Hindawi Publishing Corporation
EURASIP Journal on Audio, Speech, and Music Processing
Volume 2008, Article ID 349214, 10 pages
doi:10.1155/2008/349214

Research Article

Fast Noise Compensation and Adaptive Enhancement for Speech Separation

Rong Hu and Yunxin Zhao

Department of Computer Science, University of Missouri-Columbia, Columbia, MO 65211, USA

Correspondence should be addressed to Yunxin Zhao, zhaoy@missouri.edu

Received 4 December 2007; Revised 12 March 2008; Accepted 12 May 2008

Recommended by D. Wang

We propose a novel approach to improve adaptive decorrelation filtering- (ADF-) based speech source separation in diffuse noise. The effects of noise on system adaptation and separation outputs are handled separately. First, fast noise compensation (NC) is developed for adaptation of separation filters, forcing ADF to focus on source separation; next, output noises are suppressed by speech enhancement. By tracking noise components in output cross-correlation functions, the bias effect of noise on the system adaptation objective function is compensated, and by adaptively estimating output noise autocorrelations, the speech separation output is enhanced. For fast noise compensation, a blockwise fast ADF (FADF) is implemented. Experiments were conducted on real and simulated diffuse noises. Speech mixtures were generated by convolving TIMIT speech sources with acoustic path impulse responses measured in a real room with reverberation time $T_{60} = 0.3$ second. The proposed techniques significantly improved separation performance and phone recognition accuracy of ADF outputs.

1. INTRODUCTION

Interference speech and diffuse noise present double folds of challenges for hands-free automatic speech recognition (ASR) and speech communication. For practical applications of blind source separation (BSS), it is important to address the effects of noise in speech separation: (1) noise may degrade the conditions of BSS and hence hurt the separation performances; (2) BSS aims at source separation and has limited ability in suppressing diffuse noise. Although "bias removal" has been identified as a general approach for improving speech separation in noise [1], the performance depends largely on specific separation algorithms. Some noise compensation (NC) methods, for example [2], were proposed for a natural gradient-based separation algorithm. Other reported studies either focused primarily on theoretical issues, for example [3], or handled only conditions like uncorrelated noises, for example [4], or simplified mixing models, such as anechoic mixing [5]. The limitations of BSS in noise suppression were reported previously. Araki et al. [6, 7], established the mechanism similarities between BSS and the adaptive null beamformer. Asano et al. [8] grouped the two approaches into "spatial inverse" processing and pointed out that they are only able to suppress directional interferences but not omnidirectional ambient noises. Therefore, when both interference speech and diffuse noise are present, output noise suppression is needed in addition to separation processing. On the other hand, speech enhancement algorithms that are formulated for stationary noises cannot be applied directly in this scenario, because the adaptation of separation filters makes the output noise statistics time varying. Such variation may happen frequently when the mixing acoustic paths change, for example when a speaker moves.

In our previous works [9, 10], the separation model of adaptive decorrelation filtering (ADF) [11, 12] was significantly improved for noise-free speech mixtures in both aspects of convergence rate and steady-state filter estimation accuracy. A noise-compensated ADF [4] was proposed for speech mixtures contaminated by white uncorrelated noises. However, in real sound fields, diffuse noises are colored and spatially correlated in low frequency which deteriorate ADF performance more severely than uncorrelated noises [13]. It appears that noise can be removed from speech inputs prior to ADF separation. But such a noise prefiltering deteriorates the condition for subsequent source separation, due to

nonlinear distortions introduced by speech enhancement [13].

In the current work, we propose to address the challenge of speech separation and diffuse noise suppression by an effective two-step strategy. First, a noise compensation (NC) [14] algorithm is developed to improve speech separation performances; effective blockwise implementations of compensation processing and ADF filtering are derived in FFT. As separation filters change over time, output noise statistics of cross-correlations are tracked so that filter adaptation bias can be removed. Second, output noise autocorrelations are estimated and used to enhance the speech signals separated in the first step [15], so as to improve speech quality. Speech separation, enhancement, and phone recognition experiments were conducted, and the results are presented to show the performances of the proposed separation and enhancement techniques.

2. ADF MODEL IN NOISE

In the following, we use variables in bold lower case for vectors, bold upper case for matrices, superscript T for transposition, $\mathbf{I}$ for the identity matrix, "$*$" for convolution, and $E\{\}$ for expectation. The correlation matrix formed by vectors $\mathbf{a}$ and $\mathbf{b}$ is defined as $\mathbf{R}_{\mathbf{ab}} = E\{\mathbf{a}\mathbf{b}^T\}$, and the correlation vector between a scalar a and a vector $\mathbf{b}$ as $\mathbf{r}_{a\mathbf{b}} = E\{a\mathbf{b}\}$. N and K denote filter and block lengths, respectively. Speech and noise signal vectors contain N consecutive samples up to current time t, and their counterparts with $2N-1$ samples up to time t are marked with tilde.

The noisy speech mixing and ADF separation systems are shown in Figure 1, where $\mathbf{g}_{ij} = [g_{ij}(0), \ldots, g_{ij}(N-1)]^T$, $i, j = 1, 2$, $i \neq j$, are separation filters. We formulate the I/O relations of ADF as [4]

$$\mathbf{v}_n = \mathbf{G}(\tilde{\mathbf{y}} + \tilde{\mathbf{n}}), \tag{1}$$

where $\tilde{\mathbf{y}} = [\tilde{\mathbf{y}}_1^T(t), \tilde{\mathbf{y}}_2^T(t)]^T$ and $\tilde{\mathbf{n}} = [\tilde{\mathbf{n}}_1^T(t), \tilde{\mathbf{n}}_2^T(t)]^T$ are vectors of the clean speech mixture and the noise, respectively, with $\tilde{\mathbf{y}}_i = [y_i(t), \ldots, y_i(t-2N+2)]^T$, $\tilde{\mathbf{n}}_i = [n_i(t), \ldots, n_i(t-2N+2)]^T$, $i = 1, 2$. The filter matrix

$$\mathbf{G} = \begin{bmatrix} \begin{bmatrix} \mathbf{I}_N & \mathbf{0}_{N\times(N-1)} \end{bmatrix} & -\mathbf{G}_{12} \\ -\mathbf{G}_{21} & \begin{bmatrix} \mathbf{I}_N & \mathbf{0}_{N\times(N-1)} \end{bmatrix} \end{bmatrix} \tag{2}$$

is $2N \times (4N-2)$, where $\mathbf{G}_{ij}$ is an $N \times (2N-1)$ Toeplitz matrix and its kth row is $[\mathbf{0}_{1\times(k-1)}, \mathbf{g}_{ij}^T, \mathbf{0}_{1\times(N-k)}]$, $k = 1, \ldots, N$. For the noisy ADF output $\mathbf{v}_n$, its speech-only output is denoted by $\mathbf{v} = [\mathbf{v}_1^T(t), \mathbf{v}_2^T(t)]^T$ and the noise output component by $\boldsymbol{\eta} = [\boldsymbol{\eta}_1^T(t), \boldsymbol{\eta}_2^T(t)]^T$. Then, the effect of noise in the system output correlation matrix is described by $\mathbf{R}_{\mathbf{v}_n\mathbf{v}_n} = \mathbf{R}_{\mathbf{vv}} + \mathbf{R}_{\boldsymbol{\eta}\boldsymbol{\eta}}$. The I/O relations in correlation vectors of speech are

$$\begin{aligned} \mathbf{r}_{v_i\mathbf{v}_j} &= \mathbf{r}_{y_i\mathbf{y}_j} - \mathbf{G}_{ji}\mathbf{r}_{y_i\tilde{\mathbf{y}}_i} - \mathbf{R}_{\mathbf{y}_j\mathbf{y}_j}\mathbf{g}_{ij} + \mathbf{G}_{ji}\mathbf{R}_{\tilde{\mathbf{y}}_i\mathbf{y}_j}\mathbf{g}_{ij}, \\ \mathbf{r}_{v_i\mathbf{v}_i} &= \mathbf{r}_{y_i\mathbf{y}_i} - \mathbf{G}_{ij}\mathbf{r}_{y_i\tilde{\mathbf{y}}_j} - \mathbf{R}_{\mathbf{y}_i\mathbf{y}_j}\mathbf{g}_{ij} + \mathbf{G}_{ij}\mathbf{R}_{\tilde{\mathbf{y}}_j\mathbf{y}_j}\mathbf{g}_{ij}. \end{aligned} \tag{3}$$

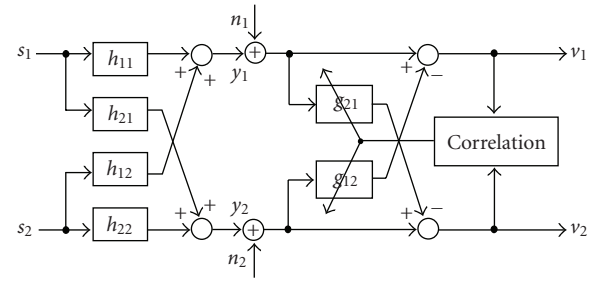

Figure 1: Speech mixing and ADF separation system in noise.

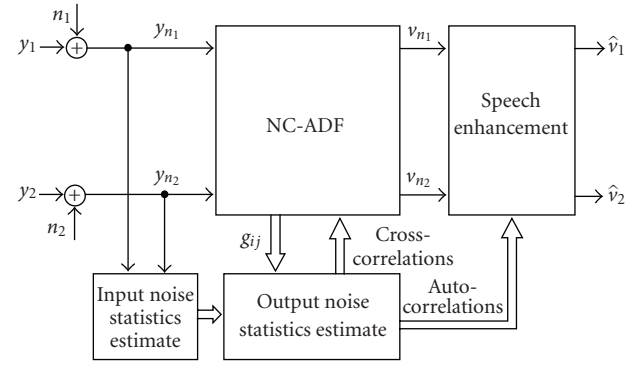

Figure 2: NC-ADF separation and adaptive enhancement system.

The noise correlation I/O relations have the same form:

$$\mathbf{r}_{\eta_i\boldsymbol{\eta}_j} = \mathbf{r}_{n_i\mathbf{n}_j} - \mathbf{G}_{ji}\mathbf{r}_{n_i\tilde{\mathbf{n}}_i} - \mathbf{R}_{\mathbf{n}_j\mathbf{n}_j}\mathbf{g}_{ij} + \mathbf{G}_{ji}\mathbf{R}_{\tilde{\mathbf{n}}_i\mathbf{n}_j}\mathbf{g}_{ij}, \tag{4}$$

$$\mathbf{r}_{\eta_i\boldsymbol{\eta}_i} = \mathbf{r}_{n_i\mathbf{n}_i} - \mathbf{G}_{ij}\mathbf{r}_{n_i\tilde{\mathbf{n}}_j} - \mathbf{R}_{\mathbf{n}_i\mathbf{n}_j}\mathbf{g}_{ij} + \mathbf{G}_{ij}\mathbf{R}_{\tilde{\mathbf{n}}_j\mathbf{n}_j}\mathbf{g}_{ij}. \tag{5}$$

In the absence of noise, the basic ADF adaptation algorithm is given in [11] as

$$\mathbf{g}_{ij}(t+1) = \mathbf{g}_{ij}(t) + \mu(t)v_i(t)\mathbf{v}_j(t). \tag{6}$$

It has been shown in [4] that by taking the decorrelation objective functions as

$$J_{ij} = \frac{1}{2}\mathbf{r}_{v_i\mathbf{v}_j}^T\mathbf{r}_{v_i\mathbf{v}_j}, \tag{7}$$

and approximating $\mathbf{r}_{v_i\mathbf{v}_j}$ by instantaneous correlations $v_i(t)\mathbf{v}_j(t)$, the same adaptation equation can be obtained. For the step-size $\mu(t)$, [12] proposed an input-normalized technique based on a convergence analysis, which was combined in [9] with variable step-size (VSS) techniques to accelerate convergence and reduce ADF estimation error.

The proposed system for improving ADF in noise works in two steps, as shown in Figure 2. In the NC step, the noise effects on the adaptation procedure (6), including the step-size computation, are reduced to improve speech separation. In the adaptive enhancement step, the ADF speech outputs are enhanced by noise reduction. The details of the techniques for these two processing steps are covered in Sections 3 and 4, respectively.

3. NOISE COMPENSATION FOR ADF

Since the objective function in the form of (7) becomes $J_{n_{ij}} = (1/2)\mathbf{r}^T_{v_{\mathbf{n}_i}\mathbf{v}_{n_j}}\mathbf{r}_{v_{\mathbf{n}_i}\mathbf{v}_{n_j}}$, the presence of noise deteriorates the adaptation performance of (6) which contains bias caused by output noise cross-correlations. As shown in (4), the noise component in output cross-correlation varies as filters $\mathbf{g}_{ij}$ adapt. The time-varying noise effect can be reduced by using an estimate of speech cross-correlation $\mathbf{r}_{v_i\mathbf{v}_j} = \mathbf{r}_{v_{n_i}\mathbf{v}_{\mathbf{n}_j}} - \mathbf{r}_{\eta_i\boldsymbol{\eta}_j}$, that is,

$$J'_{ij} = \frac{1}{2}\left(\mathbf{r}_{v_{\mathbf{n}_i}\mathbf{v}_{n_j}} - \mathbf{r}_{\eta_i\boldsymbol{\eta}_j}\right)^T\left(\mathbf{r}_{v_{\mathbf{n}_i}\mathbf{v}_{n_j}} - \mathbf{r}_{\eta_i\boldsymbol{\eta}_j}\right). \tag{8}$$

Based on (8), the noise-compensated ADF (NC-ADF) is obtained as

$$\mathbf{g}_{ij}(t+1) = \mathbf{g}_{ij}(t) + \mu(t)\left(v_{n_i}(t)\mathbf{v}_{n_j}(t) - \alpha\hat{\mathbf{r}}_{\eta_i\boldsymbol{\eta}_j}(t)\right), \tag{9}$$

where $\hat{\mathbf{r}}_{\eta_i\boldsymbol{\eta}_j}(t)$ is the estimate of output noise cross-correlation, and $0 < \alpha \leq 1$ the discount factor to prevent overcompensation. In the following, $\alpha = 0.9$ is used.

In the current work, for the computation of step-sizes, the VSS technique of [9] is extended to include a compensation of output noise powers. The effect of unequal source energies on filter estimation errors is that the lower the relative strength of the jth source, the higher the estimation error will be for the filter $\mathbf{g}_{ij}$ [9]. To reduce the ADF estimation error caused by unbalanced source energies, step-sizes can be scaled by relative short-term powers of ADF outputs as

$$\mu_{ij}(t) = \frac{\mu(t)\cdot\hat{\sigma}^2_{v_j}(t)}{\hat{\sigma}^2_{\text{av}}(t)}, \tag{10}$$

where the normalizing gain factor $\mu(t)$ was given by [12]

$$\mu(t) = \frac{\gamma}{\left(N\left(\sigma^2_{y_{n1}}(t) + \sigma^2_{y_{n2}}(t)\right)\right)}, \tag{11}$$

with $\sigma^2_{y_{ni}}(t)$ the short-term power of the ith input, and γ $(0 < \gamma < 1)$ the constant gain factor that controls convergence speed. The estimated average speech output power $\hat{\sigma}^2_{\text{av}}(t)$ is

$$\hat{\sigma}^2_{\text{av}}(t) = \frac{\left(\hat{\sigma}^2_{v_1}(t) + \hat{\sigma}^2_{v_2}(t)\right)}{2}. \tag{12}$$

The noise compensation to output power is made by subtracting noise power from the power of noisy ADF output, that is,

$$\hat{\sigma}^2_{v_j} = \hat{r}_{v_jv_j}(0) = \hat{r}_{v_{n_j}v_{n_j}}(0) - \hat{r}_{\eta_j\eta_j}(0), \tag{13}$$

and the output noise power is obtained from (5) as

$$\hat{r}_{\eta_j\eta_j}(0) = \hat{r}_{n_jn_j}(0) - 2\mathbf{g}^T_{ji}\hat{\mathbf{r}}_{n_j\mathbf{n}_i} + \mathbf{g}^T_{ji}\hat{\mathbf{R}}_{\mathbf{n}_i\mathbf{n}_i}\mathbf{g}_{ji}. \tag{14}$$

4. FAST IMPLEMENTATION OF NOISE COMPENSATION AND ADF

4.1. Fast update of compensation terms

Direct computations of noise cross-correlation vectors in NC-ADF adaptation (9) are not feasible for real-time applications since the terms in (4) require matrix-vector multiplications for every time sample. For fixed speaker locations, the changes of ADF filters are in general small within short time intervals (e.g., around 30 milliseconds). The slow change of ADF parameters and the short-term stationarity of input noise make it possible to update compensation terms in a blockwise fashion, reducing the update rate by a factor of K (block-length). To speed up NC-ADF, we first reduce the update rate for compensation terms and then utilize the Toeplitz structures of both the system and the correlation matrices to derive an FFT-based estimation of (4).

The estimate of output bias (4) can be rewritten as

$$\hat{\mathbf{r}}_{\eta_i\boldsymbol{\eta}_j} = \hat{\mathbf{r}}_{n_i\mathbf{n}_j} - \mathbf{a}_{ij} - \mathbf{b}_{ij} + \mathbf{c}_{ij}, \tag{15}$$

with $\mathbf{a}_{ij} = \mathbf{G}_{ji}\hat{\mathbf{r}}_{n_i\tilde{\mathbf{n}}_i}$, $\mathbf{b}_{ij} = \hat{\mathbf{R}}_{\mathbf{n}_j\mathbf{n}_j}\mathbf{g}_{ij}$, $\mathbf{c}_{ij} = \mathbf{G}_{ji}\mathbf{d}_{ij}$, and $\mathbf{d}_{ij} = \hat{\mathbf{R}}_{\tilde{\mathbf{n}}_i\mathbf{n}_j}\mathbf{g}_{ij}$. Computations of $\mathbf{a}_{ij}$ and $\mathbf{c}_{ij}$ share the same structure. The components of vector $\mathbf{a}_{ij}$, that is, $a_{ij}(k)$, $k = 0,\ldots,N-1$, can be expressed as the last N samples, in reversed order, of the convolution $g_{ji}(n)*\xi^a_{ij}(n)$, that is,

$$a_{ij}(k) = g_{ji}(n)*\xi^a_{ij}(n)|_{n=2N-2-k}, \tag{16}$$

where $\xi^a_{ij}(n) = \hat{r}_{n_i\tilde{n}_i}(2N-2-n)$ is the $(2N-1)$-point reverse of $\hat{\mathbf{r}}_{n_i\tilde{\mathbf{n}}_i}$. Similarly, components of $\mathbf{c}_{ij}$ are obtained by $c_{ij}(k) = g_{ji}(n)*\xi^c_{ij}(n)|_{n=2N-2-k}$, with $\xi^c_{ij}(n) = d_{ij}(2N-2-n)$. The vectors $\mathbf{b}_{ij}$ and $\mathbf{d}_{ij}$ also have a similar structure, where $b_{ij}(k) = g_{ij}(n)*\xi^b_{ij}(n)|_{n=k+N-1}$ with $\xi^b_{ij}(n) = \hat{r}_{n_j\tilde{n}_j}(n-N+1)$, and $d_{ij}(k) = g_{ij}(n)*\xi^d_{ij}(n)|_{n=k+N-1}$ with $\xi^d_{ij}(n) = \hat{r}_{n_i\tilde{n}_j}(N-1-n)$. Based on such convolutive expressions, the N-point sequences $a_{ij}(k)$, $b_{ij}(k)$, and $c_{ij}(k)$ can be computed by N_F-point FFTs $(N_F > 2N-1)$. For modularity, the $(2N-1)$-point sequence $d_{ij}(k)$ can be decomposed into two N-point subsequences and computed with two N_F-point FFT-IFFT modules. In this way, all the sequences above only need to be zero-padded to length N_F, because only N-point results are required in each module. The rest points with aliasing are irrelevant and are discarded.

From (13)–(15), the noise-free ADF output powers used in the VSS computation are estimated by

$$\hat{\sigma}^2_{v_j} \approx \mathbf{v}^T_{n_j}\mathbf{v}_{n_j}/N - \hat{r}_{n_jn_j}(0) + 2\mathbf{g}^T_{ji}\hat{\mathbf{r}}_{n_j\mathbf{n}_i} - \mathbf{g}^T_{ji}\mathbf{b}_{ji}. \tag{17}$$

4.2. Fast ADF and NC-FADF

The samplewise procedures of filtering (1) and adaptation (6) of ADF are also modified for a blockwise implementation to enable fast noise compensation. The fast computation of (1) can use the standard overlap-add fast convolution [16] under the approximation that filters are constant within each block.

By using a constant step-size in each block, a block-adaptive procedure for filter update can be obtained. For noise-free ADF, consider the mth block covering samples from t_m to $t_m + K - 1$, and let $\mathbf{g}^m_{ij} = \mathbf{g}_{ij}(t_m)$ be the filters of the current block. After obtaining ADF outputs of the mth block by a fast convolution filtering, the step-size μ^m can be estimated to update filters in the entire current block. By summing up both sides of (6) for $t = t_m,\ldots,t_m + K - 1$,

the new filters for the next block, $\mathbf{g}_{ij}^{m+1} = \mathbf{g}_{ij}(t_m + K)$, can be estimated as

$$\mathbf{g}_{ij}^{m+1} = \mathbf{g}_{ij}^{m} + \mu^m K \hat{\mathbf{r}}_{v_i\mathbf{v}_j}^{m}. \tag{18}$$

The cross-correlation estimate

$$\hat{\mathbf{r}}_{v_i\mathbf{v}_j}^{m} = \hat{\mathbf{r}}_{v_i\mathbf{v}_j}(t_m) = \frac{1}{K}\sum_{k=0}^{K-1} v_i(t_m + k)\mathbf{v}_j(t_m + k) \tag{19}$$

can be computed by an FFT-based fast implementation [16].

Similarly, the blockwise NC-FADF is obtained from (9) as

$$\mathbf{g}_{ij}^{m+1} = \mathbf{g}_{ij}^{m} + \mu_{ij}^{m} K\left(\hat{\mathbf{r}}_{v_{n_i}\mathbf{v}_{\mathbf{n}_j}}^{m} - \alpha\hat{\mathbf{r}}_{\eta_i\boldsymbol{\eta}_j}^{m}\right), \tag{20}$$

where $\hat{\mathbf{r}}_{v_{n_i}\mathbf{v}_{\mathbf{n}_j}}^{m}$ is defined by replacing v_i and v_j with their noisy counterparts in (19), $\hat{\mathbf{r}}_{\eta_i\boldsymbol{\eta}_j}^{m}$ is from (15), and the block step-size μ_{ij}^{m} is computed by (10). The normalization gain factor μ^m in (11) uses ADF input powers that are estimated from the samples of both current and previous blocks. To prevent overcompensation in NC-FADF, $\hat{\sigma}_{v_j}^2$ in (17) is set to zero when negative values occur. The denominator in (10) is also added a small positive number to avoid divide-by-zeros. Triangular windows $w(n) = (N - n)/N$, $n = 0,\ldots,N-1$, are applied to both correlation estimate $\hat{\mathbf{r}}_{\eta_i\boldsymbol{\eta}_j}^{m}$ and ADF adaptation vectors to prevent instability.

The overlap-add method requires $N \le K \le 2N$. When $K = N$ and the FFT length $N_F = 2N$, the computation of $2N$-point FFTs is distributed to the block of length N, resulting in a complexity of $O(\log N)$ per time-sample for NC-FADF, in contrast to $O(N^2)$ for a direct estimation of NC terms that are required by matrix-vector multiplications.

5. ADAPTIVE ENHANCEMENT OF SEPARATED SPEECH

5.1. Tracking of ADF output noise autocorrelations

Although NC-FADF improves the speech separation performance in noise, the separation outputs $\mathbf{v}_{\mathbf{n}_i}$ are still contaminated by noise. Thus, a speech enhancement postprocessing should be integrated with ADF to reduce noise in each output. To do so, we need to track the time-varying output noise statistics as filters evolve from block to block by a fast computation of (5). Similar to the derivations of (15), we obtain autocorrelation of ADF output noise for the mth block:

$$\hat{\mathbf{r}}_{\eta_i\boldsymbol{\eta}_i}^{m} = \hat{\mathbf{r}}_{n_i\mathbf{n}_i} - \mathbf{a}_{ii}^{m} - \mathbf{b}_{ii}^{m} + \mathbf{c}_{ii}^{m}, \tag{21}$$

where $\mathbf{a}_{ii}^{m} = \mathbf{G}_{ij}^{m}\hat{\mathbf{r}}_{n_i\tilde{\mathbf{n}}_j}$, $\mathbf{b}_{ii}^{m} = \hat{\mathbf{R}}_{\mathbf{n}_i\mathbf{n}_j}\mathbf{g}_{ij}^{m}$, $\mathbf{c}_{ii}^{m} = \mathbf{G}_{ij}^{m}\mathbf{d}_{ii}^{m}$, and $\mathbf{d}_{ii}^{m} = \hat{\mathbf{R}}_{\tilde{\mathbf{n}}_j\mathbf{n}_j}\mathbf{g}_{ij}^{m}$. Since input noise is stationary, its auto- and cross-correlations can be measured a priori during a speech inactive period. The fast mappings from input noise correlations to output noise autocorrelation, depending only on current system parameters $\mathbf{g}_{ij}^{m}$'s and $\mathbf{G}_{ji}^{m}$'s, are implemented as fast convolutions of the following signal sequences:

$$\begin{aligned}
a_{ii}^{m}(k) &= g_{ij}^{m}(n) * \xi_{ii}^{a}(n)|_{n=2N-2-k},\\
c_{ii}^{m}(k) &= g_{ij}^{m}(n) * \xi_{ii}^{c}(n)|_{n=2N-2-k},\\
b_{ii}^{m}(k) &= g_{ij}^{m}(n) * \xi_{ii}^{b}(n)|_{n=k+N-1},\\
d_{ii}^{m}(k) &= g_{ij}^{m}(n) * \xi_{ii}^{d}(n)|_{n=k+N-1},
\end{aligned} \tag{22}$$

where $\xi_{ii}^{a}(n) = \hat{r}_{n_i\tilde{n}_j}(2N-2-n)$, $\xi_{ii}^{c}(n) = d_{ij}^{m}(2N-2-n)$, $\xi_{ii}^{b}(n) = \hat{r}_{n_i\tilde{n}_j}(N-1-n)$, and $\xi_{ii}^{d}(n) = \hat{r}_{n_j\tilde{n}_j}(N-1-n)$.

5.2. Enhancement of separated speech

Utilizing the adaptively estimated noise statistics $\hat{\mathbf{r}}_{\eta_i\boldsymbol{\eta}_i}^{m}$, many algorithms can be considered for postenhancement of ADF outputs. The time domain constrained (TDC) type of the generalized subspace (GSub) method [17] is tested due to its ability to handle colored noise. The TDC-GSub processing is applied to every block of ADF outputs, where for the mth block it requires the noise autocorrelation matrix $\mathbf{R}_{\boldsymbol{\eta}_i\boldsymbol{\eta}_i}^{m}$, which can be constructed by forming a symmetric Teoplitz matrix from the output autocorrelation vector in (21). Specifically, $\hat{\mathbf{r}}_{\eta_i\boldsymbol{\eta}_i}^{m}$ constitutes the first column and the first row of $\mathbf{R}_{\boldsymbol{\eta}_i\boldsymbol{\eta}_i}^{m}$. Another piece of information that the TDC-GSub algorithm takes is the autocorrelation matrix of the noisy ADF output, $\mathbf{R}_{\mathbf{v}_{n_i}\mathbf{v}_{n_i}}$, which is estimated from ADF outputs of the current block. The TDC-GSub processing is performed on each nonoverlapping subframe of length $L = 40$ and the major steps are the same as in [17].

Step 1. Do eigendecomposition $\boldsymbol{\Sigma}_i\mathbf{U} = \mathbf{U}\boldsymbol{\Lambda}$ for matrix $\boldsymbol{\Sigma}_i = (\mathbf{R}_{\boldsymbol{\eta}_i\boldsymbol{\eta}_i}^{m})^{-1}\mathbf{R}_{\mathbf{v}_{n_i}\mathbf{v}_{n_i}} - \mathbf{I}$, with $\boldsymbol{\Lambda} = \mathrm{diag}[\lambda^1,\ldots,\lambda^M,0,\ldots,0]$, and M is the number of positive eigenvalues.

Step 2. Compute the optimal estimator $\mathbf{H} = \mathbf{U}^{-T}\mathrm{diag}[\alpha_1,\ldots,\alpha_M,0,\ldots,0]\mathbf{U}^T$, where the eigendomain filtering gains are obtained by $\alpha_k = \lambda^k/(\lambda^k + \beta)$, $k = 1,\ldots,M$, and β is determined from

$$\beta = \begin{cases} 5 & \mathrm{SNR}_{\mathrm{dB}} \le -5,\\ 1 & \mathrm{SNR}_{\mathrm{dB}} \ge 20,\\ \dfrac{4.2 - (\mathrm{SNR}_{\mathrm{dB}})}{6.25} & \text{otherwise,} \end{cases} \tag{23}$$

with $\mathrm{SNR}_{\mathrm{dB}} = 10\log_{10}(\sum_{k=1}^{M}\lambda^k/L)$.

Step 3. Enhance the ith ADF output by $\hat{\mathbf{v}}_i^m = \mathbf{H}\mathbf{v}_{n_i}^m$.

The computations of matrix inversion, multiplication, and eigendecomposition become acceptable when a small value is used for L (2.5 milliseconds). In addition, a measure is taken to speed up TDC-GSub by utilizing the short-term stationary property of separated speech signals $\mathbf{v}_{n_i}$'s. Within 20 milliseconds, the variations of $\mathbf{R}_{\mathbf{v}_{n_i}\mathbf{v}_{n_i}}$'s are relatively small, obviating the need for updating their eigendecompositions in every subframes. In practice, the computation rate for both steps 1 and 2 are thus reduced to every 12.5 milliseconds, without introducing significant degradations.

Table 1: Counts of real multiplications.

Computation	Complexity estimates		Gain
	Direct	Fast	
ADF filtering	$2N$	$\frac{(8N_F\log_2 N_F)}{K}$	$\frac{N}{(8\log_2(2N))}$
ADF adapt	$2N$	$\frac{(8N_F\log_2 N_F)}{K}$	$\frac{N}{(8\log_2(2N))}$
$\hat{\mathbf{r}}_{\eta_i\eta_j}$'s	$10N^2$	$\frac{(40N_F\log_2 N_F)}{K}$	$\frac{N^2}{(8\log_2(2N))}$
$\hat{\mathbf{r}}_{\eta_i\eta_i}$'s	$10N^2$	$\frac{(40N_F\log_2 N_F)}{K}$	$\frac{N^2}{(8\log_2(2N))}$
SS	$\frac{8K_F\log_2 K_F}{K}$, $(K_F \geq K)$		
TDC-GSub	$O(L^2)$		

6. COMPLEXITY ANALYSIS

The complexities of the major computation steps in terms of the average number of real multiplications per time-sample are listed in Table 1. Trivial computation overheads are ignored. The gain of the fast over the direct implementations are evaluated for $N = K$ and $N_F = 2N$. The counts for FFT are based on the regular radix-2 method. It is possible to further reduce the complexities of computations. In Table 1, only a coarse complexity estimate is made for TDC-GSub, based on direct implementations of matrix operations. Faster computation techniques for TDC-GSub and complexity analyses are out of the scope of this paper.

7. EXPERIMENTS

7.1. Experimental data and setup

Speech mixtures were generated from a convolution of clean speech sources in TIMIT database with real acoustic impulse responses measured in a room of reverberation time $T_{[60]} = 0.3$ second [18]. The speakers were approximately 2 m away from two microphones that were mounted 21 cm apart on a circular array of radius 15 cm, and the distance between the two speakers was 2.6 m. The target speech was sampled at 16 kHz and had 40 sentences from 4 speakers (faks0, felc0, mdab0, mreb0). The competing speech contained randomly selected TIMIT sentences. Both simulated and real diffuse noise conditions were tested. The simulated noise is speech-shaped and was generated by the following procedure:

$$\begin{aligned} n_1(t) &= 0.65\sum_{k=1}^{2} a_k^{(1)} n_1(t-k) + 0.35n_2(t) + \varepsilon_1(t), \\ n_2(t) &= 0.6\sum_{k=1}^{3} a_k^{(2)} n_2(t-k) + 0.4n_1(t) + \varepsilon_2(t), \end{aligned} \tag{24}$$

where $\varepsilon_i(t)$'s are white Gaussian excitations and $a_k^{(i)}$'s are linear prediction coefficients (LPC) estimated from clean

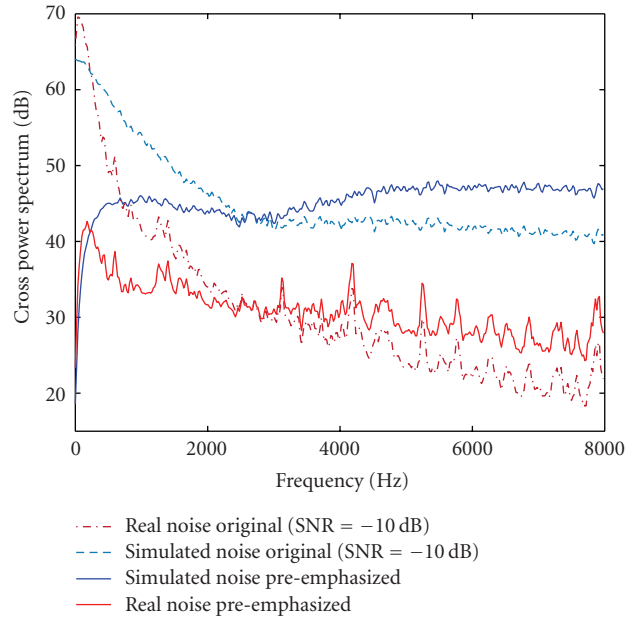

Figure 3: Cross-power spectra of two types of diffuse noises: simulated speech-shaped noise and real lab noise.

TIMIT data. Real diffuse noises were recorded in a computer lab with a pair of omnidirectional microphones placed in the center of the lab, where the microphones were the same distance apart as that of the array microphone pair. Ventilation and air-conditioning systems and 8 desktop workstations were working simultaneously, generating diffuse noises that fit the stationary assumption. As a default setting, a 2-second speech inactive segment immediately preceding the speech was used to estimate input noise statistics. Figure 3 illustrates the cross-power spectra for both types of noises.

The basic setup for ADF was $N = 400$ and $\gamma = 0.01$ and the separation filters were initialized with zeros, representing a totally blind condition (if certain prior knowledge of

Table 2: Gain in TIR (dB) (simulated speech-shaped noise).

Original SNR	Pre-emphsized SNR(y_1, y_2)	Baseline v_{n_1}, v_{n_2}	FADF v_{n_1}, v_{n_2}	NC-FADF v_{n_1}, v_{n_2}
3 dB	0.2, −1.3	1.7, 2.1	1.7, 2.0	6.3, 7.5
9 dB	6.2, 4.7	3.0, 3.9	2.8, 3.6	8.5, 9.2
15 dB	12.2, 10.7	4.7, 5.6	4.4, 5.2	10.0, 9.9
21 dB	18.2, 16.7	6.3, 6.8	5.9, 6.3	10.7, 10.1
27 dB	24.2, 22.7	7.5, 7.6	6.9, 6.9	11.0, 10.2

Table 3: Output SNR (dB) (simulated speech-shaped noise).

Original SNR	Pre-emphsized SNR(y_1, y_2)	Baseline v_{n_1}, v_{n_2}	FADF v_{n_1}, v_{n_2}	NC-FADF v_{n_1}, v_{n_2}
3 dB	0.2, −1.3	−0.3, −1.5	−0.1, −1.3	−0.8, −2.9
9 dB	6.2, 4.7	5.3, 3.4	5.5, 3.7	4.9, 2.9
15 dB	12.2, 10.7	10.8, 8.6	11.0, 8.9	10.6, 8.7
21 dB	18.2, 16.7	16.3, 14.0	16.5, 14.3	16.5, 14.6
27 dB	24.2, 22.7	22.0, 19.7	22.2, 19.9	22.3, 20.5

the acoustic paths can be incorporated into the initial separation filters, then ADF separation performance can be improved, especially in severe noise). In all cases, a pre-emphasis $(1 - z^{-1})$ was applied to speech mixtures to remove the 6-dB/octave tilt of speech long-term spectrum and to reduce eigenvalue dispersion for faster convergence [10]. Pre-emphasis enhances perceptually important speech components, and it also alters input noise properties as well as the relative strengths of noise and speech measured in signal-to-noise ratio (SNR): $\text{SNR} = 10\log_{10}(P_S/P_N)$, where P_S is the power of the clean speech mixture signal, and P_N is the power of the noise component. In fact, the simulated speech-shaped noise spectrum was flattened by pre-emphasis, resulting in a loss of SNR of approximately 3 dB. On the other hand, the recorded diffuse noise retained a significant amount of coloration and spatial correlation after pre-emphasis that increased SNR by 12 dB through suppressing strongly correlated low-frequency noise components (see Figure 3). In subsequent discussions, SNR and target-to-interference ratio (TIR) refer to those evaluated on pre-emphasized input and output components, where TIR is defined as $10\log_{10}(P_T/P_I)$, with P_T the power of target speech and P_I the power of interference speech component. For FADF and NC-FADF, the block length was $K = 400$ and the FFT length was $N_F = 1024$. Since VSS without NC would corrupt adaptation at high levels of noise, it was not applied to ADF (6) and FADF. In the appendix, more details are provided for the definitions of SNR and TIR.

7.2. Speech separation performance

The separation performances were evaluated by system gains in TIR, defined as $\text{TIR}_{\text{output}} - \text{TIR}_{\text{input}}$. In Tables 2 and 4, the TIR gains of NC-FADF outperform those of the baseline for both types of noises, at the cost of a slightly decreased SNR, as shown in Tables 3 and 5. Since FADF is a fast and approximate implementation of the baseline ADF, it suffered a slight degradation from the baseline and showed occasional instability in the iterative estimations of separation filters. The TIR gain values in Tables 2 and 4 are computed from the noise-free components in the noisy outputs v_{n_1} and v_{n_2}. It is interesting to observe that under severe noise conditions, for example SNR= − 12 dB (original), the baseline ADF actually increased SNR. This is consistent with the analysis in [13] that in correlated noises, the baseline ADF tends to divert from speech separation to noise cancellation. Tables 3 and 5 show that the NC algorithm can force ADF to focus on speech separation, rather than noise cancellation.

7.3. Speech enhancement and phone recognition

Experiments were conducted to compare the cases of using NC-FADF or FADF, with and without adaptive speech enhancements. Since SNR was altered by pre-emphasis differently for simulated and real diffuse noises, the range of initial SNRs were chosen differently for these two cases so that the input target speech had the same SNRs after pre-emphasis. After adaptive online speech enhancement, a de-emphasis $1/(1 - 0.98z^{-1})$ was applied to the enhanced speech.

The overall enhancement of target speech against the effects of both interfering jammer and noise are shown by the target-to-interference-and-noise ratio (TINR) in Figures 4 and 5, where TINRs are defined in the appendix for the input, the separation output, and the separation output with noise reduction. It is seen that NC-FADF outperformed FADF in both types of noises under almost all SNR conditions. At high SNRs, the TINR improvements come mainly from the separation processing of NC-FADF or FADF, as speech jammer is the dominant problem. The larger TINR gains obtained by NC-FADF over FADF were also attributed to its

Table 4: Gain in TIR (dB) (real diffuse noise).

Original SNR	Pre-emphsized SNR(y_1, y_2)	Baseline v_{n_1}, v_{n_2}	FADF v_{n_1}, v_{n_2}	NC-FADF v_{n_1}, v_{n_2}
−12 dB	0.2, 0.3	3.1, 3.9	3.1, 3.6	7.0, 8.2
−6 dB	6.2, 6.3	4.2, 5.6	1.5, 5.4	9.2, 9.3
0 dB	12.2, 12.3	6.3, 7.7	6.2, 6.9	10.4, 9.9
6 dB	18.2, 18.3	7.7, 7.9	7.2, 7.3	10.9, 10.1
12 dB	24.2, 24.3	8.1, 8.1	7.5, 7.4	11.0, 10.2

Table 5: Output SNR (dB) (real diffuse noise).

SNR	SNR(y_1, y_2)	v_{n_1}, v_{n_2}	v_{n_1}, v_{n_2}	v_{n_1}, v_{n_2}
−12 dB	0.2, 0.3	3.8, 2.4	4.2, 3.2	1.6, −1.0
−6 dB	6.2, 6.3	6.1, 5.9	6.5, 4.9	6.7, 4.5
0 dB	12.2, 12.3	12.3, 11.4	12.8, 11.6	12.4, 10.2
6 dB	18.2, 18.3	17.9, 16.4	18.0, 16.5	18.2, 16.0
12 dB	24.2, 24.3	23.4, 21.7	23.6, 21.9	24.1, 22.0

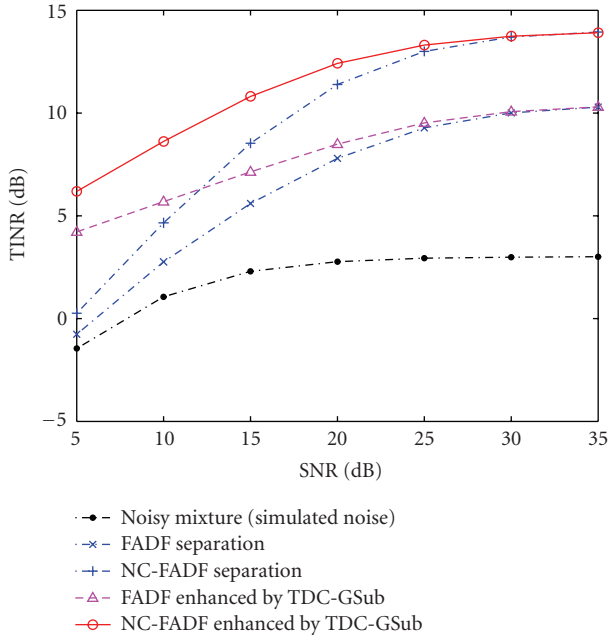

Figure 4: Target-to-interference-and-noise ratio (simulated noise).

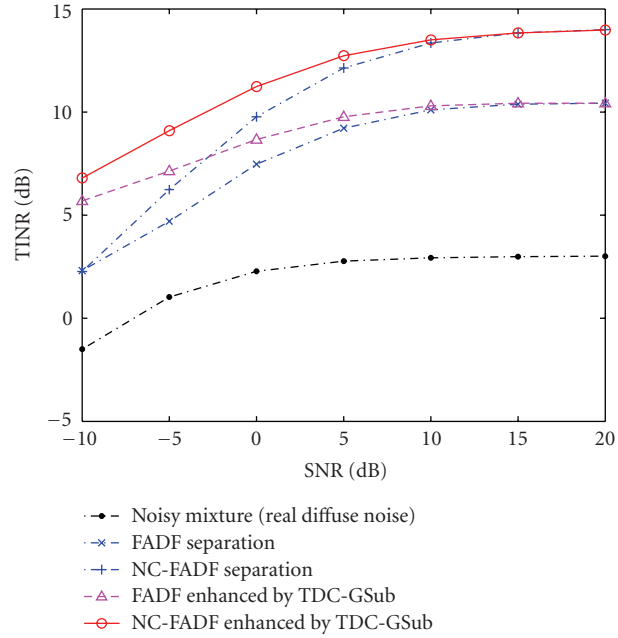

Figure 5: Target-to-interference-and-noise ratio (real diffuse noise).

use of the variable step-size adaptation defined in (10) and (12)–(14), while without noise compensation the VSS was unavailable to FADF. This advantage of using variable step size over fixed step size in ADF adaptation is consistent with the findings in [9]. At low SNRs, the TINR improvement is mainly contributed by the suppression of the noise components, and in the real diffuse noise, the separation processing had a stronger effect on TINR improvement than in the simulated noise. When the SNR is very low, where the energy of speech mixture is dominated by the noise, the TIR improvement (between target and jammer speech) by NC-FADF contributed less to the overall TINR gains, and here the enhancement processing by TDC-GSub improved TINR greatly in both types of noises.

Phone recognitions were performed by using HTK toolkit [19] for the noisy mixture, the noisy separated speech, and the enhanced separated speech of the target. The speech signals were represented by sequences of feature vectors obtained from 50% overlapped short-time analysis window of 20 milliseconds. Each feature vector consisted of 13 cepstral coefficients and their first- and second-order time derivatives. Both training and test data from TIMIT

database were processed with spectral mean subtraction. Hidden Markov modeling (HMM) was used for 39 context independent phone units, defined by the phone grouping scheme of [20]. Each phone unit had 3 emission states, with state observation probabilities modeled by size-8 Gaussian mixture densities. Phone bigram was used as "language model."

The phone accuracy results in simulated and real diffuse noise cases are shown in Figures 6 and 7, respectively. The upper limit of phone accuracy was 46.5%, which was obtained from the target speech separated from the clean speech mixtures by ADF. It is observed that when SNR is low or moderate, the adaptive enhancement techniques significantly improved the phone recognition accuracy of the separation outputs. Similar to the TINR results, at high SNRs, the improvement to phone accuracy comes mainly from speech separation, where NC-FADF is significantly better than FADF. Comparative experimental results were also generated for the proposed approach of applying TDC-GSub as a postprocessor after FADF (FADF enhanced by TDC-GSub postprocessing) and the apparent alternative of using TDC-GSub as a preprocessor prior to FADF (FADF after TDC-GSub Preprocessing). It is seen that the former performed better than the latter, especially in real diffuse noise. In general, the combination of NC-FADF with TDC-GSub postprocesing achieved the highest accuracy performance.

7.4. *Sensitivity to noise estimation*

In real applications, there are scenarios where the speech inactive periods are short, which would reduce the reliability of noise statistic estimation. It is therefore of interest to evaluate the feasibility of the proposed NC-FADF algorithm when the input noise statistics are estimated from short data segments. For this purpose, an experiment was performed to vary the speech inactive period from 0.5 second through 2.5 seconds, and the noise statistics computed from the different periods were used by NC-FADF followed by TDC-GSub to perform speech separation and enhancement. The test results confirmed that for the two types of noises investigated in the current work, there is no significant difference in the overall system performance over this range of speech-inactive intervals. Figure 8 illustrates the phone recognition performance versus the speech inactive interval lengths in real diffuse noise. It is seen that except for a performance drop when the speech inactive length was 0.5 second, phone accuracy remained essentially the same for all other speech inactive lengths. In simulated noise, the accuracy performance remained essentially the same for all of the speech inactive lengths, including the 0.5 second case. In general, in an online system a voice activity detection module is needed to identify speech inactive periods, and for fast-varying nonstationary input noises, robust algorithms are needed to estimate time-varying noise properties with adaptive memory lengths. Although this issue is practically important, it is out of the scope of the current work.

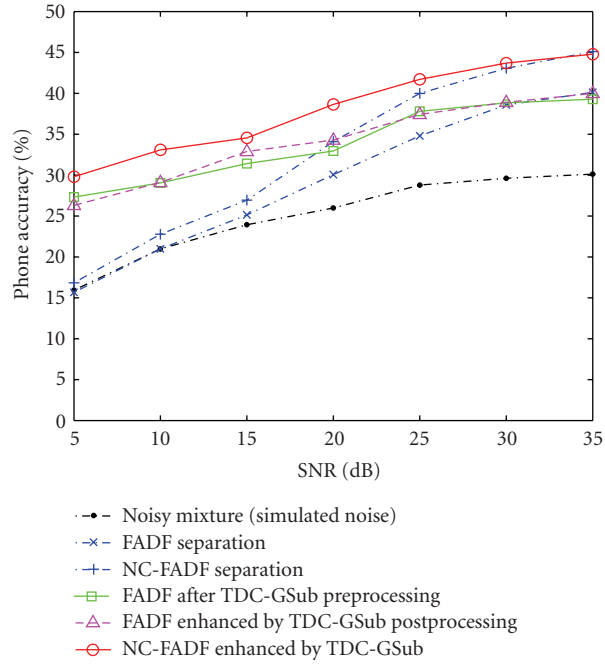

Figure 6: Phone accuracies (simulated noise).

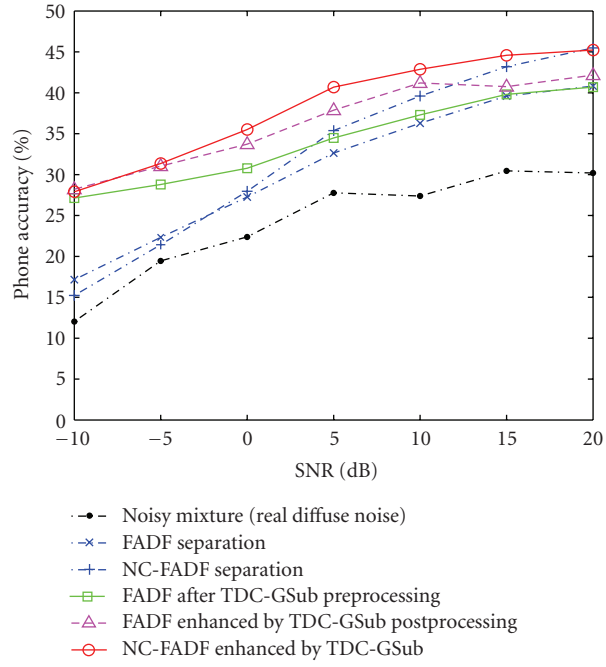

Figure 7: Phone accuracies (real diffuse noise).

8. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented methods of noise compensation and adaptive speech enhancement to improve the performances of ADF speech separation in diffuse noise. Fast implementations for ADF and noise compensation have

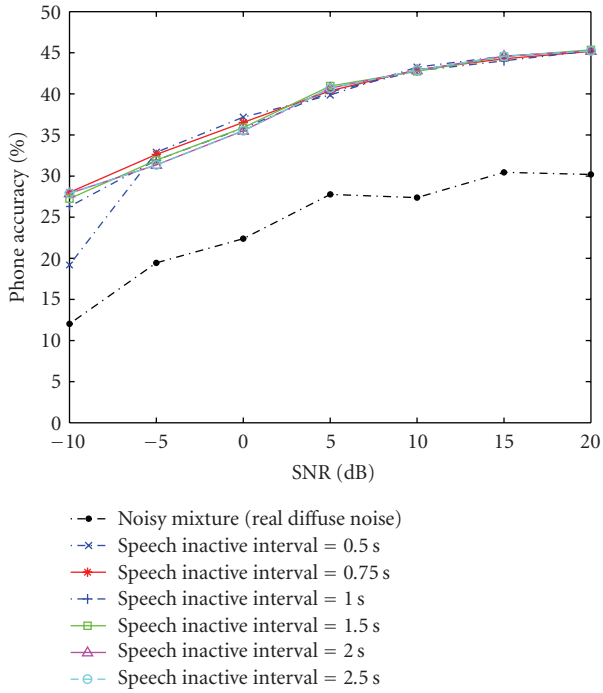

Figure 8: Effects of speech inactive interval length on noise estimation for phone recognition: NC-FADF with TDC-GSub in real diffuse noise.

been made that warrant real-time online applications. FADF has achieved performance comparable to that of ADF with a much faster speed. NC-FADF significantly improved the separation performance for speech mixtures in diffuse noise, and the integration of NC-FADF with speech enhancement significantly improved phone recognition accuracies in separated speech. Future investigations may include other enhancement algorithms and noise-reduction implementations for a more streamlined integration with the NC-FADF procedure.

APPENDIX

DEFINITIONS OF SNR, TIR, AND TINR

Since the ADF filtering model (1) is linear, the superposition principle holds, that is, its output components of target, interference, and noise can be computed separately from its respective input components. Unlike the linear model of ADF, the speech enhancement module is nonlinear and its output components cannot be separately estimated from its individual input components. Therefore, the separate computation of output TIR and SNR are not feasible for the speech enhancement module. Instead, TINRs can be estimated by taking the signal energies other than the original target as the sum of noise and interference signals. The computations of SNR, TIR, and TINR are defined below with respect to channel 1 (the definitions are similar for channel 2):

$$\begin{aligned}
\mathrm{SNR}_{y_1} &= 10\log_{10}\left(\frac{P_{y_1}}{P_{n_1}}\right),\\
\mathrm{SNR}_{v_1} &= 10\log_{10}\left(\frac{P_{v_1}}{P_{\eta_1}}\right),\\
\mathrm{TIR}_{y_1} &= 10\log_{10}\left(\frac{P_{y_{s_1}}}{P_{y_{s_2}}}\right),\\
\mathrm{TIR}_{v_1} &= 10\log_{10}\left(\frac{P_{v_{s_1}}}{P_{v_{s_2}}}\right),\\
\mathrm{TINR}_{y_1} &= 10\log_{10}\left(\frac{P_{y_{s_1}}}{P_{(y_{s_2}+n_1)}}\right),\\
\mathrm{TINR}_{v_1} &= 10\log_{10}\left(\frac{P_{v_{s_1}}}{P_{(v_{s_2}+\eta_1)}}\right),\\
\mathrm{TINR}_{\hat{v}_1} &= 10\log_{10}\left(\frac{P_{v_{s_1}}}{P_{(\hat{v}_1-v_{s_1})}}\right).
\end{aligned} \tag{A.1}$$

At ADF input, P_{y_1} and P_{n_1} are the powers of the clean mixture and the noise components, respectively; $P_{y_{s_1}}$ and $P_{y_{s_2}}$ are the powers of the target and the interference speech signals, respectively; $y_{s_2} + n_1 = y_{n_1} - y_{s_1}$ is the sum of interference speech and noise. At ADF output, P_{v_1} and P_{η_1}, $P_{v_{s_1}}$ and $P_{v_{s_2}}$, and $v_{s_2} + \eta_1 = v_{n_1} - v_{s_1}$ are the counterparts of the above components at ADF input. The component $\hat{v}_1$ is the output speech after enhancement processing.

REFERENCES

[1] A. Hyvarinen, J. Karhunen, and E. Oja, *Independent Component Analysis*, John Wiley & Sons, New York, NY, USA, 2001.

[2] R. Aichner, H. Buchner, and W. Kellermann, "Convolutive blind source separation for noisy mixtures," in *Proceedings of Joint Meeting of the German and the French Acoustical Societies (CFA/DAGA '04)*, Strasbourg, France, March 2004.

[3] S. C. Douglas, A. Cichocki, and S. Amari, "Bias removal technique for blind source separation with noisy measurements," *Electronics Letters*, vol. 34, no. 14, pp. 1379–1380, 1998.

[4] R. Hu and Y. Zhao, "Adaptive decorrelation filtering algorithm for speech source separation in uncorrelated noises," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '05)*, vol. 1, pp. 1113–1116, Philadelphia, Pa, USA, March 2005.

[5] R. Balan, J. Rosca, and S. Richard, "Scalable non-square blind separation in the presence of noise," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '03)*, vol. 5, pp. 293–296, Hong Kong, April 2003.

[6] S. Araki, S. Makino, R. Mukai, and H. Saruwatari, "Equivalence between frequency domain blind source separation and frequency domain adaptive beamforming," in *Proceedings of the 7th European Conference on Speech Communication and Technology (Eurospeech '01)*, vol. 4, pp. 2595–2598, Aalborg, Denmark, September 2001.

[7] S. Araki, R. Mukai, S. Makino, T. Nishikawa, and H. Saruwatari, "The fundamental limitation of frequency domain blind source separation for convolutive mixtures of speech," *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 2, pp. 109–116, 2003.

[8] F. Asano, S. Hayamizu, T. Yamada, and S. Nakamura, "Speech enhancement based on the subspace method," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 5, pp. 497–507, 2000.

[9] R. Hu and Y. Zhao, "Variable step size adaptive decorrelation filtering for competing speech separation," in *Proceedings of the 9th European Conference on Speech Communication and Technology (Eurospeech '05)*, vol. 1, pp. 2297–2300, Lisbon, Portugal, September 2005.

[10] Y. Zhao, R. Hu, and X. Li, "Speedup convergence and reduce noise for enhanced speech separation and recognition," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, no. 4, pp. 1235–1244, 2006.

[11] E. Weinstein, M. Feder, and A. V. Oppenheim, "Multi-channel signal separation by decorrelation," *IEEE Transactions on Speech and Audio Processing*, vol. 1, no. 4, pp. 405–413, 1993.

[12] K.-C. Yen and Y. Zhao, "Adaptive co-channel speech separation and recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 2, pp. 138–151, 1999.

[13] K. Yen, J. Huang, and Y. Zhao, "Co-channel speech separation in the presence of correlated and uncorrelated noises," in *Proceedings of the 6th European Conference on Speech Communication and Technology (Eurospeech '99)*, pp. 2587–2589, Budapest, Hungary, September 1999.

[14] R. Hu and Y. Zhao, "Fast noise compensation for speech separation in diffuse noise," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '06)*, vol. 5, pp. 865–868, Toulouse, France, May 2006.

[15] R. Hu and Y. Zhao, "Adaptive speech enhancement for speech separation in diffuse noise," in *Proceedings of the 9th International Conference on Spoken Language Processing (INTERSPEECH '06)*, pp. 2618–2621, Pittsburgh, PA, USA, September 2006.

[16] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1989.

[17] Y. Hu and P. C. Loizou, "A generalized subspace approach for enhancing speech corrupted by colored noise," *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 4, pp. 334–341, 2003.

[18] "RWCP Sound Scene Database in Real Acoustic Environments," ATR Spoken Language Translation Research Lab, Japan, 2001.

[19] J. Odell, D. Ollason, V. Valtchev, S. Young, D. Kershawl, and P. Woodland, "HTK Speech Recognition Toolkit," 1999, http://htk.eng.cam.ac.uk/docs/docs.shtml.

[20] K.-F. Lee and H.-W. Hon, "Speaker-independent phone recognition using hidden markov models," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 11, pp. 1641–1648, 1989.