EFFICIENT TECHNIQUE FOR BLIND SEPARATION OF POST-NONLINEAR MIXED SIGNALS

Usama Sayed Mohammed

Department of Electrical Engineering, Faculty of Engineering, Assiut University, Assiut, Egypt   Usama@aun.edu.eg

Hany Saber

Department of Electrical Engineering Faculty of Eng., South Valley University, Aswan, Egypt   hany@svu.edu.eg

This paper proposes a new method to solve the post-nonlinear blind source separation problem (PNLBSS). The method is based on the fact that the distribution of the output signals of the linearly mixed system are approximately Gaussian distributed. According to the central limit theory, if one can manage the probability density function (PDF) of the nonlinear mixed signals to be Gaussian distribution, then this means that the signals becomes linearly mixed in spite of the PDF of its separate components. In this paper, the short time Gaussianization utilizing the B-spline neural network is used to ensure that the distribution of the signal is converted to the Gaussian distribution. These networks are built using neurons with flexible B-spline activation functions. The fourth order moment is used as a measurement of Gaussianization. After finishing the Gaussianization step, linear blind source separation method is used to recover the original signals. Performed computer simulations have shown the effectiveness of the idea, even in presence of strong nonlinearities and synthetic mixture of real world data.

1. INTRODUCTION

The problem of blind source separation (BSS) is based on the recovery of independent sources from their mixture. This is important in several applications like speech enhancement, telecommunication, biomedical signal processing, etc. Most of the work on BSS mainly addresses the cases of instantaneous linear mixture [1-5]. Let A a real or complex rectangular (n×m; n≥m) matrix, the data model for linear mixture can be expressed as

$$X(t) = AS(t) + n(t) \qquad (1)$$

Where *S(t)* represents the statistically independent sources array and *n(t)* is noise vector while *X(t)* is the array containing the observed signals. For real world situation, however, the basic linear mixing model in (1) is too simple for describing the observed data. In many applications such as the nonlinear characteristic introduced by preamplifiers of receiving sensors, we can consider a nonlinear mixing. So a nonlinear

mixing is more realistic and accurate than linear model. For instantaneous mixtures, a general nonlinear data model can have the form

$$X(t) = f(S(t)) \qquad (2)$$

Where *f* is an unknown vector of real functions.

The linear instantaneous mixing (BSS) can be solved using the independent component analysis (ICA). The goal of the ICA is to separate the signals by finding independent component from the data signal. The ICA w commonly used in solving the problem of the linear mixing but in the nonlinear case, the ICA can not be used to solve this problem. It is important to note that if x and y are two independent random variables, any of their functions *f(x)* and *f(y)* are also independent. An even more serious problem is that in the nonlinear case, x and y can be mixed and still be statistically independent. Several authors [6-10] have addressed the important issues on the existence and uniqueness of solutions for the nonlinear ICA and BSS problems. In general, the ICA is not a strong enough constraint for ensuring separation in the nonlinear mixing case. Several known methods try to solve the nonlinear BSS problem. They can be roughly divided into algorithms with a parametric approach and algorithms with nonlinear expansion approach. With the parametric model, the nonlinearity of the mixture is estimated by parameterized nonlinearities. In [8] and [11], neural network is used to solve this problem. In the nonlinear expansion approach, the observed mixture is mapped into a high dimensional feature space and afterwards a linear method is applied to the expanded data. A common technique to turn a nonlinear problem into a linear one is introduced in [12].

1.1 Post Nonlinear Mixture Model

An important special case of the nonlinear mixture is the so-called post-nonlinear (PNL) mixture; the post nonlinear mixing model can be expressed as follows:

$$X(t) = f(AS(t)) \qquad (3)$$

Where *f* is an invertible nonlinear function that operates componentwise and A is a linear mixing matrix, more detailed:

$$X_i(t) = f_i\left(\sum_{j=1}^{m} a_{ij} \ S_j(t)\right), i = 1,.....,n \qquad (4)$$

Where a_{ij} are the elements of the mixing matrix A. The PNL model was introduced by Taleb and Jutten [14], as shown in Fig. 1. It represents an important subclass of the general nonlinear model and has therefore attracted the interest of several researchers [15-19]. Applications are found, for example, in the fields of telecommunications, where power efficient wireless communication devices with nonlinear class C amplifiers are used [20] or in the field of biomedical data recording, where sensors can have nonlinear characteristics [21].

In this work, a decoupled two-stage process to solve the PNL-BSS problem is proposed. First, linearization for each data vector is done based on short time Gaussianization to overcome the distortion of the nonlinear function f. The second step is the linear BSS techniques, in the wavelet domain, based on temporal decorrelation separation method (TDSEP) [29].
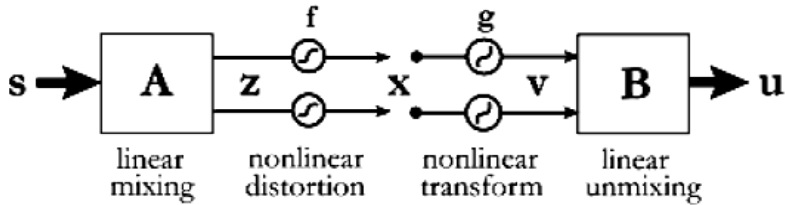
Fig.1 Building blocks of PNL mixing model and separation

1.2 Short Time Gaussianization

Short Time Gaussianization of the function *f* is a method to convert the distribution of the function into Gaussian random variable (Gaussianization). According to the central limit theory, the linearly mixed signals are most likely subjected to Gaussian distributions. Gaussianization technique is inspired by algorithms commonly used in random number surrogate generators and adaptive bin size methods [22]. The goal is to find functions h($\cdot$), such that:

$$h_i(x_i) \sim N(0, \sigma_i^2) \tag{5}$$

Where $N(0, \sigma_i^2)$ is a stander Gaussian distribution with zero mean and variance σ_i^2. The variance (σ_i^2) can be set to a constant ($\sigma_i^2 = 1$) without loss of generality. The previous method [22], which is based on the ranks of samples and percentiles of the Gaussian distribution, proceeds as follows:

Suppose that the observed samples $x_i[1]....,x_i[L]$ fulfill an ergodicity assumption and L is sufficiently large. Let **r[t]** is the rank of the tth sample **x_i[t]** and Φ ($\cdot$) is the cumulative distribution function of the standard Gaussian N (0, 1). The ranks "r" are assumed to be ordered from the smallest value to the largest value, **r[t]** = 1 for t = argmint **x_i[t]** and **r[t]**= L for t=argmaxt **x_i[t].** Then, the transformed samples will be

$$V(t) = \Phi^{-1}\left(\frac{r(t)}{L+1}\right), \qquad t = 1,.........,L \tag{6}$$

That is, the real values *v[t]* such that *Φ (v[t]) = (r[t] /L+1)* can be regarded as realizations from the standard Gaussian distribution [22]. Note that, adding 1 to the denominator is needed to avoid *v(t) = ±∞*. Furthermore, to limit the influence of outliers, it is recommended computing the modified versions of *v[t]* according to [22].

$$V(t) = \Phi^{-1}\left(c\frac{r(t)}{L+1} + (1-c)\frac{1}{2}\right)$$

However, the main drawbacks of this method are: (1) Gaussianization transform does not ensure that the transformed signals have all main features of the original nonlinear mixed signal. It depends on the choice of "c" and the rank method. (2) There is no measurement criterion of the Gaussianization to measure how far the distribution of the result signals from the Gaussian distribution. It is important to note that, the ranking process plays a role in the failure of this technique, especially when the data is sparse.

Another modified method to achieve Gaussianization is introduced in [23]. In this method, the data vector is divided into small blocks and then Gaussianization process of the blocks followed by the normalized kurtosis step is used to check the distribution of the data. Also, this method does not ensure the transformed signal have all information about the mixed signal.

1.3. B-Spline Neural Network [24]

The adaptive spline neural network (ASNN's) are designed using a neuron, called generalized sigmoid (GS-neuron), containing an adaptive parametric spline activation function. The basic network scheme is therefore very similar to classical multilayer perceptrons (MLP) structures, but with improved nonlinear adaptive activation functions. These functions have several interesting features: (1) easy to adapt, (2) retain the squashing property of the sigmoid, (3) have the necessary smoothing characteristics, (4) easy to implement both in hardware and in software. Spline activation functions are smooth parametric curves, divided in multiple tracts (spans) each controlled by four control points. Let $h(x)$ to be the nonlinear function to reproduce, then the spline activation function can be expressed as a composition of N-3 spans (cubic spline function) where N is the total number of the control points Q_j, where $j=1,2,\ldots\ldots,N$, each is depending on a local variable $u \in [0,1)$ and controlled by the control points as shown in Fig.2.

$$y = h(x) = \bar{h}(u,i) \tag{7}$$

where $\bar{h}(u,i)$ is the estimated parametric cubic spline function. The parameters can be derived by a dummy variable Z that shifts and normalizes the input.

$$Z_k = \frac{x}{\Delta x} + \frac{N-1}{2}$$

$$Z = \begin{cases} 1 & \text{if} \to Z_k < 1 \\ Z_k & \text{if} \to 1 \le Z_k \le N-3 \\ N-3 & \text{if} \to Z_k > N-3 \end{cases} \tag{8}$$

Where Δx is the fixed distance between two adjacent control points. The constraints imposed by (8) are necessary to keep the input within the active region that encloses the control points. Separating Z into integer and fractional parts using the floor operator $\lfloor . \rfloor$,

$$i = \lfloor Z \rfloor \qquad , \quad u = Z\text{-}i \tag{9}$$

In the matrix form, the output can be expressed as follows [24]:

$$y = \bar{h}(u,i) = \text{T} \bullet \text{M} \bullet Q \tag{10}$$

Where $\quad \text{T} = \begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix}$ and $Q_i = \begin{bmatrix} Q_i & Q_{i+1} & Q_{i+2} & Q_{i+3} \end{bmatrix}^{\text{T}}$ $\tag{11}$
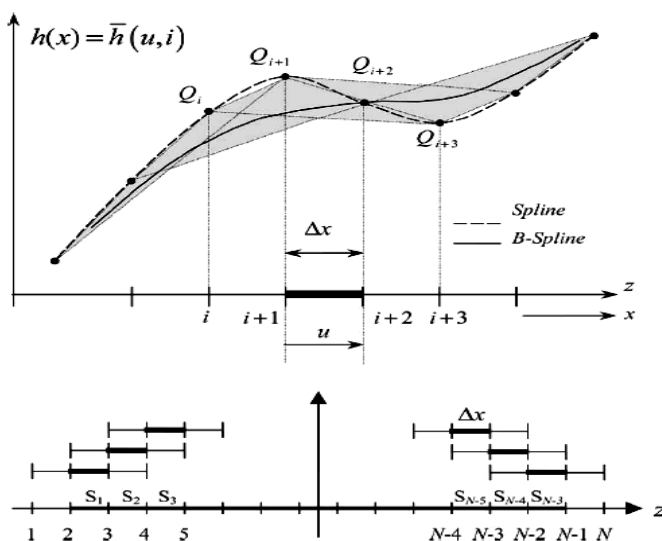
Fig. 2. Locally adaptive B-spline functions, each is defined inside a single span

$$M = \frac{1}{6}\begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 0 & 4 & 1 & 0 \end{bmatrix} \qquad \text{for } (B-spline)\,\text{function}$$

or

$$M = \frac{1}{2}\begin{bmatrix} -1 & 3 & -3 & 1 \\ 2 & -5 & 4 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 \end{bmatrix} \qquad \text{for } spline\,\text{function}$$

With $0 \le u < 1$ and M is the coefficient matrix of the Spline/B-spline interpolation function. In order to ensure the monotonously increasing characteristic of the function, the following additional constraint must be imposing:

$$Q_1 < Q_2 < Q_3 <,........< Q_N \qquad (12)$$

So in each time, only four control points will be involved in the training process. The activation function of the adaptive spline neural network is divided into two parts: GS1 which is used to determine i and u and GS2 which is used to determine Equation (10), as shown in Fig. 3.
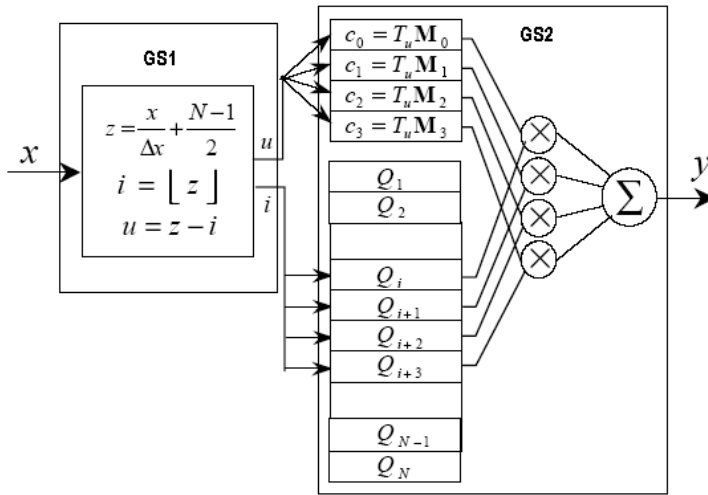
Fig.3. The spline activation function architecture.

2. THE PROPOSED SEPARATION METHOD

To solve the problem of the post-nonlinear blind source separation, we need to overcome the distortion of the nonlinear function f in (2) and then applying a linear blind source separation algorithm. As described in section 1.2, if the post nonlinear mixed signal is transformed into Gaussian distribution, the nonlinearity of the mixed signal can be eliminated and the problem will be transferred to linear blind source separation problem.

A new method for linearization using the principle of the short-time Gaussianization and the B-spline neural network is introduced. The proposed algorithm can be summarized as follows:

(1) Dividing the nonlinear mixed data in the time domain into blocks, for each data vector, and measuring the kurtosis K_i.

(2) Passing the data block through Gs1 and Gs2 (equations (8) to (11)) Fig .4.

(3) Applying Gaussianization on the output data from Gs2.

(4) Calculate the kurtosis K_{i+1} of linearization data using the normalized kurtosis. Then, check if $k_{i+1} > k_i$ go to step 1 and change the starting block size else go to the next step.

(5) Updating the control point of the Gs2 (the updating procedure will be described in Section 4).

(6) Go to step 2 and using the original block X and repeat the step 3 to step 5 until convergence of value of the kurtosis K.

(7) Applying step 1 to 7 on all data vectors.

(8) The output data Y is pre-whitened using eigenvalue decomposition of the zero time-lag correlation matrixes.

(9) n-scale wavelet packet transform is applied on the whitened subblocks resulting in n-subband signals.

(10) Evaluate N shifted covariance matrices among the two corresponding subband.

(11) The joint approximate diagonalization criteria is used with the N shifted covariance matrices to minimize the correlation among the subband and estimating the separating matrix B_i for each n-subband (B_1,….B_n) [25].

(12) For the separation matrices B_i, estimating from the subband of n-stage, each two matrices are used to compute and test the global matrices G (Equation (13) and (14)) [26].

(13) If some of subbands of the m-subband signals are correlated, then the signals will be decomposed using the wavelet packet transform into 2m-subband and repeat the process starting from step (10).

$$G_{ij} = B_i * B_j, \qquad \text{where i} \neq \text{j} \qquad (13)$$

$$PI = \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j=1}^{n} \left(\frac{[G]_{ij}}{max_j [G]_{ij}} - 1 \right) \qquad (14)$$

Where $[G]_{ij}$ is the (i,j)-element of the estimating matrix G. The smaller performance index (PI) implies usually better performance in separation.

(14) Using the separation matrix, which corresponding to the minimum correlated band, as a separation matrix of the linearized whitened-data.
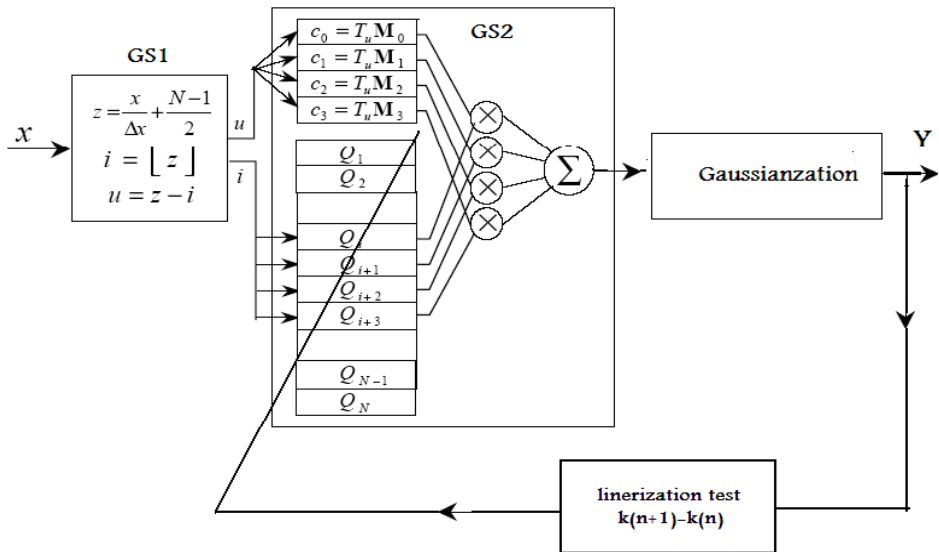

Fig.4. The block diagram of the linearization

3. ASSUMPTIONS AND DEFINITIONS

In the proposed algorithm, there are some assumptions are considered as follows: (1) the source signals must be independent, (2) the number of mixed signals are greater than or equal to the number of the source signals, (3) there are at most one source signal have Gaussian distribution, (4) the nonlinear function f is invertible function, and (5) the mixed matrix must be of full rank.

4. THE UPDATING PROCESS

In the beginning, the data vectors must be normalized to be with zero mean and unit variance. The fourth order cumulent (kurtosis) is used to measurement the efficiency of the linearization.

$$K(Y) = E\{Y^4\} - 3[(E\{Y^2\})]^2 \tag{15}$$

Since the data is normalized (zero mean and unity variance) the second term it the equation (15) E{Y2} will be equal to one, so the eq. (15) can be rewritten as follows:

$$K(Y) = E\{Y^4\} - 3 \tag{16}$$

So to drive the equation which controlling the updating of the control point vector Q, the derivative of the kurtosis with respect to the control points vector Q must be

calculated. Since the vector, Y is a function of the control point vector Q. So $\dfrac{\partial |K(Y)|}{\partial Q}$

will equal to:

$$\frac{\partial |K(Y)|}{\partial Q} = 4 \ \text{sign}(K(Y)) \left[E\left\{ \dot{Y}(Y)^3 \right\} \right] \tag{17}$$

Where $\dot{Y}$ is the derivative of the Y with respect to the control point vector Q.

$$\frac{\partial |K(Y)|}{\partial Q_i} = 4 sign(K(Y)) \left[E\left\{ \dot{T} \bullet M_m \bullet Q_i \left(T \bullet M_m \bullet Q_i \right)^3 \right\} \right] \tag{18}$$

Where

$$\dot{T} = \begin{bmatrix} 3u_y^2 & 2u_y & 1 & 0 \end{bmatrix}, \quad m \in \begin{pmatrix} 0 & 1 & 2 & 3 \end{pmatrix}$$

and M_m is the mth column of the M matrix, and the updating of the control point vector will be

$$Q_{i+m}[\rho + 1] = Q_{i+m}[\rho] - \eta \ \frac{\partial |K(Y)|}{\partial Q_i} \tag{19}$$

Where η is the learning step (equal to 0.0001 in this work). The expectation E operator can be dropped in the updating of the control points in Equation (18) and (19) but it cannot be dropped in the kurtosis Equation (15) and (16).

5. SIMULATION RESULTS AND IMPLEMENTATION

Several types of signals with are used to evaluate the performance of the proposed technique. For the following experiments $\Delta x = 0.2$ and the initial value of the control points is selected from sampling the following sigmoid function. $\left(1 - e^{-x} \middle/ 1 + e^{-x} \right)$

Experiment 1: The proposed algorithm is applied on audio signal with length of 32000 samples mixed with Gaussian noise $N(0,5)$ using Equation (20) and Equation (21). Fig. 5 shows the original, the mixing and the estimating sources. Moreover, the

scattering plot of the original signals, the mixed signals and the estimating signals are shown in Fig 6.

$$Z(t) = AS(t) \tag{20}$$

Where

$$A = \begin{bmatrix} .8 & .2 \\ .3 & .92 \end{bmatrix} \text{ and}$$

$$X_1(t) = e^{(Z_1(t)) + .3Z_1(t)} \tag{21}$$

$$X_2(t) = \cos(Z_2(t)) \, \dot{\sin}(3Z_2(t))$$

Experiment 2: The proposed algorithm is applied on audio signal with length of 32000 samples mixed with Gaussian noise $N(0,7)$ using Equation (20) and Equation (22). Fig 7 shows the original, the mixing and the estimating sources. Moreover, the scattering plot the mixed signals and the estimating signals are shown in Fig 8.

$$A = \begin{bmatrix} .85 & .35 \\ .25 & .92 \end{bmatrix} \text{ And}$$

$$X_1(t) = \sin(Z_1(t)) + 0.1Z_1(t)$$
$$X_2(t) = \cos(2Z_2(t)) + 0.1Z_2(t) \tag{22}$$
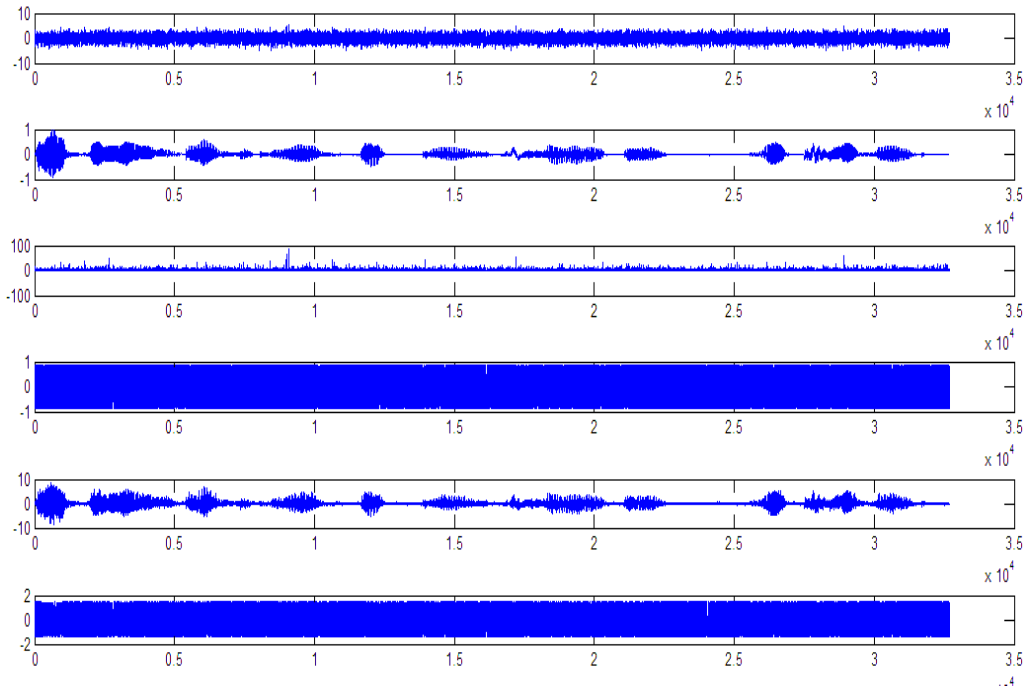

Fig. 5 the original data are the   first two signals on the top, the mixing data are in the middle and the recovering data are in the last two rows.

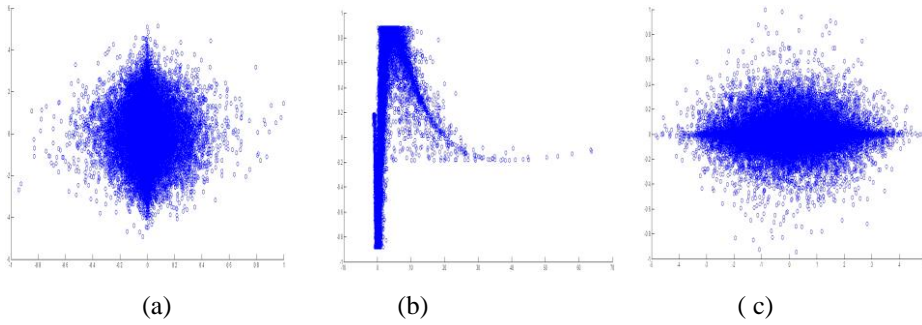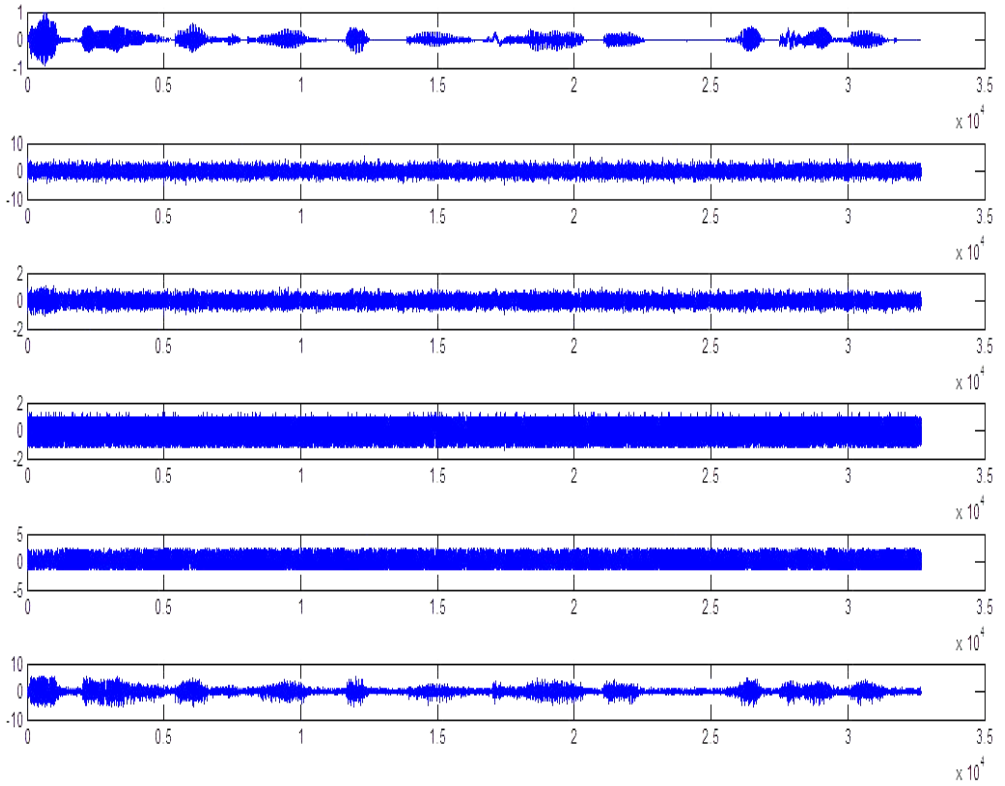(a)   (b)   (c)

Fig.6. (a) scatter plot of the original data. (b) Scatter plot of the mixed data. (c) scatter plot of the estimated data

Fig. 7 the original data are the first two signals on the top, the mixing data are in the middle and the recovering data are in the last two rows.
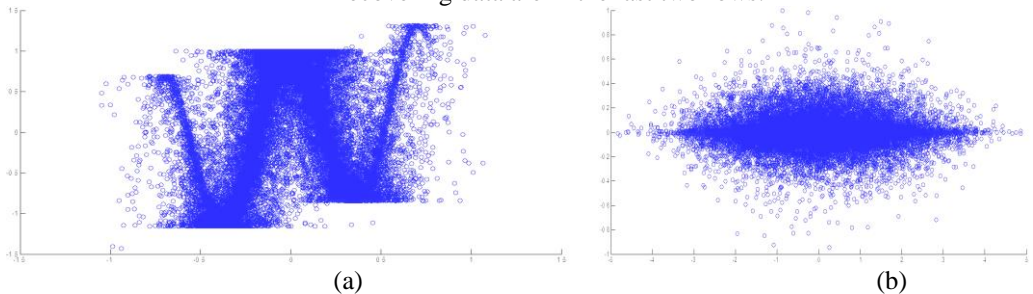

(a)   (b)

Fig.8. (a) scatter plot of the mixed data. (b) scatter plot of the estimated data

Experiment 3: The proposed algorithm is applied to two audio signals with length of 30000 samples using the following mixing system (the mixing system is introduced in [27]). Fig 9 shows the original, the mixing and the estimating sources. Moreover, the scattering plot of the original signals, the mixed signals and the estimating signals are shown in Fig 10. The mixed model uses a random matrix A and nonlinear function Where

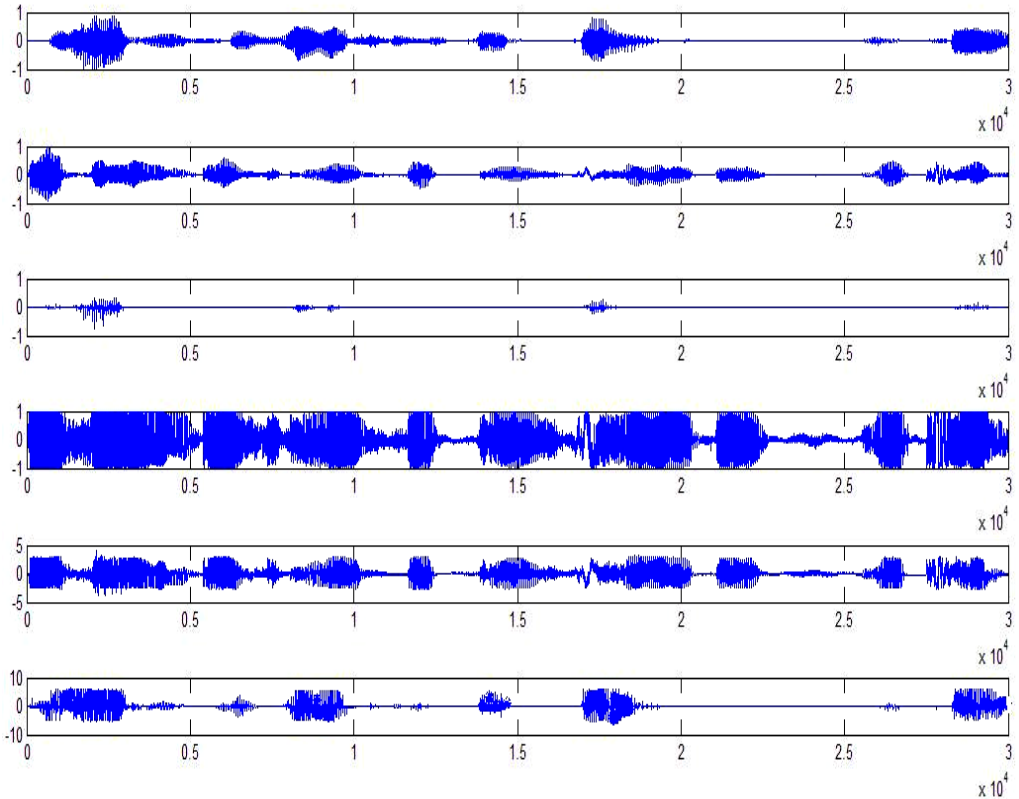$$X_1(t) = (Z_1(t))^3$$
$$X_2(t) = \tanh(10Z_2(t))$$

Fig. 9 the original data are the first two signals on the top, the mixing data are in the middle and the recovering data is in the last two rows.
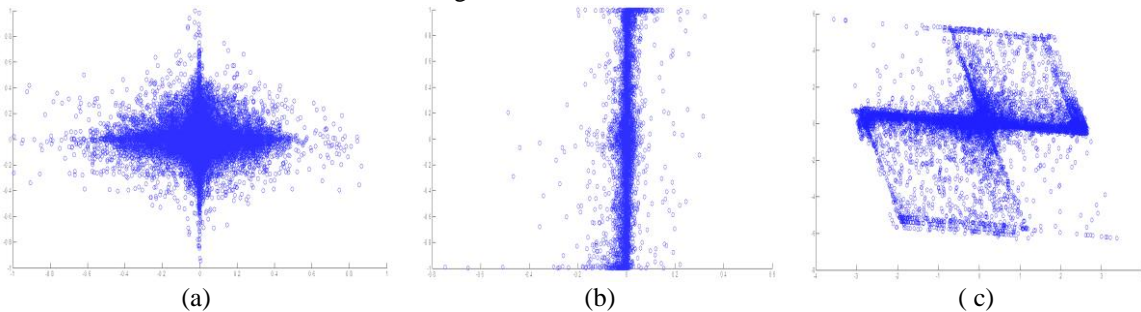

Fig.10. (a) scatter plot of the original data. (b) Scatter plot of the mixed data. (c) scatter plot of the estimated data

Experiment 4: Consider a two-channel nonlinear mixture with cubic nonlinearity: this model is introduced in [28].

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = B \begin{bmatrix} (.)^3 \\ (.)^3 \end{bmatrix} \left(A \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} \right)$$

Where the matrices B and A are defined as follows:

$$B = \begin{bmatrix} .25 & .86 \\ -.86 & .25 \end{bmatrix} \qquad A = \begin{bmatrix} .5 & .9 \\ -.9 & .5 \end{bmatrix}$$

The source vectors S(t) is given by an amplitude-modulated signal and a sinusoidal signal as follows:

$$S_1(t) = (0.5 + \sin(6\pi t))\cos(100\pi t)$$
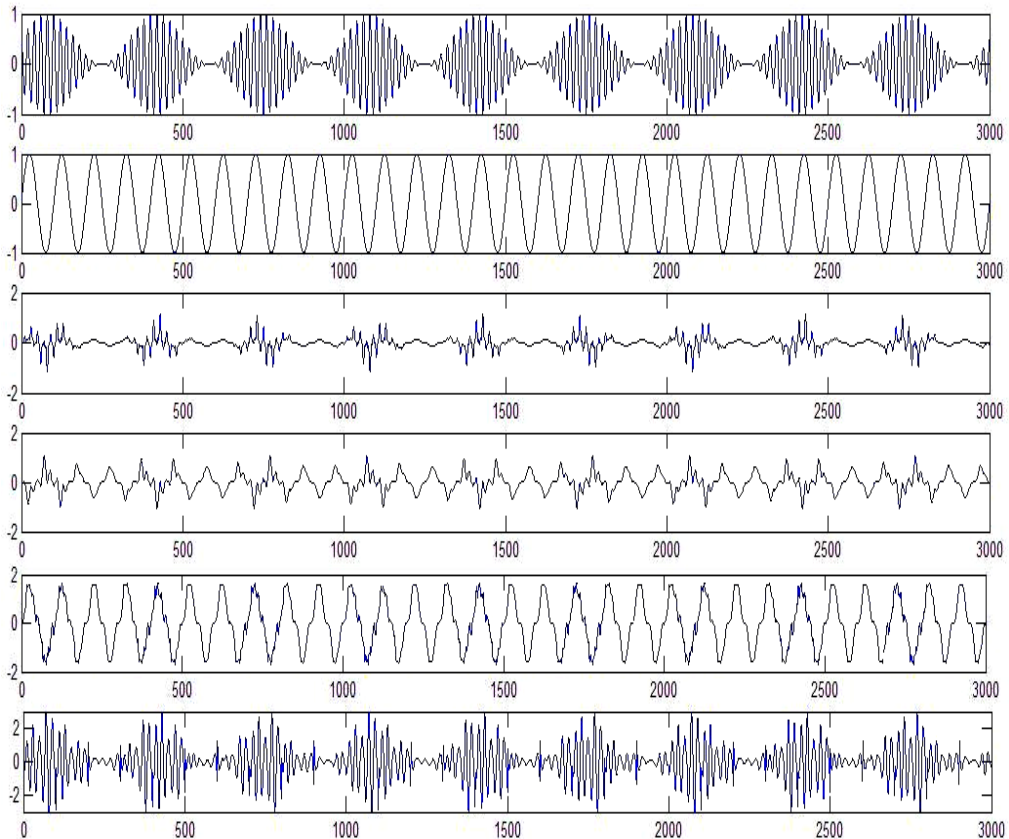$$S_2(t) = \sin(20\pi t)$$

Fig. 11 the original data are the first two signals on the top, the mixing data are in the middle and the recovering data are in the last two rows.

The performance of the proposed technique is introduced in a comparison form with the Gauss-TD separation algorithm [18] and adaptive spline neural network (ASNN) separation algorithm [28]. The last example is used to compare the

performance of the proposed technique with the Gauss-TD separation algorithm and adaptive spline neural network (ASNN) separation algorithm Fig.12 Shows the original, the mixing and the estimating sources when applying the Gauss-TD on the post-nonlinear mixed signal example 4. The results of applying the ASNN algorithm on the same signal is introduced in [28].
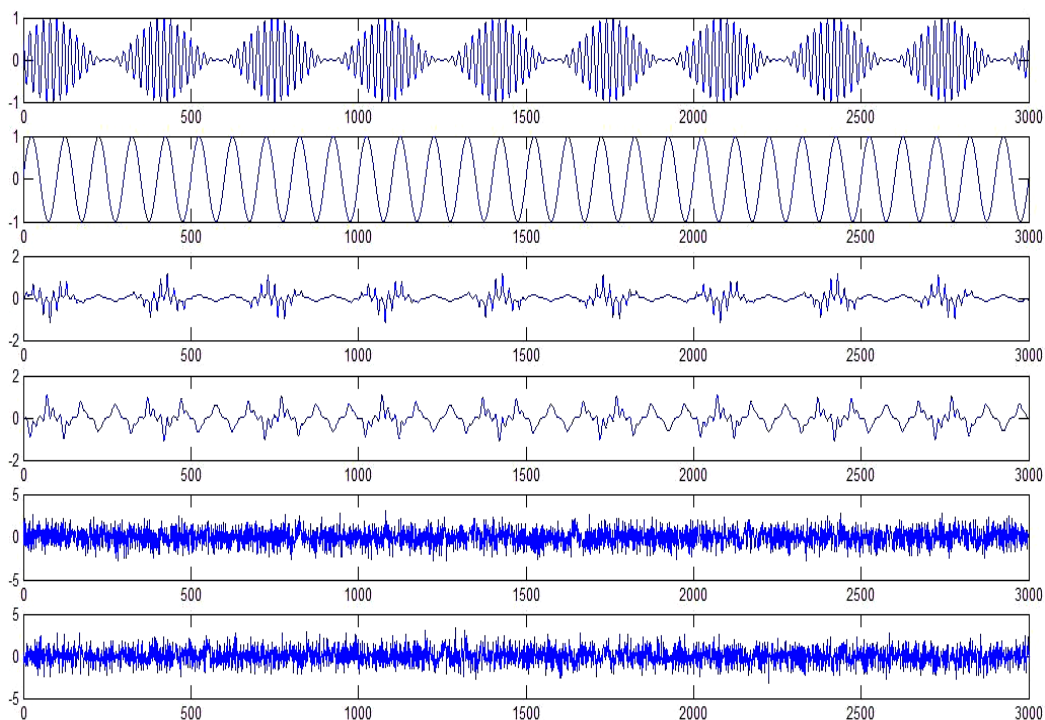

Fig. 12 The results from Gauss-TD algorithm when we applied it on experiment 4

From the results of Fig.12 it is clearly that, an improved in the separation performance is achieved by the proposed technique over the Gauss-TD separation technique. The Gauss-TD technique [18] is failed completely in recovering the signals. Moreover, when applying the ASNN algorithm on the same signals, the recovered low frequency signal is inverted. Anther important advantages of our proposed technique over the ASNN [28] can be summarized as follows: (1) the learning rule of the proposed algorithm is simpler then the complex learning rule of the ASNN algorithm, (2) our suggested separation system is simpler then ASNN separation system[28]

6. CONCLUSIONS AND DISCUSSION

In this paper, a new algorithm to solve the problem of the post nonlinear blind source separation is introduced. The algorithm depends on short time Gaussianization and B-spline neural network to perform linearization of the observed signals. By transforming the distribution of the nonlinear mixed signal into Gaussian distribution, the nonlinear distortion is reduced. Then linear blind source separation algorithm can be applied on the data to complete the separation process. The new approach works much more

efficiently than the Gauss-TD separation algorithm [18] and the adaptive spline neural network (ASNN) separation algorithm [28]. Moreover, the proposed algorithm is simple to impalement and the features of the original signals are completely recovered because the proposed algorithm does not depend on the rank of the signal. In our technique, the spline function ensures that the data is in the specific range. Our extensive experiments have confirmed that the use of the proposed procedure provides promising results even with the hard nonlinear mixing function.

REFERENCES

[1]   Amari, A. Cichocki, and H. H. Yang, A new learning algorithm for blind signal separation. In NIPS 95, MIT Press, (1996) 882–893.

[2]   Bell and T. Sejnowski, information-maximization approach to blind separation and blind deconvolution, Neural Computation,7 (1995) 1129–1159.

[3]   Cardoso, Infomax and maximum likelihood for source separation, IEEE Letters on Signal Processing, 4(1997) 112–114.

[4]   W. Lee, M. Girolami, and T. Sejnowski. ,Independent component analysis using an extended infomax algorithm for mixed sub-gaussian and super-gaussian sources, Neural Computation, 11(1999) 417–441.

[5]   D. T. Pham and J.-F. Cardoso, Blind separation of instantaneous mixtures of non-stationary sources, IEEE Trans. on Signal Processing, vol. 49(9) Sep. (2001) 1837-1848.

[6]   C. Jutten and A. Taleb, Source separation: from dusk till dawn, in Proc. 2nd Int. Workshop on Independent Component Analysis and Blind Source Separation (ICA2000), Helsinki, Finland, (2000) 15–26..

[7]   A. Hyv¨arinen and P. Pajunen, Nonlinear independent component analysis: Existence and uniqueness results, Neural Networks, 12(3) (1999) 429–439.

[8]   A. Taleb and C. Jutten, Source separation in post-nonlinear mixtures, IEEE Trans. on Signal Processing, 47(10) (1999) 2807–2820.

[9]   A. Taleb, A generic framework for blind source separation in structured nonlinear models," IEEE Trans. on Signal Processing, 50(8) (2002) 1819–1830.

[10]   J. Eriksson and V. Koivunen, Blind identifiability of class of nonlinear instantaneous ICA models, in Proc. of the XI European Signal Proc. Conf. (EUSIPCO 2002), Toulouse, France, 2,September (2002) 7–10.

[11]   Luis B. Almeida, Linear and nonlinear ICA based on mutual information, the MISEP method, Signal Processing, Special Issue on Independent Component Analysis and Beyond 84(2) (2004) 231–245.

[12]   Tobias Blaschke and Laurenz Wiskott . Independent Slow Feature Analysis and Nonlinear Blind Source Separation .Proc. of the 5th Intl. Conf. on Independent Component Analysis and Blind Signal Separation Granada, Spain, September (2004) 22-28

[14]   Taleb and C. Jutten. Nonlinear source separation: The post-nonlinear mixtures. In Proc. European Symposium on Artificial Neural Networks, Bruges, Belgium, (1997)279–284.

[15]   T. W. Lee, B.U. Koehler, and R. Orglmeister. Blind source separation of nonlinear mixing models. IEEE International Workshop on Neural Networks

for Signal Processing,(1997) 406–415.

[16]   H.H.Yang, S. Amari, and A. Cichocki. Information-theoretic approach to blind separation of sources in non-linear mixture. Signal Proc., 64(3)(1998)291–300,

[17]   A. Taleb and C. Jutten. Source separation in post-nonlinear mixtures. IEEE Trans. on Signal Processing, 47(10)(1999)2807–2820.

[18]   A. Ziehe, M. Kawanabe, S. Harmeling, and K.-R. M¨uller. Separation of post-nonlinear mixtures using ACE and temporal decorrelation. Workshop on Independent Component Analysis and Blind Signal Separation (ICA2001), San Diego, California, (2001) 433–438.

[19]   C. Jutten and J. Karhunen. Advances in nonlinear blind source separation. In Proc. of the 4th Int. Symp. on Independent Component Analysis and Blind Signal Separation (ICA2003), Nara, Japan, Invited paper in the special session on nonlinear ICA and BSS, April (2003)245–256.

[20]   L. E. Larson. Radio frequency integrated circuit technology low-power wireless communications. IEEE Personal Communications, 5(3)(1998)11–19.

[21]   A. Ziehe, K.-R. M¨uller, G. Nolte, B.-M. Mackert, and G. Curio. Artifact reduction in magnetoneurography based on time-delayed second-order correlations. IEEE Trans. Biomed. Eng., 47(1)(2000)75–87.

[22]   H. Kantz and T. Schreiber. Nonlinear time series analysis. Cambridge University Press, Cambridge,UK, 1997.

[23]   Y Yanlu Xie, Beiqian Dai and Jun Sun, Kurtosis Normalization after Short-Time Gaussianization for Robust Speaker Verification, Proceedings of the 6th World Congress on Intelligent Control and Automation, Dalian, China June(2006) 21 - 23

[24]   Stefano Guarnieri, Francesco Piazza, Multilayer Feedforward Networks with Adaptive Spline Activation Function, IEEE Transactions On Neural Networks, 10(3) May (1999).

[25]   Jean-François Cardoso and Antoine Souloumiac. Jacobi angles for simultaneous diagonalization, Siam J. Mat. Anal. Appl., 17(1)(1996)161–164.

[26]   T. Tanaka and A. Cichocki, Subband decomposition independent component analysis and new performance criteria, in Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP2004), Montreal, Canada, May (2004)541-544,

[27]   TeTe-Won Lee, Jean-Franc¸ois Cardoso, Erkki Oja and Shun-Ichi Amari. Blind Separation of Post-nonlinear Mixtures using Linearizing Transformations and Temporal Decorrelation. Journal of Machine Learning Research 4 (2003) 1319-1338 .

[28]   MiMirko Solazzi and Aurelio Uncini, "Spline Neural Networks for Blind Separation of Post-Nonlinear-Linear Mixtures" IEEE transactions on circuits and systems—i: regular papers, vol. 51, no. 4, p.p. 817-829, 2004.

[29]   A. Ziehe and K.-R. Müller.. TDSEP—an efficient algorithm for blind separation using time Structure, In Proc. Int. Conf. on Artificial Neural Networks (ICANN'98),(1998) 675–680.

طريقه فعالة لفصل الإشارات المبهمة والمخلوطة لاخطيا

في هذه المقالة تم اقتراح طريقة جديدة يمكن بها حل مشكلة الخلط المبهم الغير خطى للإشارات. تم في هذه الطريقة تحويل التوزيع الخاص بالأشارة المخلوطة الى توزيع جاوسيان اعتمادا على النظرية التى أثبتت أن الأشارات المخلوطة خطيا يكون فى الغالب توزيعها جاوسيان. بالطبع الفكرة هنا ليست جديدة ولكن الجديد فى الطريقة المقترحة هى تفادى الخطأ (فى الأبحاث السابقة) فى عملية تحويل التوزيع الخاص بالأشارة المخلوطة الى توزيع جاوسيان والذى كان من أهم عيوب هذه الطريقة. لقد تم تفادى هذا الخطأ عن طريق أستخدام الشبكة العصبية من النوع B-Spline فى عملية التحويل مع ضبطها تلقائيا لأنها تمتاز بالمرونة. وقد تم أتمام عملية الفصل بعد ذلك بأستخدام طريقة للفصل الخطى داخل التحويل المويجى والتى تمتاز أيضا بالوصول السريع الى مصفوفة الفصل. ولاختبار الطريقة المقترحة تم تنفيذها و عمل محاكاة لها بأستخدام الحاسب وتم تجريتها على أنواع مختلفة من الإشارات (الصوت, الموسيقى,....) وعدد مختلف من مصفوفات الخلط. وقد تم مقارنة كفاءة هذه الطريقة مع بعض الطرق السابقة وقد أثبتت هذه الطريقة كفاءة عالية في عملية الفصل لهذه الإشارات بالمقارنة مع هذه الطرق.