SPECTRAL DISTORTION MODEL FOR TRAINING PHASE-SENSITIVE DEEP-NEURAL NETWORKS FOR FAR-FIELD SPEECH RECOGNITION

Chanwoo Kim[1], *Tara Sainath*[1], *Arun Narayanan*[1] *Ananya Misra*[1], *Rajeev Nongpiur*[2], *and Michiel Bacchiani*[1]

[1]Google Speech, [2]Nest

{chanwcom, tsainath, arunnt, amisra, rnongpiur, michiel}@google.com

ABSTRACT

In this paper, we present an algorithm which introduces phase-perturbation to the training database when training phase-sensitive deep neural-network models. Traditional features such as log-mel or cepstral features do not have have any phase-relevant information. However more recent features such as raw-waveform or complex spectra features contain phase-relevant information. Phase-sensitive features have the advantage of being able to detect differences in time of arrival across different microphone channels or frequency bands. However, compared to magnitude-based features, phase information is more sensitive to various kinds of distortions such as variations in microphone characteristics, reverberation, and so on. For traditional magnitude-based features, it is widely known that adding noise or reverberation, often called Multistyle-TRaining (MTR) , improves robustness. In a similar spirit, we propose an algorithm which introduces spectral distortion to make the deep-learning model more robust against phase-distortion. We call this approach Spectral-Distortion TRaining (SDTR) and Phase-Distortion TRaining (PDTR). In our experiments using a training set consisting of 22-million utterances, this approach has proved to be quite successful in reducing Word Error Rates in test sets obtained with real microphones on Google Home.

***Index Terms*—** Far-field Speech Recognition, Deep-Neural Network Model, Phase-Sensitive Model Spectral Distortion Model, Spectral Distortion Training, Phase Distortion Training

1. INTRODUCTION

After the breakthrough of deep learning technology [1, 2, 3, 4, 5, 6], speech recognition accuracy has improved dramatically. Recently, speech recognition systems have begun to be employed not only in smart phones and Personal Computers (PCs) but also in standalone devices in far-field environments. The latter examples include voice assistant systems such as Amazon Alexa or Google Home [7, 8]. In far-field speech recognition, the impact of noise and reverberation is much larger than near-field cases. Traditional approaches to far-field speech recognition include noise robustness feature extraction algorithms [9, 10], on-set enhancement algorithms [11, 12], and multi-microphone approaches [13, 14]. A recent promising approach to this problem is using multi-channel features which contain temporal information between two microphones [15]. It has been known that the Inter-microphone Time Delay (ITD) or Phase Difference (PD) between two microphones may be used to identify the Angle of Arrival (AoA) [16, 17]. The Inter-microphone Intensity Difference (IID) may also serve as a cue for determining the AoA [18].

However, it is far more difficult to collect the sufficient amount of utterances to train deep-neural networks for a particular model of multi-channel devices than a model of single-channel devices. Since multi-channel utterances have device-dependent characteristics such as the number of microphones and the distance between microphones, we need to re-collect multi-channel utterances for each device model. To tackle this problem, we developed the "room simulator" [7] to generate simulated multi-microphone utterances for training multi-channel deep-neural network model. Multi-style Training (MTR) driven by this room simulator was employed in training the acoutic model for Google Home [7, 8].

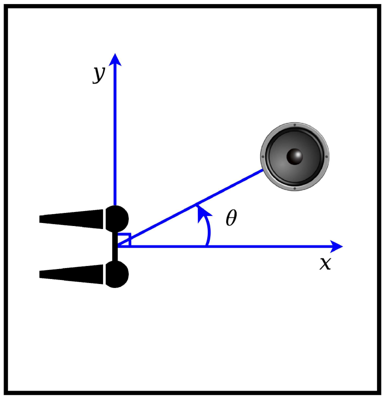
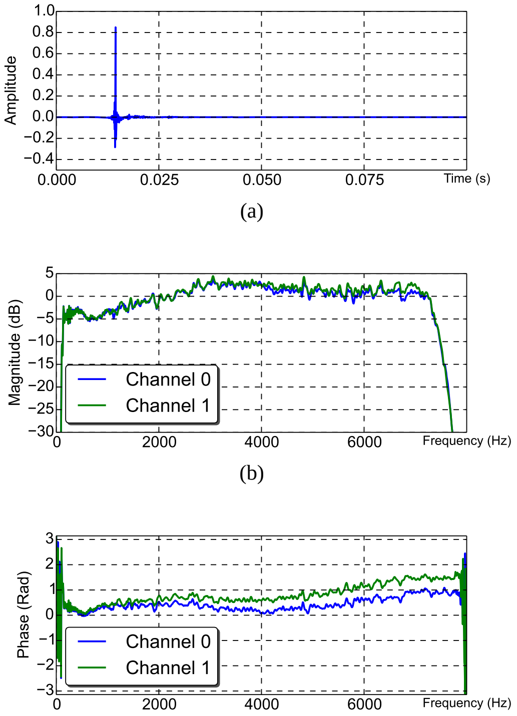


Fig. 1: A microphone array with two microphones in an anechoic chamber. The azimuth angle with respect to the x-axis is θ.

However, this room simulator still has its limitations. It assumes that all the microphones are ideal, which means that they all have zero-phase all-pass responses. Even though this assumption is very convenient, it is not true with actual microphones due to microphone spectrum distortion. In addition, there may be other distortion factors such as electrical noise in the circuit, acoustic auralization effect from the hardware surface, and various vibrations. In the conventional MTR, we usually add additive noise and reverberation effects to the training set. However we do not usually model the spectral or phase distortion across different filter bank or microphone channels. In this paper, we propose an algorithm that makes phase-sensitive deep learning model more robust by adding phase distortion to the training set.

2. SPECTRAL DISTORTION IN REAL HARDWARE RECORDING

Before discussing the Phase-Distortion TRaining (PDTR)/ Spectral-Distortion TRaining (SDTR) model in detail in the next section, we briefly examine how the magnitude-phase distortion occurs with real recording. Let us consider a configuration for recording in an "ane-



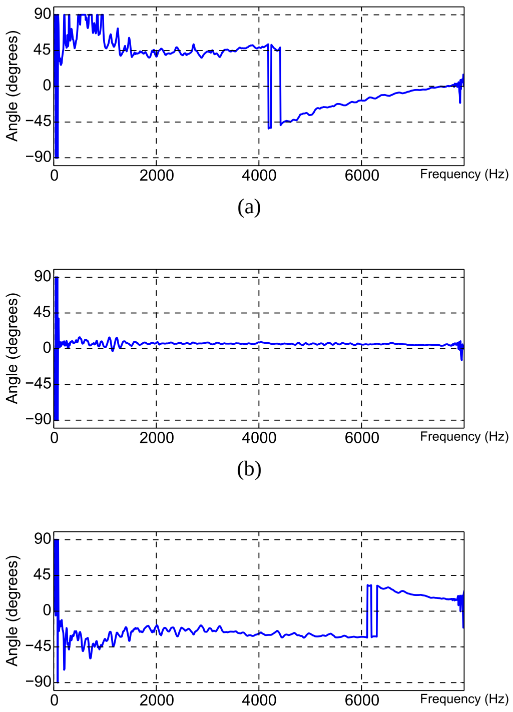
(c)

Fig. 2: *(a) A microphone response recorded in an anechoic chamber. (b) Magnitudes of two microphone transfer functions from a single microphone array, and (c) Phases of two microphone transfer functions from a single microphone array.*

choic" chamber as shown in Fig. 1. The target sound source is a high-quality loud speaker in this anechoic chamber without other noisy sound sources. In this setup, the distance between two microphones was 5.1 cm. The distance from the target sound source to the microphone array is two meters. Let us denote the azimuth angle by θ, which is the angle between the perpendicular bisector to the line connecting two microphones and the line connecting the center of the array to the loud speaker. Fig. 2 show a two-channel microphone response obtained in this configuration with $\theta = 0^o$. Impulse responses were obtained using the method proposed by A. Farina [19]. As shown in Fig. 2b and Fig. 2c, the magnitude and the phase responses are different from the ideal responses, which is the all-pass zero-phase response. Since the azimuth angle θ is zero in this case, the phase components from two microphones are expected to be the same. However, as shown in Fig. 2c, the phases from two microphones turned out to be somewhat different especially in high frequencies. The relationship between the phase difference and Angle of Arrival (AoA) θ at discrete frequency k is given by the following equation [20, 21]:

$$\theta(\omega_k) = \arcsin\left(\frac{\Delta\phi(\omega_k)c_0}{\omega_k d_m f_s}\right) \tag{1}$$

where ω_k is the discrete frequency, $\Delta\phi(\omega_k)$ is the phase difference at this frequency ω_k. f_s, d_m and c_0 are sampling rates in *Hz*, the distance between two microphones, and the speed of sound, respectively. Using this equation, the estimated Angle of Arrivals(AoAs) are obtained in Fig. 3 for three different locations with the true azimuth angle θ of 45^o, 0^0, and -45^o respectively. From this figure, the phase distortion of real hardware may be easily observed.



(c)

Fig. 3: *The estimated Angle of Arrival when the true azimuth angle θ is (a) 45^o (b) 0^o, and (c) -45^o.*

3. SPECTRAL-DISTORTION TRAINING (SDTR) FOR PHASE-SENSITIVE DEEP NEURAL NETWORKS

In this section, we explain the entire structure of Spectral-Distortion TRaining (SDTR), and its subsets Phase-Distortion TRaining (PDTR) and Magnitude Distortion TRaining (MDTR). PDTR is a subset of SDTR where distortion is only applied to the phase component without modifying the magnitude component of complex features. MDTR is a subset of SDTR where distortion is applied only to the magnitude component of such features. PDTR is devised for enhancing the robustness of phase-sensitive multi-microphone neural network models such as [8, 22].

3.1. Acoustic model training

Fig. 4 shows the structure of the acoustic model pipeline used for training multi-channel deep neural networks. This pipeline is a modified version of the architecture described in [15]. The pipeline consists of five layers of Long Short-Term Memory (LSTM) [23, 24]. Each fully connected Deep Neural Network (DNN) layer is placed before and after this stack of LSTM layers. We use the Complex Fast Fourier Transform (CFFT) feature whose window size is 32 *ms*, and the interval between successive frame is 10 *ms*. We use the FFT size of $N = 512$. Since FFT of real signals have Hermitian symmetry, we use the lower half spectrum whose size given by $N/2+1 = 257$. Since it has been shown that long-duration features represented by overlapping features are helpful [25], four frames are stacked together. Thus we use a context dependent feature consisting of 2056

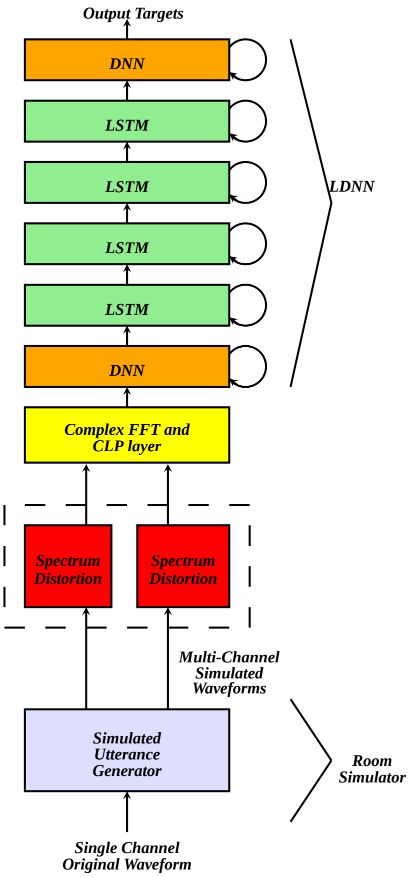

Output Targets
DNN
LSTM
LSTM
LSTM
LSTM
DNN
LDNN
Complex FFT and CLP layer
Spectrum Distortion
Spectrum Distortion
Multi-Channel Simulated Waveforms
Simulated Utterance Generator
Room Simulator
Single Channel Original Waveform

Fig. 4: A pipeline containing the Spectrum Distortion Model (SDM) for training deep-neural networks for acoustic modeling. This acoustic model pipeline is modified based on [15].

complex numbers given by 257 (the size of the lower half spectrum) x 2 (number of channels) x 4 (number of stacked frames). We apply the Complex Linear Projection (CLP) described in [26] on this two-channel complex FFT feature. The output state label is delayed by five frames, since it was observed that information about future frames improves the prediction of the current frame [27].

3.2. Spectral Distortion Model (SDM)

To make the phase-sensitive multi-channel feature more robust, we add the Spectral Distortion Model (SDM) to each channel. The actual equation we use for this SDM is described in (2). The first stage of the pipeline in Fig 4 is the room simulator to generate millions of different utterances in millions different virtual rooms [7]. The spectrum distortion procedure is summarized by the following pseudo-code:

```
for each utterance in the training set do
    for each microphone channel of the utterance do
        Creates the random transfer function in (2).
        Performs Short-Time Fourier Transform (STFT).
        Applies this transfer function to the spectrum.
        Re-synthesizes the output microphone-channel using OverLap Addition (OLA).
    end for
end for
```

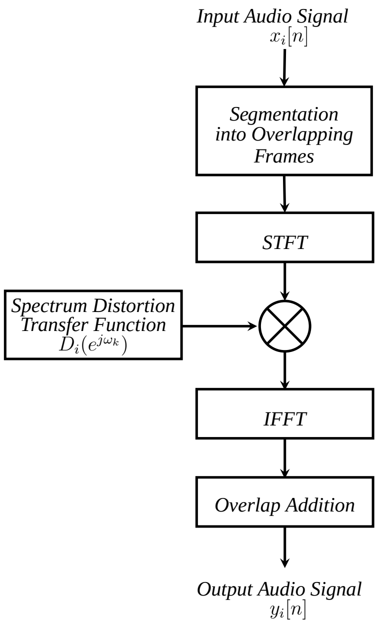


Fig. 5: A diagram showing the structure of applying magnitude-phase distortion to each microphone channel. Note that i in this diagram denotes the microphone channel index.

For each microphone channel of each utterance, we create a transfer function to distort the spectrum. Unlike the case of adding random noise to speech signals, once transfer functions are created for a specific utterance, then these transfer functions are maintained for that utterance. We use the spectral distortion model described by the following equation:

$$D_l(e^{j\omega_k}) = e^{am_l(k)+jp_l(k)}, \qquad 0 \le k \le \frac{K}{2}, \quad 0 \le l \le L-1. \tag{2}$$

where l is the microphone channel index and L is the number of microphone channels. In the case of Google Home, since we use two microphones, $L = 2$. ω_k is the discrete frequency index, which is defined by $\omega_k = \frac{2\pi k}{K}$ where K is the Discrete Fourier Transform(DFT) size. $m_l(k)$ and $p_l(k)$ are Gaussian random samples pulled from the following Gaussian distributions $\mathbf{m}$ and $\mathbf{p}$ respectively:

$$\mathbf{m} \sim \mathcal{N}(0, \sigma_m^2) \tag{3a}$$
$$\mathbf{p} \sim \mathcal{N}(0, \sigma_p^2) \tag{3b}$$

The scaling coefficient a in (2) is defined by the following equation:

$$a = \ln(10.0)/20.0 \tag{4}$$

This scaling coefficient a is introduced to make σ_m the standard deviation of the magnitude dB. From (2), it is evident that $m_l(k)$ and $p_l(k)$ are related to the magnitude and phase distortion, respectively. The magnitude distortion is accomplished by the $e^{a_l(k)}$ term. Since $e^{a_l(k)} = 20.0^{a_l(k)\log_{20}(e)}$, the standard deviation of magnitude in Decibel is given by $\log_{20}(e)\sigma_n$. For the phase term, since the complex exponential has a period of 2π, the distribution actually becomes the wrapped Gaussian distribution which is shown in [28]. After creating the spectrum distortion transfer function $D_l\left(e^{j\omega_k}\right)$ in (2), we process each channel using the structure shown in Fig. 5.

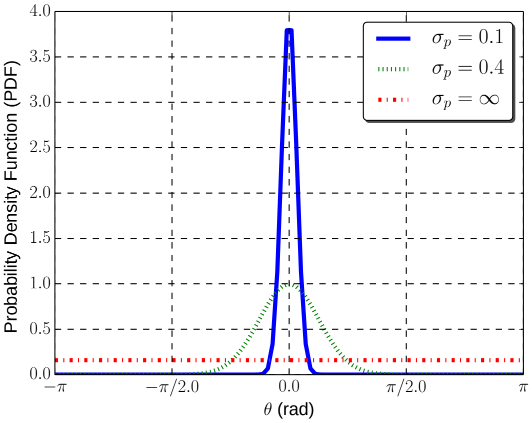


Fig. 6: The probability density functions(PDFs) of the wrapped Gaussian distribution with different values of σ_p.

We apply the Hanning window instead of the more frequently-used Hamming window to each frame. We use the Hanning window to better satisfy the OverLap-Add (OLA) constraint. After multiplying the complex spectrum of each frame with the spectrum distortion transfer function $D_l\left(e^{j\omega_k}\right)$ in the frequency domain, the time-domain signal is re-synthesized using OverLap-Add(OLA) synthesis. This processing is shown in detail in Fig. 5. The reason for going back to the time domain is because we use Complex Fast Fourier Transform (CFFT) as feature whose frame size is 32 *ms* in Fig. 4. We segment each microphone channels into successive frames with the frame length of 10 *ms*. The period between successive frames is 5 *ms*. These frame length is chosen based on the experimental results in Sec. 3.3. The spectrum distortion effects from $D_l(e^{j\omega_k})$ in Fig. 5 is not removed by either the conventional log spectral mean normalization nor Cepstral Mean Normalization (CMN), since we use complex features and SDM includes complex exponential.

3.3. Word Error Rate(WER) dependence on σ_m, σ_p and frame length

Table 1 shows speech recognition results in terms of Word Error Rate (WER) using the PDTR training with different values of σ_p and frame lengths. The PDF functions of wrapped Gaussian distributions corresponding to these σ_p values are depicted in Fig. 6. The configurations for speech recognition training and evaluation will bewill be described in detail in Sec. 4. The evaluation set used in Table 1 through Table 4 is the combinations of five rerecording sets described in Sec. 4, which are three rerecording sets using different Google Home devices, and two rerecording sets in presence of Youtube noise and interfering speakers, respectively. The best result in Table 1 (49.77 % WER) is obtained when $\sigma_p = \infty$ with the window length of 32 *ms*. Table 2 shows Word Error Rates (WERs) using the MDTR training on the same test set using the same configuration as Table 1 with different σ_m values. In these experiments, we observe significant improvement over the baseline system which shows WER of 62.0 % on the same test set.

When training acoustic models for Google Home, we have been using data generated by the room simulator [7]. Table 3 and Table 4 show the WERs when the PDTR or MDTR is applied with the Multi-style TRaining (MTR) driven by this "room simulator". Even though relative improvement over the MTR baseline in Table 3 and Table 4 is less than the relative improvement in Table 1 and Table 2, we still obtain substantial improvement over the baseline.

Table 1: Word Error Rates (WERs) using the PDTR training

	baseline	$\sigma_p = 0.1$	$\sigma_p = 0.4$	$\sigma_p = \infty$
frame length	62.00%			
10 *ms*		57.16 %	56.74 %	54.03 %
32 *ms*		59.03 %	57.14 %	**49.77** %

Table 2: Word Error Rates (WERs) using the MDTR training

	baseline	$\sigma_m = 0.5$	$\sigma_m = 1.0$	$\sigma_m = 2.0$
frame length	62.00%			
10 *ms*		60.39 %		
32 *ms*		**52.21** %	53.03 %	55.37 %

Table 3: Word Error Rates (WERs) using the PDTR and MTR training

	MTR baseline	$\sigma_p = 0.1$	$\sigma_p = 0.4$	$\sigma_p = \infty$
frame length	29.34%			
10 *ms*		28.63 %	**28.40** %	29.78 %
32 *ms*			29.28 %	30.34 %
160 *ms*		28.69 %	31.36 %	37.82 %

Table 4: Word Error Rates (WERs) using the MDTR and MTR training

	MTR baseline	$\sigma_m = 0.5$	$\sigma_m = 1.0$	$\sigma_m = 2.0$
frame length	29.34%			
10 *ms*		31.13 %		
32 *ms*		**28.46** %	28.78 %	28.70 %
160 *ms*			29.01 %	29.55 %

From the results from Table 1 to Table 4, PDTR turns out to be more effective than the MDTR. We also tried combinations of PDTR and MDTR, but we could not obtain results better than the PDTR itself. Thus, in our system, we adopt PDTR with $\sigma_p = 0.4$ as the default Spectral Distortion Model (SDM).

4. EXPERIMENTAL RESULTS

In this section, we shows experimental results obtained with the SDTR training. For training, we used an anonymized 22-million English utterances (18,000-hr), which are hand-transcribed. For training the acoustic model, instead of directly using these utterances, we use the room simulator described in [7] to generate simulated utterances for our hardware. In the simulator, we use 7.1 cm distance between two microphones. For each utterance, one room configuration was selected out of three million room configurations with varying room dimension, and varying the target speaker and noise source locations. In each room, number of noise sources may be up to three. This configuration changes for each training utterance. After every epoch, we apply a different room configuration to the utterance so that each utterance may be regenerated in somewhat different ways. For additive noise, we used Youtube videos, recordings of daily activities, and recordings at various locations inside cafes. We picked

Table 5: Word Error Rates (WERs) obtained with the PDTR ($\sigma_m = 0.0,\ \sigma_p = 0.4$) training

	Baseline	PDTR	Relative improvement (%)
Original Test Set	12.02 %	12.32 %	-2.53 %
Simulated Noise Set 1	20.34 %	20.72 %	-1.86 %
Simulated Noise Set 2	47.88 %	46.69 %	2.50 %
Rerecording using ”Device 1”	50.14 %	42.87 %	14.51 %
Rerecording using ”Device 2”	48.65 %	43.32 %	10.95 %
Rerecording using ”Device 3”	56.27 %	51.30 %	8.83 %
Rerecording with youtube background noise	76.01 %	71.42 %	6.04 %
Rerecording with multiple interfering speaker noise	78.95 %	74.80 %	5.26 %
Average from rerecording sets	**62.00** %	**56.74** %	**8.48** %

Table 6: Word Error Rates (WERs) obtained with
the PDTR ($\sigma_m = 0.0,\ \sigma_p = 0.4$) training combined with room-simulator based MTR in [7]

	MTR	PDTR + MTR	Relative improvement (%)
Original Test Set	11.97 %	11.99 %	-0.17 %
Simulated Noise Set 1	14.73 %	15.03 %	-2.04 %
Simulated Noise Set 2	19.55 %	20.29 %	-3.79 %
Rerecording using ”Device 1”	21.89 %	20.86 %	4.71 %
Rerecording using ”Device 2”	22.23 %	21.29 %	4.22 %
Rerecording using ”Device 3”	22.05 %	21.65 %	1.81 %
Rerecording with youtube background Noise	34.83 %	34.21 %	1.78 %
Rerecording with multiple interfering speaker noise	44.79 %	44.00 %	1.76 %
Average from rerecording sets	**29.34** %	**28.40** %	**3.20** %

up the SNR value from a distribution ranging from 0 dB to 30 dB, with an average of 11 dB. We used reverberation time varying from 0 ms up to 900.0 ms with an average of 500 ms. To model reverberation, we employed the image method [29]. We constructed $17^3 - 1 = 4912$ virtual sources for each real sound source. The acoustic model was trained using the Cross-Entropy (CE) minimization as the objective function after aligning each utterance. The Word Error Rates (WERs) are obtained after 120 million steps of acoustic model training.

For evaluation, we used around 15-hour of utterances (13,795 utterances) obtained from anonymized voice search data. Since our objective is evaluating speech recognition performance when our system is deployed on the actual hardware, we re-recorded these utterances using our actual devices in a real room at five different locations. We used three different devices (named ”Device 1”, ”Device 2”, and ”Device 3”) as shown in Table 5 and 6. These three devices are prototype Google Home devices. Each device is placed at five different positions and orientations in a real room with mild reverberation (around 200 *ms* reverberation time). The entire 15-hour test utterances are rerecorded using each device. We also prepared two additional rerecorded sets in presence of Youtube noise and interfering speaker noise played by real loud speakers. The noise level varies, but it is usually between 0 and 15 dB SNR. Each of these rerecording sets also contains the same 15-hour long utterances recorded at five different locations. In total, there are five rerecording test sets in Table 5 and Table 6. In addition to the real rerecorded sets, we evaluated performance on two simulated noise sets from the same utterances using the “room simulator” in [7]. Note that in these two simulated noise sets, we assume that all microphones are identical without any magnitude or phase distortion. We are interested in performance on the rerecorded sets, but we also included these simulated noise sets for comparison purporse.

In Table 5, we compare the performance of the baseline system with the PDTR system. Based on our analaysis in Sec. 3, we use the PDTR of $\sigma_m = 0.0, \sigma_p = 0.4$ in (3) as our Spectral Distortion Model (SDM). As shown in these two tables, PDTR shows significantly better results than the baseline for rerecorded sets while doing on par or slightly worse on two simulated noisy sets, which is expected.

5. CONCLUSIONS

In this paper, we described the PDTR algorithm to add phase distortion to the training set to make the trained phase-sensitive neural net model robust against various distortion in signals. Our experimental results show that the phase-sensitive neural-net trained with PDTR is much more robust against real-world distortions.

6. REFERENCES

[1] M. Seltzer, D. Yu, and Y.-Q. Wang, “An investigation of deep neural networks for noise robust speech recognition,” in *Int. Conf. Acoust. Speech, and Signal Processing*, 2013, pp. 7398–7402.

[2] D. Yu, M. L. Seltzer, J. Li, J.-T. Huang, and F. Seide, “Feature learning in deep neural networks - studies on speech recognition tasks,” in *Proceedings of the International Conference on Learning Representations*, 2013.

[3] V. Vanhoucke, A. Senior, and M. Z. Mao, “Improving the speed of neural networks on CPUs,” in *Deep Learning and Unsupervised Feature Learning NIPS Workshop*, 2011.

[4] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups,” *IEEE Signal Processing Magazine*, vol. 29, no. 6, Nov.

[5] T. Sainath, R. J. Weiss, K. W. Wilson, B. Li, A. Narayanan, E. Variani, M. Bacchiani, I. Shafran, A. Senior, K. Chin, A. Misra, and C. Kim, “Multichannel signal processing with deep neural networks for automatic speech recognition,” *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, Feb. 2017.

[6] ——, “Raw Multichannel Processing Using Deep Neural Networks,” in *A Book Chapter in New Era for Robust Speech Recognition: Exploiting Deep Learning*, 2017.

[7] C. Kim, A. Misra, K.K. Chin, T. Hughes, A. Narayanan, T. Sainath, and M. Bacchiani, “Generation of simulated utterances in virtual rooms to train deep-neural networks for far-field speech recognition in Google Home,” in *INTERSPEECH-2017*, Aug. 2017 (accepted).

[8] B. Li, T. Sainath, A. Narayanan, J. Caroselli, M. Bacchiani, A. Misra, I. Shafran, H. Sak, G. Pundak, K. Chin, K-C Sim, R. Weiss, K. Wilson, E. Variani, C. Kim, O. Siohan, M. Weintraub, E. McDermott, R. Rose, and M. Shannon, “Acoustic modeling for Google Home,” in *INTERSPEECH-2017*, Aug. 2017 (accepted).

[9] C. Kim and R. M. Stern, “Power-Normalized Cepstral Coefficients (PNCC) for Robust Speech Recognition,” *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, pp. 1315–1329, July 2016.

[10] U. H. Yapanel and J. H. L. Hansen, “A new perceptually motivated MVDR-based acoustic front-end (PMVDR) for robust automatic speech recognition,” *Speech Communication*, vol. 50, no. 2, pp. 142–152, Feb. 2008.

[11] C. Kim and R. M. Stern, “Nonlinear enhancement of onset for robust speech recognition,” in *INTERSPEECH-2010*, Sept. 2010, pp. 2058–2061.

[12] C. Kim, K. Chin, M. Bacchiani, and R. M. Stern, “Robust speech recognition using temporal masking and thresholding algorithm,” in *INTERSPEECH-2014*, Sept. 2014, pp. 2734–2738.

[13] C. Kim, K. Eom, J. Lee, and R. M. Stern, “Automatic selection of thresholds for signal separation algorithms based on interaural delay,” in *INTERSPEECH-2010*, Sept. 2010, pp. 729–732.

[14] R. M. Stern, E. Gouvea, C. Kim, K. Kumar, and H .Park, “Binaural and multiple-microphone signal processing motivated by auditory perception,” in *Hands-Free Speech Communication and Microphone Arrays, 2008*, May. 2008, pp. 98–103.

[15] T. Sainath, R. Weiss, K. Wilson, A. Narayanan, and M. Bacchiani, “Factored spatial and spectral multichannel raw waveform CLDNNs,” in *IEEE Int. Conf. Acoust., Speech, Signal Processing*, March 2016, pp. 5075–5079.

[16] C. Kim, K. Kumar, B. Raj, and R. M. Stern, “Signal separation for robust speech recognition based on phase difference information obtained in the frequency domain,” in *INTERSPEECH-2009*, Sept. 2009, pp. 2495–2498.

[17] C. Kim, K. Kumar, and R. M. Stern, “Binaural sound source separation motivated by auditory processing,” in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, May 2011, pp. 5072–5075.

[18] H. S. Colburn and A. Kulkarni, “Models of sound localization,” in *Sound Source Localization*, A. N. Popper and R. R. Fay, Eds. Springer-Verlag, 2005, pp. 272–316.

[19] A. Farina, “Simultaneous Measurement of Impulse Response and Distortion with a Swept-Sine Technique,” in *Audio Engineering Society Convention 108*, Feb. 2000, p. 7121. [Online]. Available: http://www.aes.org/e-lib/browse.cfm?elib=10211

[20] C. Kim and K. Chin, “Sound source separation algorithm using phase difference and angle distribution modeling near the target,” in *INTERSPEECH-2015*, Sept. 2015, pp. 751–755.

[21] C. Kim, C. Khawand, and R. M. Stern, “Two-microphone source separation algorithm based on statistical modeling of angle distributions,” in *IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, March 2012, pp. 4629–4632.

[22] T. Sainath, R. Weiss, K. Wilson, A. Narayanan, and M. Bacchiani, “Learning the Speech Front-end With Raw Waveform CLDNNs,” in *INTERSPEECH-2015*, Sept. 2015, pp. 1–5.

[23] H. Sak, A. Senior, and F. Beaufays, “Long short-term memory recurrent neural network architectures for large scale acoustic modeling,” in *INTERSPEECH-2014*, Sept. 2014, pp. 338–342.

[24] S. Hochreiter and Jürgen Schmidhuber, “Long Short-term Memory,” *Neural Computation*, no. 9, pp. 1735–1780, Nov. 1997.

[25] H. Sak, A. Senior, K. Rao, and F. Beaufays, “Fast and Accurate Recurrent Neural Network Acoustic Models for Speech Recognition,” in *INTERSPEECH-2015*, Sept. 2015, pp. 1468–1472.

[26] E. Variani, T. Sainath, I. Shafran, and M. Bacchiani, “Complex Linear Projection (CLP): A Discriminative Approach to Joint Feature Extraction and Acoustic Modeling,” Sept. 2016, pp. 808–812.

[27] T. Sainath, O. Vinyals, A. Senior, and H. Sak, “Convolutional, Long Short-Term Memory, Fully Connected Deep Neural Networks,” in *IEEE Int. Conf. Acoust., Speech, Signal Processing*, Apr. 2015, pp. 4580–4584.

[28] E. Breitenberger, “Analogues of the normal distribution on the circle and the sphere,” *Biometrika*, vol. 50, no. 1/2, pp. 81–88, June 1963.

[29] J. Allen and D. Berkley, “Image method for efficiently simulating small-room acoustics,” *J. Acoust. Soc. Am.*, vol. 65, no. 4, pp. 943–950, April 1979.