DNN-FREE LOW-LATENCY ADAPTIVE SPEECH ENHANCEMENT BASED ON FRAME-ONLINE BEAMFORMING POWERED BY BLOCK-ONLINE FASTMNMF

Aditya Arie Nugraha[1] *Kouhei Sekiguchi*[1] *Mathieu Fontaine*[3,1] *Yoshiaki Bando*[4,1] *Kazuyoshi Yoshii*[2,1]

[1]Center for Advanced Intelligence Project (AIP), RIKEN, Japan
[2]Graduate School of Informatics, Kyoto University, Japan
[3]LTCI, Télécom Paris, Institut Polytechnique de Paris, France
[4]National Institute of Advanced Industrial Science and Technology (AIST), Japan

2022 International Workshop on Acoustic Signal Enhancement (IWAENC) | 978-1-6654-6867-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/IWAENC53105.2022.9914729

ABSTRACT

This paper describes a practical dual-process speech enhancement system that adapts environment-sensitive frame-online beamforming (front-end) with help from environment-free block-online source separation (back-end). To use minimum variance distortionless response (MVDR) beamforming, one may train a deep neural network (DNN) that estimates time-frequency masks used for computing the covariance matrices of sources (speech and noise). Backpropagation-based runtime adaptation of the DNN was proposed for dealing with the mismatched training-test conditions. Instead, one may try to directly estimate the source covariance matrices with a state-of-the-art blind source separation method called fast multichannel non-negative matrix factorization (FastMNMF). In practice, however, neither the DNN nor the FastMNMF can be updated in a frame-online manner due to its computationally-expensive iterative nature. Our DNN-free system leverages the posteriors of the latest source spectrograms given by block-online FastMNMF to derive the current source covariance matrices for frame-online beamforming. The evaluation shows that our frame-online system can quickly respond to scene changes caused by interfering speaker movements and outperformed an existing block-online system with DNN-based beamforming by 5.0 points in terms of the word error rate.

Index Terms— speech enhancement, beamforming, blind source separation, automatic speech recognition

1. INTRODUCTION

In real environments, speech enhancement methods must be adaptive to variations in sound scenes caused by environmental changes or movements of the sound sources [1–3]. While it is important to successfully extract the speech of interest, having a low computational cost can be critical for downstream tasks that demand low-latency outputs, such as automatic speech recognition (ASR) for augmented reality applications aiming at natural human-machine interaction.

This work was supported by JSPS KAKENHI Nos. 19H04137, 20K19833, and 20K21813.

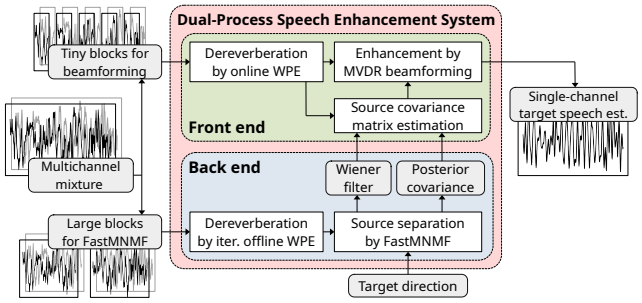

Fig. 1. The proposed low-latency speech enhancement system consisting of a frame-online front end (beamforming) informed by a block-online back end (FastMNMF).

Beamforming is a computationally-efficient multichannel source separation technique that can extract a single-channel signal coming from a target direction when an accurate steering vector or well-estimated source covariance matrices are given [3–6]. Deep neural networks (DNNs) have been popular for estimating these source covariance matrices [6–11], but these DNNs may have limited performance when the actual test environment is not covered by the training data. In contrast, multichannel blind source separation (BSS) methods, such as multichannel non-negative matrix factorization (MNMF) [12] and FastMNMF [13, 14], are expected to perform well in any environment by optimizing the parameters of the assumed source probability distributions. However, these methods require sufficient data and have a relatively high computational cost.

Semi-blind or blind source separation method has been combined with beamforming. Beamformers have been derived in a block-online processing manner using the target source mask estimated as the posterior of a complex Gaussian mixture model (cGMM) [15] or a complex angular central Gaussian mixture model (cACGMM) [16], or the source covariance matrices obtained by MNMF [17]. These systems run the estimation of mask or covariance matrices and the beamforming sequentially on the same block of data. Thus, the system latency is limited by the block size required by cGMM, cACGMM, or MNMF to provide reliable estimates. FastMNMF, which has been shown to outperform MNMF, has also been used as the

978-1-6654-6867-1/22/$31.00 ©2022 IEEE

back end of a dual-process adaptive online speech enhancement system [11] whose front end runs minimum variance distortionless response (MVDR) beamforming [5] with a DNN mask estimator [6]. The DNN mask estimator is periodically updated using the separated speech signals obtained by FastMNMF, which carry out informed source separation given the target directions, to adapt to the test environment.

This paper proposes a dual-process speech enhancement system whose front end performs a responsive adaptive online MVDR beamforming by exploiting *the parameters of the source posterior distributions*, i.e., the Wiener filters and the covariance matrices, estimated by the back-end FastMNMF, as shown in Figure 1. Using those Wiener filters and source posterior covariance matrices, we compute frame-wise second-order raw moments given the current observed mixture and accumulate them using exponential moving averages (EMAs) to obtain block-wise source covariance matrices for the beamformer. Consequently, the estimation of the source covariance matrix relies on FastMNMF, instead of a DNN-based mask estimator [11]. Directly using the average source covariance matrices estimated by FastMNMF in a way similar to [17] is also possible. However, when the back end processes a significantly larger data block than the front end, the front end processes many blocks using the same beamformer while waiting for the back end to provide new source covariance matrices. Our proposed system is thus more preferable because it can promptly respond to the sound scene changes.

The evaluation used multiple sequences of mixtures, in which the interfering speaker locations are different in separate mixtures. Our system outperformed DNN-based beamforming [11] in terms of word error rate (WER) by 5.0 points using frame-online processing with a total latency of 22 ms.

2. PROPOSED SYSTEM

Let $\mathbf{x}_{ft} \in \mathbb{C}^M$ be the short-time Fourier transform (STFT) coefficients at frequency $f \in [1, F]$ and time frame $t \in [1, T]$ of the observed multichannel mixture signal captured by M microphones and $\mathbf{x}_{nft} \in \mathbb{C}^M$ be the STFT coefficients of the so-called multichannel image of source $n \in [1, N]$, where F is the number of frequency bins, T is the total number of time frames, and N is the number of sources. The source images are assumed to sum to the observed mixture as $\mathbf{x}_{ft} = \sum_{n=1}^{N} \mathbf{x}_{nft}$. Given $\mathbf{X} \triangleq \{\mathbf{x}_{ft} | \forall f, \forall t\}$, BSS generally estimates $\forall n, \mathbf{X}_n \triangleq \{\mathbf{x}_{nft} | \forall f, \forall t\}$. In this paper, our dual-process adaptive online speech enhancement system obtains the single-channel signal estimate $\{s_{n'ft} | \forall f, \forall t\}$ of target source n' for ASR purpose.

The dual-process speech enhancement system executes a back end (Sect. 2.1) and a front end (Sect. 2.2) in parallel in a block-online processing manner, where each block is a subset of $\mathbf{X}$ consisting of a sequence of multiple time frames, as in [11]. At a time, the back end processes $\mathbf{X}_i^{\text{BSS}} \subset \mathbf{X}$ composed of T^{BSS} frames with i is the block index, while the front end processes $\mathbf{X}_j^{\text{BF}} \subset \mathbf{X}$ composed of T^{BF} frames with j is the block index. T^{BSS} can be large enough to provide reliable statistics required for good BSS performance, while T^{BF} can be small when low-latency outputs are expected.

The proposed system is similar to the system in [11]. Both systems basically use the same back-end FastMNMF. However, the front ends use different ways to obtain the source covariance matrices required to derive MVDR beamformer.

2.1. Back End

As in [11], the back end operates given a block of data $\mathbf{X}_i^{\text{BSS}}$ and one or more target directions. Offline iterative dereverberation [18] is first performed on $\mathbf{X}_i^{\text{BSS}}$ to obtain a set of less reverberant mixtures $\widehat{\mathbf{X}}_i^{\text{BSS}}$, on which FastMNMF [14] is then applied after initializing the inverse of the so-called diagonalization matrix given the target directions. Although FastMNMF typically aims for the source image estimates, we are more interested in the estimated parameters of the posterior distribution $\widehat{\mathbf{x}}_{nft} | \widehat{\mathbf{x}}_{ft}$.

The local Gaussian model assumes that each source image $\widehat{\mathbf{x}}_{nft}$ follows an M-variate complex-valued circularly-symmetric Gaussian distribution, whose covariance matrix is decomposed into power spectral density (PSD) λ_{nft} and spatial covariance matrix (SCM) $\mathbf{G}_{nf}$, as $\widehat{\mathbf{x}}_{nft} \sim \mathcal{N}_{\mathbb{C}}(\mathbf{0}, \lambda_{nft}\mathbf{G}_{nf})$ [19]. To deal with the difficult optimization of this vanilla model, the state-of-the-art BSS method called FastMNMF [14] uses a nonnegative matrix factorization (NMF)-based spectral model and a jointly-diagonalizable spatial model. The PSD is given by $\lambda_{nft} \triangleq \sum_{c=1}^{C} u_{ncf} v_{nct} \in \mathbb{R}_+$, where $u_{ncf} \in \mathbb{R}_+$ and $v_{nct} \in \mathbb{R}_+$ with $c \in [1, C]$ and C is the number of NMF components. The SCM is jointly-diagonalizable by a time-invariant diagonalization matrix shared among all sources $\mathbf{Q}_f \in \mathbb{C}^{M \times M}$ as $\mathbf{G}_{nf} \triangleq \mathbf{Q}_f^{-1} \text{Diag}(\tilde{\mathbf{g}}_n) \mathbf{Q}_f^{-\mathsf{H}}$, where $\text{Diag}(\tilde{\mathbf{g}}_n)$ is a diagonal matrix whose diagonal vector is $\tilde{\mathbf{g}}_n \triangleq [\tilde{g}_{1n}, \ldots, \tilde{g}_{Mn}]^{\mathsf{T}} \in \mathbb{R}_+^M$. Thus, the probability distributions of the n-th less reverberant source image and less reverberant mixture can be expressed as

$$\widehat{\mathbf{x}}_{nft} \sim \mathcal{N}_{\mathbb{C}}^{M}\left(\mathbf{0}, \lambda_{nft}\mathbf{Q}_f^{-1}\text{Diag}(\tilde{\mathbf{g}}_n)\mathbf{Q}_f^{-\mathsf{H}}\right), \quad (1)$$

$$\widehat{\mathbf{x}}_{ft} \sim \mathcal{N}_{\mathbb{C}}^{M}\left(\mathbf{0}, \sum\nolimits_{n=1}^{N} \lambda_{nft}\mathbf{Q}_f^{-1}\text{Diag}(\tilde{\mathbf{g}}_n)\mathbf{Q}_f^{-\mathsf{H}}\right). \quad (2)$$

Consequently, the posterior distribution is given by

$$\widehat{\mathbf{x}}_{nft} \mid \widehat{\mathbf{x}}_{ft} \sim \mathcal{N}_{\mathbb{C}}^{M}\left(\mathbf{W}_{nft}\widehat{\mathbf{x}}_{ft}, \boldsymbol{\Sigma}_{nft}\right), \quad (3)$$

$$\mathbf{W}_{nft} = \mathbf{Q}_f^{-1}\text{Diag}\left(\frac{\lambda_{nft}\tilde{\mathbf{g}}_n}{\sum_{n'=1}^{N}\lambda_{n'ft}\tilde{\mathbf{g}}_{n'}}\right)\mathbf{Q}_f, \quad (4)$$

$$\boldsymbol{\Sigma}_{nft} = (\mathbf{I} - \mathbf{W}_{nft})\,\mathbf{Q}_f^{-1}\text{Diag}\left(\lambda_{nft}\tilde{\mathbf{g}}_n\right)\mathbf{Q}_f, \quad (5)$$

where $\mathbf{I}$ is the identity matrix. After the FastMNMF parameter optimization for $\widehat{\mathbf{X}}_i^{\text{BSS}}$ is finished, we compute the exponential moving averages (EMAs), $\widetilde{\mathbf{W}}_{nfi}$ and $\widetilde{\boldsymbol{\Sigma}}_{nfi}$, with $\alpha^{\text{BSS}} = 1$ for $i = 1$, to represent the Wiener filter and the posterior covariance matrix, respectively, of source n in block i as follows:

$$\widetilde{\mathbf{W}}_{nfi} = \frac{\alpha^{\text{BSS}}}{T^{\text{BSS}}}\sum\nolimits_{t'=1}^{T^{\text{BSS}}}\mathbf{W}_{nft'} + (1-\alpha^{\text{BSS}})\widetilde{\mathbf{W}}_{nf(i-1)}, \quad (6)$$

$$\widetilde{\boldsymbol{\Sigma}}_{nfi} = \frac{\alpha^{\text{BSS}}}{T^{\text{BSS}}}\sum\nolimits_{t'=1}^{T^{\text{BSS}}}\boldsymbol{\Sigma}_{nft'} + (1-\alpha^{\text{BSS}})\widetilde{\boldsymbol{\Sigma}}_{nf(i-1)}. \quad (7)$$

2.2. Front End

Given a block of data $\mathbf{X}_j^{\text{BF}}$, the front end performs an online dereverberation to obtain a less reverberant mixture $\widehat{\mathbf{x}}_{ft}$ that is then used by a beamforming to compute the target signal estimate s_{nft}. The beamforming uses $\widetilde{\mathbf{W}}_{nfi'}$ and $\widetilde{\mathbf{\Sigma}}_{nfi'}$, where i' is the index of the latest block $\mathbf{X}_{i'}^{\text{BSS}}$ processed by the back end.

2.2.1. Online Dereverberation

We remove late reverberation from the mixture $\mathbf{x}_{ft}$ using an online variant of WPE [18,20]: $\widehat{\mathbf{x}}_{ft} = \mathbf{x}_{ft} - \mathbf{H}_{ft}^{\mathsf{H}} \widetilde{\mathbf{x}}_{f(t-\Delta)} \in \mathbb{C}^M$, where $\widetilde{\mathbf{x}}_{f(t-\Delta)} \in \mathbb{C}^{MK}$ stacks $\{\mathbf{x}_{ft'} | t' \in [t-\Delta-K+1, t-\Delta]\}$ and $\mathbf{H}_{ft} = \mathbf{R}_{ft}^{-1} \mathbf{P}_{ft} \in \mathbb{C}^{MK \times M}$ is the WPE filter with $\mathbf{R}_{ft} \in \mathbb{C}^{MK \times MK}$, $\mathbf{P}_{ft} \in \mathbb{C}^{MK \times M}$, Δ is the delay, and K is the number of filter taps. Although our front end works in a block-online processing manner, we opt for an online variant that allows us to avoid frequent matrix inversion $\mathbf{R}_{ft}^{-1}$ when T^{BF} is small. We first initialize $\mathbf{R}_{f0}^{-1} \leftarrow \mathbf{I}$ and $\mathbf{H}_{f0} \leftarrow \mathbf{0}$, where $\mathbf{0}$ is the zero matrix with appropriate dimensions. The dereverberation is then performed after updating $\mathbf{R}_{ft}^{-1}$ and $\mathbf{H}_{ft}$ as follows:

$$\phi_{ft} = (M\Delta)^{-1} \sum\nolimits_{m=1}^{M} \sum\nolimits_{t'=(t-\Delta+1)}^{t} \left| \{\mathbf{x}_{ft'}\}_m \right|^2, \quad (8)$$

$$\mathbf{K}_{ft} = \frac{\alpha^{\text{WPE}} \mathbf{R}_{f(t-1)}^{-1} \widetilde{\mathbf{x}}_{f(t-\Delta)}}{(1-\alpha^{\text{WPE}}) \phi_{ft} + \alpha^{\text{WPE}} \widetilde{\mathbf{x}}_{f(t-\Delta)}^{\mathsf{H}} \mathbf{R}_{f(t-1)}^{-1} \widetilde{\mathbf{x}}_{f(t-\Delta)}}, \quad (9)$$

$$\mathbf{R}_{ft}^{-1} = \left(1 - \alpha^{\text{WPE}}\right)^{-1} \left(\mathbf{I} - \mathbf{K}_{ft} \widetilde{\mathbf{x}}_{f(t-\Delta)}^{\mathsf{H}} \right) \mathbf{R}_{f(t-1)}^{-1}, \quad (10)$$

$$\mathbf{H}_{ft} = \mathbf{H}_{f(t-1)} + \mathbf{K}_{ft} \left(\mathbf{x}_{ft} - \mathbf{H}_{f(t-1)}^{\mathsf{H}} \widetilde{\mathbf{x}}_{f(t-\Delta)} \right)^{\mathsf{H}}, \quad (11)$$

$$\widehat{\mathbf{x}}_{ft} = \mathbf{x}_{ft} - \mathbf{H}_{ft}^{\mathsf{H}} \widetilde{\mathbf{x}}_{f(t-\Delta)}. \quad (12)$$

Our calculations for $\mathbf{K}_{ft}$ and $\mathbf{R}_{ft}^{-1}$ are slightly different from those presented in [18, 21, 22] because we formulate EMAs: $\mathbf{R}_{ft} = \alpha^{\text{WPE}} \phi_{ft}^{-1} \widetilde{\mathbf{y}}_{f(t-\Delta)} \widetilde{\mathbf{y}}_{f(t-\Delta)}^{\mathsf{H}} + (1-\alpha^{\text{WPE}}) \mathbf{R}_{f(t-1)}$ and $\mathbf{P}_{ft} = \alpha^{\text{WPE}} \phi_{ft}^{-1} \widetilde{\mathbf{y}}_{f(t-\Delta)} \mathbf{y}_{ft}^{\mathsf{H}} + (1-\alpha^{\text{WPE}}) \mathbf{P}_{f(t-1)}$. With these formulations, the EMA parameters in this paper, i.e., α^{BSS}, α^{WPE}, and α^{BF}, provide the same interpretation about the weights of a new data and the accumulated data. In terms of our α^{WPE}, the EMA parameter in [22] is $(1-\alpha^{\text{WPE}})$.

2.2.2. Online Beamforming

Assuming that $\widehat{\mathbf{x}}_{nft} \mid \widehat{\mathbf{x}}_{ft} \sim \mathcal{N}_{\mathbb{C}}^M \left(\widetilde{\mathbf{W}}_{nfi'} \widehat{\mathbf{x}}_{ft}, \widetilde{\mathbf{\Sigma}}_{nfi'} \right)$, we first compute the time-varying second-order raw moment as the covariance matrix $\mathbf{\Gamma}_{nft}$ of source n and the corresponding interference covariance matrix $\mathbf{\Upsilon}_{nft}$. EMAs $\widetilde{\mathbf{\Gamma}}_{nfj}$, $\widetilde{\mathbf{\Upsilon}}_{nfj}$ representing the source covariance matrices in block j are calculated with $\alpha^{\text{BF}} = 1$ for $j = 1$. A beamformer $\mathbf{w}_{nfj}^{\text{MV}}$ [5] is then obtained given a vector $\mathbf{u}_{m'}$, whose m'-th entry is 1 and 0 elsewhere with m' is the reference microphone index, as follows:

$$\mathbf{\Gamma}_{nft} = \widetilde{\mathbf{W}}_{nfi'} \widehat{\mathbf{x}}_{ft} \widehat{\mathbf{x}}_{ft}^{\mathsf{H}} \widetilde{\mathbf{W}}_{nfi'}^{\mathsf{H}} + \widetilde{\mathbf{\Sigma}}_{nfi'}, \quad (13)$$

$$\mathbf{\Upsilon}_{nft} = \widehat{\mathbf{x}}_{ft} \widehat{\mathbf{x}}_{ft}^{\mathsf{H}} - \mathbf{\Gamma}_{nft}, \quad (14)$$

$$\widetilde{\mathbf{\Gamma}}_{nfj} = \frac{\alpha^{\text{BF}}}{T^{\text{BF}}} \sum\nolimits_{t=1}^{T^{\text{BF}}} \mathbf{\Gamma}_{nft} + \left(1 - \alpha^{\text{BF}}\right) \widetilde{\mathbf{\Gamma}}_{nf(j-1)}, \quad (15)$$

$$\widetilde{\mathbf{\Upsilon}}_{nfj} = \frac{\alpha^{\text{BF}}}{T^{\text{BF}}} \sum\nolimits_{t=1}^{T^{\text{BF}}} \mathbf{\Upsilon}_{nft} + \left(1 - \alpha^{\text{BF}}\right) \widetilde{\mathbf{\Upsilon}}_{nf(j-1)}, \quad (16)$$

$$\mathbf{w}_{nfj}^{\text{MV}} = \left(\text{tr}\left(\widetilde{\mathbf{\Upsilon}}_{nfj}^{-1} \widetilde{\mathbf{\Gamma}}_{nfj} \right) \right)^{-1} \widetilde{\mathbf{\Upsilon}}_{nfj}^{-1} \widetilde{\mathbf{\Gamma}}_{nfj} \mathbf{u}_{m'}. \quad (17)$$

Finally, we use the beamformer for all time frames in block j, i.e., $\mathbf{w}_{nft}^{\text{MV}} \leftarrow \mathbf{w}_{nfj}^{\text{MV}}$, to obtain a single-channel enhanced signal:

$$s_{nft} = \left(\mathbf{w}_{nft}^{\text{MV}} \right)^{\mathsf{H}} \widehat{\mathbf{x}}_{ft}. \quad (18)$$

3. EVALUATION

This section presents the evaluation of our proposed method on data recorded using a Microsoft HoloLens 2 (HL2).

3.1. Experimental Settings

The evaluation was performed on the test subset of the dataset used in [11]. This subset contained eight simulated noisy mixture signals, each of which consists of multiple utterances, amounted to 18 min in total. Each mixture signal was composed of two reverberant speech signals and one diffuse noise signal ($N = 3$), which were recorded separately in a room with an RT_{60} of about 800 ms using the 5 microphones of an HL2 ($M = 5$). The dry speech signals were taken from the Librispeech dataset [23], and the noise signals were taken from the CHiME-3 dataset [24]. The noise source was located 3 m away from the HL2 in the direction of $135°$, where $0°$ was in front of the HL2, behind multiple portable room dividers to build up reflections, which characterize a diffuse noise. The target speaker and the interfering speaker were located 1.5 m away from the HL2. The target speaker was in the direction of $0°$, while the interfering speaker varied for each utterance between $\{45°, 90°, 180°, 225°, 270°, 315°\}$. Each noisy mixture signal was fed in turn to a speech enhancement method, so the method needs to handle the interfering speaker movements.

All audio signals were sampled at 16 kHz. The STFT coefficients were extracted using a 1024-point Hann window ($F = 513$) with 75% overlap. To factor out possible instability of the first few EMA computations due to improper initialization, we concatenated the last 1024 frames (≈ 16 s) of each noisy mixture to its beginning so that the compared methods processed a few blocks before the performance measurement started.

The block size of the back end was set to $T^{\text{BSS}} = 256$ frames with 75% overlap. Thus, the back end provided new $\widetilde{\mathbf{W}}_{nfi'}$ and $\widetilde{\mathbf{\Sigma}}_{nfi'}$ every 64 frames (≈ 1 s). The back-end iterative offline WPE was performed for 3 iterations using the tap length of 5 and the delay of 3. The number of NMF components was $C = 8$ and the number of FastMNMF parameter updates was 50, including 40 warming-up iterations with the frequency-invariant source model. The front-end online WPE was performed using the tap length of 5 and the delay of 3. We loosely performed a grid search in preliminary experiments by considering $\alpha^{\text{WPE}}, \alpha^{\text{BF}}, \alpha^{\text{BSS}} \in \{0.500, 0.200, 0.100, 0.050, 0.020, 0.010, 0.005\}$. The experiments presented in this paper used $\alpha^{\text{WPE}} = 0.005$ and $\alpha^{\text{BSS}} = 0.100$. The experimental results illustrate the proposed system's top performance on the

Table 1. Average WERs [%] and computation times [ms] of the *baseline* front ends. The total latency [ms] is the sum of the block shift size and the average computation time for each block. Lower WER score and computation time are better.

Method	Block Size [ms]	Block Shift [ms]	Comp. Time [ms]	Total Latency [ms]	WER [%]
Clean (ground truth)	—	—	—	—	6.1
Noisy (observation)	—	—	—	—	92.1
Online WPE	16	16	3	19	87.8
Online WPE + DS	16	16	4	20	68.4
Online WPE + MPDR	16	16	6	22	47.1
Online WPE + MPDR	64	64	16	80	46.0
Online WPE + MPDR	256	256	54	310	47.2
WPE + MVDR with DNN-based mask estimation [11]					
(before adaptation)	3000	500	250	750	35.6
(after adaptation)	3000	500	250	750	20.4

test set because the hyperparameter tuning for α^{WPE}, α^{BF}, and α^{BSS} was performed on the same set.

The performance was evaluated in terms of the word error rate (WER) [%] and the computation time [ms]. The ASR system was based on the transformer-based acoustic and language models of the SpeechBrain toolkit [25]. We additionally perform the standard statistical test called the Matched Pair Sentence Segment Word Error (MAPSSWE) test [26] to determine whether two WERs obtained by two different systems are different [2]. It is a two-tailed test whose null hypothesis is that there is no performance difference between the two systems. The computation time was measured on Intel Xeon E5-2698 v4 (2.20 GHz) with NVIDIA Tesla V100 SXM2 (16GB).

3.2. Experimental Results and Discussion

Table 1 shows the baseline performances. The WERs for 'clean', 'observed noisy', and 'online WPE' were computed using the top center microphone of HL2. The online WPE (Sect. 2.2.1) was also used with the delay-and-sum (DS) beamforming or the minimum power distortionless response (MPDR) beamforming. For 'online WPE + DS' and 'online WPE + MPDR', we selected one vector from the set of pre-recorded steering vectors given the target directions. For 'online WPE + MPDR', we used an EMA of mixture covariance matrices computed similar to Eq. (15). Thus, its performance affected by the block size and shift size. The shown WERs for 16 ms $\left(T^{\text{BF}}=1\right)$, 64 ms $\left(T^{\text{BF}}=4\right)$, and 256 ms $\left(T^{\text{BF}}=16\right)$ were achieved using the optimal α^{BF}, i.e., 0.020, 0.100, 0.200, respectively. The performances of 'WPE + MVDR with DNN-based mask estimation' were the best ones shown in [11].

Tables 2 and 3 show the computational times and the average word error rates (WERs), respectively, of the proposed system for different T^{BF} (with no overlap) and different α^{BF}. The WERs marked with * are not statistically different (the null hypothesis is accepted at the 95% confidence level) from the best WER marked with *, i.e., 14.8% for $\left(T^{\text{BF}}=2, \alpha^{\text{BF}}=0.020\right)$. It is worth noting that for $\left(T^{\text{BF}}=1, \alpha^{\text{BF}}=0.010\right)$, the WER (i.e., 15.4%) is statistically the same as the best performance. It demonstrates that our proposed system can also perform very well even with the low-latency frame-online processing.

Table 2. Computation times and total latencies [ms] of the *proposed* front end for different T^{BF} [frames] (with no overlap). Lower computation time is better.

Block size T^{BF} [frames]	1	2	4	8	16	32
Block shift [ms]	16	32	64	128	256	512
Computation time [ms]	6	10	17	30	57	111
Total latency [ms]	22	42	81	158	313	623

Table 3. Average WERs [%] of the *proposed* system for different T^{BF} [frames] and α^{BF}. Lower WER score is better. For visualization purpose, the shading of each cell reflects the WER score. The best performance for each T^{BF} is in bold type. The top performances that are not statistically different from the overall best performance indicated by * are marked with *.

$\downarrow T^{\text{BF}}$ \ α^{BF}	0.500	0.200	0.100	0.050	0.020	0.010	0.005
32	15.8*	**15.2***	16.8	21.8	27.9	38.1	50.6
16	20.1	**15.0***	15.7*	17.8	23.7	28.1	36.8
8	39.0	18.3	**14.9***	15.7*	18.7	23.1	27.4
4	93.8	30.9	18.7	**14.9***	15.8*	19.7	23.0
2	95.4	75.7	30.9	18.8	**14.8***	16.1	20.1
1	98.3	97.2	68.3	30.7	16.8	**15.4***	16.6

The optimal α^{BF} for each T^{BF} suggests that when a small block size is used, the front end should rely more on the accumulated statistics and put less importance on the newly acquired data (cf. Eqs. (15) and (16)). Inappropriate α^{BF} may have detrimental effects, e.g., $\alpha^{\text{BF}}=0.500$ for $T^{\text{BF}}\in\{1,2,4\}$. Using optimal α^{BF}, our proposed system outperformed the baseline performances, including that of mask-based MVDR with DNN [11]. It suggests that our estimation of source covariance matrices for deriving the MVDR beamformer was more adaptive in handling the sound scene changes due to the interfering speaker movements. Accumulating the statistics using EMA seems crucial and may also benefit MVDR with DNN-based mask estimation. We leave this for future work.

4. CONCLUSION

This paper proposes a practical approach to the enhancement of adaptive speech with low latency and high performance. The system operates an online MVDR beamforming on the front end that adopts the posterior distribution obtained by the back-end BSS based on FastMNMF. Future works include considering scenarios with continuously moving sources and automating hyperparameter tuning for α^{WPE}, α^{BF}, and α^{BSS}.

5. REFERENCES

[1] X. Alameda-Pineda, E. Ricci, and N. Sebe, *Multimodal Behavior Analysis in the Wild: Advances and Challenges*, Elsevier Science, 2018.

[2] D. Jurafsky and J. H. Martin, *Speech and Language Processing*, 3rd edition, Draft of Jan. 12, 2022. [Online]. Available: https://web.stanford.edu/~jurafsky/slp3/.

[3] E. Vincent, T. Virtanen, and S. Gannot, Eds., *Audio Source Separation and Speech Enhancement*, John Wiley & Sons, 2018.

[4] K. Kumatani, J. McDonough, and B. Raj, "Microphone array processing for distant speech recognition: From close-talking microphones to far-field sensors," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 127–140, 2012.

[5] M. Souden, J. Benesty, and S. Affes, "On optimal frequency-domain multichannel linear filtering for noise reduction," *IEEE Trans. Audio, Speech, Language Process.*, vol. 18, no. 2, pp. 260–276, Feb. 2010.

[6] J. Heymann, L. Drude, and R. Haeb-Umbach, "A generic neural acoustic beamforming architecture for robust multi-channel speech processing," *Computer Speech & Language*, vol. 46, pp. 374–385, 2017.

[7] A. A. Nugraha, *Deep neural networks for source separation and noise-robust speech recognition*, Ph.D. thesis, Université de Lorraine, 2017.

[8] S. Sivasankaran, E. Vincent, and D. Fohr, "SLOGD: Speaker location guided deflation approach to speech separation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2020, pp. 6409–6413.

[9] R. Gu, S.-X. Zhang, Y. Zou, and D. Yu, "Complex neural spatial filter: Enhancing multi-channel target speech separation in complex domain," *IEEE Signal Process. Lett.*, vol. 28, pp. 1370–1374, 2021.

[10] J. Casebeer, J. Donley, D. Wong, B. Xu, and A. Kumar, "NICE-Beam: Neural integrated covariance estimators for time-varying beamformers," 2021, arXiv:2112.04613v1.

[11] K. Sekiguchi, A. A. Nugraha, Y. Du, Y. Bando, M. Fontaine, and K. Yoshii, "Direction-aware adaptive online neural speech enhancement with an augmented reality headset in real noisy conversational environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, arXiv:2207.07296v1.

[12] A. Ozerov and C. Févotte, "Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 18, no. 3, pp. 550–563, 2009.

[13] N. Ito and T. Nakatani, "FastMNMF: Joint diagonalization based accelerated algorithms for multichannel nonnegative matrix factorization," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, May 2019, pp. 371–375.

[14] K. Sekiguchi, Y. Bando, A. A. Nugraha, K. Yoshii, and T. Kawahara, "Fast multichannel nonnegative matrix factorization with directivity-aware jointly-diagonalizable spatial covariance matrices for blind source separation," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 28, pp. 2610–2625, 2020.

[15] T. Higuchi, N. Ito, S. Araki, T. Yoshioka, M. Delcroix, and T. Nakatani, "Online MVDR beamformer based on complex Gaussian mixture model with spatial prior for noise robust ASR," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 25, no. 4, pp. 780–793, 2017.

[16] C. Boeddecker, J. Heitkaemper, J. Schmalenstroeer, L. Drude, J. Heymann, and R. Haeb-Umbach, "Front-end processing for the CHiME-5 dinner party scenario," in *Proc. Int. Workshop Speech Process. Everyday Environs.*, 2018, pp. 35–40.

[17] K. Shimada, Y. Bando, M. Mimura, K. Itoyama, K. Yoshii, and T. Kawahara, "Unsupervised speech enhancement based on multichannel NMF-informed beamforming for noise-robust automatic speech recognition," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 27, no. 5, pp. 960–971, 2019.

[18] L. Drude, J. Heymann, C. Boeddeker, and R. Haeb-Umbach, "NARA-WPE: A python package for weighted prediction error dereverberation in numpy and tensorflow for online and offline processing," in *Proc. ITG Symp. Speech Commun.*, 2018, pp. 1–5.

[19] N. Q. K. Duong, E. Vincent, and R. Gribonval, "Under-determined reverberant audio source separation using a full-rank spatial covariance model," *IEEE Trans. Audio, Speech, Language Process.*, vol. 18, no. 7, pp. 1830–1840, 2010.

[20] T. Yoshioka and T. Nakatani, "Generalization of multi-channel linear prediction methods for blind MIMO impulse response shortening," *IEEE Trans. Audio, Speech, Language Process.*, vol. 20, no. 10, pp. 2707–2720, 2012.

[21] J. Caroselli, I. Shafran, A. Narayanan, and R. Rose, "Adaptive multichannel dereverberation for automatic speech recognition," in *Proc. INTERSPEECH*, 2017, pp. 3877–3881.

[22] J.-M. Lemercier, J. Thiemann, R. Koning, and T. Gerkmann, "Customizable end-to-end optimization of online neural network-supported dereverberation for hearing devices," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2022, pp. 171–175.

[23] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2015, pp. 5206–5210.

[24] E. Vincent, S. Watanabe, A. A. Nugraha, J. Barker, and R. Marxer, "An analysis of environment, microphone and data simulation mismatches in robust speech recognition," *Comput. Speech Lang.*, vol. 46, pp. 535–557, 2017.

[25] M. Ravanelli, T. Parcollet, P. Plantinga, A. Rouhe, S. Cornell, L. Lugosch, C. Subakan, N. Dawalatabad, A. Heba, J. Zhong, J.-C. Chou, S.-L. Yeh, S.-W. Fu, C.-F. Liao, E. Rastorgueva, F. Grondin, W. Aris, H. Na, Y. Gao, R. D. Mori, and Y. Bengio, "SpeechBrain: A general-purpose speech toolkit," 2021, arXiv:2106.04624v1.

[26] L. Gillick and S. Cox, "Some statistical issues in the comparison of speech recognition algorithms," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 1989, vol. 1, pp. 532–535.