COMPARISON OF FINITE INTERVAL CONSTANT MODULUS ALGORITHM AND FAST-ICA FOR BLOCK SISO BLIND EQUALIZATION

Petr Tichavský[1], *Zbyněk Koldovský*[1], *Antonín Heřmánek*[1,2] *and Phillip Regalia*[2]

[1]Institute of Information Theory and Automation,
Academy of Sciences of the Czech Republic, P.O.Box 18,
182 08 Prague 8, Czech Republic

[2]Institute National des Télécommunications/GET
Dept. Communication, Images and Information Processing, Evry, France

ABSTRACT

This paper shows similarity between the finite interval constant modulus algorithm (CMA) [1] and the so called "Fast-ICA" algorithm, originally designed to solve the independent component analysis problem [2,3] in application to blind SISO deconvolution. The latter algorithm was found to be a generalization of the former algorithm. The two algorithms may optimize the same criterion, but they differ in numerical methods used for optimization. While CMA is based on the steepest descent procedure, Fast-ICA uses approximate Newton method. However, for certain choice of the step in the steepest descent procedure, both algorithms are shown to be identical. Advantages of both of the algorithm are demonstrated in simulations in blind deconvolution of QAM, and V29 signals passed through a real-world microwave channel and distorted by additive (complex) Gaussian noise. The methods exhibit good performance compared to that of a popular super-exponential algorithm [4].

1. THE BLIND EQUALIZATION PROBLEM FORMULATION

We consider the blind deconvolution problem in which we observe a noisy output of an unknown possibly nonminimum phase linear system from which we want to recover its input using an adjustable linear filter (equalizer). This is a well known problem, see [1,4,5].

Let the unknown linear system be denoted $\mathcal{H} = \{h_n\}$ and let it be fed by unobserved input a_t which is a sample function from a discrete-time white (iid) random process. In particular, we shall assume that a_t are uniformly distributed in a finite alphabet $\mathcal{A}$. The observed data is

$$y_t = h_t \star a_t + e_t \tag{1}$$

This work was supported by the Grant Agency of the Czech Republic through grant 102/01/0021.

where $\star$ denotes the convolution, e_t is an additive noise which is assumed to be independent of a_t, white and Gaussian. The channel coefficients h_t and the source signal a_t can be both complex-valued. Examples of real-world channels that can be found in microwave applications can be found in [6]. Note that these channels are complex-valued; in this case we assume that the additive noise is complex circular Gaussian.

We want to adjust an equalizer $\mathcal{C} = \{c_n\}$ so that its output z_t approximates the input a_t up to a constant delay and possibly a constant phase shift. Thus, if we denote by

$$s_n = h_n \star c_n = \sum_l c_l \, h_{n-l} \tag{2}$$

the unit sample response of the unknown equalizer, then we want to set c_n so that

$$s_n = e^{j\phi}\delta_{n-k} = \begin{cases} e^{j\phi}, & n = k \\ 0, & n \neq k \end{cases} \tag{3}$$

As a measure of equalization performance we take the intersymbol-interference (ISI) defined by

$$\mathrm{ISI}(\mathbf{s}) = \frac{\sum_n |s_n|^2 - |s_{max}|^2}{|s_{max}|^2} \tag{4}$$

where s_{max} is the component of s_n having the maximal absolute value (the leading tap or cursor). Clearly, a small value of ISI indicates the proximity to the desired solution.

2. REVIEW OF THE METHODS

2.1. Super-Exponential Algorithm

In this paper we consider the version of the super-exponential algorithm [1] which utilizes fourth-order cumulants and is suitable for symmetrically distributed input sources a_t. The algorithm starts with an initial estimate

$\mathbf{c}_0 = (0, \ldots, 0, 1, 0, \ldots, 0)^T$ where "1" is placed somewhere around the middle of the vector, and iterates

$$\begin{aligned} \mathbf{c}'_{m+1} &= \hat{R}^{-1}\hat{\mathbf{d}}^{(m)} \\ \mathbf{c}_{m+1} &= \mathbf{c}'_{m+1}/||\mathbf{c}'_{m+1}|| \end{aligned} \quad (5)$$

for $m = 0, 1, 2, \ldots$ until the convergence is achieved; In (5), $\hat{R}$ is the $L \times L$ matrix whose elements are

$$\hat{R}_{k\ell} = \frac{1}{N}\sum_{t=1}^{N} y_{t-k}y^*_{t-\ell} \quad (6)$$

and $\hat{\mathbf{d}}^{(m)}$ is the $L \times 1$ vector whose elements are

$$\begin{aligned} \hat{\mathbf{d}}_n^{(m)} &= \widehat{\text{cum}}(z_t, z_t, z_t^*, y^*_{t-n}) \\ &= \frac{1}{N}\sum_{t=1}^{N} |z_t|^2 z_t y^*_{t-n} - \frac{2}{N}\sum_{t=1}^{N} |z_t|^2 \frac{1}{N}\sum_{t=1}^{N} z_t y^*_{t-n} \\ &\quad -\frac{1}{N}\sum_{t=1}^{N} z_t^2 \frac{1}{N}\sum_{t=1}^{N} z_t^* y^*_{t-n} \end{aligned} \quad (7)$$

and $z_t = z_t^{(m)} = y_t \star \mathbf{c}_m$ is the equalizer output at iteration m (the superscript "(m)" of z_t is omitted in (7) for brevity).

2.2. Constant-Modulus Algorithm (CMA)

This algorithm is based on minimizing the constant modulus criterion $J(\mathbf{c}) = \sum_{t=1}^{N}(1 - |z_t|^2)^2$ where again, z_t is the equalizer output, $z_t = \mathbf{c} \star y_t$. An equivalent formulation of the algorithm is that the equalizer $\mathbf{c}$ should minimize the criterion [1]

$$F(\mathbf{c}) = \left(\sum_{t=1}^{N} |z_t|^4\right) / \left(\sum_{t=1}^{N} |z_t|^2\right)^2 \quad (8)$$

This algorithm is tailored for the case when the whole alphabet of information symbols a_t lies on the unit circle in the complex plane, $|a_t| = 1$ for all t. However, the algorithm was shown to perform well even in cases when a_t do not have all the same modulus, such as the V29 sources.

The algorithm implementation proposed in [4] is based on gradient descent technique. It begins with forming the data matrix X which consists of delayed versions of the input data y_t, such that the equalizer output can be written as

$$\mathbf{z} = [z_1, \ldots, z_N]^T = \mathbf{X}\mathbf{c} \quad (9)$$

The algorithm proceeds in 3 steps.
1. Run the QR decomposition of $\mathbf{X}$, $\mathbf{X} = \mathbf{QR}$ so that columns of $\mathbf{Q}$ provide an orthonormal basis of the column space of $\mathbf{X}$.
2. Starting with an $L \times 1$ initial vector $\mathbf{b}_0$ with a unit norm (as $\mathbf{c}_0$ in the super-exponential algorithm), iterate

$$\begin{aligned} \mathbf{b}'_{m+1} &= \mathbf{b}_m - \mu_m \mathbf{Q}^T(\mathbf{z} \odot \mathbf{z} \odot \mathbf{z}^*) \\ \mathbf{b}_{m+1} &= \mathbf{b}'_{m+1}/||\mathbf{b}'_{m+1}|| \end{aligned} \quad (10)$$

until convergence is achieved, where $\odot$ denotes the elementwise product, and $\mathbf{z}$ stands for the vector of equalizer output at iteration m,

$$\mathbf{z} = \mathbf{z}^{(m)} = \mathbf{Q}\mathbf{b}_m \quad (11)$$

and $\mu_m = \left(\sum(|z_t^{(m)}|^4\right)^{-1}$ is a step-size.
3. $\mathbf{c} = \mathbf{R}^{-1}\mathbf{b}_\infty$

2.3. Fast-ICA

This algorithm can be viewed as a generalized version of the CM algorithm, where the gradient descent minimization is replaced by approximate Newton method [2,3]. First, the data are "whitened", i.e. the data matrix $\mathbf{X}$ is decomposed as $\mathbf{X} = \mathbf{QR}$, where, aqain, columns of $\mathbf{Q}$ provide an orthonormal basis of the column space of $\mathbf{X}$. It is not a must, but usually $\mathbf{R}$ is determined as the Hermitian symmetric square root of $\mathbf{X}^H\mathbf{X}$.

Below we summarize complex valued version of the algorithm [3]. The algorithm searches for such a linear combination of the data that minimizes the expected value of certain suitable nonlinear function of data ($\mathbf{Q}$), E$[G(|\mathbf{Qb}|^2)]$ subject to constrant $||\mathbf{b}|| = 1$, where the nonlinear function $G(|\cdot|^2)$ is applied elementwise. The idea behind that is that probability distribution of separated sources (estimated original sources) should have different from Gaussian distribution as much as possible, and the non-gaussianity can be mesured by expectation of suitable nonlinear function of the data. In this paper, we prefer the option $G(x) = x^2$ which makes the algorithm equivalent to CMA.

Let $g(\cdot)$ and $g'(\cdot)$ denote the first and second derivative of $G(\cdot)$. The algorithm proceeds recursively, starting with an $L \times 1$ initial vector $\mathbf{b}_0$ with a unit norm iterates

$$\begin{aligned} \mathbf{b}'_{m+1} &= \mathbf{Q}^T[\mathbf{z}^* \odot g(|\mathbf{z}|^2)]\sqrt{N} \\ &\quad -\mathbf{1}^T[g(|\mathbf{z}|^2) + |\mathbf{z}|^2 \odot g'(|\mathbf{z}|^2)]\mathbf{b}_m \\ \mathbf{b}_{m+1} &= \mathbf{b}'_{m+1}/||\mathbf{b}'_{m+1}|| \end{aligned} \quad (12)$$

where $\mathbf{z}$ is defined as in (11), until convergence is achieved. In (13), the functions g and g' are applied elementwise, and $\mathbf{1}$ stands for a column vector of 1's. Finally, $\mathbf{z}^{(m)} = \mathbf{Q}\mathbf{b}_m$ and $\mathbf{c} = \mathbf{R}^{-1}\mathbf{b}_\infty$, like in the CM algorithm.

For the option $G(x) = x^2$ we receive the algorithm

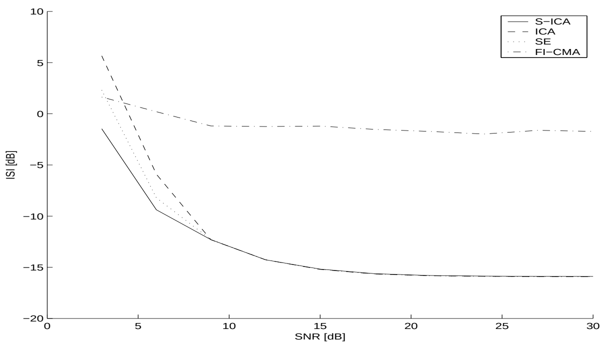
$$\begin{aligned} \mathbf{b}'_{m+1} &= \mathbf{Q}^T(\mathbf{z} \odot \mathbf{z} \odot \mathbf{z}^*)\sqrt{N} - 2\mathbf{b}_m \\ \mathbf{b}_{m+1} &= \mathbf{b}'_{m+1}/||\mathbf{b}'_{m+1}|| \end{aligned} \quad (13)$$

We can see that the above algorithm is equivalent to CMA with the step size $\mu_m = \sqrt{N}/2$. However, we found that the algorithm is more stable with the choice $\mu_m = \sqrt{N}/3$. The former (original) algorithm is referred as ICA, the latter algorithm is referred as S-ICA (stabilized ICA).

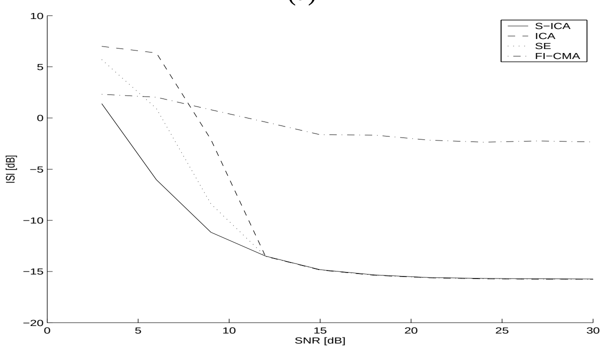
3. SIMULATIONS

We tested the 4 above mentioned algorithm using QAM and V29 signals, commonly used in communications [4], with the length $N = 500$ and $N = 1000$. The data were convolved with (i) the channel presented in [4], that has length 7 (tap values $(0.4, 1, -0.7, 0.6, 0.3, -0.4, 0.1)$) and (ii) with the real-world complex-valued microwave channel [6]. In particular we used *chan1.mat* from the database [6], re-sampled to baud-rate and truncated to length 9 (keeping the most significant tap values). The tap values of the latter filter were $-0.0118 + 0.0017i$, $0.0698 + 0.0016i$, $0.9143 - 0.0103i$, $0.3874 - 0.0130i$, $-0.0901 + 0.0056i$, $0.0193 - 0.0071i$, $-0.0015 + 0.0036i$, $0.0011 + 0.0023i$, $0.0117 + 0.0031i$. In addition, the received signal was burried in additive complex Gaussian noise.

The result are shown in Figures 1-4. In all figures, solid line, dashed line, dot-dashed line and dotted line represents S-ICA, ICA, finite interval CMA and super-exponenential algorithm, respectively.

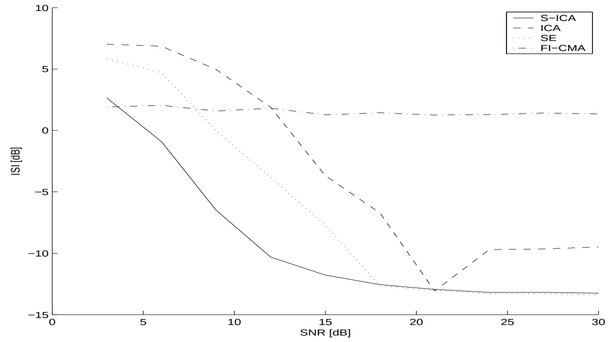


(a)

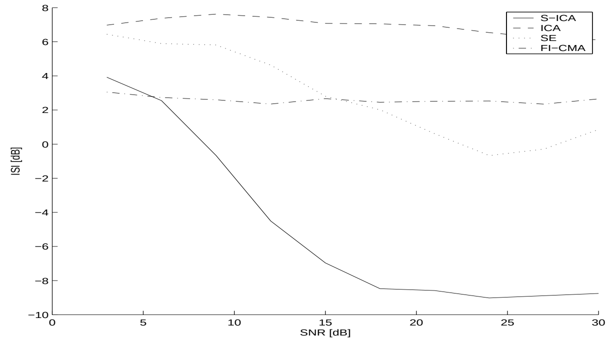


(b)

Fig. 1 Performance of the algorithms for QAM source and the SW's channel (a) for $N = 1000$ and (b) for $N = 500$

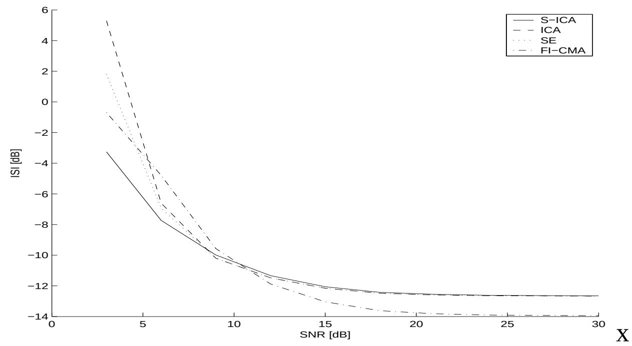


(a)

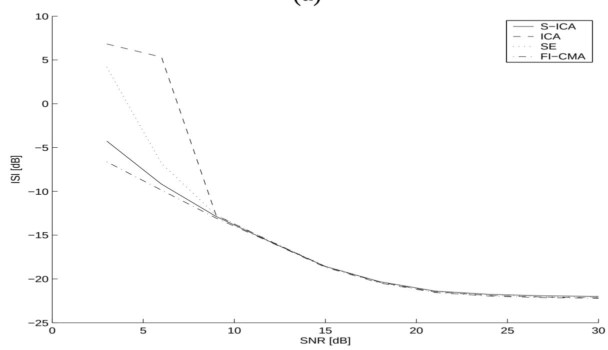


(b)

Fig. 2 Performance of the algorithms for V29 source and the SW's channel (a) for $N = 1000$ and (b) for $N = 500$



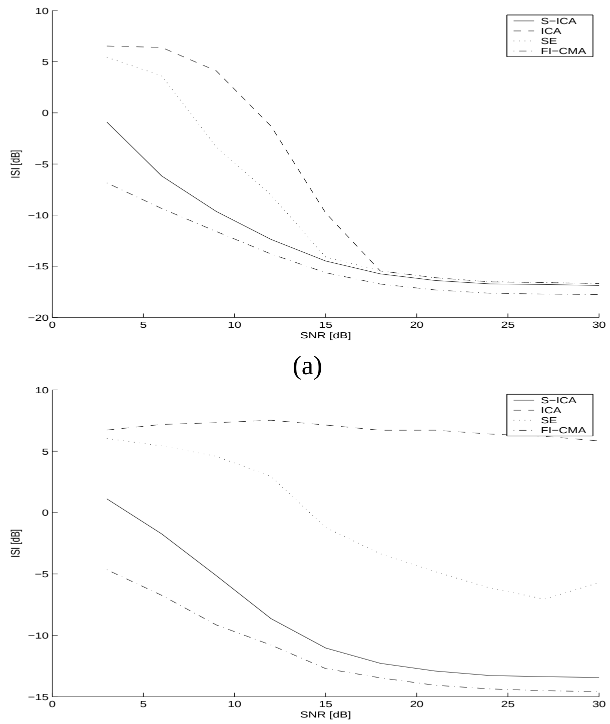
(a)



(b)

Fig. 3 Performance of the algorithms for QAM source and the real-world channel (a) for $N = 1000$ and (b) for

$N = 500$



(a)

(b)

Fig. 4 Performance of the algorithms for V29 source and the real-world channel (a) for $N = 1000$ and (b) for $N = 500$

4. CONCLUSIONS

We showed that the finite interval CMA is, for a certain step size in the iterative algorithm equivalent to algorithm Fast-ICA with kurtosis-based non-linearity.

The performance of the algorithms depends on the channel properties. Stabilized Fast-ICA exhibits relatively good performance on both channels The FI-CMA outperforms the former algorithm in the case of the particular real-world channel, but does not seem to converge well in the case of the Shalvi-Weinstein's channel for these short data records. The stabilized ICA outperforms the original ICA and superexponential algorithm. Its sensitivity to additive noise is better both for the QAM and V29 alphabet.

References

[1] P.A. Regalia, "Finite-interval constant modulus algorithm", in *Proc. 2002 IEEE Int. Conf. Acoustics, Speech, Signal Procesing*, Orlando, FL, May 2002, Vol. 3, pp. 2285-2288.

[2] A. Hyvärinen and E. Oja, "A fast fixed-point algorithm for independent component analysis", *Neural Comput.*, Vol. 6 (1997), pp.1483-1492.

[3] E. Bingham and A. Hyvrinen, "A fast fixed-point algorithm for independent component analysis of complex-valued signals." *Int. J. of Neural Systems*, Vol. 10, No.1, pp.1-8, January 2000.

[4] O. Shalvi and E. Weinstein, "Super-exponential methods for blind deconvolution", *IEEE Trans. Information Theory*, Vol. 39, No.2, pp. 504-519, March 1993.

[5] C.Y.Chi, C.Y.Chen, C.H.Chen and C.C. Feng, "Batch processing algorithms from blind equalization using higher-order statistics", *Signal Processing Magazine*, Vol. 20, No.1, pp. 25-49, January 2003.

[6] http://spib.rice.edu/spib/microwave.html