



**HAL**  
open science

## Fast Approximation of nonlinearities for improving inversion algorithms of PNL mixtures and Wiener systems

Jordi Solé, Christian Jutten, Dinh-Tuan Pham

► **To cite this version:**

Jordi Solé, Christian Jutten, Dinh-Tuan Pham. Fast Approximation of nonlinearities for improving inversion algorithms of PNL mixtures and Wiener systems. *Signal Processing*, 2005, 85 (9), pp.1780-1786. 10.1016/j.sigpro.2004.11.030 . hal-00379453

**HAL Id: hal-00379453**

**<https://hal.science/hal-00379453>**

Submitted on 28 Apr 2009

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Fast Approximation of Nonlinearities for improving inversion algorithms of PNL mixtures and Wiener systems <sup>★</sup>

J. Solé-Casals <sup>a</sup>, C. Jutten <sup>b,\*</sup>, D.T. Pham <sup>c</sup>

<sup>a</sup>*Signal Processing Group, University of Vic, Sagrada Família 7, 08500, Vic, Catalonia, Spain*

<sup>b</sup>*Laboratoire des Images et des Signaux (CNRS UMR n°5083), INPG, 46 Avénue Félix Viallet, 38031 Grenoble Cedex, France*

<sup>c</sup>*Laboratoire de Modélisation et de Calcul (CNRS UMR n°5523), IMAG, BP53, 38041 Grenoble Cedex 9, France*

---

## Abstract

This paper proposes a very fast method for blindly approximating a nonlinear mapping which transforms a sum of random variables. The estimation is surprisingly good even when the basic assumption is not satisfied. We use the method for providing a good initialization for inverting post-nonlinear mixtures and Wiener systems. Experiments show that the algorithm speed is strongly improved and the asymptotic performance is preserved with a very low extra computational cost.

*Key words:* Nonlinear source separation, Nonlinear ICA, Wiener systems

---

## 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,

---

<sup>★</sup> This work has been partly funded by the European project BLInd Source Separation and applications (BLISS, IST 1999-14190) and by the Universitat de Vic under the grant R0912.

\* Corresponding author

*Email addresses:* `jordi.sole@uvic.es` (J. Solé-Casals),  
`Christian.Jutten@inpg.fr` (C. Jutten), `Dinh-Tuan.Pham@imag.fr` (D.T. Pham).

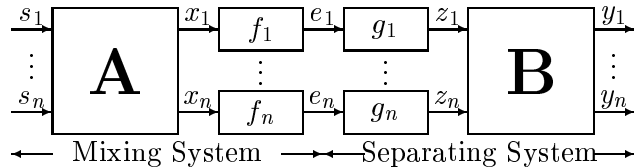


Fig. 1. The mixing-separating system for PNL mixtures.

mainly for linear (instantaneous as well as convolutive) mixtures. More recently, a few researchers [6,5,7,2,10,8,4,11] addressed the problem of source separation in nonlinear mixtures, whose observations are  $\mathbf{e} = f(\mathbf{s})$ , where  $f(\cdot)$  is an invertible nonlinear mapping. Especially, Taleb and Jutten [8] 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:

$$e_i(t) = f_i\left(\sum_j a_{ij}s_j(t)\right), i = 1, \dots, n \quad (1)$$

where  $s_j(t)$ ,  $j = 1, \dots, n$  are the independent sources,  $e_i(t)$  is the  $i$ -th observation,  $a_{ij}$  denotes the entries of the unknown mixing matrix  $\mathbf{A}$ , and  $f_i$  is the unknown nonlinear mapping on the component  $i$ .

With a suitable parameterization, 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

$$e(t) = f\left(\sum_k h(k)s(t-k)\right) \quad (2)$$

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  $h$  and  $f$  is the unknown nonlinear mapping, assumed invertible and memoryless.

Blind separation or inversion of the above models requires to estimate the inverses of the nonlinear mapping and of the linear part (mixing matrix or filter). This can be done by minimizing the mutual information of the inversion structure output. However, it leads to slow algorithms, since the two parts are in cascade and optimized with the same criterion.

In this paper, we propose a simple and very fast method for roughly estimating the inverse of the nonlinear mapping. This estimation provides a good initialization point, which can be used for any algorithm, remaining unchanged the asymptotic performance, but increasing strongly the algorithm speed, with a very small extra computational cost. Section 2 explains the principles. Section

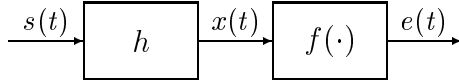


Fig. 2. A Wiener system consists of a filter followed by a distortion

3 shows experimentally the robustness and the performance of the method, before a short conclusion.

## 2 Principles

### 2.1 The basic assumption

In (1), consider the signal just before the nonlinear mapping. The  $i$ -th component,  $x_i(t) = \sum_j a_{ij}s_j(t)$ , 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 far from Gaussian. 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 coarse. The robustness of the method, with respect to this assumption, will be discussed in Section 3.

Similarly, in the Wiener systems 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 a Gaussian random variable. Of course, the vicinity to a Gaussian variable depends on the filter (especially on its number of taps), but  $X$  is closer to a Gaussian distribution than  $S$ . We then propose to approximate the inverse of  $f(\cdot)$  by the function  $g(\cdot)$  such that  $g(E)$  is Gaussian.

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

### 2.2 Cumulative density function

The simplest approach for computing  $g_i$  is based on the property of the cumulative density function (cdf). Consider the random variable  $E$ , and denote its cdf  $F_E(u) = Pr(E < u)$  where  $Pr(\cdot)$  denotes the probability. The random variable  $U = F_E(E)$  is then uniformly distributed in  $[0, 1]$ . Denoting  $\Phi(u)$

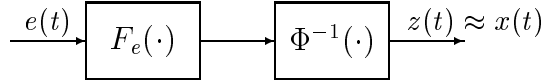


Fig. 3. The system for approximating nonlinearity

the Gaussian cdf, 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 (see Fig. 3):

$$\hat{g} = \Phi^{-1} \circ F_E \tag{3}$$

### 2.3 Maximization of Shannon entropy

Let  $p_Z(u)$  denote the probability density function of  $Z$ , the Shannon entropy of the unit variance random variable  $Z$ , defined by:

$$H(Z) = \int -\log p_Z(u) p_Z(u) du \tag{4}$$

is maximum if  $Z$  is Gaussian [1]. Then  $g$  can be estimated so that  $H(Z) = H(g(E))$  is maximum (under the constraint of unit variance).

### 2.4 Algorithms

Using the previous results, one can propose two algorithms for the rough estimation of the inverse of the nonlinear mapping  $f$ . The first algorithm is based on the formula (3). The Matlab code is very simple and very fast. A second algorithm, based on (4), consists in adjusting a nonlinear mapping  $g$  so that the 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 consider the algorithm based on (3).

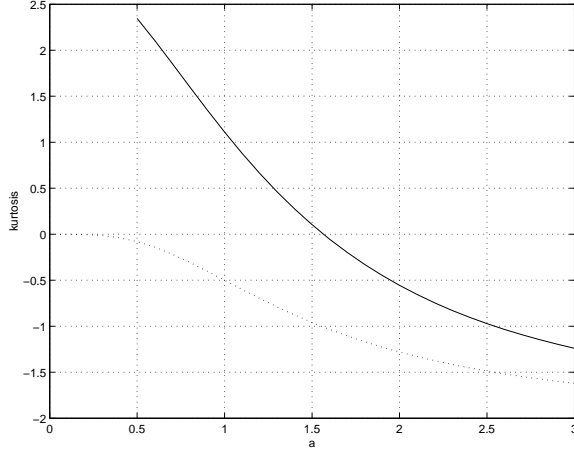


Fig. 4. Kurtosis of STL (solid line) and STG (dashed line) distributions versus  $a$

### 3 Experimental results

#### 3.1 Protocol

In order to test the robustness of the main assumption of the method (Gaussianization of  $x$ ), we did experiments using source signals whose distribution is the sum of two Laplacian (STL) or two Gaussian (STG) distributions. STL signals are interesting because one can obtain positive as well as negative signs of kurtosis by adjusting one or two parameters. As shown in [3], the pdf of the STL signal is:

$$p(x) = \frac{b}{4}[\exp(-b|x - a|) + \exp(-b|x + a|)] \quad (5)$$

and its kurtosis is:

$$K_x = 2 \frac{6 - a^4 b^4}{4 + 4a^2 b^2 + a^4 b^4} \quad (6)$$

Clearly the sign of the kurtosis can change according to the value of the parameters  $a$  and  $b$ :  $K_x \geq 0$  if  $0 < ab \leq 6^{\frac{1}{4}}$  and  $K_x < 0$  if  $ab > 6^{\frac{1}{4}}$ . We will use this signal, with  $b = 1$  and  $a$  in the range  $[0.5, 3]$ , *i.e.* the two Laplacian distributions only differ from the mean, ( $+a$  or  $-a$ ). The kurtosis of STL versus  $a$  is plotted in figure 4 (solid line).

In order to compare the results with the optimal (Gaussian) signal, we use STG signals. Each Gaussian distribution only differs by its mean ( $+a$  or  $-a$ ). The kurtosis is again a function of the distance  $2a$  between the means of the two distributions. It is equal to  $K_x = -2a^4/(1 + a^2)^2$ , and is plotted versus  $a$

Table 1

Filters used in the experiments

Low-pass filters :
$h_1 = [1, 0.5]$
$h_2 = [-0.1, -0.6579, -0.1]$
$h_3 = [1, 0.7, -0.5, 0.2]$
Band-pass filters :
$h_4 = [-0.1, 0.9, 0.1]$
$h_5 = [-0.1, 0, 0.9, 0, -0.1]$
$h_6 = [-0.0082, 0, -0.1793, 0, 0.6579, 0, -0.1793, 0, -0.0082]$
High-pass filters :
$h_7 = [1, -0.8]$
$h_8 = [-0.1, 0.6579, -0.1]$
$h_9 = [-0.1793, 0.6579, -0.1793, 0, -0.0082]$
All-pass filter
$h_{10} = [1]$

in Fig. 4 (dashed line).

Mixing of variables can be done according to either linear mixtures  $\mathbf{A}$  or linear filtering  $h$ . Each process (mixing or filtering) provides a sum of random variables which is closer to the Gaussian. For simplicity and due to the lack of space, we restrict the results to linear filtering according to (2), but similar results are obtained with (1). In fact, a  $n$ -tap filter provides a weighted sum of  $n$  delayed signals, similar to the mixture of  $n$  sources with the same distribution. In the source separation problem, sources will have different distributions, but the mixtures are still a weighted sum of sources and tend toward a Gaussian.

Then, the STL or STG signal feeds a Wiener system. Ten different filters (low-pass, high-pass or band-pass with various orders, see table 1) have been used for providing filtered signals with various distributions <sup>1</sup>. In Fig. 5 we plot the kurtosis of  $s(t)$  versus the kurtosis of  $x(t)$  for 13 inputs  $s(t)$  with various values of kurtosis and for the ten filters of Table 1. The figure shows, as expected, that the kurtosis of the filtered signal  $x(t)$  (which is a weighed sum of random variables) is moved towards zero which is the kurtosis value of the Gaussian kurtosis: the more taps the filter has, the closer to zero the

<sup>1</sup> As explained in the previous paragraph, a study with mixtures of 2 to 6 sources will give very similar results

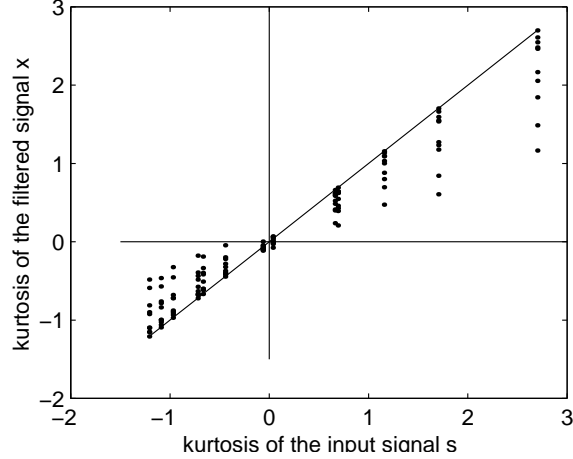


Fig. 5. Kurtosis of the input signal  $s$  versus kurtosis of the filtered signal  $x$ , for the ten filters of Table 1

kurtosis of the sum is ; the closer to zero the kurtosis of  $s(t)$  is, the closer to zero the kurtosis of the sum  $x(t)$  is. Then, the nonlinear mapping  $f$  provides  $e(t) = f(x(t))$ . We first check, 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  $Z$ . In fact, the accuracy of the estimation only depends on the distribution of  $X$ .

### 3.2 Results

The accuracy index  $\epsilon$  of the compensation will be simply the empirical mean square error<sup>2</sup> :

$$\epsilon = \frac{1}{T} \sum_{t=1}^T [(\Phi^{-1} \circ F_E \circ f)(x(t)) - x(t)]^2 \quad (7)$$

which measures the divergence between  $\Phi^{-1} \circ F_E \circ f(x)$  and the linear function  $x$ .

Fig. 6 and 7 show the accuracy index versus the kurtosis of the filtered signals  $x(t)$ , for STG and STL cases, obtained with the different sources and filters. One can remark that the error is minimum when the signal kurtosis is close to zero, *i.e.*  $X$  is close to a Gaussian, and increases as the kurtosis moves away from zero. As expected, the efficacy of the method is only related on the

<sup>2</sup> For computing  $\epsilon$ ,  $x(t)$  and  $z(t) = (\Phi^{-1} \circ F_E \circ f)(x(t))$  have to be normalized



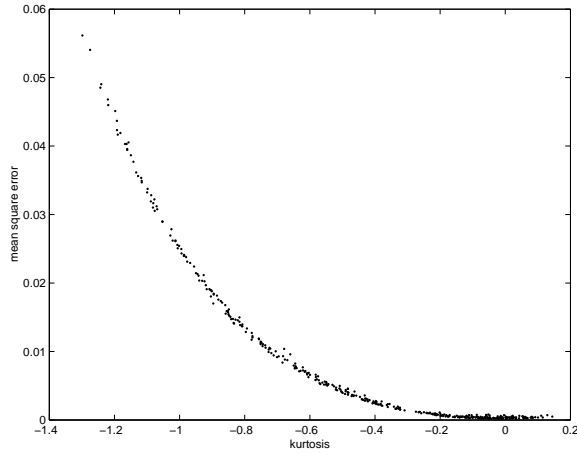


Fig. 6. Accuracy index  $\epsilon$  for STG distributions versus kurtosis

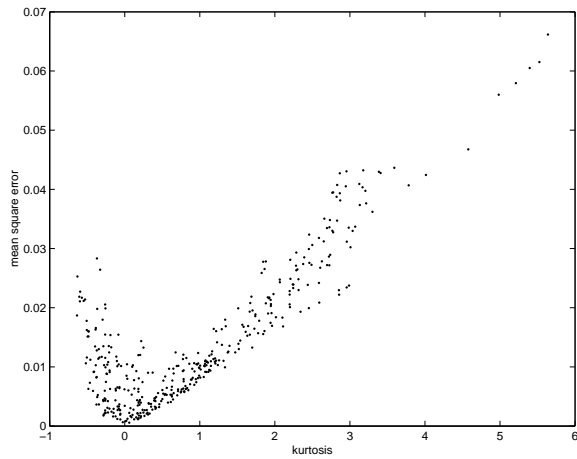


Fig. 7. Accuracy index  $\epsilon$  for STL distributions versus kurtosis

distribution of  $X$ , just before the nonlinearity  $f$ : the closer to the Gaussian the distribution is, the closer to zero the accuracy index  $\epsilon$  is. In Fig. 8, we show an example of the best (bottom) and the worst (up) compensations obtained in STL case. The best approximation corresponds to a kurtosis very close to zero ( $K_x = 0.0238$ ) and the worst case to  $K_x = -1.2631$ .

### 3.3 Application to blind inversion of PNL mixtures and Wiener systems

This experiment consists of 10 runs with STL sources whose kurtosis varies in  $[-1.7, +6.3]$ . The PNL mixtures is characterized by hard nonlinearities  $f_i(u) =$

Table 2

Performance of PNL with or without initialization

	average	std.	min	max
PI with init. (dB)	16.1	3.2	10.5	20
PI without init. (dB)	15.6	3.6	9.7	20.2
CS with init.	12.0	9.4	2	30
CS without init.	35.5	18.2	11	78

$0.1u + \tanh(10u)$ ,  $i = 1, 2$  and the following mixing matrix

$$\mathbf{A} = \begin{pmatrix} 1 & 0.4 \\ 0.7 & 1 \end{pmatrix} \quad (8)$$

The kurtoses of the linear mixtures  $X$  vary in the range  $[-1.2, +5.6]$ . The nonlinear part of separating structure is initialized with the method detailed above, and the linear part is initialized so that outputs are spatially decorrelated, *i.e.*  $E\mathbf{y}\mathbf{y}^T = \mathbf{I}$ . With this starting parameters, we used the algorithm [8]. Two performance indexes are measured, after normalisation of  $y_i$ :  $PI = 10 \log(E[s_i^2]/E[(y_i - s_i)^2])$ , which measures the separation performance, and the convergence speed,  $CS$ , *i.e.* the iteration number from which PI is over 90% of the asymptotic PI. The  $PI$ s (without (dashed line) and with (solid line) initialization) versus iterations of one typical run are shown in Fig. 9. The results (average, standard deviation, minimum and maximum values of PI and CS) of the 10 runs are presented in the Table 2.

The second set of experiments consists of 25 runs, in which the sources are 25 random STL signals  $s(t)$  with kurtosis in the range  $[-1.73, 0.79]$  filtered by the Wiener system, consisting of the filter  $h = [1, 0.5]$  and the nonlinear mapping  $f(u) = \tan(3u)$ . The kurtosis of  $x(t)$ , after the linear filter, is in the range  $[-1.15, 0, 78]$ . The nonlinear part of the separating structure is initialized by the method proposed in this paper, while the linear filter  $w$  is initialized so that its output  $y(t)$  is temporally decorrelated. Fig. 10 shows the  $PI$ 's for one run versus iterations, obtained with the algorithm [9] without (dashed line) and with (solid line) initialization. The table 3 give the averages, standard deviations, minimum and maximum of PI and CS, with and without algorithm initialization.

In the two experiments, one observes that the starting point does not affect the asymptotic performance since we use (with or without initialization) the same algorithm. On the contrary, the starting point leads to a significant convergence speed improvement, whatever the kurtosis of  $X$  is, *i.e.* even if the

Table 3

Performance of blind Wiener system inversion with or without initialization

	average	std.	min	max
PI with init. (dB)	17.5	3.3	10.2	23.9
PI without init. (dB)	15.4	3.5	9.6	23.3
CS with init.	12.3	15.6	1	44
CS without init.	89.7	37.1	34	154

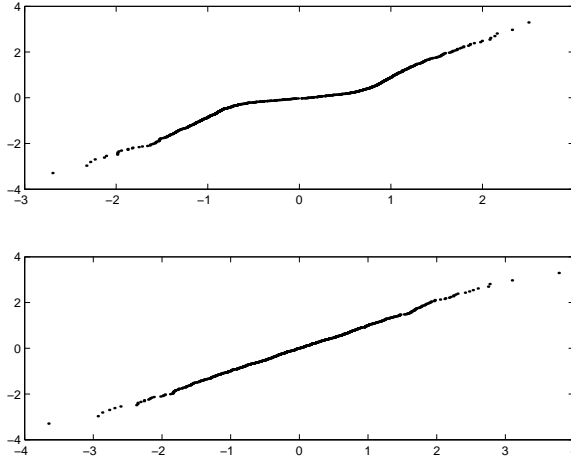


Fig. 8. Worst (up) and best (bottom) nonlinear function approximations for STL signals

approximation of  $g$  is very coarse. Moreover, in a few cases (with very particular sources, or very strong nonlinear mapping  $f$ ) where algorithms without initialization fail, algorithms with initialization converge. Finally, if the kurtosis of  $X$  is far from Gaussian, the approximation  $\hat{g}$  will be coarse; but provided that  $f$  is strongly nonlinear, the starting condition  $\hat{g} \circ f$  is closer to linearity than  $f$ , hence better.

## 4 Conclusion

In this paper, we propose a very simple and fast method for blindly approximating a nonlinear mapping. The method is based on the assumption that the input variable of the nonlinear mapping is Gaussian due to mixture or filtering. The results show the method is robust to this assumption. We then recommend to use this approximation for providing a good starting point in post nonlinear BSS or Wiener system inversion algorithms: with a very low

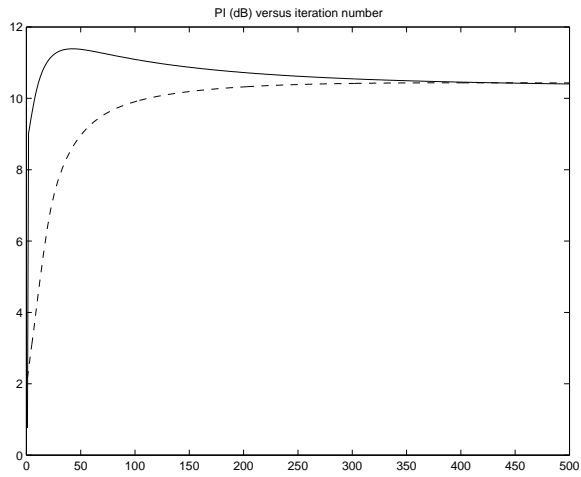


Fig. 9. PI (in dB) versus iterations for separating PNL without (dashed) or with (solid) initialization

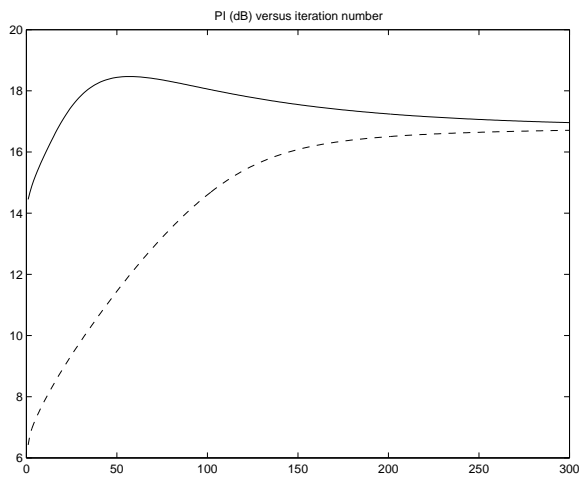


Fig. 10. PI (in dB) versus iterations for inverting Wiener system without (dashed) or with (solid) initialization

cost and the same SNR (PI), the average number of iterations for achieving convergence is divided by a coefficient 3 to 7.

## References

- [1] T.M. Cover, J.A. Thomas, Elements of Information Theory, Wiley Series in Telecommunications, 1991.
- [2] T.-W. Lee, B. Koehler, R. Orglmeister, Blind Separation of nonlinear mixing models, Proc. IEEE NNSP, Florida, USA, 1997, pp. 406-415.
- [3] A. Mansour, C. Jutten, What Should We Say About the Kurtosis?, IEEE Signal Processing Letters (6) (1999) 321-322.
- [4] G.C. Marques, L.B. Almeida, Separation of nonlinear mixtures using pattern repulsion, Proc. ICA'99, Aussois, France, 1999, pp. 277-282.
- [5] P. Pajunen, A. Hyvärinen, J. Karhunen, Non linear source separation by self-organizing maps, Proc. ICONIP 96, Hong-Kong, 1996.
- [6] L. Parra, G. Deco, S. Miesbach, Statistical independence and novelty detection with information preserving nonlinear maps, Neural Computation (8) (1996) 260-269.
- [7] A. Taleb, C. Jutten, Entropy Optimization, Application to Blind Source Separation, Proc. ICANN 97, Lausanne, Switzerland, October 1997, pp. 529-534.
- [8] A. Taleb, C. Jutten, Source separation in post nonlinear mixtures, IEEE Trans. on Signal Processing (47) (1999) 2807-2820.
- [9] A. Taleb, J. Solé-Casals, C. Jutten, Quasi-Nonparametric Blind Inversion of Wiener Systems, IEEE Trans. on Signal Processing (49) (2001) 917-924.

- [10] H.H. Yang, S. Amari, A. Cichocki, Information-theoretic approach to blind separation of sources in nonlinear mixture, *Signal Processing* (64) (1998) 291-300.
- [11] A. Ziehe, M. Kawanabe, S. Harmeling, K.-R. Müller, Separation of post-nonlinear mixtures using ACE and temporal decorrelations, *Proc. ICA'01*, San Diego, EE.UU., 2001, pp. 433-438.