

# A Dynamic Algorithm for Blind Separation of Convolutive Sound Mixtures

Jie Liu\*, Jack Xin\*, and Yingyong Qi †

## Abstract

We study an efficient dynamic blind source separation algorithm of convolutive sound mixtures based on updating statistical information in the frequency domain, and minimizing the support of time domain demixing filters by a weighted least square method. The permutation and scaling indeterminacies of separation, and concatenations of signals in adjacent time frames are resolved with optimization of  $l^1 \times l^\infty$  norm on cross-correlation coefficients at multiple time lags. The algorithm is a direct method without iterations, and is adaptive to the environment. Computations on recorded and synthetic mixtures of speech and music signals show excellent performance.

*Keywords:* Convolutive Mixtures, Indeterminacies, Dynamic Statistics Update, Optimization, Blind Separation.

---

\*Department of Mathematics, UC Irvine, Irvine, CA 92697, USA.

†Qualcomm Inc, 5775 Morehouse Drive, San Diego, CA 92121, USA.

# 1 Introduction

Blind source separation (BSS) methods aim to extract the original source signals from their mixtures based on the statistical independence of the source signals without knowledge of the mixing environment. The approach has been very successful for instantaneous mixtures. However, realistic sound signals are often mixed through a media channel, so the received sound mixtures are linear convolutions of the unknown sources and the channel transmission functions. In simple terms, the observed signals are unknown weighted sums of the signals and its delays. Separating convolutive mixtures is a challenging problem especially in realistic settings.

In this paper, we study a dynamic BSS method using both frequency and time domain information of sound signals in addition to the independence assumption on source signals. First, the convolutive mixture in the time domain is decomposed into instantaneous mixtures in the frequency domain by the fast Fourier transform (FFT). At each frequency, the joint approximate diagonalization of eigen-matrices (JADE) method is applied. The JADE method collects second and fourth order statistics from segments of sound signals to form a set of matrices for joint orthogonal diagonalization, which leads to an estimate of de-mixing matrix and independent sources. However, there remain extra degrees of freedom: permutation and scaling of estimated sources at each frequency. A proper choice of these parameters is critical for the separation quality. Moreover, the large number of samples of the statistical approach can cause delays in processing. These issues are to be addressed by utilizing dynamical information of signals in an optimization framework. We propose to dynamically update statistics with newly received signal frames, then use such statistics to determine permutation in the frequency domain by optimizing an  $l^1 \times l^\infty$  norm of channel to channel cross-correlation coefficients with multiple time lags. Though cross channel correlation functions and related similarity measure were proposed previously to fix permutation [13], they allow cancellations and may not measure similarity as accurately and reliably as the norm (metric) we introduced here. The freedom in scaling is fixed by minimizing the support of the estimated de-mixing matrix elements in the time domain. An efficient weighted least square method is formulated to achieve this purpose directly in contrast to iterative method in [17]. The resulting dynamic BSS algorithm is both direct and adapted to the acoustic environment. Encouraging results on satisfactory separation of recorded sound mixtures are reported.

The paper is organized as follows. In section 2, a review is presented on frequency domain approach, cumulants and joint diagonalization prob-

lems, and indeterminacies. Then the proposed dynamic method is presented, where objective functions of optimization, statistics update and efficient computations are addressed. Numerical results are shown and analyzed to demonstrate the capability of the algorithm to separate speech and music mixtures in both real room and synthetic environments. Conclusions are in section 3.

## 2 Convolutional Mixture and BSS

Let a real discrete time signal be  $s(k) = [s_1(k), s_2(k), \dots, s_n(k)]$ ,  $k$  a discrete time index, such that the components  $s_i(k)$  ( $i = 1, 2, \dots, n$ ), are zero-mean and mutually independent random processes. For simplicity, the processing will divide  $s$  into partially overlapping frames of length  $T$  each. The independent components are transmitted and mixed to give the observations  $x_i(k)$ :

$$x_i(k) = \sum_{j=1}^n \sum_{p=0}^{P-1} a_{ij}(p) s_j(k-p), \quad i = 1, 2, \dots, n; \quad (2.1)$$

where  $a_{ij}(p)$  denote mixing filter coefficients, the  $p$ -th element of the  $P$ -point impulse response from source  $i$  to receiver  $j$ . The mixture in (2.1) is convolutional, and an additive Gaussian noise may be added. The sound signals we are interested in are speech and music, both are non-Gaussian [1]. We shall consider the case of equal number of receivers and sources, especially  $n = 2$ .

An efficient way to decompose the nonlocal equation (2.1) into local ones is by a  $T$ -point discrete Fourier transform (DFT) [2],  $X_j(\omega, t) = \sum_{\tau=0}^{T-1} x_j(t+\tau) e^{-2\pi J\omega\tau}$ , where  $J = \sqrt{-1}$ ,  $\omega$  is a frequency index,  $\omega = 0, 1/T, \dots, (T-1)/T$ ,  $t$  the frame index. Suppose  $T > P$ , and extend  $a_{ij}(p)$  to all  $p \in [0, T-1]$  by zero padding. Let  $H_{ij}(\omega)$  denote the matrix function obtained by  $T$ -point DFT of  $a_{ij}(p)$  in  $p$ ,  $S_j(\omega, t)$  the  $T$ -point DFT of  $s_j(k)$  in the  $t$ -th frame. If  $P \ll T$ , then to a good approximation [17]:

$$X(\omega, t) \approx H(\omega) S(\omega, t), \quad (2.2)$$

where  $X = [X_1, \dots, X_n]^{Tr}$ ,  $S = [s_1, \dots, s_n]^{Tr}$ ,  $Tr$  is short for transpose. The components of  $S$  remains independent of each other, the problem is converted to a blind separation of instantaneous mixture in (2.2). Note that  $P$  is on the order of 40 to 50 typically, while  $T$  is 256 or 512, so the assumption  $P \ll T$  is reasonable.

## 2.1 Instantaneous Mixture and JADE

Let us briefly review an efficient and accurate method, so called joint approximate diagonalization eigen-matrices (JADE) [6] for BSS of instantaneous mixture. There are many other approaches in the literature [4], e.g. info-max method [1] which is iterative and based on maximizing some information theoretical function. JADE is essentially a direct method for reducing covariance. We shall think of  $S$  as a random function of  $t$ , and suppress  $\omega$  dependence. First assume that by proper scaling  $E[|S_j(t)|^2] = 1$ ,  $j = 1, \dots, n$ . It follows from independence of sources that ( $'$  conjugate transpose):

$$E[S(t)S(t)'] = I_n, \quad R_X \equiv E[X(t)X(t)'] = HH', \quad (2.3)$$

the latter identity is a factorization of the Hermitian covariance matrix of the mixture. However, there is non-uniqueness in the ordering and phases of columns of  $H$ . Suppose that (1) the mixing matrix  $H$  is full rank; (2) the  $S_j(t)$ 's are independent at any  $t$ ; (3) the process  $S(t)$  is stationary. Let  $W$  be a matrix such that  $I_n = WR_XW' = WHH'W'$ ,  $W$  is called a whitening matrix. Then  $WH$  is an orthogonal matrix, denoted by  $U$ . Multiplying  $W$  from the left onto (2.2), one finds that:

$$Z(t) \equiv WX(t) = US(t). \quad (2.4)$$

The 4th order statistics are needed to determine  $U$ . The 4th-order cumulant of four mean zero random variables is:

$$\text{Cum}[a, b, c, d] = E(abcd) - E(ab)E(cd) - E(ac)E(bd) - E(ad)E(bc), \quad (2.5)$$

which is zero if  $a, b, c, d$  split into two mutually independent groups. For source vector  $S$ ,  $\text{Cum}[S_i, S_j, S_k, S_l] = \text{kurt}_i \delta_{ijkl}$ ,  $\text{kurt}_i = \text{Cum}[S_i, S_i, S_i, S_i]$  is the kurtosis. If  $\text{kurt}_i \neq 0$ , the  $i$ -th source is called kurtic. Kurtosis is zero for a mean zero Gaussian random variable. The last assumption of JADE is that (4) there is at most one non-kurtic source.

Define cumulant matrix set  $Q_Z(M)$  from  $Z$  in (2.4) as the linear span of the Hermitian matrices  $Q = (q_{ij})$  satisfying (\* complex conjugate):

$$q_{ij} = \sum_{k,l=1}^n \text{Cum}(Z_i, Z_j^*, Z_k, Z_l^*) m_{lk}, \quad 1 \leq i, j \leq n, \quad (2.6)$$

where matrix  $M = (m_{ij}) = e_l e_k'$ ,  $e_l$  being the unit vector with zero components except the  $l$ -th component equal to one. Equations (2.4) and (2.6)

imply that ( $u_p$  is the  $p$ -th column of  $U$ ):

$$Q = \sum_{p=1}^n (\text{kurt}_p u_p' M u_p) u_p u_p', \quad \forall M, \quad (2.7)$$

or  $Q = UDU'$ ,  $D = \text{diag}(\text{kurt}_1 u_1' M u_1, \dots, \text{kurt}_n u_n' M u_n)$ . Hence,  $U$  is the joint diagonalizer of the matrix set  $Q_Z(M)$ . Once  $U$  is so determined, the mixing matrix  $H = W^{-1}U$ . It can be shown [6] using identity (2.7) that the joint diagonalizer of  $Q_Z(M)$  is equal to  $U$  up to permutation and phase, or up to a matrix multiplier  $P$  where  $P$  has exactly one unit modulus entry in each row and column. Such a joint diagonalizer is called essentially equal to  $U$ .

The algorithm of finding the joint diagonalizer is a generalization of Jacobi method or Givens rotation method [9]. As the cumulant matrices are estimated in practice, exact joint diagonalizer may not exist, instead, an approximate joint diagonalizer, an orthogonal matrix  $V$ , is sought to maximize the quantity:  $C(V, B) = \sum_{r=1}^{n^2} |\text{diag}(V' B_r V)|^2$ , where  $B = \{B_1, B_2, \dots, B_{n^2}\}$  is a set of basis (or eigen) matrices of  $Q_Z(M)$ ,  $|\text{diag}(A)|^2$  is the sum of squares of diagonals of a matrix  $A$ . Maximizing  $C(V, B)$  is same as minimizing off diagonal entries, which can be achieved in a finite number of steps of Givens rotations. The costs of joint diagonalization is roughly  $n^2$  times that of diagonalizing a single Hermitian matrix.

Though stationarity is assumed for the theoretical analysis above, JADE turns out to be quite robust even when stationarity is not exactly satisfied for signals such as speech or music.

## 2.2 Dynamic Method of Separating Convolutional Mixture

For each frequency  $\omega$ , equation (2.2) is a BSS problem of instantaneous mixtures. The speech or music signals in reality are stationary over short time scales and nonstationary over longer time scales, which depend on the production details. For speech signals, human voice is stationary for a few 10 ms, and becomes non-stationary for a time scale above 100 ms due to envelope modulations [8, 13]. The short time stationarity permits FFT to generate meaningful spectra in equation (2.2) within each frame. For a sampling frequency of 16,000 Hertz, each frame of 512 points lasts 32 ms. The mixing matrix  $H$  may depend on  $t$  over longer time scales, denoted by  $H = H(\omega, t)$ , unless the acoustic environment does not change as in most synthetic mixing. A demixing method with potential real time application should be able to capture the dynamic variation of mixing.

Our approach consists of four steps. Step I is to find an initialization for  $H(\omega, t)$ . After receiving the initial  $n_T$  frames of mixtures, compute their FFT and obtain  $X(\omega, t)$ ,  $t = 1, 2, \dots, n_T$ , to collect  $n_T$  samples at each discrete frequency. For each  $\omega$ , perform JADE, and estimate the mixing matrix denoted by  $H_0(\omega)$ . To ensure a good statistical estimate,  $n_T$  is on the order of 80 to 100, and may be properly reduced later.

Step I gives separated components of signals over all frequencies. However, such JADE output has indeterminacies in amplitude, order and phase. This benign problem for instantaneous mixtures becomes a major issue when one needs to assemble the separated individual components. For example, the permutation mismatches across frequencies can degrade the quality of separation seriously.

Step II is to use nonstationarity of signals to sort out a consistent order of separated signals in the frequency domain. Such a method for batch processing was proposed in [13]. A separation method requiring the entire length of the signal is called batch processing. The sorting algorithm of [13] proceeds as follows. (1) Estimate the envelope variation by a moving average over a number of frames (beyond stationarity time scale) for each separated frequency component. The envelope is denoted by  $\text{Env}(\omega, t, i)$ , where  $i$  is the index of separated components. (2) Compute a similarity measure equal to the sum of correlations of the envelopes of the separated components at each frequency. The similarity measure is  $\text{sim}(\omega) = \sum_{i \neq j} \rho(\text{Env}(\omega, t, i), \text{Env}(\omega, t, j))$ , where  $\rho(\cdot, \cdot)$  is the normalized correlation coefficients (see (2.9)) involving time average over the *entire signal length* to approximate the ensemble average so the  $t$  dependence drops out. (3) Let  $\omega_1$  be the one with lowest similarity value where separation is the best. The  $\omega_1$  serves as a reference point for sorting. (4) At other frequencies  $\omega_k$  ( $k = 2, 3, \dots$ ), find a permutation  $\sigma$  to maximize  $\sum_{i=1}^n \rho(\text{Env}(\omega_k, t, \sigma(i)), \sum_{j=1}^{k-1} \text{Env}_s(\omega_j, t, i))$ , among all permutations of  $1, 2, \dots, n$ . Here  $\text{Env}_s$  denotes the sorted envelopes in previous frequencies. (5) Permute the order of separated components at the  $k$ -th frequency bin according to  $\sigma$  in step (4), and define  $\text{Env}_s(\omega_k, t, i)$ . Repeat (4) and (5) until  $k = T$ .

We shall modify the above sorting method in three aspects. The first is to use segments of signal instead of the entire signal to compute statistics (correlations) to minimize delay in processing. The second is to use correlation coefficients of separated signals at *unequal times* or *multiple time lags* in step (2) to better characterize the degree of separation. Moreover, we notice that the similarity measure of [13] as seen above is a sum of correlation coefficients of potentially both signals, and so can be nearly zero due to can-

cellations even though each term in the sum is not small in absolute value. We introduce an  $l^1 \times l^\infty$  norm below to characterize more accurately channel similarity by taking sum of absolute values of correlation coefficients and maximum of time lags. The third is to simplify the maximization problem on  $\sigma$  to avoid comparing correlations with summed envelopes at all previous frequencies. We also do not use envelopes of signals inside correlation functions. The reason is that the smoothing nature of envelope operation reduces the amount of oscillations in the signals and may yield correlation values less accurate for capturing the degree of independence. Specifically, let  $\hat{s}_i(\omega, t) = a_i(\omega, t) e^{j\phi_i(\omega, t)}$  be the  $i$ -th separated signal at frequency  $\omega$ , where  $a_i(\omega, t) = |\hat{s}_i(\omega, t)|$ ,  $\phi_i$  the phase functions,  $t$  the frame index. The correlation function of two time dependent signals over  $M$  frames is:

$$\text{cov}(a(\omega, t), b(\omega', t)) = M^{-1} \sum_{t=1}^M a(\omega, t) b^*(\omega', t) - M^{-2} \sum_{t=1}^M a(\omega, t) \sum_{t=1}^M b^*(\omega', t), \quad (2.8)$$

and the (normalized) correlation coefficient is:

$$\rho(a(\omega, t), b(\omega', t)) = \frac{\text{cov}(a(\omega, t), b(\omega', t))}{\sqrt{\text{cov}(a(\omega, t), a(\omega, t)) \text{cov}(b(\omega', t), b(\omega', t))}}. \quad (2.9)$$

From speech production viewpoint, frequency components of a speech signal do not change drastically in time, instead are similarly affected by the motion of the speaker's vocal chords. The correlation coefficient is a natural tool for estimating coherence of frequency components of a speech signal. A similar argument may be applied to music signals as they are produced from cavities of instruments.

Now with  $M = n_T$  in (2.8), define

$$C(\omega) = \sum_{i \neq j} \max_{k \in \{-K_0, \dots, K_0\}} |\rho(|\hat{s}_i(\omega, t)|, |\hat{s}_j(\omega, t - k)|)|, \quad \text{for } \omega \in [\omega_L, \omega_U] \quad (2.10)$$

with some positive integer  $K_0$ . Find  $\omega_1$  between  $\omega_L$  and  $\omega_U$  to minimize  $C(\omega)$ . With  $\omega_1$  as reference, at any other  $\omega$ , find the permutation  $\sigma$  to maximize:

$$\sigma = \arg \max_{i=1}^n \sum_{k \in \{-K_0, \dots, K_0\}} \max_{k \in \{-K_0, \dots, K_0\}} |\rho(|\hat{s}_i(\omega_1, t)|, |\hat{s}_{\sigma(i)}(\omega, t - k)|)|. \quad (2.11)$$

Notice that the objective functions in (2.10)-(2.11) are exactly the  $l^1 \times l^\infty$  norms over the indices  $i(j)$  and  $k$ . Multiple time lag index  $k$  is to accomodate

the translational invariance of sound quality to the ear. Maximizing over  $k$  helps to capture the correlation of the channels, and sum of  $i$  ( $j$ ) reflects the total coherence of a vector signal.

Step III fixes the scaling and phase indeterminacies in  $\hat{s}(\omega, t)$ . Each *row* of the de-mixing matrix  $H_0^{-1}(\omega)$  may be multiplied by a complex number  $\lambda_i(\omega)$  ( $i = 1, 2 \dots, n$ ) before inverse FFT (ifft) to reconstruct demixing matrix  $h^{(0)}(\tau)$  in the time domain. The idea is to minimize the support of each row of the inverse FFT by a weighted least square method. In other words, we shall select  $\lambda_i$ 's so that the entries of  $\text{ifft}(H_0^{-1})(\tau) \equiv h^{(0)}(\tau)$  are real and nearly zero if  $\tau \geq Q$  for some  $Q < T$ ,  $Q$  as small as possible,  $T$  being the length of FFT. Smaller  $Q$  improves the local approximation, or accuracy of equation (2.2). To be more specific, using  $H_{0,i}^{-1}(\omega)$  to denote the  $i$ -th *row vector* of  $H_0^{-1}(\omega)$ , we can explicitly write the equation to shorten the support of inverse FFT:

$$\text{ifft}(\lambda_i(\omega)H_{0,i}^{-1}(\omega))(\tau) = 0 \quad (2.12)$$

in terms of the real and imaginary parts of  $\lambda_i(\omega)$  for  $\omega = 0, 1/T, \dots, (T-1)/T$ . Those real and imaginary parts are the variables and the equations are *linear*. Now, we let  $\tau$  run from  $q$  to  $T - 1$ . If we want small support,  $q$  should be small, then there are more equations than unknowns. So we multiply a weight to each equation and minimize in the least square sense. Equation (2.12) for larger  $\tau$  is multiplied by a larger weight in the hope that the value of the left hand side of (2.12) will be closer to zero during the least square process. If we choose the weighting function to be the exponential function  $\beta^\tau$  for some  $\beta > 1$ , then the above process can be mathematically written as

$$[\lambda_i(0), \dots, \lambda_i((T-1)/T)] = \underset{\tau=q}{\text{argmin}} \sum_{\tau=q}^{T-1} |\beta^\tau \text{ifft}(\lambda_i(\omega)H_{0,i}^{-1}(\omega))(\tau)|^2 \quad (2.13)$$

where  $H_{0,i}^{-1}(\omega)$  is the  $i$ -th *row vector* of  $H_0^{-1}(\omega)$ .

A few comments are in order. First, since the mixing matrix  $H_0(\omega)$  is the FFT of a real matrix, we impose that  $H_0(\omega) = H_0(1-\omega)^*$ . So, supposing  $T$  is even, we only need to apply JADE to obtain  $H_0(\omega)$  for  $\omega = 0, 1/T, \dots, 1/2$ ;  $H_0(0)$  and  $H_0(1/2)$  will automatically be real. When fixing the freedom of scaling in each  $\omega$ , we choose  $\lambda(0)$  and  $\lambda(1/2)$  real, and  $\lambda(\omega) = \lambda(1-\omega)^*$  for other  $\omega$ . Second, to fix the overall scaling and render the solution nontrivial, we set  $\lambda(0) = 1$ . Third, the weighted least square problem (2.13) can be solved by a direct method or matrix inversion (chapter 6 in [9]).

Note that when  $n = 2$ , among the  $2(T - q)$  equations from (2.12) with  $\tau = q, \dots, T - 1$ , there are  $T - 1$  variables including  $\lambda_i(1/2)$ , the real and



imaginary parts of  $\lambda_i(\omega)$  for  $\omega = 1/T, \dots, 1/2 - 1/T$ . So, we can make roughly half of  $h_i^{(0)}(\tau) \approx 0$ , the best one can achieve in general. Separated signals, denoted by  $\tilde{s}^{(0)}(t)$ , are then produced, for  $t \in [0, n_T]$ ,  $t$  the frame index.

The last step IV is to update  $h^{(0)}(\tau)$  when  $\delta n_T \ll n_T$  many new frames of mixtures arrive. The steps I to III are repeated using frames from  $\delta n_T + 1$  to  $\delta n_T + n_T$ , to generate a new time domain demixing matrix  $h^{(1)}(\tau)$ ,  $\tau \in [0, T - 1]$ , and separated signal  $\tilde{s}^{(1)}(\tau)$ ,  $\tau \in [T(n_T - \Delta n_T) + 1, T(n_T + \delta n_T)]$  with  $T$  the size of one frame. We use  $\tau$  here instead of  $t$  because in the most part of the paper,  $t$  is the frame index. Now,  $\tilde{s}^{(1)}(\tau)$  and  $s^{(0)}(\tau)$  share a common interval of size  $T\Delta n_T$ . On this common interval,  $\tilde{s}^{(1)}(\tau)$  and  $s^{(0)}(\tau)$  will be the same if we are doing a perfect job and if the ordering of  $\tilde{s}^{(1)}$  is consistent with that of  $s^{(0)}$ . In order to determine the ordering of  $\tilde{s}^{(1)}(\tau)$ , we compute  $\rho(\tilde{s}_i^{(0)}(\tau), s_j^{(1)}(\tau - k))$  on this common interval with different  $k$  and  $i, j = 1, \dots, n$ . Then we determine the permutation  $\sigma$  of the components of  $\tilde{s}^{(1)}(t)$  by minimization:

$$\sigma = \operatorname{argmax}_{i=1}^n \max_{k \in \{-K_1, \dots, K_1\}} \left| \rho\left(s_i^{(0)}(\tau), \tilde{s}_{\sigma(i)}^{(1)}(\tau - k)\right) \right| \quad (2.14)$$

with some constant  $K_1$ . After doing the necessary permutation of  $\tilde{s}^{(1)}$ , the separated signals are then extended to the extra frames  $\delta n_T + n_T$  by concatenating the newly separated  $\delta n_T$  many frames of  $\tilde{s}^{(1)}$  with those of  $\tilde{s}^{(0)}$ . The continuity of concatenation is maintained by requiring that  $\max_{\tau} |h_{ii}^{(k)}(\tau)|$ 's ( $i = 1, 2, \dots, n$ ) are invariant in  $k$ , where  $k = 1, 2, \dots$ , labels the updated filter matrix in time. The procedure repeats with the next arrival of mixture data, and is a direct method incorporating dynamic information.

Because sorting order depends only on the relative values of channel correlations, we observed in practice that the  $\max_{k \in \{-K, \dots, K\}}$  in equations (2.10), (2.11), (2.14) may be replaced by  $\sum_{k=-K}^K$ , with a different choice of  $K$  value. The  $\max_{k \in \{-K, \dots, K\}}$  is a more accurate characterization however.

### 2.3 Adaptive Estimation and Cost Reductions

Cumulants and moments are symmetric functions in their arguments [15]. For example when  $n = 2$ , there are 16 joint fourth order cumulants from (2.5), however, only six of them need to be computed, the others follow from

symmetry. Specifically, among the 16 cumulants:

$$\begin{aligned}
Q(1) &= \text{Cum}(y_1, y_1^*, y_1^*, y_1), & Q(2) &= \text{Cum}(y_1, y_1^*, y_1^*, y_2) \\
Q(3) &= \text{Cum}(y_1, y_1^*, y_2^*, y_1), & Q(4) &= \text{Cum}(y_1, y_1^*, y_2^*, y_2) \\
Q(5) &= \text{Cum}(y_1, y_2^*, y_1^*, y_1), & Q(6) &= \text{Cum}(y_1, y_2^*, y_1^*, y_2) \\
Q(7) &= \text{Cum}(y_1, y_2^*, y_2^*, y_1), & Q(8) &= \text{Cum}(y_1, y_2^*, y_2^*, y_2) \\
Q(9) &= \text{Cum}(y_2, y_1^*, y_1^*, y_1), & Q(10) &= \text{Cum}(y_2, y_1^*, y_1^*, y_2) \\
Q(11) &= \text{Cum}(y_2, y_1^*, y_2^*, y_1), & Q(12) &= \text{Cum}(y_2, y_1^*, y_2^*, y_2) \\
Q(13) &= \text{Cum}(y_2, y_2^*, y_1^*, y_1), & Q(14) &= \text{Cum}(y_2, y_2^*, y_1^*, y_2) \\
Q(15) &= \text{Cum}(y_2, y_2^*, y_2^*, y_1), & Q(16) &= \text{Cum}(y_2, y_2^*, y_2^*, y_2)
\end{aligned}$$

we have the relations:  $Q(2) = Q(3)^* = Q(5)^* = Q(9)$ ,  $Q(4) = Q(6) = Q(11) = Q(13)$ ,  $Q(7) = Q(10)^*$ ,  $Q(8) = Q(15) = Q(12)^* = Q(14)^*$ , where  $*$  is complex conjugate. For  $N$  samples, we only need to compute the following six  $1 \times N$  vectors

$$\begin{aligned}
Y_1 &= (y_1^1 y_1^1, \dots, y_1^N y_1^N), & Y_2 &= (y_1^1 y_2^1, \dots, y_1^N y_2^N), \\
Y_3 &= (y_2^1 y_2^1, \dots, y_2^N y_2^N), & Y_4 &= (y_1^1 y_1^{1*}, \dots, y_1^N y_1^{N*}), \\
Y_5 &= (y_1^1 y_2^{1*}, \dots, y_1^N y_2^{N*}), & Y_6 &= (y_2^1 y_2^{1*}, \dots, y_2^N y_2^{N*}),
\end{aligned}$$

then all the 4th order and 2nd order statistical quantities can be reconstructed. For example,

$$Q(1) = \frac{1}{N} Y_4 \cdot Y_4^{Tr} - \frac{1}{N^2} (2 \text{sum}(Y_4) \text{sum}(Y_4) + \text{sum}(Y_1) (\text{sum}(Y_1)^*)) \quad (2.15)$$

where  $\text{sum}(Y_i)$  is the summation of the  $N$  components of  $Y_i$ .

As formula (2.5) suggests, cumulants are updated through moments when  $\delta n_T$  early samples are replaced by the same number of new samples. As  $\delta n_T$  is much less than the total number of terms  $n_T$  in the empirical estimator of expectation, the adjustment costs  $2\delta n_T$  flops for each second moments and  $6\delta n_T$  flops for each joint fourth order moment. The contributions of the early samples are subtracted from the second and fourth moments, then the contributions of the new samples are added. The cumulant update approach is similar to cumulant tracking method of moving targets ([12] and references therein).

Due to dynamical cumulants update, the prewhitening step at each frequency is performed after cumulants are computed from  $X(\omega)$ . This is different from JADE [6] where the prewhitening occurs before computing the commulants. This way, it is more convenient to make use of the previous cumulant information and updated  $X(\omega)$ . Afterward, we use the multilinearity of the cummulants to transform them back to the commulants of the prewhitened  $X(\omega)$ , before joint diagonalization.

It is desirable to decrease  $n_T$  to lower the number of samples for cumulants estimation. However, this tends to increase the variance in the estimated cumulants, and render estimation less stable in time. Numerical experiments indicated that with  $n_T$  as low as 40, the separation using overlapping frames is still reliable with reasonable quality.

It is known [8] that the identity of a speaker is carried by pitch (perception of the fundamental frequency in speech production) which varies in the low frequency range of a few hundred Hertz. We found that instead of searching among all frequencies for the reference frequency  $\omega_1$  in step II, it is often sufficient to search in the low frequency range. The smaller searching range alleviates the workload in sorting and permutation correcting. This is similar to a feature oriented method, see [16, 18, 3] among others.

## 2.4 Experimental Results

The proposed algorithm with adaptivity and cost reduction considerations was implemented in Matlab. The original code of JADE by J.-F. Cardoso is obtained from a open source (<http://web.media.mit.edu/~paris/>) maintained by P. Smaragdis. Separation results with both dynamic and batch processing of three different types of mixtures are reported here:

- (1) real room recorded data;
- (2) synthetic mixture of speech and music;
- (3) synthetic mixture of speech and speech noise.

They will be called case (1), (2) and (3) in the following discussion.

The values of the parameters used in the three cases are listed in Table 1. In the table, " $n_T$  (dyn.)" is the initial value of  $n_T$  in dynamic processing and " $n_T$  (bat.)" is the  $n_T$  in batch processing. Other than  $n_T$ , dynamic and batch process share the same parameters. The frame size is  $T$ , "overlap" is the overlapping percentage between two successive frames,  $\delta n_T$  and  $\Delta n_T$  are as in step IV,  $K_0$  and  $K_1$  are from (2.11) and (2.14),  $\beta$  is in (2.13), and  $q$  is the lower limit of  $\tau$  in (2.12).

Note that the values of  $\omega_L$  and  $\omega_U$  from (2.10) are not listed in the table. In our computation, we use the following two choices

- (A)  $\omega_L = 0$ ,  $\omega_U = 1/2$ .
- (B)  $\omega_L = \omega_U = 4/T$ , namely fixing reference frequency  $\omega_1 = 4/T$ .

For the three cases reported in this paper, *both choices work and generate very similar results*. As a consequence, we will only plot the results of the first choice. The first choice is more general while the second is motivated by the pitch range of speech signal and is computationally more favorable. However, we do not know precisely the robustness of the latter.

case	T	overlap	$n_T$ (dyn.)	$\delta n_T$	$\Delta n_T$	$K_0$	$K_1$	$\beta$	$q$	$n_T$ (bat.)
(1)	512	0%	100	20	30	4	10	1.04	2	200
(2)	256	50%	100	20	40	15	20	1.04	2	160
(3)	256	50%	100	20	40	10	20	1.04	2	160

Table 1: Parameters used in both dynamic and batch processing.

For a quantitative measure of separation in all three cases, we compute the maximal correlation coefficient over multiple time lags:

$$\bar{\rho}(a, b) = \max_{k \in \{-K_2, \dots, K_2\}} |\rho(a(\tau), b(\tau + k))| \quad (2.16)$$

with  $\rho$  defined in (2.9). The  $\bar{\rho}$  is computed for the mixtures, the sources and the separated signals for both batch and dynamic processing. An exception is the lack of sources in case (1). We choose  $K_2 = 20$  in all the computations. The results are listed in Table 2 which shows that the  $\bar{\rho}$  values of the mixtures are much larger than those of the dynamically separated signals, which are on the same order as the  $\bar{\rho}$  values of the batch separated signals. In the synthetic cases (2) and (3), the  $\bar{\rho}$  values of the batch separated signals are on the same order of the  $\bar{\rho}$  values of the source signals or  $10^{-2}$ . In cases (2) and (3), we use the ratio  $\bar{\rho}(x, s_1)/\bar{\rho}(x, s_2)$  to measure the relative closeness of a signal  $x$  to source signals  $s_1$  and  $s_2$ . Table 3 lists these ratios for  $x$  being the separated signals by dynamic and batch methods with  $A$  and  $B$  denoting the two ways of setting the reference frequency  $\omega_1$ . The outcomes are similar no matter  $x = \tilde{s}_1$  or  $x = \tilde{s}_2$  (first or second separated signal) in either dynamic or batch cases and either way of selecting the reference frequency  $\omega_1$ .

In case (1), the recorded data [13] consists of 2 mixtures of a piece of music (source 1) and a digit (one to ten) counting sentence (source 2) recorded in a normal office size room. The sampling frequency is 16 kHz, and 100 k data points are shown in Fig. 1. The signals last a little over 6 seconds. The result of dynamic BSS algorithm is shown in Fig 2. As a comparison, we show in Fig. 3 result of batch processing of steps I to III of the algorithm

$\bar{\rho}(\cdot, \cdot)$ of 3 cases	mixture	dyn. separation	bat. separation	sources
(1)-A	0.8230	0.0269	0.0160	N/A
(1)-B	0.8230	0.0225	0.0159	N/A
(2)-A	0.6240	0.0503	0.0673	0.0201
(2)-B	0.6240	0.0182	0.0600	0.0201
(3)-A	0.4613	0.0351	0.0378	0.0243
(3)-B	0.4613	0.0267	0.0677	0.0243

Table 2: Values of the correlation coefficient  $\bar{\rho}(x, y)$ ,  $(x, y)$  being either the two mixtures or the two sources or the two separated signals by dynamic and batch methods. The A and B in the first column denote the two different ways of selecting the reference frequency  $\omega_1$ .

$\bar{\rho}(x, s_1)/\bar{\rho}(x, s_2)$	case(2)	case(3)
x= dyn. $\tilde{s}_1(A)$	4.5899	4.5096
x= dyn. $\tilde{s}_2(A)$	0.1086	0.2852
x= dyn. $\tilde{s}_1(B)$	5.3083	5.8411
x= dyn. $\tilde{s}_2(B)$	0.0494	0.2799
x= bat. $\tilde{s}_1(A)$	15.0912	1.4632
x= bat. $\tilde{s}_2(A)$	0.0760	0.1665
x= bat. $\tilde{s}_1(B)$	6.2227	25.8122
x= bat. $\tilde{s}_2(B)$	0.0636	0.1719

Table 3: Ratios of  $\bar{\rho}(x, s_1)$  and  $\bar{\rho}(x, s_2)$ ,  $x$  being a separated signal on the first column by dynamic or batch method,  $s_1$  and  $s_2$  are source signals. The ratio measures the relative closeness of  $x$  to  $s_1$  and  $s_2$ . If the ratio is larger (smaller) than one,  $x$  is closer to  $s_1$  ( $s_2$ ). The A and B in the first column denote the two different ways of selecting  $\omega_1$ .

with  $n_T = 200$ . The batch processing gives a clear separation upon listening to the separated signals. The dynamic processing is comparable. The filter coefficients in the time domain  $h_{ij}(\tau)$  at the last update of dynamic processing are shown in Fig. 4. Due to weighted least square optimization in step III, they are localized and oscillatory with support length  $Q$  close to half of the FFT size  $T$ .

For cases (2) and (3), we show the *envelopes of the absolute values* of the mixtures or the separated signals. The signal envelope was computed using the standard procedure of amplitude demodulation, i.e., lowpass filtering the rectified signal. The filter was an FIR filter with 400 taps and the cutoff frequency was 100 Hz. Signal envelopes help to visualize and compare source and processed signals. We have normalized all the envelopes so that the maximum height is 1. The values of  $a_{ij}$  in (2.1), which are used to synthetically generate the mixtures, are shown in Fig. 5 (see [19, (8)]). Fig. 6 and Fig. 7 show the mixtures and separated signals of case (2). Fig. 8 and Fig. 9 show the mixtures and separated signals of case (3). In view of these plots, Table 2 and Table 3, separation is quite satisfactory, which is also confirmed by hearing the separated signals.

The processing time in MATLAB on a laptop can be a factor of 5 to 8 above the real time signal duration, however, the time is expected to be closer to real time with the computation is executed by Fortran or C directly or with additional cost reduction techniques. A breakdown of time consumption in the algorithm shows that 40% of the processing is spent on computing cumulants, 30 % on sorting in frequency and time domains, 15% on fixing scaling functions, 3% on joint diagonalization, the rest on other operations such as computing lower order statistics, FFT, IFFT etc.

### 3 Conclusions

A dynamic blind source separation algorithm is proposed to track the time dependence of signal statistics and to be adaptive to the potentially time varying environment. Besides an efficient updating of cumulants, the method made precise the procedure of sorting permutation indeterminacy in the frequency domain by optimizing a metric (the  $l^1 \times l^\infty$  norm) on multiple time lagged channel correlation coefficients. A direct and efficient weighted least square approach is introduced to compactify the support of demixing filter to improve the accuracy of frequency domain localization of convolutive mixtures. Experimental results show robust and satisfactory separation of real recorded data and synthetic mixtures. An interesting line of future work

will be concerned with various strategies to reduce computational costs.

## 4 Acknowledgements

The work was partially supported by NSF grants ITR-0219004, DMS-0549215, NIH grant 2R44DC006734; the CORCLR (Academic Senate Council on Research, Computing and Library Resources) faculty research grant MI-2006-07-6, and a Pilot award of the Center for Hearing Research at UC Irvine.

## References

- [1] A. Bell, T. Sejnowski, *An Information-Maximization Approach to Blind Separation and Blind Deconvolution*, *Neural Computation*, 7(1995), pp 1129–1159.
- [2] P. Brémaud, “Mathematical Principles of Signal Processing: Fourier and Wavelet Analysis”, Springer-Verlag, 2002.
- [3] T. Chan, C. Wong, *Total variation blind deconvolution*, *IEEE Transactions Image Processing*, 7(1998), pp 370-375.
- [4] S. Choi, A. Cichocki, H. Park, S. Lee, *Blind Source Separation and Independent Component Analysis: A Review*, *Neural Information Processing -Letters and Reviews*, Vol. 6, No. 1, 2005, pp 1-57.
- [5] J-F. Cardoso, *Blind signal separation: statistical principles*, *Proceedings of IEEE*, V. 9, No. 10, pp 2009-2025, 1998.
- [6] J-F. Cardoso, A. Souloumiac, *Blind Beamforming for Non-Gaussian Signals*, *IEEE Proceedings-F*, vol. 140, no. 6, pp 362-370, 1993.
- [7] J-F. Cardoso, A. Souloumiac, *Jacobi angles for simultaneous diagonalization*, *SIAM J. Matrix Analysis*, vol. 17, pp 161-164, 1996.
- [8] L. Deng, D. O’Shaughnessy, “Speech Processing — A Dynamic and Optimization-Oriented Approach”, Marcel Dekker Inc., New York, 626 pages, 2003.
- [9] G. Golub, C. Van Loan, “Matrix Computations”, John Hopkins University Press, 1983.

- [10] S. Greenberg, W. Ainsworth, A. Popper, R. Fay, *Speech Processing in the Auditory Systems*, Springer Handbook of Auditory Research, Chapters 7 and 8, Springer, 2004.
- [11] M. Kawamoto, K. Matsuoka, N. Ohnishi, *A method of blind separation for convolved non-stationary signals*, Neurocomputing 22(1998), pp 157-171.
- [12] T. Liu, J. Mendel, *Cumulant-based subspace tracking*, Signal Processing, 76(1999), pp 237-252.
- [13] N. Murata, S. Ikeda, A. Ziehe, *An approach to blind separation based on temporal structure of speech signals*, Neurocomputing 41(2001), pp 1-24.
- [14] A. Nandi, eds, "Blind Estimation Using Higher-Order Statistics", Kluwer Academic Publishers, 1999.
- [15] C. Nikias, A. Petropulu, "Higher-Order Spectra Analysis: A Nonlinear Signal Processing Framework", Prentice-Hall Signal Processing Series, ed. A. Oppenheim, 1993.
- [16] S. Osher and L. Rudin, *Feature-Oriented Image Enhancement Using Shock Filters*, SIAM J. Numer. Analysis, Vol. 27, No. 4, pp 919-940, 1990.
- [17] L. Parra, C. Spence, *Convulsive Blind Separation of Non-Stationary Sources*, IEEE Transactions on Speech and Audio Processing, Vol. 8 (2000), No. 5, pp 320 –327.
- [18] Y. Qi, J. Xin, *A Perception and PDE Based Nonlinear Transformation for Processing Spoken Words*, Physica D 149 (2001),143-160.
- [19] K. Torkkola, *Blind separation of convolved sources based on information maximization*, Neural Networks Signal Processing, VI(1996), pp 423-432.



## Figure Captions

Fig 1: Case (1), two recorded signals in a real room where a speaker was counting ten digits with music playing in the background.

Fig 2: Case (1) with choice A, separated digit counting sentence (bottom) and background music (top) by the proposed dynamic method. Choice B gives similar results.

Fig 3: Case (1) with choice A, separated digit counting sentence (bottom) and background music (top) by batch processing using the proposed steps I to III. Choice B gives similar results.

Fig 4: Case (1) with choice A, the localized and oscillatory filter coefficients in the time domain at the last frame of dynamic processing. Choice B gives similar results.

Fig 5: The weights  $a_{ij}$  used in generating synthetic mixtures of cases (2) and (3), as proposed in [19].

Fig 6: Case (2), the synthetic mixtures are generated by a female voice and a piece of instrumental music.

Fig 7: Case (2) with choice A, the envelopes of the separated signals from mixtures whose envelopes are in Fig. (6). The small amplitude portion of the music is well recovered. Choice B gives similar results.

Fig 8: Case (3), the synthetic mixtures of a female voice and a speech noise with signal to noise ratio equal to  $-3.8206$  dB. The  $x_1$  plot shows a speech in a strong noise, the valley structures in the speech signal are filled by noise.

Fig 9: Case (3) with choice A, the envelopes of the separated signals, noise (top) and speech (bottom). The envelopes of the two mixtures are in Fig. 8. The strongly noisy  $x_1$  in Fig. 8 has been cleaned, the valleys in the envelope re-appeared. Choice B gives an even better result.

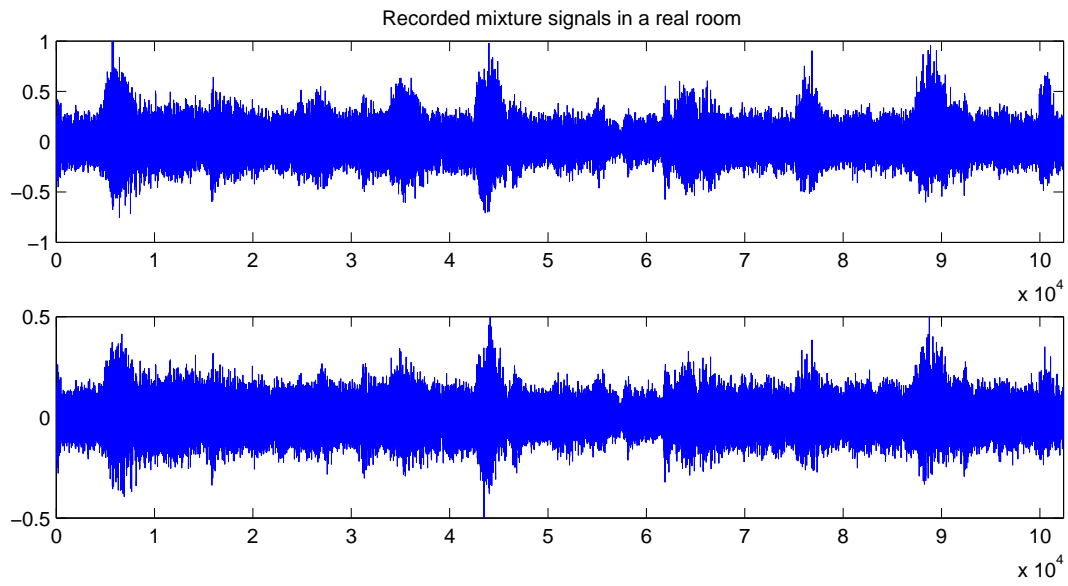


Figure 1:

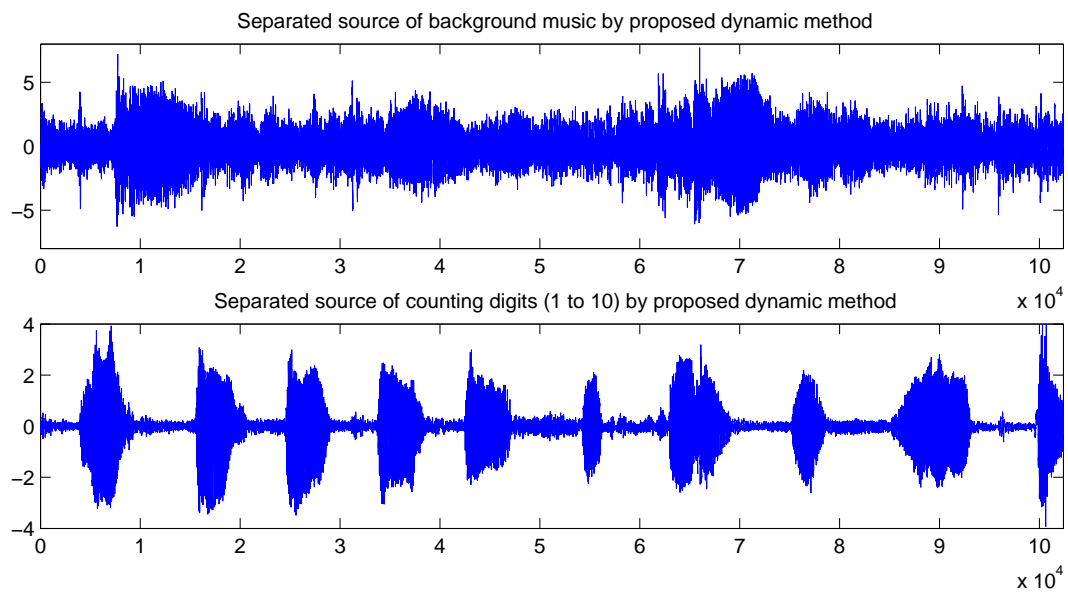


Figure 2:

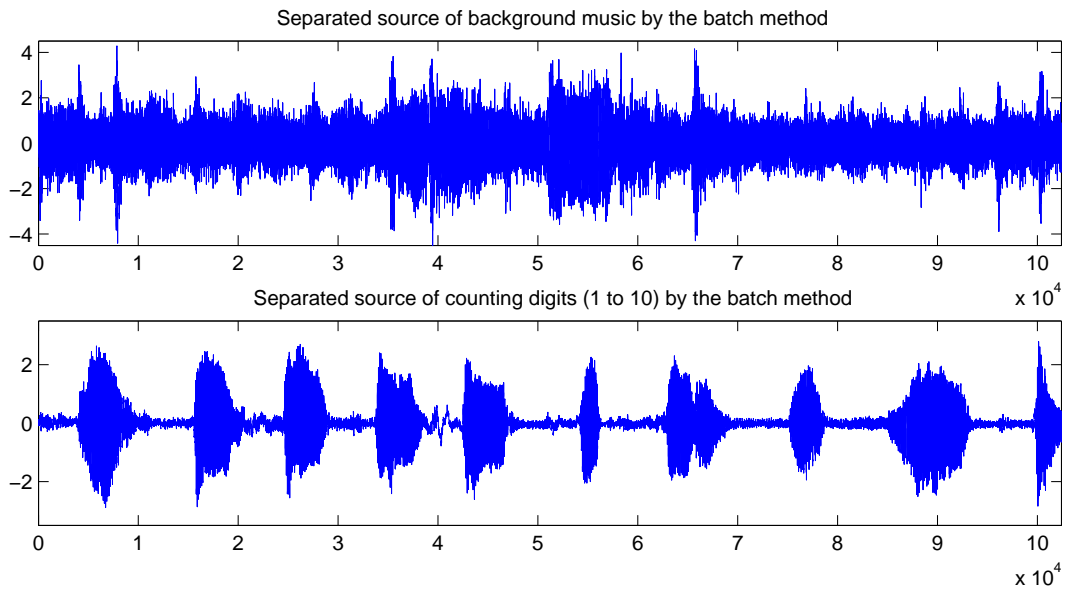


Figure 3:

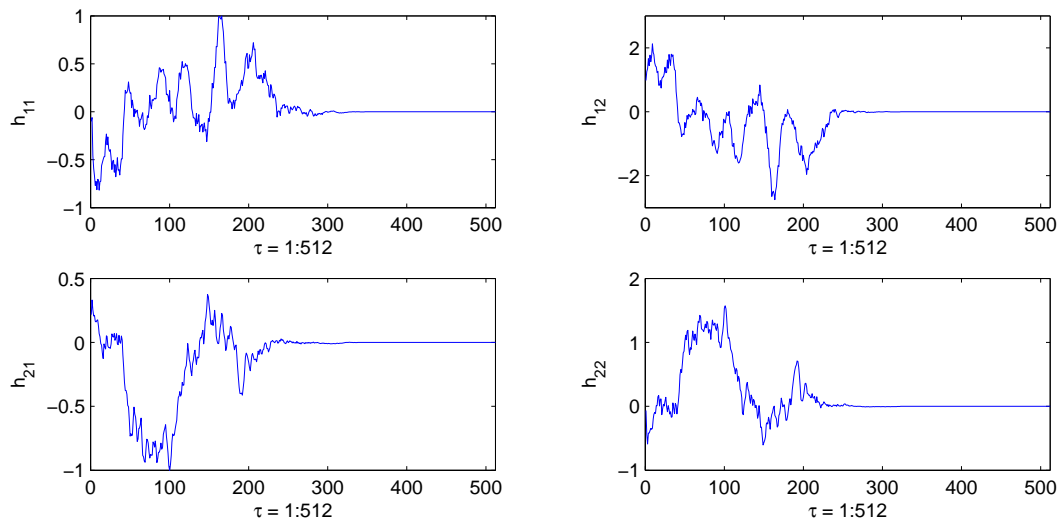


Figure 4:

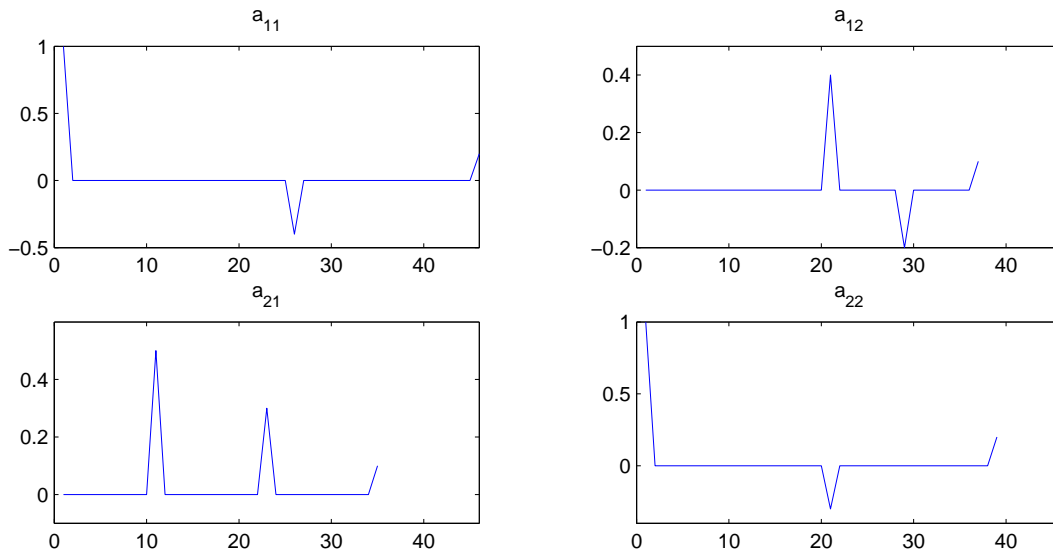


Figure 5:

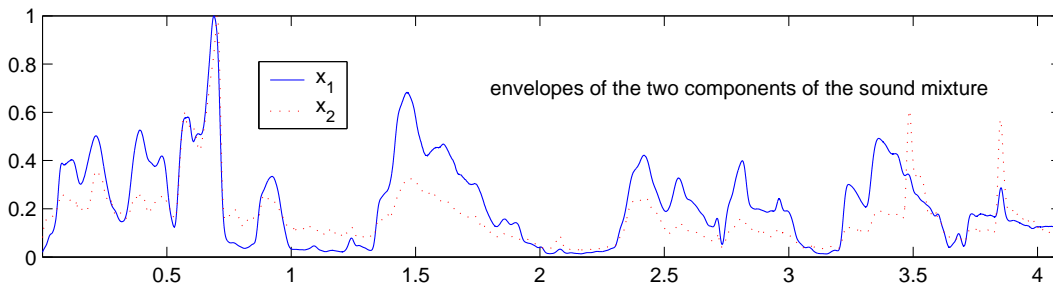


Figure 6:

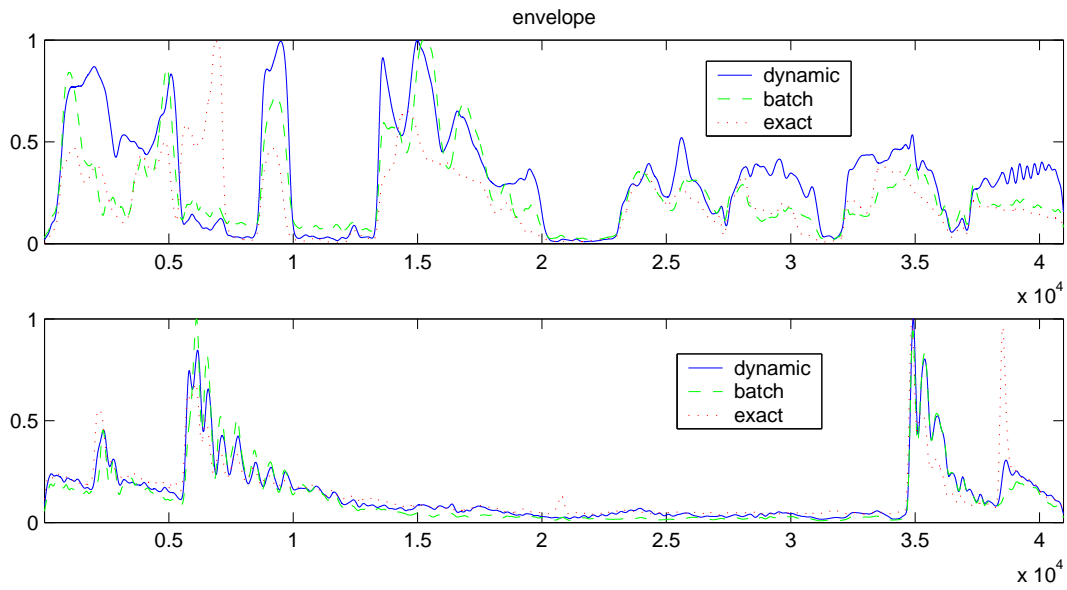


Figure 7:

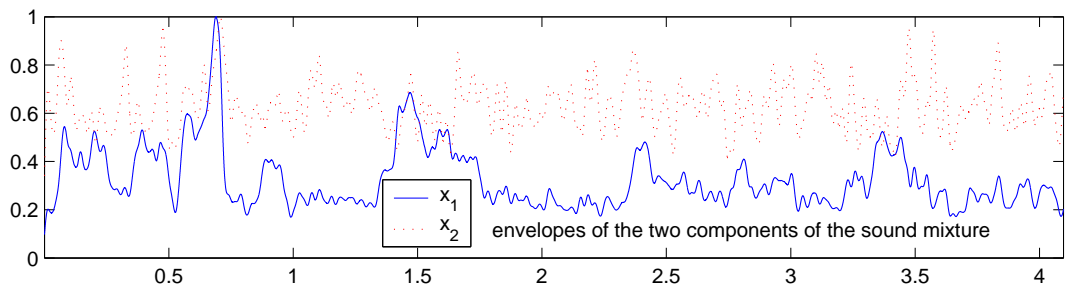


Figure 8:

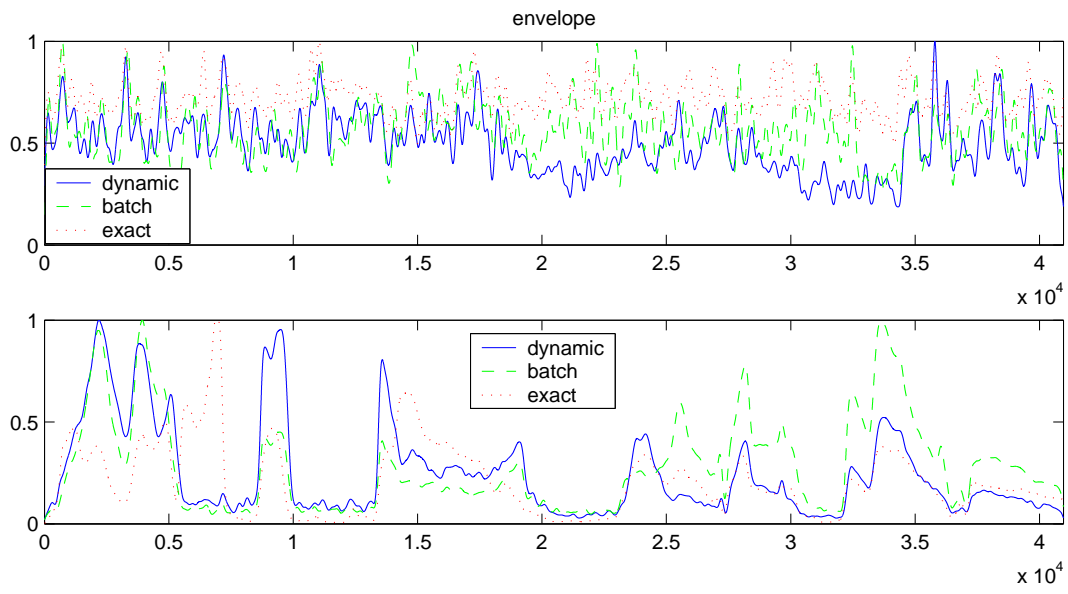


Figure 9: