Blind separation of sources, Part II: Problems statement

Pierre Comon
THOMSON-SINTRA, Parc de Sophia Antipolis, BP 138, F-06561 Valbonne Cedex, France

Christian Jutten and Jeanny Herault
Laboratory TIRF, INPG, 46 av. Felix Viallet, F-38031 Grenoble, France

Received 2 April 1990
Revised 8 October 1990 and 27 February 1991

Abstract. Though it arouses more and more curiosity, the HJ iterative algorithm has never been derived in mathematical terms to date. We attempt in this paper to describe it from a statistical point of view. For instance the updating term of the synaptic efficacies matrix cannot be the gradient of a single C^2 functional contrary to what is sometimes understood. In fact, we show that the HJ algorithm is actually searching common zeros of n functionals by pipelined stochastic iterations. Based on simulation results, advantages and limitations as well as possible improvements are pointed out after a short theoretical analysis.

Zusammenfassung. Obwohl er mehr und mehr Interesse weckt, ist der iterative HJ-algorithmus noch nie auf mathematischem Wege hergeleitet worden. In diesem Beitrag versuchen wir, ihn von einem statistischen Gesichtspunkt aus zu beschreiben. Zum Beispiel kann der Erneuerungsterm der synaptischen Wirksamkeitsmatrix nicht der Gradient eines einzelnen C^2-Funktionals sein - obwohl er gelegentlich so interpretiert wird. Wir zeigen, daß der HJ-Algorithmus in der Tat gemeinsame Nullstellen von n Funktionalen mittels stochastischer, im Pipelining ausgeführter Iterationen sucht. Auf der Grundlage von Simulationen werden Vorteile und Bergrenzungen sowie mögliche Verbesserungen nach kurzer theoretischer Analyse dargelegt.

Résumé. Bien qu'il éveille maintenant de plus en plus de curiosité, l'algorithme itératif HJ n'a jusqu'à présent jamais été justifié en termes mathématiques rigoureux. Par exemple, le terme de remise à jour des poids synaptiques ne peut pas être le gradient d'une fonctionnelle unique de classe C^2, contrairement à ce qui est parfois reconnu. Nous essayons dans cet article de le présenter en empruntant une approche statisique. Nous montrons que l'algorithme HJ recherche en réalité les zéros communs de n fonctionnelles par des itérations stochastiques pipelinées. Après une courte analyse théorique, et en nous appuyant sur des simulations, nous soulignons les avantages et les limitations de l'algorithme, ainsi qu'un certain nombre d'améliorations possibles.

Keywords. Signal and image processing, stochastic processes, mixture, neural networks, principal components, independent components, inverse problem.

Introduction

This paper deals with the so-called 'source separation' problem, which is still quite unknown by the signal processing community. Because of its numerous possible applications, in particular in image processing or antenna array processing, this subject begins to receive a greater attention. It may be described in a few words as follows. Suppose we receive n random signals on n discrete sensors, and suppose these signals are propagating from n source locations through a deterministic linear medium. Assuming the source signals are statistically independent, the problem consists of recovering them from observed signals only. The assumptions we are allowed to resort to are only the linearity of the transformation and the independence assumption. The terminology is not yet established for this kind of problem, and we can talk about blind equalization as well as deconvolution. But what is important to stress here is that we deal with a multichannel problem, which confers it very particular features. As shown in [2], multichannel deconvolution can formally be split into monic deconvolution and instantaneous source separation. We shall restrict our attention to the latter problem, called in short 'source separation'.

This simplified problem has been already studied following various approaches, including [2, 4, 6], that we shall not investigate here. We rather devote the paper to an alternative presentation of the HJ algorithm introduced in [5], attempting to solve the problem in an iterative manner. Our aim is not to give a perfect solution, since to date there does not exist any, but to point out what are the advantages and limitations of the selected algorithm, and to raise a few open questions.

1. Properties inherent in the problem

Let $x_i(t)$ denote n unknown processes, $1 \leq i \leq n$; these processes will be referred to as 'source' processes in the rest of the paper. They are assumed zero-mean, stationary up to the fourth order, and mutually independent for convenience, though these assumptions are not necessarily always required in practice. Denote by $e_i(t)$ the observations obtained by an unknown memoryless linear transform of the sources. Then we have in compact form the relation between both vector-processes e and x:

$$e(t) = Ax(t), \tag{1}$$

where A is an unknown $n \times n$ matrix. The goal is to retrieve the unknown source processes $x_i(t)$ from $e_j(t)$, by using solely their statistical independence. It is obvious that the solution, if it is obtainable, is necessarily given by a linear transform:

$$s(t) = Fe(t). \tag{2}$$

Notice that if a process $x(t)$ has independent components, then so do the process $PAx(t)$, where P is any permutation matrix and Λ any regular diagonal matrix; in other words, we may have $FA = P\Lambda$ [2]. The consequence is that among the n^2 unknowns (the entries of F), n of them are undetermined. It suffices to impose n entries in F to get a well-posed problem. One way of doing this would be to impose that the variances of the outputs $s_i(t)$ be all equal to 1 [2]. Another way is to look for a matrix A with unit diagonal entries, that is, for a matrix F of the form

$$F = (I + C)^{-1}, \quad \text{with } \mathrm{Diag}\{C\} = 0. \tag{3}$$

This constraint is assumed in the HJ algorithm. Now, it remains to find matrix C, that is, only $n(n-1)$ unknowns. Ideally, there are *several* equivalent solutions given by

$$C = AP[\mathrm{Diag}(AP)]^{-1} - I, \tag{4}$$

where P describes the set of permutations in dimension n, namely $n!$ distinct solutions. For instance, for $n = 2$ and if we denote by a_{ij} the entries of A, there exist two solutions since two different permutations exist:

$$[c_{12}, c_{21}] = [a_{12}/a_{22}, a_{21}/a_{11}]$$

and

$$[c_{12}, c_{21}] = [a_{11}/a_{21}, a_{22}/a_{12}].$$

The first idea we could think of is to resort to a second-order description of the stochastic processes, namely to use a quadratic optimisation criterion. Unfortunately, this will leave us with a large indetermination. In fact, there are $n(n-1)/2$ second order cross moments and their cancellation provides us with $n(n-1)/2$ quadratic equations, which is half less than the number of unknowns. For instance, the solution given by the principal components analysis (PCA) of the covariance matrix $\mathrm{E}\{s(t)s(t)^{\mathrm{T}}\}$ provides uncorrelated outputs $s_i(t)$, that may not be pairwise independent whenever they are non-Gaussian, PCA is not the unique solution of the system of second order equations, but it is one solution that has orthogonal columns. Moreover, Gaussian processes in the strict sense are rarely observed in the real world. For these two reasons the PCA solution is not satisfactory. The need of $n(n-1)/2$ additional equations may be faced by utilizing higher-order statistics.

2. A measure of statistical independence

Let s be a random vector with entries s_i. Mutual independence of the variables $\{s_i \mid 1 \leqslant i \leqslant n\}$ is satisfied if and only if the joint probability density function, $p_s(v)$, equals the product of the marginal densities, $p_{s_i}(v_i)$:

$$p_s(v) = \prod p_{s_i}(v_i). \tag{5a}$$

This criterion is not easy to use under this form so that one prefers to resort to characteristic functions. Define the first characteristic function as $\phi_s(u) = \mathrm{E}\{\exp(iu^{\mathrm{T}}s)\}$, and the second as $\psi_s(u) = \ln \phi_s(u)$. If the variables s_i are mutually independent, then by definition the joint characteristic function $\phi_s(u)$ splits into the product of n marginal characteristic functions $\phi_{s_i}(u_i)$. This yields by taking the logarithm:

$$\psi_s(u) = \sum \psi_{s_i}(u_i). \tag{5b}$$

This equation means that the function of n variables, $\psi_s(u)$, separates into a sum of n functions each of one variable. This property seems much easier to use than (5a). For this purpose, it suffices to express u in terms of the unknown coefficients, F_{ij}, and to write the cancellation of all the terms involving several variables in the left-hand side. For instance, if $n=2$, we get

$$\psi_s(u_1, u_2) = \psi_{s_1}(u_1) + \psi_{s_2}(u_2),$$

which implies

$$\psi_e(F_{11}u_1 + F_{21}u_2,\ F_{12}u_1 + F_{22}u_2) = \psi_{s_1}(u_1) + \psi_{s_2}(u_2).$$

Such a procedure is possible if the characteristic function of the observation, $\psi_e(v)$, is known. In practice, it can always be estimated since e is observed. However, the estimation of probability density functions and characteristic functions requires a large superflous effort as explained in the following. Suppose we calculate the Taylor expansion of both sides of (5b). Then all the terms in the left-hand side where products between different variables u_i enter must be zero since there are no such terms in the right-hand side. Equating terms of both sides for any degree gives new equations. The coefficients of terms of degree N in the expansion are called *cumulants of order N*. For instance, for $n=2$, there are three cross-cumulants of order 4:

$$\begin{aligned} \Gamma_{13} &= M_{13} - 3M_{20}M_{11} = 0, \\ \Gamma_{22} &= M_{22} - M_{20}M_{02} - 2M_{11}^2 = 0, \\ \Gamma_{31} &= M_{31} - 3M_{02}M_{11} = 0, \end{aligned} \tag{6}$$

where $M_{ij} = \mathrm{E}\{s_1^i s_2^j\}$ denote the moments of order $i+j$, and the Γ'_{ij}s are the corresponding cumulants, $\Gamma_{ij} = \mathrm{cum}\{s_1^i, s_2^j\}$. General expressions of cumulants may be found in [1].

In this framework, one will restrict ourselves to pairwise independence. But pairwise independence of the outputs implies mutual independence because outputs are *linear functions* of the sources (remind that sources are assumed mutually independent). Pairwise cumulants have been shown to provide a sufficiently large set of equations to conclude, even for $n>2$ [2]. Now let us turn to the HJ algorithm which is an adaptive means to cancel high-order cumulants.

3. HJ adaptation rule

3.1. *Searching the zeros of a function*

Let $\Phi(z)$ be a real deterministic function defined as the expectation of a random function $\varphi[z, s(z)]$, $s(z)$ being a random variable depending on the unknown parameter, z. One of the simplest ways of finding the zeros of $\Phi(z)$ is to run the algorithm

$$z(k+1) = z(k) + \mu\Phi(z(k)),$$

$$\mu \text{ is a constant}, \qquad 0 < \mu < 1.$$

It is easy to see that if $\Phi(z)>0$, then z increases, whereas if $\Phi(z)<0$, z decreases. This iteration is thus able to find the zeros of $\Phi(z)$ that have a negative derivative. A similar iteration may be used to find the zeros with positive derivative just by changing the sign of μ. In a stochastic approximation context, $\Phi(z)$ is unknown to the user as well as the probability density of φ. However, one may substitute $\varphi[z, s(z)]$ for $\Phi(z)$ in the above iteration, yielding the Robbins-Monro stochastic iteration [7, pp. 42-43]:

$$z(k+1)=z(k)+\mu\varphi[z(k), s(z(k))]. \tag{7}$$

Convergence in probability of the Markov process $z(k+1)$ to a root of $\Phi(z)$ needs among other conditions that μ depends on k [8]. More precisely, the sequence $\mu(k)$ must be non-summable and $\mu(k)^2$ must be summable. If $\mu(k)$ does not tend to zero, then $z(k)$ will be able only to achieve a neighborhood of the root but there will be a non-zero residual. This is what occurs in the HJ algorithm, but it could be easily palliated if required, for instance by setting $\mu(k)=\mu_0/k$. However, a minimal positive value of $\mu(k)$ is desired in a non-stationary environment in order to track a slowly varying phenomenon. It turns out as we shall see in the next subsection that the HJ algorithm also defines a Markov process, and therefore could be studied with the help of the standard tools for this purpose [7]. As announced in our introduction, this is however out of our present scope and we postpone this study to a future paper.

3.2. Searching common zeros of several functions

Now, let the dimension of z be larger than 1, say n, and assume we observe n functions $\varphi_i[z, s(z)]$. Denote $\Phi_i(z)=\mathrm{E}\{\varphi_i[z, s(z)]\}$ and assume we know in advance that the system $\{\Phi_i(z)=0 \mid 1\leq i\leq n\}$ admits at least one solution. The previous iteration may be used along each coordinate in turn. In such a case, the ith component of z is updated only once each n steps, if n is the dimension of z, according to a relaxation scheme:

$$\text{step}\quad k=qn+r:\quad z_r(k+1)=z_r(k)$$
$$+\mu\varphi_r[z(k), s(z(k))],$$
$$0<r\leq n,\quad q\in\mathbb{N}.$$

For instance if $n=2$, the relaxation can be written as

k odd:

$$z_1(k+1)=z_1(k)+\mu\varphi_1[z(k), s(z(k))],$$

k even:

$$z_2(k+1)=z_2(k)+\mu\varphi_2[z(k), s(z(k))].$$

3.3. HJ algorithm as a pipelined search

The HJ algorithm may be seen as a pipelined version of the above algorithm. It acts exactly in the same manner as if n relaxations were executed in parallel. For the sake of clarity let us explicit the iteration in the $n=2$ case.

iteration k:

$$z_i(k+1)=z_i(k)+\mu\varphi_i[z(k), s(z(k))]$$
$$\forall i\in\{1,\ldots,n\}. \tag{8a}$$

The difference with the relaxation scheme described earlier is that iterations along each coordinate are computed all together at the same time step. If the step is very small, the principle does not change much and the difference may be viewed as a pipeling operation. On the other hand, if the step is too large, then algorithm (8) may not converge.

Omitting the high-pass filter that was originally implemented together with function g, the adaptation rule used in the HJ algorithm may be written as

$$c_{ij}(k+1)=c_{ij}(k)+\mu\varphi_{ij}[C(k), s(C(k))],$$
$$i\neq j,\ i,j\leq n,$$

where

$$\varphi_{ij}[C(k), s(C(k))]=f[s_i(C(k))]g[s_j(C(k))],$$

$$\text{with } s(C(k))=[I+C(k)]^{-1}e(k), \tag{8b}$$

and the functions $f(s)$ and $g(s)$ are suggested to be (i) different, (ii) odd [5]. Clearly, iteration (8b) is attempting to simultaneously cancel the $n(n-1)$ functions $\mathrm{E}\{f(s_i)g(s_j)\}$ with respect to the entries of C. Now assume the sources have zero-mean even probability densities. Then all odd order moments $\mathrm{E}\{x_i^{2p+1}\}$ are zero, and consequently all odd order moments $\mathrm{E}\{s_i^{2p+1}\}$ aim to vanish. This implies that if f and g are odd, $\mathrm{E}\{f(s_i)g(s_j)\}$ expand into a combination of cross cumulants of the form $\mathrm{cum}\{s_i^{2p+1}, s_j^{2q+1}\}$. This makes the connection between the choice of functions $f(s)$ and $g(s)$ made in [5] and the criterion defined in Section 2.

Let us review the conditions recommended for functions f and g: (i) They must be different from each other in order for the matrix C to be able to take unsymmetric values, this is necessry. Condition (ii) is sufficient (but not necessary) to insure the functions $F_{ij}(C)$ to coincide with a combination of cross-cumulants, thus vanishing when the components of $s(t)$ are independent. In the next section, we shall point out some of the convergence limitations which are definitely due not only to the search algorithm but also to the choice of the functionals themselves.

3.4. Separation of statistically dependent signals

As shown in the previous sections, it is sufficient that the sources x_1 and x_2 satisfy for at least two values of the pair $(i,j)\in\mathbb{N}^2$: $\mathrm{cum}\{x_1^i, x_2^j\}=0$. It turns out that variables x_1 and x_2 are statistically independent if and only if *all the cross-cumulants* are equal to zero. But it is possible to exhibit two variables having ony a small number of null cross-cumulants. The separation is then still possible if there are at least two of them (known in advance) that are null. Of course the exact form of the cumulants must be known in advance. For instance, let h be any even function and x a zero-mean random variable symmetrically distributed. Then a mixture of the sources $x_1=x$ and $x_2=h(x)$ may be identified by using 2 functionals of the form $\mathrm{E}\{x_1^p x_2^q\}$ with p odd, since these moments are always null.

4. Behavior of the HJ algorithm

The purpose of this section is to show various conditionings of the HJ algorithm, and in particular those that cause problems. In order to bring out more insights in its behavior, the functionals on which algorithm (8b) is based are accurately estimated on a few examples in Section 4.3.

4.1. Explicit computation of the functionals cancelled implicitly by HJ

For given inputs, it is possible to estimate accurately both functionals as functions of the unknown transform parameter, C. For this purpose, assume we observe $e(t)$ for $1\leqslant t\leqslant T$. Then the functionals $\Phi_{ij}(C)$ may be approximated by

$$\bar{\Phi}_{ij}(C)=\frac{1}{T}\sum_{t=1}^{T} f[s_i(t)]g[s_j(t)],$$

$$\text{with } s(t)=[I+C]^{-1}e(t). \qquad (9)$$

For each value of C to be scanned, a large average must be calculated involving a rather important computational burden. For this reason, we have decided to restrict ourselves to a one-dimensional study ($n=2$ but one parameter), since incidentally this does not restrict the generality as pointed out below.

4.2. Restriction to a one-dimensional study

Our present investigation focuses on the case $n=2$ for the sake of clarity; nonetheless the results are representative of the behavior at higher dimensions since all outputs are processed pairwise. Now, matrix C has only two free parameters, c_{12} and c_{21}. The first goal of the algorithm is to provide uncorrelated outputs, and the next is concerned with higher-order independence. This first task is less interesting to study for two reasons: (a) there exist many classical methods carrying it out, (b) the HJ algorithm has been shown to converge quite rapidly to uncorrelated outputs whereas the next phase dealing with higher order independence takes a fairly larger amount of iterations [5]. In this section, we shall restrict the study to uncorrelated (but statistically dependent) inputs. In such a case, the matrix C to estimate is necessarily skew-symmetric. So denote from now on $c_{12} = \theta$ and $c_{21} = -\theta$. Assume the inputs $e_1(t)$ and $e_2(t)$ are uncorrelated (at order 2). Then algorithm HJ is attempting to cancel simultaneously

$$\Phi_{12}[\theta] = \mathrm{E}\{f(s_1)g(s_2)\}$$

and

$$\Phi_{21}[\theta] = \mathrm{E}\{f(s_2)g(s_1)\},$$

where

$$s_1(t) = \frac{e_1(t) - \theta e_2(t)}{1+\theta^2}$$

$$\text{and} \quad s_2(t) = \frac{e_2(t) + \theta e_1(t)}{1+\theta^2}. \tag{10}$$

4.3. Simulation examples

In this section, we have retained $f(s) = s^3$ and $g(s) = \mathrm{atan}(10\, s)$. Figures 1 and 2 show the functionals $\bar{\Phi}_{ij}(\theta)$ obtained for an average of length $T = 2000$ with two types of inputs. Inputs are obtained from independent sources via the orthogonal transform

$$A = [0.319 \;\; 0.948; \; -0.948 \;\; 0.319].$$

In the first case (Fig. 1), the sources are identically and uniformly distributed white processes, with zero-mean and unit variance. The ideal solutions are, from (4), $\theta = 3$ and $\theta = -0.3$. Attention of the reader is attracted on the fact that Figs. 1(b) and 2(b) plot Φ_{21} as a function of $-c_{21}$ and not c_{21}, in order to make it easier to compare solutions θ obtained in Figs. 1(a, b) and 2(a, b). Thus, attractors of HJ are the zeros with *positive derivative* in Figs. 1(b) and 2(b), and *negative derivative* in Figs. 1(a) and 2(a).

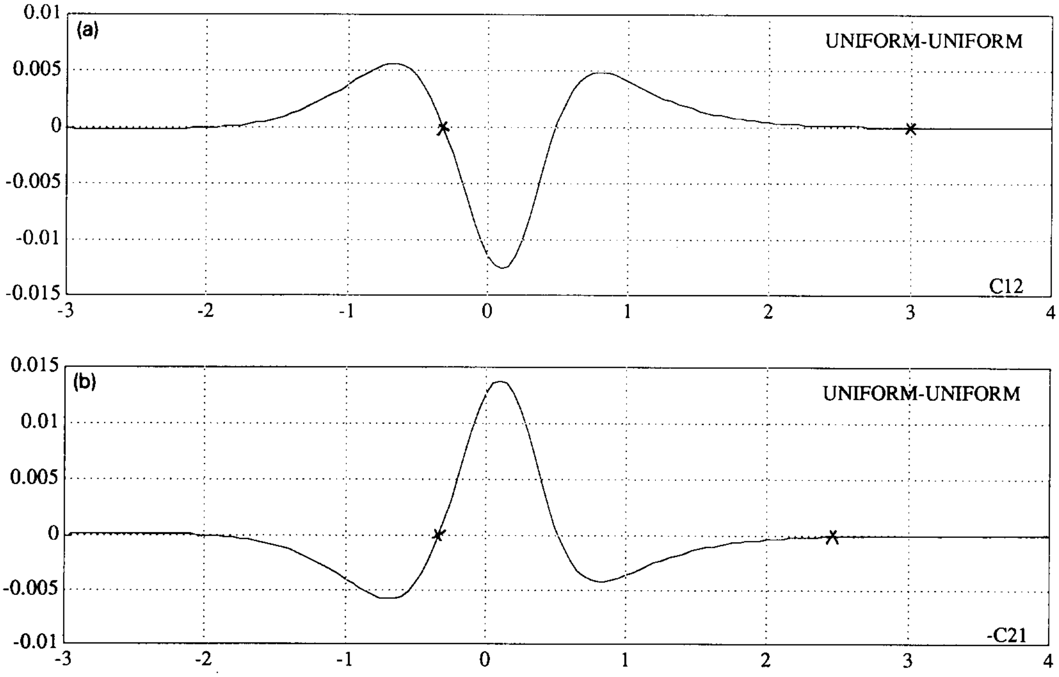

Fig. 1. Separation of two identically and uniformly distributed processes mixed by an orthogonal transform. (a) Functional $\Phi_{12}[\theta]$ as a function of $\theta = c_{12}$. Attractors for c_{12} are −0.3, +3 with associated attraction basins:]−2.1, 0.5[,]0.5, +∞[. (b) Functional $\Phi_{21}[\theta]$ as a function of $\theta = -c_{21}$. Attractors for $-c_{21}$ are −0.3 and +2.4 with associated attraction basins:]−1.8, 0.5[and]0.5, +∞[. Attractor +2.4 is inaccurately estimated (we expected +3) because the derivative of $\Phi_{21}[\theta]$ is almost null; an other consequence is that this attractor is fairly slow. Attractors are spotted with a cross.

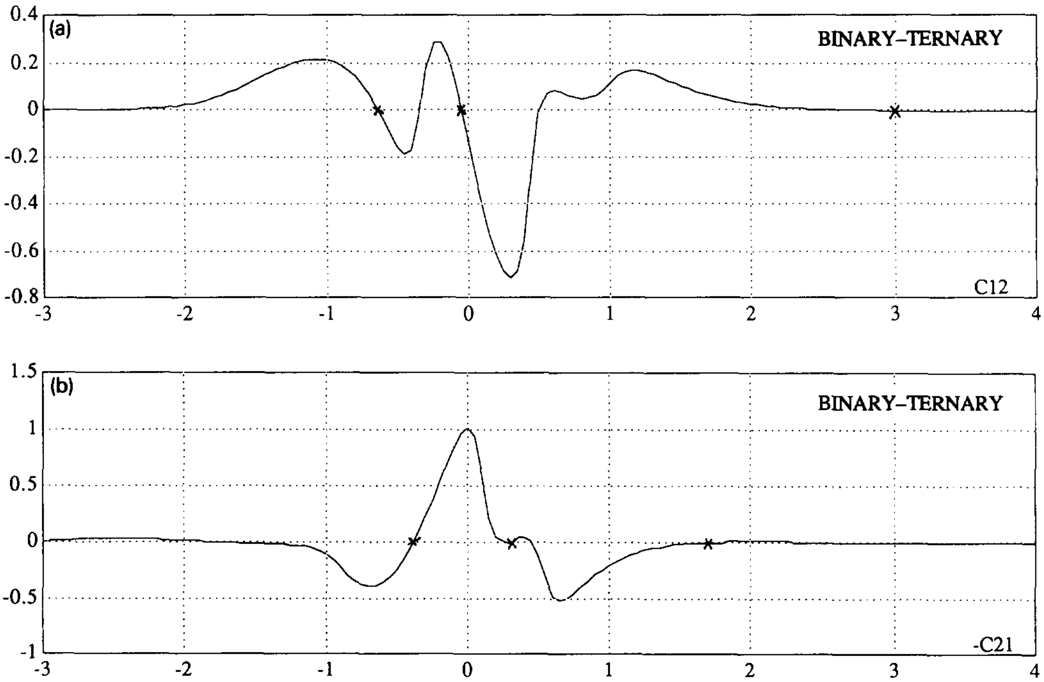

Fig. 2. Separation of binary and ternary white random processes mixed by an orthogonal transform. (a) Functional $\Phi_{12}[\theta]$ as a function of $\theta = c_{12}$. (b) Functional $\Phi_{21}[\theta]$ as a function of $\theta = -c_{21}$. After a precise computation of the functionals about the attractors and unstable points we obtain the following attractor basins: attractors for c_{12} are -0.6, -0.1 and $+3$ with attraction basins $]-2, -0.3[$, $]-0.3, 0.5[$ and $]0.5, +\infty[$; attractors for $-c_{21}$ are -0.3, $+0.3$ and $+1.7$ with attraction basins $]-1.8, 0.2[$, $]0.2, 0.5[$ and $]0.5, +\infty[$.

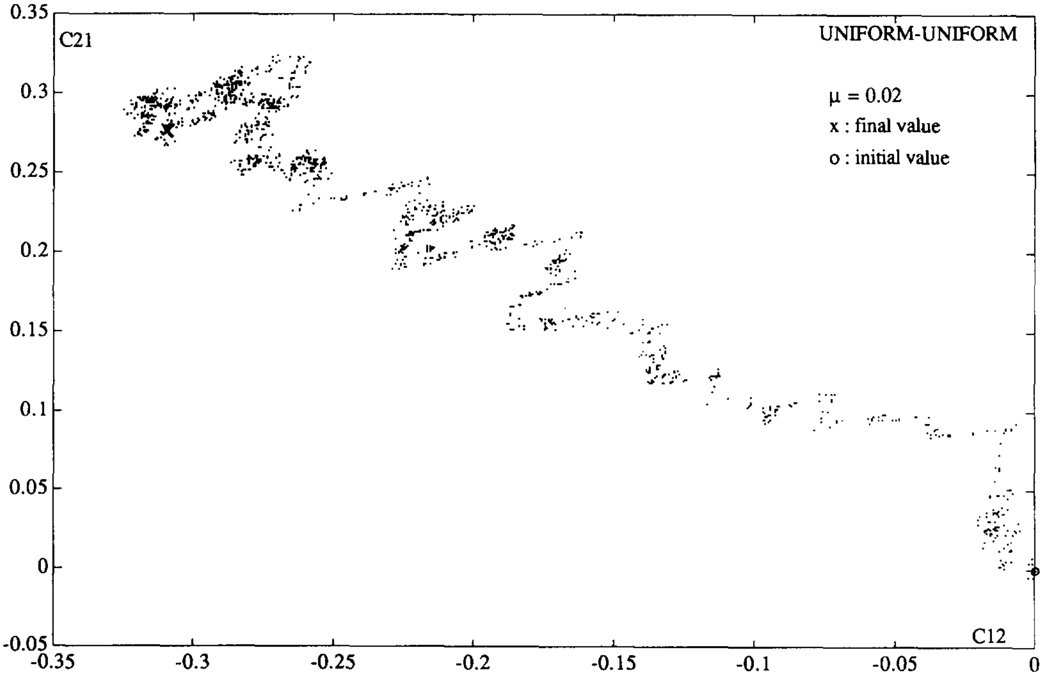

Fig. 3. Evolution of 2000 successive values taken by (c_{12}, c_{21}) in the conditions described in Fig. 1. The values of parameters (c_{12}, c_{21}) were left free so that we may have $c_{12} \neq -c_{21}$. The algorithm converges to $(-0.3, 0.3)$ which is the value expected. The cross indicates the averaged value of the last 10 iterates.

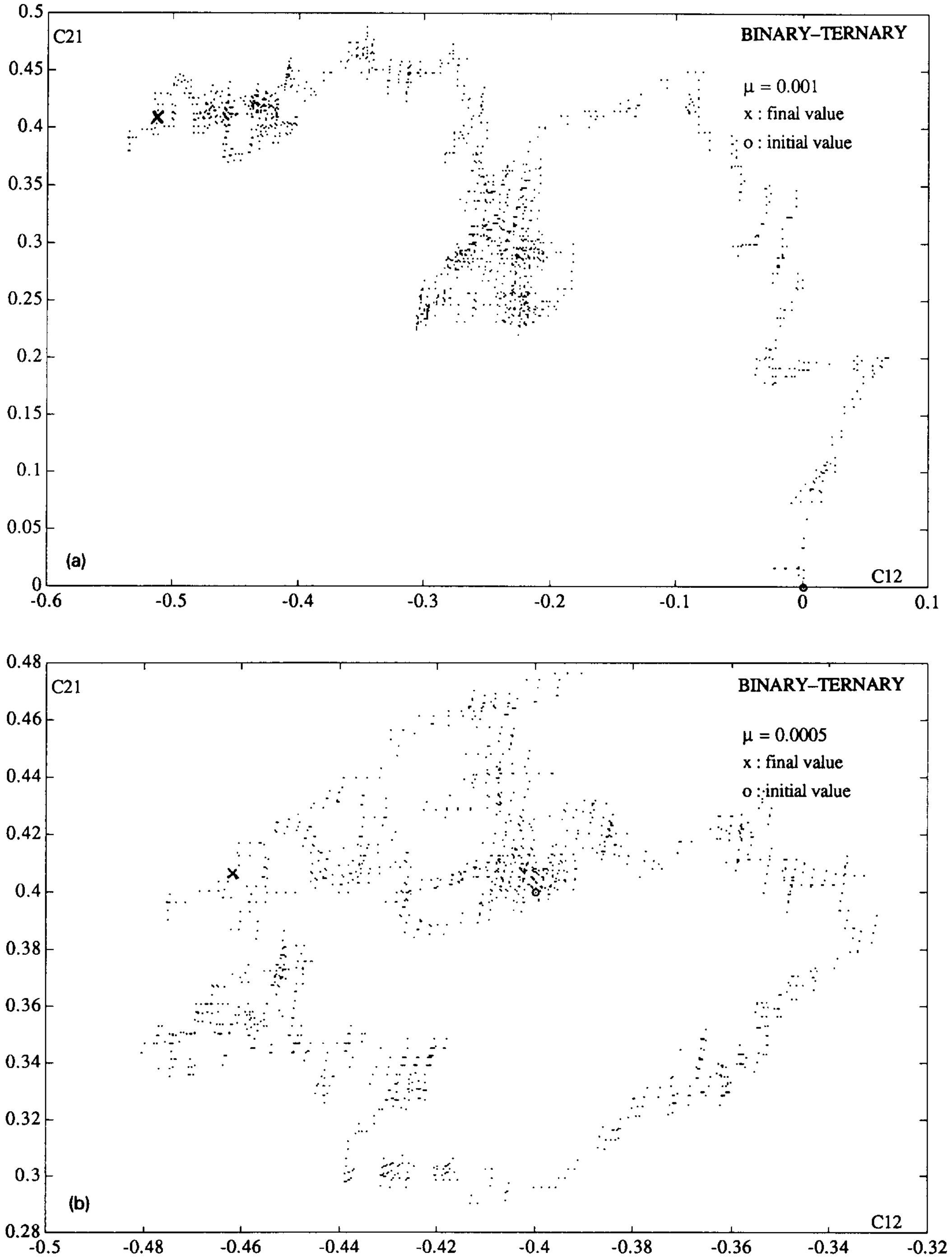

Fig. 4. Evolution of the values taken by (c_{12}, c_{21}) in the conditions described in Fig. 2. Again parameters (c_{12}, c_{21}) were left free. (a) The algorithm seems to converge to a neighborhood of $(-0.5, 0.4)$, which is not correct but anticipated by Fig. 2. The cross indicates the averaged value of the last 10 iterates. (b) This shows more precisely the behavior of the algorithm around the point $(-0.5, 0.4)$ with a starting value $(-0.4, 0.4)$ and a smaller step imposed by $\mu = 5 \cdot 10^{-4}$. The algorithm is not converging but describes a cycle in the plane. Within 2000 iterations, the algorithm has described one round and half. This shows the drawbacks that may appear when running the algorithm with unsymmetrically distributed sources.

In Fig. 1 we can see that the attractors obtained (spotted with a cross) coincide approximately with the solutions expected. However, one can check that if $\{c_{12}, -c_{21}\}$ is intialized to $\{-2.5, -2.5\}$, then the algorithm diverges to $\{-\infty, -\infty\}$; if it is initialized to $\{3, 3\}$ then it converges fajrly slowly. As an example, the behavior of the actual adaptive algorithm is represented in the $\{c_{12}, c_{21}\}$-plane in Fig. 3. The correct value is attained after 2000 iterations with $\mu = 0.02$.

On the other hand, our second example sheds light on more embarassing results: some correct attractors have split into two close spurious attractors (Fig. 2). In this example, the sources x_1 and x_2 take only values in $\{-0.65, 1.53\}$ and $\{-\sqrt{2}, 0, \sqrt{2}\}$ respectively, with probabilities $\{0.70, 0.30\}$ and $\{1/4, 1/2, 1/4\}$. As a consequence, the sources are zero-mean and of unit variance, and source x_1 is unsymmetrically distributed. Let us dwell a little longer on this particular example. Figure 2(a) shows that attractor -0.3 does not exist any more, but attractors -0.6 and -0.1 have appeared; as a result there is an unstable basin limit at -0.3. Figure 4(a) reports a typical experiment run with null starting value that led to a solution $\{c_{12}, -c_{21}\}$ belonging to a neighborhood of $\{-0.6, -0.3\}$. Figure 4(b) shows that the algorithm is actually cycling in the $\{c_{12}, c_{21}\}$ plane. This may be observed in Fig. 5 as well, where the values taken by the outputs tend periodically to cluster. Thus we get a quite inaccurate solution, which was anticipated by our analysis. Note that attractor -0.1 has been skipped in that case, because of too little averaging at the beginning of the process. Recall that Fig. 2 corresponds to sources with unsymmetric probability densities.

In the stochastic HJ algorithm, there is always a phase of fast convergence corresponding to the decorrelation up to order 2, and a phase of slow evolution where the algorithm tries to obtain out-

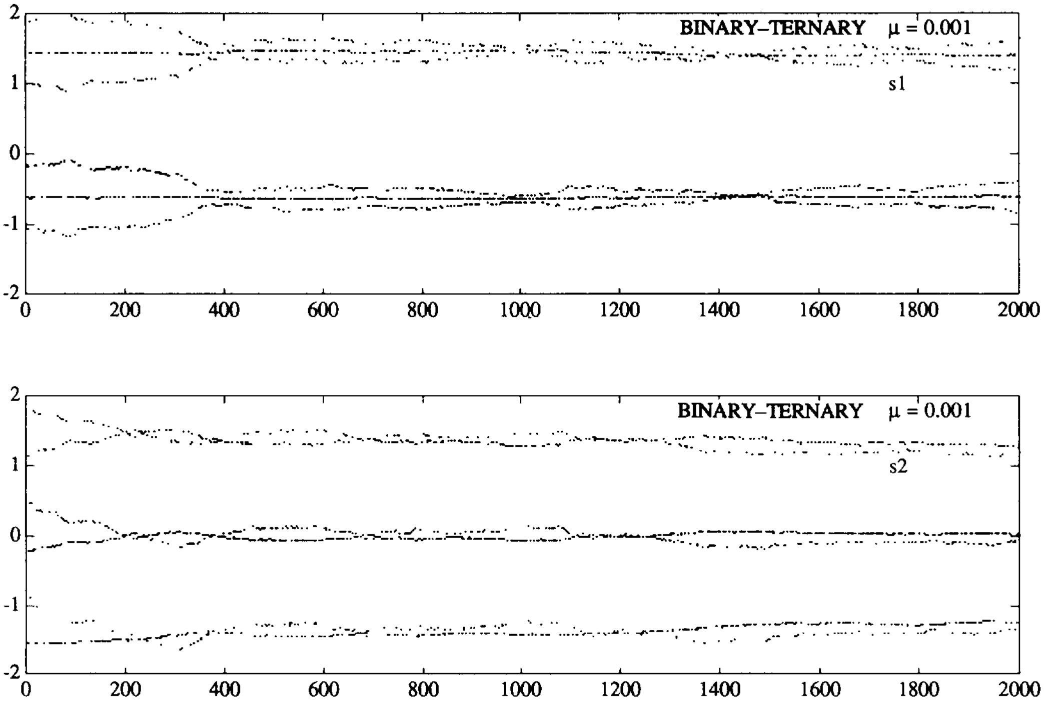

Fig. 5. Since sources take discrete values in finite sets, it is possible to check convergence of the algorithm in a glance by looking at the output signals. We may notice that the values are clustering periodically, but do not tend to converge as the number of iterations tend to infinity. This confirms the remarks made in Fig. 4(b).

puts statistically independent up to higher orders. The success of the second phase is in general compromised when one source is unsymmetrically distributed. Our simulations focused on the second phase since the mixing matrix was orthogonal.

5. Improvements to the method

The Robbins-Monro algorithm has specific limitations that are shared with the HJ algorithm. For instance, the speed of convergence of a sequence $x(k)$ to a zero x_0 of $\Phi(x)$ depends upon the ratio $\min\{\Phi(x)/(x-x_0)\}$. On the other hand, odd moments are not always the adequate tool for testing statistical independence. The possible improvements are two-fold. Firstly, the iterative search can be speeded up, or even replaced by a direct computation when it is feasible [3] in order to get rid of the initialization sensitivity. In the case where the functionals f and g are polynomials, this could also be taken into account in the zero-finding routine. Secondly, one can change the nature of the functionals, by using cumulants for instance. More precisely, a single functional could be defined as the sum of the squares of all cross-cumulants of order three and four, and would then be a polynomial in the (unknown) mixing components and in the (known) observation cumulants. Its minimization would consequently be very easy to perform and would allow us to cope with sources with unsymmetric densities. In [2, 3] some of these improvements have been implemented and first extensions to convolution mixings are discussed.

As a conclusion, it must be said that this problem is related to the solution of overdetermined systems of non-linear equations with *several variables* which are difficult to solve, even if the non-linearity is merely polynomial because the ring of polynomials is not principal any more in this case.

References

[1] D.R. Brillinger, *Time Series, Data Analysis and Theory*, Holden-Day, San Francisco, CA, 1981.

[2] P. Comon, "Separation of sources using higher-order cumulants", *SPIE Conference*, Vol. 1152, Advanced Algorithms and Architectures for Signal Processing IV, San Diego, California, 6-11 August 1989, pp. 170-181.

[3] P. Comon, "High-order separation, application to detection and localization", in: L. Torres, E. Masgrau and M.A. Lagunas, eds., *Signal Processsing V, Theory, and Applications, Proceedings of EUSIPCO-90*, Barcelona, 18-21 September 1990, Elsevier, Amsterdam, 1990, pp. 277-280.

[4] P. Comon and J.F. Cardoso, "Eigenvalue decomposition of a cumulant tensor with applications", *SPIE Conference*, Vol. 1348, Advanced Algorithms and Architectures for Signal Processing IV, San Diego, California, 10-12 July 1990, pp. 361-372.

[5] C. Jutten and J. Herault, "Separation of sources, Part I", *Signal Processing*, Vol. 24, No. 1, July 1991, pp. 1-10.

[6] J.L. Lacoume and P. Ruiz, "Sources identification: A solution based on cumulants", *IEEE ASSP Workshop*, Minneapolis, August 1988.

[7] L. Ljung and T. Soderstrom, *Theory and Practice of Recursive Identification*, MIT Press, Cambridge, MA, 1983.

[8] H. Robbins and S. Monro, "A stochastic approximation method", *Annals of Math. Statist.*, 1951, pp. 400-407.