Blind Separation of Signals with Mixed Kurtosis Signs Using Threshold Activation Functions

Heinz Mathis, Thomas P. von Hoff, and Marcel Joho

***Abstract*—A parameterized activation function in the form of an adaptive threshold for a single-layer neural network, which separates a mixture of signals with any distribution (except for Gaussian), is introduced. This activation function is particularly simple to implement, since it neither uses hyperbolic nor polynomial functions, unlike most other nonlinear functions used for blind separation. For some specific distributions, the stable region of the threshold parameter is derived, and optimal values for best separation performance are given. If the threshold parameter is made adaptive during the separation process, the successful separation of signals whose distribution is unknown is demonstrated and compared against other known methods.**

***Index Terms*—Adaptive activation function, blind separation, mixed kurtosis, threshold function.**

I. Introduction

Blind signal separation using higher-order statistics either explicitly or implicitly has attracted many researchers whose main goal is to separate a set of mixed signals as fast as possible with the smallest residual mixing. Most approaches require complete or at least some knowledge of the source distributions. If sources of different distributions are mixed, such techniques may fail to work. In that case, an approach which is independent of the source distribution must be sought.

Throughout this paper we assume a linear mixing and separation process, where the measured signals $\boldsymbol{x} = [x_1, \ldots, x_{M_s}]^T$ to be processed are linear combinations of the original source signals $\boldsymbol{s} = [s_1, \ldots, s_{M_s}]^T$, weighted by scalars, which are the elements of the mixing matrix $\boldsymbol{A}$. M_s denotes the number of sources as well as the number of sensors. Recovery of the signals is carried out by a blind adaptive algorithm adjusting the coefficients of the separation matrix $\boldsymbol{W}$. The output of the algorithm is therefore

$$\boldsymbol{u} = [u_1, \ldots, u_{M_s}]^T = \boldsymbol{W}\boldsymbol{x} = \boldsymbol{W}\boldsymbol{A}\boldsymbol{s} = \boldsymbol{P}\boldsymbol{s}. \tag{1}$$

In order to successfully separate the signals, $\boldsymbol{P} = \boldsymbol{W}\boldsymbol{A}$ should approximate as closely as possible a scaled permutation matrix. A possible learning equation for the separation matrix $\boldsymbol{W}$ results from the ML estimator [1] and applying the natural gradient [2]

$$\boldsymbol{W}_{t+1} = \boldsymbol{W}_t + \mu \left(\boldsymbol{I} - \boldsymbol{g}(\boldsymbol{u})\boldsymbol{u}^T\right) \boldsymbol{W}_t \tag{2}$$

where

μ	learning rate parameter;
$\boldsymbol{I}$	identity matrix;
$\boldsymbol{g}(\boldsymbol{u})$	suitable nonlinearity.

The separation process can be modeled as a single-layer neural network with an equal number of input and output nodes, where the coefficients w_{ij} of the separation matrix $\boldsymbol{W}$ are simply the weights from the input to the output nodes. The activation function at the output nodes is used for the training mode only, while the problem itself is linear (since the mixing is a linear process, its inverse operation is linear, too), so that for the successful separation of a linear mixture, a linear combination of the available input signal is adequate.

II. The Threshold Activation Function

The activation function plays a central role in blind signal separation. Its nature is defined by the objective or contrast function. The maximum-likelihood approach [1] leads to

$$g_i(u_i) = -\frac{\partial \log p_S(u_i)}{\partial u_i} = -\frac{p_S'(u_i)}{p_S(u_i)}, \qquad i = 1, \ldots, M_s \tag{3}$$

the so-called *score function*, where $p_S(u_i)$ and $p_S'(u_i)$ are the probability density function (pdf) and its derivative, respectively, of the source signals. In the following, the range of i will be assumed as that given in (3) if not indicated otherwise. From (3) it can be seen, that super-Gaussian signals typically have sigmoidal activation functions such as $\mathrm{sign}\,(.)$ or $\tanh(.)$, whereas sub-Gaussian signals can be separated using an activation function of the form $g_i(u_i) = a \cdot |u_i|^{p-1} u_i$, $p > 1$. A very simple activation function for the separation of sub-Gaussian signals has been derived in [3] starting from the pdf of a generalized Gaussian signal

$$p_S(u_i) = \frac{\alpha}{2\beta\Gamma\left(\dfrac{1}{\alpha}\right)} e^{-(|u_i|/\beta)^\alpha} \tag{4}$$

where $\alpha > 2$ for sub-Gaussian signals. Differentiating (4) with respect to u_i leads to

$$p_S'(u_i) = -\alpha \left(\frac{|u_i|}{\beta}\right)^{\alpha-1} \frac{\mathrm{sign}\,(u_i)}{\beta} \frac{\alpha}{2\beta\Gamma\left(\dfrac{1}{\alpha}\right)} e^{-(|u_i|/\beta)^\alpha}. \tag{5}$$

If we divide (5) by (4) and flip the sign we get

$$\begin{aligned} g_i(u_i) &= -\frac{p_S'(u_i)}{p_S(u_i)} = \alpha \left(\frac{|u_i|}{\beta}\right)^{\alpha-1} \frac{\mathrm{sign}\,(u_i)}{\beta} \\ &= \frac{\alpha}{\beta^\alpha} |u_i|^{\alpha-1} \mathrm{sign}\,(u_i). \end{aligned} \tag{6}$$

Manuscript received January 18, 2000; revised November 29, 2000 and February 7, 2001.

The authors are with the Signal and Information Processing Laboratory, Swiss Federal Institute of Technology, ETH Zurich, Switzerland (e-mail: mathis@isi.ee.ethz.ch; vonhoff@isi.ee.ethz.ch; joho@isi.ee.ethz.ch).

Publisher Item Identifier S 1045-9227(01)03566-4.

For unit variance, we can find β from the general expression for the nth-order moment of a generalized Gaussian signal [4]

$$E\left\{|X|^n\right\} = \frac{\Gamma\left(\dfrac{n+1}{\alpha}\right)}{\Gamma\left(\dfrac{1}{\alpha}\right)}\beta^n. \tag{7}$$

$\Gamma(.)$ is the gamma function given by $\Gamma(a) = \int_0^\infty x^{a-1} \exp(-x)\,dx$. For $n = 2$, (7) gives

$$E\left\{|X|^2\right\} = \frac{\Gamma\left(\dfrac{3}{\alpha}\right)}{\Gamma\left(\dfrac{1}{\alpha}\right)}\beta^2 \tag{8}$$

or, for unit variance, we have

$$\beta = \sqrt{\frac{\Gamma\left(\dfrac{1}{\alpha}\right)}{\Gamma\left(\dfrac{3}{\alpha}\right)}}. \tag{9}$$

Inserting this value for β into (6) yields the activation function

$$g_i(u_i) = \alpha\left(\frac{\Gamma\left(\dfrac{3}{\alpha}\right)}{\Gamma\left(\dfrac{1}{\alpha}\right)}\right)^{\alpha/2} \operatorname{sign}(u_i)\cdot|u_i|^{\alpha-1}. \tag{10}$$

Apart from a scaling constant, this nonlinearity has also been derived in [5].

Using $\Gamma(x)\cdot\Gamma(1-x) = \pi/\sin(\pi x)$ (see, for example, [6]) leads to

$$g_i(u_i) = \alpha\left(\frac{\dfrac{\pi}{\sin\dfrac{3\pi}{\alpha}}}{\dfrac{\pi}{\sin\dfrac{\pi}{\alpha}}}\cdot\frac{\Gamma\left(1-\dfrac{\pi}{\alpha}\right)}{\Gamma\left(1-\dfrac{3\pi}{\alpha}\right)}\right)^{\alpha/2} |u_i|^{\alpha-1}\operatorname{sign}(u_i). \tag{11}$$

Both terms $\Gamma(1-(\pi/\alpha))$ and $\Gamma(1-(3\pi/\alpha))$ are close to $\Gamma(1) = 1$ for large values of α, so that simplification of (11) yields

$$g_i(u_i)|_{\alpha\gg 1} \approx \alpha\left(\frac{\sin\left(\dfrac{\pi}{\alpha}\right)}{\sin\left(\dfrac{3\pi}{\alpha}\right)}\right)^{\alpha/2} \operatorname{sign}(u_i)\cdot|u_i|^{\alpha-1}. \tag{12}$$

The first term of the Taylor expansion of a sine function for a small argument is just the argument itself, leading to

$$g_i(u_i)|_{\alpha\gg 1} \approx \alpha\left(\frac{1}{3}\right)^{\alpha/2} \operatorname{sign}(u_i)\cdot|u_i|^{\alpha-1} = \alpha\frac{1}{u_i}\left(\frac{u_i^2}{3}\right)^{\alpha/2}. \tag{13}$$

We are now interested in the form of $g_i(.)$ as α approaches infinity, in which case (4) corresponds to a uniform distribution. As a consequence of the behavior of $\lim_{b\to\infty} a^b$ depending on a being less or greater than one, we can write the *threshold activation function* as

$$\lim_{\alpha\to\infty} g_i(u_i) = \begin{cases} 0, & |u_i| < \sqrt{3} \\ \infty\cdot\operatorname{sign}(u_i), & |u_i| \geq \sqrt{3}. \end{cases} \tag{14}$$

The infinite gain in (14) will of course cause convergence problems for a finite learning rate parameter μ. The gain can therefore be traded off against a lower threshold ϑ for a specified output power. If we aim at unity output power, we need to scale the activation function. By the scaling invariance, which is an inherent property of blind signal separation, it is impossible to recover the original power of the source signals without further knowledge. When the original power of the sources is unknown, it is reasonable to normalize the power after the separation matrix to $E\{\boldsymbol{uu}^T\} = \boldsymbol{I}$. To this end we need to scale $g_i(u_i)$ such that

$$\int_{-\infty}^{\infty} p_S(u_i)g_i(u_i)u_i\,du_i = 1 \tag{15}$$

if $p_S(.)$ is a source distribution with unit variance $\sigma_S^2 = 1$. Note that for the score function (3) of almost all distributions, (15) is satisfied without further scaling. Solving (15) for the uniform distribution and a given threshold ϑ results in a finite gain of

$$a = \frac{2\sqrt{3}}{3-\vartheta^2} \tag{16}$$

for $0 \leq \vartheta < \sqrt{3}$. The resulting threshold activation function is

$$g_i(u_i) = \begin{cases} 0, & |u_i| < \vartheta \\ a\operatorname{sign}(u_i), & |u_i| \geq \vartheta \end{cases} \tag{17}$$

and is depicted in Fig. 1. Note that a is always positive for the assigned range of ϑ.

In [3], the application of (17) to sub-Gaussian pdfs other than the uniform distribution (e.g., M-PAM or M-QAM) was demonstrated with the possible need for adjusting gain a for normalized output power. By experiment, a good value for the threshold was found to be $\vartheta = 1.5$. On the other hand, for the Laplacian distribution, which is an example of a super-Gaussian distribution and can be written in the form of (4) with $\alpha = 1$, (10) simplifies to

$$g_i(u_i) = \sqrt{2}\operatorname{sign}(u_i) \tag{18}$$

which is the same as (17) for $a = \sqrt{2}$ and $\vartheta = 0$. The signum function can hence be regarded as a threshold function with threshold ϑ set to zero.

III. Stability Analysis

In [7] it was shown that a necessary and sufficient stability criterion for the separation of the ith and jth $(i \neq j)$ source is that the eigenvalues of the Hessian sub-matrix

$$\boldsymbol{\Xi} = \begin{bmatrix} E\left\{g_i'(s_i)\right\}E\left\{s_j^2\right\} & E\left\{g_i(s_i)s_i\right\} \\ E\left\{g_j(s_j)s_j\right\} & E\left\{g_j'(s_j)\right\}E\left\{s_i^2\right\} \end{bmatrix} \tag{19}$$

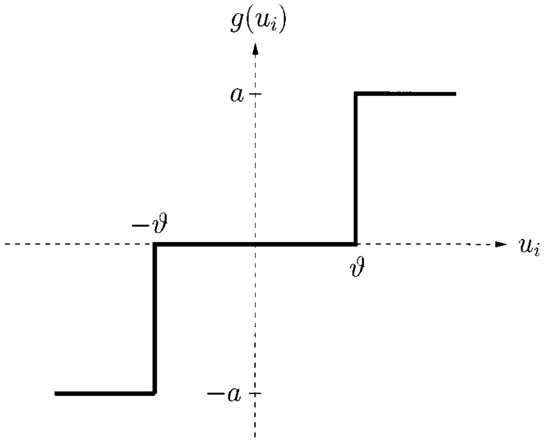

Fig. 1. Threshold activation function with parameter ϑ.

are positive. For equal source distributions and activation functions, the eigenvalues of $\boldsymbol{\Xi}$ are given by

$$\kappa_+ = E\left\{g_i'(s_i)\right\} E\left\{s_i^2\right\} + E\left\{g_i(s_i)s_i\right\} \tag{20}$$

$$\kappa_- = E\left\{g_i'(s_i)\right\} E\left\{s_i^2\right\} - E\left\{g_i(s_i)s_i\right\}. \tag{21}$$

Note that this analysis concerns local stability, hence the statistics of $\boldsymbol{s}$ and $\boldsymbol{u}$ are interchangeable near convergence. Similar analyzes have been carried out in [8] and [9]. The source power is assumed to be normalized to one, i.e., $E\{s_i^2\} = 1$, and we scale the activation function such that $E\{g_i(s_i)s_i\} = 1$. Although the threshold function is not differentiable at $s_i = \pm\vartheta$, we can derive $E\{g_i'(s_i)\}$ by the use of δ-functions and assuming a symmetric distribution

$$\begin{aligned} E\left\{g_i'(u_i)\right\} &= \int_{-\infty}^{\infty} p_S(u_i) g_i'(u_i)\,du_i \\ &= \int_{-\infty}^{\infty} p_S(u_i) a(\delta(u_i+\vartheta) + \delta(u_i-\vartheta))\,du_i \\ &= 2a \cdot p_S(\vartheta). \end{aligned} \tag{22}$$

Equations (20) and (21) can therefore be written as

$$\kappa_+ = 2a \cdot p_S(\vartheta) + 1 \tag{23}$$

$$\kappa_- = 2a \cdot p_S(\vartheta) - 1. \tag{24}$$

For a positive scaling factor a, (23) is always positive. To make (24) positive, we must ensure that $2a \cdot p_S(\vartheta) > 1$ by suitable choice of ϑ. For the threshold activation function and symmetric distributions, (15) can be written as

$$2a \int_{\vartheta}^{\infty} p_S(u_i) u_i\,du_i = 1 \tag{25}$$

or, if solved for the scaling factor

$$a = \frac{1}{2\int_{\vartheta}^{\infty} p_S(u_i) u_i\,du_i}. \tag{26}$$

Thus, the stability condition for the threshold activation function results in

$$\frac{p_S(\vartheta)}{\int_{\vartheta}^{\infty} p_S(u_i) u_i\,du_i} > 1. \tag{27}$$

Equation (27) defines a stable region for ϑ depending on the source distribution. In order to find the optimal values for ϑ (in the sense of quality of separation), we have to minimize the term $\gamma_+/\kappa_+ + \gamma_-/\kappa_-$ [7], with κ_+ and κ_- defined by Equations (20) and (21), respectively, and

$$\gamma_+ = E\left\{g_i^2(s_i)\right\} E\left\{s_i^2\right\} + \left(E\left\{g_i(s_i)s_i\right\}\right)^2 \tag{28}$$

$$\gamma_- = E\left\{g_i^2(s_i)\right\} E\left\{s_i^2\right\} - \left(E\left\{g_i(s_i)s_i\right\}\right)^2. \tag{29}$$

For the threshold activation function we can write

$$E\left\{g_i^2(s_i)\right\} = 2a^2 \int_{\vartheta}^{\infty} p_S(u_i)\,du_i. \tag{30}$$

We then get

$$\begin{aligned} \gamma_+/\kappa_+ + \gamma_-/\kappa_- &= \frac{2a^2 \int_{\vartheta}^{\infty} p_S(u_i)\,du_i + 1}{2ap_S(\vartheta) + 1} \\ &\quad + \frac{2a^2 \int_{\vartheta}^{\infty} p_S(u_i)\,du_i - 1}{2ap_S(\vartheta) - 1} \\ &= \frac{8a^3 p_S(\vartheta) \int_{\vartheta}^{\infty} p_S(u_i)\,du_i - 2}{4a^2 p_S^2(\vartheta) - 1} \end{aligned} \tag{31}$$

so the optimal value for the threshold is

$$\vartheta_{\text{opt}} = \arg\min_{\vartheta} \frac{4a^3 p_S(\vartheta) \int_{\vartheta}^{\infty} p_S(u_i)\,du_i - 1}{4a^2 p_S^2(\vartheta) - 1}. \tag{32}$$

Fig. 2 shows the stability region and the optimal values of ϑ of the threshold activation function for the generalized Gaussian distribution. For signals with $\alpha \leq 1$, the stable range for ϑ is roughly between zero and 0.5 with an optimal value of zero. For this threshold value, the threshold function is the true score function for the Laplacian distribution. Interestingly, the optimal value for $1 < \alpha < 2$ is slightly higher than zero. At the other extreme is the uniform distribution with $\alpha = \infty$. While the upper limit of the stability range is $\sqrt{3}$, the lower limit is one. The optimal value approaches $\sqrt{3}$ as the pdf gets close to the uniform distribution. Note that only for the uniform distribution the threshold function is the score function (except for the finite gain).

Generally speaking, for sub-Gaussian signals, the stable range is between 1 and $\sqrt{3}$. By decreasing α we approach the normal distribution ($\alpha = 2$), which clearly poses a singularity in the stability plot. This can be seen by the broken stability regions around $\alpha = 2$.

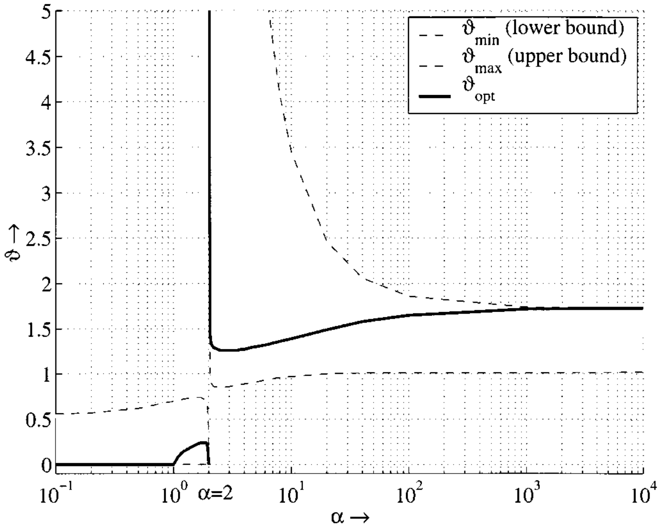

Fig. 2. Stable range of threshold ϑ (shaded area) as a function of the generalized Gaussian parameter α.

IV. Blind Separation of Arbitrary Sources

A. Known Methods

In practice, the distributions of the sources are often identical, since the nature of their origin is related. In that case, the activation function will be the same for all output nodes. However, we may find the situation where some sources have different distributions, possibly with a different sign of their respective kurtoses. If in a mixture some sources are sub-Gaussian (negative kurtosis) and some super-Gaussian (positive kurtosis) distributed, the appropriate activation function might be chosen in advance, as long as the number of sub- and super-Gaussian sources is known. Doing so, the neural network is deprived of some degree of freedom, due to the restriction of permutation within the group of equal kurtosis sign. In other words, once an activation function is chosen, only a signal with the appropriate kurtosis sign can be separated at that specific output node. Other signals are forced away to output nodes with the appropriate activation function. The learning speed can be greatly accelerated by letting the network choose its permutation closest to some initial mixing condition. This can be achieved by an adaptive activation function. If the number of sub-Gaussian and the number of super-Gaussian sources is unknown, adaptive activation functions are a necessity.

Douglas *et al.* [10] switch between two nonlinearities, namely

$$g_N(u_i) = u_i^3 \quad \text{and} \quad g_P(u_i) = \tanh(10u_i) \tag{33}$$

where $g_N(.)$ and $g_P(.)$ separate sub- and super-Gaussian signals, respectively. The algorithm does not try to normalize its output power regardless of the distribution. The stability condition [7] is therefore

$$\frac{E\{g_i'(u_i)\}\,E\{u_i^2\}}{E\{g_i(u_i)u_i\}} \cdot \frac{E\{g_j'(u_j)\}\,E\{u_j^2\}}{E\{g_j(u_j)u_j\}} > 1 \tag{34}$$

for any two outputs $i \neq j$. A stronger and thus sufficient condition for (34) to hold is

$$\frac{E\{g_i'(u_i)\}\,E\{u_i^2\}}{E\{g_i(u_i)u_i\}} > 1 \tag{35}$$

for each output i. A sufficient stability condition for the nonlinearities in (33) is therefore

$$E\{g_i'(u_i)\}\,E\{u_i^2\} - E\{g_i(u_i)u_i\} > 0. \tag{36}$$

The left-hand side of (36) is constantly evaluated for the two nonlinearities $g_N(.)$ and $g_P(.)$. The larger value decides which activation function is applied.

Similarly, Lee *et al.* [11] present an extended Infomax algorithm, where the learning equation for the separation matrix is formulated as

$$\boldsymbol{W}_{t+1} = \boldsymbol{W}_t + \mu\left(\boldsymbol{I} - \boldsymbol{K}\tanh(\boldsymbol{u})\boldsymbol{u}^T - \boldsymbol{u}\boldsymbol{u}^T\right)\boldsymbol{W}_t \tag{37}$$

with $\boldsymbol{K} = \operatorname{diag}[k_1, \ldots, k_{M_s}]^T$ being a diagonal matrix of signs. k_i is positive for a super-Gaussian and negative for a sub-Gaussian signal, respectively. The activation function can then be written as

$$g_i(u_i) = k_i \tanh(u_i) + u_i. \tag{38}$$

If the distributions are unknown, the sign might be switched according to a kurtosis estimation at the output node or some parameter expressing the stability of the nonlinearity currently used as the activation function. Similarly to (36) it follows:

$$k_i\left(\left(1 - E\left\{\tanh^2(u_i)\right\}\right)E\left\{u_i^2\right\} - E\left\{\tanh(u_i)u_i\right\}\right) > 0. \tag{39}$$

By choosing k_i the same sign as the rest of (39), the algorithm is stabilized. Thus, the sign k_i must be learned as we go along

$$k_i = \operatorname{sign}\left(\left(1 - E\left\{\tanh^2(u_i)\right\}\right)E\left\{u_i^2\right\} - E\left\{\tanh(u_i)u_i\right\}\right) \tag{40}$$

or in more compact form by using the substitution $\operatorname{sech}^2(.) = 1 - \tanh^2(.)$

$$k_i = \operatorname{sign}\left(\left(E\left\{\operatorname{sech}^2(u_i)\right\}\right)E\left\{u_i^2\right\} - E\left\{\tanh(u_i)u_i\right\}\right). \tag{41}$$

Again, output powers are not normalized, and depend on the source distributions.

Yet another method was presented by Cichocki *et al.* [12], where basically two nonlinearities $f(.)$ and $g(.)$ are used in the update expression

$$\boldsymbol{W}_{t+1} = \boldsymbol{W}_t + \mu\left(\boldsymbol{I} - \boldsymbol{f}(\boldsymbol{u})\boldsymbol{g}\left(\boldsymbol{u}^T\right)\right)\boldsymbol{W}_t \tag{42}$$

according to

$$f_i(u_i) = \begin{cases} \tanh(10u_i), & \hat{\kappa}_4 > 0.1 \\ u_i, & \hat{\kappa}_4 < 0.1 \end{cases} \tag{43}$$

$$g_i(u_i) = \begin{cases} \tanh(10u_i), & \hat{\kappa}_4 < -0.1 \\ u_i, & \hat{\kappa}_4 > -0.1 \end{cases} \tag{44}$$

where $\hat{\kappa}_4$ is the sample kurtosis.

B. Adaptive Threshold Activation Function

Since we know that any non-Gaussian distribution can be separated by the threshold function with either $\vartheta = 0$ or $\vartheta \approx 1.5$, we can set up a neural network in which the learning equation for the separation matrix is given by (2) with

$$g_i(u_i) = \begin{cases} 0, & |u_i| < \vartheta_i \\ a_i \operatorname{sign}(u_i), & |u_i| \geq \vartheta_i. \end{cases} \tag{45}$$

Each threshold ϑ_i is chosen from {0, 1.5} as that value which maximizes the right-hand side of (21) with $g_i(.)$ of (45). The use of the stability equation to switch between the two threshold values has two important disadvantages. First, the value of $p_S(\vartheta)$ is difficult to work out since the function $p_S(.)$ is unknown. Second, although the threshold function successfully separates discrete distributions, $p_S(\vartheta)$ is generally zero for discrete distributions as used in data communications, making the switching criterion invalid.

A better alternative is to train the threshold vector $\boldsymbol{\vartheta} = [\vartheta_1, \ldots, \vartheta_{M_s}]^T$ according to

$$\boldsymbol{\vartheta}_{t+1} = \boldsymbol{\vartheta}_t - \mu_\vartheta \hat{\boldsymbol{\kappa}}_t \tag{46}$$

where μ_ϑ is a properly chosen positive constant, and $\hat{\boldsymbol{\kappa}}_t = [\hat{\kappa}_{1,t}, \ldots, \hat{\kappa}_{M_s,t}]^T$ is an estimate of the output kurtoses of the vector $\boldsymbol{u}$ at sample time t. Additionally, the elements of $\boldsymbol{\vartheta}_{t+1}$ are clipped at zero and 1.5 to keep them inside a meaningful region.

C. Output Normalization

In the analysis of the stability we have shown that the scaling factors in the vector $\boldsymbol{a} = [a_1, \ldots, a_{M_s}]^T$ have to be chosen according to (25) in order to obtain output signals with unit variance. In an environment where the probability distributions are given, (25) can be evaluated off-line and $\boldsymbol{a}$ is thus fixed during learning and separation. If the distributions are unknown, however, $\boldsymbol{a}$ itself has to be found during the learning mode. To this end, we note that for unimodal, symmetric distributions, a_i is a monotonic decreasing function of the standard deviation σ_i of the ith output. Vice versa, σ_i can be written as $\sigma_i = f_i(a_i)$, where $f_i(.)$ is a monotonic decreasing function for $a_i > 0$, hence

$$\frac{\partial f_i(a_i)}{\partial a_i} < 0. \tag{47}$$

The exact course of $f_i(.)$ depends on the pdf of the ith source. For convenience we denote $\boldsymbol{f}(.) = [f_1(.), \ldots, f_{M_s}(.)]^T$. We define our error function $\boldsymbol{e} = [e_1, \ldots, e_{M_s}]^T$ by the deviation from unit variance

$$e_i = 1 - \hat{\sigma}_i^2 \tag{48}$$

and its sum of squares as the cost function

$$J(\boldsymbol{a}) = \boldsymbol{e}^T\boldsymbol{e} = \sum_{i=1}^{M_s} e_i^2 = \sum_{i=1}^{M_s} \left(1 - \hat{\sigma}_i^2\right)^2. \tag{49}$$

$\hat{\sigma}_i^2$ denotes the estimation of the output power σ_i^2. The derivative of the cost function $J(\boldsymbol{a})$ with respect to the gain $\boldsymbol{a}$ is

$$\begin{aligned} \nabla_{\boldsymbol{a}} J(\boldsymbol{a}) &= 2\boldsymbol{e} \odot \left[\frac{\partial e_1}{\partial a_1}, \ldots, \frac{\partial e_{M_s}}{\partial a_{M_s}}\right]^T \\ &= -4\left[\left(1 - \hat{\sigma}_1^2\right) \hat{\sigma}_1 \frac{\partial f_1(a_1)}{\partial a_1}, \ldots, \right. \\ &\qquad \left. \left(1 - \hat{\sigma}_{M_s}^2\right) \hat{\sigma}_{M_s} \frac{\partial f_{M_s}(a_{M_s})}{\partial a_{M_s}}\right]^T. \end{aligned} \tag{50}$$

$\odot$ denotes the elementwise multiplication of two vectors. We can now develop a stochastic gradient algorithm to train the gain vector

$$\boldsymbol{a}_{t+1} = \boldsymbol{a}_t - \mu_a \nabla_{\boldsymbol{a}} J(\boldsymbol{a}_t). \tag{51}$$

Using (47) and the fact that $\hat{\sigma}_i > 0$, we can incorporate $\hat{\sigma}_i$ and $\partial f_i(a_i)/\partial a_i$ into a different learning rate parameter μ_a and write

$$\boldsymbol{a}_{t+1} = \boldsymbol{a}_t - \mu_a \left(\mathbf{1} - \hat{\boldsymbol{\sigma}}_t^2\right) \tag{52}$$

with $\mathbf{1}$ being a vector of ones and $\hat{\boldsymbol{\sigma}}_t^2 = [\hat{\sigma}_{1,t}^2, \ldots, \hat{\sigma}_{M_s,t}^2]^T$ the vector of power estimates, respectively. Equation (52) is a simple automatic gain control (AGC) algorithm, which normalizes the output powers of the separation process. It runs along with the training of $\boldsymbol{W}$ and $\boldsymbol{\vartheta}$. Alternately, the normalization can of course be performed by a separate AGC stage after the separation process. This is for example necessary if the mixture contains binary sources. It is straightforward to see that a normalized source with symbol values ±1 produces zero output after the threshold function with $\vartheta = 1.5$.

In summary, the adaptive threshold activation function algorithm is given as follows.

Adaptive Threshold Activation Function Algorithm

Initialization:

$$\boldsymbol{W}_{t=0} = \boldsymbol{I} \qquad (53)$$

$$\boldsymbol{\vartheta}_{t=0} = 0.75 \cdot \mathbf{1} \qquad (54)$$

$$\boldsymbol{a}_{t=0} = 2 \cdot \mathbf{1} \qquad (55)$$

Separation:

$$\boldsymbol{u}_t = \boldsymbol{W}_t \boldsymbol{x}_t \qquad (56)$$

Weights learning:

$$\boldsymbol{W}_{t+1} = \boldsymbol{W}_t + \mu \left(\boldsymbol{I} - \boldsymbol{g}(\boldsymbol{u}_t)\boldsymbol{u}_t^T\right) \boldsymbol{W}_t \qquad (57)$$

with

$$g_i(u_i) = \begin{cases} 0, & |u_i| < \vartheta_i \\ a_i \operatorname{sign}(u_i), & |u_i| \geq \vartheta_i \end{cases} \qquad (58)$$

Statistics estimation:

$$\hat{\sigma}_{i,\,t}^2 = \frac{1}{L}\sum_{\tau=0}^{L-1} u_{i,\,t-\tau}^2 \qquad (59)$$

$$\hat{\kappa}_{i,\,t} = \frac{L\sum_{\tau=0}^{L-1} u_{i,\,t-\tau}^4}{\hat{\sigma}_{i,\,t}^4} - 3 \qquad (60)$$

Activation function learning:

$$\vartheta_{i,\,t+1} = \begin{cases} 0, & \vartheta_{i,\,t} - \mu_\vartheta \hat{\kappa}_{i,\,t} < 0 \\ 1.5, & \vartheta_{i,\,t} - \mu_\vartheta \hat{\kappa}_{i,\,t} > 1.5 \\ \vartheta_{i,\,t} - \mu_\vartheta \hat{\kappa}_{i,\,t}, & \text{otherwise} \end{cases} \qquad (61)$$

$$\boldsymbol{a}_{t+1} = \boldsymbol{a}_t - \mu_a \left(\mathbf{1} - \hat{\boldsymbol{\sigma}}_t^2\right). \qquad (62)$$

The structure of the resulting neural network is depicted in Fig. 3. Note that the training of the activation functions involves adapting the thresholds as well as the scaling factors.

D. Advantages of the New Method

The adaptive threshold activation function offers three main advantages over existing methods. First, it is not only easy to implement on a DSP, but would also be considerably simpler to implement in hardware as compared to polynomial nonlinearities or hyperbolic functions used in other methods, which either need look-up tables or polynomial approximations, since the set of possible output values of the threshold function only contains three values, $\pm a$ and zero. The threshold operation can thus be easily implemented by two comparators only. Second, the threshold nonlinearity can be stabilized for distributions which are impossible to separate by either $g(u) = u^3$ or $g(u) = \tanh(u)$, see also [13]. Third, normalization of the output signals is achieved independent of the distributions, which is not the case with any of the existing methods in [10]–[12].

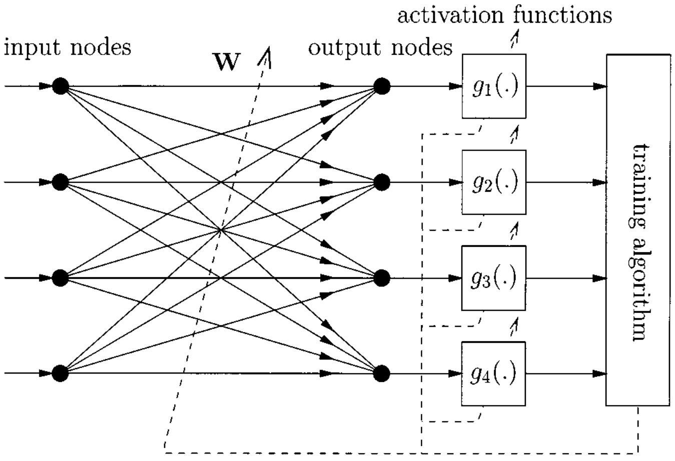

Fig. 3. Neural network with adaptive threshold activation functions.

V. Simulations

For the following simulations of the learning time analysis of blind signal separation using a single-layer neural network with an adaptive threshold device, $M_s = 10$ independent source signals were mixed by a random matrix $\boldsymbol{A}$, whose condition number is chosen $\chi(\boldsymbol{A}) = 100$ (the singular values of $\boldsymbol{A}$ are logarithmically distributed). Block processing with a block length $L = 64$ was applied. With this length the kurtosis estimation for the purpose of threshold learning is accurate enough, and inter-block memory does not offer any advantage.

In the first computer experiment we mixed three Laplacian, three uniform, three 16-PAM, and one Gaussian source. If more sources are Gaussian distributed, they can still be separated from other sources by the adaptive threshold activation function, but remain mixed among themselves, leading to a disturbed permutation matrix. This is an inherent limitation of blind separation using higher order statistics, and is usually circumvented by the restriction to at most one Gaussian source. A neural network with the adaptive threshold activation function algorithm (57)–(62) was then used to separate the signals in a block-processing manner. The learning rate parameter μ of the training was adjusted for a residual mixing of $J_{\text{ICI}}(\boldsymbol{P}) = -20$ dB, where the performance measure

$$J_{\text{ICI}}(\boldsymbol{P}) = \frac{1}{M_s}\left(\sum_{i=1}^{M_s} \frac{\sum_{k=1}^{M_s} p_{ik}^2}{\max_k p_{ik}^2}\right) - 1 \qquad (63)$$

is the average interchannel interference and is described in [14]. Practical values for the different learning parameters were $\mu = 0.0017$, $\mu_\vartheta = 0.32$, and $\mu_a = 0.05$, respectively.

Fig. 4 shows the learning process. The effect of the AGC can be observed as well as the convergence of the kurtoses of the

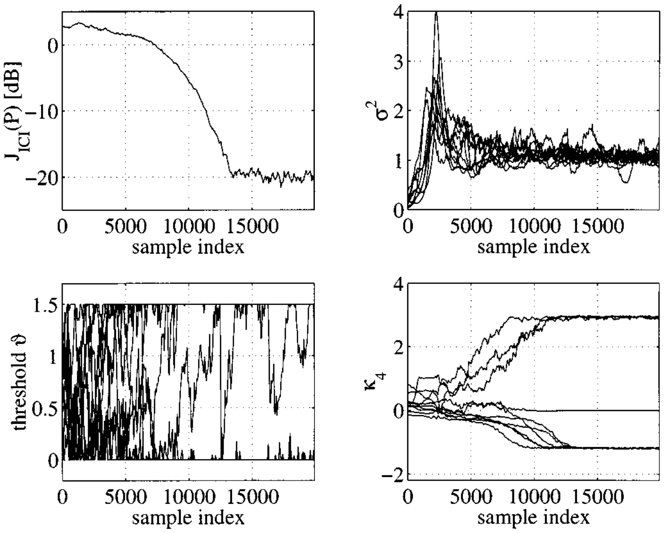

Fig. 4. Course of some statistics during the separation process of mixed-kurtosis signals. Top left: Learning curve. Top right: Output powers. Bottom left: Adaptive threshold values. Bottom right: Kurtoses of output signals.

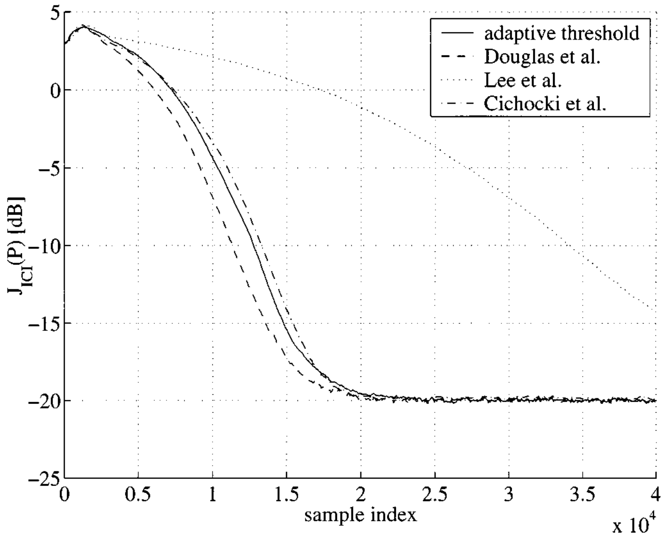

Fig. 5. Learning curves for a separation neural network with the threshold activation function.

output signals to the respective values three for Laplacian, -1.2 for uniform, -1.209 for 16-PAM, and zero for Gaussian distributions. The threshold values approach either zero or 1.5, depending on their kurtoses, and converge around 10 000 samples, except for the output node with the Gaussian distribution where the threshold value remains undecided.

In the next simulation we compared the adaptive threshold activation function algorithm with the algorithms found by Douglas *et al.* [10], Lee *et al.* [11], and Cichocki *et al.* [12]. To this end, we mixed five Laplacian and five uniform sources. The four algorithms were then run with as similar parameters as possible to allow a fair comparison. A block processing with the block size $L = 64$ was used for all algorithms. The learning rate parameters of the training algorithms were adjusted individually for a residual mixing of $J_{\text{ICI}}(\boldsymbol{P}) = -20$ dB. Averaging over 100 runs with different matrices (all with the characteristics as described above) were carried out to get typical behavior. Fig. 5 shows the separation performance for all tested algorithms. The adaptive threshold activation function algorithm, the algorithm by Douglas *et al.*, and that by Cichocki *et al.* reach the -20 dB point at exactly the same time on average, whereas the extended Infomax algorithm needs considerably more time.

VI. Conclusion

A threshold activation function for the blind separation of any non-Gaussian sources by a single-layer neural network has been derived. The threshold function is not just a simplification of polynomial functions but the true score function for the uniform distribution. The stability analysis reveals that for threshold values larger than one, separation of sub-Gaussian signals is achieved. If the threshold is reduced to zero, super-Gaussian signals can be separated. Using the kurtosis of the output signal to train the threshold parameter, a new, computationally much simpler method than those existing can be devised for the blind separation of mixed-kurtosis signals. On average it is equally fast as the fastest known, but more complex, methods and can therefore be implemented at a lower cost without sacrificing performance.

References

[1] H. H. Yang, "Serial updating rule for blind separation derived from the method of scoring," *IEEE Trans. Signal Processing*, vol. 47, p. 8, 1999.

[2] S.-I. Amari, A. Cichocki, and H. H. Yang, "A new learning algorithm for blind signal separation," *Advances Neural Inform. Processing Syst.*, vol. 8, pp. 757–763, 1996.

[3] H. Mathis, M. Joho, and G. S. Moschytz, "A simple threshold nonlinearity for blind signal separation," in *Proc. ISCAS*, vol. IV, Geneva, Switzerland, May 28–31, 2000, pp. 489–492.

[4] R. H. Lambert, "Multichannel blind deconvolution: FIR matrix algebra and separation of multipath mixtures," Ph.D. dissertation, Univ. Southern California, Los Angeles, 1996.

[5] S. Choi, A. Cichocki, and S. Amari, "Flexible independent component analysis," in *Proc. NNSP*, Amelia Island, FL, Sept. 1998, pp. 83–92.

[6] I. N. Bronshtein and K. A. Semendyayev, *Handbook of Mathematics*, 3rd ed. New York: Springer-Verlag, 1997.

[7] T. P. von Hoff, A. G. Lindgren, and A. N. Kaelin, "Transpose properties in the stability and performance of the classic adaptive algorithms for blind source separation and deconvolution," *Signal Processing*, vol. 80, no. 9, pp. 1807–1822, Aug. 2000.

[8] J.-F. Cardoso and B. H. Laheld, "Equivariant adaptive source separation," *IEEE Trans. Signal Processing*, vol. 44, pp. 3017–3030, Dec. 1996.

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

[10] S. C. Douglas, A. Cichocki, and S. Amari, "Multichannel blind separation and deconvolution of sources with arbitrary distributions," in *Proc. NNSP*, Amelia Island, FL, Sept. 1997, pp. 436–445.

[11] T.-W. Lee, M. Girolami, and T. J. Sejnowski, "Independent component analysis using an extended infomax algorithm for mixed sub-Gaussian and super-Gaussian sources," *Neural Comput.*, vol. 11, no. 2, pp. 417–441, 1999.

[12] A. Cichocki, I. Sabala, S. Choi, B. Orsier, and R. Szupiluk, "Self-adaptive independent component analysis for sub-Gaussian and super-Gaussian mixtures with an unknown number of sources and additive noise," in *Proc. NOLTA*, Honolulu, HI, Nov. 29–Dec. 2, 1997, pp. 731–734.

[13] S. C. Douglas, "Self-stabilized gradient algorithms for blind source separation with orthogonality constraints," *IEEE Trans. Neural Networks*, vol. 11, pp. 1490–1497, Nov. 2000.

[14] M. Joho, H. Mathis, and G. S. Moschytz, "An FFT-based algorithm for multichannel blind deconvolution," in *Proc. ISCAS*, Orlando, FL, May 30–June 2, 1999, pp. III–203–206.