Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2007, Article ID 31951, 12 pages
doi:10.1155/2007/31951

Research Article

gpICA: A Novel Nonlinear ICA Algorithm Using Geometric Linearization

Thang Viet Nguyen, Jagdish Chandra Patra, and Sabu Emmanuel

School of Computer Engineering, Nanyang Technological University, Singapore 639798

Received 30 September 2005; Revised 21 March 2006; Accepted 11 June 2006

Recommended by Frank Ehlers

A new geometric approach for nonlinear independent component analysis (ICA) is presented in this paper. Nonlinear environment is modeled by the popular post nonlinear (PNL) scheme. To eliminate the nonlinearity in the observed signals, a novel linearizing method named as geometric post nonlinear ICA (gpICA) is introduced. Thereafter, a basic linear ICA is applied on these linearized signals to estimate the unknown sources. The proposed method is motivated by the fact that in a multidimensional space, a nonlinear mixture is represented by a nonlinear surface while a linear mixture is represented by a plane, a special form of the surface. Therefore, by geometrically transforming the surface representing a nonlinear mixture into a plane, the mixture can be linearized. Through simulations on different data sets, superior performance of gpICA algorithm has been shown with respect to other algorithms.

1. INTRODUCTION

Independent component analysis (ICA), a technique of separating the unknown source signals from their mixtures, has been extensively studied in the last two decades [1] and successfully applied in many fields such as signal processing, biomedical engineering, medical imaging, speech enhancement, remote sensing, and data mining [2]. With a single hypothesis of the statistical independence of source signals, an ICA algorithm is able to estimate the unknown sources without any training data or a prior knowledge of these signals.

The simple linear ICA model where original signals are assumed to be linearly mixed by a mixing matrix was the first approach to ICA problem. Until now, linear ICA has become relatively well established with many effective algorithms [3–7]. However, because of its nature, linear model can perform well in the linear environment only. This drawback limits the use of linear ICA in various practical applications whose environment is naturally a nonlinear mixing system. Therefore, a general nonlinear ICA model, in which the mixing matrix is replaced by a nonlinear multidimensional mixing function, was introduced. The early methods for general model were proposed in [8, 9] with the use of self-organizing maps to estimate the independent components. Another method called LOCOCODE [10, 11] that applies minimum description length principle to optimize the number of bits used to encode data can also be considered as one of the pioneers in nonlinear ICA. The problems of general nonlinear ICA, however, are the computational complexity and the nonuniqueness of solutions [2, 12]. Without any constraint to the model, there always exists an infinite number of solutions [13]. Therefore, researchers in recent years have focused on the submodels, that is, a variation of general model with constraints, in order to develop nonlinear ICA algorithms that yield unique solution.

The post nonlinear (PNL) model introduced by Taleb and Jutten [14] is among the most popular submodels. PNL model limits the mixing system to a linear mixing stage followed by a nonlinear one-dimensional distortion for each mixture. Therefore, the model is much simpler and provides a unique solution. It has attracted the interest of various researchers [12, 14–16] and has found many applications, for example, sensor array processing [17], digital satellite and microwave communications [18], and biological systems [19].

In this paper, we propose a novel method called gpICA (geometric PNL ICA) for the nonlinear ICA problem. The gpICA method is tailored to the PNL model and employs a geometric linearizing technique to deal with the nonlinearity problem. Our major contribution, the geometric linearizing technique, comes from the following observation: in a multidimensional space, the graph of a nonlinear mixture

is a nonlinear surface. In special case, when the mixture is a linear one, its graph becomes a plane. Hence, the objective of converting a given nonlinear mixture to a linear one can be achieved by transforming a nonlinear surface (the geometric representation of the nonlinear mixture) to a plane (the geometric representation of the linear mixture). Thereafter, any algorithm which is applicable to linear ICA can be applied on the linearized mixtures to extract the original signals.

Our geometric approach was first introduced in [20] in which only midpoint transformation was used. A preliminary version of the current multiarbitrary point transformation was proposed in [21]. Compared to the previous midpoint technique, the utilization of multiarbitrary point transformation enhances the ability of gpICA in many hard nonlinear situations. In this paper, more details on theoretical development and performance comparison are provided. With the novel geometric technique, gpICA possesses the following advantages.

(1) Linearization can be done without any knowledge or assumption on the number or distribution of the unknown sources, the mixing system, or the observed signals.
(2) After completion of the linearizing phase, one can choose the most suitable linear ICA method to accomplish the demixing process. It is very useful as each ICA method works well only on specific type of applications.
(3) The linearization is carried out geometrically, therefore, gpICA does not have any constraint on the nonlinear distortion.

This paper is organized as follows. An overview of ICA models is shown in Section 2. Principles of the geometric approach are introduced in Section 3 and the details of gpICA are presented in Section 4. We provide the computer simulation results in Section 5. Finally, in Section 6, we conclude and discuss the issues related to the proposed algorithm.

2. OVERVIEW OF ICA MODELS

2.1. Linear ICA model

Consider a system with n observations which are mixtures of n unknown signals. Let s_i be the unknown sources and let x_i be the observations ($i = 1, 2, \ldots, n$). Linear ICA model presumes that the n source signals are statistically independent and each observed signal is a linear mixture of the n sources. The mixing system, therefore, is described by

$$\mathbf{x} = \mathbf{A}\mathbf{s}, \tag{1}$$

where $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$ and $\mathbf{s} = [s_1, s_2, \ldots, s_n]^T$ are the vectors representing the observed signals and unknown source signals, respectively, and $\mathbf{A}$, termed as mixing matrix, is a full rank matrix of size $n \times n$. The elements of $\mathbf{A}$, a_{ij} ($i, j = 1, 2, \ldots, n$), can have any scalar value.

After modeling the system, our objective is to estimate these unknown signals, s_i, only from the knowledge of the observed signals, x_i. Let y_i ($i = 1, 2, \ldots, n$) denote n estimates of the unknown signals. These estimates, y_i, can be obtained by finding an inverse matrix $\mathbf{W}$ of $\mathbf{A}$, that is, $\mathbf{W} \approx \mathbf{A}^{-1}$, such that

$$\mathbf{y} = \mathbf{W}\mathbf{x} \approx \mathbf{A}^{-1}\mathbf{A}\mathbf{s} = \mathbf{s}, \tag{2}$$

where $\mathbf{y} = [y_1, y_2, \ldots, y_n]^T$ is a vector of the estimated signals. The matrix $\mathbf{W}$ of size $n \times n$ is termed as demixing matrix. The process to find $\mathbf{W}$, called separating process, is accomplished by maximizing the statistical independence among the n estimated outputs, y_i. For more details on linear ICA methods, see [1, 2].

2.2. General nonlinear model

General nonlinear ICA model is a natural extension of the linear model in which the mixing matrix is replaced by a nonlinear mixing function. The mixing process (1), therefore, is reformulated as

$$\mathbf{x} = \mathcal{F}(\mathbf{s}), \tag{3}$$

where $\mathcal{F}$ is an unknown real-valued n-component mixing function [12]. This extended model can be applied for the mixing environment that cannot be modeled by a linear mixing system. For example, the multiplicative mixtures $x_j = \prod_{i=1}^{n} s_i$ which are used for modeling the gray level image as a product of incident light and reflected light [22]. Obviously, linear ICA is a special case of (3) when $\mathcal{F}(\mathbf{s}) = \mathbf{A}\mathbf{s}$.

To find the estimates of the original signals, y_i, we have to build a separating system. That is, to find a mapping $\mathcal{G} : \mathbb{R}^n \to \mathbb{R}^n$ such that

$$\mathbf{y} = \mathcal{G}(\mathbf{x}) \approx \mathbf{s}. \tag{4}$$

Some of the early approaches to nonlinear ICA can be found in [8, 9]. The authors tried to apply self-organizing maps (SOM) [8] and later, the generative topographic mapping (GTM) [9] to the model in (3). However, these approaches to general model do not yield satisfactory result. They need a lot of computation and do not provide a unique solution. It is shown that the solutions of the general nonlinear model always exist, but they are nonunique and without any constraint to the model, there are infinite number of solutions [1, 13].

2.3. Post nonlinear (PNL) model

To overcome the nonuniqueness issue, recent studies on nonlinear ICA are focused on specific subclasses of the general model. Some constraints are applied on the mixing function $\mathcal{F}$ or the source signals in order to eliminate the nonuniqueness [12]. One of the important subclasses is the post nonlinear (PNL) model, introduced by Taleb and Jutten [14].

The post nonlinear (PNL) model shown in Figure 1 constrains the mixing system to a linear one followed by a single variate nonlinear distortion for each mixture [14].

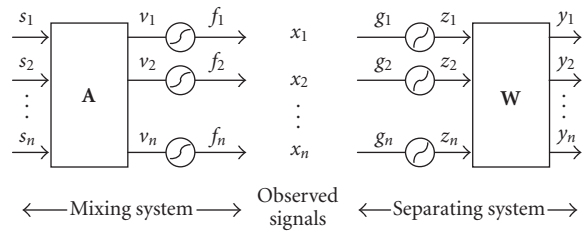

Figure 1: The post nonlinear (PNL) mixing-separating system.

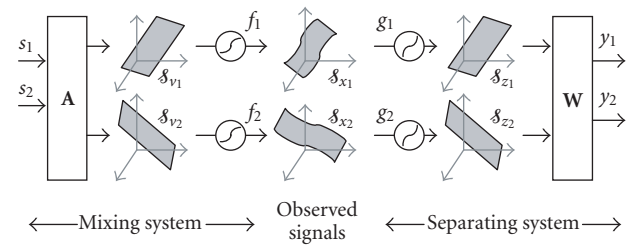

Figure 2: Geometric description of a PNL mixing and separating system in a 3D space.

Mathematically, the PNL mixing model can be specified as

$$\mathbf{v} = \mathbf{A}\mathbf{s}, \tag{5}$$

$$x_i = f_i(v_i), \quad i = 1, \ldots, n, \tag{6}$$

where $\mathbf{v} = [v_1, v_2, \ldots, v_n]^T$ is a vector of linear mixtures, and f_i are nonlinear functions. Because of its simplicity and plausibility, the PNL approach is the subject of many studies [14–16] and has found several applications [17–19].

In order to estimate the unknown source signals, a two-stage PNL separating model is applied. The PNL separating system can be model as

$$\begin{aligned} z_i &= g_i(x_i), \quad i = 1, \ldots, n, \\ \mathbf{y} &= \mathbf{W}\mathbf{z}, \end{aligned} \tag{7}$$

where $\mathbf{z} = [z_1, z_2, \ldots, z_n]^T$ and g_i are the single variate functions. In the first stage, termed as linearizing stage, the method attempts to transform the observed signals into linear mixtures. Then in the second stage, a basic linear ICA algorithm is applied to estimate the unknown sources from those linearized mixtures.

3. GEOMETRIC APPROACH FOR PNL MODEL

As we can see, linearization of observed signals is the most important task that all the PNL-tailored methods have to solve. In the early approach introduced by Taleb and Jutten [14], independent criteria were used in both stages. Therefore, the algorithm may not work well with the hard nonlinear distortions. In Howard et al.'s approach [16], the model required a lot of computation. Recently, Ziehe et al. [15] proposed a simple and effective PNL method but required the mixtures to have Gaussian or Gaussian-like distributions. In this paper, we propose a new geometric approach for the linearizing process. The proposed method does not impose any additional assumption and works effectively with the hard nonlinear distortions as well.

To begin with, we first represent the PNL problem under geometric viewpoint. The nonlinear and linear mixtures are represented in terms of nonlinear surfaces and planes in a multidimensional space. The linearizing process that changes PNL mixtures into linear ones is presented as a transformation of nonlinear surfaces into planes. The approach is first described in the case of two sources and two observed signals (the 2×2 case) so that it can be illustrated in a three-dimensional (3D) space. A solution for a general PNL case (more than two sources and observations) will be given later in Section 4.

Now let us recall the two concepts which will be used extensively in our method, the definitions of a surface and a plane.

Definition 1. Given $z = f(x, y)$, a graph of $f(x, y)$ is a set of all points $(x, y, f(x, y))$ in a 3D space XYZ and is called a surface. The function $f(x, y)$ can be any arbitrary function of x and y.

Definition 2. In a special case, when the function $f(x, y)$ is a linear one, that is, $f(x, y) = ax + by + c$, the graph of $f(x, y)$ in the 3D space is then called a plane.

Let us look at a specific 2×2 case of the PNL problem. The linear mixtures in (5), v_i, can be expressed as

$$v_i = a_{i1}s_1 + a_{i2}s_2 \tag{8}$$

and the PNL mixtures in (6), x_i, can be expressed as

$$x_i = f_i(v_i) = f_i(a_{i1}s_1 + a_{i2}s_2), \tag{9}$$

where $i = 1, 2$. Let us consider a 3D space XYZ, where the values s_1 and s_2 are shown on X- and Y-axes, respectively. The values of v_i or z_i are shown on Z-axis. Let us denote $\mathcal{S}_{v_i}$ and $\mathcal{S}_{x_i}$ as the graphs of v_i and x_i, respectively. Since v_i is described as a linear (8), its graph, $\mathcal{S}_{v_i}$, in this 3D space is clearly a plane. The PNL mixture, x_i, on the other hand, is represented by a nonlinear function (9) and its graph, $\mathcal{S}_{x_i}$, is in the form of a nonlinear surface. Therefore, PNL mixing model can be viewed as a creation of the planes followed by a surface-transforming process from these planes. Consequently, PNL separating system, as a reverse process, is to transform the input surfaces back to the planes and then uses these planes to estimate the original coordinates. An illustration of the PNL model is shown and described in Figure 2 in terms of the transformation from planes to surfaces and vice versa.

In Figure 2, a plane $\mathcal{S}_{v_1}$, the set of points having coordinates (s_1, s_2, v_1), is drawn as a graph of the linear function v_i in (8). The plane, $\mathcal{S}_{v_1}$, is transformed to a nonlinear surface $\mathcal{S}_{x_1}$ when v_1 is distorted by a nonlinear function in (9). Likewise, plane $\mathcal{S}_{v_2}$ and then surface $\mathcal{S}_{x_2}$ are created by the similar processes. An example of the mixing system is depicted in Figure 3. From 10 000 samples in the range $[-1, 1]$

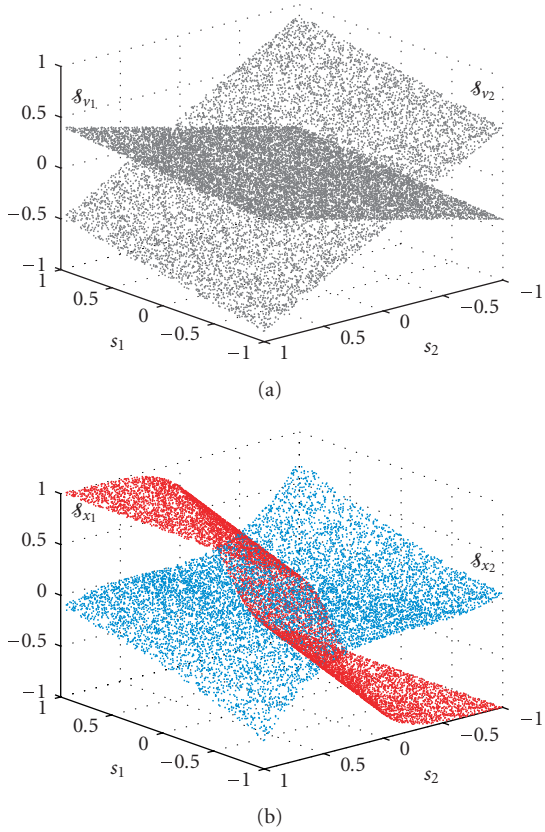

Figure 3: Example of a PNL mixing system in 3D space. (a) Planes $\mathcal{S}_{v_1}$ and $\mathcal{S}_{v_2}$ correspond to linear mixtures v_1 and v_2. (b) Nonlinear surfaces $\mathcal{S}_{x_1}$ and $\mathcal{S}_{x_2}$ correspond to PNL mixtures x_1 and x_2.

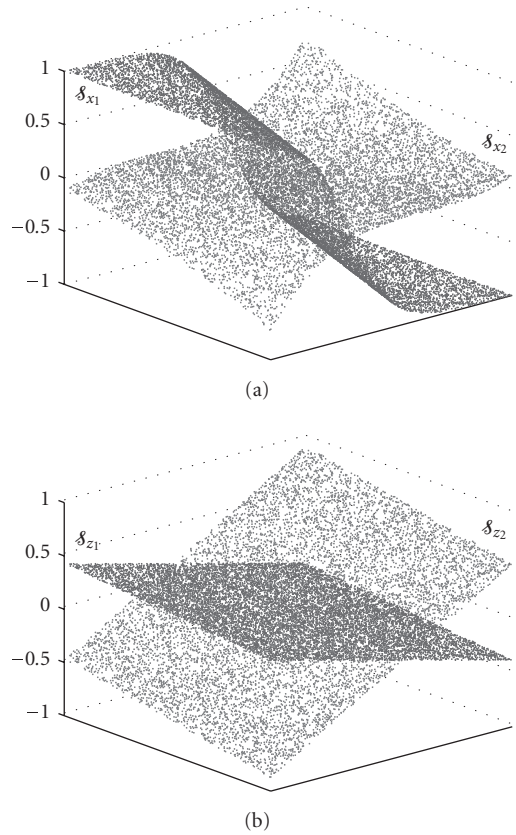

Figure 4: Example of the PNL separating system in 3D space. (a) Nonlinear surface $\mathcal{S}_{x_1}$ and $\mathcal{S}_{x_2}$ correspond to the observations. (b) Planes $\mathcal{S}_{z_1}$ and $\mathcal{S}_{z_2}$ correspond to linearized mixtures z_1 and z_2.

of random signals s_1 and s_2, the graphs of two linear mixtures $v_1 = 0.1s_1 + 0.3s_2$ and $v_2 = 0.2s_1 - 0.7s_2$ are created and shown in Figure 3(a). Next, the PNL mixtures $x_1 = \tanh(10v_1)$ and $x_2 = v_2^3$ are constructed and their graphs, $\mathcal{S}_{x_1}$ and $\mathcal{S}_{x_2}$, are shown in Figure 3(b).

Before going into the details of the PNL separating system, we need to describe the characteristics of the observed graphs. Since s_1 and s_2 are unknown in a PNL problem, the observed graph $\mathcal{S}_{x_i}$, in fact, is a set of points having coordinates $(\_, \_, x_i)$, where the symbol "$\_$" denotes an unknown value. In other words, we are given two graphs and the value of their points on the third coordinate. Our objective is to estimate the values on the first two coordinates of these points.

To achieve this objective, we apply a two-stage scheme. In the first stage, the critical linearizing process is accomplished by geometric transformation. The surfaces $\mathcal{S}_{x_i}$ $(i = 1, 2)$ are transformed to planes $\mathcal{S}_{z_i}$ which correspond to the linearized mixtures z_i. In the second stage, a basic ICA algorithm is applied on z_i to estimate the unknown sources. Figure 4 illustrates a geometric view of the PNL separating system. Note that the labels s_1 and s_2 which appeared in Figure 3 have been removed from Figure 4 in order to reflect the unavailability of the knowledge of source signals.

The problem of transforming surface to plane would be quite easy if the values s_1 and s_2 were known. The construction of $\mathcal{S}_{z_i}$ can be achieved by letting $z_i = w_{i1}s_1 + w_{i2}s_2$ and selecting any nonzero scalar value for w_{i1} and w_{i2} $(i = 1, 2)$. In PNL problem described above, however, this kind of information is not available. Thus, to carry out the transformation $\mathcal{S}_{x_i}$ into $\mathcal{S}_{z_i}$, the following issues need to be resolved.

(1) To identify whether a given graph $\mathcal{S}_z$ is a plane without prior knowledge of s_1 and s_2.
(2) To evolve a mechanism to generate $\mathcal{S}_{z_i}$ from $\mathcal{S}_{x_i}$, also without knowledge of s_1 and s_2.

3.1. Identification of a plane

From the nature of plane and surface in 3D space, the following property is observed.

Proposition 1. *Let p_1 and p_2 denote two arbitrary points lying on a surface $\mathcal{S}$ in a 3D space and let p_c denote an arbitrary point lying on a straight line joining p_1 and p_2. $\mathcal{S}$ is a plane if and only if for all $p_1, p_2 \in \mathcal{S}$, p_c lies on $\mathcal{S}$.*

An illustration of Proposition 1 is shown in Figure 5. In Figure 5(a), two points p_1 and p_2 are lying on $\mathcal{S}_v$ and p_c is an arbitrary point on $\overline{p_1 p_2}$. Since $\mathcal{S}_v$ is a plane ($v = 0.1s_1 + 0.3s_2$ in the example), we have $p_c \in \mathcal{S}_v$. Whereas, in Figure 5(b),

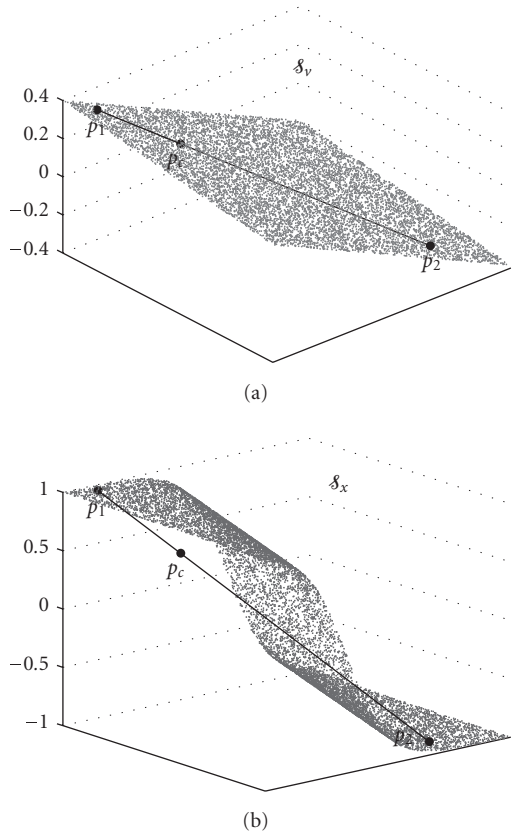

FIGURE 5: An example showing the difference between a plane and a nonlinear surface. (a) A plane: arbitrary point $p_c \in \overline{p_1 p_2}$ lies on $\mathcal{S}_v$. (b) A nonlinear surface: the point $p_c \in \overline{p_1 p_2}$ falls out of $\mathcal{S}_x$.

since $\mathcal{S}_x$ is not a plane ($z = \tanh(10v) = \tanh(10(0.1s_1 + 0.3s_2))$ in the example), p_c does not lie on $\mathcal{S}_x$.

3.2. Transformation of a nonlinear surface to a plane

In order to construct a plane $\mathcal{S}_z$ from a given nonlinear surface $\mathcal{S}_x$, we propose a heuristic method that utilizes properties of the straight line in a 3D space. Given an XYZ coordinate system, the following definitions and propositions are observed.

Definition 3. The equation of a line containing a point $p_1(x_1, y_1, z_1)$ can be written as $(x - x_1)/a = (y - y_1)/b = (z - z_1)/c$, where a, b, c are the scalars and are not all zeros.

Definition 4. Let $p_1(x_1, y_1, z_1)$ and $p_2(x_2, y_2, z_2)$ be two arbitrary points. The pair $\{p_1, p_2\}$ is called a companion pair, $\mathcal{C}_{p_1,p_2}$, if and only if $x_1 = x_2$ and $y_1 = y_2$. The point p_1 is then called a companion of p_2 and vice versa.

Proposition 2. *Given an arbitrary point, $p_1(\_, \_, z_{p1})$, and its companion point, $q_1(\_, \_, z_{q1})$, let $p_2(\_, \_, z_{p2})$ be another arbitrary point and let $q_2(\_, \_, z_{q2})$ be its companion point. Let $p_c(\_, \_, z_{pc})$ and $q_c(\_, \_, z_{qc})$ be the two arbitrary points lying on $\overline{p_1 p_2}$ and $\overline{q_1 q_2}$, respectively. If p_c is the companion of q_c, then the following expression holds true:*

$$\frac{z_{pc} - z_{p1}}{z_{p2} - z_{p1}} = \frac{z_{qc} - z_{q1}}{z_{q2} - z_{q1}}. \tag{10}$$

The proposition is a direct consequence of Thales' theorem. Its proof is provided in the appendix.

Proposition 2 is the key of our technique. Given five points p_1, p_2, q_1, q_2, and p_c, we can locate the position of the sixth point, q_c, by using (10). Now assume that a reference plane $\mathcal{S}_v$ is known, and our task is to transform a nonlinear surface $\mathcal{S}_x$ into a plane $\mathcal{S}_z$. The following transformation scheme will detail our technique to transform $\mathcal{S}_x$ into $\mathcal{S}_z$ with the help of the reference plane $\mathcal{S}_v$.

Transformation scheme

(1) Pick up any two arbitrary points p_1 and p_2 on the reference plane $\mathcal{S}_v$. Locate their respective companion points q_1 and q_2 on $\mathcal{S}_x$.
(2) Select an arbitrary point p_c on $\overline{p_1 p_2}$. Find a companion point, $q_c \in \overline{q_1 q_2}$, of p_c by using (10).
(3) Locate q_x on $\mathcal{S}_x$ such that $\{p_c, q_x\}$ is a companion pair.
(4) Pull q_x toward q_c.
(5) Use Proposition 1 to check whether $\mathcal{S}_x$ is a plane.
(6) Repeat steps (1) to (5) if it is not a plane, otherwise stop.

Figure 6 illustrates a change of point q_x in one iteration. The old location of q_x is shown in Figure 6(a). After a transformation iteration, q_x is changed to q_c (Figure 6(b)) which lies on the same straight line joining q_1 and q_2. The transformation is repeated for every point $q_x \in \mathcal{S}_x$ and at the end, the surface $\mathcal{S}_x$ is transformed to a plane $\mathcal{S}_z$.

4. gpICA ALGORITHM

In this section, the above transformation scheme will be applied to create our geometric linearizing technique. However, the following issues have to be solved due to the unavailability of s_1 and s_2.

(1) Since the values of s_1 and s_2 are not known, the first two coordinates of a point are not available. Therefore, we cannot apply Definition 4 directly to identify the companion of a given point.
(2) In PNL problem, a reference plane is also not available.
(3) The ambiguity of the point $q_c \in \overline{q_1 q_2}$. Since the first two coordinates are unknown, we cannot ensure whether $p_c \in \overline{p_1 p_2}$. Therefore, the reverse of Proposition 2 does not always hold, that is, there may be more than one point that satisfy (10) but are not the companion of p_c.

Here we intend to use the time index of a mixture to identify a companion point. Let $v_1(t_1)$ denote a sample of linear mixture v_i at time $t = t_1$. Then the point on the surface $\mathcal{S}_{v_1}$ representing this sample is given by the coordinates $(s_1(t_1), s_2(t_1), v_1(t_1))$. Similarly, at the sampling time

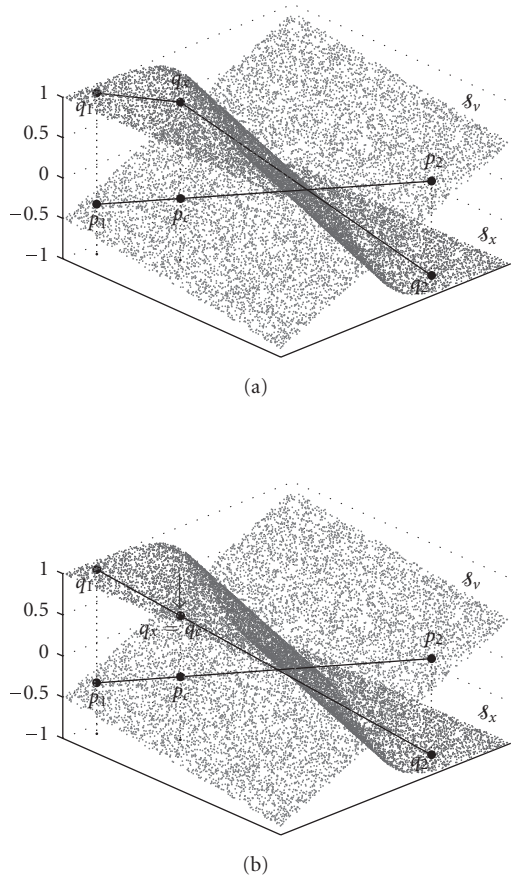

Figure 6: An illustrative example of linearizing $\mathcal{S}_x$ using reference plane $\mathcal{S}_v$. (a) The point $q_x \in \mathcal{S}_x$ before transformation. (b) After the transformation, q_x was pulled to $q_c \in \overline{q_1 q_2}$.

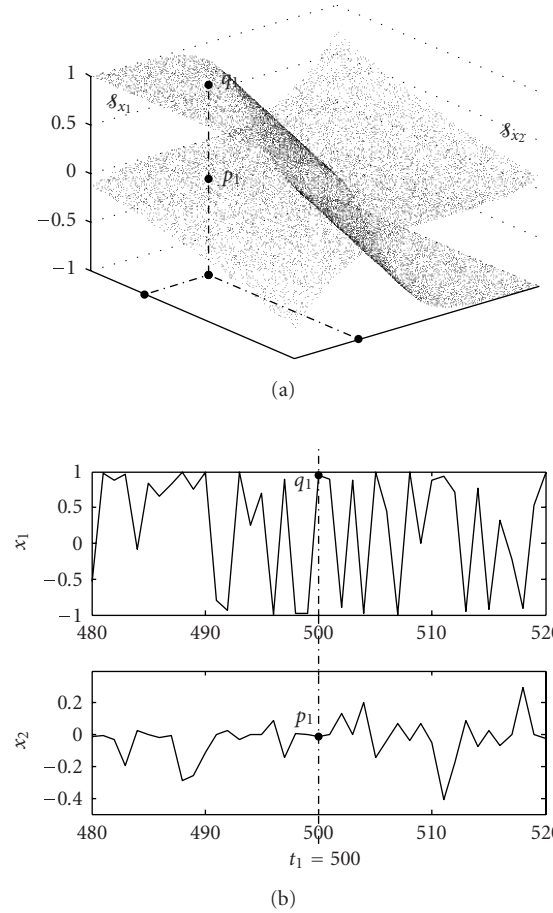

Figure 7: Example of a companion pair $\mathcal{C}_{p_1,q_1}$. (a) In a 3D space, p_1 and q_1 have the same values on the coordinates s_1 and s_2. (b) In time series plots, p_1 and q_1 are at the same time instant.

$t = t_1$, the coordinates of the corresponding points on the surfaces $\mathcal{S}_{v_2}$, $\mathcal{S}_{x_1}$, and $\mathcal{S}_{x_2}$ are given by $(s_1(t_1), s_2(t_1), v_2(t_1))$, $(s_1(t_1), s_2(t_1), x_1(t_1))$, and $(s_1(t_1), s_2(t_1), x_2(t_1))$, respectively. Considering PNL problem modeled by (8) and (9), the following proposition is observed.

Proposition 3. *Given the time instant $t = t_1$, then all the points $p_{1_v}(\_, \_, v_1(t_1)) \in \mathcal{S}_{v_1}$, $q_{1_v}(\_, \_, v_2(t_1)) \in \mathcal{S}_{v_2}$, $p_1(\_, \_, x_1(t_1)) \in \mathcal{S}_{x_1}$, and $q_1(\_, \_, x_2(t_1)) \in \mathcal{S}_{x_2}$ are the companions.*

An example of Proposition 3 is shown in Figure 7. In 3D space, p_1 and its companion q_1 are shown in Figure 7(a) as the two points having the same values on the first two coordinates. Meanwhile, this companion property is shown in the time series plots in Figure 7(b) as the two signal samples having the same time instant $t = t_1$. Thus with Proposition 3, companion points can be identified without knowing the coordinates s_1 and s_2.

For the second issue of finding a reference plane, a proposed solution is to employ a "fake plane," that is, to assume one surface as reference plane and use it to transform the other surface. The role of "fake plane" will be alternatively changed from one surface to another during the transformation process. For this reason, the transformation process mentioned in Section 3 is modified to adapt with the fake plane. Let us choose $\mathcal{S}_{x_2}$ as the fake plane and $\mathcal{S}_{x_1}$ as the surface that needs to be transformed. Given two points $p_1(\_, \_, x_2(t_1))$ and $p_2(\_, \_, x_2(t_2))$ lying on $\mathcal{S}_{x_2}$ and the other two points $q_1(\_, \_, x_1(t_1))$ and $q_2(\_, \_, x_1(t_2))$ lying on $\mathcal{S}_{x_1}$, let p_c be a point being located at $(\_, \_, x_2(t_c))$. Then the value on the third coordinate of the companion point of p_c, $q_c(\_, \_, z_1(t_c))$, can be computed from (10) as

$$z_1(t_c) = \frac{x_2(t_c) - x_2(t_1)}{x_2(t_2) - x_2(t_1)} \left(x_1(t_2) - x_1(t_1)\right) + x_1(t_1). \quad (11)$$

The last issue is the ambiguity of the point $q_c \in \overline{q_1 q_2}$. Proposition 2 implies that if $p_c \in \overline{p_1 p_2}$ and p_c and q_c are the companion points, then (10) holds true, but the reverse is not always true. There may be more than one point q_c that satisfy (10) but are not a companion of p_c. Also, we cannot ensure that p_c will lie on $\overline{p_1 p_2}$. An incorrect selection of q_c or p_c will make q_x, the point that is supposed to be changed, move to a wrong position. Currently in this paper, there is no complete solution to eliminate this ambiguity. Nevertheless, to lessen the inaccuracy, we apply a local transformation, that is,

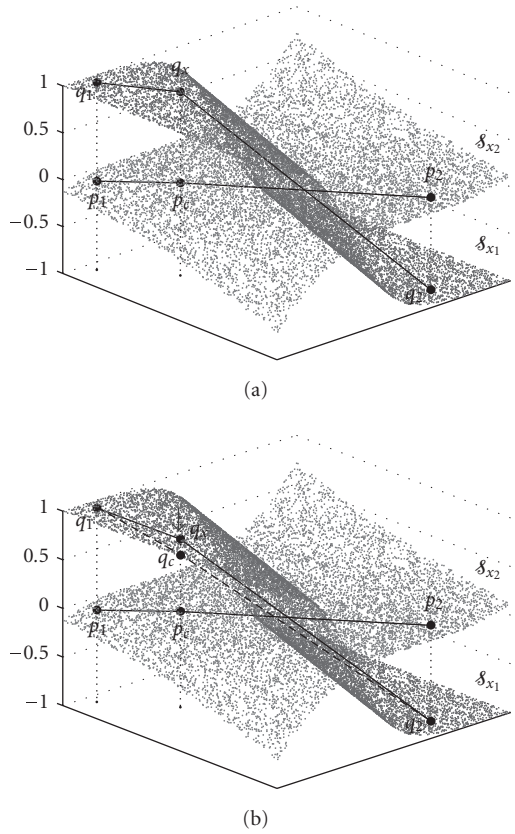

FIGURE 8: An example of gpICA method. The surface $\mathcal{S}_{x_1}$ is transformed using the fake plane $\mathcal{S}_{x_2}$. Position of q_x (a) before and (b) after the transformation. Because $\mu < 1$, q_x is pulled to a new position near q_c.

the surface is divided into small cells and the transformation process is carried out within these cells. In addition, instead of pulling q_x right to q_c, we apply a learning rate $\mu < 1$ to update q_x position. By doing this way, the transformation will take a longer time but will steadily converge to a plane. The updating function is formulated as

$$x_1^{\text{new}}(t_c) = \mu z_1(t_c) + (1-\mu)x_1^{\text{old}}(t_c). \tag{12}$$

Figure 8 shows an example of our proposed method in an iteration. The two surfaces $\mathcal{S}_{x_1}$ and $\mathcal{S}_{x_2}$ represent two nonlinear mixtures x_1 and x_2, respectively. The surface $\mathcal{S}_{x_2}$ is assumed to be a "fake plane" and is used to transform $\mathcal{S}_{x_1}$ into a plane. The position of the selected point, q_x, before the change is illustrated in Figure 8(a). The new location of q_x after the change is shown in Figure 8(b). With the learning rate $\mu < 1$, q_x does not move right to q_c but to a location near q_c.

After the proposed linearizing process, we obtained two planes, $\mathcal{S}_{z_1}$ and $\mathcal{S}_{z_2}$, which represent the linearized signals z_1 and z_2, respectively. To improve the smoothness of these surfaces, a smoothing function is applied. In this work, we use an averaging function to smooth the signals. The signal is sorted in an ascending order, and the sorted signal, z^s, is smoothed using the function

$$\widetilde{z}_i(t) = \frac{1}{L}\sum_{j=-(L-1)/2}^{(L-1)/2} z_i^s(t+j). \tag{13}$$

Then $\widetilde{z}_i$ is restored to the original order to produce signals z_i. These smoothed signals, z_i, are used as the inputs for the linear ICA algorithm applied in the demixing stage.

Finally, the algorithm is extended for the case of $n > 2$ observations. Each signal x_i $(i = 1, 2, \ldots, n)$ is represented by a surface $\mathcal{S}_{x_i}$. In an iteration, a surface $\mathcal{S}_{x_k}$ is randomly chosen and used as the "fake plane" to transform the other surfaces. Equation (11), therefore, is updated to

$$z_i(t_c) = \frac{x_k(t_c) - x_k(t_1)}{x_k(t_2) - x_k(t_1)}\left(x_i(t_2) - x_i(t_1)\right) + x_i(t_1). \tag{14}$$

Likewise, the updating equation (12) is modified as

$$x_i^{\text{new}}(t_c) = \mu z_i(t_c) + (1-\mu)x_i^{\text{old}}(t_c). \tag{15}$$

The stopping criteria "$\mathcal{S}_x$ is a plane" of the transformation scheme in Section 3 also need to be realized into a precise condition in order to complete the gpICA algorithm. In this work, we measure the difference (error) between the values of the signal after and before the change in one iteration. In each iteration, these errors are accumulated and compared to a threshold ξ. The accumulated error, ϵ, is computed by

$$\epsilon = \frac{1}{nN_k}\sum_{i=1}^{n}\sum_{j=1}^{N_k}\left(x_i^{\text{new}}(j) - x_i^{\text{old}}(j)\right)^2, \tag{16}$$

where $x_i^{\text{new}}(j)$ and $x_i^{\text{old}}(j)$ are the signal values after and before the change, respectively, and N_k is the number of updated samples in each iteration. The linearizing process is stopped when ϵ goes below a threshold ξ. After that a basic linear ICA algorithm is applied on the linearized signals, z_i $(i = 1, 2, \ldots, n)$, to produce the estimates of source signals, y_i. A framework of gpICA is described by the following pseudocode.

5. SIMULATION RESULTS

To evaluate the performance of gpICA, several computer simulations have been carried out on different data sets. The first test was done on a simple case of two sinusoidal sources. The sources, mixtures, and estimated signals, therefore, can be visualized in a 3D space. To assess the performance of the gpICA in a multiple-source PNL problem, we carried out the second simulation with four speech signals.

We have applied several typical linear ICA methods for the linear demixing stage (step (5)), such as SOBI (second-order blind identification) [5], JADETD (joint approximate diagonalization of eigen matrices with time delays) [23], and FPICA (fixed-point ICA) [6]. As results of these methods were almost identical, we only provide the outcomes that were carried out with SOBI during the simulation.

To give a perspective comparison of the algorithm's performance, we compare our method with a linear ICA algorithm and a competitive PNL method. For linear one, we

Input
$x_1, x_2, \ldots, x_n$: n observed signals with N samples each.
Parameter
μ: learning rate.
ξ: threshold to stop the linearization.
N_k: number of points to be updated in each iteration.
Output
$z_1, z_2, \ldots, z_n$: n linearized signals.
$y_1, y_2, \ldots, y_n$: n estimated signals.
function gpICA()
{
  Repeat {
    Randomly select fake plane x_k.
    For $i = 1$ to n {
      For $j = 1$ to N_k {
        Randomly generate time indices t_1, t_2, and t_c.
        Compute $z_i(t_c)$ using (14).
        Update $x_i(t_c)$ using (15).
      } /* end of j loop */
    } /* end of i loop */
    Compute ϵ using (16).
  } Until ($\epsilon < \xi$)
For $i = 1$ to n assign $z_i = x_i$.
For $i = 1$ to n smooth z_i using (13).
Apply a linear ICA algorithm on z_i to extract y_i.
} /* end of gpICA */

Algorithm 1: The Geometric PNL Algorithm: gpICA

choose SOBI, and for the PNL one, we choose Gauss-TD [15], one of the effective reported PNL methods. The performance was measured by the correlation coefficient between an original source, s, and an output signal, y. The correlation coefficient between s and y, $r_{(x,y)}$, is computed by

$$r_{(s,y)} = \frac{\sum_{t=1}^{N} (s(t) - \bar{s})(y(t) - \bar{y})}{\sqrt{\sum_{t=1}^{N} (s(t) - \bar{s})^2 \sum_{t=1}^{N} (y(t) - \bar{y})^2}}, \quad (17)$$

where $\bar{s} = (1/N)\sum_{t=1}^{N} s(t)$, $\bar{y} = (1/N)\sum_{t=1}^{N} y(t)$, with N as the number of samples.

5.1. Experiment 1: mixture of two sinusoidal signals

Using two sinusoidal signals in (18), two linear mixtures v_1 and v_2 were generated by a mixing matrix $\mathbf{A}$ whose entries were the random numbers in the range of $[-1, 1]$. The linear mixtures were then distorted by two nonlinear functions in (20) to produce the observations x_1 and x_2. The source signals, mixing matrix, and nonlinear functions are given below:

$$\begin{aligned} s_1(t) &= \sin(0.22t), \\ s_2(t) &= \sin\left(2\pi(0.1t) + 6\cos\left(2\pi(0.02t)\right)\right), \end{aligned} \quad (18)$$

$$\mathbf{A} = \begin{bmatrix} -0.605 & 0.152 \\ 0.625 & 0.056 \end{bmatrix}, \quad (19)$$

$$\begin{aligned} x_1(t) &= \left(3v_1(t)\right)^3, \\ x_2(t) &= \tanh\left(10v_2(t)\right). \end{aligned} \quad (20)$$

At first, the linearizing process was run on 3000 samples ($N = 3000$) with the window size of the smoothing function $L = 151$, the learning rate $\mu = 0.2$, and the error threshold $\xi = 0.002$. The 3D plots of the linearized signals, z_i, are shown in Figure 9(c) and are compared with the plots of linear mixtures, v_i, (Figure 9(a)) and PNL mixtures, x_i, (Figure 9(b)). As it is expected, the graphs of gpICA's result, z_i, are similar to the graphs of unknown linear mixtures, v_i. Clearly, the nonlinearity in x_i has been eliminated in z_i. Whereas, some nonlinearity is still visible in the graphs (Figure 9(d)) of the competitive Gauss-TD algorithm.

In the next stage, a linear ICA method, SOBI, was applied on z_i to extract the estimates of original signals, y_i. The plots of these estimates, y_i, are illustrated in Figure 10, together with plots of source signals, linear mixtures, PNL mixtures, and Gauss-TD and SOBI estimates. Compared with those of SOBI and Gauss-TD, gpICA's results resemble much more to the original source signals. The linear ICA algorithm provided the worst performance (Figure 10(d)). A quantitative performance was measured in terms of the correlation coefficient between a source signal and its estimate $\hat{s}_i$, $r_{(s_i,\hat{s}_i)}$. The estimate of the ith source signal, $\hat{s}_i$, is one of the outputs y_j ($j = 1, 2, \ldots, n$) whose absolute correlation coefficient, $|r_{(y_j,s_i)}|$, is the highest one. Using this index, $r_{(s_i,\hat{s}_i)}$, comparisons between gpICA and SOBI, and Gauss-TD were carried out and are reported in Table 1. As it is shown in the table, the result obtained by gpICA is better than both SOBI and Gauss-TD results, providing high correlation coefficients, $|r_{(\hat{s}_i,s_i)}| \approx 1$.

5.2. Experiment 2: mixture of four speech signals

In this simulation, 5000 samples of four speeches (taken from [24]) were chosen as the original sources. Their linear and nonlinear mixtures are given by

$$\mathbf{A} = \begin{bmatrix} -0.087 & 0.714 & -0.835 & 0.448 \\ 0.283 & -0.560 & -0.842 & -0.130 \\ -0.086 & 0.093 & -0.061 & 0.522 \\ -0.483 & -0.751 & 0.531 & 0.202 \end{bmatrix},$$

$$\begin{aligned} x_1(t) &= \tanh\left(10v_1(t)\right), \\ x_2(t) &= 0.1v_2(t) + v_2(t)^3, \\ x_3(t) &= \tanh\left(2v_3(t)\right) + v_3(t)^3, \\ x_4(t) &= v_4(t) + \tanh\left(7v_4(t)\right). \end{aligned} \quad (21)$$

The plots of the original speeches, linear mixtures, and PNL mixtures are shown in Figure 11. Our proposed algorithm, gpICA, was executed on these PNL mixtures with parameters $L = 151$, $\mu = 0.5$, and $\xi = 0.01$. The final outputs, y_i, are plotted in Figure 11(f), next to the outputs of SOBI and Gauss-TD. The correlation coefficients between the estimates $\hat{s}_i$ and their original signals s_i of gpICA, SOBI, and Gauss-TD are provided in Table 2.

In Figure 11, we can observe a good performance of gpICA. The linear ICA method (SOBI) was not able to separate the nonlinear mixtures. It could manage to estimate only one speech signal with adequate quality (with $r_{(s_2,\hat{s}_2)} \approx 0.8$). Whereas, the gpICA was capable of estimating all the four

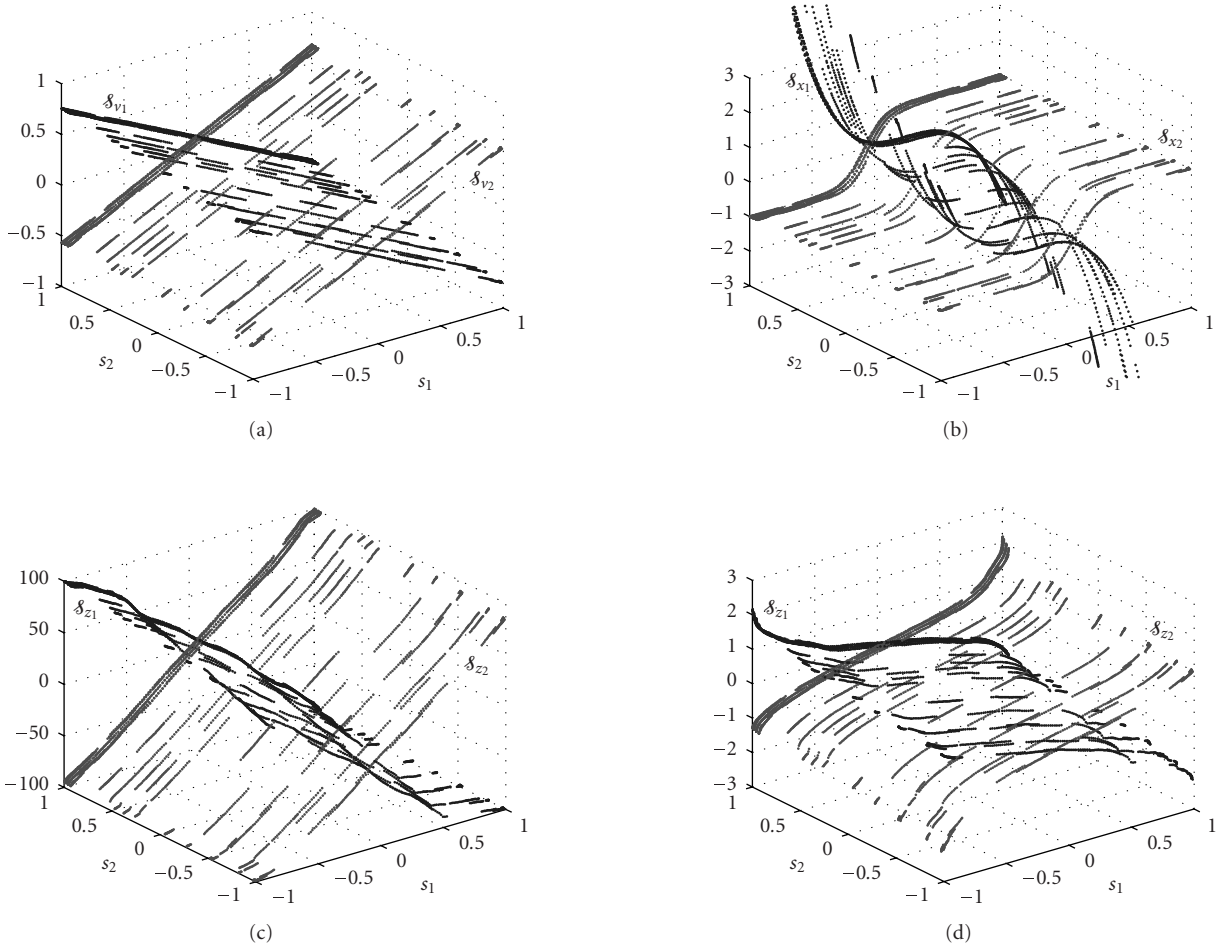

FIGURE 9: Geometric representation of signals in a 3D space. (a) Linear mixtures v_1 and v_2. (b) PNL mixtures x_1 and x_2. (c) gpICA's linearized signals z_1 and z_2. (d) Linearized signals using Gauss-TD.

TABLE 1: Experiment 1: mixture of two sinusoidal signals—correlation coefficient between the original sources and their estimates.

Method	$r_{(s_1,\hat{s}_1)}$	$r_{(s_2,\hat{s}_2)}$
SOBI	0.887	0.582
Gauss-TD	−0.982	0.888
gpICA	−0.999	−0.966

TABLE 2: Experiment 3: mixture of four speech signals—correlation coefficient between the original sources and their estimates.

Method	$r_{(s_1,\hat{s}_1)}$	$r_{(s_2,\hat{s}_2)}$	$r_{(s_3,\hat{s}_3)}$	$r_{(s_4,\hat{s}_4)}$
SOBI	−0.404	0.816	0.590	0.695
Gauss-TD	−0.770	0.984	−0.956	0.611
gpICA	−0.724	0.946	−0.875	-0.926

speech signals effectively. Our proposed algorithm continued to provide good performance with high-quality estimated signals; even the poorest output of gpICA still has correlation coefficient of over 0.72. Compared with the Gauss-TD algorithm which uses additional assumption of the Gaussianity of the mixtures, gpICA provided a similar performance.

6. CONCLUSION

A geometric approach called gpICA for the post nonlinear independent component analysis has been presented in this paper. By considering the characteristics of a plane and a nonlinear surface in a multidimensional space, a simple linearizing process has been developed to transform the post nonlinear mixtures into linear mixtures without any additional assumption. From the linearized signals, the unknown sources can be estimated by any linear ICA algorithm. Throughout extensive experiments, gpICA was able to perform well in all the simulations without changing the algorithm configuration and parameters. With the two-stage separating model and a novel geometric linearizing technique, gpICA possesses the following several advantages.

(1) Besides the PNL mixture assumption, gpICA does not use any additional assumption about the original sources as well as the mixing models. For example,

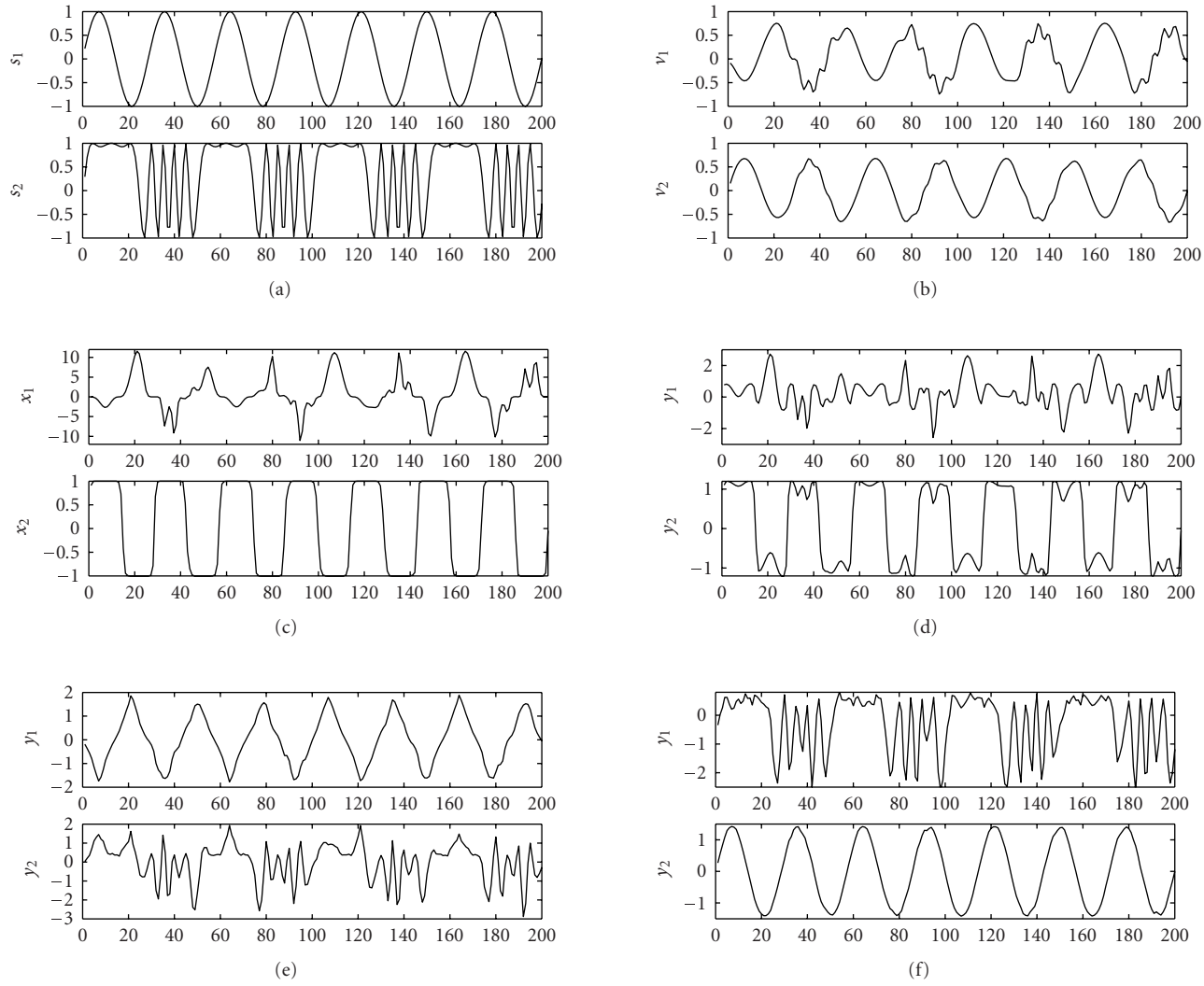

Figure 10: Plots of 200 signal samples. (a) Unknown sources. (b) Linear mixtures. (c) PNL mixtures. (d) SOBI estimates. (e) Gauss-TD estimates. (f) gpICA estimates.

unlike Gauss-TD, gpICA does not use the Gaussianity assumption about the linear mixtures.

(2) The linearization and demixing processes are independent. The users can choose any suitable linear ICA algorithm for demixing stage. Thus, it increases the performance and adaptation of gpICA in a specific environment.

(3) With a single fixed configuration, gpICA can perform the extraction effectively in various simulations with different data sets.

However, several issues still exist and require further investigations to improve the algorithm. The first issue comes from the heuristic criterion in (10). A possible solution could be another version with multiple updating points or multiple "fake planes." The convergence conditions and the criteria for selecting the algorithm's parameters are also other issues that need more study. Geometric approach to nonlinear ICA is not constrained to PNL model, an extended version of gpICA for a broader nonlinear ICA submodel is being carried out. Finally, the question of whether gpICA can be applied for optimizing the data encoding (like the LOCOCODE method [10]) is an interesting issue for the future study.

APPENDIX

PROOF OF PROPOSITION 2

Since p_1 and q_1 are the companion points, their coordinates are specified as (x_1, y_1, z_{p1}) and (x_1, y_1, z_{q1}), respectively. Similarly, p_2 and q_2 are located at (x_2, y_2, z_{p2}) and (x_2, y_2, z_{q2}), p_c and q_c are located at (x_c, y_c, z_{pc}) and (x_c, y_c, z_{qc}).

From Definition 3, as p_c lies on $\overline{p_1 p_2}$, we derive the following equation:

$$\frac{x_c - x_1}{x_2 - x_1} = \frac{y_c - y_1}{y_2 - y_1} = \frac{z_{pc} - z_{p1}}{z_{p2} - z_{p1}}. \tag{.22}$$

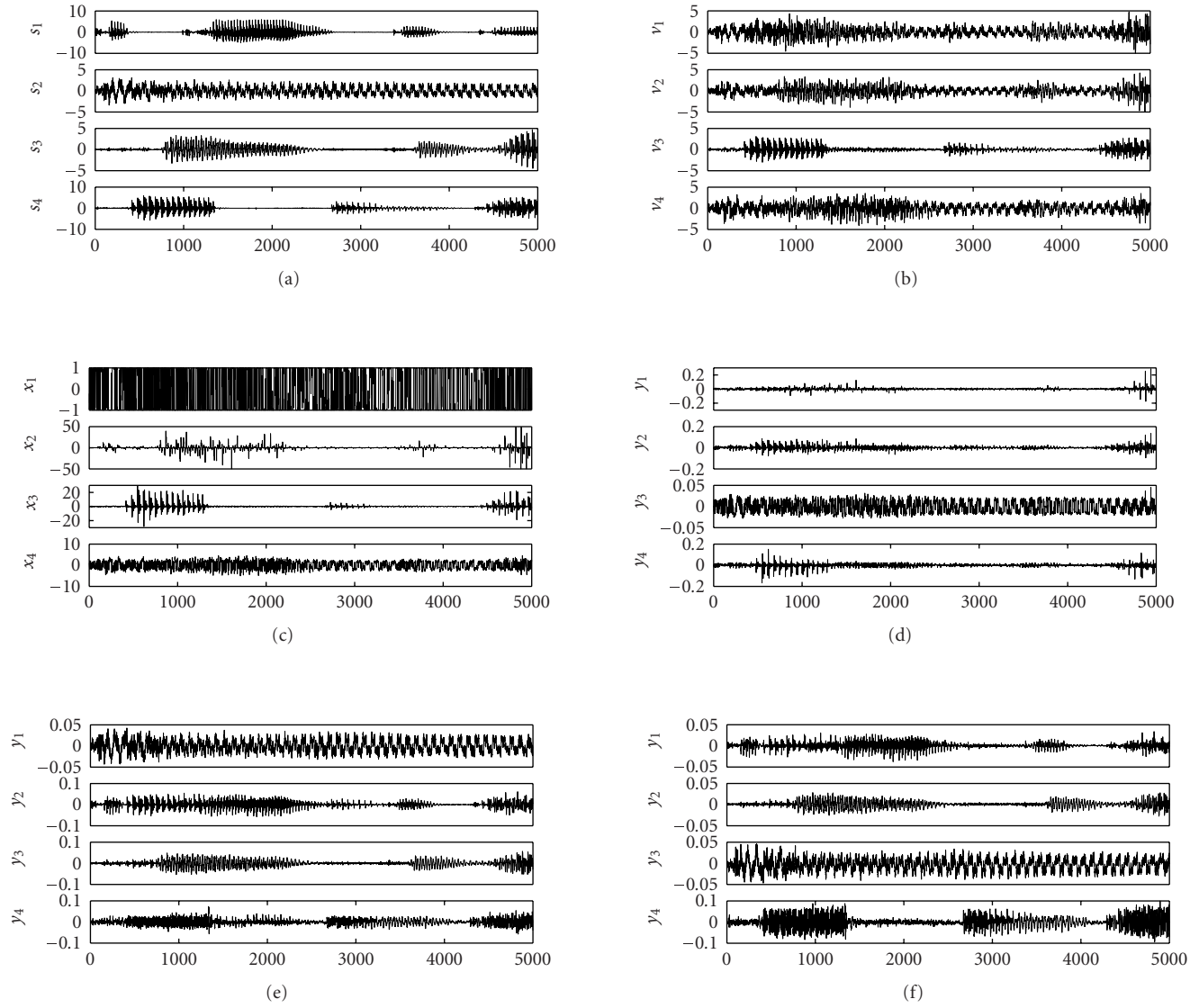

FIGURE 11: Plots of the speech signals. (a) Unknown sources. (b) Linear mixtures. (c) PNL mixtures. (d) SOBI estimates. (e) Gauss-TD estimates. (f) gpICA estimates.

Similarly, since q_c lies on $\overline{q_1 q_2}$, we have the following equation:

$$\frac{x_c - x_1}{x_2 - x_1} = \frac{y_c - y_1}{y_2 - y_1} = \frac{z_{qc} - z_{q1}}{z_{q2} - z_{q1}}. \quad (.23)$$

Combining (.22) and (.23) yields

$$\frac{z_{pc} - z_{p1}}{z_{p2} - z_{p1}} = \frac{z_{qc} - z_{q1}}{z_{q2} - z_{q1}}. \quad (.24)$$

REFERENCES

[1] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*, John Wiley & Sons, New York, NY, USA, 2001.

[2] A. Cichocki and S.-I. Amari, *Adaptive Blind Signal and Image Processing*, John Wiley & Sons, New York, NY, USA, 2002.

[3] P. Comon, "Independent component analysis, a new concept?" *Signal Processing*, vol. 36, no. 3, pp. 287–314, 1994.

[4] A. J. Bell and T. J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," *Neural Computation*, vol. 7, no. 6, pp. 1129–1159, 1995.

[5] A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, and E. Moulines, "A blind source separation technique using second-order statistics," *IEEE Transactions on Signal Processing*, vol. 45, no. 2, pp. 434–444, 1997.

[6] A. Hyvärinen, "Fast and robust fixed-point algorithms for independent component analysis," *IEEE Transactions on Neural Networks*, vol. 10, no. 3, pp. 626–634, 1999.

[7] D.-T. Pham and J.-F. Cardoso, "Blind separation of instantaneous mixtures of non stationary sources," *IEEE Transactions on Signal Processing*, vol. 49, no. 9, pp. 1837–1848, 2001.

[8] P. Pajunen, A. Hyvärinen, and J. Karhunen, "Nonlinear blind source separation by self-organizing maps," in *Proceedings of the International Conference on Neural Information Processing (ICONIP '96)*, vol. 2, pp. 1207–1210, Hong Kong, September 1996.

[9] P. Pajunen and J. Karhunen, "A maximum likelihood approach to nonlinear blind source separation," in *Proceedings of 7th International Conference on Artificial Neural Networks (ICANN '97)*, pp. 541–546, Lausanne, Switzerland, October 1997.

[10] S. Hochreiter and J. Schmidhuber, "Feature extraction through LOCOCODE," *Neural Computation*, vol. 11, no. 3, pp. 679–714, 1999.

[11] S. Hochreiter and J. Schmidhuber, "LOCOCODE performs nonlinear ica without knowing the number of sources," in *Proceedings of the 1st International Workshop on Independent Component Analysis and Signal Separation (ICA '99)*, pp. 149–154, Aussois, France, January 1999.

[12] C. Jutten and J. Karhunen, "Advances in nonlinear blind source separation," in *Proceedings of the 4th International Workshop on Independent Component Analysis and Signal Separation (ICA '03)*, pp. 245–256, Nara, Japan, April 2003.

[13] A. Hyvärinen and P. Pajunen, "Nonlinear independent component analysis: existence and uniqueness results," *Neural Networks*, vol. 12, no. 3, pp. 429–439, 1999.

[14] A. Taleb and C. Jutten, "Source separation in post-nonlinear mixtures," *IEEE Transactions on Signal Processing*, vol. 47, no. 10, pp. 2807–2820, 1999.

[15] A. Ziehe, M. Kawanabe, S. Harmeling, and K.-R. Müller, "Blind separation of post-nonlinear mixtures using linearizing transformations and temporal decorrelation," *Journal of Machine Learning Research*, vol. 4, no. 7-8, pp. 1319–1338, 2003.

[16] H. Y. Howard, S.-I. Amari, and A. Cichocki, "Information-theoretic approach to blind separation of sources in non-linear mixture," *Signal Processing*, vol. 64, no. 3, pp. 291–300, 1998.

[17] A. Parashiv-Ionescu, C. Jutten, A. Ionescu, A. Chovet, and A. Rusu, "High performance magnetic field smart sensor arrays with source separation," in *Proceedings of the 1st International Conference on Modeling and Simulation of Microsystems (MSM '98)*, pp. 666–671, Santa Clara, Calif, USA, April 1998.

[18] S. Prakriya and D. Hatzinakos, "Blind identification of LTI-ZMNLLTI nonlinear channel models," *IEEE Transactions on Signal Processing*, vol. 43, no. 12, pp. 3007–3013, 1995.

[19] M. J. Korenberg and I. W. Hunter, "The identification of nonlinear biological systems: LNL cascade models," *Biological Cybernetics*, vol. 55, no. 2-3, pp. 125–134, 1996.

[20] T. V. Nguyen, J. C. Patra, and A. Das, "A post nonlinear geometric algorithm for independent component analysis," *Digital Signal Processing*, vol. 15, no. 3, pp. 276–294, 2005.

[21] T. V. Nguyen, J. C. Patra, A. Das, and G. S. Ng, "Post nonlinear blind source separation by geometric linearization," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '05)*, vol. 1, pp. 244–249, Montreal, Canada, July-August 2005.

[22] J. Eriksson and V. Koivunen, "Blind identifiability of class of nonlinear instantaneous ica models," in *Proceedings of the 11th European Signal Processing Conference (EUSIPCO '02)*, vol. 2, pp. 7–10, Toulouse, France, September 2002.

[23] P. Georgiev and A. Chichocki, "Robust independent component analysis via time-delayed cumulant functions," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E86-A, no. 3, pp. 573–579, 2003.

[24] A. Cichocki, S.-I. Amari, K. Siwek, et al., Icalab toolboxes, 2003, http://www.bsp.brain.riken.jp/ICALAB.

Thang Viet Nguyen received his B.S. degree in computer science from the Vietnam National University, Hanoi, Vietnam, in 2000. He is currently pursuing the Ph.D. degree at the School of Computer Engineering, Nanyang Technological University, Singapore. His research interests include independent component analysis, neural networks, and genetic algorithm.

Jagdish Chandra Patra obtained the B.S. (Eng) and M.S. (Eng) degrees, both in electronics and telecommunication engineering from Sambalpur University, India, in 1978 and 1989, respectively. He received his Ph.D. degree in electronics and communication engineering from the Indian Institute of Technology, Kharagpur, in 1996. Currently he is serving as an Assistant Professor in the School of Computer Engineering, NTU, Singapore. His research interests include intelligent signal processing using neural networks, fuzzy sets, and genetic algorithms in the area of data security, sensor networks, image processing, and bioinformatics. He is a Member of IEEE (USA) and Institution of Engineers (India).

Sabu Emmanuel received his B.E. degree (electronics and communication engineering) from Regional Engineering College, Durgapur (1988), M.E. degree (electrical communication engineering) from Indian Institute of Science, Bangalore (1998), and Ph.D. degree (computer science) from National University of Singapore (2002). He is an Assistant Professor at the School of Computer Engineering, Nanyang Technological University, Singapore. His current research interests are in media processing for surveillance applications, media forensics and security, and digital rights management (DRM). He is a Member of the IEEE and has served as a member of the technical program committee of several international conferences.