Open access • Journal Article • DOI:10.1109/TNN.2006.880980

The FastICA Algorithm Revisited: Convergence Analysis — Source link

Erkki Oja, Zhijian Yuan

Institutions: Helsinki University of Technology

Published on: 01 Nov 2006 - IEEE Transactions on Neural Networks (IEEE Trans Neural Netw)

Topics: FastICA, Independent component analysis, Normalization (statistics), Local convergence and Blind signal separation

Erkki Oja and Zhijian Yuan. The FastICA Algorithm revisited: convergence analysis. *IEEE Transactions on Neural Networks.* pp.1370-1381, vol. 17:6, 2006.

The FastICA Algorithm Revisited: Convergence Analysis

Erkki Oja, *Fellow, IEEE*, and Zhijian Yuan

***Abstract*—The fast independent component analysis (FastICA) algorithm is one of the most popular methods to solve problems in ICA and blind source separation. It has been shown experimentally that it outperforms most of the commonly used ICA algorithms in convergence speed. A rigorous local convergence analysis has been presented only for the so-called one-unit case, in which just one of the rows of the separating matrix is considered. However, in the FastICA algorithm, there is also an explicit normalization step, and it may be questioned whether the extra rotation caused by the normalization will affect the convergence speed. The purpose of this paper is to show that this is not the case and the good convergence properties of the one-unit case are also shared by the full algorithm with symmetrical normalization. A local convergence analysis is given for the general case, and the global behavior is illustrated numerically for two sources and two mixtures in several typical cases.**

***Index Terms*—Convergence analysis, cubic convergence, FastICA, independent component analysis (ICA).**

I. Introduction: ICA and FastICA

THE fast independent component analysis (FastICA) algorithm [11]–[13] is one of the most popular methods to solve problems in ICA and blind source separation. FastICA was originally introduced for the instantaneous noise-free ICA model. The problem is to estimate the unknown mixing matrix $\mathbf{A}$ and n unknown independent sources s_i making up a random vector $\mathbf{s} = (s_1 \dots s_n)^T$, from the model

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

Vector $\mathbf{x}$ contains the mixtures and a sample of $\mathbf{x}$ is available. It can be assumed that the sources are zero mean by first centering $\mathbf{x}$ to zero mean. It can also be assumed that the variances of the sources are equal to one, because both $\mathbf{A}$ and $\mathbf{s}$ are unknown and any scalar magnitudes can be exchanged between them. For identification of the model, at most one of the sources can have Gaussian density.

In the basic FastICA method, the vector $\mathbf{x}$ is first whitened to obtain vector $\mathbf{z} = \mathbf{V}\mathbf{x}$ for which $\mathrm{E}\{\mathbf{z}\mathbf{z}^T\} = \mathbf{I}$. This can be accomplished with principal component analysis, in practice using the available sample on vector $\mathbf{x}$. At the same time, the dimension of $\mathbf{z}$ is made equal to $n = \dim(\mathbf{s})$. In the ideal case of (1), n is found simply as the number of nonzero eigenvalues of the covariance matrix of $\mathbf{x}$. Then, in the model

$$\mathbf{z} = \mathbf{V}\mathbf{x} = \mathbf{V}\mathbf{A}\mathbf{s} \tag{2}$$

both $\mathbf{z}$ and $\mathbf{s}$ are white; $\mathbf{z}$ because of the explicit whitening, and $\mathbf{s}$ by the assumptions of zero mean, unit variance, and independence of the sources s_i. It is easy to show that in this case the $n \times n$ square matrix $\mathbf{V}\mathbf{A}$ is orthogonal, or $(\mathbf{V}\mathbf{A})^T(\mathbf{V}\mathbf{A}) = \mathbf{I}$, which simplifies the estimation problem. If $\mathbf{A}$ were known, the sources could be directly solved from $\mathbf{s} = (\mathbf{V}\mathbf{A})^T\mathbf{z}$.

In the FastICA algorithm, the solution is sought as

$$\mathbf{y} = \mathbf{W}\mathbf{z} \tag{3}$$

with matrix $\mathbf{W}$ square and orthogonal. Vector $\mathbf{y}$ is thus a rotation of the solution $\mathbf{s}$ and has a unit (identity) covariance matrix. If $\mathbf{W}$ is equal to $(\mathbf{V}\mathbf{A})^T$, then $\mathbf{y} = \mathbf{s}$ and the independent components have been found. If $\mathbf{P}$ is any permutation matrix and $\mathbf{W}$ is equal to $\mathbf{P}(\mathbf{V}\mathbf{A})^T$, also an orthogonal matrix, then $\mathbf{y} = \mathbf{P}\mathbf{s}$. Also, this permuted version of the sources s_i is a feasible solution, because we have no way of knowing the order of the independent components.

The separation matrix $\mathbf{W}$ is found by numerical algorithms. Let us denote the rows of $\mathbf{W}$ by $\mathbf{w}_i^T, i = 1, \dots, n$. The *FastICA algorithm* [12] is an iterative method to find the local maxima of a cost function

$$\mathcal{J}_G = \sum_{i=1}^{n} \mathrm{E}\left\{G\left(\mathbf{w}_i^T\mathbf{z}\right)\right\} \tag{4}$$

with G a nonlinear function which is usually assumed even and symmetrical. The symbol E stands for expectation, which in practice would be estimated by sample mean over the whitened vectors $\mathbf{z}$. The cost function to be maximized can be negative mutual information, likelihood, some approximation of nongaussianity such as higher order cumulants, or some extension of these. For a very wide range of nonlinearities G, it can be shown that the true independent components, or rows of the true solution matrix $\mathbf{W}$, are among the local maxima; see [12]. For self-adapting nonlinearity, see [25]. A widely used cost function is the fourth-order cumulant or kurtosis, defined for any random variable τ as

$$\mathrm{kurt}(\tau) = \mathrm{E}\{\tau^4\} - 3(\mathrm{E}\{\tau^2\})^2. \tag{5}$$

In (4), the argument $\mathbf{w}_i^T\mathbf{z} = y_i$ is restricted to have unit variance, and thus its kurtosis is $\mathrm{E}\{y_i^4\} - 3$. In maximization, the second term can be dropped, and the criterion becomes

$$\mathcal{J}_G^{\mathrm{kurt}} = \sum_{i=1}^{n} \mathrm{E}\left\{\left(\mathbf{w}_i^T\mathbf{z}\right)^4\right\}. \tag{6}$$

Kurtosis maximization was the starting point for the FastICA algorithm [13], [14]. This is a very well-known and established

Manuscript received February 3, 2005; revised February 17, 2006.

The authors are with the Adaptive Informatics Research Centre, Helsinki University of Technology, 02015 HUT, Finland (e-mail: erkki.oja@hut.fi; zhijian.yuan@hut.fi).

Digital Object Identifier 10.1109/TNN.2006.880980

cost function in ICA, blind source separation, and blind deconvolution, dating back to the classical works of Donoho [6], Lacoume [17], and Comon [4]. Different maximization techniques have been used such as gradient methods [23], Givens rotations [5], or tensor algebraic techniques [2]. In [13] and [14], we introduced a faster method based on batch processing and fixed-point analysis. For one of the rows $\mathbf{w}_i^T$, the FastICA algorithm for kurtosis maximization makes the basic updating step

$$\bar{\mathbf{w}}_i = \mathrm{E}\left\{\mathbf{z}\left(\mathbf{w}_i^T\mathbf{z}\right)^3\right\} - 3\mathbf{w}_i \tag{7}$$

followed by normalization of vector $\bar{\mathbf{w}}_i$ to unit norm. This form was introduced in [13], where it was also shown that the algorithm will converge globally to one of the original sources. Note that the first term in (7) is just the gradient of (6), and the second term has been designed to give the algorithm superior convergence over a gradient method. Any constant multipliers can be omitted because of the subsequent normalization.

For the general cost function (4), the corresponding updating step is

$$\bar{\mathbf{w}}_i = \mathrm{E}\left\{\mathbf{z}g\left(\mathbf{w}_i^T\mathbf{z}\right)\right\} - \mathrm{E}\left\{g'\left(\mathbf{w}_i^T\mathbf{z}\right)\right\}\mathbf{w}_i \tag{8}$$

with function g the derivative of G and g' the derivative of g. This form of the algorithm was introduced by Hyvärinen [11] as an approximative Newton method for maximizing the cost function $\mathcal{J}_G$ in (4). The first term again is just the gradient of $\mathcal{J}_G$ with respect to $\mathbf{w}_i$, and the second term is due to the Newton approximation. Note that for $G(y) = y^4$ as in (6), we have $g(y) = 4y^3, g'(y) = 12y^2$, giving the rule (7) when the multiplier four is dropped. In addition to the cubic polynomial y^3, some popular nonlinearities used in FastICA are $g(y) = \tanh(y)$ and $g(y) = y\exp(-y^2/2)$. Note that these all are *odd* functions: $g(-y) = -g(y)$.

To compute the full matrix $\mathbf{W}$, the vectors $\bar{\mathbf{w}}_i$ must be reorthonormalized after the update because they lose their orthonormality in the updating (8). The orthonormalization can be accomplished basically in two ways: either deflationary or symmetrical orthonormalization [13]. The latter is given by

$$\mathbf{W} = (\bar{\mathbf{W}}\bar{\mathbf{W}}^T)^{-1/2}\bar{\mathbf{W}} \tag{9}$$

with $\bar{\mathbf{W}}$ the matrix with rows $\bar{\mathbf{w}}_i^T$. The square root of a symmetrical positive definite matrix is here defined as the principal square root with positive eigenvalues, which can be computed through the eigenvector–eigenvalue decomposition.

Clearly, after the normalization, it holds $\mathbf{W}\mathbf{W}^T = \mathbf{I}$ or $\mathbf{w}_i^T\mathbf{w}_j = \delta_{ij}$. In FastICA, the updating step (8) for each $\mathbf{w}_i$ in parallel followed by normalization is repeated until the matrix $\mathbf{W}$ converges. The initial matrix $\mathbf{W}$ can be chosen as an arbitrary orthogonal matrix.

A good reference to the FastICA method is the textbook [12] which also contains many pointers to practical applications and public-domain software for the method.

The key question is the *rate and order of convergence* for this algorithm. This is defined in the standard way as follows: Assume $\mathbf{W}^*$ is a *fixed point*, meaning that if it is chosen as the initial matrix, then it will not be changed in the algorithm. Assume that at a certain step in the algorithm, $\mathbf{W} = \mathbf{W}^* + \mathbf{E}_1$ and at the next step, $\mathbf{W} = \mathbf{W}^* + \mathbf{E}_2$. Then the convergence to $\mathbf{W}^*$ is *linear with convergence rate* k if $\|\mathbf{E}_2\|/\|\mathbf{E}_1\|$ is asymptotically (after a large number of iteration steps) equal to k. There $\|\cdot\|$ is any matrix norm. Convergence is *quadratic*, or the order of convergence is two, if $\|\mathbf{E}_2\|$ is asymptotically proportional to $\|\mathbf{E}_1\|^2$, and *cubic* (order of convergence is three) if $\|\mathbf{E}_2\|$ is asymptotically proportional to $\|\mathbf{E}_1\|^3$. As soon as $\|\mathbf{E}_1\|$ becomes small, quadratic, or cubic, convergence makes the error very small in a few steps.

In an empirical comparison study given by [8], FastICA turned out to live up to its name, as it outperformed some popular gradient-descent type ICA methods in convergence speed by a clear margin, with small residual error. Many other comparisons indicate the same. This is understandable because FastICA is not gradient descent but an approximative Newton method. However, despite the importance of the speed of convergence, it has been only partially answered in the literature so far.

Theoretical analysis of convergence can be approached from two directions.

1) The numerical convergence of the ideal deterministic algorithms. These are obtained when the expectations in (7) and (8) are assumed to be the theoretically correct ones, essentially meaning that there is an infinite sample of the mixture vectors $\mathbf{x}$, hence, the whitened vectors $\mathbf{z}$, in the algorithms. Questions such as asymptotic stability of the extrema of the theoretical contrast functions, and the convergence rate of the algorithms, can be discussed.
2) Behavior of the algorithms for finite samples. This is the practical situation. Then, the theoretical expectations are replaced by sample averages. Numerical convergence speed is still quite as important as for the ideal case, but now the limit of convergence is not exactly the correct solution obtained by the ideal algorithm. There is a residual error due to the finite sample size. A classical measure of error is the variance of the matrix elements. The question then is how this residual error due to a finite sample converges to zero with growing sample size, or the consistency of the estimation.

In this paper, we only address the first of these questions. The second question has been discussed elsewhere [10], [12], [15], [16], [24]. In the three latter references, the Cramer–Rao bound for linear ICA was derived, the residual error of FastICA was compared to it with favorable results, and a new efficient version of FastICA was proposed.

The first question has been partially addressed earlier, but up to now, a full analysis has been lacking. The convergence was proven in [11] and [13] for the so-called *one-unit case*, in which only one of the rows is considered and orthogonalization is reduced to just normalization of the vector to unit length after each iteration step. The one-unit algorithm has local *quadratic* or *cubic* convergence to one of the rows of the true separating matrix, under a mild condition on the nonlinear function g. It has also been shown that in the specific case of the kurtosis cost function, when $g(\tau) = \tau^3$, there are no other asymptotically stable points, so the convergence of the one-unit algorithm is

global, and the order of convergence is cubic. An analysis of the convergence for this case was also made by Douglas [7]. For a gradient algorithm and a generic cost function, monotonic convergence was analyzed by Regalia and Kofidis [22]. For stochastic gradient descent ICA algorithms in general, stability was considered by Amari *et al.* [1].

However, rigorous analysis of convergence for the full algorithm with either deflationary or symmetrical orthonormalization is missing. For the deflationary case, the convergence follows in a straightforward fashion from the convergence of the one-unit case because the vectors $\mathbf{w}_i$ can be analyzed one by one. However, in the symmetrical case, *all* the vectors $\bar{\mathbf{w}}_1, \ldots, \bar{\mathbf{w}}_n$ are rotated strongly in step (9) and there is a doubt that this rotation might destroy the convergence speed. We cannot infer the convergence speed directly from the one-unit behavior.

It is the purpose of this paper to show that this is not the case: The algorithm with symmetrical orthonormalization still has the good local convergence properties of the one-unit case. Throughout, the approach here is purely theoretical, giving rigorous mathematical results. No source separation experiments are given, because a multitude of such results have been presented in ICA literature for artificial and real data, both for the FastICA algorithm and other comparable methods; see e.g., the books [3], [9], [12], and references therein.

The contents of this paper are as follows. Section II considers the general cost function (4) and shows local convergence to the true separating matrix for the symmetrical FastICA algorithm. The convergence speed is at least quadratic as usually in the Newton method. Section III addresses the special case of the kurtosis cost function (6) and shows cubic convergence, following a preliminary result of this case earlier presented in [21]. In Section IV, a more detailed convergence analysis is made in the simple 2×2 case for various nonlinearities G in (4), illustrating why the algorithm is so fast. Section V gives some conclusions.

II. Convergence Analysis for the General Nonlinearity

A. Useful Transformation

Let us make a simplifying linear transformation by considering the matrices

$$\mathbf{U} = \mathbf{W}(\mathbf{VA}) \quad \bar{\mathbf{U}} = \bar{\mathbf{W}}(\mathbf{VA}). \tag{10}$$

Then, from (2), we have $\mathbf{Wz} = \mathbf{W}(\mathbf{VA})\mathbf{s} = \mathbf{Us}$. Denoting the rows of matrix $\mathbf{U}$ by $\mathbf{u}_i^T$, we have $\mathbf{w}_i^T\mathbf{z} = \mathbf{u}_i^T\mathbf{s}$. Multiplying both sides of the general one-unit FastICA algorithm (8) from the left by $(\mathbf{VA})^T$ and remembering that $\mathbf{s} = (\mathbf{VA})^T\mathbf{z}$ yields now

$$\bar{\mathbf{u}}_i = \mathrm{E}\{\mathbf{s}g(\mathbf{u}_i^T\mathbf{s})\} - \mathrm{E}\{g'(\mathbf{u}_i^T\mathbf{s})\}\mathbf{u}_i. \tag{11}$$

This form of the equation is much easier to analyze than (8) because now the independent source vector $\mathbf{s}$ appears explicitly and we can make use of the independence of its elements.

Let us show next that the normalization (9) is uneffected by this transformation. Multiplying both sides of (9) from the right by $(\mathbf{VA})$ we get

$$\mathbf{W}(\mathbf{VA}) = (\bar{\mathbf{W}}(\mathbf{VA})(\mathbf{VA})^T\bar{\mathbf{W}}^T)^{-1/2}\bar{\mathbf{W}}(\mathbf{VA})$$

where we have used the fact that $\mathbf{VA}(\mathbf{VA})^T = \mathbf{I}$. We have

$$\mathbf{U} = (\bar{\mathbf{U}}\bar{\mathbf{U}}^T)^{-1/2}\bar{\mathbf{U}} \tag{12}$$

giving the equivalent orthonormalization for matrix $\mathbf{U}$. The general symmetrical FastICA algorithm in the transformed coordinate system consists of (11), for $i = 1, \ldots, n$, together with the normalization (12).

B. Fixed Points

The problem is now the convergence of this algorithm: Where does it converge and what is the convergence speed?

For the one-unit convergence result, the following assumptions on function $g(\cdot)$ are made which also hold here [12].

Assumption 1: Function $g(\cdot)$ is an odd, twice differentiable function, which satisfies

$$\mathrm{E}\{s_i g(s_i) - g'(s_i)\} \neq 0 \tag{13}$$

for all sources $s_i, i = 1, \ldots, n$.

$g(\cdot)$ is odd because it is the derivative of the even function $G(\cdot)$ in the general cost function (4). Differentiability is a technical assumption that will be needed in the stability analysis. The condition (13) is quite general. However, it is violated in two special cases, as is easily shown: Either if $g(\cdot)$ is a linear function, or if s_i has a Gaussian density. It is well known that in these cases, the ICA model cannot be identified. Linearity of $g(\cdot)$ would mean that the function $G(\cdot)$ in the cost function (4) is quadratic. The ICA problem cannot be solved with second-order statistics. As for Gaussianity, in the practical FastICA algorithm finite samples are always used and all the empirical distributions are non-Gaussian.

To analyze the iteration, let us first look at its *fixed points*. We have the following.

Lemma 1: Under Assumption 1, matrix $\mathbf{U} = \mathrm{diag}(\pm 1, \ldots, \pm 1)$ is a fixed point of (11) and (12).

Proof: Now $\mathbf{u}_i^T = (0, \ldots, \pm 1, \ldots, 0)$ is the ith row of the matrix $\mathbf{U}$, thus $\mathbf{u}_i^T\mathbf{s} = \pm s_i$. Hence, $g(\mathbf{u}_i^T\mathbf{s}) = g(\pm s_i)$ is a function of variable s_i only, which is independent of other components $s_j, j \neq i$; so $\mathrm{E}\{s_j g(\mathbf{u}_i^T\mathbf{s})\} = 0$ for $i \neq j$. Therefore, from (11)

$$\begin{aligned}\bar{\mathbf{u}}_i &= \mathrm{E}\left\{\mathbf{s}g\left(\mathbf{u}_i^T\mathbf{s}\right)\right\} - \mathrm{E}\left\{g'\left(\mathbf{u}_i^T\mathbf{s}\right)\right\}\mathbf{u}_i \\ &= (0, \ldots, \mathrm{E}\{s_i g(\pm s_i)\}, \ldots, 0)^T - \mathrm{E}\{g'(\pm s_i)\}\mathbf{u}_i.\end{aligned}$$

Because $g(\cdot)$ is odd and hence $g'(\cdot)$ is even, we have $\mathrm{E}\{s_i g(\pm s_i)\} = \pm\mathrm{E}\{s_i g(s_i)\}$ and $\mathrm{E}\{g'(\pm s_i)\} = \mathrm{E}\{g'(s_i)\}$. Consider first the plus sign in $\mathbf{u}_i$ (the one nonzero element is equal to $+1$). Then, we have

$$\begin{aligned}\bar{\mathbf{u}}_i &= (0, \ldots, \mathrm{E}\{s_i g(s_i)\}, \ldots, 0)^T - \mathrm{E}\{g'(s_i)\}\mathbf{u}_i \\ &= c_i\mathbf{u}_i\end{aligned}$$

where $c_i = \mathrm{E}\{s_i g(s_i)\} - \mathrm{E}\{g'(s_i)\}$. Consider then the minus sign in $\mathbf{u}_i$. Then, we have

$$\begin{aligned}\bar{\mathbf{u}}_i &= (0,\ldots,-\mathrm{E}\{s_i g(s_i)\},\ldots,0)^T - \mathrm{E}\{g'(s_i)\}\mathbf{u}_i \\ &= c_i\mathbf{u}_i.\end{aligned}$$

Thus, in both plus and minus cases, we have $\bar{\mathbf{u}}_i = c_i\mathbf{u}_i$, in matrix form

$$\bar{\mathbf{U}} = \mathbf{K}\mathbf{U} \tag{14}$$

where $\mathbf{K} = \mathrm{diag}(c_1,\ldots,c_n)$. By Assumption 1, all the c_i are nonzero. We have

$$\bar{\mathbf{U}}\bar{\mathbf{U}}^T = \mathbf{K}\mathbf{U}\mathbf{U}^T\mathbf{K} = \mathbf{K}^2 \tag{15}$$

$$(\bar{\mathbf{U}}\bar{\mathbf{U}}^T)^{-1/2} = \mathbf{K}^{-1} \tag{16}$$

and hence

$$(\bar{\mathbf{U}}\bar{\mathbf{U}}^T)^{-1/2}\bar{\mathbf{U}} = \mathbf{K}^{-1}\mathbf{K}\mathbf{U} = \mathbf{U}. \tag{17}$$

This completes the proof.

Remark: The result also follows from the convergence result for the one-unit algorithm in a straightforward manner. However, for other fixed points, not necessarily shared by the one-unit algorithm, the analysis must be done separately for the symmetric algorithm, as was done here (see also Lemma 4 in Section III).

The significance of Lemma 1 comes from noting that, by (3), $y_i = \mathbf{w}_i^T\mathbf{z} = \mathbf{u}_i^T\mathbf{s} = \pm s_i$, meaning that the original sources have been found at the fixed point. They have a sign ambiguity which cannot be resolved with ICA techniques.

Because the ordering of the sources should have no effect, it would be desirable that also a permuted version would be a fixed point. This is shown in the following.

Lemma 2: Let $\mathbf{U}$ be the fixed point given in Lemma 1. Let $\mathbf{P}$ be an orthogonal permutation matrix for which $\mathbf{P}\mathbf{P}^T = \mathbf{I}$. Then, $\mathbf{P}\mathbf{U}$ and $\mathbf{U}\mathbf{P}$ are fixed points as well.

Proof: Both permuted matrices $\mathbf{P}\mathbf{U}$ and $\mathbf{U}\mathbf{P}$, with $\mathbf{U}$ given in Lemma 1, contain exactly one nonzero element on each row. Equation (14) can be shown to hold for them as well, when $\mathbf{U}$ and $\bar{\mathbf{U}}$ are replaced by the permuted versions and the diagonal elements of $\mathbf{K}$ are permuted correspondingly.

Consider then the symmetrical normalization in (12). Let us first show that the same relation holds between $\bar{\mathbf{U}}\mathbf{P}$ and $\mathbf{U}\mathbf{P}$. We have

$$[\bar{\mathbf{U}}\mathbf{P}(\bar{\mathbf{U}}\mathbf{P})^T]^{-1/2}\bar{\mathbf{U}}\mathbf{P} = (\bar{\mathbf{U}}\bar{\mathbf{U}}^T)^{-1/2}\bar{\mathbf{U}}\mathbf{P} = \mathbf{U}\mathbf{P}$$

which concludes the proof for $\mathbf{U}\mathbf{P}$.

For matrices $\mathbf{P}\bar{\mathbf{U}}$ and $\mathbf{P}\mathbf{U}$, we must show

$$[\mathbf{P}\bar{\mathbf{U}}(\mathbf{P}\bar{\mathbf{U}})^T]^{-1/2}\mathbf{P}\bar{\mathbf{U}} = \mathbf{P}\mathbf{U}.$$

We have now

$$\begin{aligned}[\mathbf{P}\bar{\mathbf{U}}(\mathbf{P}\bar{\mathbf{U}})^T]^{-1} &= [\mathbf{P}\bar{\mathbf{U}}\bar{\mathbf{U}}^T\mathbf{P}^T]^{-1} \\ &= \mathbf{P}(\bar{\mathbf{U}}\bar{\mathbf{U}}^T)^{-1}\mathbf{P}^T \\ &= \mathbf{P}(\bar{\mathbf{U}}\bar{\mathbf{U}}^T)^{-1/2}\mathbf{P}^T\mathbf{P}(\bar{\mathbf{U}}\bar{\mathbf{U}}^T)^{-1/2}\mathbf{P}^T\end{aligned}$$

hence, $[\mathbf{P}\bar{\mathbf{U}}(\mathbf{P}\bar{\mathbf{U}})^T]^{-1/2} = \mathbf{P}(\bar{\mathbf{U}}\bar{\mathbf{U}}^T)^{-1/2}\mathbf{P}^T$. Thus, finally

$$\begin{aligned}[\mathbf{P}\bar{\mathbf{U}}(\mathbf{P}\bar{\mathbf{U}})^T]^{-1/2}\mathbf{P}\bar{\mathbf{U}} &= \mathbf{P}(\bar{\mathbf{U}}\bar{\mathbf{U}}^T)^{-1/2}\mathbf{P}^T\mathbf{P}\bar{\mathbf{U}} \\ &= \mathbf{P}(\bar{\mathbf{U}}\bar{\mathbf{U}}^T)^{-1/2}\bar{\mathbf{U}} = \mathbf{P}\mathbf{U}.\end{aligned} \tag{18}$$

This concludes the proof.

C. Stability Analysis for the Fixed Points

Now we will show that the following holds.

Lemma 3: Under Assumption 1, the fixed point of Lemma 1 is asymptotically stable and the order of convergence is at least two.

Proof: For clarity, let us denote the fixed point $\mathrm{diag}(\pm 1,\ldots,\pm 1)$ of Lemma 1 by $\mathbf{D}$. Make the perturbation

$$u_{ij} = d_{ij} + \epsilon_{ij} \tag{19}$$

where d_{ij} are the elements of matrix $\mathbf{D}$, and ϵ_{ij} are small for all i,j, say, $|\epsilon_{ij}| \le \epsilon$ with ϵ a small positive number. Denote the rows of matrix $\mathbf{D}$ by $\mathbf{d}_i^T$ and those of the (ϵ_{ij}) matrix by $\mathbf{e}_i^T$, respectively, thus $\mathbf{u}_i = \mathbf{d}_i + \mathbf{e}_i$. Equation (11) gives

$$\begin{aligned}\bar{\mathbf{u}}_i &= \mathrm{E}\left\{\mathbf{s}g\left(\mathbf{u}_i^T\mathbf{s}\right)\right\} - \mathrm{E}\left\{g'\left(\mathbf{u}_i^T\mathbf{s}\right)\right\}\mathbf{u}_i && (20)\\ &= \mathrm{E}\left\{\mathbf{s}g\left(\mathbf{d}_i^T\mathbf{s} + \mathbf{e}_i^T\mathbf{s}\right)\right\} \\ &\quad - \mathrm{E}\left\{g'\left(\mathbf{d}_i^T\mathbf{s} + \mathbf{e}_i^T\mathbf{s}\right)\right\}(\mathbf{d}_i + \mathbf{e}_i) && (21)\\ &= \mathrm{E}\left\{\mathbf{s}g\left(\pm s_i + \mathbf{e}_i^T\mathbf{s}\right)\right\} \\ &\quad - \mathrm{E}\left\{g'\left(\pm s_i + \mathbf{e}_i^T\mathbf{s}\right)\right\}(\mathbf{d}_i + \mathbf{e}_i). && (22)\end{aligned}$$

Using Taylor series expansions of the functions $g(\tau)$ and $g'(\tau)$ at the points $\tau = \pm s_i$, and remembering that $g(\cdot)$ is an odd function, we get

$$\begin{aligned}g\left(\pm s_i + \mathbf{e}_i^T\mathbf{s}\right) &= \pm g(s_i) + g'(s_i)\mathbf{e}_i^T\mathbf{s} + \mathcal{O}(\epsilon^2) && (23)\\ g'\left(\pm s_i + \mathbf{e}_i^T\mathbf{s}\right) &= g'(s_i) \pm g''(s_i)\mathbf{e}_i^T\mathbf{s} + \mathcal{O}(\epsilon^2). && (24)\end{aligned}$$

The terms $\mathcal{O}(\epsilon^2)$ collect all the remaining terms that are quadratic or of higher order in the elements of $\mathbf{e}_i$, thus bounded by a constant times ϵ^2. Also, in the rest of the proof, this standard practice is used. Substituting the aforementioned into (22) yields

$$\begin{aligned}\bar{\mathbf{u}}_i = \pm\mathrm{E}\{\mathbf{s}g(s_i)\} - \mathrm{E}\{g'(s_i)\}\mathbf{d}_i + \mathrm{E}\left\{\mathbf{s}g'(s_i)\mathbf{e}_i^T\mathbf{s}\right\} \\ \mp\mathrm{E}\left\{g''(s_i)\mathbf{e}_i^T\mathbf{s}\right\}\mathbf{d}_i - \mathrm{E}\{g'(s_i)\}\mathbf{e}_i + \mathcal{O}(\epsilon^2).\end{aligned}$$

Since

$$\begin{aligned}&\mathrm{E}\left\{\mathbf{s}g'(s_i)\mathbf{e}_i^T\mathbf{s}\right\} \mp \mathrm{E}\left\{g''(s_i)\mathbf{e}_i^T\mathbf{s}\right\}\mathbf{d}_i - \mathrm{E}\{g'(s_i)\}\mathbf{e}_i \\ &\quad = -\left(\pm\mathrm{E}\{g''(s_i)s_i\} \pm \mathrm{E}\{g'(s_i)\} \mp \mathrm{E}\left\{g'(s_i)s_i^2\right\}\right)\epsilon_{ii}\mathbf{d}_i\end{aligned}$$

we have

$$\begin{aligned}\bar{\mathbf{u}}_i = \pm\mathrm{E}\{\mathbf{s}g(s_i)\} - \mathrm{E}\{g'(s_i)\}\mathbf{d}_i - \left(\pm\mathrm{E}\{g''(s_i)s_i\}\right. \\ \left.\pm\mathrm{E}\{g'(s_i)\} \mp \mathrm{E}\left\{g'(s_i)s_i^2\right\}\right)\epsilon_{ii}\mathbf{d}_i + \mathcal{O}(\epsilon^2).\end{aligned} \tag{25}$$

In matrix form, this becomes

$$\bar{\mathbf{U}} = \mathbf{K}\mathbf{D} - \hat{\mathbf{K}}\mathbf{D} + \mathcal{O}(\epsilon^2) \tag{26}$$

where the matrix $\mathbf{K}$ is the diagonal matrix of (14), and matrix $\hat{\mathbf{K}}$ is another diagonal matrix whose diagonal elements are $(\pm\mathrm{E}\{g''(s_i)s_i\} \pm \mathrm{E}\{g'(s_i)\} \mp \mathrm{E}\{g'(s_i)s_i^2\})\epsilon_{ii}$. This gives

$$\begin{aligned}\bar{\mathbf{U}}\bar{\mathbf{U}}^T &= (\mathbf{K}-\hat{\mathbf{K}})\mathbf{D}^2(\mathbf{K}-\hat{\mathbf{K}})^T + \mathcal{O}(\epsilon^2)\\ &= \mathbf{K}^2 - 2\hat{\mathbf{K}}\mathbf{K} + \mathcal{O}(\epsilon^2)\\ &= \mathbf{K}^2\left(\mathbf{I} - 2\hat{\mathbf{K}}\mathbf{K}^{-1}\right) + \mathcal{O}(\epsilon^2)\end{aligned}$$

where we have used the diagonality of all the matrices and the fact that $\mathbf{D}^2$ is the unit (identity) matrix. Matrix $-2\hat{\mathbf{K}}\mathbf{K}^{-1}$ is first order in ϵ, thus

$$(\bar{\mathbf{U}}\bar{\mathbf{U}}^T)^{-1/2} = \mathbf{K}^{-1}(\mathbf{I} + \hat{\mathbf{K}}\mathbf{K}^{-1}) + \mathcal{O}(\epsilon^2). \tag{27}$$

Thus

$$\begin{aligned}&(\bar{\mathbf{U}}\bar{\mathbf{U}}^T)^{-1/2}\bar{\mathbf{U}}\\ &\quad= \mathbf{K}^{-1}(\mathbf{I} + \hat{\mathbf{K}}\mathbf{K}^{-1})(\mathbf{K}-\hat{\mathbf{K}})\mathbf{D} + \mathcal{O}(\epsilon^2)\\ &\quad= (\mathbf{I} + \mathbf{K}^{-1}\hat{\mathbf{K}} - \mathbf{K}^{-1}\hat{\mathbf{K}} + \mathbf{K}^{-2}\hat{\mathbf{K}}^2)\mathbf{D} + \mathcal{O}(\epsilon^2)\\ &\quad= \mathbf{D} + \mathcal{O}(\epsilon^2).\end{aligned}$$

This proves the stability of the fixed point $\mathbf{D}$ with second-order convergence.

The proof could be extended to the permuted fixed points in a straightforward way. As for other fixed points, which are possible depending on the nonlinearity g, general results of convergence or nonconvergence are difficult, as it also depends on the densities of the independent sources s_i. An exception is the case of the kurtosis cost function, as well as other polynomial nonlinearities, for which all fixed points may be characterized and their stability behavior confirmed. In the following, this is done for the kurtosis cost function.

III. Convergence Analysis for the Kurtosis Cost Function

This case was already considered in [13] for one of the rows of matrix $\mathbf{U}$. It was shown that for the jth element of $\mathbf{u}_i^T$, denoted here u_{ij}, (11) yields the very simple update rule

$$\bar{u}_{ij} = \kappa_j u_{ij}^3 \tag{28}$$

where κ_j is the kurtosis of the jth independent component. We may assume that all the kurtoses are nonzero. This follows from Assumption 1 by substituting $g(s_i) = s_i^3$: Then, $\mathrm{E}\{s_i g(s_i)\} - \mathrm{E}\{g'(s_i)\} = \mathrm{E}\{s_i^4\} - 3$ which is exactly the kurtosis of the unit variance s_i.

The element-wise algorithm (28) for all elements of matrix $\mathbf{U}$, followed by the symmetrical orthogonalization (12), is the FastICA algorithm for kurtosis in the transformed space. This case was analyzed by one of the authors in [21]. In the following, the main results will be briefly repeated.

Let us again solve the fixed points as follows.

Lemma 4: Let $\mathbf{U}$ be an orthogonal matrix ($\mathbf{U}\mathbf{U}^T = \mathbf{I}$) such that in each column j, each $|u_{ij}|$ is either 0 or a positive constant β_j. Then, it is a fixed point of the algorithm (28) and (12).

Proof: Now $u_{ij}^3 = \beta_j^2 u_{ij}$ and from (28) it follows:

$$\bar{u}_{ij} = \beta_j^2 \kappa_j u_{ij}.$$

Thus

$$\bar{\mathbf{U}} = \mathbf{U}\mathbf{C}$$

where $\mathbf{C} = \mathrm{diag}(\beta_1^2\kappa_1, \ldots, \beta_n^2\kappa_n)$. The rest of the proof is similar to Lemma 1.

As in Lemma 2, we can easily show that also permutations of these fixed points are fixed points. The main result to be shown for the kurtosis case is that, among the fixed points mentioned in Lemma 4, only the diagonal matrix

$$\mathbf{U} = \mathbf{D} = \mathrm{diag}(\pm 1, \ldots, \pm 1) \tag{29}$$

or its permutations are stable. The speed of convergence to the stable fixed points is cubic. On the contrary, all those matrices that have more than one nonzero element on each row and column are unstable.

To show the stability of (29), let us make the perturbation

$$u_{ij} = d_{ij} + \epsilon_{ij} \tag{30}$$

where d_{ij} are the elements of matrix $\mathbf{D}$ and $|\epsilon_{ij}| \le \epsilon$ for all i, j, with ϵ small. Then, (28) gives

$$\begin{aligned}\bar{u}_{ij} &= \kappa_j(d_{ij} + \epsilon_{ij})^3 &(31)\\ &= \kappa_j\left(d_{ij}^3 + 3d_{ij}^2\epsilon_{ij} + 3d_{ij}\epsilon_{ij}^2\right) + \mathcal{O}(\epsilon^3). &(32)\end{aligned}$$

We see that the off-diagonal elements of $\bar{\mathbf{U}}$ are proportional to ϵ^3, denoted $\mathcal{O}(\epsilon^3)$, because for them $d_{ij} = 0$. Denote the diagonal matrix of the perturbations ϵ_{ii} by $\boldsymbol{\mathcal{E}}$ and remember that $\mathbf{C}$ is the diagonal matrix of the kurtoses κ_j, as before. Then, we can write (32) in the form of a diagonal matrix plus error as

$$\bar{\mathbf{U}} = \mathbf{C}(\mathbf{D} + 3\boldsymbol{\mathcal{E}} + 3\mathbf{D}\boldsymbol{\mathcal{E}}^2) + \mathcal{O}(\epsilon^3)$$

where we have used the fact that $\mathbf{D}^2 = \mathbf{I}$. This gives

$$\bar{\mathbf{U}}\bar{\mathbf{U}}^T = \mathbf{C}^2(\mathbf{I} + 15\boldsymbol{\mathcal{E}}^2 + 6\mathbf{D}\boldsymbol{\mathcal{E}}) + \mathcal{O}(\epsilon^3).$$

Thus

$$(\bar{\mathbf{U}}\bar{\mathbf{U}}^T)^{-1/2} = \mathbf{C}^{-1}(\mathbf{I} - 3\mathbf{D}\boldsymbol{\mathcal{E}} + 6\boldsymbol{\mathcal{E}}^2) + \mathcal{O}(\epsilon^3)$$

and finally

$$\begin{aligned}&(\bar{\mathbf{U}}\bar{\mathbf{U}}^T)^{-1/2}\bar{\mathbf{U}}\\ &\quad= (\mathbf{I} - 3\mathbf{D}\boldsymbol{\mathcal{E}} + 6\boldsymbol{\mathcal{E}}^2)(\mathbf{D} + 3\boldsymbol{\mathcal{E}} + 3\mathbf{D}\boldsymbol{\mathcal{E}}^2) + \mathcal{O}(\epsilon^3)\\ &\quad= \mathbf{D} + \mathcal{O}(\epsilon^3).\end{aligned} \tag{33}$$

The error is proportional to the third power of the previous error, showing that the convergence is indeed cubic.

Remark: According to Lemma 4, a typical column of a fixed point matrix $\mathbf{U}$ has from 0 to $n-1$ zeros and the rest of the elements equal to $\pm\beta$. Orthogonality of $\mathbf{U}$ constrains this further: If p is the number of nonzero elements, then $\beta = 1/\sqrt{p}$ because

each column must have unit norm. The mutual orthogonality of the columns sets further constraints on the locations and signs of the nonzero elements in each column. For the stable fixed points, there is only one nonzero element ± 1 in each column, each one on a different row.

The stability of the fixed points $\mathbf{PD}$ and $\mathbf{DP}$, with $\mathbf{P}$ an orthogonal permutation matrix, can be treated in the same way [21]. On the contrary, those fixed points that have more than one nonzero element in the rows are not stable. This can be shown by a counterexample which is the topic of Section IV.

IV. Global Analysis for Two Mixtures, Two Sources

The smallest nontrivial mixing model is one in which two independent sources are mixed to two mixtures. After the whitening and the transformation to the $\mathbf{U}$-space, the mixing matrix $\mathbf{U}$ is orthogonal. For the 2×2 case, the orthogonal matrix can be parameterized with a single parameter in the following form:

$$\mathbf{U} = \begin{pmatrix} x & \sqrt{1-x^2} \\ \sqrt{1-x^2} & -x \end{pmatrix}$$

where $|x| \leq 1$. The expression $\sqrt{1-x^2}$ can have plus or minus sign as long as the sign is the same in both off-diagonal elements, to guarantee orthogonality of the rows.

It turns out that global iteration formulas can be derived for the four matrix elements from the FastICA algorithm. They show very clearly how the algorithm behaves in this simple case, and may also give an idea of the general behavior. Unfortunately, extending such a simplified analysis to dimensions higher than two is not easy.

We could also use the polar parametrization $x = \cos\theta, y = \sin\theta$, as was done in a related approach to the global properties of the kurtosis cost function for ICA [19].

A. Kurtosis Cost Function

It may be instructive to look at the kurtosis case first as it is relatively simple and illustrative. This was earlier analyzed in [21]; see also [19]. Lemma 4 gives the fixed points: Either $x = \pm 1, x = 0$, or $x = \pm\sqrt{1/2}$. Assume that both kurtoses κ_1, κ_2 are nonzero. It follows that

$$\bar{\mathbf{U}} = \begin{pmatrix} \kappa_1 x^3 & \kappa_2(1-x^2)\sqrt{1-x^2} \\ \kappa_1(1-x^2)\sqrt{1-x^2} & -\kappa_2 x^3 \end{pmatrix}$$

implying

$$(\bar{\mathbf{U}}\bar{\mathbf{U}}^T)^{-1/2}\bar{\mathbf{U}} = \frac{1}{\sqrt{x^6 + (1-x^2)^3}} \times \begin{pmatrix} x^3 & (1-x^2)\sqrt{1-x^2} \\ (1-x^2)\sqrt{1-x^2} & -x^3 \end{pmatrix}. \quad (34)$$

It is notable that the kurtoses have disappeared altogether from the iteration. In the 2×2 case, the convergence of the algorithm is totally *independent of the signs and values of the kurtoses*, as long as they are nonzero. Note also that the "one-bit-matching" assumption [18], by which the nonlinearity $g(\cdot)$ should be adapted to the signs of the kurtoses of the sources, is not necessary for fixed-point iterations like FastICA. This iteration will find both the maxima and minima of the cost function, while a gradient–ascent type of learning rule only finds the local maxima.

Let us look at how the elements u_{ij} change in one step of the iteration. The relative change is the same for all the elements, following the algorithm:

$$x \leftarrow f(x) \quad (35)$$

where

$$f(x) = x^3/\sqrt{x^6 + (1-x^2)^3}. \quad (36)$$

Algorithm (35) is a fixed-point iteration whose convergence depends on the iteration function $f(x)$. Fig. 1 shows the graph of the iteration function.

As seen from Fig. 1 and also easily proven (see [21]), the fixed points of this iteration are $x = 0, \pm\sqrt{1/2}$, and ± 1. The points $\pm\sqrt{1/2}$ are unstable. The points $0, \pm 1$ are stable and the order of convergence to these points is three.

Typically, starting from any other initial point but the points $\pm\sqrt{1/2}$, the elements rapidly approach the closest stable fixed point and then converge very fast, because close to the fixed point the error at any step is the error at the previous step to the third power. For example, taking 0.5 as the initial point would lead to the iteration

$$\begin{aligned} x_0 &= 0.5 \\ x_1 &= f(x_0) = 0.1890 \\ x_2 &= f(x_1) = 0.0071 \\ x_3 &= f(x_4) = 0.0000. \end{aligned}$$

Actually, the last value is of the order 10^{-7}. After this, converging to zero is very rapid. This iteration sequence could be traced from the graph of $f(x)$ in Fig. 1.

The elements on the same row or column of $\mathbf{U}$ are always on the opposite sides of the unstable point, because the sum of their squares is equal to one. Thus, one of them converges to 0 if the other converges to ± 1 and vice versa, and matrix $\mathbf{U}$ becomes the diagonal matrix (29) or one with permuted columns.

B. General Cost Function

For the general odd function g, the algorithm is as follows:

$$\bar{\mathbf{u}}_i = \mathrm{E}\left\{\mathbf{s}g\left(\mathbf{u}_i^T\mathbf{s}\right)\right\} - \mathrm{E}\left\{g'\left(\mathbf{u}_i^T\mathbf{s}\right)\right\}\mathbf{u}_i \quad (37)$$

and

$$\mathbf{U} = (\bar{\mathbf{U}}\bar{\mathbf{U}}^T)^{-1/2}\bar{\mathbf{U}}. \quad (38)$$

Opening the (37) for all the elements $\bar{u}_{ij}$, we have

$$\begin{aligned} \bar{u}_{11} &= \mathrm{E}\{s_1 g(xs_1 + \sqrt{1-x^2}s_2)\} \\ &\quad - \mathrm{E}\{g'(xs_1 + \sqrt{1-x^2}s_2)\}x \quad (39) \\ \bar{u}_{12} &= \mathrm{E}\{s_2 g(xs_1 + \sqrt{1-x^2}s_2)\} \\ &\quad - \mathrm{E}\{g'(xs_1 + \sqrt{1-x^2}s_2)\}\sqrt{1-x^2} \quad (40) \end{aligned}$$

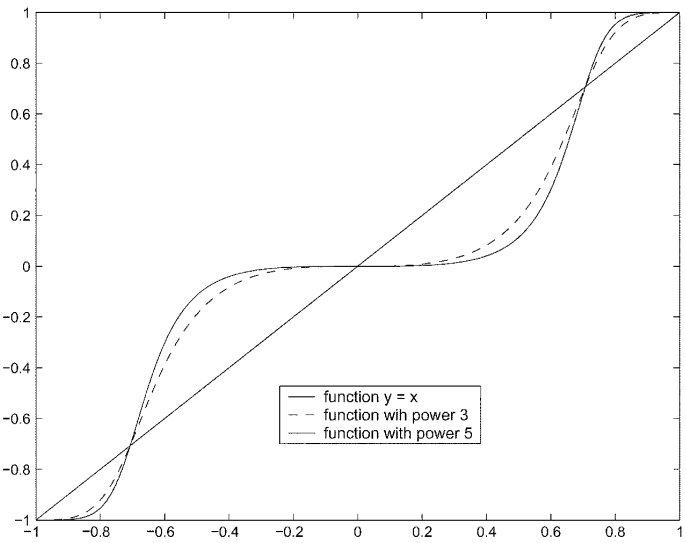

Fig. 1. Comparison of the iteration functions $f(x)$ in the cubic and fifth-degree case.

$$\bar{u}_{21} = \mathrm{E}\{s_1 g(\sqrt{1-x^2}s_1 - xs_2)\} - \mathrm{E}\{g'(\sqrt{1-x^2}s_1 - xs_2)\}\sqrt{1-x^2} \quad (41)$$

$$\bar{u}_{22} = \mathrm{E}\{s_2 g(\sqrt{1-x^2}s_1 - xs_2)\} + \mathrm{E}\{g'(\sqrt{1-x^2}s_1 - xs_2)\}x. \quad (42)$$

Thus

$$\bar{\mathbf{U}}\bar{\mathbf{U}}^T = \begin{pmatrix} \bar{u}_{11}^2 + \bar{u}_{12}^2 & \bar{u}_{11}\bar{u}_{21} + \bar{u}_{12}\bar{u}_{22} \\ \bar{u}_{11}\bar{u}_{21} + \bar{u}_{12}\bar{u}_{22} & \bar{u}_{21}^2 + \bar{u}_{22}^2 \end{pmatrix}$$

$$(\bar{\mathbf{U}}\bar{\mathbf{U}}^T)^{-1} = \frac{1}{(\bar{u}_{12}\bar{u}_{21} - \bar{u}_{11}\bar{u}_{22})^2} \times \begin{pmatrix} \bar{u}_{21}^2 + \bar{u}_{22}^2 & -\bar{u}_{11}\bar{u}_{21} - \bar{u}_{12}\bar{u}_{22} \\ -\bar{u}_{11}\bar{u}_{21} - \bar{u}_{12}\bar{u}_{22} & \bar{u}_{11}^2 + \bar{u}_{12}^2 \end{pmatrix}.$$

Assume that the determinant of the matrix $\bar{\mathbf{U}}$ is negative, that is, $\bar{u}_{12}\bar{u}_{21} - \bar{u}_{11}\bar{u}_{22} > 0$. We have

$$(\bar{\mathbf{U}}\bar{\mathbf{U}}^T)^{-1/2} = \frac{1}{\bar{u}_{12}\bar{u}_{21} - \bar{u}_{11}\bar{u}_{22}} \begin{pmatrix} a & b \\ b & c \end{pmatrix}$$

where

$$b = -\frac{\bar{u}_{11}\bar{u}_{21} + \bar{u}_{12}\bar{u}_{22}}{\sqrt{(\bar{u}_{11} - \bar{u}_{22})^2 + (\bar{u}_{21} + \bar{u}_{12})^2}} \quad (43)$$

$$a = \frac{\bar{u}_{21}^2 + \bar{u}_{22}^2 - \bar{u}_{11}\bar{u}_{22} + \bar{u}_{12}\bar{u}_{21}}{\sqrt{(\bar{u}_{11} - \bar{u}_{22})^2 + (\bar{u}_{21} + \bar{u}_{12})^2}} \quad (44)$$

$$c = \frac{\bar{u}_{11}^2 + \bar{u}_{12}^2 - \bar{u}_{11}\bar{u}_{22} + \bar{u}_{12}\bar{u}_{21}}{\sqrt{(\bar{u}_{11} - \bar{u}_{22})^2 + (\bar{u}_{21} + \bar{u}_{12})^2}}. \quad (45)$$

Therefore

$$(\bar{\mathbf{U}}\bar{\mathbf{U}}^T)^{-1/2}\bar{\mathbf{U}} = \frac{1}{\sqrt{(\bar{u}_{11} - \bar{u}_{22})^2 + (\bar{u}_{21} + \bar{u}_{12})^2}} \times \begin{pmatrix} \bar{u}_{11} - \bar{u}_{22} & \bar{u}_{12} + \bar{u}_{21} \\ \bar{u}_{12} + \bar{u}_{21} & \bar{u}_{22} - \bar{u}_{11} \end{pmatrix}.$$

Looking at the changes of all the elements in one step of iteration, they follow the same algorithm

$$x \leftarrow f(x) \quad (46)$$

where

$$f(x) = \frac{\bar{u}_{11} - \bar{u}_{22}}{\sqrt{(\bar{u}_{11} - \bar{u}_{22})^2 + (\bar{u}_{21} + \bar{u}_{12})^2}} \quad (47)$$

with $\bar{u}_{ij}$ given in (39)–(42).

In the case that the determinant of the matrix $\bar{\mathbf{U}}$ is positive, by a similar calculation, the function $f(x)$ becomes

$$f(x) = \frac{\bar{u}_{11} + \bar{u}_{22}}{\sqrt{(\bar{u}_{11} + \bar{u}_{22})^2 + (\bar{u}_{21} - \bar{u}_{12})^2}}. \quad (48)$$

To illustrate this general result, let us look at two different nonlinearities: First, the fifth power to see the difference to the earlier analyzed third power, and second, the hyperbolic tangent function, widely used in ICA algorithms.

C. Function $g(x) = x^5$

To simplify the calculation, let us assume here that both s_1 and s_2 have symmetric density functions. Thus, all central

moments of odd degree are zero. For $g(x) = x^5$, the rule (37) becomes

$$\bar{\mathbf{u}}_i = \mathrm{E}\left\{\mathbf{s}\left(\mathbf{u}_i^T\mathbf{s}\right)^5\right\} - \mathrm{E}\left\{5\left(\mathbf{u}_i^T\mathbf{s}\right)^4\right\}\mathbf{u}_i. \tag{49}$$

Equations (39)–(42) become

$$\begin{aligned}
\bar{u}_{11} &= \mathrm{E}\left\{s_1\left(\mathbf{u}_1^T\mathbf{s}\right)^5\right\} - \mathrm{E}\left\{5\left(\mathbf{u}_1^T\mathbf{s}\right)^4\right\}u_{11} \\
&= \mathrm{E}\{s_1(xs_1 + \sqrt{1-x^2}s_2)^5\} \\
&\quad - \mathrm{E}\{5(xs_1 + \sqrt{1-x^2}s_2)^4\}x \\
&= x^5\mathrm{E}\left\{s_1^6\right\} + 5x^3(2-3x^2)\mathrm{E}\left\{s_1^4\right\} \\
&\quad - 30x^3(1-x^2) \\
\bar{u}_{12} &= \mathrm{E}\left\{s_2\left(\mathbf{u}_1^T\mathbf{s}\right)^5\right\} - \mathrm{E}\left\{5\left(\mathbf{u}_1^T\mathbf{s}\right)^4\right\}u_{12} \\
&= \mathrm{E}\{s_2(xs_1 + \sqrt{1-x^2}s_2)^5\} \\
&\quad - \mathrm{E}\{5(xs_1 + \sqrt{1-x^2}s_2)^4\}\sqrt{1-x^2} \\
&= (\sqrt{1-x^2})^5\mathrm{E}\left\{s_2^6\right\} \\
&\quad + 5(\sqrt{1-x^2})^3(3x^2-1)\mathrm{E}\left\{s_2^4\right\} \\
&\quad -30x^2(\sqrt{1-x^2})^3 \\
\bar{u}_{21} &= \mathrm{E}\left\{s_1\left(\mathbf{u}_2^T\mathbf{s}\right)^5\right\} - \mathrm{E}\left\{5\left(\mathbf{u}_2^T\mathbf{s}\right)^4\right\}u_{21} \\
&= \mathrm{E}\{s_1(\sqrt{1-x^2}s_1 - xs_2)^5\} \\
&\quad -\mathrm{E}\{5(\sqrt{1-x^2}s_1 - xs_2)^4\}\sqrt{1-x^2} \\
&= (\sqrt{1-x^2})^5\mathrm{E}\left\{s_1^6\right\} \\
&\quad + 5(\sqrt{1-x^2})^3(3x^2-1)\mathrm{E}\left\{s_1^4\right\} \\
&\quad - 30x^2(\sqrt{1-x^2})^3 \\
\bar{u}_{22} &= \mathrm{E}\left\{s_2\left(\mathbf{u}_2^T\mathbf{s}\right)^5\right\} - \mathrm{E}\left\{5\left(\mathbf{u}_2^T\mathbf{s}\right)^4\right\}u_{22} \\
&= \mathrm{E}\{s_2(\sqrt{1-x^2}s_1 - xs_2)^5\} \\
&\quad + \mathrm{E}\{5(\sqrt{1-x^2}s_1 - xs_2)^4\}x \\
&= -x^5\mathrm{E}\left\{s_2^6\right\} - 5x^3(2-3x^2)\mathrm{E}\left\{s_2^4\right\} \\
&\quad + 30x^3(1-x^2).
\end{aligned}$$

Now we can calculate the function $f(x)$ as shown in (47) and (48). If $\mathrm{dct}(\bar{\mathbf{U}}) < 0$

$$\begin{aligned}
\bar{u}_{11} - \bar{u}_{22} &= x^5\left(\mathrm{E}\left\{s_1^6\right\} + \mathrm{E}\left\{s_2^6\right\}\right) \\
&\quad + 5x^3(2-3x^2)\left(\mathrm{E}\left\{s_1^4\right\} + \mathrm{E}\left\{s_2^4\right\}\right) \\
&\quad - 60x^3(1-x^2) \\
&= x^3(\alpha x^2 + \beta(1-x^2))
\end{aligned}$$

$$\begin{aligned}
\bar{u}_{12} + \bar{u}_{21} &= \sqrt{1-x^2}^5\left(\mathrm{E}\left\{s_1^6\right\} + \mathrm{E}\left\{s_2^6\right\}\right) \\
&\quad + 5\sqrt{1-x^2}^3(3x^2-1)(\mathrm{E}\left\{s_1^4\right\} + \mathrm{E}\left\{s_2^4\right\}) \\
&\quad - 60x^2(\sqrt{1-x^2})^3 \\
&= \sqrt{1-x^2}^3(\alpha(1-x^2) + \beta x^2)
\end{aligned}$$

where

$$\alpha = \mathrm{E}\left\{s_1^6\right\} + \mathrm{E}\left\{s_2^6\right\} - 5\mathrm{E}\left\{s_1^4\right\} - 5\mathrm{E}\left\{s_2^4\right\} \tag{50}$$

$$\beta = 10\mathrm{E}\left\{s_1^4\right\} + 10\mathrm{E}\left\{s_2^4\right\} - 60. \tag{51}$$

Using the relations between cumulants and central moments [20], we note that these can also be written as

$$\alpha = \kappa^6(s_1) + \kappa^6(s_2) + 10(\mathrm{kurt}(s_1) + \mathrm{kurt}(s_2)) \tag{52}$$

$$\beta = 10(\mathrm{kurt}(s_1) + \mathrm{kurt}(s_2)) \tag{53}$$

where κ^6 denotes the sixth-degree cumulant. Now

$$\begin{aligned}
(\bar{u}_{11} - \bar{u}_{22})^2 + (\bar{u}_{12} + \bar{u}_{21})^2 &= x^6((\alpha x^2 + \beta(1-x^2)))^2 \\
&\quad +(1-x^2)^3(\alpha(1-x^2) + \beta x^2)^2.
\end{aligned} \tag{54}$$

Thus, we get (55) as shown at the bottom of the page. If $\mathrm{dct}(\bar{\mathbf{U}}) > 0$, the function $f(x)$ is as in (56), shown at the bottom of the page, where

$$\alpha_1 = \kappa^6(s_1) - \kappa^6(s_2) + 10(\mathrm{kurt}(s_1) - \mathrm{kurt}(s_2)) \tag{57}$$

$$\beta_1 = 10(\mathrm{kurt}(s_1) - \mathrm{kurt}(s_2)). \tag{58}$$

This gives the fixed points again as $x = 0, x = \pm 1$, and $x = \pm\sqrt{1/2}$, or the same as in the kurtosis case. It can be shown that the points $x = \pm\sqrt{1/2}$ are unstable by looking at the derivative of the function $f(x)$.

The shape of the function depends on the two parameters α, β which in turn depend on the sixth and fourth central moments of the sources. A simple special case is $\beta = 0$, which by (53) means that the kurtoses of the two sources have different signs and they cancel out. Note that one of the sources is then sub-Gaussian, the other super-Gaussian. Then, in analogy with (36), we have

$$f(x) = \frac{x^5}{\sqrt{x^{10} + (1-x^2)^5}}.$$

For this function, the convergence is very fast, of the order five, because very close to the fixed point $x = 0$ it holds $f(x) \approx x^5$. However, this is the only such case; if $\beta \neq 0$, then close to $x = 0$ it holds $f(x) \approx x^3$, and the convergence is again cubic.

$$f(x) = \frac{x^3(\alpha x^2 + \beta(1-x^2))}{\sqrt{x^6(\alpha x^2 + \beta(1-x^2))^2 + (1-x^2)^3(\alpha(1-x^2) + \beta x^2)^2}} \tag{55}$$

$$f(x) = \frac{x^3(\alpha_1 x^2 + \beta_1(1-x^2))}{\sqrt{x^6(\alpha_1 x^2 + \beta_1(1-x^2))^2 + (1-x^2)^3(\alpha_1(1-x^2) + \beta_1 x^2)^2}} \tag{56}$$

Fig. 1 gives a more realistic example in which the source s_1 is uniformly distributed (sub-Gaussian) in the interval $[-\sqrt{3}, \sqrt{3}]$, thus, it has zero mean and unit variance. Source s_2 has Laplace distribution (super-Gaussian) with zero mean and unit variance, with the probability density function

$$p(x) = \frac{\sqrt{2}}{2} e^{-|x|\sqrt{2}}. \tag{59}$$

Now the function (56) becomes

$$f(x) = \frac{x^3(36\frac{6}{7}x^2 + 18)}{\sqrt{x^6(36\frac{6}{7}x^2 + 18)^2 + (1-x^2)^3(54\frac{6}{7} - 36\frac{6}{7}x^2)^2}}. \tag{60}$$

As seen from Fig. 1, the function is close to the cubic function near the stable fixed points.

D. Function $g(x) = \tanh(x)$

For the function $g(x) = \tanh(x)$, the behavior of the algorithm depends on the probability densities of the sources in more complex ways than for the polynomial functions. The influence cannot be expressed through a finite number of moments or cumulants as in the polynomial case. Therefore, let us assume some densities. First, take a case in which both s_1 and s_2 are uniformly distributed on the interval $[-\sqrt{3}, \sqrt{3}]$ with the probability density function

$$p(x) = \begin{cases} \frac{1}{2\sqrt{3}}, & x \in [-\sqrt{3}, \sqrt{3}] \\ 0, & \text{otherwise.} \end{cases}$$

Thus, the variances of s_1 and s_2 are 1. The rule (37) becomes

$$\bar{\mathbf{u}}_i = \mathrm{E}\left\{\mathbf{s}\tanh\left(\mathbf{u}_i^T\mathbf{s}\right)\right\} - \mathrm{E}\left\{1 - \tanh^2\left(\mathbf{u}_i^T\mathbf{s}\right)\right\}\mathbf{u}_i. \tag{61}$$

Then, (39)–(42) become (using shorthand notation a for $\sqrt{3}$)

$$\begin{aligned}
\bar{u}_{11} &= \mathrm{E}\left\{s_1\tanh\left(\mathbf{u}_1^T\mathbf{s}\right)\right\} - \mathrm{E}\left\{1 - \tanh^2\left(\mathbf{u}_1^T\mathbf{s}\right)\right\}x \\
&= \frac{1}{12}\left(\int_{-a}^{a}\int_{-a}^{a} s_1\tanh\left(\mathbf{u}_1^T\mathbf{s}\right)ds_1ds_2 \right. \\
&\quad \left. - \int_{-a}^{a}\int_{-a}^{a} x\left(1 - \tanh^2\left(\mathbf{u}_1^T\mathbf{s}\right)\right)ds_1ds_2\right) \\
\bar{u}_{21} &= \mathrm{E}\left\{s_2\tanh\left(\mathbf{u}_1^T\mathbf{s}\right)\right\} \\
&\quad - \mathrm{E}\left\{1 - \tanh^2\left(\mathbf{u}_1^T\mathbf{s}\right)\right\}\sqrt{1-x^2} \\
&= \frac{1}{12}\left(\int_{-a}^{a}\int_{-a}^{a} s_2\tanh\left(\mathbf{u}_1^T\mathbf{s}\right)ds_1ds_2 \right. \\
&\quad \left. - \int_{-a}^{a}\int_{-a}^{a} \sqrt{1-x^2}\left(1 - \tanh^2\left(\mathbf{u}_1^T\mathbf{s}\right)\right)ds_1ds_2\right) \\
\bar{u}_{12} &= \mathrm{E}\left\{s_1\tanh\left(\mathbf{u}_2^T\mathbf{s}\right)\right\} \\
&\quad - \mathrm{E}\left\{1 - \tanh^2\left(\mathbf{u}_2^T\mathbf{s}\right)\right\}\sqrt{1-x^2} \\
&= \frac{1}{12}\left(\int_{-a}^{a}\int_{-a}^{a} s_1\tanh\left(\mathbf{u}_2^T\mathbf{s}\right)ds_1ds_2 \right. \\
&\quad \left. - \int_{-a}^{a}\int_{-a}^{a} \sqrt{1-x^2}\left(1 - \tanh^2\left(\mathbf{u}_2^T\mathbf{s}\right)\right)ds_1ds_2\right) \\
\bar{u}_{22} &= \mathrm{E}\left\{s_2\tanh\left(\mathbf{u}_2^T\mathbf{s}\right)\right\} + \mathrm{E}\left\{1 - \tanh^2\left(\mathbf{u}_2^T\mathbf{s}\right)\right\}x \\
&= \frac{1}{12}\left(\int_{-a}^{a}\int_{-a}^{a} s_2\tanh\left(\mathbf{u}_2^T\mathbf{s}\right)ds_1ds_2 \right. \\
&\quad \left. + \int_{-a}^{a}\int_{-a}^{a} x\left(1 - \tanh^2\left(\mathbf{u}_2^T\mathbf{s}\right)\right)ds_1ds_2\right).
\end{aligned}$$

To calculate the function $f(x)$ as shown in (47), we first compute

$$\begin{aligned}
&12(\bar{u}_{11} - \bar{u}_{22}) \\
&= \int_{-a}^{a}\int_{-a}^{a} s_1\tanh\left(\mathbf{u}_1^T\mathbf{s}\right)ds_1ds_2 \\
&\quad - \int_{-a}^{a}\int_{-a}^{a} s_2\tanh\left(\mathbf{u}_2^T\mathbf{s}\right)ds_1ds_2 \qquad (62) \\
&\quad - x\left(\int_{-a}^{a}\int_{-a}^{a}\left(1 - \tanh^2\left(\mathbf{u}_1^T\mathbf{s}\right)\right)ds_1ds_2 \right. \\
&\quad \left. + \int_{-a}^{a}\int_{-a}^{a}\left(1 - \tanh^2\left(\mathbf{u}_2^T\mathbf{s}\right)\right)ds_1ds_2\right) \qquad (63) \\
&= 2\int_{-a}^{a}\int_{-a}^{a} s_1\tanh\left(\mathbf{u}_1^T\mathbf{s}\right)ds_1ds_2 \\
&\quad - 2\int_{-a}^{a}\int_{-a}^{a} x\left(1 - \tanh^2\left(\mathbf{u}_1^T\mathbf{s}\right)\right)ds_1ds_2 \qquad (64)
\end{aligned}$$

and

$$\begin{aligned}
&12(\bar{u}_{21} + \bar{u}_{12}) \\
&= \int_{-a}^{a}\int_{-a}^{a} s_2\tanh\left(\mathbf{u}_1^T\mathbf{s}\right)ds_1ds_2 \\
&\quad + \int_{-a}^{a}\int_{-a}^{a} s_1\tanh\left(\mathbf{u}_2^T\mathbf{s}\right)ds_1ds_2 \qquad (65) \\
&\quad - \sqrt{1-x^2}\left(\int_{-a}^{a}\int_{-a}^{a}\left(1 - \tanh^2\left(\mathbf{u}_1^T\mathbf{s}\right)\right)ds_1ds_2 \right. \\
&\quad \left. + \int_{-a}^{a}\int_{-a}^{a}\left(1 - \tanh^2\left(\mathbf{u}_2^T\mathbf{s}\right)\right)ds_1ds_2\right) \qquad (66) \\
&= 2\int_{-a}^{a}\int_{-a}^{a} s_2\tanh\left(\mathbf{u}_1^T\mathbf{s}\right)ds_1ds_2 \\
&\quad - 2\int_{-a}^{a}\int_{-a}^{a} \sqrt{1-x^2}\left(1 - \tanh^2\left(\mathbf{u}_1^T\mathbf{s}\right)\right)ds_1ds_2. \qquad (67)
\end{aligned}$$

Since

$$\begin{aligned}
\Delta &:= \frac{1}{2}\int_{-a}^{a}\int_{-a}^{a}\left(1 - \tanh^2\left(\mathbf{u}_1^T\mathbf{s}\right)\right)ds_1ds_2 \\
&= \frac{1}{x\sqrt{1-x^2}}\ln\frac{e^{xa+\sqrt{1-x^2}a} + e^{-xa-\sqrt{1-x^2}a}}{e^{xa-\sqrt{1-x^2}a} + e^{-xa+\sqrt{1-x^2}a}} \qquad (68) \\
\Delta_1 &:= \frac{1}{2}\int_{-a}^{a}\int_{-a}^{a} s_1\tanh\left(\mathbf{u}_1^T\mathbf{s}\right)ds_1ds_2 \\
&= \frac{1}{\sqrt{1-x^2}}\int_{-a}^{a} s_1\ln(e^{xs_1+\sqrt{1-x^2}a} + e^{-xs_1-\sqrt{1-x^2}a})ds_1 \qquad (69) \\
\Delta_2 &:= \frac{1}{2}\int_{-a}^{a}\int_{-a}^{a} s_2\tanh\left(\mathbf{u}_1^T\mathbf{s}\right)ds_1ds_2 \\
&= \frac{1}{x}\int_{-a}^{a} s_2\ln(e^{xa+\sqrt{1-x^2}s_2} + e^{-xa-\sqrt{1-x^2}s_2})ds_2. \qquad (70)
\end{aligned}$$

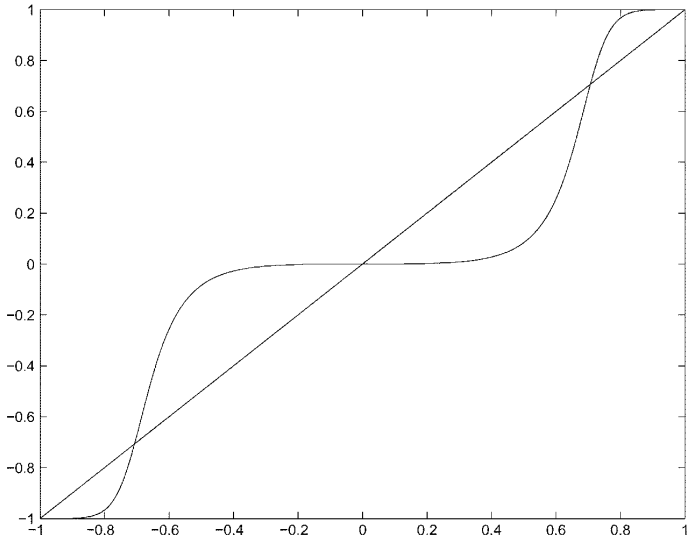

Fig. 2. Plot of the function $f(x)$ in (71).

Since the previous two integrations could not be integrated exactly, we have to resort to numerical integration. This gives the following function $f(x)$, plotted in Fig. 2:

$$f(x) = \frac{\Delta_1 - x\Delta}{\sqrt{(\Delta_1 - x\Delta)^2 + (\Delta_2 - \sqrt{1-x^2}\Delta)^2}}. \tag{71}$$

Note how the function again qualitatively looks very similar to the case of third- and fifth-degree polynomials: The fixed points are the same, including the unstable point $\pm\sqrt{1/2}$, and the convergence to the stable fixed points is very fast. For this simple 2×2 case, there are no false stable fixed points.

Second, for the $\tanh$ nonlinearity, assume both s_1 and s_2 are binary distributed: They take only values $\{-1, 1\}$ with equal probabilities. This density is sub-Gaussian. Then, the elements $\bar{u}_{ij}$ become (dropping out constant multipliers due to the probabilities)

$$\begin{aligned}\bar{u}_{11} = -\bar{u}_{22} = &\tanh(x + \sqrt{1-x^2}) \\ &+ \tanh(x - \sqrt{1-x^2}) \\ &- (2 - \tanh^2(x + \sqrt{1-x^2}) \\ &- \tanh^2(x - \sqrt{1-x^2}))x,\end{aligned}$$

and

$$\begin{aligned}\bar{u}_{21} = \bar{u}_{12} = &\tanh(x + \sqrt{1-x^2}) \\ &- \tanh(x - \sqrt{1-x^2}) \\ &- (2 - \tanh^2(x + \sqrt{1-x^2}) \\ &- \tanh^2(x - \sqrt{1-x^2}))\sqrt{1-x^2}.\end{aligned}$$

Now the function $f(x)$ is

$$f(x) = \frac{\bar{u}_{11}}{\sqrt{\bar{u}_{11}^2 + \bar{u}_{21}^2}} \tag{72}$$

and it is depicted in Fig. 3.

Note how the function again has the same general shape as the previous ones, with the same stable and unstable fixed points. Due to the flatness of the curve around the stable points, convergence is very fast.

Both aforementioned cases were sub-Gaussian. One may wonder how the function changes if the kurtoses of the two sources have different signs. Therefore, in the third case, assume the source s_1 is binary distributed as previously, thus sub-Gaussian, but the source s_2 takes the values $\{-2, 2\}$ with equal probabilities $1/8$, and zero with probability $3/4$. It is easy to check that s_2 is then super-Gaussian, and both sources have unit variance. Calculating the values of elements $\bar{u}_{ij}$, and neglecting again constant multipliers due to the probabilities, we have

$$\begin{aligned}\bar{u}_{11} = &\frac{1}{8}\tanh(x + 2\sqrt{1-x^2}) \\ &+ \frac{1}{8}\tanh(x - 2\sqrt{1-x^2}) + \frac{3}{4}\tanh(x) - x \\ &+ x\left(\frac{3}{4}\tanh^2(x) + \frac{1}{8}\tanh^2(x + 2\sqrt{1-x^2})\right. \\ &\left.+\frac{1}{8}\tanh^2(x - 2\sqrt{1-x^2})\right) \\ \bar{u}_{12} = &\frac{1}{8}\tanh(\sqrt{1-x^2} + 2x) \\ &+ \frac{1}{8}\tanh(\sqrt{1-x^2} - 2x) \\ &+ \frac{3}{4}\tanh(\sqrt{1-x^2}) - \sqrt{1-x^2} \\ &+ \sqrt{1-x^2}\left(\frac{3}{4}\tanh^2(\sqrt{1-x^2})\right. \\ &+ \frac{1}{8}\tanh^2(\sqrt{1-x^2} + 2x) \\ &\left.+\frac{1}{8}\tanh^2(\sqrt{1-x^2} - 2x)\right)\end{aligned}$$

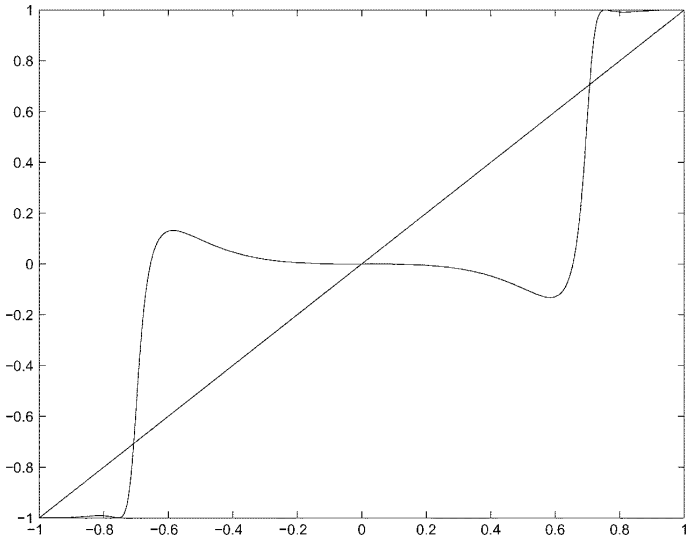

Fig. 3. Plot of the function $f(x)$ in (72).

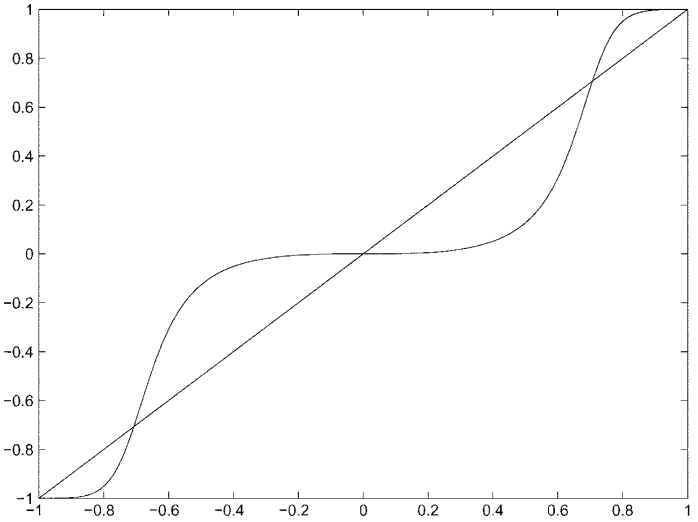

Fig. 4. Plot of the function $f(x)$ in (73).

$$\begin{aligned}\bar{u}_{22} = &\frac{1}{4}\tanh(\sqrt{1-x^2}-2x)\\ &-\frac{1}{4}\tanh(\sqrt{1-x^2}+2x)+x\\ &-x\left(\frac{3}{4}\tanh^2(\sqrt{1-x^2})\right.\\ &+\frac{1}{8}\tanh^2(\sqrt{1-x^2}+2x)\\ &\left.+\frac{1}{8}\tanh^2(\sqrt{1-x^2}-2x)\right)\end{aligned}$$

and

$$\begin{aligned}\bar{u}_{21} = &\frac{1}{4}\tanh(x+2\sqrt{1-x^2})\\ &+\frac{1}{4}\tanh(-x+2\sqrt{1-x^2})-\sqrt{1-x^2}\\ &+\sqrt{1-x^2}\left(\frac{3}{4}\tanh^2(x)\right.\\ &+\frac{1}{8}\tanh^2(x+2\sqrt{1-x^2})\\ &\left.+\frac{1}{8}\tanh^2(x-2\sqrt{1-x^2})\right).\end{aligned}$$

Since $\bar{u}_{11}\bar{u}_{22}-\bar{u}_{21}\bar{u}_{12}\geq 0$, the function $f(x)$ is

$$f(x)=\frac{\bar{u}_{11}+\bar{u}_{22}}{\sqrt{(\bar{u}_{11}+\bar{u}_{22})^2+(\bar{u}_{21}-\bar{u}_{12})^2}}. \tag{73}$$

Again, as seen from Fig. 4, the function is qualitatively very similar to the previous ones, with the same stable and unstable fixed points.

V. Conclusion

The FastICA algorithm with symmetrical orthogonalization is widely used in practice for blind source separation, based on its good accuracy and convergence speed. It would, therefore, be essential to have theoretical confirmation of the good properties. Up to now, only partial results have been available, mostly concerning the one-unit behavior. These generalize in a straightforward way to deflation approaches, in which the rows of the demixing matrix are computed sequentially. However, in symmetrical orthogonalization, all the row vectors are rotated in parallel and it is not obvious how the convergence is affected. This algorithm was analyzed here, showing local quadratic convergence to the correct solution with a generic cost function. For the kurtosis cost function, the convergence is cubic. Thus, the one-unit behavior generalizes to the parallel case as well.

It is notable that in these results the chosen nonlinearity in the cost function has very little effect on the behavior: The algorithm will find the sources as either local minima or maxima of the cost function. The score function of the sources, which is optimal for maximum likelihood and minimum entropy criteria for ICA, seems not to offer any advantages for the speed of convergence. However, as shown elsewhere [10], [15], the true score function does minimize the residual error in the finite sample case.

References

[1] S.-I. Amari, T.-P. Chen, and A. Cichocki, "Stability analysis of adaptive blind source separation," *Neural Netw.*, vol. 10, no. 8, pp. 1345–1351, 1997.

[2] J.-F. Cardoso, "Eigen-structure of the fourth-order cumulant tensor with application to the blind source separation problem," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP'90)*, Albuquerque, NM, 1990, pp. 2655–2658.

[3] A. Cichocki and S. Amari, *Adaptive Blind Signal and Image Processing*. New York: Wiley, 2002.

[4] P. Comon, "Independent component analysis—A new concept?," *Signal Process.*, vol. 36, pp. 287–314, 1994.

[5] N. Delfosse and P. Loubaton, "Adaptive blind separation of independent sources: A deflation approach," *Signal Process.*, vol. 45, pp. 59–83, 1995.

[6] D. L. Donoho, "On minimum entropy deconvolution," in *In Applied Time Series Analysis II*. New York: Academic, 1981, pp. 565–608.

[7] S. C. Douglas, "On the convergence behavior of the FastICA algorithm," in *Proc. 4th Int. Symp. Independent Component Anal. (ICA) and Blind Source Separation (BSS)*, Nara, Japan, Apr. 1–4, 2003, pp. 409–414.

[8] X. Giannakopoulos, J. Karhunen, and E. Oja, "Experimental comparison of neural algorithms for independent component analysis and blind separation," *Int. J. Neural Syst.*, vol. 9, no. 2, pp. 651–656, 1999.

[9] M. Girolami, Ed., *Advances in Independent Component Analysis*. London, U.K.: Springer-Verlag, 2000.

[10] A. Hyvärinen, "One-unit contrast functions for independent component analysis: A statistical analysis," in *Proc. IEEE Workshop Neural Netw. Signal Process.*, Amelia Island, FL, 1997, pp. 388–397.

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

[12] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*. New York: Wiley, 2001.

[13] A. Hyvärinen and E. Oja, "A fast fixed-point algorithm for independent component analysis," *Neural Comput.*, vol. 9, no. 7, pp. 1483–1492, 1997.

[14] A. Hyvärinen and E. Oja, "One-unit learning rules for independent component analysis," in *In Advances in Neural Information Processing Systems*. Cambrdige, MA: MIT, 1997, vol. 9, pp. 480–486.

[15] Z. Koldovsky, P. Tichavsky, and E. Oja, "Cramer-Rao lower bound for linear independent component analysis," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP'05)*, Philadelphia, PA, Mar. 18–23, 2005.

[16] Z. Koldovsky, P. Tichavsky, and E. Oja, "Efficient variant of algorithm FastICA for independent component analysis attaining the Cramer–Rao lower bound," *IEEE Trans. Neural Netw.*, vol. 17, no. 5, pp. 1265–1277, Sep. 2006.

[17] J.-L. Lacoume and P. Ruiz, "Sources identification: A solution based on cumulants," presented at the IEEE Acoust., Speech, Signal Process. Workshop, Minneapolis, MN, 1988.

[18] Z. Y. Liu, K. C. Chiu, and L. Xu, "One-bit-matching conjecture for independent component analysis," *Neural Comput.*, vol. 16, pp. 383–399, 2004.

[19] J. Ma, Z. Liu, and L. Xu, "A further result on the ICA one-bit-matching conjecture," *Neural Comput.*, vol. 17, pp. 331–334, 2005.

[20] C. Nikias and A. Petropulu, *Higher-Order Spectral Analysis—A Nonlinear Signal Processing Framework*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[21] E. Oja, "Convergence of the symmetrical FastICA algorithm," in *Proc. Int. Conf. Neural Inf. Process. (ICONIP'02)*, Singapore, Nov. 18–22, 2002, pp. 1368–1372.

[22] P. A. Regalia and E. Kofidis, "Monotonic convergence of fixed-point algorithms for ICA," *IEEE Trans. Neural Netw.*, vol. 14, no. 4, pp. 943–949, Jul. 2003.

[23] O. Shalvi and E. Weinstein, "New criteria for blind deconvolution of nonminimum phase systems (channels)," *IEEE Trans. Inf. Theory*, vol. 36, no. 2, pp. 312–321, Mar. 1990.

[24] P. Tichavsky, Z. Koldovsky, and E. Oja, "Performance analysis of the FastICA algorithm and Cramer-Rao bounds for linear independent component analysis," *IEEE Trans. Signal Process.*, vol. 54, no. 4, pp. 1189–1203, Apr. 2006.

[25] L. Zhang, A. Cichocki, and S. Amari, "Self-adaptive blind source separation based on activation functions adaptation," *IEEE Trans. Neural Netw.*, vol. 15, no. 2, pp. 233–244, Mar. 2004.

Erkki Oja (S'75–M'78–SM'90–F'00) received the Dr.Sc. degree from Helsinki University of Technology, Helsinki, Finland, in 1977.

He is Director of the Adaptive Informatics Research Center and the Professor of Computer Science at the Laboratory of Computer and Information Science, Helsinki University of Technology. He has been a Research Associate at Brown University, Providence, RI, and a Visiting Professor at the Tokyo Institute of Technology, Tokyo, Japan. He is the author and coauthor of more than 280 articles and book chapters on pattern recognition, computer vision, and neural computing, and three books: *Subspace Methods of Pattern Recognition* (New York: Research Studies Press and Wiley, 1983), which has been translated into Chinese and Japanese; *Kohonen Maps* (Amsterdam, The Netherlands: Elsevier, 1999), and *Independent Component Analysis* (New York: Wiley, 2001). His research interests are in the study of principal component and independent component analysis, self-organization, statistical pattern recognition, and applying artificial neural networks to computer vision and signal processing.

Prof. Oja is a member of the Finnish Academy of Sciences, Founding Fellow of the International Association of Pattern Recognition (IAPR), and Past President of the European Neural Network Society (ENNS). He is also a member of the editorial boards of several journals and has been in the program committees of several recent conferences including the International Conference on Artificial Neural Networks (ICANN), International Joint Conference on Neural Networks (IJCNN), and International Conference on Neural Information Processing (ICONIP). He is the recipient of the 2006 IEEE Computational Intelligence Society Neural Networks Pioneer Award.

Zhijian Yuan received the M.S. degree in mathematics from Beijing Institute of Technology, Beijing, China, in 1992 and the Licenciate degree in mathematics from Helsinki University of Technology, Helsinki, Finland, in 2002, where he is currently working towards the Ph.D. degree at the Laboratory of Information and Computer Science.