INTERSPEECH 2017
August 20–24, 2017, Stockholm, Sweden

Adaptive Multichannel Dereverberation for Automatic Speech Recognition

Joe Caroselli, Izhak Shafran, Arun Narayanan, Richard Rose

Google, Inc.
{jcarosel, izhak, arunnt, rickrose}@google.com

Abstract

Reverberation is known to degrade the performance of automatic speech recognition (ASR) systems dramatically in far-field conditions. Adopting the weighted prediction error (WPE) approach, we formulate an online dereverberation algorithm for a multi-microphone array. The key contributions of this paper are: (a) we demonstrate that dereverberation using WPE improves performance even when the acoustic models are trained using multi-style training (MTR) with noisy, reverberated speech; (b) we show that the gains from WPE are preserved even in large and diverse real-world data sets; (c) we propose an adaptive version for online multichannel ASR tasks which gives similar gains as the non-causal version; and (d) while the algorithm can just be applied for evaluation, we show that also including dereverberation during training gives increased performance gains. We also report how different parameter settings of the dereverberation algorithm impacts the ASR performance.
Index Terms: speech recognition, dereverberation

1. Introduction

In far-field conditions, when a user speaks to a device, the microphones on the device receive not only the wavefront arriving directly from the speaker by the shortest path, but also reflections off of surrounding objects from different directions. The reflections can vary widely in real-world applications and cannot be easily accounted for by deterministic functions or models.[1] Early research on removing the effect of these reflections or reverberations were focused entirely on improving the intelligibility of speech, for example, when transmitted to a distant listener.[2, 3, 4] More recently, research began addressing the problem of mitigating the effect of reverberation on ASR.[5, 6, 7]

An approach to dereverberation that has demonstrated considerable promise[8, 9] is the weighted prediction error (WPE) algorithm[10]. This is reviewed in Section 2 with a simplified derivation. In Section 3, we extend the approach in [11] and derive an adaptive version that is suitable for multiple channels from a microphone array. The experimental setup for evaluating our approach is detailed in Section 4 and the observed results are reported in Section 5.

2. Dereverberation

Reverberation is generally modeled as the convolution of a Room Impulse Response (RIR) with the original signal. Assuming an array of M microphones and a linear system, this can be written as

$$y_i[n] = h_i[n] * x[n] \tag{1}$$

where $x[n]$ is the source signal, $y_i[n]$ is the signal received at the i^{th} microphone at time n, and $h_i[n]$ represents the impulse of the channel from the desired source to the i^{th} microphone.

This convolution introduces correlation into the received signal that otherwise would not be present. Dereverberation can be performed removing that correlation, otherwise known as whitening the signal. The sample received at microphone i at time n can be whitened by subtracting off the portion of that sample that can be predicted from the previous N' samples received at that microphone. This is done using a finite impulse response (FIR) linear prediction filter with taps $\hat{\mathbf{w}}_i$ to obtain [12, pp. 71–72]

$$\hat{y}_i[n] = y_i[n] - \sum_{k=0}^{N'-1} \hat{w}_i[k] y_i[n-k-1]. \tag{2}$$

The taps of the filter are obtained by minimizing the Euclidean norm of the prediction error:

$$\hat{\mathbf{w}}_i = \min_{\mathbf{w_i}} \sum_n |y_i[n] - \sum_{k=0}^{N'-1} w_i[k] y_i[n-k-1]|^2. \tag{3}$$

In this way, the correlation of the current sample with the previous samples is reduced and the reverberation is mitigated.

However, this is complicated by the fact that speech itself is correlated in time and it is certainly not desirable to remove the correlation that is inherent in the speech. Fortunately, the correlation time of the speech is often much smaller than the correlation time due to the RIR. Studies have shown that ASR is hurt most by the late-reverberation or the further-out components of the correlation induced by reverberation. The filter can be focused on removing the longer term correlation while minimizing the impact to the desired speech signal by ignoring the correlations shorter than some minimum, Δ' samples. This is accomplished by shifting the minimum delay of y_m by on the right side of (3) from 1 to Δ'.

Next, as written, (3) works to minimize the power of the reverberation. However, because the desired signal is nonstationary with time-varying power, it makes more sense to maximize the signal-to-reverberation ratio. To accomplish this, each term on the right side of (5) has been normalized by $\hat{\lambda}^2[n]$ where $\hat{\lambda}^2[n]$ is an estimate of the signal power at time n. A method for calculating this estimate will be discussed in a subsequent subsection.

Finally, multipath channels, like the ones modeled by the RIRs, often have many spectral nulls. Spectral nulls are difficult to undo and pose challenges in removing the correlation due to the channel. An advantage of a microphone array is that each microphone receives a signal subject to a different RIR. Therefore, even though content at one frequency may be wiped out in the signal at one microphone, that may not be the case for the signal at a different microphone. Thus, the dereverberation can be performed more effectively when the linear predictor simultaneously uses the signals received from all the microphones in predicting the current sample of *each* microphone.

http://dx.doi.org/10.21437/Interspeech.2017-1791

Equations (2) and (3) have been revised to reflect these three enhancements below

$$\hat{y}_i[n] = y_i[n] - \sum_{m=0}^{M-1} \sum_{k=0}^{N'-1} \hat{w}_{i,m}[k] y_m[n-k-\Delta'] \quad (4)$$

where

$$\hat{\mathbf{w}}_{i,m} = \min_{\mathbf{w}_i} \sum_n \frac{1}{\hat{\lambda}^2[n]} \left| y_i[n] - \sum_{m=0}^{M-1} \sum_{k=0}^{N'-1} w_{i,m}[k] y_m[n-k-\Delta'] \right|^2. \quad (5)$$

Rewriting (4) in matrix notation

$$\hat{\mathbf{y}}[n] = \mathbf{y}_i[n] - \hat{\mathbf{W}}^T \tilde{\mathbf{y}}_m \quad (6)$$

where

$$\hat{\mathbf{y}}[n] \equiv \begin{bmatrix} \hat{y}_0[n] & \hat{y}_1[n] & \cdots & \hat{y}_{M-1}[n] \end{bmatrix}, \quad (7)$$

$$\tilde{\mathbf{y}}[n] \equiv \begin{bmatrix} \tilde{\mathbf{y}}_0 & \tilde{\mathbf{y}}_1 & \cdots & \tilde{\mathbf{y}}_{M-1} \end{bmatrix}, \quad (8)$$

and

$$\tilde{\mathbf{y}}_i[n] \equiv \begin{bmatrix} y_i[n-\Delta] & \cdots & y_i[n-\Delta'-(N'-1)] \end{bmatrix} \quad (9)$$

The tap matrix is defined as

$$\mathbf{W} \equiv \begin{bmatrix} \mathbf{w}_0^T & \mathbf{w}_1^T & \cdots & \mathbf{w}_{M-1}^T \end{bmatrix} \quad (10)$$

where

$$\mathbf{w}_i \equiv \begin{bmatrix} \mathbf{w}_{i,0} & \mathbf{w}_{i,1} & \cdots & \mathbf{w}_{i,M-1} \end{bmatrix} \quad (11)$$

and

$$\mathbf{w}_{i,j} \equiv \begin{bmatrix} w_{i,j}[0] & w_{i,j}[1] & \cdots & w_{i,j}[N'-1] \end{bmatrix}. \quad (12)$$

Note, in this formulation, the required FIR filter $\mathbf{W}[n]$, with dimension $MN' \times MN'$, is prohibitively expensive to compute on-the-fly in real-time for most applications. Specifically, the associated matrix inversions are challenging. Fortunately, this can be simplified further.

2.1. Computing Efficiently In Frequency Domain

Dereverberation is often regarded as a linear process, meaning the content in one frequency does not influence any other. Thus, when an N'-tap filter is modeled in terms of F frequency bins, the equivalent tap lengths necessary for each frequency bin is N'/F. Since the computational complexity of matrix inversion is cubic in length, this decimation in frequency drastically reduces the computationally complexity by a factor $O(F^3)$, which can be substantial when, for example, $F = 512$, as in our application. In this application, the FFT size has been selected based on the coherence bandwidth of typical channels such the channel response in adjacent bins is roughly uncorrelated.

The analogous equation to (6) in the frequency domain, accounting for the fact that the signals are now complex, is written as

$$\hat{\mathbf{Y}}_l[k] = \mathbf{Y}_l[k] - \hat{\mathbf{W}}_l \tilde{\mathbf{Y}}_l[k] \quad (13)$$

where l represents the frequency bin, k represents the ST-DFT frame index, $\mathbf{Y}_l[k]$ is a vector that contains the the l^{th} frame of the ST-DFT of the received signal for each of the M microphones, and $\mathbf{Y}_l[k]$ is a matrix whose N columns are delayed versions of the ST-DFT of the received signal for each of the M microphone corresponding to frames $l-\Delta$ to $l-\Delta-(N-1)$.

The matrix of taps for the l^{th} frequency bin is found by optimizing

$$\hat{\mathbf{W}}_l = \min_{\mathbf{W}_l} \sum_n \frac{1}{\hat{\Lambda}_l^2[k]} \left| \hat{\mathbf{Y}}_l[k] - \mathbf{W}_l^H \tilde{\mathbf{Y}}_1 \right|^2. \quad (14)$$

$\hat{\Lambda}_l^2[k]$ is the estimate of the received signal averaged across the M microphones for frame k. This is estimated using a moving average as follows

$$\hat{\Lambda}_l^2[k] = \frac{1}{M(r_1+r_2+1)} \sum_{k=-r_1}^{r^2} \mathbf{Y}_l[k]^H \mathbf{Y}_l[k] \quad (15)$$

It is straightforward to solve (14) to find

$$\hat{\mathbf{W}}_l = \mathbf{R}_{\tilde{y}\tilde{y},l}^{-1} \mathbf{P} \quad (16)$$

where

$$\mathbf{R}_{\tilde{y}\tilde{y},l} \equiv \frac{1}{|K|} \sum_{k \in K} \frac{1}{\hat{\Lambda}_l^2[k]} \tilde{\mathbf{Y}}_k[k] \tilde{\mathbf{Y}}_k[k]^H \quad (17)$$

and

$$\mathbf{P} \equiv \frac{1}{|K|} \sum_{k \in K} \frac{1}{\hat{\Lambda}_l^2[k]} \tilde{\mathbf{Y}}_l[k] \mathbf{Y}_l[k]^H. \quad (18)$$

This is equivalent to the method presented in [10] with the scaled identity matrix approximation.

3. An Adaptive Algorithm

The algorithm as described above requires the entire utterance to be obtained before the taps can be calculated and, consequently, before dereverberation can be applied. For our ASR application, this latency is not acceptable. As such, it is desirable to obtain estimates of the tap values quickly as the speech signal arrives. It is useful for the tap values to be modified as the RIRs change due to speaker motion or other causes or just to account for the non-stationarity of the signal itself.

We extended the adaptive RLS-based algorithm for single channel [11] to the multichannel case, as presented below.

The error term at each step

$$\xi_l[k] = \sum_{k'=0}^{k} \frac{\alpha^{k-k'}}{\hat{\Lambda}_l^2[k']} \left| \hat{\mathbf{Y}}_l[k'] \right|^2 \quad (19)$$

is considered, where

$$\hat{\mathbf{Y}}_l[k] \equiv \mathbf{Y}_l[k] - \hat{\mathbf{W}}_l[k]^H \tilde{\mathbf{Y}}_l[k], \quad (20)$$

$$\hat{\mathbf{W}}_l[k] = \min_{\mathbf{W}_l[k]} \xi_l[k], \quad (21)$$

and α is a forgetting factor usually set in the range $0.98 < \alpha \leq 1$. α impacts the speed of adaptation and gives exponentially less weight to older error samples.

Similar to (13) and (14), the solution to (21) can easily be found at each step to be

$$\hat{\mathbf{W}}_l[k] = \mathbf{R}_{\tilde{y}\tilde{y},l}^{-1}[k] \mathbf{P}_l[k] \quad (22)$$

where

$$\mathbf{R}_{\tilde{y}\tilde{y},l}[k] \equiv \sum_{k'=0}^{k} \frac{\alpha^{k-k'}}{\hat{\Lambda}_l^2[k']} \tilde{\mathbf{Y}}_l[k'] \tilde{\mathbf{Y}}_l[k']^H \quad (23)$$

and

$$\mathbf{P}_l[k] \equiv \sum_{k'=0}^{k} \frac{\alpha^{k-k'}}{\hat{\Lambda}_l^2[k']} \tilde{\mathbf{Y}}_l[k'] \mathbf{Y}_l[k']^H. \quad (24)$$

Recognizing the recursive relationship

$$\mathbf{R}_{\tilde{y}\tilde{y},l}[k] = \alpha \mathbf{R}_{\tilde{y}\tilde{y},l}[k-1] + \tilde{\mathbf{Y}}_l[k] \tilde{\mathbf{Y}}_l[k]^H \quad (25)$$

enables the avoidance of taking the inverse of $\mathbf{R}_{\tilde{y}\tilde{y},l}[k]$ at each time step by applying the matrix inversion lemma [13] such that

$$\begin{aligned} \mathbf{S}_{\tilde{y}\tilde{y},l}[k] &\equiv \mathbf{R}_{\tilde{y}\tilde{y},l}^{-1}[k] \\ &= \frac{1}{\alpha}[\mathbf{S}_{\tilde{y}\tilde{y},l}(k-1) - \mathbf{K}_l[k] \tilde{\mathbf{Y}}_l[k]^H \mathbf{S}_{\tilde{y}\tilde{y},l}[k-1]] \end{aligned} \quad (26)$$

where

$$\mathbf{K}_l[k] \equiv \frac{\mathbf{S}_{\tilde{y}\tilde{y},l}[k-1] \tilde{\mathbf{Y}}_l[k]}{\alpha \hat{\Lambda}_l^2[k] + \tilde{\mathbf{Y}}_l[k]^H \mathbf{S}_{\tilde{y}\tilde{y},l}[k-1] \tilde{\mathbf{Y}}_l[k]}. \quad (27)$$

is the Kalman gain. With substitution of (26) and (24) into (22), the update equation can be obtained as

$$\hat{\mathbf{W}}_l[k] = \hat{\mathbf{W}}_l[k-1] + \mathbf{K}_l[k] \hat{\mathbf{Y}}_l^H[k]. \quad (28)$$

This update equation coupled with the filtering defined in (20) will be used for dereverberation in our experiments.

4. ASR Experiments

The dereverberation algorithm described above was evaluated in a speech recognition task where the speakers interacted with the device at a distance. In this task, the speech was collected from two microphones spaced about 70mm apart, each sampling at 16KHz.

4.1. Baseline CLP Acoustic Model

The baseline acoustic model consisted of 2 factored complex-valued linear projection (fCLP) layers followed by a cascade of 4 layers of Long Short Term Memory (LSTM) layers with 1024 nodes in each LSTM layer [14]. The input to the acoustic model consisted of complex-valued features computed as follows. From each channel, 32ms frames were extracted every 10ms and their complex-valued FFTs were computed. The frames from adjacent left and right contexts were stacked and the stacked frames were then decimated by a factor of 3. The fCLP layer consisted of 5 matrices corresponding to 5 filters, each filter was shared across the four frames in each stacked input. The resulting complex-valued vector was projected to a 128 component vector using another fCLP layer, which was then converted to a real-vector by applying a log compression on each component after computing its magnitude. These vectors formed the input to the LSTMs. The output of the LSTMs was projected to 512 dimensions using a DNN layer with a rectified linear activation function. These were then sent to a softmax layer to predict 8192 categories of tied context-dependent phone units.

The parameters of the model were learned using truncated backpropagation through time (BPTT) where the computational graph was unrolled for 20 time steps. The output state labels were delayed by 5 frames for better performance. The parameters of the model were updated using asynchronous stochastic gradient descent (ASGD) optimization distributed across about 444400 multiple workers. The models were optimized to minimize cross-entropy (CE) criterion. The weights for all layers are initialized using the Glorot-Bengio strategy [15], while those of the all LSTM layers are randomly initialized using a uniform distribution between -0.02 and 0.02. The learning rate was exponentially decayed from 0.004 by a factor of 0.1 every 240 billion frames.

4.2. Corpus

The training corpus consisted of about 22M anonymized English utterances from Google's voice search application with an average length of 4.6s. The single channel utterances were synthetically reverberated using a room simulator and then corrupted with additive background noise. The simulator was configured to sample from room dimensions with T_{60} ranging from 0 to 900 ms, with an average of about 500 ms. For each of these configurations, the simulator created an RIR for the given locations of speaker and the specified microphone array. The distance between the source and the array ranged from 1 to 7 meters. During training time, each utterance is convolved with 100 such RIRs to create synthetically reverberated utterances. The background noise types for corrupting the signal included music and ambient noise from YouTube and internal collection. Noise was injected to create an SNR ranging from 0 to 30 dB, with an average of about 11 dB. The models were evaluated on both simulated and real data. The simulated evaluation data was generated with settings that had no overlap with the training configurations.

5. Results

The tap parameter N represents the number of taps used to generate each microphone output per subband and per number of microphone inputs. With our system constrained to two microphones, N is the primary determinant of the complexity in the algorithm. In order to assess the impact of dereverberation and the dependence on this parameter, five different acoustic models were trained using the corpus described in the previous section. The first used no dereverberation in the front end, N = 0. The other four models were trained with values for N of 5, 10, 15, and 20. For each of the models, evaluations were performed with each of the 5 different values for N, making a total of 25 cases.

Evaluation results performed using 3 different anonymized data sets are presented. The first, labelled $clean$, has not been corrupted with noise or reverberation. The evaluation set called *rerecorded* was generated by rerecording utterances played through a mouth-simulator in a room to create reverberation. Finally, the evaluation set *rerecorded_noisy* was generated in a similar manner to *rerecorded* except that an additional noise source was added.

Other required parameters were selected by empirical optimization with respect to WER. For power estimation as defined in (15), r_1=1 and r_2=0 were used. The forgetting factor α was set to 0.9999.

In Figure 1, performance versus the minimum delay parameter Δ is shown for the two evaluation sets *rerecorded* and *rerecorded_noisy*. The y-axis shows the relative WER degradation versus the best performance obtained. Based on these results, a value of 2 for Δ was selected.

An example of tap coefficient adaptation over time is shown in Figure 2. The real part of the tap values are shown for one frequency bin for an 8 second utterance. Each line represents a different tap. It can be seen that roughly 2 seconds are needed to converge. Times where the tap values vary in a high fre-

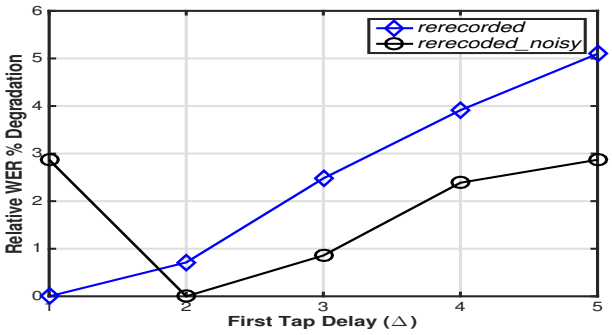

Figure 1: *Performance versus minimum delay parameter* Δ.

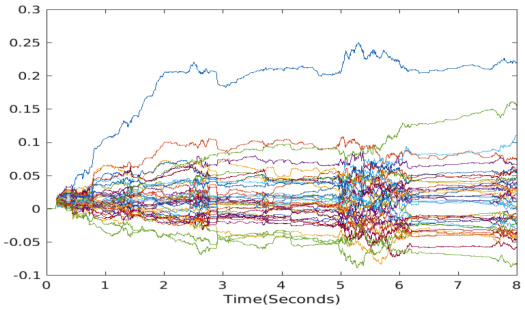

Figure 2: *Tap values versus time for one frequency bin and utterance.*

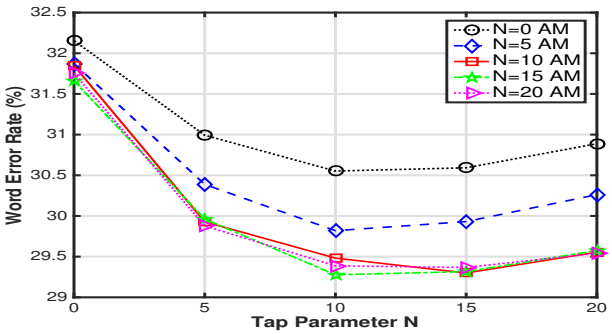

Figure 3: *Performance with rerecorded dataset for acoustic models trained with five different filter sizes.*

quency manner can be observed around 2.7s and 5-6s. These correspond to pauses in the speech.

Figure 3 plots the results for the *rerecorded_noisy* set for all of the considered training and evaluation configurations. Figure 4 does the same for the *rerecorded* set. Each line represents a different number of taps used during training. The x-axis values correspond to different values of N used during evlation. A few observations can be made. First, improvement seems to be made by training with the dereverberation algorithm even if it is not used in evaluation. This implies that there is benefit in cleaning up some of the reverberation present in the signals before training. Next, in terms of training, the biggest relative improvement appears to be gained by going from 0 to 5 taps. A smaller relative improvement can be garnered by training with more taps; however, 10, 15 or 20 taps seem to yield similar results. Finally, in evaluation, performance is not monotonically improve with the number of taps used. Using more than roughly 10 taps appears to degrade performance. We first observed this effect when looking at a model trained while using $N = 0$ taps but evaluated with more. One conjecture for the degraded performance as the number of taps used in evaluation increased, was that dereverberation was introducing a some artifacts in the processed signal that negatively impacted the ASR. Training with the algorithm reduces this mismatch during evaluations, but still the effect remains. Based on these results, it was decided that training and evaluating using a value of 10 for N provided a good balance between performance and complexity. Results comparing the output of such a model with one that had no dereverberation are presented in Table 1 for the three different evaluation sets. The results with the $clean$ set show that applying dereverberation did not negatively impact performance of non-reverberated data. With the *rereecorded*, the relative WER improvement is seenn to be about 5%. Finally, the relative improvement with the *record_noisy* set increased to over 8%. The reason for the additional improvement in noisy conditions could be due to the dereverberation allowing the multichannel processing being performed by the neural network to function better.

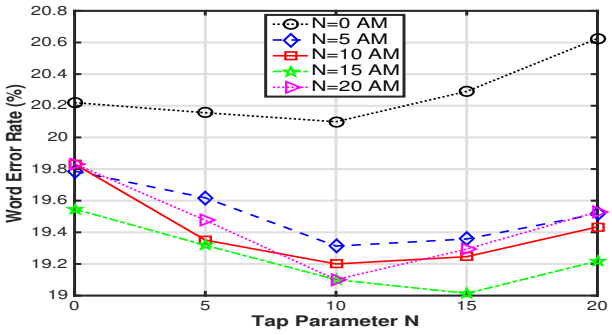

Figure 4: *Performance with rerecorded_noisy dataset for acoustic models trained with five different filter sizes.*

Table 1: *Dereverberation Impact on WER*

Model	clean	rerecorded	rerecorded-noisy
No Drvb	11.2	20.2	32.2
N =10	11.1	19.2	29.5

6. Conclusions

An adaptive multichannel dereverberation algorithm based upon WPE that is suitable for online ASR applications was developed. It was demonstrated that gains as high as 8% in relative error rate were obtained when training with noisy, reverberated data and evaluated on large, real world data sets. It was also shown that while benefits could be gained by applying the algorithm at evaluation, the gains are much more significant if the model is trained using the algorithm.

7. References

[1] H. Kuttruff, *Room acoustics*. Crc Press, 2016.

[2] R. Zelinski, "A microphone array with adaptive post-filtering for noise reduction in reverberant rooms," in *Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on*. IEEE, 1988, pp. 2578–2581.

[3] M. Miyoshi and Y. Kaneda, "Inverse filtering of room acoustics," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 2, pp. 145–152, 1988.

[4] D. Bees, M. Blostein, and P. Kabal, "Reverberant speech enhancement using cepstral processing," in *Acoustics, Speech, and Signal Processing, 1991. ICASSP-91., 1991 International Conference on*. IEEE, 1991, pp. 977–980.

[5] T. Yoshioka, A. Sehr, M. Delcroix, K. Kinoshita, R. Maas, T. Nakatani, and W. Kellermann, "Making machines understand us in reverberant rooms: Robustness against reverberation for automatic speech recognition," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 114–126, 2012.

[6] K. Kinoshita, M. Delcroix, T. Yoshioka, T. Nakatani, A. Sehr, W. Kellermann, and R. Maas, "The reverb challenge: A common evaluation framework for dereverberation and recognition of reverberant speech," in *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2013 IEEE Workshop on*. IEEE, 2013, pp. 1–4.

[7] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, "The third chimespeech separation and recognition challenge: Dataset, task and baselines," in *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*. IEEE, 2015, pp. 504–511.

[8] M. Delcroix, T. Yoshioka, A. Ogawa, Y. Kubo, M. Fujimoto, N. Ito, K. Kinoshita, M. Espi, T. Hori, T. Nakatani *et al.*, "Linear prediction-based dereverberation with advanced speech enhancement and recognition technologies for the reverb challenge," in *REVERB Workshop*, 2014.

[9] T. Yoshioka, N. Ito, M. Delcroix, A. Ogawa, K. Kinoshita, M. Fujimoto, C. Yu, W. J. Fabian, M. Espi, T. Higuchi *et al.*, "The ntt chime-3 system: Advances in speech enhancement and recognition for mobile multi-microphone devices," in *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*. IEEE, 2015, pp. 436–443.

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

[11] T. Yoshioka, H. Tachibana, T. Nakatani, and M. Miyoshi, "Adaptive dereverberation of speech signals with speaker-position change detection." in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2009, pp. 3733–3736.

[12] J. R. Barry and D. G. Messerschmitt, *Digital Communication*. New York: Springer, 2012.

[13] P. S. Diniz, *Adaptive Filtering: Algorithms and Practical Implementation*. Norwell, MA, USA: Kluwer Academic Publishers, 2002.

[14] T. N. Sainath, A. Narayanan, R. J. Weiss, K. W. Wilson, M. Bacchiani, and I. Shafran, "Improvements to Factorized Neural Network Multichannel Models," in *Proc. Interspeech*, 2016.

[15] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. International Conference on Artificial Intelligence and Statistics (AISTATS10). Society for Artificial Intelligence and Statistics*, 2010.