APPROXIMATED PARALLEL MODEL COMBINATION FOR EFFICIENT NOISE-ROBUST SPEECH RECOGNITION

Khe Chai SIM

School of Computing, National University of Singapore
Computing 1, 13 Computing Drive, 117417 Singapore

ABSTRACT

Parallel Model Combination (PMC) and Vector Taylor Series (VTS) are two model-based approaches for noise-robust speech recognition. The latter is more popular because of its simple compensation formulae for both the static and dynamic parameters. Furthermore, this VTS compensation formulation can be easily extended to noise adaptive training where the parameters of the underlying pseudo-clean speech and distortion models can be optimized. PMC lacks the above benefits because of its nonlinear variance compensation formula. In this paper, the Approximated PMC (APMC) method is proposed where linearized PMC variance compensation is used. The same approximation has also been applied to Trajectory-based APMC (TAPMC) to achieve a four-time computational saving over the Trajectory-based PMC (TPMC). The dynamic parameter compensation and noise re-estimation formulae for APMC are also derived. Experimental results on AURORA 4 show that APMC and TAPMC consistently outperformed the standard VTS and Trajectory-based VTS (TVTS) by 6.3% and 5.3% relative respectively.

Index Terms— Noise robust speech recognition, vector Taylor series, parallel model combination, trajectory-based compensation

1. INTRODUCTION

Most state-of-the-art speech recognition systems use the Hidden Markov Model (HMM) [1] to represent phonemes. The observation distribution for each HMM state is represented by a Gaussian Mixture Model (GMM) [2]. Parallel Model Combination (PMC) [3, 4] and Vector Taylor Series (VTS) [5] are two well-known model-based approaches for improving the noise robustness of HMM-based speech recognition. Using the first order Taylor series approximation, the VTS compensation formulae for both the static and dynamic parameters are much simpler and computationally more efficient compared to PMC. In particular, the cost of performing trajectory-based compensation [6] using PMC is significantly higher due to the higher dimensional cepstral trajectory statistics. Furthermore, the simple VTS formulation allows the underlying pseudo-clean speech models and distortion parameters to be updated using maximum likelihood estimation. This is essential for Noise Adaptive Training (NAT) [7].

One of the major drawbacks of PMC is its complex nonlinear variance compensation formulae. Such complexity leads to a much higher computational cost and complicates the compensation of the dynamic parameters. In this paper, the Approximated PMC (APMC) method is proposed to mitigate this problem by simplifying the variance compensation formulae of PMC through linearization of the $\exp$ and $\log$ functions. Incidentally, the resulting variance compensation is very similar to that of VTS. Therefore, APMC and be viewed as a hybrid between PMC and VTS. With the simplified variance compensation, the dynamic parameter compensation formulae as well as the noise mean update formulae for APMC can also be derived.

The remainder of this paper is organized as follows. Section 2 compares VTS and PMC in terms of the compensation formulae and the computational costs. Section 3 describes the proposed Approximated PMC (APMC) technique. Section 4 presents the re-estimation formulae for the noise mean using APMC. Section 5 presents the experimental results on the AURORA 4 dataset.

2. VTS VERSUS PMC

Parallel Model Combination (PMC) [3, 4] and Vector Taylor Series (VTS) [5] are two well-known model-based noise compensation techniques. The former uses a *log normal* approximation to convert the cepstral statistics into linear spectral statistics so that noisy speech statistics can be easily derived from clean speech and noise statistics. VTS, on the other hand, uses the Taylor series expansion to approximate the relationship between the noisy speech data and the clean speech data so that simple compensation formulae can be derived. VTS can also be used to easily compensate the dynamic parameters using continuous time approximation [5]. Although continuous time approximation can also be applied to PMC [3], the resulting compensation formulae are quite complex. Trajectory PMC (TPMC) [6] and Extended VTS [8] have recently been proposed to mitigate the complex dynamic compensation problem by representing the

observation (static and dynamic) statistics as trajectory (extended) statistics. These approaches, which avoid the need to deal with dynamic parameter compensation directly, have been shown to outperform standard PMC and VTS compensations. Although the PMC and VTS methods are based on quite different formulations, their compensation formulae are somewhat similar. In the following, a closer comparison between PMC and VTS is presented.

2.1. Vector Taylor Series (VTS)

The VTS formulation is based on the approximation of the relationship between the noisy speech data and the clean speech data using vector Taylor series expansion [5]. This leads to the following VTS compensation formulae:

$$\hat{\boldsymbol{\mu}}_m^{(\mathtt{c})} = \boldsymbol{\mu}_m^{(\mathtt{c})} + \boldsymbol{\mu}_h^{(\mathtt{c})} + g\left(\bar{\boldsymbol{\mu}}_{mhn}^{(\mathtt{c})}\right) \quad (1a)$$

$$\hat{\boldsymbol{\Sigma}}_m^{(\mathtt{c})} = \boldsymbol{G}_m \boldsymbol{\Sigma}_m^{(\mathtt{c})} \boldsymbol{G}_m{}^\top + \boldsymbol{F}_m \boldsymbol{\Sigma}_n^{(\mathtt{c})} \boldsymbol{F}_m{}^\top \quad (1b)$$

where $\bar{\boldsymbol{\mu}}_{mhn}^{(\mathtt{c})} = \boldsymbol{\mu}_n^{(\mathtt{c})} - \boldsymbol{\mu}_m^{(\mathtt{c})} - \boldsymbol{\mu}_h^{(\mathtt{c})}$. $\boldsymbol{\mu}_m^{(\mathtt{c})}$, $\boldsymbol{\mu}_h^{(\mathtt{c})}$ and $\boldsymbol{\mu}_n^{(\mathtt{c})}$ are the mean vectors for clean speech, channel and noise respectively. $\boldsymbol{h}_t^{(\mathtt{c})}$ and $\boldsymbol{n}_t^{(\mathtt{c})}$ refer to the channel and noise distortions respectively in the cepstral domain[1]. The non-linear function $g(\cdot)$ is given by

$$g(\boldsymbol{x}) = \boldsymbol{C} \log\left(1 + \exp\left(\boldsymbol{C}^\dagger \boldsymbol{x}\right)\right) \quad (2)$$

where $\boldsymbol{C}$ and $\boldsymbol{C}^\dagger$ denote the truncated Discrete Cosine Transform (DCT) used to compute the cepstral parameters and the corresponding pseudo-inverse.The channel Jacobian $\boldsymbol{G}_m$ and the noise Jacobian $\boldsymbol{F}_m$ are given by

$$\boldsymbol{G}_m = \boldsymbol{C}\hat{\boldsymbol{G}}_m\boldsymbol{C}^\dagger \quad (3)$$

$$\boldsymbol{F}_m = \boldsymbol{C}\hat{\boldsymbol{F}}_m\boldsymbol{C}^\dagger = \boldsymbol{I} - \boldsymbol{G}_m \quad (4)$$

where $\hat{\boldsymbol{F}}_m = \boldsymbol{I} - \hat{\boldsymbol{G}}_m$ and $\hat{\boldsymbol{G}}_m$ is given by

$$\hat{\boldsymbol{G}}_m = \left\{\mathrm{diag}\left(1 + \exp\left(\boldsymbol{C}^\dagger \bar{\boldsymbol{\mu}}_{mhn}^{(\mathtt{c})}\right)\right)\right\}^{-1} \quad (5)$$

$\mathrm{diag}(\boldsymbol{x})$ converts a vector $\boldsymbol{x}$ into a diagonal matrix whose leading diagonal elements are given by $\boldsymbol{x}$.

2.2. Parallel Model Combination (PMC)

As previously mentioned, PMC is based on the *log-normal* approximation to convert the statistics between the cepstral domain and the linear spectral domain so that speech statistics can be easily modified given the channel and noise statistics. The statistics conversion formulae from the cepstral domain ($\mathtt{c}$) to the linear spectral domain ($\mathtt{s}$) are given by:

$$\boldsymbol{\mu}^{(\mathtt{s})} = \exp\left(\boldsymbol{C}^\dagger \boldsymbol{\mu}^{(\mathtt{c})} + \frac{\mathrm{diag}^{-1}\left[\boldsymbol{\Sigma}^{(\mathtt{l})}\right]}{2}\right) \quad (6a)$$

$$\boldsymbol{\Sigma}^{(\mathtt{s})} = \boldsymbol{M}^{(\mathtt{s})}\left(\exp\left(\boldsymbol{\Sigma}^{(\mathtt{l})}\right) - 1\right)\boldsymbol{M}^{(\mathtt{s})} \quad (6b)$$

[1]Superscript ($\mathtt{c}$) is used to denote cepstral domain variables.

where $\boldsymbol{\Sigma}^{(\mathtt{l})} = \boldsymbol{C}^\dagger \boldsymbol{\Sigma}^{(\mathtt{c})} \boldsymbol{C}^{\dagger\top}$ denotes the covariance matrix in the log spectral domain. $\boldsymbol{M}^{(\mathtt{s})}$ is a diagonal matrix such that $\boldsymbol{\mu}^{(\mathtt{s})} = \mathrm{diag}^{-1}(\boldsymbol{M}^{(\mathtt{s})})$. The corresponding conversion formula from linear spectral to cepstral domains are given by:

$$\boldsymbol{\mu}^{(\mathtt{c})} = \boldsymbol{C}\log\left(\boldsymbol{\mu}^{(\mathtt{s})} - \frac{1}{2}\log\left(\mathrm{diag}^{-1}\left[\boldsymbol{V}^{(\mathtt{s})}\right]\right)\right) \quad (7a)$$

$$\boldsymbol{\Sigma}^{(\mathtt{c})} = \boldsymbol{C}\log\left(\boldsymbol{V}^{(\mathtt{s})} + 1\right)\boldsymbol{C}^\top \quad (7b)$$

where $\boldsymbol{V}^{(\mathtt{s})} = \boldsymbol{M}^{(\mathtt{s})-1}\boldsymbol{\Sigma}^{(\mathtt{s})}\boldsymbol{M}^{(\mathtt{s})-1}$. The compensation due to additive noise can be performed easily in the linear spectral domain as follows:

$$\hat{\boldsymbol{\mu}}_m^{(\mathtt{s})} = \boldsymbol{\mu}_m^{(\mathtt{s})} + \boldsymbol{\mu}_n^{(\mathtt{s})} \quad (8a)$$

$$\hat{\boldsymbol{\Sigma}}_m^{(\mathtt{s})} = \boldsymbol{\Sigma}_m^{(\mathtt{s})} + \boldsymbol{\Sigma}_n^{(\mathtt{s})} \quad (8b)$$

By combining all the steps (Eq. 6, Eq. 7 and Eq. 8) together, the overall PMC compensation formulae can be written as[2]:

$$\hat{\boldsymbol{\mu}}_m^{(\mathtt{c})} = \boldsymbol{\mu}_m^{(\mathtt{c})} + \boldsymbol{\mu}_h^{(\mathtt{c})} + g\left(\tilde{\boldsymbol{\mu}}_{mhn}^{(\mathtt{c})}\right) + \frac{1}{2}\boldsymbol{C}\,\mathrm{diag}^{-1}\left[\boldsymbol{\Sigma}_m^{(\mathtt{l})} - \hat{\boldsymbol{\Sigma}}_m^{(\mathtt{l})}\right] \quad (9a)$$

$$\hat{\boldsymbol{\Sigma}}_m^{(\mathtt{c})} = \boldsymbol{C}\left(\log\left(\hat{\boldsymbol{G}}_m\left(\exp\left(\boldsymbol{\Sigma}_m^{(\mathtt{l})}\right) - 1\right)\hat{\boldsymbol{G}}_m{}^\top + \hat{\boldsymbol{F}}_m\left(\exp\left(\boldsymbol{\Sigma}_n^{(\mathtt{l})}\right) - 1\right)\hat{\boldsymbol{F}}_m{}^\top + 1\right)\right)\boldsymbol{C}^\top \quad (9b)$$

where $\hat{\boldsymbol{G}}_m$ and $\hat{\boldsymbol{F}}_m$ are computed using Eq. 5 except that $\bar{\boldsymbol{\mu}}_{mhn}^{(\mathtt{c})}$ is replaced by $\tilde{\boldsymbol{\mu}}_{mhn}^{(\mathtt{c})}$, which is given by

$$\tilde{\boldsymbol{\mu}}_{mhn}^{(\mathtt{c})} = \boldsymbol{\mu}_n^{(\mathtt{c})} - \boldsymbol{\mu}_m^{(\mathtt{c})} - \boldsymbol{\mu}_h^{(\mathtt{c})} + \frac{1}{2}\boldsymbol{C}\,\mathrm{diag}^{-1}\left[\boldsymbol{\Sigma}_n^{(\mathtt{l})} - \boldsymbol{\Sigma}_m^{(\mathtt{l})}\right]$$

2.3. Comparison of Computational Complexities

Note that Eq. 9a is very similar to Eq. 1a except that:

1. $g(\cdot)$ is computed using $\tilde{\boldsymbol{\mu}}_{mhn}^{(\mathtt{c})}$ instead of $\bar{\boldsymbol{\mu}}_{mhn}^{(\mathtt{c})}$
2. there is an additional term $\frac{1}{2}\,\mathrm{diag}^{-1}\left[\boldsymbol{\Sigma}_m^{(\mathtt{l})}\right]$

The above two additional items require the diagonal elements of $\boldsymbol{\Sigma}_m^{(\mathtt{l})}$ and $\boldsymbol{\Sigma}_n^{(\mathtt{l})}$ to be computed with $\mathcal{O}(D_c D_l)$ complexity, where D_c and D_l are the dimensions of the cepstral and log spectral domains. However, due to the expensive operations of $\log(\cdot)$ and $\exp(\cdot)$, the covariance matrix compensation of PMC in Eq. 9b is computationally much more expensive than that of VTS in Eq. 1b. The difference becomes more apparent when performing trajectory-based compensation.

[2]The channel distortion mean is also included for completeness.

3. APPROXIMATED PMC (APMC)

Due to the expensive computational cost for compensating the covariance matrices using PMC, an variant of PMC, called Approximated PMC (APMC), is proposed, where the approximations $\log(\cdot)$ and $\exp(\cdot)$ are used. $\log(\boldsymbol{x}+1) \approx \boldsymbol{x}$ and $\exp(\boldsymbol{x}) - 1 \approx \boldsymbol{x}$. Therefore, Eq. 9b can be approximated as

$$\hat{\boldsymbol{\Sigma}}_m^{(\mathtt{c})} \approx \tilde{\boldsymbol{G}}_m \boldsymbol{\Sigma}_m^{(\mathtt{c})} \tilde{\boldsymbol{G}}_m{}^\top + \tilde{\boldsymbol{F}}_m \boldsymbol{\Sigma}_n^{(\mathtt{c})} \tilde{\boldsymbol{F}}_m{}^\top \tag{10}$$

where $\tilde{\boldsymbol{G}}_m$ and $\tilde{\boldsymbol{F}}_m$ are computed based on $\tilde{\boldsymbol{\mu}}_{mhn}^{(\mathtt{c})}$. Note that the resulting covariance matrix compensation formula is very similar to that of VTS as shown in Eq. 1b. Therefore, APMC can be regarded as a hybrid between PMC and VTS. The mean compensation formulae for APMC is the same as that of the standard PMC in Eq. 9a, except that $\hat{\boldsymbol{\Sigma}}_m^{(\mathtt{1})}$ is computed using the new compensated covariance matrix in Eq. 10. Besides reducing the computational cost of variance compensation, the approximation in Eq. 10 also allows the dynamic covariance matrices to be compensated using the same continuous time approximation technique as VTS. Furthermore, the dynamic mean and variance compensation for APMC can also be derived by differentiating Eq. 9a and Eq. 9b:

$$\begin{aligned}\Delta\hat{\boldsymbol{\mu}}_m^{(\mathtt{c})} &= \boldsymbol{G}_m \Delta\boldsymbol{\mu}_m^{(\mathtt{c})} + \boldsymbol{F}_m \Delta\boldsymbol{\mu}_n^{(\mathtt{c})} \\ &\quad + \frac{1}{2}\boldsymbol{C}\,\mathrm{diag}^{-1}\boldsymbol{C}^\dagger \left[\Delta\boldsymbol{\Sigma}_m^{(\mathtt{c})} - \Delta\hat{\boldsymbol{\Sigma}}_m^{(\mathtt{c})}\right] \boldsymbol{C}^{\dagger\top} \quad (11a)\\ \Delta\hat{\boldsymbol{\Sigma}}_m^{(\mathtt{c})} &= \boldsymbol{G}_m \Delta\boldsymbol{\Sigma}_m^{(\mathtt{c})} \boldsymbol{G}_m{}^\top + \boldsymbol{F}_m \Delta\boldsymbol{\Sigma}_n^{(\mathtt{c})} \boldsymbol{F}_m{}^\top \quad (11b)\end{aligned}$$

The same approximation can also be applied to the recently proposed Trajectory-based PMC (TPMC) [6] method. The resulting method will be referred to as Trajectory-based APMC (TAPMC). Since variance compensation needs to be applied to the variances at each time instant of the trajectory as well as the covariance between different time instances, the computational cost of TPMC compensation is much higher. Consequently, the approximation applied to TAPMC leads to a substantial saving in computation (see Fig. 1 in Section 5).

4. NOISE RE-ESTIMATION

The noise parameters can be re-estimated by maximizing the likelihood using the Baum-Welch algorithm [7]. The auxiliary function to be optimized is given by:

$$\mathcal{Q} = -\frac{1}{2}\sum_{i,m} \hat{\boldsymbol{\Sigma}}_m^{(\mathtt{c})} \left\{ \beta_m^{(\mathtt{i})} \log\left|\hat{\boldsymbol{\Sigma}}_m^{(\mathtt{c})}\right| + \mathrm{Tr}\left(\boldsymbol{\psi}_m^{(\mathtt{i})} \hat{\boldsymbol{\Sigma}}_m^{(\mathtt{c})-1}\right)\right\}$$

where the sufficient statistics for component m are given by

$$\begin{aligned}\boldsymbol{\nu}_m^{(\mathtt{i})} &= \sum_t \gamma_m(t)\left(\hat{\boldsymbol{x}}_t^{(\mathtt{c})} - \hat{\boldsymbol{\mu}}_m^{(\mathtt{c})}\right) \quad (12)\\ \boldsymbol{\psi}_m^{(\mathtt{i})} &= \sum_t \gamma_m(t)\left(\hat{\boldsymbol{x}}_t^{(\mathtt{c})} - \hat{\boldsymbol{\mu}}_m^{(\mathtt{c})}\right)\left(\hat{\boldsymbol{x}}_t^{(\mathtt{c})} - \hat{\boldsymbol{\mu}}_m^{(\mathtt{c})}\right)^\top \quad (13)\\ \beta_m^{(\mathtt{i})} &= \sum_t \gamma_m(t) \quad (14)\end{aligned}$$

$\gamma_m(t)$ is the posterior probability of component m at time t, obtained from the forward-backward algorithm [9]. The noise mean update formula can be obtained by equating the differential of $\mathcal{Q}$ with respect to $\boldsymbol{\mu}_n^{(\mathtt{c})}$ to zero. It turns out that the update formula for the noise mean is similar to that of the standard VTS [7], except that the noise Jacobian matrix, $\tilde{\boldsymbol{F}}_m$ is computed differently for APMC:

$$\boldsymbol{\mu}_n^{(\mathtt{c})} = \boldsymbol{\mu}_{n,0}^{(\mathtt{c})} + \boldsymbol{\zeta}_n{}^{-1}\boldsymbol{\omega}_n \tag{15}$$

where

$$\boldsymbol{\zeta}_n = \sum_m \beta_m^{(\mathtt{i})} \tilde{\boldsymbol{F}}_m \hat{\boldsymbol{\Sigma}}_m^{(\mathtt{c})} \tilde{\boldsymbol{F}}_m{}^\top, \qquad \boldsymbol{\omega}_n = \sum_m \tilde{\boldsymbol{F}}_m \hat{\boldsymbol{\Sigma}}_m^{(\mathtt{c})} \boldsymbol{\nu}_m^{(\mathtt{i})}$$

The noise variance can be updated using the Newton method [7]. However, due to the complexity of the update formula, it will not be considered in this paper.

5. EXPERIMENTAL RESULTS

In this section, experimental results on the AURORA 4 database [10] are reported. Two sets of training data (clean and multi-noise) were used. Each set comprises 7138 training utterances (approximately 12 hours of speech, 84 speakers). The multi-noise training set contains clean and noisy speech data. Six different noises were artificially added to the clean data recorded at randomly chosen signal-to-noise ratios (SNR) between 10 and 20 dB. The average SNR of the test data was 15 dB. System evaluation involves performing continuous speech recognition using a 5000-word vocabulary on 330 test utterances for each noise condition. The 330 utterances contain recordings from 8 speakers with about 40 utterances per speaker. Only the 16 kHz testing data recorded from the first microphone at an average SNR of 10 dB were used for the following evaluation. This paper only considers the effects of additive noise.

The baseline clean and multi-noise systems were HMM-based cross-word triphone systems trained using HTK [11]. Decision tree state clustering [12] was used to obtain about 3000 distinct states. Each state is represented by a 16-component GMM. The acoustic features used had 39 dimensions including 12 Mel Frequency Cepstral Coefficient (MFCC) [13] and the `C0` energy as well as the first and second order dynamic features. Table 1 compares the Word Error Rate (WER) performance of different model compensation techniques. PMC uses Eq. 9a and Eq. 9b to compensate the static parameters and the VTS compensation formulae for the dynamic parameters. For each of the noise compensation techniques, two types of noise models were evaluated. The INIT noise model was estimated using the leading and trailing 20 frames of observations of the utterances for each noise conditions. Starting with the INIT noise model, three Baum-Welch iterations were performed to re-estimate the noise mean only for each utterance. The re-estimation for

Compensation Model	Noise Model	WER (%)							
		clean	car	babble	restaurant	street	airport	train	average
Clean	—	**6.0**	36.6	52.6	50.3	61.3	45.6	59.9	44.6
Multi-noise	—	7.9	9.3	14.2	20.1	19.9	12.6	19.7	14.8
VTS	INIT	7.1	11.3	15.9	20.6	19.3	16.5	19.8	15.8
	VTS	6.3	10.1	14.7	19.4	16.9	15.0	18.2	14.4
PMC	INIT	7.0	10.6	15.1	19.7	18.2	15.7	19.5	15.1
	VTS	6.3	9.4	14.2	18.3	16.8	14.1	17.7	13.8
APMC	INIT	7.0	10.4	15.3	19.7	18.0	15.9	19.1	15.1
	APMC	6.4	9.4	13.8	18.0	16.3	13.8	16.9	13.5
TVTS	INIT	7.2	9.2	14.6	18.0	17.2	14.5	17.9	14.1
	VTS	6.3	9.1	14.2	16.8	15.7	13.3	16.9	13.2
TPMC	INIT	6.3	8.5	13.9	17.1	16.7	13.7	16.6	13.3
	VTS	6.2	**8.4**	13.5	16.1	15.4	13.0	**16.0**	12.6
TAPMC	INIT	6.9	9.3	13.8	16.5	15.4	13.2	16.6	13.1
	VTS	6.3	8.6	**13.2**	**15.8**	**14.3**	**12.9**	16.6	**12.5**

Table 1. *WER (%) performance of various systems for different testing conditions on AURORA 4.*

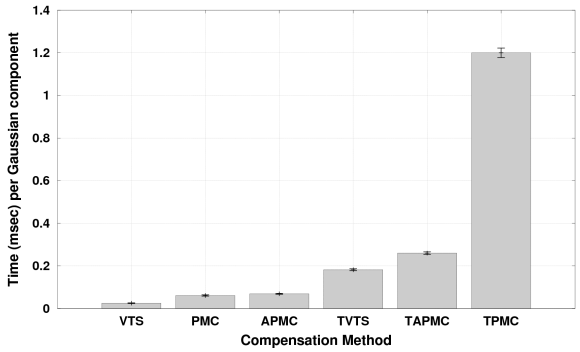

Fig. 1. Average durations (milliseconds) and the corresponding standard deviations for different compensation methods.

APMC is performed using the formula presented in Section 4. For the other compensation methods, the noise model re-estimated using VTS was used. The recognition outputs obtained from the corresponding INIT compensated systems were used as the supervision for noise re-estimation. In general, all the noise compensation methods achieved significant performance improvements over the uncompensated clean system. With the initial noise estimation (INIT), VTS achieved an average WER performance of 15.8% while PMC and APMC achieved 15.1%. Trajectory-based approaches gave better performances of 14.1%, 13.1% and 13.3% for TVTS, TAPMC and TPMC respectively. Noise re-estimation gives consistent absolute WER reductions between 0.6%–1.6%. Despite the approximation, both APMC and TAPMC achieved similar performance compared to PMC and TPMC respectively.

Finally, Fig. 1 compares the computational complexity of different compensation methods. The mean and standard deviation of the times taken to compensate 60320 Gaussian components in the system are measured on a Mac Pro server with two 2.93 GHz Quad-core Intel Xeon CPU and 8 GB memory. PMC and APMC took about the same time for compensation, which are slightly slower than VTS due to the additional terms that PMC and APMC needs to compute for mean compensationxt. Likewise, TAPMC is also slightly slower than TVTS. However, TPMC is almost four times slower than TAPMC due to the expensive variance and covariance compensation in the cepstral trajectory domain [6]. Therefore, the approximations applied to TAPMC has significantly improved its computational efficiency while retaining similar performance to TPMC.

6. CONCLUSIONS

This paper has presented the Approximated PMC (APMC) method, a modified version of PMC that approximates the compensation formulae of the variances to improve computational efficiency and allow simple compensation formulae to be derived for the dynamic parameters using continuous time approximation. The recently proposed trajectory-based compensation approach has also been applied to APMC to yield Trajectory APMC (TAPMC) so that an improved compensation for the dynamic parameters can be achieved. Experimental results on AURORA 4 show that APMC and TAPMC consistently outperformed VTS and TVTS. Despite the approximation, TAPMC retains the performance of TPMC while achieving almost four times speed up in compensation time.

7. REFERENCES

[1] L. A. Rabiner, "A tutorial on hidden Markov models and selective applications in speech recognition," in *Proc. of the IEEE*, February 1989, vol. 77, pp. 257–286.

[2] Xuedong Huang, Alex Acero, Hsiao-Wuen Hon, and Raj Reddy, *Spoken Language Processing: A Guide to Theory, Algorithm and System Development*, Prentice Hall PTR, 1st edition, 2001.

[3] M. J. F. Gales and S. J. Young, "Cepstral parameter compensation for HMM recognition in noise," *Speech Commun.*, vol. 12, pp. 231–239, July 1993.

[4] Mark John Francis Gales, *Model-Based Techniques for Noise Robust Speech Recognition*, Ph.D. thesis, Gonville and Caius College, University of Cambridge, 1996.

[5] Alex Acero, Li Deng, Trausti Kristjansson, and Jerry Zhang, "HMM adaptation using vector Taylor series for noisy speech recognition," in *Proc. of ICSLP*, 2000, vol. 3, pp. 869–872.

[6] K. C. Sim and M. Luong, "A trajectory-based parallel model combination with a unified static and dynamic parameter compensation for noisy speech recognition," in *Proc. of Automatic Speech Recognition and Understanding Workshop*, 2011.

[7] Ozlem Kalinli, Michael L. Seltzer, Jasha Droppo, and Alex Acero, "Noise adaptive training for robust automatic speech recognition," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 18, no. 8, pp. 1889–1901, 2010.

[8] R.C. van Dalen and M.J.F. Gales, "Extended VTS for noise-robust speech recognition," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, pp. 733–743, 2011.

[9] L. E. Baum and J. A. Eagon, "An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology," *Bull. Amer. Math. Soc.*, vol. 73, pp. 360–363, 1967.

[10] N. Parihar, J. Picone, D. Pearce, and H.G. Hirsch, "Performance analysis of the Aurora large vocabulary baseline system," in *Proceedings of the European Signal Processing Conference*, 2004.

[11] S. J. Young, G. Evermann, M. J. F. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. C. Woodland, *The HTK Book (for HTK version 3.4)*, Cambridge University, December 2006.

[12] S. J. Young, J. J. Odell, and P. C. Woodland, "Tree-based state tying for high accuracy acoustic modelling," in *Proceedings ARPA Workshop on Human Language Technology*, 1994, pp. 307–312.

[13] S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustic, Speech and Signal Processing*, vol. 28, no. 4, pp. 357–366, 1980.