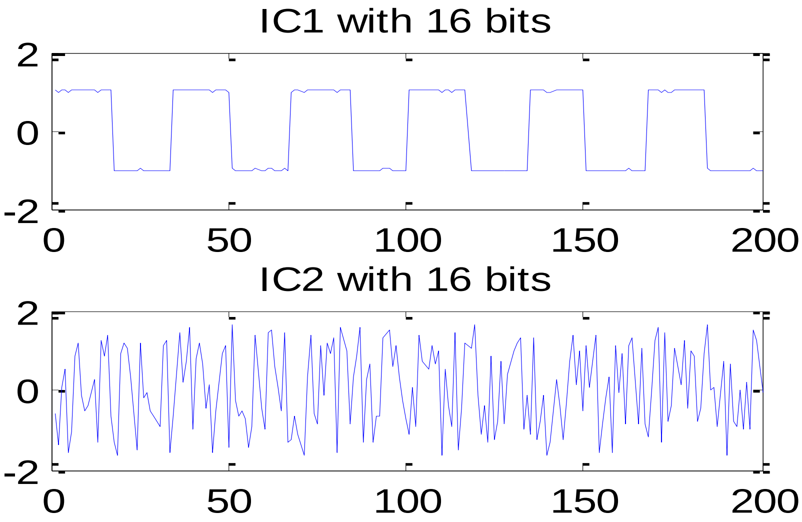

Fig 6.10 Separated Signals for Fast ICA with 16bits

The separation of signals with register lengths of 8 and 16 are shown in Figs 6.11 and 6.12 respectively for the BFOICA algorithm.

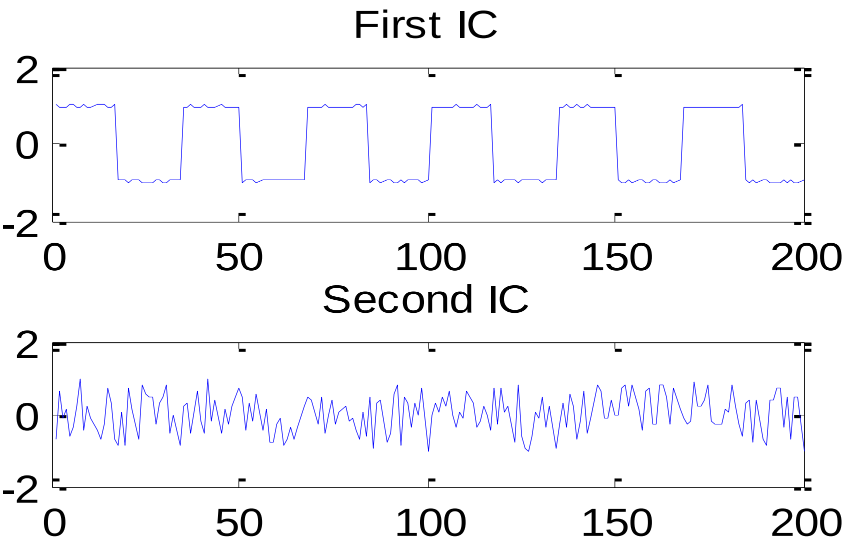

Fig 6.11 Separated Signals for BFOICA with 8 bits

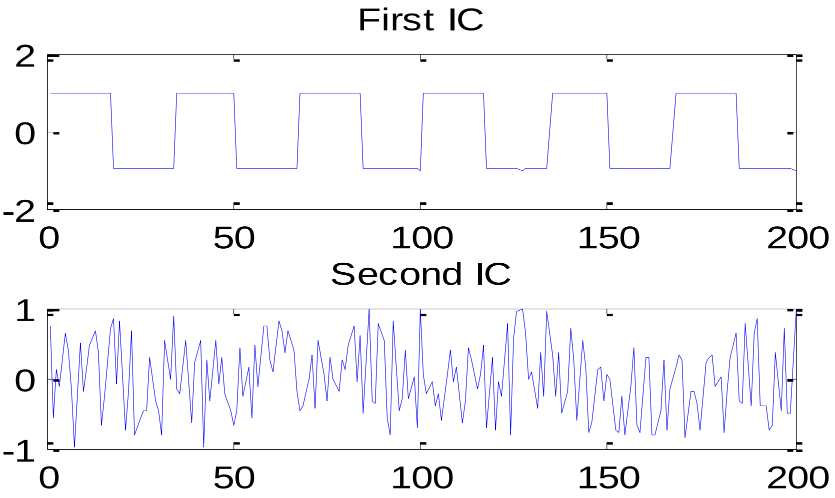

Fig 6.12 Separated Signals for BFOICA with 16bits

The signal separation by CGAICA algorithm with register lengths 8 and 16 are depicted in Fig 6.13 and Fig 6.14 respectively.

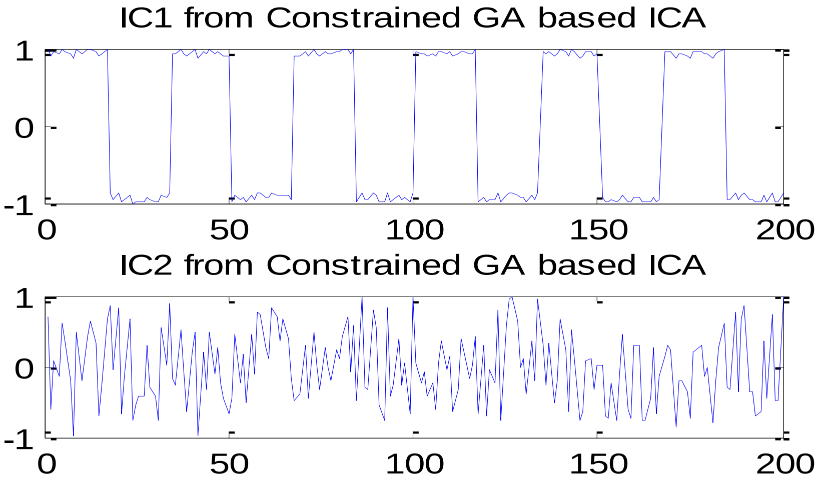

Fig 6.13 Separated Signals for CGAICA with 8bits

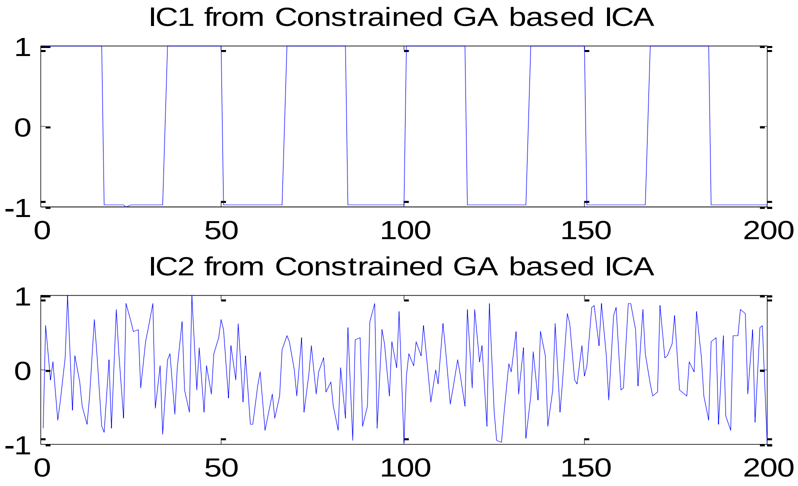

Fig 6.14 Separated Signals for CGAICA with 16 bits

It can be well observed that separation with BFOICA is better than that with CGAICA with 8 bits and also same is the case for 16 bits. But under limited word length constraint the fast ICA separation performance is better than both BFOICA and CGAICA algorithms.

As explained in previous sections an extra 5 bits is necessary to accommodate the sign bit and overflow due to fixed-point operations. With this 16 bit representation the fast ICA algorithm with kurtosis as optimization function when implemented with balanced speed and area optimization settings works with a maximum frequency of 6.59 MHz. The same when implemented with only speed optimization settings works with a maximum frequency of 6.61 MHz. The resources utilized for both cases are summarized in Table 6.2. It is also observed that for speed optimization more resources are utilized than the case when there is a balanced optimization for speed and area. The maximum frequency can be improved significantly by register pipelining in the combinational blocks.

Table 6-2 Synthesis Results with Different Optimization Settings in the Quartus II Synthesis tool

Optimization Setting	Frequency (MHz)	Combinational Elements	ALMs	Registers	DSP Blocks Utilized
Balanced	6.59	8310	5149	1031	66
Speed	6.61	9927	6352	1031	66

The FPGA implementation of fast ICA algorithm gives an approximate knowledge of speed and logic utilization in the process. The CGAICA and BFOICA algorithms being more complex than fast ICA needs dedicated efforts .This problem has not been attempted in the present thesis.

6.7 Conclusions

The present Chapter studies the effect of finite register length on the performance of two different evolutionary computation based ICA algorithms BFOICA and CGAICA and also the most popular fast ICA algorithm. Extensive simulation studies reveal that kurtosis based CGAICA yields higher MSE compared to kurtosis based BFOICA. Further for the same bit length, the fixed-point BFOICA offers substantially low MSE compared to the CGAICA (kurtosis based) and fixed-point fast ICA offers better performance than both evolutionary computation based ICA algorithms. The separation ability of fixed-point ICA depends on the number of bits used. Fixed-point BFOICA performs superior to the fixed-point CGAICA. Performance of fixed-point fast ICA in comparison to both fixed-point BFOICA and CGAICA algorithms is better due to less computational complexity involved in fast ICA algorithm. The performance of the proposed architecture for fast ICA is evaluated based on trial implementations with FPGA. It is also inferred that with speed optimization more resources are required for implementation. Further investigations in the design implementation in FPGA for timing performance and VLSI (FPGA) implementation of the complex CGAICA and BFOICA are topics of further research work.

References

[6.1] Chang-Min Kim et al. “FPGA Implementation of ICA Algorithm for Blind Signal Separation and Adaptive Noise Cancellation”, *IEEE Trans. on Neural Networks*, vol-14, No.15, pp1038-1046, 2003.

[6.2] Charoensak C. et. al.“ A Single-Chip FPGA Design for Real-time ICA-based Blind Source Separation Algorithm”, *IEEE International Symposium on Circuits and Systems,* vol-6, pp.5822-5825, 2005.

[6.3] A.M.Reza, L.Zhu, “Analysis of Error in the Fixed-Point Implementation of Two-Dimensional Discrete Wavelet Transforms”, *IEEE Trans. Circuits and Systems –I*, vol.-53, No-3, pp.641-655, Mar. 2005.

[6.4] G. Panda, R.N.Pal,.and B.Chatterjee, “Error Analysis of Good-Wino grad Algorithm Assuming Correlator Truncation Error”, *IEEE Trans. on Acoustic., Speech and Signal Processing,* pp.502-512, Apr.1983.

[6.5] A.V.Oppenheim and R.W.Schafer, *Discrete-Time Signal Processing,* International Edition, Pretice Hall, New Jersey, 1989.

[6.6] D.P.Acharya, G.Panda and Y.V.S.Lakshmi, “Fixed-point Error Evaluation of Fast ICA and Algebraic ICA Algorithms”, *IEEE International Conference on Industrial Technology,* Mumbai, India, Dec. 2006.

[6.7] D.P.Acharya, G.Panda and Y.V.S.Lakshmi, “FPGA Implementation Of A Novel Optimized Fixed-Point VLSI Architecture For Fast ICA Algorithm”, *IEEE International Conference on Signal and Image Processing*, Hubli, Karnataka, 2006.

[6.8] D.P.Acharya, G.Panda, S.Mishra and Y.V.S.Lakshmi, “Bacteria Foraging Based Independent Component Analysis”, *IEEE International Conference on Computational Intelligence and Multimedia Applications*, Sivakasi, India, Dec. 2007.

[6.9] D.P.Acharya, G.Panda and Y.V.S.Lakshmi, “Constrained Genetic Algorithm Based Independent Component Analysis”, *IEEE Congress on Evolutionary Computation*, Singapore, Sept. 2007.

[6.10] D.P.Acharya, G.Panda and Y.V.S.Lakshmi, “Effects of Finite Register Length on Constrained Genetic Algorithm based ICA and Bacteria Foraging based ICA Algorithms”, *International Conference on Signal Processing and Communication Networks*, Chennai, Jan 2008.

[6.11] R.P.Brent and F.T.Luk, “The solution of singular-value and symmetric eigen-value problems on multiprocessor arrays”, *SIAM J. Scientific and Statistical Computing*, 6, pp.69-84, 1985.

[6.12] H.Du,H.Qi and X.Wang, “Comparative Study of VLSI Solutions to Independent Component Analysis”, *IEEE Trans. Industrial Electronics*, vol.54, No.1, 2007.

[6.13] K.S.Cho and S.Y.Lee, “Implementation of InfoMax ICA algorithm with analog CMOS circuits”, *In Proc. Int. Workshop Independent Component Analysis and Blind Signal Separation,* Vancouver, BC, Canada, Dec. 2001, pp. 70-73.

[6.14] G. Cauwenberghs, “Neuromorphic autoadaptive systems and independent component analysis,” Johns Hopkins Univ., Baltimore, MD. Tech. Rep (2003).. [Online]. Available: http://bach.ece.jhu.edu/gert/yip/

[6.15] Computational NeuroSystems Laboratory, Digital Implementation of Independent Component Analysis Algorithm, 2003, Daejeon, South Korea: Dept. Biosystems, Korea Advanced Inst. Sci. Technol. Tech. Rep. [Online]. Available: http://cnsl.kaist.ac.kr/Research/kscho/icachip.htm

[6.16] www.xilinx.com

[6.17] www.altera.com

[6.18] A. B. Lim, J. C. Rajapakse, and A .R. Omondi, “Comparative study of implementing ICNNs on FPGAs,” *Proc. Int. Joint Conf. Neural Netw.,* , vol. 1, pp. 177–182, Jul. 2001.

[6.19] Xilinx, Virtex-E 1.8 V Extended Memory Field Programmable Gate Arrays, 2002. [Online]. Available: http://direct.xilinx.com/bvdocs/ publications/ds025-1.pdf

[6.20] A. Nordin, C. Hsu, and H. Szu, “Design of FPGA ICA for hyperspectral imaging processing,” *Proc. SPIE, Wavelet Appl. VIII,* vol. 4391, pp. 444–454, 2001.

[6.21] A. J. Bell and T. J. Sejnowski, “An information maximization approach to blind separation and blind deconvolution,” *Neural Comput.*, vol. 7, no. 6, pp. 1129–1159, Nov. 1995.

[6.22] H.Du and H.Qi, “An FPGA Implementation of parallel ICA for Dimensionality Reduction in Hyperspectral Images”, *Proc. of IEEE International Symposium on Geoscience and Remote Sensing,* 5, 3257 – 3260, 2004.

Chapter

VII

BLIND JAMMER SUPPRESSION IN DS-SS SYSTEMS USING ICA TECHNIQUES

7.1. Introduction

Spread spectrum communications, with its inherent interference attenuation capability, has over the years become an increasingly popular technique for use in diversity of systems. The systems range from antijam systems for military wireless communications to code division multiple access systems for commercial mobile communications, to systems designed to combat multipath. The properties such as antijamming, antiinterference, low probability of intercept, multiple user random access communications with selective addressing capability, high resolution ranging and accurate universal timing have made it so popular. In Direct Sequence Spread Spectrum (DS-SS) communications, the bandwidth of the transmitted waveform is intentionally made wider than would be necessary to transmit the information over the channel, by means of a code which is independent of the data. The data at the receiver end is despreaded and recovered by the same code being available at the receiver. The ratio of transmission and data bandwidth is called the processing gain and provides the system with a high degree of interference suppression capability. In most of the cases this gain is sufficient for proper performance of the system. However sometimes, additional interference suppression capability is needed due to limited availability of the bandwidth. Therefore, signal processing techniques have been used in addition to the DS spread spectrum receiver to enhance the processing gain, providing greater interference suppression without an increase in bandwidth [7.1, 7.2]. Traditionally filters are designed as a preprocessing step to suppress the interference and increase the signal-to-noise ratio at the correlator output of the receiver. Belouchrani and Amin [7.3] have proposed the interference mitigation scheme using blind source separation to aid conventional detection in spread spectrum receiver. They have used second order statistics based BSS technique to separate a set of independent information signals from their mixtures observed at the sensors. However second order temporal statistics makes the scheme vulnerable to signals having low temporal correlations.

Different types of ICA techniques have been discussed in Chapters 2, 3 and 4 which are used for blind source separation. ICA exploits higher order statistics and hence higher order decorrelation for jammer separation is achieved by use of ICA. K.Raju *et. al.* [7.4] used Fast ICA for jammer suppression in DS-CDMA arrays. The Evolutionary computation based ICA techniques CGAICA and BFOICA have been discussed in Chapters 3 and 4 respectively. The GA, a popular evolutionary computation method has been widely used for multi user detection (MUD) in CDMA systems which use the spread spectrum technique [7.5, 7.6, 7.7]. So it becomes quite motivating to apply the evolutionary computation based ICA techniques to the direct sequence spread spectrum scenario for jammer mitigation.

In this Chapter, a novel scheme for jammer separation in DS-SS communication systems using the CGAICA and BFOICA techniques has been proposed. Simulation studies to assess the performance of the proposed schemes have been carried out. The merits and limitations of the CGAICA and BFOICA assisted jammer suppression in comparison to fast ICA assisted one are also discussed.

This Chapter is organized as follows. In Section 7.2, we present the overview of direct sequence spread spectrum communication systems. Section 7.3 discusses different sources of interference in DS SS system. In Section 7.4, the interference rejection techniques in DSSS systems are outlined. The ICA based system model for interference rejection is described in Section 7.5. In Section 7.6 the novel BFOICA/CGAICA based jammer mitigation model is proposed. In Section 7.7 the experimental set up for validation of the model is outlined. The results of the investigation has been illustrated and discussed in Section 7.8. Finally, in Section 7.9, the conclusion of the study has been provided.

7.2. Spread Spectrum Communication Systems

Spread spectrum is a means of transmission in which the signal occupies a bandwidth in excess of the minimum necessary to send the information; the band spread is accomplished by means of a code which is independent of the data, and a synchronized reception with the code at the receiver is used for despreading and subsequent data recovery [7.8].

There are many reasons for spreading the spectrum, and if done properly, a multiplicity of benefits can accrue simultaneously. Some of these are antijamming, antiinterference, low probability of intercept, multiple user random access communications with selective addressing capability, high resolution ranging, and accurate universal timing.

The means by which the spectrum is spread is crucial. Several of the techniques are “direct-sequence” modulation in which a fast pseudorandomly generated sequence causes phase transitions in the carrier containing data, “frequency hopping,” in which the carrier is caused to shift frequency in a pseudorandom way, and “time hopping,” wherein bursts of signal are initiated at pseudorandom times. Hybrid combinations of these techniques are frequently used. Here we focus only on the direct sequence spread spectrum (DS-SS) systems. This consists of superimposing upon the data bits a wide bandwidth spreading sequence. This spreading sequence is often generated from a linear feedback shift register (LFSR) and is called pseudorandom noise (PN) sequence.

To illustrate this idea in a very simple manner, let a data signal $d(t)$ be spread by multiplying it with a binary valued spreading sequence $c(t)$ such that the bandwidth of $c(t)$ is much greater than the bandwidth of $d(t)$. If the transmitted signal $d(t)c(t)$ is received in the presence of noise $n(t)$ and interference $j(t)$ then the received waveform $r(t)$ is given by

$$r(t) = c(t)d(t) + n(t) + j(t) \tag{7.1}$$

$r(t)$ is despread at the receiver by multiplication with a locally generated version of $c(t)$. Calling this new waveform as $x(t)$, we get

$$\begin{aligned} x(t) &= c^2(t)d(t) + c(t)n(t) + c(t)j(t) \\ &= d(t) + c(t)n(t) + c(t)j(t) \text{ since } c^2(t) = 1 \end{aligned} \tag{7.2}$$

The desired signal bandwidth is reduced back down to the data bandwidth and the interference has had its bandwidth increased to at least the spread bandwidth (assuming $c(t)$ and $j(t)$ are uncorrelated), since multiplication in time corresponds to convolution in frequency domain. A time domain and frequency domain representation of this is depicted in Fig 7.1 and 7.2 respectively. The bit manipulation for spreading and dispreading at transmitter and receiver ends in time domain and frequency domain have been shown in Fig 7.3 and 7.4 respectively. Hence if $x(t)$ is passed through a lowpass filter whose bandwidth is equal to that of $d(t)$, most of the energy of $c(t)j(t)$ will be filtered out and the system will enjoy a so called processing gain advantage over the interference. The jammer power and despread jammer power remain unchanged prior to lowpass filtering, since the spreading sequence $c(t)$ is normalized so that $c^2(t) = 1$. However, the effective jammer power is reduced at the output of the lowpass filter, since all jammer components outside of the data bandwidth are rejected.

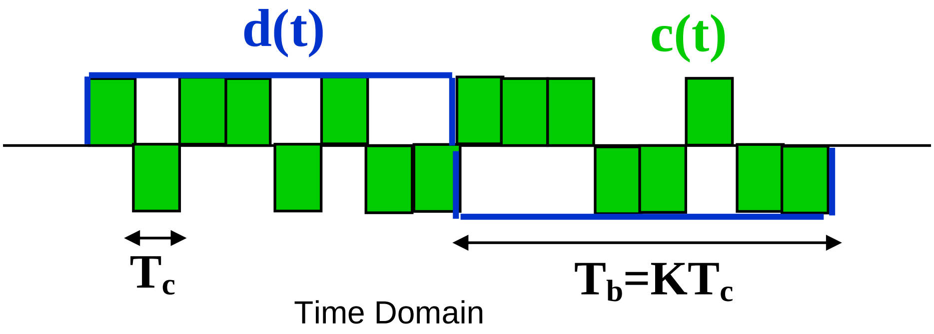

Fig 7.1 Time domain spreading

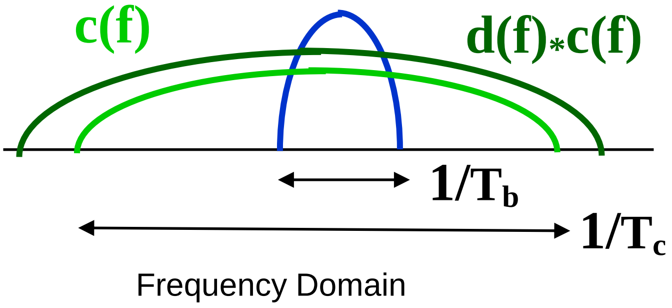

Fig. 7.2 Frequency domain Spreading

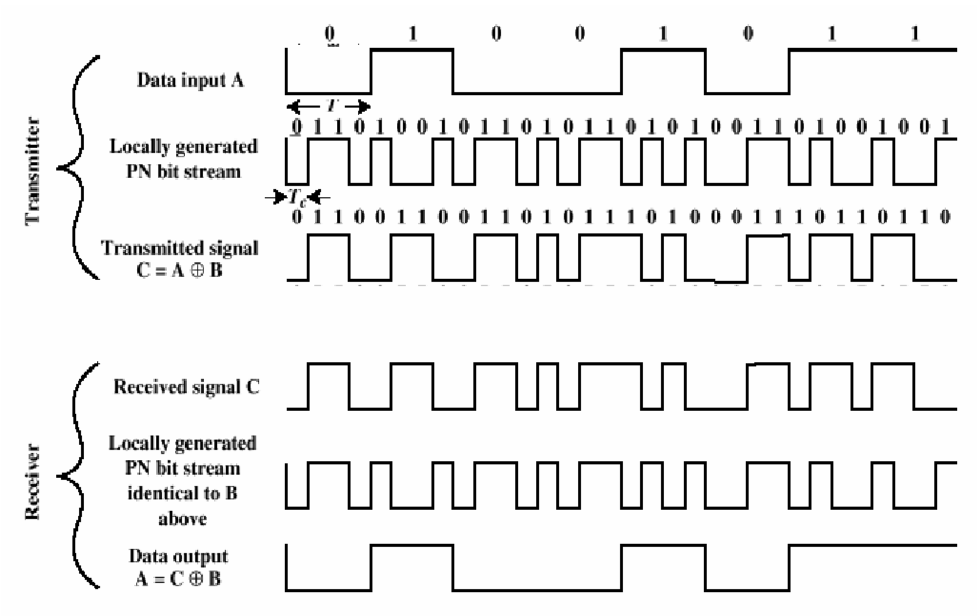

Fig. 7.3 Spreading and dispreading at Transmitter and Receiver in time domain

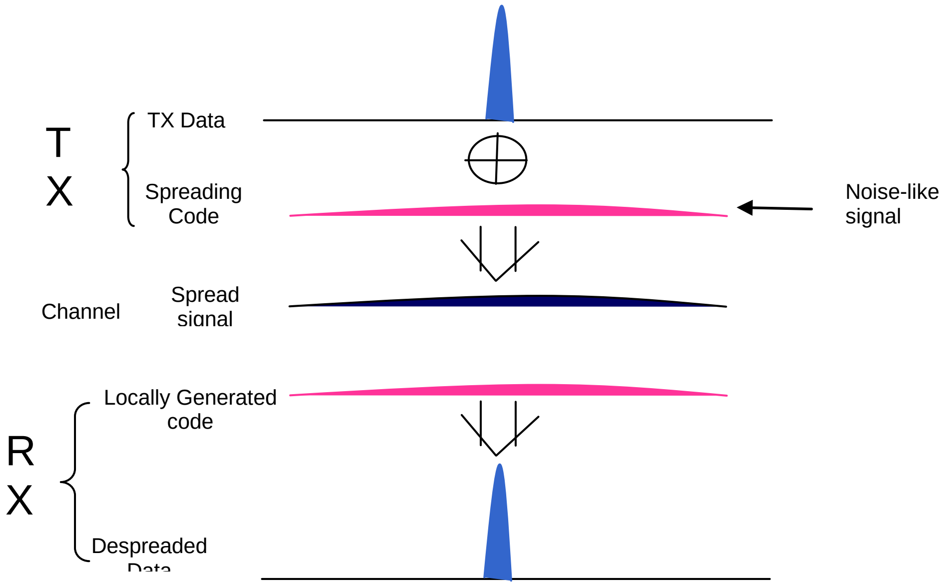

Fig. 7.4 Spreading and dispreading at Transmitter and Receiver in frequency domain

7.3. Sources of Interference

As with any (digital) communication system, interference is common in spread spectrum systems. It can range from plain noise to smart jammers. Interference can be categorized into the following types:

- interchip interference component. This is usually termed inter symbol interference (ISI) for non-spread digital modulation;
- the component due to background noise n(t);
- the other-user interference components in the cell-of-interest. This is the multiuser interference (MUI);
- interference due to users in the neighboring cells - inter-cell interference (ICI);

- external interference due to coexisting systems in the same band or due to adversaries that intentionally disrupt transmission, usually termed as jammers.

Earlier versions of spread spectrum systems were primarily designed for military communications to overcome a jamming situation, i.e., when an adversary intends to disrupt the communication. In order to disrupt the communication, the adversary needs to do two things: (a) detect that a communication is taking place, and (b) to transmit a jamming signal which is designed to confuse the receiver. Spread spectrum communications are therefore designed to make these tasks as difficult as possible. In modern CDMA systems, there is no longer a real adversary that wishes to disrupt communications. Jamming here takes place more due to the coexistence of multiple systems in the same band. This jamming is mostly unintentional by nature. Such jammers can either be continuous wave (CW, narrow-band) or pulsed (wide-band). The jamming signal can be expressed as

$$\mathrm{j(t)} = \delta_{\mathrm{p}}(\mathrm{t})\sqrt{\mathrm{J}}\exp(\mathrm{i}2\pi \mathrm{f}_{\mathrm{j}}\mathrm{t}+\phi) \qquad (7.3)$$

where $i=\sqrt{-1}$, and $\delta_p(t)=1$with a probability p during a symbol. It defines the nature of jammer. The jammer is continuous wave when $p=1$ and pulsed at the symbol level otherwise. The power, frequency and the phase of the jammer signal j(t) are denoted by J, f_j and ϕ respectively. The phase is assumed to be uniformly distributed over the interval [0, 2].

7.4. Interference Rejection in DS-SS Systems

The acquisition and tracking systems of a spread-spectrum receiver are probably the most critical components of the receiver, since if they fail to function properly, it is doubtful that the desired signal can be successfully detected. This means that the effect of interference (such as jamming) on the receiver while it is attempting to learn the correct phase position of the incoming code might be especially harmful, since the interference might not allow the receiver to acquire the signal. To address this problem, a narrow-band interference suppression filter is used to enhance the performance of a serial search acquisition scheme for a direct-sequence spread-spectrum receiver. Analytical expressions for the probabilities of error in both the search and lock modes are derived and numerical results are used to illustrate the sensitivity of the receiver to various system parameters. It is shown by Milstein [7.9] that the presence of the rejection filter can significantly improve the performance of the acquisition system. Milstein in [7.9] discusses in depth two classes of rejection schemes (both of which implement an adaptive notch filter): 1) those based upon least mean square (LMS) estimation techniques, and 2) those based upon transform domain processing structures. In these techniques the improvement is achieved subject to the constraint that the interference is relatively narrowband with respect to the DS waveform.

Interference rejection techniques for DSSS systems are numerous [7.10]. In particular, much literature exists on the adaptive notch filter as it relates to rejecting narrow band interference on a wideband DSSS signal. Decision directed adaptive filtering is another well established technique for interference rejection. Other techniques for narrowband DSSS include adaptive analog-to-digital (A/D) conversion and nonlinear adaptive filtering.

7.5. ICA Based System Model and Scheme for Mitigation

We consider a DS-SS system which uses binary phase shift keying (BPSK) for both chip and data modulation. Fig 7.5 shows the simple transmitter model where the incoming symbol sequence is spreaded by a pseudo random noise like (PN) bit sequence so that the bandwidth of the transmitted signal increases by a factor of the processing gain. The receiver is depicted in Fig 7.6. The transmitted signal which passes through the AWGN channel also gets corrupted by jammer signal and is received at the receiving antenna. The received signal is despreaded and demodulated before getting integrated over the bit duration. This comprises a matched filtering operation for the DS/BPSK signal. After dispreading, the jammer energy is spread over the PN code bandwidth and the integrator acts as a narrowband filter. The full system model with transmitter, channel and receiver is shown in Fig 7.7.

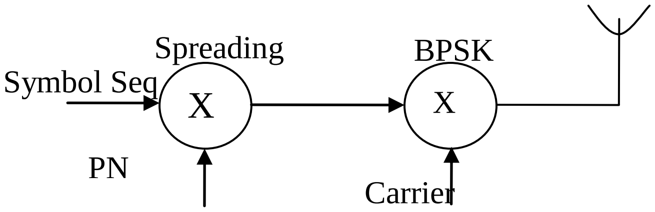

Fig. 7.5 DS-SS BPSK Transmitter

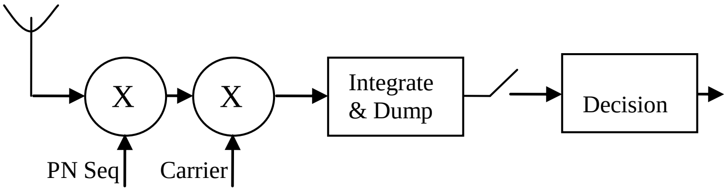

Fig.7.6 DS-SS BPSK Receiver

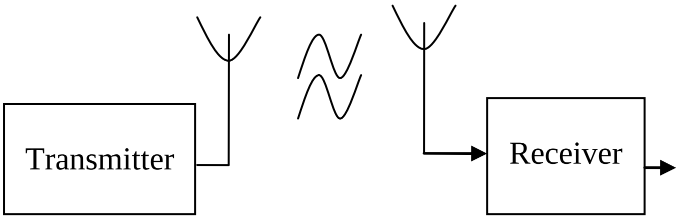

Fig. 7.7 Simple DS-SS System Model

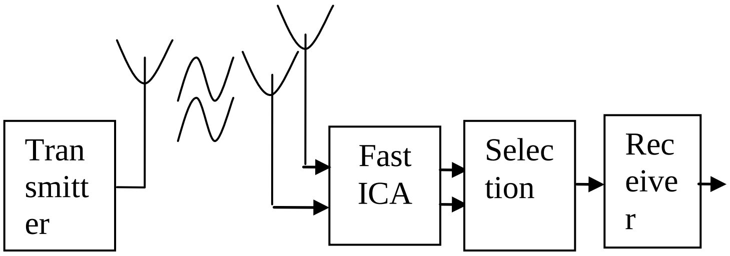

Fig. 7.8 Fast ICA based DS-SS System Model

The base band spread spectrum signal $s(t)$ is of the form:

$$s(t)=\sum_{k=-\infty}^{\infty} b_k m_k(t-kT) \qquad (7.4)$$

where $m_k(t)=\sum_{l=1}^{L} c_l^k p(t-(l-1)T_c)$ with T^{-1} as the data rate, and T_c^{-1} as the chip rate. $L=T/T_c$ is the number of chips per bit called the processing gain of the SS system. $\{b_k\}$ and $\{c_l^k\}$ represent the *k*th bit data sequence and corresponding chip sequence, and $p(t)$ is the chip pulse.

The jammer signal can be continuous wave or bit pulsed as described above. Continuous wave jammer signal is represented as

$$j(t)=\sqrt{J_p}\cos(wt+\phi) \qquad (7.5)$$

where J_p is the jammer power and w is the jammer frequency and ϕ is the phase of jammer. The jammer can also be a random jammer with a specified power J_p. Another type of jammer can be chip pulsed jamming which is a wideband jammer.

The received signal is a mixer of the desired transmitted signal $s(t)$, jammer signal $j(t)$ and additive noise $n(t)$ in the channel. At the two receiving antenna sensors the signals $r_1(t)$ and $r_2(t)$ can be modeled as

$$[r_1(t) \quad r_2(t)]^T = A * [s(t) \quad j(t)]^T + n(t) \tag{7.6}$$

where A is unknown mixing matrix which depends on the channel and $[]^T$ denotes the transpose operator.

In ICA technique for separating two signals at least two observed mixed signals are necessary. This can be achieved by having antenna diversity at the SS receiver [7.11]. Assuming the waveforms $s(t)$ and $j(t)$ to be statistically independent fast ICA algorithm is performed on the received signals $r_1(t)$ and $r_2(t)$. The ICA separates the jammer and the desired signal as depicted in Fig. 7.8. The ICA output signals are input to a selection block which finds the desired signal by rejecting the jammer signal. This selected desired signal is processed by the conventional receiver for detection. The demodulation process recovers the desired signal by dispreading and spreading the noise and any other interference signal which is easily filtered subsequently.

7.6. The Proposed BFOICA Based System Model

Though ICA is a blind technique but the permutation uncertainty inherent to ICA methods necessitates a selection block which uses a training signal at the receiver end to identify the desired spread spectrum signal. The additional selection block shown in Fig. 7.8 makes the implementation of signal processing at receiver complex due to the involvement of the signal classification problem [7.12]. To avoid this extra processing, the proposed BFOICA algorithm is used as a replacement for the fast ICA. Since in BFOICA the independent components (ICs) are recovered in a particular order, we can know the IC pertaining to the spread spectrum signal. This eliminates the necessity of the extra processing block for selection. The received signal in our scheme is incident on the BFOICA/CGAICA block which gives out the desired signal directly to the conventional spread spectrum receiver for detection as depicted in Fig. 7.9.

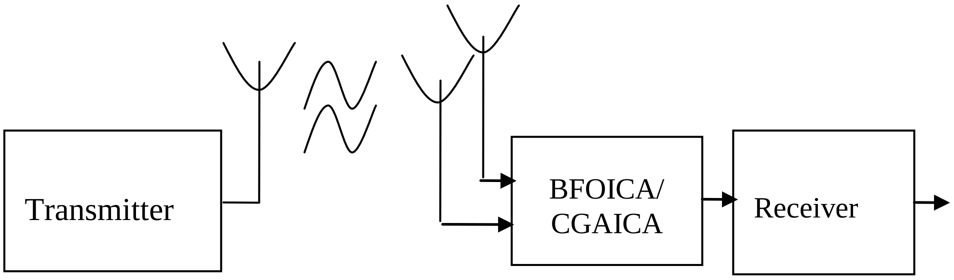

Fig. 7.9 Proposed BFOICA based DS-SS System Model

7.7. Experimental Setup

Experimental study is carried out in this section to demonstrate the performance of the proposed model and to compare the resultd with those obtained by other methods. In the simulation experiment study, an input symbol sequence of 10000 samples is considered. Programs for the transmitter, receiver and fast ICA algorithm are written to carry out computer simulations for the simple DS-SS system and ICA based DS-SS system as shown in Fig 7.7 and 7.8 respectively. Programs for CGAICA and BFOICA are also written and CGAICA/BFOICA based DS-SS system is modeled as in Fig 7.9. To the transmitted output signal, two types of weighted continuous wave jammers are added. The first type of jammer considered is a sinusoidal jammer whereas the second one is a random jammer. The transmitted signal is allowed to pass through an AWGN channel. A model of two antennas is considered to obtain the receive diversity as required for the ICA algorithm. Jammer power is changed so as to vary the signal-to-jammer ratio (SJR) in the range of -60dB to 50dB in steps of 10dB. Corresponding to each value of SJR, bit error rate

(BER) at the receiver output is computed. The variation of BER with SJR at a particular signal-to-noise ratio (SNR) is studied. The experiment is conducted at SNR values 0dB, 5dB and 10dB. The variation of BER with SJR is studied for the simple DS-SS system and DS-SS system with ICA assistance. The experiment is carried out for different PN sequence lengths. The BER is plotted against SJR for PN sequence lengths of 8, 16 and 32 bits at different SNRs. The above procedure is repeated for both types of jammers used.

7.8. Analysis of Results

[a] Case –I

The jammer considered here is a sinusoidal one. The simulation studies, on both simple DS-SS and DS-SS with fast ICA and BFOICA are performed as described in the previous section. Fig 7.10, 7.11 and 7.12 depict the plot of BER against SJR with PN sequence lengths 8, 16 and 32 bits respectively. In case of simple DS-SS system the BER is observed to decrease with increasing SJR. At higher values of SJR, BER remains almost constant at a very low value. As the length of PN sequence increases, the BER value remains constant low value. When both fast ICA and BFOICA are used for jammer mitigation the low value of BER is observed at lesser SJR values for the entire range of the SJR variation. This means at higher jammer powers also the ICA assisted receiver offers better BER values irrespective of the PN sequence length. BFOICA based technique yields almost comparable BER performance as that offered by the fast ICA.

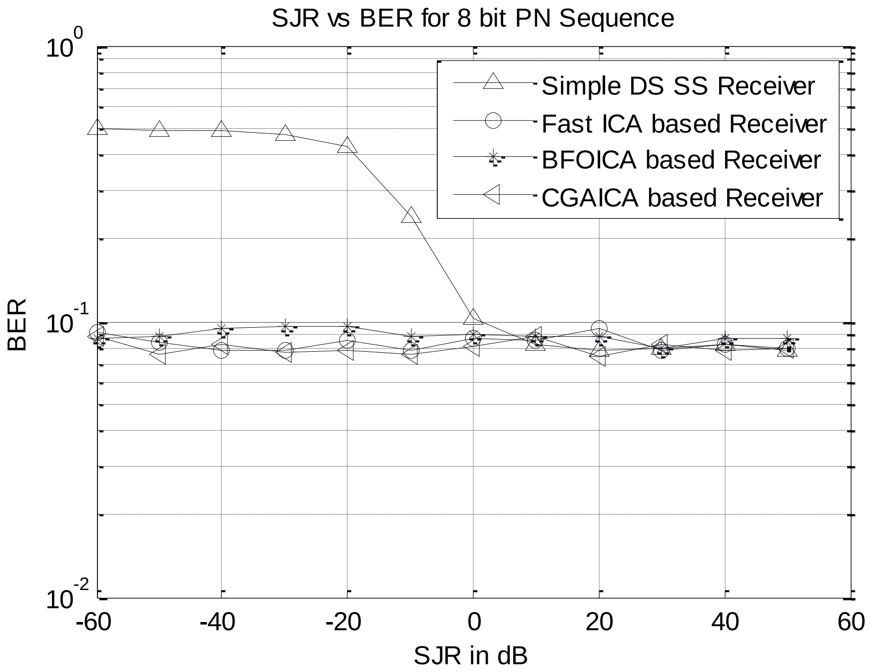

Fig 7.10 Plot of SJR vs BER without and with Fast ICA BFOICA & CGAICA for DS-SS System with 8 bits PN Sequence at SNR=0dB(Sinusoidal jammer)

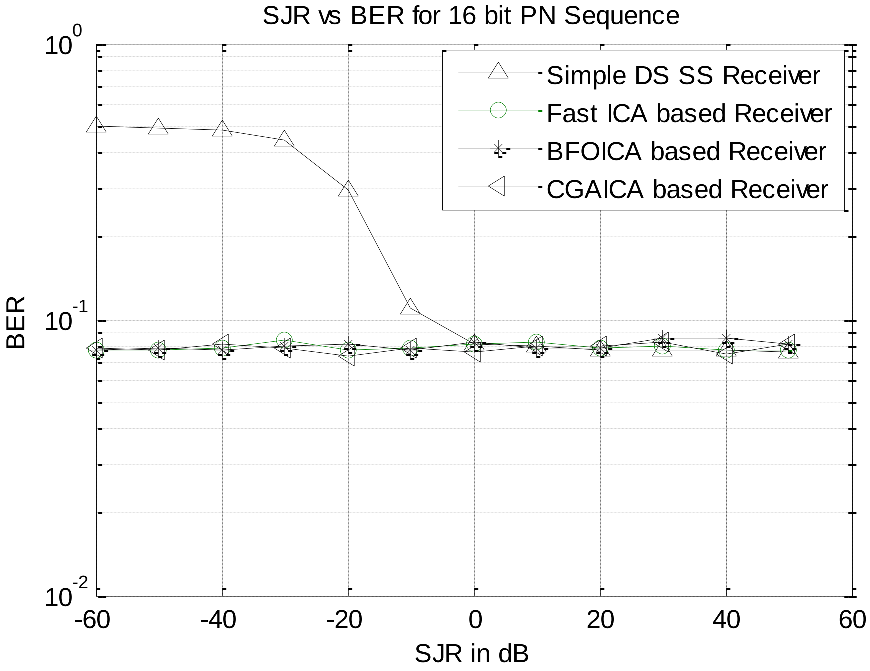

Fig.7.11 Plot of SJR vs BER without and with Fast ICA BFOICA & CGAICA for DS-SS System with 16 bits PN Sequence at SNR=0dB(Sinusoidal jammer)

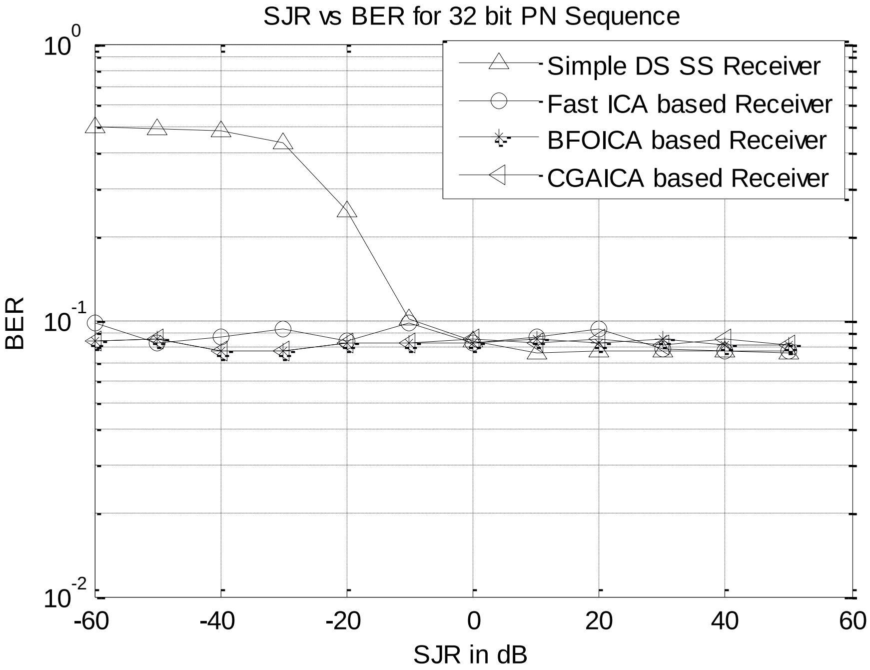

Fig.7.12 Plot of SJR vs BER without and with Fast ICA, BFOICA & CGAICA for DS-SS System with 32 bits PN Sequence at SNR=0dB(Sinusoidal jammer)

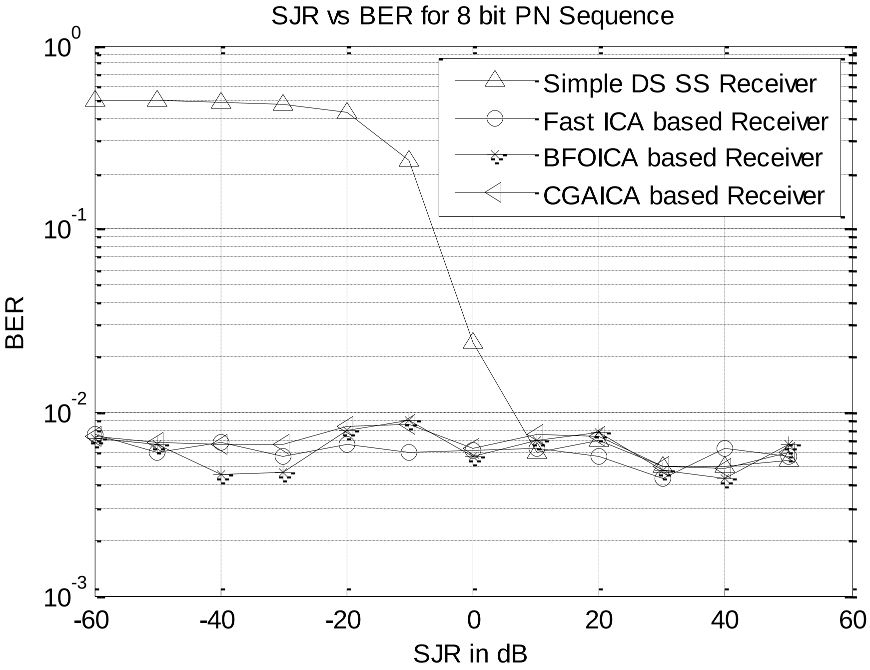

Fig 7.13 Plot of SJR vs BER without and with Fast ICA, BFOICA & CGAICA for DS-SS System with 8 bits PN Sequence at SNR=5 dB(Sinusoidal jammer)

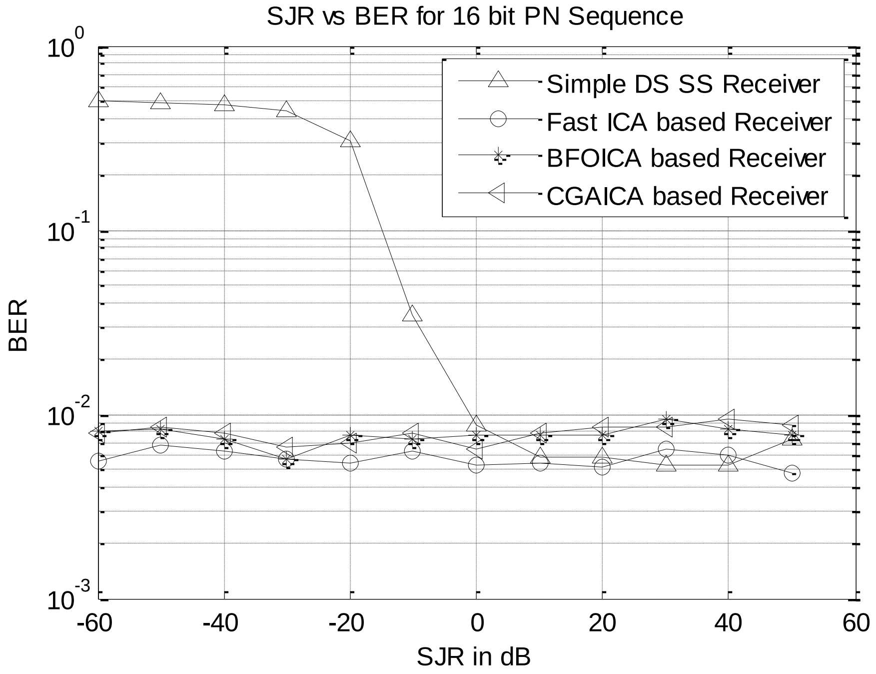

Fig 7.14 Plot of SJR vs BER without and with Fast ICA BFOICA & CGAICA for DS-SS System with 16 bits PN Sequence at SNR=5 dB(Sinusoidal jammer)

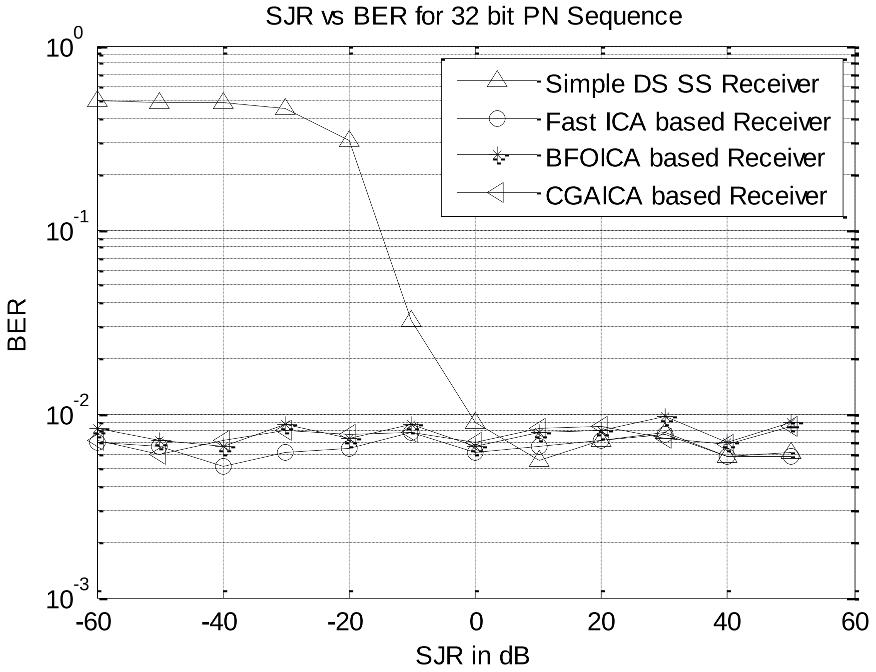

Fig.7.15 Plot of SJR vs BER without and with Fast ICA BFOICA & CGAICA for DS-SS System with 32 bits PN Sequence at SNR=5 dB (Sinusoidal jammer)

The main advantage of this ICA assisted technique is that the separation is unaffected by errors in the propagation model or any array calibration. All the above results are obtained at SNR 0dB. The results of the same experiments at SNR 5 dB are depicted in Fig. 7.13, 7.14 and 7.15 respectively. In this case the ICA assisted schemes are observed to have less BER as compared to case with 0 dB SNR. The same experiments when conducted at 10dB SNR result in constant low value of BER irrespective of jammer power.

[b] Case-II

In case of a random jammer, the jammer power and hence SJR when widely varied the BER is observed to decrease with increasing SJR values. This happens for all considered PN sequence lengths and at different SNRs (0dB and 5 dB) as observed for the previous case of sinusoidal jammers. The SJR against BER plots for PN sequence lengths 8, 16 and 32 for SNR 0dB and 5dB are depicted in Fig**.**s 7.16 -7.22 respectively.

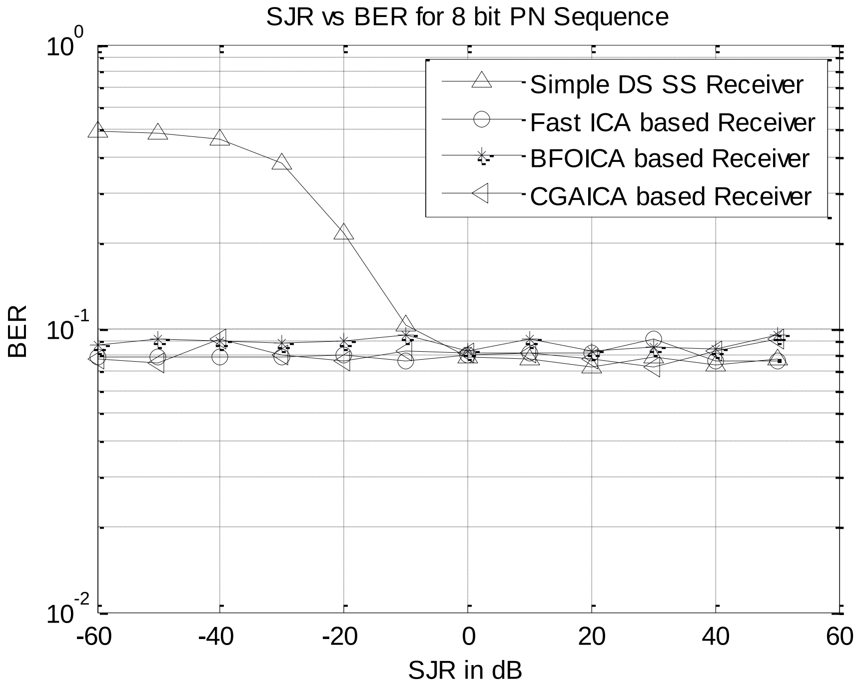

Fig 7.16 Plot of SJR vs BER without and with Fast ICA BFOICA & CGAICA for DS-SS System with 8 bits PN Sequence at SNR=0dB(Random Jammer)

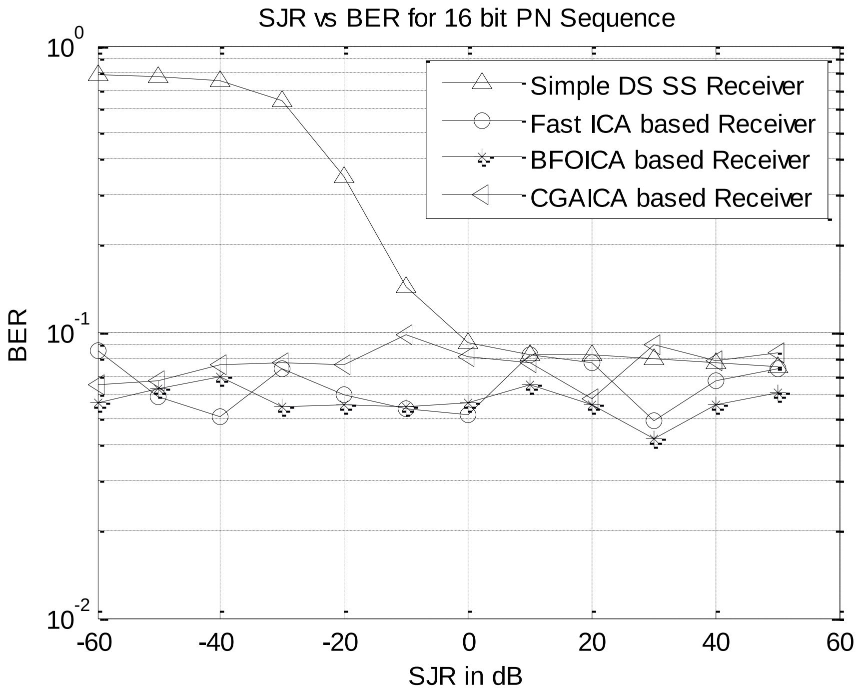

Fig 7.17 Plot of SJR vs BER without and with Fast ICA BFOICA & CGAICA for DS-SS System with 16 bits PN Sequence at SNR=0dB(Random Jammer)

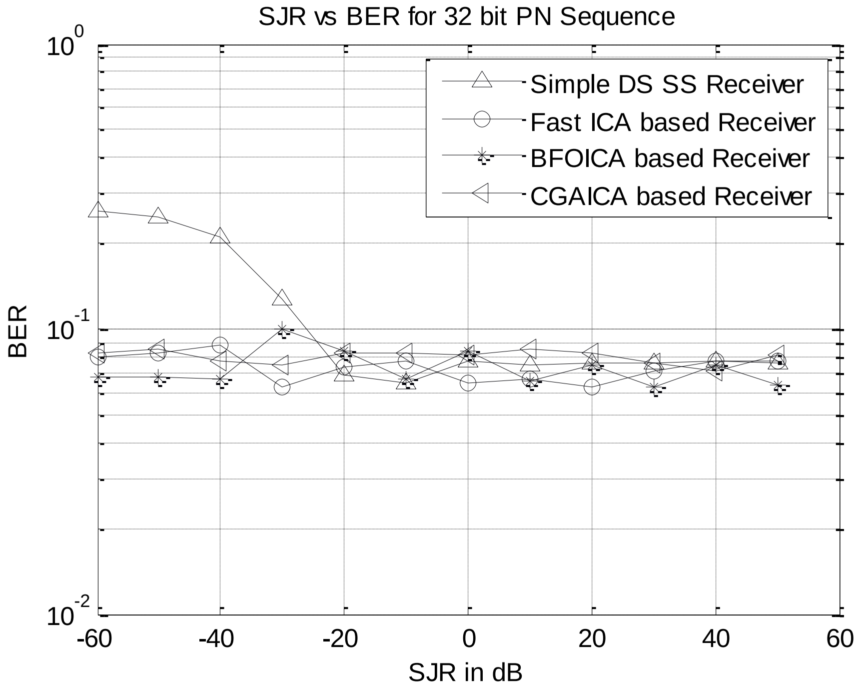

Fig 7.18 Plot of SJR vs BER without and with Fast ICA BFOICA & CGAICA for DS-SS System with 32 bits PN Sequence at SNR=0dB(Random Jammer)

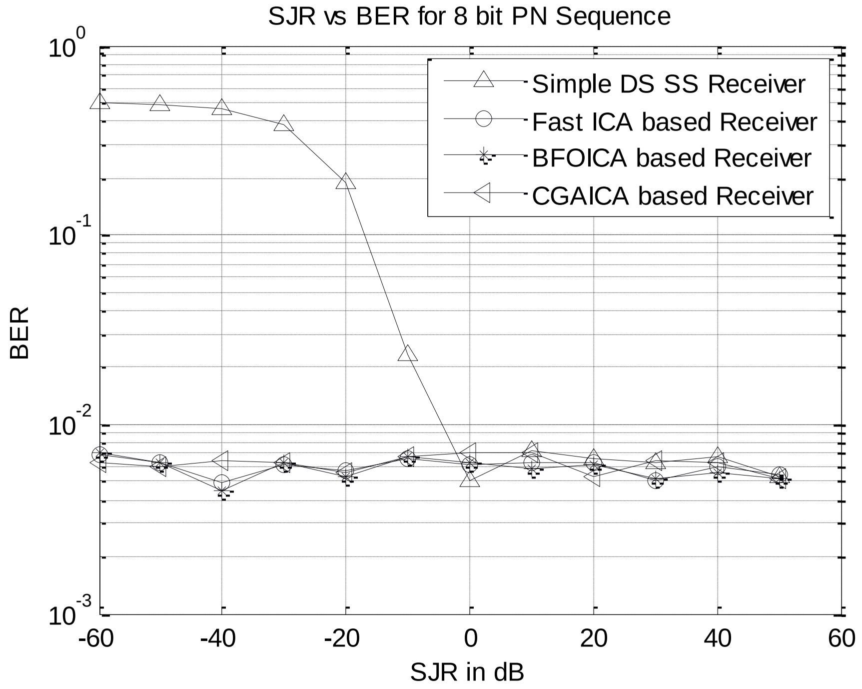

Fig 7.19 Plot of SJR vs BER without and with Fast ICA BFOICA & CGAICA for DS-SS System with 8 bits PN Sequence at SNR=5dB(Random Jammer)

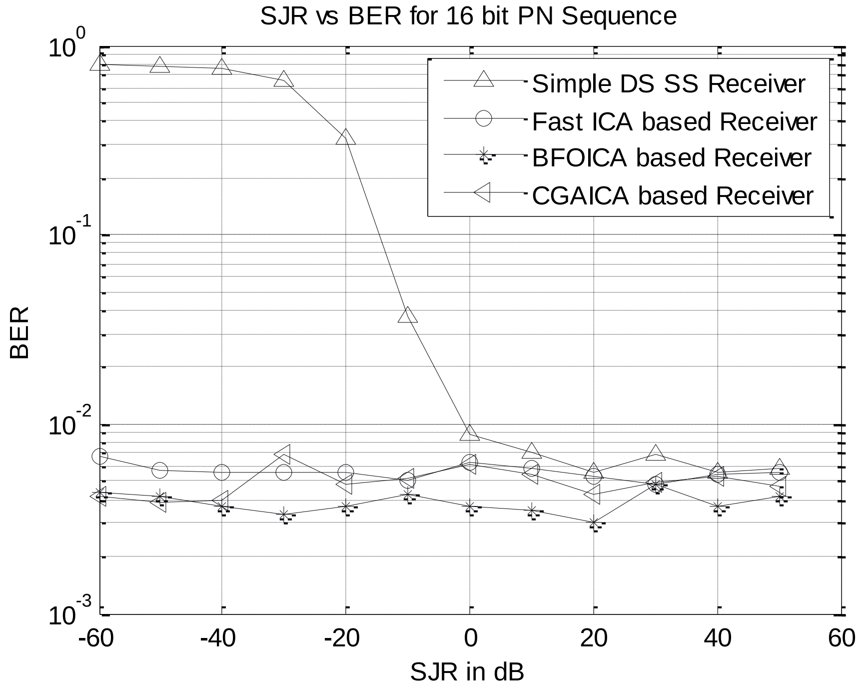

Fig 7.20 Plot of SJR vs BER without and with Fast ICA BFOICA & CGAICA for DS-SS System with 16 bits PN Sequence at SNR=5dB(Random Jammer)

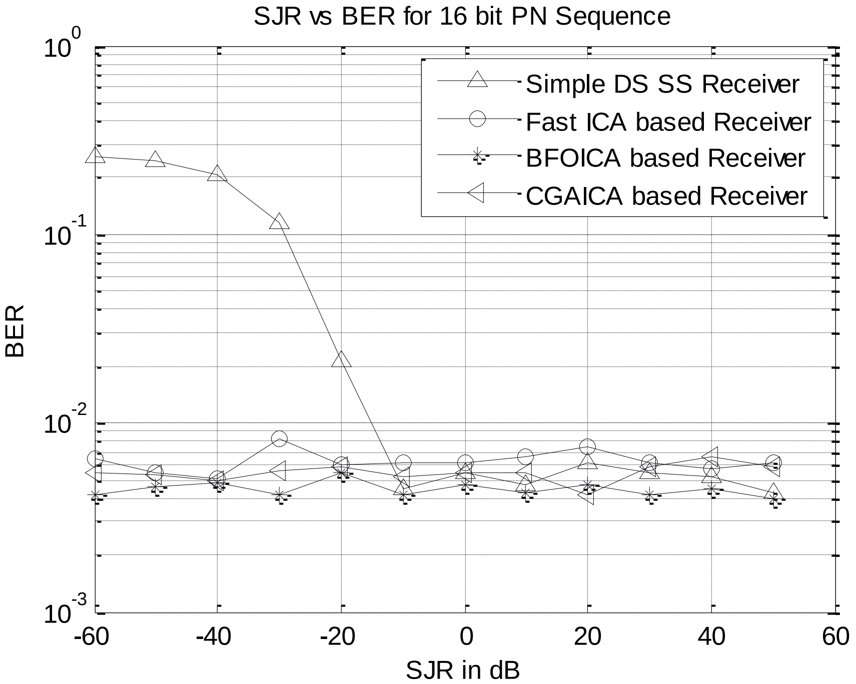

Fig 7.21 Plot of SJR vs BER without and with Fast ICA BFOICA & CGAICA for DS-SS System with 32 bits PN Sequence at SNR=5dB(Random Jammer)

However there are certain issues pertaining to the fast ICA (also other gradient based ICA) assisted DS-SS technique. Fast ICA is reported to have inherent ambiguities like permutation and scaling. Due to the permutation uncertainty the independent components recovered do not appear in a known or particular order. So at the output of the fast ICA block it is difficult to recognize the spread spectrum signal and jammer signal. To recognize this and get the desired spread spectrum signal for further processing by the receiver, a training sequence is employed and another post processing block after fast ICA separation block is used which performs selection of the desired spread spectrum signal. However this is a signal classification problem and makes the technique semi blind. Apart from this it adds to the computational complexity of the system.

Generally in typical DS-SS applications like CDMA, the pilot training sequence is available at the receiver end. To recognize the signal from the signal and jammer (appearing at the output of ICA block), the pilot training signal is correlated with those signals. But this correlation based methods often becomes difficult when the jammer signal is random in nature. This is because in such cases the signal correlation and jammer correlation with the training signal are comparable. In such case signal classification becomes more complex.

In the proposed CGAICA and BFOICA, however, there is no permutation ambiguity and we know the order in which the desired signal appears is known. In the present study the desired SS signal comes as the second independent component always. So there is no need to employ the signal selection block and use the training sequence. The proposed CGAICA and BFOICA assisted techniques are observed to be more advantageous in case of random jammer signal where correlation based identification techniques are difficult. So BFOICA assisted DS-SS system is a blind technique for jammer suppression. But the disadvantage of this technique is that it is computationally more complex than the fast ICA algorithm. The computational complexity of BFOICA depends on the parameters of bacterial foraging optimization process. However, by employing this new scheme the post processing complexity involved with fast ICA based separation can be avoided.

7.9. Conclusions

The present Chapter studies the jammer mitigation in Direct Sequence Spread Spectrum communications system by using fast ICA technique and proposes new schemes based on BFOICA and CGAICA. Both fast ICA and BFOICA assisted DS-SS models yield better BER performance even when the high jammer power is high. However selection of desired spread spectrum signal in case of fast ICA based scheme adds to the computational complexity. The proposed BFOICA assisted jammer suppression in DS-SS communication systems has almost comparable performance as the fast ICA based jammer suppression. The additional signal selection block needed in case of fast ICA and other gradient based ICA techniques for jammer mitigation in DS-SS systems, is avoided by the proposed scheme.

References

[7.1] LB.Milstein and R.A.Iltis, "Signal Processing for Interference Rejection in Spread Spectrum Communications", *IEEE ASSP Magazine*, pp.18-31, Apr. 1986.

[7.2] L.Milstein, "Interference rejection techniques in spread spectrum communications", *Proceedings of the IEEE*, vol.66, pp.657-671, June 1988.

[7.3] A.Belouchrani, M.Amin, "Jammer mitigation in spread spectrum communications using blind sources separation", *Signal Processing*, 80, pp. 723-729, 2000.

[7.4] K.Raju, et.al., "Jammer suppression in DS-CDMA arrays using Independent Component Analysis", *IEEE Transactions on Wireless Communications*, vol.5, No.1, pp.1-6, Jan.2006.

[7.5] L.M.San Jose-Revuelta, "Entropy-guided micro-genetic algorithms for multiuser detection in CDMA communications", *Signal Processing*, vol.85, Issue 8, pp. 1572-1587, Aug. 2005.

[7.6] C.Ergun, K.Hacioglu, "Multiuser detection using a GA in CDMA communication systems", *IEEE Trans. Comm.* 48 (8), pp.1374-1383, 2000.

[7.7] M.G.Shayesteh, M.B.Menhaj and B.G.Nobary, "A modified Genetic Algorithm for multiuser detection in DS/CDMA systems", *IEICE Trans. Com. E* 86-B (8), pp.2377-2388, 2003.

[7.8] A. J. Viterbi, "CDMA: Principles of Spread Spectrum Communication" Addison Wesley, 1995.

[7.9] LB.Milstein, "Interference Suppression to Aid Acquisition in Direct-Sequence Spread- Spectrum Communications", *IEEE Trans. On Communications*, vol. 36, No. 11, pp.1200-1207, Nov. 1988.

[7.10] J.D.Laster and J.H.Reed, "Interference Rejection in Digital Wireless Communications", *IEEE Signal Processing Magazine*, pp.37-62, May 1997.

[7.11] A. Belouchrani and M.Amin, "A two-sensor array blind beamformer for direct sequence spread spectrum communications", *IEEE Trans. Signal Processing*, Vol.47, pp. 2191-2199, Aug. 1999.

[7.12] L.Dominguez, J.Borrallo and J.Garcia, "A general approach to the automatic classification of radio communication signals", *Signal Processing* 22, pp. 239-250, March 1991.

Chapter

VIII

CONCLUSIONS AND SCOPE FOR FUTURE RESEARCH

8.1. General Conclusions

The research studies presented in this thesis, developed a novel constrained genetic algorithm based ICA algorithm and a new ICA algorithm based on bacterial foraging optimization and have evaluated their performance. The same BFO technique is applied to develop the post nonlinear ICA algorithm. A detailed study has been conducted for the fixed-point VLSI implementation of the ICA algorithms developed here and the most popular fast ICA algorithm. A trial FPGA implementation of fast ICA algorithm has also been presented. The ICA algorithms developed in the thesis have been applied to suppress the jammer interference blindly in direct sequence spread spectrum receiver to demonstrate their superiority.

The investigations conducted in this research work yield the following important conclusions.

1. The computationally intelligent techniques like GA and BFO are successfully applied to linear and nonlinear ICA. Such techniques always converge to a global optimum in the problem space of the contrast function used. However these optimization techniques are computationally complex as compared to the gradient based techniques. But with the availability of high end processors this issue seems to have very less importance.
2. The gradient based ICA methods have inherent permutation ambiguity. In CGAICA and BFOICA techniques the recovered independent components always appear in a fixed order depending on the relative values of the objective

function (e.g. kurtosis). Hence if the kurtosis values of the signals are relatively known then the permutation ambiguity is resolved. But these methods suffer from the limitation that if all these components have equal or very nearly equal value of the contrast function then the order of appearance of the independent components is difficult to predict.

3. BFO technique along with the gradient based linear ICA method is capable of recovering signal components from their nonlinear mixtures. In this case however the permutation ambiguity is still prevailing as BFO is used only for nonlinear parameter estimation but fast ICA (or any other gradient technique) recovers the components from their linear mixtures. The BFO based extraction of components from their nonlinear mixtures possesses faster convergence as compared to GA based nonlinear ICA.
4. VLSI (ASIC/FPGA) implementation of any ICA algorithm is carried out using fixed-point arithmetic for optimal use of resources. The register length greatly affects the separation performance of ICA algorithms. The fixed-point BFOICA performs better then the fixed-point CGAICA particularly at lower bit precisions. The fixed-point fast ICA exhibits superior performance than fixed-point CGAICA and BFOICA. A trial implementation of fast ICA algorithm shows that with speed optimization and without any register pipelining in FPGA the achievable speed is 6.61MHz. It is also observed that by trying to get optimized speed, the resource utilization increases.
5. BFOICA and CGAICA techniques are proposed for mitigating the jammer interference from the desired signal in a DS-SS receiver. BFOICA and CGAICA assisted jammer suppression technique avoids the extra signal identification overhead present in fast ICA assisted jammer mitigation. Proposed BFOICA and CGAICA based schemes for jammer suppression yield the same BER as fast ICA assisted model even at high jammer powers irrespective of signal to noise ratio in an AWGN channel. This is consistently observed irrespective of the nature of jammer.

8.2. Scope for Future Research

The proposed research can be extended in following dimensions.

1. Evolutionary ICA techniques can be extended for cases of convolutive signal/speech mixtures. Other variants of evolutionary computation technique and their hybrids can be used for optimization in ICA algorithm.
2. The BFO based nonlinear ICA can be studied in greater details for nonlinear convolutive mixtures of signals. In nonlinear ICA we have used BFO and fast ICA algorithm. Nonlinear ICA can be attempted to be solved only by BFO without using fast ICA (or any gradient based technique). The same experiment can be extended for any evolutionary technique may be for a better convergence. The identifiabililty of models for nonlinear ICA needs further research.
3. The complexity analysis for implementation of the computationally intelligent ICA techniques can be an extensive research activity. One can also carry out VLSI implementation of different ICA algorithms using the concepts of systolic arrays.
4. Interference suppression and multi user detection in DS-CDMA systems can also be carried out using various ICA techniques. ICA can also be applied to MIMO OFDM systems. Application of noisy nonlinear ICA to wireless sensor networks is another domain of research.
5. The generalized analysis of nonlinear ICA algorithm can be carried out for better understanding of the separation of signals.

CURRICULUM VITAE

Debiprasad Priyabrata Acharya

PERSONAL DETAILS

Name:	Debiprasad Priyabrata Acharya
DoB:	23rd August 1974
Sex:	Male
E-mail :	d_p_acharya@rediffmail.com
Permanent address:	AT-Rameswarpur, P.O.-Gadijang, Dist-Cuttack, Orissa. PIN-754157 Ph : +91-671-2852515 +91-9438484151

PROFESSIONAL PROFILE

- Current Organization: **National Institute of Technology, Rourkela**
- Organization: **Tata Elxsi Ltd, Bangalore.**
  - Designation : Specialist (ASIC/FPGA)
  - Job profile: FPGA Design and implementation for CDMA EvDO 1xRTT Femto Cell.
- Organization: **Wipro Technologies, Bangalore.**
  - Designation : Technical Consultant
  - Job profile: FPGA Design and implementation for Telecom products.
- Organization: **Center for Development of Telematics (C-DOT), Telecom Technology Centre of Govt. of India, Bangalore.**
  - Designation : Research Engineer
  - Job profile: Digital System solutions using FPGA and related board.

ACADEMIC QUALIFICATIONS

- Master of Engineering in Microelectronics from BITS Pilani in Jan 2002 with 8.27(CGPA).
- Master of Science in Physics with Electronics from IIT Kharagpur in June 1996 with 7.19(CGPA).
- 10+2 Science from B.J.B.College, Bhubaneswar in May 1991 with 88.8% of marks
- H.S.C. from Board of Secondary Education, Orissa in June 1989 with 89.9% of marks

RECOGNITIONS

- Awards/Scholarships:
  - National Scholarship throughout carrier
  - P.K. Memorial award for best graduate
  - GATE
- Professional Body Membership:
  - Member IEEE (USA)
- Number of Publications: 10