ABSTRACT

Most existing streaming neural-network-based multi-channel speech separation systems consist of a causal network architecture and an online spatial information extraction module. The spatial information extraction module can either be a feature calculation module that generates cross-channel features or an online beamforming module that explicitly performs frame- or chunk-level spatial filtering. While such online beamforming modules were mainly proposed in the frequency domain, recent literature have investigated the potential of learnable time-domain methods which can be jointly optimized with the entire model with a single training objective. Among those methods, the time-domain generalized Wiener filter (TD-GWF) has shown performance gain compared to conventional frequency-domain beamformers in the sequential beamforming pipeline. In this paper, we modify the offline TD-GWF to an online counterpart via a Sherman–Morrison formula-based approximation and introduce how we simplify and stabilize the training phase. Experiment results on applying various offline and online spatial filtering modules in the sequential beamforming pipeline show that the online TD-GWF can obtain better performance than an offline frequency-domain multi-channel Wiener filter (FD-MCWF) in the noisy multi-channel reverberant speech separation task.

Index Terms: Multi-channel Speech separation, Generalized Wiener filter

1. INTRODUCTION

The design and use of neural beamformers is a hot topic in the community of multi-channel speech enhancement, separation, and recognition [1][9]. A standard neural beamformer typically applies a neural network to extract an estimation of the target source out of multi-channel noisy observations, and use the estimated target source to calculate the beamforming coefficients via the procedure of a selected beamformer. Frequency-domain beamformers have long been the top choices in such pipelines [10], and time-domain systems have also been proposed in recent years to better utilize the power of end-to-end-trained neural networks [3][11][12]. Another application of beamformers is to use them in the sequential beamforming pipeline [13][16], where the output of the beamformers is treated as auxiliary inputs to a second-stage post-processing module to re-extract the target sources. In this configuration, the beamformers can be viewed as a (typically linear) spatial feature extractor that improves the performance of a second-stage nonlinear neural separator. Time-domain spatial filtering methods may perform better than the frequency-domain opponents, as they can typically be jointly optimized with the post-processing module [16].

In practice, such (sequential) neural beamforming systems often need to be modified to support streaming processing, as applications such as telecommunication and live captioning often requires a short system latency. Although neural networks can be easily adjusted to satisfy such causal requirement, the performance of beamformers can be limited as the available spatial information is constrained in such cases. Methods have been proposed for such streaming scenario, and the two main categories are the use of fixed beamformers and recursive feature estimation. Fixed beamformers were introduced as beam candidates covering a wide range of desired directions, and a beam selection module can be applied to select the best direction out of all options [17]. This bypasses the need for calculating the beamforming coefficients, however also sacrifices its adaptation ability on the beam patterns. Recursive feature estimation iteratively calculates the cumulative spatial features, e.g. the spatial covariance matrices, and use them to update the beamforming coefficients at frame level [15]. This preserves the adaptation ability of the system, but introduces additional computational costs.

In this paper, we extend our previous work on an offline neural beamformer, the time-domain generalized Wiener filter (TD-GWF) [16], to the online scenario. The offline TD-GWF calculates the Wiener filter coefficients on learnable real-valued 2-D features calculated on the waveforms of the noisy observations and the estimated target source, and the filter coefficients are defined as the solution to an utterance-level minimum mean-square error (MMSE) estimation between the features of the estimated target source and the filtered noisy observations. To modify it to support streaming processing, we follow the recursive feature estimation method and change the utterance-level filter coefficient estimation to frame-level estimation. Similar to existing works in online frequency-domain beamformers [19], we apply a Sherman–Morrison formula-based approximation [20] to bypass the frame-level calculation of matrix inverse to save the computational cost in both training and inference phase. Moreover, to tackle the numerical instability issue in the Sherman–Morrison formula, we introduce a learning-based initialization of the covariance matrix that stabilizes the training phase. Similar to the offline TD-GWF, we also apply the online TD-GWF to the sequential beamforming pipeline, where the output of the beamformer is served as an auxiliary feature for a second-stage neural separator. Experiment results on a noisy reverberant speech separation task show that the frame-wise online TD-GWF can achieve better performance than an offline frequency-domain multi-channel Wiener filter (FD-MCWF) with a same network architecture for the neural separators, proving its potential in streaming multi-channel speech separation and enhancement problems.

The rest of the paper is organized as follows. Section 2 introduces the proposed online TD-GWF, demonstrates how we simply the filter coefficient calculation and stabilize the training, and shows how it is applied in the sequential neural beamforming pipeline. Section 3 provides the experiment configurations. Section 4 presents the experiment results. Section 5 concludes the paper.
Fig. 1: Flowchart of online time-domain generalized Wiener filter (online TD-GWF) at frame $T$. The Wiener filter coefficients are updated at every frame using the information of the entire mixture history $Y_{1:T}$ and the entire estimated target source $\hat{X}_{1:T}$, but the filter coefficients are only applied to the $T$-th frame in the mixture.

2. ONLINE TIME-DOMAIN GENERALIZED WIENER FILTER

2.1. Online Time-domain Generalized Wiener Filter

Figure 1 shows the flowchart for the proposed online Time-domain Generalized Wiener filter (TD-GWF). We first transform the 1-D $M$-channel noisy input waveforms $y_{m} \in \mathbb{R}^{1 \times L}$ into 2-D features by applying a learnable linear transform:

$$Y_{m,t} = y_{m,t}B$$

(1)

where $y_{m,t} \in \mathbb{R}^{1 \times P}$ denote the $t$-th frame of the windowed waveform with $P$ sample points at the $m$-th channel, $B \in \mathbb{R}^{P \times N}$ denotes the learnable transformation matrix, or equivalently the waveform encoder in recent time-domain speech separation systems [21], and $Y_{m,t} \in \mathbb{R}^{N \times T}$ denote the generated $N$-dimension sequential feature at the $m$-th channel. $\{y_{m}\}_{m=1}^{M}$ are then split to $V$ non-overlapped groups of $\frac{N}{V}$-dimension sub-features, where we set $V = N/2$ according to the configuration in offline TD-GWF [19], and the $M$ channels of sub-features in the same group are then concatenated to form $\hat{Y} \in \mathbb{R}^{V \times \frac{MN}{V} \times T}$. Suppose that a pre-separation model extracts the estimated source-of-interests (SOIs) at a selected reference channel $\{x_{v}\}_{v=1}^{L} \in \mathbb{R}^{1 \times L}$ (we drop the subscript where there is no ambiguity). We use the same waveform encoder and group-splitting and concatenation scheme to form the 2-D features $\hat{X} \in \mathbb{R}^{V \times \frac{MN}{V} \times T}$.

Different from the offline TD-GWF where the entire sequence is jointly processed, online TD-GWF calculates the filter coefficients at frame level. For time step $T$, we use the feature between the first and the $T$-th frame in the noisy input features $\hat{Y}_{1:T} \in \mathbb{R}^{V \times \frac{MN}{V} \times T}$ and the estimated SOI feature $\hat{X}_{1:T} \in \mathbb{R}^{V \times \frac{MN}{V} \times T}$, and solve the minimum mean-square error (MMSE) problem for each group $v$:

$$W_{T}^{v} = \arg \min_{W_{T}} \|W_{T}^{v} \hat{Y}_{1:T}^{v} - \hat{X}_{1:T}^{v}\|_2, \quad v = 1, \ldots, V$$

(2)

where $\hat{X}_{1:T}^{v} \in \mathbb{R}^{V \times T}$ and $\hat{Y}_{1:T}^{v} \in \mathbb{R}^{\frac{MN}{V} \times T}$ denote the $v$-th group of SOI and noisy input features, respectively, and $W_{T}^{v} \in \mathbb{R}^{\frac{MN}{V} \times \frac{MN}{V}}$ denotes the corresponding filter coefficients. A closed-form solution for $W_{T}^{v}$ can be defined:

$$W_{T}^{v} = (\hat{Y}_{1:T}^{v} \hat{Y}_{1:T}^{v \top})^{-1}\hat{Y}_{1:T}^{v} \hat{X}_{1:T}^{v \top}$$

(3)

It is easy to observe that the proposed online TD-GWF utilizes all entire history information in the calculation of the filter coefficients at a certain time step. When $T$ reaches the total length of the input feature, the calculation of $W_{T}^{v}$ becomes equivalent to the calculation of an offline TD-GWF.

$\hat{W}_{T}^{v}$ is then applied to the $T$-th frame of the noisy input feature $\hat{Y}_{1:T}^{v} \in \mathbb{R}^{V \times \frac{MN}{V} \times T}$ to obtain the $v$-th group of the output:

$$\hat{X}_{1:T}^{v} = \hat{W}_{T}^{v \top} \hat{Y}_{1:T}^{v}$$

(4)

The final output feature $\hat{X} \in \mathbb{R}^{V \times T}$ is obtained in the same way as the offline TD-GWF by concatenating the $V$ groups of outputs $\{\hat{X}_{v}\}_{v=1}^{V}$ across the feature dimension:

$$\hat{X} = \text{Concat}(\{\hat{X}_{v}\}_{v=1}^{V})$$

(5)

The final output waveform $\hat{x} \in \mathbb{R}^{1 \times L}$ is then obtained by applying another linear transform (i.e., a waveform decoder) $D \in \mathbb{R}^{N \times P}$ to $\hat{X}$ together with the overlap-add (OLA) operation on the windows:

$$\hat{x} = \text{OLA}(D^{\top}\hat{X})$$

(6)

2.2. Simplifying and Stabilizing the Training of Online TD-GWF

Equation 3 can be calculated by standard matrix inverse operators or linear system solvers [16]. However, calculating matrix inverse or solving linear systems at every frame can be time and computation consuming, which limits the application of online TD-GWF in resource-efficient platforms. Moreover, since the beginning of a sentence can often be silence, the energy of $\hat{Y}_{1:T}^{v}$ when $T$ is small can be low, which results in extremely large entries in $\hat{Y}_{1:T}^{v \top}\hat{Y}_{1:T}^{v}$ and causes numerical instability in training. Here we introduce how
we simplify the calculation of the matrix inverse operations via a Sherman–Morrison formula-based approximation [20], and how we solve the numerical issue by adding small perturbations at the beginning.

2.2.1. Frame-level Update without Matrix Inverse

The use of Sherman–Morrison formula in online frequency-domain beamformers has been investigated in prior works [19], and we adopt it in the frame-level calculation of $W^T_T$. We first calculate the matrix inverse for $\Lambda_1 \triangleq \hat{Y}_1^T \hat{Y}_1 \in \mathbb{R}^{MN \times MN}$ (we drop the superscript $v$ for $\Lambda_1$ for simplicity), and use the Sherman–Morrison formula for all $T > 1$:

$$\Lambda_T^{-1} = (\Lambda_{T-1} + \hat{Y}_T^v \hat{Y}_T^v T)^{-1} \approx \Lambda_{T-1}^{-1} - \Lambda_{T-1}^{-1} \hat{Y}_T^v \hat{Y}_T \frac{1}{1 + \hat{Y}_T^v \Lambda_{T-1}^{-1} \hat{Y}_T^v} \Lambda_{T-1}^{-1}$$

It allows us to only calculate the matrix inverse once for the first frame and use standard matrix multiplication operations for all the other frames. Note that since Sherman–Morrison formula is valid only when $\{\Lambda_1\}_{T=1}^T$ are invertible while $\Lambda_1$ is clearly rank-deficient, equation (8) can only be an approximation instead of a closed-form solution.

2.2.2. Stabilizing the Numerical Issues

Another problem for the calculation of $\Lambda_T^{-1}$ is that when the energy of $\hat{Y}_1$ is low, entries in $\Lambda_1^{-1}$ can be extremely large. This can happen when the first frame in the noisy input waveforms is silence or has low energy. We empirically find that when directly using the pseudo-inverse of $\Lambda_1$ to initialize the Sherman–Morrison formula, the training of the entire system quickly collapses because of the numerical overflow issue. To avoid such issue, one option is to use a look-ahead window to gather multiple frames to ensure $T > \frac{MN}{K}$, but it leads to the cold-start problem and introduces a system delay. Instead, we use a learnable short sequence of features $P \in \mathbb{R}^{MN \times K}$, $K > \frac{MN}{V}$, as a part of the parameters to be optimized with the entire system, and $\Lambda_1$ is replaced by $\frac{1}{\alpha} P P^T$ where $\alpha \in \mathbb{R}^+$ is a positive rescaling scalar. The idea is to use a(n ideally) full-rank matrix (as we set $K > \frac{MN}{V}$) to replace $\Lambda_1$ so that its inverse can be properly calculated, and its energy can be constrained to a reasonable scale. $P$ can also be viewed as a low-energy frame added to the beginning of an utterance. We set the positive rescaling scalar $\alpha$ to be the Frobenius norm of $P P^T$. Note that the size of $P$ is related to the number of channels $M$, and for ad-hoc microphone array configurations, a maximum number of supported channels $M_{max}$ can be defined in advance.

2.3. Online TD-GWF in Sequential Neural Beamforming

As in the original literature of offline TD-GWF, we insert online TD-GWF in the sequential neural beamforming pipeline [6, 13, 14]. Figure 2 shows the pipeline for a sequential neural beamforming system, where the pre-separation module performs separation on the input $Y$ to generate the estimated target sources $\{\hat{X}_c^{(1)}\}_{c=1}^C$. $\{\hat{X}_c^{(1)}\}_{c=1}^C$ is then sent to the spatial filtering module (e.g., a beamforming module) to calculate the output $\{\hat{X}_c^{(1)}\}_{c=1}^C$. $Y$, $\{\hat{X}_c^{(1)}\}_{c=1}^C$, and $\{\hat{X}_c^{(1)}\}_{c=1}^C$ are then all sent to the post-separation module to re-separate the estimated target sources $\{\hat{X}_c^{(2)}\}_{c=1}^C$. Such spatial filtering and post-separation process can be repeated for multiple times, and here we only consider a single iteration. The outputs of the pre-separation module and the post-separation module are all sent to the training objective, and the gradients for the spatial filtering module and the post-separation module are cut during backpropagation so that they do not affect the gradients in the pre-separation module.

The role of spatial filtering module in the sequential neural beamforming pipeline is different from other existing neural beamforming pipelines as the output of the spatial filtering module is not directly used as the system’s final output. The output is used as an auxiliary feature that helps the post-separation module to re-separate the input and obtain a better result. In this case, the spatial filtering module can be viewed as a linear spatial feature extraction module that improves the nonlinear post-separation module. In the online configuration, the pre-separation, spatial filtering and post-separation modules should all be causal. Although different modules can use different system latency settings, here we consider the frame-level streaming configuration where no look-ahead is applied in any of the modules.

3. EXPERIMENT CONFIGURATIONS

3.1. Dataset

We use the same noisy reverberant two-speaker dataset proposed in [22] and used in the offline TD-GWF system for evaluating the effec-
Table 1. Comparison of different multi-channel pre-separation models on the simulated 6-mic circular array.

<table>
<thead>
<tr>
<th>Model</th>
<th>Speaker angle</th>
<th>SI-SDR (dB)</th>
<th>PESQ</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>&lt;15° 15-45°</td>
<td></td>
</tr>
<tr>
<td>Mixture</td>
<td>-0.5</td>
<td>-0.4</td>
<td>-0.4</td>
</tr>
<tr>
<td>DRNN</td>
<td>5.0</td>
<td>5.2</td>
<td>5.6</td>
</tr>
<tr>
<td>DRNN + offline TD-GWF</td>
<td>7.1</td>
<td>7.9</td>
<td>8.8</td>
</tr>
<tr>
<td>DRNN + offline FD-MCWF</td>
<td>5.4</td>
<td>5.7</td>
<td>6.1</td>
</tr>
<tr>
<td>DRNN + online TD-GWF</td>
<td>5.7</td>
<td>6.3</td>
<td>7.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&lt;25% 25-50% 50-75%</td>
<td>&gt;75%</td>
<td></td>
</tr>
<tr>
<td>Mixture</td>
<td>-0.4</td>
<td>-0.4</td>
<td>-0.5</td>
</tr>
<tr>
<td>DRNN</td>
<td>10.0</td>
<td>6.8</td>
<td>3.8</td>
</tr>
<tr>
<td>DRNN + offline TD-GWF</td>
<td>13.0</td>
<td>9.8</td>
<td>6.9</td>
</tr>
<tr>
<td>DRNN + offline FD-MCWF</td>
<td>10.5</td>
<td>7.4</td>
<td>4.4</td>
</tr>
<tr>
<td>DRNN + online TD-GWF</td>
<td>11.1</td>
<td>8.2</td>
<td>5.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Average</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mixture</td>
<td>-0.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DRNN</td>
<td>10.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DRNN + offline TD-GWF</td>
<td>13.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DRNN + offline FD-MCWF</td>
<td>10.5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DRNN + online TD-GWF</td>
<td>11.1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3.2. Model configurations

Since the main contribution of the paper is the online TD-GWF module, we use a simple network architecture for the pre- and post-separation modules. We use a frequency-domain deep recurrent neural network (DRNN) architecture which takes the complex-valued spectrogram as input and uses stacked residual LSTM layers followed by a multilayer perceptron (MLP) to estimate unbounded complex ratio masks (cRMs) for the target sources. The real and imaginary parts of the complex-valued spectrogram $Y$ are concatenated and passed to a linear fully-connected (FC) layer to perform dimension projection before sent to the stack residual LSTM layers. Each residual LSTM layer contains a cumulative layer normalization (cLN) module [21], a uni-directional LSTM layer and a linear FC layer for dimension projection. The input to the residual LSTM layer is first passed to the cLN module, and the output of the LSTM layer is passed to the FC layer and then added with the original input. The MLP module for mask generation also contains a cLN module to normalize its input. The configuration is similar to the one described in [23] despite the use of the deep LSTM layers. The pre-separation and post-separation modules share the same network architecture but different parameters. The size of hidden units in the LSTM layers is set to 128, and the number of residual LSTM layers is set to 3 in both modules. We only use 1 single iteration in the sequential beamforming pipeline. For a single-channel baseline, we use 6 residual LSTM layers to match the overall model size and complexity of the multi-channel systems.

To simplify the configuration, the DRNN systems in the pre-separation module in the sequential neural beamforming pipelines still takes single-channel input, where the first microphone is always selected as the reference microphone. Similar to the offline TD-GWF systems, the post-separation module is always a single-channel module. We use the offline TD-GWF module, the offline FD-MCWF module [16], and the proposed online TD-GWF module as the spatial filtering module and compare their performance. The window and hop sizes for Fourier transform and FD-MCWF are both set to 32 ms and 8 ms, respectively.

3.3. Training and Evaluation

All models are trained for 100 epochs with the Adam optimizer [24] with an initial learning rate of 0.001. Signal-to-noise ratio (SNR) is used as the training objective, and the clean reverberant SOIs are used as the training targets. The learning rate is decayed by 0.98 for every two epochs. Gradient clipping by a maximum gradient norm of 5 is applied. Early stop is applied when there is not best model found in the validation set for 10 consecutive epochs. We report the scale-invariant signal-to-distortion ratio (SI-SDR) [25] and the wideband perceptual quality evaluation of speech (PESQ) [26] for system comparison.

4. RESULTS AND DISCUSSIONS

Table 1 shows the results of the single-channel system, the sequential beamforming systems with offline spatial filtering module, and the sequential beamforming system with the proposed online TD-GWF module. Note that even though the offline spatial filtering modules are used, the separation networks are still causal, and these experiments are designed to serve as performance indicators of the improvements obtained by the spatial filtering modules. We can see that the causal DRNN system with offline TD-GWF module achieves a 2.9 dB average SI-SDR improvement compared with the single-channel causal DRNN system, while the offline FD-MCWF module only achieves a 0.4 dB average SI-SDR improvement. This again proves that TD-GWF is potentially more powerful than conventional frequency-domain beamformers in the sequential beamforming pipeline. Adding the proposed frame-level online TD-GWF module to the DRNN baseline leads to a 1.3 dB SI-SDR improvement, which is higher than the gain obtained by the offline FD-MCWF. This shows that the frame-level online TD-GWF is able to learn better spatial features that assists a second-stage post-separation module than an offline spatial filtering module.

5. CONCLUSION

In this paper, we proposed online time-domain generalized Wiener filter (TD-GWF) designed for the streaming multi-channel source enhancement and separation task. Online TD-GWF modified the offline TD-GWF by applying a frame-level calculation of the filter coefficients, and we adopted the Sherman-Morrison formula together with a learnable initial covariance matrix to simplify and stabilize the filter calculation and model training. Experiment results showed that when inserted into the sequential neural beamforming pipeline, online GWF can achieve better performance than an offline frequency-domain multi-channel Wiener filter (FD-MCWF).
6. REFERENCES


