IMPROVING ALGORITHM SPEED IN PNL MIXTURE SEPARATION AND WIENER SYSTEM INVERSION

Jordi SOLÉ-CASALS[1,2], *Massoud BABAIE-ZADEH*[2,3], *Christian JUTTEN*[2], *Dinh-Tuan PHAM*[4]

[1]Signal Processing Group, Universitat de Vic, Sagrada Família 7, 08500, Vic, Spain

[2]Lab. des Images et des Signaux, INPG, 46 Av. Félix-Viallet, 38031 Grenoble Cedex, France
[3] Electrical Engineering Department, Sharif University of Technology, Tehran, Iran
[4]Laboratoire de Modélisation et de Calcul, BP 53, 38041 Grenoble Cedex 9, France
Jordi.Sole@uvic.es, mbzadeh@yahoo.com,
Christian.Jutten@inpg.fr, Dinh-Tuan.Pham@imag.fr

ABSTRACT

This paper proposes a very simple method for increasing the algorithm speed for separating sources from PNL mixtures or inverting Wiener systems. The method is based on a pertinent initialization of the inverse system, whose computational cost is very low. The nonlinear part is roughly approximated by pushing the observations to be Gaussian; this method provides a surprisingly good approximation even when the basic assumption is not fully satisfied. The linear part is initialized so that outputs are decorrelated. Experiments shows the impressive speed improvement.

1. INTRODUCTION

Blind Separation of independent sources (BSS) is a basic problem in signal processing, which has been considered intensively in the last fifteen years, mainly for linear (instantaneous as well as convolutive) mixtures. More recently, a few researchers [1, 2, 3, 4, 5, 6, 7, 8] addressed the problem of source separation in nonlinear mixtures, whose observations are $\mathbf{e} = f(\mathbf{s})$. Especially Taleb and Jutten [6] have studied a special and realistic case of nonlinear mixtures, called post nonlinear (PNL) mixtures which are separable. As shown in Fig. 1, this two-stage system consists of a linear mixing matrix, followed by componentwise nonlinear distortions. It then provides the mixing observations:

$$e_i(t) = f_i(\sum_j a_{ij} s_j(t)) \tag{1}$$

where $s_j(t)$ are the independent sources, $e_i(t)$ is the i-th observation, a_{ij} denotes the entries of the unknown mixing

This work has been partly funded by the European project BLInd Source Separation and applications (BLISS, IST 1999-14190), by the SASI project (ELESA-IMAG) and by the Direcció General de Recerca de la Generalitat de Catalunya under a grant for Integrated Actions ACI2001, and by the Universitat de Vic under the grant R0912.

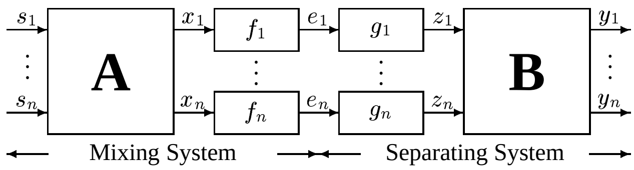

Fig. 1. The mixing-separating system for PNL mixtures.

matrix $\mathbf{A}$, and f_i is the unknown nonlinear mapping on the component i.

With a suitable parameterization, it can be easily shown that the problem of blind inversion of Wiener systems (Fig. 2) is equivalent to the source separation problem in PNL mixtures [9]. Its output writes as

$$e(t) = f(\sum_k h(k)s(t-k)) \tag{2}$$

where $s(t)$ is the independent and identically distributed (iid) input, $e(t)$ is the observation, $h(k)$ denotes the entries of the unknown filter H, assumed invertible, and f is the unknown nonlinear mapping, assumed invertible and memoryless.

Blind separation or inversion of the above models requires first to estimate the inverse of the nonlinear mapping, and then to inverse the linear part. This can be done by minimizing the mutual information of the inversion structure output. However, it leads to slow algorithms, especially with very hard nonlinearities, since the two parts are in cascade and optimized with the same criterion.

In this paper, we use a simple and very fast method for roughly estimating the inverse of the nonlinear mapping. Moreover, the linear part can be initialized so that the output vector $\mathbf{y}$ is spatially decorrelated (PNL case) or the output process y is time decorrelated (Wiener case), which is a first step to independence, very easy and fast to compute. These tricks are then used for initializing the inverse system. Section 2 and Section 3 explain the principles for initializing the nonlinear part and the linear part, respectively. Section 4 and Section 5 propose the algorithms for nonlinear part and linear part, respectively. Section 6 shows experimentally the efficacy of the method.

2. NONLINEAR PART INVERSION

In the model (1), consider the signal just before the nonlinear mapping. For the i-th component in the PNL mixture, The signal $x_i(t) = \sum_j a_{ij} s_j(t)$ is a weighted sum of random variables. According to the Central Limit Theorem, X_i tends toward a Gaussian random variable. The nonlinear mapping f_i changes the distribution, and consequently we can assume that the random variable $E_i = f_i(X_i)$ is farther from a Gaussian than X_i. Then, we propose to estimate the inverse of f_i, as the nonlinear mapping g_i which enforces the random variable $Z_i = g_i(E_i)$ to be Gaussian.

Of course, the Gaussian assumption will be satisfied if the number of sources s_j is large enough. For a small number of sources, the assumption is rough. The robustness of the method, with respect to this assumption, will be discussed in Section 6.

Similarly, in the Wiener system, the filtered signal $x(t) = \sum_k h(k)s(t-k)$, just before the nonlinearity, is a weighted sum of random variables. According to the Central Limit Theorem, the random variable X, associated to $x(t)$, tends to be closer to a Gaussian random variable than S. Of course, the quality of the approximation to a Gaussian variable depends on the filter, but X is closer to a Gaussian distribution than the distribution of the original source S. Moreover, the observed signal $E = f(X)$ generally is farther from a Gaussian than X. We then propose to approximate the inverse of f by the function g such that $g(E)$ is Gaussian.

In the following, since the two problems are very similar, we drop the index i for simplifying the notations.

2.1. Cumulative density function

The simplest approach for computing g is based on the property of the cumulative density function. Consider the random variable E, and denote $F_E(u)$ its cumulative density function:

$$F_E(u) = Pr(E < u) \tag{3}$$

where $Pr()$ denotes the probability.

The random variable $U = F_E(E)$ is then uniformly distributed in $[0, 1]$. Denoting $\Phi(u)$ the Gaussian cumulative density function, which transforms a unit variance Gaussian variable into a uniform random variable in $[0, 1]$, it is clear that $\Phi^{-1}(U)$ is a unit variance Gaussian random variable.

Then, a simple approximation of the inverse g of the nonlinear mapping f is:

$$\hat{g} = \Phi^{-1} \circ F_E \tag{4}$$

2.2. Maximization of Shannon entropy

Consider now the Shannon entropy of the unit variance random variable $Z = g(E)$

$$H(Z) = \int -\log p_Z(u) p_Z(u) du \tag{5}$$

where $p_Z(u)$ denotes the probability density function.

Since, for unit variance random variable, the Shannon entropy $H(Z)$ is maximum if Z is Gaussian [10], then g can be estimated so that $H(Z) = H(g(E))$ is maximum (under the constraint of unit variance).

3. LINEAR PART INVERSION

Since the objective of blind source separation is to recover independent signals at the output, we initialize the separation matrix $\mathbf{B}$ (see Fig.1) in order to have decorrelated signals at the output, which is a first step toward independence. This whitening process reduces the complexity of the problem, and has been used very often [11, 12, 13, 14] as a first separation stage. Here, it is only use for initializing the separation matrix $\mathbf{B}$. The whitening process can be summarized as follow: after mean substraction, we multiply $\mathbf{z}$ by the matrix $\mathbf{B_0}$ such that the covariance matrix of $\mathbf{y}$ is equal to the identity matrix $E\mathbf{y}\mathbf{y}^T = \mathbf{I}$. For determining $\mathbf{B_0}$, many algorithms exist since there are many matrices insuring decorrelation.

For the Wiener system inversion (see Fig. 2 and Fig. 3), we propose a similar idea for initializing the filter w. It consists in computing a Finite Impulse Response (FIR) filter, which produces temporally uncorrelated output samples. This part is implemented by using linear prediction. The estimate of the prediction coefficients (the initial filter w) is done by minimizing the mean-square error between the predicted signal and the actual signal.

4. NONLINEAR INVERSION ALGORITHMS

Using the previous results, one can propose two simple algorithms for the rough estimation of the inverse of the nonlinear mapping f. The first algorithm is based on the formula (4) derived in Subsection 2.1. The Matlab code is very simple and very fast.

A second algorithm, based on the result on Subsection 2.2, consists in adjusting a nonlinear mapping g so that the

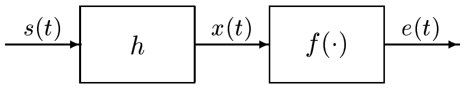

Fig. 2. A Wiener system consists of a filter followed by a distortion.

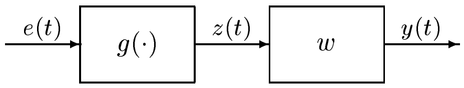

Fig. 3. A Hammerstein system consists of a distortion followed by a filter.

Shannon's entropy of z is maximum under the constraint $Ez^2 = 1$. Although the second idea is still quite simple, it leads to an algorithm which is much more complicated and longer to converge than the previous one.

Hence, in the following, we only give experimental results with the simplest and more efficient algorithm based on (4).

5. LINEAR INVERSION ALGORITHMS

As explained in section 3, the objective of the linear part (the matrix $\mathbf{B}$ or the filter w) is to decorrelate the output.

In the case of PNL mixtures, the algorithms to whiten the outputs are simple and easy to implement. We have used the method presented in [13], that writes as follows:

$$\mathbf{y} \leftarrow \mathbf{B_0}(\mathbf{y} - \langle \mathbf{y} \rangle) \tag{6}$$

with

$$\mathbf{B_0} = 2\sqrt{\langle \mathbf{y}\mathbf{y}^T \rangle^{-1}} \tag{7}$$

after which $\langle \mathbf{y}\mathbf{y}^T \rangle = \mathbf{I}$.[1]

For the Wiener system inversion, the filter w is initialized as the inverse of the prediction filter, *i.e.*: a Moving Average (MA) filter, in order to provide the innovation process at the output of the Hammerstein system (the inverse system), at the initialization. This can be done by using the LPC routine in Matlab environment, as follows:

```
% lpc filter coefficients
  w = lpc(z,Filter_Lengt);
% innovation process obtained by AR filtering
  y = filter(w,1,z);
```

[1] $\mathbf{X} = \sqrt{\mathbf{A}}$ is the positive definite square root of the matrix $\mathbf{A}$, *i.e.* $\mathbf{X} \cdot \mathbf{X} = \mathbf{A}$

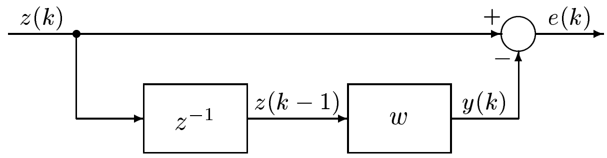

Fig. 4. LPC.

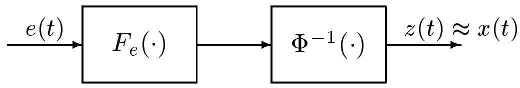

Fig. 5. The inversion system.

6. EXPERIMENTAL RESULTS

In this section we present a few examples in order to show the efficacy of the algorithm, for PNL mixtures or for Wiener systems. We checked, as expected by theory, that the algorithm is completely independent of f since, $\forall f$ the function $\Phi^{-1} \circ F_E \circ f$ transforms the random variable X to a Gaussian variable Z. If the compensation of the nonlinearity was perfect, the function $\Phi^{-1} \circ F_E \circ f$ should be the identity function. Of course, it can be rigorously true, only if X is Gaussian.

As a result, the efficacy of the method is only related on the distribution of X, just before the nonlinearity f: closer to the Gaussian the distribution X, better the approximation of $g = f^{-1}$.

6.1. PNL mixtures

For testing the algorithm of PNL source separation, we did experiments using two different separation methods, for mixtures of two uniformly distributed random sources. The first method is the algorithm proposed by Taleb and Jutten [6], denoted TJ, and the second method is the algorithm proposed by Babaie-Zadeh *et al.* [15], denoted BJN. Although the two methods are based on the minimization of the output mutual information , TJ method uses the marginal score function of the outputs while BJN uses the score function difference, which involves joint probability density functions (and consequently joint score functions). The two methods are batch gradient algorithms, *i.e.* one iteration means one adaptation step computed using the whole signals (500 samples). The mixing system is composed of:

$$\mathbf{A} = \begin{bmatrix} 1 & 0.4 \\ 0.7 & 1 \end{bmatrix} \tag{8}$$

$$f_1(x) = f_2(x) = 0.1x + \tanh(10x) \tag{9}$$

This mixture leads to the following joint distribution (Fig. 6), where the effect of the nonlinearities is clearly visible. Fig. 7 shows the scatter plot after the initialization

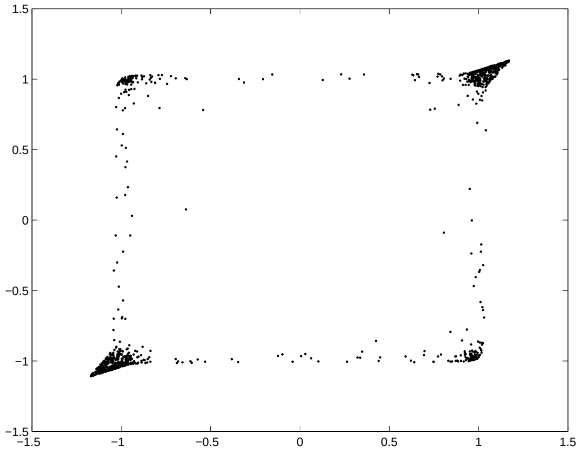

Fig. 6. Scatter plot of the observed signals.

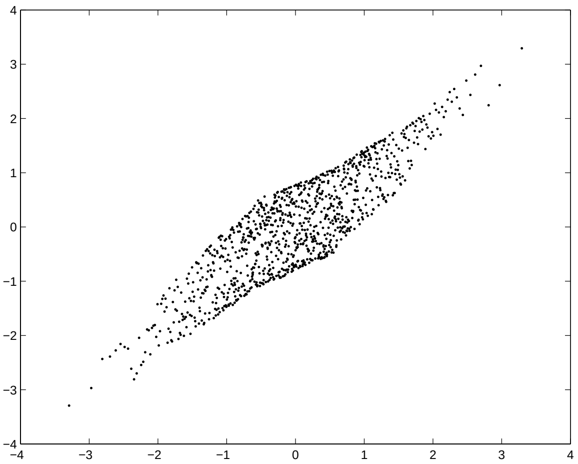

Fig. 7. Scatter plot of the signals after initializing nonlinear functions g.

of nonlinear functions g. Finally, Fig. 8 shows the scatter plot of the initialized outputs $\mathbf{y}$, where the signals are decorrelated. It is easy to see qualitatively the initialon provides an estimation $y(t)$ which is a mixture, simpler than $e(t)$.

Quantitative performance is measured with the SNR versus iterations, where SNR is:

$$\mathrm{SNR}_i = 10 \log_{10} E \left\{ \frac{s_i^2}{(y_i - s_i)^2} \right\} \qquad (10)$$

Despite hard nonlinearities ($0.05x + tanh(10x)$) are used in the experiments, the results obtained with the two methods are satisfactory. For TJ, the initialization process increases the convergence speed of the algorithm, and sometimes gives a better result in terms of output SNR (Fig. 9). For BJN, the initialization process also increases the convergence speed and, as the algorithm without initialization fails, can provide successful convergence (Fig. 10).

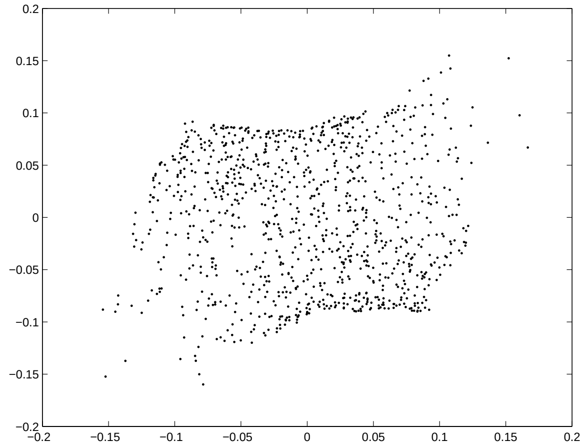

Fig. 8. Scatter plot of the decorrelated output signals.

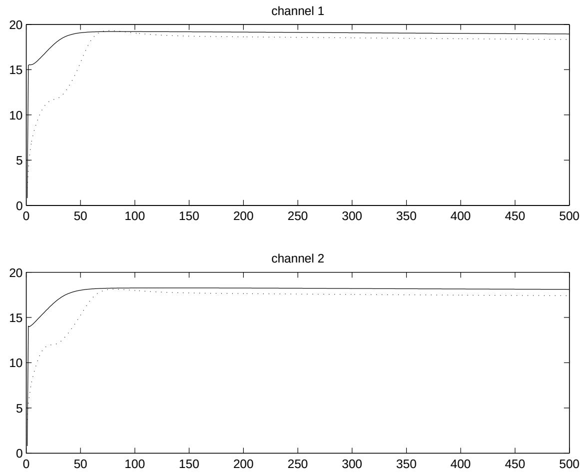

Fig. 9. SNR evolution for TJ algorithm with initialization process(solid line) and without (dashed line).

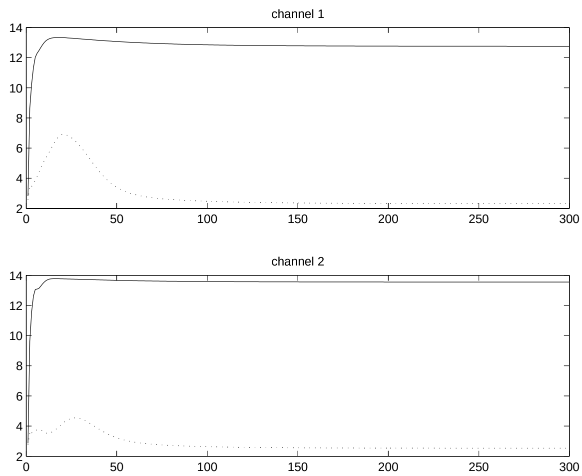

Fig. 10. SNR evolution for BJN algorithm with initialization process(solid line) and without (dashed line).

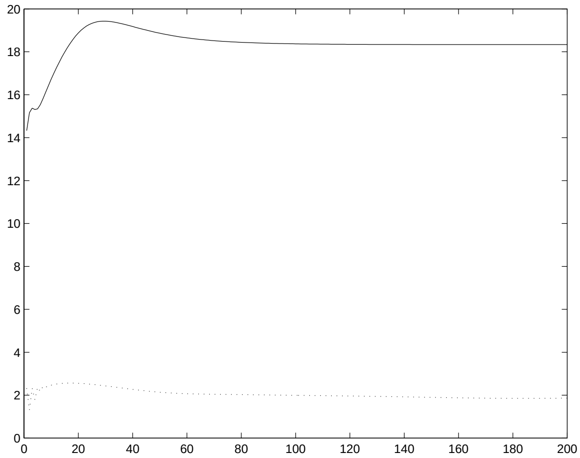

Fig. 11. SNR evolution for TSJ algorithm with initialization (solid line) and without (dashed line).

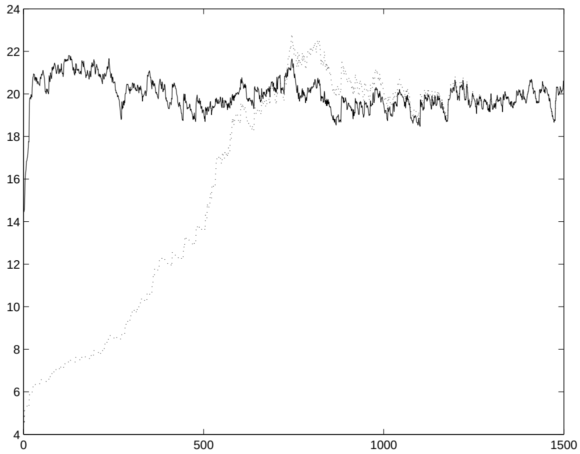

Fig. 12. SNR evolution for BSJ algorithm with initialization (solid line) and without (dashed line).

6.2. Wiener systems

For testing the algorithm in Wiener system inversion, we did experiments using two different methods. The first one, proposed by Taleb *et al.* is denoted TSJ [9] and the second one is derived from the algorithm proposed by Babaie-Zadeh *et al.* for PNL mixtures, denoted BSJP [16]. The two methods are based on the minimization of the output mutual information rate, which is the natural extension of the mutual information for stationary random processes [10]. As for PNL mixtures , TSJ method uses the marginal score function of the output, and BSJP uses the score function difference.

Fig. 11 shows output SNR versus iterations for TSJ algorithm, with a unit variance uniformly distributed random signal $s(t)$, filter $h = [1, 0.5, -0.2]$ and the nonlinearity $f(x) = x^3$. The initialization process always gives a good inversion, while without the initialization the result is poor.

Fig. 12 shows SNR versus iterations for BSJP case, with a unit variance uniformly distributed random signal $s(t)$, filter $h = [1, 0.5, -0.2]$ and the nonlinearity $0.1x + tanh(10x)$. We can observe the improvement of the speed of convergence.

The robustness of the nonlinearity compensation method, with respect to the distribution X, is studied in detail in [17] for Wiener systems, using ten different filters, and many sources $s(t)$, with positive as well as negative kurtosis, whose distributions are the averaged sum of two Laplacian distributions:

$$p(x) = \frac{b}{4}[\exp(-b|x-a|) + \exp(-b|x+a|)] \qquad (11)$$

With such signals, examples of a good and a poor nonlinear function compensation are shown in Fig. 13 and 14. The good compensation corresponds to 0.0238 kurtosis (close to a Gaussian kurtosis) and the poor case to -1.2631.

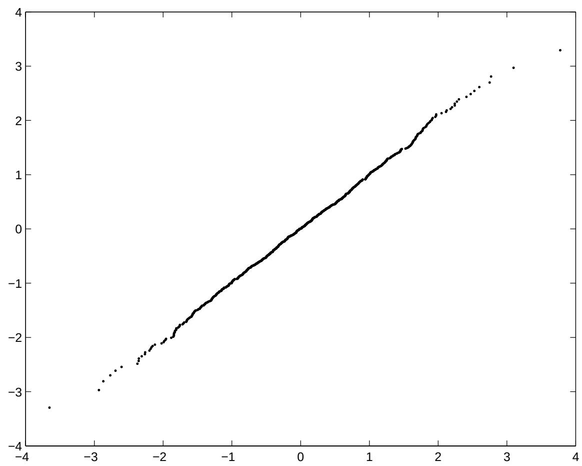

Fig. 13. Best case of nonlinear function compensation: the kurtosis of $x(t)$ is equal to 0.0238.

7. METHOD IMPROVEMENT

Of course, if the source $s(t)$ is very far from a Gaussian, *e.g.* a discrete-valued signal or signal with heavy tails, the method will give a crude approximation of g (see Fig. 14). However, using weak priors, it is possible to improve the compensation. For instance, if we know that $x(t)$ has heavy tails, we can compute z so that its distribution is Laplacian rather that a Gaussian. In fact, the algorithm will be quite similar: we replace in formula (4) Φ, the cumulative density function of the Gaussian distribution, by the cumulative density function of the Laplacian distribution. The difficulty is however that we don't know the true source distribution and even if we have some idea about this distribution, we cannot reliably deduce the distribution of X since it is a mixture of the sources with unknown mixing coefficients.

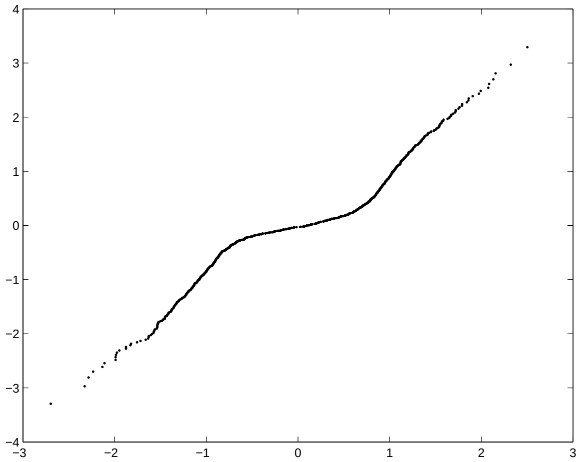

Fig. 14. Worst case of nonlinear function compensation: the kurtosis of $x(t)$ is equal to -1.2631.

8. CONCLUSION

In this paper, we propose a very simple and fast method for blindly initialize the inversion structure for PNL source separation and Wiener systems. For the nonlinear part, the method based on the assumption that the input variable of the nonlinear mapping is close to a Gaussian, leads to a rough and fast approximation of the nonlinear mapping. The linear part of PNL (or Wiener system, respectively) is initialize $\mathbf{B}$ (or w, respectively) so that the output random vector Y (random process $y(t)$, respectively) is spatially (temporally, respectively) whitened. This approach provides a good starting point which increases convergence speed of BSS or Wiener algorithms, with a very low computational cost.

9. REFERENCES

[1] L. Parra, G. Deco, and S. Miesbach, "Statistical independance and novelty detection with information preserving nonlinear maps," *Neural Computation*, vol. 8, pp. 260–269, 1996.

[2] P. Pajunen, A. Hyvärinen, and J. Karhunen, "Non linear source separation by self-organizing maps," in *ICONIP 96*, Hong-Kong, September 1996.

[3] A. Taleb and C. Jutten, "Entropy optimization, application to blind source separation," in *ICANN 97*, Lausanne (Switzerland), October 1997, pp. 529–534.

[4] T.-W. Lee, B. Koehler, and R. Orglmeister, "Blind separation of nonlinear mixing models," in *IEEE NNSP*, Florida, USA, 1997, pp. 406–415.

[5] H. H. Yang, S. Amari, and A. Cichocki, "Information-theoritic approach to blind separation of sources in nonlinear mixture," *Signal Processing*, vol. 64, no. 3, pp. 291–300, 1998.

[6] A. Taleb and C. Jutten, "Source separation in post nonlinear mixtures," *IEEE Tr. on SP*, vol. 47, no. 10, pp. 2807–2820, 1999.

[7] G. C. Marques and L. B. Almeida, "Separation of nonlinear mixtures using pattern repulsion," in *ICA 99*, Aussois (FRANCE), 1999, pp. 277–282.

[8] A. Ziehe, M. Kawanabe, S. Harmeling, and K.R. Muller, "Separation of post-nonlinear mixtures using ace and temporal decorrelations," in *ICA 2001*, San Diego - California, 2001, pp. 433–438.

[9] A. Taleb, J. Solé-Casals, and C. Jutten, "Quasi-nonparametric blind inversion of Wiener systems," *IEEE Tr. on Signal Processing*, vol. 49, no. 5, pp. 917–924, 2001.

[10] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, Wiley Series in Telecommunications, 1991.

[11] P. Comon, "Separation of sources using higher-order cumulants," in *SPIE Vol. 1152 Advanced Algorithms and Architectures for Signal Processing IV*, San Diego (CA), USA, August 8-10 1989.

[12] J.-F. Cardoso, "On the performance of orthogonal source separation algorithm," *Proceedings of EUSIPCO'94*, pp. 776–779, 1994.

[13] T. Bell and T. Sejnowski, "The 'independent components' of natural scenes are edge filters," in *Proc. Intern. Symp. on Nonlinear Theory and Applications*, 1995, pp. 43–47.

[14] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent component analysis*, Adaptive and Learning Systems for Signal Processing, Communications, and Control. John Wiley, New York, 2001.

[15] M. Babaie-Zadeh, C. Jutten, and K. Nayebi, "Blind separating convolutive post non-linear mixtures," in *Proceedings of ICA'2001*, San Diego, California, December 2001, pp. 138–143.

[16] M. Babaie-Zadeh, J. Solé-Casals, and C. Jutten, "Blind inversion of Wiener systems using a minimization-projection (MP) approach," in *Proceedings of ICA'03*, Nara, Japan, 2003, submitted.

[17] J. Solé-Casals, C. Jutten, and D.T. Pham, "Fast approximation of nonlinearities," *IEEE Signal Processing Letters*, 2002, submitted.