

On-line Blind Source Separation of Non-Stationary Signals

Lucas Parra, Clay Spence
 Sarnoff Corporation, CN-5300, Princeton, NJ 08543,
 lparra@sarnoff.com, cspence@sarnoff.com

Abstract— We have shown previously that non-stationary signals recorded in a static multi-path environment can often be recovered by simultaneously decorrelating varying second order statistics. As typical sources are often moving, however, the multi-path channel is not static. We present here an on-line gradient algorithm with adaptive step size in the frequency domain based on second derivatives, which we refer to as multiple adaptive decorrelation (*MAD*). We compared the separation performance of the proposed algorithm to its off-line counterpart and to another decorrelation based on-line algorithm.

I. INTRODUCTION

Blind source separation is an ongoing research topic. Among the unresolved problems we consider two of particular importance. First, most of the known algorithms try to invert a multi-path acoustic environment by finding a multi-path finite impulse response (*FIR*) filter that approximately inverts the forward channel. However, a perfect inversion may require fairly long FIR filters. In our experience such situations occur in strongly echoic and reverberant rooms where most, if not all, current algorithms fail. Secondly, changing forward channels due to moving sources, moving sensors, or changing environments require an algorithm that converges sufficiently quickly to maintain an accurate current inverse of the channel. It is this second problem that we address here.

In terms of separation criteria to our knowledge there are three types of algorithms for blind source separation (*BSS*) of a convolutive mixture of broad-band signals. Algorithms that diagonalize a single estimate of the second order statistics, algorithms that simultaneously diagonalize second order statistics at multiple times exploiting non-stationarity, and algorithms that identify statistically independent signals by considering higher order statistics.

The simpler algorithms generate decorrelated signals by diagonalizing second order statistics [1], [2]. They have a simple structure that can be implemented efficiently [1], [3], [4], but are not guaranteed to converge to the right solution as single decorrelation is not a sufficient condition to obtain independent model sources [5], [1], [2]. Instead, for stationary signals higher order statistics have to be considered [6], either by direct measurement and optimization of higher order statistics [6], [7], [8], [9], [10], or indirectly by making assumptions on the shape of the cumulative den-

sity function (*cdf*) of the signals [11], [12], [13]. The former methods are fairly complex and difficult to implement. The latter methods fail in cases where the assumptions on the *cdf* are not accurate. Finally, for non-stationary signals it is known [1], and has been demonstrated [14], [15], [16], [17], that one may recover the signals by simultaneously diagonalizing second order statistics estimated at different times.

On-line algorithms are available for the single decorrelation [1], [2], [3], [18] approach and the indirect higher order methods [19], [20], but they have the same limitations as their off-line counterparts. More recently a number of single decorrelation algorithms have been presented that claim good performance for non-stationary signals. Some implicitly exploit non-stationarity [21], [4] while others reduce the number of required constraints by using low dimensional parameterizations of the filters [22], [21]. It is not difficult to think also of naive ways to transform our off-line algorithm [14], [15] for multiple decorrelation into an on-line version. But a rigorous derivation is more desirable, since we need fast convergence of the non-static filters, and the data may be visited only once.

We should mention that there is an obvious difficulty here in terms of non-stationarity of the mixing filters and the non-stationarity of the signals. Our suggested criteria exploits short time non-stationarity of the signals such as in speech. During the time that multiple second order statistics are collected or estimated we assume the mixing filter to remain approximately the same. That is, the sensors (microphones) and the source locations do not change with respect to the environment. If the filter changes too quickly, one may not be able to collect enough statistics to do the decorrelation for that time segment — even if the algorithm converges immediately to the correct solution. For any algorithm that is based on non-stationarity of the signal this will place an upper bound on the permissible variability of the environment. The convergence delay of the actual optimization can only lower that bound. Convergence is therefore a critical issue which we will try to address without complicating the algorithm too much and spending too much processing power. We suggest the use of an adaptive step size in the frequency domain motivated by second derivatives of the of the cost function. This effectively amounts to an adaptive power normalization in each frequency bin.

This paper is organized as follows. In section II we derive an on-line gradient algorithm starting with a time do-

main diagonalization criteria for multiple times. In section III-A a fast implementation in the frequency domain is obtained by making an approximation which is valid in the limit of small filter sizes compared to the window size of the Fourier transform. The adaptive step size in the frequency domain is derived in section III-B. In the last section we will compare the results of the proposed multiple adaptive decorrelation (*MAD*) algorithm in simulations with another on-line separation algorithm — an improved version of the adaptive decorrelation filter [3], [18], [1].

II. ON-LINE TIME-DOMAIN DECORRELATION

Assume non-stationary decorrelated source signals $\mathbf{s}(t) = [s_1(t), \dots, s_{d_s}(t)]^T$. These signals are observed in a multi-path environment $\mathbf{A}(\tau)$ of order P as $\mathbf{x}(t) = [x_1(t), \dots, x_{d_x}(t)]^T$,

$$\mathbf{x}(t) = \sum_{\tau=0}^P \mathbf{A}(\tau) \mathbf{s}(t-\tau) + \mathbf{n}(t). \quad (1)$$

We assume at least as many sensors as sources, i.e. $d_s \leq d_x$. It is known that under certain conditions on the coefficients of $\mathbf{A}(\tau)$ the multi-path channel is invertible with an FIR multi-path filter $\mathbf{W}(\tau)$ of suitably chosen length Q [23],

$$\hat{\mathbf{s}}(t) = \sum_{\tau=0}^Q \mathbf{W}(\tau) \mathbf{x}(t-\tau). \quad (2)$$

It has been shown [1] and demonstrated [14], [15], [16], [17] that non-stationary source signals can be recovered by optimizing filter coefficients \mathbf{W} such that the estimated sources $\hat{\mathbf{s}}(t)$ have diagonal second order statistics at different times,

$$\forall t, \tau: E[\hat{\mathbf{s}}(t) \hat{\mathbf{s}}^H(t-\tau)] = \hat{\Lambda}_s(t, \tau). \quad (3)$$

Here $\hat{\Lambda}_s(t, \tau) = \text{diag}([\hat{\lambda}_1(t, \tau), \dots, \hat{\lambda}_{d_s}(t, \tau)])$ represent the autocorrelations of the source signals at times t which have to be estimated from the data. This criteria determines the estimate $\hat{\mathbf{s}}(t)$ at best up to a permuted and convolved version of the original sources $\mathbf{s}(t)$. In previous works [14], [15], [16], [17] second-order statistics at multiple times have been estimated and simultaneously diagonalized in the frequency domain. In the present work, in order to derive an on-line algorithm we will consider the time domain directly and transfer the algorithm into the frequency domain later to get a more efficient and faster-converging on-line update rule. For the expectation we will use the sample average starting at time t , i.e. $E[f(t)] = \sum_{\tau'} f(t+\tau')$. We can then define a separation criteria (3) for simultaneous diagonalization with,

$$\mathcal{J}(\mathbf{W}) = \sum_t \mathcal{J}(t, \mathbf{W}) = \sum_t \|\mathbf{J}(t, \mathbf{W})\|^2 \quad (4)$$

$$= \sum_{t, \tau} \left\| E[\hat{\mathbf{s}}(t) \hat{\mathbf{s}}^H(t-\tau)] - \hat{\Lambda}_s(t, \tau) \right\|^2 \quad (5)$$

$$= \sum_{t, \tau} \left\| \sum_{\tau'} \hat{\mathbf{s}}(t+\tau') \hat{\mathbf{s}}^H(t-\tau+\tau') - \hat{\Lambda}_s(t, \tau) \right\|^2, \quad (6)$$

where we use the Frobenius norm given by $\|\mathbf{J}\|^2 = \text{Tr}(\mathbf{J}\mathbf{J}^H)$. We may now search for a separation filter by minimizing $\mathcal{J}(\mathbf{W})$ with a gradient algorithm. It is straightforward to compute the stochastic gradient of this optimization criteria with respect to the filter parameters.¹ We mean stochastic in the conventional sense of on-line algorithms that take a gradient step $\frac{\partial \mathcal{J}(t, \mathbf{W})}{\partial \mathbf{W}(l)}$ for every t instead of summing the total gradient before updating. This leads to the following expression,

$$\begin{aligned} \frac{\partial \mathcal{J}(t, \mathbf{W})}{\partial \mathbf{W}(l)} &= \sum_{\tau} \left(\sum_{\tau'} \hat{\mathbf{s}}(t+\tau') \hat{\mathbf{s}}^H(t+\tau'-\tau) - \hat{\Lambda}_s(t, \tau) \right) \\ &\quad \times \sum_{\tau''} \hat{\mathbf{s}}(t+\tau''-\tau) \mathbf{x}^H(t+\tau''-l) \\ &+ \sum_{\tau} \left(\sum_{\tau'} \hat{\mathbf{s}}(t+\tau'-\tau) \hat{\mathbf{s}}^H(t+\tau') - \hat{\Lambda}_s(t, \tau) \right) \\ &\quad \times \sum_{\tau''} \hat{\mathbf{s}}(t+\tau'') \mathbf{x}^H(t+\tau''-\tau-l) \end{aligned}$$

To simplify this expression, we show that the first and second sums over τ can be made equal. In the gradient descent procedure we may choose to apply the different gradient terms in these sums at times other than the time t . Following that argument, we can replace t with $t-\tau$ in the second sum, which effectively uses the value of the sum in the gradient update at time $t' = t-\tau$. In addition, if the sum over τ runs symmetrically over positive and negative values, we can change the sign of τ in the second sum. It can be argued that the diagonal matrix $\hat{\Lambda}_s(t, \tau)$ remains unchanged by these transformations, at least in a quasi-stationary approximation. The resulting update at time t for lag l with a step size of μ simplifies to

$$\begin{aligned} \Delta_t \mathbf{W}(l) &= -2\mu \sum_{\tau} \left(\sum_{\tau'} \hat{\mathbf{s}}(t+\tau') \hat{\mathbf{s}}^H(t+\tau'-\tau) - \hat{\Lambda}_s(t, \tau) \right) \\ &\quad \times \sum_{\tau''} \hat{\mathbf{s}}(t+\tau''-\tau) \mathbf{x}^H(t+\tau''-l). \end{aligned} \quad (7)$$

The sums over τ' and τ'' represent the averaging operations while the sum over τ stems from the correlation in (3). Denote the estimated cross-correlation of the sensor signals as,

$$\hat{\mathbf{R}}_x(t, \tau) = E[\hat{\mathbf{x}}(t) \hat{\mathbf{x}}^H(t-\tau)]. \quad (8)$$

By inserting (2) into (7) and using (8) we obtain for the update at time t ,

$$\begin{aligned} \Delta_t \mathbf{W} &= -2\mu \mathbf{J}(t) * \mathbf{W} * \hat{\mathbf{R}}_x(t), \\ \text{with } \mathbf{J}(t) &= \mathbf{W} * \hat{\mathbf{R}}_x(t) * \mathbf{W}^H - \hat{\Lambda}_s(t). \end{aligned} \quad (9)$$

In this short hand notation convolutions are represented by $*$, and correlations by \star , and time lag indices are omitted.

¹It is easy to see that the optimal estimates of the autocorrelation $\hat{\Lambda}_s(t, \tau)$ for a given \mathbf{W} are the diagonal elements of $E[\hat{\mathbf{s}}(t) \hat{\mathbf{s}}^H(t-\tau)]$. Therefore only the gradient with respect to \mathbf{W} is needed.

To obtain this expression we assumed that the estimated cross-correlations don't change much within the time scale corresponding to one filter length, i.e. $\hat{\mathbf{R}}_x(t, \tau) \approx \hat{\mathbf{R}}_x(t + l, \tau)$ for $0 < l \leq Q$.

III. A FREQUENCY DOMAIN GRADIENT ALGORITHM

In this section we will outline an on-line frequency domain implementation of this basic gradient algorithm. We try to reduce the computational cost as well as improve the convergence properties of the gradient updates.

A. Frequency domain gradient

Because the convolutions in (9) are expensive to compute, we transform this gradient expression into the frequency domain with T frequency bins. If we assume small filters compared to the number of frequency components, i.e. $Q \ll T$, the convolutions factor. Thus to a good approximation the stochastic gradient update rule (7) in the frequency domain is

$$\Delta_t \mathbf{W}(\omega) = 2\mu \mathbf{J}(t, \omega) \mathbf{W}(\omega) \hat{\mathbf{R}}_x(t, \omega), \quad (10)$$

with $\mathbf{J}(t, \omega) = \mathbf{W}(\omega) \hat{\mathbf{R}}_x(t, \omega) \mathbf{W}^H(\omega) - \hat{\Lambda}_s(t, \omega)$.

Here $\mathbf{W}(\omega)$ and $\hat{\mathbf{R}}_x(t, \omega)$ are the T point discrete Fourier transform of $\mathbf{W}(\tau)$ and $\hat{\mathbf{R}}_x(t, \tau)$ respectively. We note that the same expression can be obtained directly in the frequency domain by using the gradient of $\sum_{t, \omega} \mathcal{J}(t, \omega) = \sum_{t, \omega} \|\mathbf{J}(t, \omega)\|^2$ with respect to \mathbf{W}^* . In a gradient rule with complex variables this represents the combination of the partial derivatives with respect to the real and complex parts of \mathbf{W} . In this formalism the update rule for a complex parameter w with learning rate μ is $\Delta w = -2\mu \frac{\partial}{\partial w^*}$ [24].

B. Power normalization

In order to improve the convergence properties of this algorithm we would like to consider some second order gradient expressions. A proper Newton-Raphson update requires the inverse of the Hessian. Computing the exact inverse Hessian seems to be quite difficult in this case. An admittedly crude, yet common, approach is to neglect the off-diagonal terms of the Hessian. This should give efficient gradient updates in the case that the coefficients are not strongly coupled. If we ignore the constraints on the filter size in the time domain, the approximate frequency domain gradients updates depend only on $\mathbf{W}(\omega)$ as one can see in (10). The parameters are therefore decoupled for different frequencies. However the several elements of $\mathbf{W}(\omega)$ at a single frequency may be strongly dependent, in which case a diagonal approximation of the Hessian would be quite poor. In fact in our experiments we obtained poor stability with gradient steps modified by this diagonalized Hessian approximation, at least when the powers on different channels were very different. Given that observation we choose not to modify the gradient directions of the matrix elements $\mathbf{W}(\omega)$ for a given frequency. Instead we follow

the original gradient but adapt the step size with a normalization factor $h(t, \omega)$ for different frequencies,

$$\Delta_t \mathbf{W}(\omega) = -\mu h^{-1}(t, \omega) \frac{\partial \mathcal{J}(t, \omega)}{\partial \mathbf{W}^*(\omega)}. \quad (11)$$

As a motivation for this step size normalization consider the following. For a real valued square cost, $J(z) = azz^*$ in the complex plane the proper second order gradient step corresponds to $(\partial^2 J(z)/\partial z \partial z^*)^{-1} \partial J(z)/\partial z^* = z$. The corresponding expression in our current cost function can be computed to $\frac{\partial^2 \mathcal{J}}{\partial^2 W_{ij}^* \partial^2 W_{ij}} = 2((\mathbf{W}\mathbf{R}_x)^H(\mathbf{W}\mathbf{R}_x))_{jj}$. It is real valued and independent of i . We want to have the same step size for all j . Therefore we sum over j , and use

$$h(t, \omega) = \sum_j \frac{\partial^2 \mathcal{J}(t, \omega)}{\partial W_{ij}^*(\omega) \partial W_{ij}(\omega)} = \|\mathbf{W}(\omega) \hat{\mathbf{R}}_x(t, \omega)\|^2, \quad (12)$$

This is effectively an adaptive power normalization. In our experiments the resulting updates were stable and lead to faster convergence.

C. Block processing

In practice we implement the MAD algorithm as a block processing procedure. The signals are windowed and transformed into the frequency domain, i.e. the segment $x_i(t), \dots, x_i(t + T - 1)$ gives frequency components $x_i(t, \omega)$ for $\omega = 0, \dots, T - 1$. These are used to compute the estimated cross-correlations directly in the frequency domain, i.e. $\hat{\mathbf{R}}_x(t, \omega) = E[\mathbf{x}(t, \omega) \mathbf{x}^H(t, \omega)]$. In an on-line algorithm the expectation operation is typically implemented as an exponentially windowed average of the past. For the cross-correlations of the observations in the frequency domain this reads

$$\hat{\mathbf{R}}_x(t, \omega) = (1 - \gamma) \hat{\mathbf{R}}_x(t, \omega) + \gamma \mathbf{x}(t, \omega) \mathbf{x}^H(t, \omega). \quad (13)$$

where γ represents a forgetting factor to be chosen according to the stationarity time of the signal.

At time t also a gradient step (11) is performed with the current estimate $\hat{\mathbf{R}}_x(t, \omega)$. As we compute the updates in the frequency domain it is quite natural to implement the actual filtering (2) with an overlap-save routine. For this, one takes signal windows of size $2T$ and steps by T samples to obtain the next block of data. We can use these blocks of data for both the gradient and estimation updates. But we may need to perform more gradient steps before visiting new data. That is, we may want to update more frequently within that time T . In general, for a given frame rate r we compute the $2T$ frequency components and update $\mathbf{W}(\omega)$ and $\hat{\mathbf{R}}_x(t, \omega)$ at $t = T/r, 2T/r, 3T/r, \dots$. Conventional overlap and save corresponds to $r = 1$.

IV. EXPERIMENTAL RESULTS

In the following we measure performance with the Signal to Interference Ratio (*SIR*), which we define for a signal $\mathbf{s}(t)$ in a multi-path channel $H(\omega)$ to be the total signal powers

of the direct channel versus the signal power stemming from cross channels,

$$SIR[H, s] = \frac{\sum_{\omega} \sum_i |H_{ii}(\omega)|^2 \langle |s_i(\omega)|^2 \rangle}{\sum_{\omega} \sum_i \sum_{j \neq i} |H_{ij}(\omega)|^2 \langle |s_j(\omega)|^2 \rangle}. \quad (14)$$

In the case of known channels and source signals we can compute the expressions directly by using a sample average over the available signal and multiplying the powers with the given direct and cross channel responses.

We compared the performance of the proposed MAD algorithm to its corresponding off-line counterpart [14], [15], and to a second decorrelation algorithm presented in [3]. This so-called transformed domain adaptive decorrelation algorithm (*TDAD*) is a faster frequency domain implementation of the adaptive decorrelation algorithm presented by Weinstein et al. in [1]. It is based on decorrelating the covariance at a given time rather than at multiple times. As a result in principle it is not guaranteed to converge to the correct solution, but it has shown good performance in some simulated environments for a two source and two microphone case [3].

The most important question to address here is the adaption speed of the algorithm. The multi-path channel may change very rapidly with moving sources. In particular with fast head movements of a human subject the room response for that speaker may change quite abruptly. The question therefore is how much signal is required for MAD if the multi-path channel has completely changed.

We used one simulated room response and one real room recording with a few seconds of signal each at 8 kHz. In all instances the separation filters were 2048 taps. In both the simulated and real room recordings 40 randomly-chosen segments out of 15-second signals were used (figure 1 and 2). The mean and standard deviation for varying signal length are reported in figure 3 and figure 4. From these graphs we can conclude the following: 1. As expected we see that with increasing signal length the performance of MAD increases. 2. In all cases not more than a few seconds (3s-6s) of signal are required in order to converge to the best achievable performance. Within the margin of signal dependent variation in these particular environments the performance of MAD approaches that of the off-line algorithm within approximately 3.5s. 3. The TDAD algorithm is very dependent on the signal. It's average performance is considerably better than our algorithm for the simulated room but it fails for our recording in a real room. This observation is consistent with the results reported in [3] on simulated environments and with the theoretical considerations on convergence to correct solutions [2].

The simulated room was computed using the algorithms of [25]², with 2048 filter taps at 8 KHz (256 ms). We restricted the current experiments to the case of two sources and two microphones. The test signals consisted of a subject counting digits and a music source (see figure 1), to

²We thank Joseph G. Desloge and Michael P. O'Connell of the Sensory Communication Group, Research Laboratory of Electronics, Massachusetts Institute of Technology for the software implementation in MATLAB.

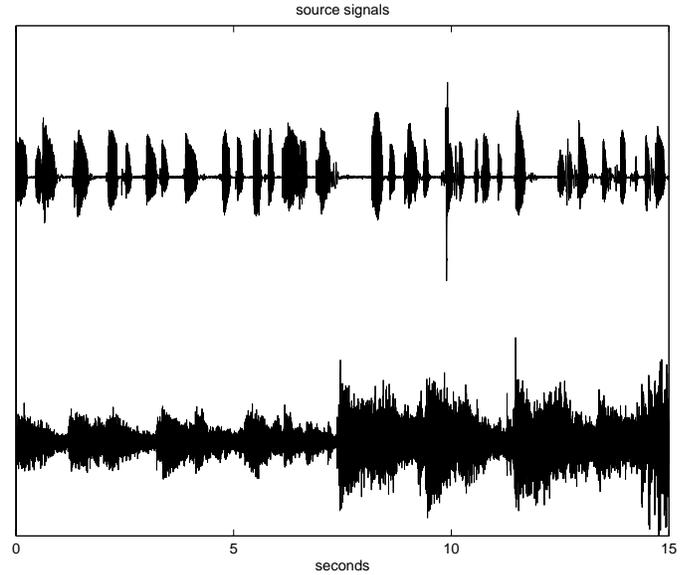


Fig. 1. Test sequences for the simulated room (see main text). These are fifteen-second speech and music sources sampled at 8 kHz. Forty different sections of varying length were used to estimate the SIR improvements.

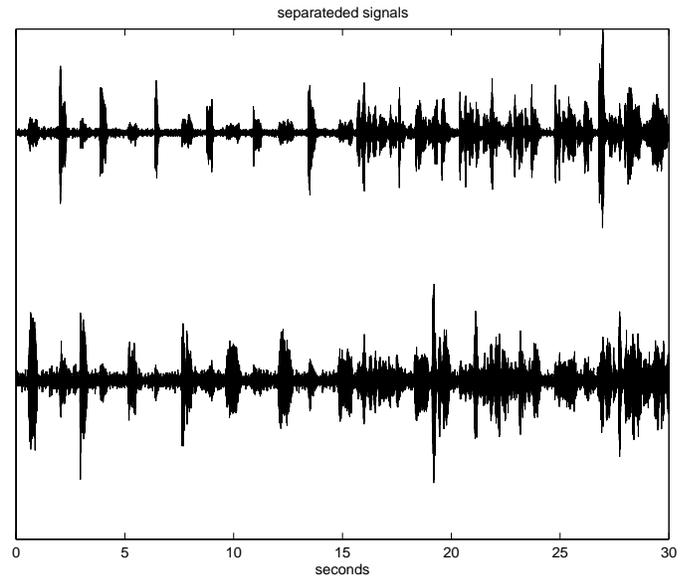


Fig. 2. Two separated speech signals recorded in a real office environment. These particular results corresponds to a 15 dB SIR improvement from 0.25 dB before separation. In the first fifteen seconds the speakers alternately spoke short digits. Because of the alternation, we could use this period to measure cross-talk power. Forty random sections of the last fifteen seconds were used for training.

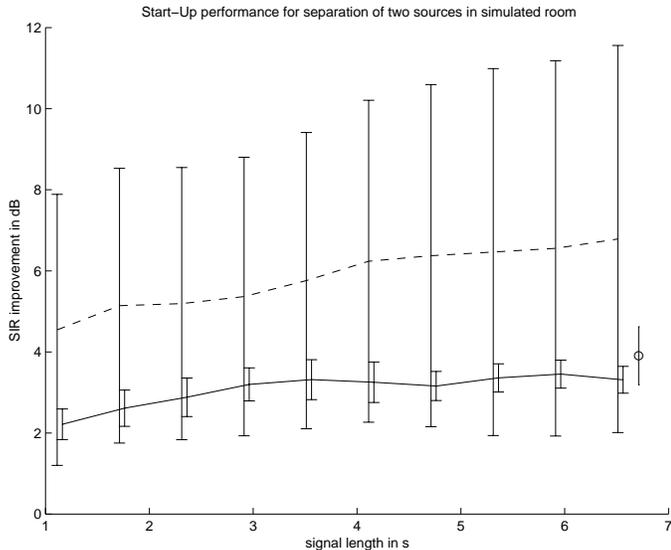


Fig. 3. Average SIR improvement before and after separation with the MAD (solid line) and TDAD (dashed line) algorithms in a simulated office environment with two microphones and two sources (see text). The bars represent the standard deviation obtained by running the algorithms on 40 different sections of the signal. The data-point on the very right represents the performance of the off-line multiple decorrelation algorithm.

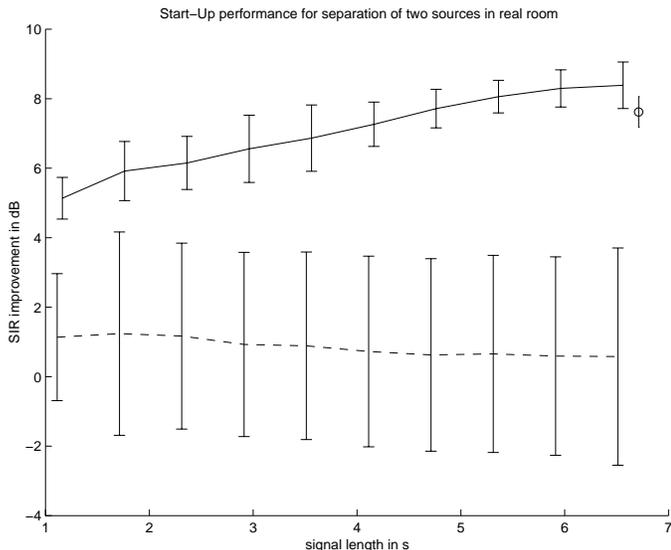


Fig. 4. Average SIR improvement before and after separation with the on-line MAD (solid line) and TDAD (dashed line) algorithms for real room recordings with two microphones and two sources (see text). The bars represent the standard deviations obtained by running the algorithms on forty different sections of the signal. The data-point on the far right is the performance of the off-line multiple decorrelation algorithm.

provide a more or less continuous stream of signal without long interruptions. The room is a 5 m x 5 m x 3 m rectangle with the absorption characteristic of a typical office room (carpet floor, sheet-rock or gypsum-board walls, and acoustic tiles in the ceiling). The two microphones are placed at one of the walls at a distance of two meters from each other. That wall was chosen to be anechoic in order to generate a directional characteristic for the microphones. The sources have been placed at a distance of 1.5 meters in front of the microphones

The real room recordings were obtained in a quiet office environment of about the same size. The first fifteen seconds of signal contain alternating speech and were used to measure the cross talk powers before and after separation. The second fifteen seconds of signal were used for training.

In the end of the previous section we pointed out that it might be advantageous for the block algorithm to use larger frame rates, though the computational cost is higher. In particular for realistic environments we are using filters of considerable length (256 ms). Updating at a slower frame rate might not be sufficient. In TDAD a frame rate of $r = 1$ is used. In the previous experiments our MAD algorithm used a frame rate of $r = 16$, which results in an update of the cross-correlation and filter coefficients every 16 ms. In order to choose the frame rates we determined the SIR improvements for the same real-room recordings as before for varying frame rates. In figure 5 we see that increasing the frame rate will in fact improve the performance. For this particular signal and environment a frame rate of 8 seems sufficient.

Finally we want to note the implementation of this algorithm in C runs in real time on a 155 kHz Intel Pentium processor for a 2 input, 2 sources problem with a frame rate of 8 at 8 kHz sampling rate and $T=2048$

V. CONCLUSION

A key result of this work is that an on-line source-separation algorithm which supposedly exploits multiple decorrelation can visit the signals only once *and yet* within three-to-four seconds achieve the same separation performance as an off-line multiple-decorrelation algorithm. At any time the on-line algorithm has only one estimate of the cross-correlation matrix. This stands in contrast to the off-line algorithm that diagonalizes multiple cross-correlation matrices by repeatedly accessing their values, which is equivalent to revisiting the data many times. Both algorithms assume non-stationarity of the signal so that multiple decorrelations provide sufficient conditions to unambiguously separate the sources. However in the on-line algorithm the multiple decorrelations are facilitated through a stochastic gradient that diagonalizes a changing average cross-correlation. To our knowledge this differs from all previous adaptive decorrelation algorithms, which drop the explicit averaging of the single cross-correlation matrix to get a stochastic update rule.

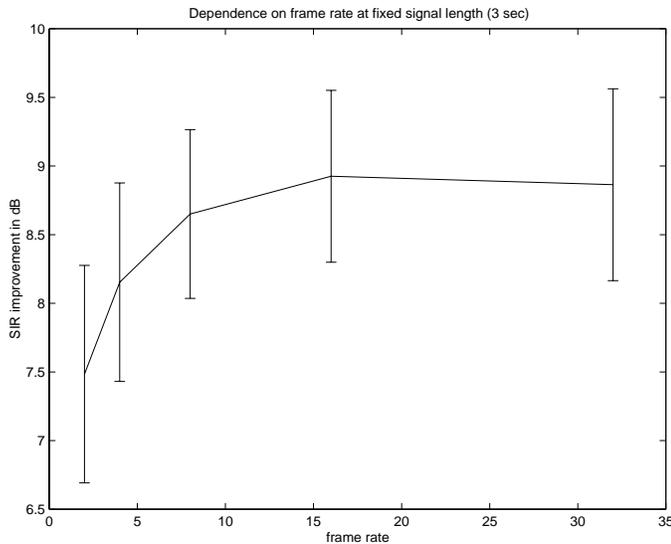


Fig. 5. Dependence of the MAD algorithm on the number of gradient updates per signal block size (filter size). For these results a filter length of 2048 taps was used. At 8 kHz a frame rate of 4 corresponds to an update every 64 ms.

REFERENCES

- [1] E. Weinstein, M. Feder, and A.V. Oppenheim, "Multi-Channel Signal Separation by Decorrelation", *IEEE Trans. Speech Audio Processing*, vol. 1, no. 4, pp. 405–413, Apr. 1993.
- [2] S. Van Gerven and D. Van Compernelle, "Signal Separation by Symmetric Adaptive Decorrelation: Stability, Convergence, and Uniqueness", *IEEE Trans. Signal Processing*, vol. 43, no. 7, pp. 1602–1612, July 1995.
- [3] K.-C. Yen and Y. Zhao, "Improvements on co-channel separation using ADF: Low complexity, fast convergence, and generalization", in *Proc. ICASSP 98*, Seattle, WA, 1998, pp. 1025–1028.
- [4] M. Kawamoto, "A method of blind separation for convolved non-stationary signals", *Neurocomputing*, vol. 22, no. 1-3, pp. 157–171, 1998.
- [5] S. Van Gerven and D. Van Compernelle, "Signal separation in a symmetric adaptive noise canceler by output decorrelation", in *Proc. ICASSP 92*, 1992, vol. IV, pp. 221–224.
- [6] D. Yellin and E. Weinstein, "Multichannel Signal Separation: Methods and Analysis", *IEEE Trans. Signal Processing*, vol. 44, no. 1, pp. 106–118, 1996.
- [7] H.-L. N. Thi and C. Jutten, "Blind source separation for convolutive mixtures", *Signal Processing*, vol. 45, no. 2, pp. 209–229, 1995.
- [8] V. Capdevielle, C. Serviere, and J.L. Lacoume, "Blind separation of wide-band sources in the frequency domain", in *Proc. ICASSP 95*, 1995, pp. 2080–2083.
- [9] S. Shamsunder and G. Giannakis, "Multichannel Blind Signal Separation and Reconstruction", *IEEE Trans. Speech Audio Processing*, vol. 5, no. 6, pp. 515–528, Nov. 1997.
- [10] S. Cruses and L. Castedo, "A Gauss-Newton method for blind separation of convolutive mixtures", in *Proc. ICASSP 98*, Seattle, WA, 1998.
- [11] R. Lambert and A. Bell, "Blind Separation of Multiple Speakers in a Multipath Environment", in *Proc. ICASSP 97*, 1997, pp. 423–426.
- [12] S. Amari, S.C. Douglas, A. Cichocki, and A.A. Yang, "Multichannel blind deconvolution using the natural gradient", in *Proc. 1st IEEE Workshop on Signal Processing App. Wireless Comm.*, 1997, pp. 101–104.
- [13] T. Lee, A. Bell, and R. Lambert, "Blind separation of delayed and convolved sources", in *Proc. Neural Information Processing Systems 96*, 1997.
- [14] L. Parra, C. Spence, and B. De Vries, "Convolutional Blind Source Separation based on Multiple Decorrelation", in *IEEE Workshop on Neural Networks and Signal Processing*, Cambridge, UK, September 1998, also presented at 'Machines that Learn' Workshop, Snowbird, April 1998.
- [15] L. Parra and C. Spence, "Convolutional blind source separation of non-stationary sources", *IEEE Trans. Signal Processing*, 1999, accepted.
- [16] N. Murata, S. Ikeda, and A. Ziehe, "An approach to blind source separation based on temporal structure of speech signals", *IEEE Trans. Signal Processing*, submitted.
- [17] J. Principe, "Simultaneous diagonalization in the frequency domain (SDIF) for source separation", in *ICA '99*, Cardoso, Jutten, and Loubaton, Eds., 1999, pp. 245–250.
- [18] K.-C. Yen and Y. Zhao, "Adaptive Co-Channel Speech Separation and Recognition", *IEEE Trans. Signal Processing*, vol. 7, no. 2, March 1999.
- [19] S. Amari, C.S. Douglas, A. Cichocki, and H.H. Yang, "Novel On-Line Adaptive Learning Algorithms for Blind Deconvolution Using the Natural Gradient Approach", in *Proc. 11th IFA C Symposium on System Identification*, Kitakyushu City, Japan, July 1997, vol. 3, pp. 1057–1062.
- [20] P. Smaragdis, "Blind separation of convolved mixtures in the frequency domain", *Neurocomputing*, vol. 22, pp. 21–34, 1998.
- [21] T. Ngo and N. Bhadkamkar, "Adaptive blind separation of audio sources by a physically compact device using second-order statistics", in *ICA '99*, Loubaton Cardoso, Jutten, Ed., 1999, pp. 257–260.
- [22] H. Sahlin and H. Broman, "Separation of real-world signals", *Signal Processing*, vol. 64, pp. 103–104, 1998.
- [23] P. Nelson, F. Orduna-Bustamante, and H. Hamada, "Inverse filter design and equalization zone in multichannel sound reproduction", *IEEE Trans. Speech Audio Processing*, vol. 3, no. 3, pp. 185–192, May 1995.
- [24] D Brandwood, "A complex gradient operator and its application in adaptive array theory", *IEE Proc.*, vol. 130, no. 1, pp. 11–16, Feb. 1983.
- [25] P.M. Peterson, "Simulating the response of multiple microphones to a single acoustic source in a reverberant room", *J. Acoust. Soc. Am.*, vol. 80, no. 5, pp. 1527–1529, Nov. 1986.