

## A Spectral Clustering Approach to Underdetermined Postnonlinear Blind Source Separation of Sparse Sources

Steven Van Vaerenbergh and Ignacio Santamaría

**Abstract**—This letter proposes a clustering-based approach for solving the underdetermined (i.e., fewer mixtures than sources) postnonlinear blind source separation (PNL BSS) problem when the sources are sparse. Although various algorithms exist for the underdetermined BSS problem for sparse sources, as well as for the PNL BSS problem with as many mixtures as sources, the nonlinear problem in an underdetermined scenario has not been satisfactorily solved yet. The method proposed in this letter aims at inverting the different nonlinearities, thus reducing the problem to linear underdetermined BSS. To this end, first a spectral clustering technique is applied that clusters the mixture samples into different sets corresponding to the different sources. Then, the inverse nonlinearities are estimated using a set of multilayer perceptrons (MLPs) that are trained by minimizing a specifically designed cost function. Finally, transforming each mixture by its corresponding inverse nonlinearity results in a linear underdetermined BSS problem, which can be solved using any of the existing methods.

**Index Terms**—Blind source separation, multilayer perceptrons, post-nonlinear (PNL) mixtures, sparse sources, spectral clustering, underdetermined source separation.

### I. INTRODUCTION

Blind source separation (BSS) is an important problem in the signal processing area, with a number of applications in communications, speech processing and biomedical signal processing. The goal of BSS is to recover the  $n$  source signals from their  $m$  observed linear or nonlinear mixtures [1], [2].

The first BSS algorithms focused only on linear mixtures. Different approaches were taken depending on the number of mixtures,  $m$ , versus the number of sources,  $n$ . For the case where as many mixtures as unknown sources are available ( $m = n$ ), a number of techniques have been developed. Most of them stem from the theory of independent component analysis (ICA) [3], [4], a statistical technique whose goal is to represent a set of random variables as linear functions of statistically independent components. If there are more mixtures than sources ( $m > n$ ), the redundancy in information can be used to achieve additional noise reduction [5].

On the other hand, if there are fewer mixtures than sources ( $m < n$ ), we have an *underdetermined* BSS problem, which can only be solved if we rely on *a priori* information about the sources. Specifically, a number of algorithms that assume sparse sources have been proposed for underdetermined BSS [6]–[8].

A considerable amount of research has also been done on the so-called postnonlinear BSS problem (PNL BSS), in which the sources are first mixed linearly and then transformed nonlinearly. For an equal number of mixtures and sources ( $m = n$ ), some algorithms have been proposed [9]–[12]. However, these algorithms cannot deal with the more restricted problem of *underdetermined* PNL BSS. An *underdetermined* PNL BSS algorithm was recently proposed in [13]; it nevertheless requires the number of active sources at each instant to be

lower than the number of mixtures  $m$  and assumes noiseless mixtures. The approach presented in this letter relaxes these restrictions on the sources and it is able to work with noisy mixtures.

The rest of the letter is organized as follows: A description of the mixing process is given in Section II. The proposed algorithm consists of two major stages. The first one is a spectral clustering method, which is described in Section III. The second part, which deals with the estimation of the inverse nonlinearity through a set of multilayer perceptrons, is explained in Section IV, and in Section V simulation results are presented. Finally, Section VI summarizes the main conclusions of this letter.

### II. PROBLEM STATEMENT

#### A. Model for Sparse Sources

In order to model sources with different degrees of sparsity we consider the probability density function (pdf)

$$p_{S_i}(s_i) = p_i\delta(s_i) + (1 - p_i)f_{S_i}(s_i), \quad i = 1, \dots, n \quad (1)$$

where  $p_i$  is the probability that a source is inactive,  $\delta(\cdot)$  denotes Dirac's delta and  $f_{S_i}(s_i)$  is the pdf of the  $i$ th source when it is active [8].

#### B. Linear Mixture Model

In a general linear mixture model, the measurement random vector  $\mathbf{y} \in \mathbb{R}^{m \times 1}$  can be described as

$$\mathbf{y} = \mathbf{A}\mathbf{s} + \mathbf{n} \quad (2)$$

where  $\mathbf{s} \in \mathbb{R}^{n \times 1}$  is an independent random vector representing the sources,  $\mathbf{A} \in \mathbb{R}^{m \times n}$  is the unknown mixing matrix,  $\mathbf{n} \in \mathbb{R}^{m \times 1}$  is an independent random vector with Gaussian white noise representing sensor noise.

For  $m \geq n$ , several algorithms exist that estimate the unmixing matrix  $\mathbf{W} = \mathbf{A}^{-1}$  sufficiently well [4]. For  $m < n$  the mixing matrix is not square and the problem cannot be solved without additional information about the sources. In the absence of noise, if only source  $i$  is active, the output signal  $\mathbf{y}$  will be aligned with the vector representing the  $i$ th column of  $\mathbf{A}$ , the  $i$ th “basis vector” [7]. Therefore, if the sources are sparse according to the model described in (1), most of the output samples  $\mathbf{y}$  will be aligned with one of the basis vectors [see Fig. 2(a)].

Using this geometrical insight a large number of estimators for the mixing matrix have been proposed, amongst them a technique using overcomplete representations [6], a line spectrum estimation method [14] and a number of geometric algorithms [15], [16]. Once the mixing matrix has been estimated, the original sources can be recovered with the shortest-path algorithm introduced in [7].

#### C. PNL Mixture Model

In a realistic scenario the  $m$  sensors that measure the mixtures show some kind of nonlinearity, which suggests the extension of (2) to a PNL mixture model

$$\mathbf{x} = \mathbf{f}(\mathbf{A}\mathbf{s}) + \mathbf{n} \quad (3)$$

where  $\mathbf{f}(\cdot) = [f_1(\cdot), f_2(\cdot), \dots, f_m(\cdot)]^T$  is a componentwise nonlinear function and  $\mathbf{x} \in \mathbb{R}^{m \times 1}$  is the measurement random vector. For the underdetermined case ( $m < n$ ) the linear BSS methods are not able to estimate the sources properly. A scatter plot example of PNL mixtures is shown in Fig. 2(b).

The proposed algorithm aims at estimating the inverse nonlinearities  $\mathbf{g} = \mathbf{f}^{-1}$ , under the condition that they are invertible and linear for small input values. This leads directly to an estimate of the linear

Manuscript received June 7, 2005; revised September 20, 2005. This work was supported by the Ministerio de Educación y Ciencia (MEC), Spain, under Grant TEC2004-06451-C05-02.

The authors are with the Department of Communications Engineering, University of Cantabria, Santander 39005, Spain (e-mail: steven@gtas.dicom.unican.es; nacho@gtas.dicom.unican.es).

Digital Object Identifier 10.1109/TNN.2006.872358

mixtures  $\mathbf{y} = \mathbf{g}(\mathbf{x})$ , which can be used to recover the original sources  $\mathbf{s}$  relying on known methods for underdetermined linear BSS.

### III. DIVIDING THE SAMPLES INTO $n$ CLUSTERS

#### A. Spectral Clustering

Spectral clustering [17] is a technique that clusters points based on a spectral analysis of the matrix of point-to-point similarities. This ‘‘affinity’’ matrix is formed as  $F_{ij} = \exp(-d^2(\mathbf{x}_i, \mathbf{x}_j)/\sigma^2)$  where  $d(\mathbf{x}_i, \mathbf{x}_j)$  is some distance measure between points  $\mathbf{x}_i$  and  $\mathbf{x}_j$  and  $\sigma$  is a scaling constant. Spectral clustering obtains good clustering performance in cases where classic methods such as k-means fail, for instance when one data set is surrounded by another one. Nevertheless, its efficiency depends on the choice of  $\sigma$ . In addition, since it requires obtaining the eigenvectors of an  $N \times N$  matrix, care should be taken to reduce the computational cost when the number of samples  $N$  is large.

In [18], Zelnik-Manor and Perona proposed to calculate the affinity matrix as  $F_{ij} = \exp(-d^2(\mathbf{x}_i, \mathbf{x}_j)/(\sigma_i\sigma_j))$  where the ‘‘local scale’’  $\sigma_i = d(\mathbf{x}_i, \mathbf{x}_L)$  is the distance between  $\mathbf{x}_i$  and its  $L$ th neighbor, with  $L$  constant and depending on the data dimension. This has certain advantages over the original spectral clustering. First, the results do not depend on the choice of  $\sigma$ . Second, in the context of the underdetermined PNL BSS problem points with a higher local scale will likely correspond to multiple active sources, since the local scale is inversely proportional to the local density of points [see Fig. 2(a) and (b)].

#### B. Preprocessing

Some preprocessing steps are taken to facilitate the spectral clustering. Basically, the mixture samples are roughly reduced to those for which only one source was active at each instant. Apart from guaranteeing the overall efficiency of the algorithm, this reduction also lowers the computational cost.

Central samples are removed because they correspond to inactive sources and are almost unaffected by the nonlinearity. If  $p_i = p, \forall i$ , the probability of having no active sources at all according to the sparse source model (1) is  $p^n$ , so the  $\nu_1 = p^n N$  samples closest to the origin can be removed. In addition, ‘‘nonsparse’’ samples, which are the result of multiple sources active at the same time, are also removed. They can be estimated as the  $\nu_2 = [1 - n(1 - p)p^{n-1} - p^n]N$  samples with highest local scale.

If the sources have different  $p_i$ -values,  $\nu_1$  and  $\nu_2$  can easily be calculated according to the previous description. In practice, rough (over-) estimates can be used for  $\nu_1$  and  $\nu_2$ . Especially when the  $p_i$  are unknown,  $\nu_1$  and  $\nu_2$  should be chosen so that after preprocessing the remaining samples can be clustered into nonoverlapping clusters.

#### C. Identification and Clustering Limitations

The performance of the clustering algorithm will depend on the distance between points of different clusters. If clusters still overlap or come too close after preprocessing, spectral clustering will not be possible. This cluster ‘‘separability’’ depends mainly on the nonlinearity  $\mathbf{f}$  and the mixing matrix.

Furthermore, it is assumed that the different sources have double sided distributions. By applying spectral clustering, it is then possible to distinguish  $2n$  clusters, one for each sidelobe of the  $n$  distributions. And since the nonlinearities are assumed to be linear for small input values, determining which pair of clusters correspond to the same source can be done by looking at which clusters have the same slopes close to the origin. Finally,  $n$  clusters are obtained, corresponding to the  $n$  sources.

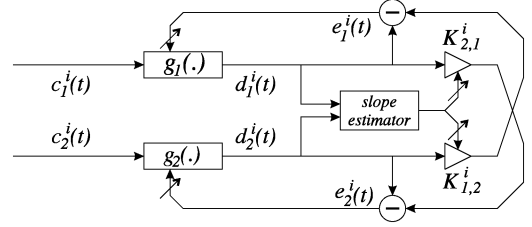


Fig. 1. The block diagram used for the MLP parameter training for  $m = 2$ . The blocks labeled  $g_1(\cdot)$  and  $g_2(\cdot)$  represent the two MLPs. The slope estimator is used to estimate the slope  $K_{1,2}^i$  of the curve formed by  $(d_1^i(t), d_2^i(t))$ . To train the upper MLP, we use as desired signal  $K_{1,2}^i d_2^i(t)$ . In this way, the error signal  $e_1^i(t) = d_1^i(t) - K_{1,2}^i d_2^i(t)$  measures the deviation from linearity of this curve. The same procedure is carried out for the lower MLP.

### IV. ESTIMATING THE INVERSE NONLINEAR FUNCTIONS

To estimate the inverse nonlinearities in [13], Theis and Amari consider that there should be a linear relationship between the same component of different clusters. Here, we exploit that there should be a linear relationship between the *different components of the same cluster*. Both approaches can lead to a reliable estimation of the inverse nonlinearities, but the latter allows to operate directly on the available data. By doing so it avoids the interpolation needed in [13] that can be problematic in cases of strong nonlinearities or in the presence of noise.

#### A. MLP Model

To represent each inverse nonlinear function  $g_j(\cdot)$  ( $j = 1, \dots, m$ ) we use a single input, single output multilayer perceptron with one hidden layer of  $r$  neurons. Once the samples are clustered into  $n$  sets by the spectral clustering algorithm, the elements of each set are used as input patterns for the  $m$  MLPs. In particular, for the  $i$ th cluster we have patterns  $\mathbf{c}^i(t) = (c_1^i(t), c_2^i(t), \dots, c_m^i(t))$ , where  $t$  is a discrete time unit. The  $j$ th component of each pattern is fed into the  $j$ th MLP, whose output is given by

$$d_j^i(t) = g_j(c_j^i(t)) = \mathbf{w}_{j,2}^T \phi(\mathbf{w}_{j,1} c_j^i(t) + \mathbf{b}_{j,1}) + b_{j,2} \quad (4)$$

where  $\mathbf{w}_{j,1}, \mathbf{w}_{j,2} \in \mathbb{R}^{r \times 1}$  are weight vectors,  $\mathbf{b}_{j,1} \in \mathbb{R}^{r \times 1}$  and  $b_{j,2} \in \mathbb{R}$  are biases and  $\phi(\cdot)$  is a neuron activation function. For all the neurons in the hidden layers we chose to use the hyperbolic tangent activation function.

#### B. Cost Function and Parameter Training

In order to train the MLP weights and biases a cost function is designed that allows training blindly on the available nonlinear cluster data. The cost should be minimal when these data are mapped onto linear clusters by the entire set of MLPs, therefore ideally there should be a linear relationship between all components  $j$  and  $k$  within the same cluster  $i$

$$K_{j,k}^i = \frac{d_j^i(t)}{d_k^i(t)} \quad \forall t \quad (5)$$

where  $K_{j,k}^i$  is the slope made up by the components  $j$  and  $k$  of cluster  $i$ . Hence, a cost function to train the weights of MLP  $j$  can be derived as

$$J_j = \sum_{i=1}^n \sum_{k=1}^m \sum_t \left[ d_j^i(t) - K_{j,k}^i d_k^i(t) \right]^2 \quad (6)$$

Notice that the elements  $K_{j,k}^i$  must also be estimated and updated in each iteration, for instance by using the histogram-based estimator described in [8]. Fig. 1 shows the training diagram corresponding to the case  $m = 2$ .

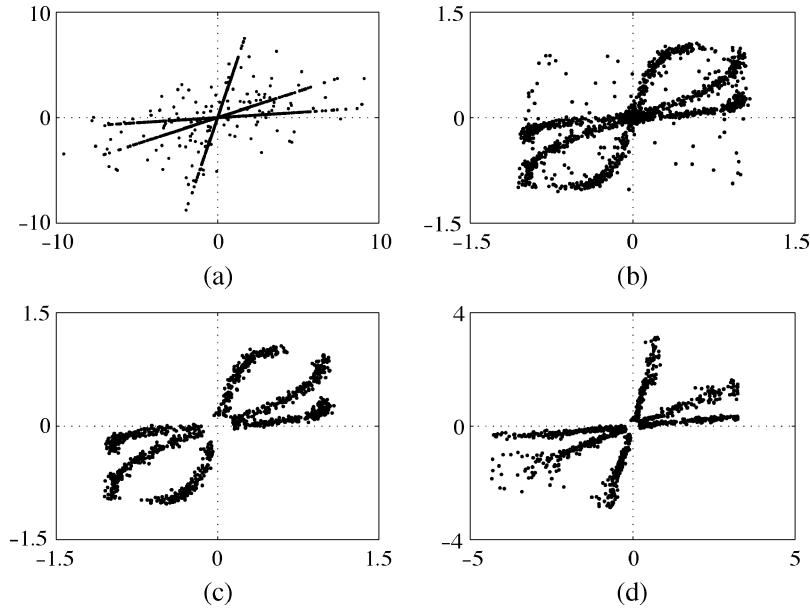


Fig. 2. Example of underdetermined BSS mixtures: scatter plots of three linear mixtures in (a) and three PNL mixtures with additive noise in (b), with  $p = 0.9$  and 20 dB SNR. The preprocessing of Section III removes some samples of (b) to obtain (c), which is then used for spectral clustering. (d) Output of the MLPs after training with the clustered data.

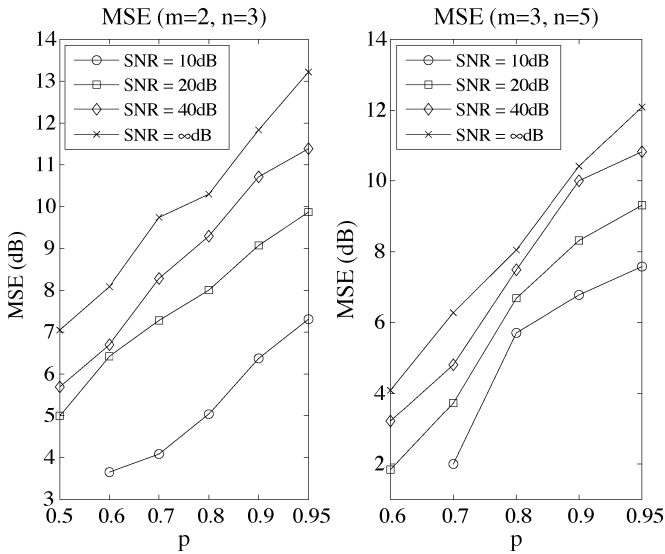


Fig. 3. MSE values for varying sparsity and SNR levels, for the 2-measurement ( $m = 2$ , left) and 3-measurement case ( $m = 3$ , right).

The MLPs are initialized to get a linear input-output transformation:  $\mathbf{d}^i(t) = \mathbf{g}(\mathbf{c}^i(t)) = \mathbf{c}^i(t)$ . This initialization is relatively “near” the optimal solution and prevents the weights from converging to a trivial (all zeroes) solution. The parameters of the  $m$  MLPs are adapted in each iteration using a batch gradient descent approach to minimize (6). And, as suggested in [13], we also assume that they pass through the origin, i.e.,  $g_j(0) = 0$ ; therefore, the bias of the output layer  $b_{j,2}$  is fixed as  $b_{j,2} = -\mathbf{w}_{j,2}^T \phi(\mathbf{b}_{j,1})$ . After this training the mixing matrix can be estimated in a straightforward way relying on the estimated  $K_{j,k}^i$ .

V. SIMULATION RESULTS

Monte Carlo simulations were performed for signals with different sparsity and SNR levels. The source signals were generated according

to (3) with a normal distribution  $f_{S_1}(s_i)$  with zero mean and variance 10. For each sparsity and SNR level, 20 different mixing matrices were generated randomly by choosing the amplitudes of the basis vectors uniformly from  $[0.1, 1]$  and the angles uniformly from  $[-\pi, \pi]$  with a minimum angle of  $\pi/10$  between every pair of basis vectors to avoid cluster overlapping. The number of samples in each case was  $2500/(1 - \nu_1 - \nu_2)$  in order to restrict the clustering to 2500 samples. After mixing by  $\mathbf{A}$ , the mixtures were transformed by the nonlinear functions  $f_j(x) = \tanh(x)$ . Finally Gaussian white noise was added to reach the specified SNR level.

A 2-measurement scenario with three mixtures ( $m = 3, n = 2$ ) as well as a 3-measurement scenario with 5 mixtures ( $m = 5, n = 3$ ) were simulated. Fine-tuning spectral clustering was applied and  $m$  MLPs with  $r = 15$  hidden neurons were trained to estimate the two inverse nonlinearities, with a learning rate of  $\mu = 0.01$  and a maximum of 1000 epochs. An illustration of the different steps of the algorithm is shown in Fig. 2.

After training, the basis vectors were estimated from  $K_{j,k}^i$  and the source signals were estimated applying the shortest-path algorithm from [7]. The results are shown in Fig. 3. Since no measures were taken to reduce the sensor noise, the obtained mean square errors (MSEs) are highly dependent on the SNR level. Although in most cases the inverse nonlinearity estimation can “linearize” the clusters sufficiently well [see, for instance, Fig. 2(d)] only a modest MSE value was obtained even for the noiseless case (SNR =  $\infty$  dB). This is due to the strong nonlinearity used and to the fact that the MLPs only represent the inverse nonlinear functions well for input points that are in the training range. Points that are outside of it, such as the “nonsparse” samples, are estimated with greater error and therefore represent the main contribution in the MSE.

VI. CONCLUSION

We presented an algorithm to invert the nonlinearities in the problem of PNL underdetermined BSS of sparse sources. The algorithm consists of two steps: first, a spectral clustering algorithm is applied to identify the active sources and second, a set of MLPs are trained to identify the inverse nonlinearity. After these two steps, the outputs of the MLPs

provide a “linearized” underdetermined BSS problem, which can easily be solved.

The presented method requires sparse sources and invertible nonlinearities that are linear for small input values. Simulation results were included for 2-measurement and 3-measurement cases, and as long as the contributions of the different sources do not overlap in the mixtures, there is no restriction on the number of sources or mixtures.

#### REFERENCES

- [1] J. Cardoso, “Blind signal separation: statistical principles,” in *Proc. IEEE. Special Issue on Blind Identification and Estimation*, vol. 9, Oct. 1998, pp. 2009–2025.
- [2] P. Comon, C. Jutten, and J. Herault, “Blind separation of sources, part II: problem statement,” *Signal Process.*, vol. 24, pp. 11–21, 1991.
- [3] P. Comon, “Independent component analysis—a new concept?,” *Signal Process.*, vol. 36, pp. 287–314, 1994.
- [4] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*. New York: Wiley Interscience, 2001.
- [5] M. Joho, H. Mathis, and R. Lambert, “Overdetermined blind source separation: using more sensors than source signals in a noisy mixture,” in *Proc. 2nd Int. Conf. on Independent Component Analysis and Blind Signal Separation*, Helsinki, Finland, Jun. 2000, pp. 81–86.
- [6] T. W. Lee, M. S. Lewicki, M. Girolami, and T. J. Sejnowski, “Blind source separation of more sources than mixtures using overcomplete representations,” *IEEE Signal Process. Lett.*, vol. 6, pp. 87–90, 1999.
- [7] P. Bofill and M. Zibulevsky, “Underdetermined blind source separation using sparse representations,” *Signal Process.*, vol. 81, no. 11, pp. 2353–2362, 2001.
- [8] D. Luengo, I. Santamaría, and L. Vielva, “A general solution to blind inverse problems for sparse input signals: deconvolution, equalization and source separation,” *Neurocomputing*, vol. 69, pp. 198–215, 2005.
- [9] M. Solazzi, R. Parisi, and A. Uncini, “Blind source separation in nonlinear mixtures by adaptive spline neural networks,” in *Proc. 3rd Int. Conf. on Independent Component Analysis and Blind Signal Separation*, San Diego, CA, Dec. 2001, pp. 254–259.
- [10] Y. Tan and J. Wang, “Nonlinear blind source separation using higher order statistics and a genetic algorithm,” *IEEE Trans. Evol. Comput.*, vol. 5, no. 6, pp. 600–612, Dec. 2001.
- [11] M. Babaie-Zadeh, C. Jutten, and K. Nayebi, “A geometric approach for separating post nonlinear mixtures,” in *Proc. XI Europ. Signal Processing Conf.*, vol. II, Toulouse, France, Sep. 2002, pp. 11–14.
- [12] A. Taleb and C. Jutten, “Source separation in post-nonlinear mixtures,” *IEEE Trans. Signal Process.*, vol. 47, no. 10, pp. 2807–2820, Oct. 1999.
- [13] F. J. Theis and S. Amari, “Postnonlinear overcomplete blind source separation using sparse sources,” in *Proc. 5th Int. Conf. Independent Component Analysis and Blind Signal Separation*, vol. 3195, Lecture Notes in Computer Science, Granada, Spain, Sep. 2004, pp. 718–725.
- [14] L. Vielva, I. Santamaría, C. Pantaleón, J. Ibáñez, D. Erdogmus, and J. C. Principe, “Estimation of the mixing matrix for underdetermined blind source separation using spectral techniques,” in *Proc. XI Europ. Signal Processing Conf.*, vol. 1, Toulouse, France, Sep. 2002, pp. 557–560.
- [15] D. Erdogmus, L. Vielva, and J. C. Principe, “Nonparametric estimation and tracking of the mixing matrix for underdetermined blind source separation,” in *Proc. 3rd Int. Conf. on Independent Component Analysis and Blind Signal Separation*, San Diego, CA, Dec. 2001, pp. 189–194.
- [16] F. J. Theis, E. W. Lang, and C. G. Puntonet, “A geometric algorithm for overcomplete linear ICA,” *Neurocomputing*, vol. 56, pp. 381–396, 2004.
- [17] A. Y. Ng, M. I. Jordan, and Y. Weiss, “On spectral clustering: analysis and an algorithm,” in *Advances in Neural Information Processing Systems 14*, T. G. Dietterich, S. Becker, and Z. Ghahramani, Eds. Cambridge, MA: MIT Press, 2002, pp. 849–856.
- [18] L. Zelnik-Manor and P. Perona, “Self-tuning spectral clustering,” in *Advances in Neural Information Processing Systems 17*, L. K. Saul, Y. Weiss, and L. Bottou, Eds. Cambridge, MA: MIT Press, 2005, pp. 1601–1608.

## Stability Analysis for Stochastic Cohen–Grossberg Neural Networks With Mixed Time Delays

Zidong Wang, Yurong Liu, Maozhen Li, and Xiaohui Liu

**Abstract**—In this letter, the global asymptotic stability analysis problem is considered for a class of stochastic Cohen–Grossberg neural networks with mixed time delays, which consist of both the discrete and distributed time delays. Based on an Lyapunov–Krasovskii functional and the stochastic stability analysis theory, a linear matrix inequality (LMI) approach is developed to derive several sufficient conditions guaranteeing the global asymptotic convergence of the equilibrium point in the mean square. It is shown that the addressed stochastic Cohen–Grossberg neural networks with mixed delays are globally asymptotically stable in the mean square if two LMIs are feasible, where the feasibility of LMIs can be readily checked by the Matlab LMI toolbox. It is also pointed out that the main results comprise some existing results as special cases. A numerical example is given to demonstrate the usefulness of the proposed global stability criteria.

**Index Terms**—Cohen–Grossberg neural networks, discrete delays, distributed delays, global asymptotic stability, linear matrix inequality (LMI), Lyapunov–Krasovskii functional, stochastic systems.

#### I. INTRODUCTION

The past few decades have witnessed tremendous developments in the research field of neural networks. Various neural networks, such as Hopfield neural networks, cellular neural networks, bidirectional associative neural networks and Cohen–Grossberg neural networks, have been widely investigated and successfully applied in many areas. Among them, the renowned Cohen–Grossberg neural network [7] has recently gained particular research attention, since it is quite general to include several well-known neural networks as its special cases, and it has promising application potentials for tasks of classification, associative memory, parallel computation and nonlinear optimization problems; see [16] and [26] for a survey.

On the other hand, time delays are unavoidably encountered in the implementation of neural networks, and may cause undesirable dynamic network behaviors such as oscillation and instability. For example, delay occurs due to the finite speeds of the switching and transmission of signals in a network. This leads to the delayed neural networks that were first explicitly introduced in [17]. Since then, the delayed neural networks have been widely studied. Recently, there has been an increasing research interest on the stability analysis problems for delayed Cohen–Grossberg neural networks, and many results have been reported in the literature. Various sufficient conditions, either delay-dependent or delay-independent, have been proposed to guarantee the asymptotic, exponential, or absolute stability for Cohen–Grossberg neural networks; see [2], [5], [6], [13], [14], and [18] for some recent results concerning *discrete* time-delays.

Manuscript received September 1, 2005; revised November 20, 2005. This work was supported in part by the Engineering and Physical Sciences Research Council (EPSRC) of the U.K. under Grant GR/S27658/01, by the Nuffield Foundation of the U.K. under Grant NAL/00630/G, and by the Alexander von Humboldt Foundation of Germany.

Z. Wang and X. Liu are with the Department of Information Systems and Computing, Brunel University, Uxbridge, Middlesex, UB8 3PH, U.K. (e-mail: Zidong.Wang@brunel.ac.uk).

Y. Liu is with the Department of Mathematics, Yangzhou University, Yangzhou 225002, P. R. China.

M. Li is with the Department of Electronic and Computer Engineering, Brunel University, Uxbridge, Middlesex, UB8 3PH, U.K.

Digital Object Identifier 10.1109/TNN.2006.872355