Further Study of Independent Component Analysis in Foreign Exchange Rate Markets

By
Zhi-bin LAI

Supervised By
Prof. Lei XU

Submitted to the Division of Computer Science and Engineering
in partial fulfillment of the requirements for the
degree of Master of Philosophy
at
the Chinese University of Hong Kong

December 1998

Further Study of Independent Component Analysis in Foreign Exchange Rate Markets

submitted by

Zhi-bin Lai

for the degree of Master of Philosophy

at The Chinese University of Hong Kong

Abstract

Recently, independent component analysis (ICA) has provided a new tool to analyze financial markets (Back and Weigend, 1997), in which the financial data are regarded as the linear mixture of a set of independent components (ICs). However, there exist two important problems need to be solved: (1) how to arrange the order of the obtained ICs under certain dominant sense; (2) how to select the appropriate number of dominant ICs which reflect the major movement of the observed financial data.

In view of the first problem, we determine the dominant ICs order under measurement of the Mean Square Error (MSE) between the original data and the reconstructed data, which is wholly different from those existing heuristic methods of sorting the order of dominant ICs according to their weights. Based on this criterion, we study a Forward Selection approach to sort the WICs into a certain order according to their dominant values measured by MSE. Considering of the different practical needs, we also determine the dominant ICs order under measurement of Tendency Error (TE) between the original data and the reconstructed data. We study a Backward Elimination Tendency Error (BETE) approach to implement this criterion. For the second problem, we develop number determination criterion under MSE and TE measurement respectively. Large number of experiments show that the dominant ICs obtained by these order-sorting approaches and number-determination criteria are better than those heuristically obtained in the MSE and TE signal reconstruction, which not only reflect the major movement of the observed financial data, but also make the reconstruction of non-dominant ICs unbiased.

Acknowledgments

The most heartfelt thanks is given to my supervisor, Prof. Xu Lei, for his guidance and patience. Not only he teach me how to do research, but also make me grasp abilities in many aspects, which are very important for my future life and work.

I sincerely thank Mr. Cheung Yiu-ming, who give me much help and suggestion in the finish of my thesis.

Also, I would like to thank Mr. Cheung Chi-chiu, Mr. Leung Wai-man and Prof. Wang Tai-jun for their help and fruitful discussion. Besides, I would also like to many of my colleagues who give me support and help.

Lastly, I am deeply grateful to my family for their care and encouragement during past years.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Recently, blind source separation (BSS) problem has attracted much attention in the fields of neural networks and signal processing. It has been widely used in speech recognition, telecommunication and medical signal processing, etc. For linear mixing cases, it can be formulated as an independent component analysis (ICA) problem, where the objective is to separate mutually independent unknown source signals from their instantaneous linear mixtures without the knowledge of the mixing process.

1.1 ICA Model

We assume the observed n financial data series (or called *sensor signals*) $\{x(t)\}_{t=1}^{N}$ with $x(t) = [x_1(t), ..., x_n(t)]$ are the instantaneous linear mixture of k unknown statistically independent source signals $\{s(t)\}_{t=1}^{N}$ with $s(t) = [s_1(t), ..., s_k(t)]$, which maybe response to any useful information such as political and economical news, investment environment as well as some unexplained noise. Hence, the model is:

$$x(t) = As(t), \text{ with } Es(t) = 0 \tag{1.1}$$

where A is a $n \times k$ mixing matrix. In the following, we only consider the case of $n = k$, and without ambiguity we will omit the time index t. The objective of ICA in this model is to recover the source signals (also called *independent components*) from the observed mixed signals x through a de-mixing process:

$$y = Wx \tag{1.2}$$

Then we can get the information passage

$$s \to x = As \to y = Wx = WAs = Vs \tag{1.3}$$

We can tune W in such a way that makes y mutually independent and recover s up to a unknown constant and a permutation of indices.

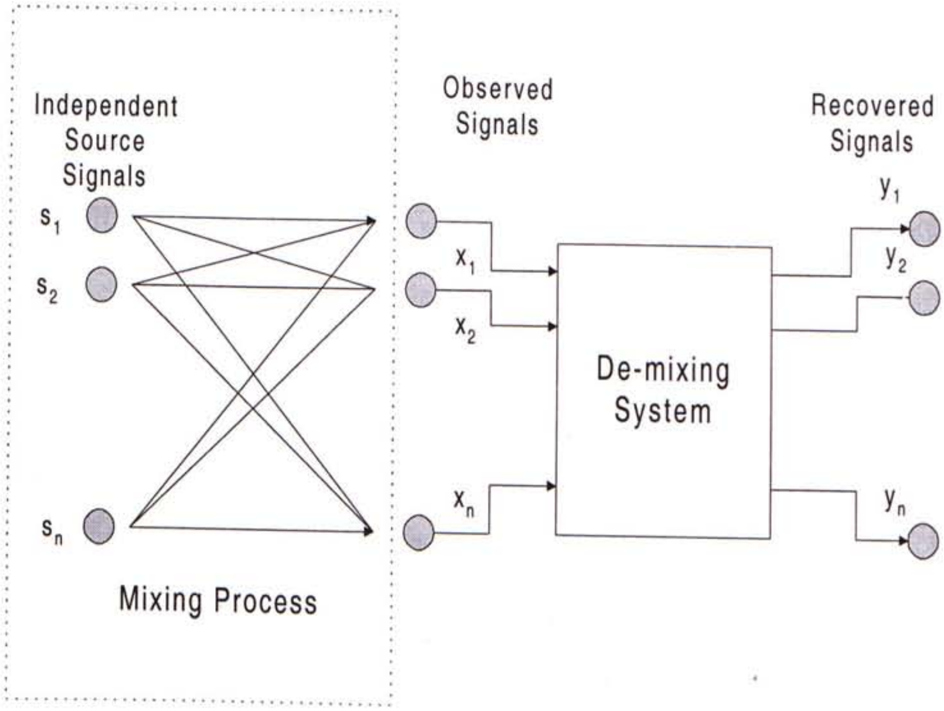


Figure 1.1: ICA Model

1.2 ICA Algorithms

Usually there exists necessary assumption that the separated source signals should be statistically independent, so in order to achieve successful separation of the mixture signals, high-order statistics should be put into consideration. Up to date, different neural approaches to BSS and ICA have been proposed.

As we know, blind separation problems was firstly inspired by Herault and Jutten (1986) and much attentions have been focused on this aspect afterwards. In the well-known Herault-Jutten (HJ) algorithm, the separating matrix W is written as the form as $W = (I + M)^{-1}$, and M is a matrix with its diagonal elements keep zero and its off-diagonal elements have the iterative equation as

$$M_{k+1} = M_k + \eta_k g(y_k) h(y_k^T) \tag{1.4}$$

where η_k is the learning rate. $g(y)$ and $h(y)$ are two odd functions. Deville and Andry (1995) showed that $g(y) = y^3$ and $h(y) = y$ can separate sub-Gaussian sources signals, $g(y) = y$ and $h(y) = y^3$ are suitable for separating super-Gaussian source signals.

Various kind of modification of this algorithm have been proposed (Cichocki et al. 1995, 1997), in which the diagonal elements of matrix M are also updated in each iteration. In order to avoid the computation of the inverse of the matrix, some approximation have also been made, such as $W_{k+1} = I - M_{k+1}$ and performance of the algorithm keeps unchanged and sometimes even better than the original algorithm.

Comon (1994) first define the concept of independent component analysis which measures the degree of independence among outputs using contrast functions approximately by the Edgeworth expansion of the Kullback-Leibler divergence. The higher order statistics is approximated by cummulants up to 4th order and require intensive computation.

ICA algorithms are usually implemented in either off-line or on-line approach. For batch algorithm, usually it is composed of two procedures. The first step is called decorrelation or whitening, in which the covariance matrix of the input signals is diagonalized. The second step is called rotation, in which a unitary rotation matrix is used to maximize a measure of the higher order statistics which results the non-Gaussian output signals are as statistically independent as possible.

One typical example is the JADE (Joint Approximate Diagonalization of Eigenmatrices) algorithm proposed by Cardoso and Souloumiac in 1993, which is also composed of above mentioned two stages. The observed signals are first whitened through eigendecomposition of their covariance matrix, then a rotation matrix is used to jointly diagonalize the eigenmatrices got from the fourth order cummulants of the whitened observed data. A key advantage of this algorithm is its computational efficiency. One extension of the JADE algorithm can be found in (Pope and Bogner 1994).

In the equivariant source separation method (Cardoso and Laheld 1996), the de-mixing is performed by serial updating of a decorrelation matrix to produce orthogonal signals. This method is based on the method called fourth order blind identification (Cardoso 1989) that consists of two steps, orthonormalization and quadratic weighting of the covariances that be used to obtain fourth order moments.

Cardoso (1996) proposed a ML blind source separation algorithm, in which the structure of source separation as a multi-dimensional location-scale model, entailing a specific form of parameterization and a specific notion of gradient as location-scale transformation form a group. The ML estimation of the source signals only depend on the particular realization of the source signal, not on the transformation. Batch and adaptive algorithms can be obtained and show same performance.

Bell and Sejnowski (1995) proposed the Maximization Entropy (ME) approach, in which some nonlinear transformation functions are suitably chosen to be the cumulative distribution function of the sources. The output entropy is maximized to ensure the separation of the blind source signals. The adaptive equation for the de-mixing matrix has the following form

$$W_{k+1} = W_k + \eta_k[(W_k^T)^{-1} + h(y)x^T] \tag{1.5}$$

Modified iteration equation is formed as

$$W_{k+1} = W_k + \eta_k(I + h(y)y^T)W_k \tag{1.6}$$

In which the natural gradient algorithm (Amari et al. 1996) is used to accelerate the convergence by multiplying the positive definite matrix W^TW to the right of the gradient to avoid the computation of the inverse of the de-mixing matrix .

Amari et al. (1996) proposed an adaptive algorithm to minimize the mutual information between the output (estimated source signals), which equals to the Kullback-Divergence between the joint density of the output signals and the product of marginal densities of the output signals. The mutual information is minimized only when the source signals have already been correctly separated. A truncated Gram-Charlier series is used to approximate the mutual information. The de-mixing matrix had the same form of iteration equation with that of the modified ME algorithm but with the different nonlinearity function which results in the different separation performance.

The bigradient algorithm (Wang et al. 1995a,b,c) has the form of

$$W_{k+1} = W_k + \eta_k m_k n(y_k^T) + \beta_k W_k(I - W_k^T W_k) \tag{1.7}$$

in which the learning parameter η_k can be either positive or negative. The first update term is actually a nonlinear Hebbian term, and the second term keeps the matrix W_k roughly orthonormal. The basic bigradient algorithm can be modified with some slightly different forms to be able to separate either sub-Gaussian or super-Gaussian source signals.

Hyvarinen and Oja (1997) proposed a fixed-point algorithm, in which the neural network learning rule can be transformed into a fixed-point iteration. The relative algorithm does not dependent on any user-defined parameters and can find all non-Gaussian independent components at a time regardless of their probability distribution. The convergence speed is cubic, much faster than gradient based algorithms.

The EASI algorithm (Cardoso and Laheld 1996) can be thought as an adaptive nonlinear PCA type algorithm. This algorithm is based on the idea of serial updating by which the uniform performance property of equivariant estimators is directly inherited by the corresponding adaptive serial algorithms. In the algorithm, a vector-to-matrix mapping is serial defined by let its symmetric part to a second order condition of independence (decorrelation) while the skew-symmetric part involves nonlinear function. Experiments show that the convergence rate and stability condition depend only on the distribution of the source signals.

Usually the source signals are assumed to be linearly mixed. It is apparent very limited and unsuitable for many practical problems. Some extensions to nonlinear mixing models have been proposed. Herrmann and Yang (1996) use self-organizing map (SOM) to extract sources from nonlinear mixture. Yang et al. (1996) employed a two-layer perceptron model as a de-mixing system by gradient method to minimize the mutual information of the outputs.

Different from above mentioned algorithms, there are still many other approaches. Pearlmutter and Parra (1997) proposed a contextual ICA algorithm which is based on maximum likelihood estimation. The source distribution are modeled and the temporal nature of the signal is used to derive the de-mixing matrix. The density function of the input signals are estimated using past values of the outputs. This algorithm is shown to be effective in separating signals having colored Gaussian distributions or low kurtosis. In (Cichocki et al. 1997) two types of cascade neural networks are applied to extract independent source signals from a linear mixture of them when the number of noisy mixed signals is equal to or larger than the number of sources. The developed learning algorithm can be considered as a generation of extension of Hebbian/anti-Hebbian rules.

Learned Parametric Mixture Based ICA Algorithm

Various approaches have been proposed to separate blind source signals, such as Minimum Mutual Information approach (Amari et al. 1996) and Maximization Entropy approach (Bell and Sejnowski 1995), which learn the de-mixing matrix W adaptively by $W^{new} = W^{old} + \eta \Delta W$, $\Delta W = (I + \phi(y)y^T)W$. However, these approaches can only separate sub-Gaussian or super-Gaussian signals due to the fixed separation nonlinearity $\phi(y)$. To tackle this problem, the learned parametric mixture based ICA (LPM) algorithm has been proposed (Xu et al. 1998), which adaptively learns the separation nonlinearity instead of fixing it with the result that this algorithm can separate any combinations of sub-Gaussian and super-Gaussian signals, this is also in conformity with the actual situation of the exchange rate markets.

Here we briefly introduce the de-mixing process of the LPM algorithm:

From (Xu et al. 1998), the cost function is

$$\begin{aligned} J(W) &= \int_x p(x) \log \frac{p(x)}{|\det[W]| \Pi_{i=1}^n g_i(w_i^T x)} dx \\ &= \int_s p(s) \log \frac{p(s)}{|\det[V]| \Pi_{i=1}^n g_i(v_i^T s)} ds \end{aligned} \tag{1.8}$$

where

$$g_i(y_i) = \sum_{j=1}^{p_i} \alpha_{ij} \varphi(u_{ij}) \tag{1.9}$$

is the form of mixture of densities to be able to approximate any function arbitrarily, where

$$\varphi(u_{ij}) = b_{ij} \phi'(u_{ij}) \tag{1.10}$$

$$u_{ij} = b_{ij}(y_i - a_{ij}) \tag{1.11}$$

$$\alpha_{ij} = \frac{\exp(\gamma_{ij})}{\sum_{k=1}^{p_i} \exp(\gamma_{ik})} \tag{1.12}$$

Here b_{ij} is a scaling factor, a_{ij} is a bias, γ_{ik} can take any real value.

One selection of $\phi(u_{ij})$ is

$$\phi(u_{ij}) = \log sig(u_{ij}) = \frac{1}{1+\exp(-u_{ij})} \tag{1.13}$$

and

$$\phi'(u_{ij}) = \phi(u_{ij})(1-\phi(u_{ij})) \tag{1.14}$$

The natural gradient algorithm derived by Amari et al (1996) is used :

$$\frac{dW}{dt} \propto -[\nabla_W J(W)]W^T W \tag{1.15}$$

its stochastic form is

$$\Delta W = \varepsilon(t)[I + h(y)y^T]W \tag{1.16}$$

where

$$h(y) = [h_1(y_1), \ldots\ldots, h_n(y_n)]', \quad h_i(y_i) = \frac{g_i^{'}(y_i)}{g_i(y_i)} \tag{1.17}$$

The $h_i(y_i)$ nonlinearity is written as

$$h_i(y_i) = \frac{1}{g_i(y_i)} \sum_{j=1}^{p_i} \alpha_{ij} b_{ij} \varphi'(u_{ij}) \tag{1.18}$$

For $\phi(u_{ij}) = \log sig(u_{ij})$, $\varphi'(u_{ij}) = b_{ij}(1-2\phi(u_{ij}))\phi'(u_{ij})$.

The parameters $\{\gamma, a, b\}$ are tuned in the gradient descent algorithm together with the tune of W to minimize the cost function iteratively on the arrival of each data point. After simplification, finally, we can get the adaptive algorithm as follows

$$\Delta\gamma_{ij} = \frac{1}{g_i(y_i)} \sum_{k=1}^{p_i} b_{ik}\phi'(u_{ik})\alpha_{ik}(\delta_{kj} - \alpha_{ij}) \tag{1.19}$$

$$\Delta b_{ij} = \frac{\alpha_{ij}}{g_i(y_i)}\{\phi'(u_{ij}) + \phi''(u_{ij})u_{ij}\} \tag{1.20}$$

$$\Delta a_{ij} = -\frac{1}{g_i(y_i)}\alpha_{ij}b_{ij}^2\phi''(u_{ij}) \tag{1.21}$$

where δ_{ij} is the Kroniker delta function.

1.3 Foreign Exchange Rate Scheme

As shown in (George and Giddy 1983; Hallwood and MacDonald 1994), the foreign exchange markets have been subject to considerable volatility, and to erratic movements in recent years. Like other asset prices, exchange rates are affected by an integrated process that includes the following elements: change in supply of and demand for money and financial assets; economic and financial conditions and developments (e.g. interest rate, inflation rate, etc.); monetary and fiscal policy; market expectations; and efficient market forces. Usually an increase in the interest rate causes the domestic currency to appreciate. A decrease in the interest rate causes the domestic currency to depreciate. An increase in inflation erodes the currency's purchasing power, causing it to depreciate. A decrease in inflation causes the domestic currency to appreciate. Exchange rates react to new information in an immediate and unbiased fashion, and since new information arrives randomly, exchange rates fluctuate randomly. For example, in the beginning of January of 1995, the Mexico encountered a monetary crisis, resulted in the sudden drop of US dollar versus other currencies owing to the relationship between Mexico and USA. Contrarily, in July of 1995, when the news of Russian president got ill came out, the exchange rate of US dollar versus other currencies rose in a large scale within a very short time. It has been recognized that the currencies rate changing is a complex and intersection process which can not be fully explained by any one factor or any limited set of explanatory variables.

1.4 Problem Motivation

Now that the financial data such as stock price, foreign exchange rates are always affected by an integrated process which includes many factors and it is difficult to directly analyze the financial data because what we really know is just the change of the market price without any idea what cause such change, by other words, the financial data can be regarded as the linear mixture of a set of independent components, a problem is proposed: whether we can use one kind of suitable ICA algorithm to separate these independent factors that influence or control the change of the financial data. In the following chapters, we will make some further study on this aspect.

1.5 Main Contribution of the Thesis

The main contribution of this thesis can be summarized as follows:

1. Firstly we sort the ICs according to their L_1 norm as shown in (Back and Weigend 1997), then we further expand this method as sorting them under L_p norm measurement. We develop a criterion to find out the appropriate number of dominant ICs under measurement of the MSE between the original data and the reconstructed data by adopting an idea given by Mr. Cheung Yiu-ming.

2. Following the suggestion of Prof. Xu, we also determine the dominant ICs order through measurement of the MSE between the original data and the reconstructed data. Because this is a discrete optimization problem and difficult to implement, we study a Forward Selection (FS) approach to sort the weighted ICs into a certain order according to their dominant value measured by MSE.

3. Sometimes MSE measurement is not suitable, similarly, we determine the domi-

nant ICs order under Tendency Error (TE) measurement between the original data and the reconstructed data. As this is also a discrete optimization problem, we study a Backward Elimination Tendency Error (BETE) approach to implement our criterion. We also develop a corresponding dominant ICs number determination criterion.

4. We have made large number of experiments to compare the performance of FS and BETE approaches based on our proposed criteria with some heuristic methods. Simulation results show that the dominant ICs obtained by these order-sorting approaches and number selection criteria are better than those heuristically obtained in the MSE and TE data reconstruction.

1.6 Other Contribution of the Thesis

1. We implement the learned parametric mixture based ICA (LPM) algorithm (Xu et al. 1998) to separate out the same number of independent components from eight foreign exchange rates.

2. On the basis of the original LPM algorithm, we propose two heuristic modified implementation algorithms to improve the convergence speed.

1.7 Organization of the Thesis

This thesis consists of six chapters

In Chapter 2 we firstly sort the ICs according to their L_1 norm as shown in (Back and Weigend 1997), then we further expand this method as sorting ICs by L_p norm measurement. We develop a dominant ICs number determination criterion under MSE measurement between the original data and the reconstructed data.

In Chapter 3 we determine the dominant ICs order under measurement of MSE between the original data and the reconstructed data. Based on this criterion, we study a Forward Selection (FS) approach (Lai et al. 1998a).

In Chapter 4 we determine the ICs order under measurement of Tendency Error (TE) between the original data and the reconstructed data. We implement this criterion by studying a Backward Elimination Tendency Error (BETE) (Lai et al. 1998b) approach. We also develop a corresponding dominant ICs number determination criterion.

In Chapter 5 we analyze the variance characteristics of the separated independent components and compare the reconstruction ability between PCA and ICA. Besides, we also study the autocorrelation and rescaled analysis on the independent components.

In Chapter 6 we make conclusion and look forward to the further work.

In Appendix We firstly give a brief review of selecting subsets from regression variables. From which we get some suggestions in proposing Forward Selection and Backward Elimination methods. On the other hand, we also give a systematically survey on unconstrained gradient based optimization algorithms. Then we introduce two heuristic modified implementation algorithms based on the original LPM algorithm (Xu et al. 1998). Comparison between those modified algorithms and original LPM algorithm and some other fixed nonlinearity ICA algorithms have been made.

Chapter 2

Heuristic Dominant ICs Sorting

2.1 L_1 Norm Sorting

Recently, independent component analysis has been successfully applied to analyze the stock price in Japan market (Back and Weigend 1997), where the observed price data are regarded as the linear mixture of a set of weighted independent components (WICs). In (Back and Weigend 1997), they use the daily closing price (from 1986 until 1989) of 28 largest firms as the observed data. After separate out the 28 independent components, they reconstruct the stock prices of the Bank of Tokyo-Mitsubishi, one of the largest bank in Japan, by using some so-called dominant ICs. In their paper, the dominant ICs order is determined by measuring the L_1 norm of the WICs, and the dominant ICs number is arbitrarily given.

The sorting procedure of L_1 Norm method is described as follows

Step 1

The L_1 norm of each WIC is computed as

$$N_{ij} = \sum_{t=1}^{N} |WIC_{ij}(t)| \tag{2.1}$$

where the k weighted ICs for the i^{th} financial data can be obtained by

$$WIC_{ij}(t) = \hat{A}_{ij}y_j(t),\ 1 \leq j \leq k. \tag{2.2}$$

with the mixing matrix A is estimated as $\hat{A} = W^{-1}$.

Step 2

Dominant ICs are sorted according to the descending order of the L_1 norm.

Apparently, by using this method, dominant ICs order is heuristically determined under the measurement of the weights of the ICs.

2.2 L_p Norm (L_3 Norm) Sorting

We can expand above mentioned L_1 norm sorting method to more general case (suggested by Mr. Cheung Yiu-ming), that is, we can sort the dominant ICs under the measurement of L_p norm ($p < \infty$). In order to make them easier to be compared, we randomly select $p = 3$ as an example (Actually we also have tried to use $p = 2$ and some other even numbers, we find that the simulation results are very similar with that of $p = 1$ under most of the cases, so here we only use $p = 3$ to represent odd number cases).

The sorting procedure of L_3 norm method is as follows

Step 1

$$N_{ij} = \sum_{t=1}^{N} |WIC_{ij}(t)|^3 \tag{2.3}$$

Step 2

Dominant ICs are sorted according to the descending order of the L_3 norm.

Of course, this is also a weight-determination heuristic sorting method.

2.3 Problem Motivation

Although the sorting of ICs and selecting of dominant ICs have many advantages such as

1. reveal some underlying structure in the data;
2. better financial model maybe set up by ignoring some non-dominant WICs, which may be some unexplained noise factors such as market expectation;

However, these existing methods will naturally arise two questions: (1) how to sort obtained WICs in a certain dominant order; (2) how to select the appropriate number of dominant ICs even if the dominant order of WICs is given.

2.4 Determination of Dominant ICs

In order to cope with the arbitrarily determination of the dominant ICs number, as shown in (Back and Weigend 1997), here we use the Mean Square Error (MSE) between the actual value and the reconstructed value as a cost function to select the suitable number of dominant ICs. Because the cost function monotonically decreases with the increasing of the number of dominant ICs, we can not find the suitable dominant ICs number in directly using this cost function. Here we define another cost function $J(m)$ as follows:

$$J(m) = Q(m) - Q(m-1), \quad m = 2, ..., n \tag{2.4}$$

with

$$\begin{aligned} Q(m) &= E[x_i - \hat{x}_i^m]^2 \\ &\approx \frac{1}{N}\sum_{t=1}^{N}[x_i(t) - \hat{x}_i^m(t)]^2, \text{ as } N \to \infty, \end{aligned} \tag{2.5}$$

where $\hat{x}_i^m(t)$ is reconstructed by the first m dominant ICs at time t as given in Section 4.6, and $m = 2, 3, \ldots, n$ is the candidate dominant ICs number.

The number selection criterion is that *the curve of cost function $J(m)$ versus m has a global minimum point at $m = m^*$, where m^* is the appropriate number of dominant components.*

Hence, in the following, as given a IC dominant order, we assume the first m^* ICs are dominant whereas the remaining $(n - m^*)$ are non-dominant.

2.5 ICA in Foreign Exchange Rate Markets

Here we use eight exchange rates (for each exchange rate, there are 1112 data points from November 26, 1991 to August 31, 1995), which are *US dollar versus German mark, Australian dollar, Canadian dollar, French franc, Swiss franc, British pound, Japanese yen and Hong Kong dollar* as shown in Figure 2.1. The ICA approach we used is the Learned Parametric Mixture Based ICA (LPM) Algorithm (Xu et al. 1998). Before we apply LPM in the separation of foreign exchange rate data, we have made large number of experiments to test the separation ability of high-dimensional source signals which include super-Gaussian (speech) signals and sub-Gaussian signals. The simulation results show that even for mixture of 10-channel source signals, the average signal-to-noise ratio can reach 30 (db), which means the source signals have been successfully separated. When we implement LPM, we firstly normalize x to fall into the range [-1,1] to remove the scaling factor. Such kind of normalization can alleviate the influence of some big shock such as that happened in the usd-cad exchange rate data (There is a big shock in the exchange rate of usd-cad, we have checked the data carefully, we think maybe this is an original printing error). The separated independent components are shown in Figure 2.2.

2.6 Comparison of Two Heuristic Methods

As given a set of independent components, we determine the dominant order of independent components based on WICs by two methods, respectively:

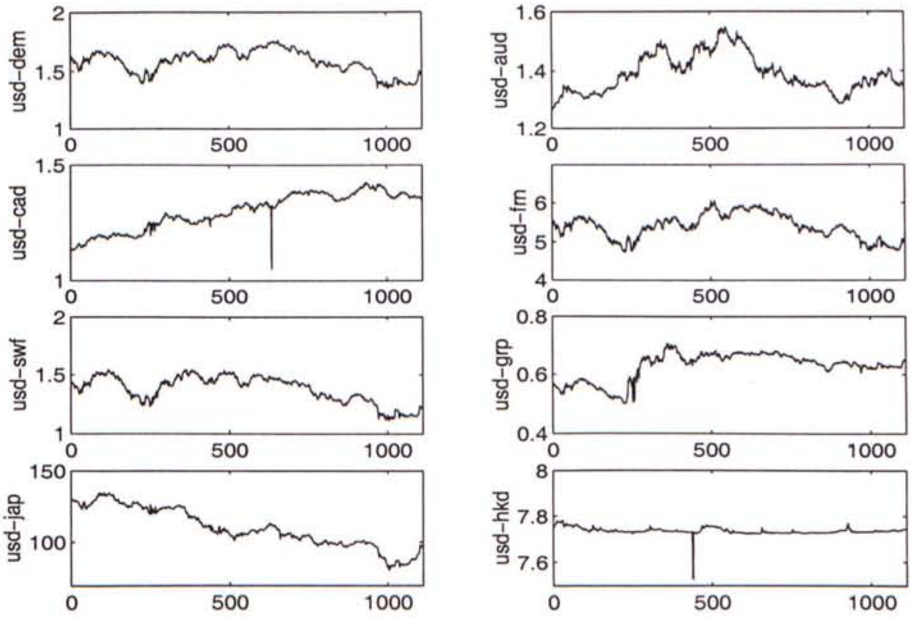


Figure 2.1: Original eight foreign exchange rates from November 26, 1991 and August 31, 1995

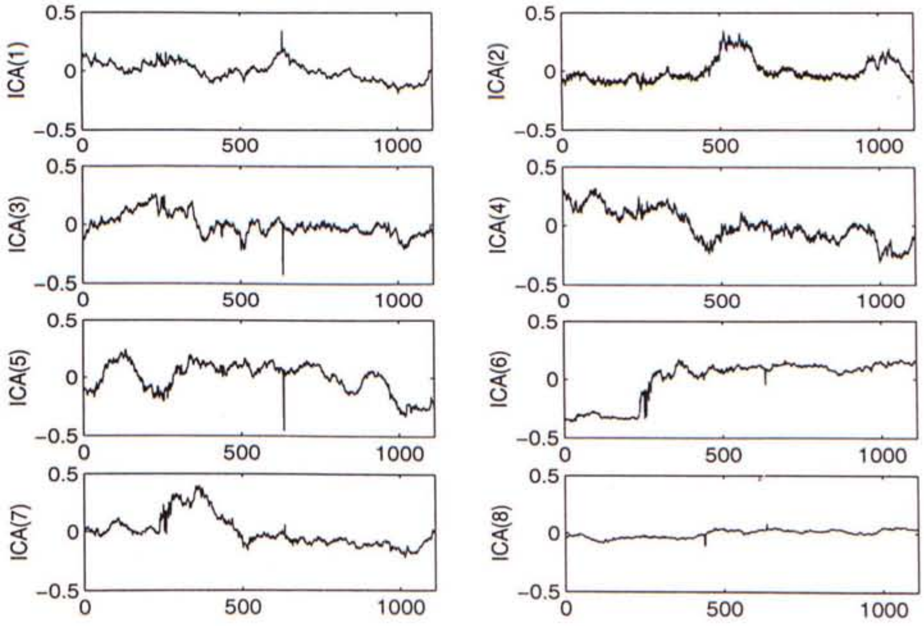


Figure 2.2: The eight separated independent components, where the label *ICA(i)* in y-axis means independent component

Method 1 L_1 Norm

Method 2 L_3 Norm [1]

We have made experiments to reconstruct four kinds of randomly selected foreign exchange rates by using above mentioned dominant ICs sorting Methods 1 and 2 respectively. Then we determine the suitable dominant ICs number through our proposed dominant Number-Determination criterion under orders got from these 2 methods.

2.6.1 Experiment 1: US Dollar vs Swiss Franc

In the following, we use USD-SWF exchange rate as an example to show the results under the measure of heuristic Methods 1 and 2.

In table 2.1, we demonstrate the procedure of sorting the ICs by L_1 norm and L_3 norm which corresponding to Method 1 and 2. We can get the orders of dominant ICs are [5,6,1,4,7,3,2,8] and [5,6,7,1,4,3,2,8] according to the measurements of Method 1 and 2 respectively.

The MSE corresponding to different dominant ICs number under measurement of Methods 1 and 2 are listed in table 2.2. The MSE curves of these three Methods are shown in the upper row of Figure 2.3. The relative cost function $J(m)$ curves are shown in the lower row of Figure 2.3.

We can see that the number m^* of dominant ICs is different under the different measure methods:

- Method 1: $m^* = 6$;
- Method 2: $m^* = 6$;

In this experiment, these two heuristic methods all select 6 dominant ICs out of total

[1] This method can be generalized into L_p norm with $p < \infty$.

Method 1		Method 2	
L_∞ Norm	Dominant Order	L_3 Norm	Dominant Order
59.6414	5	0.4380	5
31.9571	6	0.0713	6
25.8310	1	0.0438	7
25.0965	4	0.0384	1
24.1964	7	0.0342	4
20.6934	3	0.0287	3
8.9500	2	0.0024	2
4.1311	8	0.0001	8

Table 2.1: Norm measurement and corresponding dominant IC under Methods 1 and 2 (USD-SWF)

8 independent components. The MSE between the original data and the reconstructed data is very small.

No. of ICs Selected	Method 1	Method 2
1	0.0472	0.0472
2	0.0485	0.0485
3	0.0445	0.0434
4	0.0346	0.0430
5	0.0298	0.0298
6	0.0057	0.0057
7	0.0005	0.0005
8	0.0000	0.0000

Table 2.2: MSE between original signal and reconstruction signal measured by Method 1 and 2 under different dominant ICs number (USD-SWF)

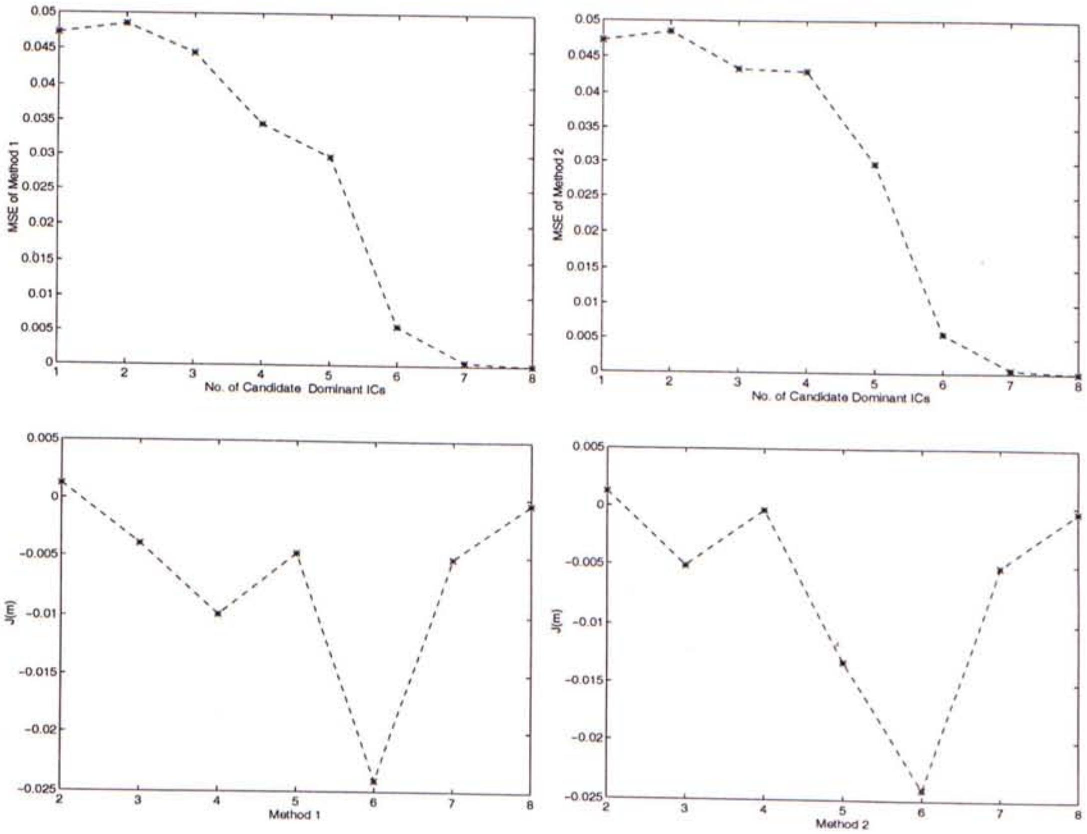


Figure 2.3: MSE under the measure of Methods 1 and 2 (upper row) and Curve of $J(m)$ vs. m under the measure of Methods 1 and 2 (lower row) (USD-SWF)

2.6.2 Experiment 2: US Dollar vs Australian Dollar

In the following, we use USD-AUD exchange rate as an example to show the results under the measure of two heuristic Methods 1 and 2.

In table 2.3, we demonstrate the procedure of sorting the ICs by L_1 norm and L_3 norm which corresponding to Method 1 and 2. We can get the orders of dominant ICs are [2,4,7,1,3,5,6,8] and [2,7,4,1,3,5,6,8] according to the measurements of Method 1 and 2 respectively.

The MSE corresponding to different dominant ICs number under measurement of Methods 1 and 2 are listed in table 2.4. The relative MSE curves of these two Methods are shown in the upper row of Figure 2.4. The relative cost function $J(m)$ curves are shown in the lower row of Figure 2.4.

We can see that the number m^* of dominant ICs is different under the different measure methods:

- Method 1: $m^* = 3$;
- Method 2: $m^* = 4$;

Method 1		Method 2	
L_∞ Norm	Dominant Order	L_3 Norm	Dominant Order
29.9634	2	0.0901	2
27.0032	4	0.0435	7
24.1512	7	0.0425	4
17.7055	1	0.0124	1
8.9158	3	0.0023	3
6.2641	5	0.0005	5
2.9960	6	0.0001	6
1.1557	8	0.0000	8

Table 2.3: Norm measurement and corresponding dominant IC under Methods 1 and 2 (USD-AUD)

No. of ICs Selected	Method 1	Method 2
1	0.1045	0.1045
2	0.1861	0.0445
3	0.0847	0.0847
4	0.0189	0.0189
5	0.0051	0.0051
6	0.0005	0.0005
7	0.0001	0.0001
8	0.0000	0.0000

Table 2.4: MSE between original signal and reconstruction signal measured by Method 1 and 2 under different dominant ICs number (USD-AUD)

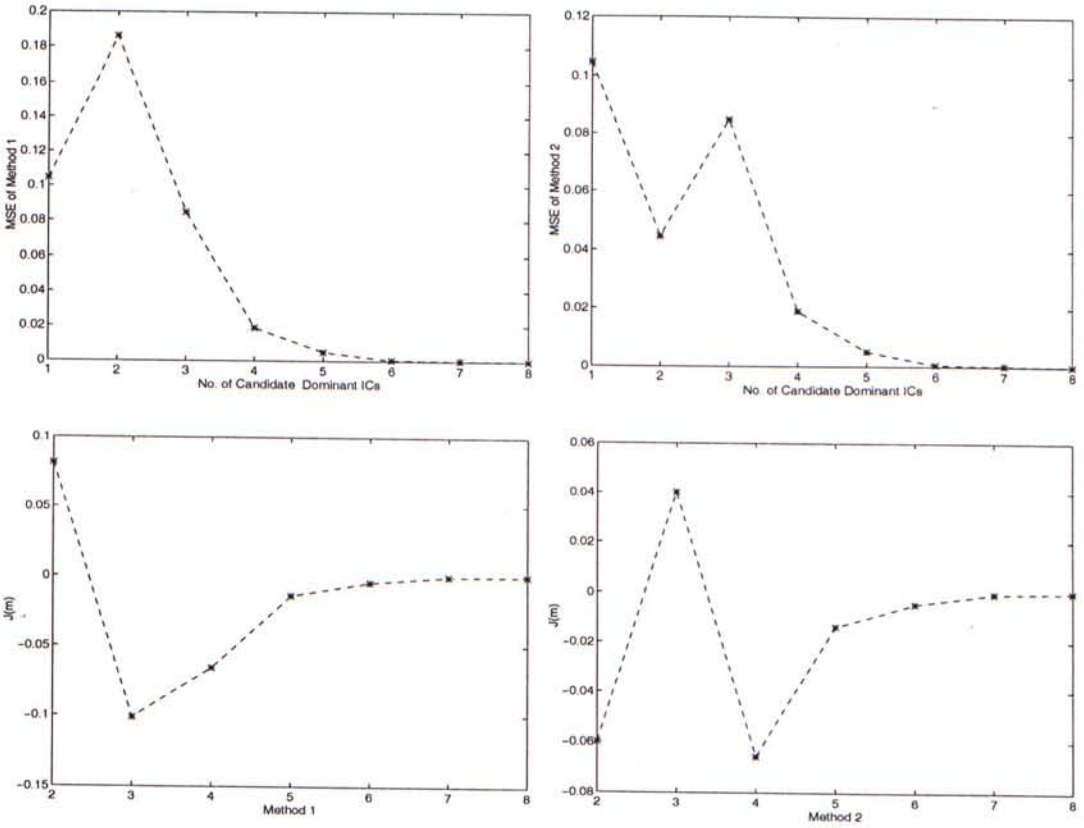


Figure 2.4: MSE under the measure of Methods 1 and 2 (upper row) and Curve of $J(m)$ vs. m under the measure of Methods 1 and 2 (lower row) (USD-AUD)

2.6.3 Experiment 3: US Dollar vs Canadian Dollar

In the following, we use USD-CAD exchange rate as an example to show the results under the measure of two heuristic Methods 1 and 2.

In table 2.5, we demonstrate the procedure of sorting the ICs by L_1 norm and L_3 norm which corresponding to Method 1 and 2. We can get the orders of dominant ICs are [6,7,4,1,3,2,5,8] and [6,7,4,3,1,2,5,8] according to the measurements of Method 1 and 2 respectively.

The MSE corresponding to different dominant ICs number under measurement of Methods 1 and 2 are listed in table 2.6. The relative MSE curves of these three Methods are shown in the upper row of Figure 2.5. The relative cost function $J(m)$ curves are shown in the lower row of Figure 2.5.

We can see that the number m^* of dominant ICs is different under the different measure methods:

- Method 1: $m^* = 5$;
- Method 2: $m^* = 5$;

These two heuristic methods all select 5 dominant ICs from total 8 independent components. We can find the MSE between the original data and the reconstructed data is less.

Method 1		Method 2	
L_∞ Norm	Dominant Order	L_3 Norm	Dominant Order
61.2565	6	0.5025	6
31.0887	7	0.0929	7
23.5736	4	0.0283	4
13.5378	1	0.0067	3
12.7090	3	0.0055	1
9.8218	2	0.0032	2
2.7104	5	0.0001	5
1.3656	8	0.0000	8

Table 2.5: Norm measurement and corresponding dominant IC under Methods 1 and 2 (USD-CAD)

No. of ICs Selected	Method 1	Method 2
1	0.0707	0.0707
2	0.0823	0.0823
3	0.0610	0.0610
4	0.0589	0.0424
5	0.0048	0.0048
6	0.0003	0.0003
7	0.0002	0.0002
8	0.0000	0.0000

Table 2.6: MSE between original signal and reconstruction signal measured by Method 1 and 2 under different dominant ICs number (USD-CAD)

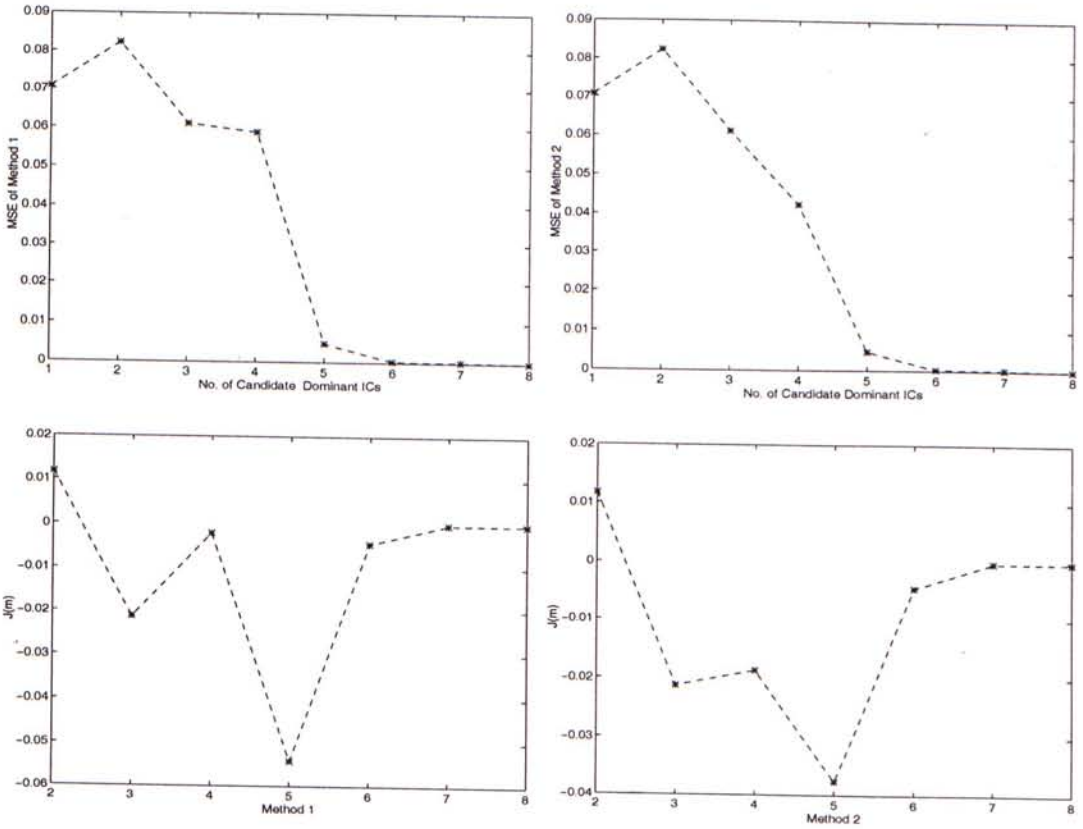


Figure 2.5: MSE under the measure of Methods 1 and 2 (upper row) and Curve of $J(m)$ vs. m under the measure of Methods 1 and 2 (lower row) (USD-CAD)

2.6.4 Experiment 4: US Dollar vs French Franc

In the following, we use USD-FRN exchange rate as an example to show the results under the measure of two heuristic Methods 1 and 2.

In table 2.7, we demonstrate the procedure of sorting the ICs by L_1 norm and L_3 norm which corresponding to Method 1 and 2. We can get the orders of dominant ICs are [5,1,3,2,4,8,7,6] and [5,1,3,2,4,8,7,6] according to the measurements of Method 1 and 2 respectively.

The MSE corresponding to different dominant ICs number under measurement of Methods 1 and 2 are listed in table 2.8. The relative MSE curves of these three Methods are shown in the upper row of Figure 2.6. The relative cost function $J(m)$ curves are shown in the lower row of Figure 2.6.

We can see that the number m^* of dominant ICs is different under the different measure methods:

- Method 1: $m^* = 3$;
- Method 2: $m^* = 3$;

Method 1		Method 2	
L_∞ Norm	Dominant Order	L_3 Norm	Dominant Order
41.6803	5	0.1495	5
25.5416	1	0.0371	1
18.5302	3	0.0206	3
6.2766	2	0.0008	2
5.2743	4	0.0003	4
4.6711	8	0.0002	7
4.0969	7	0.0001	8
0.5315	6	0.0000	6

Table 2.7: Norm measurement and corresponding dominant IC under Methods 1 and 2 (USD-FRN)

No. of ICs Selected	Method 1	Method 2
1	0.0930	0.0930
2	0.0848	0.0848
3	0.0269	0.0269
4	0.0140	0.0140
5	0.0060	0.0060
6	0.0018	0.0016
7	0.0000	0.0000
8	0.0000	0.0000

Table 2.8: MSE between original signal and reconstruction signal measured by Method 1 and 2 under different dominant ICs number (USD-FRN)

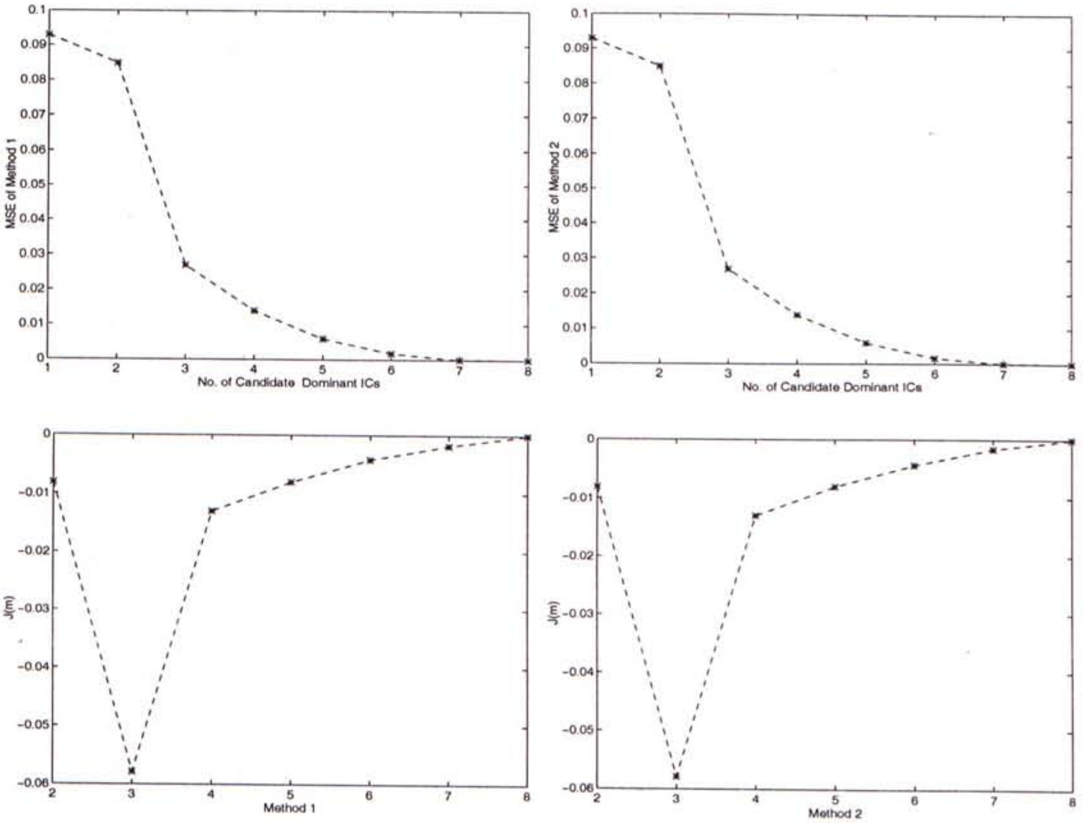


Figure 2.6: MSE under the measure of Methods 1 and 2 (upper row) and Curve of $J(m)$ vs. m under the measure of Methods 1 and 2 (lower row) (USD-FRN)

Chapter 3

Forward Selection under MSE Measurement

3.1 Order-Sorting Criterion

In the above two sections, we introduce two methods which are applied in sorting the dominant ICs. The basic idea is to determine the order by measuring the weight magnitude of the independent components, as shown in (Back and Weigend, 1997). As we can see, they are all heuristic sorting methods, and can not correctly determine the dominant ICs order because they can not control the reconstruction error between the original data and the reconstructed data. Under such cases, we propose an Order-Sorting criterion: *the dominant ICs order should be determined under measurement of the MSE between the original data and the reconstructed data.*

3.2 Order Sorting Approaches

Following above mentioned Order-Sorting criterion, large quantities of approaches can be applied to determine the dominant ICs order, within which the most basic one is the exhaustive-searching method. As we know, although this method can guarantee global minimum for the cost function defined as the reconstruction error between the original data and the reconstructed data, for high dimension data, it is usually very time consuming. Branch-and-bound is an optimized searching approach with some complexity and its searching efficiency highly depends on the data itself. Compared with some optimized but complex approaches, forward and backward searching are two simplified and more efficient methods, through which we can get some sub-optimal results.

3.3 Forward Selection Approach

In order to cope with the inaccuracy and arbitrariness of sorting dominant ICs by using some heuristic methods such as L_1 Norm and L_p Norm methods, also for simplification and fast implementation purpose, from various kinds of searching approaches we study a Forward Selection approach (Lai et al. 1998a) and define MSE as the reconstruction error between the original data and the reconstructed data. The detailed algorithm is described as follows

Step 1

Let $V = \{IC_{ij}\}_{j=1}^{k}$, and selection-order IC list $L = \{\}$

Step 2

we select that IC_{im_1} with

$$m_1 = \arg\min_j MSE\,(x_i - WIC_{ij}),\ \ 1 \leq j \leq k \tag{3.1}$$

as the first dominant IC. We let

$$L^{new} = L^{old} \cup \{\,IC_{im_1}\} \tag{3.2}$$

$$V^{new} = V^{old} - \{\,IC_{im_1}\} \tag{3.3}$$

Step 3

For each $IC_{ij} \in V$, we let

$$Z_{ij} = WIC_{ij} + WIC_L \tag{3.4}$$

where $WIC_L = \sum_{IC_{ij} \in L} WIC_{ij}$. We calculate the reconstruction MSE between x_i and Z_{ij}, then select the IC_{im_2} with

$$m_2 = \arg\min_j MSE\left(x_i - Z_{ij}\right) \tag{3.5}$$

as the second dominant IC. We let

$$\begin{aligned} L^{new} &= L^{old} \cup \{IC_{im_2}\} & (3.6) \\ V^{new} &= V^{old} - \{IC_{im_2}\} & (3.7) \end{aligned}$$

Step 4

Similar with Step 3, we can sort all the ICs in the list L with descending order under MSE measure.

3.4 Comparison of Three Dominant ICs Sorting Methods

In this section, as given a set of independent components, we determine the dominant order of independent components based on WICs by three methods, respectively:

Method 1 L_1 Norm, which is also used in (Back and Weigend 1997);

Method 2 L_3 Norm [1]

Method 3 Forward Selection (FS)

We have made experiments to reconstruct four kinds of randomly selected foreign exchange rates by using above mentioned dominant ICs sorting Methods 1, 2 and 3 respectively. Then we determine the suitable dominant ICs number through our proposed dominant Number-Selection criterion in last section.

[1] This method can be generalized into L_p norm with $p < \infty$.

After we implement Forward Selection method, we also try to apply Backward Elimination method in our experiment, but we find that the performance is inferior compared with that of using Forward Selection method, so in this chapter, we only use Forward Selection method on the separation of original foreign exchanger rate data.

3.4.1 Experiment 1: US Dollar vs Swiss Franc

In the following, we use USD-SWF exchange rate as an example to show the results under the measure of Forward Selection Method and two heuristic Methods 1 and 2. As introduced in last section, when we use the forward selection method starting from no ICs in the reconstructed signal, each time when we add ICs into the reconstructed signal, we should compute and compare the MSE under different choices and select the IC that corresponding to the smallest MSE until all the ICs have been added into the reconstructed signal or some criteria are satisfied.

The procedure of selecting the dominant ICs by Forward Selection Method is listed in table 3.1, from which we can see that the dominant order is [5,1,3,7,2,6,4,8].

The MSE corresponding to different dominant ICs number under measurement of Methods 1, 2 and 3 are listed in table 3.2. The MSE curves of these three Methods are shown in the upper row of Figure 3.1. The relative cost function $J(m)$ curves are shown in the lower row of Figure 3.1.

We can see that the number m^* of dominant ICs is different under the different measure methods:

- Method 1: $m^* = 6$;
- Method 2: $m^* = 6$;
- Method 3: $m^* = 2$;

In Figure 3.2, we use 2 dominant ICs respectively to reconstruct the USD-SWF data. The

No. of ICs Selected	MSE vs Independent Component								Selecting Order of ICs
1	0.1358	0.2745	0.2645	0.5822	0.0472	0.3415	0.1537	0.4318	5
2	0.0361	0.1144	0.0962	0.2894	0.1365	0.0594	0.2166	-	1
3	0.0728	0.0352	0.1845	0.1317	0.0530	0.1286	-	-	3
4	0.0989	0.2432	0.1028	0.0387	0.1593	-	-	-	7
5	0.0579	0.1242	0.0909	0.0900	-	-	-	-	2
6	0.1808	0.0611	0.1213	-	-	-	-	-	6
7	0.0584	0.0680	-	-	-	-	-	-	4
8	-	-	-	-	-	-	-	-	8

Table 3.1: Simulation results of sorting the dominant ICs by Forward Selection Method (USD-SWF)

No. of ICs Selected	Method 1	Method 2	Method 3
1	0.0472	0.0472	0.0472
2	0.0485	0.0485	0.0254
3	0.0445	0.0434	0.0192
4	0.0346	0.0430	0.0147
5	0.0298	0.0298	0.0136
6	0.0057	0.0057	0.0151
7	0.0005	0.0005	0.0005
8	0.0000	0.0000	0.0000

Table 3.2: MSE between original signal and reconstruction signal measured by Method 1, 2 and 3 under different dominant ICs number (USD-SWF)

MSE between the reconstructed data and the original data are 0.0485, 0.0485 and 0.0254 under Methods 1, 2 and 3 respectively. We can see that Method 3 is the best, which not only made the trend of reconstructed data similar with the original financial data, but also made the reconstruction of $n - m^*$ non-dominant ICs is unbiased. This implies that the major movements of financial data have been well controlled by the dominant ICs we determined.

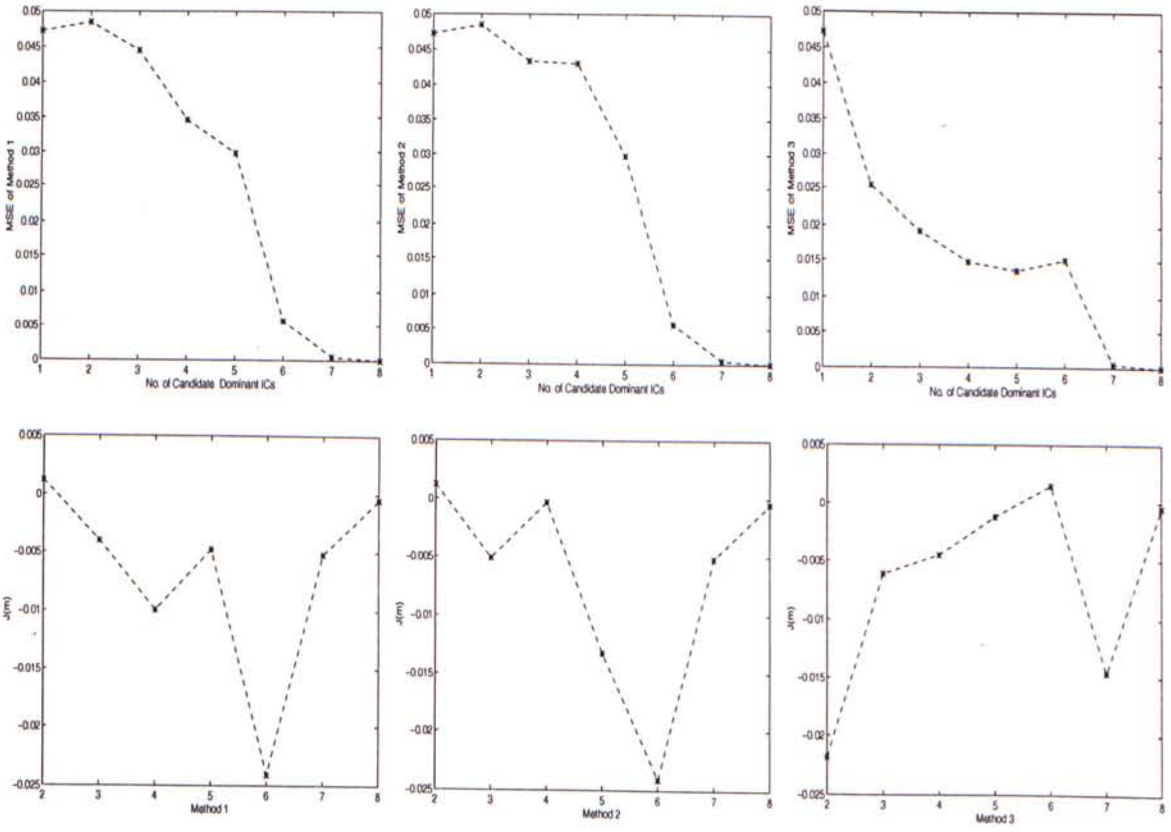
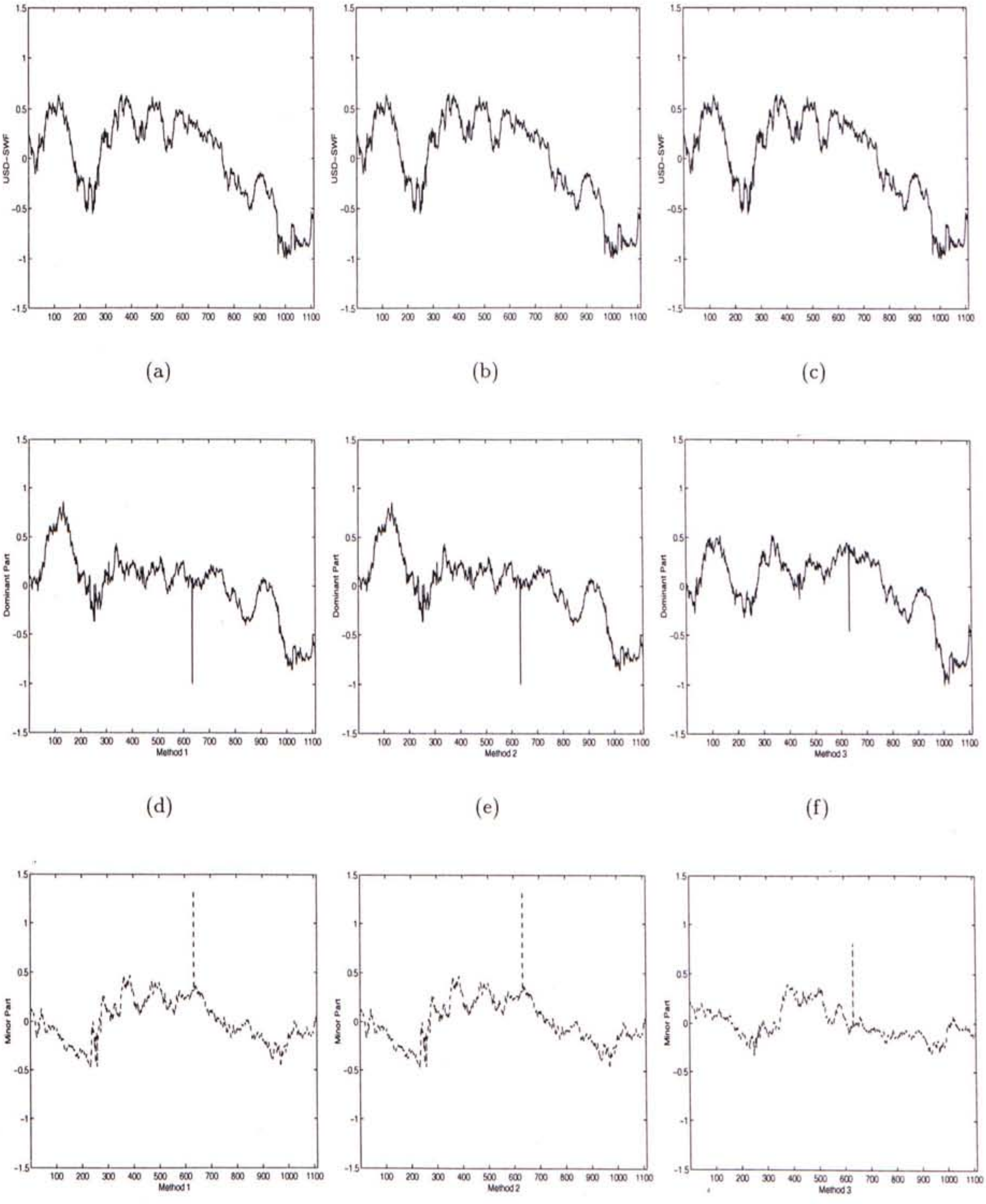


Figure 3.1: MSE under the measure of Methods 1, 2 and 3 (upper row) and Curve of $J(m)$ vs. m under the measure of Methods 1, 2 and 3 (lower row) (USD-SWF)



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 3.2: Normalized original USD-SWF data (upper low), the reconstructed signals (middle low) by using 2 dominant ICs determined by Method 1, 2 and 3 respectively from left to right, and the corresponding reconstructed signals by the left non-dominant ICs (lower row).

3.4.2 Experiment 2: US Dollar vs Australian Dollar

In the following, we use USD-AUD exchange rate as an example to show the results under the measure of Forward Selection Method and two heuristic Methods 1 and 2.

The procedure of selecting the dominant ICs by Forward Selection Method is listed in table 3.3, from which we can see that the dominant order is [2,7,1,4,3,5,6,8].

The MSE corresponding to different dominant ICs number under measurement of Methods 1, 2 and 3 are listed in table 3.4. The relative MSE curves of these three Methods are shown in the upper row of Figure 3.3. The relative cost function $J(m)$ curves are shown in the lower row of Figure 3.3.

We can see that the number m^* of dominant ICs is different under the different measure methods:

- Method 1: $m^* = 3$;
- Method 2: $m^* = 4$;
- Method 3: $m^* = 2$;

In Figure 3.4, we use 2 dominant ICs respectively to reconstruct the USD-SWF data. The MSE between the reconstructed data and the original data are 0.1861, 0.0445 and 0.0445 under Methods 1, 2 and 3 respectively. We can see that although the result of Method 3 is same as that of Method 2 here , but it is much better than Method 1 . By using Method 3, we find that which not only make the trend of reconstructed data similar with the original financial data, but also the reconstruction of $n - m^*$ non-dominant ICs keeps unbiased. We can see that the major movements of financial data have been well controlled by the dominant ICs we determined.

No. of ICs Selected	MSE vs Independent Component								Selecting Order of ICs
1	0.1863	0.1045	0.2110	0.3029	0.1759	0.2177	0.1977	0.3338	2
2	0.0880	0.0945	0.2398	0.0955	0.1732	0.0607	0.1748	-	7
3	0.8687	0.9955	1.0689	1.0335	0.9635	1.9160	-	-	1
4	0.1501	0.0334	0.1237	0.0593	0.2111	-	-	-	4
5	0.0298	0.0375	0.1021	0.0624	-	-	-	-	3
6	0.0417	0.0593	0.1123	-	-	-	-	-	5
7	0.0438	0.1284	-	-	-	-	-	-	6
8	-	-	-	-	-	-	-	-	8

Table 3.3: Simulation results of sorting the dominant ICs by Forward Selection Method (USD-AUD)

No. of ICs Selected	Method 1	Method 2	Method 3
1	0.1045	0.1045	0.1045
2	0.1861	0.0445	0.0445
3	0.0847	0.0847	0.0578
4	0.0189	0.0189	0.0189
5	0.0051	0.0051	0.0051
6	0.0005	0.0005	0.0005
7	0.0001	0.0001	0.0001
8	0.0000	0.0000	0.0000

Table 3.4: MSE between original signal and reconstruction signal measured by Method 1, 2 and 3 under different dominant ICs number (USD-AUD)

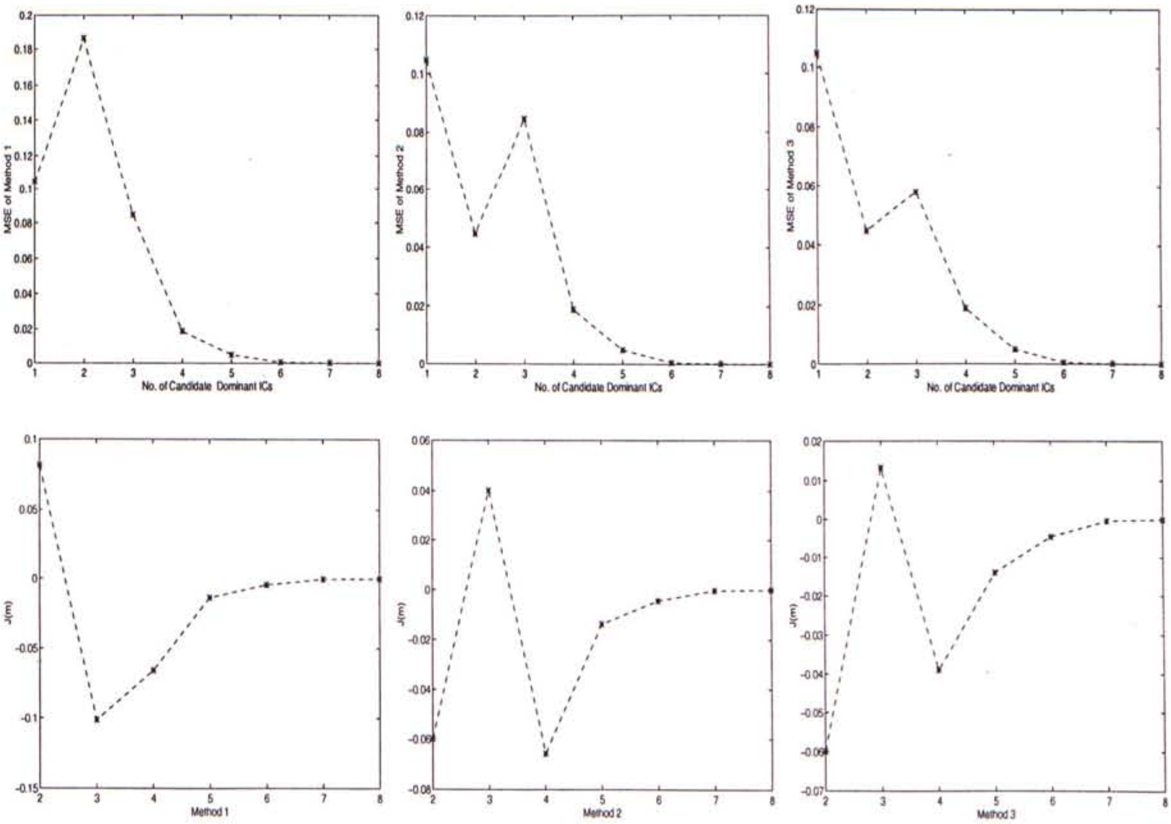
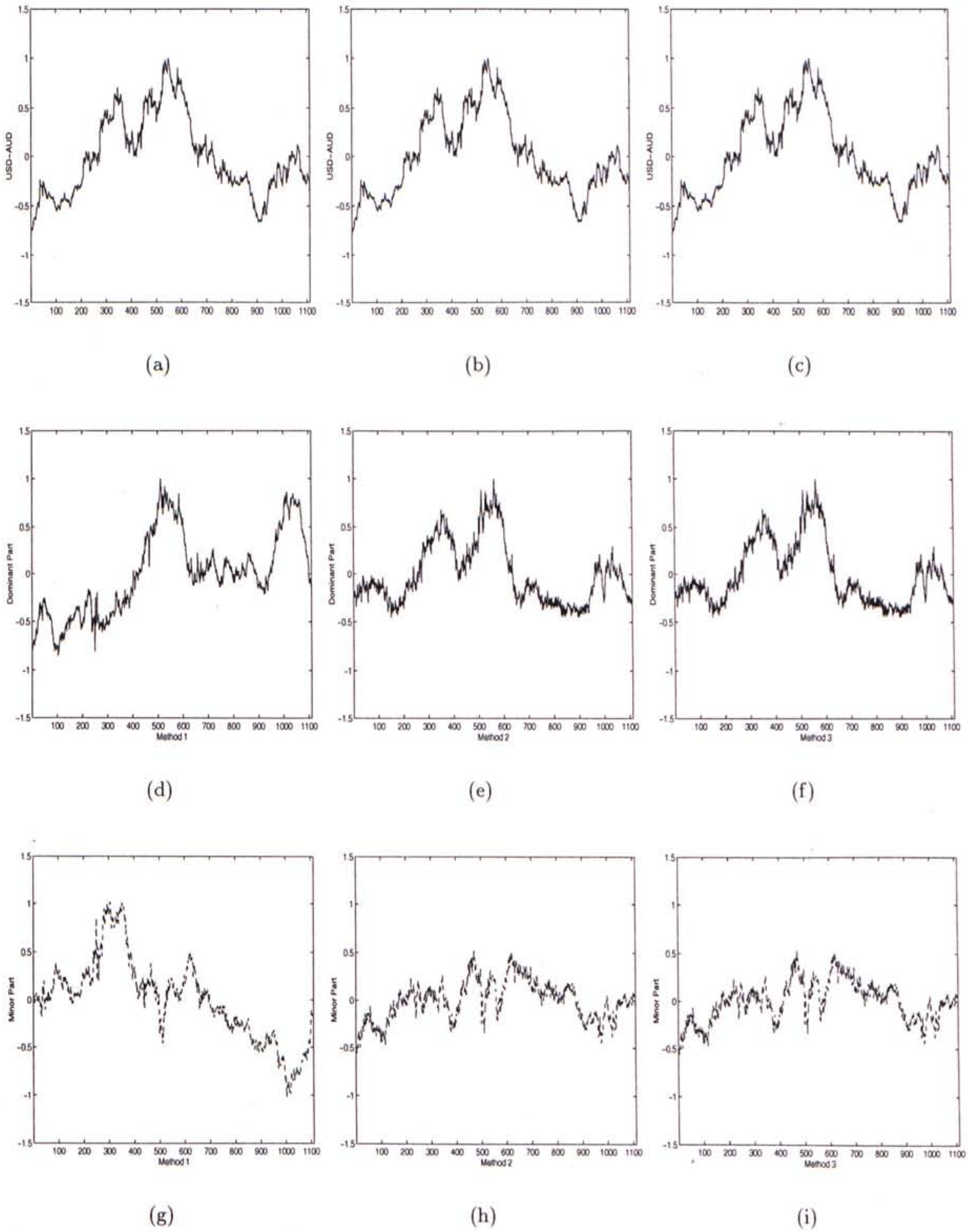


Figure 3.3: MSE under the measure of Methods 1, 2 and 3 (upper row) and Curve of $J(m)$ vs. m under the measure of Methods 1, 2 and 3 (lower row) (USD-AUD)



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 3.4: Normalized original USD-AUD data (upper row), the reconstructed signals (middle low) by using 2 dominant ICs determined by Method 1, 2 and 3 respectively from left to right, and the corresponding reconstructed signals by the left non-dominant ICs (lower row).

3.4.3 Experiment 3: US Dollar vs Canadian Dollar

In the following, we use USD-CAD exchange rate as an example to show the results under the measure of Forward Selection Method and two heuristic Methods 1 and 2.

The procedure of selecting the dominant ICs by Forward Selection Method is listed in table 3.5. From which we can see that the dominant order is [6,3,1,7,4,2,5,8].

The MSE corresponding to different dominant ICs number under measurement of Methods 1, 2 and 3 are listed in table 3.6. The relative MSE curves of these three Methods are shown in the upper row of Figure 3.5. The relative cost function $J(m)$ curves are shown in the lower row of Figure 3.5.

We can see that the number m^* of dominant ICs is different under the different measure methods:

- Method 1: $m^* = 5$;
- Method 2: $m^* = 5$;
- Method 3: $m^* = 2$;

In Figure 3.6, we use 2 dominant ICs respectively to reconstruct the USD-SWF data. The MSE between the reconstructed data and the original data are 0.0823, 0.0823 and 0.0435 under Methods 1, 2 and 3 respectively. We can see that Method 3 is the best. By using Method 3, we find that which not only the make trend of reconstructed data similar with the original financial data, but also the reconstruction of $n - m^*$ non-dominant ICs keeps unbiased. So by using Method 3, we can get most of the reconstruction of the original data with only suitable number of dominant ICs.

No. of ICs Selected	MSE vs Independent Component								Selecting Order of ICs
1	0.0734	0.2434	0.2049	0.4793	0.1480	0.0707	0.0983	0.3622	6
2	0.0563	0.0644	0.0435	0.0799	0.0741	0.0823	0.0703	-	3
3	0.0300	0.0481	0.0888	0.0433	0.0315	0.0441	-	-	1
4	0.0323	0.0405	0.0302	0.0258	0.0304	-	-	-	7
5	0.0201	0.0048	0.0291	0.0247	-	-	-	-	4
6	0.0003	0.0065	0.0041	-	-	-	-	-	2
7	0.0001	0.0006	-	-	-	-	-	-	5
8	-	-	-	-	-	-	-	-	8

Table 3.5: Simulation results of sorting the dominant ICs by Forward Selection Method (USD-CAD)

No. of ICs Selected	Method 1	Method 2	Method 3
1	0.0707	0.0707	0.0707
2	0.0823	0.0823	0.0435
3	0.0610	0.0610	0.0300
4	0.0589	0.0424	0.0258
5	0.0048	0.0048	0.0048
6	0.0003	0.0003	0.0003
7	0.0002	0.0002	0.0002
8	0.0000	0.0000	0.0000

Table 3.6: MSE between original signal and reconstruction signal measured by Method 1, 2 and 3 under different dominant ICs number (USD-CAD)

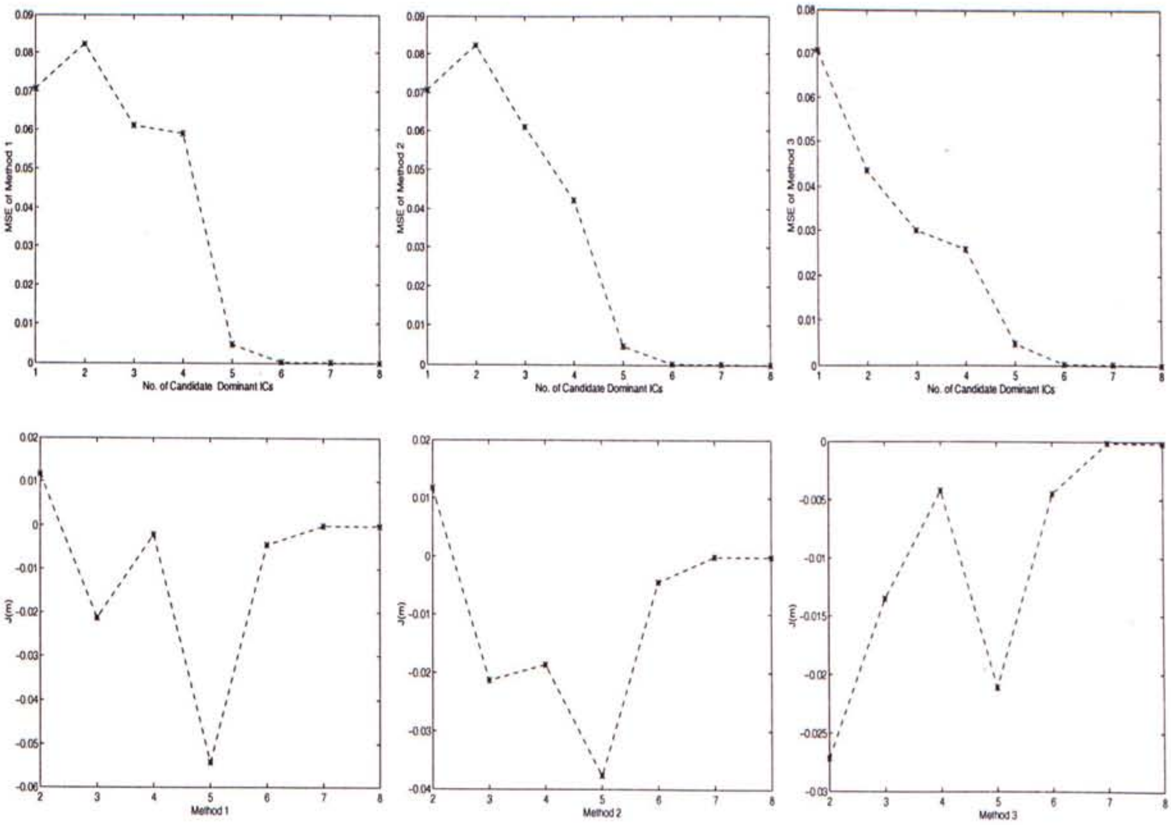
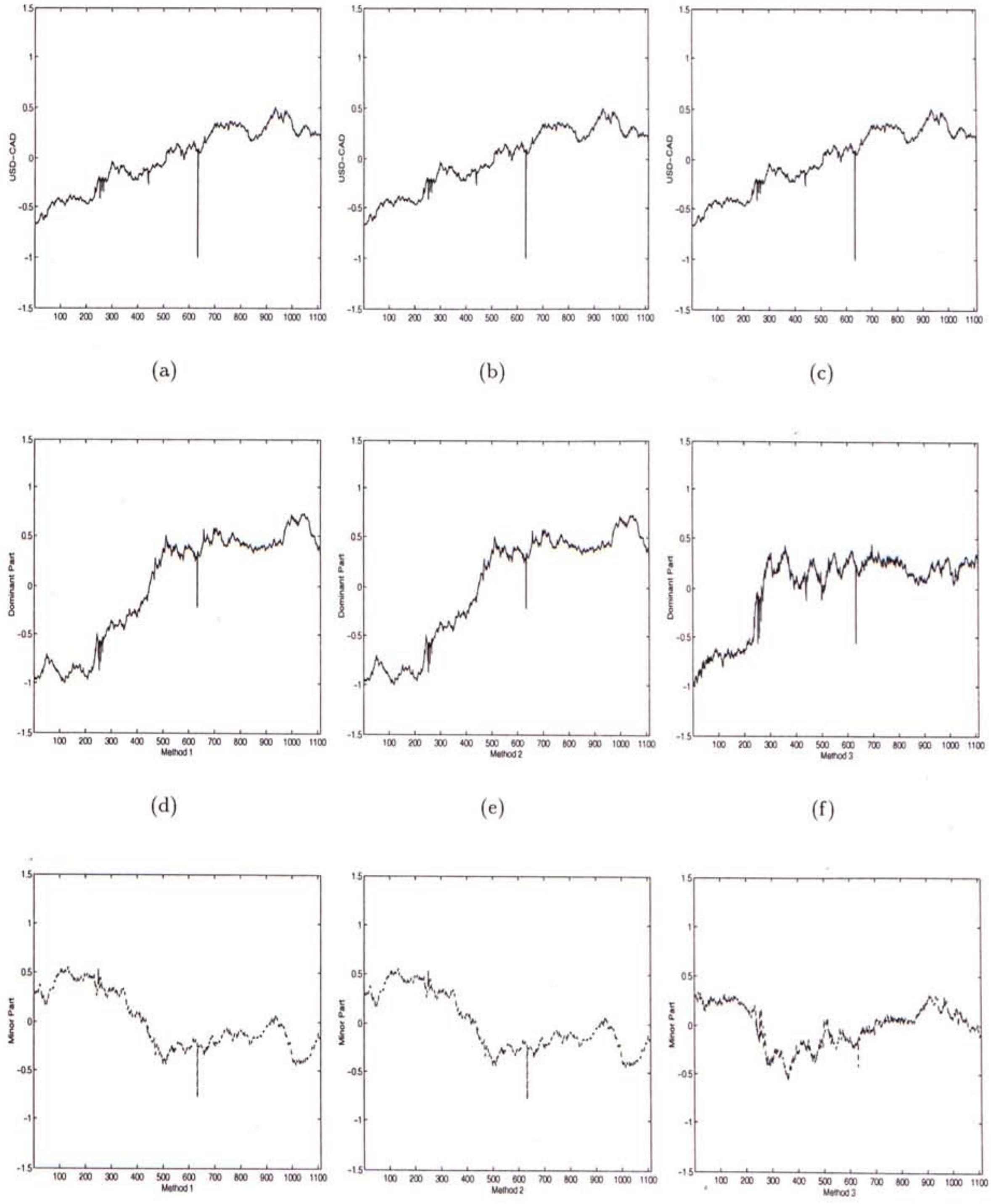


Figure 3.5: MSE under the measure of Methods 1, 2 and 3 (upper row) and Curve of $J(m)$ vs. m under the measure of Methods 1, 2 and 3 (lower row) (USD-CAD)



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 3.6: Normalized original USD-CAD data (upper row), the reconstructed signals (middle row) by using 2 dominant ICs determined by Method 1, 2 and 3 respectively from left to right, and the corresponding reconstructed signals by the left non-dominant ICs (lower row).

3.4.4 Experiment 4: US Dollar vs French Franc

In the following, we use USD-FRN exchange rate as an example to show the results under the measure of Forward Selection Method and two heuristic Methods 1 and 2.

The procedure of selecting the dominant ICs by Forward Selection Method is listed in table 3.7, from which we can see that the dominant order is [5,3,1,2,4,7,8,6].

The MSE corresponding to different dominant ICs number under measurement of Methods 1, 2 and 3 are listed in table 3.8. The relative MSE curves of these three Methods are shown in the upper row of Figure 3.7. The relative cost function $J(m)$ curves are shown in the lower row of Figure 3.7.

We can see that the number m^* of dominant ICs is different under the different measure methods:

- Method 1: $m^* = 3$;
- Method 2: $m^* = 3$;
- Method 3: $m^* = 2$;

In Figure 3.8, we use 2 dominant ICs respectively to reconstruct the USD-SWF data. The MSE between the reconstructed data and the original data are 0.0848, 0.0848 and 0.0532 under Methods 1, 2 and 3 respectively. We can see that Method 3 is the best, which not only make the trend of reconstructed data similar with the original financial data, but also make the reconstruction of $n - m^*$ non-dominant ICs keeps unbiased.

No. of ICs Selected	MSE vs Independent Component								Selecting Order of ICs
1	0.1930	0.2330	0.2412	0.5228	0.0930	0.5496	0.3803	0.3400	5
2	0.0848	0.0802	0.0532	0.0928	0.0937	0.0912	0.0807	-	3
3	0.0269	0.0535	0.0617	0.0529	0.0554	0.0519	-	-	1
4	0.0140	0.0169	0.0279	0.0191	0.0173	-	-	-	2
5	0.0060	0.0147	0.0077	0.0064	-	-	-	-	4
6	0.0064	0.0016	0.0018	-	-	-	-	-	7
7	0.0019	0.0001	-	-	-	-	-	-	8
8	-	-	-	-	-	-	-	-	6

Table 3.7: Simulation results of sorting the dominant ICs by Forward Selection Method (USD-FRN)

No. of ICs Selected	Method 1	Method 2	Method 3
1	0.0930	0.0930	0.0930
2	0.0848	0.0848	0.0532
3	0.0269	0.0269	0.0269
4	0.0140	0.0140	0.0140
5	0.0060	0.0060	0.0060
6	0.0018	0.0016	0.0016
7	0.0000	0.0000	0.0000
8	0.0000	0.0000	0.0000

Table 3.8: MSE between original signal and reconstruction signal measured by Method 1, 2 and 3 under different dominant ICs number (USD-FRN)

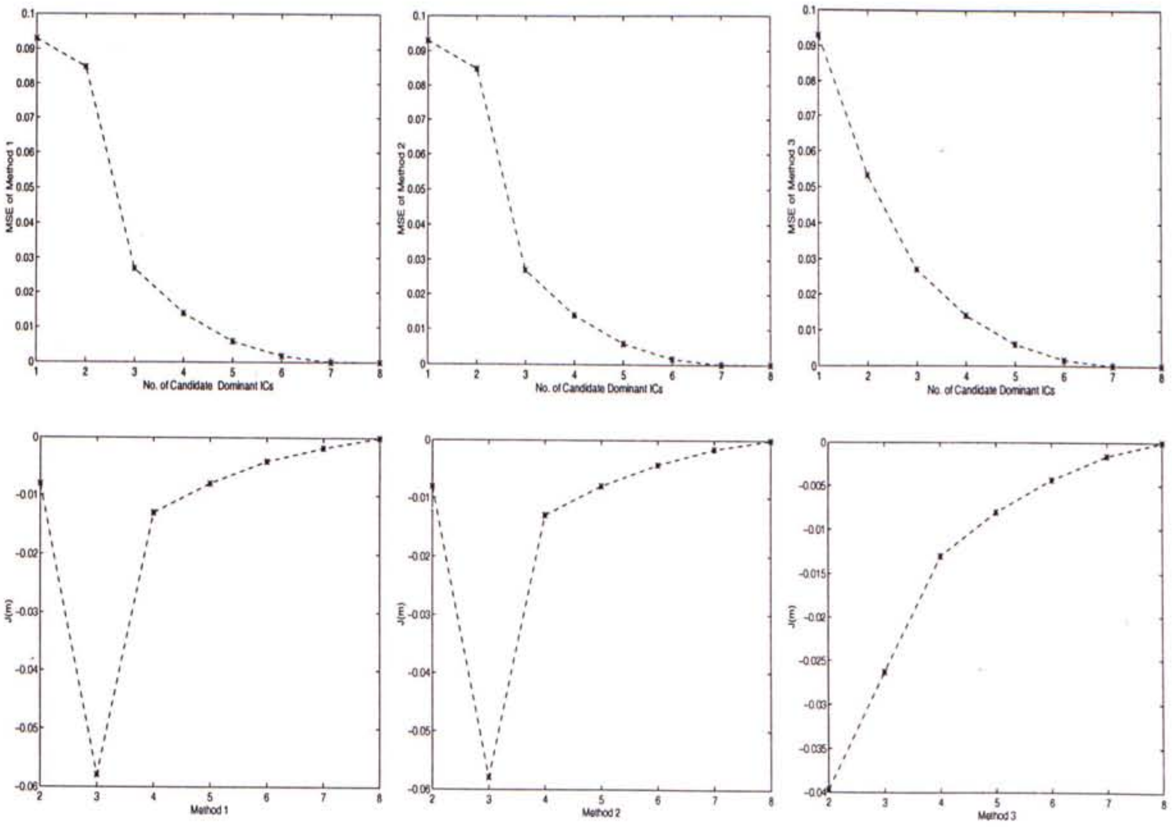
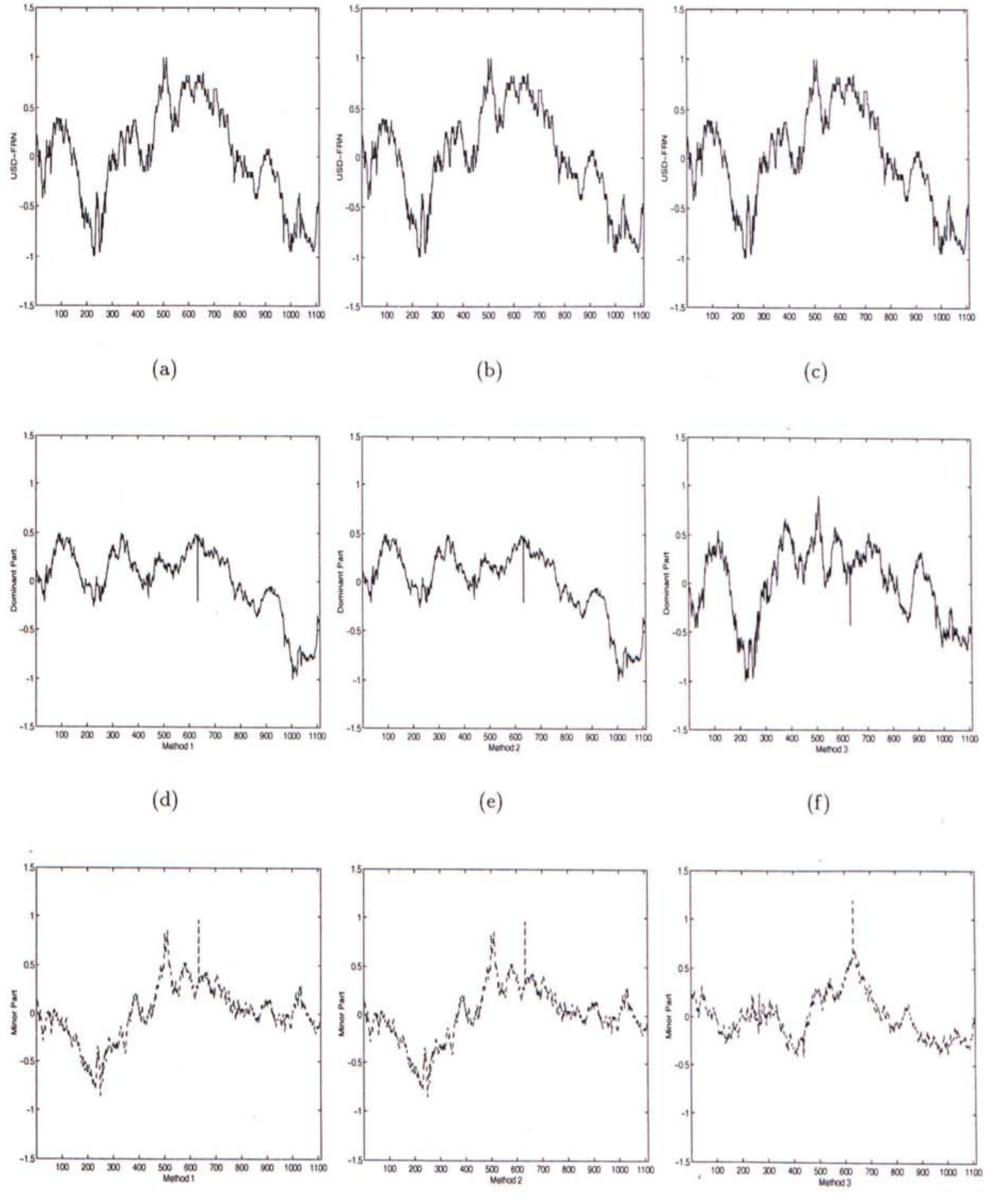


Figure 3.7: MSE under the measure of Methods 1, 2 and 3 (upper row) and Curve of $J(m)$ vs. m under the measure of Methods 1, 2 and 3 (lower row) (USD-FRN)



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 3.8: Normalized original USD-FRN data (upper row), the reconstructed signals (middle row) by using 2 dominant ICs determined by Method 1, 2 and 3 respectively from left to right, and the corresponding reconstructed signals by the left non-dominant ICs (lower row).

Chapter 4

Backward Elimination Tendency Error

4.1 Tendency Error Scheme

In the analysis of the financial markets, such as make prediction, usually the objective is to minimize the MSE between the original data and the reconstructed data, by other words, more less the MSE is, more better the reconstruction performance is. But we also often meet such situation, that is, people care more about the change tendency between the reconstructed return data and the original return data than care about the total MSE between them. Under such cases, MSE is not a suitable measurement criterion. In order to reflect the difference of the change tendency between the original data and the reconstructed data, here we define a concept of Tendency Error (TE) as follows.

Considering i^{th} financial data series $x_i(.)$ at time t, we define the returns of $x_i(.)$ as

$$\begin{cases} R_i(t) = 1 & if\ \ x_i(t) - x_i(t-1) > \delta \\ R_i(t) = 0 & if\ \ |x_i(t) - x_i(t-1)| < \delta \\ R_i(t) = -1 & if\ \ x_i(t) - x_i(t-1) < -\delta \end{cases} \tag{4.1}$$

The reconstructed return data $\widehat{R_i}(t)$ can be defined similarly.

$$\begin{cases} \widehat{R_i}(t) = 1 & if\ \widehat{x}_i(t) - \widehat{x}_i(t-1) > \delta \\ \widehat{R_i}(t) = 0 & if\ |\widehat{x}_i(t) - \widehat{x}_i(t-1)| < \delta \\ \widehat{R_i}(t) = -1 & if\ \widehat{x}_i(t) - \widehat{x}_i(t-1) < -\delta \end{cases} \tag{4.2}$$

here we set $\delta = 10^{-8}$. We also define $H_i = [h_i(2), h_i(3), \ldots, h_i(N)]$, with $h_i(t) = R_i(t) - \widehat{R_i}(t)$, the difference between the original return and the reconstructed return at t time step, n is the number of returns. We can calculate the Tendency Error (TE) between the original modified return and the reconstructed modified return as

$$TE_i = Number\ of\ Nonzero\ Elements\ in\ H_i \tag{4.3}$$

4.2 Order-Sorting Criterion

Apart from some heuristic methods which have been used to sort dominant ICs, in last chapter we have proposed a criterion which sorts the dominant ICs under MSE measurement. As we have already illustrated in last section, sometimes the Tendency Error between the original data and the reconstructed data is more important for analysts or investors. Under such cases, we propose another Order-Sorting criterion: *the dominant ICs order should be determined under measurement of the Tendency Error between the original data and the reconstructed data.*

4.3 Order Sorting Approaches

Following the previously introduced Order-Sorting criterion, here we focus on the return data and define the Tendency Error as the reconstruction error between the original signal and the reconstructed signal in sorting the WICs with the result that the dominant WICs can mostly reflect the change tendency of the original data. Similarly as introduced in last chapter, we also can use many order sorting approaches such as exhaustive searching approach, which is very time-consuming; branch-and-bound approach, which is an optimized searching method, but its searching speed highly depends on the data; forward and backward searching approaches, etc. Within these approaches, forward searching and backward searching approaches are two simple and efficient methods, so even if by using these methods, usually we only can get sub-optimal results, they are widely used in many aspects.

4.4 Backward Elimination Tendency Error Approach

From our experiment results of separately using Forward Selection and Backward Elimination methods (for simplification and fast implementation purpose, here we only compare these two methods), we find that for return data, the latter one is more robust, sometimes it performs better than the former one. So, under the aim of minimizing the Tendency Error between the original return data and the reconstructed return data, we introduce a so-called Backward Elimination Tendency Error (BETE) approach (Lai et al. 1998b) The detailed algorithm is described as follows

Step 1

Let $V = \{IC_{ij}\}_{j=1}^{k}$, and elimination-order IC list $L = \{\}$.

Step 2

For each $IC_{ij} \in V$, we let $Z_{ij} = x_i - WIC_{ij}$, and delete that IC_{im_1} with

$$m_1 = \arg\min_{j} TE\,(x_i - Z_{ij}),\;\; 1 \leq j \leq k \tag{4.4}$$

as the first non-dominant IC. We let

$$\begin{aligned} L^{new} &= L^{old} \cup \{\, IC_{im_1}\} && (4.5)\\ V^{new} &= V^{old} - \{\, IC_{im_1}\} && (4.6) \end{aligned}$$

Step 3

For each $IC_{ij} \in V$, we let

$$Z_{ij} = x_i - (WIC_{ij} + WIC_L) \tag{4.7}$$

where $WIC_L = \sum_{IC_{ij} \in L} WIC_{ij}$. We calculate the TE between x_i and Z_{ij}, and delete the IC_{im_2} with

$$m_2 = \arg\min_j TE\,(x_i - Z_{ij}) \tag{4.8}$$

as the second non-dominant IC. We let

$$\begin{aligned} L^{new} &= L^{old} \cup \{\, IC_{im_2}\} && (4.9) \\ V^{new} &= V^{old} - \{\, IC_{im_2}\} && (4.10) \end{aligned}$$

Step 4

Similar with Step 3, we can sort all the ICs in the list L with ascending order under TE measure.

4.5 Determination of Dominant ICs

Corresponding to BETE method , we use the Tendency error (TE) by using dominant ICs as the cost function to select the suitable number of dominant ICs. The cost function $J(m)$ is defined as follows:

$$J(m) = TE(m) - TE(m-1), \quad m = 2, ..., n \tag{4.11}$$

Where $TE(m)$ is the number of nonzero elements in the reconstrtuced return by using m dominant ICs, by another word, it is the Tendency Error between the original return data and the reconstructed return data of using m dominant ICs.

The number selection criterion is that *the curve of cost function $J(m)$ versus m has a global minimum point at $m = m^*$, where m^* is the appropriate number of dominant components.*

Hence, in the following, as given a IC dominant order, we assume the first m^* ICs are dominant whereas the remaining $(n - m^*)$ are non-dominant.

4.6 Comparison Between Three Approaches

In this section, we have made experiments to reconstruct four kinds of foreign exchange rates. The dominant ICs are sorted by using under mentioned Methods 1,2 and 4 respectively, then we determine the suitable dominant ICs number through our proposed dominant number selection criterion under orders got from these 3 methods.

As given a set of independent components, we determine the dominant order of independent components based on WICs by three ways, respectively:

Method 1 L_1 Norm

Method 2 L_3 Norm [1]

Method 4 Backward Elimination Tendency Error (BETE)

4.6.1 Experiment Results on USD-SWF Return

We have made experiments to reconstruct the USD-SWF return data by using BETE method to gradually eliminate the IC which is relatively not dominant in controlling the change tendency of the original data. Table 4.1 is the simulation results in selecting the eliminated ICs gradually. The order of sorting the dominant ICs (contrary to the elimination order) is [5,1,3,4,2,7,6,8].

[1] This method can be generalized into L_p norm with $p < \infty$.

No. of ICs Deleted	TE vs Independent Component								Elimination Order of IC
1	255	163	283	232	566	113	203	91	8
2	256	163	283	232	566	113	203	-	6
3	251	192	288	212	544	146	-	-	7
4	294	136	289	287	576	-	-	-	2
5	257	321	173	572	-	-	-	-	4
6	251	224	514	-	-	-	-	-	3
7	262	550	-	-	-	-	-	-	1
8	-	-	-	-	-	-	-	-	5

Table 4.1: Simulation results of sorting dominant ICs by BETE method (USD-SWF)

The TE values corresponding to different dominant ICs number under measurement of Methods 1, 2 and 4 are listed in table 4.2.

The MSE value curves under Method 1, 2 and TE values under BETE Method are shown in the upper row of Figure 4.1. The relative cost function $J(m)$ curves are demonstrated in the lower row of Figure 4.1.

- Method 1: $m^* = 6$;
- Method 2: $m^* = 6$;
- Method 4: $m^* = 3$;

In Figure 4.2, for each method, we all use 3 dominant ICs to reconstruct the original USD-SWF return data. Experiment results show the BETE method performs better than other two methods, it can reconstruct most of the change tendency of the original data. The TE value under this case is 247, 269 and 173 for method 1, 2 and BETE respectively. This result is in conformity with our observation in Figure 4.2.

No. of ICs Selected	Method 1	Method 2	Method 4
1	262	262	262
2	289	289	224
3	247	269	173
4	336	239	136
5	325	325	146
6	145	145	101
7	91	91	91
8	44	45	44

Table 4.2: TE between original signal and reconstruction signal measured by Method 1, 2 and 4 under different dominant ICs number (USD-SWF)

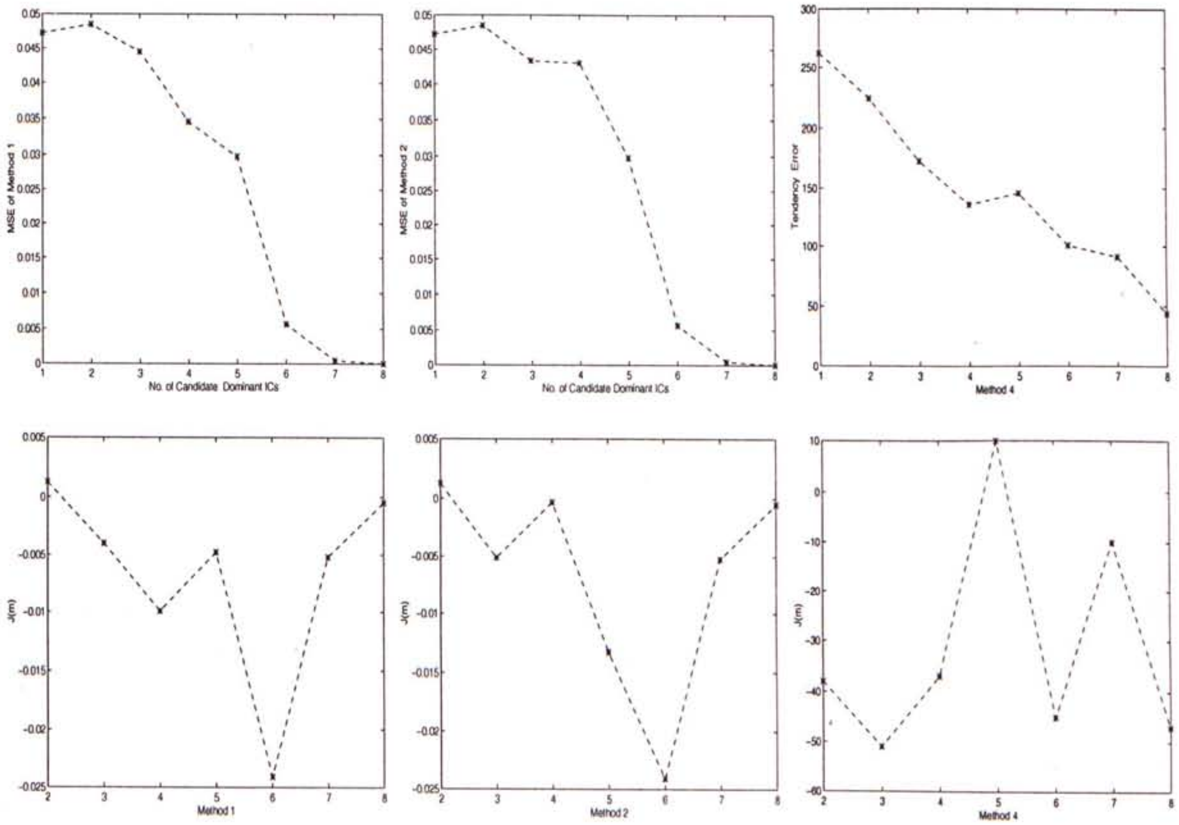
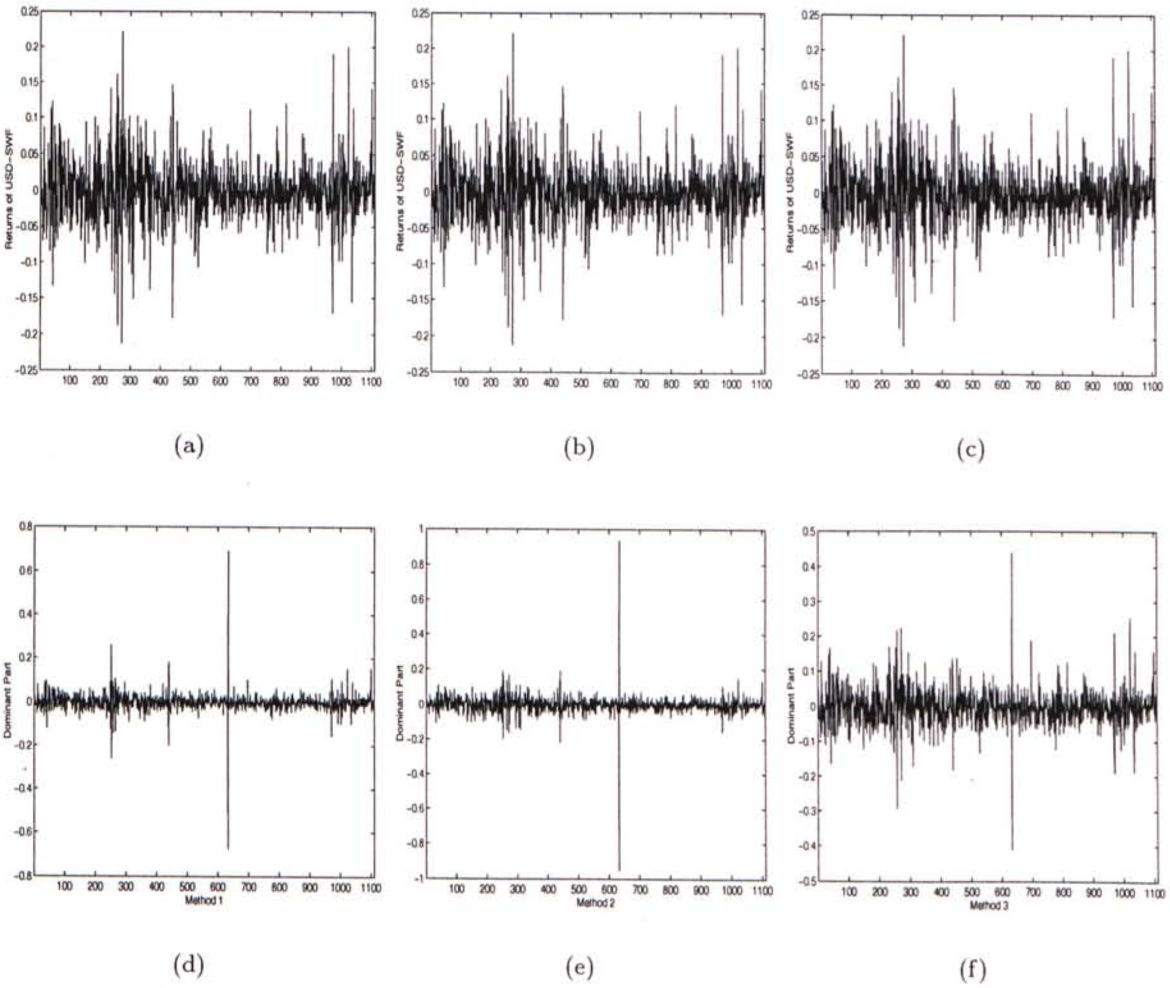


Figure 4.1: MSE under the measure of Methods 1, 2 and TE under Method 4 (upper row) and Curve of $J(m)$ vs. m under the measure of Methods 1, 2 and 4 (lower row) (USD-SWF)



(a) (b) (c)

(d) (e) (f)

Figure 4.2: Return of usd-swf data (upper low), the reconstructed signals (lower row) by using the m^* dominant ICs determined by Method 1, 2 and 4 respectively from left to right.

4.6.2 Experiment Results on USD-AUD Return

We have made experiments to reconstruct the USD-AUD return data by using BETE method to gradually eliminate the IC which is relatively not dominant in controlling the change tendency of the original data. Table 4.3 is the simulation results in selecting the eliminated ICs gradually. The order of sorting the dominant ICs (contrary to the elimination order) is [2,4,7,1,3,5,6,8].

The TE values corresponding to different dominant ICs number under measurement of Methods 1, 2 and 4 are listed in table 4.4.

The MSE value curves under Method 1, 2 and TE values under BETE Method are shown in the upper row of Figure 4.3. The relative cost function $J(m)$ curves are demonstrated in the lower row of Figure 4.3.

- Method 1: $m^* = 3$;
- Method 2: $m^* = 4$;
- Method 4: $m^* = 5$;

In Figure 4.4, for each method, we use m^* dominant ICs to reconstruct the original USD-SWF return data. According to the different dominant ICs number selected through methods 1, 2 and 4, the Tendency Error we get are 397, 304 and 184 respectively, which shows that the BETE method performs better than other two methods. By using the suitable dominant ICs selected from the BETE method, we can reconstruct most of the change tendency of the original return data.

No. of ICs Deleted	TE vs Independent Component								Elimination Order of ICs
1	322	492	285	419	176	109	318	100	8
2	322	492	285	420	176	109	318	-	6
3	321	497	282	420	184	327	-	-	5
4	317	501	304	418	343	-	-	-	3
5	397	523	477	412	-	-	-	-	1
6	546	507	464	-	-	-	-	-	7
7	552	529	-	-	-	-	-	-	4
8	-	-	-	-	-	-	-	-	2

Table 4.3: Simulation results of sorting dominant ICs by BETE method (USD-AUD)

No. of ICs Selected	Method 1	Method 2	Method 4
1	529	529	529
2	464	507	464
3	397	397	397
4	304	304	304
5	184	184	184
6	108	108	108
7	100	100	100
8	72	68	72

Table 4.4: TE between original signal and reconstruction signal measured by Method 1, 2 and 3 under different dominant ICs number (USD-AUD)

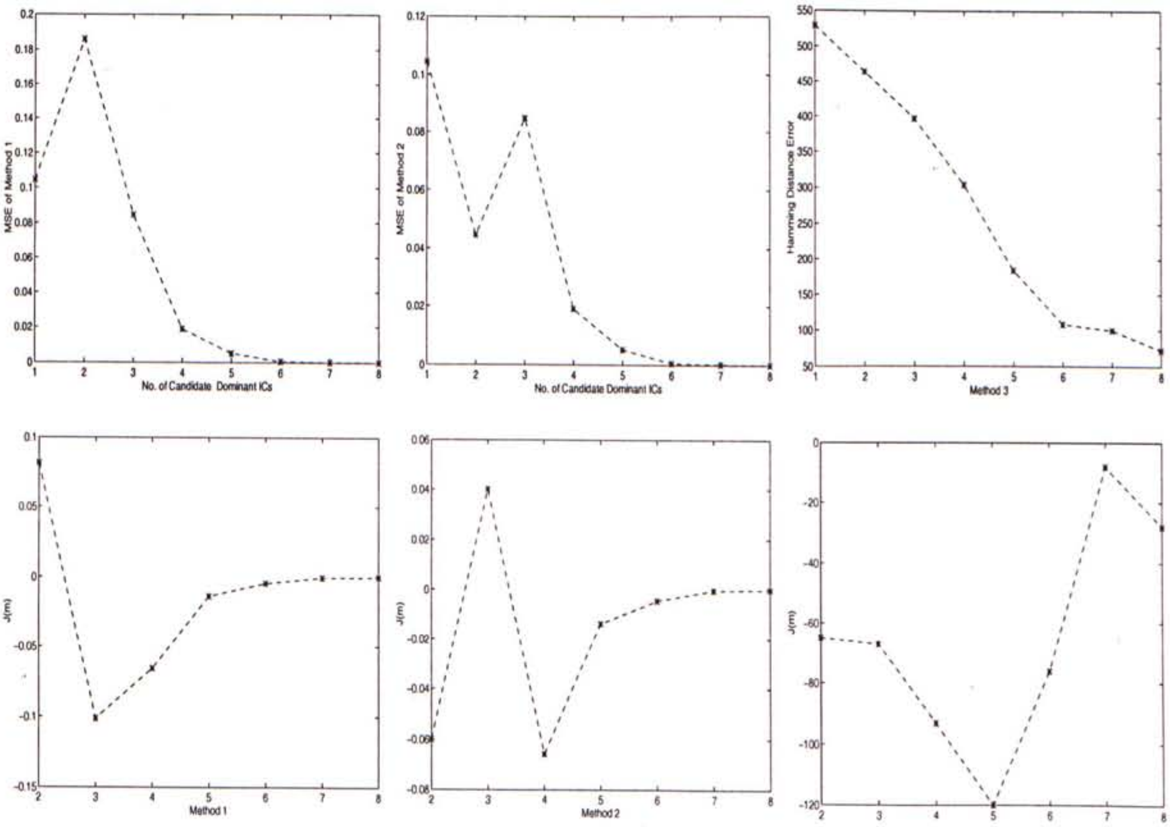
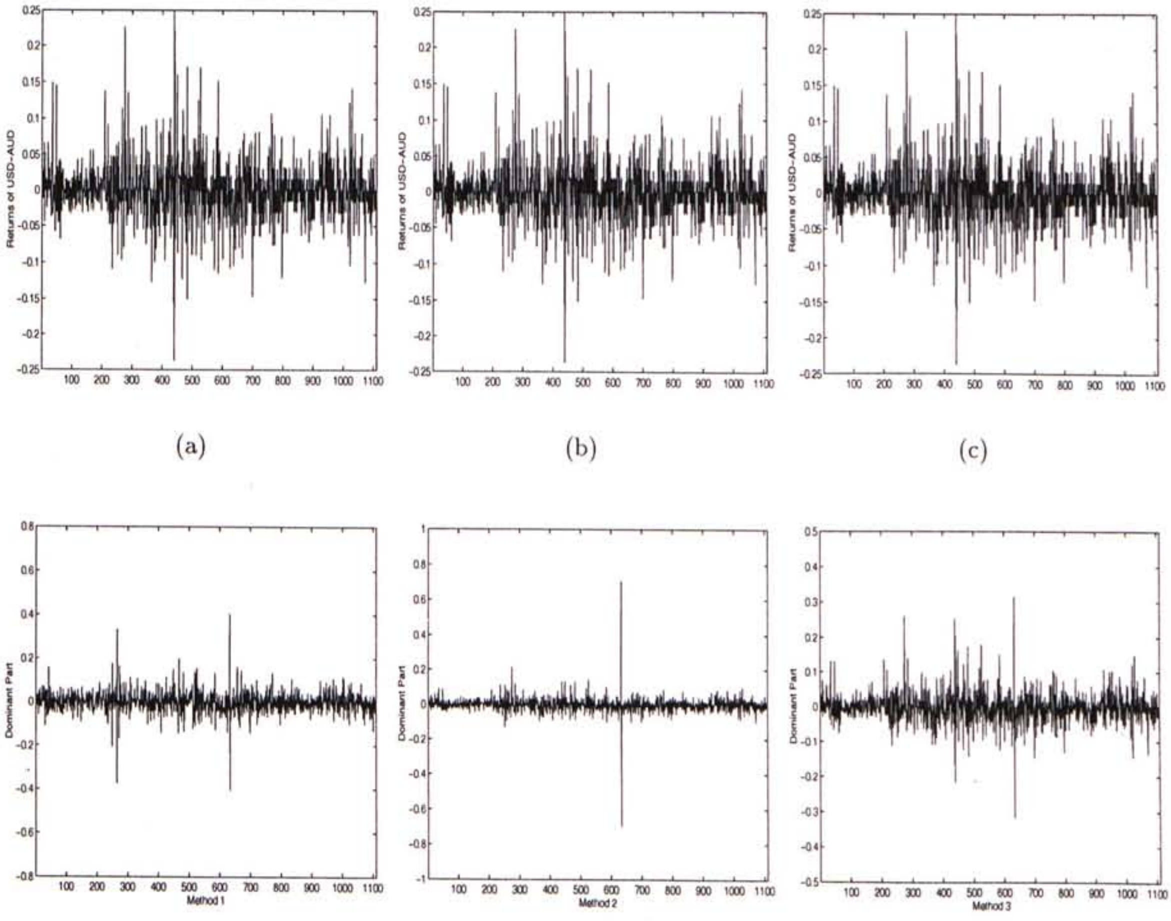


Figure 4.3: MSE under the measure of Methods 1, 2 and TE under Method 4 (upper row) and Curve of $J(m)$ vs. m under the measure of Methods 1, 2 and 4 (lower row) (USD-AUD)



(a)

(b)

(c)

(d)

(e)

(f)

Figure 4.4: Return of USD-AUD data (upper row), the reconstructed signals (lower row) by using the m^* dominant ICs determined by Method 1, 2 and 4 respectively from left to right.

4.6.3 Experiment Results on USD-CAD Return

We have made experiments to reconstruct the USD-CAD return data by using BETE method to gradually eliminate the IC which is relatively not dominant in controlling the change tendency of the original data. Table 4.5 is the simulation results in selecting the eliminated ICs gradually. The order of sorting the dominant ICs (contrary to the elimination order) is [3,4,6,7,2,1,5,8].

The TE values corresponding to different dominant ICs number under measurement of Methods 1, 2 and 4 are listed in table 4.6.

The MSE value curves under Method 1, 2 and TE values under BETE Method are shown in the upper row of Figure 4.5. The relative cost function $J(m)$ curves are demonstrated in the lower row of Figure 4.5.

- Method 1: $m^* = 5$;
- Method 2: $m^* = 5$;
- Method 4: $m^* = 6$;

In Figure 4.6, for each method, we all use m^* dominant ICs to reconstruct the original USD-SWF return data. According to the different dominant ICs number selected through methods 1, 2 and 4, the Tendency Error between the reconstructed return data and the original data are 346, 346 and 155 respectively, which shows that the BETE method performs better than other two methods, it can reconstruct most of the change tendency of the original data.

No. of ICs Deleted	TE vs Independent Component								Elimination Order of ICs
1	337	356	449	440	155	446	420	146	8
2	338	356	449	440	155	447	420	-	5
3	328	346	421	436	428	412	-	-	1
4	373	468	465	478	462	-	-	-	2
5	510	410	518	409	-	-	-	-	7
6	520	483	462	-	-	-	-	-	6
7	584	487	-	-	-	-	-	-	4
8	-	-	-	-	-	-	-	-	3

Table 4.5: Simulation results of sorting dominant ICs by BETE method (USD-CAD)

No. of ICs Selected	Method 1	Method 2	Method 4
1	510	510	487
2	468	468	462
3	510	510	409
4	499	373	373
5	346	346	328
6	155	155	155
7	146	146	146
8	98	98	91

Table 4.6: TE between original signal and reconstruction signal measured by Method 1, 2 and 4 under different dominant ICs number (USD-CAD)

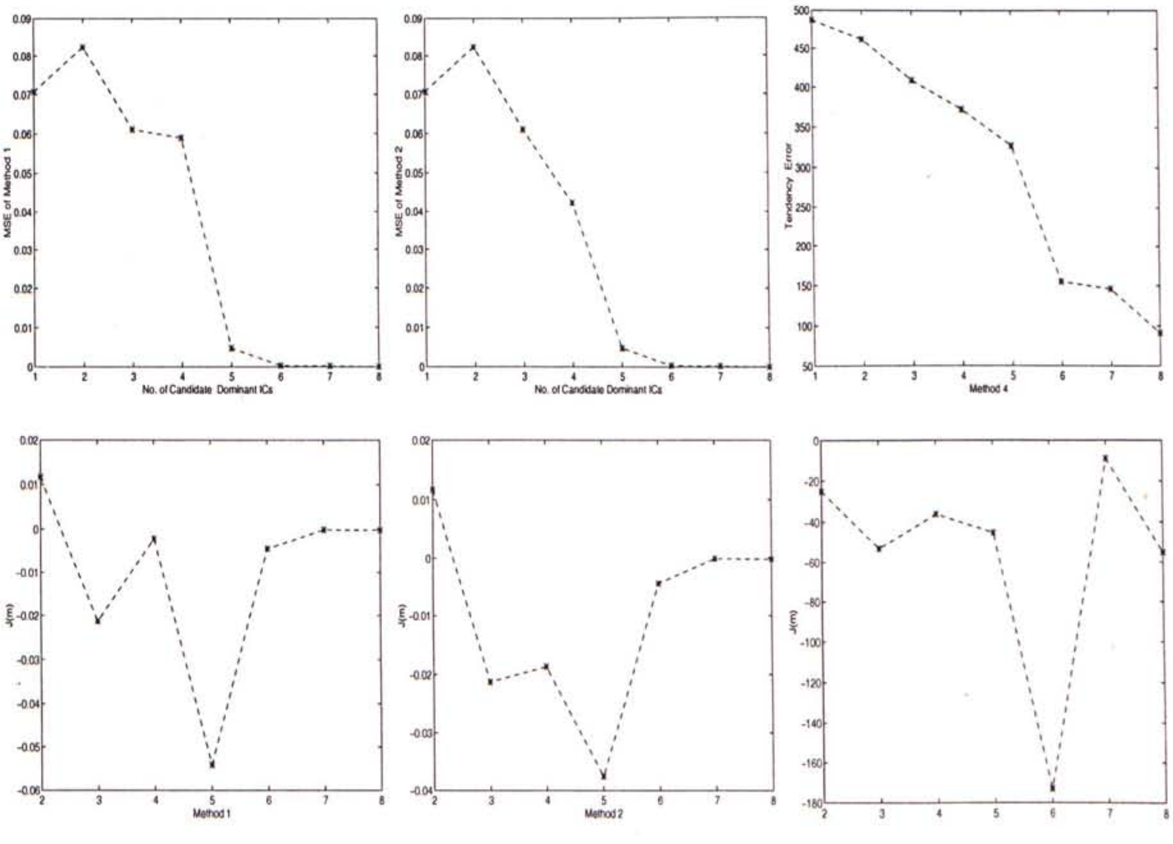
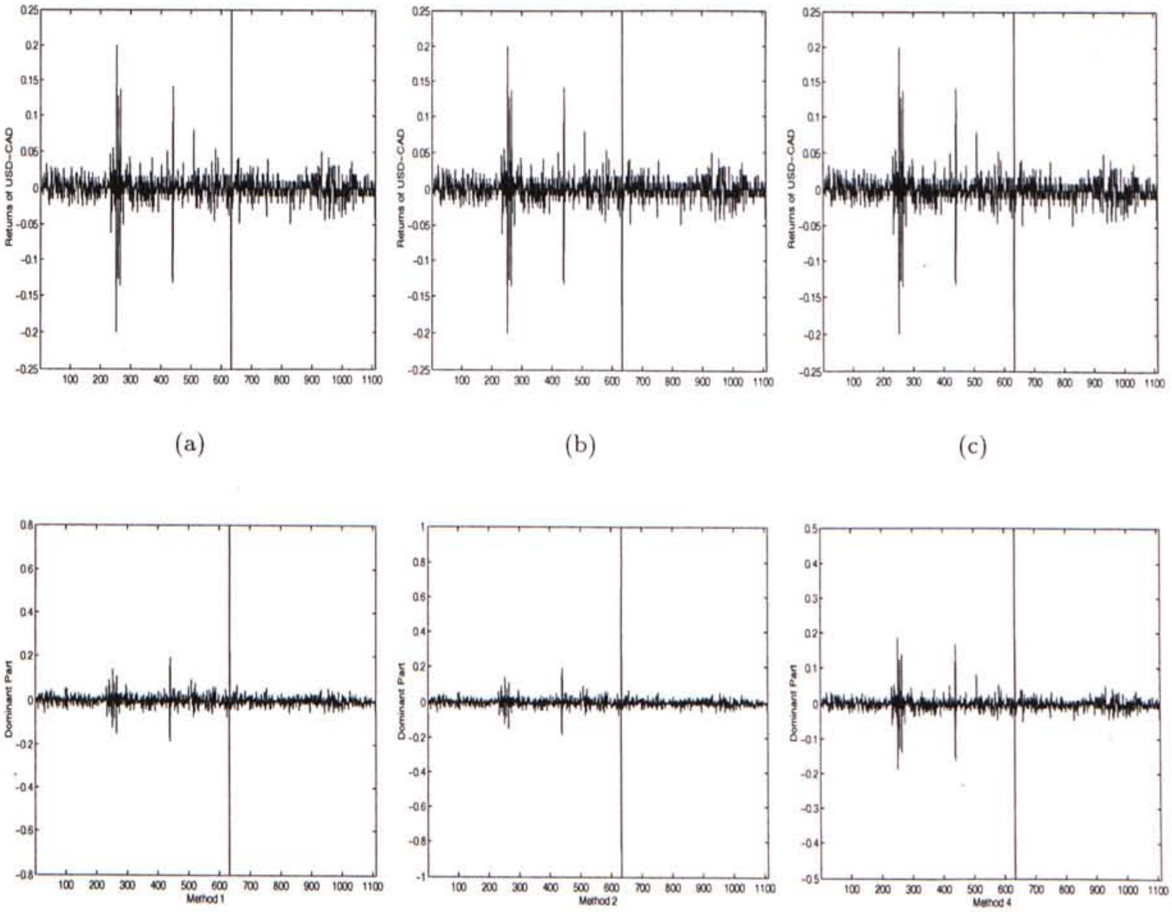


Figure 4.5: MSE under the measure of Methods 1, 2 and TE under Method 4 (upper row) and Curve of $J(m)$ vs. m under the measure of Methods 1, 2 and 4 (lower row) (USD-CAD)



(a) (b) (c)

(d) (e) (f)

Figure 4.6: Return of USD-CAD data (upper row), the reconstructed signals (lower row) by using the m^* dominant ICs determined by Method 1, 2 and 4 respectively from left to right.

4.6.4 Experiment Results on USD-FRN Return

We have made experiments to reconstruct the USD-FRN return data by using BETE method to gradually eliminate the IC which is relatively not dominant in controlling the change tendency of the original data. Table 4.7 is the simulation results in selecting the eliminated ICs gradually. The order of sorting the dominant ICs (contrary to the elimination order) is [3,5,1,8,2,4,7,6].

The TE values corresponding to different dominant ICs number under measurement of Methods 1, 2 and 4 are listed in table 4.8.

The MSE value curves under Method 1, 2 and TE values under BETE Method are shown in the upper row of Figure 4.7. The relative cost function $J(m)$ curves are demonstrated in the lower row of Figure 4.7.

- Method 1: $m^* = 3$;
- Method 2: $m^* = 3$;
- Method 4: $m^* = 7$;

In Figure 4.8, for each method, we use m^* dominant ICs to reconstruct the original USD-SWF return data. According to the different dominant ICs number selected through methods 1, 2 and 4, the Tendency Error between the reconstructed return data and the original return data are 341, 341 and 197 respectively, which shows that the BETE method performs better than other two methods, it can reconstruct most of the change tendency of the original data.

No. of ICs Deleted	TE vs Independent Component								Elimination Order of ICs
1	397	335	382	334	479	197	296	327	6
2	397	335	382	334	479	197	296	-	7
3	392	336	396	325	472	334	-	-	4
4	386	333	376	465	349	-	-	-	2
5	386	421	492	341	-	-	-	-	8
6	392	409	480	-	-	-	-	-	1
7	485	477	-	-	-	-	-	-	5
8	-	-	-	-	-	-	-	-	3

Table 4.7: Simulation results of sorting dominant ICs by BETE method (USD-FRN)

No. of ICs Selected	Method 1	Method 2	Method 4
1	485	485	477
2	409	409	392
3	341	341	341
4	349	349	333
5	334	334	325
6	295	327	295
7	197	197	197
8	171	171	164

Table 4.8: TE between original signal and reconstruction signal measured by Method 1, 2 and 4 under different dominant ICs number (USD-FRN)

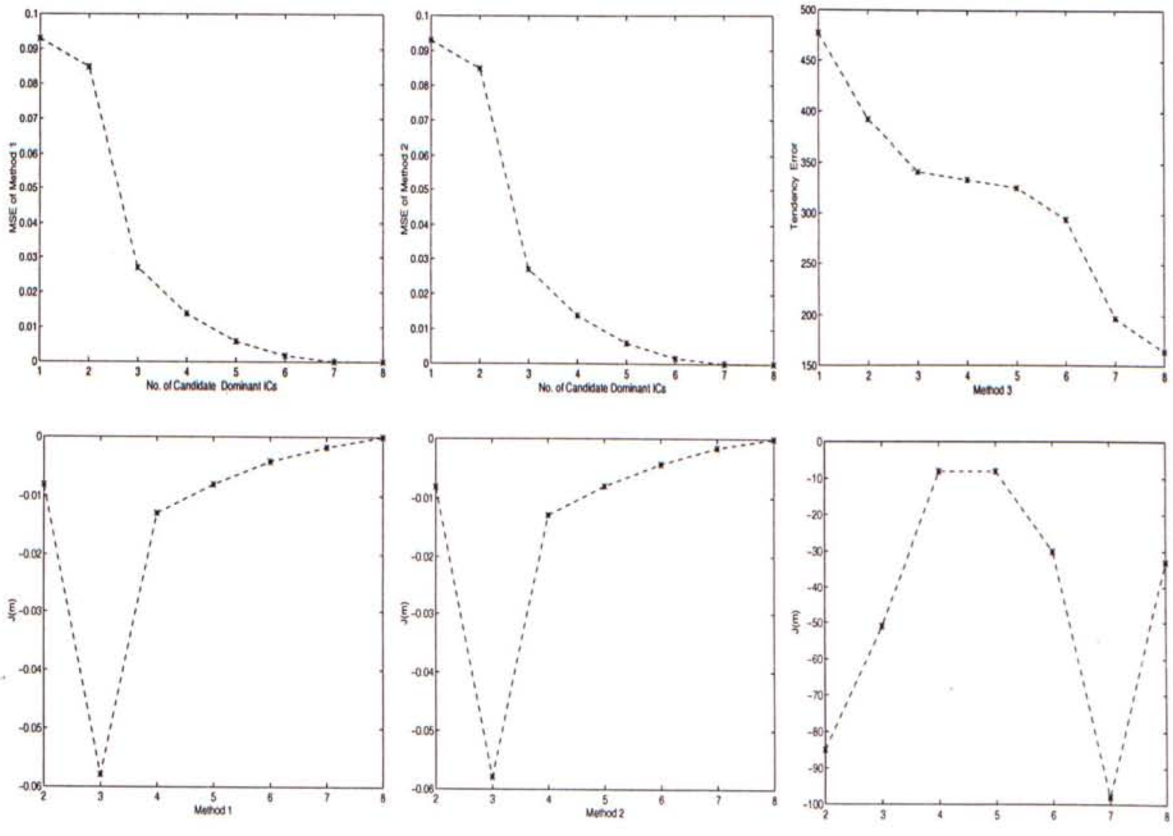
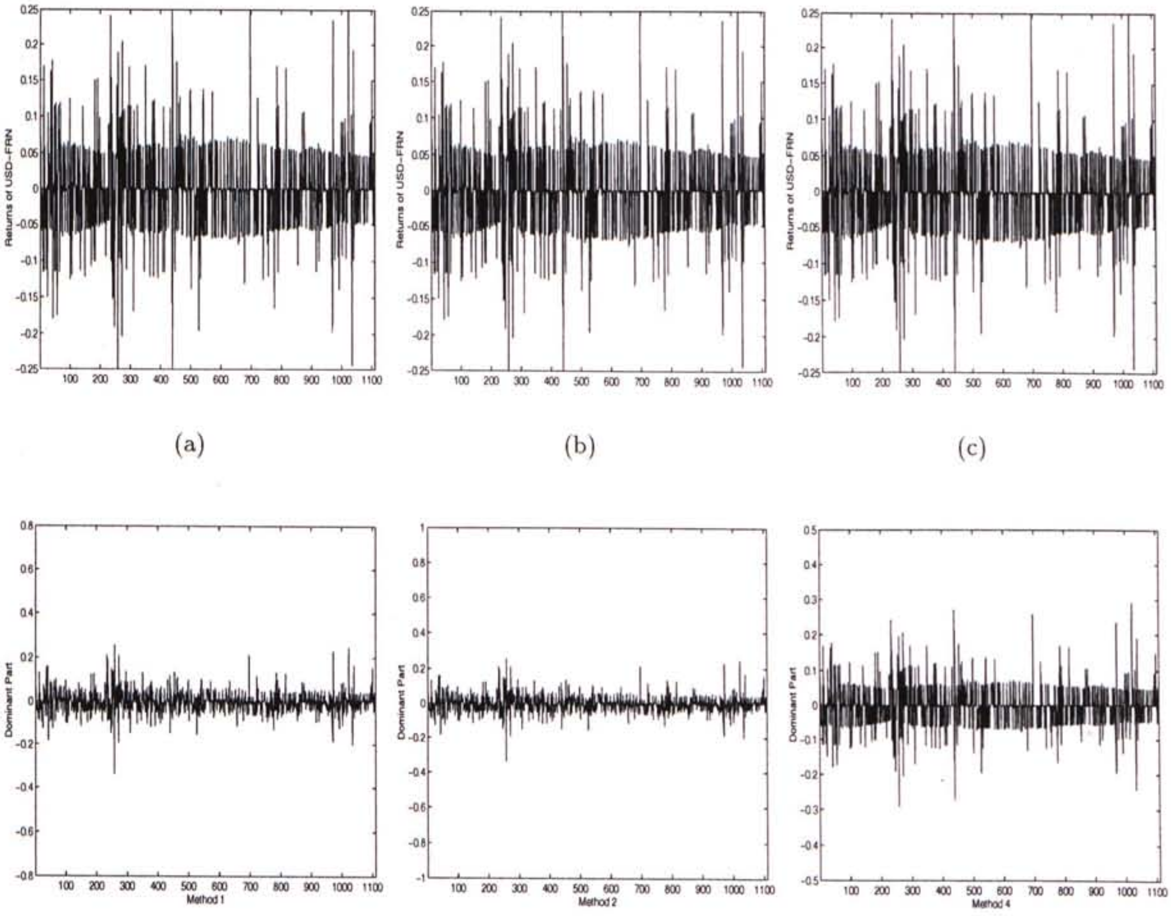


Figure 4.7: MSE under the measure of Methods 1, 2 and TE under Method 4 (upper row) and Curve of $J(m)$ vs. m under the measure of Methods 1, 2 and 4 (lower row) (USD-FRN)



(a) (b) (c)

(d) (e) (f)

Figure 4.8: Return of USD-FRN data (upper row), the reconstructed signals (lower row) by using the m^* dominant ICs determined by Method 1, 2 and 4 respectively from left to right.

Chapter 5

Other Analysis of ICA in Foreign Exchange Rate Markets

5.1 Variance Characteristics of ICs and PCs

We first analyze the variance change of the independent components (ICs) and compare them with those of the principal components (PCs) got from principal component analysis (PCA). The variances of each independent components (here 21 windows are used, each window contains 100 data points, within which there is an overlap of 50 data points) is shown in Figure 5.1, from which we can see large change of variance exists in each independent components, that is to say, each independent component makes its particular effect in the change of exchange rates. For different exchange rate, their influence may be different. On the other hand, for that of PCA as shown in Figure 5.2 with the same configuration of the windows as that in Figure 5.1, the large change in variance are mainly located in the first 4 principal components, that is to say, most of the informations are contained in these 4 components. Figure 5.3 demonstrates the percentage of each principal component, from which we also can find that the first two principal components already can represent most of the original signal.

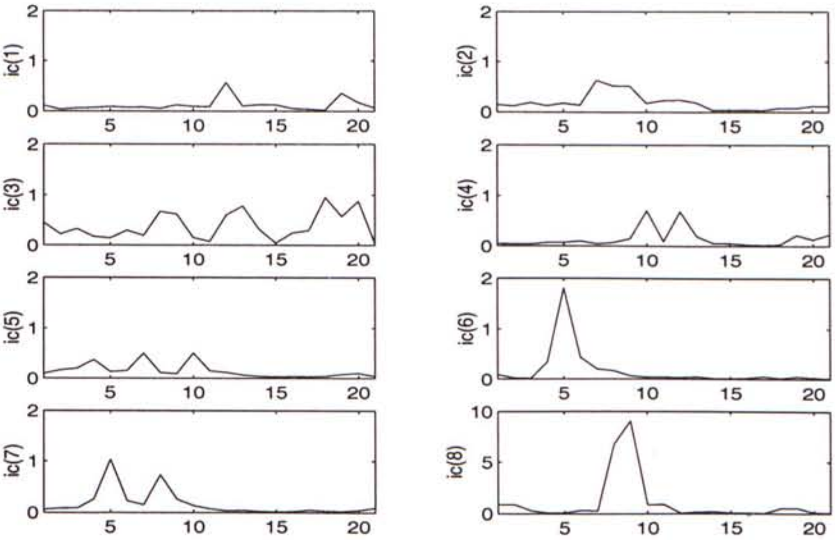


Figure 5.1: Variance of independent components

5.2 Reconstruction Ability between PCA and ICA

As that of in (Back and Weigend 1997), here we also randomly select an foreign exchange rate (usd-aud) as an example to compare the reconstruction ability of PCA and ICA. We separately use two major principal components and two dominant independent components to reconstruct the original usd-aud data. Simulation results can be found in Figure 5.4. It is apparent that by using ICA method, the original exchange rate data can be reconstructed very well, the summation of other weighted independent components only performs like one kind of random disturbance with less magnitude change. But for PCA, the reconstruction result is not very satisfied, there are still some big shocks left in the minor components and can not be reconstructed, such kinds of shocks usually represent some new information that often change the movement directions of the exchange rates.

5.3 Properties of Independent Components

For eight separated independent components, we notice that independent component 1 plays an important role in the change of almost all exchange rates. Components 3 and 5

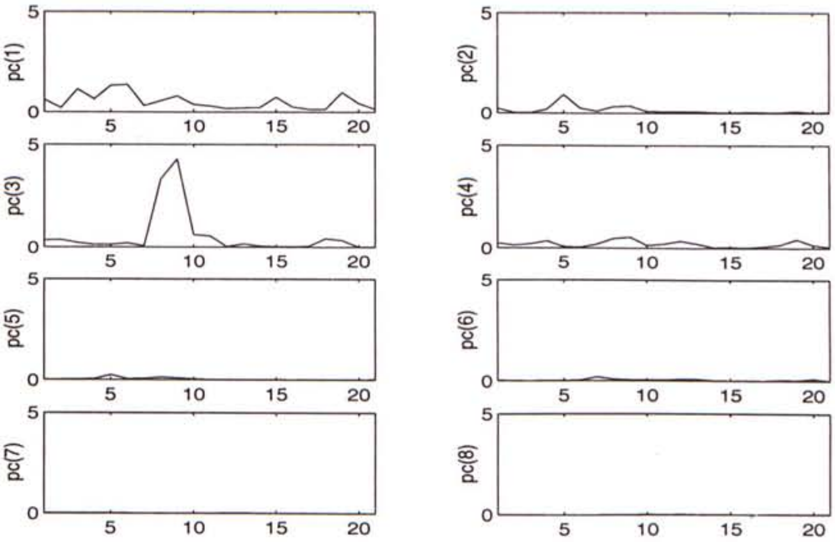


Figure 5.2: Variance of principal components

mainly influence European currencies, such as *German mark, French franc, Swiss franc and British pound.* Components 6 and 7 mainly influence Japanese yen and Canadian dollar. Component 2 mainly influence Australian dollar. Component 4 mainly influence non-European currencies. Weights of components 6 and 7 on Canadian dollar and Japanese yen have the contrary sign, which result in the adverse movement tendencies of these two currencies.

Figure 5.5 demonstrates the weighted independent components of usd-dem. From the change magnitude of each components, we can clearly see that components 1, 3 and 5 are three major components. Apparently the factor $y1$ looks like making a periodical floating movement. According to the history data of these exchange rates, we find that such kinds of regular movements are usually caused by the periodical government intervention such as adjusting the interest rates, etc. Some random but very important events that influence these exchange rates also can be observed from another majòr factor $y5$. In $y5$ we find that the curve starts with a ascending period, this is because from the beginning of 1992, two countries of previous Russia fought for the Black sea fleet, which results the ascending of US dollar versus other currencies, especially European currencies. The second obvious ascending period happens in the September of 1992. Because of the different attitude

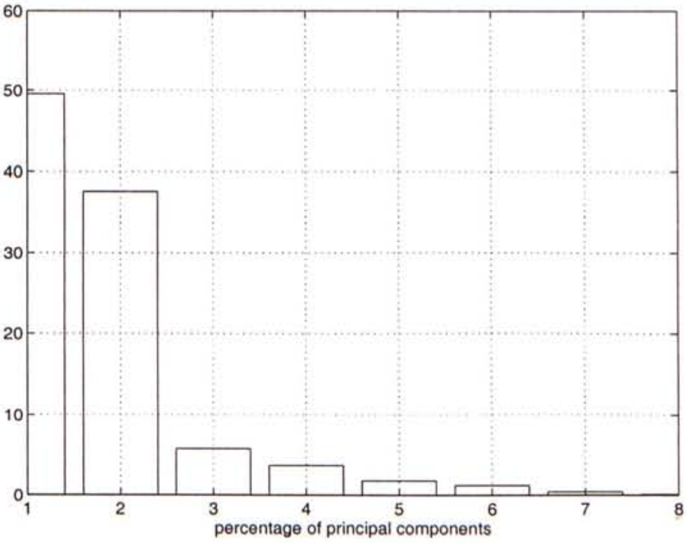


Figure 5.3: percentage of principal components

of some European countries in joining Maastricht's treaty, a currency crisis happened which leaded to a storm in the European foreign exchange rates markets. Through this crisis, many European countries widened their fluctuating amplitude of these currencies versus US dollar. We can see such adjustments actually greatly influence the European exchange markets and play roles in quite a long time. The 3rd ascending period in curve $y5$ happened in the August of 1994 as American Federal Reserve Bureau rushed into the markets and bought enormous US dollars, which also strongly influenced the movement of the markets.

Skewness usually refers to the asymmetry of a distribution. A distribution that is positively skewed has a long tail on the right side of the distribution and its mean is typically greater than its mean, which in turn, is greater than its mode. Because the mean exceeds the median, most of the returns are below the mean, but they are of smaller magnitude than the fewer returns that are above the mean. A distribution that is negatively skewed has the contrary effects. In Figure 5.6 we demonstrate the skewness property of each independent component.

5.4 Autocorrelation

Of all the independent components that influence the usd-jap exchange rate, components 1 and 6 are two major ones. In order to analyze the change characteristics of these two factors, we compute their autocorrelation coefficients (AC) by using 21 windows and show the simulation results in Figure 5.7, where the solid line represents the autocorrelation coefficient of component 1 and the dotted line represents that of component 6. We find that during the total computational period, for component 1, usually the AC value is larger than 0.9, which means the components in consecutive time steps are highly correlated. There is just one value smaller than 0.8, which can be explained that there are some big changes happened in this component during this period. For component 6, similarly, during most of the time, the AR value is larger than 0.95, there is only one value around 0.75. The results hint that these two components are highly autocorrelated, some big changes suddenly decrease their autocorrelation, such kinds of changes usually represent some big events or some market intervention.

5.5 Rescaled Analysis

In order to make it clear whether the change of the exchange rates are controlled by the deterministic factors or some random noise factors, we use the rescaled range (R/S) analysis. It is a method that frequently applied to natural phenomena to detect any biases in behavior over time. For the observed signal x_t, the detailed equations are listed as follows:

$$m(N, t_0) = \sum_{t=t_0+1}^{t_0+N} r_t/N, \quad r_t = x_t - x_{t-1} \tag{5.1}$$

$$S(N, t_0) = \{\frac{1}{N} \sum_{t=t_0+1}^{t_0+N} [r_t - m(N, t_0)]^2\}^{\frac{1}{2}} \tag{5.2}$$

$$X(N, t_0, i) = \sum_{t=t_0+1}^{t_0+i} (r_t - m(N, t_0)),\ 1 \leq i \leq N \tag{5.3}$$

$$R(N, t_0) = \max_i X(N, t_0, i) - \min_i X(N, t_0, i) \tag{5.4}$$

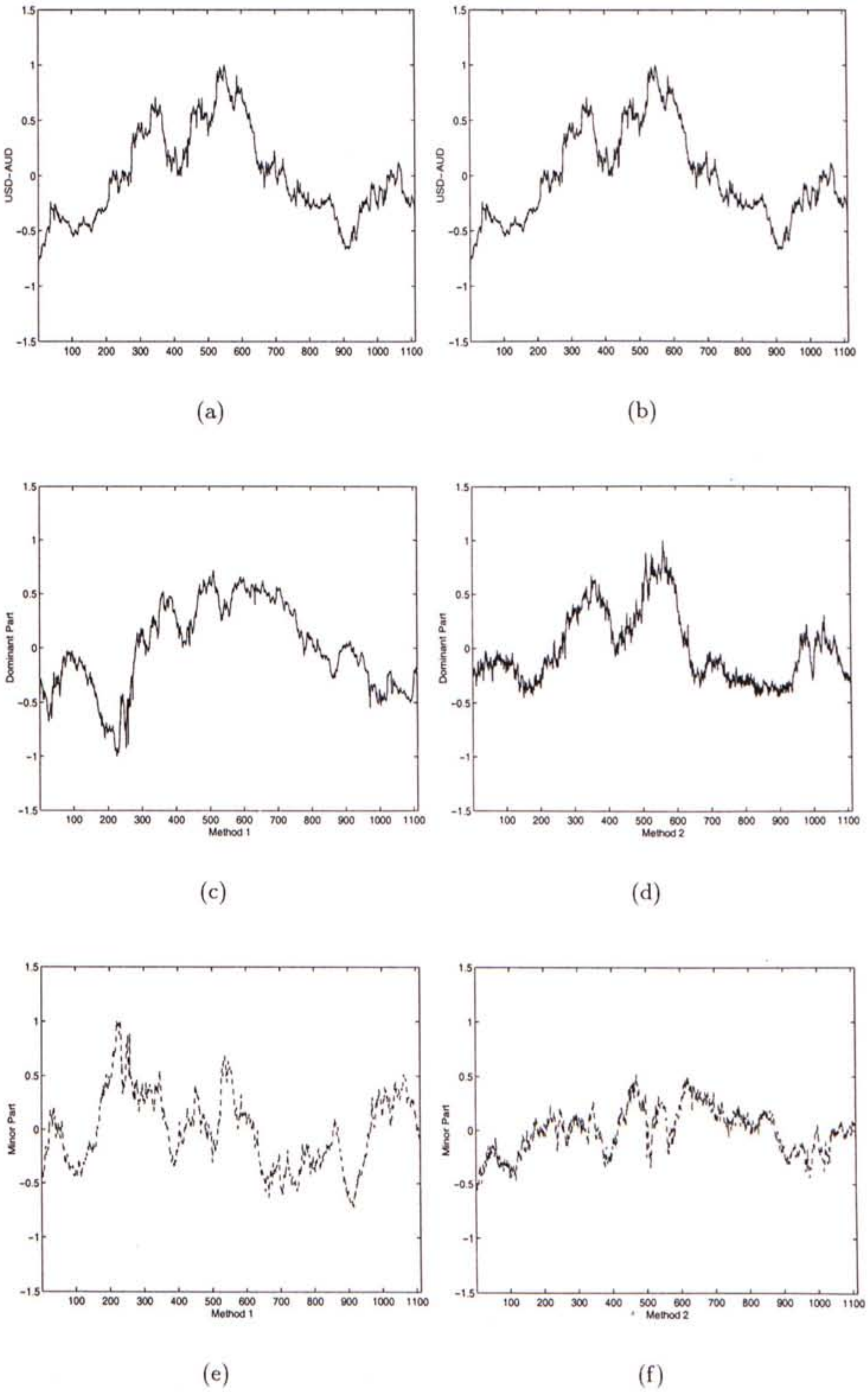
$$[R/S](N) = \frac{\sum_{t_0} R(N, t_0)}{\sum_{t_0} S(N, t_0)} \tag{5.5}$$

$$[R/S](N) \approx (N)^H \tag{5.6}$$

The Hurst exponent H is defined as:

$$H = \log(R/S) \,/\, \log(N) \tag{5.7}$$

As we know, H can range between 0 and 1. An H equal to 0.5 implies pure random walk behavior. An H between 0 and 0.5 implies anti-persistent behavior. An H greater than 0.5 but less than or equal to 1 implies persistent behavior. Figure 5.8 demonstrates the rescaled analysis for the 8 independent components. The simulation results show that during most of the time the Hurst exponent of component 1, 3 and 5 are larger than 0.5, which mean that these three components all show persistent behavior. We have mentioned before that the exchange rates of European currencies are mainly influenced by these three components, under such cases, naturally here we conclude that the change of the European currency are mainly influenced by some persistent change components. But for some non-European currency exchange rates, the Hurst exponent of their major independent components are less than 0.5, which means that the change of these non-European currencies are controlled by some anti-persistent factors. This new finding tells us the reason why the movement characteristics of European currencies and those of non-European currencies are usually different.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 5.4: Normalized original USD-AUD data (upper low), the reconstructed signals (middle low) by using the first 2 dominant components determined by PCA and ICA respectively from left to right, and the corresponding reconstructed signals by the left non-dominant components (lower row).

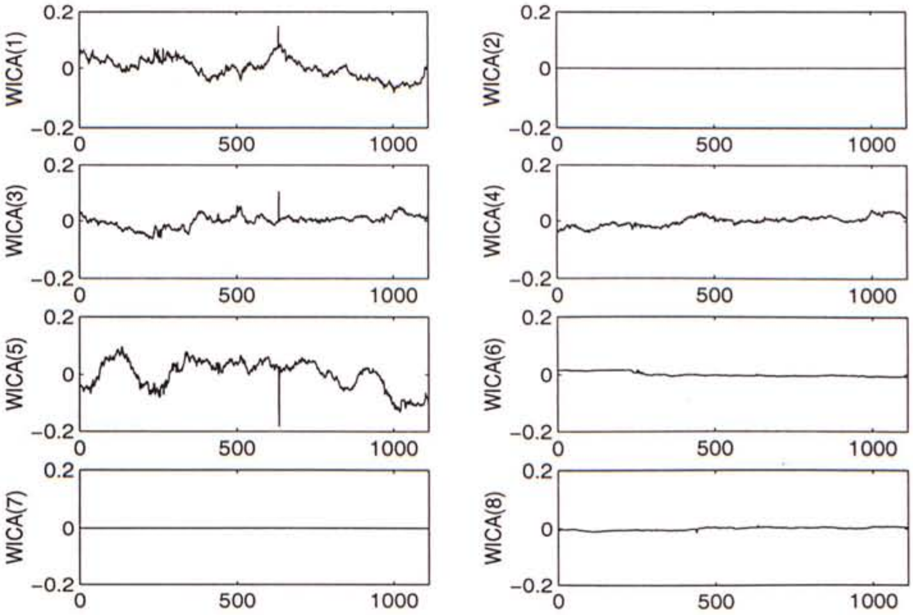


Figure 5.5: Weighted ICs of usd-dem

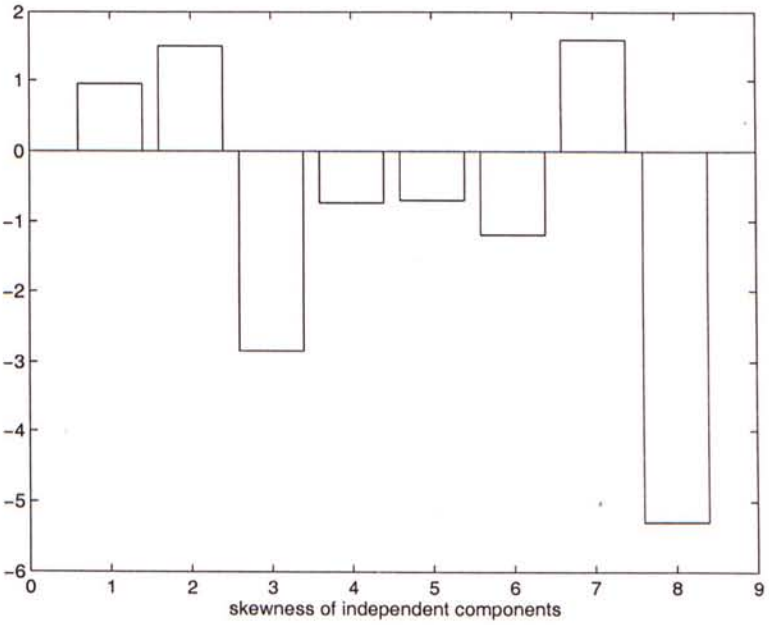


Figure 5.6: Skewness of independent components

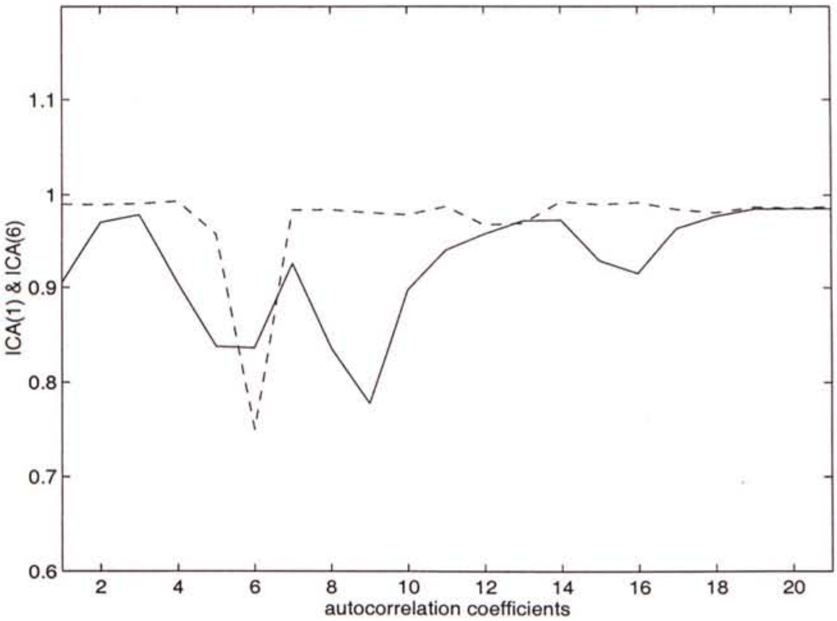


Figure 5.7: Autocorrelation coefficient of IC(1) and IC(6)(by windows)

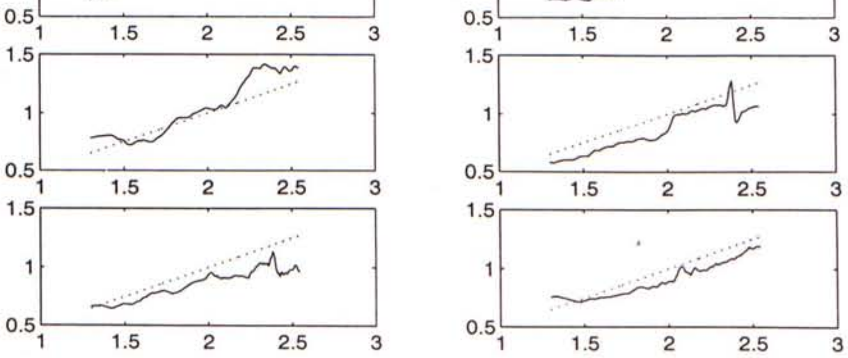


Figure 5.8: Rescaled analysis of independent components

Chapter 6

Conclusion and Further Work

6.1 Conclusion

We sort the ICs according their L_1 norm as shown in (Back and Weigend 1997), then we expand this method as sorting them by L_p norm measurement. we develop a criterion to find out the appropriate number of dominant WICs under MSE measurement between the original data and the reconstructed data.

We also determine the dominant ICs order under measurement of the MSE between the original data and the reconstructed data. Based on this criterion, we study a Forward Selection approach (Lai et al. 1998a) to sort the WICs into a certain order according to their dominant values measured by MSE.

Considering of the different practical needs, we determine the dominant ICs order according to the Tendency Error (TE) between the original data and the reconstructed data. We study a Backward Elimination Tendency Error (BETE) approach (Lai et al. 1998b) to implement this criterion. We also develop a corresponding number determination criterion.

Experiments show that the dominant WICs obtained by these order-sorting approaches and number-determination criteria are better than those heuristically obtained in the MSE and TE signal reconstruction. Furthermore, we have noticed that both WIC dominant order and its dominant number vary with different measure method used in order-sorting approaches and the cost function of number-selection criterion. To obtain an appropriate set of dominant WICs, we suggest the measurement in dominant WICs evaluation, dominant order determination and number selection criterion should be consistent.

Additionally, we have also proposed two heuristic modified implementation algorithms based on the original LPM algorithm. Experiments show that these modified algorithms can efficiently accelerate the convergence speed.

6.2 Further Work

The further work consists of two aspects:

1. Also our proposed *Forward Selection* and *Backward Elimination Tendency Error* approaches can determine the suitable dominant ICs and reconstruct the original signal well, some other optimized approaches such as branch-and-bound approach have not been applied in this aspect. How to find other number-selection criteria also need to be further studied.

2. After we separate out those independent components that influence or control the original financial data, characteristic of each independent component should be deeply studied, such as their change periodicity, move tendency, correlation between different ICs, relation between correlation and volatility, etc. These analysis will be more benefit in the understanding of the financial markets and help us to make correct prediction or other applications.

Appendix A

Fast Implement of LPM Algorithm

A.1 Review of Selecting Subsets from Regression Variables

The problem of determining the "best" subset of regression variables has long been of interest to applied statisticians and received considerable attention in the statistical literature, especially during 1960 and 1970 periods. .Usually linear models and the least squares criterion are considered. In (Miller 1984), reasons of using only some of the variables or possible predictor variables are explained:

(1) to estimate or predict at lower cost by reducing the number of variables

(2) to predict accurately by eliminating uninformative variables

(3) to describe a multivariate data set parsimoniously

(4) to estimate regression coefficients with small standard errors

Algorithms for finding best-fitting subsets of variables to a set of data requires a search strategy and a computational algorithm. Garside (1971) and others have proposed methods for generating the residual sum of squares for all subsets of all sizes. They use Gauss-Jordan methods operating upon sums of squares and product matrices. Alternatively, the planar-rotation algorithm of Gentleman (1973, 1974) can be used to change the order of variables within a triangular factorization, as described for instance by Elden (1972), Hammarling (1974) and Clarke (1981)

Apparently an exhaustive search can be very time consuming if a large number of possible subsets have to be examined. Under such cases, some more efficient methods based on the procedure of sequentially introducing the variables into the model have been proposed. Two of them are called Forward Selection and Backward Elimination (Efroymson, 1966 or Draper and Smith, 1966). Forward Selection method starts with no variables in the equation and adds one variable at a time until either all variables are in or until a stopping criterion is satisfied. The variable considered for inclusion at any step is the one yielding the largest single degree of freedom $F-$ratio among those eligible for inclusion. Backward Elimination method starts with all variables are included in the equation, variables are eliminated one at a time. At any step, the variable with smallest $F-$ratio as computed from the current regression, is eliminated if this $F-$ratio does not exceed a specified value. Some combination of these two methods have also been proposed, the most popular one is described by Efroymson (1960), which is a variation on forward selection. In this method, after each variable (other than the first) is added to the set of selected variables, a test is made to see if any of the previously selected variables can be deleted without appreciably increasing the residual sum of squares. Forward selection and the Efroymson algorithm can be used when there are more predictors than observations, while backward elimination is usually not feasible in such cases.

An alternative to the Efroymson algorithm, which often finds better-fitting subsets, is that of replacing predictors rather than deleting them. Suppose that we have 26 potential predictors denoted by the letters A to Z and currently we are looking for subset of four predictors. We can start with for example the subset ABCD. We can consider first replace predictor A from the remaining 22 which gives the smallest residual sum of squares in a subset with B, C and D. If no reduction can be obtained then A is not replaced. Then we can try replace B, then C, then D and then back to the new first predictor, continuing until no further reduction can be found.

Many variations on the basic replacement algorithm are used. A variation is to find the best replacement for A, but not to make the replacement. Similarly the best replacements for B, C and D are found but only the best of the four replacement is implemented. The process is repeated until no further improvement can be fund. A sequential replacement algorithm is possible, that is it is carried out sequentially for one, two, three, four predictors, etc., taking the final subset of $p-1$ predictors plus one other predictor as the starting point for finding a subset of p predictors. Another variation is to use randomly chosen starting subset of each size, which is particularly useful when there is a large number of predictors. Replacement methods require more computation than forward or the Efroymson algorithm, but it is still feasible to apply to problems with several hundred variables when subsets of up to 20-30 variables are required.

Some automatic methods start by finding the simple correlations between the predictand and each of the predictors, and then check scatter diagrams for those predictors with the largest correlations. This may reflect the need for a transformation, or adding polynomial terms, or the presence of outliers. After selecting one predictor, the process is repeated using the residuals from fitting this predictor, continuing until nothing more can be seen in the data. This approach is a extension of forward selection and suffers from the weakness of that method. A formalized version of this approach has been called "projection pursuit" by Friedman and Stuetzle (1981).

One technique which has attracted a considerable attention is the ridge regression technique of Hoerl and Kennard [1970a,b]. They suggested that using all the available variables, biased estimators, $b(d)$, of the regression of coefficients may be obtained by $b(d) = (X'X + dI)^{-1}X'Y$ for a range of positive values of the scalar d. They recommended that the predictor variables should first be standardized to have zero mean so that the sum of squares of elements in any column of X should be one, such that $X'X$ should be replaced with the correlation matrix. $b(d)$ is plotted against d, this plot was termed the 'ridge trace'. Visual examination of the trace usually shows some regression coefficients which are 'stable', that is they only change slowly and others either decrease or change sign rapidly. The latter variables are then deleted. Usually this method will tend to select those variables which both yield regression coefficients with the same sign in single variable regressions and which show up early in forward selection.

The branch-and-bound technique is particularly valuable in reducing the number of subsets to be considered when there are "dominant" predictors such that there are no subsets which fit well that do not contain them. Suppose that we are looking for the subset of 5 variables out of 26 which gives the smallest RSS. Let the variables be denoted the letters A-Z. We could divide all the possible subsets into two branches, those which contain variable A and those which do not. Within each branch we can have sub-branches including and excluding variable B, etc Now suppose we have found a subset of five variables containing A or B or both which give RSS=100. If we start to examine that sub-branch which excludes both A and B. A lower bound on the smallest RSS which can be obtained from this sub-branch is the RSS for all of the other 24 variables. If this lower bound is larger than 100 then no subset of 5 variables from this sub-branch can do better than this, so this whole sub-branch can be skipped.

This technique has been first used in subset selection by Beale, Kendall and Mann (1967), and by Hocking and leslie (1967). It is further exploited by LaMotte and Hocking (1970). This method has the advantage of exhaustive search that guarantee to find the best-fitting subsets and meanwhile it also greatly reduce the computational time by skipping some subsets in the searching process. This method can be applied with advantage with most other criteria of goodness-of-fit. One such application has been made by Edwards and Havranek (1987) to derive so-called minimal adequate sets.

Gorman and Toman (1966) proposed a procedure based on a fractional factorial scheme in an effort to identify the better models with a moderate amount of computation. For the same purpose, Barr and Goodnight (1971) in the Statistical Analysis System (SAS) regression program proposed a scheme based on maximum-R^2-improvement. This is essentially an extension of the stepwise concept but the search is more extensive. For example, to determine the best p-term equation, starting with a given p-1 term equation, the currently excluded variable causing the greatest increase in R^2 is adjoined to that subset. Given this subset, a comparison is made to see if replacing a variable by one currently excluded will increase R^2, then the best switch is made. This process is continued until it is found that no switch will increase R^2. The resulting p-term equation is thought as the best, but this subset can be inferior to the one determined by SELECT.

We have already known the least squares L_2 is usually used as the major criterion in selecting subset of regression variables. Besides, many other criteria have been proposed. These criteria are stated in terms of the behavior of certain functions of the variables included in the subsets. Many of these criteria functions are simple functions of the residual sum of square for the p-term equation denoted by RSS_p. The L_1 criterion is used as an alternative to the L_2 criterion for its resistance to outlier in the data. The framing of the L_1 estimation problem as a linear program by Charnes, Cooper and Ferguson [1955], made L_1 estimation computationally feasible and allowed for the derivation of a number of properties of the L_1 estimates. A number of modifications of linear programming algorithms, such as that by Barrodale and Roberts (1974), have been devised to enhance the computational efficiency of L_1 estimation. Roodman (1974) gives a partial enumerative search procedure using a simplex algorithm with upper and lower bounds on the coefficients to specify the subset of variables being considered at each stage. Narula and Wellington (1976) describe an all-subsets procedure that uses both a primal and a dual simplex algorithm along with a pre-optimality check to move rapidly to the best subset. Hanson (1977) incorporated an implicit enumeration scheme with fathoming directly in the Barrodale and Roberts (1974) procedure. The resulting algorithm is very fast in finding the best subset. Narula and Wellington (1977, 1979) and Wellington and Narula (1981) have presented an algorithm for finding the best-fitting subsets of regression variables based on the criterion that of minimizing the sum of absolute deviations. Some other criteria including log-liner model that fitting to categorical data(Goodman 1971; Brown, 1976), in which the measure of goodness-of-fit is either a log-likelihood or a chi-square quantity. Other measures which have been used in subset selection have included that of minimizing the maximum deviation from the model, known simply as minimax fitting or as L_∞ fitting.

A.2 Unconstrained Gradient Based Optimization Methods Survey

Optimization techniques have been widely used in neural networks fields. Usually, the objective is to minimize a cost function (also called an error function or an energy function) by using some iterative methods. Of all the different methods, the gradient descent algorithm is the most basic one, in which the gradient of the specified cost function with respect to its parameters are computed. This algorithm has been used in many neural networks problems, for example, the popular back-propagation problem. Although this algorithm has the advantage of simple and easy to implement, but its drawback of slow convergence is also very apparent and impede its application in practical problems. Up to date, many modified algorithms aiming at fast implementation have been proposed.

Different from the gradient descent algorithm that use only the first derivatives of the cost function, the Newton method directly incorporate the Hessian matrix (second derivatives of the cost function). This method has the fast convergence speed than the gradient descent algorithm if the initial point is close to the optimal point. But it also has some disadvantages. First, it requires a good initial estimate of the solution, which is usually not available in many cases. Further, each iteration requires the computation of the Hessian matrix and also its inverse, which often introduces computational difficulties and singular problems. Finally, for a non-convex function, this method can converge to a local maximum, saddle point or minimum (Becker and LeCun 1988).

Also Newton method may converge faster than gradient descent method by taking into account additional information about the cost function, however, for many connectionist problems, the Hessian matrix is too large to compute and too expensive to invert. An approximation of the Hessian matrix is used by only considering its diagonal term (Becker and LeCun 1988). Since this approximate Hessian has only diagonal elements, it is not only trivial to invert, and the diagonal approximation can capture most of the curvature information.

Quasi-Newton technique combine Newton's method with some other convergent algorithm and overcome many of the drawbacks described above. One of the best techniques is the Broyden-Fletcher-Goldfarb-Shanno algorithm (Battiti and Masulli 1990). This algorithm is stable since the searching direction is always a descent direction. Also the evaluation of second-order derivative is not needed since the positive definite approximations of the inverse Hessian matrix can be obtained solely from gradient information. On the other hand, the storage requirement are extremely large for problems with large number of variables.

The conjugate gradient method compute the actual search direction as a linear combination of the current gradient vector with the previous search directions. Such kind of method requires much less storage than the Quasi-Newton method. Also they require an exact determination of the learning rate and some other parameters in each iteration step. Moreover, the conjugate gradient methods require approximately twice as many gradient evaluations as the Quasi-Newton methods, but they save time and memory needed for computing inverse of the Hessian matrix for large problems.

Moller (1993) proposed a scaled conjugate gradient algorithm. This method is fully-automated, include no critical user-dependent parameters and avoids a time consuming line search that is often used to to determine the step size in each iteration. Simulation results show that this method have much fast convergence speed than conjugate gradient and BFGS methods.

In most of the optimization methods, the learning rates are usually chosen arbitrarily. This naturally arise many drawbacks. One simple way to improve the learning process is to smooth the weight changes by adding the momentum term. The momentum factor can determine the relative contribution of the current and past partial derivatives to the current weight change. When consecutive derivative of a weight possess the same sign, the weight is adjusted by a large amount which will accelerate the convergence process. Similarly, when consecutive derivative of a weight possess opposite signs, the weight is adjusted by a small amount which can avoid the algorithm from oscillation (Jacobs 1988).

LeCun et al (1993) proposed a technique of computing the optimal learning rate in gradient descent algorithm. By using this scheme, only the principal eigenvalues and eigenvectors of the objective function's second derivative matrix need to be computed, not the Hessian matrix itself. This will greatly decrease the computational time and complexity. The optimal learning rate is estimated to be the inverse of the largest eigenvalue of the Hessian matrix.

As we know, usually the learning process covers the whole data size, but because we can not avoid that some data are contradictory and some are redundant. In this case, it will take us a long time to finish the training process. Zhang (1994) implemented the learning on an increasing number of selected training examples, starting with a small training set. Experiments show that such kind of incremental learning can speed up the convergence and achieve a reasonable performance. In this paper, not only the criterion of selecting the critical example is proposed, but also an efficient method of scheduling their training order is given out.

Yu et al. (1995) proposed an efficient method to derive the first and second derivatives of the objective function with respect to the learning rate. Several learning rate optimization approaches are proposed based on linear expansion of the actual outputs and line searches with suitable descent values and Newton-like methods. Yu and Chen (1997) proposed several approaches to accelerate the back-propagation learning procedure by dynamically updating the learning rate and the momentum factor. Optimization of learning rate was considered by using the first two derivative information of the cost function with respect to learning rate.

Some other heuristical methods of adaptively updating the learning rates have also been proposed, such as Delta-Bar-Delta algorithm (Jacobs 1988), Super SAB algorithms (Fahlman 1988; Schiffman 1992), Search-Then-Converge algorithm (Darken et al. 1990, 1992), Averaging algorithm (Polyak 1990), RPROP algorithm (Riedmiller and Braun 1992) and Quickprop algorithm (Fahlman 1988; Lebiere and Lebiere 1990; Veitch and Holmes 1991).

Salomon and Hemmen (1996) presented a dynamic self-adaptation genetic algorithm to accelerate steepest descent as it is used in iterative procedures. The basic idea is to take the learning rate of the previous step and increase or decrease it slightly, to evaluate the cost function for both new values of the learning rate and select the one that has the lower cost function value. The dynamic self-adaptation algorithm is composed of two steps start with a mutation of comparing the new value with the previous one and select the best one, in which gradient descent algorithm can be performed without normalization or with normalization.

Vitthal et al. (1995) modified the gradient descent algorithm by adding the integral and derivatives terms of the gradient. It can be seen through an appropriate tuning of the proportional-integral-derivative (PID) parameters. By using this method, the convergence rate can be greatly improved and the local minima can also be overcome. In this paper, the principle of how to appropriately tune the PID parameters and an "integral suppression scheme that effectively uses the PID principles are all proposed.

Baba et al. (1994) proposed a new algorithm which combines the modified BP method and the random optimization method of Solis and Wets (1981) to find a global minimum of the total error function of a neural network in a small number of steps. As the original BP method is based on the steepest descent method, which is one of the simplest optimization algorithm in the field of nonlinear programming that uses a fixed step size that does not perform line searches, so it cannot ensure convergence even to a local minimum of the objective function. In this paper, a new modified BP method is proposed that uses conjugate gradient method and performs a line search using quadratic polynomial approximation of the total error function in the search direction. To prevent the algorithm from stopping on a local minimum of the total error function, the random optimization method that ensures convergence to a global minimum of the objective function in a compact region is combined with the modified BP method to form the hybrid algorithm.

A.3 Characteristics of the Original LPM Algorithm

We recall that the original LPM algorithm (Xu et al. 1998) is mainly implemented by using the natural gradient method to tune W and the gradient descent algorithm to tune other parameters. Unlike the famous FFT algorithm, in which the properties of the triangular function's periodicity and odd-even characteristic of the number of the data are fully made use of and directly result in the great simplification of the computation process, here the consideration of periodicity and whether the number of the mixture or the number of data points is odd or even does not make effect. Besides, the observed signals are time series, they can not be separated into several parts and computed independently, so the parallel algorithm is not suitable to be adopted. Moreover, when W is a $n \times n$ matrix, the Hessian matrix of the cost function $J(W)$ is a $n^2 \times n^2$ matrix. For high dimension cases, it is prohibitive in computing, even if we can make some approximation and do not need to really calculate the inverse of Hessian matrix, usually it is still unworkable owing to the complexity of the cost function.

A.4 Constrained Learning Rate Adaptation Method

In the original LPM algorithm (Xu et al. 1998), the natural gradient algorithm (Amari et al. 1996) has been extensively used to optimize the related parameters and shows fast convergence speed than the gradient algorithm. However, in practical application, such as the analysis of the independent components that influence the financial markets, usually there are multiple source signals. From the experience of my work in using this algorithm in high dimension source signal cases, the convergence speed is unsatisfactory. Naturally, a question is proposed: whether we can make some modification to this algorithm and let it still perform well in solving practical problems that with high dimension source signals.

In this section, we present our first modified method, the *Constrained Learning Rate Adaptation Method.* The main purpose is to adjust the de-mixing matrix W with adaptively adjusting the learning rate in each iteration. Under same conditions, experiments have been made to compare the modified algorithm with the original algorithm.

As regards to the gradient descent algorithm, the iteration equation at $k+1$ step is

$$W_{k+1} = W_k + \eta_k \Delta W_k \tag{A.1}$$

$$J(W_{k+1}) = \min_{\eta} J(W_k + \eta \Delta W_k) \tag{A.2}$$

The cost function at $k+1$ time step is:

$$J(W_{k+1}) = -\log|\det W_{k+1}| - \sum_{i=1}^{n} \log g_i(y_i) \tag{A.3}$$

From

$$\begin{aligned} \det W_{k+1} &= \begin{vmatrix} w_{11}+\eta\Delta w_{11} & w_{12}+\eta\Delta w_{12} \\ w_{21}+\eta\Delta w_{21} & w_{22}+\eta\Delta w_{22} \end{vmatrix} \\ &= a\eta^2 + b\eta + c \end{aligned} \tag{A.4}$$

where

$$\begin{aligned} a &= \Delta w_{11}\Delta w_{22} - \Delta w_{12}\Delta w_{21} \\ b &= \Delta w_{11} w_{22} + \Delta w_{22} w_{11} - \Delta w_{12} w_{21} - \Delta w_{21} w_{12} \\ c &= w_{11}w_{22} - w_{12}w_{21} \end{aligned}$$

So we can get

$$\frac{\partial\,|\det W_{k+1}|}{\partial\eta} = |2a\eta + b| \tag{A.5}$$

and

$$\frac{|2a\eta+b|}{|a\eta^2+b\eta+c|} + h(y)' \cdot \Delta W \cdot X_{k+1} = 0 \tag{A.6}$$

In the second part of $J(W_{k+1})$, y_i should be the output value at $k+1$ step, under such cases, the derivative of the cost function is too difficult to be calculated out, so here we use an approximation, this is, we assume the output value at two consecutive time step, k time step and $k+1$ time step are very similar, thus for the derivative of the cost function at $k+1$ time step, we use the output value at k time step to approximate its value. We define $h(y)'\Delta W \cdot X = m$, then we have

$$|2a\eta + b| + m \cdot \left|a\eta^2 + b\eta + c\right| = 0 \tag{A.7}$$

Here we assume $2a\eta + b$ and $a\eta^2 + b\eta + c$ are all larger than zero.

The formula is simplified as

$$ma\eta^2 + (mb + 2a)\eta + mc + b = 0 \tag{A.8}$$

This is a quadratic equation. We can figure out the two roots.

$$\begin{aligned} \eta_1 &= \frac{-(mb+2a) + \sqrt{(mb+2a)^2 - 4ma(mc+b)}}{2ma} \\ \eta_2 &= \frac{-(mb+2a) - \sqrt{(mb+2a)^2 - 4ma(mc+b)}}{2ma} \end{aligned} \tag{A.9}$$

and their simplified forms are

$$\begin{aligned} \eta_1 &= \frac{-(mb+2a) + \sqrt{m^2(b^2 - 4ac) + 4a^2}}{2ma} \\ \eta_2 &= \frac{-(mb+2a) - \sqrt{m^2(b^2 - 4ac) + 4a^2}}{2ma} \end{aligned}$$

From the two roots η_1 and η_2, we choose the real part of the smaller root and take its absolute value as η.

After simplify and solve out the above quadratic equation, we can have two roots, here we choose the real part of the smaller one. In order to prevent the computation from overflowing, we give a selecting criteria in the determination of the learning rate, that is

```
if  $\eta < N$
  $\eta = n * \eta$
else                                        (A.10)
  $\eta = \eta_o$
```

In the experiment, we choose $N = 0.8$, $n = 0.01$, η_o is fixed at 0.0001.

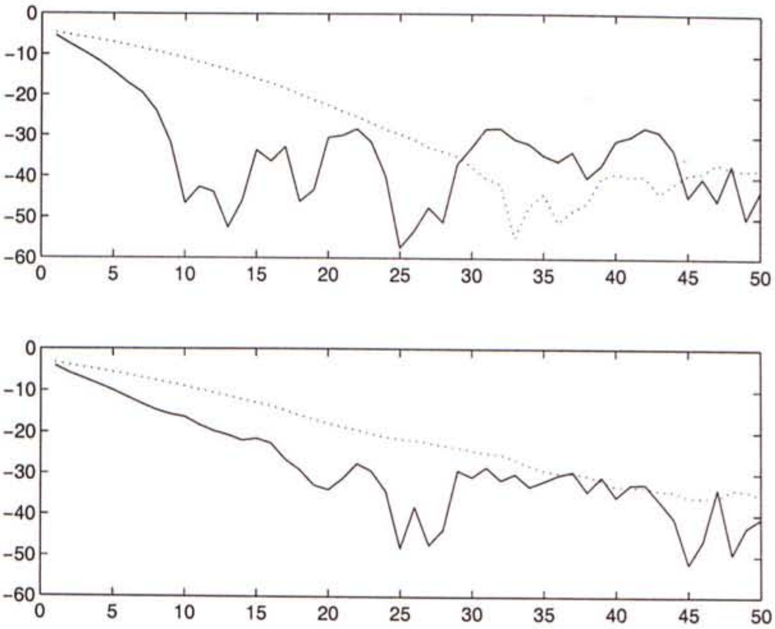


Figure A.1: Method 1 (solid) and old method (dotted) with 2 sub-Gaussian signals for 30,000 data

Experiment Results

The performance of the separation is determined by how close to PD the matrix $V = WA$ is. The element v_{ij} represents the amplitude of source s_j goes into recovered signal y_j and v_{ij}^2 represents the power. The greatest v_{ij}^2 in a row in V is regarded as the power of the signal and the sum of other v_{ij}^2 of the same row is regarded as the power of the noise. Here we defined the following noise-to-signal power ratio of channel i as the performance index of source separation.

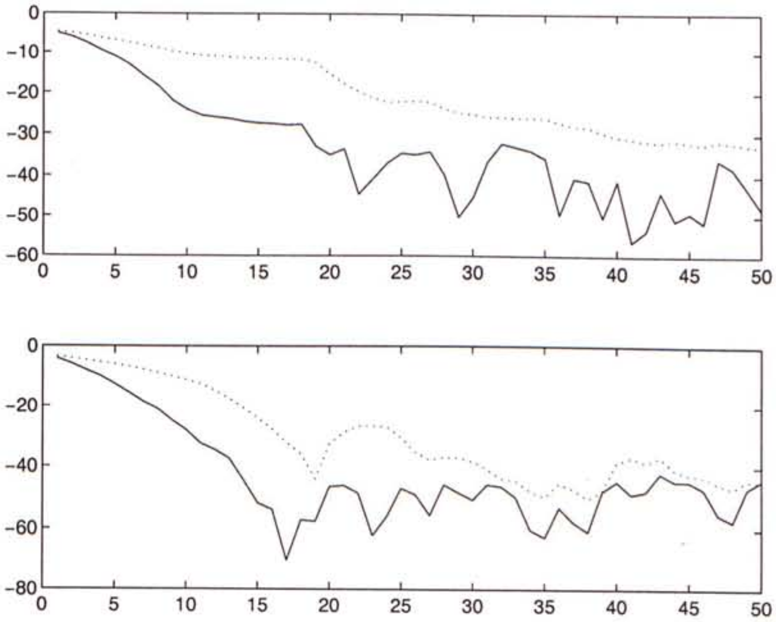


Figure A.2: Method 1 (solid) and old method (dotted) with 1 sub-Gaussian and 1 speech signal for 30,000 data

$$N\,/\,S_i = 10 * \log 10(\frac{\sum_{j \neq k} v_{ij}^2}{v_{ik}^2}), \qquad k = \arg\max_l v_{il}^2 \tag{A.11}$$

Initialization

We have made experiments to compare the convergence speed between our modified algorithm and the original algorithm. For the original algorithm, the learning rates of W, γ, a, b are kept at 0.0001, 0.001, 0.01, 0.001 respectively. For the modified algorithm , the learning rates of γ, a, b are the same as those of the original algorithm, but the learning rate of W is adjusted adaptively according to the new criterion. The initialization of other parameters of these two algorithms are all the same. W is initialized as an identity matrix. The mixing matrix A is set as

$$A = \begin{bmatrix} 1 & 0.6 \\ 0.7 & 1 \end{bmatrix} \tag{A.12}$$

All γ_{ij} are initialized as 0.25. All b_{ij} elements are initialized in the range of (0.5,15). All a_{ij} elements are initialized as 0. Four components are used in the mixture of densities.

Experiment Results on LPM Algorithm

In trial 1, 2-channel artificially generated independent and identically distributed signals with uniform distribution in [-1, 1] are used. In trial 2, we use one uniform distributed sub-Gaussian signal and one super-Gaussian speech signal which is recorded from a man telling a story. Each channel consists of 30000 data points. The performance graphs of trial 1 and 2 are plotted in Figure A.1 and Figure A.2. From the simulation results we see that if we can suitably choose related parameters (such as m and n), then the learning rate of W can be automatically adapted in each iteration, and the convergence rate are accelerated.

In trial 3, the observed signals are the mixture of one normal distributed Gaussian source signal and one super-Gaussian speech signal. After scanning for 20000 data points, a snapshot of matrix $V(V = W * A)$ (here we denote it $V1$) from our modified algorithm is

$$V1 = \begin{bmatrix} 0.5482 & 0.0324 \\ -0.0118 & 0.8982 \end{bmatrix} \tag{A.13}$$

A snapshot of matrix V (here we denote it $V2$) from the original algorithm is

$$V2 = \begin{bmatrix} 0.7975 & 0.3396 \\ -0.0117 & 0.7021 \end{bmatrix} \tag{A.14}$$

The average signal-to-noise ratio is 31.10 (dB) and 21.49 (dB) respectively.

In trial 4, the observed signals are the mixture of two super-Gaussian speech signals. After scanning for 30000 data points, a snapshot of matrix V (we denote it $V1$) from our modified algorithm is

$$V1 = \begin{bmatrix} 1.9201 & -0.0134 \\ -0.0228 & 0.8288 \end{bmatrix} \tag{A.15}$$

A snapshot of matrix V (we denote it $V2$) from the original algorithm is

$$V2 = \begin{bmatrix} 1.0625 & -0.0044 \\ 0.2124 & 0.7544 \end{bmatrix} \tag{A.16}$$

The average signal-to-noise ratio is 37.17 (dB) and 29.34 (dB) for our modified algorithm and the original algorithm respectively.

In order to observe and compare the actual separation performance of our modified method and the original method, corresponding to trial 3 and 4, in Figure A.3 and Figure A.4, we separately list the source signals, the mixture signals, the separated signals by modified method and the separated signals by the original method

Experiment Results on Fixed Nonlinearity

As shown before, fixed nonlinearity function also can separate some particular source signal. In the following experiments, we test the separation ability of our modified algorithm and the original algorithm within same training process, by other words, to test their convergence speed on fixed nonlinearity respectively.

In trial 5, we use the nonlinearity $h_i(y_i) = -{y_i}^3$ on two sub-Gaussian source signals. After scanning for 30000 data points, a snapshot of matrix V (we denote it $V1$) from our modified algorithm is

$$V1 = \begin{bmatrix} 2.7818 & -0.0168 \\ 0.0178 & 2.6965 \end{bmatrix} \tag{A.17}$$

A snapshot of matrix V (we denote it $V2$) from the original algorithm is

$$V2 = \begin{bmatrix} 2.9506 & 0.0659 \\ 0.1809 & 2.9174 \end{bmatrix} \tag{A.18}$$

The average signal-to-noise ratio is 44 (dB) and 28.59 (dB) for our modified algorithm and the original algorithm respectively.

In trial 6, the nonlinearity function $h_i(y_i) = -{y_i}^{1/3}$ are used on two super-Gaussian speech signals. After scanning for 30000 data points, a snapshot of matrix V (we denote it $V1$) from our modified algorithm is

$$V1 = \begin{bmatrix} 4.6647 & 0.0098 \\ -0.0207 & 1.8332 \end{bmatrix} \tag{A.19}$$

A snapshot of matrix V (we denote it $V2$) from the original algorithm is

$$V2 = \begin{bmatrix} 4.5370 & 0.0369 \\ 0.1470 & 2.3859 \end{bmatrix} \tag{A.20}$$

The average signal-to-noise ratio is 46.25 (dB) and 33 (dB) for our modified algorithm and the original algorithm respectively.

In trial 7, we use the nonlinearity $h_i(y_i) = 1 - 2logsig(y_i)$ on two super-Gaussian speech source signals. After scanning for 30000 data points, a snapshot of matrix V (we denote is $V1$) from our modified algorithm is

$$V1 = \begin{bmatrix} 15.4118 & -0.0141 \\ -0.0492 & 5.2662 \end{bmatrix} \tag{A.21}$$

A snapshot of matrix V (we denote it $V2$) from the original algorithm is

$$V2 = \begin{bmatrix} 3.5174 & 0.8363 \\ 0.6984 & 2.5532 \end{bmatrix} \tag{A.22}$$

The average signal-to-noise ratio is 50.68 (dB) and 11.87 (dB) for our modified algorithm and the original algorithm respectively.

A.5 Gradient Descent with Momentum Method

In last section we propose a criterion for the adjustment of the learning rate in each iteration, but as we know, with the increase of the dimension number, the formula for solving out the learning rate also becomes much more complicated, so this criterion is not suitable for high dimension cases. Naturally we turn to another possibility of improving the algorithm, that is how to modify the searching direction p_k at kth iteration, which enables us to speed up the convergence rate.

For the gradient descent algorithm, we have the following iteration equation,

$$W_{k+1} = W_k + \eta_k p_k \tag{A.23}$$

η_k is the learning rate. $p_k = -\nabla J(W_k)$ is the negative gradient of the cost function $J(W)$ at W_k. Because the steepest descent direction of $J(W)$ at W_k is its negative gradient direction, so at W_{k+1}, the gradient of $J(W)$ along this direction should be zero, that is

$$\frac{d}{d\eta} J(W_k + \eta p_k)|_{\eta=\eta_k} = 0 \tag{A.24}$$

or

$$\nabla J(W_{k+1})^T p_k = 0 \tag{A.25}$$

which means that the gradient of $J(W)$ at W_{k+1} is perpendicular to the searching direction p_k. In other words, the searching directions between two continuous iteration steps are perpendicular. Now that have such a conclusion, we can assume that the value of W_{k+1} not only has relation with the searching direction at kth iteration, but also has relation with the searching direction two consecutive time steps before. The iteration equation is written as

$$W_{k+1} = W_k + \eta_k p_k + \eta_{k-1} p_{k-1} + \eta_{k-2} p_{k-2} \tag{A.26}$$

For high dimension cases, we find that our proposed modified method performs better than the original gradient descent with momentum algorithm of considering searching direction just one time step before.

Experiment Results on Two-Dimension Cases

In order to test whether such assumption can speed up the convergence, we also have made experiments between our modified algorithm and the original algorithm. The learning rates of W, γ, a, b are kept at [0.0001, 0.001, 0.01, 0.001]. The initializations of other parameters are the same as those used in the last section.

In trial 8, the observed signals are the mixture of one uniformly distributed sub-Gaussian source signal and one super-Gaussian speech signal. For the modified algorithm, after convergence, a snapshot of matrix V (here we denote it $V1$)

$$V1 = \begin{bmatrix} 1.3512 & 0.0238 \\ -0.0011 & 2.0102 \end{bmatrix} \tag{A.27}$$

Meanwhile, a snapshot of matrix V (here we denote it $V2$) from the original algorithm is

$$V2 = \begin{bmatrix} 1.0308 & 0.0459 \\ -0.0087 & 1.1305 \end{bmatrix} \tag{A.28}$$

The average signal-to-noise ratio is 50.16 (dB) and 34.66 (dB) for our modified algorithm and the original algorithm respectively.

In trial 9, we use one normally distributed Gaussian signal and one super-Gaussian speech signal as two source signals. For the modified algorithm, after convergence, a snapshot of matrix V (we denote it $V1$)

$$V1 = \begin{bmatrix} 0.6270 & 0.0197 \\ -0.0065 & 1.8962 \end{bmatrix} \tag{A.29}$$

Meanwhile, a snapshot of matrix V (we denote it $V2$) from the original algorithm is

$$V2 = \begin{bmatrix} 0.7802 & 0.3454 \\ -0.0089 & 1.1598 \end{bmatrix} \tag{A.30}$$

The average signal-to-noise ratio is 39.68 (dB) and 24.69 (dB) for our modified algorithm and the original algorithm respectively.

Corresponding to trial 8 and 9, in Figure A.5 and Figure A.6, we separately list the source signals, the mixture signals, the separated signals by modified method and the separated signals by the original method.

Besides, we have also made experiments under the cases of source signals are two uniformly distributed sub-Gaussian signals and two super-Gaussian speech signals. The performance graph is shown in Figure A.7 and Figure A.8 respectively, in which the solid line represents the signal-to-noise ratio curve of our modified algorithm, contrarily, the dotted line represents that of the original algorithm. We can see that compared with the original algorithm, when we put the previous two step's searching direction into consideration, actually the convergence rate can be improved.

Experiment Results on Eight-Dimension Cases

This modified method also can be extended to high dimension cases. We also have made experiments to compare the convergence performance between our modified method and the original method under such cases. The learning rates of W, γ, a, b are kept at [0.0001, 0.001, 0.01, 0.001]. The initializations of other parameters are the same as those used in the last section. The mixing matrix A is set as

$$A = \begin{bmatrix} 0.6 & 0.6 & 0.3 & 0.2 & 0.3 & 0.2 & 0.1 & 0.4 \\ 0.7 & 0.6 & 0.5 & 0.4 & 0.1 & 0.4 & 0.3 & 0.2 \\ 0.1 & 0.3 & 1.0 & 0.6 & 0.2 & 0.4 & 0.5 & 0.3 \\ 0.4 & 0.5 & 0.2 & 1.0 & 0.7 & 0.6 & 0.3 & 0.2 \\ 0.4 & 0.2 & 0.1 & 0.5 & 1.0 & 0.7 & 0.5 & 0.4 \\ 0.2 & 0.3 & 0.1 & 0.6 & 0.5 & 1.0 & 0.6 & 0.4 \\ 0.2 & 0.4 & 0.3 & 0.5 & 0.1 & 0.4 & 1.0 & 0.3 \\ 0.5 & 0.6 & 0.3 & 0.7 & 0.1 & 0.2 & 0.4 & 1.0 \end{bmatrix} \tag{A.31}$$

In trial 10, 8-channel uniform distributed sub-Gaussian signals are used. For the modified algorithm, after convergence, a snapshot of matrix V (we denote it $V1$)

$$V1 = \begin{bmatrix} -0.0113 & 0.8765 & -0.0078 & -0.0384 & 0.0273 & -0.0154 & 0.0398 & -0.0405 \\ 1.4953 & 0.0068 & 0.0107 & 0.0121 & 0.0040 & -0.0166 & 0.0115 & 0.0185 \\ -0.0008 & 0.0056 & 2.4763 & 0.0117 & -0.0023 & 0.0120 & 0.0048 & -0.0114 \\ -0.0039 & 0.1213 & -0.0126 & 1.8131 & 0.0017 & 0.0163 & -0.0127 & 0.0281 \\ -0.0127 & -0.0596 & 0.0080 & 0.0247 & 2.2034 & -0.0044 & 0.0178 & -0.0017 \\ -0.0069 & 0.0076 & -0.0148 & 0.0157 & -0.0053 & 2.0113 & -0.0092 & 0.0074 \\ 0.0071 & 0.0252 & -0.0060 & -0.0197 & 0.0082 & -0.0029 & 2.0963 & 0.0046 \\ 0.0002 & 0.0437 & -0.0132 & 0.0124 & 0.0071 & -0.0244 & -0.0143 & 2.1597 \end{bmatrix} \tag{A.32}$$

Meanwhile, a snapshot of matrix V (we denote it $V2$) from the original algorithm is

$$V2 = \begin{bmatrix} -0.0827 & 0.5603 & 0.0366 & -0.1758 & 0.0770 & -0.0058 & -0.0089 & -0.0499 \\ 0.9035 & 0.1073 & 0.0695 & -0.0846 & -0.0729 & 0.0376 & 0.0023 & -0.0841 \\ -0.0894 & 0.0098 & 1.8277 & 0.2743 & 0.0256 & 0.0957 & 0.0639 & 0.0389 \\ 0.0357 & 0.3972 & -0.1525 & 1.0227 & 0.0791 & -0.0192 & -0.0678 & -0.3028 \\ 0.1984 & -0.1978 & -0.0255 & 0.0196 & 1.5069 & 0.1208 & 0.0751 & 0.1212 \\ -0.0792 & 0.0050 & -0.0647 & 0.0436 & -0.0629 & 1.3417 & -0.0027 & 0.0510 \\ 0.0187 & 0.1305 & -0.0412 & 0.0135 & -0.0271 & -0.0129 & 1.4896 & -0.0419 \\ 0.2085 & 0.2681 & -0.0604 & 0.3047 & -0.1227 & -0.0754 & 0.0005 & 1.3817 \end{bmatrix} \tag{A.33}$$

In trial 11, we use 7-channel uniform distributed sub-Gaussian signals together with one speech signals. For the modified algorithm, after convergence, a snapshot of matrix V (here we denote it $V1$)

$$V1 = \begin{bmatrix} 0.0029 & 0.8721 & -0.0017 & -0.0217 & 0.0200 & -0.0131 & -0.0073 & -0.0468 \\ 1.5173 & -0.0098 & -0.0158 & 0.0341 & -0.0070 & 0.0070 & -0.0098 & 0.0323 \\ -0.0021 & 0.0046 & 2.4735 & -0.0059 & -0.0048 & -0.0045 & -0.0189 & -0.0526 \\ -0.0079 & 0.0842 & 0.0037 & 1.8998 & -0.0061 & 0.0147 & -0.0213 & 0.0599 \\ -0.0064 & -0.0447 & 0.0101 & -0.0023 & 2.2482 & 0.0119 & 0.0030 & -0.0006 \\ -0.0055 & 0.0020 & -0.0288 & -0.0193 & -0.0023 & 2.0720 & -0.0152 & -0.0210 \\ 0.0074 & 0.0280 & 0.0300 & 0.0286 & 0.0270 & 0.0153 & 2.1384 & -0.0222 \\ 0.0193 & 0.0145 & 0.0039 & 0.0276 & -0.0122 & 0.0149 & -0.0012 & 9.1458 \end{bmatrix} \tag{A.34}$$

Meanwhile, a snapshot of matrix V (here we denote it $V2$) from the original algorithm is

$$V2 = \begin{bmatrix} -0.0676 & 0.5641 & 0.0229 & -0.1692 & 0.0833 & 0.0078 & -0.0355 & -0.0011 \\ 0.8958 & 0.1084 & 0.0312 & -0.0452 & -0.0658 & 0.0308 & -0.0225 & -0.0436 \\ -0.0786 & 0.0100 & 1.8400 & 0.2507 & 0.0393 & 0.0753 & 0.0504 & -0.0422 \\ 0.0272 & 0.3943 & -0.1200 & 1.0835 & 0.0610 & -0.0067 & -0.0787 & 0.0435 \\ 0.2175 & -0.2081 & -0.0240 & -0.0029 & 1.5071 & 0.0841 & 0.0695 & -0.0091 \\ -0.0665 & 0.0018 & -0.0619 & 0.0096 & -0.0457 & 1.3856 & -0.0017 & -0.0156 \\ 0.0089 & 0.1403 & -0.0118 & 0.0370 & -0.0246 & 0.0070 & 1.4882 & -0.0433 \\ 0.0307 & 0.0262 & 0.0065 & -0.0015 & -0.0121 & 0.0083 & -0.0084 & 3.5756 \end{bmatrix} \tag{A.35}$$

The performance graphs of trial 10 and 11 are shown in Figure A.9 and Figure A.10. The simulation results demonstrate that our modified algorithm actually performs better than the original algorithm.

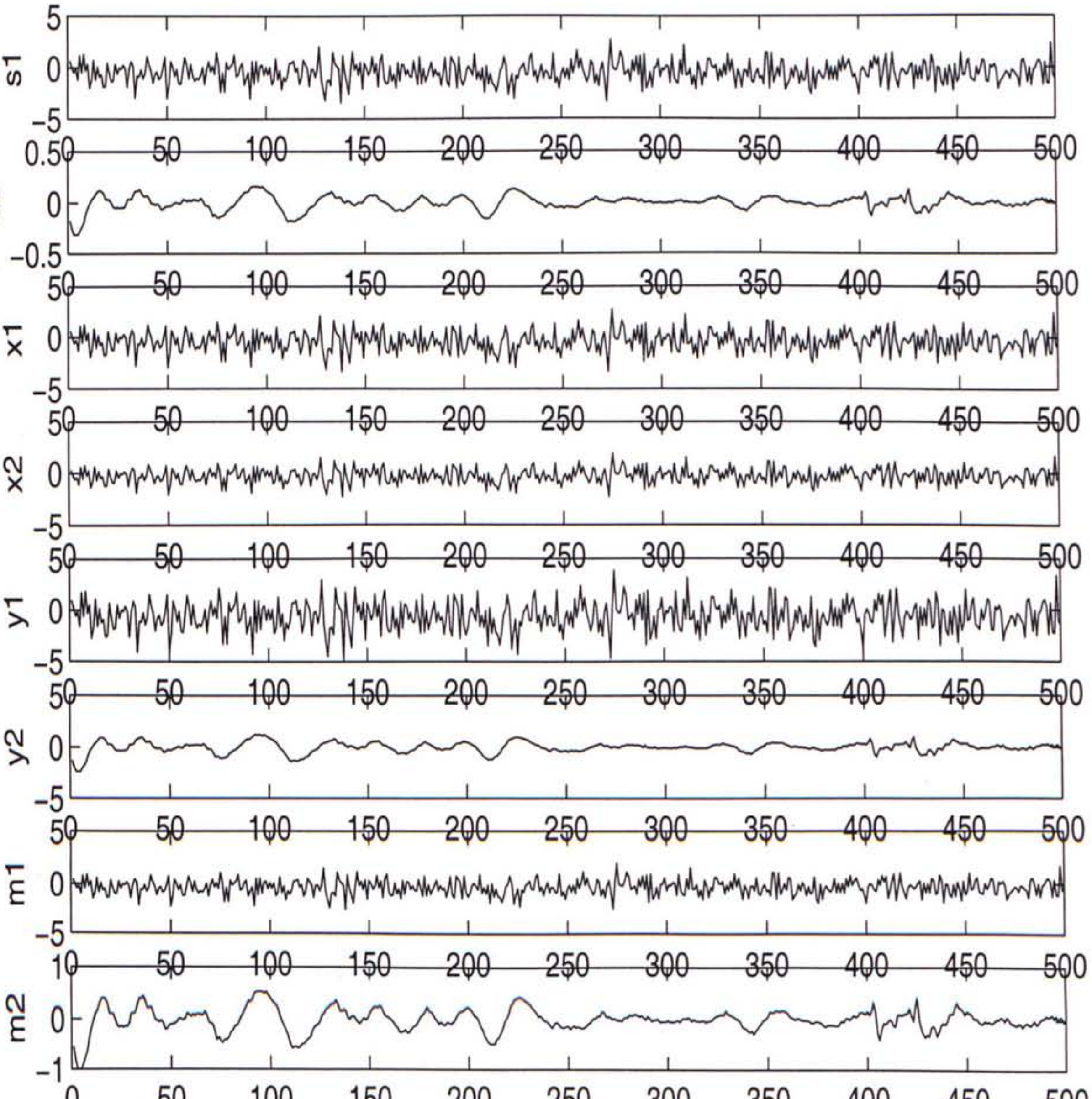


Figure A.3: Experiment results in trial 3, we list the original source signals (first two rows), mixed signals (3 and 4 rows), separated source signals by modified method 1 (5 and 6 rows) and separated source signals by original method (last two rows)

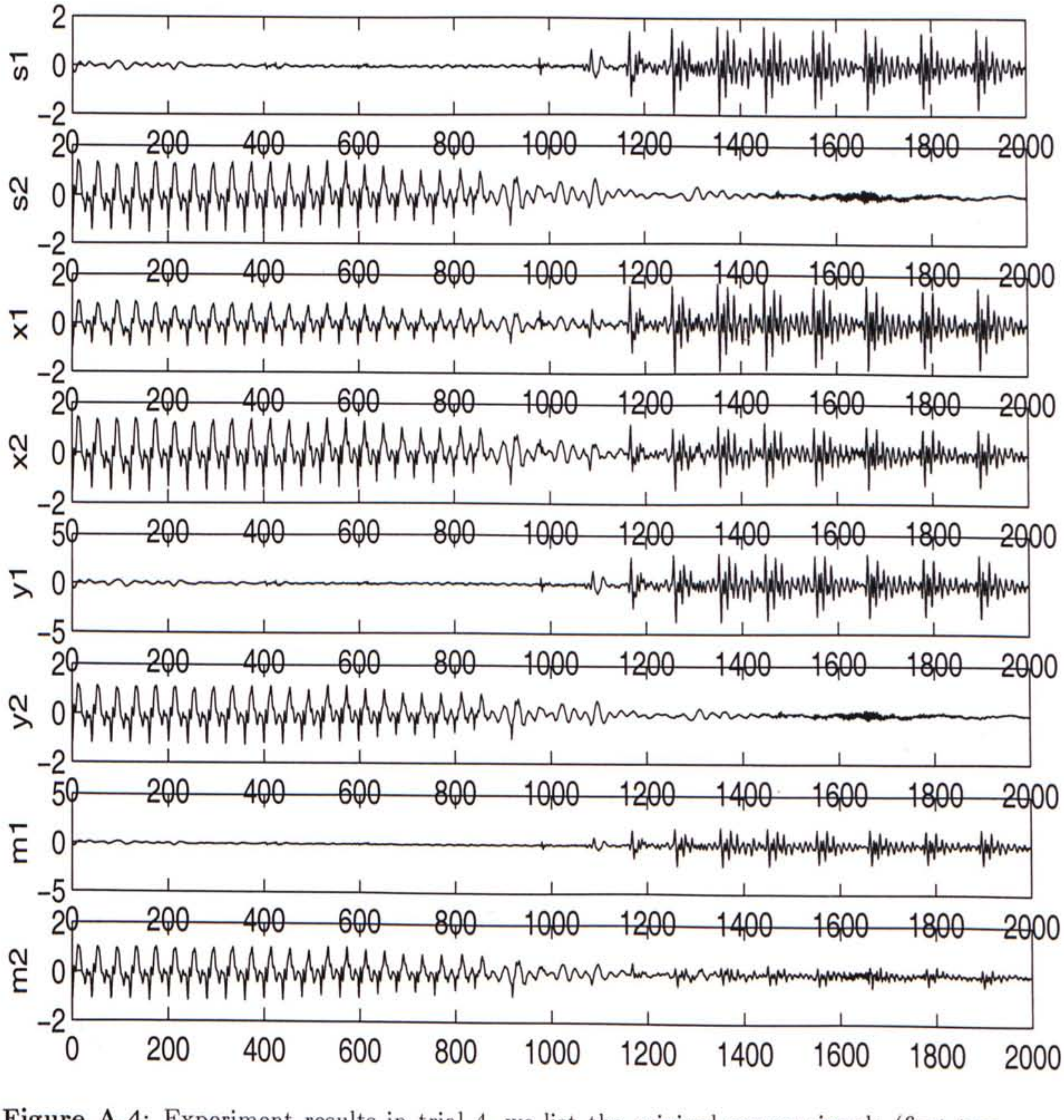


Figure A.4: Experiment results in trial 4, we list the original source signals (first two rows), mixed signals (3 and 4 rows), separated source signals by modified method 1 (5 and 6 rows) and separated source signals by original method (last two rows)

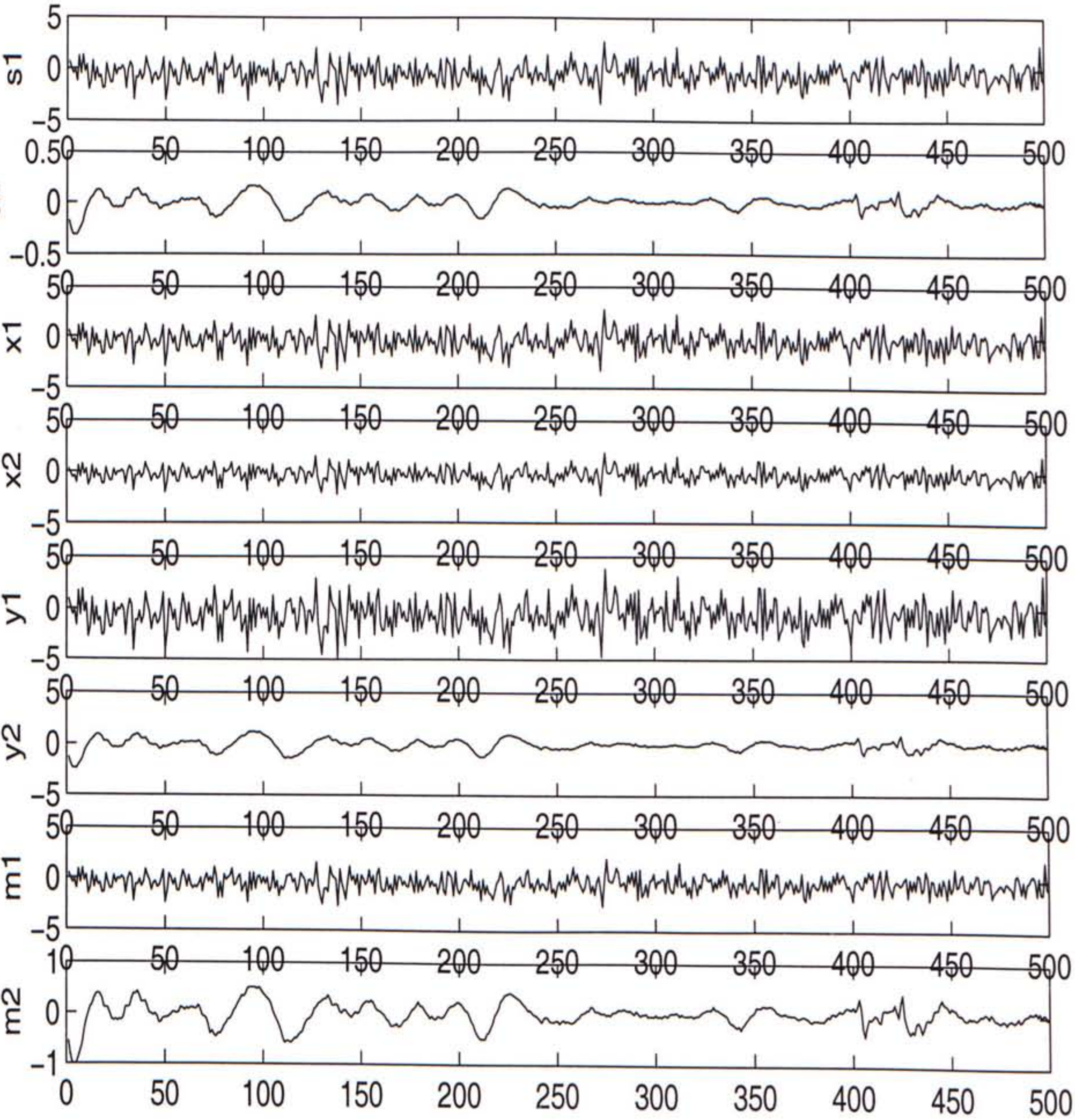


Figure A.5: Experiment results in trial 8, we list the original source signals (first two rows), mixed signals (3 and 4 rows), separated source signals by modified method 2 (5 and 6 rows) and separated source signals by original method (last two rows)

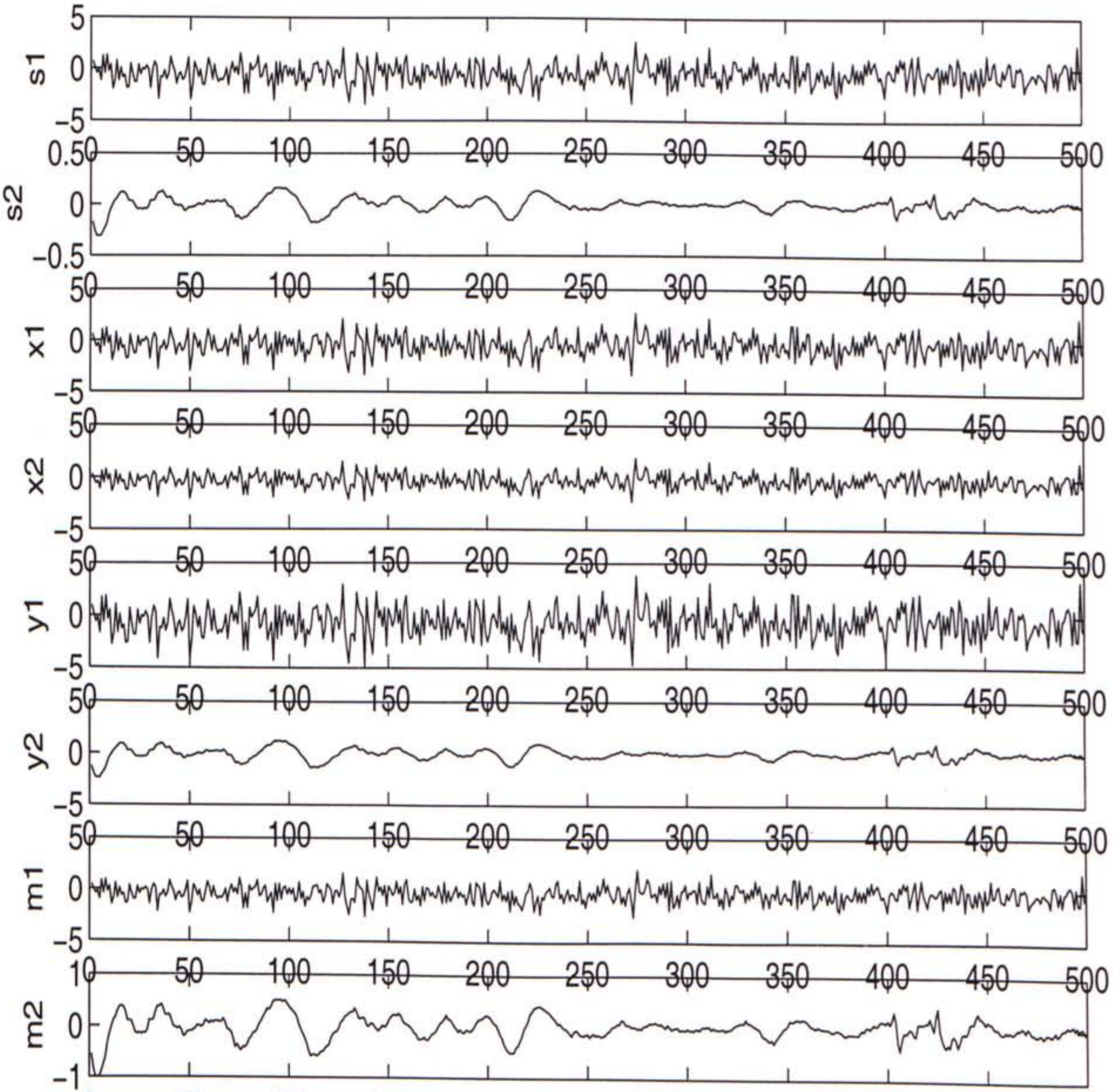


Figure A.6: Experiment results in trial 9, we list the original source signals (first two rows), mixed signals (3 and 4 rows), separated source signals by modified method 2 (5 and 6 rows) and separated source signals by original method (last two rows)

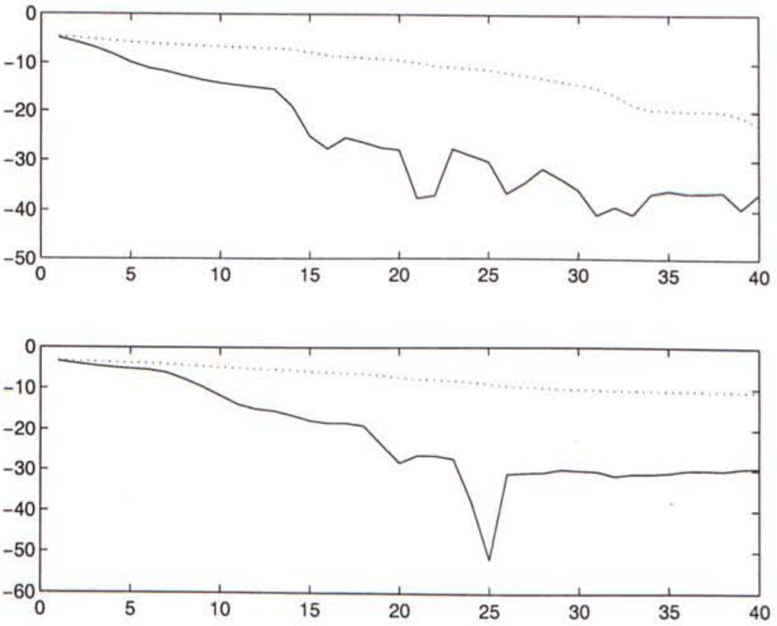


Figure A.7: Method 2 (solid) and original method (dotted) with 2 sub-Gaussian signals for 30,000 data

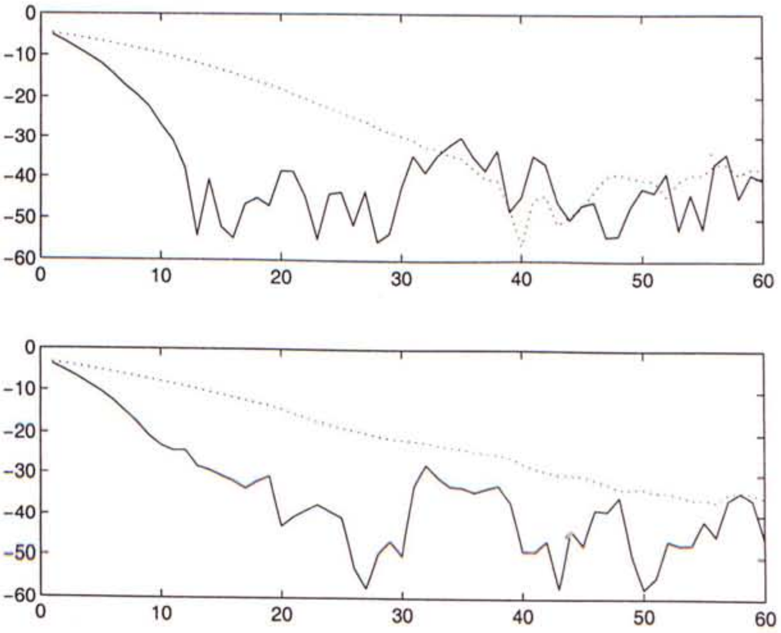


Figure A.8: Method 2 (solid) and original method (dotted) with 2 super-Gaussian signals for 30,000 data

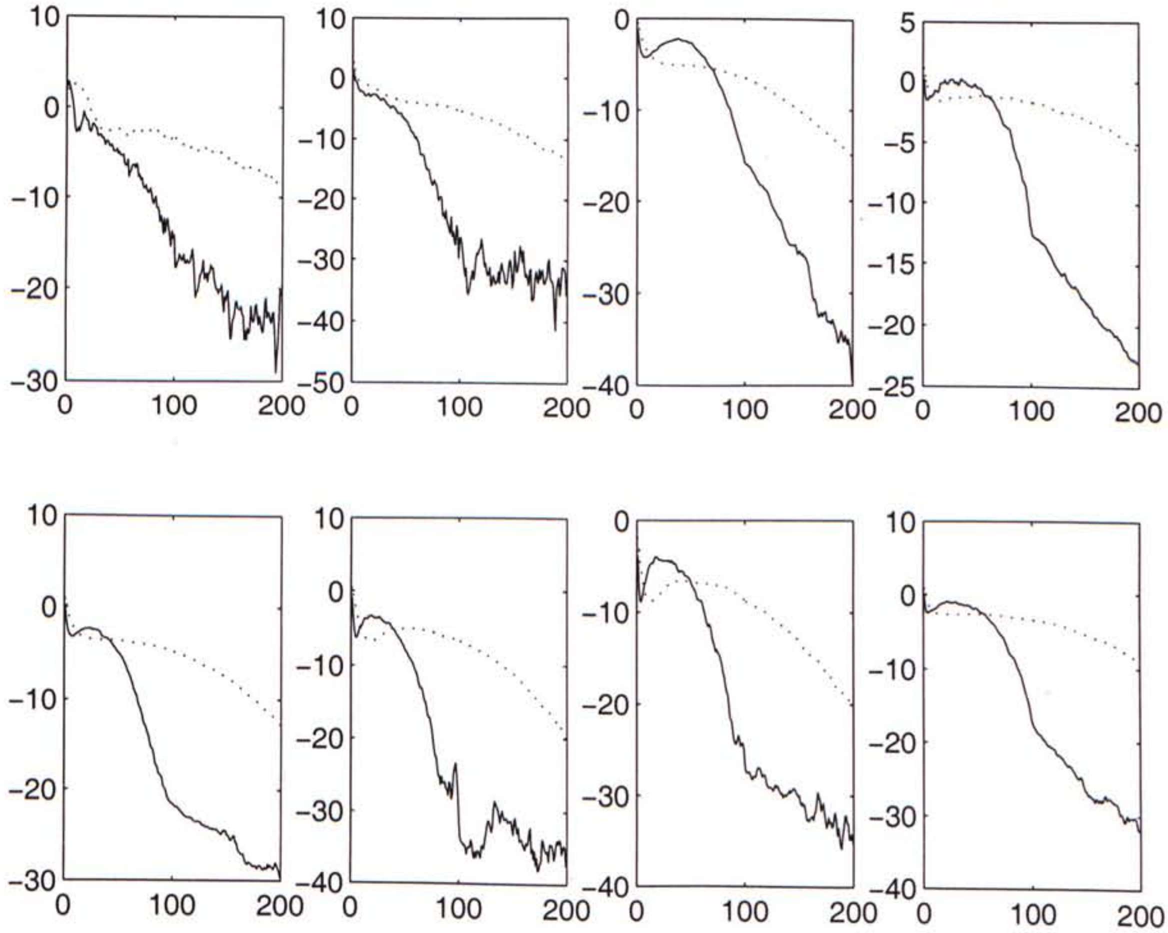


Figure A.9: Method 2 (solid) and old method (dotted) with 8 sub-Gaussian signals for 80,000 data

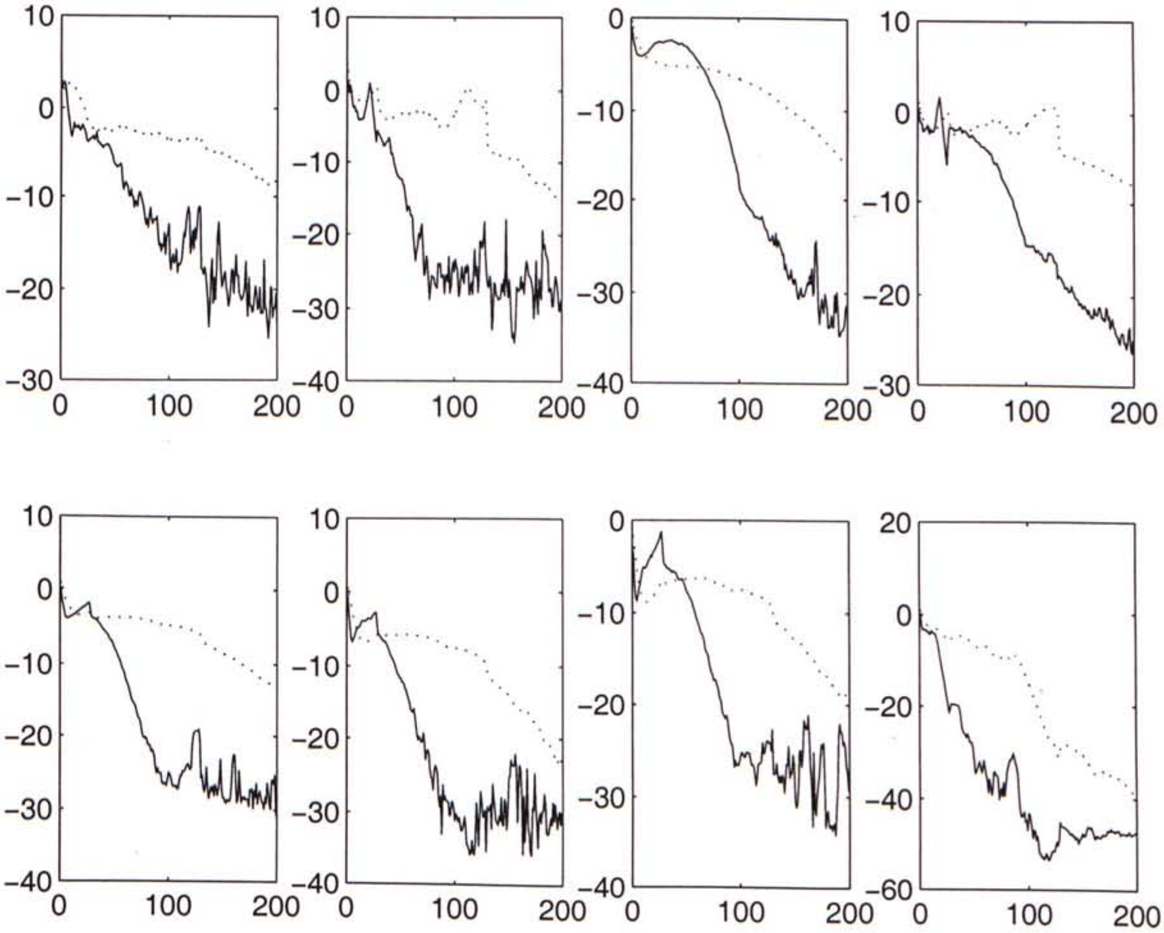


Figure A.10: Method 2 (solid) and old method (dotted) with mixture of sub-Gaussian and super-Gaussian source signals for 80,000 data

Bibliography

[1] S.-I. Amari, A. Cichocki, H. Yang (1996), "A new learning algorithm for blind signal separation", in G. Tesauro, D. Touretzky and T. Leen (eds), *Advances in Neural Information Processing 8*, The MIT Press, Cambridge, MA, pp. 757–763, 1996.

[2] N. Baba, Y. Mogami, M. Kohzaki (1994), Y. Shiraishi and Y. Yoshida, "A hybrid algorithm for finding the global minimum of error function of neural networks and its applications", *Neural Networks*, Vol. 7, No. 8, pp. 1253-1265, 1994

[3] A.D. Back, A.S. Weigend (1997), "A first application of independent component analysis to extracting structure from stock returns", *International Journal of Neural Systems*, Vol. 8, No. 4, pp. 473–484, 1997.

[4] R. Battiti and F. Masulli (1990), "BFGS Optimization for faster and automated supervised learning". *INCC 90 Paris, International Neural Network Conference*, 2, pp.757-760

[5] S. Becker and Y. Le Cun (1988), "Improving the convergence of backpropagation learning with second order methods", in *Proceedings of the 1988 Connectionist Models Summer School*, David S. Touretzky, Geoffrey E. Hinton, and Terrence J. Sejnowski, Eds. San Mateo, CA: Morgan Kaufmann, 1988, pp.29-37

[6] A.J. Bell and T.J. Sejnowski (1995), "An information-maximization approach to blind separation and blind deconvolution", *Neural Computation*, Vol. 7, pp. 1129–1159, 1995.

[7] J. Cardoso and A. Souloumiac (1993), "Blind Beamforming for non-Gaussian signals", *IEEE Proc. F.*, 140(6): 771-774, 1993

[8] J. Cardoso and B. Laheld (1996), "Equivariant adaptive source separation", *IEEE Trans. Signal Processing*, 44(12): 3017-3030, 1996

[9] J. Cardoso (1989), "Source separation using higher order moments", *International Conference on Acoustics, Speech and Signal Processing*, pp. 2109-2112, 1989

[10] J. Cardoso (1996), "Informax and Maximum Likelihood for Blind Source Separation",to appear in *IEEE Signal Processing Letters*

[11] A. Cichocki, R. Thawonmas and S. Amari (1997), "Dual cascade networks for blind signal extraction", in *Proc. 1997 International Conference on Neural Networks*, Houston, June, Vol. 4, pp. 2135-2140, 1997

[12] A. Cichocki and R. Unbehauen (1993), *Neural Networks for Optimization and Signal Processing*, John Wiley and Sons Press, 1993

[13] A. Cichocki and R. Unbehauen (1997), "Robust neural networks with on-line learning for blind identification and blind separation of sources", submitted to *IEEE Trans. on Circuits and Systems*,

[14] A. Cichocki, W. Kasprzak and S. Amari (1995), "Multi-layer neural networks with a local adaptive learning rule for blind separation of source signals", in *Proc. 1995 Int. Symp. on Nonlinear Theory and Applications*, Las Vegas, USA, December 1995, Vol. 1, pp. 61-66

[15] P. Comon (1994), "Independent component analysis - a new concept?", *Signal Processing*, Vol. 36, pp. 287-314, 1994

[16] C. Darken and J. Moody (1990), "Towards faster stochastic gradient search", in the book *Advances in Neural Information Processing Systems 4*, Morgan Kaufman, San Mateo 1990, pp. 1009-1016

[17] C. Darken, J. Chang and J. Moody (1992), "Learning rate schedules for faster stochastic gradient search", in *Nueral Networks for Signal Processing 2-Proc, 1992 IEEE Workshop*, IEEE Press, New York 1992

[18] Y. Deville and L. Andry (1995), "Application of blind source separation techniques to multi-tag contactless identification systems", in *Proc. 1995 Int. Symp. on Nonlinear Theory and Applications*, Las Vegas, USA, December 1995, Vol. 1, pp. 73-78

[19] S. E. Fahlman (1988), "Faster learning variations on backpropagation: an empirical study", in *Proc. 1988 Connectionist Models Summer School*, Morgan Kaufmann, Los Altos, CA 1988, pp. 38-51

[20] S. E. Fahlman and C. Lebiere (1990), "The cascade-correlation learning architecture", in the book *Advances in Neural Information Processing Systems 2*, Eds. D. S. Touretzky, Morgan Kaufmann, Los Altos, CA 1990, pp. 524-532

[21] A.M. George and I.H. Giddy (1983), *International Finance Handbook*, Wiley Press,

[22] P.C. Hallwood and R. MacDonald (1994), *International Money and Finance*, 2nd edition, Blackwell Press.

[23] J. Herault and C. Jutten (1986), "Space or time adaptive signal processing by neural network models", in J. S. Denker (ed.), *Neural Networks for Computing, Proceedings of AIP Conference*, American Institute of Physics, New York, pp. 206-211

[24] M. Herrmann and H. H. Yang (1996), "Perspective and limitation of self-organizing maps in blind separation of source signals", in *Progress in Neural Information Processing: Proceedings of ICONIP*96*, pp. 1211-1216, Springer, September 1996

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

[26] R. A. Jacobs (1988), "Increased rates of convergence through learning rate adaptation", *Neural Networks*, Vol. 1. pp. 295-307, 1988

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

[28] Z. B. Lai, Y. M. Cheung and L. Xu (1998a), "Further study of ICA in financial data analysis", submitted to 1999 *IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing* (NSIP'99)

[29] Z. B. Lai, Y. M. Cheung and L. Xu (1998b), "Further Study on Independent Components in Analysis of Financial Data", accepted by *Computational Finance* (CF99)

[30] Y. LeCun, P. Y. Simard, and B. A. Pearlmutter (1993), "Automatic learning rate maximization by on-line estimation of the Hessian's Eigenvectors", in *Advances in Neural Information Processing Systems 5 (NIPS*92)*, S. J. Hanson, J. Cowan. and L. Giles, Eds. San Mateo, CA: Morgan Kaufmann, 1993, pp.156-163

[31] M.F. Moller (1993), "A scaled conjugate gradient algorithm for fast supervised learning", *Neural Networks*, Vol. 6, no. 4, pp.525-533, 1993

[32] P. Pajunen and Juha Karhunen (1997), "Least-squares methods for blind source separation based on nonlinear PCA", submitted to *International Journal of Neural Systems*, October 1997

[33] B. A. Pearlmutter and L. C. Parra (1997), "Maximum likelihood blind source separation: A context-sensitive generalization of ICA", in M. C. Mozer, M. I. Jordan and T. Petsche (eds), *Advance in Neural Information Processing System 9 (NIPS*96)*,The MIT Press, Cambridge, MA, pp. 613-619, 1997

[34] B. T. Polyak (1990), "New method of stochastic approximation type", *Automat. Remote Control*, Vol. 51, 1990, pp.937-946

[35] K. Pope and R. Bogner (1994), "Blind separation of speech signals", *Proc. of the Fifth Australian Int. Conf. on Speech Science and Technology*, Perth, Western Australia, pp. 46-50

[36] M. Riedmiller and H. Braun (1992), "RPROP -a fast adaptive learning algorithm", Technical Report, University Karlsrube 1992

[37] R. Salomon and J. L. Hemmen (1996), "Accelerating backpropagation through dynamic self-adaptation", *Neural Networks*, Vol. 9, No. 4, pp. 589-601, 1996

[38] W. Schiffman, M. Joost and R. Werner (1992), "Optimization of the backpropagation algorithm for training multilayer perceptrons", Technical Report, University of Koblenz, Institute of Physics, 1992

[39] F. J. Solis and J. B. Wets (1981), "Minimization by random search techniques", *Mathematics of Operations Research*, 6, 19-30, 1981

[40] T. Tollenaere (1990), "Super SAB: fast adaptive backpropagation with good scaling properties", *Neural Networks*, Vol. 3, pp. 561-573, 1990

[41] A. C. Veitch and G. Holmes (1991), "A modified quickprop algorithm", *Neural Computation*, Vol. 9, pp. 310-311, 1991

[42] R. Vitthal, P. Sunthar and C. D. Rao (1995), "The generalized proportional-integral-derivative (PID) gradient descent back propagation algorithm", *Neural Networks*, Vol. 8, No. 4, pp. 563-569, 1995

[43] L. Wang and J. Karhunen (1995a), "A unified neural bigradient algorithm for robust PCA and MCA", to appear in *Int. J. of Neural Systems*

[44] L. Wang, J. Karhunen and E. Oja (1995b), "A bigradient optimization approach for robust PCA, MCA, and source separation", in *Proc. 1995 IEEE Int. Conf. on Neural Networks*, Perth, Australia, November 1995, pp. 1684-1689

[45] L. Wang, J. Karhunen, E. Oja and R. Vigario (1995c), "Blind separation of sources using nonlinear PCA type learning algorithm", in *Proc. Int. Conf. on Neural Networks and Signal Processing*, Nanjing, China, December 1995, pp. 847-850

[46] L. Xu, C.C. Cheung, and S.-I. Amari (1998),"Learned Parametric Mixture Based ICA Algorithm", *Neurocomputing*, Volume 22, Issue 1-3, pp. 69-80, 1998

[47] L. Xu (1998), "Bayesian Kullback Ying-Yang dependence reduction theory", *Neurocomputing*, Volume 22, Issue 1-3, pp. 81-111, 1998

[48] H. H. Yang, S. Amari and A. Cichocki (1996), "Information back-propagation for blind separation of sources in nonlinear mixture", Riken Technical Report BIP-96-0013

[49] X. H. Yu, G. A. Chen and S. X. Cheng (1995), "Dynamic learning rate optimization of the backpropagation algorithm", *IEEE Transactions on Neural Networks*, Vol. 6, No. 3, 1995

[50] X. H. Yu and G. A. Chen (1997), "Efficient backpropagation learning using optimal learning rate and momentum", *Neural Networks*, Vol. 10, No. 3, pp. 517-527, 1997

[51] B. T. Zhang (1994), "Accelerated learning by active example selection", *International Journal of Neural Systems*. Vol. 5, No. 1, pp. 67-75, 1994