A Novel Dynamic Parameters Calculation Approach For Model Compensation

Suliang Bu, Yanmin Qian, Kai Yu

Key Lab. of Shanghai Education Commission for Intelligent Interaction and Cognitive Engineering
MOE-Microsoft Key Lab. for Intelligent Computing and Intelligent Systems
Department of Computer Science and Engineering, Shanghai Jiao Tong University
09210240096@fudan.edu.cn, {yanminqian, kai.yu}@sjtu.edu.cn

Abstract

Model compensation approach has been successfully applied to various noise robust speech recognition tasks. In this paper, based on Continuous Time (CT) approximation, the dynamic mismatch function is derived without further approximation. With such mismatch function, a novel approach to deriving the formula for calculating the dynamic statistics is presented. Besides, we also provide an insight on the processing of the pseudo inverse of non-square discrete cosine transform (DCT) matrix during model compensation. Experiments on Aurora 4 showed that the proposed approach obtained 23.2% relative WER reduction over traditional first-order Vector Taylor Series (VTS) approach.

Index Terms: robust speech recognition, model based compensation, Continuous Time approximation, Vector Taylor Series

10.21437/Interspeech.2014-159

1. Introduction

It is well known that the performance of automatic speech recognition (ASR) system degrades greatly when additive noise exists and the system is trained only with clean speech. An effective way to address this problem is to adopt the model compensation approach, which adapts the clean model to the noise. And as observed in several previous papers [1, 2], accurate dynamic parameters are the key for a good performance.

Generally speaking, there are two ways to calculate the dynamic parameters. One is to use the linear transform of the static parameters as the dynamic ones [3, 4]. The underlying justification is that the dynamic features are usually computed by a linear regression of the static ones. However this way is often computationally expensive. Thus the most common way is to use the Continuous Time (CT) approximation [5], which assumes that the dynamic features are the time derivatives of the statics. Empirically, this assumption works well.

Previously, under CT assumption, we take the derivative of the approximation of noisy speech to derive the dynamic mismatch function. Obviously, this approximation will result in some inaccuracy. Secondly, since non-square DCT matrix has no real inverse, its pseudo inverse is used during model compensation. Besides, first-order, not higher order, VTS is often used to expand static mismatch function. All of these could introduce errors during model compensation process, and consequently decrease the performance.

This work was performed while the first author was an intern at SJTU. It was supported by the Program for Professor of Special Appointment (Eastern Scholar) at Shanghai Institutions of Higher Learning, the China NSFC project No. 61222208 and JiangSu NSF project No. 201302060012 .

To deal with these above problems, in this paper, we first use second-order VTS instead of first-order to calculate static parameters to reduce residual errors. Then the dynamic mismatch function is directly derived from noisy speech without further approximation. Finally we make some modification related to the pseudo inverse of DCT matrix. As will be shown in experiment, these processes actually improve the performance.

This paper is organized as follows. In section 2, the formulation to calculate static and dynamic statistics is described. The modification related to the pseudo inverse of DCT matrix is shown in section 3. And experiment results on aurora 4 are analyzed in section 4, finally we conclude in section 5.

2. Model-based Compensation

In this study, only additive noise is considered, and the channel distortion is ignored. The formulation to calculate static and dynamic statistics will be described respectively.

2.1. Static Statistics

2.1.1. Formula to Calculate Static Statistics

In the static cepstral domain, the nonlinear effect of additive noise can be expressed as:

$$\mathbf{y_s} = \mathbf{x_s} + \mathbf{C} \cdot \ln\left(\mathbf{1} + \mathrm{e}^{\mathbf{C^{-1}(n_s - x_s)}}\right) \tag{1}$$

where $\mathbf{y_s}$, $\mathbf{x_s}$ and $\mathbf{n_s}$ are static features corresponding to noisy speech, clean speech and additive noise respectively; $\mathbf{C}$ is the truncated DCT matrix. Since it is not square, there is no real inverse for it. Therefore, its pseudo inverse matrix $\mathbf{C^{-1}}$ is usually used instead. The subscript "s" indicates static parameters. Here both $\mathbf{x_s}$ and $\mathbf{n_s}$ are assumed to have independent Gaussian distributions with mean $\boldsymbol{\mu_{xs}}$, $\boldsymbol{\mu_{ns}}$ and covariance $\mathbf{\Sigma_{xs}}$, $\mathbf{\Sigma_{ns}}$, respectively.

Next, we use $\mathbf{z_s}$ to denote $\mathbf{C^{-1}(n_s - x_s)}$. Since 1) each component of $\mathbf{x_s}$ and $\mathbf{n_s}$ is Gaussian distributed, 2) all these vector components are assumed to be mutually independent and 3) $\mathbf{C^{-1}}$ is a linear transformation, $\mathbf{z_s}$ is also Gaussian distributed. Please note that components in $\mathbf{z_s}$ are not mutually independent, nor $\mathbf{z_s}$ and $\mathbf{x_s}$ are independent.

The mean and covariance for $\mathbf{z_s}$ are given by

$$\boldsymbol{\mu_{zs}} = \mathbf{C^{-1}}(\boldsymbol{\mu_{ns}} - \boldsymbol{\mu_{xs}}) \tag{2}$$

$$\mathbf{\Sigma_{zs}} = \mathbf{C^{-1}}(\mathbf{\Sigma_{ns}} + \mathbf{\Sigma_{xs}})(\mathbf{C^{-1}})^{\mathbf{T}} \tag{3}$$

Now take expectation of the noisy speech, we would have:

$$E[\mathbf{y_s}] = E[\mathbf{x_s}] + \mathbf{C} \cdot E[\ln(\mathbf{1} + \mathrm{e}^{\mathbf{z_s}})] \tag{4}$$

The second-order vector Taylor series expansion of $\ln(\mathbf{1}+\mathrm{e}^{\mathbf{z_s}})$ around $\boldsymbol{\mu_{zs}}$ can be written as

$$\begin{aligned}\ln(\mathbf{1}+\mathrm{e}^{\mathbf{z_s}}) \approx\ & \mathbf{f}^{(0)}+\mathbf{f}^{(1)}\odot(\mathbf{z_s}-\boldsymbol{\mu}_{\mathrm{zs}}) \\ & +\frac{1}{2}\mathbf{f}^{(2)}\odot(\mathbf{z_s}-\boldsymbol{\mu_{zs}})\odot(\mathbf{z_s}-\boldsymbol{\mu_{zs}})\end{aligned} \tag{5}$$

where

$$\mathbf{f}^{(0)}=\ln(\mathbf{1}+\mathrm{e}^{\boldsymbol{\mu_{zs}}}) \tag{6}$$

$$\mathbf{f}^{(1)}=\mathbf{1}-(\mathbf{1}+e^{\boldsymbol{\mu_{zs}}})^{-1} \tag{7}$$

$$\mathbf{f}^{(2)}=(\mathbf{1}+e^{\boldsymbol{\mu_{zs}}})^{-2}\odot e^{\boldsymbol{\mu_{zs}}} \tag{8}$$

$\mathbf{f}^{(1)}$ and $\mathbf{f}^{(2)}$ correspond to the first and second derivatives of $\ln(\mathbf{1}+\mathrm{e}^{\mathbf{z_s}})$ at $\mathbf{z_s}=\boldsymbol{\mu_{zs}}$ and $\odot$ denotes element-by-element multiplication.

Compute the expectation of Eq. (5), we could have

$$E\left[\ln\left(\mathbf{1}+\mathbf{e}^{\mathbf{z_s}}\right)\right]\approx\mathbf{f}^{(0)}+\frac{1}{2}\mathbf{f}^{(2)}\odot diag^{-1}(\boldsymbol{\Sigma}_{\mathbf{zs}}) \tag{9}$$

where $diag^{-1}(\cdot)$ denotes the operation of extracting the diagonal elements of a matrix as a column vector.

Then together with Eq.(4) and (9), the static mean of the noisy speech is given by

$$\boldsymbol{\mu_{ys}}\approx\boldsymbol{\mu_{xs}}+\mathbf{C}\cdot\left\{\mathbf{f}^{(0)}+\frac{1}{2}\left[\mathbf{f}^{(2)}\odot diag^{-1}(\boldsymbol{\Sigma}_{\mathbf{zs}})\right]\right\} \tag{10}$$

As for the static covariance of noisy speech, it could be calculated by

$$\begin{aligned}\boldsymbol{\Sigma}_{\mathbf{ys}}\approx E(\mathbf{y_s}-\boldsymbol{\mu_{ys}})(\mathbf{y_s}-\boldsymbol{\mu_{ys}})^T=\boldsymbol{\Sigma}_{\mathbf{xs}}-\mathbf{K_1} \\ -\mathbf{K_1}^T+\mathbf{C}\cdot[\boldsymbol{\Sigma}_{\mathbf{zs}}\odot\mathbf{F_1}]\cdot\mathbf{C}^T+\mathbf{K_2}\end{aligned} \tag{11}$$

where

$$\mathbf{K_1}=\boldsymbol{\Sigma}_{\mathbf{xs}}(\mathbf{C}^{-1})^T\cdot diag(\mathbf{f}^{(1)})\cdot\mathbf{C}^T \tag{12}$$

$$\mathbf{K_2}=\frac{1}{2}\mathbf{C}\cdot[\boldsymbol{\Sigma}_{\mathbf{zs}}\odot\boldsymbol{\Sigma}_{\mathbf{zs}}\odot\mathbf{F_2}]\cdot\mathbf{C}^T \tag{13}$$

$$\mathbf{F_1}=\mathbf{f}^{(1)}\left(\mathbf{f}^{(1)}\right)^T \tag{14}$$

$$\mathbf{F_2}=\mathbf{f}^{(2)}\left(\mathbf{f}^{(2)}\right)^T \tag{15}$$

In Eq. (12), $diag(\cdot)$ denotes the operation of generating diagonal matrix from a column vector.

2.2. Dynamic Statistics

2.2.1. Theoretical Analysis

In the following formula, the subscript "Δ" is used to denote delta statistics, and "$\Delta\Delta$" to denote delta delta ones.

To compute dynamic mean and covariance matrix, here CT approximation [5] was used. Previously, people take the derivative of the approximation of $\mathbf{y_s}$ with respect to time to derive the dynamic mismatch function. For example, [6, 7, 8] uses VTS approximation. The basic reason might be that such approximation could simplify subsequent calculation. However, this inevitably causes some errors. Therefore, here we directly take derivative of $\mathbf{y_s}$ with respect to time:

$$\frac{\partial\mathbf{y_s}}{\partial t}=\frac{\partial\mathbf{x_s}}{\partial t}+\mathbf{C}\cdot\left(\frac{\mathbf{e}^{\mathbf{z_s}}}{\mathbf{1}+\mathbf{e}^{\mathbf{z_s}}}\odot\frac{\partial\mathbf{z_s}}{\partial\mathbf{t}}\right) \tag{16}$$

For $\frac{\mathbf{e}^{\mathbf{z_s}}}{\mathbf{1}+\mathbf{e}^{\mathbf{z_s}}}$, its i'th element is given by $\frac{\mathbf{e}^{(\mathbf{z_s})_\mathbf{i}}}{\mathbf{1}+\mathbf{e}^{(\mathbf{z_s})_\mathbf{i}}}$. Similarly for some following vector variables. For the sake of convenience, let's first define some notations:

$$\mathbf{L_1}=E\left[\frac{\mathbf{e}^{\mathbf{z_s}}}{\mathbf{1}+\mathbf{e}^{\mathbf{z_s}}}\right]=E\left[\frac{\mathbf{1}}{\mathbf{1}+\mathbf{e}^{-\mathbf{z_s}}}\right] \tag{17}$$

$$\mathbf{L_2}=E\left[\frac{\mathbf{e}^{\mathbf{z_s}}}{(\mathbf{1}+\mathbf{e}^{\mathbf{z_s}})^{\mathbf{2}}}\right]=E\left[\frac{\mathbf{1}}{\mathbf{e}^{\mathbf{z_s}}+\mathbf{2}+\mathbf{e}^{-\mathbf{z_s}}}\right] \tag{18}$$

$$\mathbf{L_{11}}=E\left[\frac{\mathbf{e}^{\mathbf{z_s}}}{\mathbf{1}+\mathbf{e}^{\mathbf{z_s}}}\right]\left[\frac{\mathbf{e}^{\mathbf{z_s}}}{\mathbf{1}+\mathbf{e}^{\mathbf{z_s}}}\right]^T \tag{19}$$

$$\mathbf{L_{22}}=E\left[\frac{\mathbf{e}^{\mathbf{z_s}}}{(\mathbf{1}+\mathbf{e}^{\mathbf{z_s}})^{\mathbf{2}}}\right]\left[\frac{\mathbf{e}^{\mathbf{z_s}}}{(\mathbf{1}+\mathbf{e}^{\mathbf{z_s}})^{\mathbf{2}}}\right]^T \tag{20}$$

$$\mathbf{L_{21}}=E\left[\frac{\mathbf{e}^{\mathbf{z_s}}}{(\mathbf{1}+\mathbf{e}^{\mathbf{z_s}})^{\mathbf{2}}}\right]\left[\frac{\mathbf{e}^{\mathbf{z_s}}}{\mathbf{1}+\mathbf{e}^{\mathbf{z_s}}}\right]^T \tag{21}$$

Here we assume $\frac{\partial\mathbf{n_s}}{\partial t}$ and $\mathbf{n_s}$ are independent, similarly for speech. Since $\mathbf{C^{-1}}$ is a linear transformation, it is easy to prove $\frac{\partial\mathbf{z_s}}{\partial t}$ and $\mathbf{z_s}$ are independent. Then take expectation of Eq. (16), we would have:

$$\begin{aligned}\boldsymbol{\mu_{y\Delta}}&=\boldsymbol{\mu_{x\Delta}}+\mathbf{C}\cdot\left(E\left[\frac{\mathbf{e}^{\mathbf{z_s}}}{\mathbf{1}+\mathbf{e}^{\mathbf{z_s}}}\right]\odot E\left[\frac{\partial\mathbf{z_s}}{\partial t}\right]\right) \\ &=\boldsymbol{\mu_{x\Delta}}+\mathbf{C}\cdot(\mathbf{L_1}\odot\boldsymbol{\mu_{z\Delta}})\end{aligned} \tag{22}$$

As for delta covariance, it could be computed by:

$$\begin{aligned}\boldsymbol{\Sigma}_{\mathbf{y\Delta}}=\boldsymbol{\Sigma}_{\mathbf{x\Delta}}-\mathbf{K_3}-\mathbf{K_3}^T+\mathbf{C}\cdot[\boldsymbol{\Sigma}_{\mathbf{z\Delta}}\odot\mathbf{L_{11}}]\cdot\mathbf{C}^T \\ +\mathbf{C}\cdot\left[\left(\mathbf{L_{11}}-\mathbf{L_1}(\mathbf{L_1})^T\right)\odot\left(\boldsymbol{\mu_{z\Delta}}(\boldsymbol{\mu_{z\Delta}})^T\right)\right]\cdot\mathbf{C}^T\end{aligned} \tag{23}$$

where $\boldsymbol{\mu_{z\Delta}}$, $\boldsymbol{\Sigma_{z\Delta}}$ could be computed like Eq. (2), (3), and

$$\mathbf{K_3}=\boldsymbol{\Sigma}_{\mathbf{x\Delta}}(\mathbf{C}^{-1})^{\mathbf{T}}\cdot diag\left(\mathbf{L_1}\right)\cdot\mathbf{C}^T \tag{24}$$

Similarly, take derivative of $\frac{\partial\mathbf{y_s}}{\partial t}$, the delta delta mismatch function could be computed as:

$$\begin{aligned}\frac{\partial^2\mathbf{y_s}}{\partial t^2}=\frac{\partial^2\mathbf{x_s}}{\partial t^2}+\mathbf{C}\left[\frac{\mathbf{e}^{\mathbf{z_s}}}{(\mathbf{1}+\mathbf{e}^{\mathbf{z_s}})^{\mathbf{2}}}\odot\frac{\partial\mathbf{z_s}}{\partial\mathbf{t}}\odot\frac{\partial\mathbf{z_s}}{\partial\mathbf{t}}\right] \\ +\mathbf{C}\left(\frac{\mathbf{e}^{\mathbf{z_s}}}{\mathbf{1}+\mathbf{e}^{\mathbf{z_s}}}\odot\frac{\partial^{\mathbf{2}}\mathbf{z_s}}{\partial\mathbf{t^2}}\right)\end{aligned} \tag{25}$$

As in previous steps, the delta delta statistics can be computed by:

$$\boldsymbol{\mu_{y\Delta\Delta}}=\boldsymbol{\mu_{x\Delta\Delta}}+\mathbf{C}\cdot(\mathbf{L_1}\odot\boldsymbol{\mu_{z\Delta\Delta}}+\mathbf{L_2}\odot\mathbf{K_4}) \tag{26}$$

$$\begin{aligned}\boldsymbol{\Sigma}_{\mathbf{y\Delta\Delta}}=\boldsymbol{\Sigma}_{\mathbf{x\Delta\Delta}}-\mathbf{K_5}-\mathbf{K_5}^T+\mathbf{C}\left(\mathbf{K_6}+\mathbf{K_7}+\mathbf{K_7}^T\right)\mathbf{C}^T \\ +\mathbf{C}\left[\boldsymbol{\Sigma}_{\mathbf{z\Delta\Delta}}\odot\mathbf{L_{11}}+\mathbf{K_8}\right]\mathbf{C}^T\end{aligned} \tag{27}$$

where

$$\mathbf{K_4}=diag^{-1}(\boldsymbol{\Sigma}_{\mathbf{z\Delta}})+\boldsymbol{\mu_{z\Delta}}\odot\boldsymbol{\mu_{z\Delta}} \tag{28}$$

$$\mathbf{K_5}=\boldsymbol{\Sigma}_{\mathbf{x\Delta\Delta}}\left(\mathbf{C}^{-\mathbf{1}}\right)^{\mathbf{T}}\cdot diag\left(\mathbf{L_1}\right)\cdot\mathbf{C}^T \tag{29}$$

$$\begin{aligned}\mathbf{K_6}=\boldsymbol{\Sigma}_{\mathbf{z\Delta}}\odot\left(2\boldsymbol{\Sigma}_{\mathbf{z\Delta}}+4\boldsymbol{\mu}_{\mathbf{z\Delta}}(\boldsymbol{\mu}_{\mathbf{z\Delta}})^T\right)\odot\mathbf{L_{22}} \\ +\left[\mathbf{L_{22}}-\mathbf{L_2}(\mathbf{L_2})^T\right]\odot\left[\mathbf{K_4}(\mathbf{K_4})^T\right]\end{aligned} \tag{30}$$

$$\mathbf{K_7}=\left[\mathbf{L_{21}}-\mathbf{L_2}(\mathbf{L_1})^T\right]\odot\left[\mathbf{K_4}(\boldsymbol{\mu_{z\Delta\Delta}})^T\right] \tag{31}$$

$$\mathbf{K_8}=\left[\mathbf{L_{11}}-\mathbf{L_1}(\mathbf{L_1})^T\right]\odot\left[\boldsymbol{\mu_{z\Delta\Delta}}(\boldsymbol{\mu_{z\Delta\Delta}})^T\right] \tag{32}$$

2.2.2. Formula to Calculate Dynamic Statistics

To calculate the dynamic statistics, we need to know the value of $\mathbf{L_1}$, $\mathbf{L_2}$, $\mathbf{L_{11}}$, $\mathbf{L_{21}}$, $\mathbf{L_{22}}$. However, none of them could be easily computed. Therefore, making some approximation is necessary. One way is to use VTS to expand each corresponding term and then calculate the resultant expectations. However there is a problem with this idea: Even if assume the VTS way could get more accurate value, it has a big disadvantage, that is it would result in very complicated formula, especially for covariance matrices. This not only makes the implementation complex, but also increases the computational cost. To avoid these problems, we make the following approximation [1]:

$$\mathbf{L_1} \approx \frac{\mathbf{1}}{\mathbf{E}\left[\mathbf{1}+\mathbf{e}^{-\mathbf{z_s}}\right]} \tag{33}$$

$$\mathbf{L_2} \approx \frac{\mathbf{1}}{\mathbf{E}\left[\mathbf{e}^{\mathbf{z_s}}+\mathbf{2}+\mathbf{e}^{-\mathbf{z_s}}\right]} \tag{34}$$

$$\mathbf{L_{11}} \approx \mathbf{L_1}\left(\mathbf{L_1}\right)^T \tag{35}$$

$$\mathbf{L_{22}} \approx \mathbf{L_2}\left(\mathbf{L_2}\right)^T \tag{36}$$

$$\mathbf{L_{21}} \approx \mathbf{L_2}\left(\mathbf{L_1}\right)^T \tag{37}$$

For a Gaussian variable x with mean μ and variance Σ, we have:

$$E\left[e^x\right] = exp(\mu + \Sigma/2) \tag{38}$$

With Eq. (38), it is easy to derive the formula for $\mathbf{L_1}$ and $\mathbf{L_2}$. Then with all these above, the formula to calculate the dynamic statistics are available. Particularly, covariance is simplified:

$$\mathbf{\Sigma_{y\Delta}} \approx \mathbf{\Sigma_{x\Delta}} - \mathbf{K_3} - \mathbf{K_3}^T + \mathbf{C}\left[\mathbf{\Sigma_{z\Delta}} \odot \mathbf{L_{11}}\right]\mathbf{C}^T \tag{39}$$

$$\begin{aligned}\mathbf{\Sigma_{y\Delta\Delta}} \approx & \mathbf{\Sigma_{x\Delta\Delta}} - \mathbf{K_5} - \mathbf{K_5}^T + \mathbf{C}\left[\mathbf{\Sigma_{z\Delta\Delta}} \odot \mathbf{L_{11}}\right]\mathbf{C}^T \\ & + \mathbf{C}\left[\mathbf{\Sigma_{z\Delta}} \odot \left(2\mathbf{\Sigma_{z\Delta}} + 4\boldsymbol{\mu}_{\mathbf{z\Delta}}\left(\boldsymbol{\mu}_{\mathbf{z\Delta}}\right)^T\right) \odot \mathbf{L_{22}}\right]\mathbf{C}^T\end{aligned} \tag{40}$$

Considering that the simple approximations of Eq. (35) - (37) might result in inaccurate dynamic covariance, alternatively we use second-order VTS [2] to derive the formula for them.

To do model compensation, noise parameters are needed. In this paper, noise is modeled by a single Gaussian. Usually, the noise parameters are iteratively estimated using EM-like algorithms. However, when high order VTS is used, it is not easy to directly derive the formula for re-estimation even for the mean. For comparison, in this study noise parameters are estimated by the first and last 20 frames of each test utterance.

In practice, when noise is relatively stationary, $\boldsymbol{\mu}_{\boldsymbol{n\Delta}}$ and $\boldsymbol{\mu}_{\boldsymbol{n\Delta\Delta}}$ are set to be zero, and covariance matrices are usually diagonalized for computational convenience.

3. About the Pseudo Inverse

In this section, we deal with the pseudo inverse of the truncated DCT matrix. The basic motivation here is to see how the pseudo inverse influences the mean and covariance, specifically those $\boldsymbol{\mu}_{\boldsymbol{z}}$ and $\mathbf{\Sigma_z}$, which are directly related to the pseudo inverse (see Eq. (2) and (3)). After all, if square DCT matrix is really used, there is no need to care about them.

In experiment, we compared $\mathbf{C_r^{-1}(cov)(C_r^{-1})}^T$ with $\mathbf{C^{-1}(cov_t)(C^{-1})}^T$, where $\mathbf{C_r^{-1}}$ is the real inverse of square DCT matrix, $\mathbf{cov}$ is a diagonal matrix, used to simulate the covariance matrix and $\mathbf{C^{-1}}$ is the pseudo inverse of a truncated DCT matrix and $\mathbf{cov_t}$ is the corresponding truncated matrix. The size of the $\mathbf{C_r}$ could be less than 10*10 for observation convenience. In experiment, we find $\mathbf{C^{-1}(cov_t)(C^{-1})}^T$ seems to spread some of its main diagonal value among its two neighbors, that is the first diagonal above the main diagonal, and the first one below that. This observation inspires us to do the following process about those "$\mathbf{z}$" covariance matrices:

1, Calculate original covariance matrix according to Eq. (3)
2, Calculate the new tridiagonal as following:

$$\begin{aligned}new\ diag(\mathbf{\Sigma_{zs}}) = & [diag(\mathbf{\Sigma_{zs}},\ \mathbf{1}) + diag(\mathbf{\Sigma_{zs}}, -\mathbf{1})] \\ & * \alpha + diag(\mathbf{\Sigma_{zs}})\end{aligned} \tag{41}$$

$$new\ diag(\mathbf{\Sigma_{zs}},\ \mathbf{1}) = diag(\mathbf{\Sigma_{zs}},\ \mathbf{1}) * (1-\alpha) \tag{42}$$

$$new\ diag(\mathbf{\Sigma_{zs}}, -\mathbf{1}) = diag(\mathbf{\Sigma_{zs}}, -\mathbf{1}) * (1-\alpha) \tag{43}$$

3, Replace the original tridiagonal with the new one, and keep the other entries in the covariance matrix unchanged.

In the above three equations, $diag(\mathbf{\Sigma_{zs}})$, $diag(\mathbf{\Sigma_{zs}}, 1)$ and $diag(\mathbf{\Sigma_{zs}}, -1)$ denote the main diagonal, the first diagonal above the main diagonal, and the first one below that, respectively. And α is a value between 0 and 1.

The basic objective of Eq. (41) is to assign some value of $diag(\mathbf{\Sigma_{zs}}, 1)$ and $diag(\mathbf{\Sigma_{zs}}, -1)$ to the main diagonal. And Eq. (41) is actually row-oriented. It operates like this: For each row, the element in the main diagonal is added by its row neighbors. As for the first and last main diagonal element, only its right or left neighbor is considered.

Another interesting finding about $\mathbf{C_r^{-1}(cov)(C_r^{-1})}^T$ and $\mathbf{C^{-1}(cov_t)(C^{-1})}^T$ is that the sum of each row is the same within the same matrix, though the sums are different for the two matrices. In order to be in accordance with this finding, Eq. (42) and (43) are used to make sure the sum of each row is the same as before.

A final important thing is to make sure the new main diagonal is positive since each element in the main diagonal is actually variance. Empirically, we find if some reserved value for α, such as 0.2 is used, the main diagonal is always positive.

Similarly, $\mathbf{\Sigma_{z\Delta}}$, $\mathbf{\Sigma_{z\Delta\Delta}}$ could also be computed. However, as will be shown below, the joint modification seems does not improve the performance.

So far, these above are related to the covariance. As for the mean, we did not observe anything particular.

4. Experiments and Results

4.1. Experiment Setup

In order to test the effect of the proposed approaches, we conducted a series of experiments on Aurora 4, which is based on the Wall Street Journal 5k database. In this study, speech models were trained on clean data, which comprises 7138 utterances. And decision tree state clustering was used to get about 3000 tied triphone states.

Since this paper only considers additive noise, the experiments were conducted on the test set B of Aurora 4 corpus, which was recorded using the same microphone as the training data did. Therefore, channel distortion could be omitted. Six different noises at various SNRs were artificially added to turn original clean data into the noisy database. Each noise condition has 330 test utterances from 8 speakers. And only 16kHz testing data were used for evaluation.

[1] For $\mathbf{L_1}$ and $\mathbf{L_2}$, we actually could calculate their upper bounds or lower bounds, and use either bound as an approximation of real expectation. However, they are not necessarily more accurate than the simple approximation as described above.

We used 12 Mel Frequency Cepstral Coefficient (MFCC) and C0 as well as the delta and delta-delta features. HTK [9] software was used to build the system, in which bigram language model was adopted. Each speech state was represented by 16 Gaussian components while 32 Gaussian components were used for the silence state model.

4.2. Experiment Results

In the following experiments, we first construct the GMM-HMM baseline, which results are illustrated in Table 1.

clean	car	babble	rest.	street	airport	train	avg
6.8	37.0	55.1	55.0	64.7	48.8	63.9	54.1

Table 1: WER (%) of the baseline system on test B of AURORA 4 using HMMs trained on clean speech.

Then we compare the following four approaches: 1, 2) both static and dynamic statistics are computed using first-order VTS [7] or second-order VTS [2], denoted as fVTS and sVTS respectively; 3) the first approach proposed in section 2.2.2, denoted as "SIMP" because of its simplicity; 4) the alternative one proposed in section 2.2.2, which is based on second-order VTS to calculate dynamic covariance. It is denoted as "MIX".

Please note that since it is difficult to re-estimate the noise parameters for high order VTS as said in previous section, here we only adopt simple noise estimation by using the first and last 20 frames. More accurate noise parameters estimation for second-order VTS will be in our future work.

	car	babble	rest.	street	airport	train	avg
fVTS	14.9	21.3	28.3	24.4	22.0	25.6	22.8
sVTS	11.5	18.7	23.7	20.5	18.3	21.6	19.0
MIX	**11.0**	**17.7**	**22.5**	**19.4**	**17.3**	**20.5**	**18.1**
SIMP	11.5	18.0	22.7	19.9	17.9	20.8	18.5

Table 2: WER (%) of four methods on test B

Compare fVTS with sVTS, we can see higher order VTS indeed improves the performance. The probable reason is higher order VTS could reduce the residual errors. At the same time, considering the only difference between sVTS and "MIX" is that the latter uses the simple approximation to calculate the dynamic means, the relatively large improvement over sVTS shows the necessity of accurate dynamic mismatch function. On the other hand, though "SIMP" is simple, it is still better than pure VTS-based approaches.

Next, we will start to consider the α factor, which has been explained in section 3. In the following Table 3 and 4, α will be assigned 0.1 and 0.2 respectively to calculate new $\mathbf{\Sigma_{zs}}$ [1].

Compare Table 3, 4 with Table 2, we can see that the introduction of α indeed improves the average performance, but for some specific conditions, the performance might decrease. On the other hand, it seems second-order VTS is less subject to the α factor than first-order. One possible reason is that sVTS is close to the performance limit of such pure VTS-based model compensation approaches. This explanation could be tested if higher-order VTS method is available. In other words, it is the VTS that hinders the improvement. In this case, more accurate values do not necessarily help.

[1]When α factor is considered, the fVTS formula will be different from [7] since we need the formula which explicitly include $\mathbf{\Sigma_{zs}}$. Actually the new formula based on first-order VTS can be easily inferred from section 2 and section 3 in [2].

	car	babble	rest.	street	airport	train	avg
fVTS	14.4	20.8	26.7	23.4	21.1	24.9	21.9
sVTS	11.3	18.4	23.2	20.3	17.7	22.0	18.8
MIX	**11.1**	**17.2**	**21.6**	**19.1**	**16.5**	20.7	**17.7**
SIMP	11.3	17.7	21.6	19.3	16.5	**20.4**	17.8

Table 3: WER (%) of four methods on test B: $\alpha = 0.1 : \mathbf{\Sigma_{zs}}$

	car	babble	rest.	street	airport	train	avg
fVTS	14.3	20.4	26.3	23.2	21.0	24.4	21.6
sVTS	11.2	18.0	23.4	20.2	17.2	22.1	18.7
MIX	**10.8**	**16.2**	22.2	**19.3**	16.2	21.0	17.6
SIMP	11.0	17.1	**21.2**	19.4	**15.9**	**20.2**	**17.5**

Table 4: WER (%) of four methods on test B: $\alpha = 0.2 : \mathbf{\Sigma_{zs}}$

In [10], the WER of their fVTS system on test B is 22.4%, then after one iteration of noise parameters re-estimation, the WER reduced to 15.8%. In Table 4, even without noise re-estimation, "SIMP" has reached 17.5% WER. Compared to 22.8% in Table 1, "SIMP" obtained 23.2% relative WER reduction over traditional fVTS approach.

Now it is natural to wonder whether modification to the dynamic covariance is also helpful for the performance. In next experiment, we set $\alpha = 0.1$ to calculate both $\mathbf{\Sigma_{zs}}$ and $\mathbf{\Sigma_{z\Delta}}$.

	car	babble	rest.	street	airport	train	avg
fVTS	14.5	20.7	26.1	23.0	20.9	24.7	21.7
sVTS	11.2	18.4	23.4	20.3	17.9	22.0	18.9
MIX	**10.9**	**17.2**	22.2	19.3	16.5	20.9	17.8
SIMP	11.3	17.4	**21.6**	**19.1**	**16.4**	**20.3**	**17.7**

Table 5: WER (%) of four methods on B: $\alpha = 0.1 : \mathbf{\Sigma_{zs}}, \mathbf{\Sigma_{z\Delta}}$

Compare Table 3 with Table 5, we find that modification to both covariance almost does not improve the performance. One possible explanation is that the CT approximation is not accurate enough so that more accurate parameters do not necessarily lead to better results. Another possible explanation is that the same modification to $\mathbf{\Sigma_{z\Delta}}$ is not accurate or reasonable. After all, the value of the delta covariance nearly has an order of magnitude difference compared to the value of the static one.

5. Conclusion

Based on CT approximation, this paper derives the dynamic mismatch function without further approximation. We then present the theoretical analysis on dynamic parameters calculation. Considering the difficulty to calculate $\mathbf{L_1}$, $\mathbf{L_2}$, etc., we make some expedient approximations. With these simplifications, the resulting formula is very neat. At the same time, it could be reasonable to expect the performance could improve even further if we know how to accurately calculate $\mathbf{L_1}$, $\mathbf{L_2}$, etc.. This could be one research direction.

This study also provides an insight on how to process the pseudo inverse of the DCT matrix. Experiment results show that the proposed ad hoc method could effectively improve the performance. Compared to traditional fVTS, "SIMP" with $\alpha = 0.2$ achieved 23.2% relative WER reduction.

6. References

[1] Jinyu Li, Li Deng, Dong Yu, Yifan Gong, and Alex Acero, "High-performance HMM adaptation with joint compensation of additive and convolutive distortions via vector taylor series," in *IEEE Workshop on Automatic Speech Recognition & Understanding*, 2007.

[2] Suliang Bu, Yanmin Qian, Khe Chai Sim, Yongbin You, and Kai Yu, "Second order vector Taylor series based robust speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2014.

[3] Rogier C van Dalen and Mark JF Gales, "Extended VTS for noise-robust speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, 2011.

[4] Khe Chai Sim and Minh-Thang Luong, "A trajectory-based parallel model combination with a unified static and dynamic parameter compensation for noisy speech recognition," in *IEEE Workshop on Automatic Speech Recognition and Understanding*, 2011.

[5] P. S. Gopalakrishnan S. Balakrishnan-Aiyer R. A. Gopinath, M. J. F. Gales and M. A. Picheny, "Robust speech recognition in noise - performance of the IBM continuous speech recognizer on the ARPA noise spoke task," in *ARPA Workshop on Spoken Language System Technology*, 1995.

[6] Pedro J Moreno, Bhiksha Raj, and Richard M Stern, "A vector Taylor series approach for environment-independent speech recognition," in *IEEE International Conference onAcoustics, Speech, and Signal Processing*, 1996.

[7] Alex Acero, Li Deng, Trausti T Kristjansson, and Jerry Zhang, "HMM adaptation using vector Taylor series for noisy speech recognition," in *INTERSPEECH*, 2000.

[8] Ozlem Kalinli, Michael L Seltzer, and Alex Acero, "Noise adaptive training using a vector Taylor series approach for noise robust automatic speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2009.

[9] Steve Young, Gunnar Evermann, Dan Kershaw, Gareth Moore, Julian Odell, Dave Ollason, Valtcho Valtchev, and Phil Woodland, "The htk book," *Cambridge University Engineering Department*, 2002.

[10] Liang Lu, KK Chin, Arnab Ghoshal, and Steve Renals, "Joint uncertainty decoding for noise robust subspace Gaussian mixture models," *IEEE Transactions on Audio, Speech and Language Processing*, 2012.