2004

General Interference Suppression Technique For Diversity Wireless Rece

Tianyu Yang
University of Central Florida

GENERAL INTERFERENCE SUPPRESSION TECHNIQUE FOR DIVERSITY WIRELESS RECEIVERS IN FADING CHANNELS BASED ON INDEPENDENT COMPONENT ANALYSIS

by

TIANYU YANG
B.S. EE, Zhejiang University, P.R. China, 2001

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the Department of Electrical and Computer Engineering
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Fall Term
2004

Major Professor: Dr. Wasfy B. Mikhael

ABSTRACT

The area of wireless transceiver design is becoming increasingly important due to the rapid growth of wireless communications market as well as diversified design specifications. Research efforts in this area concentrates on schemes that are capable of increasing the system capacity, providing reconfigurability/reprogrammability and reducing the hardware complexity. Emerging topics related to these goals include *Software Defined Radio*, *Multiple-Input-Multiple-Output (MIMO) Systems, Code Division Multiple Access, Ultra-Wideband Systems,* etc.

This research adopts space diversity and statistical signal processing for digital interference suppression in wireless receivers. The technique simplifies the analog front-end by eliminating the anti-aliasing filters and relaxing the requirements for IF bandpass filters and A/D converters. Like MIMO systems, multiple antenna elements are used for increased frequency reuse. The suppression of both image signal and Co-Channel Interference (CCI) are performed in DSP simultaneously.

The signal-processing algorithm used is Independent Component Analysis (ICA). Specifically, the fixed-point Fast-ICA is adopted in the case of static or slow time varying channel conditions. In highly dynamic environment that is typically encountered in cellular mobile communications, a novel ICA algorithm, OBAI-ICA, is developed, which outperforms Fast-ICA for both linear and abrupt time variations.

Several practical implementation issues are also considered, such as the effect of finite arithmetic and the possibility of reducing the number of antennas.

This work is dedicated to Jesus Christ, my Lord and Savior, for His enduring love and perfect plan for my life.

"In his heart a man plans his course, but the LORD determines his steps."

Proverbs 16:9 (NIV)

ACKNOWLEDGMENTS

I wish to dedicate my wholehearted praise and thanksgiving to Jesus Christ for giving me not only the opportunity to know Him in America, but also a prepared heart to accept Him as my Lord and personal savior when I was introduced to Him.

I am deeply indebted to my advisor Dr. Wasfy B. Mikhael for his seasoned guidance and help throughout my Ph.D. study, both academically and personally.

I appreciate the help from members of my dissertation committee.

I am grateful to my parents for their encouragement and support during my study in America.

Last but not least, I thank my brothers and sisters at Bread of Life Christian Church in Orlando, because it is through them that I began to realize and understand God's unfailing love and amazing grace.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER ONE: INTRODUCTION

The wireless communications market has been growing rapidly world wide since 1990's. The growth is even stronger since we enter the 21st century, especially in Asian countries such as China and India.

Being the largest consumer product industry in history, the wireless/mobile industry attracted huge amount of research efforts that significantly reduced the system cost, which in turn fueled further demand [1]. As a result, the technology had to evolve constantly to keep up with such demand.

Many concepts have been proposed to enable higher transmission rates, better bandwidth usage efficiency, and service provision diversity. Also, reducing the size and cost of the handheld devices and improve the Quality of Service (QoS) are major challenges.

Software Defined Radio

The term software radio implies radio functionalities defined by software. This requests the use of DSPs to replace dedicated hardware to execute the necessary software (SW) [2].

Software radio has many desirable features. In software radio, SW controls and programs transceiver architecture. Therefore, flexibility is provided to achieve the capability of reconfiguration. Also, software radio supports multiple modes and multiple standards.

At both the receiver and transmitter side, most functions should be defined by SW. These include the selection and identification of frequency band and bandwidth, modulation and coding schemes, as well as definition and implementation of user application.

SW radio is compatible with any other radio mobile. Also, DSP implementations are in real time.

In order to achieve SW radio, two primary tasks have to be accomplished. First, the ADC (at the receiver side) and DAC (at the transmitter side) have to be moved near the antenna, thus signal processing tasks can be performed in digital domain. Second, dedicated hardware needs to be replaced by DSPs.

MIMO Systems

The third generation (3G) systems are being deployed throughout the world. Currently, 3G has two principal standards, namely, Wideband Code Division Multiple Access and 1XRTT [3]. Yet, the increase in the demand for higher data rate and better QoS, together with the scarcity of the Radio Frequency (RF) spectrum, make it necessary to adopt novel techniques.

The use of multiple antennas, known as space diversity, at the transceiver is proved to be a feasible technique for better performance. The technique is named Multiple-Input-Multiple-Output (MIMO). In particular, space diversity at the receiver side effectively exploits multipath fading, originally a drawback, for the separability of fading channels [4]. Also, many applications of adaptive signal processing at the receiver side for interference reduction have been introduced, which prove to be promising.

There are several different ways to exploiting MIMO structure [3].

The first one is *Array Gain*. By combining the signals received via multiple antennas, the Signal to Noise Ratio (SNR) can be substantially improved at the receiver side.

The second one is *Spatial Multiplexing* (SM). Without increasing the power consumption, using multiple antennas provides a linear increase in the system capacity for a fixed amount of bandwidth. Only MIMO channels make SM possible.

Finally, *Co-Channel Interference* (CCI) suppression can be achieved by exploiting independence between the spatial signatures of the desired signal and CCI's. This is extremely important for frequency reuse systems.

Spread Spectrum Technology

Spread spectrum modulation uses a transmission bandwidth many times greater than the information bandwidth or data rate of any user. Spread spectrum applications fall into several broad categories [5]:

- High capability to tolerate interference (both intentional and unintentional);
- More accurate position location and velocity estimation;
- Much lower detectability by an unintended receiver;
- Multiple access communication: a large number of typically uncoordinated users can share a common spectral allocation.

CDMA is such a technique that transmits signals that appear noiselike and random. This is achieved by coding the input symbols by pseudorandom sequence. The same pseudorandom sequence is used at the receiver to decode the transmitted symbols.

Adopting Direct-Sequence (DS) or Frequency Hopping (FH) technique, CDMA accesses the local exchange carrier through wireless terminals. The frequency reuse problem is avoided because the users are separated by orthogonal codes. Each CDMA cell occupies the same 1.25 MHz band simultaneously. In this respect, frequency planning is substantially simplified. Also, the capacity of CDMA is significantly increased.

In CDMA scheme, accurate power control is essential in limiting interference. Every user is in effect a noise to all the other users sharing the same channel. Also, the noise from every user is accumulated. This represents a soft limit to the number of users a system can accommodate. Therefore, the power control scheme should be able to maintain each mobile's power consumption at the smallest possible level and ensure QoS at the same time.

Another important scheme in this category is *Ultra-Wide Band* (UWB). The UWB signals span over an extremely wide frequency range. Therefore, they are very hard to detect. Typically, a UWB signal occupies 1/4 of the center frequency or more.

A very popular UWB scheme is to use non-sinusoidal pulses with widths smaller than one nanosecond.

Although UWB promises low power, low cost, high data rates and precise positioning capability, it is widely believed that it will not be fully implemented unless an additional interference suppression mechanism is introduced.

Motivation and Scope of the Research

The suppression of interfering signals is becoming critically important for modern communication systems.

If the network loading is very low, incorporating interference measurements in resource management helps to provide interference avoidance. However, if the network loading is high, avoidance technique is no longer effective, so it is incapable of preventing the degradation of Quality of Service (QoS). Therefore, it is necessary to reduce interference after it has already occurred.

In this research, the suppression of two important types of interfering signals is discussed, namely, the image signal and the Co-Channel Interference (CCI).

The image problem arises from out-of-band users due to the adoption of IF stage [6-7]. Traditionally, image rejection is performed by BandPass Filters (BPFs) at the RF stage. The stringent requirement for these filters adds to the front-end's complexity and makes the receiver costly. As receivers are required to work at newly allocated high RF band, this problem is becoming more serious.

On the other hand, CCI occurs when different users are occupying the same frequency band. The suppression of CCI is a critical issue, for it determines the system capacity. It is not possible to attenuate CCI by analog filters.

Interference suppression in the digital domain is highly desirable, because it can reduce the complexity of the front-end, which makes the receiver easier to be integrated and less costly. Moreover, it is an important step towards the long-term objective of transceiver design: *Software-Defined Radio*. As the computational power of Digital

Signal Processor (DSP) is increasing rapidly, implementing radio functionalities digitally is becoming more and more feasible.

Diversity reception with *Independent Component Analysis* (ICA) stands out as a strong candidate for digital interference suppression [8-9].

Adopted in MIMO systems, the use of multiple antennas effectively exploits multipath fading. In ICA, statistical independence between signal components is exploited for separation of the interfering signals from the desired signal.

ICA's ability of suppressing strong interferers helps to solve the *near-far* problem, because ICA has no requirement for the components' relative strength. This is especially beneficial for high data rates applications [10]. As long as the resulting mixing matrix is not singular, the performance is unaffected if the power of the interference sources is changed. Moreover, ICA is robust against erroneous estimation of system parameters, since it is based purely on high-order statistics.

In this research, we propose using this technique for interference suppression in wireless receivers operating in fading conditions [11-14]. It provides simultaneous rejection for image signal and Co-Channel Interferer (CCI). The order ambiguity inherent to ICA is addressed, and an efficient solution is presented [15].

Further, to make the proposed technique feasible in highly dynamic environment, a novel ICA algorithm is proposed based on block adaptive signal processing and Taylor series expansion [16]. Simulation results confirm that the presented algorithm has much superior performance to the Fast-ICA algorithm, which has been dominant in various applications of ICA. It is well known that Fast-ICA cannot operate well in rapid time-varying conditions. Also, it is worth mentioning that the new algorithm represents a new family of ICA algorithms that are also applicable in other areas.

Besides, brief study has been carried out regarding practical implementation issues. First, the effect of finite arithmetic is studied by computer simulations [17]. Second, we prove that in the case of *Binary Phase Shift Keying* (BPSK) receivers with one interferer present, the number of antenna can be reduced to one without any performance degradation by a simple operation in the digital domain [18]. Therefore, the hardware complexity can be reduced by half with negligible increase in computational complexity. Finally, it is shown that the proposed technique should be disabled when the received signal already has a large signal to interference ratio, i.e., the desired signal is much stronger than the interference. In such a scenario, additional suppression is not needed.

Organization of the Document

The document is organized as follows.

Chapter 2 gives a brief overview of the statistical technique used in this research, namely, ICA.

Chapter 3 presents the novel ICA algorithm that has superior performance for highly dynamic environment.

Chapter 4 introduced the proposed interference suppression technique for diversity wireless receivers, where the Fast-ICA algorithm is applied assuming the channel is quasi-stationary.

Chapter 5 applies the new ICA algorithm to the same scheme assuming rapid time-varying channel conditions. It is also shown that Fast-ICA cannot work properly in this scenario.

Chapter 6 discusses practical implementation issues of the proposed technique.

Finally, Chapter 7 summarizes the research work and suggests possible direction for future research.

CHAPTER TWO: INDEPENDENT COMPONENT ANALYSIS

Independent Component Analysis (**ICA**) is a statistical signal processing technique that utilizes the statistical independence of signals hidden in sets of measurements.

In a typical ICA model, the variables, also known as the observations, are linear or nonlinear combinations of certain source signals of interest. These signals are statistically independent and nongaussian. It is the ICA algorithms' goal to recover the source signals. Although the criterion adopted in the ICA processing varies, almost all ICA algorithms share the same characteristic in that they take advantage of higher-order statistics.

While most classic statistical processing techniques utilize only second order statistics, ICA is a more powerful method because it relies on more stringent assumption. Fortunately, in practice most source signals underlying the observations do satisfy the independence (not only uncorrelated) assumption.

ICA is capable of processing data in many areas, such as *Feature Extraction* [19], *Telecommunications* [20-21], *Financial Engineering* [22], *Brain Imaging* [23] and *Text Document Analysis* [24].

General Background

In the area of adaptive filtering, *supervised learning* refers to the design of filters assuming that a training sequence is available with specification of the desired response of the filter. The other class, known as *unsupervised learning,* does not need the specification of such response. Instead, unsupervised learning algorithms adjust the free parameters of adaptive filters according to certain desired properties of input-output mapping.

ICA is such an unsupervised learning technique that specifies the mutual independence of the output signals. Of course, the validity of ICA processing lies in the assumption that the observations are indeed generated by independent sources that have physical meanings [25].

The problem ICA processing attempts to solve is also known as Blind Source Separation (**BSS**). Some examples of BSS problems include different speech signals detected by a set of microphones at different locations, multiple signals received by a mobile handset, or several time series generated during certain financial process.

Basic ICA Model

The linear ICA model is: **X**=**AS**. Here, **X** is the given observation matrix, and each row of **X** is one observation; **A** is a square mixing matrix whose dimensionality is the same as the number of rows in **X**; **S** is the source signal matrix, and each row of **S** is one independent component. Typically, the number of observations is the same as the number of independent components.

The goal of ICA procedures is to obtain **S** from **X** when **A** is unknown. As mentioned in the previous sections, ICA algorithms are based on high order statistics, so they need the assumption of statistical independence between the source signals. Also, at most, one source signal can be gaussian.

The objective of ICA is to find a separation matrix **W**, such that **S** can be recovered when the observation matrix **X** is multiplied by **W.** This is achieved by making each component in **WX** as independent as possible. Each row in **W** is a demixing vector that is able to extract one component when applied to **X**.

Principles of ICA

Maximization of Nongaussianity

The simplest principle of ICA is *Maximization of Nongaussianity*, which is adopted in this research.

Nongaussianity is fundamental in ICA estimation. The *central limit theorem* states that, if the source signals are nongaussian, their sum is more gaussian than any one of them individually. Thus, a linear combination of the observations will be maximally nongaussian if it equals one of the independent components up to a multiplicative constant.

Maximum Likelihood Estimation

Maximum Likelihood (ML) Estimation is the fundamental method of statistical estimation. In ML estimation, values that give the highest probability for the observations are taken as the estimation of the parameters.

The key to ML estimation is the probability densities of the independent components, since the likelihood is a function of these densities. If this information is available, a very simple gradient algorithm can be derived. Otherwise, the problem is much more complicated, because the estimation of densities is a *nonparametric* problem, which means the problem cannot be reduced to the estimation of a finite number of parameters. A common way to solve this problem is to approximate the densities of the independent components by a family of densities that are specified by a limited number of parameters.

The well-known *Bell-Sejnowski* algorithm and *Natural gradient* algorithm belong to this category.

Minimization of Mutual Information

Minimization of Mutual Information is inspired by information theory.

The motivation of this approach is that it may not be very realistic in many cases to assume that the data follows the basic ICA model. Therefore, a general-purpose measure of dependence is more useful. The classic information-theoretic measure of statistical independence, *Mutual information,* is desirable in these scenarios.

However, minimization of mutual information actually leads to similar algorithms as maximization of nongaussianity. Thus, the main utility of mutual information is theoretical, and it provides a unifying framework for other estimation principles, especially ML estimation and maximization of nongaussianity.

Tensorial Methods

Tensorial Methods use higher-order cumulant tensor. Tensors are generalizations of matrices, or linear operators. Cumulant tensors are generalizations of covariance matrix. The covariance matrix is the second-order cumulant tensor.

The rationale behind tensorial methods is, if we make the fourth-order cumulants zero, a higher-order decorrelation is achieved. This gives one class of methods for ICA estimation.

The tensorial methods typically involve the eigenvalue decomposition operation, which is computationally inefficient if the dimensions of the data are high. Therefore, they are becoming less popular.

Measure of Independence

Since ICA relies on statistical independence to perform signal separation, a quantitative measure of independence is needed. The most important measure of independence is nongaussianity.

In general, the *negentropy* is the optimal measure of nongaussianity. For a random variable *y* with probability density function *p(y)*, its entropy is defined as:

$$H(y) = -\int_{-\infty}^{\infty} p(y)\log_2 p(y)dy \qquad (2.1)$$

The entropy represents the level of uncertainty within the random variable. Among all the random variables with equal variance, the gaussian distribution has the largest entropy.

In practice, it is desirable to have a measure that is zero for gaussian distribution and nonnegative for all others. Thus, the negentropy is defined:

$$G(y) = H(y_{gaussian}) - H(y) \qquad (2.2)$$

where $y_{gaussian}$ is a gaussian variable with equal variance as *y*.

To use negentropy, the density function has to be known. Otherwise, estimation has to be used. This task makes the negentropy based method computationally highly demanding.

The simplest quantitative measure of nongaussianity is *kurtosis*, whose absolute value measures the departure of a random vector from a gaussian random vector. The kurtosis of a zero-mean random variable ξ is defined as a fourth order cumulant:

$$\mathrm{kurt}(\xi) = E\{\xi^4\} - 3(E\{\xi^2\})^2 \tag{2.3}$$

A distribution with a negative kurtosis is called subgaussian, while a distribution with a positive kurtosis is called supergaussian. Subgaussian probability density functions (pdf) are flatter than the gaussian pdf. On the other hand, a supergaussian pdf has a sharper peak and/or longer tails than the gaussian pdf.

The main reason for kurtosis being widely used a measure of non-gaussianity is its computational simplicity, because it can be easily computed from the moments of the sampled data. However, due to the same reason, kurtosis is sensitive to outliers.

Fast-ICA Algorithm for Extraction of One Component

A natural choice for maximizing the absolute value of kurtosis is the gradient algorithm. However, the convergence of such algorithm is slow, and depends on a proper choice of the learning rate sequence. The fast fixed-point algorithm is a promising alternative that makes learning considerably faster and more reliable.

Fast-ICA is a very simple and efficient method that has been shown to have excellent convergence properties [26-27]. It is dominant in various application areas.

A straightforward way to understand this Newton based method is based on the observation that, at the convergence point, the demixing vector **w** should point to the same direction as the gradient vector. Therefore, if the measure of non-gaussianity is kurtosis, **w** is updated at each iteration to be the gradient of the absolute value of kurtosis at the previous iteration.

Since the expectation operator in the gradient vector has to be estimated, the algorithm should be operated on a block of data, usually referred to as a “frame”, so the sample means can be used as the estimation. Thus, the algorithm operates in batch mode.

Below is a description of Fast-ICA based on kurtosis. As a convention, the variance of the data is assumed to be unity, so the definition of kurtosis (2.3) can be simplified. In the description, **X** is a N by N_f data matrix, where N is the number of observations, which is also the number of independent components; and N_f is length of the processing frame.

Step 1. Get the whitened data $\mathbf{X_1}$ by decomposing **X**’s covariance matrix. Set the counter $p = 1$;

Step 2. Initialize the p^{th} row of the separation matrix $\mathbf{w_p}$ to a random vector of unity length;

Step 3. Set $\mathbf{w_p} = \frac{1}{N_f}\sum_{n=1}^{N_f}\{[\mathrm{w_p X_1(n)}]^3 \mathrm{X_1(n)}\} - 3\mathrm{w_p}$;

Step 4. Make the estimate of $\mathbf{w_p}$ orthogonal to the subspace spanned by the already extracted rows of **W** using

$$\mathbf{w_p} = \mathbf{w_p} - \sum_{v=1}^{p-1} (\mathrm{w}_p^{\mathrm{T}} \mathrm{w}_v)\mathrm{w}_v \ ;$$

Step 5. Normalize $\mathbf{w_p}$ to unity length;

Step 6. Check the convergence of $\mathbf{w_p}$. If it is not reached, go back to *Step 3*, otherwise proceed;

Step 7. Set $p = p + 1$. If $p \leq N$, go back to step 2.

In *step 6*, the convergence means that the old and new values of $\mathbf{w_p}$ point in almost the same direction, i.e., the difference between their dot-product and 1 is less than a predefined threshold value. In our simulations, it is found that a threshold value of 10^{-6} is appropriate to achieve satisfying performance while maintaining fast convergence.

In *Step 1*, the whitening is performed by Eigenvalue decomposition (EVD). For noisy ICA model, i.e., if there exists significant Gaussian noise, the whitening should be replaced by *quasiwhitening*, where the covariance matrix of the noise-free data should be used instead of the covariance matrix of the noisy data.

The above parallel algorithm is computationally simple. It eliminates the choice of a learning rate sequence, and the estimate of the probability density function is not needed. The convergence is globally cubic. Moreover, it is proved that the algorithm is locally consistent, which means we don't need to distinguish between maximization and minimization of the kurtosis based on whether the signals are supergaussian or subgaussian.

The frame length N_f is an important parameter to be chosen. Ideally, all the available data to be processed needs to be used to have a good estimate of the expectation operation. However, the computational complexity in terms of multiplications and divisions per iteration is of O(N_f), so the computations may become too demanding if the sample size is very large. Thus, there is a tradeoff between the performance and the computational complexity.

Complex-valued Fast-ICA

Separation of complex valued signals is a frequently arising problem in many situations, such as frequency-domain applications and communication systems. For example, in separation of convolutive mixtures, the data is usually transformed into Fourier domain before being processed.

The distributions for the complex variables are often spherically symmetric. Thus, only the modulus is of interest.

The kurtosis for a complex random variable can be defined as:

$$\mathrm{kurt}(y) = E\{|y|^4\} - 2(E\{|y|^2\})^2 - |E\{y^2\}|^2 = E\{|y|^4\} - 2 \qquad (2.4)$$

where y is white, i.e., the real and imaginary parts of y are uncorrelated and their variances are equal. Using this definition, a fixed-point complex-valued Fast-ICA algorithm based on deflationary separation can be developed.

Let **X** and N_f denote the observation matrix and the processing frame length, respectively, and N is the number of components. Also, the source signals are assumed to have zero mean and uncorrelated real and imaginary parts of equal variance. The Fast-

ICA algorithm for complex-valued signals [28] is described below, where the contrast function is chosen according to kurtosis.

Step 1. Get the whitened data $\mathbf{X_1}$ by decomposing $\mathbf{X}$'s covariance matrix. Set the counter p = 1;

Step 2. Initialize $\mathbf{w_p}$ to a random vector of unity length;

Step 3. Set $\mathbf{w_p} = \frac{2}{N_f}\sum_{n=1}^{N_f}\{\mathrm{X}_1(\mathrm{n})[\mathrm{w}_\mathrm{p}^\mathrm{H}\mathrm{X}_1(\mathrm{n})]^* - 2\mathrm{w}_\mathrm{p}\}\,|\,\mathrm{w}_p^\mathrm{H}\mathrm{X}_1(\mathrm{n})\,|^2$;

Step 4. Make the estimate of $\mathbf{w_p}$ orthogonal to the subspace spanned by the already extracted rows of $\mathbf{W}$ using $\mathbf{w_p} = \mathbf{w_p}$ **-** $\sum_{v=1}^{p-1}(\mathrm{w}_p^\mathrm{H}\mathrm{w}_v)\mathrm{w}_v$ **;**

Step 5. Normalize $\mathbf{w_p}$ to unity length;

Step 6. Check the convergence of $\mathbf{w_p}$. If the convergence is not reached, go back to *Step 3*, otherwise proceed;

Step 7. Set $p = p + 1$. If $p \leq$ N, go back to *step 2*.

It is easily seen that the algorithm is a straightforward extension of real-valued Fast-ICA algorithm, thus it exhibits the same advantages. In the complex case, the transpose operation should be replaced by the Hermitian operation.

Estimating Multiple Components

To extract *n* independent components, the previous mentioned one-unit ICA algorithm needs to be run *n* times. In addition, the orthogonalization operation is also required to ensure uncorrelatedness of all the extracted components.

A simple way is deflationary orthogonalization using the Gram-Schmidt method, in which the components are estimated one by one. The basic idea is that after one demixing vector $\mathbf{w_p}$ is found, it should be projected on the space that is orthogonal to the subspace spanned by columns of **W** previously found. This procedure is already incorporated into the above outlined Fast-ICA algorithms.

The deflationary orthogonalization has the drawback of error accumulation. The error made in the estimation of first several components is accumulated and it contributes to the error in the estimation of later extracted components during the orthogonalization operation. As a solution, a symmetric orthogonalization can be used.

In symmetric orthogonalization, all extracted components of the demixing matrix are treated as equal. The orthogonalization is performed as a linear transformation:

$$\overline{W} = S^{-1/2}W \tag{2.5}$$

where W is the demixing matrix whose rows need to be orthogonalized and $S^{-1/2}$ is the transformation matrix from the Singular Value Decomposition (SVD) of WW^T.

Ambiguities in ICA

When applying an ICA algorithm, three ambiguities should be resolved.

The first is amplitude ambiguity, which means the amplitudes of the separated signals are obtained within arbitrary multiplicative constants. In practice, it is assumed that the source signals have unit variance during ICA processing, so the amplitude of the extracted signals has to be scaled by a gain corresponding to the variance of the signals.

The second is the sign ambiguity, which is a special case of the amplitude ambiguity for a scalar constant of –1.

The third is the order ambiguity, which means the order of the extracted components cannot be identified by ICA processing. Although in deflationary separation, the components tend to separate in the order of decreasing non-Gaussianity, this is not guaranteed.

The second and third ambiguities are typically resolved via certain prior information already available, such as a training sequence.

Drawbacks of Fast-ICA Algorithm

Unlike gradient-based online learning algorithms, Fast-ICA cannot adapt to fast variations in the mixing matrix in a nonstationary environment. It needs the assumption that within one processing frame, the mixing matrix should stay approximately constant.

Therefore, in applications that require real-time adaptation, Fast-ICA exhibits performance degradation if the mixing matrix changes significantly over one processing frame.

In this scenario, the ICA algorithm faces two opposing requirements regarding the choice of the frame size N_f. On one hand, to establish sufficient statistical independence, a large number of symbols should be used to obtain an accurate estimate of the expectation operation. This implies that N_f needs to be large. On the other hand, increasing N_f violates the quasi-stationarity requirement for **A** if **A** is fast time-varying.

Therefore, new algorithms that combine the advantages of both online learning and fixed-point ICA are needed. The algorithm should converge fast and able to track fast time variations. In the next chapter, such an algorithm will be developed.

Conclusions

In this chapter, Independent Component Analysis technique is briefly introduced. Different principles of ICA are overviewed, and the most important one, i.e., *Maximization of Nongaussianity*, is presented.

The basic ICA operations include prewhitening, extracting one component, orthogonalization and techniques solving the inherent amplitude and order ambiguity.

The most popular ICA algorithm, Fast-ICA, is described. Both real-valued and complex-valued versions of Fast-ICA are included. Also, the drawbacks of Fast-ICA are discussed.

CHAPTER THREE: ICA WITH OPTIMUM BLOCK ADAPTATION WITH INDIVIDUAL ADAPTATION PARAMETERS (OBAI)

As described in the previous chapter, the Newton-based Fast fixed-point ICA algorithm has been dominant in most applications. This is because it converges very fast, and in most cases the choice of learning rate is avoided.

However, in the case that the mixing matrix is highly dynamic, Fast-ICA cannot successfully track the time variation. Thus, a gradient-based algorithm is more desirable in this scenario.

The disadvantages of the online gradient-based algorithm are slow convergence and difficulty in the choice of learning rate. An improper choice of learning rate can destroy convergence.

In this chapter, a new adaptive algorithm named ICA with Optimum Block Adaptation with Individual adaptation parameters (OBAI) [29] is developed, which combines the advantages of the gradient-based online learning and Fast-ICA. From OBAI-ICA, a family of gradient ICA algorithms are derived: Individual Adaptation (IA) ICA, Homogeneous Adaptation (HA) ICA [30], Optimum Block Adaptation (OBA) ICA [31], and Block-ICA. Moreover, it is also shown that the online gradient ICA algorithms can be obtained from OBAI-ICA.

Background and Motivation

The Fast-ICA algorithm is a block algorithm. It uses a block of data to establish statistical properties. Specifically, the “expectation” operator is estimated by the average over L data points, where L is the frame size. The performance is better when the estimation is more accurate, i.e., L is larger.

However, it is very important that the mixing matrix stays approximately constant within one processing block, i.e., quasi-stationary. Thus, the problem arises when the mixing matrix is rapidly time varying, in which case a large L violates the assumption of quasi-stationarity.

On the other hand, the online gradient-based algorithm, which updates the separation matrix once for every received symbol, can better track the time variation of the mixing matrix. But it directly drops the "expectation" operator, which results in unstable convergence and worse performance than a block algorithm.

Therefore, a block algorithm is needed that can better accommodate time variations within one processing frame.

Our idea is to tailor the learning rates in a gradient-based block algorithm to each iteration and every coefficient in the separation matrix, in order to optimize a cost function that corresponds to a measure of independence. In [29], Mikhael and Wu used a similar idea to develop a fast block-LMS adaptive algorithm for FIR filters, which yielded useful results, especially when adapting to time-varying systems.

The algorithms developed in subsequent sections are used for estimating one row of the demixing matrix $\mathbf{W}$. Other ICA related operations, such as whitening, orthogonalization and solving ambiguities, are identical as the procedures mentioned in the previous chapter.

Formulation of OBAI-ICA

First, the following parameters are defined:

j: iteration index.

M: number of observations.

L: length of the processing frame.

$\underline{w(j)} = [w_1(j)\ w_2(j)\ \ldots\ldots\ w_M(j)]^T$: the current row of the separation matrix for the jth iteration. (i = 1, 2, …, M)

$x_{l,i}(j)$: the ith signal in the lth observation data vector for the jth iteration. (l = 1, 2, …, L)

$\underline{X_l(j)} = [x_{l,1}(j)\ x_{l,2}(j)\ \ldots\ldots\ x_{l,M}(j)]^T$: lth signal observation for the jth iteration.

$[G]_j = \left[\underline{X_1(j)} \quad \underline{X_2(j)} \quad \ldots.. \quad \underline{X_L(j)}\right]^T$: Observation matrix for the jth iteration

The lth kurtosis value for the jth iteration.

$$kurt_l(j) = \mathrm{E}\{[\underline{w^T(j)}\ \underline{X_l(j)}]^4\} - 3 \qquad (3.1)$$

where it is assumed that the signals and $\underline{w(j)}$ both have been normalized to unit variance.

Then, the kurtosis vector for the jth iteration is:

$$\underline{kurt(j)} = [kurt_1(j)\ kurt_2(j) \ldots\ldots\ kurt_L(j)]^T \qquad (3.2)$$

Now we can write the updating formula in a matrix-vector form as:

$$\underline{w(j+1)} = \underline{w(j)} + [MU]_j \nabla_B(j) \qquad (3.3)$$

where

$$\nabla_B(j) = \frac{\partial\{\underline{kurt^T(j)kurt(j)}\}}{\partial \underline{w(j)}} = \frac{1}{L}\left[\frac{\partial\{\underline{kurt^T(j)kurt(j)}\}}{\partial w_1(j)} \ldots \frac{\partial\{\underline{kurt^T(j)kurt(j)}\}}{\partial w_M(j)}\right]^T \qquad (3.4)$$

and

$$[MU]_j = \begin{bmatrix} \mu_{B1}(j) & ..... & 0 \\ ..... & ..... & ..... \\ 0 & ..... & \mu_{BM}(j) \end{bmatrix} \tag{3.5}$$

Note that in (3.3), a "+" sign is used instead of "-" as in the steepest descent algorithm. Because our cost function is kurtosis rather than error signal, we wish to maximize the cost function to achieve maximal nongaussianity.

To evaluate (3.4), we have:

$$\frac{\partial\{\underline{kurt^T(j)}\underline{kurt(j)}\}}{\partial w_i(j)} = \sum_{l=1}^{L} \frac{\partial[E\{[\underline{w^T(j)}\underline{X_l(j)}]^4\}-3]^2}{\partial w_i(j)} = 8\sum_{l=1}^{L}[\underline{w^T(j)}\underline{X_l(j)}]^3 kurt_l(j)x_{l,i}(j) \tag{3.6}$$

In derivation of (3.6), the expectation operator was dropped.

So the block gradient vector can be written as:

$$\nabla_B(j) = \frac{8}{L}[\sum_{l=1}^{L}[\underline{w^T(j)}\underline{X_l(j)}]^3 kurt_l(j)x_{l,1}(j) \ \ ... \ \ \sum_{l=1}^{L}[\underline{w^T(j)}\underline{X_l(j)}]^3 kurt_l(j)x_{l,M}(j)]^T$$

$$= \frac{8}{L}[G]_j^T[C]_j^3\underline{kurt(j)} \tag{3.7}$$

Where

$$[C]_j = \begin{bmatrix} \underline{w^T(j)}\underline{X_1(j)} & ..... & 0 \\ ..... & ..... & ..... \\ 0 & ..... & \underline{w^T(j)}\underline{X_L(j)} \end{bmatrix} \tag{3.8}$$

is a diagonal matrix.

From (3.7), the updating formula (3.3) becomes

$$\underline{w(j+1)} = \underline{w(j)} + \frac{8}{L}[MU]_j[G]_j^T[C]_j^3\underline{kurt(j)} \tag{3.9}$$

Now, the primary task is to identify the matrix $[MU]_j$ in an optimal sense, so that the total squared kurtosis $\underline{kurt^T(j)}\underline{kurt(j)}$ is maximized. In order to do that, we express the lth kurtosis value in the $(j+1)$th iteration by Taylor's series expansion.

$$kurt_l(j+1) = kurt_l(j) + \sum_{i=1}^{M} \frac{\partial kurt_l(j)}{\partial w_i(j)} \Delta w_i(j) + \frac{1}{2!}\sum_{m=1}^{M}\sum_{n=1}^{M} \frac{\partial^2 kurt_l(j)}{\partial w_m(j)\partial w_n(j)} \Delta w_m(j)\Delta w_n(j) + \ldots$$

$$l = 1, 2, \ldots, L \qquad (3.10)$$

where

$$\Delta w_i(j) = w_i(j+1) - w_i(j) \qquad i = 1, 2, \ldots, M \qquad (3.11)$$

The high order derivative terms in (3.10) are complicated and requires large amount of computation. However, if $\Delta w_i(j)$ is small enough, they can be omitted. In our experimentation, it is found that this is indeed the case.

Based on (3.1), if the expectation operator is dropped, we have:

$$\frac{\partial kurt_l(j)}{\partial w_i(j)} = 4x_{l,i}(j)[\underline{w^T(j)}\underline{X_l(j)}]^3 \qquad (3.12)$$

Then, (3.10) becomes:

$$kurt_l(j+1) = kurt_l(j) + 4[\underline{w^T(j)}\underline{X_l(j)}]^3 \sum_{i=1}^{M} x_{l,i}(j)\Delta w_i(j)$$

$$= kurt_l(j) + 4[\underline{w^T(j)}\underline{X_l(j)}]^3[\underline{X_l^T(j)}\underline{\Delta w(j)}] \qquad (3.13)$$

Write (3.13) for every l, the matrix-vector form of the Taylor expansion becomes:

$$\underline{kurt(j+1)} = \underline{kurt(j)} + 4[C]_j^3[G]_j\underline{\Delta w(j)} \qquad (3.14)$$

From (3.9),

$$\underline{\Delta w(j)} = \frac{8}{L}[MU]_j[G]_j^T[C]_j^3\underline{kurt(j)} \qquad (3.15)$$

Substitute (3.15) into (3.14),

$$\underline{kurt(j+1)} = \underline{kurt(j)} + \frac{32}{L}[C]_j^3[G]_j[MU]_j[G]_j^T[C]_j^3\underline{kurt(j)} \tag{3.16}$$

Define

$$\underline{q(j)} = [G]_j^T[C]_j^3\underline{kurt(j)} = [q_1(j)...q_M(j)]^T \tag{3.17}$$

$$[R]_j = [G]_j^T[C]_j^6[G]_j = [R_{mn}(j)] \qquad 1 \le \text{m, n} \le \text{N} \tag{3.18}$$

the total squared kurtosis for the (*j+1*)th iteration can be written as:

$$\underline{kurt^T(j+1)}\underline{kurt(j+1)} = S_1 + S_2 + S_3 \tag{3.19a}$$

where

$$S_1 = \underline{kurt^T(j)}\underline{kurt(j)} \tag{3.19b}$$

$$S_2 = \frac{64}{L}\sum_{i=1}^{M} q_i^2(j)\mu_{Bi}(j) \tag{3.19c}$$

$$S_3 = \frac{1024}{L^2}\underline{q^T(j)}[MU]_j[R]_j[MU]_j\underline{q(j)} \tag{3.19d}$$

In order to identify optimal $[MU]_j$, the following condition must be met:

$$\nabla\frac{\partial\{\underline{kurt^T(j+1)}\underline{kurt(j+1)}\}}{\partial\mu_{Bi}(j)} = 0 \qquad i = 1,2,\dots, M \tag{3.20}$$

Combine (3.19a) and (3.20), we have:

$$\frac{\partial S_1}{\partial\mu_{Bi}(j)} + \frac{\partial S_2}{\partial\mu_{Bi}(j)} + \frac{\partial S_3}{\partial\mu_{Bi}(j)} = 0 \tag{3.21}$$

Substitute (3.19b), (3.19c) and (3.19d) into (3.21), and use the symmetry property of the matrix $[R]_j$ given in (3.18), the following is obtained:

$$\sum_{k=1}^{M}[q_k(j)\mu_{BK}^*(j)r_{ki}(j)] = -\frac{L}{32}q_i(j) \tag{3.22}$$

where * denotes the optimal value.

Write (3.22) for every *i*, the following matrix-vector equation is obtained:

$$[R]_j[MU]_j^*\underline{q(j)} = -\frac{L}{32}\underline{q(j)} \tag{3.23}$$

From (3.23), we have:

$$[MU]_j^*\underline{q(j)} = -\frac{L}{32}[R]_j^{-1}\underline{q(j)} \tag{3.24}$$

From (3.17), (3.24) and (3.9), the OBAI-ICA algorithm is obtained:

$$\underline{w(j+1)} = w(j) + \frac{8}{L}(-\frac{L}{32})[R]_j^{-1}\underline{q(j)} = \underline{w(j)} - 0.25[R]_j^{-1}\underline{q(j)} \tag{3.25}$$

where $[R]_j$ and $\underline{q(j)}$ are given by (3.17) and (3.18).

Obtain Online Gradient ICA from OBAI-ICA

Now we show that online gradient ICA can be obtained from the general formulation presented above. Let $L = 1$ and

$$\mu_{B1}(j) = \mu_{B2}(j) = \ldots = \mu_{BM}(j) = \mu_B(j) \tag{3.26}$$

OBAI-ICA (3.25) simplifies to:

$$\underline{w(j+1)} = \underline{w(j)} - 0.25\mu_B^*(j)\underline{X(j)}[\underline{w^T(j)}\underline{X(j)}]^3 kurt(j) \tag{3.27}$$

where

$$\mu_B^*(j) = \frac{1}{[\underline{w^T(j)}\underline{X(j)}]^6[\underline{X^T(j)}\underline{X(j)}]} \tag{3.28}$$

If we let $\mu = 0.25\mu_B^*(j)\,|\,kurt(j)\,|$, the following online gradient ICA equation is obtained [25, pp177]:

$$\underline{w(j+1)} = \underline{w(j)} - \mu(sign[kurt(j)]\underline{X(j)}[\underline{w^T(j)}\underline{X(j)}]^3) \tag{3.29}$$

ICA with IA, HA, OBA and BLOCK ICA

The Block ICA

In (3.9), the diagonal elements in the matrix $[MU]_j$ are set equal and time invariant, i.e.,

$$\mu_{B1}(j) = \mu_{B2}(j) = ... = \mu_{BM}(j) = \mu_B \tag{3.30}$$

Then, the weight update equation becomes:

$$\underline{w(j+1)} = \underline{w(j)} + \frac{8}{L}\mu_B[G]_j^T[C]_j^3\underline{kurt(j)} \tag{3.31}$$

This is the block ICA algorithm.

The OBA-ICA algorithm

In order to derive OBA-ICA, it is assumed that the time varying convergence factor is the same for all the coefficients of $\underline{w(j)}$, i.e., equation (3.26) holds. So equation (3.16) becomes:

$$\underline{kurt(j+1)} = \underline{kurt(j)} + \frac{32\mu_B(j)}{L}[C]_j^3[G]_j[G]_j^T[C]_j^3\underline{kurt(j)} \tag{3.32}$$

Then, the total squared kurtosis for the (*j+1*)th iteration (3.19) is modified as follows:

$$\underline{kurt^T(j+1)}\underline{kurt(j+1)} = S_1 + S_2 + S_3 \tag{3.33a}$$

where

$$S_1 = \underline{kurt^T(j)}\underline{kurt(j)} \tag{3.33b}$$

$$S_2 = \frac{64\mu_B(j)}{L}\sum_{i=1}^{M} q_i^2(j) \tag{3.33c}$$

$$S_3 = \frac{1024\mu_B^2(j)}{L^2}\underline{q^T(j)}[R]_j\underline{q(j)} \tag{3.33d}$$

where $[R]_j$ and $\underline{q(j)}$ are given by (3.17) and (3.18).

Taking the derivative with respect to $\mu_B(j)$, and set it to zero:

$$\frac{64}{L}\underline{q^T(j)}\underline{q(j)} + \frac{2048}{L^2}\mu_B^*(j)\underline{q^T(j)}[R]_j\underline{q(j)} = 0 \tag{3.34}$$

So the optimum convergence factor is:

$$\mu_B^*(j) = -\frac{L}{32}\frac{\underline{q^T(j)}\underline{q(j)}}{\underline{q^T(j)}[R]_j\underline{q(j)}} \tag{3.35}$$

In this scenario, the weight update equation (3.9) simplifies to:

$$\underline{w(j+1)} = \underline{w(j)} + \frac{8}{L}\mu_B(j)\underline{q(j)} \tag{3.36}$$

Using the optimum convergence factor (3.35), OBA-ICA is obtained:

$$\underline{w(j+1)} = \underline{w(j)} - 0.25\frac{\underline{q^T(j)}\underline{q(j)}}{\underline{q^T(j)}[R]_j\underline{q(j)}}\underline{q(j)} \tag{3.37}$$

Note that unlike OBAI-ICA (3.25), OBA-ICA (3.37) does not require matrix inversion, which is more computationally efficient. Due to the same reason, OBA-ICA is applicable in cases where the length of the processing block is less than the order of the system, i.e., *L*<*N*. In this scenario, OBAI-ICA does not work, because it involves the inverse of matrix *R*, which is *N* by *N*.

The IA-ICA algorithm

If the block size L=1, a blockwise process reduces to a sequential one. In this case, $[R]_j$ and $\underline{q(j)}$ defined in (3.17) and (3.18) becomes:

$$[R]_j = \underline{X(j)}[\underline{w^T(j)}\underline{X(j)}]^6 \underline{X^T(j)} \tag{3.38}$$

$$\underline{q(j)} = \underline{X(j)}[\underline{w^T(j)}\underline{X(j)}]^3 kurt(j) \tag{3.39}$$

Note that in (3.39), the kurtosis vector has degenerated to a scalar.

From (3.38)

$$[R]_j^{-1} = \frac{[\underline{X(j)}\underline{X^T(j)}]^{-1}}{[\underline{w^T(j)}\underline{X(j)}]^6} \tag{3.40}$$

Substitute (3.39) and (3.40) into OBAI-ICA (3.25), the IA-ICA is obtained:

$$\underline{w(j+1)} = \underline{w(j)} - 0.25kurt(j)\frac{[\underline{X(j)}\underline{X^T(j)}]^{-1}\underline{X(j)}}{[\underline{w^T(j)}\underline{X(j)}]^3} \tag{3.41}$$

The HA-ICA algorithm

HA-ICA is the sequential version of OBA-ICA. Substitute (3.38) and (3.39) into OBA-ICA (3.37), the following weight update equation is obtained after straightforward derivation:

$$\underline{w(j+1)} = \underline{w(j)} - 0.25\frac{kurt(j)\underline{X(j)}}{[\underline{w^T(j)}\underline{X(j)}]^3[\underline{X^T(j)}\underline{X(j)}]} \tag{3.42}$$

Implementation Issues of OBAI-ICA

Elimination of the matrix inversion operation

OBAI-ICA algorithm (3.25) gives the optimal updating formula to extract one row of the separation matrix **W**. The update equation involves the inversion of the $[R]$ matrix, whose dimensionality is equal to the order of the system N. This operation could be inefficient in the case of high order system, because the computational complexity is $O(N^3)$.

When the order of the system is large, an estimate of $[R]$ can be used. The method proposed here is to use a diagonal matrix $[R]_D$ which contains only the diagonal elements of $[R]$. Thus, the complexity of the inverse operation becomes $O(N)$. The adaptive system can repair itself from this approximation and converge to the right solution. The only price is that the number of iterations required for convergence is increased.

Computational complexity

Having eliminated the inversion problem, the dominant factor determining the computational complexity is the block size L for most applications of ICA, because it is typically larger than the order of the system N. It is easily seen that the number of multiplication and divisions of OBAI-ICA is $O(L)$ per iteration, which is equivalent to Fast-ICA.

An additional scaling factor

In practice, it is desirable to introduce one additional adaptation parameter for the equation, based on two considerations.

First, since the high order derivative terms in (3.10) are dropped in our formulation, an additional adaptation parameter can help to ensure reliable convergence.

Second, according to the speed of the time variation, a mechanism should be available to adjust $\underline{\Delta w(j)}$, so that the algorithm can track the time variation no matter how fast the mixing matrix is changing.

Therefore, the following updating formula is obtained based on (3.25):

$$\underline{w(j+1)} = \underline{w(j)} - 0.25\mu[R]_j^{-1}\underline{q(j)} \tag{3.43}$$

The choice of μ should be made according to the convergence property and the speed of mixing matrix's time variation. It is found in our simulation that the optimal value of μ varies according to the speed at which the mixing matrix coefficients vary. On the other hand, Fast-ICA does not have the capability to adapt to time variation even if an additional scaling factor is introduced.

Two versions of OBAI-ICA

Now two versions of OBAI-ICA are introduced depending on whether the same block of data are used from one iteration to another.

In OBAI-ICA1, the data observation matrix $[G]$ stays the same for every iteration, until the algorithm converges. For this case, dependency on iteration index j can be dropped.

In OBAI-ICA2, a completely new block of data is used for every iteration. Upon convergence, the converged demixing vector processes all the data that have been used.

An Alternative for OBAI

The OBAI-ICA algorithm (3.25) requires that the block size *L* is larger than the order of the system *N*, because it involves the inverse of the matrix [*R*] defined in (3.18). This condition is satisfied in most applications, such as the wireless communications application considered in this dissertation.

However, in the case of high order systems, *L*<*N* might arise in certain applications. Although OBA-ICA (3.37) is applicable for *L*<*N*, the performance of OBA-ICA is worse than OBAI-ICA.

In this section, we propose an alternative algorithm for OBAI-ICA, assuming that the original distribution of the signal components is known. The algorithm is named General Optimum Block Adaptation ICA (GOBA-ICA). It is general in the sense that it does not impose any constraint on *L* and *N*.

From Taylor series expansion (3.14), we have the relationship between the kurtosis vectors in the *j*th and (*j+1*)th iteration. Our purpose is to choose $\Delta \underline{w(j)}$ such that finally the kurtosis vector consists of the kurtosis values of individual components.

Assume the independent components are binary signals taking values of either -1 and 1. The kurtosis of the discrete uniform distribution is -2. Therefore, if we achieve convergence in the (*j+1*)th iteration, the kurtosis vector should become a vector of length *L* in which every element is -2.

Based on the above argument, we can let $\underline{kurt(j+1)}$ in (3.14) to be

$$\underline{kurt(j+1)} = -\underline{K} = [-2 \quad -2 \quad \ldots \quad -2]^T \tag{3.44}$$

Let

$$[R]_j = 4[C]_j^3[G]_j$$

(3.45)

Then, according to (3.14), the change in the weight vector becomes

$$\Delta \underline{w(j)} = -[R]_j^{\#}(\underline{Kurt(j)} + \underline{K}) \tag{3.46}$$

where

$$[R]_j^{\#} = \begin{cases} [R]_j^T([R]_j[R]_j^T)^{-1} & L < N \\ ([R]_j^T[R]_j)^{-1}[R]_j^T & L \geq N \end{cases} \tag{3.47}$$

is the pseudo-inverse of the matrix $[R]_\mathrm{j}$ defined in (3.45).

The weight update equation for GOBA-ICA (3.46) is the *Least Square* (LS) solution for (3.14) assuming (3.44).

Note that GOBA-ICA implies the convergence is achieved in one iteration. Thus it is a Newton-based approach. OBAI-ICA falls into the same category in the sense that by finding the optimum values of convergence factor, we are maximizing the energy of the kurtosis vector in one iteration, equivalently.

As with OBAI-ICA, an additional scaling factor can be introduced for GOBA-ICA. So (3.46) becomes

$$\Delta \underline{w(j)} = -\mu[R]_j^{\#}(\underline{Kurt(j)} + \underline{K}) \tag{3.48}$$

The evaluation of pseudo-inverse in GOBA-ICA also involves the matrix inversion operation. The dimensionality of the matrix requiring inversion is the smaller one of *L* and *N*. As with OBAI-ICA, it is sometimes desirable to approximate the matrix in order to simplify the inversion operation.

Conclusions

This chapter develops the Optimum Block Adaptation with Individual adaptation parameters ICA algorithm (OBAI-ICA). The idea is to tailor the learning rates in a gradient-based block algorithm to each iteration and every coefficient in the separation matrix, in order to optimize a cost function that corresponds to a measure of independence. Taylor series expansion is used in conjunction with gradient based block adaptation equation to identify the optimal set of individual convergence factors.

Since OBAI-ICA involves matrix inversion, a simplification of the inversion operation is proposed by approximate the matrix to be inverted using only its diagonal elements. Also, an additional scaling factor is introduced for practical implantation of OBAI-ICA to provide a mechanism for adjusting to different speed of time variation.

The computational complexity of OBAI-ICA per iteration in terms of the block size L is equivalent to Fast-ICA, which is very efficient.

From OBAI-ICA, a family of ICA algorithm can be derived. They are: IA-ICA, HA-ICA, OBA-ICA and Block-ICA. Moreover, the existing online gradient ICA algorithm can be obtained as a simplification of the more general OBAI-ICA algorithm.

To have an individualized block adaptation ICA algorithm that works for the case of $L<N$, an alternative algorithm named GOBA-ICA is also developed. It utilizes the pseudo-inverse of an L by N matrix. The assumption needed in GOBA-ICA is that the distribution of the independent components is known.

Both OBAI-ICA and GOBA-ICA are Newton-based methods, as Fast-ICA. However, because they are adaptive techniques, they possess the capability of tracking time variations in the mixing matrix's coefficients.

CHAPTER FOUR: A GENERAL INTERFERENCE SUPPRESSION TECHNIQUE FOR DIVERSITY WIRELESS RECEIVERS ADOPTING ICA

In Chapter two, Independent Component Analysis technique is summarized and Fast-ICA algorithm is described. In Chapter three, a new ICA algorithm OBAI-ICA and its branches are developed.

In this chapter, an application of ICA is considered based on Fast-ICA, and computer simulation results are presented to confirm the advantages of ICA. In the next chapter, OBAI-ICA is adopted for the same application in a highly dynamic environment, and its performance is compared with Fast-ICA.

For an overview of the cellular communication and smart antenna systems, see [86].

Introduction

As mentioned in the introduction part, digital interference suppression is becoming increasingly important, because the computation power of modern DSPs has developed substantially, and digital implementation of radio functionalities is consistent with the ultimate goal of telecommunication transceiver design: *Software Defined Radio*.

Recently, diversity receiver structures have been proposed which utilizes ICA to suppress CCI [8-9]. In this chapter, a more general scheme is presented that is able to reject simultaneously CCI and image interference. Then a technique solving the order ambiguity inherent to ICA is introduced, which performs the signal selection while extracting components. By exploiting the training sequence embedded into the air interface frame, the method achieves higher computational efficiency. Also, the selection of the related threshold value is convenient.

Wireless Communications in Fading Channels

For most practical wireless communication channels, signal propagation takes place in the atmosphere and near the ground. A signal can travel from transmitter to receiver over multiple reflective paths, which causes fluctuations in the received signal's amplitude, phase, and angle of arrival [32]. This phenomenon, known as *multipath fading*, should be taken into consideration to describe the channel behavior and predict the system performance.

There are two types of fading characterizing mobile communications: large-scale fading and small-scale fading.

Large-scale fading represents the average signal power attenuation or the path loss due to motion over large areas. It is often described in terms of a mean-path loss and a log-normally distributed variation about the mean.

Small-scale fading occurs when there are small changes between a receiver and transmitter, which cause dramatic changes in signal amplitude and phase.

Since the signals travel along random paths, they have random phases. Even small differences in the length of the radio path result in large carrier phase shifts. As a result, the phase response of the wireless channels, for mobile systems operating at frequencies higher than 800MHz, can be modeled as uniformly distributed over the interval [0, 2π).

Regarding the amplitude response, when there is no direct line-of-sight between the transmitter and the receiver, the signal arrives at the receiver as a superposition of many components from a wide range of incident angles. The envelope of such a received signal is statistically described by a *Rayleigh* probability density function (pdf). Consequently, this type of fading is called *Rayleigh* fading. In cellular systems, this occurs if the height of the cellular tower is small. On the other hand, in the presence of a dominant nonfading signal component that is propagating along a line-of-sight path, the fading is called *Rician* fading [33].

Receiver Structure

Figure 1 shows the diversity receiver architecture.

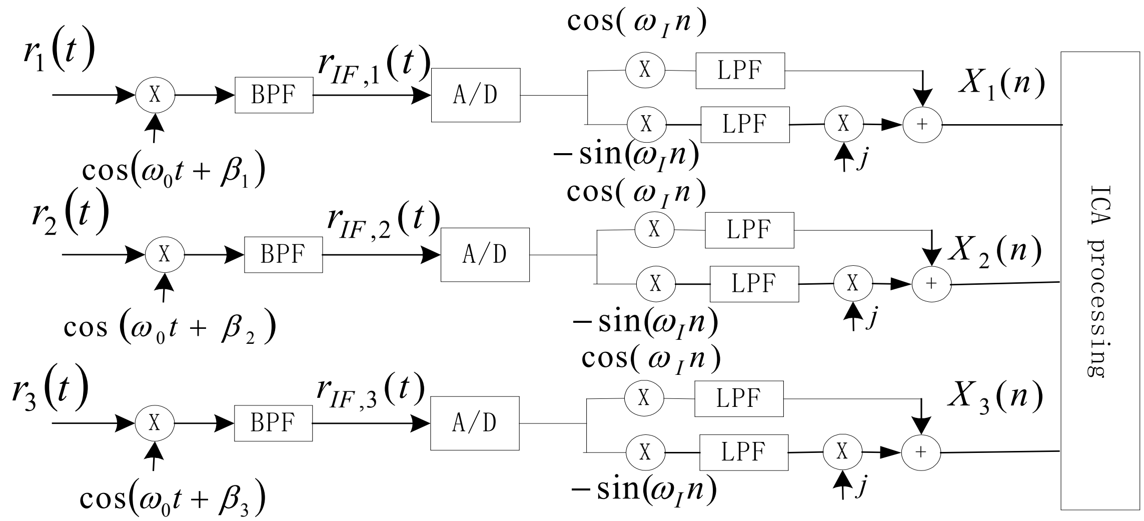

Figure 1. Proposed diversity wireless receiver structure

Three antennas are used, because it is assumed that there are only one principal CCI signal and one principal image signal. Extension to multiple CCI signals and image signals is straightforward and is achieved by addition of antenna elements.

It is required that there is a separation of at least 0.5 wavelengths between each pair of two antennas in order to obtain signals that fade independently.

In Fig.1, the three RF signals received through three antennas are denoted by $r_k(t)$, where k=1, 2 or 3 is the antenna index. ω_0 and ω_I denote the frequencies of the first and second local oscillators. The IF signals are expressed as $r_{IF,\,k}(t)$. Three baseband complex-valued observations are denoted by $X_k(n)$, where n is the discrete time index.

Note that for IF-baseband downconversion, the IF signals are multiplied by $cos(\omega_I n)$ and $-sin(\omega_I n)$. Since the desired signal is situated at positive IF, the IF signal needs to be multiplied by $e^{-j\omega_I n}$. The inputs to DSP are obtained by combining the baseband signal components from both inphase and quadrature branches.

Non-coherent detection is used, thus no phase synchronization is needed in the front-end, and the phase difference of the incoming signal and Local Oscillator (LO) signals are introduced as β_1, β_2 and β_3, respectively. They are all assumed to be uniform over $[0, 2\pi)$.

It is worth mentioning that the downconversion from IF to baseband is performed in the digital domain quadraturely in Fig. 1. This is only necessary if the source signals are QAM signals. Otherwise, the downconversion can be performed in the analog domain with single LO signal. In that case, the A/D converters can be place after LPF at baseband, and the signal observations becomes real-valued.

Signal Model

The channel's fading coefficients are defined as:

$$f_{sk} = \alpha_{sk}\, e^{j\psi_{sk}} \tag{4.1}$$

$$f_{ck} = \alpha_{ck}\, e^{j\psi_{ck}} \tag{4.2}$$

$$f_{ik} = \alpha_{ik}\, e^{j\psi_{ik}} \tag{4.3}$$

where f_{sk}, f_{ck}, f_{ik} are fading coefficients for the desired, CCI, and image signals, respectively; α_{sk}, α_{ck}, α_{ik} and ψ_{sk}, ψ_{ck}, ψ_{ik} are the channel's amplitude and phase responses, respectively.

α_{sk}, α_{ck}, α_{ik} are determined by the type of the fading channel. The distribution of them can be Rayleigh, Rician, or Nakagami. Since the signals travel along random paths, they have random phases. So ψ_{sk}, ψ_{ck}, ψ_{ik} are modeled as uniformly distributed over the interval $[0, 2\pi)$.

In (4.1) ~ (4.3), the time dependency of the fading coefficients is dropped because it is assumed that the wireless channel is quasi-stationary. Thus, the algorithm is applicable to slow time-varying channels. Also, frequency-flat fading is assumed.

Let $s(t)$, $c(t)$ and $i(t)$ denote the complex envelopes of the desired, CCI and image signals, respectively. Thus, the received signal of the kth antenna $r_k(t)$ could be expressed as:

$$r_k(t) = 2\mathrm{Re}\{(s(t)f_{sk} + c(t-\tau_1)f_{ck}\, e^{-j(\omega_0+\omega_I)\tau_1})\, e^{j(\omega_0+\omega_I)t} + i(t-\tau_2)f_{ik}\, e^{-j(\omega_0-\omega_I)\tau_2}\; e^{j(\omega_0-\omega_I)t}\} \tag{4.4a}$$

where $Re\{.\}$ denotes the real part of a signal. The multiplication by 2 is introduced for convenience. τ_1 and τ_2 are the timing delays of $c(t)$ and $i(t)$ relative to the symbol timing reference of *s(t)*. In general, they are independent of each other and can be assumed uniform over the interval [0, *T*], where *T* is the symbol interval.

Equation (4.4a) can be rewritten as:

$$\begin{aligned} r_k(t) = {} & (s(t)f_{sk} + c(t-\tau_1)f_{ck}\,e^{-j(\omega_0+\omega_I)\tau_1})\,e^{j(\omega_0+\omega_I)t} \\ & + (s^*(t)f^*_{sk} + c^*(t-\tau_1)f^*_{ck}\,e^{j(\omega_0+\omega_I)\tau_1})\,e^{-j(\omega_0+\omega_I)t} \\ & + i(t-\tau_2)f_{ik}\,e^{-j(\omega_0-\omega_I)\tau_2}\ e^{j(\omega_0-\omega_I)t} + i^*(t-\tau_2)f^*_{ik}\,e^{j(\omega_0-\omega_I)\tau_2}\ e^{-j(\omega_0-\omega_I)t} \end{aligned} \tag{4.4b}$$

where * denotes complex conjugate.

After the first mixer, the signals are downconverted to IF stage. Then BPFs with a center frequency of ω_I are employed to select the channel and suppress the high frequency components. The output signal of the BPF $r_{IF,\,k}(\mathrm{t})$ is given by:

$$\begin{aligned} r_{IF,\,k}(\mathrm{t}) = {} & (s(t)f_{sk} + c(t-\tau_1)f_{ck}\,e^{-j(\omega_0+\omega_I)\tau_1})\,e^{-j\beta_k}\ e^{j\omega_I t} \\ & + (s^*(t)f^*_{sk} + c^*(t-\tau_1)f^*_{ck}\,e^{j(\omega_0+\omega_I)\tau_1})\,e^{j\beta_k}\ e^{-j\omega_I t} \\ & + i(t-\tau_2)f_{ik}\,e^{-j(\omega_0-\omega_I)\tau_2}\ e^{-j\omega_I t}\ e^{-j\beta_k} + i^*(t-\tau_2)f^*_{ik}\,e^{j(\omega_0-\omega_I)\tau_2}\ e^{j\omega_I t}\ e^{j\beta_k} \end{aligned} \tag{4.5}$$

At this point, A/D conversion is performed. Finally, the IF signals are processed by quadrature downconverters, followed by lowpass filters. After straightforward manipulations, the signal observation corresponding to the *k*th antenna $X_k(n)$ is given by:

$$X_k(n) = s(n)f_{sk}\ e^{-j\beta_k} + c(n)\,f_{ck}\,e^{-j[(\omega_0+\omega_I)\tau+\beta_k]} + i^*(n)f^*_{ik}\,e^{j[(\omega_0-\omega_I)\tau+\beta_k]} \tag{4.6}$$

Thus, the following signal model is obtained:

$$\mathbf{X(n)=AS(n)} \tag{4.7}$$

Where, $\mathbf{X(n)} = \begin{bmatrix} X_1(n) \\ X_2(n) \\ X_3(n) \end{bmatrix};$ (4.8)

$$\mathbf{A} = \begin{bmatrix} f_{s1}\mathrm{e}^{\text{-j}\beta_1} & f_{c1}\mathrm{e}^{\text{-j}[(\omega_0+\omega_I)\tau_1+\beta_1]} & f_{i1}^{*}\mathrm{e}^{\mathrm{j}[(\omega_0-\omega_I)\tau_2+\beta_1]} \\ f_{s2}\mathrm{e}^{\text{-j}\beta_2} & f_{c2}\mathrm{e}^{\text{-j}[(\omega_0+\omega_I)\tau_1+\beta_2]} & f_{i2}^{*}\mathrm{e}^{\mathrm{j}[(\omega_0-\omega_I)\tau_2+\beta_2]} \\ f_{s3}\mathrm{e}^{\text{-j}\beta_3} & f_{c3}\mathrm{e}^{\text{-j}[(\omega_0+\omega_I)\tau_1+\beta_3]} & f_{i3}^{*}\mathrm{e}^{\mathrm{j}[(\omega_0-\omega_I)\tau_2+\beta_3]} \end{bmatrix}; \tag{4.9}$$

and $$\mathbf{S(n)} = \begin{bmatrix} s(n) \\ c(n) \\ i^{*}(n) \end{bmatrix}. \tag{4.10}$$

The signal model (4.7) can be identified by ICA algorithm as long as the mixing matrix **A** is nonsingular, and all signal components are statistically independent.

Since there is no need for analog image rejection, the IF can be chosen to be low. This relaxes the requirements for A/D converters' speed and BPFs' selectivity at the IF stage. In addition, because the interference suppression is performed at baseband, the processing speed required for DSP is lower than techniques that process signals at the IF stage.

This model does not include thermal noise, because in most cases we are dealing with interference-limited conditions. However, even if the strength of the thermal noise is comparable to that of the interferers, the proposed technique is still applicable with a slight change in the algorithm's whitening procedure.

The model developed here adopts complex notation. However, this is optional. Even if the signals are QAM, it is still possible to use real-valued processing algorithms in the DSP, as long as the real and imaginary parts of the baseband observation are not combined. This way, the dimensionality of the signal model is doubled, but the computational complexity stays the same.

Solving the Order Ambiguity

Order ambiguity is inherent to ICA algorithms. Therefore, additional information is needed to identify which one of the extracted components is the desired signal. In deflationary separation, the components tend to separate in the order of decreasing non-Gaussianity. However, this property is not guaranteed. For instance, if BPSK signals amd 16QAM signals are mixed together, in which case they have different non-Gaussianity, it is found in our experiment that one third of the time 16QAM signal will be extracted first. But 16QAM is more Gaussian than BPSK. Even if the property holds anytime, it is still not quite useful in wireless communications scenario. In practice, the signals from different users often adopt the same modulation scheme, which leads to identical non-Gaussianity.

Fortunately, in most digital cellular and Personal Communication Systems (PCS) standards, training sequences are included in the frame structure. For example, within each IS-54/IS-136 burst and GSM's normal burst, there are 28 symbols and 26 symbols that are arranged into eight possible training sequences, respectively. They allow the adaptive equalizer in the mobile or base station receiver to analyze the radio channel characteristics before decoding the user data.

Figures 2 shows the structure of IS-54/IS-136 burst and GSM normal burst, respectively.

6	6	16	28	122	12	12	122
G	R	Data	Training Sequence	data	SACCH	CDVCC	data

IS-54/IS-136 TDMA burst

3	57	1	26	1	57	8.25
Guard	Data	Flag	Training Sequence	Flag	Data Bits	Tail

GSM normal burst

Figure 2. Frame structure of IS-54/IS-136 burst and GSM normal burst

Assuming that *s(t), i(t),* and *c(t)* have different training sequences, the ICA receiver can use this information to perform signal selection. In the process of network frequency planning for cellular systems, special attention is paid to avoidance of the use of the same training sequence between two calls.

When all the components have been extracted, the desired signal can be identified based on the minimization of Euclidian distance:

$$d = \sum_{i=1}^{N_{TS}} | \hat{b}_i - b_{0i} |^2 \tag{4.11}$$

where N_{TS} is the length of the training sequence, $\hat{b}_i$ is the estimate and b_{0i} is the known value of the training sequence symbol.

However, it is obvious that there is a significant amount of computations performed to separate the interfering signals. A practical technique is proposed where the signal selection and ICA separation are performed simultaneously. Thus, the algorithm can be stopped once it is determined that the desired signal has been extracted.

Each time a component is separated, the Euclidian distance *d* between the training sequences of the extracted component and the desired signal is computed. If *d* is less than a predefined threshold value, it is concluded that the desired signal is obtained, and there is no need to continue the ICA processing.

Separation using Kurtosis based Fast-ICA

The separation process is implemented employing the complex-valued deflationary Fast-ICA algorithm. The contrast function is chosen according to the measure of kurtosis because of its simplicity, superior performance for subgaussian signals and robustness against thermal noise.

The procedures of the algorithm have been described in Chapter 2. However, an additional step is needed to solve the order ambiguity. The following eight steps describe the kurtosis-based fast-ICA algorithm, where a signal selection procedure is incorporated in *Step 7.* This procedure provides a mechanism to stop the algorithm once the desired signal has been extracted. As before, *N* source signals are assumed; **W** denotes the separation matrix; **X** represents the observation matrix, which is defined in (4.8). N_f is the processing frame length, and ε is the predefined threshold value. $\mathbf{x_p}$ denotes the *p*th extracted signal.

Step 1. Get the whitened data $\mathbf{X_1}$ by decomposing **X**'s covariance matrix. Set the counter p = 1;

Step 2. Initialize $\mathbf{w_p}$ to a random vector of unity length;

Step 3. Set $\mathbf{w_p} = \frac{2}{N_f}\sum_{n=1}^{N_f}\{X_1(n)[w_p^H X_1(n)]^* \,|\, w_p^H X_1(n)\,|^2 \text{ -}2w_p\}$;

Step 4. Make the estimate of $\mathbf{w_p}$ orthogonal to the subspace spanned by the already extracted rows of **W** using $\mathbf{w_p} = \mathbf{w_p}$ - $\sum_{v=1}^{p-1}(w_p^H w_v)w_v$;

Step 5. Normalize $\mathbf{w_p}$ to unity length;

Step 6. Check the convergence of $\mathbf{w_p}$. If the convergence is not reached, go back to *Step 3*, otherwise proceed;

Step 7. Compute the extracted component $\mathbf{x_p}$ using $\mathbf{w_p}$, and then compute the corresponding Euclidian distance d defined in (4.11). If $d < \varepsilon$, terminate the program and return $\mathbf{x_p}$ as the desired signal, otherwise proceed;

Step 8. Set $p = p + 1$. If $p \leq N$, go back to *step 2*.

The signal selection is performed in *Step 7*. $\mathbf{x_p}$ is obtained differentially in *step 7,* because the phase of the source signals cannot be preserved due to an arbitrary complex-valued multiplicative constant. Moreover, because there is also sign ambiguity associated with ICA processing, the Euclidian distance d should be computed for both $\mathbf{x_p}$ and - $\mathbf{x_p}$. If it is found that $\mathbf{x_p}$ is the negative of the desired signal, - $\mathbf{x_p}$ should be returned.

This ICA algorithm takes advantage of the fact that there is no need to estimate the mixing matrix **A** for this particular application. Also, only one of the components is of interest. Therefore, it is possible to reduce the amount of computation by deflationary separation.

If the training sequences of interfering signals are also known, a more reliable signal selection scheme can be obtained. First, the Euclidian distances between $\mathbf{x_p}$'s training sequence and the known training sequences of all signal components are computed. Then, the minimum of the distances is determined. If the minimum corresponds to the known training sequence of the desired signal, it is concluded that $\mathbf{x_p}$, or possibly - $\mathbf{x_p}$, is the desired signal. Thus, the choice of a threshold value ε is avoided.

Simulation Results

The presented technique is evaluated via extensive Monte Carlo simulations in Matlab for QPSK receivers. Sample representative results are presented here for Rayleigh fading and signal components of equal strength.

However, it is found that the result remains unchanged if the relative signal strengths are varied or different fading conditions are assumed. This is because the basic requirement for successful ICA separation is that the mixing matrix should be nonsingular. Typically, the source signals are all normalized to unit variance and their relative powers are included in the mixing matrix. Therefore, the only parameters affecting the performance of the proposed technique are the length of the processing frame (N_f) and the number of interfering signals (N). Note that the number of source signals is N+1.

The performance is measured by the Signal-to-Interference Ratio (SIR) defined as:

$$\text{SIR} = 10 \log_{10} \{\frac{1}{N_f}\sum_{k=1}^{N_f}\frac{|s(k)|^2}{[||s(k)|^2 - |y(k)|^2|]}\} \tag{4.16}$$

where *s(k)* is the *k*th sample of the desired signal, *y(k)* is the estimate of the *s(k)* obtained at the output of the ICA processing unit. The SIR represents the average ratio of the desired signal power to the power of estimation error signal.

First, it is assumed that the training sequence is known for all signal components. Thus, no threshold value is needed. N_f is varied from 100 to 1000 symbols with step size 50. For each N_f, simulation is performed 100 times to get the average performance. N=2, 3, and 4 are used. The resulted average SIR is shown in Fig. 3.

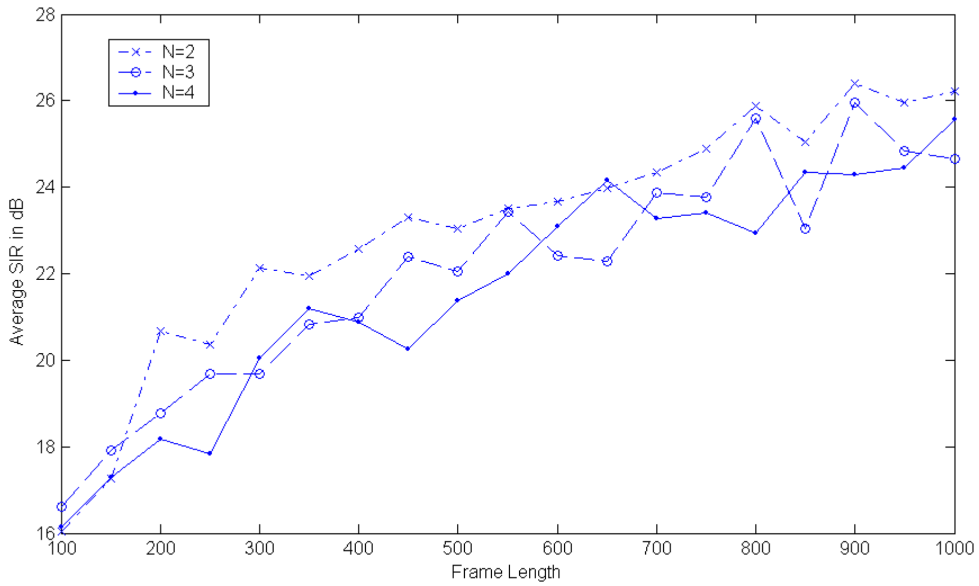

Figure 3. Performance of ICA processing in the presence of *N* interferers

Convergence speed is shown in Figs. 4 - 6 in terms of the number of iterations required. For comparison, the convergence speed for conventional ICA, where all the components are extracted before signal selection, is also plotted. This corresponds to the case where *Step 7* in the procedures described in the previous section is omitted.

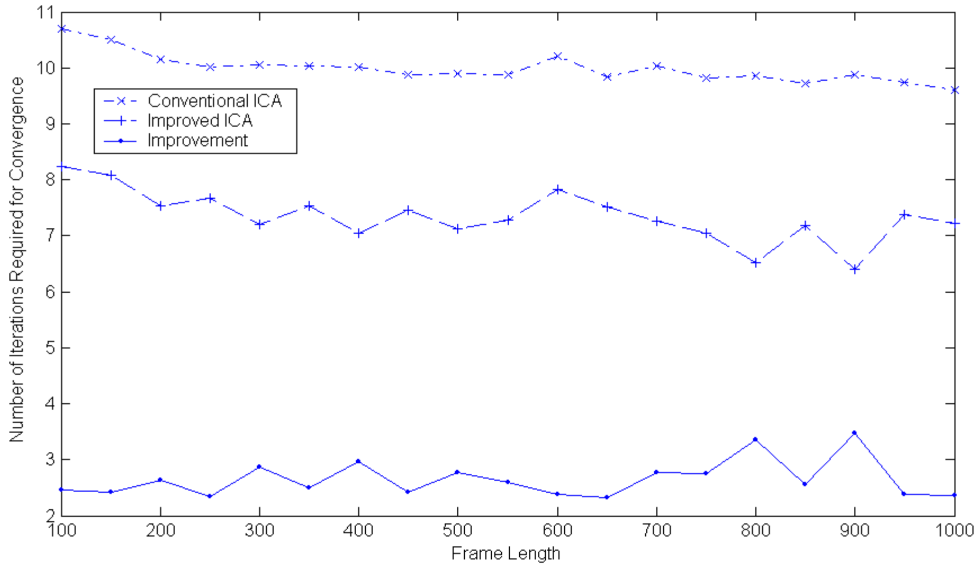

Figure 4. Convergence speed for two interfering signals (*N*=2)

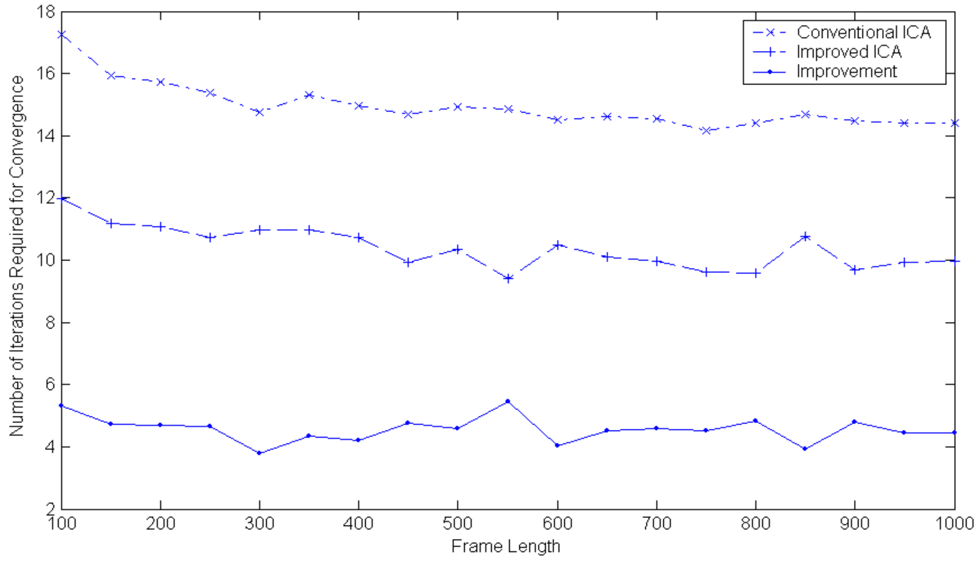

Figure 5. Convergence speed for three interfering signals (*N*=3)

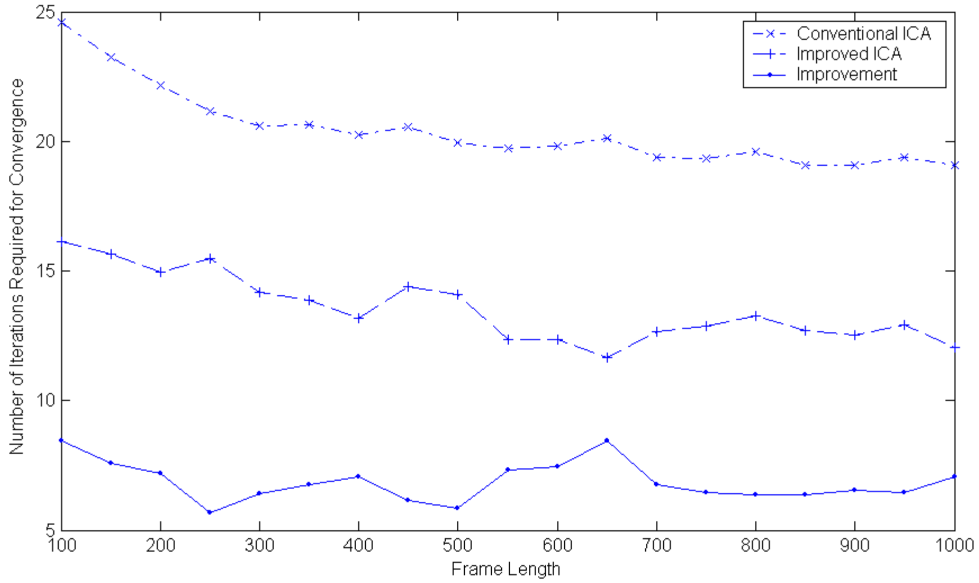

Figure 6. Convergence speed for four interfering signals (*N*=4)

It is observed from Fig. 3 that an average SIR of 16dB to 26dB is achievable. The performance generally improves as N_f increases, because the statistical independence is easier to establish for longer frames.

In Figs. 4 - 6, it is obvious that the improved version of ICA achieves faster convergence than the conventional ICA, and the improvement becomes more significant as *N* increases. The percentage saving in computations is 22.9% to 35.2% for *N*=2, 25.6% to 36.6% for *N*=3, and 26.8% to 41.9% for *N*=4.

Now we make a brief comment regarding the selection of the threshold value ε in the case that only the training sequence of the desired signal is known.

GSM burst is used as an example. Each time slot has 156.25 bits, 26 of which are training bits. Therefore, the length of the training sequence is 26/156.25 = 16.64% of the processing frame length.

Through simulations for different frame lengths, it is found as the length of the training sequence increases, the allowable range of ε also increases. Consequently, the selection of ε becomes easier.

However, even for relative short frame lengths such as 300 symbols, which means the training sequence has 50 symbols, any value of ε between 1 and 7 is allowable. This is a large enough range for ε to be easily identified.

Here we provide a comparison of the kurtosis-based Fast-ICA with the Equivariant Adaptive Separation via Independence (EASI) algorithm presented in [34]. For this purpose, one interferer is used and the performance measure Signal-to-Interference-Rejection-Ratio (SIRR) is redefined by elements of the matrix **H=WA** as:

$$\mathrm{SIRR} = \frac{1}{2}\left[\left|20\log_{10}\left(\frac{|H_{11}|}{|H_{12}|}\right)\right| + \left|20\log_{10}\left(\frac{|H_{22}|}{|H_{21}|}\right)\right|\right] \tag{4.17}$$

The resulting performance obtained by kurtosis-based Fast-ICA is plotted in Fig. 7.

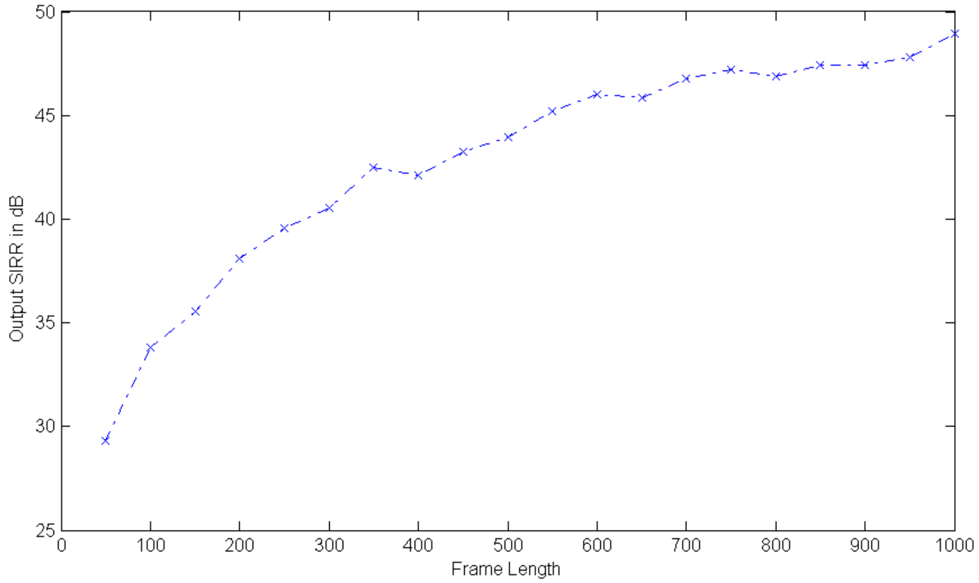

Figure 7. ICA performance using the alternative performance measure (*N*=1)

It is clear that an SIR of 30 to 50 dB for frame lengths of 50 to 1000 symbols is achieved. According to Fig. 9 in [34], the achievable SIR of EASI is around 33dB. Our simulation also shows that kurtosis-based Fast-ICA achieves convergence in 5 iterations, where the EASI algorithm converges after 200 iterations, according to Fig. 10 in [34]. This is not surprising, because Fast-ICA is essentially Newton's method, and EASI is a sequential algorithm. Therefore, for quasi-stationary channel conditions, Fast-ICA is preferable to EASI.

<u>Fast-ICA with Overlapping Blocks for Rapidly Time-varying Channels</u>

The above-presented technique assumes that the wireless channel is quasi-stationary, which means the channel response stays approximately constant within one processing frame.

For the case of highly dynamic environment, the performance of the proposed method will degrade. In this section, a straightforward modification is introduced to deal with this situation.

Figure 8 shows the architecture of data reusing ICA processor adopting shifting technique.

In this scheme, the processing block is shifted right one symbol at a time, with the oldest symbol dropped and a new symbol incorporated. We can have two choices as to how to generate the symbol estimation. First, take only the first symbol in each estimated block; second, average over N_f consecutive block estimations to generate the estimated symbol.

It should be noted that the variations of the mixing matrix should be *slow* in the sense that its time scale is much larger than the sampling interval. In practical communication systems, this is a reasonable assumption.

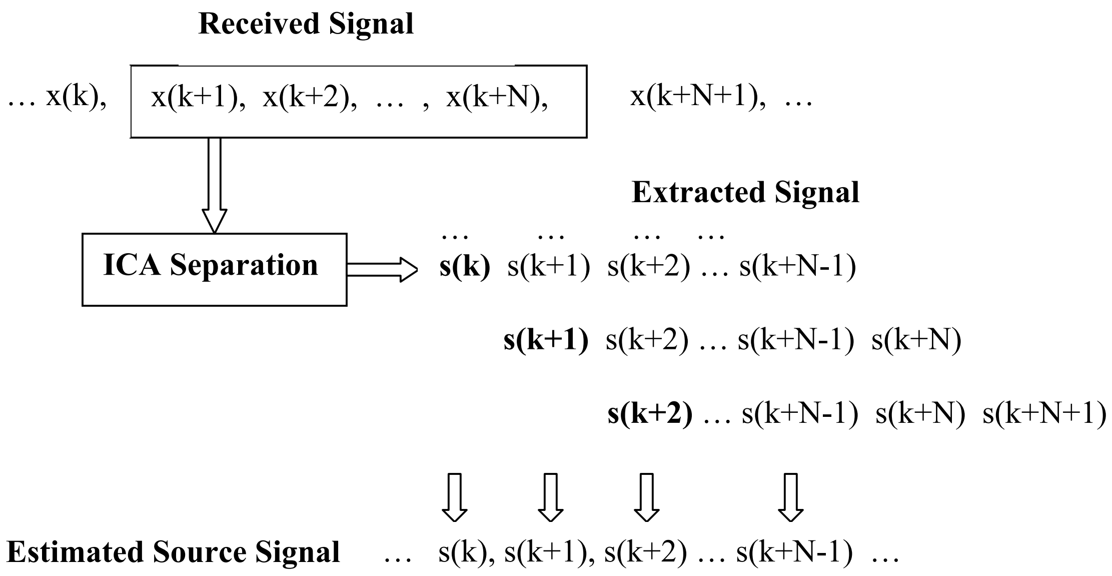
Received Signal
… x(k),
x(k+1), x(k+2), … , x(k+N),
x(k+N+1), …
Extracted Signal
ICA Separation
… … … …
s(k) s(k+1) s(k+2) … s(k+N-1)
s(k+1) s(k+2) … s(k+N-1) s(k+N)
s(k+2) … s(k+N-1) s(k+N) s(k+N+1)
Estimated Source Signal … s(k), s(k+1), s(k+2) … s(k+N-1) …

Figure 8. ICA processing with overlapping blocks

The ICA processor described here is a special case of the overlapping technique, where *K* oldest symbols are dropped and *K* new symbols are incorporated. In our case, *K*=1. It is always possible to vary the value of *K* to provide tradeoff between the computational complexity and the resulting separation performance. In practice, as computational power of DSP hardware is continuously becoming cheaper and more widely available, it is preferable to shift only one symbol at a time.

To evaluate the effectiveness of the modified processing technique, time-variations are simulated as a changing mixing matrix:

$$\mathbf{A} = \begin{bmatrix} 1 + je^{-\frac{t}{T}} & 2 + j \\ 2 - j & 1 - je^{\frac{t}{2T}} \end{bmatrix}. \qquad (4.18)$$

Where T=10,000. The selection of the time-varying parameter is arbitrary, but it should ensure that the time constant of the variation is much larger than the sampling period.

To compare the performance of overlapping ICA with conventional ICA, the average SIR defined in (4.16) obtained by both algorithms in the presence of three interferers is plotted in Fig. 9.

The simulation set up is identical as before.

For overlapping ICA, we experimented taking the average of N_f consecutive estimations for each symbol, as well as taking the first component directly.

It is clearly seen that overlapping ICA achieves better results for frame lengths less than 200 symbols. Smaller N_f achieves better performance because longer frames violates the quasi-stationarity requirement of the mixing matrix. Also, taking the first component is superior to averaging over N_f estimations. This is probably due to the fact that averaging over non-stationary channel parameters will deteriorate the performance.

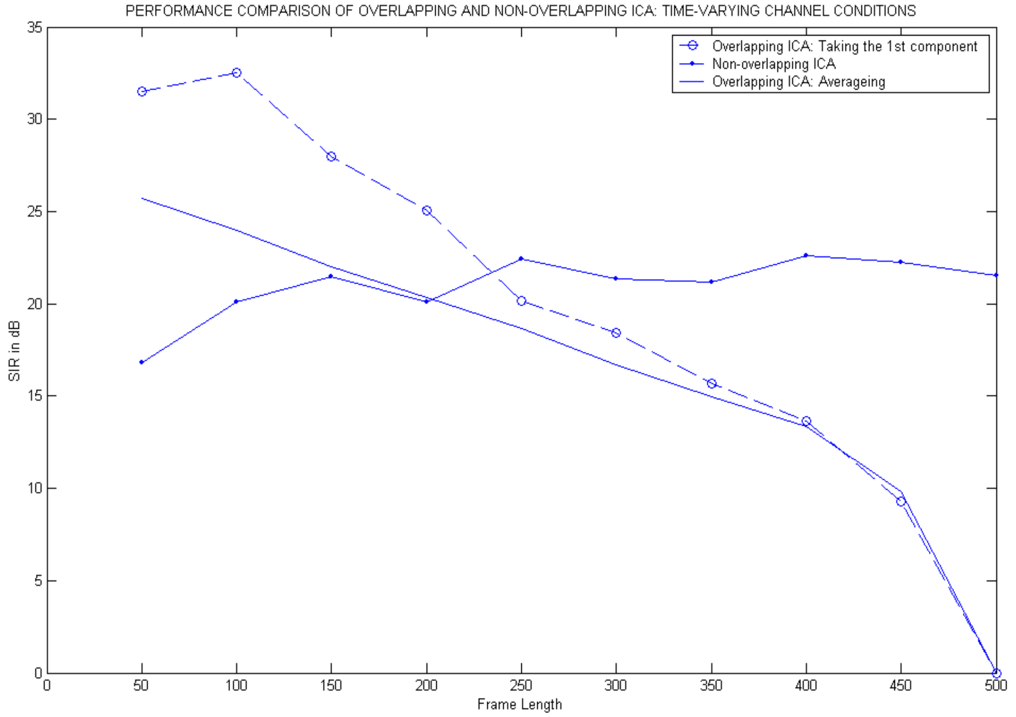

Figure 9. **Performance comparison of overlapping and conventional ICA in the presence of three interferers**

If for some reason, e.g. the available input buffer is small, we have to use a small N_f, adopting overlapping ICA is beneficial even if the channel is quasi-stationary, because it can help solving the problem of insufficient independence.

Although it is shown that the overlapping technique can help to deal with a rapidly time-varying mixing matrix, it has two significant drawbacks. First, the frame needs to be very small, i.e., less than 250 symbols. This is a disadvantage if longer frames are desirable because of the data rate requirement. Second, the overlapping incurs a significant amount of increase in the computational complexity.

Therefore, a better solution is needed. In the next chapter, OBAI-ICA developed in Chapter three is applied. It will illustrate that OBAI-ICA is a superior algorithm that has attractive features in dealing with time-varying environment.

Conclusions

In this chapter, the Fast-ICA algorithm is utilized to perform image signal and CCI suppression simultaneously in diversity wireless receivers. The advantages of the technique include: there is no need for phase synchronization and image rejection filters [35]; IF can be chosen low, so that requirements for BPFs' selectivity and A/D converters' speed are relaxed; the processing speed is low because of baseband processing. Also, no training stage is required [36-37].

Simulation results indicates that, the performance of the presented method is robust to input Signal to Interference Ratios, which means it provides a solution to the *near-far* problem. Also, it is robust to the type of fading channels, and the performance does not degrade for large number of interferers.

The presented technique exploits the fact that there is no need to estimate the mixing matrix, and only the desired signal is of interest. Hence, the amount of computations can be reduced if signal selection utilizing training sequences is implemented simultaneously with signal separation. Computer simulations show that compared with conventional ICA, a saving of 23% to 42% in computations is achievable for QPSK receivers with 2 to 4 interfering signals. This represents considerable computational saving and significantly faster detection. Also, the threshold value associated with the improved method is easy to be identified.

The last section also presents a data-reusing version of the proposed technique to deal with fast time-varying channel conditions. Simulation results confirm its effectiveness.

CHAPTER FIVE: APPLICATION OF OBAI-ICA IN HIGHLY DYNAMIC ENVIRONMENT

In Chapter four, a general interference suppression technique is proposed for diversity wireless receivers. Fast-ICA algorithm is utilized to perform the separation of the desired signal from image signal and co-channel interference. Computer simulations confirm the effectiveness and advantages of the technique.

However, it is assumed in Chapter four that the parameters of the wireless channel remain unchanged over the period of the processing frame, i.e., the channel is quasi-stationary. This assumption is no longer valid when the channel is rapidly time varying. For cellular communications, this could arise from very high user mobility or handover between two base stations.

In this scenario, if Fast-ICA is still to be used, there is a dilemma in choosing the processing frame length, as mentioned in Chapter three. The performance will deteriorate significantly. In fact, the convergence of Fast-ICA may be problematic.

In this chapter, OBAI-ICA is used in comparison with Fast-ICA. Rapid time variation is simulated and the performance of OBAI-ICA and Fast-ICA is studied by Matlab simulation.

Receiver Structure and Formulation of the Signal Model

Figure 10 shows the architecture of a dual-antenna BPSK receiver. It is assumed that there is only one interferer, which can be either image signal or CCI. Note that the highly selective image rejection at the RF stage, which is usually required for signal branch receiver, is eliminated.

First the received signal $r_k(t)$ is downconverted from RF to IF (k = 1 or 2 is the antenna index), followed by a bandpass filter to perform adjacent channel suppression. Then, the IF signal $r_{IF,k}(t)$ is downconverted to baseband and lowpass filtered. The baseband signal $r_{BB,k}(t)$ is digitized to obtain signal observation $X_k(n)$, which is fed into DSP for further processing.

ω_0 and ω_I are the frequencies of the first and second LO signals, α_1 and α_2 are phase differences between the LO signals and the received signals. $X_1(n)$ and $X_2(n)$ are two baseband signal observations.

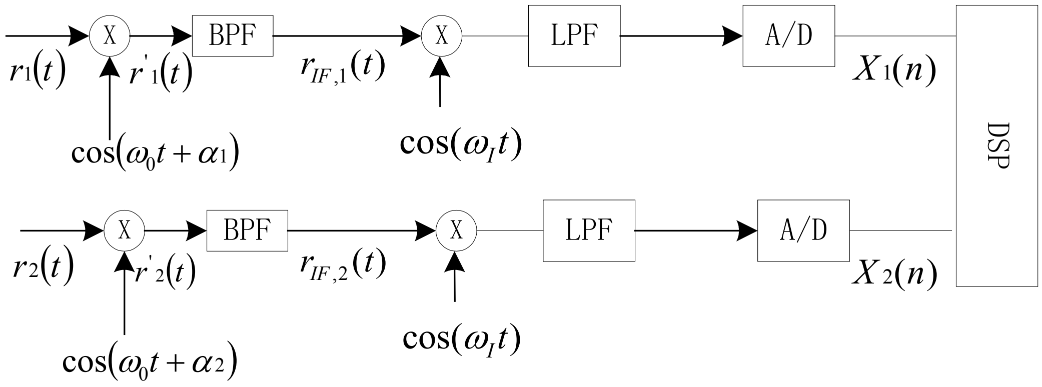

Figure 10. The structure of a dual-antenna BPSK receiver

Note that the only difference between the structure presented here and the one in Chapter four is that, the downconversion from IF to baseband is no longer quadrature since the signals are BPSK modulated. Therefore, both the mixing matrix coefficients and the source signals are real valued.

After similar derivation as in Chapter four, the following signal model is obtained assuming the interferer is an image signal:

$$\mathbf{X(n)} = \begin{bmatrix} X_1(n) \\ X_2(n) \end{bmatrix} = \begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \end{bmatrix} \begin{bmatrix} s(n) \\ i(n) \end{bmatrix} = \mathbf{AS(n).} \qquad (5.1)$$

where, $a_k = Re\{f_{s,k} e^{-j\alpha_k}\}$ and $b_k = Re\{f_{i,k} e^{-j(\omega_0 - \omega_I)\tau} e^{-j\alpha_k}\}$. $f_{s,k}$ and $f_{i,k}$ are the fading coefficients for the desired and interfering signals, respectively. τ is the timing delay of the image signal with respect to the symbol timing reference of the desired signal.

In our simulations, the mixing matrix coefficients a_k and b_k will be varied with time over each processing frame. Both OBAI-ICA and Fast-ICA are applied to signal model (5.1) to separate the two source signals.

As mentioned in Chapter three, OBAI-ICA has two versions, i.e., OBAI-ICA1 and OBAI-ICA2, depending on whether the same block of data are used from one iteration to another. Obviously, OBAI-ICA1 is more similar and comparable with Fast-ICA, because they both process the same block of data until convergence.

Types of Time Variation

There are two types of time variation that can arise in cellular mobile communications.

If the mobile user is moving at a high speed, it is more appropriate to model the channel change as continuous linear time variation in mixing matrix's coefficients. In this case, the ICA algorithm will seek a compromising separation matrix that could recover the source signals with minimum error.

If the user is experiencing handover between two service towers, an abrupt change in mixing matrix's coefficients is suitable. In this scenario, the ICA algorithm will seek to converge to the new demixing matrix as soon as possible. Note that the ICA processing will only be affected when the abrupt change occurs within one processing block. This is the case studied in our simulation. It is expected that the separation performance will degrade for the frame in which the abrupt change occurs, but we are interested in how fast the system can track the variation and adjust itself to a new demixing matrix. In practice, it is desirable to identify the exact location of the abrupt change within the processing block. In the next section, such a technique is introduced.

It is worth mentioning that in practice, it is quite possible that both types of time variation are present.

A Binary Search Technique for Locating the Position of Abrupt Changes

When an abrupt change occurs within a processing block, the performance for the block degrades significantly, especially when the block size is large. This is because the converged demixing vector is a compromise between two completely different channel parameters. In order to deal with this situation, we propose to locate the position of abrupt changes within the block. This technique will improve the performance if the performance degradation is indeed due to a sudden change in the channel condition.

In the search procedure, we utilize the demixing matrices obtained through the previous block, **W1,** and the subsequent block **W2.**

First, the block is evenly divided into two sub-blocks. **W1** is used to process the first sub-block, while **W2** is used to process the second sub-block.

If the separation performance for the second sub-block is better, we may conclude that the abrupt change occurs within the first sub-block. Otherwise, we conclude that the abrupt change occurs within the second sub-block.

Thus, after the above procedure, the location of the abrupt change is narrowed down to a sub-block. The search process can be continued by dividing that sub-block evenly and using **W1** and **W2** to process the two sub-blocks, respectively. This procedure can be repeated until the location of the abrupt change is narrowed down to a very small range.

Once the location is identified, the symbols before the abrupt change are processed by **W1**, and the symbols after the abrupt change are processed by **W2**.

Simulation Results

The performance measure is Signal to Interference Ratio (SIR) defined as:

$$\mathbf{SIR} = 10 \log_{10} \left(\frac{1}{N_f} \sum_{k=1}^{N_f} \frac{s(k)^2}{[s(k) - y(k)]^2} \right) \qquad (5.2)$$

where *s(k)* is the *k*th sample of the desired signal, *y(k)* is the estimate of the *s(k)* obtained at the output of the ICA processing unit. **SIR** represents the average ratio of the desired signal power to the power of the estimation error.

As mentioned earlier, the relative strength of *s(t)* and *i(t)* and the type of fading channel does not affect the performance. This is because ICA algorithms only require the mixing matrix to be non-singular. As long as that requirement is met, the exact values of coefficients do not affect the separation performance.

For continuous linear time variation, the mixing matrix simulated is chosen as:

$$A = \begin{bmatrix} 1 + l\Delta & 0.5 \\ 0.7 & 2 + l\Delta \end{bmatrix} \tag{5.3}$$

Where $l = 1, 2, \ldots, N_f$, and Δ is parameter reflecting the speed of channel variation. Here, it is assumed that the channel's transfer function is frequency-flat over the signal band. Also, the sampling interval of the receiver's A/D converter is much smaller than the scale of the channel's time variation.

In our simulation, the block size N_f is varied from 50 symbols to 1000 symbols, with a step size of 50. For each block size, the calculation of performance is averaged over 100 simulation runs. The total number of symbols estimated is 1000.

Figure 11 shows the performance of OBAI-ICA1 and Fast-ICA under stationary conditions. It is clearly seen that their performance is equivalent.

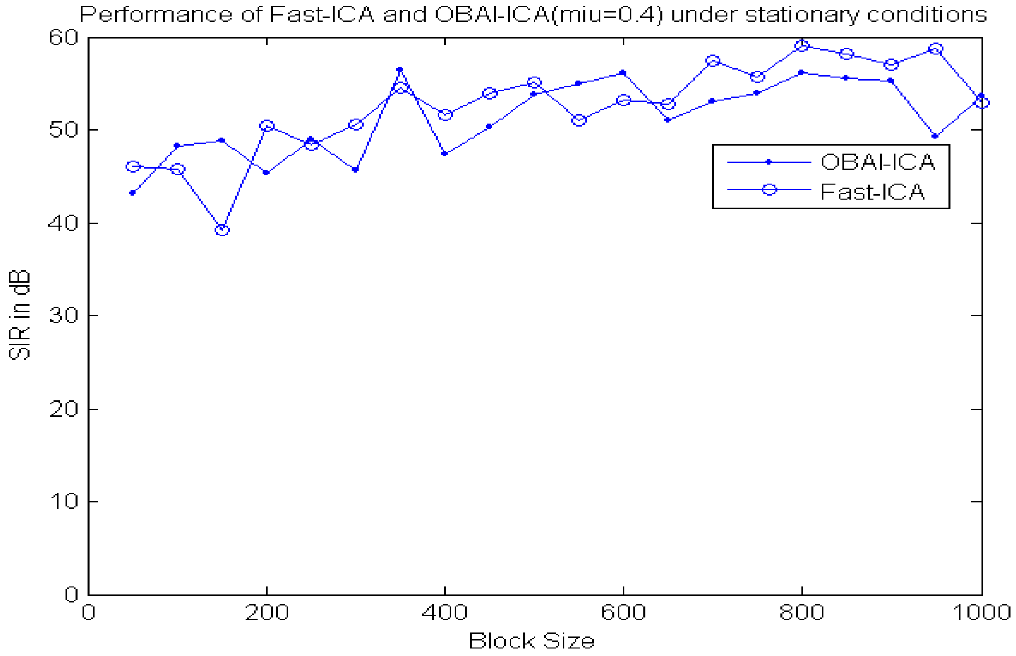

Figure 11. Performance of Fast-ICA and OBAI-ICA under stationary channel conditions

Figures 12 and 13 show the performance and convergence speed of OBAI-ICA1 and Fast-ICA for Δ = 0.01. The additional step size μ in OBAI used is 0.5.

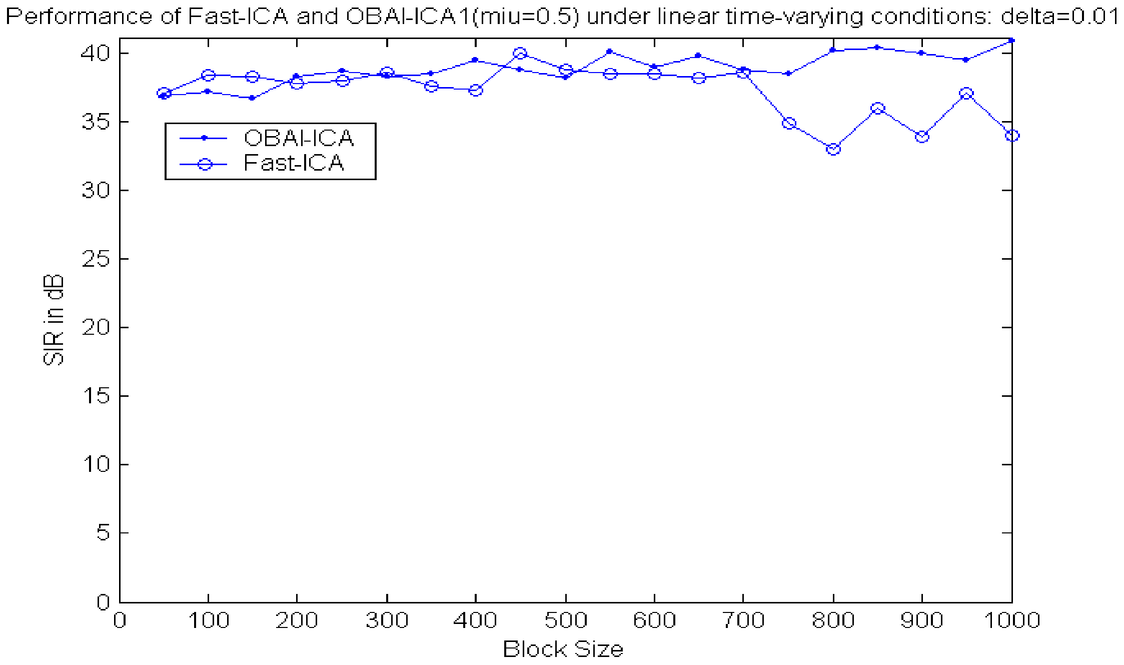

Figure 12. Performance of Fast-ICA and OBAI-ICA1 for slow linear time variation

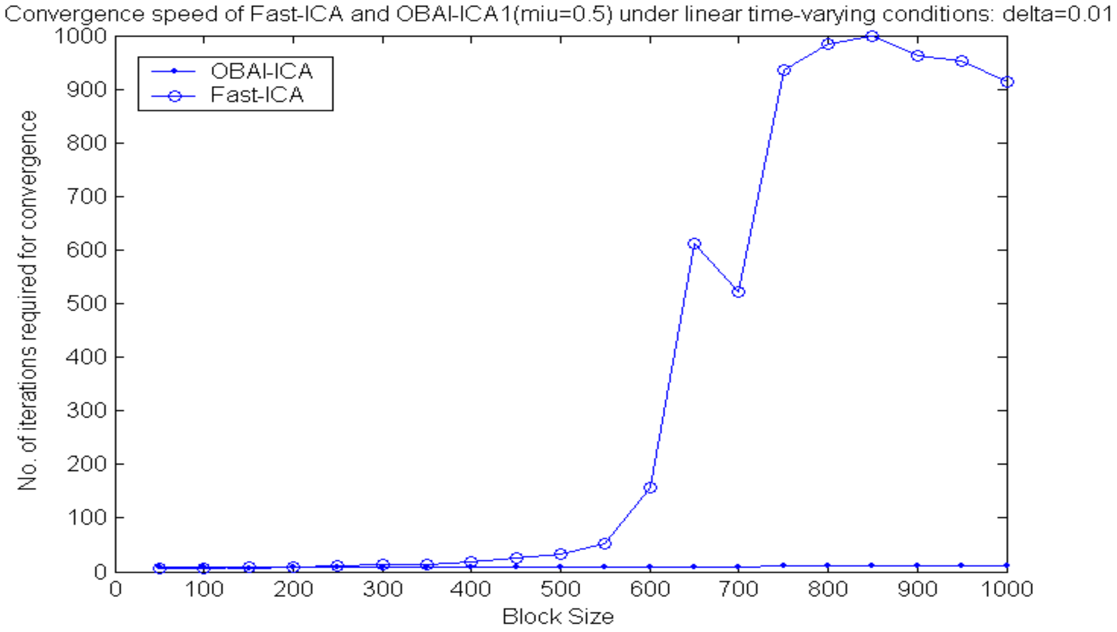

Figure 13. Convergence of Fast-ICA and OBAI-ICA1 for slow linear time variation

These two figures show the performance of OBAI-ICA1 and Fast-ICA under slow linear time variation conditions.

It is seen that the two algorithms have similar performance except for longer blocks, in which case OBAI has better performance. This indicates OBAI has better capability in dealing with time variation within one processing block. Also, Fast-ICA converges very slowly for long blocks, while OBAI-ICA1 always converges within 20 iterations regardless of the block size.

For faster time-variation, i.e., Δ = 0.1, 0.5, 1, Fast-ICA fails to converge within several hundred iterations, so only OBAI-ICA is used. The performance and convergence speed for OBAI-ICA1 are given in Figs. 14 and 15. The optimal μ values are given for every Δ.

Again, it is seen that OBAI-ICA1 always converges within 20 iterations. It is observed that a larger μ should be used for faster time variation, as expected.

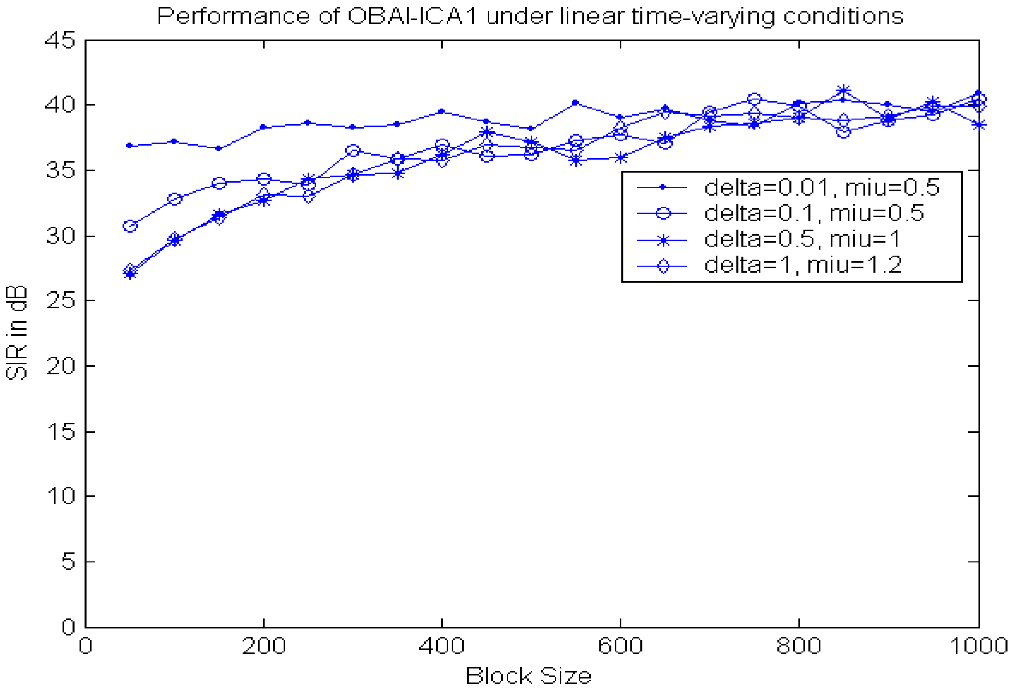

Figure 14. Performance of OBAI-ICA1 under various linear time varying conditions

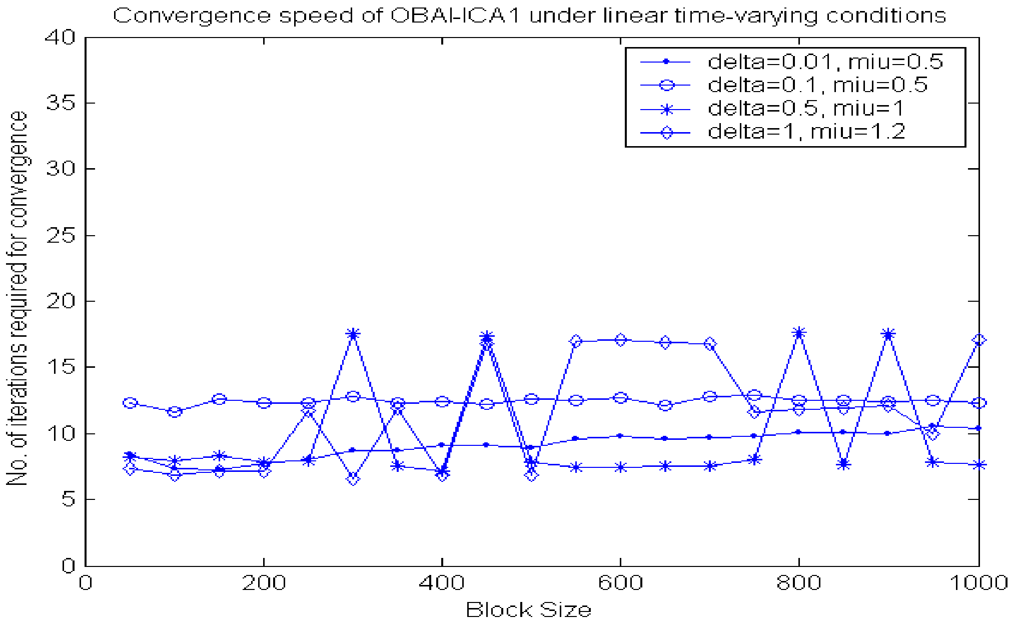

Figure 15. Convergence of OBAI-ICA1 under various linear time varying conditions

Next, the tracking capability of Fast-ICA and OBAI-ICA1 is compared under abrupt changing channel condition. To simulate this condition, an abrupt change of mixing matrix is introduced in the middle of each processing block.

Figures 16 and 17 show their performance and convergence speed. As expected, the performance of both algorithms degrades compared with the case of continuous time variation. However, OBAI-ICA1 converges much faster than Fast-ICA. This means OBAI-ICA1 has much superior tracking capability to abrupt time variation.

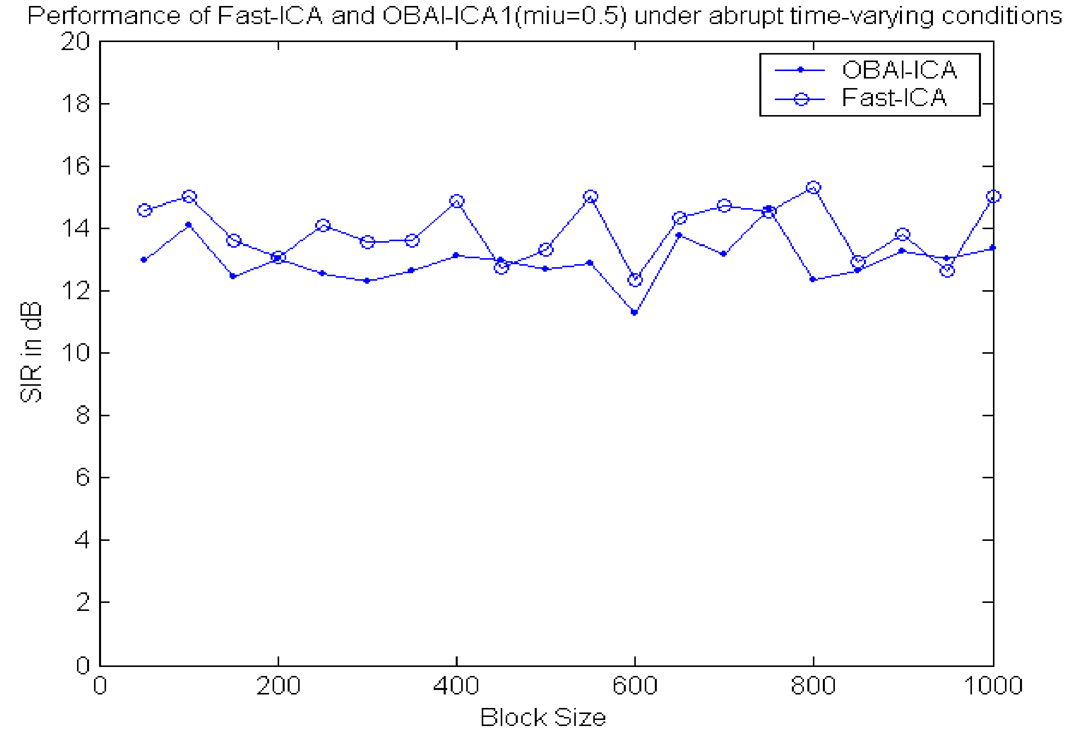

Figure 16. Performance of OBAI-ICA1 and Fast-ICA for abruptly time variation

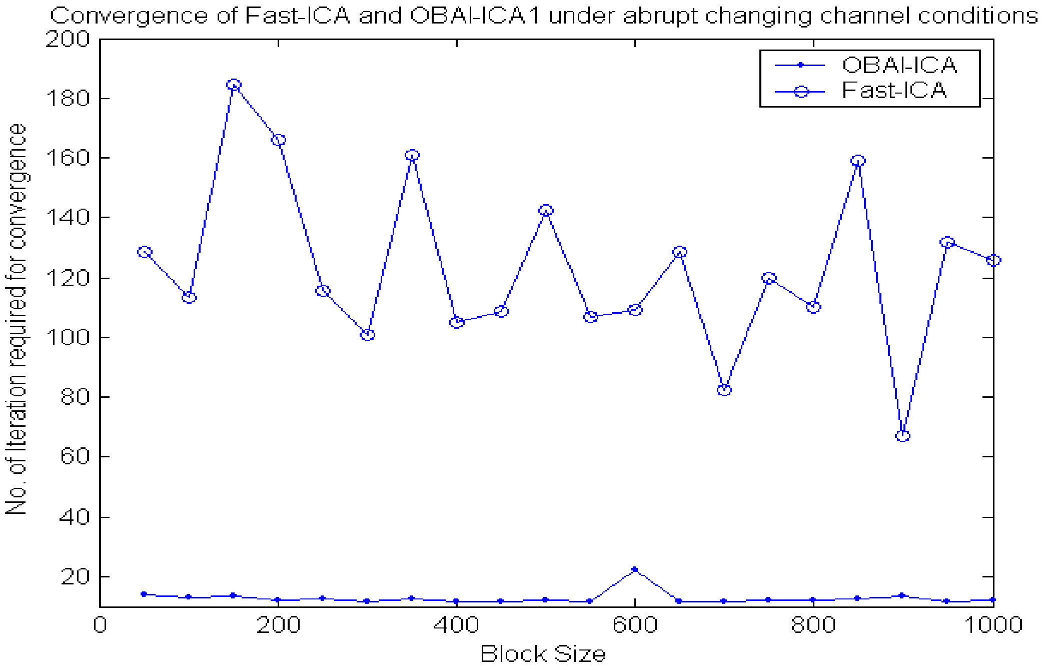

Figure 17. Convergence of OBAI-ICA1 and Fast ICA for abruptly time variation

Following the detection of an abrupt change within a certain block, the binary search technique introduced in the previous section is simulated to detect the location of the abrupt change. In Figs. 18 and 19, the performance for three consecutive blocks is shown, when an abrupt change in the middle block is simulated. Figure 18 shows the performance without binary search, while Fig. 19 plots the performance after applying the technique.

As seen, the performance for the block in which the abrupt time change occurs is substantially improved when the binary search is employed to locate the position of the channel change within the block.

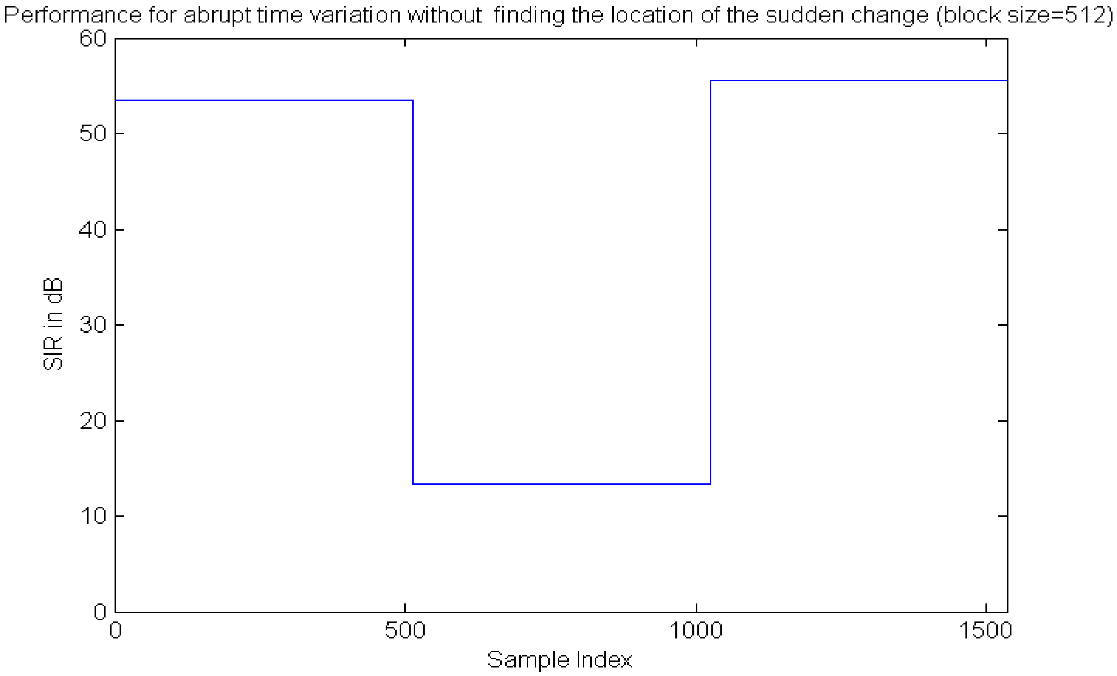

Figure 18. Performance for abrupt time variation without binary search

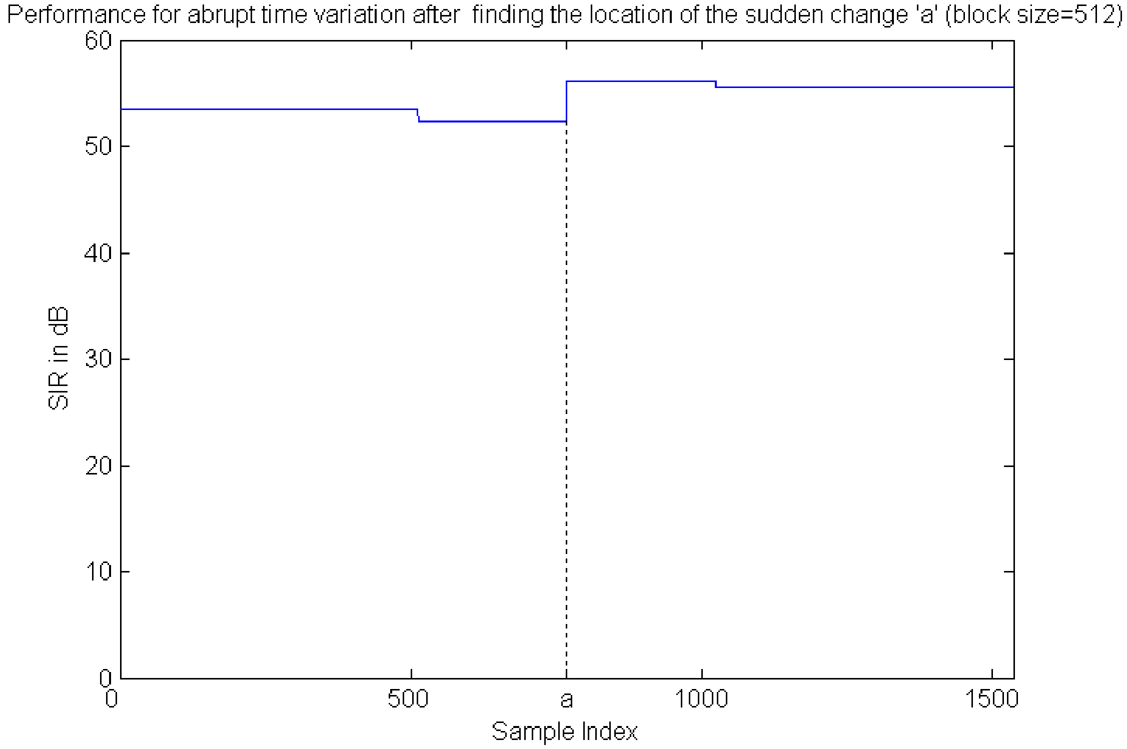

Figure 19. Performance for abrupt time variation after locating the position of the sudden change

OBAI-ICA2 is also simulated in our experimentation. It is found that OBAI-ICA2 converges slowly for abrupt time variation.

For continuous linearly time varying channels, however, it is advantageous over OBAI-ICA1, because the choice of the additional adaptation size μ is robust to the speed of channel's time variation. Figure 20 shows the performance of OBAI-ICA2. For Δ = 0.1, 0.5, 1, the optimal μ is fixed at 0.1. One more advantage of OBAI-ICA2 is that, all 1000 symbols can be extracted every time the algorithm converges, while for OBAI-ICA1, only the data in the processing block can be obtained.

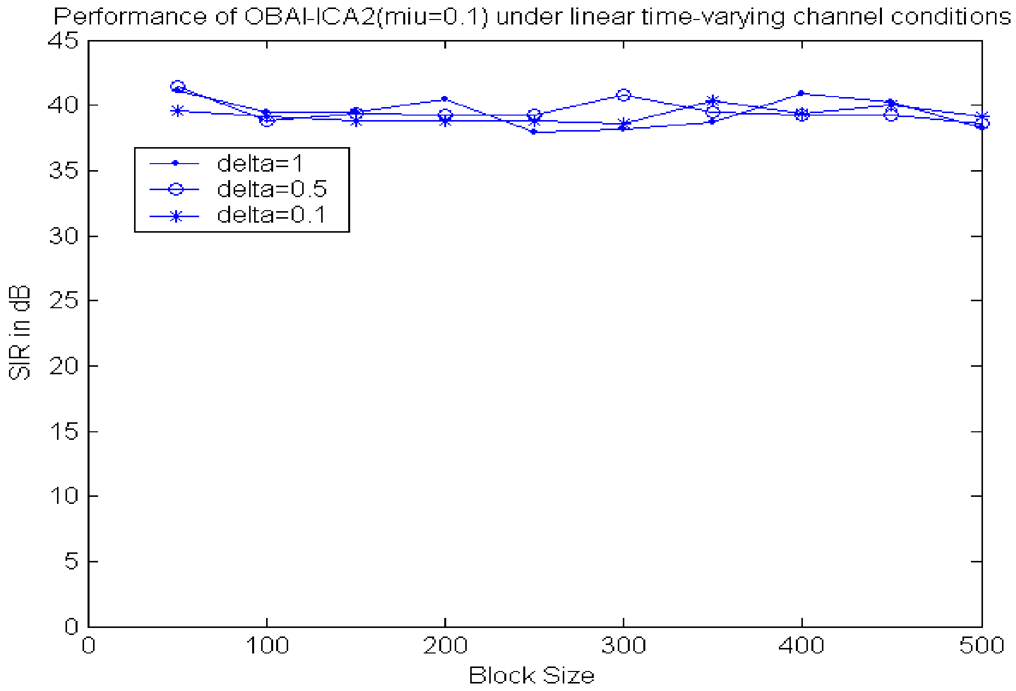

Figure 20. **Performance of OBAI-ICA2 under linearly time varying conditions**

In addition, the GOBA-ICA algorithm, as defined in (3.48), is also simulated under the same simulation conditions. The results are plotted in Figs. 21-25. It is observed that GOBA-ICA performs similarly as OBAI-ICA, but its convergence property is slightly worse.

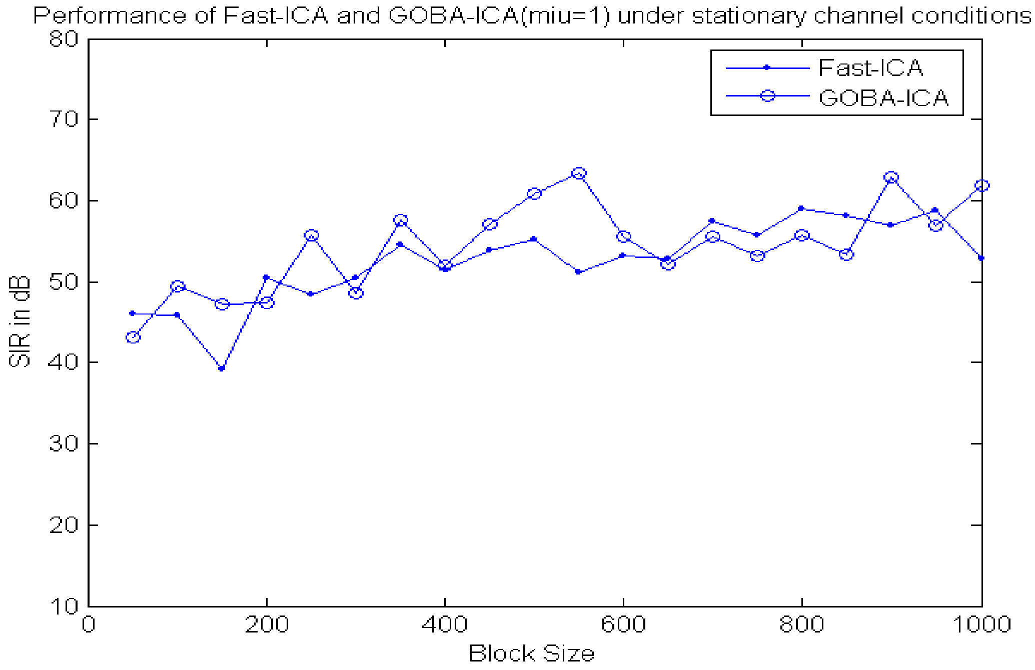

Figure 21. Performance of GOBA-ICA and Fast-ICA in stationary channels

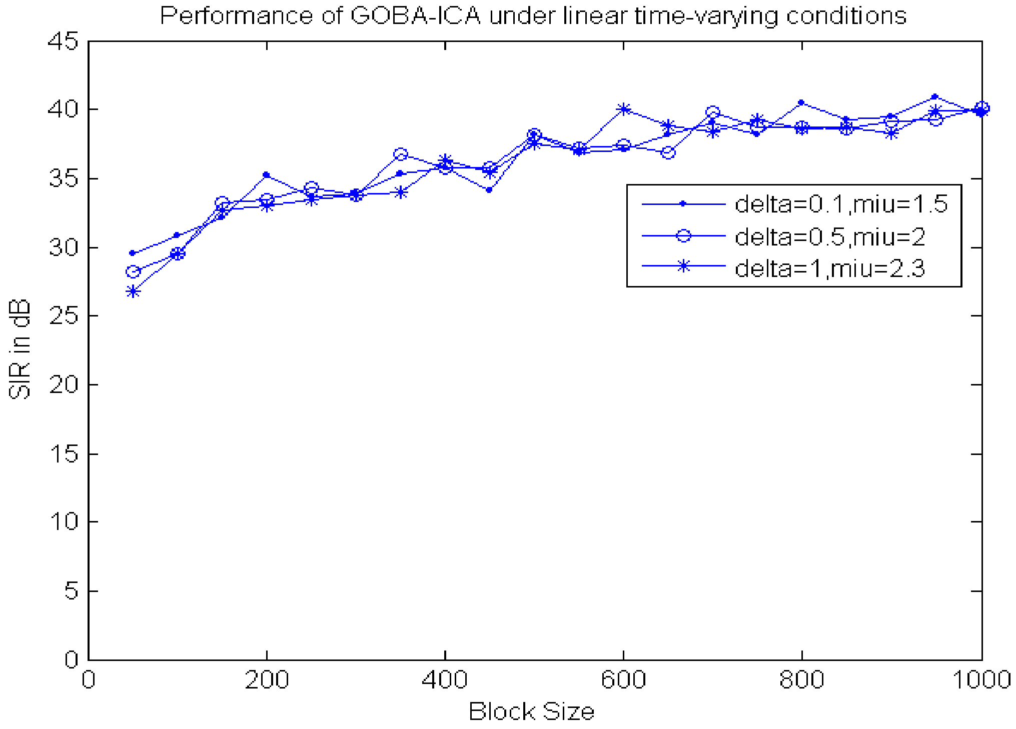

Figure 22. Performance of GOBA-ICA with linear time variation

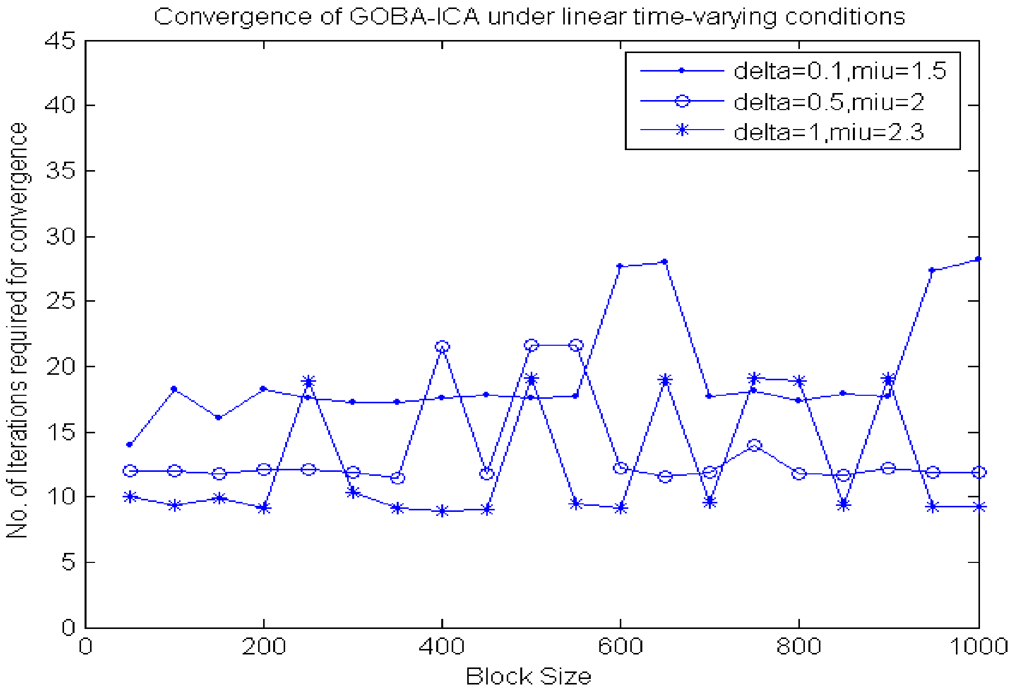

Figure 23. Convergence of GOBA-ICA with linear time variation

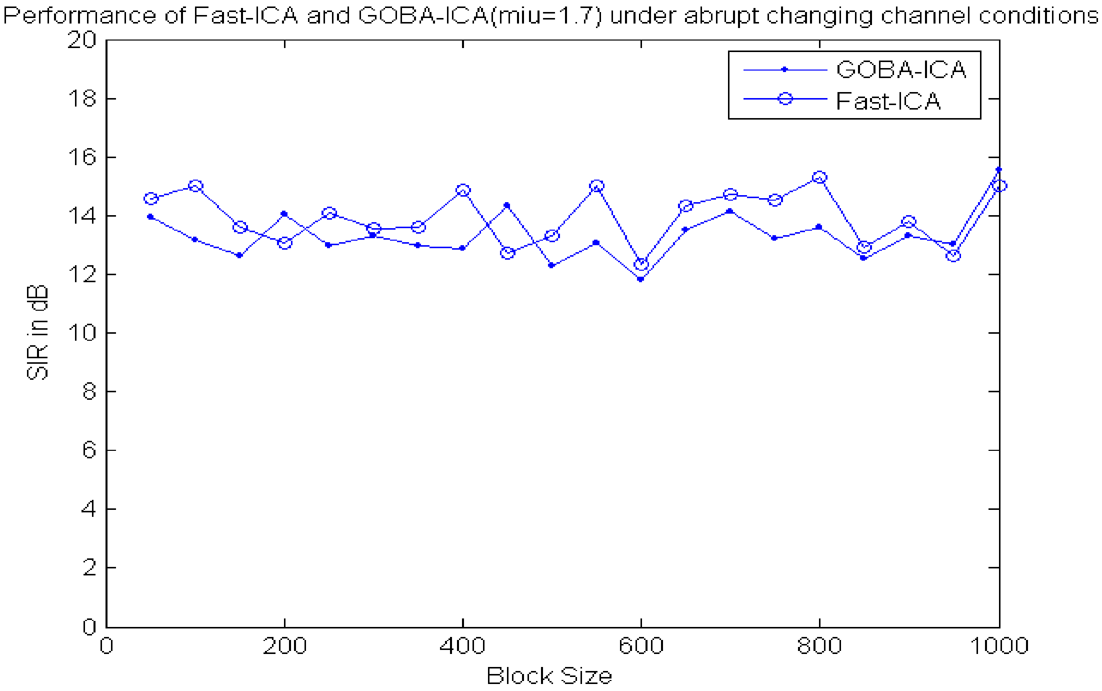

Figure 24. Performance of GOBA-ICA with abrupt time variation

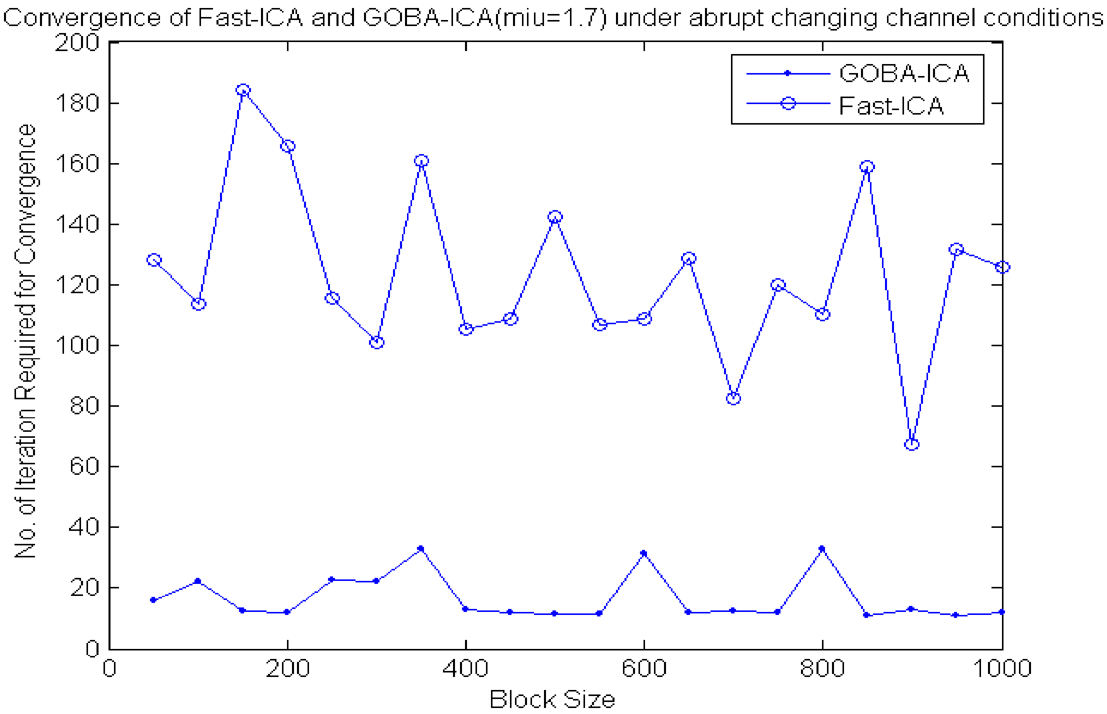

Figure 25. Convergence of GOBA-ICA with abrupt time variation

As a comment about the application of OBAI-ICA, it is worth mentioning that SIR used above is not the only index of performance. As a part of our experiment, the residue error of ICA separation is also studied. The residue error is defined as the difference between the kurtosis value of the extracted component and that of the original signal source. Thus, it reflects how the extracted component deviates from the true value. In our simulations, it is found that the residue error increases with the frame length. Therefore, though OBAI-ICA always converges very fast regardless of the choice of frame length, it is desirable to use smaller frame lengths. This is understandable, since more compromise is made when ICA tries to converge to a single demixing matrix for longer frames.

Conclusions

In this chapter, the performance of OBAI-ICA is studied via computer simulations and compared with Fast-ICA, under various time-varying conditions. OBAI tailors the learning rate for each coefficient in the separation vector and updates those rates at each block iteration. Thus, OBAI-ICA is much superior to Fast-ICA in terms of separation performance and convergence speed in highly dynamic environment. Two versions of OBAI-ICA are both simulated. OBAI-ICA1 is preferable for abrupt time-variation, while OBAI-ICA2 is desirable in continuous linear time varying conditions. If both types of time variation are present, OBAI-ICA1 needs to be used.

For the case of abrupt time varying channels, as encountered by mobile users experiencing handover between two service towers, a binary search technique is proposed to prevent performance degradation for the block in which the abrupt change occurs. Simulation results verify the effectiveness of the technique.

Also, the GOBA-ICA algorithm is simulated with the same conditions. It yields equivalent performance as OBAI-ICA.

CHAPTER SIX: PRACTICAL IMPLEMENTATION ISSUES FOR WIRELESS RECEIVERS EMPLOYING ICA

In Chapters four and five, wireless receivers employing ICA for interference suppression are introduced. Simulation results show that the technique can substantially improve the receiver's performance without additional cost of spectrum.

This chapter deals with practical implementation issues of such receivers, i.e., the effect of finite arithmetic and the possibility of reducing the front-end hardware complexity. In addition, it is illustrated that the proposed algorithm is robust to additive thermal noise, but it needs to be disabled in the case of large SIR, i.e., the desired signal is much stronger than the interference.

Effect of Finite Arithmetic

As mentioned in Chapter one, software radio requires the implementation of most radio functionalities in digital hardware.

In general, DSP data representation can be categorized into two types. The first type, floating-point data, has a larger dynamic range and higher precision of processing. However, the second type, fixed-point data, is of interest because of two reasons [38]. First, its required hardware is less costly, especially in massive production products. Second, the logic circuits of fixed-point hardware have reduced complexity. This leads to smaller chip size and less power consumption comparing with floating-point hardware.

The binary fixed-point numbers are represented in three different ways: Sign/magnitude, One's complement, or Two's complement [39]. The most commonly used representation is Two's complement. The word length is usually in the form of (WL, FP), where *WL* stands for *Word Length*, and *FP* stands for *Fraction Part*. The word length often takes values such as 8, 16, or 32, etc.

Now the simulation results for the fixed-point implementation are presented. QPSK receivers as introduced in Chapter four are used, and the performance measure SIRR as defined in (4.17) is adopted. Fast-ICA is simulated with finite arithmetic.

Extensive simulations are performed for different word lengths. For illustration, performance of the Finite-Precision ICA (FPICA) algorithm for finite word lengths of (32, 26) and (16, 10) are given. The results are compared with those for ICA implemented using double precision in Matlab.

In addition, the FPICA with mixed word lengths is also simulated, where the word length of the signal observations is (16, 10), and the word length adopted within the DSP hardware is (32, 26) [40]. The reason is that, in order to reduce the hardware complexity and cost, it is highly desirable to reduce the requirement for the number of bits in the A/D converters, and consequently the word length for the signal observations.

Ten simulations are performed for each frame length, and the averages of SIRR are plotted in Fig. 26. It is seen that, FPICA (32, 26) and FPICA with mixed word lengths are able to achieve equivalent average SIRR as floating-point ICA, and there is performance degradation when FPICA (16, 10) is employed. This can be easily explained, since as the word length becomes larger, the quantization noise is lowered. However, as the word length increases, the hardware cost also becomes higher.

It is useful to note that, the performance of FPICA with mixed word lengths is as good as FPICA (32, 26). This implies that the requirement for A/D converters can be relaxed without performance degradation.

Referring to Fig. 26, it is worth mentioning that, when the frame length is less than 250 symbols, there is no significant performance difference between the four implementations. This is because the independence between the source signals for a short frame length is not well established. As a result, the effect of finite word length is not the only dominant source of performance degradation. This leads to an important conclusion: if small frame length is used, there is no need to employ large word length.

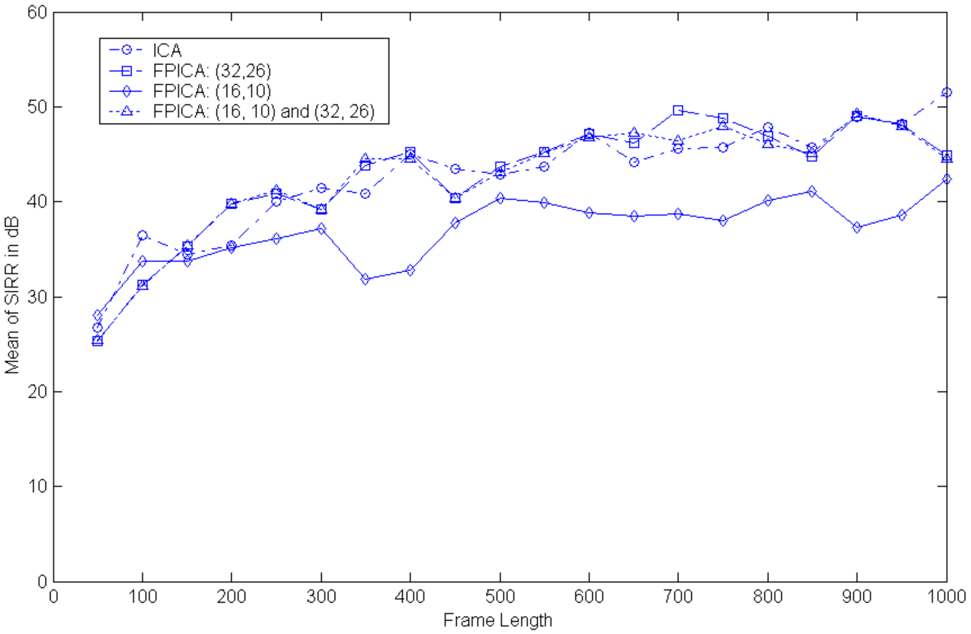

Figure 26. Average SIRR of ICA and Finite-Precision ICA (FPICA) algorithms

Figure 27 shows the convergence speed in terms of the number of iterations required for convergence. It is clear that FPICA (16, 10) converges faster than the other three cases. This suggests that smaller word length has faster convergence, which means it is preferable if the resulting performance is acceptable.

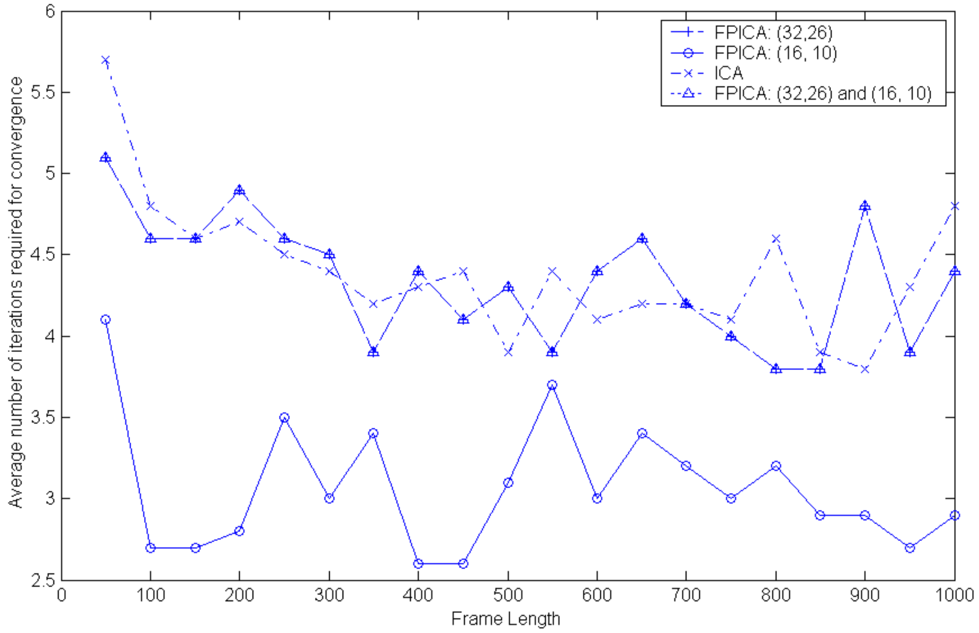

Figure 27. Convergence Speed of ICA and FPICA algorithms

In conclusion, Finite-Precision ICA algorithm with word lengths (16, 10) for the signal observations and (32, 26) in DSP are able to achieve equivalent performance as floating-point ICA. Meanwhile, it is also shown that shorter word length such as (16, 10) for DSP is preferable in some cases because it has lower hardware cost and faster convergence without degrading the performance when small N_f is adopted.

Reduced Hardware Complexity for BPSK receivers with One Interferer

An obvious drawback of the proposed wireless receiver technique is that one antenna is required for an additional interferer. This increases the receiver's front-end hardware complexity. Therefore, it is highly desirable to have reduced number of antenna and/or downconversion paths. In this section, it is shown that for BPSK receivers, it is possible to use only one antenna to reject one interferer without any performance degradation. Thus, the complexity of the front-end is reduced by half.

Figure 28 is a single branch BPSK receiver. It is similar to the receiver structure in Chapter five, except that only one antenna and downconversion path is used. ω_0 and ω_I denote the frequencies of the first and second local oscillators. α is the phase difference between the received signal and the first local oscillator.

Akin to the derivation in Chapter five, the baseband signal observation can be expressed as:

$$X(n) = as(n) + bi(n) \tag{6.1}$$

Where $a = Re\{f_s e^{-j\alpha}\}$, and $b = Re\{f_i e^{-j(\omega_0-\omega_I)\tau}\ e^{-j\alpha}\}$. As before, f_s and f_i are the fading coefficients for the desired and interfering signals, respectively. τ is the timing delay of the interferer with respect to the symbol timing reference of the desired signal.

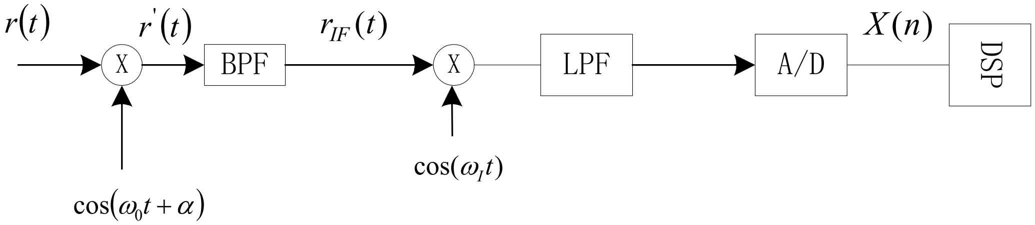

Figure 28. **A single branch BPSK receiver**

X(n), given by (6.1), is a linear combination of two independent signals, *s(n)* and *i(n)*. In principle, two independent observations are needed to recover the two signals. The problem of fewer antennas than signals is generally known as *ICA with overcompletes bases*.

Since one antenna is employed, only one observation *X(n)* is available. A second linearly independent observation *Y(n)* is artificially generated from *X(n)*. Then two signals *s(n)* and *i(n)* can be separated based on *X(n)* and *Y(n)*. The requirement is that, *Y(n)* should be linearly independent of *X(n)*.

Actually, *Y(n)* can be obtained by raising *X(n)* to the p^{th} power, where p is an odd number and $p \geq 3$. The reason is described as follows, assuming $p = 3$.

Since *s(n)* and *i(n)* are both binary numbers taking values of either +1 or -1, $s^2(n) = 1$ and $i^2(n) = 1$. Thus,

$$Y(n) = [as(n) + bi(n)]^3 = a^3s(n) + b^3i(n) + 3a^2bi(n) + 3ab^2s(n)$$

$$= (a^3 + 3ab^2)s(n) + (b^3 + 3a^2b)i(n) \tag{6.2}$$

When *Y(n)* is appended to *X(n)*, the signal observation becomes:

$$\mathbf{X(n)}=\begin{bmatrix} X(n) \\ Y(n) \end{bmatrix}=\begin{bmatrix} \mathrm{a} & b \\ a^3+3ab^2 & b^3+3a^2b \end{bmatrix}\begin{bmatrix} s(n) \\ i(n) \end{bmatrix}=\mathbf{AS(n)} \tag{6.3}$$

If the determinant of the mixing matrix **A** is nonzero, **S(n)** can be recovered from the observation **X(n)**.

$$\det(\mathbf{A}) = ab(b^2 +3a^2) – ab(a^2 +3b^2) = 2ab(a^2 – b^2) \tag{6.4}$$

Therefore, as long as $2ab(a^2 – b^2) \neq 0$, *s(n)* and *i(n)* can be successfully recovered.

The above argument can be easily extended to the case when $p > 3$. In fact, *Y(n)* can also be obtained by applying an odd function to *X(n)*.

Computer simulations are carried out to examine the performance of the proposed technique. Fast-ICA is adopted to separate the desired signal from the interferer. The simulation set up is identical to Chapter five. The resulting average SIR is plotted in Fig. 29. The performance of quadrature or dual-antenna receivers, where two observations are directly available, is also plotted for comparison.

It is clearly seen from Fig. 29 that the proposed technique achieves equivalent performance as quadrature or dual-antenna receivers. This means that the hardware complexity of the receiver can be substantially simplified without affecting the performance.

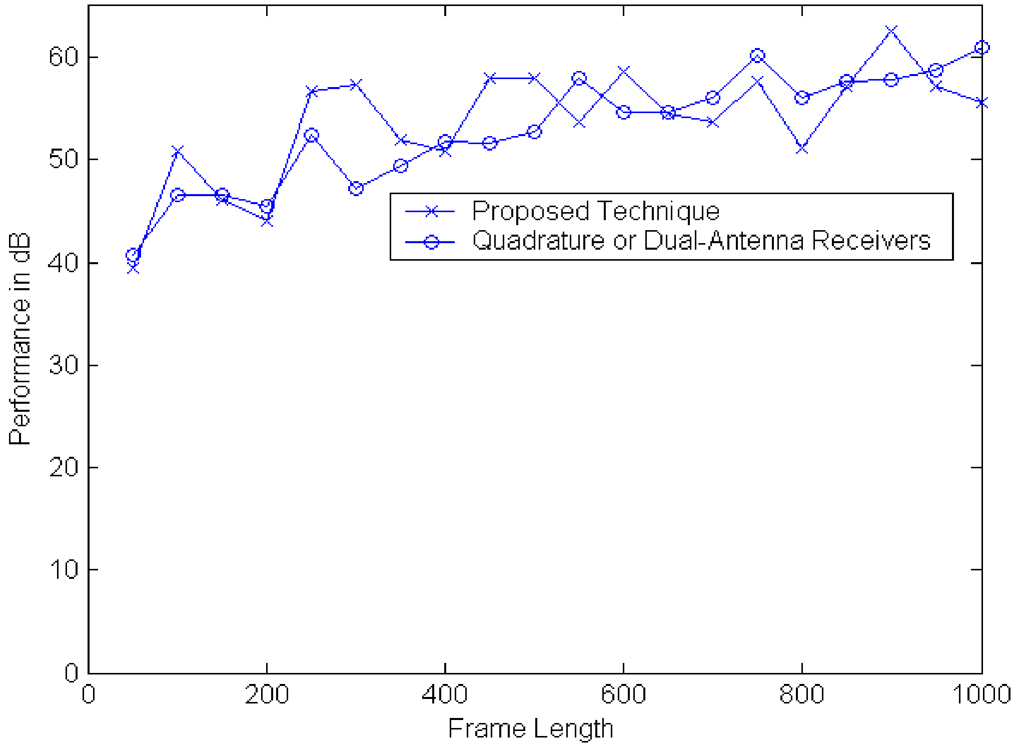

Figure 29. Performance comparison of the proposed technique and quadrature or dual-antenna receivers

Effect of Thermal Noise

In Chapter four, kurtosis is chosen as the measure of independence because of its robustness against thermal noise. Also, it is mentioned that the performance of ICA processing is not affected by the relative strength of the desired signal and the interference, but rather how well the statistical properties of the signals are established.

Here, these properties are verified by simulation results.

Figure 30 plots the average SIR after ICA processing versus the input SIR in the case of one interferer. The frame length is fixed at 200 symbols. In addition, the processing gain, which is the improvement in SIR by ICA processing, is also plotted. Moreover, an additive Gaussian thermal noise of 20 dB below the desired signal level is added to the signal observations to study its effect.

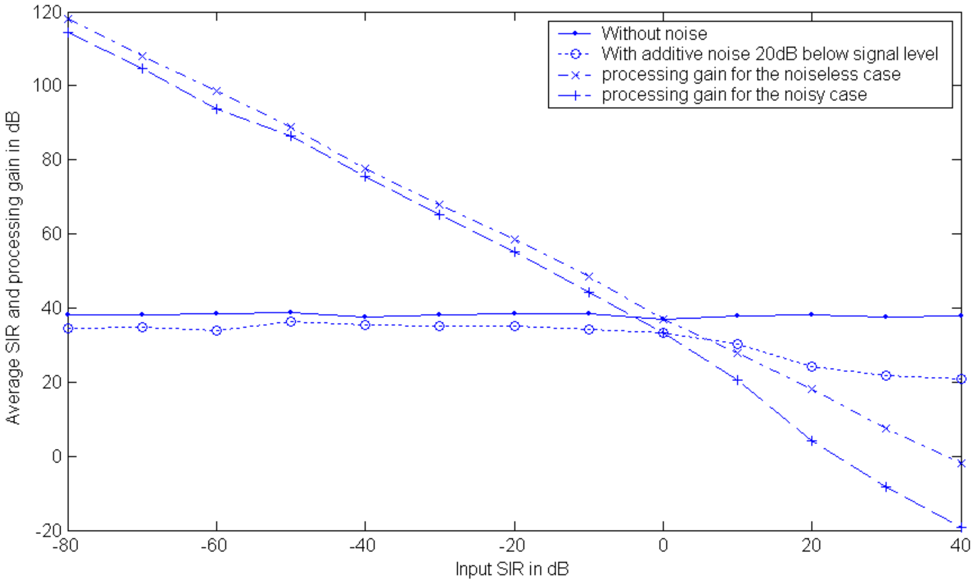

Figure 30. **Average SIR and processing gain for varied input SIR's**

As seen, the performance is not significantly affected by the thermal noise, and it is robust against input SIR's.

For the scheme proposed in the previous section, the BPSK receiver with single downconversion path is also simulated with an additive Gaussian thermal noise of 20 dB below the desired signal level added. The average SIR achieved is compared with the performance obtained by quadrature or dual-antenna receivers, where two observations are directly available. Figure 31 plots the results.

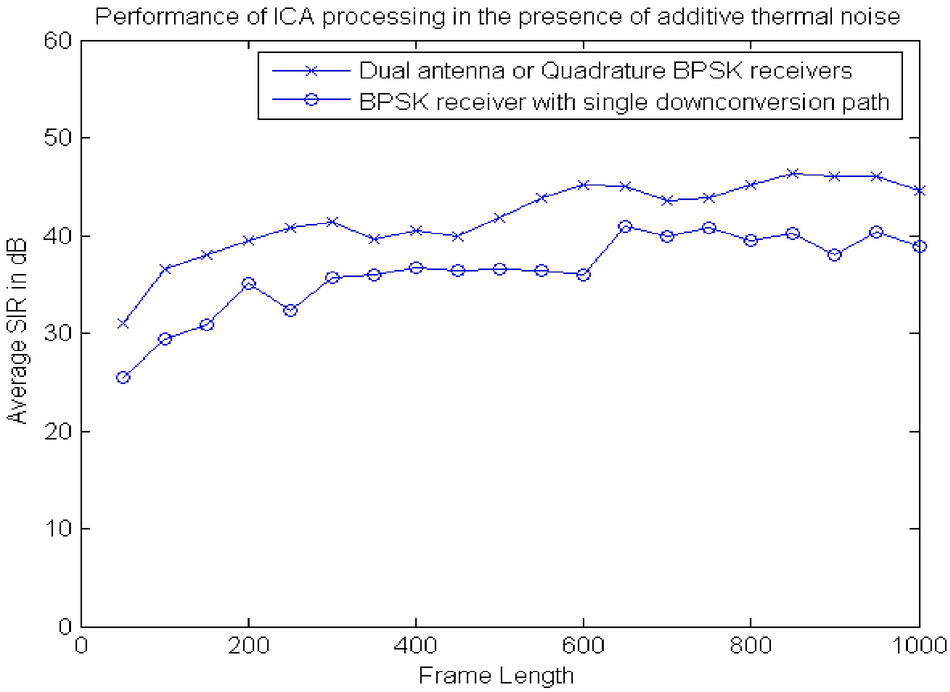

Figure 31. The performance of ICA processing in the presence of thermal noise

From Fig. 31, it is obvious that the scheme with reduced hardware complexity is less robust against thermal noise than quadrature or dual-antenna receivers. This is due to the fact that the noise part of the artificially generated observation is correlated with the noise of the available observation. Also, the cubing operation introduces multiplicative noises in the second signal.

Also, finite arithmetic may have a larger effect to this technique because of propagation of quantization error when generating the second observation. Figure 32 illustrates this phenomenon.

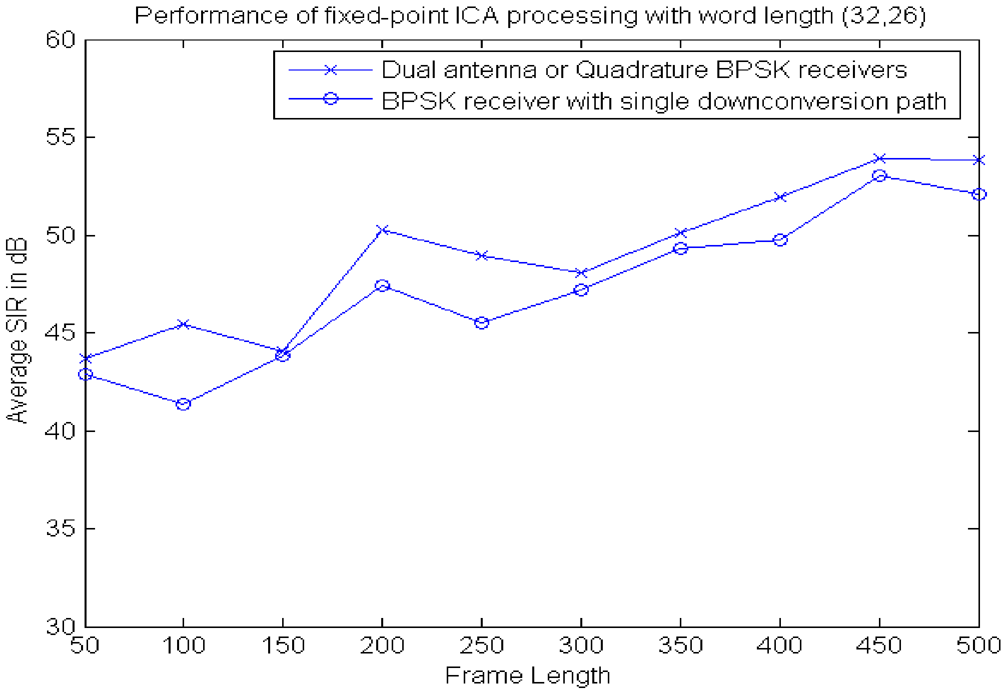

Figure 32. **The performance of fixed-point ICA for BPSK receivers**

<u>The Case of Large Input SIR</u>

From Fig. 30, it is also observed that in the case of small input SIR, the processing gain is significant. However, when the input SIR exceeds 25dB, the processing gain drops below zero dB. This implies that the algorithm actually deteriorates SIR.

A natural solution to this problem is to disable the algorithm in this scenario. Fortunately, additional interference suppression is not needed in this case, because typically the desired signal is already strong enough. Of course, a power estimation mechanism should be introduced in the receiver to estimate the input SIR.

Conclusions and Discussions

In this chapter, two implementations issues are discussed for the general interference suppression scheme proposed in Chapter Four.

The first one is the effect of finite arithmetic. It is shown that better performance can be achieved as the word length is increased, which indicates that the quantization error is directly associated with the word length. A useful result is that each signal can be quantized in a different way to get an optimum result with reduced hardware complexity.

In reality, cost, power dissipation of hardware and performance of the systems are highly dependent on the bit resolution [41]. An additional important consideration is that, the word lengths should be carefully selected to preserve the algorithm stability [42]. For this purpose, the automatic word length determination methods were recently proposed [43]. For example, finding the minimum word length by computing dissimilarities between fixed-point and floating-point results can help to achieve optimal word length [44]. More detailed studies in the area of finding the optimum word length automatically can be found in [45-46].

Second, in the special case of BPSK receivers with one interferer, it is shown that one antenna and single downconversion path is enough to achieve interference suppression. A second observation can be artificially generated from the available observation that is linearly independent. Without performance degradation in the noiseless case, the technique achieves significantly hardware simplification with negligible computational cost.

Finally, it is illustrated that the proposed algorithm is indeed robust to thermal noise, but it needs to be disabled in the case of large input SIR. This implies power estimation is needed at the receiver side.

CHAPTER SEVEN: CONCLUSIONS AND FUTURE WORK

Our research involves the application of Independent Component Analysis for interference suppression in diversity wireless receivers. In particular, in the case of highly dynamic channel conditions, a novel ICA algorithm, OBAI-ICA, is proposed for fast adaptation, because the classic fixed-point ICA fails to track fast time variation. This is of particular interest to mobile communications where the user is moving at a high speed or experiencing handover between two service towers. Important implementation issues, such as the effect of finite arithmetic and thermal noise, are also addressed.

This chapter summarizes the contributions of the research and concludes with suggestions for future research.

Importance of the Contributions

The methodology adopted in our research work is consistent with current trend in the area of wireless transceiver design. In Chapter one, it is shown that the proposed scheme is helpful towards the realization of *Software Defined Radio*, and space diversity at the receiver side is compatible with space-time communications, i.e., MIMO systems.

As the frequency spectrum is becoming more and more expensive, interference suppression for frequency-reuse systems is becoming critical. Our proposed method utilizes purely statistical properties to provide interference rejection. Thus, no additional spectrum or hardware is required. This is attractive because it may lead to more intensive frequency reuse and more integrated systems. Also, it provides additional interference suppression capability that might help newly emerging schemes such as CDMA and UWB.

It is worth mentioning that the proposed technique does not distinguish between image and co-channel interference. Thus, the scheme not only increases the degree of frequency reuse, but also eliminates the need for a highly selective anti-aliasing bandpass filter at the RF stage, which could be extremely costly.

The idea of avoiding the use of an analog image rejection filter has attracted huge amount of research efforts [7, 34-37, 47-51]. Ideally, quadrature downconversion can solve the image problem, but the mismatch between the Inphase and Quadrature phase local oscillator signals always necessitates additional analog circuitry or digital algorithm for compensation. Moreover, a lot of technique requires a test tone and separate training stage for calibration, which impairs their efficiency.

On the other hand, our method does not need to deal with the I/Q mismatch problem and it does not require a test stage.

Although various ICA algorithms have been reported [52-70], the novel ICA algorithm proposed in this dissertation, OBAI-ICA, is especially useful in the case of highly dynamic environment. In particular, it is shown that for time-varying systems, OBAI-ICA is superior to Fast-ICA, which has been dominant in various applications. OBAI-ICA is based on similar techniques originally designed for updating FIR filter coefficients [71-78]. OBAI-ICA has the capability of tracking rapid time variation in the mixing matrix coefficients. This makes OBAI-ICA an attractive alternative in the cases where real-time adaptation is important.

Key Contributions

The research presented contains the following contributions:

- The gradient-based OBAI-ICA is derived in Chapter three based on block adaptation and Taylor series expansion. From OBAI-ICA, a family of gradient-based algorithm is developed. OBAI-ICA can be simplified to obtain online gradient ICA already existing in literature. Also, the computational complexity and the estimation of matrix inversion are addressed [16, 80].
- Chapter four presents the proposed diversity receiver with ICA-based interference rejection. An efficient way of solving the inherent order ambiguity is introduced. Simulation results confirm the advantages of the scheme, and a performance comparison is provided between Fast-ICA and EASI. The results are reported in [11-15, 51, 81]. Further, in the case of rapid changing channel conditions, a data-reusing technique is adopted for better performance [82].
- Chapter five investigates the application of OBAI-ICA to the proposed scheme in Chapter four in a highly dynamic environment. It is demonstrated that OBAI-ICA outperforms Fast-ICA in both linear and abrupt time varying channels conditions, especially in terms of convergence speed. Especially for abrupt time variation, a binary search technique is proposed to identify the location of the change within the block to prevent performance degradation for that specific block. The results are reported in [16, 79, 80].
- Chapter six discusses several implementation issues of the proposed scheme. First, the effect of finite arithmetic is studied. It is shown that the hardware complexity can be significantly reduced without performance degradation by carefully selecting the word lengths for each quantity. The results are reported in [17, 83]. Second, it is shown that it is possible to use only one antenna and single

downconversion path to rejection one interferer in BPSK receivers. With negligible increase in computation, the hardware complexity is reduced by half without any performance degradation in the noiseless case. The results are reported in [18, 84, 85]. Third, it is illustrated that our proposed scheme is robust to thermal noise, but it should be disabled in the case of large input SIR, in which case additional interference suppression is not necessary.

Areas of Future Research

The presented research work can be extended in many directions.

The most important extension will be reducing the number of antennas and/or downconversion paths. Since one additional antenna is required for one interferer, the proposed scheme is more feasible to be adopted at base station. However, on the mobile side, it is not practical in the case of many interferers, because the cost might be too high, and installing multiple antenna elements to mobile units is difficult because of their limited size. Also, sometimes it is impossible to know before ahead the exact number of interferers, and the number may not be fixed. Therefore, it is highly desirable to use fewer antennas than the number of signal components. Chapter six illustrates that, in the case of BPSK with one interferer, it is possible to use only one antenna to achieve interference suppression. In general, the problem is known as *Underdetermined ICA* [87, 88].

When some interferers are absent, the number of antennas is effectively larger than the number of independent components. We found that the performance in this scenario degrades substantially. The problem is commonly known as *Overdetermined*

ICA. Some techniques have been proposed in the literature to deal with this situation [89, 90].

It is assumed in this document that the fading channels are frequency-flat for the signal band of interest, and no *Inter-Symbol Interference* (ISI) is present. In reality, frequency-selective fading is often present for wide-band signals. Thus, convolutive mixtures are obtained at the receiver and blind deconvolution is needed. One possible solution is frequency domain ICA, which performs ICA processing to the Fourier transformed data. This operation has an additional advantage of saving computations, because fast transform method is applicable in this scenario [36].

It is mentioned in Chapter six that the algorithm should be disabled in the case of large input SIR. Under certain circumstances, there is no prior knowledge regarding the input SIR. Therefore, a blind estimation of input SIR will be very useful.

Also, in order to increase the data rate, the adoption of higher-order QAM signals has to be studied. Because of the relatively small distance between adjacent symbols in the signal constellation, additional performance enhancement is needed if the technique presented in this dissertation does not provide enough interference suppression.

APPENDIX

MATLAB CODE FOR COMPUTER SIMULATIONS

1. Fast-ICA for real valued sources: ficai.m

```
% The function accepts observation matrix r and accuracy epsilon, and outputs source
% signals and demixing matrix, as well as estimation of mixing matrix
function [Shat, W, A, all_iter] = ficai (r, epsilon);

% Determine the dimensionality of the data
[m,n] = size(r);

% mean center the data
M = mean(r');
for i = 1:m
    r(i,:) = r(i,:)-M(i);
end

% Whiten the data by eigenvalue decomposition
Rrr = zeros(m,m);
for i = 1:n
    Rrr = Rrr+1/n*r(:,i)*r(:,i)';
end

[E,D] = eig(Rrr);
V = diag(diag(D).^(-1/2))*E';
```

```
x = V*r;

% Extract the components by fixed-point ICA, measure of independence is kurtosis
W = zeros(m,m);
all_iter = 0;                                   % Total number of iteration
for p = 1:m
        W(p,:) = 0.01*randn(1,m);               % Initialization
        W(p,:) = W(p,:)/norm(W(p,:));           % Normalization
        delta = 1;                              % For convergence
        iter = 0;

        % Iterate for one row of demixing matrix
        while delta > epsilon & iter < 1000 ;
                iter = iter+1;
                wnew = zeros(m,1);
                % Adapt according to fixed-point algorithm
                for i = 1:n
                wnew = wnew + 1/n*(W(p,:)*x(:,i))^3*x(:,i);
                end
                wnew = wnew-3*W(p,:)';
                % Orthogonalization with respect to already extracted rows
                if p > 1
                        for j = 1:p-1
                wnew = wnew-wnew'*W(j,:)'*W(j,:)';
                end
            end
            wnew = wnew/norm(wnew);  % Renormalization
            delta = abs(abs(W(p,:)*wnew)-1);     % Convergence criterion
            W(p,:) = wnew';
        end
        all_iter = all_iter+iter;
end


% Add back the mean
for i = 1:m
   r(i,:) = r(i,:)+M(i);
end


W = W*V;      % Demixing matrix
Shat = W*r;   % Estimation of the sources
A = inv(W);   % Estimation of the mixing matrix
```

2. Fast-ICA for complex-valued sources: ficaic.m

```
% The file is similar to ficai.m, excepted that the adaptation equation is complex-valued
function [Shat, Ahat, What, all_iter] = ficaic(r,epsilon);

[m,n] = size(r);
M = mean(r,2);
for i = 1:m
   r(i,:) = r(i,:)-M(i);
end

Rrr = 1/n*r*r';
[E,D] = eig(Rrr);
V = diag(diag(D).^(-1/2))*E';
x = V*r;

W = zeros(m,m);
all_iter = 0;
for p = 1:m
   W(:,p) = randn(m,1);
   W(:,p) = W(:,p)/norm(W(:,p));
   delta = 1;
   iter = 0;
```

```
        while delta > epsilon & iter < 1000 ;
            iter = iter+1;
            wnew = zeros(m,1);
            % Update the current row with complex-valued fixed-point ICA (kurtosis based)
            for i = 1:n
                wnew = wnew + 2/n*(x(:,i)*conj(W(:,p)'*x(:,i))-2*W(:,p))*abs(W(:,p)'*x(:,i))^2;
            end
            if p > 1
                for j = 1:p-1
                    wnew = wnew-W(:,j)*W(:,j)'*wnew;
                end
            end
            wnew = wnew/norm(wnew);
            delta = abs(abs(W(:,p)'*wnew)-1);
            W(:,p) = wnew;
        end
        all_iter=all_iter+iter;
    end


    for i = 1:m
        r(i,:) = r(i,:)+M(i);
    end
```

```
What = W'*V;

Shat = What*r;

Ahat = inv(What');
```

3. OBAI-ICA algorithm: ficai_obai.m

```
% The file is similar to ficai.m, except that the adaptation rule for each row is OBAI-ICA
% The additional input parameter miu can be adjusted according to the time-variation
function [Shat, W, A, all_iter] = ficai_obai (r,epsilon,miu);

[m,n] = size(r);
M = mean(r');
for i = 1:m
    r(i,:) = r(i,:)-M(i);
end

Rrr = zeros(m,m);
for i = 1:n
    Rrr = Rrr+1/n*r(:,i)*r(:,i)';
end
[E,D] = eig(Rrr);
V = diag(diag(D).^(-1/2))*E';
x = V*r;
W = zeros(m,m);
all_iter = 0;
for p = 1:m
   W(:,p) = 0.01*randn(m,1);
```

```
W(:,p) = W(:,p)/norm(W(:,p));
 delta = 1;
 iter = 0;
 while delta > epsilon & iter < 1000;
    iter = iter+1;
    G = x';                 % Define observation matrix G for OBAI-ICA
    temp=[];
    kurt=[];
    for j=1:n
        temp=[temp (W(:,p)'*x(:,j))^3];         % For matrix C in OBAI-ICA
        kurt=[kurt (W(:,p)'*x(:,j))^4-3];        % Kurtosis vector
    end
    C = diag(temp);
    kurt = kurt';
    q=G'*C*kurt;      % Define q vector for OBAI-ICA
    R=G'*C*C*G;    % Define matrix R for OBAI-ICA
    invR=inv(R);  % For high order systems, this line should be: invR=diag(1./diag(R));
    wnew = zeros(m,1);
    wnew = W(:,p) - 0.25*miu*invR*q;      % OBAI-ICA adaptation
    if p > 1
        for j = 1:p-1
             wnew = wnew-W(:,j)*W(:,j)'*wnew;
         end
        end
        wnew = wnew/norm(wnew);
        delta = abs(abs(W(:,p)'*wnew)-1);
        W(:,p) = wnew;
    end
    all_iter=all_iter+iter;
end

for i = 1:m
    r(i,:) = r(i,:)+M(i);
end

W = W'*V;
Shat = W*r;
A = inv(W);
```

4. GOBA algorithm: ficai_goba.m

```
% The file is similar to ficai_obai, but the adaptation rule for each row is GOBA-ICA
function [Shat, W, A, all_iter] = ficai1(r,epsilon,miu);

[m,n] = size(r);
M = mean(r');
for i = 1:m
   r(i,:) = r(i,:)-M(i);
end

Rrr = zeros(m,m);
for i = 1:n
   Rrr = Rrr+1/n*r(:,i)*r(:,i)';
end
[E,D] = eig(Rrr);
V = diag(diag(D).^(-1/2))*E';
x = V*r;

W = zeros(m,m);
all_iter = 0;
for p = 1:m
  W(:,p) = 0.01*randn(m,1);
```

```
W(:,p) = W(:,p)/norm(W(:,p));
delta = 1;
 iter = 0;
 while delta > epsilon & iter < 1000;
    iter = iter+1;
    G = x';
    temp=[];
    kurt=[];
    for j=1:n
       temp=[temp (W(:,p)'*x(:,j))^3];
       kurt=[kurt (W(:,p)'*x(:,j))^4-3];
    end
    kurt = kurt';
    C = diag(temp);
    R=4*C*G;                   % Define R matrix for GOBA-ICA
    R_star=inv(R'*R)*R';       % Define pseudo inverse of R for GOBA-ICA
    twos=2*ones(n,1);
    wnew = zeros(m,1);
    wnew = W(:,p)-miu*R_star*(kurt+twos);        % Adaptation for GOBA-ICA
    if p > 1
       for j = 1:p-1
          wnew = wnew-W(:,j)*W(:,j)'*wnew;
       end
```

```
        end
        wnew = wnew/norm(wnew);
        delta = abs(abs(W(:,p)'*wnew)-1);
        W(:,p) = wnew;
    end
    all_iter = all_iter+iter;
end

for i = 1:m
    r(i,:) = r(i,:)+M(i);
end

W = W'*V;
Shat = W*r;
A = inv(W);
```

5. Generation of BPSK source signals, processing with ICA and calculation of SIR: main.m

```
% Generate channel parameters, assuming Rayleigh fading and one interferer
as1=raylrnd(1/sqrt(2),[1 1]);   % Amplitude of the response to the desired signal
ai1=raylrnd(1/sqrt(2),[1 1]);   % Amplitude of the response to the interfering signal
fais1=2*pi*rand(1);
faii1=2*pi*rand(1);
fs1=as1*exp(j*fais1);
fi1=ai1*exp(j*faii1);
alpha=2*pi*rand(1);
as1=real(fs1*exp(-j*alpha));  % The first channel response to the desired signal
ai1=real(fi1*exp(-j*alpha));   % The first channel response to the interfering signal

as2=raylrnd(1/sqrt(2),[1 1]);
ai2=raylrnd(1/sqrt(2),[1 1]);
fais2=2*pi*rand(1);
faii2=2*pi*rand(1);
fs2=as2*exp(j*fais2);
fi2=ai2*exp(j*faii2);
alpha=2*pi*rand(1);
as2=real(fs2*exp(-j*alpha));  % The second channel response to the desired signal
ai2=real(fi2*exp(-j*alpha));   % The second channel response to the interfering signal
```

```
% Generate the BPSK source signals of block size N
de = floor(rand(N,1)*2);        % The desired signal
in = floor(rand(N,1)*2);        % The interferer
de = 2*de-1;
in = 2*in-1;

% Define mixing matrix, source signal matrix and observation matrix
Am = [as1 as2; ai1 ai2];
D = [de'; in'];
y = Am*D;

% If desired, add thermal noise
yn = y +randn(2,N)*0.1;         % Add Gaussian noise of 20 dB below the signal level

% Perform the ICA separation
[Shat, What, A, all_iter] = ficai(y,1e-6);        % use yn for the noisy case

% Decide the sign and the order of the extracted signals
S = zeros(size(Shat));
for i = 1:2
   d1 = norm(Shat(i,:)-de');
   d2 = norm(Shat(i,:)+de');
   d3 = norm(Shat(i,:)-in');
```

```
    d4 = norm(Shat(i,:)+in');

    d = [d1 d2 d3 d4];

    [m,I] = min(d);

    switch I

    case 1

        S(1,:) = Shat(i,:);

    case 2

        S(1,:) = -Shat(i,:);

    case 3

        S(2,:) = Shat(i,:);

    case 4

        S(2,:) = -Shat(i,:);

    end

end


% Determine the SIR

sum=0;

for k=1:N

    sum=sum+de(k)^2/(de(k)-S(1,k))^2;

end

sir=1/N*sum;

sir=10*log10(sir);
```

6. Generation of QPSK source signals, processing with ICA and calculation of SIR: main1.m

```
% This is similar to main.m, except that the processing is complex-valued
% Generate channel parameters
as1=raylrnd(1/sqrt(2),[1 1]);
as2=raylrnd(1/sqrt(2),[1 1]);
ac1=raylrnd(1/sqrt(2),[1 1]);
ac2=raylrnd(1/sqrt(2),[1 1]);

fais1=2*pi*rand(1);
fais2=2*pi*rand(1);
faic1=2*pi*rand(1);
faic2=2*pi*rand(1);

fs1=as1*exp(j*fais1);
fs2=as2*exp(j*fais2);
fc1=ac1*exp(j*faic1);
fc2=ac2*exp(j*faic2);

% Generate QPSK source signals
sr=floor(rand(N,1)*2);
sr=2*sr-1;
```

```
si=floor(rand(N,1)*2);
si=2*si-1;
de=sr+j*si;
de=de/sqrt(2);                    % Desired signal

cr=floor(rand(N,1)*2);
cr=2*cr-1;
ci=floor(rand(N,1)*2);
ci=2*ci-1;
in=cr+j*ci;
in=in/sqrt(2);                    % Interferer

Am=[fs1 fc1; fs2 fc2];
D=[de'; in'];
y = Am*D;

% If desired, add thermal noise
yn = y +randn(2,N)*0.1;

% Perform the ICA separation
[Shat, Ahat, What, W, all_iter] = ficaic(y,1e-6);       % use yn in the noisy case

Ihat=abs(What*Am);                % The product of demixing matrix and mixing matrix
```

```
Shat1=Ihat*D;


% If higher order QAM signals, scale with the energy of the constallation
% Shat = Shat*2.2361;           % For 16-QAM


% Select the desired signal and its sign
S = zeros(size(Shat));
for i = 1:2
   d1 = norm(abs(Shat1(i,:)-de'));
   d2 = norm(abs(Shat1(i,:)+de'));
   d3 = norm(abs(Shat1(i,:)-in'));
   d4 = norm(abs(Shat1(i,:)+in'));
   d = [d1 d2 d3 d4];
   [m,I] = min(d);
   switch I
   case 1
      S(1,:) = Shat(i,:);
   case 2
      S(1,:) = -Shat(i,:);
   case 3
      S(2,:) = Shat(i,:);
   case 4
      S(2,:) = -Shat(i,:);
```

```
    end
end

% Determine the SIR
sum=0;
soriginal=de';
for k=1:N
    sum=sum+(abs(soriginal(k)))^2/abs((abs(soriginal(k)))^2-(abs(S(1,k)))^2);
end
sir=1/N*sum;
sir=10*log10(sir);

% The performance measure can also be defined by SIRR in terms of H=What*Am
%H = What*Am;
%SIRR =
%(abs(20*log10(abs(H(1,1))/abs(H(1,2))))+abs(20*log10(abs(H(2,1))/abs(H(2,2)))))/2;
```

7. ICA with overlapping blocks: main2.m

```
% QPSK source signals
close all;
sr=floor(rand(1000,1)*2);             % the total number of symbols is one thousand
sr=2*sr-1;
si=floor(rand(1000,1)*2);
si=2*si-1;
signal=sr+j*si;
signal=signal/sqrt(2);                % Desired signal

imager=floor(rand(1000,1)*2);
imager=2*imager-1;
imagei=floor(rand(1000,1)*2);
imagei=2*imagei-1;
image=imager-j*imagei;
image=image/sqrt(2);                  % Interfering signal

T=10000;
for t=1:1000
  A(:,:,t)=[1+j*exp(-t/T) 0.5;        % Exponentially varying channel parameters
            0.7 2-j*exp(t/(2*T))];
end
```

```
D=[signal';image'];

for t=1:1000
    y(:,:,t)=A(:,:,t)*D(:,t);
end

ym=y(:,:,1);
for t=2:1000
    ym=[ym y(:,:,t)];              % Signal observation matrix
end

Dhat=zeros([1 N]);
Destimate=zeros([1 1000]);    % Estimation of overlapping ICA
x=zeros([2 N]);
for t=1:(1000-N+1)              % shift one symbol at a time
  x=ym(:,t:t+N-1);              % The current block of observations
  % perform ICA processing for the current block
  [Dhat, all_iter]=ficaic(x,D(1,t:t+N-1),D(2,t:t+N-1),A(:,:,t),D(:,t:t+N-1),1e-6);
  Destimate(t)=Dhat(1);         % Take the first estimate as our estimation
end

sum=0;
```

```
for k=N:1000-N+1
    sum=sum+(abs(D(1,k)))^2/abs(((abs(D(1,k)))^2-abs(Destimate(k))^2));
end
sir=1/N*sum;
sir=10*log10(sir);
```

8. OBAI-ICA (or GOBA-ICA) for linear time-varying channels: main3.m

```
delta=0.1;        % delta determines the speed of channel’s time variation
for t=1:N
 A(:,:,t)=[1+delta*t 0.5;
              0.7 2+delta*t];
end

de = floor(rand(N,1)*2);
in = floor(rand(N,1)*2);
de = 2*de-1;    % The desired signal
in = 2*in-1;    % The interferer

D=[de'; in'];
for t=1:N
   y(:,t)=A(:,:,t)*D(:,t);           % Construct the observation matrix
end

% Perform the ICA separation, the additional scaling factor needs to be adjusted
% If GOBA-ICA is to be used, the function ficai_goba should be called instead
[Shat, What, all_iter] = ficai_obai (y,1e-6, 0.5);

% Solving the order ambiguity
```

```
S = zeros(size(Shat));
for i = 1:2
   d1 = norm(Shat(i,:)-de');
   d2 = norm(Shat(i,:)+de');
   d3 = norm(Shat(i,:)-in');
   d4 = norm(Shat(i,:)+in');
   d = [d1 d2 d3 d4];
   [m,I] = min(d);
   switch I
   case 1
      S(1,:) = Shat(i,:);
   case 2
      S(1,:) = -Shat(i,:);
   case 3
      S(2,:) = Shat(i,:);
   case 4
      S(2,:) = -Shat(i,:);
   end
end

sum=0;
for k=1:N
   sum=sum+de(k)^2/(de(k)-S(1,k))^2;
```

```
end

sir=1/N*sum;

sir=10*log10(sir);
```

9. A binary search technique for abrupt changing channels: main4.m

```
% Once performance degradation indicates a sudden change, the algorithm searches for
% the exact location of the change within the processing block.

% The channel response before the change occurs
as1=raylrnd(1/sqrt(2),[1 1]);
as2=raylrnd(1/sqrt(2),[1 1]);
ai1=raylrnd(1/sqrt(2),[1 1]);
ai2=raylrnd(1/sqrt(2),[1 1]);

% The channel response after the change occurs
aq1=raylrnd(1/sqrt(2),[1 1]);
aq2=raylrnd(1/sqrt(2),[1 1]);
at1=raylrnd(1/sqrt(2),[1 1]);
at2=raylrnd(1/sqrt(2),[1 1]);

% Generating the data block in which a sudden change occurs
de = floor(rand(N,1)*2);
in = floor(rand(N,1)*2);
de = 2*de-1;
in = 2*in-1;
```

```
% Generating the previous and next data blocks

previous1 = floor(rand(1,N)*2);

next1 = floor(rand(1,N)*2);

previous1 = 2*previous1-1;

next1 = 2*next1-1;

previous2 = floor(rand(1,N)*2);

next2 = floor(rand(1,N)*2);

previous2 = 2*previous2-1;

next2 = 2*next2-1;


previous=[previous1;previous2];

next=[next1;next2];


Am = [as1 as2; ai1 ai2];

D1 = [Amc(1:N/2)'; Ams(1:N/2)'];             % The source data before the abrupt change

Am1 = [aq1 aq2; at1 at2];

D2= [Amc(N/2+1:N)'; Ams(N/2+1:N)'];          % The source data after the abrupt change

D=[Amc';Ams'];                               % The whole source data block

y1 = Am*D1;

y2 = Am1*D2;

y=[y1 y2];                             % The observation block containing sudden change
```

```
y_previous=Am*previous;
y_next=Am1*next;
% Obtain the demixing matrix before the change occurs
[Shat_previous, W_previous, all_iter1] = ficai(y_previous,1e-6);

% Solving the order ambiguity
H_previous=W_previous*Am;
if abs(H_previous(1,1))<abs(H_previous(1,2))
    temp=W_previous(1,:);
    W_previous(1,:)=W_previous(2,:);
    W_previous(2,:)=temp;
end

% Solving the sign ambiguity
H_previous=W_previous*Am;
for i=1:2
    if H_previous(i,i)<0
        W_previous(i,:)=-W_previous(i,:);
    end
end

% Obtain the demixing matrix after the change occurs
[Shat_next, W_next, all_iter2] = ficai(y_next,1e-6);
```

```
H_next=W_next*Am1;
if abs(H_next(1,1))<abs(H_next(1,2))
   temp=W_next(1,:);
   W_next(1,:)=W_next(2,:);
   W_next(2,:)=temp;
end
H_next=W_next*Am1;
for i=1:2
   if H_next(i,i)<0
      W_next(i,:)=-W_next(i,:);
   end
end


before=[];        % For storing the data before the change occurs
after=[];         % For storing the data after the change occurs
start=0;          % Starting index of the current sub-block under search
t=N/2;            % Length of the sub-blocks under search
all_iter=0;
while t>=2
 test1=W_previous*y(:,start+1:start+t);   % use W_previous process the first sub-block
 sum1=0;
 for k1=start+1:start+t
   sum1=sum1+D(1,k1)^2/(D(1,k1)-test1(1,k1-start))^2;
 end

 sir1=1/t*sum1;

 sir1=10*log10(sir1);             % Calculate the performance for the first sub-block

  test2=W_next*y(:,start+t+1:start+2*t);  % use W_next process the second sub-block

 sum2=0;

 for k2=start+t+1:start+2*t

    sum2=sum2+D(1,k2)^2/(D(1,k2)-test2(1,k2-start-t))^2;

 end

 sir2=1/t*sum2;

 sir2=10*log10(sir2);             % Calculate the performance for the second sub-block


 if sir1>sir2

    % If the performance for the first sub-block is better, it does not contain the change

    before=[before y(:,start+1:start+t)];

    start=start+t;          % Reset start to be the beginning of the second sub-block

 else

    % If the performance for the second sub-block is better, it does not contain the change

    after=[y(:,start+t+1:start+2*t) after];

 end

 t=floor(t/2);              % Reset the length of the sub-blocks to be searched

all_iter=all_iter+1;

end
```

```
before=[before y(:,start+1)];

after=[y(:,start+2) after];

before=W_previous*before; % Obtain the estimation for symbols before the change

after=W_next*after;       % Obtain the estimation for symbols after the change

est=[before after];       % Estimation for the whole block


% Calculate the performance of the estimation est

sum1=0;

 [m1, n1]=size(before);

  for k=1:n1

    sum1=sum1+1/((D(1,k)-est(1,k))^2);

  end

  sir1=1/n1*sum1;

  sir1=10*log10(sir1);    % Average performance for samples before the change


sum2=0;

  [m2,n2]=size(after);

  for k=n1+1:N

    sum2=sum2+1/((D(1,k)-est(1,k))^2);

  end

  sir2=1/n2*sum2;

  sir2=10*log10(sir2);    % Average performance for samples after the change
```

10. The effect of finite arithmetic: finitelength.m, ficaic_finite.m, main5.m

```
% finitelength.m: generate the fixed-point data. Note that the matlab filter design toolbox
% is needed
function [x_fixed]=finitelength(x);    % accept double precision, output fixed-point data

q=quantizer('fixed', [32 26]);  % The word length [16, 10] is also used
[m n]=size(x);
x_fixed = zeros(m,n);
for k=1:m
   for l=1:n
      x_real=real(x(k,l));
      x_imag=imag(x(k,l));
      x_r_bin=num2bin(q,x_real);      % conversion to binary representation
      x_r_fixed=bin2num(q,x_r_bin);           % conversion to fixed-point representation
      x_i_bin=num2bin(q,x_imag);
      x_i_fixed=bin2num(q,x_i_bin);
      x_fixed(k,l)=x_r_fixed+i*x_i_fixed;
   end
end
```

```
% ficaic_finite.m: the finite precision Fast-ICA algorithm

function [Shat, Ahat, What, W, all_iter] = ficaic_finite(r,epsilon);


if nargin < 2

   epsilon = 1e-6;

end


[m,n] = size(r);

M = mean(r,2);

for i = 1:m

   r(i,:) = r(i,:)-M(i);

end


Rrr = finitelength(1/n*r*r');    % convert the current parameter into fixed-point data

[E,D] = eig(Rrr);

E=finitelength(E);

D=finitelength(D);

V = finitelength(diag(diag(D).^(-1/2))*E');

x = finitelength(V*r);


W = zeros(m,m);

all_iter = 0;
```

```
for p = 1:m
   W(:,p) = randn(m,1);
   W(:,p) = finitelength(W(:,p)/norm(W(:,p)));
   delta = 1;
   iter = 0;
   while delta > epsilon & iter < 1000 ;
      iter = iter+1;
      wnew = zeros(m,1);
      for i = 1:n
                  wnew = wnew +
                  2/n*finitelength((finitelength(x(:,i)*conj(finitelength(W(:,p)'*x(:,i))))-
                  2*W(:,p))*finitelength(abs(W(:,p)'*x(:,i))^2));
      end
      if p > 1                                    % Deflation algorithm
         for j = 1:p-1
            wnew = wnew-finitelength(W(:,j)*W(:,j)'*wnew);
         end
      end
      wnew = finitelength(wnew/norm(wnew));
      delta = abs(abs(finitelength(W(:,p)'*wnew))-1);
      W(:,p) = wnew;
   end
   all_iter=all_iter+iter;
```

```
end
for i = 1:m
   r(i,:) = r(i,:)+M(i);
end

What = finitelength(W'*V);
Shat = finitelength(What*r);
Ahat = finitelength(inv(What'));

% main5.m: generating fixed-point observations, and call ficaic_finite.m
close all;
as1=raylrnd(1/sqrt(2),[1 1]);
as2=raylrnd(1/sqrt(2),[1 1]);
ai1=raylrnd(1/sqrt(2),[1 1]);
ai2=raylrnd(1/sqrt(2),[1 1]);

fais1=2*pi*rand(1);
fais2=2*pi*rand(1);
faii1=2*pi*rand(1);
faii2=2*pi*rand(1);

fs1=as1*exp(j*fais1);
fs2=as2*exp(j*fais2);
```

```
fi1=ai1*exp(j*faii1);

fi2=ai2*exp(j*faii2);


sr=floor(rand(N,1)*2);

sr=2*sr-1;

si=floor(rand(N,1)*2);

si=2*si-1;

de=sr+j*si;

de=de/sqrt(2);


imager=floor(rand(N,1)*2);

imager=2*imager-1;

imagei=floor(rand(N,1)*2);

imagei=2*imagei-1;

in=imager-j*imagei;

in=in/sqrt(2);


Am=[fs1 fi1; fs2 fi2];

D=[de';in'];

y = finitelength(Am*D);         % fixed-point observation


[Shat, Ahat, What, W, all_iter] = ficaic_finite (y,1e-6);   % call finite precision Fast-ICA

S = zeros(size(Shat));
```

```
for i = 1:2
   d1 = norm(abs(Shat(i,:)-de'));
   d2 = norm(abs(Shat(i,:)+de'));
   d3 = norm(abs(Shat(i,:)-in'));
   d4 = norm(abs(Shat(i,:)+in'));
   d = [d1 d2 d3 d4];
   [m,I] = min(d);
   switch I
   case 1
      S(1,:) = Shat(i,:);
   case 2
      S(1,:) = -Shat(i,:);
   case 3
      S(2,:) = Shat(i,:);
   case 4
      S(2,:) = -Shat(i,:);
   end
end

H = What*Am;
SIRR =
(abs(20*log10(abs(H(1,1))/abs(H(1,2))))+abs(20*log10(abs(H(2,1))/abs(H(2,2)))))/2
```

11. Simplified BPSK with single antenna and one downconversion path: main6.m

```
close all;

as1=raylrnd(1/sqrt(2),[1 1]);

ai1=raylrnd(1/sqrt(2),[1 1]);

fais1=2*pi*rand(1);

faii1=2*pi*rand(1);

fs1=as1*exp(j*fais1);

fi1=ai1*exp(j*faii1);

alpha=2*pi*rand(1);

a=real(fs1*exp(-j*alpha));

b=real(fi1*exp(-j*alpha));


de = floor(rand(N,1)*2);

in = floor(rand(N,1)*2);

de = 2*de-1;

in = 2*in-1;


Am=[a b];

D=[de';in'];

y = Am*D;     % The available observation
```

```
% If necessary, add thermal noise
%sigma = 1/10;
%yn = y +randn(1,N)*sigma;

temp=y.^3;      % Obtain the second signal from the available observation
y1 =[y;temp];  % Append the second signal to the available observation
[Shat, W, all_iter] = ficai(y1,1e-6);    % Separation based on the two signals

S = zeros(size(Shat));
for i = 1:2
   d1 = norm(abs(Shat(i,:)-de'));
   d2 = norm(abs(Shat(i,:)+de'));
   d3 = norm(abs(Shat(i,:)-in'));
   d4 = norm(abs(Shat(i,:)+in'));
   d = [d1 d2 d3 d4];
   [m,I] = min(d);
   switch I
   case 1
      S(1,:) = Shat(i,:);
   case 2
      S(1,:) = -Shat(i,:);
   case 3
      S(2,:) = Shat(i,:);
   case 4
       S(2,:) = -Shat(i,:);
   end
end


sum=0;
for k=1:N
       sum=sum+de(k)^2/(de(k)-S(1,k))^2;
end
sir=1/N*sum;
sir=10*log10(sir);
```

12. The Monte-Carlo simulation for average performance: run.m

```
% The file specifies the number of simulations runs, the block size to be used, and
% call one of the main functions to execute experiments.  Results are stored.

clear all;
clc;

Nruns = 100;                      % Specify the number of simulation runs
Nf = 100:50:1000;                 % Specify the length of the processing frame
ResSir = zeros(length(Nf), 1);    % Storing the average signal to interference ratio
ResIter=zeros(length(Nf),1);      % Storing the average number of iterations to convergence

for i = 1:length(Nf)              % Simulate for every frame length
   for iter = 1:Nruns             % Iterate for Nruns times
      N = Nf(ii);                 % Extract the frame length
      main                        % Call one of the main functions
      Rs(iter) = sir;             % Store the SIR value for each run
      Ri(iter)=all_iter;          % Store the number of iterations for each run
   end
ResSir(ii) = mean(Rs);            % Compute the average SIR value
ResIter(ii)=mean(Ri);             % Computer the average convergence speed
end
```

LIST OF REFERENCES

[1]. R. Esmailzadeh, M. Nakagawa, “TDD-CDMA for wireless communications”, Artech House, Inc., 2003.

[2]. E. Buracchini, “The Software Radio Concept”, *IEEE Communications Magazine,* September 2000.

[3]. A. Paulraj, R. Nabar, D. Gore, “Introduction to Space-Time Wireless Communications”, Cambridge University Press, 2003.

[4]. D. Gesbert, M. Shafi, D. Shiu, P. J. Smith, A. Naguib, “From Theory to Practice: An Overview of MIMO Space-Time Coded Wireless Systems”, *IEEE Journal on Selected Areas in Communications*, Vol. 21, No. 3, April 2003.

[5]. A. J. Viterbi, “CDMA: Principles of Spread Spectrum Communication”, Addison-Wesley Publishing Company, 1995.

[6]. Won Namgoong, Teresa H. Meng, “Direct-Conversion RF Receiver Design”, *IEEE Transactions on Communications*, vol.49, NO. 3, March 2001

[7]. Jan Crols, Michiel S. J. Steyaert, “Low-IF Topologies for High-Performance Analog Front Ends of Fully Integrated Receivers”, *IEEE Transactions on Circuits and Systems*, vol. 45, no. 3, March 1998

[8]. I. Kostanic and W. Mikhael, “Blind source separation technique for the reduction of co-channel interference”, *IEE Electronics Letters*, Vol. 38, No. 20, September 2002.

[9]. I. Kostanic and W. Mikhael, “Rejection of the Co-Channel Interference Using Non-Coherent Independent Component Analysis Based Receiver”, *the 45th*

IEEE International Midwest Symposium On Circuits and Systems, Vol. 2, August 2002.

[10]. T. Ristaniemi and J. Joutsensalo, "Advanced ICA-based receivers for block fading DS-CDMA channels", *Signal Processing*, Vol. 82, No. 3, pp. 417-431, March 2002.

[11]. T. Yang, W. Mikhael, "A General Approach for Image and Co-channel Interference Suppression in Diversity Wireless Receivers Employing ICA", *Journal of Circuits, Systems, and Signal Processing (**CSSP**)*, VOL. 24, No. 3, 2004, pp. 317-327.

[12]. T. Yang, W. Mikhael, "Baseband Image Rejection for Diversity Superheterodyne Receivers", *IEEE Wireless Communications and Networking Conference (**WCNC**) 2004*, Atlanta, Georgia, March 2004.

[13]. T. Yang, W. Mikhael, "Diversity Wireless Receivers with Efficient Co-Channel Interference Suppression", *2004 IEEE/Sarnoff Symposium on Advances in Wired and Wireless Communication*, Princeton, New Jersey, April 2004.

[14]. T. Yang, W. Mikhael, "A General Interference Suppression Scheme for Diversity Wireless Receivers", **Invited paper**, *Workshop on Wireless Circuits and Systems (**WoWCAS**)*, University of British Columbia, Vancouver, Canada, May 2004.

[15]. T. Yang, W. Mikhael, "ICA With Improved Computational Efficiency For Interference Suppression In Wireless Receivers", *The 8th World Conference*

*on Systemics, Cybernetics, and Informatics (**SCI** 2004)*, Orlando, Florida, July 2004.

[16]. W. Mikhael, T. Yang, “A Gradient–based Optimum Block Adaptation ICA Technique for Interference Suppression in Highly Dynamic Communication Channels”, submitted to *IEEE Transactions on Circuits and Systems*, August 2004.

[17]. T. Yang, W. Mikhael, “Practical Implementation Issues of ICA Based Image Rejection for IF Wireless Receivers”, ***SCI** 2004*, Orlando, Florida, July 2004.

[18]. T. Yang, W. Mikhael, “Practical Image Rejection for Single Branch BPSK Receivers”, *IEE Electronics Letters*, Vol. 40, No. 22, October 2004.

[19]. T. W. Lee, M. S. Lewicki, and T. J. Sejnowski, “ICA mixture models for unsupervised classification of non-gaussian sources and automatic context switching in blind signal separation”, *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 22(10): 1-12, 2000.

[20]. A. Belouchrani and M. Amin, “Jammer mitigation in spread spectrum communications using blind source separation”, *Signal Processing*, 80(4): 723-729, 2000.

[21]. L. Castedo, C. Escudero and A. Dapena, “A blind signal separation method for multiuser communications”, *IEEE Transactions on Signal Processing*, 45(5): 1343-1348, 1997.

[22]. S. Malaroiu, K. Kiviluoto and E. Oja, “Time series prediction with independent component analysis”, Proceedings of *International Conference on Advanced Investment Technology*, Gold Coast, Australia, 2000.

[23]. M. McKeown, S. Makeig, S. Brown, T. P. Jung, S. Kindermann, A.J. Bell, V. Iragui and T. Sejnowski, "Blind separation of functional magnetic resonance imaging (fMRI) data", *Human Brain Mapping*, 6(5-6): 368-372, 1998.

[24]. C.L. Isbell and P. Viola, "Restructuring sparse high-dimensional data for effective retrieval", *Advances in Neural Information Processing Systems*, Volume 11. MIT Press, 1999.

[25]. A. Hyvarienen, J. Karhunen and E. Oja, "Independent Component Analysis", John Wiley and Sons, 2001.

[26]. A. Hyvärinen and E. Oja, "A fast fixed-point algorithm for independent component analysis", *Neural Computation*, 9(7): 1483-1492, 1997.

[27]. A. Hyvärinen, "Fast and robust fixed-point algorithms for independent component analysis", *IEEE Transactions on Neural Networks*, Vol. 10, No. 3, May 1999.

[28]. E. Bingham and A. Hyvärinen, "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.

[29]. W. Mikhael and F. Wu, "A fast block FIR adaptive digital filtering algorithm with individual adaptation of parameters", *IEEE Transactions on Circuits and Systems*, Vol. 36, No. 1, January 1989.

[30]. W. Mikhael, F. Wu, L.G. Kazovsky, G. S. Kang and L. J. Fransen, "Adaptive Filters with Individual Adaptation of Parameter", *IEEE Transactions on Circuits and Systems*, Vol. 33, No. 7, July 1986.

[31]. W. Mikhael and F. Wu, "Fast algorithms for block FIR adaptive digital filtering", *IEEE Transactions on Circuits and Systems*, Vol. 34, No. 10, October 1987.

[32]. B. Sklar, "Digital Communications: Fundamentals and Applications" (Second Edition), Prentice Hall, 2002.

[33]. J. G. Proakis, "Digital Communications" (Third Edition), McGraw-Hill, Inc., 1995.

[34]. M. Valkama, M. Renfors, V. Koivunen, "Advanced methods for I/Q imbalance compensation in communication receivers", *IEEE Transactions On Signal Processing*, vol. 49, no. 10, October 2001.

[35]. Jan Crols, Michiel S. J. Steyaert, "A single-chip 900 MHz CMOS receiver front-end with a high performance Low-IF topology", *IEEE J. Solid-State Circuits*, vol. 30, no. 12, Dec. 1995.

[36]. W. Mikhael, R. C. Chehata, D. Chester, B. Myers, "Efficient Adaptive Algorithm Implementations For Performance Enhancement Of Zero IF Receivers", *Proceedings of Systemic, Cybernetics and Informatics*, vol 15, pp. 362-366, July, 2002.

[37]. Jack P.F. Glas, "Digital I/Q imbalance compensation in a low-IF receiver", *Global Telecommunications Conference*, 1998.

[38]. E. R. Ferrara, "Fast implementation of LMS adaptive filters", *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. ASSP-28, No. 4, August 1980.

[39]. C. H. Roth, Jr., "Fundamentals of Logic Design", Fourth Edition, West, 1992.

[40]. K. Han, I. Eo, K. Kim, H. Cho, “Numerical word-length optimization for CDMA demodulators”, *IEEE International Symposium on Circuits and Systems*, pp 290-3, Vol. 4, 2001.

[41]. M. Cantin, Y. Savaria, P. Lavoie, “A comparison of automatic word length optimization procedure”, *IEEE International Symposium on Circuits and Systems*, Vol. 2, pp 612-615, 2002.

[42]. H. Keding, M. Willems, M. Coors, H. Meyr, “FRIDGE: a fixed point-design and simulation environment”, *Design, Automation and Test in Euro*pe, pp. 429-435, 1998.

[43]. M. Cantin, Y. Blaguiere, Y. Sarvaria, P. Lavoie, E. Granger, “Analysis of quantization effects in a digital hardware implementation of a fuzzy ART neural network algorithm”, *IEEE International Symposium on Circuits and Systems*, pp. 141-144, vol. 3, 2000.

[44]. M. Cantin, Y. Blaguiere, Y. Sarvaria, P. Lavoie, “An automatic word length determination method”, *IEEE International Symposirum on Circuits and Systems*, Vol. 5, pp. 53-56, 2001.

[45]. S. Kim, K. Kum, W. Sung, “Fixed-point optimization utility for C and C++ based signal processing programs”, *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing”,* vol. 45, no. 11, 1998.

[46]. K. Kum, W. Sung, “Word-length optimization for high-level synthesis of digital signal processing systems”, *IEEE Workshop on Signal Processing Systems*, pp. 569-578, 1998.

[47]. L. Yu and W. M. Snelgrove, “A novel adaptive mismatch cancellation system for quadrature IF radio receivers,” *IEEE Transactions on Circuits and Systems II*, vol. 46, pp. 789–801, June 1999.

[48]. Cetin, E.; Kale, I.; Morling, R.C.S., “Adaptive compensation of analog front-end I/Q mismatches in digital receivers”, *2001 IEEE International Symposium on Circuits and Systems,* Vol. 4, pp. 370 -373, May 2001.

[49]. Kong-Pang Pun, J.E. Franca, C. Azeredo-Leme, “A switched-capacitor image rejection filter, for complex IF receivers”, *IEEE APCCAS*, pp. 140 –143, 2000.

[50]. Nam-Soo Kim, Jung-Ki Choi, Shin-Chol Kim, Sang-Gug Lee, Chan-Gu Lee, Hae-Won Jung, Hyun-Kyu Yu, “An image rejection down conversion mixer architecture”, *TENCON*, vol. 1, pp. 287-289, 2000.

[51]. T. Yang, W. Mikhael, “Baseband Digital Image-Suppression in Low-IF Receivers by Complex-Valued ICA”, *The 46th IEEE International Midwest Symposium on Circuits and Systems* ***(MWSCAS),*** Cairo, Egypt, December 2003.

[52]. A. Hyvärinen, E. Oja, “Independent Component Analysis by general nonlinear Hebbian-like learning rules”, *Signal Processing*, 64(3): 301-313, 1998.

[53]. Z. Malouche, O. Macchi, “Adaptive unsupervised extraction of one component of a linear mixture with a single neuron”, *IEEE Transactions on Neural Networks*, 9(1): 123-138, 1998.

[54]. M. Gaeta, J.-L. Lacoume, “Source separation without prior knowledge: the maximum likelihood solution”, in *Proc. EUSIPCO’90*, pp. 621-624, 1990.

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

[56]. D.-T. Pham, “Blind separation of instantaneous mixture sources via an independent component analysis”, *IEEE Transactions on Signal Processing*, 44(11): 2768-2779, 1996.

[57]. D.-T. Pham, P. Garrat, “Blind separation of mixture of independent sources through a quasi-maximum likelihood approach”, *IEEE Transactions on Signal Processing*, 45(7): 1712-1725, 1997.

[58]. J.-F. Cardoso, “Infomax and maximum likelihood for source separation”, *IEEE letters on Signal Processing*, 4:112-114, 1997.

[59]. S.-I. Amari, “Neural learning in structured parameter space – natural Riemannian gradient”, In *Advances in Neural Information Processing Systems 9*, pp. 127-133, MIT Press, 1997.

[60]. S.-I. Amari, A. Cichocki, H.H. Yang, “A new learning algorithm for blind source separation”, In *Advances in Neural Information Processing Systems 8*, pp. 757-763, MIT Press, 1996.

[61]. P. Comon, “Independent Component Analysis – a new concept”, *Signal Processing*, 36:287-314, 1994.

[62]. Z. He, L. Yang, J. Liu, Z. Lu, C. He, Y. Shi, " Blind source separation using clustering – based multivariate density estimation algorithm", *IEEE Transactions on Signal Processing*, 48(2): 575-579, 2000.

[63]. D.-T. Pham, "Blind separation of instantaneous mixture of sources based on order statistics", *IEEE Transactions on Signal Processing*, 48(2): 363-375, 2000.

[64]. J. Principe, D. Xu, J. W. Fisher III, " Information-theoretic learning", in S. Haykin, editor, *Unsupervised Adaptive Filtering*, Vol. I, pp. 265-319, Wiley, 2000.

[65]. J.-F. Cardoso, "Source separation using higher order moments", Proc. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2109-2112, Glasgow, UK, 1989.

[66]. J.-F. Cardoso, A. Souloumiac, "Blind beamforming for non Gaussian signals", *IEE Proceedings-F*, 140(6):362-370, 1993.

[67]. L. De Lathauwer, "Signal Processing by Multilinear Algebra", Ph.D. thesis, Faculty of Engineering, K. U. Leuven, Leuven, Belgium, 1997.

[68]. P. Comon, B. Mourrain, "Decomposition of quantics in sums of powers of linear forms", *Signal Processing*, 53(2): 93-107, 1996.

[69]. J.-F. Cardoso, "High-order contrasts for independent component analysis", *Neural Computation*, 11(1):157-192, 1999.

[70]. A. Yeredor, "Blind source separation via the second characteristic function", *Signal Processing*, 80:897-902, 2000.

[71]. W. Mikhael, F. Wu, G. Kang, L. Fransen, “Optimum adaptive algorithms with applications to noise cancellation”, *IEEE Transactions on Circuits and Systems,* vol. 31, pp. 312-315, March 1984.

[72]. W. Mikhael and F. Wu, “Fast Gradient Algorithms For Block FIR Adaptive Digital Filters”, *Nineteenth Asilomar Conference on Circuits, Systems and Computers*, November 1985.

[73]. M. Barsoum and W. Mikhael, “A conjugate-gradient-based block adaptive algorithm, its applications and performance evaluation”, *The 37th IEEE Midwest Symposium on Circuits and Systems*, August 1994.

[74]. W. Mikhael, F. Wu, “Gradient algorithms for FIR adaptive filtering-a tutorial”, *IEEE International Symposium on Circuits and Systems,* June 1991.

[75]. W. Mikhael, S. Ghosh, “Adapting: From Wiener to Widrow”, *Twenty-fourth Asilomar Conference on Circuits, Systems and Computers,* November 1990.

[76]. T. Wang, C. Wang, “In the optimum design of the block adaptive FIR digital filter”, *IEEE Transactions on Signal Processing*, Vol. 41, pp. 2131-2140, June 1993.

[77]. J. Lim, C. Un, “Conjugate gradient algorithm for block FIR adaptive filtering”, *IEE Electronics Letters,* Vol. 29, pp. 428-429, March 1993.

[78]. T. Wang, C. Wang, “A new optimum block adaptive FIR filtering algorithm”, *IEEE Region 10 International Conference*, TENCON 1992.

[79]. W. Mikhael, T. Yang, “Optimum Block Adaptive Algorithm for Gradient Based Independent Component Analysis for Time Varying Wireless

channels", submitted to *2005 IEEE International Symposium on Circuits and Systems*, Kobe Japan, May 2005.

[80]. W. Mikhael, T. Yang, "Optimum Block Adaptive ICA Algorithm for Dynamic Environments", submitted to *IEE Electronics Letters*, November 2004.

[81]. T. Yang, W. Mikhael, "Improved ICA Technique for Interference Suppression in Diversity Wireless Receivers", submitted to *Journal of Communications and Networking (**JCN**)* in November 2003, revised and resubmitted in April 2004.

[82]. T. Yang, W. Mikhael, "ICA-based Interference Rejection for Diversity Receivers in Non-Stationary Channels", *IASTED International Conference on Circuits, Signals and Systems (CSS 2004)*, Clearwater Florida, November 2004.

[83]. T. Yang, W. Mikhael, "Computational Complexity and Effect of Finite Arithmetic for ICA Based Image Rejection in Wireless Receivers", submitted to *Digital Signal Processing – A Review Journal*, October 2003.

[84]. W. Mikhael, T. Yang, "Efficient Digital Image Rejection for Single Branch BPSK Receiver", to appear in proceedings of *2005 IEEE International Conference on Speech, Acoustics, and Signal Processing (ICASSP 2005),* Philadelphia, Pennsylvania, March 2005.

[85]. W. Mikhael, T. Yang, "A Novel Digital Image Rejection Technique for Simplified BPSK Wireless Receivers", submitted to *IEEE Transactions on Wireless Communications*, October 2004.

[86]. The *International Engineering Consortium* Web ProForum tutorials, http://www.iec.org/online/tutorials/ , “Cellular Communications”, “Smart Antenna Systems”.

[87]. M. Davies, N. Mitianoudis, “Simple mixture model for sparse overcomplete ICA”, *IEE Proceedings on Vision, Image and Signal Processing*, Vol. 151, No. 1, February 2004.

[88]. T. Lee, M. Lewicki, M. Girolami, T. Sejnowski, “Blind Source Separation of More Sources Than Mixtures Using Overcomplete Representations”, *IEEE Signal Processing Letters*, Vol. 6, No. 4, April 1999.

[89]. L. Zhang, A. Cichocki, S. Amari, “Natural gradient algorithm for blind separation of overdetermined mixture with additive noise”, *IEEE Signal Processing Letters*, Vol. 6, Issue 11, November 1999.

[90]. S. Winter, H. Sawada, S. Makino, “Geometrical understanding of the PCA subspace method for overdetermined blind source separation”, *ICASSP '03*, Vol. 2, pp. 6-10, April 2003.