An Improved VTS Feature Compensation using Mixture Models of Distortion and IVN Training for Noisy Speech Recognition

Jun Du and Qiang Huo, *Member, IEEE*

***Abstract*—In our previous work, we proposed a feature compensation approach using high-order vector Taylor series (VTS) approximation for noisy speech recognition. In this paper, we report new progress on making it more powerful and practical in real applications. First, mixtures of densities are used to enhance the distortion models of both additive noise and convolutional distortion. New formulations for maximum likelihood (ML) estimation of distortion model parameters, and minimum mean squared error (MMSE) estimation of clean speech are derived and presented. Second, we improve the feature compensation in both efficiency and accuracy by applying higher order information of VTS approximation only to the noisy speech mean parameters, and a temporal smoothing operation for the posterior probability of Gaussian mixture components in clean speech estimation. Finally, we design a procedure to perform irrelevant variability normalization (IVN) based joint training of a reference Gaussian mixture model (GMM) for feature compensation and hidden Markov models (HMMs) for acoustic modeling using VTS-based feature compensation. The effectiveness of our proposed approach is confirmed by experiments on Aurora3 benchmark database for a real-world in-vehicle connected digits recognition task. Compared with ETSI advanced front-end, our approach achieves significant recognition accuracy improvement across three "training-testing" conditions for four languages.**

***Index Terms*—Feature compensation, irrelevant variability normalization, mixture model of distortion, noisy speech recognition, vector Taylor series.**

I. Introduction

BEFORE the recent breakthrough of deep learning approaches to automatic speech recognition (ASR) (e.g., [16] and the references therein), historically, most of ASR systems use Mel-frequency cepstral coefficients (MFCCs) and their derivatives as speech features, and a set of Gaussian mixture continuous density HMMs (CDHMMs) for modeling basic speech units. It is well known that the performance of such an ASR system trained with clean speech will degrade significantly when the testing speech is corrupted by additive noises and convolutional distortions. One type of approaches to dealing with the above problem is the so-called feature compensation approach using *explicit* model of environmental distortions (e.g., [1]), which is also the topic of this paper.

For our approach, it is assumed that in time domain, "corrupted" speech $y[t]$ is subject to the following *explicit* distortion model:

$$y[t] = x[t] \otimes h[t] + n[t] \tag{1}$$

where independent signals $x[t]$, $h[t]$ and $n[t]$ represent the tth sample of clean speech, the convolutional (e.g., transducer and transmission channel) distortion and the additive noise, respectively; and $\otimes$ is the convolution operator. In log-power-spectral domain denoted by superscript 'l', the distortion model can be expressed *approximately* (e.g., [1]) as

$$\exp(\mathbf{y}^{\mathrm{l}}) = \exp(\mathbf{x}^{\mathrm{l}} + \mathbf{h}^{\mathrm{l}}) + \exp(\mathbf{n}^{\mathrm{l}}) \tag{2}$$

where $\mathbf{y}^{\mathrm{l}}$, $\mathbf{x}^{\mathrm{l}}$, $\mathbf{h}^{\mathrm{l}}$ and $\mathbf{n}^{\mathrm{l}}$ are log power-spectra of noisy speech, clean speech, convolutional term and noise, respectively. In MFCC domain denoted by superscript 'c', the distortion model becomes

$$\mathbf{y}^{\mathrm{c}} = \mathbf{C}\log[\exp(\mathbf{C}^{+}(\mathbf{x}^{\mathrm{c}} + \mathbf{h}^{\mathrm{c}})) + \exp(\mathbf{C}^{+}\mathbf{n}^{\mathrm{c}})] \tag{3}$$

where $\mathbf{C}$ is a $D^{\mathrm{c}} \times D^{\mathrm{l}}$ truncated discrete cosine transform (DCT) matrix, $\mathbf{C}^{+}$ denotes the Moore-Penrose inverse of $\mathbf{C}$ (refer to [1], [19], [17] for details), D^{c} is the dimension of MFCC feature vector, and D^{l} is the number of channels of the Mel-frequency filter bank used in MFCC feature extraction. In most ASR systems, $D^{\mathrm{c}} < D^{\mathrm{l}}$. The log and exp functions in the above equations operate element-by-element on the corresponding vectors. The nonlinear nature of the above distortion model makes statistical modeling and inference of the above variables difficult, therefore certain approximations have to be made.

Understandably, a simple linear approximation, namely first-order vector Taylor series (VTS) approximation, has been tried in the past (e.g., [22], [23], [19]). The related works of VTS-based feature compensation can be divided into several categories. The first category is on the more precise expression of the distortion model in Eq. (2). An example is given in [7], where the phase relationship between clean speech and additive noise is incorporated into the distortion model. In [14], [15], Monte-Carlo methods are used for approximating the required intractable integrals involving the so-called "devil function" generated from the interaction model (i.e., exact distortion

Manuscript received November 07, 2013; revised January 28, 2014; accepted July 16, 2014. Date of publication July 23, 2014; date of current version August 01, 2014. This work was supported in part by the National Natural Science Foundation of China under Grant 61305002. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Shinji Watanabe.

J. Du is with the National Engineering Laboratory for Speech and Language Information Processing (NEL-SLIP), University of Science and Technology of China, Hefei 230027, China (e-mail: jundu@ustc.edu.cn).

Q. Huo is with Microsoft Research Asia, Beijing 100080, China (e-mail: qianghuo@microsoft.com).

Digital Object Identifier 10.1109/TASLP.2014.2341912

model) associated with probability densities. The second category is on the more accurate approximation of the nonlinear distortion model. In [20], a linear function is found to approximate the high-order Taylor series expansion of the nonlinear distortion model with only additive noise by minimizing the mean-squared error incurred by this approximation. Given the linear function, the remaining inference is the same as in using the traditional first-order Taylor series expansion to approximate the nonlinear distortion model directly under the Gaussian assumptions for both speech and noise. In [8], high-order Taylor series expansion is used to approximate the nonlinear portion of the distortion function by expanding with respect to $\mathbf{n}^l - \mathbf{x}^l$ instead of $(\mathbf{x}^l, \mathbf{n}^l)$. Both approaches work for each feature dimension independently by ignoring correlations between different channels of the filter bank. In [27], the nonlinear distortion model is approximated by a second-order VTS. Using this relation, the mean vector of the relevant noisy speech feature vector can be derived, which naturally includes a term related to the second-order term in high-order VTS. In terms of using high-order Taylor series expansion, the main difference of the approaches in [27], [20], [8] is that the approximation operation is performed in MFCC domain where the vector form of Taylor series expansion is adopted in [27] while the scale version of Taylor series expansion for each channel of the filter bank is used in [20], [8]. More recently, we proposed a high-order VTS based formulation for maximum likelihood (ML) estimation of both additive noise and convolutional distortion, and minimum mean squared error (MMSE) estimation of clean speech in [9] where correlations between different channels of the filter bank can be considered. The third category is on improving the recognition accuracy in non-stationary environments. In [7], [27], sequential noise estimation is performed to deal with non-stationary noise. The last category is on extension from traditional VTS-based feature compensation under clean-condition training to real scenarios, where noisy speech can also be included in training data. In [21], noise adaptive training (NAT) (e.g., [18]) is used to train a Gaussian mixture model (GMM) for VTS-based feature compensation.

In our recent work [10], [11], new progress has been made to improve [9], which can be summarized as follows: 1) both efficiency and accuracy of feature compensation can be improved by applying higher order information of VTS approximation only to the noisy speech mean parameters, and a temporal smoothing operation for the posterior probability of Gaussian mixture components in clean speech estimation, 2) irrelevant variability normalization (IVN) based joint training of a reference GMM for feature compensation and HMMs for acoustic modeling is proposed, which outperforms using the IVN-based method in [17] for GMM training, 3) mixture models for modeling both additive noise and convolutional distortion are adopted to improve the recognition accuracy in non-stationary noise environments. This is related to a recent work in [12], where a similar idea of using noise mixture model is proposed. However, the method in [12] differs significantly from ours. First, the estimation of distortion parameters and clean speech associated with VTS approximation is conducted in Log-Mel-Filter-Bank (LMFB) domain in [12], where the correlations between different channels of the filter bank are not taken into consideration as all the covariance matrices in the distributions of clean speech and noise are diagonal. However, in our approach the estimation is performed in MFCC domain where the final feature vector is fed to the recognizer and the problem of correlation between different dimensions is alleviated. Second, mixture model is only used for additive noise and the estimation of noise mixture model and bias vector (i.e., convolutional distortion in our approach) is in an alternate manner of switching between different auxiliary functions by using MMSE estimation of clean speech and noise in [12], while we use mixture models for both additive noise and convolutional distortion and closed-form formulations can be derived by jointly optimizing all the parameters of the distortion model using a unique auxiliary function for ML estimation. Furthermore, our formulations are generalized to VTS with any order. Third, for noise suppression, a Mel-scaled Wiener filter is exploited in [12] while we use MMSE estimation of clean speech. Finally, the method is verified under clean-condition training where VTS-based feature compensation is only performed on the testing set with synthesized noisy speech in [12], while we use IVN-based joint training to extend VTS-based feature compensation to any "training-testing" condition and verify our approach on noisy speech from real environments. In this paper, we expand on the above work, providing a more detailed description of the formulation and derivation, new experiments, and an expanded discussion of results.

The rest of the paper is organized as follows. In Section II, we give a brief introduction of conventional VTS-based feature compensation framework. In Section III, we describe an improved version of VTS-based feature compensation using mixture models of distortion and several other modifications. In Section IV, we present a detailed procedure for IVN-based joint training of GMM and HMMs using VTS-based feature compensation. In Section V, we report experimental results and analysis. Finally, we conclude the paper in Section VI.

II. Overview of Conventional VTS-Based Feature Compensation Framework

The flowchart of a conventional VTS-based feature compensation framework is illustrated in Fig. 1. In training stage, a reference GMM for VTS-based feature compensation and HMMs for acoustic modeling are trained from clean speech using MFCC features under maximum likelihood (ML) criterion. In recognition stage, first both the feature vector of an unknown utterance and the reference GMM are transformed from MFCC domain to log-power-spectral domain. Then by applying high-order VTS approximation to explicit distortion model, the required statistics are calculated and transformed back to MFCC domain, followed by estimation of parameters for additive noise and convolutional distortion (channel) under ML criterion. Finally, clean speech is estimated using MMSE criterion and fed to the recognizer. The details of VTS-based feature compensation module will be described in Section III, which is an extension of the formulations in [9].

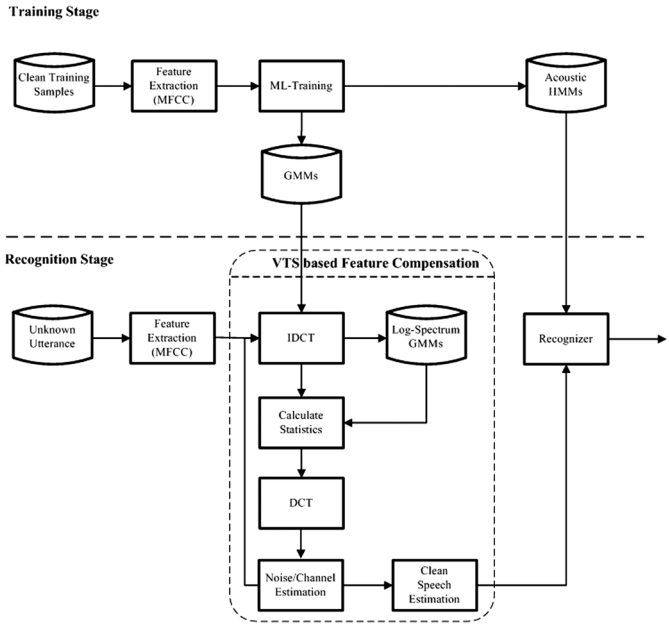


Fig. 1. Flowchart of a conventional VTS-based feature compensation framework.

III. Improved VTS-Based Feature Compensation

A. Mixture Models of Distortion

In [9], the clean speech is modeled by a GMM as follows:

$$p(\mathbf{x}_t^{\mathrm{c}}) = \sum_{m=1}^{M} \omega_{\mathbf{x},m} \mathcal{N}(\mathbf{x}_t^{\mathrm{c}}; \boldsymbol{\mu}_{\mathbf{x},m}^{\mathrm{c}}, \Sigma_{\mathbf{x},m}^{\mathrm{c}}) \tag{4}$$

where $\boldsymbol{\mu}_{\mathbf{x},m}^{\mathrm{c}}$, $\Sigma_{\mathbf{x},m}^{\mathrm{c}}$, and $\omega_{\mathbf{x},m}$ are mean vector, diagonal covariance matrix, and mixture weight of the mth Gaussian component, respectively. For each utterance, it is often assumed that the additive noise $\mathbf{n}_t^{\mathrm{c}}$ follows a Gaussian probability density function (PDF) while the convolutional distortion $\mathbf{h}_t^{\mathrm{c}}$ has a PDF of the Kronecker delta function. In this work, to enhance the modeling power for distortions, mixture models are employed to model both additive noise and convolutional distortion as follows:

$$p(\mathbf{n}_t^{\mathrm{c}}) = \sum_{l=1}^{L} \omega_{\mathbf{n},l} \mathcal{N}(\mathbf{n}_t^{\mathrm{c}}; \boldsymbol{\mu}_{\mathbf{n},l}^{\mathrm{c}}, \Sigma_{\mathbf{n},l}^{\mathrm{c}}) \tag{5}$$

$$p(\mathbf{h}_t^{\mathrm{c}}) = \sum_{k=1}^{K} \omega_{\mathbf{h},k} \delta(\mathbf{h}_t^{\mathrm{c}} - \mathbf{h}_{\mathrm{const},k}^{\mathrm{c}}) \tag{6}$$

where GMM and mixture of Kronecker delta functions are used for modeling additive noise and convolutional distortion, respectively. $\boldsymbol{\mu}_{\mathbf{n},l}^{\mathrm{c}}$, $\Sigma_{\mathbf{n},l}^{\mathrm{c}}$, and $\omega_{\mathbf{n},l}$ are mean vector, diagonal covariance matrix, and mixture weight of the lth Gaussian component for the distribution of additive noise $\mathbf{n}_t^{\mathrm{c}}$, respectively. $\mathbf{h}_{\mathrm{const},k}^{\mathrm{c}}$ and $\omega_{\mathbf{h},k}$ are deterministic vector and mixture weight of the kth component for the distribution of convolutional distortion $\mathbf{h}_t^{\mathrm{c}}$. In our implementation, the mixture number of additive noise L is set equal to the mixture number of convolutional distortion K as we assume that each pair of mixture component can roughly model a stationary segment of an utterance. Also we should define a new random vector, $\mathbf{z}_t^{\mathrm{c}} = \mathbf{x}_t^{\mathrm{c}} + \mathbf{h}_t^{\mathrm{c}}$, whose PDF can be derived as follows:

$$p(\mathbf{z}_t^{\mathrm{c}}) = \sum_{m=1}^{M} \sum_{k=1}^{K} \omega_{\mathbf{x},m} \omega_{\mathbf{h},k} \mathcal{N}(\mathbf{z}_t^{\mathrm{c}}; \boldsymbol{\mu}_{\mathbf{x},m}^{\mathrm{c}} + \mathbf{h}_{\mathrm{const},k}^{\mathrm{c}}, \Sigma_{\mathbf{x},m}^{\mathrm{c}}). \tag{7}$$

The above unknown distortion model parameters can be estimated as follows:

Step 1: *Initialization*

For each utterance, we determine the mixture number by setting $L = K = \left\lceil \frac{T}{T_{Seg}} \right\rceil$, where T_{Seg} and T are the length of a relatively stationary segment and the current utterance, respectively. First the parameters of a single Gaussian for $\mathbf{n}_t^{\mathrm{c}}$ and a Kronecker delta function for $\mathbf{h}_t^{\mathrm{c}}$ are estimated using the whole utterance as in [9]. Then based on this global set of parameters as the initialization for each segment, the second-pass re-estimation is performed using frames of each segment separately to obtain L sets of parameters. Finally, each component pair of mixture models of distortion in Eq. (5) and Eq. (6) is initialized by the set of parameters in the corresponding segment. And all mixture weights are set to equal.

Step 2: *Computation of required statistics*

First transform all parameters from cepstral domain to log-power-spectral domain as follows:

$$\boldsymbol{\mu}_{\mathbf{z},mk}^{\mathrm{l}} = \mathbf{C}^{+}(\boldsymbol{\mu}_{\mathbf{x},m}^{\mathrm{c}} + \mathbf{h}_{\mathrm{const},k}^{\mathrm{c}}) \tag{8}$$

$$\Sigma_{\mathbf{z},m}^{\mathrm{l}} = \mathbf{C}^{+}\Sigma_{\mathbf{x},m}^{\mathrm{c}}(\mathbf{C}^{+})^{\top} \tag{9}$$

$$\boldsymbol{\mu}_{\mathbf{n},l}^{\mathrm{l}} = \mathbf{C}^{+}\boldsymbol{\mu}_{\mathbf{n},l}^{\mathrm{c}} \tag{10}$$

$$\Sigma_{\mathbf{n},l}^{\mathrm{l}} = \mathbf{C}^{+}\Sigma_{\mathbf{n},l}^{\mathrm{c}}(\mathbf{C}^{+})^{\top}. \tag{11}$$

Then with those parameters in log-power-spectral domain, use high-order VTS approximation which is elaborated in Section III-B, to calculate the relevant statistics, $\boldsymbol{\mu}_{\mathbf{y},mkl}^{\mathrm{l}}$, $\Sigma_{\mathbf{y},mkl}^{\mathrm{l}}$, $\Sigma_{\mathbf{zy},mkl}^{\mathrm{l}}$, $\Sigma_{\mathbf{ny},mkl}^{\mathrm{l}}$, which are required for re-estimation of distortion model parameters and estimation of clean speech. $\Sigma_{\mathbf{zy},mkl}^{\mathrm{l}}$ is the covariance matrix of $\mathbf{z}$ and $\mathbf{y}$ while $\Sigma_{\mathbf{ny},mkl}^{\mathrm{l}}$ is the covariance matrix of $\mathbf{n}$ and $\mathbf{y}$ for mixture component mkl in log-power-spectral domain. Finally, transform the statistics back to cepstral domain as follows:

$$\boldsymbol{\mu}_{\mathbf{y},mkl}^{\mathrm{c}} = \mathbf{C}\boldsymbol{\mu}_{\mathbf{y},mkl}^{\mathrm{l}} \tag{12}$$

$$\boldsymbol{\Sigma}_{\mathbf{y},mkl}^{\mathrm{c}} = \mathbf{C}\boldsymbol{\Sigma}_{\mathbf{y},mkl}^{\mathrm{l}}(\mathbf{C})^{\top} \tag{13}$$

$$\boldsymbol{\Sigma}_{\mathbf{zy},mkl}^{\mathrm{c}} = \mathbf{C}\boldsymbol{\Sigma}_{\mathbf{zy},mkl}^{\mathrm{l}}(\mathbf{C})^{\top} \tag{14}$$

$$\boldsymbol{\Sigma}_{\mathbf{ny},mkl}^{\mathrm{c}} = \mathbf{C}\boldsymbol{\Sigma}_{\mathbf{ny},mkl}^{\mathrm{l}}(\mathbf{C})^{\top}. \tag{15}$$

Step 3: *Joint re-estimation of distortion model parameters*

Use Eq. (17) to Eq. (21) to re-estimate the distortion model parameters. Note that the cepstral domain indicator "c" in relevant variables has been dropped for notational convenience. The detailed derivations for joint re-estimation can be referred to Appendix A, which can be extended from those in

[26], [9]. Several items used in Eq. (17) to Eq. (21) are evaluated in Eq. (22) to Eq. (25), where the statistics $\boldsymbol{\mu}_{\mathbf{y},mkl}$, $\boldsymbol{\Sigma}_{\mathbf{y},mkl}$, $\boldsymbol{\Sigma}_{\mathbf{zy},mkl}$, $\boldsymbol{\Sigma}_{\mathbf{ny},mkl}$ are calculated in Step 2.

Step 4: *Repeat Step 2 and Step 3 N_{VTS} times*

Given the noisy speech and the estimated distortion model parameters, MMSE estimation of clean speech can be calculated as

$$\hat{\mathbf{x}}_t = E_{\mathbf{x}}\left[\mathbf{x}_t|\mathbf{y}_t\right] = \sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{l=1}^{L} P(m,k,l|\mathbf{y}_t)(E_{\mathbf{z}}[\mathbf{z}_t|\mathbf{y}_t,m,k,l] - \mathbf{h}_{\text{const},k}). \tag{16}$$

B. Use of Higher Order Information

In [9], high-order VTS approximation of the nonlinear distortion function is applied to the calculation of all required statistics in log-power-spectral domain. First, the explicit distortion model in Eq. (2) is reformulated in the scalar form as follows:

$$y = f(z,n) = \log(\exp(z) + \exp(n)) \tag{26}$$

where $z = x + h$ and the log-power-spectral domain indicator "l" is dropped for notational convenience. Then the P-order Taylor series of $f(z,n)$ with the expansion point (μ_z, μ_n) can be represented as

$$\begin{aligned} &f_P(z,n) \\ &= \sum_{p=0}^{P}\frac{1}{p!}\left[(z-\mu_z)\frac{\partial}{\partial z} + (n-\mu_n)\frac{\partial}{\partial n}\right]^p f(\mu_z,\mu_n) \\ &= \sum_{p=0}^{P}\sum_{r=0}^{p} A(p,r)(z-\mu_z)^{p-r}(n-\mu_n)^r \end{aligned} \tag{27}$$

where

$$A(p,r) = \frac{1}{r!(p-r)!}\left.\frac{\partial^p f(z,n)}{\partial z^{p-r}\partial n^r}\right|_{(\mu_z,\mu_n)} \tag{28}$$

and

$$\left.\frac{\partial^p f(z,n)}{\partial z^{p-r}\partial n^r}\right|_{(\mu_z,\mu_n)} = \begin{cases} \log(\exp(\mu_z)+\exp(\mu_n)), & p=0, r=0 \\ 1 - \frac{1}{1+\exp(\mu_n-\mu_z)}, & p=1, r=1 \\ \frac{1}{1+\exp(\mu_n-\mu_z)}, & p=1, r=0 \\ (-1)^{p-r}\sum_{q=1}^{p}\frac{B(p,q)}{[1+\exp(\mu_n-\mu_z)]^q}, & p>1 \end{cases}. \tag{29}$$

When $p > 1$ and $p \geq q \geq 1$, the coefficients $B(p,q)$ in Eq. (29) can be evaluated by using the following recursive relation

$$B(p,q) = (q-1)B(p-1,q-1) - pB(p-1,q) \tag{30}$$

with the initial condition

$$B(1,1) = -1, B(p,0) = B(p,p+1) = 0, \quad p \geq 1. \tag{31}$$

Given the above notations and results, the required statistics $\boldsymbol{\mu}^{\text{l}}_{\mathbf{y},mkl}$, $\boldsymbol{\Sigma}^{\text{l}}_{\mathbf{y},mkl}$, $\Sigma^{\text{l}}_{\mathbf{zy},mkl}$, $\Sigma^{\text{l}}_{\mathbf{ny},mkl}$ can be calculated using relevant expectations of $f_P(z,n)$ in Eq. (27). For notational convenience, the channel index "l" and mixture component index "mkl" have been dropped hereinafter. Let's use $\mu_y(i)$ to denote the ith element of the vector $\boldsymbol{\mu}_{\mathbf{y}}$, and use $\sigma^2_y(i,j)$, $\sigma^2_{zy}(i,j)$,

$$\overline{\omega}_{\mathbf{n},l} = \frac{1}{T}\sum_{t=1}^{T}\sum_{m=1}^{M}\sum_{k=1}^{K} P(m,k,l|\mathbf{y}_t) \tag{17}$$

$$\overline{\boldsymbol{\mu}}_{\mathbf{n},l} = \frac{\sum_{t=1}^{T}\sum_{m=1}^{M}\sum_{k=1}^{K} P(m,k,l|\mathbf{y}_t)E_{\mathbf{n}}[\mathbf{n}_t|\mathbf{y}_t,m,k,l]}{\sum_{t=1}^{T}\sum_{m=1}^{M}\sum_{k=1}^{K} P(m,k,l|\mathbf{y}_t)} \tag{18}$$

$$\overline{\Sigma}_{\mathbf{n},l} = \frac{\sum_{t=1}^{T}\sum_{m=1}^{M}\sum_{k=1}^{K} P(m,k,l|\mathbf{y}_t)E_{\mathbf{n}}[\mathbf{n}_t\mathbf{n}_t^{\top}|\mathbf{y}_t,m,k,l]}{\sum_{t=1}^{T}\sum_{m=1}^{M}\sum_{k=1}^{K} P(m,k,l|\mathbf{y}_t)} - \overline{\boldsymbol{\mu}}_{\mathbf{n},l}\overline{\boldsymbol{\mu}}_{\mathbf{n},l}^{\top} \tag{19}$$

$$\overline{\omega}_{\mathbf{h},k} = \frac{1}{T}\sum_{t=1}^{T}\sum_{m=1}^{M}\sum_{l=1}^{L} P(m,k,l|\mathbf{y}_t) \tag{20}$$

$$\overline{\mathbf{h}}_{\text{const},k} = \left[\sum_{t=1}^{T}\sum_{m=1}^{M}\sum_{l=1}^{L} P(m,k,l|\mathbf{y}_t)\boldsymbol{\Sigma}^{-1}_{\mathbf{x},m}\right]^{-1} \left[\sum_{t=1}^{T}\sum_{m=1}^{M}\sum_{l=1}^{L} P(m,k,l|\mathbf{y}_t)\boldsymbol{\Sigma}^{-1}_{\mathbf{x},m}(E_{\mathbf{z}}[\mathbf{z}_t|\mathbf{y}_t,m,k,l] - \boldsymbol{\mu}_{\mathbf{x},m})\right] \tag{21}$$

$$P(m,k,l|\mathbf{y}_t) = \frac{\omega_{\mathbf{x},m}\omega_{\mathbf{h},k}\omega_{\mathbf{n},l}\mathcal{N}(\mathbf{y}_t;\boldsymbol{\mu}_{\mathbf{y},mkl},\boldsymbol{\Sigma}_{\mathbf{y},mkl})}{\sum_{m=1}^{M}\sum_{k=1}^{K}\sum_{l=1}^{L}\omega_{\mathbf{x},m}\omega_{\mathbf{h},k}\omega_{\mathbf{n},l}\mathcal{N}(\mathbf{y}_t;\boldsymbol{\mu}_{\mathbf{y},mkl},\boldsymbol{\Sigma}_{\mathbf{y},mkl})} \tag{22}$$

$$E_{\mathbf{n}}[\mathbf{n}_t|\mathbf{y}_t,m,k,l] = \boldsymbol{\mu}_{\mathbf{n},l} + \boldsymbol{\Sigma}_{\mathbf{ny},mkl}\boldsymbol{\Sigma}^{-1}_{\mathbf{y},mkl}(\mathbf{y}_t - \boldsymbol{\mu}_{\mathbf{y},mkl}) \tag{23}$$

$$E_{\mathbf{z}}[\mathbf{z}_t|\mathbf{y}_t,m,k,l] = (\boldsymbol{\mu}_{\mathbf{x},m} + \mathbf{h}_{\text{const},k}) + \boldsymbol{\Sigma}_{\mathbf{zy},mkl}\boldsymbol{\Sigma}^{-1}_{\mathbf{y},mkl}(\mathbf{y}_t - \boldsymbol{\mu}_{\mathbf{y},mkl}) \tag{24}$$

$$E_{\mathbf{n}}[\mathbf{n}_t\mathbf{n}_t^{\top}|\mathbf{y}_t,m,k,l] = E_{\mathbf{n}}[\mathbf{n}_t|\mathbf{y}_t,m,k,l]E^{\top}_{\mathbf{n}}[\mathbf{n}_t|\mathbf{y}_t,m,k,l] + \boldsymbol{\Sigma}_{\mathbf{n},l} - \boldsymbol{\Sigma}_{\mathbf{ny},mkl}\boldsymbol{\Sigma}^{-1}_{\mathbf{y},mkl}\boldsymbol{\Sigma}_{\mathbf{yn},mkl} \tag{25}$$

$\sigma^2_{ny}(i,j)$, to denote the (i,j)th element of the matrix $\mathbf{\Sigma_y}$, $\mathbf{\Sigma_{zy}}$, $\mathbf{\Sigma_{ny}}$, respectively. Then using the definition of those parameters and Eq. (27), we have

$$\begin{aligned}\mu_y(i) &\approx E^i_{zn}[f_P(z,n)] \\ &= \sum_{p=0}^{P}\sum_{r=0}^{p} A^i(p,r)E^i_{zn}[(z-\mu_z)^{p-r}(n-\mu_n)^r] \\ &= \sum_{p=0}^{P}\sum_{r=0}^{p} A^i(p,r)M^i_n(r)M^i_z(p-r) \qquad (32)\\ \sigma^2_y(i,j) &\approx E^{ij}_{zn}[f_P(z,n),f_P(z,n)] - \mu_y(i)\mu_y(j) \\ &= \sum_{p_1=0}^{P}\sum_{r_1=0}^{p_1}\sum_{p_2=0}^{P}\sum_{r_2=0}^{p_2}[A^i(p_1,r_1)A^j(p_2,r_2) \\ &\quad M^{ij}_n(r_1,r_2)M^{ij}_z(p_1-r_1,p_2-r_2)] - \mu_y(i)\mu_y(j) \qquad (33)\\ \sigma^2_{zy}(i,j) &= E^{ij}_{zn}[(z-\mu_z),(y-\mu_y)] \\ &\approx \sum_{p=0}^{P}\sum_{r=0}^{p} A^j(p,r)M^j_n(r)M^{ij}_z(1,p-r) \qquad (34)\\ \sigma^2_{ny}(i,j) &= E^{ij}_{zn}[(n-\mu_n),(y-\mu_y)] \\ &\approx \sum_{p=0}^{P}\sum_{r=0}^{p} A^j(p,r)M^{ij}_n(1,r)M^j_z(p-r). \qquad (35)\end{aligned}$$

where

$$M^i_\Delta(q) = \begin{cases} 0, & \text{if } q \text{ is odd} \\ (q-1)!!\sigma^q_\Delta(i), & \text{otherwise} \end{cases} \qquad (36)$$

and

$$M^{ij}_\Delta(q,s) = \begin{cases} 0, & \text{if } q+s \text{ is odd} \\ q!s!2^{-\frac{q+s}{2}}\sum_{0\le u\le min(q,s)}^{q-u} \text{ is even} & \\ \frac{2^u}{u!(\frac{q-u}{2})!(\frac{s-u}{2})!}\sigma^{q-u}_\Delta(i,i) & \\ \sigma^{2u}_\Delta(i,j)\sigma^{s-u}_\Delta(j,j), & \text{otherwise} \end{cases}. \qquad (37)$$

Δ represents 'z' or 'n'. $A^i(p,r)$ is the value of Eq. (28) for the ith dimension.

However, according to our experiments, inconsistent improvements of recognition performance are observed on different Aurora3 tasks by applying higher order VTS approximation to all statistics. One possible reason is that the approximated calculation of statistics is not accurate due to our model assumptions, especially for those variance and covariance parameters. In this study, higher order information of VTS approximation is only applied to the calculation of noisy speech mean parameters. Our new experiments show that consistent improvements of recognition performance can be achieved, yet its computational complexity is much lower than that of the original high-order VTS so that the additional computation cost can be ignored compared with full operations of first-order VTS.

C. Use of Acoustic Context Information

We use acoustic context information in clean speech estimation to further improve the accuracy. Acoustic context information has been widely used in several feature extraction/transformation methods, such as TANDEM [13] and fMPE [24], where in addition to the current frame, the information from several neighboring frames in the left and right context is also used. In our VTS-based feature compensation, MMSE estimation of clean speech feature vector $\mathbf{x}_t$ in Eq. (16) only uses the information of noisy speech in the tth frame $\mathbf{y}_t$. To leverage acoustic context information, we calculate the new posterior probability by a weighted average among neighboring frames as follows:

$$[P(m,k,l|\mathbf{y}_t)]_{\text{new}} = \frac{\sum_{\tau=-\Delta}^{\Delta}(\Delta+1-|\tau|)P(m,k,l|\mathbf{y}_{t+\tau})}{\sum_{\tau=-\Delta}^{\Delta}(\Delta+1-|\tau|)} \qquad (38)$$

where Δ is the size of acoustic context. In Eq. (38), a temporal smoothing operation of posterior probabilities using weighted auto-regression and moving-average (ARMA) filter is adopted, which is more effective than the traditional ARMA filtering in [6].

IV. IVN-Based Joint Training of GMM and HMMs

A. System Overview

In the traditional framework of VTS-based feature compensation, both HMMs for recognition and reference GMM for feature compensation are trained on clean speech data. In real scenarios, the training data may include noisy speech data. So the reference GMM and HMMs are first initialized from multi-condition data. Then those models are updated by compensated (pseudo-clean) features after VTS-based feature compensation, which are called as *generic* (pseudo-clean) GMM and HMMs. In [17], IVN-based HMM training using VTS-based model compensation is used to train *generic* HMMs from mixed clean and noisy speech data. In this work, we propose a novel procedure to perform IVN-based joint training of GMM and HMMs using VTS-based feature compensation, which is illustrated in Fig. 2. In the training stage, the procedure is as follows:

Step 1: *Initialization*
First, the reference GMM for feature compensation and HMMs for recognition are trained from multi-condition training data using MFCC features with cepstral mean normalization (CMN).

Step 2: *VTS-based feature compensation*
Given the reference GMM, VTS-based feature compensation is applied to each training utterance.

Step 3: *Joint training of GMM and HMMs*
Based on the compensated features of training set, single pass retraining (SPR) [29] is performed to generate the *generic* GMM and HMMs by using the last updated GMM and HMMs with the corresponding feature set. The SPR works as follows: given one set of well-trained models, a new set matching a different training data parameterization can be generated in a single re-estimation pass, which is done by computing the forward and backward probabilities using the original models together with the original training data and then

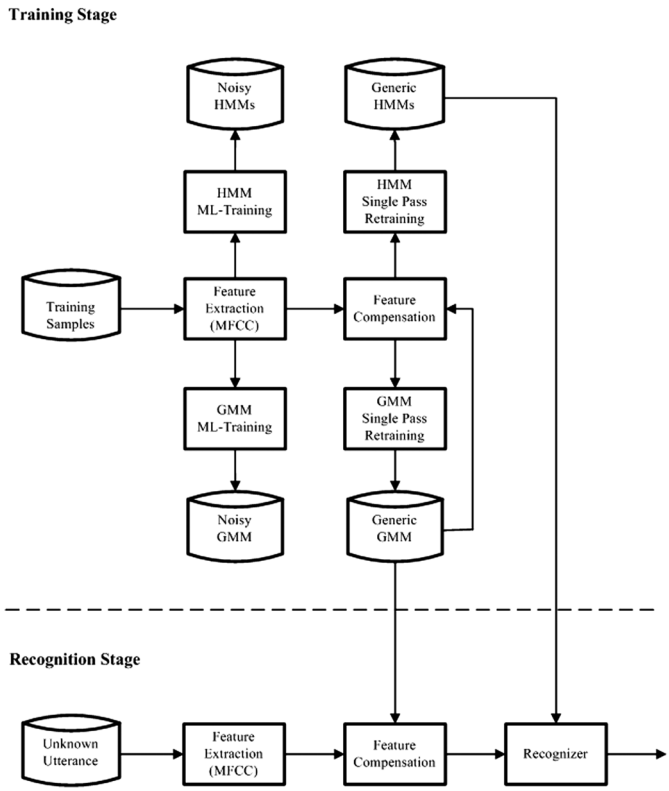


Fig. 2. Flowchart of IVN training using VTS-based feature compensation.

switching to the new training data to compute the parameter estimation for the new set of models. In our case, the original model and training data correspond to the models and compensated features in the last iteration while a new set matching a different training data parameterization refers to the models and corresponding features to be updated using VTS-based feature compensation in the current iteration.

Step 4: *Repeat Step 2 and Step 3 N_{IVN} times*

In the recognition stage, after feature extraction for an unknown utterance, we perform VTS-based feature compensation using the *generic* GMM and then do recognition using the *generic* HMMs.

B. Discussions

In the above procedure, the IVN concept is implemented by SPR using VTS-based feature compensation which is denoted as IVN-1. Actually, there are other two alternatives which can also achieve this goal. One method (denoted as IVN-2) is to use the compensated features to retrain GMM from scratch and then use the new GMM to compensate features again in an iterative way. Finally a *generic* GMM can be generated. The other method (IVN-3) is to use a similar procedure as in [17] to generate a *generic* GMM. In [17], IVN training using VTS-based model compensation is adopted to generate a *generic* HMM. For those two methods, the *generic* HMMs can be trained from scratch using compensated features based on *generic* GMM. The main differences among IVN-1, IVN-2, and IVN-3 are how to generate the GMM for feature compensation and HMM for acoustic modeling. For both GMM and HMM, IVN-1 uses SPR training while IVN-2 adopts the conventional retraining procedure by using the compensated features. However, IVN-3 employs a VTS-based model compensation procedure for GMM and a retraining procedure for HMM. As a comparison, SPR-based IVN training has two advantages: 1) GMM and HMMs are jointly trained in each iteration, 2) both GMM and HMMs are progressively updated, which brings stable improvements of recognition performance. Our experimental results also confirm that SPR-based IVN training can achieve better recognition performance, which is recommended as a practical solution.

V. Experiments and Results

A. Experimental Setup

In order to verify the effectiveness of the proposed approach on real-world ASR, Aurora3 databases were used, which contained utterances of digit strings recorded in real automobile environments for German, Danish, Finnish and Spanish, respectively. A full description of the above databases and the corresponding test frameworks are given in [2], [3], [4], [5].

In our ASR systems, each feature vector consisted of 13 MFCCs (including C_0) plus their first and second order derivatives. The number of Mel-frequency filter banks was 23. MFCCs were computed based on power spectrum. Each digit was modeled by a whole-word left-to-right CDHMM, which consisted of 16 emitting states, each having 3 Gaussian mixture components. Three "training-testing" conditions were designed for Aurora3. The first one was high-mismatch (HM) condition, where training data included utterances recorded by close-talking (CT) microphone, which could be considered as "clean", while testing data was recorded by hands-free (HF) microphone. The second one was well-matched (WM) condition, where both training and testing data were recorded by CT and HF microphones. The last one was mid-mismatch (MM) condition, where training data included quiet and low noisy data recorded by HF microphone while testing data included high noisy data recorded by HF microphone. The relevant control parameters were set as $M = 256$, $\Delta = 3$, $T_{Seg} = 60$, $N_{VTS} = 4$, $N_{IVN} = 4$. Other control parameters related to our previous work on VTS-based feature compensation could be found in [9]. Our baseline system used CMN for feature normalization. In all the experiments, tools in HTK [29] were used for training and testing. VTS-based feature compensation was applied to the static 13-dimensional MFCC features while the dynamic features were extracted from the compensated static features.

B. Effects of VTS Approach

In the first set of experiments, we study the effectiveness of several innovations in our proposed VTS approach under the "clean-condition" training. Table I summarizes a performance (word accuracy in %) comparison of several robust ASR systems using VTS-based feature compensation in the high-mismatch (HM) condition on Aurora3 databases. VTS-Old refers to the practical solution of feature compensation recommended in [9], namely CMN + VTS(N,H)(MMSE-VTS0) where a first-order VTS approximation in distortion model parameter

TABLE I
PERFORMANCE (WORD ACCURACY IN %) COMPARISON OF SEVERAL ROBUST ASR SYSTEMS USING VTS-BASED FEATURE COMPENSATION IN THE HIGH-MISMATCH (HM) CONDITION ON AURORA3 DATABASES

Methods	German	Danish	Finnish	Spanish
VTS-Old	91.03	76.92	86.29	85.35
VTS-HO-1	90.98	76.85	87.94	86.48
VTS-HO-2	91.52	77.28	89.13	86.83
VTS	91.77	77.39	90.46	87.46
MMD-VTS	92.46	77.94	91.17	89.44
IVN-SMMD-VTS	92.41	80.29	91.63	89.86
IVN-MMD-VTS	92.74	80.64	92.61	91.28

TABLE II
PERFORMANCE (WORD ACCURACY IN %) COMPARISON OF THE BASELINE SYSTEM AND SEVERAL ROBUST ASR SYSTEMS USING VTS-BASED FEATURE COMPENSATION WITH DIFFERENT IVN TRAINING APPROACHES UNDER THREE "TRAINING-TESTING" CONDITIONS ON AURORA3 DATABASE

Methods		German	Danish	Finnish	Spanish
Baseline	HM	83.77	54.78	77.07	80.99
	MM	82.43	75.42	84.06	89.39
	WM	92.49	90.84	93.09	93.57
IVN-1	HM	92.09	**79.98**	**91.55**	**88.90**
	MM	89.24	78.53	**87.48**	**91.44**
	WM	**94.93**	**92.91**	**95.64**	**95.57**
IVN-2	HM	**92.28**	75.94	90.32	88.18
	MM	89.46	78.39	86.25	91.11
	WM	94.67	92.33	94.40	95.15
IVN-3	HM	92.09	75.63	89.86	87.70
	MM	**89.53**	**78.81**	86.46	91.13
	WM	94.77	91.45	94.74	94.90

estimation for both additive noise and convolutional distortion (N,H), and a zero-order VTS approximation in MMSE estimation (MMSE-VTS0) for clean speech is used. VTS-HO-1 uses a second-order VTS approximation to calculate all required parameters while VTS-HO-2 only applies a second-order VTS approximation to the calculation of noisy speech mean parameters. VTS is an improved version of VTS-Old using higher order information (second order here) and acoustic context information described in Section III. MMD-VTS uses mixture models of distortion on top of VTS. IVN-MMD-VTS adds IVN-1 to MMD-VTS. IVN-SMMD-VTS is a simple version of IVN-MMD-VTS without the step of joint re-estimation of distortion model parameters in Section III-A. Several observations can be made. First, consistent and significant improvements of recognition performance can be achieved by using higher order information of VTS approximation to the noisy mean parameters and acoustic context information. Second, MMD-VTS system outperforms VTS system for four languages, which indicates that mixture models play an important role in modeling non-stationary distortions in real applications. Third, IVN-MMD-VTS system using IVN training yields further improvements over the MMD-VTS system for all testing cases in the HM condition, especially on Finnish and Spanish databases where more training data is provided than German and Danish databases. Finally, joint re-estimation of distortion model parameters based on the whole utterance in IVN-MMD-VTS system can achieve better performance than just initializing the parameters of mixture models in each segment of the utterance in IVN-SMMD-VTS system.

The second set of experiments are designed to examine the effectiveness of several IVN training approaches. Table II gives a performance (word accuracy in %) comparison of the baseline system and several robust ASR systems using VTS-based feature compensation with different IVN training approaches under three "training-testing" conditions on Aurora3 databases. Note that MMD is not used here. From those results, first all robust ASR systems using VTS-based feature compensation with IVN training significantly outperform the baseline system without using feature compensation under all "training-testing" conditions for four languages. On Finnish and Spanish databases, IVN-1 system consistently achieves much better performance than IVN-2 and IVN-3 systems, which can also be applied to the WM condition for all languages. For very few cases where IVN-1 system slightly underperforms IVN-2 and IVN-3 systems, the main reason may be due to the lack of enough training data under HM and MM conditions on German and Danish databases. Overall, IVN-1 approach using SPR can achieve the best performance in most cases, which is used in all the other experiments.

TABLE III
PERFORMANCE (WORD ACCURACY IN %) COMPARISON OF TWO ROBUST ASR SYSTEMS USING VTS-BASED FEATURE COMPENSATION AND AFE UNDER THREE "TRAINING-TESTING" CONDITIONS ON AURORA3 DATABASES

Methods		German	Danish	Finnish	Spanish
IVN-MMD-VTS	HM	92.74	80.64	92.61	91.28
	MM	89.24	80.08	88.30	91.97
	WM	95.13	92.91	96.23	95.78
AFE	HM	90.61	77.39	87.46	83.76
	MM	88.65	78.25	85.02	91.28
	WM	94.67	92.21	95.01	95.20

TABLE IV
PERFORMANCE (WORD ACCURACY IN %) COMPARISON OF THREE ROBUST ASR SYSTEMS USING OUR PROPOSED MMD-VTS APPROACH, DNA APPROACH, AND NTT APPROACH ON AURORAII + DNA DATASET

Methods	15dB	10dB	5dB	0dB	-5dB
Baseline	94.34	84.47	69.24	52.25	38.13
DNA	98.15	97.09	94.89	91.43	82.62
NTT	98.48	97.56	96.53	93.72	89.76
MMD-VTS	98.68	98.39	97.65	95.67	92.37

C. Comparison of VTS and other Approaches

In this section, we first compares our VTS approach to a representative noise robust front-end, namely ETSI advanced front-end (AFE) [30]. AFE was designed as the standard of the activity for distributed speech recognition by ETSI. This standard includes the feature extraction module and feature compression/transmission part at both the terminal and server sides. In the feature extraction part, first a two-stage Mel-warped Wiener filter is designed for noise reduction. Then the waveform processing is applied to the denoised signal and cepstral features are calculated. Finally, the blind equalization is applied to the cepstral features, which are then fed to the further compression process for channel transmission. In our evaluation, we ignore all the operations of that standard at the server side in the distributed speech recognition. Actually AFE is a very competitive noise robust front-end on Aurora3 task. Table III shows a performance (word accuracy in %) comparison of two robust ASR systems using VTS-based feature compensation and AFE under three "training-testing"

TABLE V
PERFORMANCE (WORD ACCURACY IN %) COMPARISON OF TWO ROBUST ASR SYSTEMS USING A SIMPLE VTS-BASED FEATURE COMPENSATION (IVN-SVTS) DURING THE RECOGNITION STAGE AND DIFFERENT VERSIONS OF VTS DURING THE TRAINING STAGE UNDER THREE "TRAINING-TESTING" CONDITIONS ON AURORA3 DATABASES

Training	Testing	Condition	German	Danish	Finnish	Spanish
IVN-SVTS	IVN-SVTS	HM	91.07	75.47	88.83	83.91
		MM	88.95	76.84	83.52	90.40
		WM	94.43	91.42	95.08	94.63
IVN-MMD-VTS	IVN-SVTS	HM	91.44	79.55	90.35	86.59
		MM	89.31	78.67	85.50	90.78
		WM	94.61	92.40	95.73	95.53

conditions on Aurora3 databases. We can observe that our proposed IVN-MMD-VTS system can achieve consistent and significant improvements of recognition performance over AFE system for all conditions and languages, especially under the HM condition. This is reasonable as IVN-MMD-VTS has a more powerful modeling capability for clean speech and distortions compared with AFE.

To further verify the effectiveness of our approach on non-stationary noises, we make a comparison with two other approaches, namely dynamic noise adaptation (DNA) approach in [25] and the approach proposed by NTT researchers in [12], under the $\mathrm{AuroraII} + \mathrm{DNA}$ framework which is a benchmark designed by Rennie [25]. The $\mathrm{AuroraII} + \mathrm{DNA}$ framework has been designed as an extension of the Aurora II task, created for the evaluation of adaptive speech denoising algorithms. Table IV lists a performance (word accuracy in %) comparison of three robust ASR systems using our proposed MMD-VTS approach, DNA approach, and NTT approach on $\mathrm{AuroraII} + \mathrm{DNA}$ dataset. Note that both the baseline and DNA results are cited from [25] while we evaluate the NTT and MMD-VTS approaches with the same configuration. In NTT approach, 3 mixture components are used for noise mixture model. Obviously, our proposed MMD-VTS approach yields consistently the best performance for all SNRs, especially low SNRs. The reason why both NTT and MMD-VTS approaches outperform DNA might be more precise models are used for both clean speech and noise in NTT and MMD-VTS. Meanwhile, the convolutional distortion is not considered in DNA. Our approach shares the similar VTS framework with NTT approach. But the implementations of parameter estimation for both clean speech and distortions described in Section I are quite different, which demonstrates the superiority of our approach.

D. A Practical Version of VTS Approach

Although our IVN-MMD-VTS approach yields very promising results on Aurora3 task, one concern is the computational complexity during the recognition stage. The main overhead comes from the re-estimation of distortion model parameters and the clean speech estimation using mixture models of distortion. To make our approach more practical, a simplified VTS-based feature compensation (IVN-SVTS) without any re-estimation of distortion model parameters, in which mixture models of distortion are not used and only the parameters of additive noise are initialized by first several frames of the current utterance according to Step 1 of Section II in [9], is adopted in the recognition stage, which is a tradeoff between recognition accuracy and run-time overhead. Table V lists a performance (word accuracy in %) comparison of two robust ASR systems using IVN-SVTS during the recognition stage and different versions of VTS during the training stage under three "training-testing" conditions on Aurora3 databases. With the same run-time overhead in the recognition stage, the system using IVN-MMD-VTS approach in Table III during the training stage consistently outperforms the system using IVN-SVTS approach during the training stage for all conditions and languages, which indicates that even two different versions of VTS-based feature compensation are used for training and recognition stages, joint re-estimation of distortion model parameters only for the training stage can bring further gain of recognition performance during the recognition stage. Compared with Table III, the simplified system still outperforms the AFE system in most cases though its performance is slightly worse than that of IVN-MMD-VTS system but much faster in run-time. Table VI gives readers an idea of computational complexity during the run-time for two VTS-based systems. The timing experiment is conducted on a "Pentium-4" PC with a clock rate of 2.66 GHz by using utterances from Aurora3 database. It is obvious that IVN-MMD-SVTS system which uses IVN-MMD-VTS in the training stage and IVN-SVTS in the recognition stage as in Table V is much faster than IVN-MMD-VTS system as in Table III. Another benefit from using IVN-MMD-SVTS system is that it can be implemented in an online manner without the re-estimation of parameters for distortion model using the whole utterance.

TABLE VI
SUMMARY OF REAL TIME FACTOR (RTF) FOR TWO VTS-BASED SYSTEMS

Methods	IVN-MMD-VTS	IVN-MMD-SVTS
RTF	5.68	0.45

VI. CONCLUSION AND DISCUSSIONS

In this paper, we propose to use mixture models for modeling both additive noise and convolutional distortion to improve the recognition accuracy in non-stationary environments. Combined with IVN-based joint training of a reference GMM for feature compensation and HMMs for acoustic modeling using VTS-based feature compensation, significant performance gain can be achieved under all the "training-testing" conditions on Aurora3 task. This paper improves our previous work in [9] in both algorithms and practicality for real applications. As for our future work, we aim to further improve our estimation method for parameters of distortion model and verify our approach on other tasks, such as CHiME challenge data [28].

APPENDIX A
DERIVATION OF ML TRAINING OF DISTORTION MODEL PARAMETERS

In this appendix, we summarize how to derive a procedure for the estimation of the parameters of explicit distortion model by maximizing the likelihood function defined on a given set of noisy observations in cepstral domain.

First we make assumptions that both $\mathbf{z}$ and $\mathbf{n}$ are modeled by GMMs. The likelihood function is defined as:

$$\begin{aligned}\mathcal{L}(\mathbf{Y}|\mathbf{\Lambda}) &= p(\mathbf{Y}|\mathbf{\Lambda}) = p(\mathbf{Y}|\mathbf{\Lambda_z}, \mathbf{\Lambda_n}) \\ &= \sum_{\mathbf{M_z}} \sum_{\mathbf{M_n}} p(\mathbf{Y}, \mathbf{M_z}, \mathbf{M_n}|\mathbf{\Lambda_z}, \mathbf{\Lambda_n}) \end{aligned} \tag{39}$$

where $\mathbf{\Lambda_z}$ and $\mathbf{\Lambda}_n$ are model parameter sets for $\mathbf{z}$ and $\mathbf{n}$, respectively. $\mathbf{Y}$ is the sequence of the noisy observation vectors in the current utterance. $\mathbf{M_z}$ and $\mathbf{M_n}$ are the sequences of Gaussian component indices for $\mathbf{z}$ and $\mathbf{n}$, respectively. $p(\mathbf{Y}, \mathbf{M_z}, \mathbf{M_n}|\mathbf{\Lambda_z}, \mathbf{\Lambda_n})$ can be expressed as

$$\begin{aligned}&p(\mathbf{Y}, \mathbf{M_z}, \mathbf{M_n}|\mathbf{\Lambda_z}, \mathbf{\Lambda_n}) \\ &= \iint_C \prod_{t=1}^{T} \omega_{\mathbf{z}}(m_{\mathbf{z}}^t)\omega_{\mathbf{n}}(m_{\mathbf{n}}^t) p(\mathbf{z}^t|m_{\mathbf{z}}^t) p(\mathbf{n}^t|m_{\mathbf{n}}^t) d\mathbf{Z} d\mathbf{N}\end{aligned} \tag{40}$$

where $m_{\mathbf{z}}^t$ and $m_{\mathbf{n}}^t$ are the hidden Gaussian component indices at the tth frame for $\mathbf{z}$ and $\mathbf{n}$, respectively; $\omega_{\mathbf{z}}(m_{\mathbf{z}}^t)$ and $\omega_{\mathbf{n}}(m_{\mathbf{n}}^t)$ denote the weights of the corresponding Gaussian components for $\mathbf{z}^t$ and $\mathbf{n}^t$ respectively; $p(\mathbf{z}^t|m_{\mathbf{z}}^t)$ and $p(\mathbf{n}^t|m_{\mathbf{n}}^t)$ are PDFs of Gaussian components for $\mathbf{z}$ and $\mathbf{n}$, respectively; and the notation $\iint_C$ represents the T-fold iterated integral, each component of which is along the contour C_t defined by the explicit model $f(\mathbf{z}^t, \mathbf{n}^t) = \mathbf{y}^t$. It is important to note that one can define a particular model for the corruption of the clean speech by noise simply by defining particular contours of integration C_t.

It is impossible to obtain the closed-form ML estimation directly by maximizing the likelihood function in Eq. (39). Here we adopt an iterative EM algorithm to solve the problem. The M-Step of the EM algorithm is to maximize the following auxiliary function:

$$\begin{aligned}&\mathcal{Q}(\bar{\mathbf{\Lambda}}|\mathbf{\Lambda}) \\ &= E[\log p(\mathbf{Z}, \mathbf{N}, \mathbf{M_z}, \mathbf{M_n}|\bar{\mathbf{\Lambda}})|\mathbf{Z}, \mathbf{N}, \mathbf{\Lambda}] \\ &= \sum_{\mathbf{M_z}} \sum_{\mathbf{M_n}} \iint_C p(\mathbf{Z}, \mathbf{N}, \mathbf{M_z}, \mathbf{M_n}|\mathbf{\Lambda}) \\ &\quad \log p(\mathbf{Z}, \mathbf{N}, \mathbf{M_z}, \mathbf{M_n}|\bar{\mathbf{\Lambda}}) d\mathbf{Z} d\mathbf{N}\end{aligned} \tag{41}$$

where $\mathbf{\Lambda}$ and $\bar{\mathbf{\Lambda}}$ are the sets of old and new model parameters, respectively. If we assume that the observations are independent in time, and further assume that random processes representing $\mathbf{Z}$, $\mathbf{N}$, $\mathbf{M_z}$, and $\mathbf{M_n}$ are independent, then:

$$\begin{aligned}&p(\mathbf{Z}, \mathbf{N}, \mathbf{M_z}, \mathbf{M_n}|\mathbf{\Lambda}) \\ &= \prod_{t=1}^{T} \omega_{\mathbf{z}}(m_{\mathbf{z}}^t)\omega_{\mathbf{n}}(m_{\mathbf{n}}^t) p(\mathbf{z}^t|m_{\mathbf{z}}^t) p(\mathbf{n}^t|m_{\mathbf{n}}^t).\end{aligned} \tag{42}$$

Furthermore, as $\mathbf{z} = \mathbf{x} + \mathbf{h}$, and $p(\mathbf{z})$ follows Eq. (7), $\mathcal{Q}(\bar{\mathbf{\Lambda}}|\mathbf{\Lambda})$ can be reduced to

$$\begin{aligned}\mathcal{Q}(\bar{\mathbf{\Lambda}}|\mathbf{\Lambda}) = &\sum_{t=1}^{T} \sum_{k_{\mathbf{x}}=1}^{K_{\mathbf{x}}} \sum_{k_{\mathbf{h}}=1}^{K_{\mathbf{h}}} \sum_{k_{\mathbf{n}}=1}^{K_{\mathbf{n}}} \iint_C \gamma_t(k_{\mathbf{x}}, k_{\mathbf{h}}, k_{\mathbf{n}}) \\ &\log\left[\bar{\omega}_{\mathbf{x}}(k_{\mathbf{x}})\bar{\omega}_{\mathbf{h}}(k_{\mathbf{h}})\bar{\omega}_{\mathbf{n}}(k_{\mathbf{n}})\bar{p}(\mathbf{z}^t|k_{\mathbf{x}}, k_{\mathbf{h}})\bar{p}(\mathbf{n}^t|k_{\mathbf{n}})\right] d\mathbf{Z} d\mathbf{N}\end{aligned} \tag{43}$$

where

$$\begin{aligned}\gamma_t(k_{\mathbf{x}}, k_{\mathbf{h}}, k_{\mathbf{n}}) = &\sum_{\mathbf{M_x}} \sum_{\mathbf{M_h}} \sum_{\mathbf{M_n}} \delta(k_{\mathbf{x}}, k_{\mathbf{h}}, k_{\mathbf{n}}, \mathbf{M_x}, \mathbf{M_h}, \mathbf{M_n}) \\ &p(\mathbf{Z}, \mathbf{N}, \mathbf{M_x}, \mathbf{M_h}, \mathbf{M_n}|\mathbf{\Lambda});\end{aligned} \tag{44}$$

$k_{\mathbf{x}}, k_{\mathbf{h}}$, and $k_{\mathbf{n}}$ are the component indices for $\mathbf{x}$, $\mathbf{h}$ and $\mathbf{n}$, respectively. $\mathbf{M_x}$ and $\mathbf{M_h}$ are the sequences of component indices for $\mathbf{x}$ and $\mathbf{h}$, respectively. $\delta(k_{\mathbf{x}}, k_{\mathbf{h}}, k_{\mathbf{n}}, \mathbf{M_x}, \mathbf{M_h}, \mathbf{M_n})$ is an indicator function defined as follows:

$$\begin{aligned}&\delta(k_{\mathbf{x}}, k_{\mathbf{h}}, k_{\mathbf{n}}, \mathbf{M_x}, \mathbf{M_h}, \mathbf{M_n}) \\ &= \begin{cases} 1 & \text{if} \quad m_{\mathbf{x}}^t = k_{\mathbf{x}}, m_{\mathbf{h}}^t = k_{\mathbf{h}}, m_{\mathbf{n}}^t = k_{\mathbf{n}} \\ 0 & \text{otherwise.} \end{cases}\end{aligned} \tag{45}$$

Individually maximizing $\mathcal{Q}(\bar{\mathbf{\Lambda}}|\mathbf{\Lambda})$ in Eq. (43) with respect to each of the model parameters in $\bar{\mathbf{\Lambda}}_{\mathbf{n}}$ is straightforward. Maximizing Eq. (43) with respect to $\bar{\omega}_{\mathbf{n}}(k_{\mathbf{n}})$ under the constraint $\sum_{k_{\mathbf{n}}=1}^{K_{\mathbf{n}}} \bar{\omega}_{\mathbf{n}}(k_{\mathbf{n}}) = 1$ gives

$$\begin{aligned}&\bar{\omega}_{\mathbf{n}}(k_{\mathbf{n}}) = \\ &\frac{\sum_{t=1}^{T} \sum_{k_{\mathbf{x}}=1}^{K_{\mathbf{x}}} \sum_{k_{\mathbf{h}}=1}^{K_{\mathbf{h}}} \iint_C \gamma_t(k_{\mathbf{x}}, k_{\mathbf{h}}, k_{\mathbf{n}}) d\mathbf{Z} d\mathbf{N}}{\sum_{t=1}^{T} \sum_{k_{\mathbf{x}}=1}^{K_{\mathbf{x}}} \sum_{k_{\mathbf{h}}=1}^{K_{\mathbf{h}}} \sum_{k_{\mathbf{n}}=1}^{K_{\mathbf{n}}} \iint_C \gamma_t(k_{\mathbf{x}}, k_{\mathbf{h}}, k_{\mathbf{n}}) d\mathbf{Z} d\mathbf{N}}.\end{aligned} \tag{46}$$

Meanwhile, it is easy to prove that

$$\iint_C \gamma_t(k_{\mathbf{x}}, k_{\mathbf{h}}, k_{\mathbf{n}}) d\mathbf{Z} d\mathbf{N} = p(\mathbf{Y}|\mathbf{\Lambda}) p(k_{\mathbf{x}}, k_{\mathbf{h}}, k_{\mathbf{n}}|\mathbf{y}^t, \mathbf{\Lambda}) \tag{47}$$

where $p(k_{\mathbf{x}}, k_{\mathbf{h}}, k_{\mathbf{n}}|\mathbf{y}^t, \mathbf{\Lambda})$ defined in Eq. (48) is the posterior probability of hidden random variables $k_{\mathbf{x}}$, $k_{\mathbf{h}}$ and $k_{\mathbf{n}}$. Substituting Eq. (47) into Eq. (46), the final updating formula for $\bar{\omega}_{\mathbf{n}}(k_{\mathbf{n}})$ is as follows:

$$\bar{\omega}_{\mathbf{n}}(k_{\mathbf{n}}) = \frac{1}{T} \sum_{t=1}^{T} \sum_{k_{\mathbf{x}}=1}^{K_{\mathbf{x}}} \sum_{k_{\mathbf{h}}=1}^{K_{\mathbf{h}}} p(k_{\mathbf{x}}, k_{\mathbf{h}}, k_{\mathbf{n}}|\mathbf{y}^t, \mathbf{\Lambda}). \tag{49}$$

The mean and covariance matrix of Gaussian components can be estimated similarly. By setting the following partial derivative

$$p(k_{\mathbf{x}}, k_{\mathbf{h}}, k_{\mathbf{n}}|\mathbf{y}^t, \mathbf{\Lambda}) = \frac{\omega_{\mathbf{x}}(k_{\mathbf{x}})\omega_{\mathbf{h}}(k_{\mathbf{h}})\omega_{\mathbf{n}}(k_{\mathbf{n}})p(\mathbf{y}^t|k_{\mathbf{x}}, k_{\mathbf{h}}, k_{\mathbf{n}}, \mathbf{\Lambda})}{\sum_{k_{\mathbf{x}}=1}^{K_{\mathbf{x}}}\sum_{k_{\mathbf{h}}=1}^{K_{\mathbf{h}}}\sum_{k_{\mathbf{n}}=1}^{K_{\mathbf{n}}}\omega_{\mathbf{x}}(k_{\mathbf{x}})\omega_{\mathbf{h}}(k_{\mathbf{h}})\omega_{\mathbf{n}}(k_{\mathbf{n}})p(\mathbf{y}^t|k_{\mathbf{x}}, k_{\mathbf{h}}, k_{\mathbf{n}}, \mathbf{\Lambda})} \tag{48}$$

$$\bar{\boldsymbol{\mu}}_{\mathbf{n}}(k_{\mathbf{n}}) = \frac{\sum_{t=1}^{T}\sum_{k_{\mathbf{x}}=1}^{K_{\mathbf{x}}}\sum_{k_{\mathbf{h}}=1}^{K_{\mathbf{h}}} p(k_{\mathbf{x}}, k_{\mathbf{h}}, k_{\mathbf{n}}|\mathbf{y}^t, \mathbf{\Lambda})E_{\mathbf{n}}[\mathbf{n}^t|\mathbf{y}^t, k_{\mathbf{x}}, k_{\mathbf{h}}, k_{\mathbf{n}}, \mathbf{\Lambda}]}{\sum_{t=1}^{T}\sum_{k_{\mathbf{x}}=1}^{K_{\mathbf{x}}}\sum_{k_{\mathbf{h}}=1}^{K_{\mathbf{h}}} p(k_{\mathbf{x}}, k_{\mathbf{h}}, k_{\mathbf{n}}|\mathbf{y}^t, \mathbf{\Lambda})} \tag{53}$$

$$\bar{\mathbf{\Sigma}}_{\mathbf{n}}(k_{\mathbf{n}}) = \frac{\sum_{t=1}^{T}\sum_{k_{\mathbf{x}}=1}^{K_{\mathbf{x}}}\sum_{k_{\mathbf{h}}=1}^{K_{\mathbf{h}}} p(k_{\mathbf{x}}, k_{\mathbf{h}}, k_{\mathbf{n}}|\mathbf{y}^t, \mathbf{\Lambda})E_{\mathbf{n}}[\mathbf{n}^t(\mathbf{n}^t)^\top|\mathbf{y}^t, k_{\mathbf{x}}, k_{\mathbf{h}}, k_{\mathbf{n}}, \mathbf{\Lambda}]}{\sum_{t=1}^{T}\sum_{k_{\mathbf{x}}=1}^{K_{\mathbf{x}}}\sum_{k_{\mathbf{h}}=1}^{K_{\mathbf{h}}} p(k_{\mathbf{x}}, k_{\mathbf{h}}, k_{\mathbf{n}}|\mathbf{y}^t, \mathbf{\Lambda})} - \bar{\boldsymbol{\mu}}_{\mathbf{n}}(k_{\mathbf{n}})\bar{\boldsymbol{\mu}}_{\mathbf{n}}^\top(k_{\mathbf{n}}) \tag{54}$$

$$\frac{\partial \mathcal{Q}(\bar{\mathbf{\Lambda}}|\mathbf{\Lambda})}{\partial \bar{\boldsymbol{\mu}}_{\mathbf{n}}(k_{\mathbf{n}})} = \sum_{t=1}^{T}\sum_{k_{\mathbf{x}}=1}^{K_{\mathbf{x}}}\sum_{k_{\mathbf{h}}=1}^{K_{\mathbf{h}}}\sum_{k_{\mathbf{n}}=1}^{K_{\mathbf{n}}}\iint_C \gamma_t(k_{\mathbf{x}}, k_{\mathbf{h}}, k_{\mathbf{n}})\frac{\partial \log \bar{p}(\mathbf{n}^t|k_{\mathbf{n}})}{\partial \bar{\boldsymbol{\mu}}_{\mathbf{n}}(k_{\mathbf{n}})}d\mathbf{Z}d\mathbf{N} \tag{50}$$

equal to zero with the Gaussian PDF $\bar{p}(\mathbf{n}^t|k_{\mathbf{n}})$, we have

$$\bar{\boldsymbol{\mu}}_{\mathbf{n}}(k_{\mathbf{n}}) = \frac{\sum_{t=1}^{T}\sum_{k_{\mathbf{x}}=1}^{K_{\mathbf{x}}}\sum_{k_{\mathbf{h}}=1}^{K_{\mathbf{h}}}\iint_C \gamma_t(k_{\mathbf{x}}, k_{\mathbf{h}}, k_{\mathbf{n}})\mathbf{n}^t d\mathbf{Z}d\mathbf{N}}{\sum_{t=1}^{T}\sum_{k_{\mathbf{x}}=1}^{K_{\mathbf{x}}}\sum_{k_{\mathbf{h}}=1}^{K_{\mathbf{h}}}\iint_C \gamma_t(k_{\mathbf{x}}, k_{\mathbf{h}}, k_{\mathbf{n}})d\mathbf{Z}d\mathbf{N}}. \tag{51}$$

The integral in the numerator of Eq. (51) can be reduced to

$$\iint_C \gamma_t(k_{\mathbf{x}}, k_{\mathbf{h}}, k_{\mathbf{n}})\mathbf{n}^t d\mathbf{Z}d\mathbf{N} = p(\mathbf{Y}|\mathbf{\Lambda})p(k_{\mathbf{x}}, k_{\mathbf{h}}, k_{\mathbf{n}}|\mathbf{y}^t, \mathbf{\Lambda})E_{\mathbf{n}}[\mathbf{n}^t|\mathbf{y}^t, k_{\mathbf{x}}, k_{\mathbf{h}}, k_{\mathbf{n}}, \mathbf{\Lambda}] \tag{52}$$

where $E_{\mathbf{n}}[\mathbf{n}^t|\mathbf{y}^t, k_{\mathbf{x}}, k_{\mathbf{h}}, k_{\mathbf{n}}, \mathbf{\Lambda}]$ is the conditional expectation of $\mathbf{n}^t$ given $\mathbf{y}^t$ for components $k_{\mathbf{x}}$, $k_{\mathbf{h}}$ and $k_{\mathbf{n}}$. By substituting Eq. (47) and Eq. (52) into Eq. (51), the updating formula for $\bar{\boldsymbol{\mu}}_{\mathbf{n}}(k_{\mathbf{n}})$ can be obtained as Eq. (53). By a similar procedure, the updating formula for the covariance matrix $\bar{\mathbf{\Sigma}}_{\mathbf{n}}(k_{\mathbf{n}})$ can also be obtained as Eq. (54) where $E_{\mathbf{n}}[\mathbf{n}^t(\mathbf{n}^t)^\top|\mathbf{y}^t, k_{\mathbf{x}}, k_{\mathbf{h}}, k_{\mathbf{n}}, \mathbf{\Lambda}]$ is the conditional expectation of $\mathbf{n}^t(\mathbf{n}^t)^\top$ given $\mathbf{y}^t$ for components $k_{\mathbf{x}}$, $k_{\mathbf{h}}$ and $k_{\mathbf{n}}$.

Now let's consider how to derive the updating formula for the parameters of convolutional distortion. First the updating formula for $\bar{\omega}_{\mathbf{h}}(k_{\mathbf{h}})$ can be similarly derived as:

$$\bar{\omega}_{\mathbf{h}}(k_{\mathbf{h}}) = \frac{1}{T}\sum_{t=1}^{T}\sum_{k_{\mathbf{x}}=1}^{K_{\mathbf{x}}}\sum_{k_{\mathbf{n}}=1}^{K_{\mathbf{n}}} p(k_{\mathbf{x}}, k_{\mathbf{h}}, k_{\mathbf{n}}|\mathbf{y}^t, \mathbf{\Lambda}). \tag{55}$$

Based on Eq. (7), we have

$$\bar{p}(\mathbf{z}^t|k_{\mathbf{x}}, k_{\mathbf{h}}) = \mathcal{N}(\mathbf{z}^t; \boldsymbol{\mu}_{\mathbf{x}}(k_{\mathbf{x}}) + \bar{\mathbf{h}}(k_{\mathbf{h}}), \mathbf{\Sigma}_{\mathbf{x}}(k_{\mathbf{x}})). \tag{56}$$

The partial derivative of $\mathcal{Q}(\bar{\mathbf{\Lambda}}|\mathbf{\Lambda})$ in Eq. (43) with respect to $\bar{\mathbf{h}}(k_{\mathbf{h}})$ can be written as

$$\frac{\partial \mathcal{Q}(\bar{\mathbf{\Lambda}}|\mathbf{\Lambda})}{\partial \bar{\mathbf{h}}(k_{\mathbf{h}})} = \sum_{t=1}^{T}\sum_{k_{\mathbf{x}}=1}^{K_{\mathbf{x}}}\sum_{k_{\mathbf{h}}=1}^{K_{\mathbf{h}}}\sum_{k_{\mathbf{n}}=1}^{K_{\mathbf{n}}}\iint_C \gamma_t(k_{\mathbf{x}}, k_{\mathbf{h}}, k_{\mathbf{n}})\frac{\partial \log \bar{p}(\mathbf{z}^t|k_{\mathbf{x}}, k_{\mathbf{h}})}{\partial \bar{\mathbf{h}}(k_{\mathbf{h}})}d\mathbf{Z}d\mathbf{N}. \tag{57}$$

Using Eq. (56) and setting the above expression equal to zero, the updating formula for $\bar{\mathbf{h}}(k_{\mathbf{h}})$ can be derived as

$$\bar{\mathbf{h}}(k_{\mathbf{h}}) = \left[\sum_{t=1}^{T}\sum_{k_{\mathbf{x}}=1}^{K_{\mathbf{x}}}\sum_{k_{\mathbf{n}}=1}^{K_{\mathbf{n}}} p(k_{\mathbf{x}}, k_{\mathbf{h}}, k_{\mathbf{n}}|\mathbf{y}^t, \mathbf{\Lambda})\mathbf{\Sigma}_{\mathbf{x}}^{-1}(k_{\mathbf{x}})\right]^{-1} \left[\sum_{t=1}^{T}\sum_{k_{\mathbf{x}}=1}^{K_{\mathbf{x}}}\sum_{k_{\mathbf{n}}=1}^{K_{\mathbf{n}}} p(k_{\mathbf{x}}, k_{\mathbf{h}}, k_{\mathbf{n}}|\mathbf{y}^t, \mathbf{\Lambda})\mathbf{\Sigma}_{\mathbf{x}}^{-1}(k_{\mathbf{x}}) \left(E_{\mathbf{z}}[\mathbf{z}^t|\mathbf{y}^t, k_{\mathbf{x}}, k_{\mathbf{h}}, k_{\mathbf{n}}, \mathbf{\Lambda}] - \boldsymbol{\mu}_{\mathbf{x}}(k_{\mathbf{x}})\right)\right] \tag{58}$$

where $E_{\mathbf{z}}[\mathbf{z}^t|\mathbf{y}^t, k_{\mathbf{x}}, k_{\mathbf{h}}, k_{\mathbf{n}}, \mathbf{\Lambda}]$ is the conditional expectation of $\mathbf{z}^t$ given $\mathbf{y}^t$ for components $k_{\mathbf{x}}$, $k_{\mathbf{h}}$ and $k_{\mathbf{n}}$.

Finally, Eq. (49), Eq. (53), Eq. (54), Eq. (55), and Eq. (58), are corresponding to Eq. (17) to Eq. (21) with the setting of $k_{\mathbf{x}} = m$, $k_{\mathbf{h}} = k$, $k_{\mathbf{n}} = l$.

REFERENCES

[1] A. Acero, *Acoustic and Environment Robustness in Automatic Speech Recognition*. Norwell, MA, USA: Kluwer, 1993.

[2] Aurora document AU/217/99, "Availability of Finnish SpeechDat-Car database for ETSI STQ WI008 front-end standardisation," Nokia, Nov. 1999.

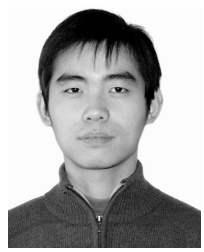
[3] "Spanish SDC-Aurora database for ETSI STQ Aurora WI008 advanced DSR front-end evaluation: Description and baseline results," UPC, Aurora document AU/271/00, Nov. 2000.

[4] "Description and baseline results for the subset of the SpeechDat-Car German database used for ETSI STQ Aurora WI008 Advanced DSR Front-end Evaluation," Texas Instruments, Aurora document AU/273/00, Dec. 2001.

[5] "Danish SpeechDat-Car digits database for ETSI STQ-Aurora advanced DSR," Aalborg Univ., Aurora document AU/378/01, Jan. 2001.

[6] C.-P. Chen and J. A. Bilmes, "MVA processing of speech features," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 1, pp. 257–270, Jan. 2007.

[7] L. Deng, J. Droppo, and A. Acero, "Enhancement of log mel power spectra of speech using a phase-sensitive model of the acoustic environment and sequential estimation of the corrupting noise," *IEEE Trans. Speech Audio Process.*, vol. 12, no. 2, pp. 133–143, Mar. 2004.
[8] G.-H. Ding and B. Xu, "Exploring high-performance speech recognition in noisy environments using high-order Taylor series expansion," in *Proc. ICSLP*, 2004, pp. 149–152.
[9] J. Du and Q. Huo, "A feature compensation approach using high-order vector Taylor series approximation of an explicit distortion model for noisy speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 8, pp. 2285–2293, Nov. 2011.
[10] J. Du and Q. Huo, "IVN-based joint training of GMM and HMMs using an improved VTS-based feature compensation for noisy speech recognition," *Proc. Interspeech*, 2012.
[11] J. Du and Q. Huo, "A VTS-based feature compensation approach to noisy speech recognition using mixture models of distortion," *Proc. ICASSP*, pp. 7078–7082, 2013.
[12] M. Fujimoto, S. Watanabe, and T. Nakatani, "Noise suppression with unsupervised joint speaker adaptation and noise mixture model estimation," in *Proc. ICASSP*, 2012, pp. 4713–4716.
[13] H. Hermansky, D. P. W. Ellis, and S. Sharma, "Tandem connectionist feature extraction for conventional HMM systems," in *Proc. ICASSP*, 2000, pp. 1635–1638.
[14] J. R. Hershey, P. A. Olsen, and S. J. Rennie, "Signal interaction and the devil function," in *Proc. INTERSPEECH*, 2010, pp. 334–337.
[15] J. R. Hershey, S. J. Rennie, and J. Le Roux, "Factorial models for noise robust speech recognition," in *Techniques for noise robustness in automatic speech recognition*, T. Virtanen, R. Singh, and B. Raj, Eds. New York, NY, USA: Wiley, 2013, pp. 311–345.
[16] G. Hinton, L. Deng, D. Yu, G. 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," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012.
[17] Y. Hu and Q. Huo, "Irrelevant variability normalization based HMM training using VTS approximation of an explicit model of environmental distortions," in *Proc. Interspeech*, 2007, pp. 1042–1045.
[18] O. Kalinli, M. L. Seltzer, and A. Acero, "Noise adaptive training using a vector Taylor series approach for robust automatic speech recognition," in *Proc. ICASSP*, 2009, pp. 3825–3828.
[19] D.-Y. Kim, C.-K. Un, and N.-S. Kim, "Speech recognition in noisy environments using first-order vector Taylor series," *Speech Commun.*, vol. 24, pp. 39–49, 1998.
[20] N. S. Kim, "Statistical linear approximation for environment compensation," *IEEE Signal Process. Lett.*, vol. 5, no. 1, pp. 8–10, Jan. 1998.
[21] J. Li, M. L. Seltzer, and Y. Gong, "Improvements to VTS feature enhancement," in *Proc. ICASSP*, 2012, pp. 4677–4680.
[22] P. J. Moreno, B. Raj, and R. M. Stern, "A vector Taylor series approach for environment-independent speech recognition," in *Proc. ICASSP*, 1996, pp. 733–736.
[23] P. J. Moreno, "Speech recognition in noisy environments," Ph.D. dissertation, Carnegie Mellon Univ., Pittsburgh, PA, 1996.
[24] D. Povey, B. Kingsbury, L. Mangu, G. Saon, H. Soltau, and G. Zweig, "fMPE: Discriminatively trained features for speech recognition," in *Proc. ICASSP*, 2005, pp. 961–964.
[25] S. Rennie, T. Kristjansson, P. Olsen, and R. Gopinath, "Dynamic noise adaptation," in *Proc. ICASSP*, 2006, pp. 1197–1200.
[26] R. C. Rose, E. M. Hofstetter, and D. A. Reynolds, "Integrated models of signal and background with application to speaker identification in noise," *IEEE Trans. Speech Audio Process.*, vol. 2, no. 2, pp. 245–257, Apr. 1994.
[27] V. Stouten, "Robust automatic speech recognition in time-varying environments," Ph.D. dissertation, Katholieke Univ. Leuven, Leuven, Belgium, 2006.
[28] E. Vincent, J. Barker, S. Watanabe, J. L. Roux, F. Nesta, and M. Matassoni, "The second CHiME speech separation and recognition challenge: Datasets, tasks and baselines," in *Proc. ICASSP*, 2013, pp. 126–130.
[29] S. Young *et al.*, *The HTK Book (for HTK v3.4)* 2006.
[30] *Speech processing, transmission and quality aspects (STQ); Distributed speech recognition; Advanced front-end feature extraction algorithm; Compression algorithms*, ETSI ES 202 050 v1.1.1 (2002-10), Oct. 2002, ETSI standard document.



Jun Du received the B.Eng. and Ph.D. degrees from the Department of Electronic Engineering and Information Science, University of Science and Technology of China (USTC), Hefei, China, in 2004 and 2009, respectively. From 2004 to 2009, he was with the iFlytek Speech Lab of USTC, where he conducted research on speech recognition. During the above period, he worked as an Intern twice for 9 months at Microsoft Research Asia (MSRA), Beijing, China, doing research on discriminative training and noise robust front-end for speech recognition, and speech enhancement. In 2007, he also worked as a Research Assistant for 6 months at the Department of Computer Science, The University of Hong Kong, doing research on robust speech recognition. From July 2009 to June 2010, he worked at iFlytek Research on speech recognition. From July 2010 to January 2013, he joined MSRA as an Associate Researcher, working on handwriting recognition, OCR, and speech recognition. Since February 2013, Dr. Du worked at National Engineering Laboratory for Speech and Language Information Processing (NEL-SLIP) of USTC.

Qiang Huo (M'95) is a Research Manager in Microsoft Research Asia (MSRA), Beijing, China. Prior to joining MSRA on August 1, 2007, he had been a faculty member at the Department of Computer Science, The University of Hong Kong since 1998, where he also did his Ph.D. research on speech recognition during 1991 to 1994. From 1995 to 1997, Dr. Huo worked at the ATR Interpreting Telecommunications Research Laboratories, Kyoto, Japan. In the past 25 years, he has been doing research and making contributions in the areas of speech recognition, handwriting recognition, OCR, gesture recognition, biometric-based user authentication, hardware design for speech and image processing. Dr. Huo received the B.Eng. degree from the University of Science and Technology of China (USTC), Hefei, China, in 1987, the M.Eng. degree from Zhejiang University, Hangzhou, China, in 1989, and the Ph.D. degree from the USTC in 1994, all in electrical engineering.