A MAP-BASED APPROACH TO BLIND NONLINEAR UNDERDETERMINED MIXTURE

C. WEI, W.L. WOO, S.S. DLAY and L.C. KHOR
School of Electrical, Electronic and Computer Engineering
University of Newcastle
Newcastle upon Tyne, NE1 7RU
United Kingdom
Email: w.l.woo@ncl.ac.uk

Abstract: In this paper, a new learning algorithm is proposed to solve the separation problem of the blind nonlinear underdetermined mixtures. The mixing system is characterised by the post-nonlinear structure and concurrently the number of sensors is less than the number of sources. The proposed algorithm utilises the Generalised Gaussian Distribution to model the prior probability distribution of the source signals and the mixing variables. A novel iterative technique based on alternate optimisation within the Bayesian framework has been developed for estimating the source signals, mixing matrix and the nonlinear distortion. In this paper, it is shown that through formal Bayesian derivation the update of the mixing matrix can be decomposed into two separate constituents given by the linear and nonlinear parts. Furthermore, the post-nonlinear distortion functions in the mixing model are approximated by a set of polynomials and the coefficients are found by solving a least square error problem. Simulations have been carried out to verify the effectiveness in separating signals under nonlinear and underdetermined conditions. An average margin of 130% improvement has been obtained when compared with the existing linear algorithm.

Key-Words: Independent Component Analysis (ICA), Maximum a Posteriori (MAP), blind source separation, post-nonlinear model, underdetermined mixture and statistical signal processing.

1. INTRODUCTION

The conventional Independent Component Analysis (ICA) aims to recover unknown statistically independent sources $\mathbf{s} = [s_1, ..., s_N]^T$ from a set of observations $\mathbf{x} = [x_1, ..., x_N]^T$, which are presupposed to be linear and instantaneous mixing of the original sources. The canonical model of the mixture is given by

$$\mathbf{x} = \mathbf{A}\mathbf{s} + \mathbf{n} \tag{1}$$

where $\mathbf{A}$ is conventionally assumed to be a square matrix with dimension $N \times N$, and $\mathbf{n}$ represents the residue of decomposition or, in real applications, noise. Based on (1), many de-mixers have been proposed and a review of existing ICA techniques can be found in [1-6]. Due to the diverse range of applications, ICA has been perceived to be attractive and promising. Nevertheless, the assumptions of $\mathbf{A}$ with dimension $N \times N$ and the mixing process is linear in (1) are rather restrictive and have subsequently narrowed down the scope of utilisation. Recently, a new research direction has emerged by relaxing either of these two assumptions. Some works are already underway which have extended the study of the ICA on linear and complete[1] mixtures to either linear underdetermined mixture or nonlinear complete mixture exclusively [7-20].

ICA with underdetermined mixture is a branch of ICA family where the number of observation $\mathbf{x}$ is less than the number of sources $\mathbf{s}$. In other words, the mixing matrix $\mathbf{A}$ becomes a rectangular matrix with dimension $M \times N$ where $M < N$. Hence, it does not admit the left invertible matrix as conventional ICA does. A number of methodologies have already been proposed to estimate the sources [4]. The simplest method assumes the mixing matrix is known and a de-mixing model is built which utilise the pseudoinverse of mixing matrix. In the noise free case, it has the form of

$$\tilde{\mathbf{s}} = \mathbf{A}^T \left(\mathbf{A}\mathbf{A}^T \right)^{-1} \mathbf{x} \tag{2}$$

From (2) it can be inferred that the optimal estimation of the source signals can still be found but the solution favours the Gaussian distribution. In ICA, however, non-gaussian source signals are of primary interest. In

[1] Complete mixture refers to a mixture where the number of sources equals to the number of sensors.

addition, the assumption that $\mathbf{A}$ is known is often violated and hence, this prevents the de-mixer from restoring the original signals in a blind mode. In the underdetermined Blind Signal Separation (BSS) and ICA, estimating the mixing matrix $\mathbf{A}$ can be regarded as part of estimating $\mathbf{s}$. One technique is to use the Maximum Likelihood (ML) estimation given as follows:

$$\mathbf{s}_{ML} = \arg\max_{\mathbf{s}} P(\mathbf{x}|\mathbf{A}) \tag{3}$$

Similar to the ML approach, the Maximum a Posteriori (MAP) probability approach can also be applied to estimate $\mathbf{s}$ and $\mathbf{A}$ as expressed below.

$$\left(\tilde{\mathbf{s}}, \tilde{\mathbf{A}}\right) = \arg\max_{\mathbf{s},\mathbf{A}} P(\mathbf{s}, \mathbf{A}|\mathbf{x}) \tag{4}$$

Generally, the estimation of both $\mathbf{s}$ and $\mathbf{A}$ can be outlined as follows: Firstly, the cost function based on the joint probability of $\mathbf{s}$ and $\mathbf{A}$ is constructed. Both estimates of $\mathbf{s}$ and $\mathbf{A}$ are then updated (or refined) until their optimal values [21] are found. An alternative approach is to establish the marginal probability of $\mathbf{s}$ and $\mathbf{A}$, and jointly optimised as follows:

$$\begin{aligned}\left(\tilde{\mathbf{s}}, \tilde{\mathbf{A}}\right) &= \arg\max_{\mathbf{s},\mathbf{A}} P(\mathbf{s}, \mathbf{A}|\mathbf{x}) = \arg\max_{\mathbf{s},\mathbf{A}} \frac{P(\mathbf{s}, \mathbf{A}, \mathbf{x})}{P(\mathbf{x})} \\ &= \arg\max_{\mathbf{s},\mathbf{A}} \frac{P(\mathbf{s}|\mathbf{A}, \mathbf{x})P(\mathbf{A}|\mathbf{x})}{P(\mathbf{x})} \propto \arg\max_{\mathbf{s},\mathbf{A}} P(\mathbf{s}|\mathbf{A}, \mathbf{x})P(\mathbf{A}|\mathbf{x})\end{aligned} \tag{5}$$

Both methods in (4) and (5) can be complicated and the computational complexity will be very high especially when non-Gaussian variables are of primary interest. In this paper, instead of maximising the joint probability directly we propose to use an iterative optimisation approach where the maximisation of (4) is carried out in an alternate manner between the two separate steps as follows:

$$\tilde{\mathbf{s}} = \arg\max_{\mathbf{s}} P(\mathbf{s}|\mathbf{A}, \mathbf{x}) \tag{6}$$

$$\tilde{\mathbf{A}} = \arg\max_{\mathbf{A}} P(\mathbf{A}|\mathbf{x}) = \arg\max_{\mathbf{A}} \int P(\mathbf{x} \mid \mathbf{A}, \mathbf{s})P(\mathbf{A})P(\mathbf{s})d\mathbf{s} \tag{7}$$

Detailed derivation of (6) and (7) will be discussed in Section 4.

Until now, the study of ICA with underdetermined mixture has concentrated solely on the linear mixture model (1). However, linear ICA methods will fail to recover the original source signals if the mixing environment contains nonlinear distortion. In recent surveys, many applications have been found to involve some degree of nonlinear mixing. For example, in biomedical signal processing, many physiological signals such as the auditory nervous system are nonlinearly distorted and identification of nonlinearity of the system should be taken into consideration [22, 23]. Another example is the nonlinear distortion introduced by sensitive sensors such as carbon-button microphones. Hence, nonlinear ICA solutions have drawn considerable amount of attention in science and industry for the practical reasons [24, 25]. A well-known but simple nonlinear mixture is the post-nonlinear model proposed by Taleb and Jutten [26], which includes a linear mixing matrix followed by one layer of one-to-one nonlinear distortion function. The model is particularly suited for problems that involve the use of nonlinear sensors. The separability analysis of the post nonlinear mixture has also been derived explicitly in [26, 27]. However, the algorithm proposed in [26] for the post-nonlinear model is still by far limited to the complete case where the number of observed signals is equal to the number of sources.

The contribution of this paper is to tackle the problem where the mixing process is characterised by the post-nonlinear underdetermined mixture. As far as the authors are concerned, this problem has not been previously undertaken and this is the first work to bridge the gaps between these two areas and to provide a principled approach towards finding the solution. The fundamental theory of our algorithm is based on the Bayesian framework. The MAP-based cost functions are derived to iteratively estimate $\mathbf{s}$ and $\mathbf{A}$ alternately. We further show that through formal Bayesian derivation, the update of the mixing matrix can be decomposed into two separate constituents given by the linear and nonlinear parts. Between these two constituents, the nonlinear part has more effects on the quality of signal recovery since it compensates for the lost details caused by the nonlinearity. The implementation of the nonlinear part requires an appropriate assumption on the nonlinear function whose shape is as close as possible to the true one. Unfortunately, due to the blind nature of the problem, it is difficult to make any valid assumptions about the nonlinear mixing process and therefore, this leads to nonlinear function mismatch. To minimise this mismatch, this paper

proposes a self-adaptive algorithm using polynomials where the adaptation is designed to provide the optimal estimation of the true nonlinear mixing process.

The paper is organised as follows: In Section 2, the post nonlinear underdetermined mixing model is described. In Section 3, the Generalised Gaussian Distribution (GGD) model is introduced to model the source signals and mixing matrix. In Section 4, the cost functions for estimating the source signals and the mixing matrix are derived; furthermore, an effective method for adapting the polynomials to minimise the nonlinearity mismatch is present and the updates of the hyper-parameters involved in the estimation algorithm is lay out. Finally, simulation results and analysis are presented in Section 5 to verify the effectiveness at the proposed algorithm.

2. POST NONLINEAR UNDERDETERMINED MIXING MODEL

Fig.1 shows the post nonlinear underdetermined mixing model. It is featured by a linear mixing matrix $\mathbf{A}$ whose dimension is $M \times N$ (where the number of observation is less than the number of sources, i.e. $M < N$) and a layer of one-to-one nonlinear distortion function $f_m(.)$. Assuming that the nonlinearity in the mixture is zero-preserving, the post nonlinear underdetermined mixing model for the observations $\mathbf{x} = [x_1, ...x_m, ..., x_M]^T$ can be expressed as

$$\mathbf{x} = F(\mathbf{As}) + \mathbf{n} \tag{8}$$

where $F(\mathbf{As}) = \left[f_1(\sum_{n=1}^{N} a_{1n} s_n), \quad \cdots \quad f_m(\sum_{n=1}^{N} a_{mn} s_n), \quad \cdots \quad f_M(\sum_{n=1}^{N} a_{Mn} s_n) \right]^T$, $\mathbf{s} = [s_1, ...s_n, ..., s_N]^T$ and $\mathbf{n} = [n_1, ...n_m, ..., n_M]^T$ are the mixing process, the source signals and noise, respectively. In current literature, nonlinear ICA is achieved by combining with different kinds of neural networks. These methods can be generally classified into two categories: generative approach and signal transformation method. The former aims to find a specific model to describe how the observations are generated and the solution consists of estimating both the source signals and the mixing mapping whereas for the signal transformation method, a separation system is firstly constructed and the unknown source signals are then estimated directly at the

output. In this paper, the generative approach is adopted. An optimal solution of $\mathbf{s}$ in (8) is difficult to obtain since $\mathbf{s}$ is both embedded in an underdetermined mixture and nonlinearly distorted by the function $f_m(.)$. In general, there exist infinite numbers of solutions to $\mathbf{s}$ which result from insufficient information extracted from the nonlinear underdetermined mixture. In order to narrow down the solution set, this paper proposes to use the MAP estimation which allows prior knowledge in the form of statistical information to be incorporated into the solution. As a result, the proposed MAP algorithm will seek the most probable value given the available observations. Following (6), the first step to derive a MAP-based algorithm to recover $\mathbf{s}$ is to consider the posterior probability of $\mathbf{s}$ given $\mathbf{x}$ and $\mathbf{A}$:

$$\begin{aligned} P(\mathbf{s}|\mathbf{x},\mathbf{A}) &\propto P(\mathbf{x}|\mathbf{s},\mathbf{A})P(\mathbf{s}|\mathbf{A}) \\ &\propto \ln P(\mathbf{x}|\mathbf{s},\mathbf{A}) + \ln P(\mathbf{s}|\mathbf{A}) \\ &\propto \ln P(\mathbf{x}|\mathbf{s},\mathbf{A}) + \ln P(\mathbf{s}) \end{aligned} \tag{9}$$

One key advantage of the formulation in (9) lies in the fact that it offers increased flexibility to accommodate various kinds of probability distributions. In (9), the prior knowledge of $P(\mathbf{s})$ is used. In this paper, we utilise the GGD model as the general model for the probability distribution function (PDF).

3. GENERALISED GAUSSIAN DISTRIBUTION (GGD) MODEL

The GGD model [28] is defined as follows:

$$g(u;p,\lambda) = \frac{\lambda p}{2\Gamma(1/p)} e^{-(\lambda|u|)^p} \qquad g(.) = \begin{cases} \text{super-gaussian, } 0<p<2 \\ \text{gaussian, } \quad p=2 \\ \text{sub-gaussian, } p>2 \end{cases} \tag{10}$$

where $\Gamma(.)$ is the standard gamma function, λ is the inverse of generalised variance, p controls the shape of distribution and therefore the kurtosis of the signal. The GGD model has been used to model a wide range of distribution functions. Although there exist more powerful PDF approximating models such as the Gaussian Mixture Model (GMM), the GGD model is preferred since it requires substantially lower computational complexity. Furthermore, the parameters of the GGD model can be easily estimated either on-

line or off-line during the update process.

Accordingly, we assume that the source signals $\mathbf{s}=[s_1,.....,s_N]^T$ are mutually independent with each other and the prior distribution of $\mathbf{s}$ is modelled by GGD. Not necessarily a limiting factor but for the sake of simplicity, we choose to model the prior probability distribution of $\mathbf{A}$ as the GGD model as well. The probability distribution of $\mathbf{s}$ and $\mathbf{A}$ may then assume the following forms:

$$\begin{aligned}\ln P(\mathbf{s}) &= \ln P(s_1,.....,s_N) \\ &= \ln\left(\frac{\lambda_{s_1} p_{s_1}}{2\Gamma(1/p_{s_1})}\exp\left(-\left(\lambda_{s_1}|s_1|\right)^{p_{s_1}}\right)\right)+\cdots+\ln\left(\frac{\lambda_{s_N} p_{s_N}}{2\Gamma(1/p_{s_N})}\exp\left(-\left(\lambda_{s_N}|s_N|\right)^{p_{s_N}}\right)\right) \\ &\propto -\sum_{n=1}^{N}\left(\lambda_{s_n}|s_n|\right)^{p_{s_n}}\end{aligned} \tag{11}$$

$$\begin{aligned}\ln p(\mathbf{A}) &= \ln\left(\frac{\lambda_{\mathbf{A}} p_{\mathbf{A}}}{2\Gamma(1/p_{\mathbf{A}})}\exp\left(-\sum_{n=1}^{N}\sum_{m=1}^{M}\left(\lambda_{\mathbf{A}}|a_{mn}|\right)^{p_{\mathbf{A}}}\right)\right) \\ &\propto -\sum_{n=1}^{N}\sum_{m=1}^{M}\left(\lambda_{\mathbf{A}}|a_{mn}|\right)^{p_{\mathbf{A}}}\end{aligned} \tag{12}$$

where λ_{s_n} and p_{s_n} are the inverse of generalised variance and shape distribution parameter, respectively for the n^{th} signal source while $\lambda_{\mathbf{A}}$ and $p_{\mathbf{A}}$ represents the same parameters but for the mixing matrix $\mathbf{A}$.

Following similar line of derivation, the joint probability distribution of the noise becomes:

$$\begin{aligned}\ln P\left(\mathbf{x}\middle|\mathbf{A},\mathbf{s}\right) &= \ln P(\mathbf{n}) \\ &= \ln P(\mathbf{x}-F(\mathbf{As})) \\ &\propto -\sum_{m=1}^{M}\left(\lambda_{n_m}\left|x_m - f_m(\sum_{n=1}^{N}a_{mn}s_n)\right|\right)^{p_{n_m}}\end{aligned} \tag{13}$$

where λ_{n_m} and p_{n_m} are the inverse of generalised variance and distribution parameter respectively for the m^{th} noise variable. For simplicity, we assume that the noise is Gaussian distributed i.e. $p_n=2$. Therefore, (13) can be simplified to the following:

$$\begin{aligned}\ln P(\mathbf{n}) &\propto -\sum_{m=1}^{M}\left(\lambda_{n_m} \left| x_m - f_m(\sum_{n=1}^{N} a_{mn} s_n) \right| \right)^2 \\ &= -\left\| diag\left(\lambda_{n_1},\cdots,\lambda_{n_M}\right)\left(\mathbf{x} - F(\mathbf{As})\right)\right\|^2 \end{aligned} \tag{14}$$

As stated in Section 1, $\mathbf{n}$ can be viewed as the residue of decomposition. In other words, (14) measures the Mean Square Error (MSE) of the residue between $\mathbf{x}$ and $F(\mathbf{As})$.

4. ESTIMATIONS OF THE PARAMETERS

4.1 Source signal estimation

The proposed learning algorithm first targets on estimating the original source signals which is in accord to the goal of BSS and ICA. Following the discussion on MAP in Section 2, the GGD model can be merged with (9) to form the cost function of the proposed generative network as:

$$\tilde{\mathbf{s}} = \arg\max_{\mathbf{s}} J\left(\mathbf{s}\right) \tag{15}$$

where

$$\begin{aligned} J\left(\mathbf{s}\right) &= \ln P(\mathbf{x}|\mathbf{s},\mathbf{A}) + \ln P(\mathbf{s}) \\ &\propto -\left\| diag\left(\lambda_{n_1}^2,\cdots,\lambda_{n_M}^2\right)\left(\mathbf{x} - F(\mathbf{As})\right)\right\|^2 - \sum_{n=1}^{N}\left(\lambda_{s_n}\left|s_n\right|\right)^{p_{s_n}} \end{aligned} \tag{16}$$

We note that (15) and (16) can be utilised for most situations under nonlinear underdetermined mixture. The derivative of the cost function with respect to $\mathbf{s}$ simply becomes

$$\nabla_{\mathbf{s}} J(\mathbf{s}) = 2diag\left(\lambda_{n_1}^2,\cdots,\lambda_{n_M}^2\right)\mathbf{A}^T diag(F'(\mathbf{As}))(\mathbf{x} - F(\mathbf{As})) - diag\left(\lambda_{s_n}^{p_{s_n}}\right) diag\left(p_{s_n}\right) diag\left(\left|s_n\right|\right)^{(p_s-2)}\mathbf{s} \tag{17}$$

where $F'(\mathbf{As}) = \left[f_1'(\sum_{n=1}^{N} a_{1n}s_n),\ \cdots\ f_m'(\sum_{n=1}^{N} a_{mn}s_n),\ \cdots\ f_M'(\sum_{n=1}^{N} a_{Mn}s_n) \right]^T$ and $f_m'(\cdot)$ is the first order derivative of $f_m(\cdot)$. It can be easily inferred that the optimal solution of $\mathbf{s}$ has the property of $\nabla_{\mathbf{s}} J(\mathbf{s}) = 0$. The adaptation of the sources can follow the gradient-based learning algorithm which is intended to drive $\mathbf{s}$

to reach the maxima of $J(\mathbf{s})$:

$$\mathbf{s}(t+1) = \mathbf{s}(t) + \mu_s \nabla_s J(\mathbf{s}) \tag{18}$$

where μ_s is the learning rate of the source estimation.

4.2 Mixing matrix estimation

In ICA, the direct estimation of the mixing system can be regarded as the key feature which distinguishes the generative approach from the conventional signal transformation method. The source signal estimation in (15) requires that the mixing matrix $\mathbf{A}$ has been estimated prior to iterating the update recursion in (18). Following (7), the mixing matrix can be estimated by constructing a separate cost function for $\mathbf{A}$ based on its marginal probability given the available observed data as

$$\begin{aligned} P(\mathbf{A}|\mathbf{x}) &= \int P(\mathbf{A},\mathbf{s}\,|\,\mathbf{x}) d\mathbf{s} \\ &\propto \int P(\mathbf{x}\,|\,\mathbf{A},\mathbf{s}) P(\mathbf{A}) P(\mathbf{s}) d\mathbf{s} \\ &\propto \ln P(\mathbf{A}) + \ln \int P(\mathbf{x}\,|\,\mathbf{A},\mathbf{s}) P(\mathbf{s}) d\mathbf{s} \end{aligned} \tag{19}$$

Equation (19) is seen to consist of two components: the prior probability distribution of $\mathbf{A}$ and an integral of log likelihood function with the source signals distribution function. When dealing with the integration of the second term, derivation of the integration becomes difficult and intractable since the mixture is underdetermined. To overcome this problem, we apply an approximation by using the Gaussian integral as follows:

$$\int y(\mathbf{s}) d\mathbf{s} \approx (2\pi)^{k/2}\, y(\tilde{\mathbf{s}}) \det\left(-\nabla\nabla \ln y(\tilde{\mathbf{s}})\right)^{-1/2} \tag{20}$$

where $\nabla\nabla$ denotes the Hessian function around $\tilde{\mathbf{s}}$ and $\tilde{\mathbf{s}}$ can be computed in (15). If we define $y(\tilde{\mathbf{s}}) = P(\mathbf{x}\,|\,\mathbf{A},\tilde{\mathbf{s}}) P(\tilde{\mathbf{s}})$, the integration then becomes

$$\int P(\mathbf{x}\,|\,\mathbf{A},\mathbf{s}) P(\mathbf{s}) d\mathbf{s} \approx (2\pi)^{k/2} P(\mathbf{x}\,|\,\mathbf{A},\tilde{\mathbf{s}}) P(\tilde{\mathbf{s}}) \det\left(-\nabla\nabla\left(\ln\left(P(\mathbf{x}\,|\,\mathbf{A},\tilde{\mathbf{s}}) P(\tilde{\mathbf{s}})\right)\right)\right)^{-1/2} \tag{21}$$

Hence, the cost function is established by substituting (21) into (19) as

$$\begin{aligned}\tilde{\mathbf{A}} &= \arg\max_{\mathbf{A}}\left(\ln P(\mathbf{A})+\ln P(\tilde{\mathbf{s}})+\ln P(\mathbf{x}\mid\mathbf{A},\tilde{\mathbf{s}})-\frac{1}{2}\ln\det\left(-\nabla\nabla\left(\ln\left(P(\tilde{\mathbf{s}})P(\mathbf{x}\mid\mathbf{A},\tilde{\mathbf{s}})\right)\right)\right)\right)\\ &= \arg\max_{\mathbf{A}}\left(\ln P(\mathbf{A})+\ln P(\tilde{\mathbf{s}})+\frac{1}{2}\ln\det\left(\nabla\nabla\left(\ln P(\tilde{\mathbf{s}})\right)\right)+\ln P(\mathbf{x}\mid\mathbf{A},\tilde{\mathbf{s}})+\frac{1}{2}\ln\det\left(\nabla\nabla\left(\ln P(\mathbf{x}\mid\mathbf{A},\tilde{\mathbf{s}})\right)\right)\right)\\ &= \arg\max_{\mathbf{A}}\left(\ln P(\mathbf{A})+\xi\left(\ln P(\tilde{\mathbf{s}})\right)+\xi\left(\ln P(\mathbf{x}\mid\mathbf{A},\tilde{\mathbf{s}})\right)\right)\end{aligned} \tag{22}$$

where $\xi(u)=u+\frac{1}{2}\ln\det\left(\nabla\nabla u\right)$. Note that (22) consists of three terms where each term represents a distribution related to either the mixing matrix, source signals or noise. In the following sub-sections, we will provide the detailed derivations for the gradient of each term consecutively.

4.2.1 Derivation of $\frac{\partial \ln P(\mathbf{A})}{\partial \mathbf{A}}$

Assuming that the prior probability distribution for $\mathbf{A}$ is given by the GGD model, then we obtain

$$\begin{aligned}\frac{\partial \ln P(\mathbf{A})}{\partial \mathbf{A}} &\propto -\frac{\partial}{\partial \mathbf{A}}\left(\sum_{n=1}^{N}\sum_{m=1}^{M}\left(\lambda_{\mathbf{A}}\left|a_{mn}\right|\right)^{p_{\mathbf{A}}}\right)\\ &= -\lambda_{\mathbf{A}p}\left|\mathbf{A}\right|^{(p_{\mathbf{A}}-2)}\circ\mathbf{A}\end{aligned} \tag{23}$$

where $\lambda_{\mathbf{A}p}=p_{\mathbf{A}}\lambda_{\mathbf{A}}^{p_{\mathbf{A}}}$, $\left|\mathbf{A}\right|^{(p_{\mathbf{A}}-2)}=\begin{bmatrix}\left|a_{11}\right|^{p_{\mathbf{A}}-2} & \cdots & \left|a_{1N}\right|^{p_{\mathbf{A}}-2}\\ \vdots & \ddots & \vdots\\ \left|a_{M1}\right|^{p_{\mathbf{A}}-2} & \cdots & \left|a_{MN}\right|^{p_{\mathbf{A}}-2}\end{bmatrix}$ and '$\circ$' represents the element-by-element Hadamard product. Specifically, if the mixing matrix has a Gaussian distribution which implies $p_{\mathbf{A}}=2$, then (23) reduces to $\frac{\partial \ln P(\mathbf{A})}{\partial \mathbf{A}}=-2\lambda_{\mathbf{A}}\mathbf{A}$.

4.2.2 Derivation of $\frac{\partial \xi\left(\ln P(\tilde{\mathbf{s}})\right)}{\partial \mathbf{A}}$

The second term of (22) is $\ln P(\tilde{\mathbf{s}})$ which influences the most probable value of $\tilde{\mathbf{s}}$ by updating $\mathbf{A}$. From (22), we have

$$\xi\left(\ln P(\tilde{\mathbf{s}})\right) = \ln P(\tilde{\mathbf{s}}) + \frac{1}{2}\ln\det\left(\nabla\nabla\left(\ln P(\tilde{\mathbf{s}})\right)\right) \tag{24}$$

Through our derivation and analysis, we find that $\xi\left(P(\tilde{\mathbf{s}})\right)$ does not involve any nonlinear process. Due to this characteristic, the derivation of $\frac{\partial\xi\left(\ln P(\tilde{\mathbf{s}})\right)}{\partial\mathbf{A}}$ is very similar to its linear counterpart in [28, 29]. Instead of providing the derivation formally, we simply state the final learning rule of $\frac{\partial\xi\left(\ln P(\tilde{\mathbf{s}})\right)}{\partial\mathbf{A}}$ in this paper as follows:

$$\frac{\partial\xi\left(P(\tilde{\mathbf{s}})\right)}{\partial\mathbf{A}} \approx -\mathbf{V}^T\frac{\partial\ln P(\tilde{\mathbf{s}})}{\partial\tilde{\mathbf{s}}}\tilde{\mathbf{s}}^T \tag{25}$$

where $\mathbf{V}$ is defined to satisfy $\mathbf{A}^T\mathbf{V}^T = \mathbf{I}$ and is computed by $\mathbf{V} = \mathbf{A}^T\left(\mathbf{A}\mathbf{A}^T + \delta\mathbf{I}\right)^{-1}$ in which $\delta << 1$ can be regarded as a small perturbation according to the matrix computation.

4.2.3 Derivation of $\frac{\partial\xi\left(\ln P(\mathbf{x}\,|\,\mathbf{A},\tilde{\mathbf{s}})\right)}{\partial\mathbf{A}}$

Starting from this sub-section, we move onto one of the key points of this paper in that $\frac{\partial\xi\left(\ln P(\mathbf{x}\,|\,\mathbf{A},\tilde{\mathbf{s}})\right)}{\partial\mathbf{A}}$ introduces the nonlinear term in the recovery process to account for the existence of the nonlinearity in the mixing process. From (22), we have

$$\xi\left(\ln P(\mathbf{x}\,|\,\mathbf{A},\tilde{\mathbf{s}})\right) = \ln P(\mathbf{x}\,|\,\mathbf{A},\tilde{\mathbf{s}}) + \frac{1}{2}\ln\det\left(\nabla\nabla\left(\ln P(\mathbf{x}\,|\,\mathbf{A},\tilde{\mathbf{s}})\right)\right) \tag{26}$$

The derivatives of the first term of (26) is given by

$$\begin{aligned}\frac{\partial \ln P(\mathbf{x}\,|\,\mathbf{A},\tilde{\mathbf{s}})}{\partial \mathbf{A}} &= \frac{\partial}{\partial a_{ij}}\sum_{m=1}^{M}\left\| x_m - f_m(\sum_{n=1}^{N} a_{mn}\tilde{s}_n)\right\|^2 \\ &= -2\left(x_i - f_i(\sum_{n=1}^{N} a_{in}\tilde{s}_n)\right) f_i'(\sum_{n=1}^{N} a_{in}\tilde{s}_n)\tilde{s}_j - 2\left(x_i - f_i(\sum_{n=1}^{N} a_{in}\tilde{s}_n)\right) f_i'(\sum_{n=1}^{N} a_{in}\tilde{s}_n) a_{ij}\frac{\partial \tilde{s}_n}{\partial a_{ij}} \\ &= \phi_i \tilde{s}_j - \phi_i a_{ij} a_{ij}^{-1}\tilde{s}_j \\ &= \phi_i \tilde{s}_j - \phi_i \tilde{s}_j \\ &= 0\end{aligned} \tag{27}$$

where $\phi_i = -2\left(x_i - f_i(\sum_{n=1}^{N} a_{in}\tilde{s}_n)\right) f_i'(\sum_{n=1}^{N} a_{in}\tilde{s}_n)$.

As far as the second term is concerned, for simplicity, we define

$$H(\tilde{\mathbf{s}}) = \nabla\nabla\left(\ln P(\mathbf{x}\,|\,\mathbf{A},\tilde{\mathbf{s}})\right) \tag{28}$$

Based on the chain rule,

$$\frac{\partial \ln\det H(\tilde{\mathbf{s}})}{\partial a_{ij}} = \frac{1}{\det H(\tilde{\mathbf{s}})}\sum_{l=1}^{N}\sum_{k=1}^{N}\frac{\partial \det H(\tilde{\mathbf{s}})}{\partial h_{kl}(\tilde{\mathbf{s}})}\frac{\partial h_{kl}(\tilde{\mathbf{s}})}{\partial a_{ij}} \tag{29}$$

The middle term on the right hand side of (29) admits the following identity:

$$\frac{\partial \det H(\tilde{\mathbf{s}})}{\partial h_{kl}(\tilde{\mathbf{s}})} = (\det H(\tilde{\mathbf{s}}))h_{lk}^{-1}(\tilde{\mathbf{s}}) \tag{30}$$

By substituting (30) into (29), this leads to

$$\frac{\partial \ln\det H(\tilde{\mathbf{s}})}{\partial \mathbf{A}} = \begin{bmatrix} \sum_{l=1}^{N}\sum_{k=1}^{N} h_{lk}^{-1}(\tilde{\mathbf{s}})\frac{\partial h_{kl}(\tilde{\mathbf{s}})}{\partial a_{11}} & \cdots & \sum_{l=1}^{N}\sum_{k=1}^{N} h_{lk}^{-1}(\tilde{\mathbf{s}})\frac{\partial h_{kl}(\tilde{\mathbf{s}})}{\partial a_{1N}} \\ \vdots & \ddots & \vdots \\ \sum_{l=1}^{N}\sum_{k=1}^{N} h_{lk}^{-1}(\tilde{\mathbf{s}})\frac{\partial h_{kl}(\tilde{\mathbf{s}})}{\partial a_{M1}} & \cdots & \sum_{l=1}^{N}\sum_{k=1}^{N} h_{lk}^{-1}(\tilde{\mathbf{s}})\frac{\partial h_{kl}(\tilde{\mathbf{s}})}{\partial a_{MN}} \end{bmatrix} \tag{31}$$

Hence, from (28) we can compute

$$
\begin{aligned}
H(\tilde{\mathbf{s}}) &= \nabla\nabla \ln P(\mathbf{x} \mid \mathbf{A}, \tilde{\mathbf{s}}) \\
&= 2\lambda_n^2 \mathbf{A}^T \left(diag\left(\left\| \mathbf{x} - F(\mathbf{A}\tilde{\mathbf{s}}) \right\| \right) diag\left(F''(\mathbf{A}\tilde{\mathbf{s}}) \right) - diag\left(F'(\mathbf{A}\tilde{\mathbf{s}}) \right)^2 \right) \mathbf{A} \\
&= 2\lambda_n^2 \mathbf{A}^T diag\left(\eta \right) \mathbf{A} \\
&= 2\lambda_n^2 \begin{bmatrix} a_{11}^2\eta_1 + a_{21}^2\eta_2 + \ldots + a_{M1}^2\eta_M & \cdots & a_{11}a_{1N}\eta_1 + a_{21}a_{2N}\eta_2 + \ldots + a_{M1}a_{MN}\eta_M \\ \vdots & \ddots & \vdots \\ a_{11}a_{1N}\eta_1 + a_{21}a_{2N}\eta_2 + \ldots + a_{M1}a_{MN}\eta_M & \cdots & a_{1N}^2\eta_1 + a_{2N}^2\eta_2 + \ldots + a_{MN}^2\eta_M \end{bmatrix}
\end{aligned} \tag{32}
$$

Each component of (32) can be expressed as

$$
h_{kl}(\tilde{\mathbf{s}}) = 2\lambda_n^2 \sum_{i=1}^{M} a_{ik} a_{il} \eta_i \tag{33}
$$

where

$$
\eta_i = \left(x_i - f_i(\sum_{n=1}^{N} a_{in}\tilde{s}_n) \right) f_i''(\sum_{n=1}^{N} a_{in}\tilde{s}_n) - f_i'(\sum_{n=1}^{N} a_{in}\tilde{s}_n)^2 \tag{34}
$$

where $f_i''(.)$ is the second order derivative of the function $f_i(.)$ while $f_i'(.)$ is previously defined in (17).

The final term of (29) can be expressed as:

$$
\frac{\partial h_{kl}(\tilde{\mathbf{s}})}{\partial a_{ij}} = 2\lambda_n^2 \left(\psi(i,j,k,l)\eta_i + a_{ik}a_{il}\eta_i'\tilde{s}_j \right) \tag{35}
$$

where $\psi(i,j,k,l) = \begin{cases} 2\tilde{a}_{ij} & j = k = l \\ \tilde{a}_{ik} & j = l \neq k \\ \tilde{a}_{il} & j = k \neq l \\ 0 & j \neq l \neq k \end{cases}$

and

$$
\eta_i' = \left(x_i - f_i(\sum_{n=1}^{N} a_{in}\tilde{s}_n) \right) f_i'''(\sum_{n=1}^{N} a_{in}\tilde{s}_n) - 3f_i'(\sum_{n=1}^{N} a_{in}\tilde{s}_n) f_i''(\sum_{n=1}^{N} a_{in}\tilde{s}_n) \tag{36}
$$

The term $f_i'''(.)$ represents the third order derivative of $f_i(.)$.

Returning to (30), we have

$$\sum_{l=1}^{N}\sum_{k=1}^{N} h_{lk}^{-1}(\tilde{\mathbf{s}})\frac{\partial h_{kl}(\tilde{\mathbf{s}})}{\partial a_{ij}} = 2\lambda_n^2\left(\sum_{l=1}^{N}\sum_{k=1}^{N} h_{lk}^{-1}(\tilde{\mathbf{s}})\psi(i,j,k,l)\eta_i + \sum_{l=1}^{N}\sum_{k=1}^{N} h_{lk}^{-1}(\tilde{\mathbf{s}})a_{ik}a_{il}\eta_i'\tilde{s}_j\right) \tag{37}$$

Note that the matrix form of the first term in (37) can be described as

$$\begin{aligned}\sum_{l=1}^{N}\sum_{k=1}^{N} h_{lk}^{-1}(\tilde{\mathbf{s}})\boldsymbol{\psi}(k,l)diag(\eta) &= \begin{bmatrix} \left(a_{11}h_{11}^{-1}+a_{12}h_{12}^{-1}+\ldots+a_{1N}h_{1N}^{-1}\right)\eta_1 & \cdots & \left(a_{11}h_{N1}^{-1}+a_{12}h_{N2}^{-1}+\ldots+a_{1N}h_{NN}^{-1}\right)\eta_1 \\ \vdots & \ddots & \vdots \\ \left(a_{M1}h_{11}^{-1}+a_{M2}h_{12}^{-1}+\ldots+a_{MN}h_{1N}^{-1}\right)\eta_M & \cdots & \left(a_{M1}h_{N1}^{-1}+a_{M2}h_{N2}^{-1}+\ldots+a_{MN}h_{NN}^{-1}\right)\eta_M \end{bmatrix} \\ &= 2diag(\eta)\mathbf{A}H^{-1}(\tilde{\mathbf{s}}) \\ &= 2diag(\eta)\mathbf{A}\left(-2\lambda_n^2\mathbf{A}^T diag(\eta)\mathbf{A} - \nabla\nabla\log P(\tilde{\mathbf{s}})\right)^{-1}\end{aligned} \tag{38}$$

where $\boldsymbol{\psi}(k,l) = \begin{bmatrix} \psi(1,1,k,l) & \cdots & \psi(1,N,k,l) \\ \vdots & \ddots & \vdots \\ \psi(M,1,k,l) & \cdots & \psi(M,N,k,l) \end{bmatrix}$. As defined in (10), λ_n is the inverse of generalised variance. If the noise level is small enough, then $\nabla\nabla\log P(\tilde{\mathbf{s}})$ is negligible compared with $2\lambda_n^2\mathbf{A}^T diag(\eta)\mathbf{A}$. Thus, $\nabla\nabla\log P(\tilde{\mathbf{s}})$ can be considered as a small perturbation of the dominant component $2\lambda_n^2\mathbf{A}^T diag(\eta)\mathbf{A}$ which further substantiates the inverse matrix to be always stable. Hence

$$\mathbf{A}\left(-2\lambda_n^2\mathbf{A}^T diag(\eta)\mathbf{A} - \nabla\nabla\log P(\tilde{\mathbf{s}})\right)^{-1}\mathbf{A}^T \approx -\frac{1}{2}\lambda_n^{-2}diag^{-1}(\eta) \tag{39}$$

Note that $\mathbf{A}^T\mathbf{V}^T = \mathbf{I}$, and applying this identity to (38), this yields

$$\begin{aligned}\sum_{l=1}^{N}\sum_{k=1}^{N} h_{lk}^{-1}(\tilde{\mathbf{s}})\boldsymbol{\psi}(k,l)diag(\eta)\mathbf{A}^T\mathbf{V}^T &= 2diag(\eta)\left[\mathbf{A}\left(-2\lambda_n^2\mathbf{A}^T diag(\eta)\mathbf{A} - \nabla\nabla\log P(\tilde{\mathbf{s}})\right)^{-1}\mathbf{A}^T\right]\mathbf{V}^T \\ &\approx -\lambda_n^{-2}\mathbf{V}^T\end{aligned} \tag{40}$$

The matrix form of the second term of (37) is shown as

$$
\begin{aligned}
&\begin{bmatrix}
\left(\sum_{n=1}^{N} a_{1n}^2 h_{nn}^{-1} + 2\sum_{\substack{p=1,q=1\\p\neq q}}^{N} a_{1p}a_{1q}h_{pq}^{-1}\right)\eta_1'\tilde{s}_1 & \cdots & \left(\sum_{n=1}^{N} a_{1n}^2 h_{nn}^{-1} + 2\sum_{\substack{p,q=1\\p\neq q}}^{N} a_{1p}a_{1q}h_{pq}^{-1}\right)\eta_1'\tilde{s}_N \\
\vdots & \ddots & \vdots \\
\left(\sum_{n=1}^{N} a_{Mn}^2 h_{nn}^{-1} + 2\sum_{\substack{p=1,q=1\\p\neq q}}^{N} a_{Mp}a_{Mq}\right)\eta_M'\tilde{s}_1 & \cdots & \left(\sum_{n=1}^{N} a_{Mn}^2 h_{nn}^{-1} + 2\sum_{\substack{p=1,q=1\\p\neq q}}^{N} a_{Mp}a_{Mq}h_{pq}^{-1}\right)\eta_M'\tilde{s}_N
\end{bmatrix} \\
&= diag(\eta')\left\langle \mathbf{A}H^{-1}(\tilde{\mathbf{s}})\mathbf{A}^{\mathbf{T}}\right\rangle\tilde{\mathbf{s}}^T \\
&= diag(\eta')\left\langle \mathbf{A}\left(-2\lambda_n^2\mathbf{A}^T diag(\eta)\mathbf{A} - \nabla\nabla\log P(\tilde{\mathbf{s}})\right)^{-1}\mathbf{A}^{\mathbf{T}}\right\rangle\tilde{\mathbf{s}}^T \\
&= -\frac{1}{2}\lambda_n^{-2} diag(\eta')\left[\eta_1^{-1}\ldots\eta_M^{-1}\right]^T\tilde{\mathbf{s}}^T \\
&= -\frac{1}{2}\lambda_n^{-2}\left[\eta_1'\eta_1^{-1} \quad \cdots \quad \eta_M'\eta_M^{-1}\right]^T\tilde{\mathbf{s}}^T
\end{aligned}
\tag{41}
$$

where $\langle . \rangle$ defines the vector extraction of the diagonal component of a matrix and $diag(.)$ represents the construction of a diagonal matrix from a vector.

Substituting (40) and (41) into (37), we obtain the following:

$$
\begin{aligned}
\frac{\partial \ln\det H(\tilde{\mathbf{s}})}{\partial \mathbf{A}} &= 2\lambda_n^2\left(-\lambda_n^{-2}\mathbf{V}^T - \frac{1}{2}\lambda_n^{-2}\left[\eta_1'\eta_1^{-1} \quad \cdots \quad \eta_M'\eta_M^{-1}\right]^T\tilde{\mathbf{s}}^T\right) \\
&= -2\mathbf{V}^T - \left[\eta_1'\eta_1^{-1} \quad \cdots \quad \eta_M'\eta_M^{-1}\right]^T\tilde{\mathbf{s}}^T
\end{aligned}
\tag{42}
$$

where η and η' are defined in(34) and (36), respectively.

4.2.4 Simplification in the update of $\mathbf{A}$

From the three sub-sections presented above, we conclude that the final expression of the learning rule for the mixing matrix $\mathbf{A}$ is given by

$$
\begin{aligned}
\Delta\mathbf{A} &= -\lambda_{\mathbf{A}p}\left|\mathbf{A}\right|^{p_{\mathbf{A}}-2}\circ\mathbf{A} - \mathbf{V}^T\frac{\partial \ln P(\tilde{\mathbf{s}})}{\partial\tilde{\mathbf{s}}}\tilde{\mathbf{s}}^T + \frac{1}{2}\left(-2\mathbf{V}^T - \left[\eta_1'\eta_1^{-1} \quad \cdots \quad \eta_M'\eta_M^{-1}\right]^T\tilde{\mathbf{s}}^T\right) \\
&= -\lambda_{\mathbf{A}p}\left|\mathbf{A}\right|^{p_{\mathbf{A}}-2}\circ\mathbf{A} - \mathbf{V}^T\left(\frac{\partial \ln P(\tilde{\mathbf{s}})}{\partial\tilde{\mathbf{s}}}\tilde{\mathbf{s}}^T + \mathbf{I}\right) - \frac{1}{2}\left[\eta_1'\eta_1^{-1} \quad \cdots \quad \eta_M'\eta_M^{-1}\right]^T\tilde{\mathbf{s}}^T \\
&= \mathbf{L} + \mathbf{N}
\end{aligned}
\tag{43}
$$

where we define $\mathrm{L} = -\lambda_{\mathbf{A}p} \left|\mathbf{A}\right|^{p_{\mathrm{A}}-2} \circ \mathbf{A} - \mathbf{V}^T \left(\frac{\partial \ln P(\tilde{\mathbf{s}})}{\partial \tilde{\mathbf{s}}} \tilde{\mathbf{s}}^T + \mathbf{I} \right)$ as the linear constituent and $\mathbf{N} = -\frac{1}{2}\left[\eta_1' \eta_1^{-1} \quad \cdots \quad \eta_M' \eta_M^{-1} \right]^T \tilde{\mathbf{s}}^T$ as the nonlinear constituent. The terms η and η' are defined in (34) and (36), and their existences are the direct outcome of the nonlinear distortion functions embedded in the mixture. Hence, to facilitate the update of $\mathbf{A}$ in (43) this requires some form of assumptions on the nonlinear distortion function. For a blind system, the mixing process is unknown and therefore, if crude assumptions on nonlinear distortion function were to be imposed at this stage, significant mismatch between the hypothetical and the true nonlinear distortion functions will be resulted. As demonstrated in Section 5, this mismatch will lead to performance degradation. In this paper, we propose to develop a self-adaptive algorithm based on the polynomial so that the constructed nonlinear distortion function is as close as possible to the true function $\{f_m\}_{m=1}^M$ in terms of the mean square error criterion.

4.3 A systematic update rule to minimise nonlinear mismatch

A self-adaptive algorithm is essential in the case where the nonlinear distortion function cannot be accessed directly. The Weierstrass Approximation Theorem [30] proves that for every continuous function $f(.)$, there always exists a polynomial expression which can uniformly approximate $f(.)$ in the form of

$$\delta(u, b_k\big|_0^K) = b_0 + b_1 u + \cdots + b_K u^K = \sum_{k=0}^{K} b_k u^k \tag{44}$$

where K represents the order and b_k is the coefficient of the polynomial.

As long as the nonlinear functions $\{f_m\}_{m=1}^M$ are wrongly specified, the estimated $\mathbf{s}$ and $\mathbf{A}$ will still yield a mismatch during the recovery process. Since the function f_m is a one-to-one mapping, there is a clear cut of role as to how the estimation of $\mathbf{A}$, $\mathbf{s}$ and $\{f_m\}_{m=1}^M$ can be managed. Thus, this allows us to formulate a least square error criterion in terms of the polynomial coefficient to minimise the mismatch between the true observed signal and the estimated observed signal as follows:

$$\left\{\hat{b}_{m,k}\right\}_{k=0}^{K} = \underset{\{b_{m,k}\}_{k=0}^{K}}{\arg\min} \frac{1}{T}\sum_{t=1}^{T}\left(x_m(t) - \sum_{k=0}^{K} b_{m,k}(\mathbf{a}_m\tilde{\mathbf{s}}(t))^k \right)^2 \qquad \forall m = 1,\ldots,M \tag{45}$$

where $\mathbf{a}_m$ is the m^{th} row of the estimated mixing matrix $\mathbf{A}$. Let $\mathbf{b}_m = \left[b_{m,0}\ b_{m,1}\ \cdots\ b_{m,K}\right]^T$ and $\mathbf{z}_m(t) = \left[1\ (\mathbf{a}_m\tilde{\mathbf{s}}(t))\ \cdots\ (\mathbf{a}_m\tilde{\mathbf{s}}(t))^K\right]^T$, then (45) can be written as

$$\begin{aligned}\hat{\mathbf{b}}_m &= \underset{\mathbf{b}_m}{\arg\min} \frac{1}{T}\sum_{t=1}^{T}\left(x_m(t) - \sum_{k=0}^{K} \hat{\mathbf{b}}_m^T\mathbf{z}_m(t) \right)^2 \\ &= \left[\sum_{t=1}^{T}\mathbf{z}_m(t)\mathbf{z}_m^T(t)\right]^{-1}\sum_{t=1}^{T} x_m(t)\mathbf{z}_m(t)\end{aligned} \tag{46}$$

In terms of on-line update, (46) can be recast as follows:

$$\mathbf{b}_m(t+1) = \mathbf{b}_m(t) + 2\mu_b\left(x_m(t) - \mathbf{b}_m^T\mathbf{z}_m(t)\right)\mathbf{z}_m(t) \tag{47}$$

where μ_b is the step size. Once the coefficients of polynomial converge, the new estimated nonlinear function is substituted into (18) and (43) to obtain refined estimates of $\tilde{\mathbf{s}}$ and $\tilde{\mathbf{A}}$.

4.4 Hyper-parameters estimation

In the previous sections, we have introduced the main algorithm which estimates $\mathbf{s}$, $\mathbf{A}$ and $f(.)$ in (15), (22) and (45), respectively. Within these parameters, there are the hyper-parameters which need to be estimated during the update process. These hyper-parameters include λ_s, p_s, $\lambda_{\mathbf{A}}$ and $p_{\mathbf{A}}$. Under high signal-to-noise ratio (SNR), λ_n will contribute very little and thus, it is negligible and can be neglected. The hyper-parameters λ_s, p_s, $\lambda_{\mathbf{A}}$, and $p_{\mathbf{A}}$ depict the degree of similarity between the estimated prior probability distribution and the real probability distribution. It is preferred that the initial estimation is as close as possible to the true probability distribution. However, this is often not the case and hence, a self-adaptive algorithm for each parameter is more suitable to minimise the mismatch of the probability distribution model. The correct Bayesian analysis requires us to integrate out the hyper-parameters from the marginal probability.

This technique is tedious and computationally very demanding. On the other hand, rather than using a full Bayesian analysis we adopt the maximum likelihood approach to estimate these hyper-parameters. The update method for these parameters is very similar since they are all based on the GGD model. Because of this similarity, we only give a general form which use λ and p to symbolise all λ related parameters and p related parameters, respectively.

From (10), we know that the expectation of the log-likelihood function of GGD model is given by

$$E\left[\ln P\left(\mathbf{u}\middle|\lambda,p\right)\right]=\ln\lambda+\ln p-\ln 2-\ln\Gamma\left(1/p\right)-\lambda^{p}E\left[\left|\mathbf{u}\right|^{p}\right] \tag{48}$$

where $E[\cdot]$ symbolises the expectation. Equation (48) is utilised as the cost function and the derivative with respect to λ gives the gradient of λ as

$$\frac{\partial\ln P\left(\mathbf{u}\middle|\lambda,p\right)}{\partial\lambda}=\frac{1}{\lambda}-pE\left[\left|\mathbf{u}\right|^{p}\right]\lambda^{(p-1)} \tag{49}$$

Equating (49) to zero and solving for λ will give the estimate of the inverse variance. Similarly, the derivative of $\ln P\left(\mathbf{u}\middle|\lambda,p\right)$ with respect to p gives the gradient of p as

$$\frac{\partial\ln P\left(\mathbf{u}\middle|\lambda,p\right)}{\partial p}=\frac{1}{p}-\frac{\Gamma'\left(1/p\right)}{\Gamma\left(1/p\right)}-\lambda^{p}\ln\lambda E\left[\left|\mathbf{u}\right|^{p}\right]\ln\left(E\left[\left|\mathbf{u}\right|\right]\right) \tag{50}$$

The parameter p can be estimated either using gradient or Newton's method. Equations (49) and (50) are the general derivation of λ and p. If individual derivative such as λ_s and p_s, are required, then this can simply be done by substituting them into (49) and (50) directly and this two equations becomes dedicated for λ_s and p_s.

Finally, we summarise the whole framework of our algorithm as follows:

Step 1: Initialise $\mathbf{s}$, $\mathbf{A}$, $\{f_m\}_{m=1}^{M}$, λ_s, p_s, $\lambda_{\mathbf{A}}$ and $p_{\mathbf{A}}$.

Step 2: Substitute $\mathbf{A}$, $\{f_m\}_{m=1}^{M}$ into (18) and update $\mathbf{s}$ until it converges to a fixed value $\tilde{\mathbf{s}}$.

Step 3: Apply the estimated $\tilde{\mathbf{s}}$ from Step 2 into (43) and iterate until $\mathbf{A}$ converges to a fixed value $\tilde{\mathbf{A}}$.

Step 4: Repeat Step 2 and 3 until both $\tilde{\mathbf{s}}$ and $\tilde{\mathbf{A}}$ remain relatively stable i.e. $\left\|\tilde{\mathbf{s}}(n+1)-\tilde{\mathbf{s}}(n)\right\|^2 < \varsigma$ and $Trace\left[\mathbf{E}(n)\mathbf{E}^T(n)\right] < \varsigma$ where $\mathbf{E}(n) = \tilde{\mathbf{A}}(n+1) - \tilde{\mathbf{A}}(n)$ and ς is a small constant (set to 0.01).

Step 5: Apply $\tilde{\mathbf{s}}$ and $\tilde{\mathbf{A}}$ from Step 4 into (47), (49) and (50) to obtain a new optimised nonlinear function $\{f_m\}_{m=1}^{M}$ and the set of hyper-parameters λ and p.

Step 6: Return to step 4 until all parameters converge.

We note that Step 1 to Step 4 are the essential steps to recover the sources $\mathbf{s}$ and the mixing matrix $\mathbf{A}$. They compose the main body of the algorithm. Step 5 and Step 6 concern about updating the nonlinear function $\{f_m\}_{m=1}^{M}$ and the hyper-parameters which aim to minimise both the mismatches of nonlinearity and the probability distribution. The latter two steps improve the quality of recovery but at the cost of higher computational complexity. We emphasise that part of the computational complexity of the algorithm depends on the type of applications whether Steps 5 and 6 are required to be executed.

5 SIMULATION RESULTS

In this section, we design several experimental simulations under different conditions to examine our proposed iterative algorithm. Before presenting the results, we introduce a performance index to assess the acquired results. The Mean Square Error (MSE) criterion is a reliable performance index measuring the similarity between two signals. However, it is sensitive to the variability of both scale and phase of the signals. In the context of BSS, the estimated signals can be subject to scale and phase reversal ambiguities and therefore, the MSE criterion is not suitable for direct comparison between the original and the estimated sources. As an alternative, we propose the following performance index:

$$\mathrm{P} = 2\left(1 - \frac{1}{N}\sum_{i=1}^{N} |\rho_i|\right) \tag{51}$$

where

$$\rho_i = \frac{E\left[(s_i - E[s_i])^* (\tilde{s}_i - E[\tilde{s}_i])\right]}{\sqrt{E\left[|s_i - E[s_i]|^2\right] E\left[|\tilde{s}_i - E[\tilde{s}_i]|^2\right]}} \tag{52}$$

where ρ_i is the normalized cross-correlation. In above, '$*$' and '$|\cdot|$' denote the complex conjugate and absolute operation, respectively. It can be shown that the proposed performance index P is essentially a variant of the MSE criterion that implicitly takes into account the scale and phase reversal ambiguities.

5.1 Performance improvement compared with linear algorithm

In this experiment, three source signals as shown in Fig.3 (top) are generated by the GGD model which has a Laplacian distribution and they are mixed into two mixtures as depicted in Fig.3 (middle). The mixing matrix is randomly generated which has a Gaussian distribution. The actual post-nonlinear process is set to tanh(.) and the estimated nonlinear process is assumed to be identical to the true nonlinear distortion function. Hence, there is no mismatch in the nonlinear process. For all experiments to follow in this paper, the initial value of $\mathbf{A}$ is selected randomly and the initial value of $\tilde{\mathbf{s}}$ is set to

$$\tilde{\mathbf{s}} = \mathbf{A}^{+}\mathbf{x} \tag{53}$$

where $\mathbf{A}^{+}$ is pseudoinverse of mixing matrix $\mathbf{A}$. The noise is gaussianly distributed and is used to perturb the sensors. The initial hyper-parameters for the estimated mixing matrix are $p_{\mathbf{A}} = 2$, $\lambda_{\mathbf{A}} = 1$ while for the estimated source signals, they are computed directly from (53). All hyper-parameters are updated in (49) and (50).

Fig.3 (bottom) shows the recovered source signals under SNR=20dB. Comparing Fig.3 (top) with Fig.3 (bottom), there is a closed resemblance between the recovered source signals and the original sources. To

further examine the results, we utilise the performance index given in (51). Fig.4. (top) depicts the performance index between the sources and the estimated sources. It is seen that the performance index converges to a small value after 400 iterations. To investigate the effect of noise on our algorithm, we simulated different levels of noise magnitude. Fig.4 (bottom) shows the SNR as a function of performance index. The figure clearly shows that reasonably good performance index is obtained when SNR is above 10dB.

In order to demonstrate that our algorithm has improved performance over the linear recovery algorithm, we applied the well-known FOCal Underdetermined System Solver (FOCUSS) algorithm [31] to the same mixing condition. Fig. 4 (dotted line) shows the performance index of the FOCUSS algorithm where it is seen that performance improvement of 130% is obtained over FOCUSS under SNR=20dB. In the case of different levels of noise, significant performance gain over FOCUSS has been obtained in all cases of SNR>5dB. To compare the computational time taken between the FOCUSS and our proposed algorithm, only $\mathbf{s}$ and $\{f_m\}_{m=1}^{M}$ will be estimated while $\mathbf{A}$ is kept fixed. For 3×10^4 samples, the time taken to run the FOCUSS algorithm and the proposed algorithm is 10.6s and 15.2s, respectively using CPU (Pentium IV 3GHz) with memory 1GB. Hence, the proposed algorithm is only 1.43 times more complex compared to the FOCUSS algorithm.

5.2. Nonlinear mismatch problem

In Section 5.1, the nonlinear function used in recovery process is set to $\tanh(.)$, which is identical to the true nonlinear distortion function. However, perfect knowledge of the true nonlinear distortion function is not always possible since the mixing process is unknown to the receiving end. In this experiment, we assume that we have no knowledge of the degree of nonlinearity in the mixture. Any mismatch of estimated nonlinear process will lead to performance degradation. We design the following experiment to investigate the effects of nonlinear mismatch and to evaluate the performance of the proposed self-adaptive polynomial in approximating the true nonlinear distortion function.

The true nonlinear distortion function is set to $\mathbf{x}+0.8\mathbf{x}^3$, and the hypothetical nonlinear functions used in the algorithm are $\mathbf{x}+0.8\mathbf{x}^3$, $\mathbf{x}^5$, and $\tanh(.)$, respectively. Fig. 5 shows the results obtained in terms of the performance index of matched and mismatched nonlinear functions. The figure reveals that the matched nonlinear function $\mathbf{x}+0.8\mathbf{x}^3$ yields the best performance which is 55% better compared to using $\mathbf{x}^5$ while a further 136% performance improvement is obtained over $\tanh(.)$.

In Section 4.3, we introduce a systematic approach to minimise the nonlinear mismatch by using the self-adaptive polynomial. In this experiment, three different types of nonlinearity, $\mathbf{x}+0.8\mathbf{x}^3$, $\tanh(.)$ and $\mathbf{x}^5$ are applied as the nonlinear distortion function respectively and the polynomials are expected to adaptively approximate the true nonlinear function. The order of the polynomials is set to 15. The polynomials are initialised as a linear function but the coefficients are subsequently updated according to (46). Fig. 6 shows the approximated polynomial function after the update process. It is clear that the estimated nonlinear function converges very close to the true nonlinear distortion function. Although minor nonlinear mismatch is still visible in some cases, this mismatch is considered as negligible when compared with the results if arbitrary selection of the hypothetical nonlinear function were to be used.

5.3 Capability to accommodate source signals with different distributions

In section 5.1, we showed that our algorithm is capable of separating three super-gaussian signals. In this simulation, we consider source signals which are not limited to only super-gaussianly distributed. We generate three source signals as follows: one Laplacian distribution, one Gaussian distribution and one sub-gaussian distribution. For the sub-gaussian signal, we apply a sine wave with frequency equals to 100Hz i.e. $\sin(2\pi\times 100t)$. We note that the sine wave has a bimodal distribution and since the GGD model can only approximate unimodal distribution, there is mismatch in terms of the probability model between the true distribution and the hypothesised distribution. Maintaining other parameter identical to Section 5.1, the following results are obtained.

Fig.7 shows the original source, mixture and recovered sources. Visually, there is a high degree of similarity

between the original sources and the recovered sources. Detailed examination on the results reveals that the proposed algorithm can still maintain high performance as measured by the Performance Index and the Performance Index versus SNR as depicted in Fig.8. However, comparing to Section 5.1, the performance is seen to degrade by 45% under SNR=20dB when dealing with different distributions and this discrepancy is attributed by the mismatch of the GGD model to describe the probability distribution of the sine wave. From the obtained results after convergence, the p_s value of the estimated Gaussian source signal converges close to 2; however, for the estimated Laplacian source signal, the p_s value converges to a value slightly higher than the true value while for the estimated sine wave, the p_s value converges to a value that ranges from 8 to 11. Although the latter corresponds to a sub-gaussian distribution, the true source signal distribution is bimodal and this subsequently leads to waveform mismatch as observed in the estimated sine wave. Thus, this result points out the need to use a more general signal model for modelling multimodal probability distribution if high performance is to be maintained.

5.4 Speech sources

To investigate the effectiveness of our algorithm on speech signals, three speech signals are recorded through microphones which are stored in a computer as three independent source signals. By calculating the kurtosis of speech signals, it is found that the distribution of the speech signals is very similar to the Laplacian distribution. Therefore, all parameters are set identical to those used in Section 5.1 except that the sources are replaced by speech signals and the nonlinear distortion function is $\mathbf{x}+0.8\mathbf{x}^3$. The estimation process follows the steps as proposed in Section 4. Fig. 9 depicts the true speech signals (top), mixed signals (middle), and recovered speech signals (bottom). From the visual perspective, the estimated speech signals closely resemble the original signals. The hyper-parameters of the estimated speech signals have converged close to the true values. Fig.10 presents the Performance Index under SNR=20dB and the Performance Index as a function of SNR. Very similar to the result in Section 5.1, performance improvement of 128% is obtained over FOCUSS algorithm for the speech signals under SNR=20dB.

6 CONCLUSION AND FUTURE WORK

In this paper, a novel algorithm for blind nonlinear underdetermined mixtures is proposed. The algorithm originates from the Bayesian framework and culminates to the MAP-based solution. The proposed algorithm is general and powerful enough to deal with both super-gaussian and sub-gaussian signals. Furthermore, both problems of nonlinearity and probability distribution mismatches have been considered and a systematic solution is proposed altogether. Computer simulations have shown that separation of blind nonlinear underdetermined mixture is feasible and that the MAP-based approach has yielded a considerable level of high performance. It is reported that an average of 130% performance gain has been achieved compared to linear algorithm. However, there still remain a few challenges that need to be considered in future work. The proposed algorithm is currently augmented with high computational complexity since it requires relatively large number of loops during the parameter updates. A way forward is to reduce the complexity by removing the terms in the algorithm that contribute very little in the update after a certain number of iterations. In this case, it is essential that the characteristics and the behaviours of each term in the algorithm are studied in details. Secondly, the proposed algorithm uses the GGD model exclusively to approximate the prior probability distribution of the parameters. The GGD model is limited to unimodal distribution and for parameters that exhibit multimodal distribution, a more efficient approximating model is therefore required. A suitable candidate would be the Gaussian Mixture Model (GMM) which can asymptotically accommodate any continuous distributions. Finally, the issue of model order selection i.e. the determination of the number of source signals has been completely set aside in this paper. A viable approach would be to consider a full Bayesian analysis where the number of source signals is considered as a parameter to be optimised and the marginal probability expression would need to take this into consideration in deriving the posterior update.

7 REFERENCES

1. CARDOSO, J.F.: 'Blind signal separation: Statistical principles', Proceedings of the IEEE, Oct. 1998, 86, (10), pp. 2009-2025.
2. AMARI, S. and CICHOCKI, A.: 'Adaptive blind signal processing - Neural network approaches', Proceedings of the IEEE, Oct. 1998, 86, (10), pp. 2026-2048.
3. AMARI, S., HYVARINEN, A., LEE, S.-Y., LEE, T.W., and SANCHEZ A, V.D.: 'Blind signal separation and independent component analysis', Neurocomputing, 2002, 49, (1-4), pp. 1-5.
4. HYVARINEN, A., KARHUNEN, J., and OJA, E.: 'Independent Component Analysis', (J. Wiley, New York, 2001).
5. KWON, O.W. and LEE, T.W.: 'Phoneme recognition using ICA-based feature extraction and transformation', Signal Processing, 2004, 84, (6), pp. 1005-1019.
6. KANO, M., HASEBE, S., HASHIMOTO, I. and OHNO, H.: 'Evolution of multivariate statistical process control: Application of independent component analysis and external analysis', Computers & Chemical Engineering, 2004, 28, (6-7), pp. 1157-1166.
7. WOO, W.L., and DLAY, S.S.: 'Neural Network Approach to Blind Signal Separation of Mono-nonlinearly Mixed Signals', IEEE Trans. on Circuits and System – Part 1, 2005, 28, (2), pp. 1236-1247.
8. DAVIES, M. and MITIANOUDIS, N.: 'Simple mixture model for sparse overcomplete ICA', IEE Proceedings on Vision, Image & Signal Processing (Special Section on Nonlinear and Non-Gaussian Signal Processing), 2004, 151, (1), pp. 35-43.
9. TAKIGAWA, I., KUDO, M. and TOYAMA,J.: 'Performance Analysis of Minimum l_1-Norm Solutions for Underdetermined Source Separation', IEEE Transactions on Signal Processing, 2004, 52, (3), pp. 582-591
10. WIPF, D.P. and RAO, B,D.: 'Sparse Bayesian Learning for Basis Selection', IEEE Transactions on Signal Processing, 2004, 52, (8), pp. 2153-2164
11. WOO, W.L. and DLAY, S.S.: 'Regularised Nonlinear Blind Signal Separation using Sparsely Connected Network', IEE Proc. on Vision, Image and Signal Processing, 2005, 152, (1), pp. 61-73.
12. GIROLAMI, M.: 'A Variational Method for learning Sparse and Overcomplete representations', Neural

Computation, 2001, 13, pp. 2517-2532

13. ZIBULEVSKY, M., PEARLMUTTER, B.A., BOFILL, P., KISILEV, P.: 'Blind source separation by sparse decomposition', In ROBERT S.J., EVERSON, R.M., ed.: Independent Component Analysis: Principles and Practice. (Cambridge University Press, 2001).
14. OLSHAUSEN, B.A., MILLAMAN, K.J.: 'Learning sparse codes with a mixture-of-Gaussians prior'. In S. A. SOLLA, T.K.L., ed.: Advances in Neural Information Processing Systems. (MIT press 2000)
15. WOO, W.L. and SALI, S.: 'General Multilayer Perceptron Demixer Scheme for Nonlinear Blind Signal Separation', IEE Proceedings on Vision, Image and Signal Processing, Oct. 2002, 149, (5), pp. 253-262.
16. LEE, T.W.: 'Nonlinear Approaches to Independent Component Analysis'. Proceedings of the American Institute of Physics, Feb, 2000, 501, (1), pp. 302-316.
17. TAN, Y., WANG, J., and ZURADA, J.: 'Nonlinear Blind Source Separation using a Radial Basis Function Network', IEEE Transactions on Neural Network, 2001, 12, (1), pp. 124-134.
18. MARTINEZ, D. and BRAY, A,: 'Nonlinear blind source separation using kernels', IEEE Transactions on Neural Networks, 2003, 14, (1), pp. 228-235
19. KHOR, L.C., WOO, W.L. and DLAY, S.S.: 'Nonlinear Blind Signal Separation with Intelligent Controlled Learning', IEE Proc. on Vision, Image and Signal Processing, 2005, 152, (3), pp. 297-306.
20. YANG, H.H., AMARI, S., and CICHOCKI, A.: 'Information-theoretic approach to blind separation of sources in nonlinear mixture', Signal Processing, 1998, 64, pp. 291-300
21. WEI, C., WOO, W.L. and DLAY, S.S.: 'A FOCUSS-Based Algorithm for Nonlinear Overcomplete Independent Component Analysis', WSEAS Transactions on Information Science and Applications, 2004, 6, (1), pp. 1688-1693.
22. MAKEIG, S., JUNG, T., BELL, A.J., GAHREMANI, D., and SEJNOWSKI, T.J.: 'Blind Separation of Event-related Brain Response into Spatial Independent Components', Proceedings of the National Academy of Sciences, 1997, 94, pp. 10979-10984
23. MAKEIG, S., and INLOW, M.: 'Changes in the EEG Spectrum Predict Fluctuations in Error Rate in an Auditory Vigilance Task', Society for Psychophysiology, 1993, 28, S39
24. QUATIERI, T.F., REYNOLDS, D.A., and O'LEARY, G.C.: 'Estimation of handset nonlinearity with application to speaker recognition', IEEE Transactions on Speech and Audio Processing, 2000, 8, (5),

pp. 567-584.

25. FRANK, W., REGER, R., and APPEL, U.U.: 'Loudspeaker nonlinearities - Analysis and compensation', in Proc. Int. Asilomar on Signals, Systems and Computers, Pacific Groove, California, USA, 1992, pp. 756-760.
26. TALEB, A. and JUTTEN, C.: 'Source Separation in Post-Nonlinear Mixtures', IEEE Transactions on Signal Processing, Ort. 1999, 47, (10), pp. 2807-2820.
27. JUTTEN, C., BABAIE-ZADEH, M., and HOSSEINI, S.: 'Three easy ways for separating nonlinear mixtures?' Signal Processing, 2004, 84, (2), pp. 217-229.
28. RAO, B.D., ENGAN, K., COTTER, S.F., PALMER, J. and KREUTZ-DELGADO, K.: 'Subset Selection in Noise Based on Diversity Measure Minimization', IEEE Transactions on Signal Processing, 2004, 51, (3), pp. 760-770
29. LEWICKI, M.S. and SEJNOWSKI, T.J.: 'Learning Overcomplete Representation', Neural Computation, 2000, 12, (2), pp. 337-365
30. CANTRELL, C.D.: 'Modern Mathematical Methods for Physicists and Engineers', (Cambridge University Press, 2000).
31. RAO, B.D. and KREUTZ-DELGADO. K.: 'An Affine Scaling Methodology for Best Basis Selection', IEEE Transactions on Signal Processing', 1999, 47, (1), pp. 187-200

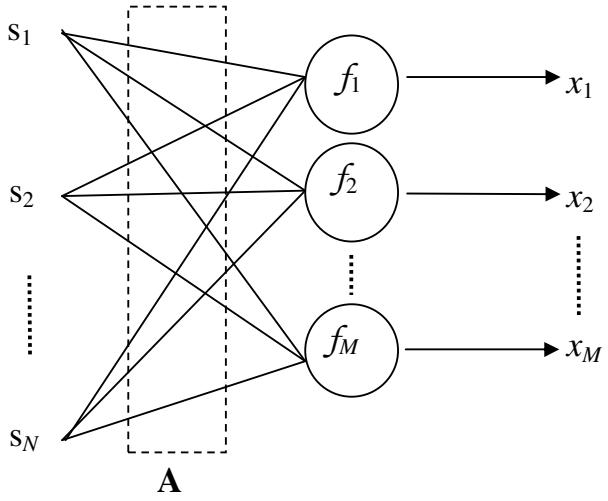
s1
s2
sN
f1
f2
fM
x1
x2
xM
A

Fig.1

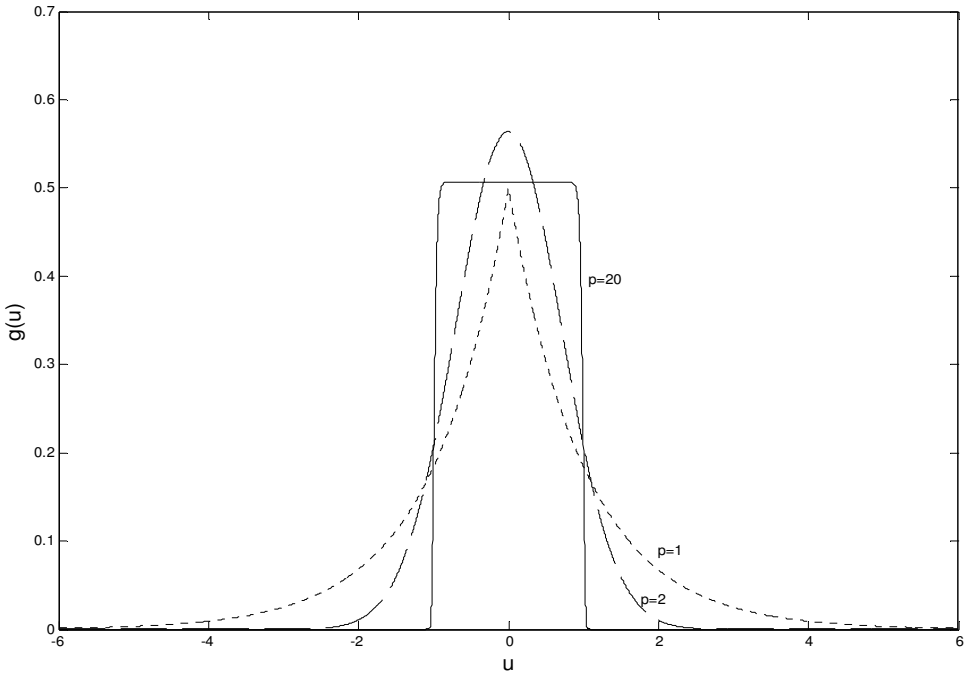
g(u)
u
p=20
p=1
p=2
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-6
-4
-2
0
2
4
6

Fig.2

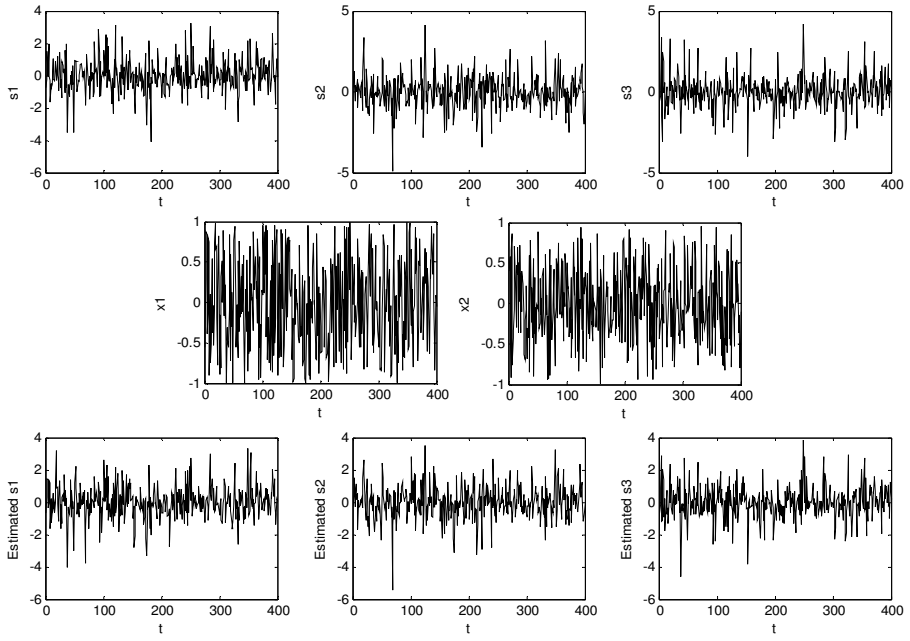
s1
s2
s3
x1
x2
Estimated s1
Estimated s2
Estimated s3
t

Fig.3

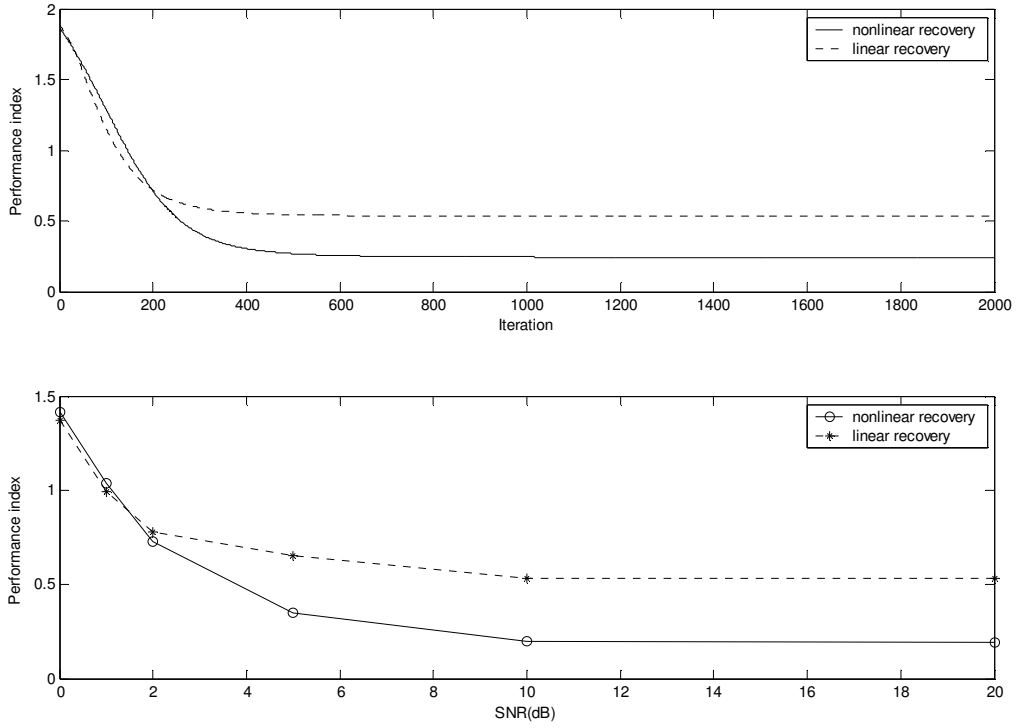
nonlinear recovery
linear recovery
Performance index
Iteration
SNR(dB)

Fig.4

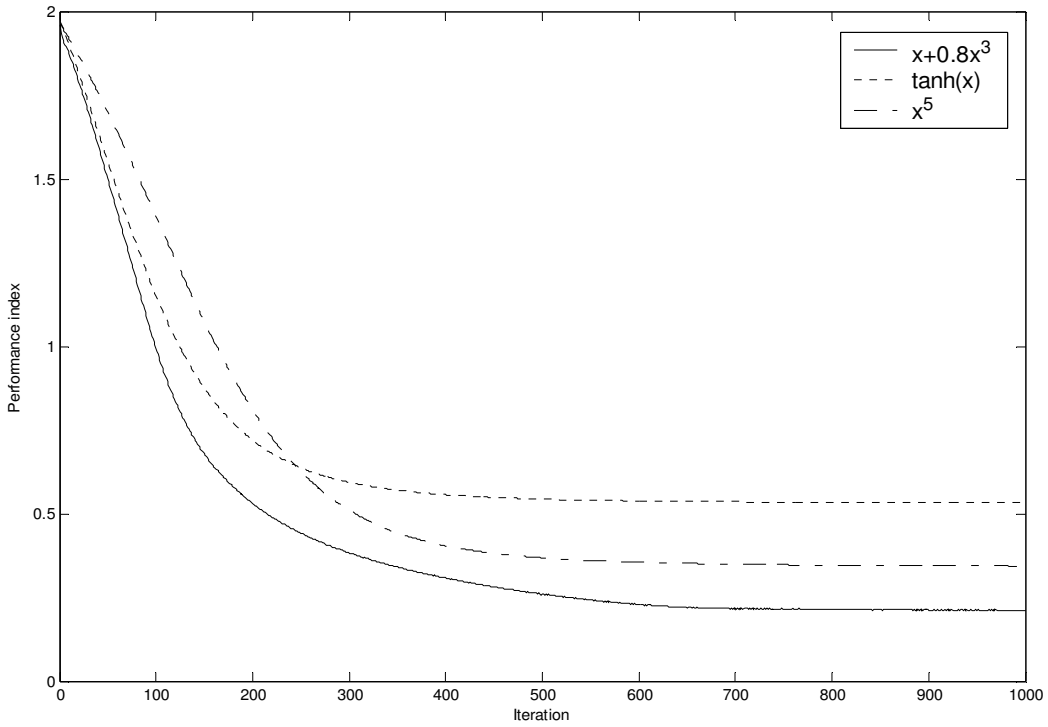
x+0.8x3
tanh(x)
x5
Performance index
Iteration
0
0.5
1
1.5
2
0
100
200
300
400
500
600
700
800
900
1000

Fig.5

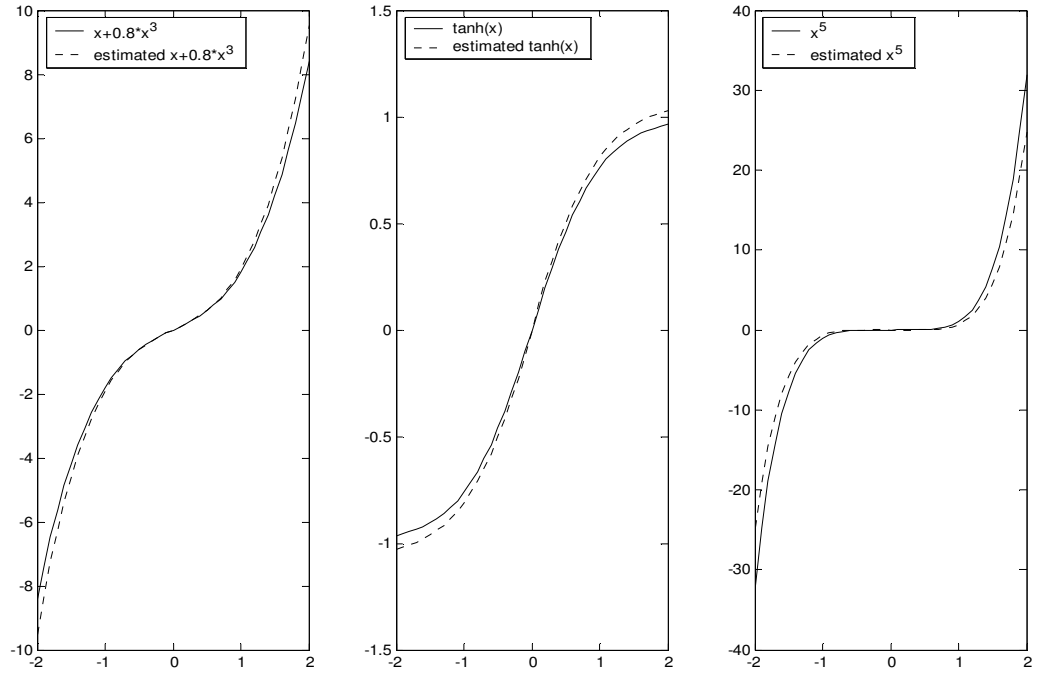
x+0.8*x3
estimated x+0.8*x3
tanh(x)
estimated tanh(x)
x5
estimated x5

Fig.6

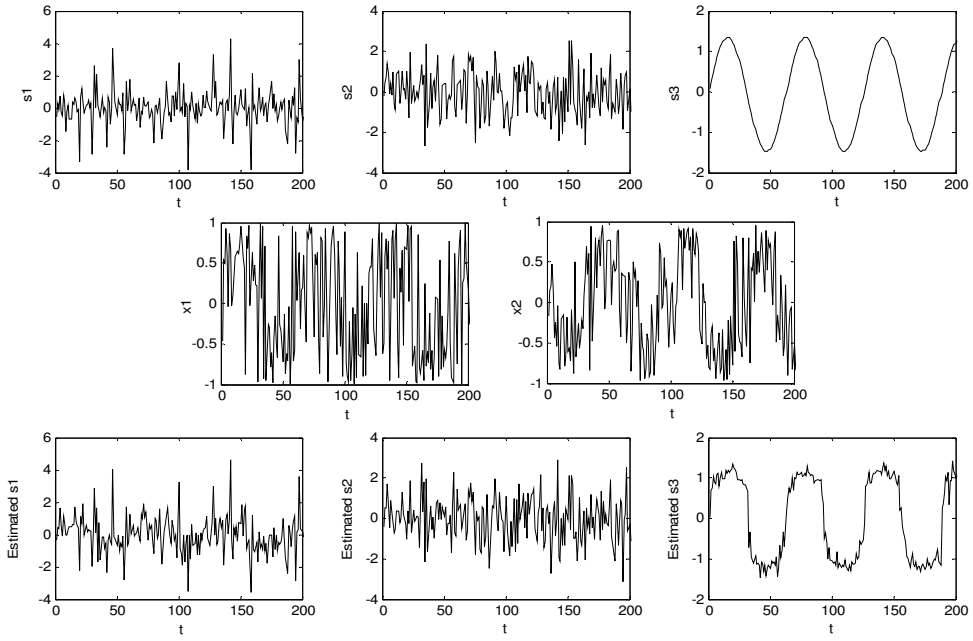
s1
s2
s3
x1
x2
Estimated s1
Estimated s2
Estimated s3
t

Fig.7

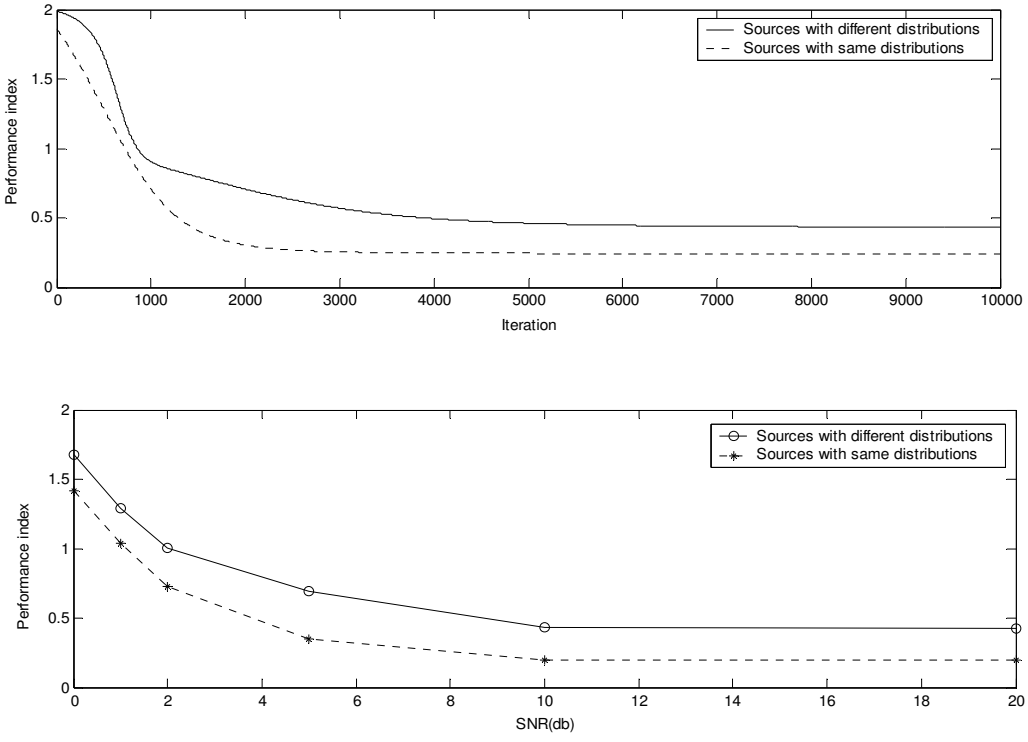
Sources with different distributions
Sources with same distributions
Performance index
Iteration
SNR(db)

Fig.8

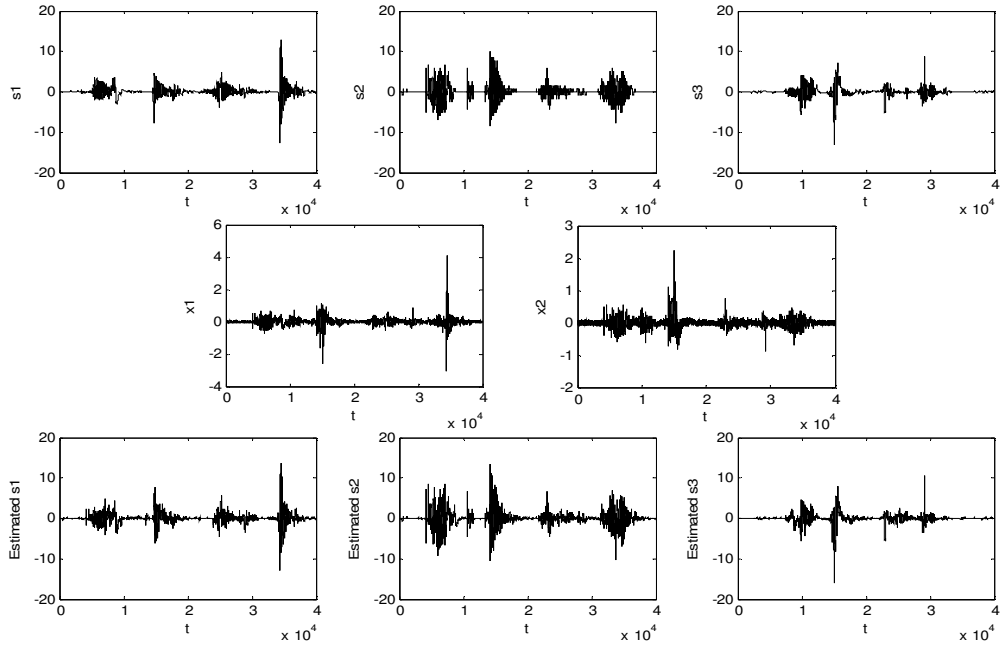
s1
s2
s3
x1
x2
Estimated s1
Estimated s2
Estimated s3
t
x 10^4

Fig.9

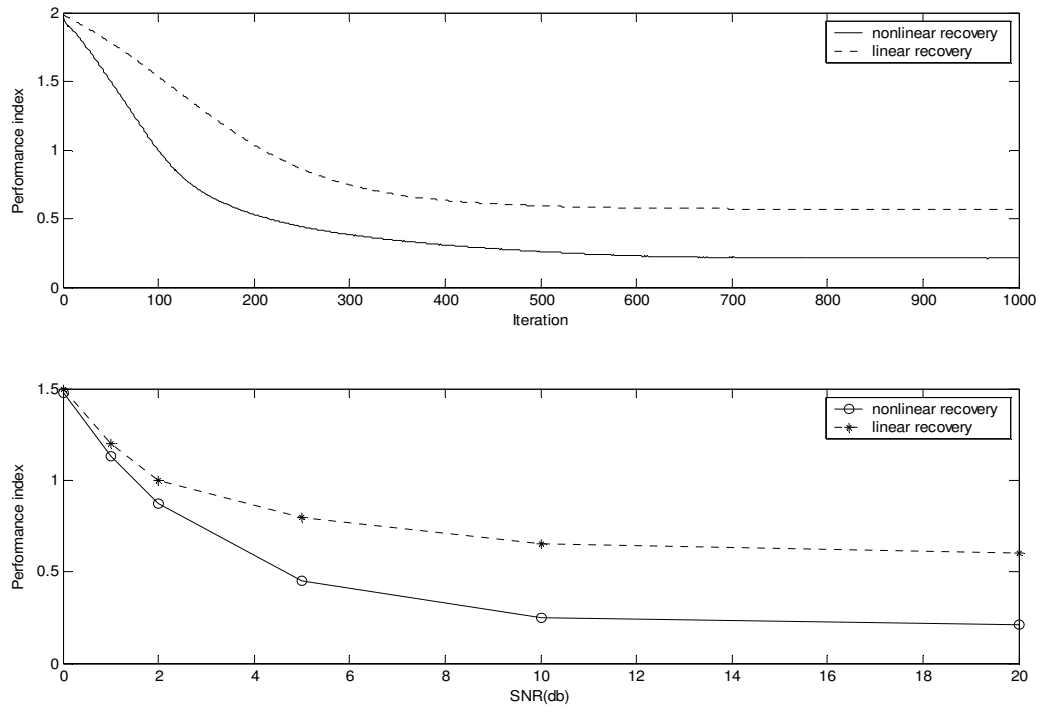
nonlinear recovery
linear recovery
Performance index
Iteration
SNR(db)

Fig.10

FIGURE CAPTION

Fig.1 Post nonlinear mixing model.

Fig.2 GGD model.
(1) p=1, super-gaussian distribution.
(2) p=2, normal distribution.
(3) p=20, sub-gaussian distribution.

Fig.3 Simulation using three Laplacian signals.
Top: three source signals.
Middle: two mixtures.
Bottom: three recovered signals.

Fig.4 Performance index of sources and recovered signals when SNR=20dB.
Top: nonlinear and linear algorithm comparison.
Bottom: Performance index as a function of SNR.

Fig.5 Performance index of matched and mismatched nonlinear function when SNR=20dB.

Fig.6 Estimated polynomial nonlinear function compared with true nonlinear distortion function.

Fig.7 Simulation using signals with different distributions.
Source signals: Laplacian (*top left*), Gaussian (*top central*) and sine wave (*top right*).
Mixtures (*middle*).
Recovered signals (*bottom*).

Fig.8 Performance Index.
Top: Performance index of recovered signals under SNR=20dB.
Bottom: Performance index as a function of SNR.

Fig.9 Simulation using speech signals.
Top: three source signals.
Middle: two mixtures.
Bottom: three recovered signals.

Fig.10 Performance Index.
Top: performance index of recovered speech under SNR=20dB.
Bottom: Performance index as a function of SNR.