# CAN BACK-ENDS BE MORE ROBUST THAN FRONT-ENDS ? INVESTIGATION OVER THE AURORA2 DATABASE

*Alexis Bernard, Yifan Gong*

Speech Technologies Laboratory
Texas Instruments Inc., Dallas, TX
{bernard,yifan.gong}@ti.com

*Xiaodong Cui**

Department of Electrical Engineering
University of California, Los Angeles, CA
xdcui@icsl.ucla.edu

## ABSTRACT

We present a back-end solution developed at Texas Instruments for noise robust speech recognition. The solution consists of three techniques: 1) a joint additive and convolutive noise compensation (JAC) which adapts speech acoustic models, 2) an enhanced channel estimation procedure which extends JAC performance towards lower SNR ranges, and 3) an N-pass decoding algorithm. The performance the proposed back-end is evaluated on the Aurora-2 database. With 20% less model parameters and without the need for second order derivative of the recognition features, the performance of the proposed solution is 91.86%, which outperforms that of the ETSI Advanced Front-End standard (88.19%) by more than 30% relative word error rate reduction.

## 1. INTRODUCTION

A speech recognizer trained on clean speech data and operating in different environments has lower performance due to at least two distortion sources [1]: background noise and microphone changes. Handling simultaneously the two is critical to the performance of the recognizer.

Many *front-end* solutions, e.g. [2, 3], have been developed and have shown give promising results for connected digit recognition applications in very noisy environments. For instance, methods such as the ETSI advanced DSR front-end [3] handles both channel distortion and background noise. These techniques does not require any noisy training data. To be effective in noise reduction, they typically require an accurate instantaneous estimate of the noise spectrum.

Alternative solutions consist, instead, of modifying the *back-end* of the recognizer to compensate for the mismatch between the training and recognition environments. More specifically, in the acoustic model space, a convolutive (e.g. channel) component and an additive (e.g. background noise) component can be introduced to model the two distortion sources [4, 5, 6, 7, 8, 9]. The effect of the two distortions introduces in the log spectral domain non-linear parameter changes, which can be approximated by linear equations [10, 11].

In this paper, we present an extension of a framework recently developed at TEXAS INSTRUMENTS, JAC (**J**oint compensation of **A**dditive and **C**onvolutive distortions), that handles simultaneously both background noise and channel distortions for speaker-independent speech recognition [9]. Perform of the novel back-end solutions over the Aurora2 database is analyzed.

*Part of the work was performed while the author was a summer intern at TEXAS INSTRUMENTS.

The JAC algorithm was developed and optimized for mobile device speech recognition applications, for which typical SNR ranges are higher than the lowest SNRs in the artificially created Aurora2 database. For this evaluation, novel enhancements to the JAC technique have been developed, including making the compensation system robust to large channel mismatches and low SNR signals, as well performing N-pass recognition. Such enhancements contributed significantly to extending performance robustness over wider range of noise and channel conditions imposed by the Aurora2 framework. The new scheme is referred to as Enhanced-JAC or E-JAC.

## 2. BACK-END MODEL COMPENSATION

### 2.1. Joint additive and convolutive noise compensation

A speech signal $x(n)$ can only be observed in a given acoustic environment. An acoustic environment can be modelled by a background noise $b'(n)$ and a distortion channel $h(n)$. For typical mobile speech recognition, $b'(n)$ consists, for instance, of office noise, vehicle engine or road noise, and $h(n)$ consists of the microphone type or its relative position to the speaker. Let $y(n)$ be the speech observed in the environment involving $b'(n)$ and $h(n)$: $y(n) = (x(n) + b'(n)) * h(n)$. In typical speech recognition applications, $b'(n)$ cannot be measured directly. What is available is $b'(n) * h(n)$. Let $b(n) = b'(n) * h(n)$, our model of distorted speech becomes:

$$y(n) = x(n) * h(n) + b(n) \tag{1}$$

Or, in the power spectral domain,

$$\mathbf{Y}(k) = \mathbf{X}(k)\mathbf{H}(k) + \mathbf{B}(k) \tag{2}$$

Representing the above quantities in logarithmic scale, we have:

$$\mathbf{Y}^l(k) = \mathbf{g}(\mathbf{X}^l, \mathbf{H}^l, \mathbf{B}^l)(k) \tag{3}$$

$$\triangleq \log(\exp(\mathbf{X}^l(k) + \mathbf{H}^l(k)) + \exp(\mathbf{B}^l(k))) \tag{4}$$

Assuming the log-normal distribution [12] and ignoring the variance, we have, in the acoustic model space,

$$\mathrm{E}[\mathbf{Y}^l] \triangleq \hat{\mathbf{m}}^l = \mathbf{g}(\mathbf{m}^l, \mathbf{H}^l, \mathbf{B}^l) \tag{5}$$

where $\mathbf{m}^l$ is the original Gaussian mean vector and $\hat{\mathbf{m}}^l$ is the Gaussian mean vector compensated for the distortions caused by channel $\mathbf{H}^l$ and environment noise $\mathbf{B}^l$.

### 2.2. Estimation of channel and noise components

Our goal is to derive the Hidden Markov Models (HMMs) of $\mathbf{Y}$, the speech signal under both additive noise and convolutive distortions. The key problem is to obtain an estimate of the channel $\mathbf{H}^l$ and noise $\mathbf{B}^l$. We assume that some speech data recorded in the noisy environment is available, and that the starting HMM models for $\mathbf{X}$ are trained on clean speech in the feature space.

Applying the Expectation-Maximization (EM) procedure [13], it can be shown [9, 14] that $\mathbf{H}^l$ and $\mathbf{B}^l$ are given by the solution to the equation

$$\mathbf{u}(\mathbf{H}^l, \mathbf{B}^l) = \sum_{j \in \Omega_s} \sum_{k \in \Omega_m} \sum_{r=1}^{R} \sum_{t=1}^{T^r} \gamma_t^r(j,k) \cdot \left\{ \mathbf{g}(\mathbf{m}_{j,k}^l, \mathbf{H}^l, \mathbf{B}^l) - \mathcal{DFT}(\mathbf{o}_t^r) \right\} = 0, \quad (6)$$

where $\gamma_t^r(j,k)$ is the probability of being in state $j$ with mixing component $k$ at time $t$ given utterance $r$, and $\mathbf{o}_t^r$ is the observation feature vector at time $t$ for utterance $r$.

#### 2.2.1. Estimation of noise component

Eq. 6 can be used to solve both $\mathbf{H}^l$ and $\mathbf{B}^l$. However, in this paper, we assume $\mathbf{B}^l$ to be stationary, and use the first $P$ non-speech frames as an estimate of $\mathbf{B}^l$. We calculate an estimate of noise in the log domain $\hat{\mathbf{B}}^l$ as the average of the $P$ noise frames in the log domain

$$\hat{\mathbf{B}}^l = \frac{1}{P} \sum_{t=1}^{P} \mathcal{DFT}(\mathbf{y}_t). \quad (7)$$

#### 2.2.2. Solving channel equation

To solve $\mathbf{H}^l$ for $\mathbf{u}(\mathbf{H}^l, \mathbf{B}^l = \hat{\mathbf{B}}^l) = 0$, we use Newton's method, which has interesting convergence property for on-line estimation of the parameters. The method is iterative, which gives a new estimate $\mathbf{H}^l_{[i+1]}$, at iteration $i+1$, of $\mathbf{H}^l$ using

$$\mathbf{H}^l_{[i+1]} = \mathbf{H}^l_{[i]} - \frac{\mathbf{u}(\mathbf{H}^l_{[i]}, \hat{\mathbf{B}}^l)}{\mathbf{u}'(\mathbf{H}^l_{[i]}, \hat{\mathbf{B}}^l)}, \quad (8)$$

where $\mathbf{u}'(\mathbf{H}^l, \hat{\mathbf{B}}^l)$ is the derivative of $\mathbf{u}(\mathbf{H}^l, \hat{\mathbf{B}}^l)$ with respect to channel $\mathbf{H}^l$. As initial condition for Eq. 8, we can set $\mathbf{H}^l_{[0]} = \mathbf{0}$

#### 2.2.3. Compensation for time derivatives

The distortion caused by channel and noise also affects the distribution of dynamic (e.g. time derivative of) MFCC coefficients. According to definition, the compensated time derivative of cepstral coefficients $\dot{\mathbf{Y}}^c$ is the time derivative of compensated cepstral coefficients $\mathbf{Y}^c$ [7]. It can be shown [7, 14] that both first and second order time derivatives are respectively a function of

$$\eta(k) = \exp(\mathbf{H}^l(k))\psi(k), \quad (9)$$

where $\psi(k) = \dfrac{\exp(\mathbf{X}^l(k))}{\exp(\mathbf{B}^l(k))}$ is the SNR in the linear scale at the frequency bin $k$.

### 2.3. Enhancements to JAC (E-JAC)

#### 2.3.1. Introduction

While jointly estimating and compensating for additive (acoustic) and convolutive (channel) noise allows for a better recognition performance, special attention must be paid for low SNR speech signals. When the SNR is too low or the noise is highly non-stationary, it becomes difficult to make a correct noise estimate $\hat{\mathbf{B}}^l$. In that case, the channel estimate $\mathbf{H}^l_{[i+1]}$ will not only reflect channel mismatch but will also represent residual additive noise, erratic in nature. A simple solution to this problem consists of adding inertia to JAC channel estimate and to force the amplitude of channel estimates to be within a certain range, as explained hereafter.

#### 2.3.2. Inertia added to JAC channel estimation

At the beginning of a recognition task in a particular noise and channel condition, the recognizer may be suddenly exposed to a new type of background noise and microphone. It may be hard for the JAC algorithm to immediately give a good estimate of the channel after one utterance, since not enough statistics have been collected to represent the channel. A solution consists of separating the running channel estimate $\mathbf{H}^l_{[i+1]}$ from the channel estimate $\bar{\mathbf{H}}^l_{[i+1]}$ used for model compensation (Eq. 5). $\bar{\mathbf{H}}^l_{[i+1]}$ approaches $\mathbf{H}^l_{[i+1]}$ gradually, as more channel statistics are collected with the increasing number $q$ of observed utterances. After an SNR-dependent $Q$ utterances have been recognized, we have $\bar{\mathbf{H}}^l_{[i+1]} = \mathbf{H}^l_{[i+1]}$. When $q < Q$, $\bar{\mathbf{H}}^l_{[i+1]}$ is given by

$$\bar{\mathbf{H}}^l_{[i+1]} = \bar{\mathbf{H}}^l_{[i]} + \frac{q}{Q}(\mathbf{H}^l_{[i+1]} - \bar{\mathbf{H}}^l_{[i]}). \quad (10)$$

#### 2.3.3. Limits on JAC channel estimation

Despite the additional robustness provided by the new iterative procedure of Eq. 10, the channel estimate can still be inaccurate, especially with sudden exposure to new noise and channel conditions at low SNR. In this case, it may be beneficial to limit the amplitudes of the channel estimate that can be applied by JAC. This is done by forcing the amplitudes of the channel estimates to be within a certain range, as specified by

$$\mathbf{H}^l_{[i+1]} = \max(\min(\mathbf{H}^l_{[i+1]}, \text{JAC\_LIM}), -\text{JAC\_LIM}) \quad (11)$$

which guarantees that $\mathbf{H}^l_{[i+1]} \in [-\text{JAC\_LIM}, \text{JAC\_LIM}]$. Note that if $\text{JAC\_LIM} = 0$, we have $\mathbf{H}^l_{[0]} = \mathbf{0}$, and only background noise compensation can be applied.

### 2.4. N-pass decoding

The JAC algorithm finds the channel estimates that maximize the likelihood of observing the sequence of feature vectors given a segmentation provided by the Viterbi decoding. Once we obtain a better estimate of the channel, we may want to analyze how the new parameters may affect the sentence segmentation. In order to do this, we simply perform another iteration of the recognition back-end algorithm. This operation can be repeated for as many passes as desired, hence the name N-pass decoding.

## 3. EVALUATION OVER THE AURORA2 DATABASE

### 3.1. Experimental conditions

We evaluate the performance of the proposed Enhanced-JAC back-end algorithm (E-JAC) on the Aurora-2 database for the purpose of benchmarking it with the new ETSI Advanced Front-End (AFE) standard [3].

We used the standard Aurora-2 testing procedure, which averages out recognition performance over 10 different noise conditions (two with channel mismatch in Test C) at 5 different SNR levels (20dB, 15dB, 10dB, 5dB and 0dB). Since the clean data and the data at -5dB are not used in the average performance evaluation, we have not tested our algorithm at those noise levels.

As a reminder, performance of AFE standard on the Aurora-2 database is established using the following configuration: a 39-dimensional feature vector (13 AFE features with $1^{st}$ and $2^{nd}$ order derivative) extracted every 10 ms and 16 states word HMM models with 20 Gaussian mixtures per state. According to the official baseline for Eurospeech 2003, the average performance of the AFE standard over the entire database is 88.19%, which breaks down in the following percentages for each SNR condition, from 20dB to 0dB: 98.92, 97.78, 94.61, 85.99 and 63.66.

In the evaluation of our solution, we move slightly away from the feature vector being used, while keeping the HMM model topology the same. We use a 32-dimensional feature vector, which corresponds to a 16 dimensional MFCC vector and its $1^{st}$ order derivative only. For memory/benefit ratio concerns, TI's low footprint solution typically does not use second order derivative. While better results could obtained by using the second order derivative, it was decided not to use the acceleration features. Note that this means that our system operates on fewer features (about 20% less) than the AFE standard. In the experiments, a speech model variance adjustment is also applied.

### 3.2. Performance of E-JAC back-end

Table-2(a) summarizes the performance of our back-end solution on the Aurora-2 database using E-JAC with 2 passes. It can be seen that we obtain an average performance level of 91.86%, which corresponds to a 31% relative improvement over AFE (88.19%). We believe that such performance level obtained using clean training is in line with some of the best front-end processing solutions using multi-conditional training.

### 3.3. Comparison between one pass and two pass decoding

The improvement in performance provided by the N-pass decoding method can be measured by analyzing the average recognition increase given the number of passes.

Table 1 summarizes the average (over 10 noise conditions) recognition performance on Aurora-2 database with E-JAC (enhanced JAC) using MFCCs, for each SNR level with respect to the number of passes. It can be seen that increasing the number of passes improves recognition at each SNR level, with more pronounced gains for low SNR conditions. On average, two passe recognition provides a 5% relative error rate reduction. We do not expect the recognition improvements to be significant beyond two passes.

| SNR | 20 dB | 15 dB | 10 dB | 5 dB | 0 dB | MEAN |
|---|---|---|---|---|---|---|
| **1 pass** | 98.95 | 98.16 | 95.98 | 90.44 | 73.82 | **91.47** |
| **2 pass** | 99.00 | 98.29 | 96.29 | 90.99 | 74.42 | **91.86** |

**Table 1**. Word accuracy improvement of recognition performance over Aurora2 database for each SNR as a function of the number of passes.

### 3.4. Improvements of E-JAC over AFE

For a better reading of the results, Table-2(b) shows relative percentage improvements of E-JAC over the AFE standard. Several trends can be observed. First, on average, the E-JAC solution provides improvements over all noise conditions and SNR level. Second, improvements are more pronounced for low SNR signals. The smaller improvements for cleaner signal are explained by the fact that we are not using the 2nd order derivatives of the features. Third, improvements are larger (37%) for the test set C, with channel mismatch, which suggests that JAC has been able to accurately estimate the channel. In fact, absolute results indicate that there is no more degradation in performance between test C over test A and B, indicating that the channel mismatch in Aurora-2 is no longer an issue. Fourth, note that the four noise types for which our improvements over AFE are less pronounced (less than 20%) are Station, Babble, Restaurant and Airport, which are the three least stationary noises.

### 3.5. Combination of AFE and E-JAC

Given the performance and quality of both the Advanced Front-End and the E-JAC solutions, we decided to establish the performance level with a combined AFE front-end and E-JAC speech recognizer. We use AFE front-end and E-JAC back-end with 32 dimensional feature vector (16 static and 16 first order dynamic coefficients). We observe that the average performance of the combined system over all 50 noise conditions is 90.25%. This results indicates that while the addition of the E-JAC back-end solution could improve AFE performance by 17% relative, its performance was not in this case nearly as good as that of a back-end solution only approach. The reason why adding AFE does not help can be attributed to several factors, including: 1) the adaptive channel equalization in AFE may make the residual channel parameter time-varying, which makes it more difficult for the E-JAC algorithm to produce accurate channel estimation, and 2) the non-linearity of speech signal introduced by the spectral subtraction as used in AFE, especially in low SNR conditions, may affect adversely the speech quality.

## 4. CONCLUSIONS

The proposed E-JAC back-end compensation technique identifies two log-domain components from incoming speech signal: one for the channel or microphone distortion (convolutive), and the other for the background noise (additive). Superior performance can be obtained since a more accurate channel estimation is achieved by embedding HMM in the channel estimation process.

Experimental results show that E-JAC, although simple, is efficient in improving speaker-independent recognition performance on Aurora-2 database application task. With 20% less model parameter size, the method gives about 30% relative word error rate reduction over AFE, to yield an average performance of 91.86%.

(a)

| Aurora 2: Clean training, multicondition testing – MFCC-OD, E-JAC | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Test Set A | | | | | Test Set B | | | | | Test Set C | | | |
| | Subway | Babble | Car | Exhibit | Average | Restaurant | Street | Airport | Station | Average | SubwayM | StreetM | Average | **Average** |
| 20 dB | 99.11 | 98.76 | 99.08 | 99.07 | 99.01% | 99.03 | 98.73 | 99.25 | 99.29 | 99.08% | 99.08 | 98.61 | 98.85% | **99.00%** |
| 15 dB | 98.74 | 98.19 | 98.66 | 98.40 | 98.50% | 97.76 | 97.94 | 97.98 | 98.70 | 98.10% | 98.53 | 98.07 | 98.30% | **98.30%** |
| 10 dB | 97.42 | 95.47 | 97.44 | 96.79 | 96.78% | 94.88 | 95.62 | 95.59 | 96.79 | 95.72% | 97.36 | 95.65 | 96.51% | **96.30%** |
| 5 dB | 93.37 | 88.57 | 94.39 | 90.81 | 91.79% | 88.12 | 90.08 | 90.25 | 91.27 | 89.93% | 93.21 | 90.66 | 91.94% | **91.07%** |
| 0 dB | 81.30 | 65.90 | 80.05 | 76.98 | 76.06% | 68.50 | 73.25 | 75.66 | 74.45 | 72.97% | 78.45 | 71.80 | 75.13% | **74.63%** |
| Average | 93.99 | 89.38 | 93.92 | 92.41 | **92.43%** | 89.66 | 91.12 | 91.75 | 92.10 | **91.16%** | 93.33 | 90.96 | **92.14%** | **91.86%** |

(b)

| Relative word error rate reduction (in %) of E-JAC with respect to ETSI's AFE standard | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Test Set A | | | | | Test Set B | | | | | Test Set C | | | |
| | Subway | Babble | Car | Exhibit | Average | Restaurant | Street | Airport | Station | Average | SubwayM | StreetM | Average | **Average** |
| 20 dB | 25.83 | -10.71 | -6.98 | 18.42 | 6.64% | -8.99 | 6.62 | 0.00 | -9.23 | -2.90% | 33.33 | 6.08 | 19.71% | **5.44%** |
| 15 dB | 46.61 | 24.27 | 11.84 | 26.94 | 27.42% | 17.04 | 19.84 | -18.82 | 22.16 | 10.05% | 42.35 | 25.77 | 34.06% | **21.80%** |
| 10 dB | 57.14 | 24.37 | 26.65 | 39.55 | 36.93% | 10.33 | 25.26 | -12.79 | 26.71 | 12.38% | 58.88 | 36.03 | 47.45% | **29.21%** |
| 5 dB | 54.37 | 29.31 | 41.80 | 33.21 | 39.68% | 30.16 | 25.08 | 20.28 | 25.32 | 25.21% | 56.70 | 42.06 | 49.38% | **35.83%** |
| 0 dB | 45.56 | 21.65 | 30.78 | 33.20 | 32.80% | 24.41 | 22.26 | 27.32 | 19.60 | 23.40% | 46.37 | 30.76 | 38.56% | **30.19%** |
| Average | 45.90 | 17.78 | 20.82 | 30.26 | **28.69%** | 14.59 | 19.81 | 3.20 | 16.91 | **13.63%** | 47.53 | 28.14 | **37.83%** | **24.49%** |

**Table 2**. (a) Performance of E-JAC back-end on the Aurora-2 database (clean training). (b) Relative percentage improvements of E-JAC over AFE DSR standard.

## 5. REFERENCES


[1] Y. Gong, "Speech recognition in noisy environments: A survey," *Speech Communication*, vol. 16, no. 3, pp. 261–291, April 1995.

[2] M. Lieb and A. Fischer, "Experiments with the Philips continuous ASR system on the AURORA noisy digits database," in *Proceedings of European Conference on Speech Communication and Technology*, 2001.

[3] D. Macho, L. Mauuary, B. Noe, Y. M. Cheng, D. Ealey, D. Jouvet, H. Kelleher, D. Pearce, and F. Saadoun, "Evaluation of a noise-robust DSR front-end on AURORA databases," in *Proc. Int. Conf. on Spoken Language Processing*, Colorado, USA, September 2002, pp. 17–20.

[4] M. Afify, Y. Gong, and J.-P. Haton, "A general joint additive and convolutive bias compensation approach applied to noisy lombard speech recognition," *IEEE Trans. on Speech and Audio Processing*, vol. 6, no. 6, pp. 524–538, November 1998.

[5] J. L. Gauvain, L. Lamel, M. Adda-Decker, and D. Matrouf, "Developments in continuous speech dictation using the ARPA NAB news task," in *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Detroit, 1996, pp. 73–76.

[6] Y. Minami and S. Furui, "A maximum likelihood procedure for a universal adaptation method based on HMM composition," in *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Detroit, 1995, pp. 129–132.

[7] M. J. F. Gales, *Model-Based Techniques for Noise Robust Speech Recognition*, Ph.D. thesis, Cambridge University, U.K., 1995.

[8] Y. Gong, "A robust continuous speech recognition system for mobile information devices (invited paper)," in *Proc. of International Workshop on Hands-Free Speech Communication*, Kyoto, Japan, April 2001.

[9] Y. Gong, "Model-space compensation of microphone and noise for speaker-independent speech recognition," in *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Hong Kong, April 2003.

[10] S. Sagayama, Y. Yamaguchi, and S. Takahashi, "Jacobian adaptation of noisy speech models," in *Proceedings of IEEE Automatic Speech Recognition Workshop*, Santa Barbara, CA, USA, DEC 1997, pp. 396–403, IEEE Signal Processing Society.

[11] N. S. Kim, "Statistical linear approximation for environment compensation," *IEEE Signal Processing Letters*, vol. 5, no. 1, pp. 8–10, Jan. 1998.

[12] M. J. F. Gales and S. Young, "An improved approach to the hidden Markov model decomposition of speech and noise," in *Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, U.S.A., April 1992, vol. I, pp. 233–236.

[13] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society*, vol. 39, no. 1, pp. 1–38, 1977.

[14] Y. Gong, "A method of joint compensation of additive and convolutive distortions for speaker-independent speech recognition," *IEEE Trans. on Speech and Audio Processing*, (submitted for publication) 2002.