DESIGN AND OPTIMIZATION OF A SPEECH RECOGNITION FRONT-END FOR DISTANT-TALKING CONTROL OF A MUSIC PLAYBACK DEVICE

Ramin Pichevar[1]*, Ali Ziaei*[2*]*, Jason Wung*[1]*, Daniele Giacobello*[1]*, and Joshua Atkins*[1]

[1]Beats Electronics, LLC, Culver City, CA
[2]Center for Robust Speech Systems (CRSS), Univ. of Texas at Dallas, Richardson, TX
ramin.pichevar@beatsbydre.com

ABSTRACT

This paper addresses the challenging scenario for the distant-talking control of a music playback device, a common portable speaker with four small loudspeakers in close proximity to one microphone. The user controls the device through voice, where the speech-to-music ratio can be as low as -40 dB during music playback. We propose a speech enhancement front-end that relies on known robust methods for echo cancellation, double-talk detection, and noise suppression, as well as an adaptive quasi-binary mask that is well suited for speech recognition. The optimization of the front-end system is then formulated as a large scale nonlinear programming problem where the recognition rate is maximized and the optimal values for the system parameters are found through a genetic algorithm. The back-end speech recognition system is designed using two methodologies: deep neural networks and subspace Gaussian mixture models. We validate our methodology by testing over the TIMIT database for different music playback levels and noise types. Finally, we show that the proposed front-end allows a natural interaction with the device for limited-vocabulary voice commands.

Index Terms— Speech recognition, echo cancellation, speech enhancement, genetic algorithm, neural networks

1. INTRODUCTION

The human interaction paradigm with music playback devices has seen a dramatic shift as devices get smaller and more portable. Well-established interaction media such as remote controls are no longer adequate. Automatic speech recognition (ASR) interfaces offer a natural solution to this problem, where these devices are typically used in hands-busy, mobility-required scenarios [1]. Performing ASR on these small devices is highly challenging due to the music playback itself, the environmental noise, and the general environmental acoustics, e.g., reverberation [2]. In particular, due to the severe degradation of the input signal, the ASR performance drops significantly when the distance between the user and the microphone increases [3]. In the past decade, the literature on distant-talking speech interfaces provided several solutions to the problem, e.g., the DICIT project [4]. However, to the authors' knowledge, the available solutions rely heavily on large microphone arrays [5], which may be infeasible for handheld portable devices.

In this work, we present a robust front-end speech enhancement and ASR solution for a single-microphone limited-vocabulary system during continuous monaural music playback. In contrast to previous studies, the microphone in our system is placed in close proximity to the loudspeakers, and the voice command still needs to be recognized at a very low speech-to-echo ratio (SER) while the music is playing and at a low speech-to-noise ratio (SNR) (due to background noise).

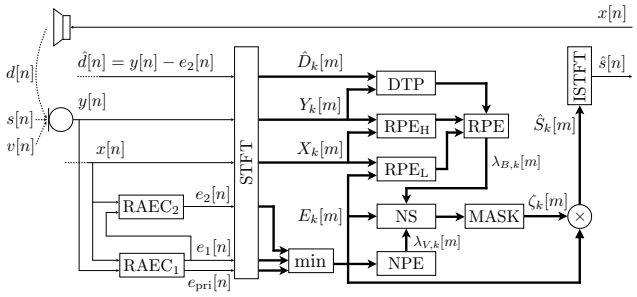

Fig. 1. A block diagram of the front-end enhancement system.

The front-end algorithm design effort can be divided into two parts. Firstly, we tailor double-talk robust solutions for echo cancellation and speech enhancement to retrieve a clean estimate of the command [7, 8, 9]. Secondly, we propose a novel noise reduction method, where we combine a traditional minimum mean-squared error (MMSE) speech enhancement approach [10] with an estimate of the ideal binary mask [11]. The parameters of the algorithm are tuned for maximum recognition rate by casting the tuning problem as a nonlinear program, solved efficiently through a genetic algorithm (GA) [12]. A similar approach was used in [13, 14] to maximize the objective perceptual quality of a speech enhancement system for full-duplex communication. The training and evaluation corpora are generated through a synthetic mixture of clean speech (from the TIMIT database [16]) and music, both convolved with separate impulse responses, and further mixed with a background noise to cover as many deployment scenarios as possible. Our proposed approach is then applied on a command recognizer for the hands free device using real recordings.

The paper is organized as follows. In Section 2, we describe the speech enhancement algorithm and outline the parameters to be tuned. In Section 3, we briefly describe the back-end recognizer. The tuning of these parameters by nonlinear optimization is presented in Section 4. The experimental results in Section 5 are divided in two parts. Firstly, we present the results of the training and evaluation of the front-end and acoustic models using the TIMIT database. Sec-

*Research conducted as an intern at Beats Electronics.

The authors thank Stephen Nimick for recording the voice commands used in the experimental evaluation.

ondly, we change the language model and implement our ASR system for a limited vocabulary command recognizer in very adverse conditions.

2. SPEECH ENHANCEMENT SYSTEM

Let $y[n]$ be the near-end microphone signal, which consists of the near-end speech $s[n]$ and noise $v[n]$, mixed with the acoustic echo $d[n] = h[n] * x[n]$ (music playback in our case), where $h[n]$ is the impulse response of the system, $x[n]$ is the far-end reference signal, and $*$ is the convolution operator. The overall block diagram of the speech enhancement algorithm is shown in Figure 1, which consists of two robust acoustic echo cancelers (RAECs), a double-talk probability (DTP) estimator, two residual power estimators (RPEs), a noise power estimator (NPE), and a noise suppressor (NS).

2.1. Robust Acoustic Echo Canceler

Since strong near-end interference may corrupt the error signal of the acoustic echo canceler (AEC) and cause the adaptive filter to diverge, the RAEC system [7, 9] is used, where the error recovery nonlinearity and robust adaptive step-size control allows for continuous tracking of the echo path during double talk. To reduce the delay of the frequency-domain adaptive filter [17], the multi-delay adaptive filter structure [18] is used. A cascaded structure similar to the system approach of [8] is used: the output of the first RAEC is fed to the input of the second RAEC, which is different from the original system approach in [8] where the input to the second RAEC is still the microphone signal (a parallel structure instead of the cascaded structure used in this work).

The tuning parameters for each of the RAECs consist of the frame size N_{AEC}, the number of partitioned blocks M_{AEC}, the number of iterations N_{iter}, the step-size μ_{AEC}, the tuning parameter γ_{AEC} for the robust adaptive step-size, and the smoothing factor α_{AEC} for the power spectral density estimation.

2.2. Residual Echo Power Estimator

Since the AEC cannot cancel all the echo signal due to modeling mismatch, further enhancement from the residual echo suppressor (RES) is required to improve the voice quality. A coherence based method similar to [21, 22] is used for the RPE, and a modified version of the DTP estimator similar to [23] is used for a more accurate estimate of the residual echo power. As shown in Figure 1, the DTP estimator differs from that in [23] since the coherence is calculated between the RAEC estimated echo signal $\hat{d}$ and the microphone signal y rather than between the loudspeaker signal x and the microphone signal y. This is possible since the estimated echo signal $\hat{d}$ can be reliably obtained even during double talk due to the *robust* echo path tracking performance of the RAEC.

In this work, we propose to estimate the residual echo power by utilizing the output of the double talk probability estimator. Ideally, when the double-talk probability is high, the level of residual echo power estimate should be low so as to not distort the near-end speech when suppressing the residual echo. On the other hand, when the double-talk probability is low, the level of residual echo power estimate should be high to suppress as much residual echo as possible. The high level residual echo power $\lambda_{B_{\text{H}},k}$ is estimated based on the coherence of the microphone signal Y_k and the reference signal X_k, while the low level residual echo power $\lambda_{B_{\text{L}},k}$ is estimated based on the coherence of the error signal E_k and the reference signal X_k. Finally, the residual echo power $\lambda_{B,k}$ is estimated by utilizing the double-talk probability estimate P_k^{DT} obtained from DTP to combine $\lambda_{B_{\text{H}},k}$ and $\lambda_{B_{\text{L}},k}$:

$$\lambda_{B,k}[m] = (1 - [m]P_k^{\text{DT}}[m])\lambda_{B_{\text{H}},k}[m] + P_k^{\text{DT}}[m]\lambda_{B_{\text{L}},k}[m], \quad (1)$$

where k is the frequency bin and m is the time frame.

The tuning parameters for the DTP consist of the transition probabilities a_{01}, a_{10}, b_{01}, and b_{10}, the smoothing factors α_{DTP} and β_{DTP}, the frequency bin range $[k_{\text{begin}}, k_{\text{end}}]$, the frame duration T_{DTP}, and the adaptation time constants τ. The tuning parameters for the RPE consist of the numbers of partitions $M_{\text{RPE}_{\text{H}}}$ and $M_{\text{RPE}_{\text{L}}}$ to calculate the coherence and the smoothing factors $\alpha_{\text{RPE}_{\text{H}}}$ and $\alpha_{\text{RPE}_{\text{L}}}$ for the power spectral density estimation.

2.3. Noise Suppressor

In this work, we combine the RPE and NPE for residual echo and noise suppression using a single noise suppressor, as shown in Figure 1. The low complexity MMSE noise power estimator [20] is used for the NPE, and the Ephraim and Malah log-spectral amplitude (LSA) estimator [10] is used for the combined residual echo and noise suppression:

$$G_k^{\text{LSA}}[m] = \frac{\xi_k[m]}{1+\xi_k[m]} \exp\left(\frac{1}{2}\int_{\frac{\xi_k[m]\gamma_k[m]}{1+\xi_k[m]}}^{\infty} \frac{e^{-t}}{t}\,\mathrm{d}t\right). \quad (2)$$

The estimation of the *a priori* SNR ξ_k is done using the decision-directed (DD) approach [19]:

$$\xi_k[m] = \alpha_{\text{DD}} \frac{|\hat{S}_k[m-1]|^2}{\lambda_{V,k}[m] + \lambda_{B,k}[m]} + (1-\alpha_{\text{DD}})\max\{\gamma_k[m] - 1, 0\},$$

where

$$\gamma_k[m] = \lambda_{E,k}[m]/(\lambda_{V,k}[m] + \lambda_{B,k}[m])$$

and $\lambda_{E,k}$, $\lambda_{V,k}$, and $\lambda_{B,k}$ are the residual error signal power, the noise power, and residual echo power respectively.

The tuning parameters of the NPE consist of the fixed *a priori* SNR ξ_{H_1}, the threshold P_{TH}, and the smoothing factors α_P and α_{NPE}. The tuning parameters of the the NS consist of the smoothing factor for the SNR estimator α_{DD}.

2.4. Generation of Speech Enhancement Mask

It has been recently shown that the speech recognition accuracy in noisy conditions can be greatly improved by direct binary masking [11] when compared to marginalization [24] or spectral reconstruction [25]. Given our application scenario, we propose to combine the direct masking approach, particularly effective at low overall SNRs, with the NS output mask G_k^{LSA}, as shown in Figure 1. In particular, we exploit the estimated bin-based *a priori* SNR ξ_k to determine the type of masking to be applied to the spectrum. However, given that an accurate estimation of the binary mask is very difficult for very low SNRs, we elect to use the LSA estimated gain for those cases (as described in [15]). Our masking then becomes:

$$\zeta_k[m] = \begin{cases} [(1-G_{\text{min}})G_k^{\text{LSA}}[m] + G_{\text{min}}], & \xi_k[m] \le \theta_1, \\ \frac{\alpha}{2}, & \theta_1 < \xi_k[m] < \theta_2, \\ \frac{2+\alpha}{2}, & \xi_k[m] \ge \theta_2, \end{cases}$$

where G_{min} is the minimum suppression gain [14], and the output is then:

$$\hat{S}_k[m] = \zeta_k[m]E_k[m]. \quad (3)$$

Tuning parameters for the direct masking consist of the minimum gain G_{min}, the thresholds θ_1 and θ_2, and tuning parameter α.

3. BACK-END SPEECH RECOGNIZER

The signal processed by the front-end of Figure 1 is then processed by the back-end recognizer which extracts features and proceeds with the sequence likelihood calculation based on the designed acoustic model distributions. As an alternative to commonly used hidden Markov models (HMMs), we chose two recently introduced statistical paradigms for modeling the distributions: one based on deep neural networks (DNN) [26], and the other is based on subspace Gaussian mixture models (SGMM) [27]. In both cases, a 40-dimensional feature vector is processed by the back-end recognizer consisting of perceptual linear prediction (PLP), linear discriminant analysis (LDA), maximum likelihood linear transform (MLLT), and feature-space maximum likelihood linear regression (fMLLR) [30, 27]).

3.1. Deep Neural Networks

DNNs evaluate the likelihood of a particular sequence using a feed-forward neural network that takes several frames of features as input and produces posterior probabilities over hidden Markov model (HMM) states as output. DNNs help efficiently model data that lie on or near a nonlinear manifold in the data space [28]. Thus, DNNs with many hidden layers have been shown to outperform GMMs on a variety of speech recognition benchmarks, sometimes by a large margin [28]. The DNN architecture consists of 3 hidden layers with 1024 neurons in each layer and 11 frame inputs (5 past frames and 5 future frames).

3.2. Subspace Gaussian Mixture Models

In contrast with standard GMM-HMM systems where state level observation densities consist of a dedicated mixture of multivariate Gaussian mixtures in subspace GMM share a common structure. In this formalism, the means and mixture weights are controlled by a global mapping from a vector space, through one or more state projection vectors, to the GMM parameter space (for more detail see [27]).

4. THE TUNING PROBLEM

The tuning problem can be formalized as an optimization problem. In our case, the objective function to maximize is the ASR recognition rate $\mathbf{R}\left(\hat{s}[n]\right)$, where $\hat{s}[n]$ is the processed speech, i.e., the output of the speech enhancement system. To restrict the search region, we can impose inequality constraints on the variables that simply determine lower and upper bounds for the components of the solution vector. Our optimization problem then becomes:

$$\begin{aligned} &\text{maximize} && \mathbf{R}\left(\hat{s}[n,\mathbf{p}]\right) \\ &\text{subject to} && \mathbf{L} \leq \mathbf{p} \leq \mathbf{U}, \end{aligned} \tag{4}$$

where $\mathbf{p}$ is now the vector of the parameters that need tuning, $\hat{s}[n,\mathbf{p}]$ is the speech enhancement system output obtained with these parameters, and $\mathbf{L}$ and $\mathbf{U}$ represent, respectively, lower and upper bounds for the values each variable. The basic concept of a GA is to apply genetic operators, such as *mutation* and *crossover*, to evolve a set of M solutions, or *population*, $\mathbf{\Pi}^{(k)} = \{\mathbf{p}_m^{(k)}, m = 1, \ldots, M\}$ in order to find the solution that maximizes the cost function [12, 29]. The steps are as follows.

1. An initial population of M solutions $\mathbf{\Pi}^{(k)}$, $k = 0$ is generated by randomly choosing in the space of feasible values $[\mathbf{L}, \mathbf{U}]$.
2. Compute the ASR accuracy $\mathbf{R}\left(\hat{s}[n, \mathbf{p}_m^{(k)}]\right)$ for each candidate solution of the population.
3. Combine sets of candidates with best accuracy through crossover and randomize the coefficients of the worst candidates through mutation to determine a better candidate.
4. Repeat steps 2 and 3 for K iterations (*generations*) or until a halting criterion is reached. The set of parameters $\mathbf{p}_m^{(K)} \in \mathbf{\Pi}^{(K)}$ that maximizes the cost function will be our estimate:

$$\hat{\mathbf{p}} = \underset{\mathbf{p}_m^{(K)} \in \mathbf{\Pi}^{(K)}}{\arg\max}\ \mathbf{R}\left(\hat{s}[n, \mathbf{p}_m^{(K)}]\right). \tag{5}$$

5. EXPERIMENTAL RESULTS

In this section, we present the results from our designed speech enhancement front-end with the tuned parameters using the optimization method presented in Section 3. In order to obtain the set of parameters that maximize the recognition rate, we optimized and tuned the system on a noisy TIMIT database and on our real-world command recordings.

5.1. Processing on TIMIT Database

5.1.1. Noisy TIMIT Database Generation

The database was generated by simulating the interaction between the user and the playback device. In this scenario, music is played from a loudspeaker system in which a microphone is placed one centimeter away from the loudspeaker. The microphone signal $y[n]$ was then generated according to:

$$y[n] = s[n] + \sigma_1 d[n] + \sigma_2 v_2[n],$$

which consisted of the speech $s[n]$, the acoustic echo from the music $d[n]$ and the background noise $v_2[n]$ (babble noise). For each file in the TIMIT database, the SER and SNR were chosen from uniform distributions ranging from -30 dB to 10 dB and from 0 dB to 30 dB, respectively. We used 12 impulse responses recorded on the device in real rooms randomly picked and normalized to unitary energy. The values of σ_1 and σ_2 were calculated based on SER and SNR. The music sound, $d[n]$, was randomly selected from five different music tracks of different genres with random starting points.

5.1.2. Recognition on noisy TIMIT

In order to optimize the parameters of our front-end speech enhancement system, we followed an iterative approach outlined below.

- **Iteration 0 (initialization):**
  1. Train the acoustic models (DNN and SGMM) on clean TIMIT.
  2. Tune the font-end parameters with our proposed tuning algorithm.
- **Iteration 1 and higher:**
  1. Process simulated noisy TIMIT with our tuned front-end and generate processed TIMIT utterances.

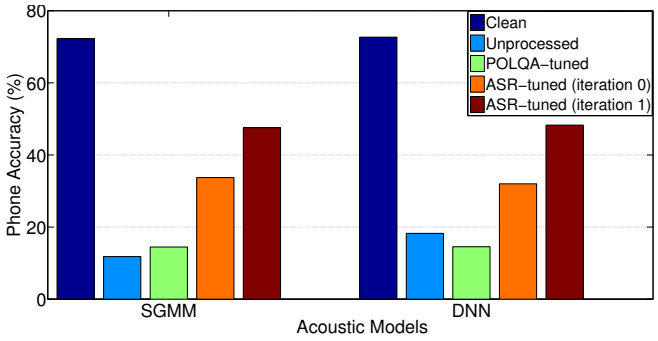

Fig. 2. Phone accuracy (in %) for different noise conditions and different tuning parameters for both DNN and SGMM on noisy TIMIT. SNR is between 0 dB to 30 dB and SER is between -30 dB to 10 dB.

2. Train the acoustic models with a mixture of clean TIMIT and processed TIMIT utterances.
3. Use the adapted acoustic models to re-tune the front-end with our proposed tuning algorithm using the database of clean and processed utterances.

In other words, at iteration 0, the tuning algorithm determines a set of parameter that output the "cleanest" speech, in order to match the clean TIMIT database characteristics. Due to the mismatch between the enhanced speech and the clean acoustic model, further iterations help reduce this mismatch and improve the ASR performance. Regardless of the iteration number, the GA had a population of $M = 40$ possible candidates and $K = 3$ were enough to reach convergence. These values were chosen empirically by balancing the complexity and the accuracy of the results.

Figure 2 shows the phone accuracy of the clean TIMIT database, the unprocessed noisy database, the speech enhancement front-end tuned with the the Perceptual Objective Listening Quality Assessment (POLQA, [31]) subjective measure [13], and the the speech enhancement front-end tuned with ASR. The acoustic models were trained on the clean TIMIT database except for the ASR optimized system tuned with 1 iteration. The system tuned with ASR outperforms the system tuned with POLQA. The phone accuracy on the clean TIMIT database with a triphone language model were 74.30% for DNN and 75.26% for SGMM, comparable to the performance reported in [26] and [27], respectively. Figure 2 also shows the phone accuracy when an iteration in the presented approach is used.

Although used in a different setup, the results obtained with the proposed method compared favorably to some prior results [32, 33], where authors investigated joint echo cancellation and speech enhancement at higher SERs and SNRs.

5.2. Processing on Real Commands

We used the system to recognize four commands: play, next, back, pause, as well as a garbage model. In this section two different scenarios are considered. We first use the set of tuned parameters for the speech enhancement system from our analysis on the TIMIT database to study the feasibility of speech recognition on limited vocabulary in extremely challenging conditions and assess the generalization of our tuning approach to unseen data (system trained on TIMIT but tested on commands). We then conducted another set of experiments where the tuning of the parameters was done on real recordings of actual commands.

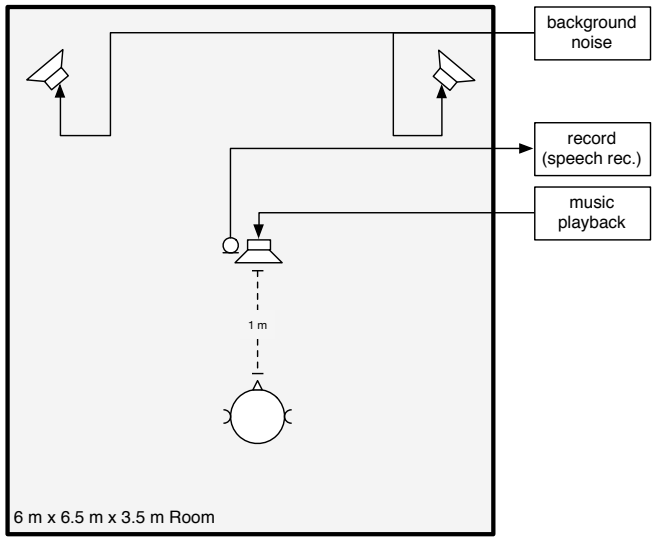

Fig. 3. Experimental setup for voice recording.

5.2.1. *Recording of Commands Database*

Figure 3 shows the setup for voice recording, where eight subjects (male/female, native/non-native English speakers) uttered a list of commands at a distance of 1 meter from the microphone of the *Beats Pill*™ portable speaker while music was playing. We used eight different music tracks, where the starting point of the track was chosen randomly. Subjects uttered the following commands towards the speakers: play, next, back, pause. Ambient noise, recorded in a shopping mall containing a mixture of babble and other environmental noise, was played through separate loudspeakers facing the wall of the room to simulate diffuse noise. The music playback levels were set to three different levels: off, medium, and loud. We estimated that the range of SER for the medium and loud are -35 to -30 dB and -40 to -35 dB, respectively. The SNR of the ambient noise was set to 5 to 10 dB for the medium and loud scenarios. There is neither background noise nor music playback for the clean recordings. The estimation of the SERs was made possible thanks to a close-talking microphone that recorded the near-end speech. Furthermore, we measured the average C-weighted SPL of the music playback at the microphone to be 92 dBC and 102 dBC for the medium and loud cases respectively.

5.2.2. *Recognition on Noisy Commands*

Figure 4 shows the command recognition rates over the four commands for different noise and echo levels, different acoustic models, and different tuning parameters. The acoustic models in both DNN and SGMM were both trained using the clean TIMIT database. The parameters for the POLQA-tuned and ASR-tuned (TIMIT) cases were the system tuned when the target optimization function was an objective speech quality metric (POLQA) [15] and the system tuned when the target was to maximize phone accuracy on noisy TIMIT (as described in Section 5.1.2), respectively. For the clean commands the accuracy for both DNN and SGMM was 100%, as expected for this small command list. The command recognition rate of the close talk microphone degrades slightly when there was music and background noise but was still around or above 90% in all cases. For the *Beats Pill*™ microphone recording during music playback and background noise, we obtained the best accuracy when the tuning

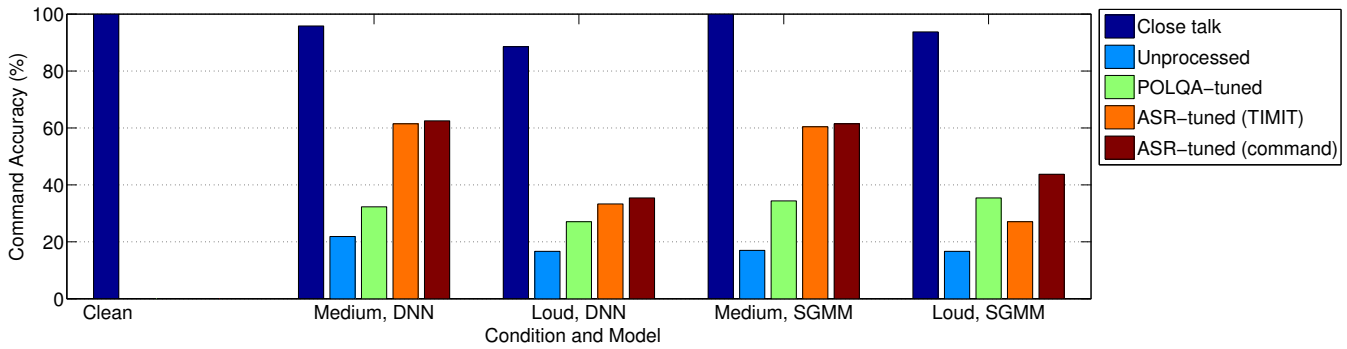

Fig. 4. Command Accuracy in % averaged over the four commands for different noise conditions and for different acoustic models. Clean: SNR=∞, SER=∞. Medium: SER=$[-35, -30]$ dB, SNR=$[0, 5]$ dB. Loud: SER=$[-40, -35]$ dB, SNR=$[0, 5]$ dB. First bar represents accuracy on clean recordings for both DNN and SGMM acoustic models. Second and third groups of bars represent results for DNN, while the fourth and fifth groups represent results for SGMM. ASR-tuned (TIMIT): GA tuning over the TIMIT database. ASR-tuned (command): GA tuning over the commands.

was done on a mixture of both medium and loud conditions as in ASR-tuned (command).

Since the recording condition was not always known in advance, the command recognition on the mixed condition was also a good indication of the generalization capacity of our proposed algorithm. Furthermore, command accuracy from the optimization performed on TIMIT was within 2% absolute of the results obtained while optimizing on commands for the medium level scenario, which was a good indication of the generalization capacity of our proposed approach. The accuracy gap was wider between TIMIT-based optimization and the loud-level commands due to a mismatch between SER and SNR of the simulated TIMIT and loud commands. Our results also clearly showed that our proposed tuning based on ASR optimization outperforms the POLQA-based tuning. The difference in performance seemed to derive from the POLQA optimization being less aggressive on noise in order to preserve speech quality.

6. CONCLUSION

We proposed a robust ASR front-end and a related tuning methodology combined with a state-of-the-art speech recognition systems (DNN- and SGMM-based). The proposed speech enhancement front-end consists of a cascaded robust AEC, a residual echo power estimator based on a double-talk probability estimator, and a *quasi*-binary masking that utilizes the classical MMSE-based method at very low SNRs. The tuning improved the speech recognition rate substantially on the TIMIT database. The optimized front-end was then tested in realistic environments for the remote control of a music playback device with a limited-sized command dictionary. The result showed a fairly high recognition rate for voice commands at a speech-to-music ratio as low as -40 dB and SNR as low as 0 dB, scenarios hardly seen through the literature. In our experiments, SGMM outperformed DNN in noisy conditions. However, training the DNN on a larger corpus can potentially improve recognition results on DNN. In fact, training the DNN on a larger dataset including noisy and clean TIMIT improved the overall recognition rate of the DNN when our proposed iterative approach was used. We also showed that training the back-end ASR and tuning our front-end speech enhancement system in the iterative approach improved the overall recognition results.

7. REFERENCES

[1] M. G. Helander, T. K. Landauer, and P. V. Prabhu, *Handbook of human-computer interaction*, Elsevier, 1997.

[2] J. Li, L. Deng, Y. Gong, and R. Haeb-Umbach, "An overview of noise-robust automatic speech recognition," *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 22, no. 4, pp. 745–777, 2014.

[3] M. Wölfel and J. McDonough, *Distant speech recognition*, John Wiley & Sons, 2009.

[4] L. Marquardt, P. Svaizer, E. Mabande, A. Brutti, C. Zieger, M. Omologo, and W. Kellermann, "A natural acoustic front-end for interactive TV in the EU-project DICIT," in *Proc. IEEE Pacific Rim Conf. on Communications, Computers and Signal Processing*, pp. 894–899, 2009.

[5] M. L. Seltzer, "Microphone array processing for robust speech recognition," Ph.D. dissertation, Carnegie Mellon University, 2003.

[6] J. Wung, D. Giacobello, and J. Atkins, "Robust Acoustic Echo Cancellation in the Short-Time Fourier Transform Domain Using Adaptive Crossband Filters," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 1314–1318, 2014.

[7] T. S. Wada and B.-H. Juang, "Acoustic echo cancellation based on independent component analysis and integrated residual echo enhancement," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 205–208, 2009.

[8] J. Wung, T. S. Wada, B.-H. Juang, B. Lee, M. Trott, and R. W. Schafer, "A system approach to acoustic echo cancellation in robust hands-free teleconferencing," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 101–104, 2011.

[9] T. S. Wada and B.-H. Juang, "Enhancement of residual echo for robust acoustic echo cancellation," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 175–189, 2012.

[10] Y. Ephraim and D. Malah, "Speech enhancement using a minimum mean-square error log-spectral amplitude estima-

tor," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 33, no. 2, pp. 443–445, 1985.

[11] W. Hartmann, A. Narayanan, E. Fosler-Lussier, and D. Wang, "A direct masking approach to robust ASR," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 21, pp. 1993–2005, 2013.

[12] D. E. Goldberg, *Genetic algorithms in search, optimization, and machine learning*, Addison-Wesley, 1989.

[13] D. Giacobello, J. Wung, R. Pichevar, and J. Atkins, "Tuning methodology for speech enhancement algorithms using a simulated conversational database and perceptual objective measures," in *Proc. 4th Joint Workshop on Hands-free Speech Communication and Microphone Arrays*, 2014.

[14] D. Giacobello, J. Atkins, J. Wung, and R. Prabhu, "Results on automated tuning of a voice quality enhancement system using objective quality measures," in *Proc. 135th Audio Engineering Society Convention*, 2013.

[15] D. Giacobello, J. Wung, R. Pichevar, and J. Atkins, "A computationally constrained optimization framework for implementation and tuning of speech enhancement systems," in *Proc. International Workshop on Acoustic Signal Enhancement*, 2014.

[16] J. S. Garofolo, *et al.*, *TIMIT: acoustic-phonetic continuous speech corpus*, Linguistic Data Consortium, 1993.

[17] J. J. Shynk, "Frequency-domain and multirate adaptive filtering," *IEEE Signal Processing Magazine*, vol. 9, no. 1, pp. 14–37, 1992.

[18] J. S. Soo and K. K. Pang, "Multidelay block frequency domain adaptive filter," *IEEE Trans. on Acoustics, Speech and Signal Processing,* , vol. 38, no. 2, pp. 373–376, 1990.

[19] Y. Ephraim and D. Malah, "Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 32, no. 6, pp. 1109–1121, 1984.

[20] T. Gerkmann and R. C. Hendriks, "Unbiased MMSE-based noise power estimation with low complexity and low tracking delay," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 20, no. 4, pp. 1383–1393, 2012.

[21] G. Enzner, R. Martin, and P. Vary, "Unbiased residual echo power estimation for hands-free telephony," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 1893–1896, 2002.

[22] S. Goetze, M. Kallinger, and K. Kammeyer, "Residual echo power spectral density estimation based on an optimal smoothed misalignment for acoustic echo cancellation," in *Proc. International Workshop on Acoustic Echo and Noise Control*, pp. 209–212, 2005.

[23] I. J. Tashev, "Coherence based double talk detector with soft decision," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 165–168, 2012.

[24] M. Cooke, P. Green, L. Josifovski, and A. Vizinho, "Robust automatic speech recognition with missing and unreliable acoustic data," *Speech Communication*, vol. 34, pp. 267–285, 2001.

[25] B. Raj, M. Seltzer, and R. M. Stern, "Reconstruction of missing features for robust speech recognition," *Speech Communication*, vol. 43, pp. 275–296, 2004.

[26] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large vocabulary speech recognition," *IEEE Trans. on Audio, Speech, and Language Proc.*, vol. 20, no. 1, pp. 30–42.

[27] D. Povey et al., "Subspace Gaussian mixture models for speech recognition," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2010.

[28] Hinton et al., "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups," *IEEE Signal Processing Magazine*, vol. 29, pp. 82–97, 2012.

[29] R. Duda, P. Hart, and D. Stork, "Pattern Classification", *John Wiley and Sons*, 2012.

[30] D. Povey et al., "The Kaldi Speech Recognition Toolkit," *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, pp. 4330–4333, 2011.

[31] *Perceptual Objective Listening Quality Assessment*, ITU-T Rec. P.863, 2010.

[32] W. Herbordt, S. Nakamura, and W. Kellerman, "Joint optimization of LCMV beamforming and acoustic echo cancellation for automatic speech recognition," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pp. 77–80, 2005.

[33] G. Reuven, S. Gannot, and I. Cohen, "Joint noise reduction and acoustic cancellation using the transfer-function generalized sidelobe canceller," *Speech communication*, vol. 49, pp. 623–635, 2007.