Adaptive Algorithms
for
Blind Source Separation

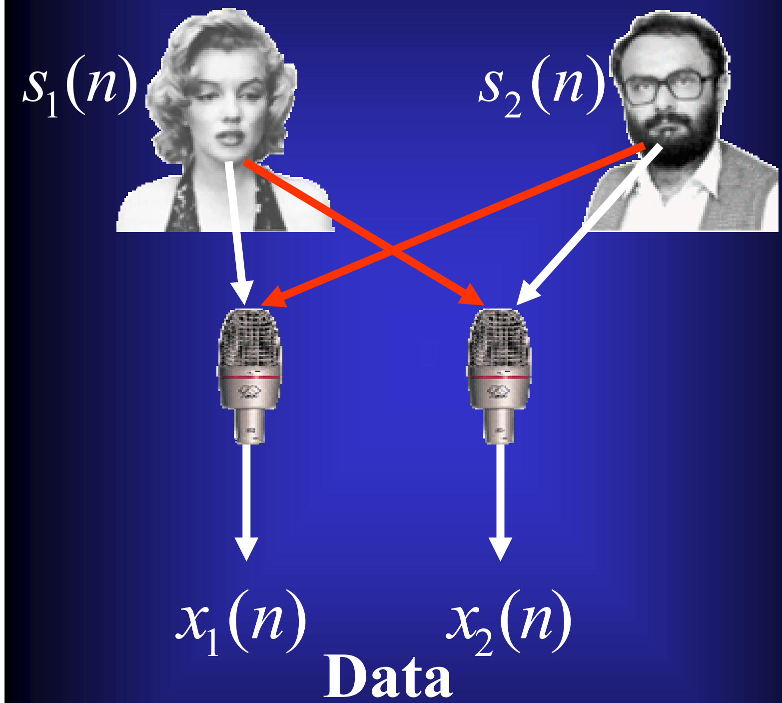
George Moustakides

Department of Computer Engineering and Informatics
UNIVERSITY OF PATRAS, GREECE

Outline of the Presentation

- Problem definition
- Existing adaptive technique
- Analysis of existing algorithm
- General adaptive scheme
- Performance measure
- Optimum algorithms
- Conclusion

Problem Definition



$$x_1(n) = a_{11}s_1(n) + a_{12}s_2(n)$$

$$x_2(n) = a_{21}s_1(n) + a_{22}s_2(n)$$

$$S(n) = \begin{bmatrix} s_1(n) \\ s_2(n) \end{bmatrix}, X(n) = \begin{bmatrix} x_1(n) \\ x_2(n) \end{bmatrix}$$

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

$$X(n) = \mathbf{A}S(n)$$

Estimate B = $\mathbf{A}^{-1}$

$$S(n)=\begin{bmatrix} s_1(n) \\ s_2(n) \\ \vdots \\ s_M(n) \end{bmatrix}, X(n)=\begin{bmatrix} x_1(n) \\ x_2(n) \\ \vdots \\ x_N(n) \end{bmatrix}, \mathbf{A}=\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1M} \\ a_{21} & a_{22} & \cdots & a_{2M} \\ \vdots & \vdots & \vdots & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NM} \end{bmatrix}$$

$$X(n)=\mathbf{A}S(n)$$

$N \geq M$ Matrix **A** must be full rank

Existing Adaptive Technique

Let N=M=2 and $\mathbf{B}(n)$ denote the estimate of $\mathbf{A}^{-1}$ at time n.
At time n available: $\mathbf{B}(n\text{-}1)$, $X(n)$

$$\hat{S}(n) = \mathbf{B}(n-1)X(n)$$

$$\mathbf{B}(n) = \mathbf{B}(n-1) - \mu\mathbf{H}\left(\hat{S}(n)\right)\mathbf{B}(n-1),\ \mathbf{B}(0) = \mathbf{I}$$

$$\mathbf{H}(Z) = \begin{bmatrix} z_1^2 - 1 & z_1 z_2 + g(z_1)z_2 - g(z_2)z_1 \\ z_1 z_2 - g(z_1)z_2 + g(z_2)z_1 & z_2^2 - 1 \end{bmatrix}$$

μ is a constant (step size) with $0 < \mu << 1$.
$g(z)$ is an odd univariate function (i.e. $g(-z) = -g(z)$) with $g(z) \neq \alpha z$.

Does $\mathbf{B}(n)$ converge to $\mathbf{A}^{-1}$? or equivalently does

$$\mathbf{C}(n) = \mathbf{B}(n)\mathbf{A}$$

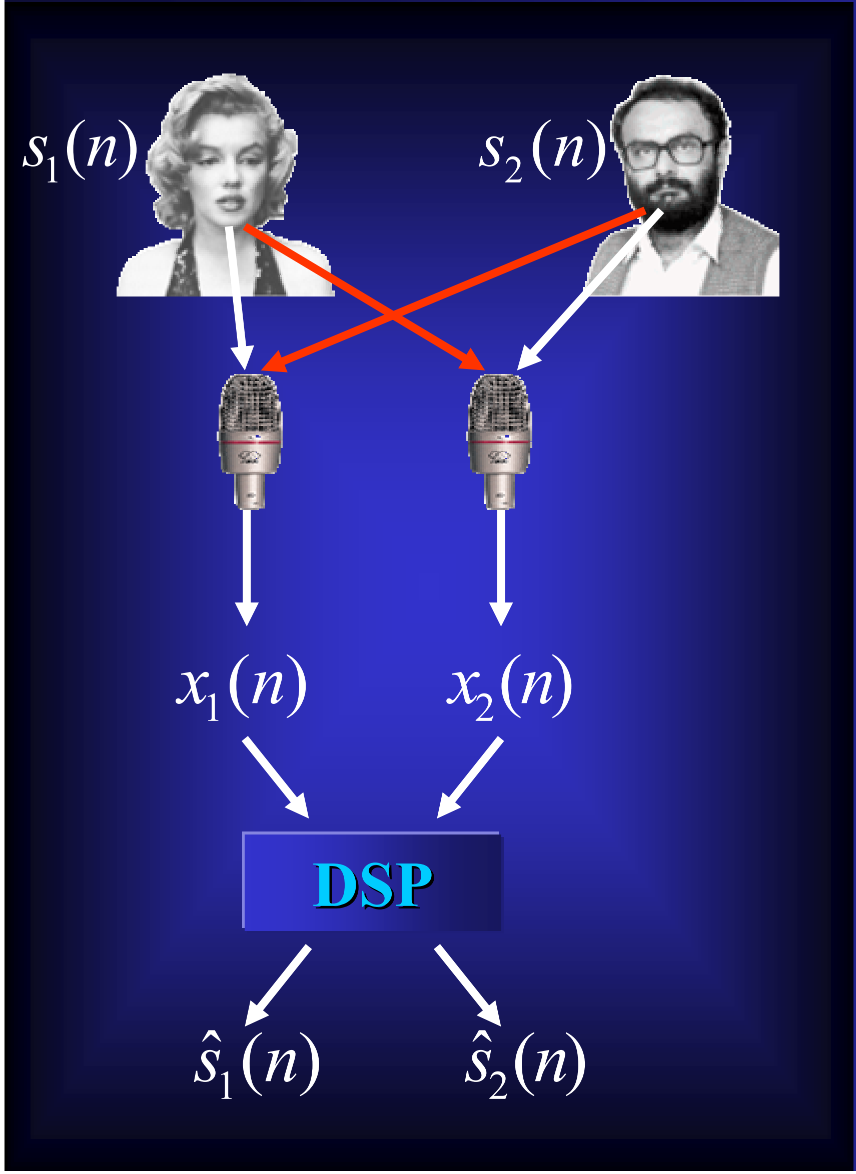
converge to the identity matrix $\mathbf{I}$?

Theorem 1. If $s_1(n)$ and $s_2(n)$ are independent, unit variance random variables with symmetric density functions and with at most one being Gaussian, then

- $\lim_{n\to\infty} E\{\mathbf{C}(n)\} = \begin{bmatrix} \pm 1 & 0 \\ 0 & \pm 1 \end{bmatrix}$ or $\begin{bmatrix} 0 & \pm 1 \\ \pm 1 & 0 \end{bmatrix}$

- The covariance of $\mathbf{C}(n)$ is $O(\mu)$.

Example (revisited)

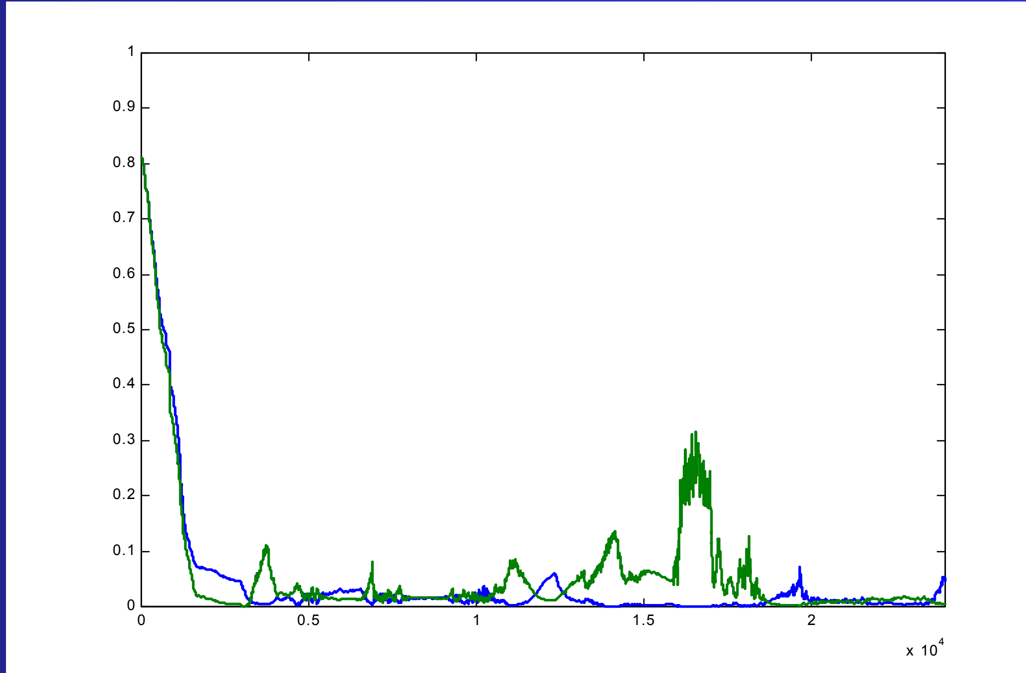


$$\hat{S}(n) = \mathbf{B}(n)X(n) = \mathbf{B}(n)\mathbf{A}S(n) =$$

$$\mathbf{C}(n)S(n) = \begin{bmatrix} c_{11}(n) & c_{12}(n) \\ c_{21}(n) & c_{22}(n) \end{bmatrix} \begin{bmatrix} s_1(n) \\ s_2(n) \end{bmatrix}$$

$$|c_{12}(n)| / |c_{11}(n)| \ll 1, |c_{21}(n)| / |c_{22}(n)| \ll 1$$

$$g(z) = \alpha z^3$$



Analysis of Existing Algorithm

$$\hat{S}(n) = \mathbf{B}(n-1)X(n)$$

$$\mathbf{B}(n) = \mathbf{B}(n-1) - \mu\mathbf{H}\left(\hat{S}(n)\right)\mathbf{B}(n-1),\ \mathbf{B}(0) = \mathbf{I}$$

$$\hat{S}(n) = \mathbf{C}(n-1)S(n)$$

$$\mathbf{C}(n) = \mathbf{C}(n-1) - \mu\mathbf{H}\left(\hat{S}(n)\right)\mathbf{C}(n-1),\ \mathbf{C}(0) = \mathbf{A}$$

Why does the adaptation work when

$$\mathbf{H}(Z)=\begin{bmatrix} z_1^2-1 & z_1z_2+g(z_1)z_2-g(z_2)z_1 \\ z_1z_2-g(z_1)z_2+g(z_2)z_1 & z_2^2-1 \end{bmatrix}$$

$$\mathbf{H}_1(Z)=\begin{bmatrix} z_1^2-1 & z_1z_2 \\ z_1z_2 & z_2^2-1 \end{bmatrix}$$

forces regular correlation to become zero

$$\mathbf{H}_2(Z)=\begin{bmatrix} 0 & g(z_1)z_2-g(z_2)z_1 \\ -g(z_1)z_2+g(z_2)z_1 & 0 \end{bmatrix}$$

forces some nonlinear correlation to become zero.

Analysis using Stochastic Approximation theory

From the theory of adaptive algorithms we know that an algorithm of the form

$$\hat{S}(n) = \mathbf{C}(n-1)S(n)$$

$$\mathbf{C}(n) = \mathbf{C}(n-1) - \mu\mathbf{H}\left(\hat{S}(n)\right)\mathbf{C}(n-1)$$

converges (in the mean) to **stable** equilibrium matrices **C** that satisfy

$$\mathbf{C} = \mathbf{C} - \mu E\left\{\mathbf{H}\left(\mathbf{C}S(n)\right)\right\}\mathbf{C}$$

or equivalently $E\left\{\mathbf{H}\left(\mathbf{C}S(n)\right)\right\} = 0$

Since

$$\mathbf{H}(Z) = \begin{bmatrix} h_{11}(z_1, z_2) & h_{12}(z_1, z_2) \\ h_{21}(z_1, z_2) & h_{22}(z_1, z_2) \end{bmatrix}$$

relation $$E\left\{\mathbf{H}\left(\mathbf{C}S(n)\right)\right\} = 0$$

is equivalent to a system of four (in general nonlinear) equations

$$E\left\{h_{ij}\left(\mathbf{C}S(n)\right)\right\} = 0, \ \ i, j = 1,2$$

in four unknowns (the four elements of matrix **C**) that can be solved to identify the equilibrium matrices.

If we are interested in imposing a specific matrix $\mathbf{C}_0$ as equilibrium, then the elements of $\mathbf{H}(Z)$ must satisfy

$$E\left\{h_{ij}\left(\mathbf{C}_0 S(n)\right)\right\}=0,\ \ i,j=1,2$$

In particular if $\mathbf{C}_0=\mathbf{I}$, the four relations become

$$E\left\{h_{ij}\left(S(n)\right)\right\}=0,\ \ i,j=1,2$$

$$\mathbf{H}(S)=\begin{bmatrix} s_1^2-1 & s_1 s_2+g(s_1)s_2-g(s_2)s_1 \\ s_1 s_2-g(s_1)s_2+g(s_2)s_1 & s_2^2-1 \end{bmatrix}$$

The existing function H(Z) is one possible choice, it is definitely not the only one!!!

General Adaptive Scheme

$$\hat{S}(n) = \mathbf{B}(n-1)S(n)$$

$$\mathbf{B}(n) = \mathbf{B}(n-1) - \mu\mathbf{H}\left(\hat{S}(n)\right)\mathbf{B}(n-1)$$

$$\mathbf{H}(Z) = \begin{bmatrix} h_{11}(z_1, z_2) & h_{12}(z_1, z_2) \\ h_{21}(z_1, z_2) & h_{22}(z_1, z_2) \end{bmatrix}$$

If we select $h_{ij}(z_1, z_2)$ to satisfy

$$E\left\{h_{ij}\left(\pm s_1, \pm s_2\right)\right\} = E\left\{h_{ij}\left(\pm s_2, \pm s_1\right)\right\} = 0$$

then we guarantee that the recursion has the same equilibrium matrices as the original algorithm.

If we would like however the algorithm to behave in exactly the same way (as far as mean trajectory and second order statistics is concerned) when converging to any desired equilibrium matrix, we need to impose the following structure on $\mathbf{H}(Z)$

$$\mathbf{H}(Z) = \begin{bmatrix} h(z_1, z_2) & q(z_2, z_1) \\ q(z_1, z_2) & h(z_2, z_1) \end{bmatrix} \quad \text{with} \quad \begin{matrix} h(z_1, z_2) \neq h(z_2, z_1) \\ q(z_1, z_2) \neq q(z_2, z_1) \end{matrix}$$

$$h(-z_1, z_2) = h(z_1, -z_2) = h(z_1, z_2)$$
$$q(-z_1, z_2) = q(z_1, -z_2) = -q(z_1, z_2)$$

$$E\{h(s_1, s_2)\} = E\{h(s_2, s_1)\} = 0$$
$$E\{q(s_1, s_2)\} = E\{q(s_2, s_1)\} = 0 \text{ (for free)}$$

The proposed structure of $\mathbf{H}(Z)$ guarantees that

- we have the same equilibrium matrices as the classical algorithm
- the trajectory follows symmetric trajectories when initialized from symmetric points, therefore not favoring any equilibrium matrix in any sense.

The classical matrix

$$\mathbf{H}(Z) = \begin{bmatrix} z_1^2 - 1 & z_1 z_2 + g(z_1) z_2 - g(z_2) z_1 \\ z_1 z_2 - g(z_1) z_2 + g(z_2) z_1 & z_2^2 - 1 \end{bmatrix}$$

satisfies the mentioned constraints, therefore it belongs to the class of functions we propose.

Example

Existing scheme

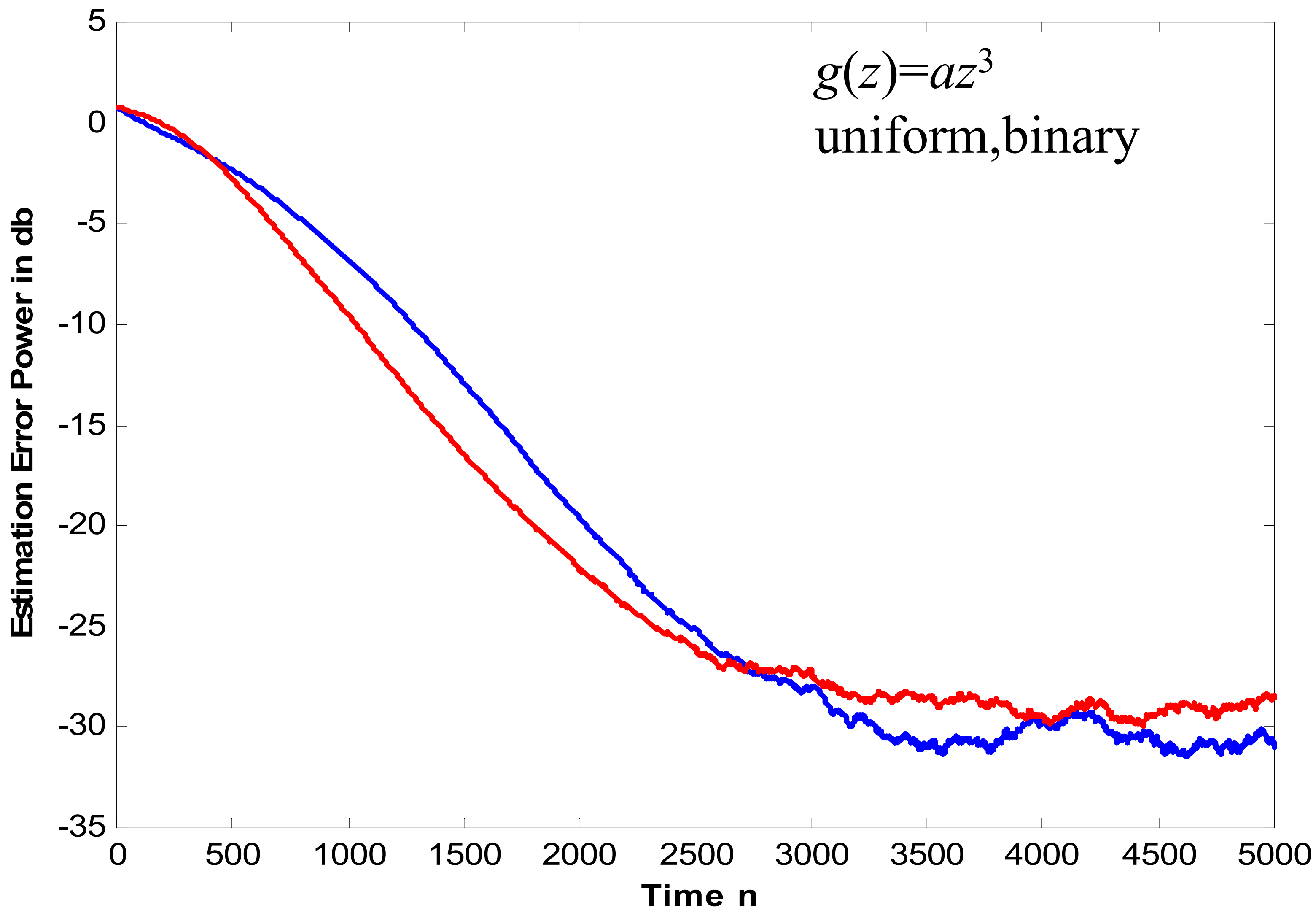
$$\mathbf{H}(Z)=\begin{bmatrix} z_1^2-1 & z_1z_2+g(z_1)z_2-g(z_2)z_1 \\ z_1z_2-g(z_1)z_2+g(z_2)z_1 & z_2^2-1 \end{bmatrix}$$

Select

$$\mathbf{H}(Z)=\begin{bmatrix} |z_1|-1 & \dfrac{\operatorname{sgn}(z_2)g(z_1)}{1+z_1^2+z_2^2} \\ \dfrac{\operatorname{sgn}(z_1)g(z_2)}{1+z_1^2+z_2^2} & |z_2|-1 \end{bmatrix}$$

The two matrices **are not related in any sense!**

$$\left\| \mathbf{C}(n) - E\left\{\mathbf{C}(\infty)\right\} \right\|^2 = \sum_{i,j} \left(\mathbf{C}_{ij}(n) - E\left\{\mathbf{C}_{ij}(\infty)\right\} \right)^2$$



Performance Measure

To propose a performance measure we will work with the algorithm that uses $\mathbf{C}(n)$, we recall

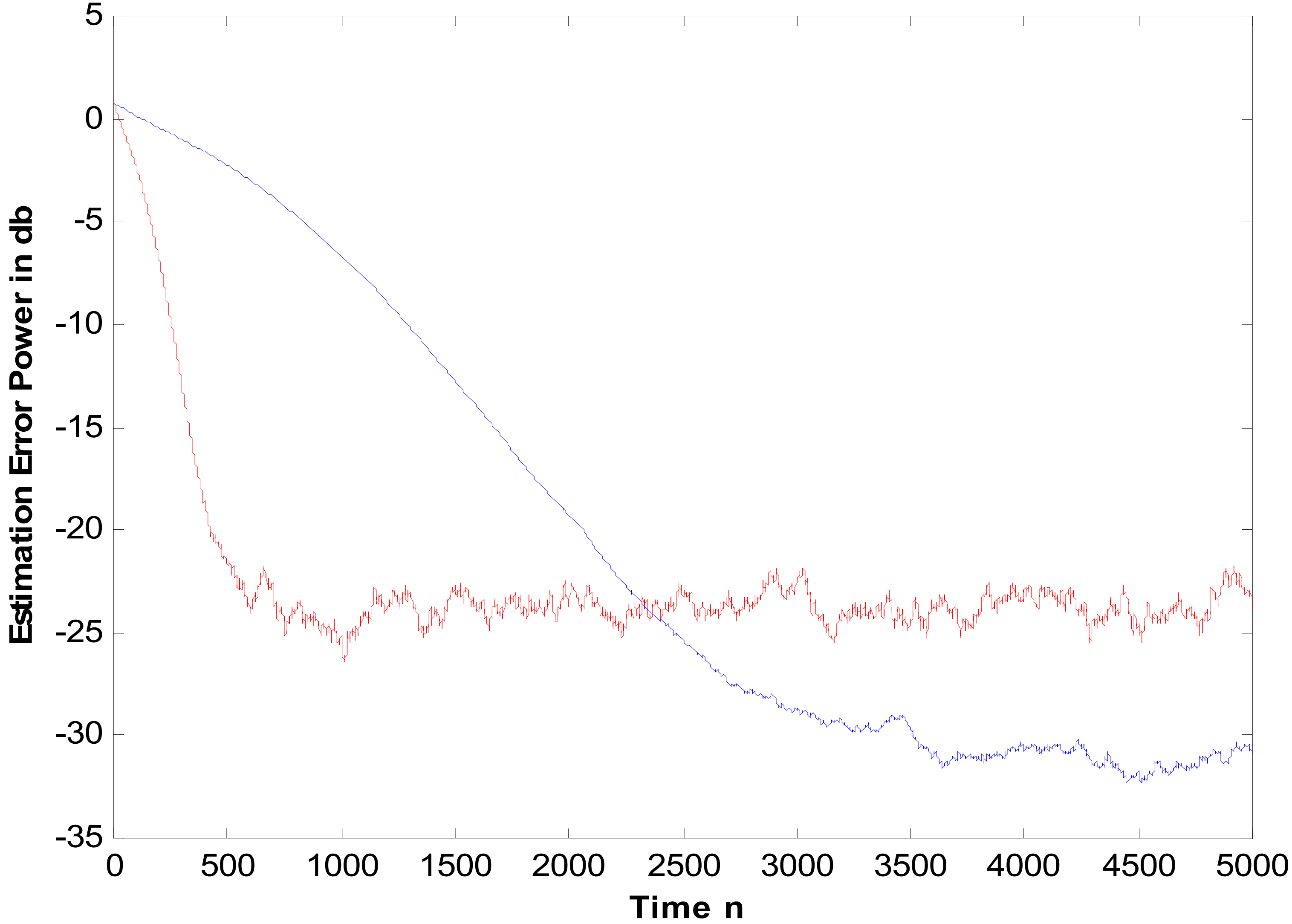
$$\hat{S}(n) = \mathbf{C}(n-1)S(n)$$

$$\mathbf{C}(n) = \mathbf{C}(n-1) - \mu\mathbf{H}\left(\hat{S}(n)\right)\mathbf{C}(n-1)$$

There are usually two quantities that are of interest in adaptive algorithms, namely

- Rate of convergence
- Steady state error power

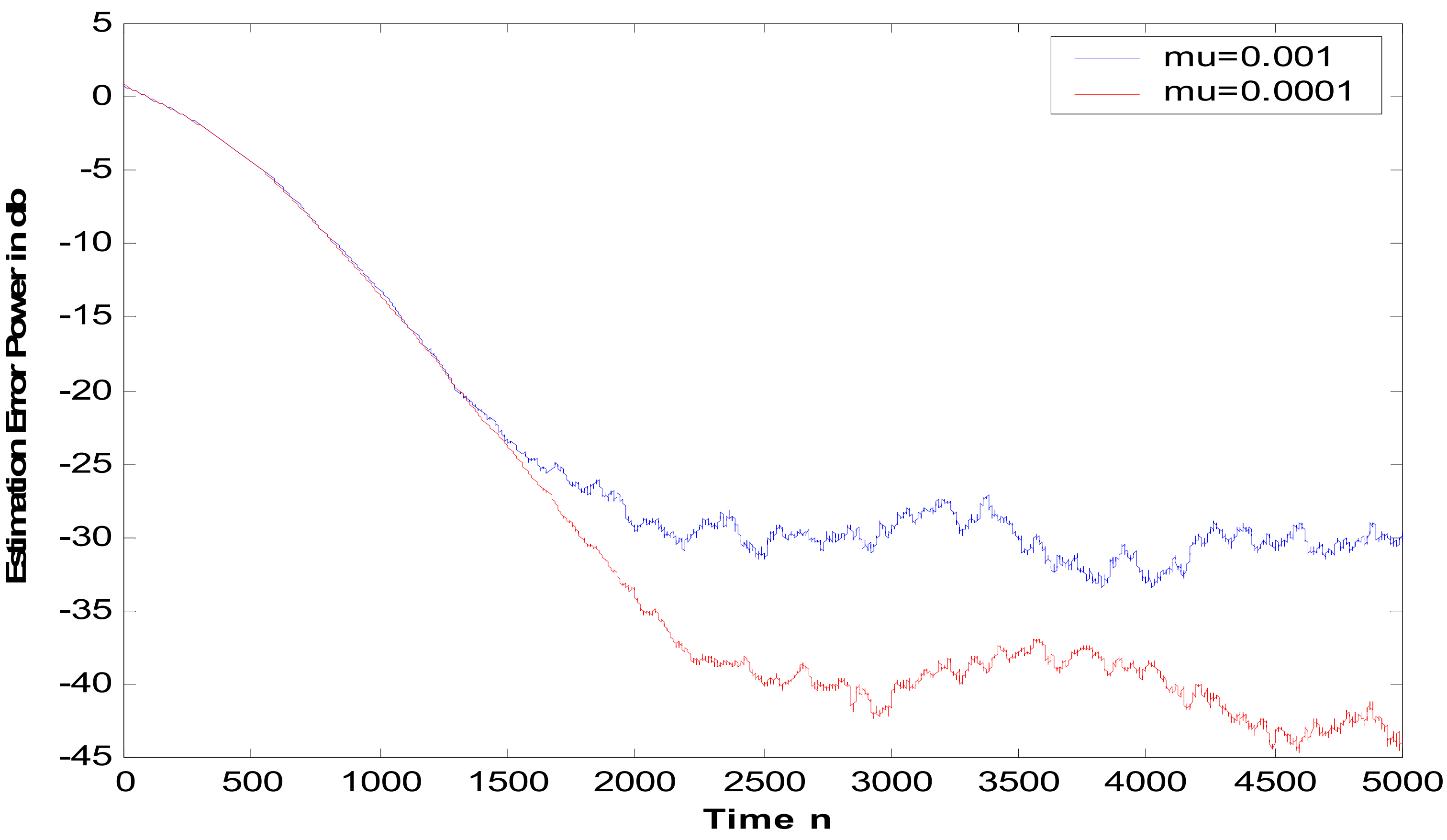
$$\left\| \mathbf{C}(n) - E\{\mathbf{C}(\infty)\} \right\|^2 = \sum_{i,j} \left(\mathbf{C}_{ij}(n) - E\{\mathbf{C}_{ij}(\infty)\} \right)^2$$



$\Delta(n) = (\mathbf{I} -$

Convergence rate

Although the recurrence is nonlinear, in the vicinity of the equilibrium matrix the behavior can be well approximated by a linear recursion.



$$-\lim_{n\to\infty} \frac{\log\left(\left\| E\{\mathbf{C}(n)\} - E\{\mathbf{C}(\infty)\}\right\|\right)}{n} = \mu \min_i \left\{\mathrm{Re}\left(\lambda_i\right)\right\} + o(\mu)$$

λ_i are the eigenvalues of $\mathbf{Q}_1, \mathbf{Q}_2$, where

$$\mathbf{Q}_1 = E\left\{\begin{bmatrix} h(s_1, s_2) \\ h(s_2, s_1) \end{bmatrix}\begin{bmatrix} s_1 l_1(s_1) - 1 & s_2 l_2(s_2) - 1 \end{bmatrix}\right\}$$

$$\mathbf{Q}_2 = E\left\{\begin{bmatrix} q(s_1, s_2) \\ q(s_2, s_1) \end{bmatrix}\begin{bmatrix} s_1 l_2(s_2) & s_2 l_1(s_1) \end{bmatrix}\right\}$$

$$l_i(z) = -\frac{f_i'(z)}{f_i(z)},$$ $f_i(z)$ pdfs of the two sources

For (local) stability of the equilibrium: $\min_i \left\{\mathrm{Re}\left(\lambda_i\right)\right\} > 0$

Steady state error power

$$\lim_{n\to\infty} E\left\{\left\|\mathbf{C}(n)-E\{\mathbf{C}(\infty)\}\right\|^2\right\} = \mu\,\mathrm{tr}(\mathbf{P}_1+\mathbf{P}_2)+o(\mu)$$

$$\mathbf{Q}_i\mathbf{P}_i+\mathbf{P}_i\mathbf{Q}_i^t=\mathbf{R}_i,\ i=1,2$$

$$\mathbf{R}_1=E\left\{\begin{bmatrix}h(s_1,s_2)\\ h(s_2,s_1)\end{bmatrix}\begin{bmatrix}h(s_1,s_2) & h(s_2,s_1)\end{bmatrix}\right\}$$

$$\mathbf{R}_2=E\left\{\begin{bmatrix}q(s_1,s_2)\\ q(s_2,s_1)\end{bmatrix}\begin{bmatrix}q(s_1,s_2) & q(s_2,s_1)\end{bmatrix}\right\}$$

To compare algorithms we will first adjust the step size μ in each algorithm so that all have the same steady state error power and then compare the exponential rates.

The exponential rate of an algorithm, given that the steady state error power has a given value σ^2, is **σ^2 Eff** where

$$\mathrm{Eff} = \frac{\min_i \left\{ \mathrm{Re}\left(\lambda_i \right) \right\}}{\mathrm{tr}\left(\mathbf{P}_1 + \mathbf{P}_1 \right)}$$

The Efficacy Eff is a function of $h(z_1,z_2)$, $q(z_1,z_2)$. Since the efficacy expresses rate of convergence, an algorithm is better if it has larger Efficacy. Consequently for an algorithm to be optimum, it must

maximize the Efficacy over $h(z_1,z_2)$, $q(z_1,z_2)$

Optimum Algorithms

Theorem 2. Let the two sources have the same pdfs, then the optimum functions $h(z_1,z_2)$, $q(z_1,z_2)$ are given by

$$h(z_1,z_2)=a\left(z_1 l(z_1)-1\right)$$

$$q(z_1,z_2)=b z_1 l(z_2)+c z_2 l(z_1)$$

$$l(z)=-\frac{f'(z)}{f(z)},$$

$f(z)$ common pdf

where a, b and c proper constants that guarantee that the two matrices $\mathbf{Q}_1$, $\mathbf{Q}_2$ have all their eigenvalues equal to unity.

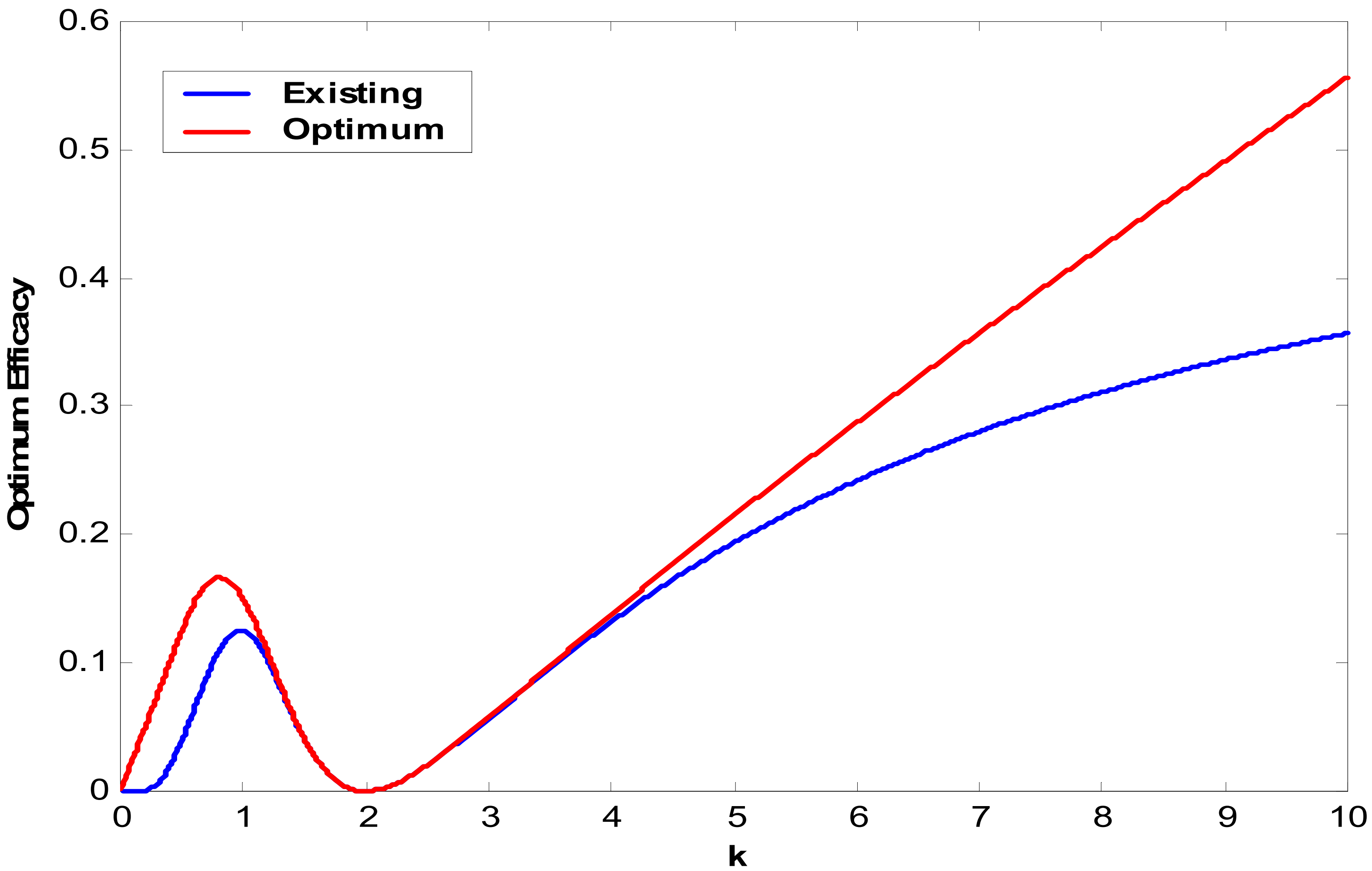
This should be compared to the (optimum) existing scheme

$$h(z_1,z_2)=z_1^2-1$$

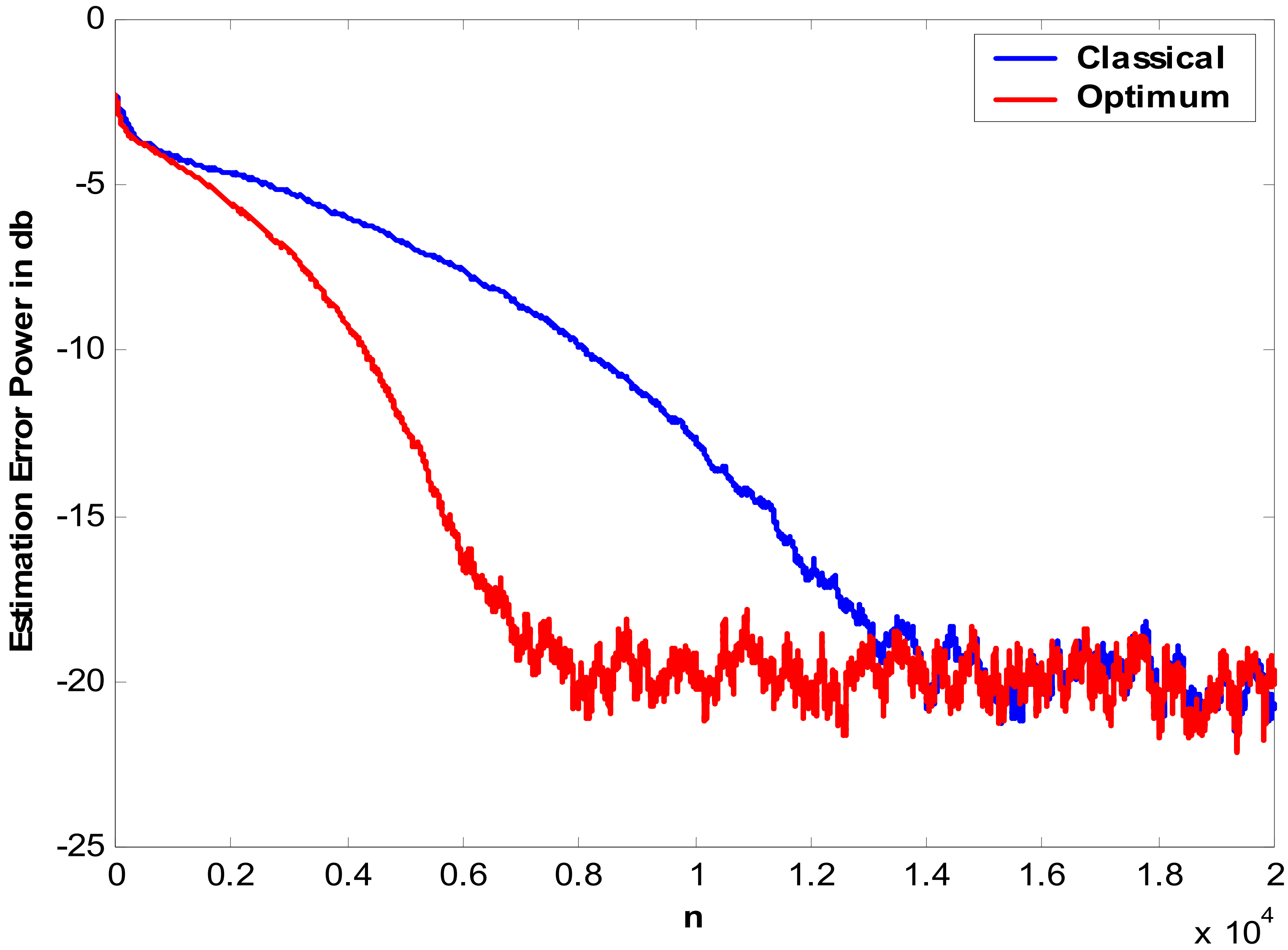
$$q(z_1,z_2)=z_1 z_2+a\left(z_1 l(z_2)-z_2 l(z_1)\right)$$

Generalized Gaussian

$$f(s) = K(k)\, e^{-\left(\frac{|s|}{A(k)}\right)^k}, \quad l(s) = \frac{k}{[A(k)]^k} |s|^{k-1} \operatorname{sgn}(s)$$



Generalized Gaussian, $k = 0.6$. $\frac{\text{Eff}_O(0.6)}{\text{Eff}_C(0.6)} = 2.3$



Conclusion

- We have presented a rich class of adaptive algorithms that can solve the blind source separation problem.
- We have introduced an analytic performance measure that allows us to compare algorithms from our class.
- We have found algorithms that optimize the performance measure for equi-distributed sources.

END