

Batch and Online underdetermined source separation using Laplacian Mixture Models

Nikolaos Mitianoudis, *Member IEEE*, and Tania Stathaki,

Abstract—In this paper, we explore the problem of sound source separation and identification from a two-sensor instantaneous mixture. The estimation of the mixing and the sources is performed using *Laplacian Mixture Models (LMM)*. The proposed algorithm fits the model using batch processing of the observed data and performs separation using either a hard or a soft decision scheme. An extension of the algorithm to online source separation, where the samples are arriving in a real-time fashion, is also presented. The proposed scheme exhibits good performance as far as separation quality and convergence speed are concerned. The online version also demonstrates several promising source separation possibilities in the case of nonstationary mixing.

Index Terms—Overcomplete source separation, Laplacian Mixture Models, Expectation Maximisation (EM), online processing.

I. INTRODUCTION

LET a set of M sensors $\underline{x}(n) = [x_1(n), \dots, x_M(n)]^T$ observe a set of N sound sources $\underline{s}(n) = [s_1(n), \dots, s_N(n)]^T$. We will consider the case of instantaneous mixing, i.e. each sensor captures a scaled version of each signal with no delay in transmission. Moreover, the possible additive noise will be considered negligible. The above instantaneous mixing model can be expressed in mathematical terms, as follows:

$$\underline{x}(n) = A\underline{s}(n) \quad (1)$$

where A represents the *mixing matrix* and n the sample index. The blind source separation problem provides an estimate of the source signals \underline{s} given the sensor signals \underline{x} . Usually, most separation approaches are *semi-blind*, which implies that some general assumptions about the statistical structure of the source signals are usually made. A number of algorithms have been proposed to solve the overdetermined and complete source separation problem ($M \geq N$) with great success. The additional assumption of statistical independence between the sources led to a whole group of source separation algorithms, summarised under the general term *Independent Component Analysis (ICA)*. Starting from different interpretations of statistical independence, most algorithms perform source separation with great accuracy. An overview of current ICA and general blind source separation algorithms can be found in tutorial books on ICA by Hyvärinen et al. [1] and Cichocki-Amari [2].

The underdetermined source separation problem ($M \leq N$) is more challenging, since in this case, the estimation of the mixing matrix A is not sufficient for the estimation of the source signals \underline{s} . Assuming Gaussian distributions for the sources, one could estimate the sources using the pseudo-inverse of matrix A [3]. As the source signals are assumed to be nonGaussian in our case, the above linear operation is not sufficient to estimate the sources. Therefore, the underdetermined source separation problem can be divided into two sub-problems: i) estimating the mixing matrix A and ii) estimating the source signals \underline{s} .

The existence of a unique solution for the overcomplete source separation problem is always under question, since it is an ill-conditioned problem that has an infinite number of solutions. Any linear system with less equations than unknowns has infinite solutions [4]. However, according to Eriksson and Koivunen [5], the linear generative model of (1) can have a *unique* and *identifiable* solution for the overcomplete case, as long i) there are no Gaussian sources present in the mixture, ii) the mixing matrix A is of full column rank, i.e. $\text{rank}(A) = M$ and iii) none of the source variables has a characteristic function featuring a component in the form $\exp(Q(u))$, where $Q(u)$ is a polynomial of a degree of at least two. This implies that this intractable problem may have a non-infinite number of solutions, under several constraints and probabilistic criteria for the sources.

One probabilistic profile that satisfies the assumptions set above are *sparse* distributions. *Sparsity* is mainly used to describe signals that are mostly close to zero with the exception of several large values. Common models that can be used for approximating sparsity are minimum \mathcal{L}_0 or \mathcal{L}_1 norms [3], *Mixture of Gaussians (MoG)* [6], [7], [8] or *factorable Laplacian distributions* [9]. The separation quality for the underdetermined case seems to improve with sparsity, as usually the performance of source separation algorithms is closely connected with the nonGaussianity of the source signals [10]. However, in a lot of practical applications, the source data are not sparse. For example, some musical instrument signals tend to be less sparse than speech signals in the time-domain. Speech contains a lot of silent segments that guarantees sparsity, however, this might not be the case with many instrument signals. Therefore, the assumed sparse models are not accurate enough to describe the statistical properties of the signals in the time-domain. However, many natural signals can have sparser representations in other transform domains. Many transforms have sparsifying properties, such as the Fourier transform, the Wavelet transform and the Modified Discrete Cosine Transform (MDCT). As long as the transformation is linear, it is straightforward to estimate the generative model

Manuscript received September 6, 2006; revised December 20, 2006.

The authors are with the Department of Electrical and Electronic Engineering, Imperial College London, Exhibition Road, SW7 2AZ London, UK (e-mail: n.mitianoudis@imperial.ac.uk, tel: +44 (0)207 594 6229, fax: +44 (0)207 594 6234).

and the sources in the transform domain. There are also alternative methods, where one can generate sparse representations for a specific dataset [11]. In the following analysis, MDCT is employed to provide sparser representations.

The overcomplete source separation problem has been covered extensively in the literature. Lewicki [3] provided a complete Bayesian approach, assuming Laplacian source priors to estimate both the mixing matrix and the sources in the time domain. In [12], Lee et al. applied the previous algorithm to the source separation problem. Girolami [9] employed the factorable Laplacian distribution and variational EM to estimate the mixing matrix and the sources. More complete sparse source models, such as the *Student-t* distribution, were employed by Févotte et al. [13]. The parameters of the model, the mixing matrix and the source signals were estimated using either *Markov Chains Monte Carlo* (MCMC) simulations [13] or a *Variational Expectation Maximisation* (EM) algorithm [14], featuring robust performance, however, computationally expensive. Vincent and Rodet [15] performed underdetermined source separation exploiting both the spatial diversity of the sources in the mixing matrix and more structured source priors in the form of trained *Hidden Markov Models* (HMM). Clustering solutions were introduced by Hyvärinen [16] and Zibulevsky et al. [17], also featuring good results and lower computational complexity. In this case, the mixing matrix and the source signals are estimated by performing clustering in a sparser representation of the signals in the transform domain. Bofill-Zibulevsky [18] presented a shortest path algorithm based on \mathcal{L}_1 minimisation that could estimate the mixing matrix and the sources. O’Grady and Pearlmutter [19] proposed an algorithm to perform separation via Oriented Lines Separation (LOST) using clustering along lines (Hard-Lost) in a similar manner to Hyvärinen [16]. In addition, they proposed a soft-thresholding technique using an EM on a mixture of oriented lines to assign points to more than one source [20]. Davies and Mitianoudis [6] employed two-state Gaussian Mixture Models (GMM) to model the source densities in a sparse representation and also the possible additive noise. An *Expectation Maximization* (EM)-type algorithm was used to estimate the parameters of the two-state models and perform the clustering of the coefficients. The latter approach can be considered a joint Bayesian and clustering approach. A two-sensor more-sources setup, modelling also some delays between the sensors, was addressed using the DUET algorithm [21] that can separate the sources, by calculating amplitude differences (AD) and phase differences (PD) between the sensors. An online version of the algorithm was also proposed [22].

In this paper, the authors explore the overcomplete source separation case of a two-sensor setup with no additive noise. Exploiting the desired spatial diversity of the source signals, the source separation problem can become a one-dimensional (1D) optimal detection problem. Therefore, the phase difference between the two-sensor data, i.e. the direction of arrival of sources, is mainly employed as the separation criterion. A *Laplacian Mixture Model* (LMM) is fitted to the phase difference between the two sensors, using an EM-type algorithm. The LMM model is then used to perform separation

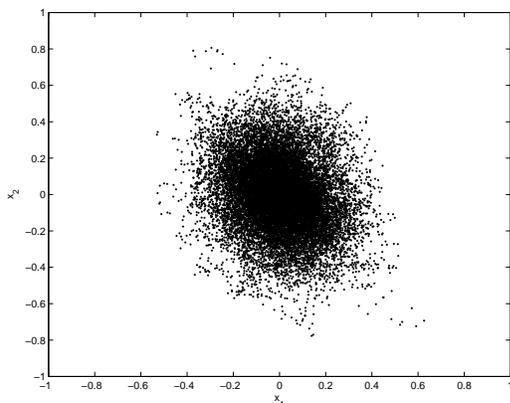
using either a *soft* or a *hard threshold*. A preliminary form of this model was presented briefly in [23], however, the authors have extended the proposed idea with more detailed analysis and experiments. In addition, the proposed batch algorithm is extended for online processing via a stochastic update of the model’s parameters. The online system is also benchmarked in the case of slowly-varying mixing, i.e. nonstationary mixing.

There are several potential applications of the two-sensor instantaneous setup. The combination of several instruments into a stereo mixture in a recording studio follows the instantaneous mixing model of (1). Therefore, the proposed approach can be used to decompose a studio recording into the separate instruments that exist in the mixture for many possible applications, such as music transcription, object-based audio coding, audio remixing and other Music Information Retrieval (MIR) tasks [24]. In addition, the mixing of the instrument in a recording usually changes over time to emphasize the song’s structure and progress, such as moving sources spatially and changing the volume of several instruments for a soloing spot.

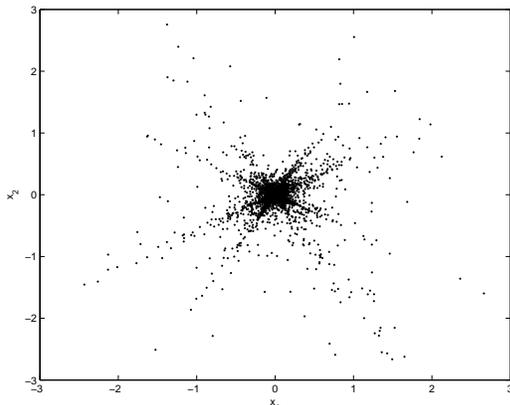
The paper is structured as follows: in Section II, we introduce the two-sensor setup that is used in our analysis. In Section III, we derive the training algorithm for the Laplacian Mixture Models, both for batch and online operation. In Section IV, the method to perform source separation using LMM is presented in detail. In Section V, some results on batch and real-time source separation are presented, using the proposed schemes. The case of nonstationary mixing is also examined. Finally, in Section VI we outline the main contributions of this paper.

II. A TWO-SENSOR APPROACH

In this paper, we will assume a two sensor instantaneous mixing approach. In Figure 1(a), one can see the scatter plot of the two sensor signals, in the case of two sensors and four sources. The four sources are 7 sec of speech, accordion, piano and violin signals. In the time-domain representation, no directions of the input sources are visible in the mixture. Consequently, the separation problem seems very difficult to solve. To get a sparser representation of the data, the *Modified Discrete Cosine Transform* (MDCT) or the *Short-Time Fourier Transform* (STFT) can be applied on the observed signals. The MDCT is a linear, real transform that has excellent sparsifying properties for most audio and speech signals. The harmonic content of most speech and musical instrument signals can be represented by harmonically related sinusoidals with great accuracy (excluding transient and percussive parts in audio and unvoiced segments in speech). Consequently, using a transformation that projects the audio data into sinusoidal bases will most probably result into a more compact and sparse representation of the original data. The MDCT is more preferable than the STFT, since it is real and retains all the required sinusoidal signal structure. The need for sparser representations in overcomplete source separation is discussed more rigorously in [6]. When the sources are sparse, smaller coefficients are more probable, whereas all the signal’s energy is concentrated in few large values. Therefore, the density of the data in the mixture space shows a tendency to cluster along



(a) Time domain



(b) MDCT domain

Fig. 1. Scatter plot of a two-sensor four-sources mixture in the time domain and in the sparse MDCT domain. The almost Gaussian-like structure of the time-domain representation is enhanced using the MDCT and the four sources can be clearly identified in the mixture.

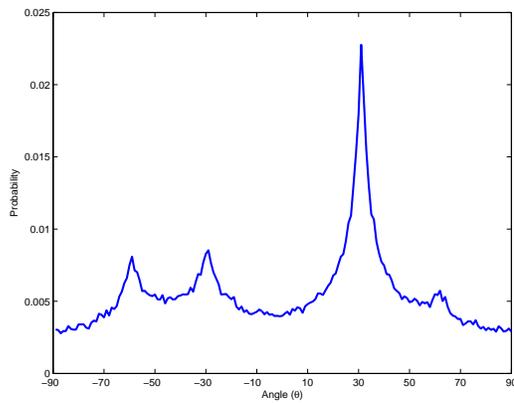
the directions of the mixing matrix columns [17]. Observing the scatter plot in Figure 1(b), it is clear that the phase difference between the two sensors can be used to identify and separate the sources in the mixture. That is to say, the two-dimensional (2D) problem can be transformed to a one-dimensional (1D) problem, as the main important parameter is the angle θ_n of each point.

$$\theta_n = \text{atan} \frac{x_2(n)}{x_1(n)} \quad (2)$$

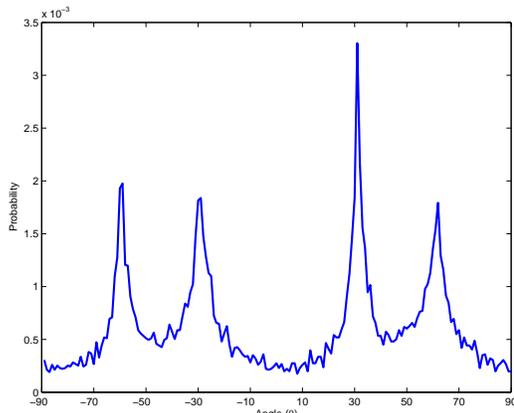
Using the phase difference information between the two sensors is equivalent to mapping all the observed data points on the unit-circle. This is equivalent to the concept of mapping all the observed data points to the half-unit N -dimensional sphere, as proposed by Zibulevsky et al. [17]. In Figure 2(a), we plot the histogram of the observed data angle θ_n in the previous example. The strong superGaussian characteristics of the individual components in the MDCT domain are preserved in the angle representation θ_n . We can also define the amplitude r_n of each point $\underline{x}(n)$, as follows:

$$r_n = \sqrt{x_1(n)^2 + x_2(n)^2} \quad (3)$$

As the points that are close to the origin have more Gaussian-like structure and do not contribute to superGaussianity, we



(a) Original histogram



(b) Modified histogram

Fig. 2. Histograms of angle θ_n in the four sources example of Fig. 1. The four sources are identifiable in the original histogram (a), however, keeping only the most superGaussian components (b), we can facilitate the separation process, as the directions of arrival are more clearly identifiable.

can use a “reduced” representation of the original data in order to estimate the columns of the mixing matrix more accurately. In Figure 2(b), we can see a histogram of those points n , whose amplitude r_n is above a threshold, e.g. $r_n > 0.1$. The four components are more clearly identifiable in this reduced representation, which can facilitate the estimation of the columns of the mixing matrix, i.e. the directions of arrival for each source.

One can model the observed density $p(\theta_n)$ of the complete data set, by fitting a *Laplacian Mixture Model* (LMM). Each of the Laplacians in the mixture will represent each individual source. Using the estimated Laplacians, source separation can be performed by optimal detection schemes.

III. LAPLACIAN MIXTURE MODELLING

There are a number of probabilistic models that can be used to represent sparsity. One model, that is commonly used in the literature, is the *Laplacian* density function. The definition for the Laplacian probability density function (pdf) is given by the following expression:

$$\mathcal{L}(\theta, c, m) = ce^{-2c|\theta-m|} \quad (4)$$

where m defines the mean and $c > 0$ controls the “width” (approximate standard deviation) of the distribution. In a

similar fashion to Gaussian Mixture Models (GMM), one can employ *Laplacian Mixture Models* (LMM) in order to model a mixture of “heavy-tailed signals”. A *Laplacian Mixture Model* can thus be defined, as follows:

$$p(\theta) = \sum_{i=1}^N \alpha_i \mathcal{L}(\theta, c_i, m_i) = \sum_{i=1}^N \alpha_i c_i e^{-2c_i|\theta - m_i|} \quad (5)$$

where α_i , m_i , c_i represent the weight, mean and width of each Laplacian respectively and all weights should sum up to one, i.e. $\sum_{i=1}^N \alpha_i = 1$. A common method that can be employed to train a mixture model is the *Expectation-Maximization* (EM) algorithm. A complete derivation of an EM algorithm was presented by Dempster et al. [25] and has been employed to fit a GMM on a training data set [26]. Similarly, the EM can be employed to train a LMM using an EM algorithm over a training set (batch-EM) or even adapt the parameters of a LMM in a real-time fashion, as each data point is accessed by the algorithm (Online-LMM).

A. Batch-EM

In this section, we derive the EM algorithm to train a Laplacian Mixture Model, based on Bilmes’s analysis [26] for the estimation of a GMM using the EM. Bilmes estimates Maximum Likelihood mixture density parameters using the EM [26]. Assuming K training samples for θ_n and Laplacian Mixture densities (5), the log-likelihood of these training samples takes the following form:

$$I(\alpha_i, c_i, m_i) = \sum_{n=1}^K \log \sum_{i=1}^N \alpha_i \mathcal{L}(\theta_n, c_i, m_i) \quad (6)$$

Introducing unobserved data items that can identify the components that “generated” each data item, we can simplify the log-likelihood of (6) for Laplacian Mixtures, as follows:

$$J(\alpha_i, c_i, m_i) = \sum_{n=1}^K \sum_{i=1}^N (\log \alpha_i + \log c_i - 2c_i|\theta_n - m_i|) p(i|\theta_n) \quad (7)$$

where $p(i|\theta_n)$ represents the probability of sample θ_n belonging to the i^{th} Laplacian of the LMM. In a similar fashion to Gaussian Mixture Models, the updates for $p(i|\theta_n)$ and α_i can be given by the following equations:

$$p(i|\theta_n) = \frac{\alpha_i \mathcal{L}(\theta_n, m_i, c_i)}{\sum_{i=1}^N \alpha_i \mathcal{L}(\theta_n, m_i, c_i)} \quad (8)$$

$$\alpha_i^+ \leftarrow \frac{1}{K} \sum_{n=1}^K p(i|\theta_n) \quad (9)$$

The updates for m_i and c_i are estimated by setting $\partial J(\alpha_i, c_i, m_i)/\partial m_i = 0$ and $\partial J(\alpha_i, c_i, m_i)/\partial c_i = 0$ respectively. Following some derivation (see Appendix I), we get the following update rules:

$$m_i^+ \leftarrow \frac{\sum_{n=1}^K \frac{\theta_n}{|\theta_n - m_i|} p(i|\theta_n)}{\sum_{n=1}^K \frac{1}{|\theta_n - m_i|} p(i|\theta_n)} \quad (10)$$

$$c_i^+ \leftarrow \frac{\sum_{n=1}^K p(i|\theta_n)}{2 \sum_{n=1}^K |\theta_n - m_i| p(i|\theta_n)} \quad (11)$$

The four update rules are iterated until convergence. As observed in the previous section, removing the data points that are placed close to the origin and do not necessarily carry enough information about the sources’ angles, the Laplacian structure of the sources can be emphasised (see Figure 1(b)). Enhancing the sparsity in the angle representation will increase the convergence speed of the EM and will provide more accurate estimates for the sources’ angles. Therefore, we train the LMM with a subset consisting of those data points n that satisfy $r(n) = \sqrt{x_1(n)^2 + x_2(n)^2} > B$, where B is a threshold. Once convergence is achieved, an adequate accuracy is obtained for the means m_i . However, to get more accurate estimates for α_i and c_i , we need to run the EM for the whole dataset. During the second stage of the adaptation, the estimated m_i are not updated.

Since the EM algorithm performs Maximum Likelihood estimation, the LMM will be fitted around the median value of each cluster but not the mean value. Thus, m_i will represent the median but not the mean of each cluster. However, since audio data are nearly symmetrical in the transform domain, the mean and the median should be relatively close and thus this should not affect the source separation quality.

B. Online-LMM

A simple method for the stochastic estimation of the parameters of a GMM is outlined by Bishop [27]. In this section, this concept is extended for the online training of a LMM.

In a real-time scenario, the cost function that is maximised, is the same as in the offline case, but in this case the data points are arriving at the processor one-by-one. Therefore, one can look for a sequential update scheme for the parameters, so that the new values are estimated based only on the previous values of the parameters and the current sample. The notation m_i^K implies that the current value of m_i was estimated using K data points. The objective will be to derive an expression, where the model’s parameters for the $K + 1$ data point are calculated based on the parameters estimated for the previous K data points and the current data point θ_{K+1} .

Given the already-trained model using the K -th sample, the probability of the $K + 1$ -th sample belonging to the i -th Laplacian is similarly:

$$p(i|\theta_{K+1}) = \frac{\alpha_i^K c_i^K e^{-2c_i^K |\theta_{K+1} - m_i^K|}}{\sum_{i=1}^N \alpha_i^K c_i^K e^{-2c_i^K |\theta_{K+1} - m_i^K|}} \quad (12)$$

In this online scheme, a “hard-thresholding” scheme will be used only, since the “soft-thresholding” technique will simply confuse the algorithm. Consequently, Eq. (12) can be used for the classification of each incoming point to the Laplacian that corresponds to the greatest $p(i|\theta_{K+1})$. To update the weight of each Laplacian, we just have to count the number of data points that have been attributed to each Laplacian so far. Normalising to the total number of points will give the actual values for α_i^{K+1} .

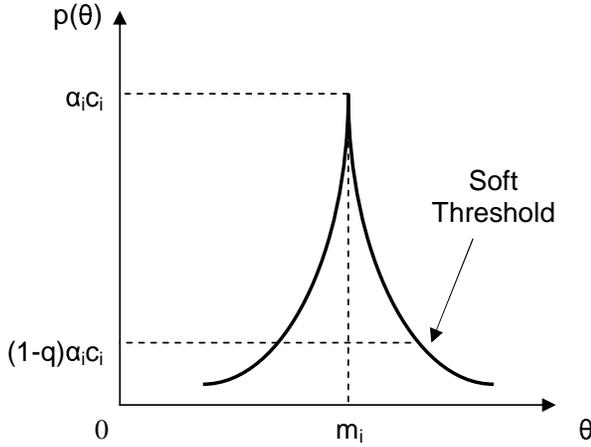


Fig. 3. In the soft thresholding scheme, the points that satisfy $p(\theta) \geq (1-q)\alpha_i c_i$ are used to reconstruct the i -th source.

To find the updates for the mean m_i^{K+1} and the width c_i^{K+1} , one can derive an expression that involves the current sample θ_{K+1} and the previous parameters m_i^K and the width c_i^K (see Appendix II). The following online update rules are obtained:

$$m_i^{K+1} = m_i^K + \frac{1}{\sum_{n=1}^{K+1} \frac{p(i|\theta_n)}{|\theta_n - m_i^K|}} \text{sgn}(\theta_{K+1} - m_i^K) \quad (13)$$

$$v_i^{K+1} = v_i^K + \frac{1}{K+1} (2|\theta_{K+1} - m_i^K| - v_i^K) \quad (14)$$

where $v_i = 1/c_i$. Unfortunately, in the estimation of m_i^{K+1} , it is not possible to derive an expression depending on the model's parameters and the current sample only. However, the expression inside the summation needs to be calculated each time a new sample arrives and add it to a variable that stores the summation in memory. This variable needs to be reset to zero, occasionally, as it will be discussed further on.

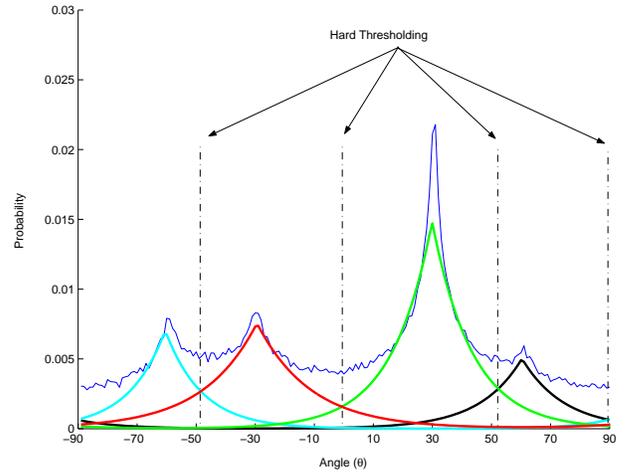
IV. OVERCOMPLETE SOURCE SEPARATION

Once the LMM is trained, optimal detection theory and the estimated individual Laplacians can be applied on the estimates of the sources. The centre of each Laplacian m_i should represent a column of the mixing matrix A in the form of $[\cos(m_i) \sin(m_i)]^T$. Each Laplacian should model the statistics of each source in the transform domain. Thus, using either a *hard* or a *soft decision threshold*, we can perform overcomplete source separation.

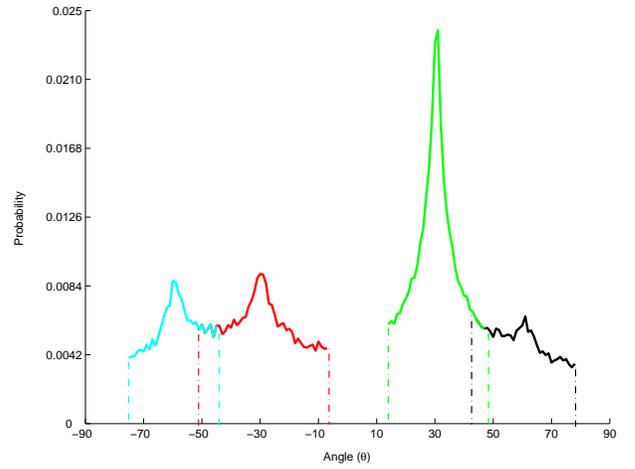
A. Hard threshold - "Winner takes all"

The "Winner takes all" strategy attributes each point of the scatter plot of Fig. 1(b) to only one of the sources. This is performed by setting a hard threshold at the intersections between the trained Laplacians. Consequently, the source separation problem becomes an *optimal decision* problem. The decision thresholds θ_{ij}^{opt} between the i -th and the j -th neighbouring Laplacians are given by the following equation:

$$m_{ij}^{opt} = \frac{\ln \frac{\alpha_i c_i}{\alpha_j c_j} + 2(c_i m_i + c_j m_j)}{2(c_i + c_j)} \quad (15)$$



(a) Hard Thresholding



(b) Soft Thresholding

Fig. 4. A two-sensors four-sources scenario, separated using the batch-EM and LMMs. In (a), the four trained Laplacians are depicted along with the actual density function and the imposed hard thresholds. Applying soft thresholds, the classification shown in (b) is achieved, which allows some overlapping between adjacent sources.

Using these thresholds, the algorithm can attribute the points with $m_{ij}^{opt} < \theta_n < m_{jk}^{opt}$ to source j , where i, j, k are neighbouring Laplacians (sources). Figure 4(a) depicts the fitted Laplacians using the batch EM-LMM algorithm, in a two sensors - four audio sources (voice, piano, accordion and violin) example and the hard thresholds imposed using the above equation. The points that belong to each of the four clusters, shown in Fig. 4(a), are attributed and are used to reconstruct each source.

B. Soft threshold

Observing the histograms of Fig. 2, we can attribute points that are distant from the centre of the 2D representation to each source with great confidence. In contrast, points that are closer to the centre ($r_n \rightarrow 0$) can not be attributed to any of the sources with great confidence. These points may belong to more than one source. One can then relax the hard threshold strategy, by allowing points belonging to more than one source simultaneously. A "soft-thresholding" strategy can

attribute points that constitute a chosen ratio q (i.e. $0.7 - 0.9$) of the density of each Laplacian to the corresponding source (see Fig. 3). Hence, the i^{th} source can be associated with those points θ_n , for which $p(\theta_n) \geq (1-q)\alpha_i c_i$, where $p(\theta_n)$ is given by (5). A large value for q allows more points to belong to more than one Laplacian. A small value for q imposes stricter criteria for the points to belong to a Laplacian and essentially becomes a hard thresholding approach. This scheme can be effective, only if the estimated Laplacians are concentrated around m_i . In the opposite case, there will be components that will dominate the pdf and therefore be attributed with more points than it should and therefore they would contain contamination from other sources. In Fig. 4(b), we can see the four sources, as classified by the soft thresholding strategy. The different colours represent different clusters, i.e. different sources. We can see that several points are attributed to both the first and the second sources and both the third and fourth sources by the soft classification scheme.

C. Initialisation issues

There is an initialisation issue for all clustering algorithms. First of all, the number of clusters (sources) N needs to be known a priori. Depending on the initialisation of the cluster centres m_i , a clustering algorithm may create different clusters, which may not be the optimal or the desired ones. In other words, two different Laplacians may converge to represent the same cluster, which is not the desired objective. Hence, we must ensure that our LMM-EM will converge to the clusters (sources) centered around the actual Direction Of Arrival (DOA) to the sensor array. One can use a *K-Means* [27] initialisation step to get an initial guess for the Laplacian centres m_i .

In this problem, the solution space is well-known. We already assume that the overcomplete mixture should be separable and identifiable in order to perform separation. Therefore, the sources should be positioned in fairly distinct and well separated angles. Hence, it seems reasonable to initialise m_i in equal intervals in the space $(-90^\circ, 90^\circ)$. This initialisation will spread the centres uniformly all over the solution space and thus is more probable to cover most source positions in the mixture.

D. Edge effects

The proposed approach is likely to be affected by some “edge effects” in several cases. The Laplacian density, as described in (4), is valid $\forall \theta \in (-\infty, +\infty)$. However, the range of θ_n is not only bounded to the $(-90^\circ, 90^\circ)$ interval and the two boundaries are actually connected, due to the $\text{atan}(\cdot)$ function. Assume that you have a concentration of points close to one of the boundaries. The EM algorithm will attempt to fit a Laplacian around this cluster, however, assuming a linear support on θ . As a result, the algorithm can not attribute points that belong to the same cluster, but are on the other end of θ , due to the assumed linear support. Therefore, the algorithm will impose some bias in the estimation of m_i that are close to the edge boundaries.

Assuming that the Laplacians, describing the sources, are quite concentrated around m_i , we can propose the following strategy to alleviate this problem. The algorithm is initialised as previously. In each update, we check whether any of the centres are closer to any of the boundaries (less than 20° distance from $\pm 90^\circ$). If this occurs, we estimate the distances between the estimated centres m_i . The next step is to rotate all the data points and the estimated centres m_i , so that the affected boundary (-90° or 90°) is mapped to the middle of the centres m_i that feature the greatest distance. In other words, all the data points and centres are rotated so that the middle between the centres that are most apart is mapped to either -90° or 90° . Finally, we map the parts that exist in $(-270^\circ, -90^\circ)$ or $(90^\circ, 270^\circ)$ to $(-90^\circ, 90^\circ)$, in order to form the actual clusters.

To address this in a more eloquent manner, one could introduce mixtures of Laplacian featuring a modulo- π wrapping, as this is the periodicity of the $\text{atan}(\cdot)$ function. Although, this solution seems very rational, we found it did not produce satisfactory results in practice. Perhaps, the use of *circular statistics* and *distributions* [28] might provide a more complete solution to the problem.

E. Online source separation

There are certain issues concerning the online version of the EM, compared to its batch version. First of all, it is rational to let each sample update only the parameters of the Laplacian it is attributed to by (12). Letting each sample update all the parameters of the model is conceptually inappropriate and is also found to cause instability in practice.

In the batch version of the algorithm, the update rule is presented with the complete dataset, which implies that the algorithm will be presented with samples from all the sources. In a real-time scenario, the algorithm may temporarily deal with statistics that correspond to some of the sources present but not all of them. In this case, the Laplacians corresponding to these sources may be overtrained and dominate the whole density function, whilst others may disappear. Therefore, some restrictions need to be imposed to guarantee that no Laplacian disappears from the LMM and also no Laplacian dominates the pdf. To achieve this, one may impose some upper and lower bounds on the “width” of each Laplacian. The lower bound c_{min} ensures that no Laplacian disappears from the mixture and the upper bound c_{max} ensures that each source has limited spatial range and does not dominate the whole pdf. Therefore, after each update we must ensure that c_i^{K+1} remains between these limits:

$$c_{min} \leq c_i^{K+1} \leq c_{max} \quad (16)$$

The update term for the width c_i in (14) includes a scaling term $1/(K+1)$. Since K is constantly increasing, the influence of the update term vanishes quickly. This might accelerate the algorithm’s convergence, however, it will not be able to track any probably changes or any sources that have not appeared so far after the first couple of samples. Consequently, it might be beneficial to replace $1/(K+1)$ with a term, such as $1/(\text{mod}(K, T_1)+1)$. In this sense, the training procedure of these parameters is reinitialised after a number of samples defined

by T_1 . The same procedure is applied to the denominator that scales the update in (13), as it is set to $p(i|\theta_n)/|\theta_{K+1} - m_i^K|$ after T_2 samples. This step will also enable the algorithm to tackle non-stationary mixing, as it will be shown in the following section.

V. EXPERIMENTS

In this section, we evaluate the algorithms proposed in the previous sections. It is not our purpose to perform a complete comparison with all state-of-the-art algorithms in the field. We will use Hyvärinen’s clustering approach [16] and O’Grady and Pearlmutter’s [19], [20] Soft and Hard LOST algorithm’s (available online from [29]) to demonstrate several trends using artificial mixtures or publicly available datasets. All these algorithms belong to the same category of BSS methods and address the problem modelled by (1) In order to quantify the performance of the algorithms, we are using the BSS_EVAL Toolbox, as available by Févotte et al. [30]. In this toolbox, the authors decompose the separated signal $u_j(n)$ into the following components for the noiseless case:

$$u_j(n) = s_i(n) + e_{interf}(n) + e_{artif}(n) \quad (17)$$

where u_j is the algorithm’s estimation of source s_i , e_{interf} is an allowed deformation of the sources which accounts for the interferences of the unwanted sources and e_{artif} is an “artifact” term that may correspond to artifacts or deformations induced by the separation algorithm [30]. The *Signal-to-Distortion Ratio* (SDR) can be evaluated, as follows:

$$SDR_{dB} = 10 \log \frac{\|s_i\|^2}{\|e_{interf} + e_{artif}\|^2} \quad (18)$$

where $\|\cdot\|$ is the \mathcal{L}_2 -norm. The results are also compared with the average SDR of the mixtures for each source, in order to demonstrate the achieved improvement. In [30], two other distortion measurements are proposed, which are not evaluated, due to lack of space.

The experimental section is divided into a section testing the batch-processing version of the LMM-EM algorithm and two other sections testing the online version of the algorithm for stationary and non-stationary mixing¹.

A. Batch processing

For the first part of the experiment, a number of synthetic instantaneous mixtures examples are created, using the following form of stationary mixing matrix:

$$A = \begin{bmatrix} \cos(\psi_1) & \cos(\psi_2) & \dots & \cos(\psi_N) \\ \sin(\psi_1) & \sin(\psi_2) & \dots & \sin(\psi_N) \end{bmatrix} \quad (19)$$

The frame length for the MDCT analysis is set to $64msec$ for the test signals sampled at $16KHz$ and to $46.4msec$ for those at $44.1KHz$. We initialise the parameters of the Batch-LMM, as follows: $\alpha_i = 1/N$ and $c_i = 0.001$. Starting the width of the Laplacians from a small value seemed to be a stable choice for the majority of the experiments. The centres m_i are

either randomly initialised in the parameter space $(-90^\circ, 90^\circ)$ or initialised using a *K-means* step. The initialisation of m_i is important, as if we choose two initial values for m_i that are really close, then it is very probable that the individual Laplacians may not converge to different clusters. To provide a more accurate estimation of m_i , training is initially performed using a “reduced” dataset, containing all points that satisfy $r_n > 0.2$, provided that the input signals are scaled to $[-1, 1]$. The second phase is to use the “complete” dataset to update the values for α_i and c_i .

1) *Five uncorrelated audio recordings*: In this experiment, we use 5 solo audio uncorrelated recordings (a saxophone, an accordion, an acoustic guitar, a violin and a female voice) of sampling frequency $16KHz$ and duration $8.19ms$. We examine the following two cases of source positioning in order to evaluate the performance of the algorithm: in the first one the sources are placed quite distinctly and in the middle of the observed space $(-90^\circ, 90^\circ)$, whereas in the second case one source is placed quite at the left far end (-80°) and three of them are placed rather closely.

	ψ_1	ψ_2	ψ_3	ψ_4	ψ_5
Case 1	-60	-30	0	30	60
Case 2	-80	-40	-20	10	60

The algorithm is initialised as previously mentioned, with $q = 0.7$ for the LMM-EM_Soft algorithm. In Fig. 5, we observe the convergence of the training parameters m_i, c_i and α_i initially for the “reduced” and then for the “complete” set for the mixing in Case 1. In Fig. 5 (a), the difference in convergence speed using K-Means initialisation (dashed lines) and a successful random initialisation (continuous lines) is depicted. As previously, mentioned a random initialisation that features distinct distributed initial centres is bound to converge to the desired optimum of the likelihood, however, a random initialisation with very localised centres may lead to local optimum of the likelihood but not the desired one. Although, the original K-means is not immune to initialisation, it served as a valid starting point for the LMM in all our experiments. Therefore, it has been used consequently employed for initialisation in the following runs of the Batch-LMM.

In Fig. 6, we depict the five fitted Laplacians and the two separation schemes using “hard” and “soft” thresholding for Case 2, that feature closed spaced sources and sources that might be affected by the “Edge Effects”. We can see that the algorithm managed to estimate the sources without bias, due to the edge effects or the close distance of the three sources.

In Table I, we compare the performance of the two batch schemes with Hyvärinen’s clustering approach and soft_LOST and hard_LOST in terms of *SDR* [30] for these two cases. Hyvärinen’s approach is also prone to initialisation, however, the results in Table I are acquired using the best run of the algorithm. To appreciate the results of this rather difficult separation problem, as the achieved SDRs for all methods seem rather low, we estimate the average SDR of the two input signals, treating them as estimates of each input signal. Then, the improvement performed by the method can be understood more clearly. It seems that the proposed algorithms perform similarly to Hyvärinen’s approach. This is fairly logical as

¹All the experimental audio results, described in the following sections, are available online at:

<http://www.commsp.ee.ic.ac.uk/~nikolao/lmm.htm>

the LMM-EM_Hard and Hyvärinen’s algorithm attempt to partition the solution space into the desired sources. All three approaches outperformed both versions of the LOST algorithm. In general, the Soft_Lost algorithm managed to separate the sources in most cases, however, there were some audible artifacts and clicks that reduced the calculated quality measure. The difference between the soft and the hard thresholding schemes of the LMM approach is not really clear in the SDR measurements, however, it is audible. The soft thresholding scheme allows more audible interference from neighbouring sources, since it attributes points with small r_n to multiple sources. The hard thresholding scheme does not allow much interference from neighbouring sources, however, more artifacts and “pipe” noise may be audible in the separated sources.

TABLE I

SEPARATION RESULTS IN TERMS OF SDR (DB) FOR THE TWO CASES OF FIVE UNCORRELATED SOURCES MIXTURE. THE PROPOSED LMM APPROACHES (LMM-EM_SOFT AND LMM-EM_HARD) ARE COMPARED WITH HYVÄRINEN’S, SOFT_LOST, HARD_LOST AND THE AVERAGE SDR OF THE MIXTURES.

Case 1	s_1	s_2	s_3	s_4	s_5
Mixed Signals	-6.00	-13.37	-26.26	-6.67	-6.81
LMM-EM_soft	6.37	-1.29	7.60	3.59	5.52
LMM-EM_hard	6.69	0.32	7.66	3.65	5.55
Hyvärinen	6.53	-1.16	7.60	4.14	5.79
hard_LOST	-2.87	-	0.07	-3.25	0.34
soft_LOST	4.58	-4.01	5.09	1.67	3.93
Case 2	s_1	s_2	s_3	s_4	s_5
Mixed Signals	-10.30	-13.09	-4.14	-10.85	-7.06
LMM-EM_soft	4.89	-2.40	6.97	4.037	3.62
LMM-EM_hard	6.10	-1.61	6.67	4.40	6.03
Hyvärinen	6.02	-1.72	7.25	4.87	5.86
hard_LOST	-0.60	-	2.66	-0.78	0.94
soft_LOST	3.77	-3.40	6.84	3.11	2.56

2) *Four correlated audio recordings*: In this section, we tested the two batch LMM approaches and the other three approaches with some challenging source separation datasets, available by the Blind Source Separation Database (BASS-DB) [31]: the *Latino 1*, *Latino 2* and *Groove* datasets, sampled at 44.1KHz. In [31], we can also find the source signals that will be used to quantify the separation results. The two “Latino” mixtures contain actually five sources, i.e. the two drums sources come in fact from the same source placed in two different positions. As their linear combination is seen by the algorithm as one source in a new position, we will regard them as a single source and therefore, we will be looking for 4 sources. Both “Latino” mixtures feature very closely spaced sources. The “Groove” dataset features four widely spaced sources: bass (far left), distortion guitar (center left), clean guitar (center right) and drums (far right).

The LMM-EM approaches were initialised as previously, however, setting a $q = 0.6$ for the LMM-EM_Soft approach. The results for all five approaches are depicted in Table II. Due to the K-means initialisation, the LMM-EM approaches

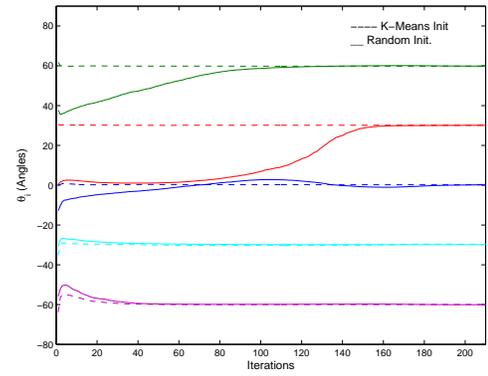
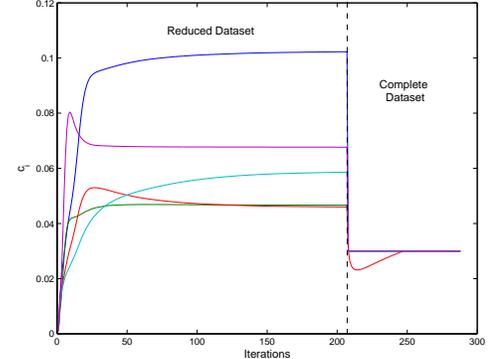
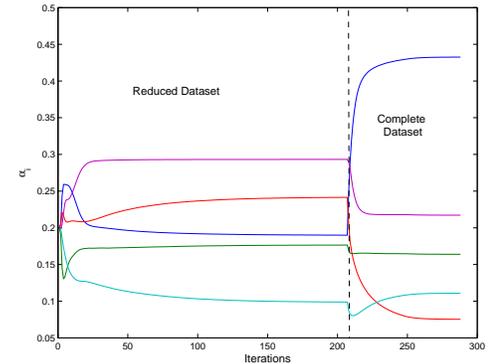
(a) Convergence of m_i (b) Convergence of c_i (c) Convergence of α_i

Fig. 5. Convergence of the parameters of the LMM in the case of five uncorrelated audio sources and the first mixing scenario. In (a), the difference in convergence speed using K-Means initialisation (dashed lines) and random initialisation (continuous lines) is shown.

are essentially producing similar results in each run, which again is not the case for Hyvärinen’s algorithm and we picked the best run for the evaluation. Both Soft and Hard LOST approaches produced similar results in each run. We can notice that the LMM-EM approaches managed to perform reasonable separation, despite the small spacing of the sources or the source being placed at the edges of the solution space. This implies that the shifting solution proposed to tackle the “edge effects” seems to be able to resolve these shortcomings. The proposed approaches managed to perform reasonably well, outperforming the Hyvärinen’s approach in the “Groove” dataset. Unfortunately, the LOST algorithm did not manage to perform reasonably good separation and in some cases the

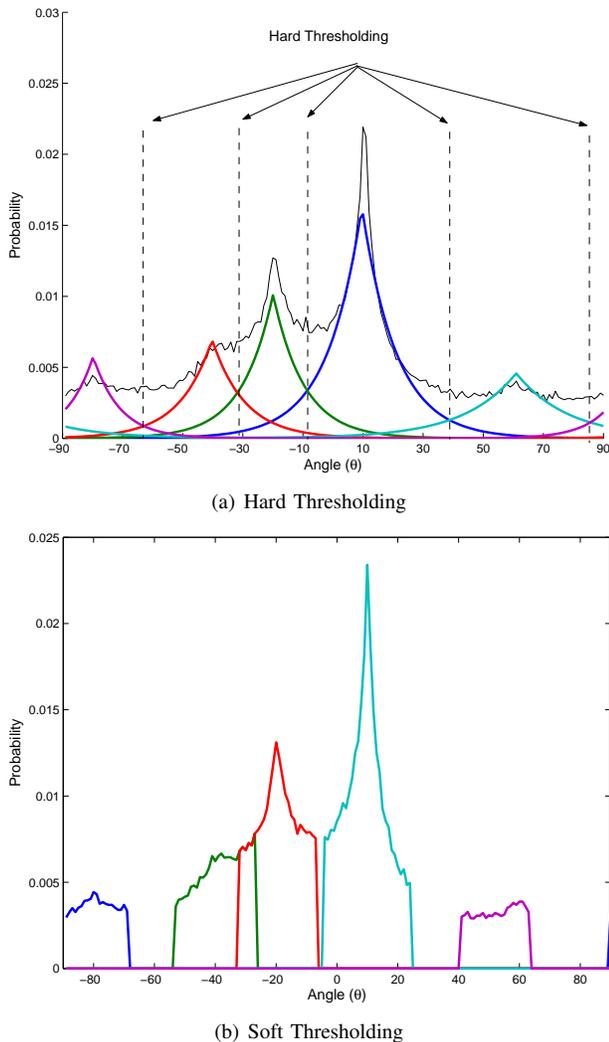


Fig. 6. Clustering results in the case of five uncorrelated audio sources and the second mixing scenario, using hard thresholding (a) and soft thresholding (b) for $q = 0.4$.

Hard_LOST did not manage to separate one of the sources (marked with “-”) or appears as a mixture with another source.

As far as computational complexity is concerned, the proposed Batch-LMM algorithms seemed to be able to converge in about 3 – 4 minutes for 8 secs of the input audio tested on Pentium 4 2.4 MHz PC with 1 GB RAM using a non-optimised implementation on MATLAB, which seems quite reasonable for these kind of problems.

B. Online processing with stationary mixing

In the online version of the algorithm, some extra assumptions are necessary. First of all, we need to assume that the data are arriving in a real-time fashion in the transform domain, i.e. the MDCT domain. As the MDCT is actually a frame-based transformation, we assume there is a pre-processing step that estimates the transformation of each frame and provides the data in a real-time fashion to the processor. A similar post-processing step is used to estimate the inverse transformation. In addition, as the input data arrive in a real-time basis, there might not be equal contribution from all the sources. In other

TABLE II
SEPARATION RESULTS IN TERMS OF SDR (DB) FOR THE LATINO1, LATINO2 AND GROOVE DATASETS [31]. THE PROPOSED LMM APPROACHES (LMM-EM_SOFT AND LMM-EM_HARD) ARE COMPARED WITH HYVÄRINEN’S, SOFT_LOST AND HARD_LOST.

Latino 1	s_1	s_2	s_3	s_4
Mixed Signals	-3.84	-5.93	-3.27	-10.05
LMM-EM_soft	7.32	0.55	7.70	2.59
LMM-EM_hard	8.95	3.89	6.74	5.25
Hyvärinen	8.57	3.06	7.25	4.69
hard_LOST	5.22	-	1.26	-3.19
soft_LOST	10.16	0.91	3.68	1.74
Latino 2	s_1	s_2	s_3	s_4
Mixed Signals	0.06	-4.62	-16.89	-11.28
LMM-EM_soft	6.55	1.54	-6.34	1.51
LMM-EM_hard	7.05	1.89	-9.32	2.59
Hyvärinen	7.86	1.90	-9.98	1.39
hard_LOST	-0.61	-7.23	-	-0.27
soft_LOST	9.96	1.52	-	4.41
Groove	s_1	s_2	s_3	s_4
Mixed Signals	-30.02	-10.25	-6.14	-21.24
LMM-EM_soft	4.21	-4.38	-1.99	-5.13
LMM-EM_hard	2.85	-4.47	-0.86	3.28
Hyvärinen	3.79	-3.72	-1.13	1.49
hard_LOST	2.20	-	-1.53	4.07
soft_LOST	4.54	-5.77	-1.74	3.62

words, at several time intervals we may encounter samples from several but not all the sources. Therefore, we have to ensure that none of the Laplacians disappears or dominates the whole density function. This may be achieved by limiting $0.01 \leq c_i \leq 1$ at every step of the iteration. In addition, in order to enhance the superGaussian nature of the signals in the transform domain, samples that are close to the origin in the scatter plot of x_1, x_2 , i.e. $r_n < 0.1$, are not processed (again input signals are scaled to $[-1, +1]$). For these samples, zeros are transmitted to all separated sources. This can also be employed as a denoising step to remove possible additive Gaussian noise of known variance that will determine a proper value for the threshold. This is similar to denoising approaches via sparse code shrinkage [32].

To compare the proposed Online-LMM, we adapted Hyvärinen’s algorithm to work in a non-overlapping frame-by-frame basis. To prevent different permutations at each frame, two constraints were imposed: a) the mixing matrix estimated at each frame is used as initialisation for the mixing matrix in the next frame, b) after convergence, the inner product of the columns of the estimated matrix with the previous matrix, in order to ensure the correct permutation is chosen.

We used three solo recordings from three instruments (piano, accordion, acoustic guitar) of sampling frequency 16KHz and duration 8.19ms and created two mixtures using the mixing matrix in (19) for $N = 3$: Case 1 for distant spaced sources and Case 2 for more closely spaced sources.

	ψ_1	ψ_2	ψ_3
Case 1	-60	10	60
Case 2	-30	10	40

The Online-LMM methodology, proposed in Section III-B, and the frame-adapted Hyvärinen’s algorithm are used to

tackle the separation problem of samples that arrive in a real-time fashion. To cater for sources that do not appear uniformly in all time slots, the estimation of c_i is restarted after 2000 samples and the estimation of m_i after 200 samples (sampling frequency $16KHz$). The frame size for Hyvärinen's algorithm was 500 samples. These values were found to yield promising results during the experiments and were used in both online experiments. The initialisation of the means seems quite important and the K-means can not be used in this case. However, using a initialisation that spans the solution space uniformly, such as $-45, 0$ and 45 , seems to tackle this indeterminacy in both cases. In Fig. 7, we can see the convergence of the mean of the three Laplacians over time for the two cases. The means seem to converge to the theoretical means, however, we can see small disturbances caused by the restarting of the mean, which is essential in order to avoid getting trapped in local minima, due to the local signal statistics. In the second case, we can see some perturbation between the separated sources, which is also audible, that might be due to the local statistics. The overall separation quality seems to be quite good, as depicted in Table II. The Hyvärinen's algorithm, although not really online, managed to perform reasonably well, however, there were some perturbations in the order of the separated sources that were not resolved by the two proposed steps. Consequently, its performance was low compared to the Online-LMM. An average for the three sources SDR measurement over time (blocks of 128 samples) for the two methods and the two cases is depicted in Fig. 8. We can see that both algorithms follow the same trends, apart from the parts that the Hyvärinen algorithm is on the wrong permutations and is not "really" online. The Online-LMM managed to retain the correct permutation of sources for the cases examined. However, if the distance between the sources becomes smaller, then we saw in experiments that the online-LMM may also feature source permutation changes along time.

C. Non-stationary mixing

In the third experiment, we applied the online source separation scheme in the case of moving sources, i.e. the mixing environment is non-stationary. Therefore, we created two different instantaneous mixtures of the three sources that were used in the previous experiment, using the mixing matrix in (19) for $N = 3$, where ψ_1, ψ_2, ψ_3 are now a function of time. In the first case, we simulated a slowly-varying parallel movement of the sources around their original position. The maximum drift from the original position to either side is 11.25 degrees. In this case, all sources move quite distantly from each other and in the same manner.

$$\begin{aligned}\psi_1(n) &= -\pi/2.5 + \pi \cos(2\pi n/164000)/16 \\ \psi_2(n) &= \pi/8 + \pi \cos(2\pi n/164000)/16 \\ \psi_3(n) &= \pi/3 + \pi \cos(2\pi n/164000)/16\end{aligned}\quad (20)$$

where n is the sample index. In the second case, we simulated a different movement, where the sources move independently and some of them may come closer to each other, but do

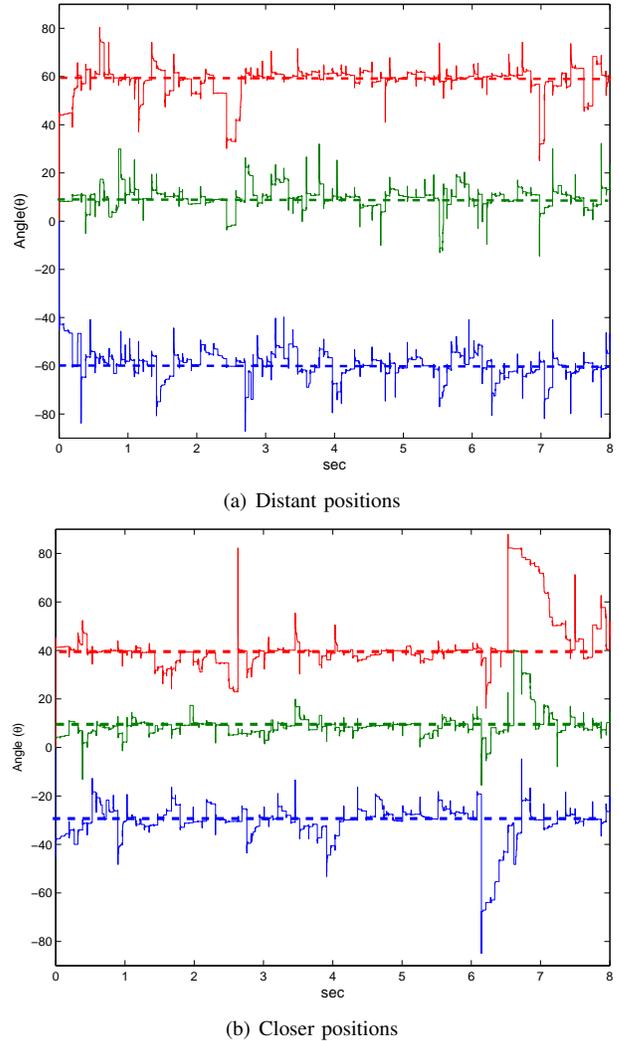


Fig. 7. Convergence of the mean m_i of the individual Laplacians for the Online-LMM, in the case of two stationary mixings, featuring distant source positions (a) and closer source positions (b). The dash lines show the real DOA of the sources and the continuous lines show the adaptation of the Laplacian means.

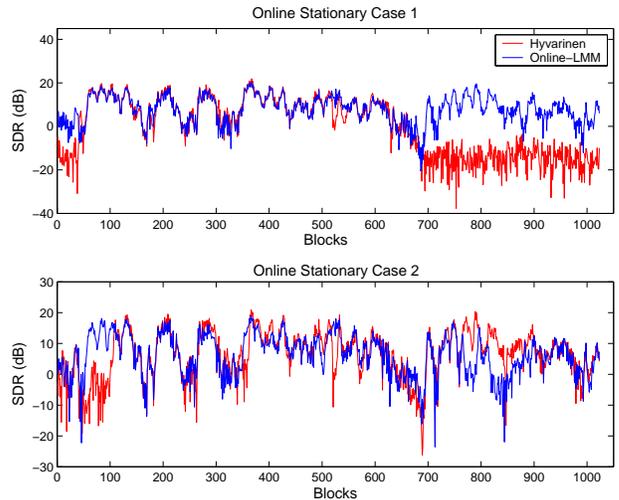


Fig. 8. Average SDR per blocks (frames of 8 ms) for the two stationary mixing cases. The online LMM (blue) is compared with the frame-adapted Hyvärinen's algorithm (red).

not crossover. The following paths were used to simulate the proposed movement.

$$\begin{aligned}\psi_1(n) &= -\pi/3 + \pi d_{st}(n)/16 \\ \psi_2(n) &= \pi/8 + \pi \cos(2\pi n/164000)/16 \\ \psi_3(n) &= \pi/3 + \pi \sin(2\pi n/30000)/16\end{aligned}\quad (21)$$

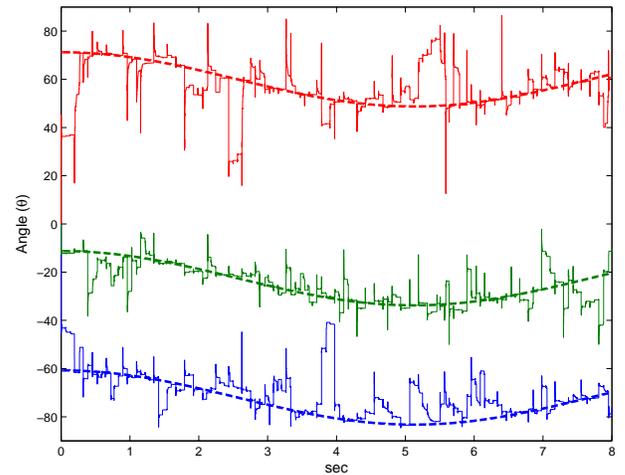
were $d_{st}(n)$ is one period of a saw-tooth signal with duration equal to the duration of the signal and unit amplitude.

The online-LMM scheme and the adapted Hyvärinen algorithm were employed to tackle these challenging source separation cases, using the same initialisation as in the previous example. In Fig. 9, we can see the tracking attempt of the LMM to capture the sources' movement in both cases, denoted with thick dash-dot lines. In Table III, the SDRs for the estimated sources using the online-LMM and the Hyvärinen algorithm is shown. An average for the three sources SDR measurement over time (blocks) for the two methods and the two cases is similarly depicted in Fig. 10. The LMM manages to track the motion of the sources with quite good accuracy. In general, we can say that the online-LMM managed to follow the movement of the three sources in both cases. We notice the perturbations that begin, when the direction of arrival for two of the sources get closer. The algorithm managed to retain a quite good quality of separation, however, this is not always the case. Generally, we can say that the algorithm is able to follow the movement of sources that remain quite distinct from other sources in the solution space and their movement does not crossover or gets closer to other sources. The Hyvärinen algorithm again was prone to permutation changes over time in both cases and therefore was not successful in separating the sources. This can also be verified by Fig. 10, where the Hyvärinen algorithm seems to be on wrong permutation for the majority of the time, especially in Case 2. In the other parts, the performance of the Online-LMM is similar to Hyvärinen's, which is important, as we are comparing an online method with a frame-based method. The SDRs of the complete waveforms in Table III also verify the above conclusions.

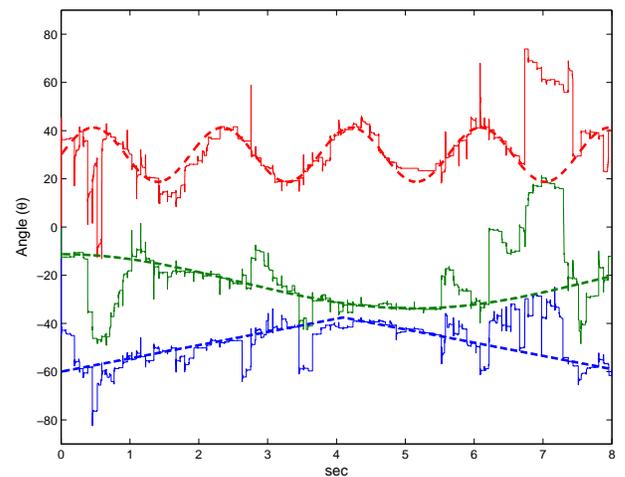
As far as computational complexity is concerned, the online version seems to be quite compact in terms of calculations, with the only exception for the update for the mean. The algorithm was tested again using a non-optimised implementation on MATLAB, on the previous machine and required about 7–10secs for the online experiments described, that featured segments of 8secs.

VI. CONCLUSION

In this paper, the use of Laplacian Mixture Model (LMM) for overcomplete source separation is introduced. The proposed method can be interpreted as either a Bayesian or a clustering approach to the separation case, where there are less observations than sources. The proposed approach was tested, using a number of synthetic and publicly proposed source separation datasets with promising results. Laplacian Mixture Modelling was adapted to estimate the LMM parameters in a real-time scenario, where the input data arrive in a sample-by-sample basis. The algorithm is suited for a real-time processing



(a) Sinusoidal movement



(b) Independent source movement

Fig. 9. Tracking of the sources in the case of non-stationary mixing. The dashed lines show that the actual movement of the sources, in terms of DOA, and the continuous lines show the real-time adaptation of the three Laplacian means m_i for two scenarios: one of similar sinusoidal movement (a) and one of independent source movement (b).

environment, as it estimates the model's parameters using only the current input sample, the previous model's parameters and a few stored variables. The proposed framework was tested in the case of both stationary and non-stationary mixing with promising results.

The experiments also demonstrated some of the limitations of the proposed approach and of underdetermined source separation in general. The algorithm, either in batch or online mode, is not able to separate very closely spaced sources. The required margin seems to be greater for online operation than in batch. This can be a limitation of overcomplete source separation that requires the columns of the mixing matrix to be different. In the opposite case, the sources that correspond to similar columns are considered as a single mixed source by these type of approaches.

In addition, the proposed approach is able to track the sources that are moving quite distinctively in the solution space, without interfering and crossing the movement of other sources in their path. Although, this is a significant

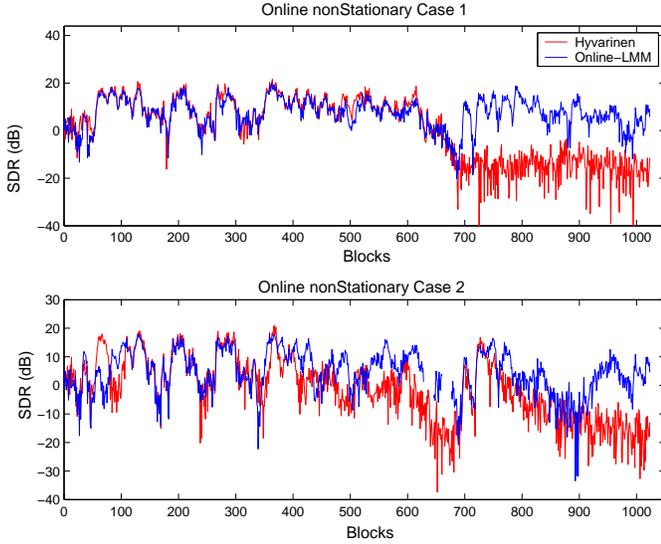


Fig. 10. Average SDR per blocks (frames of 8 ms) for the two non-stationary mixing cases. The online LMM (blue) is compared with the frame-adapted Hyvärinen's algorithm (red).

assumption, the proposed approach seems to be a promising step for tackling this challenging source separation case. This assumption may also be quite reasonable in the case of real-time separation of stereo recording where the sources usually do not move considerably in the spatial image.

One has to note that the proposed algorithms are sensitive to initialisation of the means, together with most basic clustering approaches. The proposed algorithm should always start from means that span the solution space uniformly. In the opposite case that two or more centres start from quite neighbouring points, it will be rather difficult for the algorithm to separate them. There is also a similar assumption that the actual source positions should be quite distinct for the separation problem to be identifiable. The initialisation using K-means for the batch version seemed to improve the stability of the algorithm.

Although the proposed solution seemed to be able to tackle possible “edge effects”, for future work, the authors would like to pursue the development of mixture models using circular distributions, such as the *von Mises* distribution [28]. In this case, the distribution is wrapped around $(-90^\circ, 90^\circ)$ and might give a more complete solution for sources that appear at the edges of the solution space.

APPENDIX I

DERIVATION OF THE BATCH-EM ALGORITHM

To estimate the updates for m_i and c_i for each Laplacian of the LMM model, we have to set the partial derivatives of the cost function $J(\alpha_i, c_i, m_i)$ to c_i and m_i (eq. (7)) equal to zero and solve for the two unknowns. The derivation follows:

- *Update for m_i*

The function $J(\alpha_i, c_i, m_i)$ is not differentiable at $m_i = \theta_n$, because of the term $|m_i - \theta_n|$. To calculate the derivative of this term, we can express it more analytically and then calculate

TABLE III

SDR (dB) FOR THE FOUR ONLINE STATIONARY AND NON-STATIONARY CASES. THE PROPOSED LMM APPROACH (LMM-EM_HARD) ARE COMPARED WITH HYVÄRINEN'S AND THE AVERAGE SDR OF THE MIXTURES.

Online Stat 1	s_1	s_2	s_3
Mixed Signals	-4.8411	-10.0617	-0.5054
LMM-EM_hard	9.5175	7.4534	10.9407
Hyvärinen	-2.2500	-3.9986	-1.7068
Online Stat 2	s_1	s_2	s_3
Mixed Signals	-4.4535	-10.3890	1.4630
LMM-EM_hard	8.6630	3.3047	9.0341
Hyvärinen	5.3501	0.5571	11.3578
Online NonStat 1	s_1	s_2	s_3
Mixed Signals	-6.8832	-5.7047	-0.4776
LMM-EM_hard	6.3652	8.1873	12.3729
Hyvärinen	-3.5182	-1.9740	-1.3002
Online NonStat 2	s_1	s_2	s_3
Mixed Signals	-3.8468	-6.4882	-0.4400
LMM-EM_hard	3.2055	0.5667	12.0853
Hyvärinen	-1.9140	-7.0309	3.9916

the derivative as follows:

$$G(m_i) = |m_i - \theta_n| = \begin{cases} m_i - \theta_n & , \text{if } m_i \geq \theta_n \\ -m_i + \theta_n & , \text{if } m_i < \theta_n \end{cases} \quad (22)$$

$$\frac{\partial G}{\partial m_i} = \begin{cases} 1 & , \text{if } m_i > \theta_n \\ -1 & , \text{if } m_i < \theta_n \end{cases} = \text{sgn}(m_i - \theta_n) \quad (23)$$

Therefore, we can find an analytic expression for the derivative of $G(m_i)$ that is not defined at $m_i = \theta_n$. In practice, this is not a serious problem and it is quite common in all probabilistic approaches that use Laplacian priors. One common solution is to threshold points $|\theta_n - m_i| < \delta$, where δ is a small constant (~ 0.001) to avoid possible ambiguities. Having clarified this point, the estimation of m_i can now be performed as follows:

$$\frac{\partial J}{\partial m_i} = 0 \quad (24)$$

$$\sum_{n=1}^K (-2c_i \frac{\partial}{\partial m_i} |\theta_n - m_i|) p(i|\theta_n) = 0 \quad (25)$$

$$\sum_{n=1}^K 2c_i \text{sgn}(\theta_n - m_i) p(i|\theta_n) = 0 \quad (26)$$

$$\sum_{n=1}^K \text{sgn}(\theta_n - m_i) p(i|\theta_n) = 0 \quad (27)$$

An exact solution for θ_n can not be easily traced, because of the non-separable $\text{sgn}(\cdot)$ function. However, we can estimate a solution for m_i , following a “fixed-point” optimisation approach, by reformulating (27) into the form $m_i = F(m_i)$.

$$\sum_{n=1}^K \text{sgn}(\theta_n - m_i) p(i|\theta_n) = \sum_{n=1}^K \frac{\theta_n - m_i}{|\theta_n - m_i|} p(i|\theta_n) = 0 \quad (28)$$

$$\sum_{n=1}^K \frac{\theta_n}{|\theta_n - m_i|} p(i|\theta_n) = \sum_{n=1}^K \frac{m_i}{|\theta_n - m_i|} p(i|\theta_n) \quad (29)$$

$$m_i = F(m_i) = \frac{\sum_{n=1}^K \frac{\theta_n}{|\theta_n - m_i|} p(i|\theta_n)}{\sum_{n=1}^K \frac{1}{|\theta_n - m_i|} p(i|\theta_n)} \quad (30)$$

Consequently, we can estimate m_i , by iterating $m_i^+ = F(m_i)$, i.e.

$$m_i^+ \leftarrow \frac{\sum_{n=1}^K \frac{\theta_n}{|\theta_n - m_i|} p(i|\theta_n)}{\sum_{n=1}^K \frac{1}{|\theta_n - m_i|} p(i|\theta_n)} \quad (31)$$

The derivation of the above update rule is similar to the derivation of the update of the means for the K-Harmonic Means (KHM) clustering method [33]. It is not straightforward, in either case, to provide a proof that the function $F(m_i)$ satisfies the Banach fixed-point theorem, however, it seems to converge in all our experiments.

- Update for c_i

$$\frac{\partial J}{\partial c_i} = \sum_{n=1}^K \left(\frac{1}{c_i} - 2|\theta_n - m_i| \right) p(i|\theta_n) = 0 \quad (32)$$

$$\frac{1}{c_i} \sum_{n=1}^K p(i|\theta_n) = \sum_{n=1}^K 2|\theta_n - m_i| p(i|\theta_n) \quad (33)$$

$$c_i^+ \leftarrow \frac{\sum_{n=1}^K p(i|\theta_n)}{2 \sum_{n=1}^K |\theta_n - m_i| p(i|\theta_n)} \quad (34)$$

APPENDIX II

DERIVATION OF THE ONLINE-LMM ALGORITHM

Following a similar technique for calculating stochastic updates for the parameters of a GMM [27], we can express the parameters m_i^{K+1} and c_i^{K+1} for the $K+1$ sample based on the $K+1$ sample θ_{k+1} and the previous parameters m_i^K and c_i^K . The derivations follow:

- Update for m_i^{K+1}

Starting from Eq. (31), the following online expression can be derived:

$$\begin{aligned} m_i^{K+1} &= \frac{\sum_{n=1}^{K+1} \frac{\theta_n}{|\theta_n - m_i^{K+1}|} p(i|\theta_n)}{\sum_{n=1}^{K+1} \frac{1}{|\theta_n - m_i^{K+1}|} p(i|\theta_n)} \quad (35) \\ &= \frac{\sum_{n=1}^K \frac{\theta_n}{|\theta_n - m_i^K|} p(i|\theta_n)}{\sum_{n=1}^{K+1} \frac{1}{|\theta_n - m_i^K|} p(i|\theta_n)} + \frac{\frac{\theta_{K+1}}{|\theta_{K+1} - m_i^K|} p(i|\theta_{K+1})}{\sum_{n=1}^{K+1} \frac{1}{|\theta_n - m_i^K|} p(i|\theta_n)} \\ &= \frac{\sum_{n=1}^K \frac{1}{|\theta_n - m_i^K|} p(i|\theta_n)}{\sum_{n=1}^{K+1} \frac{1}{|\theta_n - m_i^K|} p(i|\theta_n)} m_i^K + \frac{\frac{\theta_{K+1}}{|\theta_{K+1} - m_i^K|} p(i|\theta_{K+1})}{\sum_{n=1}^{K+1} \frac{1}{|\theta_n - m_i^K|} p(i|\theta_n)} \\ &= m_i^K - \frac{\frac{1}{|\theta_{K+1} - m_i^K|} p(i|\theta_{K+1})}{\sum_{n=1}^{K+1} \frac{1}{|\theta_n - m_i^K|} p(i|\theta_n)} m_i^K + \frac{\frac{\theta_{K+1}}{|\theta_{K+1} - m_i^K|} p(i|\theta_{K+1})}{\sum_{n=1}^{K+1} \frac{1}{|\theta_n - m_i^K|} p(i|\theta_n)} \\ &= m_i^K + \frac{\frac{p(i|\theta_{K+1})}{|\theta_{K+1} - m_i^K|}}{\sum_{n=1}^{K+1} \frac{1}{|\theta_n - m_i^K|} p(i|\theta_n)} (\theta_{K+1} - m_i^K) \end{aligned}$$

$$= m_i^K + \frac{p(i|\theta_{K+1})}{\sum_{n=1}^{K+1} \frac{1}{|\theta_n - m_i^K|} p(i|\theta_n)} \text{sgn}(\theta_{K+1} - m_i^K) \quad (36)$$

- Update for c_i^{K+1}

Let $v_i^{K+1} = 1/c_i^{K+1}$. Starting from eq. (34), the following online expression can be derived:

$$\begin{aligned} v_i^{K+1} &= \frac{2 \sum_{n=1}^{K+1} |\theta_n - m_i| p(i|\theta_n)}{\sum_{n=1}^{K+1} p(i|\theta_n)} \quad (37) \\ &= \frac{2 \sum_{n=1}^K |\theta_n - m_i| p(i|\theta_n)}{\sum_{n=1}^{K+1} p(i|\theta_n)} + \frac{2|\theta_{K+1} - m_i| p(i|\theta_{K+1})}{\sum_{n=1}^{K+1} p(i|\theta_n)} \\ &= \frac{\sum_{n=1}^K p(i|\theta_n)}{\sum_{n=1}^{K+1} p(i|\theta_n)} v_i^K + \frac{2|\theta_{K+1} - m_i| p(i|\theta_{K+1})}{\sum_{n=1}^{K+1} p(i|\theta_n)} \\ &= v_i^K - \frac{p(i|\theta_{K+1})}{\sum_{n=1}^{K+1} p(i|\theta_n)} v_i^K + \frac{2|\theta_{K+1} - m_i| p(i|\theta_{K+1})}{\sum_{n=1}^{K+1} p(i|\theta_n)} \\ &= v_i^K + \frac{p(i|\theta_{K+1})}{\sum_{n=1}^{K+1} p(i|\theta_n)} (2|\theta_{K+1} - m_i| - v_i^K) \quad (38) \end{aligned}$$

In the batch version, we usually set $\alpha_i^{K+1} = \frac{1}{K+1} \sum_{n=1}^{K+1} p(i|\theta_n)$. However, if we assume no knowledge of the past and use only the current sample θ_{K+1} , then conceptually $\alpha_i^{K+1} \approx p(i|\theta_{K+1})$. Consequently, the update for v_i^{K+1} can be estimated as follows:

$$v_i^{K+1} \approx v_i^K + \frac{1}{K+1} (2|\theta_{K+1} - m_i| - v_i^K) \quad (39)$$

ACKNOWLEDGMENT

The authors would like to thank Dr. Laurent Daudet for providing the code for the MDCT. The authors would like to thank the anonymous reviewers for their kind suggestions and corrections that helped to improve the quality of the paper.

REFERENCES

- [1] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*, John Wiley, New York, 2001, 481+xxii pages.
- [2] A. Cichocki and S.I. Amari, *Adaptive Blind Signal and Image Processing*, John Wiley and Sons, 2002.
- [3] M. Lewicki and T.J. Sejnowski, "Learning overcomplete representations," *Neural Computation*, vol. 12, pp. 337–365, 2000.
- [4] E. Kreyszig, *Advanced Engineering Mathematics*, Wiley, 1998.
- [5] J. Eriksson and V. Koivunen, "Identifiability and separability of linear ICA models revisited," in *Proc. Int. Workshop on Independent Component Analysis and Blind Signal Separation (ICA2003)*, Nara, Japan, 2003, pp. 23–27.
- [6] M. Davies and N. Mitianoudis, "A simple mixture model for sparse overcomplete ICA," *IEE proceedings in Vision, Image and Signal Processing*, vol. 151, no. 1, pp. 35–43, 2004.
- [7] E. Moulines, J.-F. Cardoso, and E. Gassiat, "Maximum likelihood for blind separation and deconvolution of noisy signals using mixture models," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP'97)*, Munich, Germany, 1997, pp. 3617–3620.
- [8] H. Attias, "Independent factor analysis," *Neural Computation*, vol. 11, no. 4, pp. 803–851, 1999.
- [9] M. Girolami, "A variational method for learning sparse and overcomplete representations," *Neural Computation*, vol. 13, no. 11, pp. 2517–2532, 2001.
- [10] J.-F. Cardoso, "Blind signal separation: statistical principles," *Proceedings of the IEEE*, vol. 9, no. 10, pp. 2009–2025, 1998.
- [11] M. Davies and L. Daudet, "Sparsifying subband decompositions," in *Proc. Int. IEEE WASPAA*, New Paltz, New York, 2003, pp. 107–110.

- [12] 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 Processing Letters*, vol. 4, no. 5, 1999.
- [13] C. Févotte, S.J. Godsill, and P.J. Wolfe, "Bayesian approach for blind separation of underdetermined mixtures of sparse sources," in *Proc. 5th International Conference on Independent Component Analysis and Blind Source Separation (ICA 2004)*, Granada, Spain, 2004, pp. 398–405.
- [14] A.T. Cemgil, C. Févotte, and S.J. Godsill, "Blind separation of sparse sources using variational EM," in *Proc. 13th European Signal Processing Conference (EUSIPCO05)*, Antakya, Turkey, 2005.
- [15] E. Vincent and X. Rodet, "Underdetermined source separation with structured source priors," in *Proc. 5th International Conference on Independent Component Analysis and Blind Source Separation (ICA 2004)*, Granada, Spain, 2004, pp. 1197–1204.
- [16] A. Hyvärinen, "Independent component analysis in the presence of gaussian noise by maximizing joint likelihood," *Neurocomputing*, vol. 22, pp. 49–67, 1998.
- [17] M. Zibulevsky, P. Kisilev, Y.Y. Zeevi, and B.A. Pearlmutter, "Blind source separation via multinode sparse representation," *Advances in Neural Information Processing Systems*, vol. 14, pp. 1049–1056, 2002.
- [18] P. Bofill and M. Zibulevsky, "Underdetermined blind source separation using sparse representations," *Signal Processing*, vol. 81, no. 11, pp. 2353–2362, 2001.
- [19] P.D. O'Grady and B.A. Pearlmutter, "Soft-LOST: EM on a mixture of oriented lines," in *Proc. International Conference on Independent Component Analysis 2004*, Granada, Spain, 2004, pp. 428–435.
- [20] P.D. O'Grady and B.A. Pearlmutter, "Hard-LOST: Modified K-Means for oriented lines," in *Proceedings of the Irish Signals and Systems Conference*, Ireland, 2004, pp. 247–252.
- [21] O. Yilmaz and S. Rickard, "Blind separation of speech mixtures via time-frequency masking," *IEEE Trans. Signal Processing*, vol. 52, no. 7, pp. 1830–1847, 2004.
- [22] S. Rickard, R. Balan, and J. Rosca, "Real-time time-frequency based blind source separation," in *Proc. ICA2003*, San Diego, CA, 2001, pp. 651–656.
- [23] N. Mitianoudis and T. Stathaki, "Overcomplete source separation using laplacian mixture models," *IEEE Signal Processing Letters*, vol. 18, no. 4, pp. 277–280, 2004.
- [24] S.J. Downie, "Music information retrieval (chapter 7)," *Annual Review of Information Science and Technology*, vol. 37, pp. 295–340, 2003.
- [25] A. P. Dempster, N. Laird, and D. Rubin, "Maximum likelihood for incomplete data via the EM algorithm," *J. of the Royal Statistical Society, ser. B*, vol. 39, pp. 1–38, 1977.
- [26] J.A. Bilmes, "A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian Mixture and Hidden Mixture Models," Tech. Rep., Department of Electrical Engineering and Computer Science, U.C. Berkeley, California, 1998.
- [27] C. M. Bishop, *Neural Networks for Pattern Recognition*, Clarendon Press, 1995.
- [28] N.I. Fisher, *Statistical Analysis of Circular Data*, Cambridge University Press, 1993.
- [29] P.D. O'Grady, "Code for soft and hard LOST," Available at <http://www.hamilton.ie/paul/>.
- [30] C. Févotte, R. Gribonval, and E. Vincent, "BSS EVAL Toolbox User Guide," Tech. Rep., IRISA Technical Report 1706, Rennes, France, April 2005, http://www.irisa.fr/metiss/bss_eval/.
- [31] E. Vincent, R. Gribonval, C. Févotte, A. Nesbit, M.D. Plumbley, M.E. Davies, and L. Daudet, "BASS-dB: the blind audio source separation evaluation database," Available at <http://bass-db.gforge.inria.fr/BASS-dB/>.
- [32] A. Hyvärinen, P. O. Hoyer, and E. Oja, "Image denoising by sparse code shrinkage," in *Intelligent Signal Processing*, S. Haykin and B. Kosko, Eds. IEEE Press, 2001.
- [33] B. Zhang, M. Hsu, and U. Dayal, "K-Harmonic Means: A spatial clustering algorithm with boosting," in *Proceedings of the First International Workshop on Temporal, Spatial, and Spatio-Temporal Data Mining*, Lyon, France, 2000, pp. 31–45.



image fusion and computer vision.

Nikolaos Mitianoudis received the diploma in Electronic and Computer Engineering from the Aristotle University of Thessaloniki, Greece in 1998. He received a MSc in Communications and Signal Processing from Imperial College London, UK in 2000 and the PhD in Audio Source Separation using Independent Component Analysis from Queen Mary, University of London, UK in 2004. Currently, he is working as a Research Associate at Imperial College London, UK. His research interests include Independent Component Analysis, audio signal processing,



Tania Stathaki was born in Athens, Hellas. In September 1991 she received the Masters degree in Electronics and Computer Engineering from the Department of Electrical and Computer Engineering of the National Technical University of Athens (NTUA) and the Advanced Diploma in Classical Piano Performance from the Orfeion Athens College of Music. She received the Ph.D. degree in Signal Processing from Imperial College in September 1994. She is currently a Senior Lecturer in the Department of Electrical and Electronic Engineering of Imperial College and the Image Processing Group leader of the same department. Previously, she was Lecturer in the Department of Information Systems and Computing of Brunel University in UK, Visiting Lecturer in the Electrical Engineering Department of Mahanakorn University in Thailand and Assistant Professor in the Department of Technology Education and Digital Systems of the University of Piraeus in Greece. Her current research interests lie in the areas of image processing, data fusion, non-linear signal processing, signal modelling and biomedical engineering. Dr. Stathaki is the author of 90 journal and conference papers.