Elsevier on-line version:
http://dx.doi.org/10.1016/j.sigpro.2008.09.016

Recurrent networks for separating extractable-target nonlinear mixtures. Part I: non-blind configurations

Yannick Deville and Shahram Hosseini

Laboratoire d'Astrophysique de Toulouse-Tarbes
Université de Toulouse, CNRS
14 Av. Edouard Belin, 31400 Toulouse, France.
Email: ydeville@ast.obs-mip.fr , shahram.hosseini@ast.obs-mip.fr
Phone: +33 5 61 33 28 24, fax: +33 5 61 33 28 40

(corresponding author: Yannick Deville; see address above)

Abstract While most reported source separation methods concern linear mixtures, we here address the nonlinear case. Even for a known nonlinear mixing model, creating a system which implements the exact inverse of this model is not straightforward for most nonlinear models. We first define a large class of possibly nonlinear models, i.e. "Additive-Target Mixtures" (ATM), for which this inversion may be achieved thanks to the nonlinear recurrent networks that we propose to this end. We then further extend this approach to the "extractable-target mixtures" (ETM) that we also introduce in this paper. We illustrate these general approaches for two specific classes of mixtures, i.e. linear-quadratic mixtures, and quadratic ones. We then focus on our networks suited to linear-quadratic mixtures and we provide a detailed analysis of their equilibrium points and their stability. This allows us to introduce an automated procedure for selecting their free weights so as to guarantee the stability of a separating point for any source signals. Test results show the effectiveness of this approach for various types of source signals.

Keywords source separation, inverse problem, nonlinear mixture, nonlinear recurrent neural network, stability.

1 Problem statement

Source separation is a generic signal, image, and data processing problem [1]. The situation that it addresses may be defined as follows, considering one-dimensional signals for the sake of simplicity. A set of P observed signals $x_i(n)$ are available, e.g. from sensors. These observations are "mixtures" of a set of N unknown source signals $s_j(n)$, i.e. the observation vector $x(n) = [x_1(n), \ldots, x_P(n)]^T$ may be expressed with respect to the source vector $s(n) = [s_1(n), \ldots, s_N(n)]^T$ (where ^{T} denotes transposition), according to a mixing operator M:

$$x(n) = M[s(n)], \tag{1}$$

where the class of functions corresponding to this model M is usually fixed, and its parameter values may be known or not, as detailed below. Source separation methods then aim at retrieving the unknown source signals $s_j(n)$ from their observed mixtures $x_i(n)$. The most natural methods therefore consist in applying an unmixing (or separating) operator U to these observations, to obtain the output vector

$$y(n) = U[x(n)]. \tag{2}$$

The issue is then essentially to make the model U equal to the inverse of the model M, so that $y(n)$ becomes equal to the desired vector $s(n)$ of source values.

Within this overall framework, different cases may be defined, especially depending: (i) on the class of functions considered for the mixing model M and (ii) whether the parameter values of this model are known or not. Up to now, most reported investigations [1] have been focused on relatively simple mixing models, i.e. linear ones. The development of unmixing models corresponding to a linear model with given parameters has only required limited effort, and the emphasis has been mainly put on the difficulty which arises when the mixing parameters are unknown. One then has to estimate these parameters only from the observations, thus achieving *blind* source separation (BSS). Many BSS methods have been developed during the last decades, mainly using the concept of Independent Component Analysis [1], and more recently by resorting to Sparse Component Analysis or Non-negative Matrix Factorization (see e.g. the overview in [2] and references therein). BSS methods essentially aim at recovering the waveforms of the source signals, i.e. at estimating the source signal values e.g. up to fixed scale factors and additive constants, and with possible permutations between the indices of the original source signals and their restored versions (this corresponds to classical BSS indeterminacies [1]).

Our focus in this paper is different, i.e. we aim at providing source separation methods for *nonlinear* mixtures. Such mixtures have already been considered in the literature (see e.g. the survey in [3]) but in a much less detailed way than linear mixtures. It is well known [4], [5] that the independence hypothesis is not sufficient for achieving BSS from general nonlinear mixtures, because of the very large indeterminacies which make the nonlinear BSS problem ill-posed. A natural idea for reducing the indeterminacies is to constrain the structure of mixing and separating models to belong to a certain class of transforms. This additional constraint can be viewed as a regularization of the initially ill-posed problem [6], [7]. This is the approach that we use in this paper, where we consider a particular class of mixing models. Anyway, nonlinear mixtures are much more difficult to handle than linear ones, even when considering a nonlinear mixing model M corresponding to a fixed function with known parameter values: in such a *non-blind* configuration, creating a system which implements the inverse of the considered mixing model (and which could then be extended to the blind configuration) is not straightforward for most nonlinear models, as detailed further in this paper. This is therefore the problem that we aim at solving in the first part of this paper, for a large class of nonlinear mixtures that we introduce to this end. We reach this goal thanks to new recurrent artificial neural network structures, that we derive hereafter as extensions of the source separation networks proposed in previous papers for linear mixtures.

More precisely, in the simplest source separation configuration considered in the literature, the observations are Linear Instantaneous (LI) mixtures of the source signals. The above mixing operator M may then be defined by a matrix A, composed of mixing coefficients a_{ij}, and the mixing model (1) may be expressed by the matrix-vector product

$$x(n) = As(n). \tag{3}$$

One of the very first reported solutions for this LI source separation problem is the Hérault-Jutten artificial neural network [8]. This network has a recurrent (or feedback) structure: each of its outputs $y_i(n)$ consists of an LI combination of

- all *other* outputs $y_j(n)$ with $j \neq i$, using combination coefficients, i.e. neural network "weights", denoted l_{ij} hereafter,
- and input $x_i(n)$, with a weight fixed to 1.

In a blind configuration, the weights l_{ij} are estimated from the output signals $y_i(n)$, using the unsupervised adaptation (i.e. "training") algorithm which was first proposed by Hérault and Jutten [8] and then analyzed by several authors [9]-[12].

For this class of mixtures, a recurrent structure is not mandatory however, i.e. the same class of LI mappings from the signals $x_i(n)$ to the signals $y_i(n)$ may also be achieved by a feedforward structure. In the latter structure, each output $y_i(n)$ is derived only as an LI combination of all inputs, typically with a weight equal to 1 for the input $x_i(n)$ which has the same index i as the considered output $y_i(n)$, and with weights again denoted l_{ij} for all *other* inputs $x_j(n)$, i.e. for $j \neq i$. This structure was e.g. used by Macchi and Moreau [13]-[14]. This is the mostly used LI network today, because it is simpler than the LI recurrent structure and it avoids the stability constraints of that structure.

Still for LI mixtures, extended structures were also proposed, with additional weights l_{ii} between each input and output having the *same* index i. Corresponding adaptation algorithms for the blind context were e.g. introduced by Cichocki et al. in [15] and analyzed by Deville in [16]. Beyond LI mixtures, many investigations reported in the literature are devoted to convolutive mixtures. The same evolution, from recurrent to feedforward separating structures, also occurred for these convolutive mixtures, because feedforward convolutive networks provide the considered convolutive mappings while avoiding some stability constraints of recurrent convolutive structures.

On the contrary, feedforward structures are much less attractive for nonlinear mixtures, which are now receiving increasing attention. This restriction stems from the fact that feedforward structures consist in expressing the desired network outputs, i.e. the source signals, directly with respect to the observed signals. In other words, they require us to derive an explicit analytic expression of the *inverse* of the mixing (i.e. direct) model M, which is not possible for given arbitrary nonlinear models M: for instance, if M is a higher-order polynomial model, creating an associated feedforward separating structure would mean deriving the analytic solution of the higher-order polynomial mixing equations, which is not feasible. For the sake of clarity, let us contrast this problem with the case of linear mixtures: to derive a feedforward structure for the latter mixtures, we only have to solve *linear* equations, which is straightforward and yields the linear feedforward structure that we discussed above. While feedforward structures are therefore not suited to nonlinear mixtures, we will show in the next section that the above recurrent structures may be extended to a large class of nonlinear mixtures. Let us stress again that this first part of our investigation is focused on separating structures themselves and therefore on the non-blind configuration, as opposed to the adaptation of the parameters of such structures in a blind context. The latter topic will be addressed in the second part of this paper.

The remainder of this first part of our paper is therefore organized as follows. Section 2 deals with separating structures. Based on the above discussion, we focus on recurrent structures in that section. We first provide a (presumably new) analysis of the operation of existing linear recurrent structures. This then allows us to extend them to new classes of nonlinear mixtures, that we introduce at this stage. We illustrate the proposed general approach by applying it to several specific classes of nonlinear mixtures, especially linear-quadratic mixtures. We conclude this section by defining how this investigation may then be continued, once our approach has been used for creating networks suited to a given type of mixtures. To illustrate the proposed procedure, we then focus on linear-quadratic mixtures in Section 3. We derive the fixed points of the proposed networks and analyze their stability. We thus propose a method for selecting their weights so as to avoid the divergence issue exhibited by the basic version of our networks. Various tests are used in Section 4 to confirm this theoretical analysis and to show the performance of our networks for different types of source signals. Conclusions are drawn from this investigation in Section 5.

We only consider real-valued signals and models hereafter, but several aspects of this paper are directly applicable or extendable to the complex case. When dealing with separating structures, we consider the classical "determined" configuration, i.e. the case when the numbers P and N of observations and sources are equal.

2 Proposed recurrent separating structures for nonlinear mixtures

2.1 Analysis of linear recurrent separating structures

Based on the above considerations, we will focus on recurrent separating structures in the remainder of this paper and we will consider increasingly complex mixing models. The simplest case therefore concerns $P = 2$ LI mixtures of $N = 2$ sources. The LI mixing model (3) then reads explicitly

$$x_1(n) = a_{11}s_1(n) + a_{12}s_2(n) \tag{4}$$

$$x_2(n) = a_{21}s_1(n) + a_{22}s_2(n). \tag{5}$$

It may also be expressed with respect to the "normalized" source signals which are defined as[1]

$$s'_i(n) = a_{ii}s_i(n) \qquad \forall\ i \in \{1, 2\} \tag{6}$$

and which are equal to the contribution of the source with index i in the observation with the same index i. Defining

$$L_{ij} = -\frac{a_{ij}}{a_{jj}} \qquad \forall\ i \neq j \in \{1, 2\}, \tag{7}$$

(where we assume all a_{jj} are non-zero), the above mixing equations (4)-(5) may be rewritten as

$$x_1(n) = s'_1(n) - L_{12}s'_2(n) \tag{8}$$

$$x_2(n) = -L_{21}s'_1(n) + s'_2(n). \tag{9}$$

Source separation then aims at extracting both signals $s'_i(n)$ from these mixtures (up to classical indeterminacies [1]). As stated in the previous section, this may be achieved by using the Hérault-Jutten recurrent structure. We here express the operation of this structure in a way which has not been reported in the literature to our knowledge, and which will then allow us to extend this structure to much more general mixtures. For each time n, this structure receives the couple of observations $(x_1(n), x_2(n))$ and computes the values of its outputs y_i by performing a recurrence. We denote as m the index associated to this recurrence and $y_i(m)$ the successive values of each output in this recurrence performed at time n [2]. This recurrence reads

$$y_1(m+1) = x_1(n) + l_{12}y_2(m) \tag{10}$$

$$y_2(m+1) = x_2(n) + l_{21}y_1(m) \tag{11}$$

where l_{ij} are the weights of this neural network[3]. The new values $y_i(m+1)$ are then used as the input data of the next occurence of the loop associated to this recurrence. The corresponding network may then also be considered as a looped structure (see Fig. 1), where the outputs $y_i(m)$ are fed back and combined with the inputs $x_i(n)$, to create new adder outputs $y_i(m+1)$, which then become the new outputs $y_i(m)$ for the next round through the loop. Now consider the situation when the network weights are set to [8]:

$$l_{ij} = L_{ij} \qquad \forall\ i \neq j \in \{1, 2\}. \tag{12}$$

[1] The dual normalization, i.e. $s'_i(n) = a_{ij}s_j(n)$ $\forall\ i \neq j \in \{1, 2\}$, may be used for this mixing model and for the other models subsequently considered in this paper. For the sake of brevity, this is not detailed hereafter.

[2] These successive output values therefore also depend on n. This index n is omitted in the notations $y_i(m)$, in order to improve readability and to focus on the recurrence on outputs for given input values $x_1(n)$ and $x_2(n)$.

[3] We use lower-case letters in notations for the weights of the separating network and upper-case letters for the parameters of the mixing model.

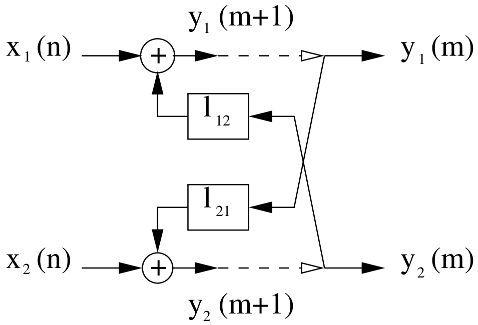

Figure 1: Loop of linear instantaneous recurrent network.

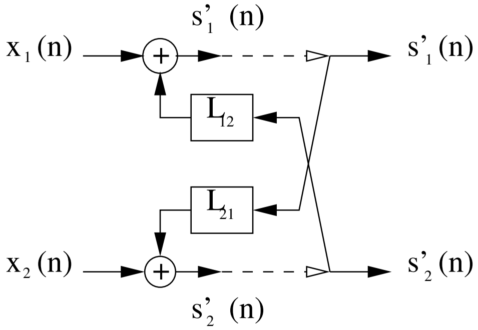

Figure 2: Fixed point of loop of linear instantaneous recurrent network.

Let us assume that, at step m of the recurrence, we obtained

$$y_i(m) = s'_i(n) \qquad \forall\ i \in \{1, 2\} \tag{13}$$

and let us consider the next occurence of the recurrence loop. Fig. 1 here becomes Fig. 2 and the network operates as follows. The outputs $y_i(m)$ are fed back, with the corresponding network weights l_{ij}, and added to the observations. Each observation $x_i(n)$ defined by (8)-(9) consists of: (i) a "target term", which is the part of this observation that we would like to keep and which is equal to $s'_i(n)$ and (ii) an "interfering term", which is the part of this observation that we would like to remove and which is equal to $-L_{ij}s'_j(n)$, with $j \neq i$. The major phenomenon is that, in the conditions considered here, each feedback term $l_{ij}y_j(m)$ becomes an exact "cancelling term", i.e. it exactly compensates for the above interfering term, so that the output of each adder becomes exactly equal to the target term $s'_i(n)$. We thus get

$$y_i(m+1) = s'_i(n) \qquad \forall\ i \in \{1, 2\}, \tag{14}$$

i.e. condition (13) is still met after running one occurence of the loop. It will therefore be met endlessly if the loop is executed again and again in these conditions. In more formal terms, this means that, in the conditions defined by (8)-(9) and (12), the point defined by (13) is a fixed point (i.e. an equilibrium point) of recurrence (10)-(11). This may be directly checked mathematically from all these equations, instead of considering Fig. 1 and 2.

In this paper, we aim at extending this analysis to nonlinear mixtures, and we therefore now introduce a first class of nonlinear mixtures suited to this approach.

2.2 Proposed additive-target mixtures

2.2.1 General model

Based on the above results, we here introduce a general class of possibly nonlinear mixtures[4]. We call this model "Additive-Target Mixtures" (ATM), since it corresponds to observations which read

$$x_i(n) = T_i[s(n)] - I_i[s(n)] \qquad \forall\, i \in \{1, \dots, P\} \tag{15}$$

where $T_i[s(n)]$ and $I_i[s(n)]$ are respectively the target and interfering terms of the considered model, i.e. the components of $x_i(n)$ that we aim at keeping and removing in the network outputs.

The target terms $T_i[s(n)]$ may take different forms in our approach. In the simplest configuration, they are equal to the source signals $s_i(n)$, up to classical scale (or filter) indeterminacies. In other words, they are typically equal to the signals $s'_i(n)$ which appear in the normalized version of the mixing equations, which was developed above for LI mixtures (see (8)-(9)), and which will be detailed below for specific nonlinear mixtures. The mixing model (15) then becomes

$$x_i(n) = s'_i(n) - I_i[s(n)] \qquad \forall\, i \in \{1, \dots, P\}. \tag{16}$$

Other configurations involve more general forms for the target terms $T_i[s(n)]$. Anyway, since each target term $T_i[s(n)]$ aims at extracting a source, it only depends on *one* source, which is selected to be $s_i(n)$. Each target term may then be expressed as $T_i[s_i(n)]$. Each single-input single-output function T_i is assumed to be invertible[5] (at least for the domain that $s_i(n)$ is allowed to span). Its inverse is denoted T_i^{-1}.

We now illustrate the above two mixing configurations by means of specific examples, for which we will then also detail associated separating networks.

2.2.2 First application: linear-quadratic mixtures

One of the simplest extensions of the LI model (4)-(5) consists in considering instantaneous mixtures which also include quadratic cross-terms, i.e. terms proportional to the product of both source signals. The observations then read

$$x_1(n) = a_{11}s_1(n) + a_{12}s_2(n) + b_1 s_1(n)s_2(n) \tag{17}$$
$$x_2(n) = a_{21}s_1(n) + a_{22}s_2(n) + b_2 s_1(n)s_2(n). \tag{18}$$

Still using (6)-(7) and now defining

$$Q_i = -\frac{b_i}{a_{11}a_{22}} \qquad \forall\, i \in \{1, 2\}, \tag{19}$$

(where we still assume all a_{jj} are non-zero), the above mixing equations (17)-(18) may be rewritten as

$$x_1(n) \;=\; s'_1(n) - L_{12}s'_2(n) - Q_1 s'_1(n)s'_2(n) \tag{20}$$
$$x_2(n) \;=\; -L_{21}s'_1(n) + s'_2(n) - Q_2 s'_1(n)s'_2(n). \tag{21}$$

So, this mixture model is indeed a specific case of the class of models defined in (15), where each target term $T_i[s(n)]$ is restricted to $s'_i(n)$, and each interfering term $I_i[s(n)]$ consists of all other components of $x_i(n)$ in (20)-(21)[6].

[4] This model itself may be defined for arbitrary numbers P and N of observations and sources. However, as stated above, when subsequently developing corresponding separating structures, we will focus on determined mixtures, i.e. $P = N$.

[5] This assumption is made for the sake of simplicity, i.e. it may even be avoided for specific mixtures, especially when each interfering term $I_i[s(n)]$ may be expressed with respect to all target terms $T_i[s_i(n)]$ instead of all signals $s_i(n)$, and provided one only aims at extracting the target terms $T_i[s_i(n)]$, not at subsequently retrieving the signals $s_i(n)$ from them.

[6] Up to the minus sign that we introduced in (15) to simplify the correspondence between mixing and separating parameters. This comment also applies to the mixing model considered in Section 2.2.3.

2.2.3 Second application: quadratic mixtures

Another simple nonlinear mixing model only consists of quadratic terms, i.e. auto-terms and cross-terms. It reads

$$x_1(n) = [a_{11}s_1(n)]^2 + [a_{12}s_2(n)]^2 + b_1 s_1(n)s_2(n) \tag{22}$$

$$x_2(n) = [a_{21}s_1(n)]^2 + [a_{22}s_2(n)]^2 + b_2 s_1(n)s_2(n). \tag{23}$$

Still using (6), (19) and now defining

$$L_{ij} = -\frac{a_{ij}^2}{a_{jj}^2} \qquad \forall\, i \in \{1,2\}, \tag{24}$$

the above mixing equations (22)-(23) may be rewritten as

$$x_1(n) = [s_1'(n)]^2 - L_{12}[s_2'(n)]^2 - Q_1 s_1'(n)s_2'(n) \tag{25}$$

$$x_2(n) = -L_{21}[s_1'(n)]^2 + [s_2'(n)]^2 - Q_2 s_1'(n)s_2'(n). \tag{26}$$

This mixture model is therefore also a specific case of the class of models defined in (15), where each target term $T_i[s(n)]$ is equal to $[s_i'(n)]^2$, and each interfering term $I_i[s(n)]$ consists of all other components of $x_i(n)$ in (25)-(26). Both functions T_i with $i \in \{1,2\}$ are then equal to the squaring function. They are therefore invertible, provided the source signals are constrained to have a constant sign.

Many other specific mixing models belonging to the class defined by (15) may also be considered by the reader, depending on his/her needs. To handle them and all ATM, we now propose general separating structures suited to the global model (15), and we then detail their specific versions corresponding to the above two applications of this mixing model.

2.3 Proposed networks for additive-target mixtures

2.3.1 General structures

We here extend the principles that we introduced in Section 2.1 to the class of mixtures that we defined in Section 2.2. We first consider the basic version (16) of these mixtures and we propose to process it with a recurrent structure defined as follows. The behavior of this network is governed by a set of operators C_i. These operators may be static or dynamic, depending on the nature of the mixing model itself. For each time n, this structure receives the observation vector $x(n)$ and computes the values of its output vector y by performing a recurrence. We again denote as m the index associated to this recurrence and $y(m)$ the successive values of the output vector in this recurrence performed at time n [7]. This recurrence reads

$$y_i(m+1) = x_i(n) + C_i[y(m)] \;\; \forall\, i \in \{1, \ldots, P\}. \tag{27}$$

The structure of this recurrent network is shown in Fig. 3 for the simplest configuration, i.e. $P = N = 2$.

Now consider the case when the network operators C_i are "fitted" to the interfering terms of the mixing model, i.e.

$$C_i[s'(n)] = I_i[s(n)] \qquad \forall\, i \in \{1, \ldots, P\}. \tag{28}$$

Assume we start step m of the recurrence (27) with

$$y_i(m) = s_i'(n) \qquad \forall\, i \in \{1, \ldots, P\}. \tag{29}$$

Then, using the same approach as in Section 2.1, it may be shown easily that, at the end of this step of the recurrence, we still have

$$y_i(m+1) = s_i'(n) \qquad \forall\, i \in \{1, \ldots, P\}. \tag{30}$$

[7] Again, these successive output values therefore also depend on n, but this index n is omitted in the notations.

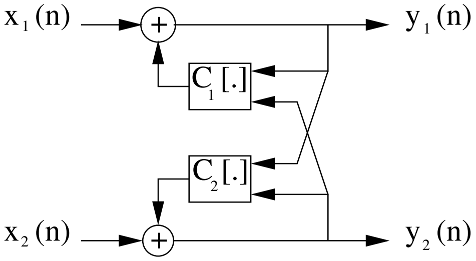

Figure 3: Proposed recurrent network for basic additive-target mixtures.

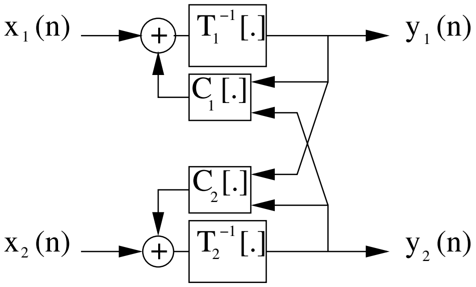

Figure 4: Proposed recurrent network for general additive-target mixtures, with adder outputs inverted before feedback.

In other words, the above-defined choice for the operators C_i is such that: (i) they exactly cancel the interfering terms of the mixing model, so that these C_i may be called "cancellation operators", and (ii) the output state (29) is a fixed point of the recurrence (27) associated to the considered network.

Let us now extend this approach to the general mixing model (15). The proposed method then essentially makes it possible to extract the target terms $T_i[s_i(n)]$ from the observations, as "intermediate outputs". To then retrieve the sources $s_i(n)$ themselves[8] from the extracted target terms, one just has to transfer each above intermediate output through the corresponding inverse mapping T_i^{-1}. This yields two alternative structures for our recurrent separating systems, depending whether the signals fed back to the inputs are the "final outputs" corresponding to $s_i(n)$ or the intermediate outputs associated to $T_i[s_i(n)]$. These two separating structures are respectively shown in Fig. 4 and 5 for two sources. One derives in the same way as above the recurrence equations corresponding to these structures (for any number of sources) and shows that the values of the operators C_i for which the extracted sources define a fixed point of these recurrences read

$$C_i[.] = I_i[.] \qquad \forall\, i \in \{1, \ldots, P\} \tag{31}$$

for the structure shown in Fig. 4 and

$$C_i[T[.]] = I_i[.] \qquad \forall\, i \in \{1, \ldots, P\} \tag{32}$$

for the network of Fig. 5, where the notation $T[.]$ in (32) represents a vector obtained by transferring each of its entries with index i through the mapping T_i.

All the recurrent networks that we thus proposed therefore have the following main features:

- *All these structures do possess a fixed point for which they extract the source signals.*

- *These structures avoid the inversion issue of feedforward networks: the operators C_i that they use are essentially "equal" to the* direct *mixing model, i.e. they are directly related to its interfering part I_i, not to its inverse, as shown by (28), (31) and (32). These structures*

[8] The following discussion may be straightforwardly modified by considering $s'_i(n)$ instead of $s_i(n)$.

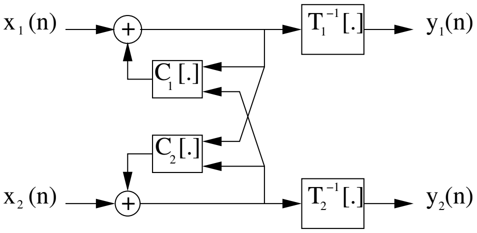

Figure 5: Proposed recurrent network for general additive-target mixtures, with adder outputs not inverted before feedback.

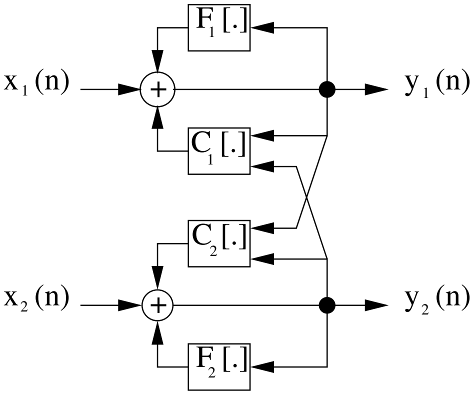

Figure 6: Proposed recurrent network for basic additive-target mixtures, with additional self-feedback.

> *therefore do not require us to derive an analytic model of the* inverse *of the considered mixing model.*

As noted in Section 1, extended structures for linear mixtures have been proposed in the literature by adding feedback loops from each output to each input having the *same* index i. We may also consider such extensions for our approach, thus introducing self-feedback operators F_i in each way i of our above structures. For instance, the structure in Fig. 3 is thus transformed into the network of Fig. 6. This extended approach provides additional flexibility in the design of the proposed structures. This flexibility is indeed required, as shown further in this paper when applying these structures to the specific mixture model that we introduced in Section 2.2.2.

2.3.2 First application: linear-quadratic mixtures

We here consider observations defined by (20)-(21). As explained in Section 2.2.2, each target term $T_i[s(n)]$ is then restricted to $s_i'(n)$. The basic approach that we propose in this case is therefore defined by the recurrence equation (27) and the structure of Fig. 3. Applying these general results to the mixing model (20)-(21), this recurrence becomes

$$y_1(m+1) = x_1(n) + l_{12}y_2(m) + q_1y_1(m)y_2(m) \tag{33}$$

$$y_2(m+1) = x_2(n) + l_{21}y_1(m) + q_2y_1(m)y_2(m) \tag{34}$$

where l_{ij} and q_i are the weights of the basic neural network that we propose to process this type of mixtures. The structure of this network is shown in Fig. 7.

An extended form of this network, corresponding to the general structure of Fig. 6, may then be derived by introducing linear self-feedback in each way i of the above structure. Denoting l_{ii} the self-feedback weights, the recurrence equations then become:

$$y_1(m+1) \quad = \quad x_1(n) + l_{11}y_1(m) + l_{12}y_2(m)$$

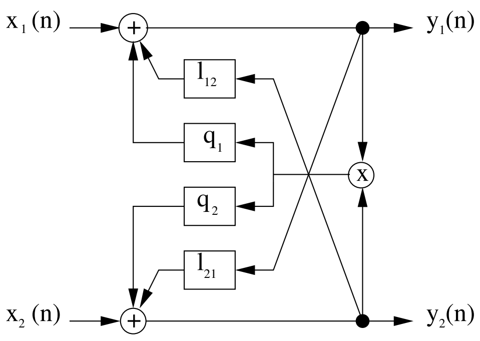

Figure 7: Proposed recurrent network for linear-quadratic mixtures: basic version, i.e. without self-feedback.

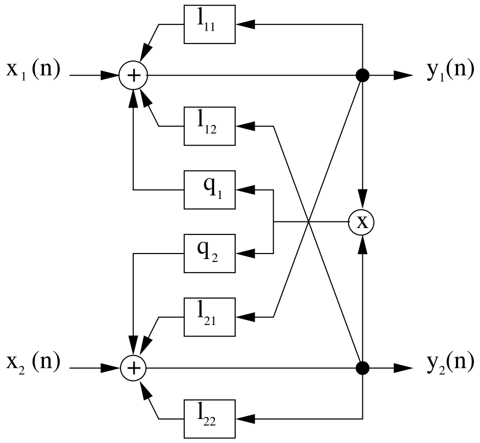

Figure 8: Proposed recurrent network for linear-quadratic mixtures: extended version, i.e. with self-feedback.

$$\begin{aligned} & +q_1 y_1(m) y_2(m) && (35) \\ y_2(m+1) &= x_2(n) + l_{21} y_1(m) + l_{22} y_2(m) \\ & +q_2 y_1(m) y_2(m). && (36) \end{aligned}$$

The corresponding extended structure is shown in Fig. 8.

2.3.3 Second application: quadratic mixtures

Now consider observations defined by (25)-(26). Since we showed in Section 2.2.3 that the target terms $T_i[s(n)]$ are here equal to $[s_i'(n)]^2$, we may now apply any of the structures of Fig. 4 and 5. For the mixing model considered here, a direct application of the approach of Fig. 4 to positive source signals yields the structure of Fig. 9. The other general structures that we proposed may be applied to this specific mixing model in the same way.

2.4 Extension to extractable-target mixtures

The above approach may be further extended by introducing a much wider class of mixing models, that we call "extractable-target mixtures" (ETM) due to their definition that we now provide. These models also involve target and interfering terms $T_i[s(n)]$ and $I_i[s(n)]$, but which are now combined according to a quite general function $f(u,v)$ to form the observations, i.e.

$$f(T_i[s_i(n)], I_i[s(n)]) = x_i(n) \qquad \forall\, i \in \{1, \ldots, P\}. \tag{37}$$

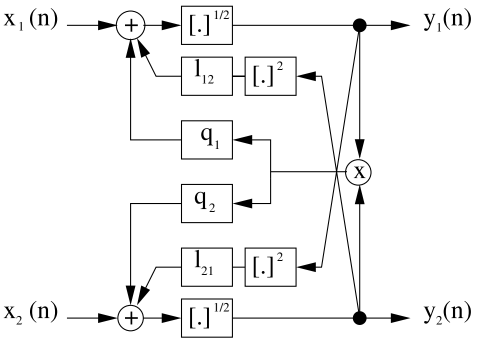

Figure 9: Proposed recurrent network for quadratic mixtures, with adder outputs inverted before feedback, without self-feedback.

The only constraint that we set on f is that there should exist an associated function[9] $g(w,z)$, whose analytic expression may be derived from that of f and which is such that

$$g(f(u,v),v) = u. \tag{38}$$

The motivation for setting this constraint is that, when fed with $w = x_i(n)$ and $z = I_i[s(n)]$, the operator $g(w,z)$ provides $T_i[s_i(n)]$, as shown by (37) and (38). The target terms $T_i[s_i(n)]$ thus become extractable when combining the function f corresponding to the undesired mixing phenomenon and the function g that we introduce in our separating networks to cancel the effect of f. More precisely, we build these networks by starting from those dedicated to ATM and shown in Fig. 4 and 5, and by just replacing their two-input adders by the two-input function g (the approach may then be extended to more than two sources). By applying the ideas developed in the previous sections to this new mixing and separating configuration it may be checked easily that, for adequate operators C_i again directly related to the interference operators I_i, these new networks have a fixed point which corresponds to the extraction of the source signals.

While the above description aimed at being general and was therefore rather formal, let us now illustrate it briefly with two practical examples. As suggested by the above discussion itself, the ATM that we introduced in Section 2.2.1 are a specific case of ETM. They correspond to

$$\begin{aligned} f(u,v) &= u - v \qquad (39)\\ g(w,z) &= w + z, \qquad (40)\end{aligned}$$

as may be checked by the reader.

Similarly, we can define "multiplicative-target mixtures" (MTM) by

$$\begin{aligned} f(u,v) &= u/v \qquad (41)\\ g(w,z) &= w \times z. \qquad (42)\end{aligned}$$

Concretely, this means that we consider mixtures defined as

$$x_i(n) = T_i[s_i(n)]/I_i[s(n)] \qquad \forall\ i \in \{1,\ldots,P\}, \tag{43}$$

as shown by (37) and (41). The networks that we propose for separating such mixtures are then derived from those shown in Fig. 4 and 5, by just replacing their adders by multipliers[10].

[9] One could consider different classes of functions for f and g in each way i of the mixing and separating stages. This case is however also contained in the description provided here, by considering a single but general couple of functions f and g, which each define a class of functions which includes all sub-classes considered in each way i.

[10] Note that various specific ETM may alternatively be transformed into ATM (of *transformed* signals) by preprocessing the observations. For instance, one may take their logarithms for MTM.

2.5 Achievements and remaining tasks

At this stage, we have mainly obtained the following results. We have defined a general procedure which allows us to create several recurrent separating networks dedicated to any given mixing model belonging to the ETM family, especially including ATM. We have also described this procedure in more detail for simple nonlinear mixtures, i.e. especially linear-quadratic mixtures, which result in the basic and extended networks respectively shown in Fig. 7 and 8.

Before using any such network, one would ideally like to know its properties, especially what are the fixed points of this dynamic structure and which of them are stable. This is of special importance here because, since we are considering nonlinear mixtures, the separating recurrent networks that we created are nonlinear dynamic systems, and it is well-known that such systems may yield unstable or even chaotic behavior [17]. Deriving all fixed points is only feasible for possibly nonlinear but still relatively simple mixtures, since it is directly related to the derivation of an analytic expression of the inverse of the mixing model[11]. At least, the following result about fixed points holds for any ETM model: we showed above that our structures have a fixed point by construction, which is the point corresponding to the extraction of all sources (also called the separating point), i.e. the main point of interest. One may then analyze the stability of this point.

To illustrate these principles, we will focus again on linear-quadratic mixtures in the next section. We will derive all the fixed points of the proposed basic and extended networks, and we will analyze their stability. As the extended structure of Fig. 8 is a superset of the basic network of Fig. 7, we will analyze the former one and thus obtain results for the latter as a specific case, i.e. by setting

$$l_{11} = 0 \quad \text{and} \quad l_{22} = 0. \tag{44}$$

It may be mentioned at this stage that, before proposing the extended structure of Fig. 8 (and considering new mixing models) in this paper, we described in the short communications [18]-[20] preliminary investigations where we only addressed linear-quadratic mixtures and we only introduced the basic structure of Fig. 7. Despite its attractive features, the experimental tests that we performed with this specific structure showed that it may become unstable, thus confirming our above warning about the general behavior of nonlinear dynamic systems. These limitations of the basic network are a strong motivation for here introducing its extended version of Fig. 8: this new structure has extra parameters, i.e. the linear feedback weights l_{11} and l_{22}, that we will aim at selecting in order to guarantee the stability of this structure. On the contrary, we will demonstrate that the constraint (44) corresponding to our initial network does not allow it to handle all signal values.

3 Analysis of linear-quadratic networks

3.1 Fixed and separating points

We here analyze the fixed points (i.e. equilibrium points) of recurrence (35)-(36), for observations expressed as (20)-(21). They are defined as the points (y_1^E, y_2^E) which are such that

$$y_1(m+1) = y_1(m) = y_1^E \text{ and } y_2(m+1) = y_2(m) = y_2^E. \tag{45}$$

We again consider the non-blind configuration, i.e. the case when the mixing coefficients L_{ij} and Q_i of (20)-(21) are known. Let us stress that this non-blindness constraint is not highly restrictive: it not only includes the specific case when the mixing coefficients L_{ij} and Q_i are known a priori (e.g. thanks to physical knowledge about the considered system), but also the more general situation when they are derived once for all from a set of *known* source values during a first step of the approach. In the latter case, the proposed procedure consists in using these known source values and measuring the corresponding observed signals. This yields a set of linear equations with unknowns L_{ij} and Q_i, as shown by (20)-(21). Numerically solving these equations yields the

[11] This will appear clearly below for linear-quadratic mixtures, by comparing the resolution of the fixed-point equations (56)-(57) to the inversion of the mixing equations (17)-(18).

mixing parameter values L_{ij} and Q_i. In the second step of our approach, these fixed parameter values are then used for *unknown* source signal values in the separating networks considered in the remainder of this paper[12]. Note that the same type of procedure may also be defined for other mixing models, such as the quadratic mixtures introduced in Section 2.2.3.

So, considering a set of known mixing coefficients, we aim at selecting network weights which meet the condition: "one of the associated fixed points corresponds to source separation without permutation and with possible scale factors", because we aim our network at converging to such a point. We therefore first determine all network weights which meet the above condition. This condition corresponds to the separating point (more precisely, the class of points) defined by

$$y_1^E = k_1 s_1'(n) \quad \text{and} \quad y_2^E = k_2 s_2'(n) \tag{46}$$

where k_1 and k_2 are two arbitrary scale factors. Combining (20)-(21), (35)-(36) and (45)-(46) yields

$$\begin{aligned}[(l_{11}-1)k_1+1]s_1'(n)+[l_{12}k_2-L_{12}]s_2'(n) & & \qquad (47)\\ +[q_1k_1k_2-Q_1]s_1'(n)s_2'(n) &= 0 \\ [l_{21}k_1-L_{21}]\,s_1'(n)+[(l_{22}-1)k_2+1]s_2'(n) & & \qquad (48)\\ +[q_2k_1k_2-Q_2]s_1'(n)s_2'(n) &= 0.\end{aligned}$$

For given mixing parameters in (20)-(21), the network weights l_{ij} and q_i and scale factors k_i should be selected so that (47)-(48) are met whatever the normalized source values $s_i'(n)$. All source coefficients in these equations should therefore be zero, which yields

$$l_{11} = -\frac{1}{k_1}+1 \tag{49}$$

$$l_{12} = \frac{L_{12}}{k_2} = L_{12}l_{22}' \tag{50}$$

$$q_1 = \frac{Q_1}{k_1k_2} = Q_1l_{11}'l_{22}' \tag{51}$$

$$l_{21} = \frac{L_{21}}{k_1} = L_{21}l_{11}' \tag{52}$$

$$l_{22} = -\frac{1}{k_2}+1 \tag{53}$$

$$q_2 = \frac{Q_2}{k_1k_2} = Q_2l_{11}'l_{22}' \tag{54}$$

with

$$l_{11}' = 1-l_{11} \quad \text{and} \quad l_{22}' = 1-l_{22}. \tag{55}$$

The first expressions in (50)-(52) and (54) show that we thus obtain an infinite number of solutions, due to the two arbitrary scale factors k_i with which the sources appear in the network outputs. There is a one-to-one correspondence between these factors and the two network weights l_{11} and l_{22} as shown by (49) and (53). One may therefore consider the weights l_{11} and l_{22} as the primary parameters, select them (freely at this stage), and then assign accordingly the other network weights, using the second expressions in (50)-(52) and (54).

For these weight values, we know by construction that the network has at least one fixed point, i.e. the point defined by (46) with (49) and (53). We must then determine *all* fixed points for these weight values, because the network may converge to any of these points (depending on their

[12] There should be no confusion between this approach and alternating estimation techniques, applied to mixture parameters and sources, that one might aim at developing for the blind configuration: in the approach proposed here, we first consider once for all known source values, from which we derive the exact mixing parameter values (i.e. no statistical parameter estimation is required here). Then, the network is always operated with the unknown source values that we aim at restoring, in a non-blind mode (from the mixture point of view), i.e. by using the exact mixing parameter values that we first derived in our first step (therefore without estimating nor adapting the mixing parameter values in that second step).

stability) and we should especially determine whether each of them yields separated sources. To this end, we again consider the equations which define all fixed points, i.e. (35)-(36) combined with (45), which yields

$$\begin{aligned} x_1(n) - l'_{11}y_1^E + l_{12}y_2^E + q_1 y_1^E y_2^E &= 0 && (56) \\ x_2(n) + l_{21}y_1^E - l'_{22}y_2^E + q_2 y_1^E y_2^E &= 0. && (57) \end{aligned}$$

These equations may e.g. be solved by first deriving a linear combination of them which cancels their cross-terms $y_1^E y_2^E$, such as $q_2 \times (56) - q_1 \times (57)$. This yields a linear relationship between y_1^E and y_2^E, from which one then derives the expression of y_2^E vs y_1^E, i.e.

$$y_2^E = \frac{(q_2 l'_{11} + q_1 l_{21}) y_1^E - [q_2 x_1(n) - q_1 x_2(n)]}{q_2 l_{12} + q_1 l'_{22}}. \qquad (58)$$

Inserting the latter expression of y_2^E in (56), one gets a second-order polynomial equation for y_1^E, which is then straightforwardly solved, thus yielding

$$y_1^E = \frac{-[q_2 x_1(n) - q_1 x_2(n) + l'_{11} l'_{22} - l_{12} l_{21}] + \epsilon_{y_1} \Delta_{y_1}^{1/2}}{-2[q_2 l'_{11} + q_1 l_{21}]} \qquad (59)$$

where

$$\epsilon_{y_1} = \pm 1 \qquad (60)$$

defines which solution of the second-order polynomial equation in y_1^E is considered, and Δ_{y_1} is the discriminant of this equation. The corresponding expressions of y_2^E may then by derived from its above-mentioned expression (58) vs y_1^E.

Subsequent expressions of Δ_{y_1}, y_1^E and y_2^E are then obtained by inserting in the above ones: (i) the expressions (50)-(52) and (54)-(55) of the considered network weights and (ii) the expressions (20)-(21) of the observations with respect to the source signals. Eq. (50)-(52) and (54)-(55) thus first yield

$$\Delta_{y_1} = (l'_{11} l'_{22})^2 \delta_{y_1} \qquad (61)$$

with

$$\begin{aligned} \delta_{y_1} &= [Q_2 x_1(n) - Q_1 x_2(n) + \gamma]^2 \\ &\quad -4\alpha[x_1(n) + x_2(n) L_{12}] && (62) \end{aligned}$$

where

$$\begin{aligned} \alpha &= Q_2 + Q_1 L_{21} && (63) \\ \gamma &= 1 - L_{12} L_{21}. && (64) \end{aligned}$$

Inserting (20)-(21) in the above equations, (62) becomes

$$\delta_{y_1} = (-\alpha s'_1(n) + \beta s'_2(n) + \gamma)^2 \qquad (65)$$

where

$$\beta = -(Q_2 L_{12} + Q_1). \qquad (66)$$

It should be noted that (65), combined with (61), guarantees that Δ_{y_1} is positive in the considered conditions[13]. The above equations therefore do have real solutions y_1^E and hence y_2^E, i.e. the network does have fixed points. As mentioned above, we already knew by construction that it has at least one fixed point, i.e. the separating point. The result obtained here is more accurate, i.e. it has exactly two (possibly equal) fixed points, defined by the above equations.

[13] On the contrary, this is not guaranteed for arbitrary network weights.

The selection of one of the two solutions y_1^E of (59) is based on the sign of ϵ_{y_1} (see (60)) and on the signs of the quantities involved in $\Delta_{y_1}^{1/2}$, i.e. $l'_{11}l'_{22}$ and $(-\alpha s'_1(n)+\beta s'_2(n)+\gamma)$, due to (61) and (65). It is therefore convenient to introduce the "total sign variable"

$$\epsilon_{T1}=\epsilon_{y_1}\ sgn(l'_{11}l'_{22})\ sgn(-\alpha s'_1(n)+\beta s'_2(n)+\gamma). \tag{67}$$

The two fixed points may then be expressed as

$$\begin{aligned} y_1^E &= \frac{1}{2l'_{11}\alpha}\{[\alpha s'_1(n)+\beta s'_2(n)+\gamma] \\ &\quad -\epsilon_{T1}[-\alpha s'_1(n)+\beta s'_2(n)+\gamma]\} \qquad (68) \\ y_2^E &= \frac{1}{\beta l'_{22}}\{-\alpha l'_{11}y_1^E+[\alpha s'_1(n)+\beta s'_2(n)]\}. \qquad (69) \end{aligned}$$

Their explicit expressions, depending on ϵ_{T1}, are as follows. The first solution, corresponding to $\epsilon_{T1}=1$, reads

$$y_1^E=\frac{1}{l'_{11}}s'_1(n)\ \text{ and }\ y_2^E=\frac{1}{l'_{22}}s'_2(n). \tag{70}$$

This solution yields source separation without permutation (and with scale factors). It is nothing but the above solution (46), as shown by (49), (53) and (55). The second solution, corresponding to $\epsilon_{T1}=-1$, reads

$$y_1^E=\frac{1}{l'_{11}}\left[\frac{\beta}{\alpha}s'_2(n)+\frac{\gamma}{\alpha}\right],\ \ y_2^E=\frac{1}{l'_{22}}\left[\frac{\alpha}{\beta}s'_1(n)-\frac{\gamma}{\beta}\right]. \tag{71}$$

This additional solution yields source separation with a permutation (and with scale factors and additive constants). Therefore, for these weight values, this network does not yield spurious fixed points, i.e. fixed points such that the outputs are still mixtures of the sources. We now analyze the stability of the above two fixed points.

3.2 Stability condition

The considered recurrent network is a two-dimensional nonlinear dynamic system, since the evolution of its state vector $[y_1(m),y_2(m)]^T$ is defined by the nonlinear equations (35)-(36). The linear stability of any such system at a given fixed point may be analyzed by considering a first-order approximation of its evolution equations at that point[14]. The new value of a small disturbance added to the state vector is thus expressed as the product of a matrix $\mathbf{H}$, which defines the first-order approximation of the system, by the former value of that disturbance (see e.g. details in [17]). The (asymptotic) stability of the system at the considered point is guaranteed by constraining the moduli of both eigenvalues of the corresponding matrix $\mathbf{H}$ to be lower than 1. Deriving these eigenvalues and applying the above constraint set on them, it may be shown that this stability condition reads

$$\left\{\begin{array}{l} T>-D-1 \\ T<D+1 \\ D<1 \end{array}\right. \tag{72}$$

where D and T are respectively the determinant and trace of $\mathbf{H}$. The stability region thus obtained in the (D,T) plane is bounded by a triangle which includes the origin (see the figure in [17], p. 153, which also indirectly confirms (72)).

The above general procedure is here applied by computing a first-order approximation of the recurrence (35)-(36) at any fixed point. We thus get

$$\mathbf{H}=\left(\begin{array}{cc} q_1y_2^E+l_{11} & q_1y_1^E+l_{12} \\ q_2y_2^E+l_{21} & q_2y_1^E+l_{22} \end{array}\right). \tag{73}$$

[14] That approximation therefore only allows us to analyze the *local* convergence of the system at the considered point, as opposed to its global convergence from any initial state.

This directly yields

$$T = q_1 y_2^E + l_{11} + q_2 y_1^E + l_{22}. \tag{74}$$

Additional manipulations using (58)-(59) then show that

$$D = T - 1 + \epsilon_{y_1} \sqrt{\Delta_{y_1}}. \tag{75}$$

Condition (72) then becomes

$$\begin{aligned} 2T &> -\epsilon_{y_1} \sqrt{\Delta_{y_1}} && (76) \\ 0 &< \epsilon_{y_1} \sqrt{\Delta_{y_1}} && (77) \\ T + \epsilon_{y_1} \sqrt{\Delta_{y_1}} &< 2. && (78) \end{aligned}$$

Eq. (77) directly shows that the fixed point corresponding to $\epsilon_{y_1} = -1$ is then always unstable. We therefore only investigate the fixed point corresponding to $\epsilon_{y_1} = 1$ hereafter, and we assume Δ_{y_1} is *strictly* positive, so that (77) is met. The overall condition (76)-(78) then becomes

$$\begin{aligned} 2T + \sqrt{\Delta_{y_1}} &> 0 && (79) \\ T + \sqrt{\Delta_{y_1}} &< 2. && (80) \end{aligned}$$

To use this condition, we then express T and $\sqrt{\Delta_{y_1}}$ with respect to known quantities, i.e. the network weights or corresponding mixing coefficients, and the observations $x_1(n)$ and $x_2(n)$ (as opposed to the source signals, i.e. we do not use (20)-(21) here). Using (61), we derive

$$\sqrt{\Delta_{y_1}} = |l'_{11} l'_{22}| \sqrt{\delta_{y_1}} \tag{81}$$

with δ_{y_1} defined in (62). T is then obtained by combining (74) and (58)-(59), where we set $\epsilon_{y_1} = 1$ and we use (50)-(52), (54) and (81). This yields

$$T = -A l'_{11} - B l'_{22} + 2 \tag{82}$$

with

$$\begin{aligned} A &= \frac{Q_1}{2\beta} \Big[-Q_2 x_1(n) + Q_1 x_2(n) + \gamma \\ &\quad - sgn(l'_{11} l'_{22}) \sqrt{\delta_{y_1}} \Big] + 1 && (83) \\ B &= \frac{Q_2}{2\alpha} \Big[-Q_2 x_1(n) + Q_1 x_2(n) - \gamma \\ &\quad + sgn(l'_{11} l'_{22}) \sqrt{\delta_{y_1}} \Big] + 1. && (84) \end{aligned}$$

Inserting (81) and (82) in the stability condition (79)-(80), the latter condition becomes

$$\begin{aligned} |l'_{11} l'_{22}| \sqrt{\delta_{y_1}} - 2A l'_{11} - 2B l'_{22} + 4 &> 0 && (85) \\ |l'_{11} l'_{22}| \sqrt{\delta_{y_1}} - A l'_{11} - B l'_{22} &< 0. && (86) \end{aligned}$$

It should be noted that A and B depend on l'_{11} and l'_{22} only because of their factor $sgn(l'_{11} l'_{22})$. Moreover, δ_{y_1} does not depend on l'_{11} and l'_{22}. When l'_{11} and l'_{22} are varied, but in such a way that the sign of $l'_{11} l'_{22}$ remains constant, we have the following two phenomena, for given mixing parameters and observations: (i) A, B and δ_{y_1} remain constant and (ii) the part of the (l'_{11}, l'_{22}) plane for which condition (85)-(86) is met may be defined as follows: it is the intersection of two sub-parts of the (l'_{11}, l'_{22}) plane, each of these parts being bounded by a curve defined as:

$$2c_1 l'_{11} l'_{22} + 2c_2 l'_{11} + 2c_3 l'_{22} + c_4 = 0, \tag{87}$$

where all c_i are constants which depend on the mixing parameters and observations, and which are specific to each of the two sub-conditions (85) and (86). Each corresponding condition (87) defines an hyperbola [21] in the (l'_{11}, l'_{22}) plane.

3.3 Limitations of our basic network

The stability condition (85)-(86), is applicable to any values of l'_{11} and l'_{22}. It therefore especially applies to our basic network, which corresponds to (44) and hence to

$$l'_{11} = 1 \quad \text{and} \quad l'_{22} = 1. \tag{88}$$

Inserting these values in (85)-(86) shows that this basic network yields a stable fixed point only for some values of the mixing coefficients and observed signals, and therefore of the source signals. To clearly illustrate this phenomenon with an example, we consider the simple case

$$L_{12} = 0, \quad L_{21} = 0, \quad Q_1 = Q_2 = Q > 0. \tag{89}$$

Applying this condition and (88) to (20)-(21), (63)-(66) and (83)-(84) allows us to rewrite stability condition (85)-(86). It may thus be easily shown that our basic network then only yields stability for a limited domain of source values, which consists of a strip in the $(s'_1(n), s'_2(n))$ plane, defined by

$$-s'_1(n) - \frac{1}{Q} < s'_2(n) < -s'_1(n) + \frac{3}{Q}. \tag{90}$$

3.4 A method for stabilizing our extended network

3.4.1 Analysis of stability condition

Condition (85)-(86) is of high theoretical interest, because it completely defines the stability of the considered fixed point. However, it does not show easily if and how l'_{11} and l'_{22} may be selected in order to ensure stability at the considered fixed point for any given observed signal values. To address that topic, we introduce a method which consists in considering l'_{11} as the primary variable and l'_{22} as the secondary variable, and in expressing it as $l'_{22} = \lambda l'_{11}$, where λ is a parameter. For any fixed λ, we first investigate whether there exist values of l'_{11} such that (85)-(86) is met, in order to eventually determine if there exist values of l'_{11} and λ (and therefore l'_{22}) such that (85)-(86) is met. In other words, we first determine the intersection of the part of the (l'_{11}, l'_{22}) plane where (85)-(86) is met and of a given line in that plane, defined by $l'_{22} = \lambda l'_{11}$ (with $\lambda \neq 0$). Using the latter expression of l'_{22}, Eq. (85)-(86) become

$$|\lambda|\sqrt{\delta_{y1}}(l'_{11})^2 - 2(A + B\lambda)l'_{11} + 4 \quad > \quad 0 \tag{91}$$
$$|\lambda|\sqrt{\delta_{y1}}(l'_{11})^2 - (A + B\lambda)l'_{11} \quad < \quad 0. \tag{92}$$

For a given λ, (62) and (83)-(84) show that A and B do not depend on l'_{11} and l'_{22}. Eq. (91)-(92) then yield two inequalities with respect to l'_{11}, which are solved as follows. We first only derive the solution of (92). It is thus shown easily that, for (92), any (non-zero) λ is suitable and the solutions of (92) for a given λ are

$$l'_{11} = \mu \frac{A + B\lambda}{|\lambda|\sqrt{\delta_{y1}}} \quad \text{with} \quad 0 < \mu < \mu_{max} \tag{93}$$

where $\mu_{max} = 1$. To solve (91)-(92) we then determine, for a given λ, which of the above solutions of (92) are also solutions of (91). We therefore insert the solutions l'_{11} defined in (93) into (91), which yields

$$(\mu^2 - 2\mu)\frac{(A + B\lambda)^2}{|\lambda|\sqrt{\delta_{y1}}} + 4 > 0 \tag{94}$$

where we only have to consider $0 < \mu < 1$, due to the above condition on μ in (93). Eq. (94) then defines all the allowed values of μ. By studying the variations of the left-hand term of (94) with respect to μ (for a given λ), it may then be shown that the overall solution of (91)-(92) is still defined by (93), but now with μ_{max} defined as follows. Denoting

$$C(\lambda) = \frac{(A + B\lambda)^2}{|\lambda|\sqrt{\delta_{y1}}} \tag{95}$$

we have

$$\mu_{max} = \begin{cases} 1 & \text{if} \qquad C(\lambda) \leq 4 \\ 1 - \sqrt{1 - \frac{4}{C(\lambda)}} & \text{otherwise.} \end{cases} \tag{96}$$

The above analysis shows that any value of λ yields a non-empty interval of solutions for l'_{11}. For a given λ, a simple and safe approach therefore consists in selecting for l'_{11} the value situated in the middle of the allowed range (93), i.e.

$$l'_{11} = \frac{\mu_{max}}{2} \frac{A + B\lambda}{|\lambda|\sqrt{\delta_{y_1}}} \tag{97}$$

with μ_{max} defined by (96). We should eventually propose a method for selecting λ. As stated above, any (non-zero) value of λ is acceptable. A simple solution is $\lambda = \pm 1$, which gives the same "weight" to l'_{11} and $l'_{22} = \lambda l'_{11}$. Moreover, the sign of λ may be chosen as follows. As explained above, the considered fixed point corresponds to $\epsilon_{y_1} = 1$. Since $l'_{22} = \lambda l'_{11}$ in addition, (67) here reduces to

$$\epsilon_{T1} = sgn(\lambda)\ sgn(-\alpha s'_1(n) + \beta s'_2(n) + \gamma). \tag{98}$$

Therefore, if λ has the same sign as $(-\alpha s'_1(n) + \beta s'_2(n) + \gamma)$, then $\epsilon_{T1} = 1$, so that the considered stable fixed point yields the non-permuted sources of (70). Otherwise, the permuted sources of (71) are obtained. We thus guarantee that both solutions are obtained by successively applying our approach with two opposite values of λ. Moreover, if the source and mixing coefficients are such that the sign of $(-\alpha s'_1(n) + \beta s'_2(n) + \gamma)$ is known, then selecting λ also with that sign guarantees (local) convergence to the non-permuting point[15].

3.4.2 Summary of proposed method

Based on the above analysis, the following procedure is guaranteed to yield local convergence towards a separating point:

1. Select λ as explained above.
2. Set l'_{11} according to (97), taking into account $sgn(l'_{11}l'_{22}) = sgn(\lambda)$.
3. Set $l'_{22} = \lambda l'_{11}$.
4. Set all other network parameters according to (50)-(52) and (54)-(55).

Note that this procedure is applied independently at each time n, for different observed values. This yields different values of (l'_{11}, l'_{22}) at each time n, so that when reaching the non-permuting separating point (70), the signals $s'_i(n)$ are obtained with scale factors $1/l'_{ii}$ which vary with n. In most applications, this should be avoided, in order to obtain the sources $s_i(n)$ with *constant* scales, such as in $s'_i(n) = a_{ii}s_i(n)$. This may be achieved by post-processing the outputs $y_i(n)$ of the network of Fig. 8, obtained for each time n after the convergence of the recurrence corresponding to m. Assuming this recurrence converges to the non-permuting separating point (70), this post-processing consists in multiplying each output signal $y_i(n)$ by $l'_{ii} = 1 - l_{ii}$, so as to obtain $z_i(n) = s'_i(n)$ (up to convergence errors). This approach may be extended to the other neural network structures introduced in this paper.

4 Test results

4.1 Tests with artificial sources

We first illustrate the performance of our approach, by considering linear-quadratic mixtures, i.e. observations defined by (20)-(21), at a single time n. We use $s'_1(n) = -1$, $s'_2(n) = -2$ and mixing coefficients defined by (89), with $Q = 0.5$. We first implement our basic network by running 10

[15] Otherwise, this version of our approach may yield permutation, and this could be further investigated.

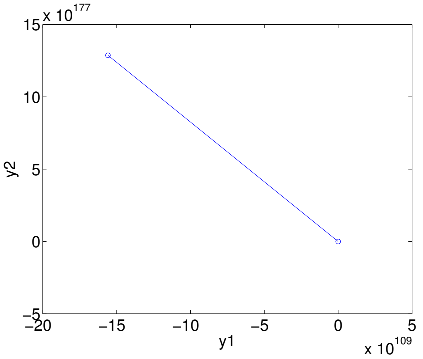

Figure 10: Divergence of basic network.

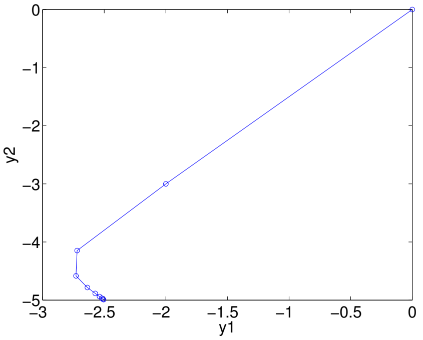

Figure 11: Convergence of extended network.

steps of the recurrence (35)-(36) with (44), (50)-(52) and (54), starting from $y_1(1) = 0$, $y_2(1) = 0$. The resulting trajectory of $(y_1(m), y_2(m))$ is provided in Fig. 10. This shows that the network diverges very rapidly. This is in agreement with the fact that condition (90) is not met here. We then implement our extended network by running the recurrence (35)-(36) as above but with parameters selected as explained in Section 3.4.2 (with $\lambda = 1$). The network then converges to the solution (70), as shown in Fig. 11 (we here have $l'_{11} = l'_{22} = 0.4$). Moreover, it converges very rapidly. This shows the effectiveness of the proposed approach.

We then determine the statistical performance of our extended network. To this end, we generate 1000 random samples of each source with a uniform distribution in $[-0.5, 0.5]$. We mix these sources in the same conditions as above (which guarantees the stability of the non-permuting separating point for $\lambda = 1$). For each source vector $s(n)$, we use the initialization $y_i(1) = x_i(n) \ \ \forall i \in \{1, 2\}$, and then run the recurrence (35)-(36) with parameters selected as above. At each step m of the recurrence performed for $s(n)$, the root-mean-square (RMS) error of the network may be defined as

$$RMS(n,m) = \sqrt{\frac{1}{2}\sum_{i=1}^{2}[y_i(m)l'_{ii} - s'_i(n)]^2}. \tag{99}$$

The average accuracy, over all source vectors $s(n)$, of our network after m steps of the recurrence is measured by the mean of $RMS(n,m)$ over n. The evolution of this mean RMS error vs m is shown in Fig. 12. This confirms that our network yields very good accuracy after just a few recurrence steps.

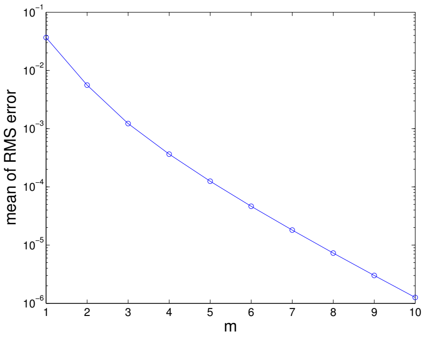

Figure 12: Mean of RMS error vs step m of recurrence.

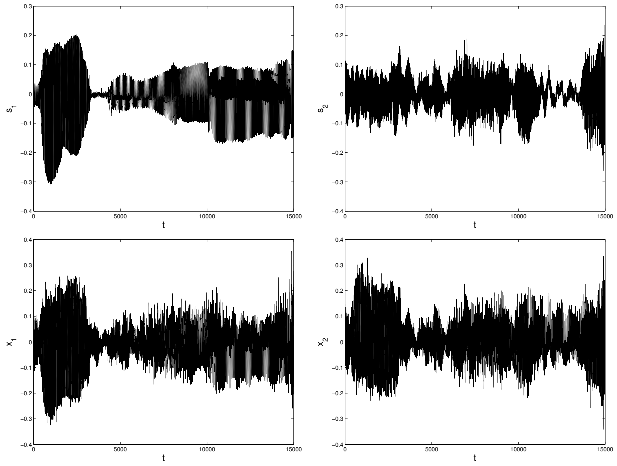

Figure 13: Speech sources $s_1(n)$ and $s_2(n)$ (top) and mixed signals $x_1(n)$ and $x_2(n)$ (bottom) used in our experiment.

4.2 Test with speech sources

Another experiment is performed with two speech signals $s_1(n)$ and $s_2(n)$, which are shown in Fig. 13. These source signals are again mixed using the linear-quadratic model (20)-(21), with $L_{12} = 0.7$, $L_{21} = 0.8$, $Q_1 = 0.8$, $Q_2 = 0.9$, to generate the two observations $x_1(n)$ and $x_2(n)$ shown in Fig. 13. The distributions of the sources and of the observations are provided in Fig. 14.

For each sample n, a recurrent extended network with parameters selected as explained previously is implemented and initialized with $y_i(1) = x_i(n) \;\; \forall i \in \{1, 2\}$. Then, the recurrence (35)-(36) is performed and the RMS error of the network is computed at each step m. The mean RMS error vs m is shown in Fig. 15 and confirms the good performance of the proposed method with real one-dimensional signals and a general choice of the mixture parameters.

4.3 Test with images

The last experiment concerns artificial mixtures of real images. The two photographic images provided by [22] and shown in Fig. 16 are vectorized, centered, and normalized to generate the two source vectors s_1 and s_2 containing $331 \times 438 = 144978$ samples. These sources are then mixed using the linear-quadratic model (20)-(21), with $L_{12} = -0.8$, $L_{21} = 0.7$, $Q_1 = -0.4$, $Q_2 = 0.3$, to generate the two observations x_1 and x_2 shown in Fig. 17. The distributions of the sources and of the observations are provided in Fig. 18.

Once more, for each sample n, a recurrent extended network initialized by the observations is

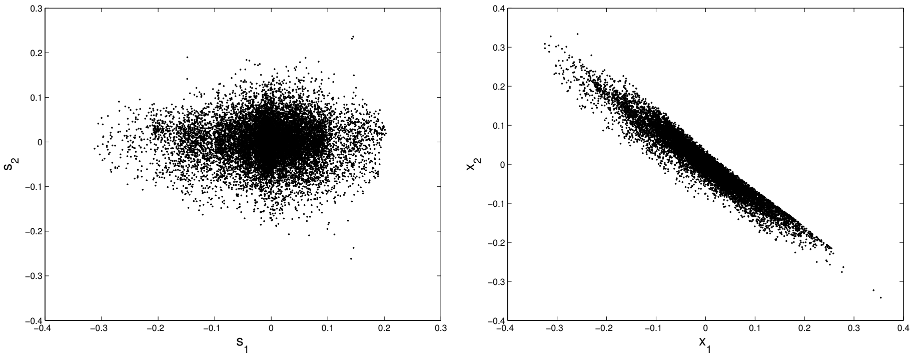

Figure 14: Distribution of the speech sources $s_1(n)$ and $s_2(n)$ (left) and of their mixtures $x_1(n)$ and $x_2(n)$ (right).

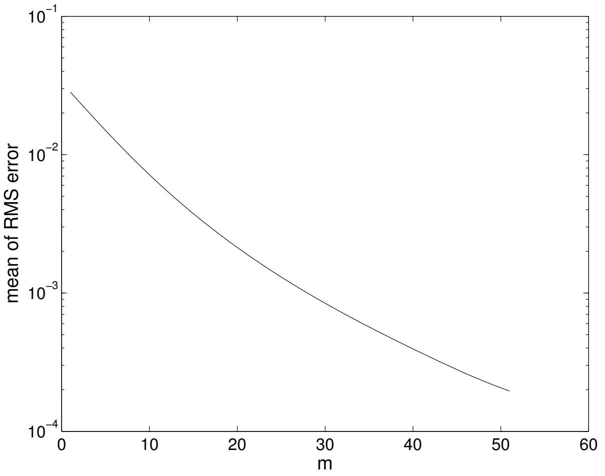

Figure 15: Mean of RMS error vs step m of recurrence for experiment using real-world speech signals.

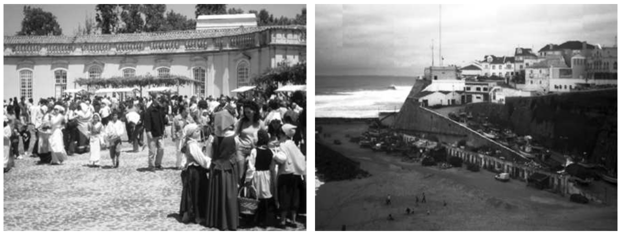

Figure 16: Photographic real-world images used in our experiment.

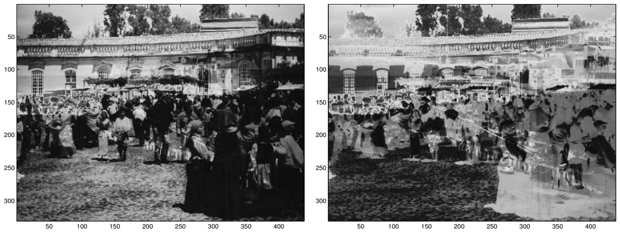

Figure 17: Mixtures of real-world images used in our experiment.

used and the RMS error of the network is computed at each step m. The mean RMS error vs m is shown in Fig. 19 and confirms the good performance of the proposed method with images and another choice of the mixture parameters.

5 Conclusion

In this first part of our paper, we introduced recurrent source separation neural network structures applicable to a large class of new nonlinear mixtures. We analyzed in detail their versions dedicated to linear-quadratic mixtures, and we especially showed how to select their free weights in order to guarantee stability. We experimentally showed that these networks yield fast and accurate convergence for various types of artificial and real sources.

Let us stress again that an attractive feature of the proposed networks is that they are tailored to each considered type of mixtures: given the mixing model, we derive a corresponding separating structure which is such that its equilibrium point of interest *exactly* restores the source values, and the network converges towards this equilibrium point quickly and with the desired

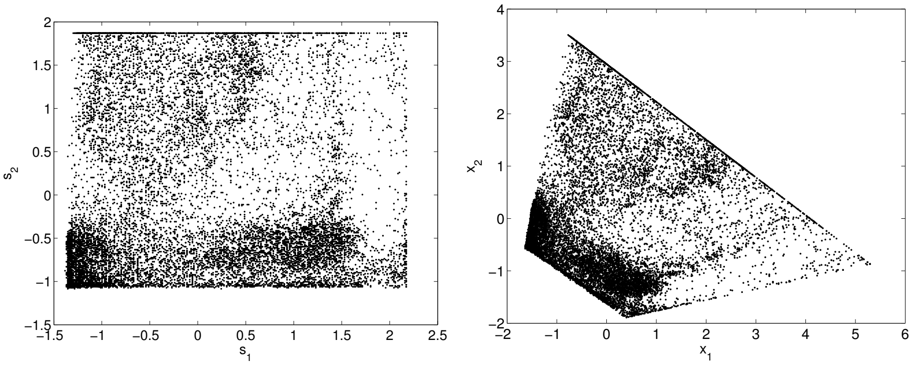

Figure 18: Distribution of the sources $s_1(n)$ and $s_2(n)$ (left) and of the observations $x_1(n)$ and $x_2(n)$ (right).

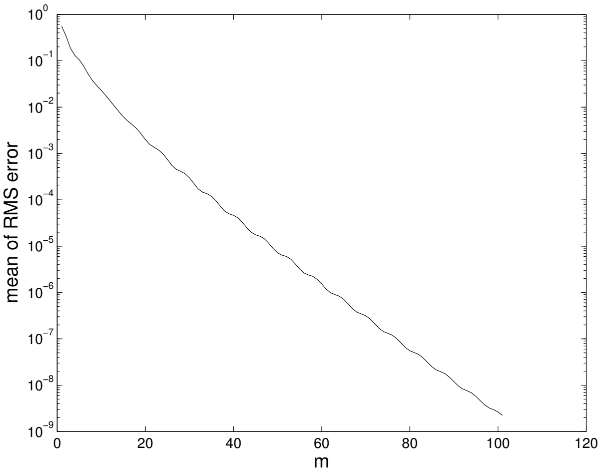

Figure 19: Mean of RMS error vs step m of recurrence for experiment using real-world images.

accuracy[16], as stated above. This should be contrasted with several existing nonlinear optimization/approximation methods, which do not yield such attractive features. For instance, one might consider using a universal approximator (e.g. a feedforward multi-layer perceptron) in order to achieve an approximation of the inverse of the mixing mapping. However, a major drawback of such a structure with given parameter values is that, for a given arbitrary input, it can only provide an approximation of the desired output with bounded accuracy, instead of the exact desired output (i.e. actual source values) that we restore. Note also that our approaches are attractive because they do not require one to manually tune parameters, thanks to the automated procedure that we introduced above.

We plan to further develop the versions of the proposed networks suited to quadratic mixtures and to general polynomial mixtures, which may be considered as Taylor series approximations of general nonlinear models. While we defined the proposed structures for an arbitrary number of sources in this paper, we then did not analyze their properties for more than two sources. An extension of this analysis to any number of sources could be considered in future work, although this is likely to be a difficult task.

More specifically, in the second part of this paper, we will mainly[17] present adaptation algorithms for all these networks, in order to perform *blind* source separation: as compared to several existing nonlinear optimization methods, the approaches that we proposed are attractive because they yield a natural extension towards the blind configuration, by keeping their structure and introducing a training algorithm for adapting their weights.

Acknowledgment

Y. Deville would like to thank Y. Naudet for drawing his attention to linear-quadratic mixtures. This first resulted in the basic network presented in [18]-[20], and this investigation then evolved towards the much more general approach introduced in this paper.

References

[1] A. Hyvarinen, J. Karhunen, E. Oja, "Independent Component Analysis", Wiley, New York, 2001.

[16] By selecting the number of recurrence steps.

[17] Other related topics will also be addressed, e.g. sample-by-sample output permutation.

[2] M. W. Berry, M. Browne, A. N. Langville, V. P. Pauca, R. J. Plemmons, "Algorithms and applications for approximate nonnegative matrix factorization", Computational statistics & data analysis, vol. 52, no. 1, pp. 155-173, 2007.

[3] C. Jutten, J. Karhunen, Advances in Blind Source Separation (BSS) and Independent Component Analysis (ICA) for Nonlinear Mixtures, International Journal of Neural Systems, vol. 14, no. 5, pp. 267-292, 2004.

[4] A. Hyvarinen, P. Pajunen, "Nonlinear independent component analysis: Existence and uniqueness results", Neural Networks, vol. 12, no. 3, pp. 429-439, 1999.

[5] A. Taleb, C. Jutten, "Source separation in post-nonlinear mixtures", IEEE Transactions on Signal Processing, vol. 47, no. 10, pp. 2807-2820, 1999.

[6] A. Taleb, "A generic framework for blind source separation in structured nonlinear models", IEEE Transactions on Signal Processing, vol. 50, no. 8, pp. 1819-1830, August 2002.

[7] C. Jutten, M. Babaie-Zadeh, S. Hosseini, "Three easy ways for separating nonlinear mixtures?", Signal Processing, 84, pp. 217-229, 2004.

[8] C. Jutten, J. Hérault, "Blind separation of sources, Part I: An adaptive algorithm based on neuromimetic architecture", Signal Processing, vol. 24, no. 1, pp. 1-10, July 1991.

[9] P. Comon, C. Jutten, J. Hérault, "Blind separation of sources, Part II: Problems statement", Signal Processing, vol. 24, no. 1, pp. 11-20, 1991.

[10] E. Sorouchyari, "Blind separation of sources, Part III: Stability analysis", Signal Processing, vol. 24, no.1, pp. 21-29, July 1991.

[11] J.-C. Fort, "Stabilité de l'algorithme de séparation de sources de Jutten et Hérault", Traitement du Signal, vol. 8, no. 1, pp. 35-42, Jan. 1991.

[12] Y. Deville, "A unified stability analysis of the Hérault-Jutten source separation neural network", Signal Processing, vol. 51, no. 3, pp. 229-233, June 1996.

[13] O. Macchi, E. Moreau, "Self-adaptive source separation, Part I: convergence analysis of a direct linear network controled by the Hérault-Jutten algorithm", IEEE Transactions on Signal Processing, vol. 45, no. 4, pp. 918-926, April 1997.

[14] E. Moreau, O. Macchi, "Self-adaptive source separation, Part II: comparison of the direct, feedback and mixed linear network", IEEE Transactions on Signal Processing, vol. 46, no. 1, pp. 39-50, Jan. 1998.

[15] A. Cichocki, W. Kasprzak, S.I. Amari, "Multi-layer neural networks with a local adaptive learning rule for blind separation of source signals", Proceedings of NOLTA '95, Las Vegas, U.S.A, Dec. 10-14, pp. 61-65.

[16] Y. Deville, "Analysis of the convergence properties of self-normalized source separation neural networks", IEEE Transactions on Signal Processing, vol. 47, no. 5, pp. 1272-1287, May 1999.

[17] J.M.T. Thompson, H.B. Stewart, "Nonlinear dynamics and chaos", Wiley, Chichester, England, 2002.

[18] Y. Deville, "Méthode de séparation de sources pour mélanges linéaires-quadratiques" ("a source separation method for linear-quadratic mixtures", in French), private communication, Sept. 6, 2000.

[19] S. Hosseini, Y. Deville, "Blind separation of linear-quadratic mixtures of real sources using a recurrent structure", Proceedings of IWANN 2003, vol. 2, pp. 241-248, J. Mira and J. R. Alvarez eds (Springer), Mao, Menorca, Spain, June 3-6, 2003.

[20] S. Hosseini, Y. Deville, "Blind maximum likelihood separation of a linear-quadratic mixture", Proceedings of ICA 2004, pp. 694-701, ISSN 0302-9743, ISBN 3-540-23056-4, Springer-Verlag, vol. LNCS 3195, Granada, Spain, Sept. 22-24, 2004.

[21] J. W. Harris, H. Stocker, "Handbook of mathematics and computational science", Springer-Verlag, New York, 1998.

[22] Web site of ICA Central, http://www.tsi.enst.fr/icacentral/base_multi.html