FREQUENCY-DOMAIN IMPLEMENTATION OF BLOCK ADAPTIVE FILTERS FOR ICA-BASED MULTICHANNEL BLIND DECONVOLUTION

Kyungmin Na, Sang-Chul Kang, Kyung Jin Lee, and Soo-Ik Chae

School of Electrical Engineering
Seoul National University
Shinlim-Dong, Gwanak-Gu, Seoul 151-742 KOREA
e-mail: {nkm, kang, jiny73, chae}@sdgroup.snu.ac.kr

ABSTRACT

In this paper, we present frequency-domain implementations of two adaptive multichannel blind deconvolution filters that employ the independent component analysis principle. The proposed implementations achieve considerable computational gains, which is shown by performing detailed analysis on the computational complexity. Particularly, our implementations incorporate a nonholonomic constraint to deal with overdetermined cases. The developed algorithms were successfully applied to the blind separation of real-world speech signals.

1. INTRODUCTION

Several real-world audio signals recorded simultaneously in a room with an array of microphones are usually modeled as different linear convolutive mixtures of unknown original audio signals such as speech and music [1]. To separate these signals, adaptive finite impulse response (FIR) filters with thousands of filter coefficients are often employed. When the impulse response of the room is not given *a priori*, the separation task is reduced to multichannel blind deconvolution (MBD) problem. The well-known *cocktail party problem* is one of typical examples of the MBD task.

Time-domain multichannel adaptive FIR filters with a number of filter coefficients need much computation if they employ sample-by-sample updating strategy [2], [3], [4]. One way to reduce the computational complexity is to use adaptive infinite impulse response (IIR) filters. However, they suffer from instability and local minima problems. An alternative approach is to employ block updating strategy in which the filter coefficients are kept fixed during a block of data and then are updated once at the end of the block. Such block adaptive FIR filters can be efficiently implemented in the frequency domain using fast Fourier transform (FFT)-based block processing. Considerable savings in the computational complexity are achieved by performing fast convolution and correlation with a proper data sectioning technique such as the overlap-save and overlap-add methods [4].

THIS WORK IS SUPPORTED BY BRAIN SCIENCE RESEARCH CENTER (BSRC).

In general, there are two approaches in obtaining those frequency-domain adaptive filters. One is to derive them directly in the frequency domain and the other is to realize time-domain block adaptive filters equivalently in the frequency domain [4]. For least mean-square (LMS) adaptive filters, it is known that both approaches are equivalent to each other. However, this is not the case for adaptive MBD filters [3]. Thus, these two approaches have been investigated separately so far [5]. The former approach allows to decompose a convolutive mixing problem in the time domain into a number of instantaneous mixing problems in the frequency domain and then solves each instantaneous mixing problem independently for the corresponding frequency bin. However, this approach suffers from scaling and permutation indeterminacy problem at each frequency bin. In addition, it is not fully revealed yet what nonlinear function is suitable for the adaptation in the frequency domain. Although there have been several heuristic methods to cope with them, they are still open problems.

This paper focuses on the latter approach since such problems do not occur in that approach [5]. In particular, we are interested in two famous adaptive MBD algorithms using the independent component analysis (ICA) principle: the standard gradient and natural (or relative) gradient MBD algorithms. By applying the FIR polynomial matrix algebra, Lee *et al.* directly formulated these two algorithms in the frequency domain and mentioned that they could be realized with block LMS techniques [6], [7], [8], [9]. However, their formulations did not correspond precisely to the time-domain block implementations of the two algorithms and detailed computational complexity analysis was not included, which motivates this work.

Therefore, the goals of this paper are twofold: 1) to present how to implement block versions of the two time-domain adaptive MBD algorithms precisely in the frequency domain and 2) to analyze their computational complexities in detail. To remove the matrix inversion operation therein and to deal with overdetermined cases efficiently, modified versions of the two adaptive MBD algorithms are employed in our implementations, which also discriminates them from those of Lee *et al.*. For a demonstration purpose, the proposed implementations were tested for 2 channel real-world speech signals provided by TeWon Lee [6], [7], [8], [9].

2. MULTICHANNEL BLIND DECONVOLUTION BY INDEPENDENT COMPONENT ANALYSIS

2.1. Multichannel Blind Deconvolution

Recently, multichannel blind deconvolution has attracted much interest due to its promising applications in signal processing and wireless communications [10]. In MBD tasks, an n-dimensional vector of observations $\mathbf{x}(t) = [x_1(t)\cdots x_n(t)]^T$ at a discrete time t is assumed to be given by a linear convolutive mixture of an m-dimensional vector of unobservable source signals $\mathbf{s}(t) = [s_1(t)\cdots s_m(t)]^T$:

$$\mathbf{x}(t) = \sum_{k=-\infty}^{\infty} \mathbf{H}_k \mathbf{s}(t-k), \qquad (1)$$

where $\mathbf{H}_k$ is an $(n \times m)$ mixing matrix at time lag k and the superscript T denotes matrix transpose.

Then, the goal of MBD is to obtain a deconvolution filter of $(m \times n)$ separating matrices, $\mathbf{W}_k$, $-\infty < k < \infty$, such that the m-dimensional recovered vector $\mathbf{y}(t) = [y_1(t)\cdots y_m(t)]^T$ given by

$$\mathbf{y}(t) = \sum_{k=-\infty}^{\infty} \mathbf{W}_k \mathbf{x}(t-k) \qquad (2)$$

corresponds to a scaled, permuted, and delayed version of the source vector $\mathbf{s}(t)$. Most MBD algorithms have tried to estimate this filter adaptively with the following adaptation rule: for $-\infty < k < \infty$,

$$\mathbf{W}_k(t+1) = \mathbf{W}_k(t) + \mu(t)\Delta\mathbf{W}_k(t), \qquad (3)$$

where $\mathbf{W}_k(t)$ is the estimate of $\mathbf{W}_k$ at a discrete time t, $\Delta\mathbf{W}_k(t)$ is an adaptation matrix, and $\mu(t)$ is a small positive step-size. Then, the filter coefficients are adjusted with time such that

$$\lim_{t\to\infty} \mathbf{W}(z,t)\mathbf{H}(z) = \mathbf{P}D(z) \qquad (4)$$

where $\mathbf{W}(z,t) = \sum_{k=-\infty}^{\infty} \mathbf{W}_k(t) z^{-k}$, $\mathbf{H}(z) = \sum_{k=-\infty}^{\infty} \mathbf{H}_k z^{-k}$, $\mathbf{P}$ is a permutation matrix, and $\mathbf{D}(z)$ is a diagonal matrix whose i-th diagonal entry is $a_i z^{-k_i}$ when a_i is a non-zero scaling value and k_i is an integer delay value [10].

In practice, the following L-tap FIR filter is employed, instead of the doubly-infinite deconvolution filter in (2):

$$\mathbf{y}(t) = \sum_{k=0}^{L-1} \mathbf{W}_k(t)\mathbf{x}(t-k). \qquad (5)$$

There have been several adaptive methods to estimate the deconvolution filter $\mathbf{W}_k(t)$, $k = 0, \cdots, L-1$. Among them, ICA has received particular interest because of its physical plausibility for various blind signal processing problems. Thus, we briefly summarize two existing ICA-based MBD algorithms, the standard gradient and natural gradient MBD algorithms, in the following.

2.2. The Standard Gradient MBD Algorithm

The standard gradient MBD algorithm was first proposed by Bell and Sejnowski [11] and was extended by Torkkola [1] later. It is described as

$$\Delta\mathbf{W}_0(t) = (\mathbf{W}_0^T(t))^{-1} - \varphi(\mathbf{y}(t))\mathbf{x}^T(t) \qquad (6)$$

$$\Delta\mathbf{W}_k(t) = -\varphi(\mathbf{y}(t))\mathbf{x}^T(t-k),\ k = 1, \cdots, L-1, \qquad (7)$$

where $(\mathbf{W}_0^T(t))^{-1}$ is the inverse matrix of $\mathbf{W}_0^T(t)$ and $\varphi(\mathbf{y}(t)) = (\varphi_1(y_1(t))\cdots\varphi_m(y_m(t)))^T$ is a component-wise nonlinear function called the *score function*.

However, the adaptation rule (6) for the zero delay separating matrix $\mathbf{W}_0(t)$ has two problems: (1) it requires computationally-expensive matrix inversion and (2) it is not suitable for overdetermined cases $(m < n)$ since the magnitude of each $y_i(t), i = 1, \cdots, m$ is controlled by it. To overcome these problems, Choi *et al.* [12] modified the adaptation rule (6) as follows:

$$\Delta\mathbf{W}_0(t) = \{\Gamma(t) - \varphi(\mathbf{y}(t))\mathbf{x}^T(t)\mathbf{W}_0^T(t)\}\mathbf{W}_0(t) \qquad (8)$$

where $\Gamma(t)$ is a diagonal matrix for a nonholonomic constraint whose diagonal entries are equal to the diagonal entries of the matrix $\varphi(\mathbf{y}(t))\mathbf{x}^T(t)\mathbf{W}_0^T(t)$. Here and in the following, we assume that the dimension of the recovered vector is n, i.e. $\mathbf{y}(t) = [y_1(t)\cdots y_n(t)]^T$, and that $n \geq m$. Then, the nonholonomic constraint allows us to deal with overdetermined cases by letting $(n-m)$ components of the recovered vector converge to zero quickly. Note that (8) is exactly equivalent to the natural gradient algorithm with the nonholonomic constraint for instantaneous mixing cases.

2.3. The Natural Gradient MBD Algorithm

The natural gradient MBD algorithm was first derived by Amari *et al.* [10] and was modified later by Choi *et al.* [13]

with the nonholonomic constraint as follows: for $k = 0, \cdots, L-1$,

$$\Delta \mathbf{W}_k(t) = \Gamma_0(t)\mathbf{W}_k(t) - \varphi(\mathbf{y}(t-L+1))\mathbf{u}^T(t-k), \quad (9)$$

where

$$\mathbf{u}(t) = \sum_{q=0}^{L-1} \mathbf{W}_{L-1-q}^T(t)\mathbf{y}(t-q) \quad (10)$$

and $\Gamma_0(t)$ is a diagonal matrix for the nonholonomic constraint whose i-th diagonal element is $\varphi_i(y_i(t-L+1))y_i(t-L+1)$. If $\Gamma_0(t)$ becomes an identity matrix, then (9) is reduced to the original natural gradient MBD algorithm. Note that the $(L-1)$-sample delayed values are used in (9) to avoid the noncausality problem.

3. BLOCK IMPLEMENTATIONS OF ICA-BASED ADAPTIVE MBD FILTERS

The adaptive algorithms in (7), (8), and (9) update the filter coefficients each time a new sample vector $\mathbf{x}(t)$ is received. An alternative updating strategy is a block updating strategy in which the filter coefficients are kept fixed during a block of sample vectors and then are updated once at the end of the block using the adaptation matrix accumulated during the block length [2]. Such block updating strategy is attractive since FIR block adaptive filter can be implemented efficiently in the frequency domain. Therefore, we consider the block implementations of the previous formulations. For notational convenience, we describe them in terms of direct filters $\mathbf{W}_{ii}(t)$, $i = 1, \cdots, n$ and cross-filters $\mathbf{W}_{ij}(t)$, $i \neq j, i, j = 1, \cdots, n$, where $\mathbf{W}_{ij}(t) = [w_{ij0}(t) \cdots w_{ij(L-1)}(t)]^T$ and $w_{ijk}(t)$ is the (i, j) element of $\mathbf{W}_k(t)$.

Let $b = 0, 1, 2, \cdots$ denote the block number and M the block length. From (5), then, the component-wise output at the b-th block becomes as follows: for $i = 1, \cdots, n$,

$$\mathbf{y}_i(bM) = [y_i(bM) \cdots y_i((b+1)M-1)]^T, \quad (11)$$

where

$$y_i(bM+p) = \sum_{j=1}^{n} \mathbf{W}_{ij}^T(bM)\mathbf{x}_j(bM+p), \quad (12)$$

with $\mathbf{x}_j(bM+p) = [x_j(bM+p) \cdots x_j(bM+p-L+1)]^T$. Thus, (12) corresponds to the sum of n linear convolutions.

Similarly, from (3), the filter coefficients are updated in blocks according to

$$\mathbf{W}_{ij}((b+1)M) = \mathbf{W}_{ij}(bM) + \mu(bM)\Delta\mathbf{W}_{ij}(bM) \quad (13)$$

where $\Delta\mathbf{W}_{ij}(bM) = [\Delta w_{ij0}(bM) \cdots \Delta w_{ij(L-1)}(bM)]^T$ is an averaged estimate of the gradient vector at block b. For the standard gradient MBD algorithm (7) and (8), each component of $\Delta\mathbf{W}_{ij}(bM)$ is given by

$$\Delta w_{ijk}(bM) = -c_{ijk}(bM), \ k = 1, \cdots, L-1, \quad (14)$$

and

$$\Delta w_{ij0}(bM) = -\sum_{p \neq i}^{n} \sum_{q=1}^{n} c_{iq0}(bM) w_{pq0}(bM) w_{pj0}(bM) \quad (15)$$

where

$$c_{ijk}(bM) = \sum_{l=bM}^{(b+1)M-1} \varphi_i(y_i(l)) x_j(l-k). \quad (16)$$

For the natural gradient MBD algorithm (9), each component of $\Delta\mathbf{W}_{ij}(bM)$ is obtained from

$$\Delta w_{ijk}(bM) = \sum_{l=bM}^{(b+1)M-1} \gamma_i(l) w_{ijk}(bM) - \sum_{l=bM}^{(b+1)M-1} \varphi_i(y_i(l-L+1)) u_j(l-k), \quad (17)$$

where $\gamma_i(l) = \varphi_i(y_i(l-L+1))y_i(l-L+1)$.

4. FREQUENCY-DOMAIN IMPLEMENTATIONS OF ICA-BASED ADAPTIVE MBD FILTERS

In the following, frequency-domain implementations of the block FIR adaptive MBD filters will be described. For this purpose, we employ the overlap-save method with the block length of $M = L$ and 50 % overlap since it provides the most efficient implementation [4]. Thus, the corresponding FFT size is $N = 2L$. For notational simplicity, we define $\mathbf{F}$ as the $N \times N$ discrete Fourier transform (DFT) matrix with its elements $F_{uv} = e^{-j2\pi uv/N}$. Its inverse is denoted by $\mathbf{F}^{-1}$. In real implementations, however, $\mathbf{F}$ and $\mathbf{F}^{-1}$ are performed by an FFT algorithm and thus considerable computational gains can be achieved.

4.1. Frequency-domain Implementation of MBD Filters

First consider the process of obtaining the block filter output in (11). Define $\bar{\mathbf{W}}_{ij}(b)$ and $\bar{\mathbf{X}}_j(b)$ as follows:

$$\bar{\mathbf{W}}_{ij}(b) = \mathbf{F}[\mathbf{W}_{ij}^T(bL) \ \underbrace{0 \cdots 0}_{L \text{ zeros}}]^T, \ i, j = 1, \cdots, n, \quad (18)$$

and

$$\bar{\mathbf{X}}_j(b) = \mathbf{F}[\underbrace{x_j((b-1)L)\cdots}_{(b-1)\text{-th block}}\underbrace{\cdots x_j((b+1)L-1)}_{b\text{-th block}}]^T. \quad (19)$$

Then, (11) can be obtained by the overlap-save method as follows: for $i = 1, \cdots, n$,

$$\mathbf{y}_i(bL) = \text{last } L \text{ elements of } \mathbf{F}^{-1}\bar{\mathbf{Y}}_i(b), \quad (20)$$

where

$$\bar{\mathbf{Y}}_i(b) = \sum_{j=1}^{n} \bar{\mathbf{W}}_{ij}(b) \odot \bar{\mathbf{X}}_j(b), \ i = 1, \cdots, n, \quad (21)$$

and $\odot$ denotes component-wise multiplication.

4.2. Frequency-domain Implementation of the Standard Gradient MBD Algorithm

Consider $\mathbf{C}_{ij}(bL) = [c_{ij0}(bL) \cdots c_{ij(L-1)}(bL)]^T$. Since (16) corresponds to a correlation, $\mathbf{C}_{ij}(bL)$ is calculated as follows:

$$\mathbf{C}_{ij}(bL) = \text{first } L \text{ elements of } \mathbf{F}^{-1}\{\bar{\mathbf{f}}_i(b) \odot \bar{\mathbf{X}}^*_j(b)\}, \quad (22)$$

where

$$\bar{\mathbf{f}}_i(b) = \mathbf{F}[\underbrace{0\cdots0}_{L \text{ zeros}} \ \varphi_i(y_i(bL)) \cdots \varphi_i(y_i((b+1)L-1))]^T \quad (23)$$

and the superscript $*$ denotes conjugate. Then, it is manifest from (14) and (15) that $\Delta\mathbf{W}_{ij}(bL)$ can be easily obtained from $\mathbf{C}_{ij}(bL)$. In the final step, the resulting time-domain gradient $\Delta\mathbf{W}_{ij}(bL)$ is transformed into the frequency domain and then is added to $\mathbf{W}_{ij}(bL)$ in order to produce the updated frequency-domain filter coefficients $\bar{\mathbf{W}}_{ij}(b+1)$. Since $\mathbf{W}_{ij}(bL)$ is followed by L zeros in (18), this is performed by

$$\bar{\mathbf{W}}_{ij}(b+1) = \bar{\mathbf{W}}_{ij}(b) + \mu(bL)\mathbf{F}[\Delta\mathbf{W}^T_{ij}(bL) \ \underbrace{0\cdots0}_{L \text{ zeros}}]^T. \quad (24)$$

In the frequency-domain implementation of block LMS adaptive filters, the time-domain filter coefficients are not involved in reality and thus the memory space for them is not necessary. However, this is not the case for the block standard gradient MBD algorithm since (15) needs the zero-delayed filter coefficients. Thus, it is necessary to keep their updated values at each block. This can be done simply by

$$w_{ij0}((b+1)L) = w_{ij0}(bL) + \mu(bL)\Delta w_{ij0}(bL) \quad (25)$$

for $i, j = 1, \cdots, n$. Thus, the additional memory space (n^2) is needed, which is relatively trivial since $n << L$.

4.3. Frequency-domain Implementation of the Natural Gradient MBD Algorithm

The first term in the right-hand side of (17) is just the product of $\sum_{l=bL}^{(b+1)L-1} \varphi_i(y_i(l-L+1))y_i(l-L+1)$ and $w_{ijk}(bL)$ and the second term can be calculated in the same way we get $c_{ijk}(bL)$ since it corresponds to a correlation. Let $h_i(bL) = \sum_{l=bL}^{(b+1)L-1} \varphi_i(y_i(l-L+1))y_i(l-L+1)$. Then, $\Delta\mathbf{W}_{ij}(bL)$ is obtained from

$$\Delta\mathbf{W}_{ij}(bL) = \text{first } L \text{ elements of} \\ \mathbf{F}^{-1}\{h_i(bL)\bar{\mathbf{W}}_{ij}(b) - \bar{\mathbf{f}}_i(b-1) \odot \bar{\mathbf{U}}^*_j(b)\} \quad (26)$$

where

$$\bar{\mathbf{U}}_j(b) = \mathbf{F}[\underbrace{u_j((b-1)L)\cdots}_{(b-1)\text{-th block}}\underbrace{\cdots u_j((b+1)L-1)}_{b\text{-th block}}]^T \quad (27)$$

Therefore, $\mathbf{u}_j(bL) = [u_j(bL) \cdots u_j((b+1)L-1)]^T$ should be calculated in advance to attain $\bar{\mathbf{U}}_j(b)$. From (10), it can be given by

$$\mathbf{u}_j(bL) = \text{first } L \text{ elements of} \\ \mathbf{F}^{-1}\{\sum_{i=1}^{n} \bar{\mathbf{W}}^*_{ij}(b) \odot \bar{\mathbf{Y}}_i(b)\} \quad (28)$$

for $j = 1, \cdots, n$, where $\bar{\mathbf{Y}}_j(b)$ is obtained by

$$\bar{\mathbf{Y}}_j(b) = \mathbf{F}[\underbrace{y_j((b-1)L)\cdots}_{(b-1)\text{-th block}}\underbrace{\cdots y_j((b+1)L-1)}_{b\text{-th block}}]^T \quad (29)$$

As in the previous, the frequency-domain filter coefficients $\bar{\mathbf{W}}_{ij}(b+1)$ are updated using (24).

4.4. Computational Complexity Ratios

Although several issues such as the number of additions, storage requirements, and hardware design would have to be considered for the computational complexity, it is often reasonable to examine the total number of multiplications for each implementation [4]. In the following, we assume that input data are real-valued and that a radix-2 FFT algorithm is employed. Then, each N-point FFT or inverse FFT (IFFT) requires approximately $N\log_2(N)$ real multiplications.

For each output sample, the n-channel standard gradient MBD algorithm with L-tap FIR filters requires n^2L real multiplications to compute its filter outputs and $n^2L + 2n^3(n-1)$ real multiplications to update the filter coefficients. Thus, a total of $2n^2L^2 + 2n^3(n-1)L$ real multiplications are needed to produce L output samples. Its frequency-domain implementation requires $2nN\log_2(N) + 4n^2N$ real multiplications to obtain its filter outputs in (20) and $2n^2N\log_2(N) + nN\log_2(N) + 4n^2N +$

$2n^3(n-1)$ real multiplications to update the filter coefficients. Since $N = 2L$, the computational complexity ratio is reduced to

$$CCR_{SG} = \frac{(2n+3)L\log_2(L) + (10n+3)L + n^3 - n^2}{nL^2 + n^2(n-1)L}. \tag{30}$$

When $n = 2$ and $L = 1024$, the frequency-domain implementation is roughly 22 times faster than the corresponding time-domain standard gradient MBD algorithm.

Similarly, for each output sample, the n-channel natural gradient MBD algorithm with L-tap FIR filters needs n^2L real multiplications to get its filter outputs and $n + 3n^2L$ real multiplications to update the filter coefficients. So, totally $4n^2L^2 + nL$ real multiplications are required to produce L output samples. Its frequency-domain implementation needs $2nN\log_2(N) + 4n^2N$ real multiplications to obtain its filter outputs and $2n^2N\log_2(N) + 4nN\log_2(N) + 10n^2N + nL$ real multiplications to update the filter coefficients. Thus, the computational complexity ratio is given by

$$CCR_{NG} = \frac{4(n+3)\log_2(L) + 32n + 13}{4nL + 1}. \tag{31}$$

When $n = 2$ and $L = 1024$, the frequency-domain implementation is roughly 30 times faster than the corresponding time-domain natural gradient MBD algorithm.

5. APPLICATIONS TO REAL-WORLD SPEECH SIGNAL SEPARATION

The developed frequency-domain adaptive MBD algorithms are applied to the blind separation of real-world speech signals in Fig. 1. These signals are provided by TeWon Lee [9]. These 2 channel speech signals were recorded in a real room in which two speakers spoke simultaneously digits in English and Spanish, respectively. In Fig. 2, Lee's separation results are depicted.

In these simulations, $\varphi_i(y_i) = y_i + \tanh(y_i)$ and $L = 1024$ taps FIR filters were employed. The step-size parameter was 0.001 and adaptation was run during 30 iterations of the speech signals. Fig. 3-4 show the plots of separated signals from the frequency-domain standard and natural gradient algorithms, respectively.

It was confirmed through simulations that the proposed implementations were much faster than their time-domain implementations and also that signals were well separated.

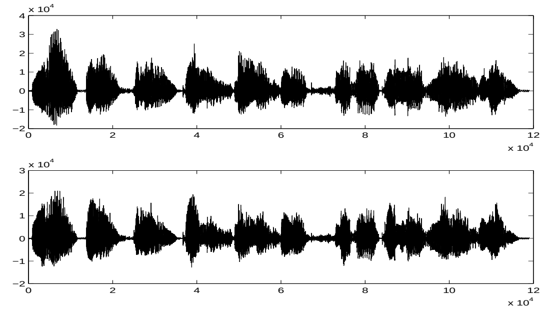

Figure 1: Real-world speech signals provided by Lee [7], [9].

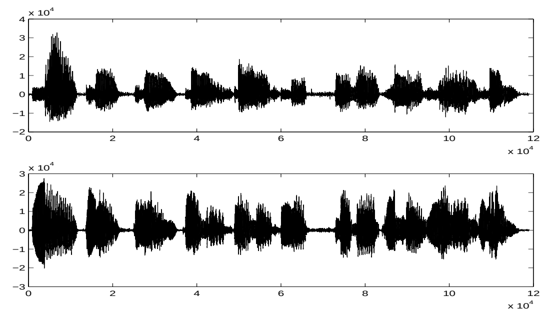

Figure 2: Separated signals from Lee [9].

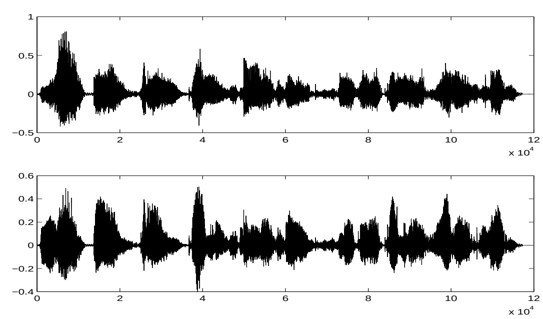

Figure 3: Separated signals from the standard gradient.

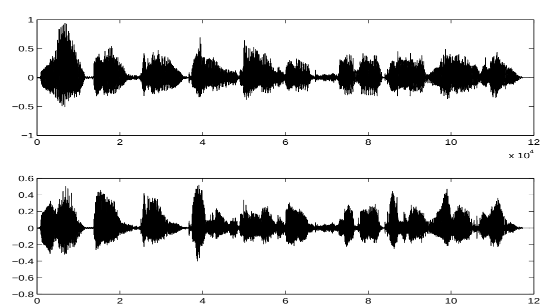

Figure 4: Separated signals from the natural gradient.

6. CONCLUSION

In this paper, we developed frequency-domain implementations of two time-domain adaptive FIR MBD filters based on the ICA principle. With the overlap-save data sectioning and FFT algorithm, the proposed ones precisely perform block adaptive MBD filtering. The detailed computational complexity ratios were presented and it was shown that considerable computational complexity reductions were achievable. To remove matrix inversion and to deal with overdetermined cases, modified algorithms were considered in our implementations. Though great computational gains are attained, the developed implementations have several weaknesses. They need more data points to get a desired performance since they work in blocks, not in samples. In addition, they cannot follow quickly the rapid changes of environments. Thus, some modifications to improve these problems may be needed. We are considering adaptive learning of the step-size parameter for this purpose. Also, we are currently trying to implement the proposed algorithms in DSP.

7. REFERENCES

[1] K. Torkkola, "Blind separation of convolved sources based on information maximization", *IEEE Workshop on Neural Networks for Signal Processing*, pp. 423–432, 1996.

[2] G. A. Clark, S. K. Mitra, and S. R. Parker, "Block implementation of adaptive digital filters", *IEEE Trans. Circuits and Systems*, vol. CAS-28, no. 6, pp. 584–592, June 1981.

[3] G. A. Clark, S. R. Parker, and S. K. Mitra, "A unified approach to time- and frequency-domain realization of fir adaptive digital filters", *IEEE Trans. Acoust., Speech, Sig. Proc.*, vol. ASSP-31, no. 5, pp. 1073–1083, Oct. 1983.

[4] J. J. Shynk, "Frequency-domain and multirate adaptive filtering", *IEEE Signal Processing Magazine*, vol. 9, no. 1, pp. 15–37, Jan. 1992.

[5] K. Torkkola, "Blind separation for audio signals - are we there yet ?", *Proc. ICA'99*, pp. 239–244, 1999.

[6] T.-W. Lee, A. Bell, and R. H. Lambert, "Blind separation of delayed and convolved sources", *Advances in Neural Information Processing Systems 9*, pp. 758–764, 1997.

[7] T.-W. Lee, A. Bell, and R. Orglmeister, "Blind source separation of real world signals", *Proc. International Conference on Neural Networks*, pp. 2129–2134, 1997.

[8] T.-W. Lee, A. Bell, and R. Orglmeister, "A contextual blind separation of delayed and convolved sources", *Proc. ICASSP*, pp. 1199–1202, 1997.

[9] T.-W. Lee, A. Ziehe, R. Orglmeister, and T. Sejnowski, "Combining time-delayed decorrelation and ica: towards solving the cocktail party problem", *Proc. ICASSP*, pp. 1249–1252, 1998.

[10] S. Amari, S. Douglas, A. Cichocki, and H. H. Yang, "Novel on-line adaptive learning algorithm for blind deconvolution using the natural gradient algorithm", *Proc. 11th IFAC Symposium on System Identification*, pp. 1057–1062, 1997.

[11] Bell and Sejnowski, "An information-maximization approach to blind separation and deconvolution", *Neural Computation*, vol. 7, no. 6, pp. 1129–1159, 1995.

[12] S. Choi, A. Cichocki, and S. Amari, "Two spatio-temporal decorrelation learning algorithms and their application to multichannel blind deconvolution", *Proc. Int. Conf. Acoust., Speech, and Signal Processing*, pp. 1085–1088, 1999.

[13] S. Choi, S. Amari, A. Cichocki, and R. Liu, "Natural gradient learning with a nonholonomic constraint for blind deconvolution of multiple channels", *Proc. ICA'99*, pp. 371–376, 1999.