音声言語情報処理 34－5
(2000. 12. 21)

Extended Kalman Particle filters applied to model-based noise compensation for noisy speech recognition

Kaisheng Yao, Tomoko Matsui and Satoshi Nakamura

ATR Spoken Language Translation Research Laboratories

2-2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0288 JAPAN

E-Mail:{kyao, tmatsui, nakamura}@slt.atr.co.jp

Abstract We suggest viewing noisy speech recognition based on Jump Markov State Space model. In this model, noise parameters and state sequences are hidden and estimated by a computational Bayesian approach for parameter estimation. Particularly, the Monte-Carlo particle filters were adopted to estimate time-varying additive noise parameter for model-based noise compensation. Each particle corresponds to a certain state space of noise. The particles randomly transit to new state spaces of noise according to the transition probability given by acoustic models and language models for speech recognition. Higher likelihood particles generate larger number of new particles with newly evolved state space, whereas the lower likelihood particles may be stopped by a selection step. The state space after a particular transition was evolved using an extended Kalman filter. Likelihood of each state space contributes to Minimum Mean Square Error (MMSE) estimation of the noise parameter from all the particles. Primary experiments on N-Best rescoring are shown in this paper.

Key words Speech recognition, Noise compensation, State space model, Kalman filter, Monte-Carlo method, Particle filter.

拡張型カルマン・パーティクル・フィルタを用いた雑音下音声認識

姚開盛、松井知子、中村哲

ATR 音声言語通信研究所

〒619-0288 京都府相楽郡精華町光台 2-2-2

E-Mail:{kyao, tmatsui, nakamura}@slt.atr.co.jp

あらまし 本稿では、ジャンプ・マルコフ状態空間モデルに基づく、雑音下の音声認識手法について述べる。このモデルでは雑音パラメータと状態列は隠れた変数として、計算ベイズアプローチによって推定する。本方法はモデルレベルの雑音補償法であり、時間的に変化する加算性雑音のパラメータを、モンテ・カルロ・パーティクル・フィルタを用いて推定する。パーティクルは各時刻における雑音の状態を表す空間（雑音状態空間）に相当する。概念的には、ある時刻のパーティクルは、音声認識中に得られる音響・言語モデルから計算される遷移確率に応じて、次の時刻の雑音状態空間の候補となる、新たな複数のパーティクルに確率的に遷移していくと考える。その際、大きい尤度の値を示すパーティクルは遷移先として、より多くの候補に展開され、小さな尤度を示すパーティクルは場合によっては、それ以上の展開は行わないと判定される。なお、それらの展開には拡張型カルマン・フィルタを用いる。雑音パラメータは、各状態空間の尤度に基づいて、最小2乗誤差推定法によって推定する。本稿では、予備的な実験として、本方法をN-bestリスコアリングに適用した結果を示す。

キーワード 音声認識、雑音補償、状態空間モデル、カルマン・フィルタ、モンテ・カルロ法、パーティクル・フィルタ

1. INTRODUCTION

For speech recognition in noise, several representative noise compensation methods [1, 2, 3, 4] have been proposed, and have shown their potential applications for robust speech recognition in real noisy environments. A survey of the methods can be seen in [5]. Most of the methods assume that the noise statistics are constant, so that noise parameter can be estimated in advance, e.g., based on the Maximum Likelihood estimation [5], and plugged into the noise compensation procedures. Though it is possible to use more Gaussian mixtures or states to represent time-varying noise statistics, the modeled noise statistics can not adapt to unseen environments.

As a result, for speech recognition in non-stationary noise, the above methods may lack effectiveness. In order to give more insights on noisy speech recognition in non-stationary noise, we view speech recognition in noise as a non-stationary parameter estimation problem. Accordingly, we suggest to represent noisy speech recognition in the framework of the Jump Markov State Space model [6], i.e.,

$$x(t+1) = A(x(t)) + B\nu(t) \quad (1)$$

$$y(t) = F_{s_t}(x(t)) + Dv(t) \quad (2)$$

$$s_{t+1} = \Phi(s_t) \quad (3)$$

where $x(t) \in R$, $y(t) \in R$ and $s_t \in Z$ each represents continuous state, observation, and the discrete state. $A(\cdot)$, $F_{s_t}(\cdot)$ and $\Phi(\cdot)$ each represents the environment transition function, observation function and the state transition function. They can be linear or non-linear. $\nu(t)$ and $v(t)$ respectively represent continuous state driving noise and the measurement noise. The objective is to estimate the hidden continuous state parameter $x(t)$ and the hidden discrete state parameter s_t given observation $y(t)$.

We note that, given state s_t, the combination of the Equation (1) and Equation (2) represents a state space model. Also, given $x(t)$, the combination of the Equation (2) and Equation (3) represents Hidden Markov Model (HMM).

For noisy speech recognition, the $x(t)$ represents changing environment parameter and the sequence of $s(t)$ is (phoneme) state sequence. This can be understood in the following way. Given a correct state sequence s_t, we can estimate noise parameter $x(t)$; Given the noise parameter estimation $x(t)$, we can do HMM decoding to decide the state sequence s_t.

Normally, joint estimate the noise parameter $x(t)$ and the discrete state sequence s_t is an N-P hard problem. One suboptimal way can be done as an iterative way as follows.

1. Step 1: Noise parameter $x(t)$ can be estimated given a (hypothesized) state sequence, and the estimated noise parameter can be plugged back to Equation (2) to do noise compensation in feature space or model space.

2. Step 2: The compensated feature or model can be used for HMM decoding to decide a new state sequence which may be used for noise estimation in Step 1.

If we assume that the noise is stationary, as we have said previously, we can do one pass of the above steps and then plug the estimated noise parameter for speech recognition afterwards.

On the other hand, if the noise statistic is changing during recognition, we may have to run Step 1 and Step 2 in several iterations, where each iteration will assume that the previous estimation of $x(t)$ or s_t is reliable for the current parameter estimation. For example, we can use a sequential Expectation Maximization (EM) algorithm [7] to do iterative sequential compensation of the mean vector of acoustic models, and the compensated model are used for noisy speech recognition.

Kalman filter can also be used in model-based noise compensation [8, 9]. The method in [8] uses a single-state multiple-observation state space, which corresponds to estimate the environment noise parameter by using only one state function plus a large amount of observation functions from state mixtures. Once a state mixture has been invoked, it will last until it reaches to the end of a utterance or it is pruned out during speech recognition. The interacting-multiple-model (IMM) based method [9] uses a parallel set of Gaussian mixture representing a set of state space models. Each Gaussian mixture will be valid during the whole utterance. From the above explanation, we can view the above two methods using Kalman filter as the case of setting $\Phi(\cdot)$ as a $1(\cdot)$ in Equation (3). For this reason, the above two methods are in the category of the deterministic Gaussian mixture approximation of the posterior probability [6] for estimating non-stationary noise parameter for noisy speech recognition problem. Intuitively, the deterministic Gaussian mixture approximation is not consistent to the situation in speech recognition, since the phonemes are changing in an utterance. It is better to select the state space according to some probability.

Another possible strategy is to compute a grid approximation to the filtered posterior density. The grid point is called as "particle", and evolves in either a fixed way (General particle filter) or a randomized adaptive way (Monte Carlo particle filter). The particles might cover the evolution of the noisy speech statistics.

This paper is the first step trying to introduce Monte Carlo particle filters into the noise parameter estimation for noisy speech recognition. We will briefly outline our method in section 2. Detailed procedure will be shown in section 3. Experiments and discussions are in section 4 and section 5.

2. THE MONTE CARLO PARTICLE FILTER FOR NOISE PARAMETER ESTIMATION IN NOISY SPEECH RECOGNITION

Following our statement that we can formulate noisy speech recognition by Equation (1) to Equation (3), specifically, for our problem of noisy speech recognition with the Log-Add noise compensation [7], assuming that there is no correlation among log-spectral filter banks, we can write Equation (1) to Equation (3) by the following equations for each filter bank j in particle $s = s_t$, i.e.,

$$\mu^l_{nj}(t+1) = \mu^l_{nj}(t) + \nu_j(t) \quad (4)$$

$$y_j(t) = \mu^l_{sj}(t) + \log\left(1 + \exp\left(\mu^l_{nj}(t) - \mu^l_{sj}(t)\right)\right)$$

$$+ \quad v_{sj}(t) \tag{5}$$

$$s_{t+1} = \Phi(s_t) \tag{6}$$

Note that the Equation (5) is derived as the observation function in the Log-Add noise compensation [7]. Subscript j represents log-spectral filter bank index, and J is the total number of the log-spectral filter-banks. $\nu_j(t)$ is state driving noise in Normal distribution $N(0, \Xi_j(t))$, and $v_{sj}(t)$ is the measurement driving noise in Normal distribution $N(0, V_{sj}(t))$, where $N(\mu, \Sigma)$ represents Normal distribution with mean μ and covariance Σ. Superscript l represents log-spectral domain. $\Phi(\cdot)$ represents the state transition function.

For each filter bank j in particle s, the state space given by Equation (4) and Equation (5) is nonlinear. The extended Kalman filter may be used to update the corresponding state space model analytically. The updating formula is given by [10],

$$\textit{Prior estimate at } t:$$

$$\mu^l_{nj}(t|t-1) = \hat{\mu}^l_{nj}(t-1) \tag{7}$$

$$R_{nj}(t|t-1) = R_{nj}(t-1) + \Xi_j(t) \tag{8}$$

$$\textit{One} - \textit{step forecast}:$$

$$f_{sj}(t|t-1) = \mu^l_{sj}(t) \tag{9}$$

$$+ \quad \log(1 + \exp(\mu_{nj}(t|t-1) - \mu^l_{sj}(t))) \tag{10}$$

$$Q_{sj}(t|t-1) = F_{sj}(t)' R_{nj}(t|t-1) F_{sj}(t)$$

$$+ \quad V_{sj}(t) \tag{11}$$

$$\textit{Posterior estimate at } t:$$

$$K_{sj}(t) = R_{nj}(t|t-1) F_{sj}(t) Q_{sj}(t|t-1)^{-1} \tag{12}$$

$$e_{sj}(t) = y_j(t) - f_{sj}(t|t-1) \tag{13}$$

$$\hat{\mu}^l_{nj}(t) = \mu^l_{nj}(t|t-1) + K_{sj}(t) e_{sj}(t) \tag{14}$$

$$R_{nj}(t) = R_{nj}(t|t-1) - K_{sj}(t) Q_{sj}(t|t-1) K_{sj}(t)' = \frac{R_{nj}(t|t-1) V_{sj}(t)}{Q_{sj}(t|t-1)} \tag{15}$$

where $F_{sj}(t) = \frac{\partial y_j(t)}{\partial \mu^l_{nj}(t)}|_{s_t=s} = \frac{\exp(\mu^l_{nj}(t) - \mu^l_{sj}(t))}{1+\exp(\mu^l_{nj}(t) - \mu^l_{sj}(t))}$.

For speech recognition, the sequence of s_t in Equation (6) represents the state mixture sequence with constraints from the language model and acoustic model, i.e., $\Phi(s_t) = p(s_{t+1}|s_t)$, where $p(l|m)$ is the transition probability from mixture m to mixture l. Note that the above state space evolution by Equation (7) to Equation (15) is carried out at state $s = s_t$, where s_t is stochastic during speech recognition.

Our method will generate a set of particles by sampling the state transition probability given by acoustic model and language model. Once a particle has been given, the state space related with this particle evolves by the above extended Kalman filters. The likelihood of the state space will give a weight of this particle for Minimum Mean Square Error (MMSE) estimation of the noise parameter. A selection step is followed to keep particles with high weights and accordingly let the particle with high weight give more newly evolved state space. After the selection step, all the particles will be assigned to equal weights, however, since the number of particles closer to the higher posterior probability is much larger than those particles far away from the ranges of higher posterior probability, the parameter estimation will be efficient. MMSE estimation can be carried out at this stage. A Monte-Carlo Markov Chain is followed to possibly reduce the total variation norm of the current distribution of the particles with respect to the 'target' posterior distribution.

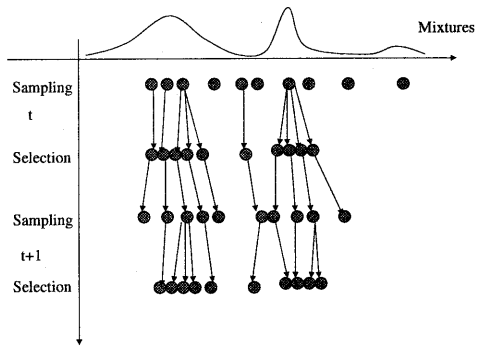

Fig. 1. Monte-Carlo evolution of the particles. The curve represents the 'target' posterior probability. Sampling will expand the search space. Selection step will condense the search space according to the weight of each particles. Note that some particle may give more 'children', while others may be dead after the selection step. As a result, particles are concentrating in the ranges of higher posterior probability.

Detailed explanation of the procedure is in the following sections.

3. PROCEDURES IN THE MONTE CARLO PARTICLE FILTERS

3.1. Minimum Mean Square Error estimation of noise parameters by Monte Carlo Particle Filters

For the aim of estimating $\mu^l_{nj}(t)$, which is used by the Log-Add noise compensation [7], we apply the Minimum Mean Square Error (MMSE) estimation. The estimation of the noise parameter in filter bank j from time 0 to t is given by,

$$\hat{\mu}^l_{nj}(0:t) = \sum_{s_{0:t}} \int \mu^l_{nj}(0:t) p(s_{0:t}, \mu^l_{nj}(0:t)|y(1:t)) d\mu^l_{nj}(0:t)$$

where $y(1:t)$ is the observation vector sequence with element of $y_j(1:t)$, $j = 1 \cdots J$. $s_{0:t}$ denotes the state sequence from 0 to t. $p(s_{0:t}, \mu^l_{nj}(0:t)|y(1:t))$ is the posterior probability of state sequence $s_{0:t}$ and hidden continuous state $\mu^l_{nj}(0:t)$ given observation $y(1:t)$.

Empirical distribution of the posterior probability is given

by,

$$\bar{P_M}(ds_{0:t}, d\mu^l_{nj}(0:t)|y(1:t)) = \frac{1}{M}\sum_{i=1}^{M}\delta(s^{(i)}_{0:t} = ds_{0:t}, \mu^{l(i)}_{nj} = d\mu^l_{nj}(0:t))$$

where M is the total number of particles. $s^{(i)}_{0:t}$ is the state sequence of particle i. $\mu^{l(i)}_{nj}(0:t)$ is the estimation of noise parameter in filter bank j from 0 to t at particle filter i. $\delta(\cdot)$ is the delta function. $ds_{0:t}$ and $d\mu^l_{nj}(0:t)$ each denotes a small 'area' of $s_{0:t}$ and $\mu^l_{nj}(0:t)$.

Using the empirical estimate of the posterior probability to approximate the true posterior probability, we can approximately estimate $I(g_{t|t})$ for any function $g_{t|t}$.

$$\begin{aligned}\bar{I_M}(g_{t|t}) = & \sum_{s_{0:t}}\int g_{t|t}(s_{0:t}, \mu^l_{nj}(0:t)) \\ & \bar{P_M}(ds_{0:t}, d\mu^l_{nj}(0:t)|y(1:t)) \\ = & \frac{1}{M}\sum_{i=1}^{M} g_{t|t}(s^{(i)}_{0:t}, \mu^{l(i)}_{nj}(0:t))\end{aligned}$$

This estimate is unbiased and form the strong law of large number, i.e., as $M \to +\infty$, $\bar{I_M}(g_{t|t}) \to I(g_{t|t})$. $I(g_{t|t})$ is the MMSE estimate of the $\mu^l_{nj}(t)$ in this paper.

Thus, one of the key points in the algorithm is to estimate the empirical posterior probability by particle filters. To estimate the empirical posterior probability and $I(g_{t|t})$, the Bayesian Importance Sampling (BIS) [10] method is used. This method assumes the existence of an arbitrary *importance distribution* $\pi(s_{0:t}, \mu^l_{nj}(0:t)|y(1:t))$ which can be easily simulated from, and $p(s_{0:t}, \mu^l_{nj}(0:t)|y(1:t)) > 0$ implies $\pi(s_{0:t}, \mu^l_{nj}(0:t)|y(1:t)) > 0$. Using this distribution, $I(g_{t|t})$ can be expressed as,

$$I(g_{t|t}) = \frac{E_{\pi(s_{0:t},\mu^l_{nj}(0:t)|y(1:t))}[g_{t|t}(s_{0:t}, \mu^l_{nj}(0:t))w(s_{0:t}, \mu^l_{nj}(0:t))]}{E_{\pi(s_{0:t},\mu^l_{nj}(0:t)|y(1:t))}[w(s_{0:t}, \mu^l_{nj}(0:t))]}$$

where $E_{\pi(\cdot)}(\cdot)$ is the expectation with respect to $\pi(\cdot)$, and the importance weight $w(s_{0:t}, \mu^l_{nj}(0:t))$ is given by,

$$w(s_{0:t}, \mu^l_{nj}(0:t)) \propto \frac{p(s_{0:t}, \mu^l_{nj}(0:t)|y(1:t))}{\pi(s_{0:t}, \mu^l_{nj}(0:t)|y(1:t))} \quad (16)$$

Accordingly, a Monte Carlo estimate of $I(g_{t|t})$ is given by,

$$\bar{I}_M(g_{t|t}) = \sum_{i=1}^{M}\bar{w}^{(i)}_{0:t} g_{t|t}(s^{(i)}_{0:t}, \mu^{l(i)}_{nj}(0:t)) \quad (17)$$

where the normalized importance weights are

$$\bar{w}^{(i)}_{0:t} = \frac{w(s^{(i)}_{0:t}, \mu^{l(i)}_{nj}(0:t))}{\sum_{j=1}^{M} w(s^{(j)}_{0:t}, \mu^{l(j)}_{nj}(0:t))} \quad (18)$$

3.2. Rao-Blackwellisation

Variance reduction of the above estimation procedure can be achieved by Rao-Blackwellization [11], where importance distribution is the marginal distribution of $s_{0:t}$ given $y(1:t)$, i.e.,

$$\pi(s_{0:t}|y(1:t)) = \int \pi(s_{0:t}, \mu^l_{nj}(0:t)|y(1:t))d\mu^l_{nj}(0:t) \quad (19)$$

As a result,

$$w(s_{0:t}) \propto \frac{p(s_{0:t}|y(1:t))}{\pi(s_{0:t}|y(1:t))} \quad (20)$$

$$\hat{I_M}(g_{t|t}) = \sum_{i=1}^{M}\tilde{w}^{(i)}_{0:t} E_{p(\mu^l_{nj}(t)|s^{(i)}_{0:t}, y(1:t))}[g_{t|t}(s^{(i)}_{0:t}, \mu^{l(i)}_{nj}(t))] \quad (21)$$

$$\tilde{w}^{(i)}_{0:t} = \frac{w(s^{(i)}_{0:t})}{\sum_{j=1}^{M} w(s^{(j)}_{0:t})} \quad (22)$$

As we have seen in Section 2, the expectation operation of $E_{p(\mu^l_{nj}(t)|s^{(i)}_{0:t}, y(1:t))}[g_{t|t}(s^{(i)}_{0:t}, \mu^{l(i)}_{nj}(t))]$ can be done by the extended Kalman filter on the non-linear state space model shown in Equation (7) to Equation (15).

3.3. Sequential Importance Sampling (SIR)

Factorization the posterior probability and the importance distribution allows a recursive evaluation of the importance weights, i.e., $w(s_{0:t}) = w(s_{0:t-1})w(s_t)$, with,

$$\begin{aligned} w(s_t) &= \frac{p(y(t)|y(1:t-1), s_{1:t})p(s_t|s_{t-1})}{p(y(t)|y(1:t-1))\pi(s_t|y(1:t), s_{1:t-1})} \\ &\propto \frac{p(y(t)|y(1:t-1), s_{1:t})p(s_t|s_{t-1})}{\pi(s_t|y(1:t), s_{1:t-1})} \end{aligned} \quad (23)$$

Specifically, we choose $\pi(s_t|y(1:t), s_{1:t-1}) = p(s_t|s_{t-1})$, the state transition probability representing the acoustic model and the language model, as the proposal importance distribution. Thus, we have,

$$w(s_t) \propto p(y(t)|y(1:t-1), s_{1:t}) \quad (24)$$

Calculation of the above parameter needs only one step of Kalman prediction, which is given by $\prod_{j=1}^{J} N(f_{sj}(t|t-1), Q_{sj}(t|t-1))$ in Equation (10) and Equation (11).

However, the weight is only related with likelihood instead of the posterior probability, which can possibly increase estimation variance.

3.4. Selection step

The variance of the above estimation procedure is increasing over time [12]. This is observed as, after a few above iterations, all but one of the normalized importance weights are very close to zero and a large computational burden is devoted to updating trajectories whose contribution to the final estimation is almost zero. A selection step is followed

by using residual re-sampling [13], which assigns the number of the "children" of a particle according to the weight of this particle. The larger the weight, the larger the number of its "children". After the selection step, all the particles are assigned with equal weight.

3.5. Monte Carlo Markov Chain (MCMC) step

After the selection step at time t, we obtain M particles distributed approximately according to $p(s_{0:t}|y(1:t))$. Note that the discrete nature of the approximation can lead to a skewed importance weights distribution. Particles may have no children ($M_s = 0$), whereas others have a large number of children. The extreme case being $M_s = M$ for a particular value s. In this case, there is a severe depletion of samples. A strategy for improving the results is introducing MCMC steps of invariant distribution $p(s_{0:t}|y(1:t))$ on each particle. In this paper, Metropolis-Hastings step [14] is used as follows,

1. Sampling
   - $v \sim U_{[0,1]}$, where $U_{[0,1]}$ is the flat distribution between 0 and 1.
2. Sample the proposal candidate
   - $s_t^{\star(i)} \sim p(s_t|s_{t-1}^{(i)})$
3. Move
   - If $v \leq \frac{\prod_{j=1}^{J} p(y_j(t)|\mu_{n_j}^L(t-1), s_t^{\star(i)})}{\prod_{j=1}^{J} p(y_j(t)|\mu_{n_j}^L(t-1), \tilde{s}_t^{(i)})}$
     - – then accept move: $s_{0:t}^{(i)} = \{\tilde{s}_{0:t-1}^{(i)}, s_t^{\star(i)}\}$
     - – else reject move: $s_{0:t}^{(i)} = \tilde{s}_{0:t}^{(i)}$

     End if.

The method can possibly spread the particles in a mode.

4. EXPERIMENTAL RESULTS

4.1. Experimental setup

Speaker-Independent TI-Digits recognition experiments were carried out with a Viterbi recognizer to test the application of the particle filters for noisy speech recognition. The digits models and background noise model were trained on clean speech utterances. The contaminated speech for the test was generated by artificially adding different levels of noise to the clean speech. All noise signals were from a Noisex-92 database.

Five hundred connected digits utterances from 15 speakers and 100 connected digits utterances from four speakers unseen in the training set were used for training and testing, respectively. There were 11 whole word models for 10 digits (zero is pronounced as oh or zero) and one silence model. Each digit was modeled by a four-Gaussian-mixture 10-state (including a non-emitting initial and final state) left-to-right HMM without skip states. Gaussian output probability distributions with diagonal covariance matrices were used for each state. The silence model was a four-Gaussian-mixture 3-state (with a first and last non-emitting state) HMM.

The speech signals were down-sampled from 20kHz to 16kHz. The window size was 25.0ms with a 10.0ms shift. Twenty-six filters were used in the binning stage. The features were the static MFCC with the dynamic MFCC.

State driving noise of all the particle filters were intialized to be the variance of the estimated noise parameter at each log-spectral filter bank. The initialized noise parameter was estimated from 5 seconds of noise before recognition. The measurement noise in all of the state space is set to be the same as the state driving noise.

4.2. Results

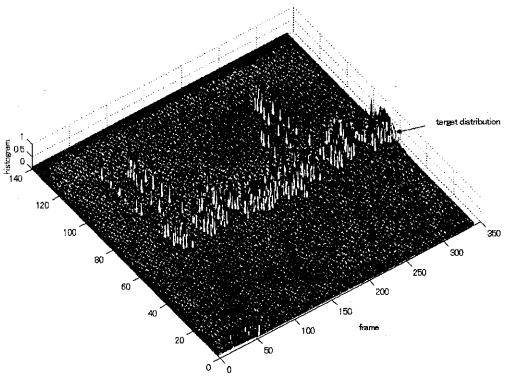

Fig. 2. Evolution of the histogram. The target distribution is the up-right most curve.

We firstly look at the evolution of the histogram of the estimated noise parameters. Specifically, in Figure 2, we plot the evolution of the histogram of the estimated noise parameter at 10-th log-spectral filter bank. The up-right most curve is the target distribution of the contaminating noise parameter at the filter bank. 357 particle filters were used to do noise parameter estimation. Utterance 15ooa[1] was contaminated by White noise in signal-to-noise ratio of 2.87dB. We did the parameter estimation of the utterance in two times, and then concatenated the histograms of the estimated noise parameter. As can be seen from the figure, the histogram of the estimated noise parameter goes from its initial distribution far away from the true distribution, and at most of the time, the histograms of the estimated noise parameters are in the range of the true distribution of the contaminating noise parameter. However, we notice that there some jumpings which result in wrong parameter estimation. We notice that the jumping is almost periodic, showing that it might be because of the large state driving noise we set in our experiments.

[1] One utterance in TI-Digits database

We then use particle filters for N-Best re-scoring. 10-Best lists were generated by a speech recognition system compensated by Log-Add method [7] in each SNR noisy environments.

We tested the performance of the particle filters with different number of particle filters. Performances of the particle filters are shown in the following tables. Particle (20), particle (50), and particle (150) each represents performance by using 20, 50,and 150 particle filters for noise parameter estimation and noise compensation afterwards.

Table 1. Word Error Rate (in %) of noise compensation methods in White noise.

SNR (dB)	8.8	16.0	20.4	40.4
Baseline	81.3	62.0	37.3	22.3
particle (20)	61.7	42.7	10.3	10.0
particle (50)	59.3	41.3	13.0	11.0
particle (150)	61.7	32.0	8.7	3.7

Table 2. Word Error Rate (in %) of noise compensation methods in Babble noise.

SNR (dB)	0.7	9.4	12.9	32.6
Baseline	90.7	97.7	44.0	20.7
particle (20)	80.7	57.0	13.7	7.0
particle (50)	79.7	48.7	9.0	6.7
particle (150)	74.7	54.0	11.7	6.7

Accuracy of the parameter estimation by particle filters is important to noise compensation afterwards. As we can see from the tables, in order to improve the accuracy of the particle filter for parameter estimation, we may have to increase the number of particle filters.

5. DISCUSSIONS

A Monte-Carlo method for parameter estimation in noisy speech recognition has been presented in this paper. The Monte-Carlo method provides a general way for parameter estimation in the framework of the Jump Markov State Space model. We notice that the accuracy of the parameter estimation is critical to the performance for noisy speech recognition. For this reason, it is better to increase the number of particle filter for parameter estimation. Also, the importance distribution for sequential sampling is important for variance reduction in parameter estimation. Currently, we use only the transition probability without consideration of likelihood during recognition. Better variance reduction and parameter estimation can be expected with importance distribution considering likelihood, which corresponds to weighting parameter estimation from each particles by posterior probability instead of the likelihood we used in our primary experiments.

6. REFERENCES

[1] M. G. Rahim and B.-H. Juang, "Signal bias removal by maximum likelihood estimation for robust telephone speech recognition," *IEEE Trans. on SAP*, vol. 4, no. 1, pp. 19–30, January 1996.

[2] M.J.F.Gales and S.J.Young, "Robust speech recognition in additive and convolutional noise using parallel model combination," *Computer Speech and Language*, vol. 9, pp. 289–307, 1995.

[3] P.J. Moreno, B. Raj, and R. M. Stern, "A vector taylor series approach for environment-independent speech recognition," in *ICASSP*, 1996, vol. 2, pp. 733–736.

[4] Ananth Sankar and Chin-Hui Lee, "A maximum–likelihood approach to stochastic matching for robust speech recognition," *IEEE Trans. on Speech and Audio Processing*, vol. 4, no. 3, pp. 190–201, 1996.

[5] Y. Zhao, "Maximum likelihood joint estimation of channel and noise for robust speech recognition," in *ICASSP*, 2000, vol. 2, pp. 1109–1113.

[6] A. Doucet, N. J. Gordon, and V. Krishnamurthy, "Particle filters for state estimation of jump markov linear systems," Tech. Rep. CUED-TR 359, Cambridge University, 2000.

[7] K. Yao, B. E. Shi, S. Nakamura, and Z. Cao, "Residual noise compensation by a sequential em algorithm for robust speech recognition in nonstationary noise," in *ICSLP*, 2000, vol. 1, pp. 770–773.

[8] K. Yao, B. E. Shi, P. Fung, and Z. Cao, "Residual noise compensation for robust speech recognition in nonstationary noise," in *ICASSP*, 2000, vol. 2, pp. 1125–1128.

[9] N. S. Kim, "Imm–based estimation for slowly evolving environments," *IEEE Signal Processing Letters*, vol. 5, no. 6, pp. 146–149, June 1998.

[10] M. West and J. Harrison, *Bayesian Forecasting and Dynamic Models*, Springer, 2 edition, 1997.

[11] G. Casella and C. P. Robert, "Rao-blackwellisation of sampling schemes," *Biometrika*, vol. 81, no. 1, 1996.

[12] A. Kong, J. S. Liu, and W. H. Wong, "Sequential imputations and bayesian missing data problems," *J. Am. Stat. Assoc*, vol. 89, pp. 278–288, 1994.

[13] J. S. Liu and R. Chen, "Sequential monte carlo methods for dynamic systems," *J. Am. Stat. Assoc*, vol. 93, pp. 1032–1044, 1998.

[14] W. K. Hastings, "Monte carlo sampling methods using markov chains and their applications," *Biometrika*, vol. 57, pp. 97–109, 1970.