EURASIP Journal on Applied Signal Processing 2004:15, 2366–2384

Time-Varying Noise Estimation for Speech Enhancement and Recognition Using Sequential Monte Carlo Method

Kaisheng Yao
Institute for Neural Computation, University of California, San Diego, 9500 Gilman Drive, La Jolla, CA 92093-0523, USA
Email: kyao@ucsd.edu

Te-Won Lee
Institute for Neural Computation, University of California, San Diego, 9500 Gilman Drive, La Jolla, CA 92093-0523, USA
Email: tewon@ucsd.edu

Received 4 May 2003; Revised 9 April 2004

We present a method for sequentially estimating time-varying noise parameters. Noise parameters are sequences of time-varying mean vectors representing the noise power in the log-spectral domain. The proposed sequential Monte Carlo method generates a set of particles in compliance with the prior distribution given by clean speech models. The noise parameters in this model evolve according to random walk functions and the model uses extended Kalman filters to update the weight of each particle as a function of observed noisy speech signals, speech model parameters, and the evolved noise parameters in each particle. Finally, the updated noise parameter is obtained by means of minimum mean square error (MMSE) estimation on these particles. For efficient computations, the residual resampling and Metropolis-Hastings smoothing are used. The proposed sequential estimation method is applied to noisy speech recognition and speech enhancement under strongly time-varying noise conditions. In both scenarios, this method outperforms some alternative methods.

Keywords and phrases: sequential Monte Carlo method, speech enhancement, speech recognition, Kalman filter, robust speech recognition.

1. INTRODUCTION

A speech processing system may be required to work in conditions where the speech signals are distorted due to background noise. Those distortions can drastically drop the performance of automatic speech recognition (ASR) systems, which usually perform well in quiet environments. Similarly, speech-coding systems spend much of their coding capacity encoding additional noise information.

There have been great interests in developing algorithms that achieve robustness to those distortions. In general, the proposed methods can be grouped into two approaches. One approach is based on front-end processing of speech signals, for example, speech enhancement. Speech enhancement can be done either in time-domain, for example, in [1, 2], or more widely used, in spectral domain [3, 4, 5, 6, 7]. The objective of speech enhancement is to increase signal-to-noise ratio (SNR) of the processed speech with respect to the observed noisy speech signal.

The second approach is based on statistical models of speech and/or noise. For example, parallel model combination (PMC) [8] adapts speech mean vectors according to the input noise power. In [9], code-dependent cepstral normalization (CDCN) modifies speech signals based on probabilities from speech models. Since methods in this model-based approach are devised in a principled way, for example, maximum likelihood estimation [9], they usually have better performances than methods in the first approach, particularly in applications such as noisy speech recognition [10].

However, a main shortcoming in some of the methods described above lies in their assumption that the background noise is stationary (noise statistics do not change in a given utterance). Based on this assumption, noise is often estimated from segmented noise-alone slices, for example, by voice-activity detection (VAD) [7]. Such an assumption may not hold in many real applications because the estimated noise may not be pertinent to noise in speech intervals in nonstationary environments.

Recently, methods have been proposed for speech enhancement in nonstationary noise. For example, in [11], a method based on sequential Monte Carlo method is applied to estimate time-varying autocorrelation coefficients of speech models for speech enhancement. This algorithm is more advanced in its assumption that autocorrelation coefficients of speech models are time varying. In fact, sequential Monte Carlo method is also applied to estimate noise parameters for robust speech recognition in nonstationary noise [12] through a nonlinear model [8], which was recently found to be effective for speech enhancement [13] as well.

The purpose of this paper is to present a method based on sequential Monte Carlo for estimation of noise parameter (time-varying mean vector of a noise model) with its application to speech enhancement and recognition. The method is based on a nonlinear function that models noise effects on speech [8, 12, 13]. Sequential Monte Carlo method generates particles of parameters (including speech and noise parameters) from a prior speech model that has been trained from a clean speech database. These particles approximate posterior distribution of speech and noise parameter sequences given the observed noisy speech sequence. Minimum mean square error (MMSE) estimation of the noise parameter is obtained from these particles. Once the noise parameter has been estimated, it is used in subtraction-type speech enhancement methods, for example, Wiener filter and perceptual filter,[1] and adaptation of speech mean vectors for speech recognition.

The remainder of the paper is organized as follows. The model specification and estimation objectives for the noise parameters are stated in Section 2. In Section 3, the sequential Monte Carlo method is developed to solve the noise parameter estimation problem. Section 4.3 demonstrates application of this method to speech recognition by modifying speech model parameters. Application to speech enhancement is shown in Section 4.4. Discussions and conclusions are presented in Section 5.

Notation

Sets are denoted as $\{\cdot,\cdot\}$. Vectors and sequences of vectors are denoted by uppercased letters. Time index is in the parenthesis of vectors. For example, a sequence $Y(1:T) = (Y(1)\ \ Y(2)\ \cdots\ Y(T))$ consists of vector $Y(t)$ at time t, where its ith element is $y_i(t)$. The distribution of the vector $Y(t)$ is $p(Y(t))$. Superscript T denotes transpose.

The symbol X (or x) is exclusively used for original speech and Y (or y) is used for noisy speech in testing environments. N (or n) is used to denote noise.

By default, observation (or feature) vectors are in log-spectral domain. Superscripts lin, l, c denote linear spectral domain, log-spectral domain, and cepstral domain. The symbol $*$ denotes convolution.

[1] A model for frequency masking [14, 15] is applied.

2. PROBLEM DEFINITION

2.1. Model definitions

Consider a clean speech signal $x(t)$ at time t that is corrupted by additive background noise $n(t)$.[2] In time domain, the received speech signal $y(t)$ can be written as

$$y(t) = x(t) + n(t). \tag{1}$$

Assume that the speech signal $x(t)$ and noise $n(t)$ are uncorrelated. Hence, the power spectrum of the input noisy signal is the summation of the power spectra of clean speech signal and those of the noise. The output at filter bank j can be described by $y_j^{\text{lin}}(t) = \sum_m b(m)|\sum_{l=0}^{L-1} v(l)y(t-l)e^{-j2\pi lm/L}|^2$, summing the power spectra of the windowed signal $v(t) * y(t)$ with length L at each frequency m with binning weight $b(m)$. $v(t)$ is a window function (usually a Hamming window) and $b(m)$ is a triangle window.[3] Similarly, we denote the filter bank output for clean speech signal $x(t)$ and noise $n(t)$ as $x_j^{\text{lin}}(t)$ and $n_j^{\text{lin}}(t)$ for jth filter bank, respectively. They are related as

$$y_j^{\text{lin}}(t) = x_j^{\text{lin}}(t) + n_j^{\text{lin}}(t), \tag{2}$$

where j is from 1 to J, and J is the number of filter banks.

The filter bank output exhibits a large variance. In order to achieve an accurate statistical model, in some applications, for example, speech recognition, logarithm compression of $y_j^{\text{lin}}(t)$ is used instead. The corresponding compressed power spectrum is called log-spectral power, which has the following relationship (derived in Appendix A) with noisy signal, clean speech signal, and noise:

$$y_j^l(t) = x_j^l(t) + \log\left(1 + \exp\left(n_j^l(t) - x_j^l(t)\right)\right). \tag{3}$$

The function is plotted in Figure 1. We observed that this function is convex and continuous. For noise log-spectral power $n_j^l(t)$ that is much smaller than clean speech log-spectral power $x_j^l(t)$, the function outputs $x_j^l(t)$. This shows that the function is not "sensitive" to noise log-spectral power that is much smaller than clean speech log-spectral power.[4]

We consider the vector for clean speech log-spectral power $X^l(t) = (x_1^l(t), \ldots, x_J^l(t))^T$. Suppose that the statistics of the log-spectral power sequence $X^l(1:T)$ can be modeled by a hidden Markov model (HMM) with output density at each state s_t $(1 \le s_t \le S)$ represented by mixtures of Gaussian $\sum_{k_t=1}^{M} \pi_{s_t k_t} \mathcal{N}(X^l(t); \mu_{s_t k_t}^l, \Sigma_{s_t k_t}^l)$, where M denotes the number

[2] Channel distortion and reverberation are not considered in this paper. In this paper, $x(t)$ can be considered as a speech signal received by a close-talking microphone, and $n(t)$ is the background noise picked up by the microphone.

[3] In Mel-scaled filter bank analysis [16], $b(m)$ is a triangle window centered in the Mel scale.

[4] We will discuss later in Sections 3.5 and 4.2 that such property may result in larger-than-necessary estimation of the noise log-spectral power.

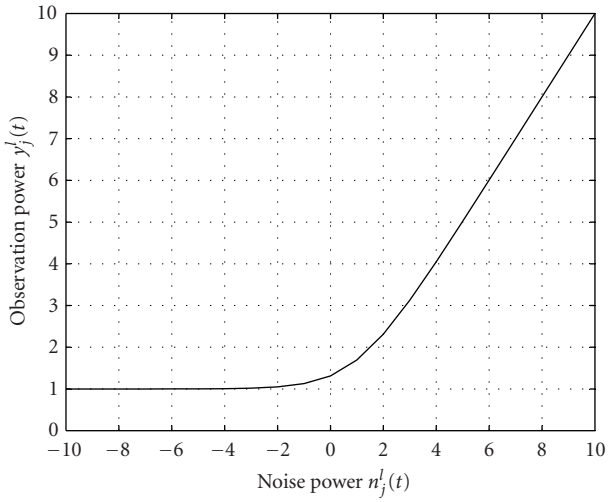


FIGURE 1: Plot of function $y_j^l(t) = x_j^l(t) + \log(1 + \exp(n_j^l(t) - x_j^l(t)))$. $x_j^l(t) = 1.0$; $n_j^l(t)$ ranges from -10.0 to 10.0.

of Gaussian densities in each state. To model the statistics of noise log-spectral power $N^l(1:T)$, we use a single Gaussian density with a time-varying mean vector $\mu_n^l(t)$ and a constant diagonal variance matrix V_n^l.

With the above-defined statistical models, we may plot the dependence among their parameters and observation sequence $Y^l(1:t)$ by a graphical model [17] in Figure 2. In this figure, the rectangular boxes correspond to discrete state/mixture indexes, and the round circles correspond to continuous-valued vectors. Shaded circles denote observed noisy speech log-spectral power.

The state $s_t \in \{1,\ldots,S\}$ gives the current state index at frame t. State sequence is a Markovian sequence with state transition probability $p(s_t|s_{t-1}) = a_{s_{t-1}s_t}$. At state s_t, an index $k_t \in \{1,\ldots,M\}$ assigns a Gaussian density $\mathcal{N}(\cdot;\mu_{s_tk_t}^l, \Sigma_{s_tk_t}^l)$ with prior probability $p(k_t|s_t) = \pi_{s_tk_t}$. Speech parameter $\mu_{s_tk_t}^l(t)$ is thus distributed in Gaussian given s_t and k_t; that is,

$$s_t \sim p(s_t|s_{t-1}) = a_{s_{t-1}s_t}, \tag{4}$$

$$k_t \sim p(k_t|s_t) = \pi_{s_tk_t}, \tag{5}$$

$$\mu_{s_tk_t}^l(t) \sim \mathcal{N}(\cdot;\mu_{s_tk_t}^l, \Sigma_{s_tk_t}^l). \tag{6}$$

Assuming that the variances of $X^l(t)$ and $N^l(t)$ are very small (as done in [8]) for each filter bank j, given s_t and k_t, we may relate the observed signal $Y^l(t)$ to speech mean vector $\mu_{s_tk_t}^l(t)$ and time-varying noise mean vector $\mu_n^l(t)$ with the function

$$Y^l(t) = \mu_{s_tk_t}^l(t) + \log\left(1 + \exp\left(\mu_n^l(t) - \mu_{s_tk_t}^l(t)\right)\right) + w_{s_tk_t}(t), \tag{7}$$

where $w_{s_tk_t}(t)$ is distributed in $\mathcal{N}(\cdot;0,\Sigma_{s_tk_t}^l)$, representing the possible modeling error and measurement noise in the above equation.

Furthermore, to model time-varying noise statistics, we assume that the noise parameter $\mu_n^l(t)$ follows a random walk function; that is,

$$\begin{aligned}\mu_n^l(t) &\sim p(\mu_n^l(t)|\mu_n^l(t-1)) \\ &= \mathcal{N}(\mu_n^l(t);\mu_n^l(t-1), V_n^l).\end{aligned} \tag{8}$$

We collectively denote these parameters $\{\mu_{s_tk_t}^l(t), s_t, k_t, \mu_n^l(t); \mu_{s_tk_t}^l(t) \in \mathbb{R}^J,\ 1 \le s_t \le S,\ 1 \le k_t \le M,\ \mu_n^l(t) \in \mathbb{R}^J\}$ as $\theta(t)$. It is clearly seen from (4)–(8) that they have the following prior distribution and likelihood at each time t:

$$\begin{aligned}&p(\theta(t)|\theta(t-1)) \\ &\quad = a_{s_{t-1}s_t}\pi_{s_tk_t} \\ &\qquad \times \mathcal{N}(\mu_{s_tk_t}^l(t);\mu_{s_tk_t}^l,\Sigma_{s_tk_t}^l)\,\mathcal{N}(\mu_n^l(t);\mu_n^l(t-1),V_n^l),\end{aligned} \tag{9}$$

$$\begin{aligned}&p(Y^l(t)|\theta(t)) \\ &\quad = \mathcal{N}(Y^l(t);\mu_{s_tk_t}^l(t) \\ &\qquad + \log\left(1 + \exp\left(\mu_n^l(t) - \mu_{s_tk_t}^l(t)\right)\right), \Sigma_{s_tk_t}^l).\end{aligned} \tag{10}$$

Remark 1. In comparison with the traditional HMM, the new model shown in Figure 2 may provide more robustness to contaminating noise, because it includes explicit modeling of the time-varying noise parameters. However, probabilistic inference in the new model can no longer be done by the efficient Viterbi algorithm [18].

2.2. Estimation objective

The objective of this method is to estimate, up to time t, a sequence of noise parameters $\mu_n^l(1:t)$ given the observed noisy speech log-spectral sequence $Y^l(1:t)$ and the above defined graphical model, in which speech models are trained from clean speech signals. Formally, $\mu_n^l(1:t)$ is calculated by the MMSE estimation

$$\hat{\mu}_n^l(1:t) = \int_{\mu_n^l(1:t)} \mu_n^l(1:t)\, p(\mu_n^l(1:t)|Y^l(1:t))\, d\mu_n^l(1:t), \tag{11}$$

where $p(\mu_n^l(1:t)|Y^l(1:t))$ is the posterior distribution of $\mu_n^l(1:t)$ given $Y^l(1:t)$.

Based on the graphical model shown in Figure 2, Bayesian estimation of the time-varying noise parameter $\mu_n^l(1:t)$ involves construction of a likelihood function of observation sequence $Y^l(1:t)$ given parameter sequence $\Theta(1:t) = (\theta(1),\ldots,\theta(t))$ and prior probability $p(\Theta(1:t))$ for $t = 1,\ldots,T$. The posterior distribution of $\Theta(1:t)$ given observation sequence $Y^l(1:t)$ is

$$p(\Theta(1:t)|Y^l(1:t)) \propto p(Y^l(1:t)|\Theta(1:t))\, p(\Theta(1:t)). \tag{12}$$

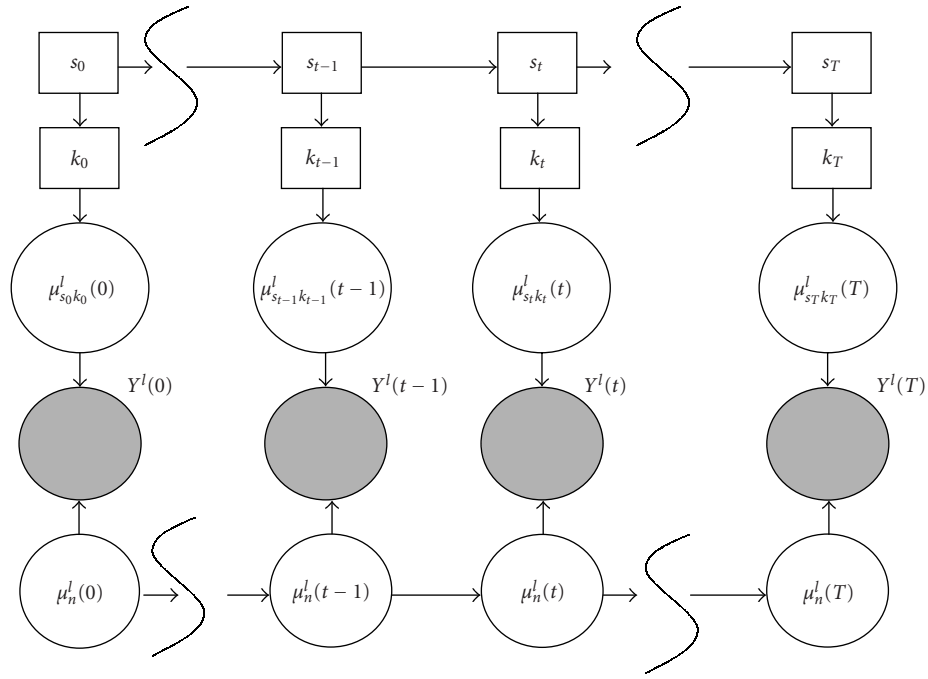


Figure 2: The graphical model representation of the dependence of the speech and noise model parameters. s_t and k_t denote the state and Gaussian mixture at frame t in speech model. $\mu^l_{s_tk_t}(t)$ and $\mu^l_n(t)$ denote the speech and noise parameters. $Y^l(t)$ is the observed noisy speech signal at frame t.

Due to the Markovian property shown in (9) and (10), the above posterior distribution can be written as

$$p(\Theta(1:t)|Y^l(1:t)) \propto \prod_{\tau=2}^{t} p(Y^l(\tau)|\theta(\tau))\,p(\theta(\tau)|\theta(\tau-1))\,p(Y^l(1)|\theta(1))\,p(\theta(1)). \tag{13}$$

Based on this posterior distribution, MMSE estimation in (11) can be achieved by

$$\begin{aligned}\hat{\mu}^l_n(1:t) = &\int_{\mu^l_{1:n}(1:t)} \mu^l_{1:n}(1:t) \\ &\times \sum_{s_{1:t},k_{1:t}} \int_{\mu^l_{s_{1:t}k_{1:t}}(1:t)} p(\Theta(1:t)|Y^l(1:t)) \\ &d\mu^l_{s_{1:t}k_{1:t}}(1:t)\,d\mu^l_n(1:t).\end{aligned} \tag{14}$$

Note that there are difficulties in evaluating the MMSE estimation. The first relates to the nonlinear function in (10), and the second arises from the unseen state sequence $s_{1:t}$ and mixture sequence $k_{1:t}$. These unseen sequences, together with nodes $\{\mu^l_{s_tk_t}(t)\}$, $\{Y^l(t)\}$, and $\{\mu^l_n(t)\}$, form loops in the graphical model. These loops in Figure 2 make exact inferences on posterior probabilities of unseen sequences $s_{1:t}$ and $k_{1:t}$, computationally intractable. In the following section, we devise a sequential Monte Carlo method to tackle these problems.

3. SEQUENTIAL MONTE CARLO METHOD FOR NOISE PARAMETER ESTIMATION

This section presents a sequential Monte Carlo method for estimating noise parameters from observed noisy signals and pretrained clean speech models. This method applies sequential Bayesian importance sampling (BIS) in order to generate particles of speech and noise parameters from a proposal distribution. These particles are selected according to their weights calculated with a function of their likelihood. It should be noted that the application here is one particular case of a more general sequential BIS method [19, 20].

3.1. Importance sampling

Suppose that there are N particles $\{\Theta^{(i)}(1:t);\ i = 1,\ldots,N\}$. Each particle is denoted as

$$\Theta^{(i)}(1:t) = \left\{s^{(i)}_{1:t}, k^{(i)}_{1:t}, \mu^{l(i)}_{s^{(i)}_{1:t}k^{(i)}_{1:t}}(1:t), \mu^{l(i)}_n(1:t)\right\}. \tag{15}$$

These particles are generated according to $p(\Theta(1:t)|Y^l(1:t))$. Then, these particles form an empirical distribution of $\Theta(1:t)$, given by

$$\tilde{p}_N(\Theta(1:t)|Y^l(1:t)) = \frac{1}{N}\sum_{i=1}^{N} \delta_{\Theta^{(i)}(1:t)}(d\Theta(1:t)), \tag{16}$$

where $\delta_x(\cdot)$ is the Dirac delta measure concentrated on x.

Using this distribution, an estimate of the parameters of interests $\bar{f}_\Theta(1:t)$ can be obtained by

$$\begin{aligned}\bar{f}_\Theta(1:t) &= \int f_\Theta(1:t)\bar{p}_N(\Theta(1:t)|Y^l(1:t))d\Theta(1:t)\\ &= \frac{1}{N}\sum_{i=1}^{N} f_\Theta^{(i)}(1:t),\end{aligned} \tag{17}$$

where, for example, function $f_\Theta(1:t)$ is $\Theta(1:t)$ and $f_\Theta^{(i)}(1:t) = \Theta^{(i)}(1:t)$ if $\bar{f}_\Theta(1:t)$ is used for estimating posterior mean of $\Theta(1:t)$. As the number of particles N goes to infinity, this estimate approaches the true estimate under mild conditions [21].

It is common to encounter the situation that the posterior distribution $p(\Theta(1:t)|Y^l(1:t))$ cannot be sampled directly. Alternatively, importance sampling (IS) method [22] implements the empirical estimate in (17) by sampling from an easier distribution $q(\Theta(1:t)|Y^l(1:t))$, whose support includes that of $p(\Theta(1:t)|Y^l(1:t))$; that is,

$$\begin{aligned}\bar{f}_\Theta(1:t) &= \int f_\Theta(1:t)\frac{p(\Theta(1:t)|Y^l(1:t))}{q(\Theta(1:t)|Y^l(1:t))}\\ &\quad\times q(\Theta(1:t)|Y^l(1:t))d\Theta(1:t)\\ &= \frac{\sum_{i=1}^{N} f_\Theta^{(i)}(1:t)w^{(i)}(1:t)}{\sum_{i=1}^{N} w^{(i)}(1:t)},\end{aligned} \tag{18}$$

where $\Theta^{(i)}(1:t)$ is sampled from distribution $q(\Theta(1:t)|Y^l(1:t))$, and each particle (i) has a weight given by

$$w^{(i)}(1:t) = \frac{p(\Theta^{(i)}(1:t)|Y^l(1:t))}{q(\Theta^{(i)}(1:t)|Y^l(1:t))}. \tag{19}$$

Equation (18) can be written as

$$\bar{f}_\Theta(1:t) = \sum_{i=1}^{N} f_\Theta^{(1:i)}(t)\tilde{w}^{(i)}(1:t), \tag{20}$$

where the normalized weight is given as $\tilde{w}^{(i)}(1:t) = w^{(i)}(1:t)/\sum_{j=1}^{N} w^{(j)}(1:t)$.

3.2. Sequential Bayesian importance sampling

Making use of the Markovian property in (13), we can have the following sequential BIS method to approximate the posterior distribution $p(\Theta(1:t)|Y^l(1:t))$. Basically, given an estimate of the posterior distribution at the previous time $t-1$, the method updates estimate of $p(\Theta(1:t)|Y^l(1:t))$ by combining a prediction step from a proposal sampling distribution in (24) and (25), and a sampling weight updating step in (26).

Suppose that a sequence of parameters $\hat{\Theta}(1:t-1)$ up to the previous time $t-1$ is given. By Markovian property in (13), the posterior distribution of $\Theta(1:t) = (\hat{\Theta}(1:t-1)\theta(t))$ given $Y^l(1:t)$ can be written as

$$\begin{aligned}p(\Theta(1:t)|Y^l(1:t)) &\propto p(Y^l(t)|\theta(t))p(\theta(t)|\hat{\theta}(t-1))\\ &\quad\times\prod_{\tau=2}^{t-1} p(Y^l(\tau)|\hat{\theta}(\tau))p(\hat{\theta}(\tau)|\hat{\theta}(\tau-1))\\ &\quad\times p(Y^l(1)|\hat{\theta}(1))p(\hat{\theta}(1)).\end{aligned} \tag{21}$$

We assume that the proposal distribution is in fact given as

$$\begin{aligned}&q(\Theta(1:t)|Y^l(1:t))\\ &\quad= q(Y^l(t)|\theta(t))q(\theta(t)|\hat{\theta}(t-1))\\ &\qquad\times\prod_{\tau=2}^{t-1} q(\hat{\theta}(\tau)|\hat{\theta}(\tau-1))q(Y^l(\tau)|\hat{\theta}(\tau))\\ &\qquad\times q(Y^l(1)|\hat{\theta}(1))q(\hat{\theta}(1)).\end{aligned} \tag{22}$$

Plugging (21) and (22) into (19), we can update weight in a recursive way; that is,

$$\begin{aligned}w^{(i)}(1:t) &= \frac{p(Y^l(t)|\theta^{(i)}(t))p(\theta^{(i)}(t)|\hat{\theta}^{(i)}(t-1))}{q(Y^l(t)|\theta^{(i)}(t))q(\theta^{(i)}(t)|\hat{\theta}^{(i)}(t-1))}\\ &\quad\times\frac{\prod_{\tau=2}^{t-1} p(\hat{\theta}^{(i)}(\tau)|\hat{\theta}^{(i)}(\tau-1))p(Y^l(\tau)|\hat{\theta}^{(i)}(\tau))}{\prod_{\tau=2}^{t-1} q(\hat{\theta}^{(i)}(\tau)|\hat{\theta}^{(i)}(\tau-1))q(Y^l(\tau)|\hat{\theta}^{(i)}(\tau))}\\ &\quad\times\frac{p(Y^l(1)|\hat{\theta}^{(i)}(1))p(\hat{\theta}^{(i)}(1))}{q(Y^l(1)|\hat{\theta}^{(i)}(1))q(\hat{\theta}^{(i)}(1))}\\ &= w^{(i)}(1:t-1)\frac{p(Y^l(t)|\theta^{(i)}(t))p(\theta^{(i)}(t)|\hat{\theta}^{(i)}(t-1))}{q(Y^l(t)|\theta^{(i)}(t))q(\theta^{(i)}(t)|\hat{\theta}^{(i)}(t-1))}.\end{aligned} \tag{23}$$

Such a time-recursive evaluation of weights can be further simplified by allowing proposal distribution to be the prior distribution of the parameters. In this paper, the proposal distribution is given as

$$q(Y^l(t)|\theta^{(i)}(t)) = 1, \tag{24}$$

$$q(\theta^{(i)}(t)|\hat{\theta}^{(i)}(t-1)) = a_{s_{t-1}^{(i)}s_t^{(i)}}\pi_{s_t^{(i)}k_t^{(i)}}\mathcal{N}\left(\mu^{l(i)}_{s_t^{(i)}k_t^{(i)}}(t);\mu^{l}_{s_t^{(i)}k_t^{(i)}},\Sigma^{l}_{s_t^{(i)}k_t^{(i)}}\right). \tag{25}$$

Consequently, the above weight is updated by

$$w^{(i)}(t) \propto w^{(i)}(t-1)p(Y^l(t)|\theta^{(i)}(t))p(\mu_n^{l(i)}(t)|\hat{\mu}_n^{l(i)}(t-1)). \tag{26}$$

Remark 2. Given $\hat{\Theta}(1:t-1)$, there is an optimal proposal distribution that minimizes variance of the importance weights. This optimal proposal distribution is in fact the posterior distribution $p(\theta(t)|\hat{\Theta}(1:t-1), Y^l(1:t))$ [23, 24].

3.3. Rao-Blackwellization and the extended Kalman filter

Note that $\mu_n^{l(i)}(t)$ in particle (i) is assumed to be distributed in $\mathcal{N}(\mu_n^{l(i)}(t); \mu_n^{l(i)}(t-1), V_n^l)$. By the Rao-Blackwell theorem [25], the variance of weight in (26) can be reduced by marginalizing out $\mu_n^{l(i)}(t)$. Therefore, we have

$$w^{(i)}(t) \propto w^{(i)}(t-1) \times \int_{\mu_n^{l(i)}(t)} p\left(Y^l(t)|\theta^{(i)}(t)\right) \times p\left(\mu_n^{l(i)}(t)|\hat{\mu}_n^{l(i)}(t-1)\right) d\mu_n^{l(i)}(t). \tag{27}$$

Referring to (9) and (10), we notice that the integrand $p(Y^l(t)|\theta^{(i)}(t))p(\mu_n^{l(i)}(t)|\hat{\mu}_n^{l(i)}(t-1))$ is a state-space model by (7) and (8). In this state-space model, given $s_t^{(i)}$, $k_t^{(i)}$, and $\mu_{s_t^{(i)}k_t^{(i)}}^{l(i)}(t)$, $\mu_n^{l(i)}(t)$ is the hidden continuous-valued vector distributed in $\mathcal{N}(\mu_n^{l(i)}(t); \hat{\mu}_n^{l(i)}(t-1), V_n^l)$, and $Y^l(t)$ is the observed signal of this model. This integral in (27) can be analytically obtained if we linearize (7) with respect to $\mu_n^{l(i)}(t)$. The linearized state-space model provides an extended Kalman filter (EKF) (see Appendix B for the detail of EKF), and the integral is $p(Y^l(t)|s_t^{(i)}, k_t^{(i)}, \mu_{s_t^{(i)}k_t^{(i)}}^{l(i)}(t), \hat{\mu}_n^{l(i)}(t-1), Y^l(t-1))$, which is the predictive likelihood shown in (B.1). An advantage of updating weight by (27) is its simplicity of implementation.

Because the predictive likelihood is obtained from EKF, the weight $w^{(i)}(t)$ may not asymptotically approach the target posterior distribution. One way to achieve asymptotically the target posterior distribution may follow a method called the *extended Kalman particle filter* in [26], where the weight is updated by

$$w^{(i)}(t) \propto w^{(i)}(t-1) \frac{p\left(Y^l(t)|\theta^{(i)}(t)\right) p\left(\mu_n^{l(i)}(t)|\hat{\mu}_n^{l(i)}(t-1)\right)}{q\left(\mu_n^{l(i)}(t)|\hat{\mu}_n^{l(i)}(t-1), s_t^{(i)}, k_t^{(i)}, \mu_{s_t^{(i)}k_t^{(i)}}^{l(i)}(t), Y^l(t)\right)}, \tag{28}$$

and the proposal distribution for $\mu_n^{l(i)}(t)$ is from the posterior distribution of $\mu_n^{l(i)}(t)$ by EKF; that is,

$$q\left(\mu_n^{l(i)}(t)|\hat{\mu}_n^{l(i)}(t-1), s_t^{(i)}, k_t^{(i)}, \mu_{s_t^{(i)}k_t^{(i)}}^{l(i)}(t), Y^l(t)\right) = \mathcal{N}\left(\mu_n^{l(i)}(t); \mu_n^{l(i)}(t-1) + G^{(i)}(t)\alpha^{(i)}(t-1), K^{(i)}(t)\right), \tag{29}$$

where Kalman gain $G^{(i)}(t)$, innovation vector $\alpha^{(i)}(t-1)$, and posterior variance $K^{(i)}(t)$ are respectively given in (B.7), (B.2), and (B.4).

However, for the following reasons, we did not apply the stricter *extended Kalman particle filter* to our problem. First, the scheme in (28) is not Rao-Blackwellized. The variance of sampling weights might be larger than the Rao-Blackwellized method in (27). Second, although observation function (7) is nonlinear, it is convex and continuous. Therefore, linearization of (7) with respect to $\mu_n^l(t)$ may not affect the mode of the posterior distribution $p(\mu_n^l(1:t)|Y^l(1:t))$. By the asymptotic theory (see [25, page 430]), under the mild condition that the variance of noise $N^l(t)$ (parameterized by V_n^l) is finite, bias for estimating $\hat{\mu}_n^l(t)$ by MMSE estimation via (17) with weight given by (27) may be reduced as the number of particles N grows large. (However, unbiasedness for estimating $\hat{\mu}_n^l(t)$ may not be established since there are zero derivatives with respect to the parameter $\mu_n^l(t)$ in (7).) Third, evaluation of (28) is computationally more expensive than (27), because (28) involves calculation processes on two state-space models. We will show some experiments in Section 4.1 to support the above considerations.

Remark 3. Working in linear spectral domain in (2) for noise estimation does not require EKF. Thus, if the noise parameter in $\Theta(t)$ and the observations are both in the linear spectral domain, the corresponding sequential BIS can achieve asymptotically the target posterior distribution (12). In practice, however, due to the large variance in the linear spectral domain, we may frequently encounter numerical problems that make it difficult to build an accurate statistical model for both clean speech and noise. Compressing linear spectral power into log-spectral domain is commonly used in speech recognition to achieve more accurate models. Furthermore, because the performance by adapting acoustic models (modifying mean and variance of acoustic models) is usually higher than enhanced noisy speech signals for noisy speech recognition [10], in the context of speech recognition, it is beneficial to devise an algorithm that works in the domain for building acoustic models. In our examples, acoustic models are trained from cepstral or log-spectral features, thus, the parameter estimation algorithm is devised in the log-spectral domain, which is linearly related to the cepstral domain. We will show later that the estimated noise parameter $\hat{\mu}_n^l(t)$ substitutes $\hat{\mu}_n^l$ using a log-add method (36) to adapt acoustic model mean vectors. Thus, to avoid inconsistency due to transformations between different domains, the noise parameter may be estimated in log-spectral domain, instead of linear spectral domain.

3.4. Avoiding degeneracy by resampling

Since the above particles are discrete approximations of the posterior distribution $p(\Theta(1:t)|Y^l(1:t))$, in practice, after several steps of sequential BIS, the weights of not all but some particles may become insignificant. This could cause a large variance in the estimate. In addition, it is not necessary to compute particles with insignificant weights. Selection of the particles is thus necessary to reduce the variance and to make efficient use of computational resources.

Many methods for selecting particles have been proposed, including sampling-importance resampling (SIR) [27], residual resampling [28], and so forth. We apply residual resampling for its computational simplicity. This method basically avoids degeneracy by discarding those particles with insignificant weights, and in order to keep the number of the

particles constant, particles with significant weights are duplicated. The steps are as follows. Firstly, set $\tilde{N}^{(i)} = \lfloor N\tilde{w}^{(i)}(1:t)\rfloor$. Secondly, select the remaining $\tilde{N} = N - \sum_{i=1}^{N}\tilde{N}^{(i)}$ particles with new weights $\acute{w}^{(i)}(1:t) = \tilde{N}^{-1}(\tilde{w}^{(i)}(1:t)N - \tilde{N}^{(i)})$, and obtain particles by sampling in a distribution approximated by these new weights. Finally, add the particles to those obtained in the first step. After this residual sampling step, the weight for each particle is $1/N$. Besides computational simplicity, residual resampling is known to have smaller variance $\operatorname{var} N^{(i)} = \tilde{N}\acute{w}^{(i)}(1:t)(1-\acute{w}^{(i)}(1:t))$ compared to that of SIR (which is $\operatorname{var} N^{(i)}(t) = N\tilde{w}^{(i)}(1:t)(1-\tilde{w}^{(i)}(1:t))$). We denote the particles after the selection step as $\{\tilde{\Theta}^{(i)}(1:t);\ i = 1\cdots N\}$.

After the selection step, the discrete nature of the approximation may lead to large bias/variance, of which the extreme case is that all the particles have the same parameters estimated. Therefore, it is necessary to introduce a resampling step to avoid such degeneracy. We apply a Metropolis-Hastings smoothing [19] step in each particle by sampling a candidate parameter given the currently estimated parameter according to the proposal distribution $q(\theta^\star(t)|\tilde{\theta}^{(i)}(t))$. For each particle, a value is calculated as

$$g^{(i)}(t) = g_1^{(i)}(t)g_2^{(i)}(t), \tag{30}$$

where $g_1^{(i)}(t) = p((\tilde{\Theta}^{(i)}(t-1)\theta^\star(t))|Y^l(1:t))/p(\tilde{\Theta}^{(i)}(1:t)|Y^l(1:t))$ and $g_2^{(i)}(t) = q(\tilde{\theta}^{(i)}(t)|\theta^\star(t))/q(\theta^\star(t)|\tilde{\theta}^{(i)}(t))$. Within an acceptance possibility $\min\{1, g^{(i)}(t)\}$, the Markov chain then moves towards the new parameter $\theta^\star(t)$; otherwise, it remains at the original parameter.

To simplify calculations, we assume that the proposal distribution $q(\theta^\star(t)|\tilde{\theta}^{(i)}(t))$ is symmetric.[5] Note that $p(\tilde{\Theta}^{(i)}(1:t)|Y^l(1:t))$ is proportional to $\tilde{w}^{(i)}(1:t)$ up to a scalar factor. With (27), (B.1), and $\tilde{w}^{(i)}(1:t-1) = 1/N$, we can obtain the acceptance possibility as

$$\min\left\{1, \frac{p\left(Y^l(t)|s_t^{\star(i)}, k_t^{\star(i)}, \mu^{l(i)}_{s_t^{\star(i)}k_t^{\star(i)}}(t), \hat{\mu}_n^{l(i)}(t-1), Y^l(t-1)\right)}{p\left(Y^l(t)|\tilde{s}_t^{(i)}, \tilde{k}_t^{(i)}, \tilde{\mu}^{l(i)}_{\tilde{s}_t^{(i)}\tilde{k}_t^{(i)}}(t), \hat{\mu}_n^{l(i)}(t-1), Y^l(t-1)\right)}\right\}. \tag{31}$$

Denote the obtained particles hereafter as $\{\hat{\Theta}^{(i)}(1:t);\ i = 1,\ldots,N\}$ with equal weights.

3.5. Noise parameter estimation via the sequential Monte Carlo method

Following the above considerations, we present the implemented algorithm for noise parameter estimation. Given that, at time $t-1$, N particles $\{\hat{\Theta}^{(i)}(1:t-1);\ i = 1,\ldots,N\}$ are distributed approximately according to $p(\Theta(1:t-1)|Y^l(1:t-1))$, the sequential Monte Carlo method proceeds as follows at time t.

Algorithm 1.

Bayesian importance sampling step

(1) Sampling. For $i = 1,\ldots,N$, sample a proposal $\hat{\Theta}^{(i)}(1:t) = (\hat{\Theta}^{(i)}(1:t-1)\hat{\theta}^{(i)}(t))$ by
   (a) sampling $\hat{s}_t^{(i)} \sim a_{s_{t-1}^{(i)}s_t}$;
   (b) sampling $\hat{k}_t^{(i)} \sim \pi_{\hat{s}_t^{(i)}k_t}$;
   (c) sampling $\hat{\mu}^{l(i)}_{\hat{s}_t^{(i)}\hat{k}_t^{(i)}}(t) \sim \mathcal{N}(\mu^l_{\hat{s}_t^{(i)}\hat{k}_t^{(i)}}(t); \mu^l_{\hat{s}_t^{(i)}\hat{k}_t^{(i)}}, \Sigma^l_{\hat{s}_t^{(i)}\hat{k}_t^{(i)}})$.

(2) Extended Kalman prediction. For $i = 1,\ldots,N$, evaluate (B.2)–(B.7) for each particle by EKFs. Predict noise parameter for each particle by

$$\hat{\mu}_n^{l(i)}(t) = \hat{\mu}_n^{l(i)}(t|t-1), \tag{32}$$

where $\hat{\mu}_n^{l(i)}(t|t-1)$ is given in (B.3).

(3) Weighting. For $i = 1,\ldots,N$, evaluate the weight of each particle $\hat{\Theta}^{(i)}$ by

$$\hat{w}^{(i)}(1:t) \propto \hat{w}^{(i)}(1:t-1)p\Big(Y^l(t)|\hat{s}_t^{(i)}, \hat{k}_t^{(i)}, \hat{\mu}^{l(i)}_{\hat{s}_t^{(i)}(t)\hat{k}_t^{(i)}}(t), \hat{\mu}_n^{l(i)}(t-1), Y^l(t-1)\Big), \tag{33}$$

where the second term in the right-hand side of the equation is the predictive likelihood, given in (B.1), of the EKF.

(4) Normalization. For $i = 1,\ldots,N$, the weight of the ith particle is normalized by

$$\tilde{w}^{(i)}(1:t) = \frac{\hat{w}^{(i)}(1:t)}{\sum_{i=1}^{N}\hat{w}^{(i)}(1:t)}. \tag{34}$$

Resampling

(1) Selection. Use residual resampling to select particles with larger normalized weights and discard those particles with insignificant weights. Duplicate particles of large weights in order to keep the number of particles as N. Denote the set of particles after the selection step as $\{\tilde{\Theta}^{(i)}(1:t);\ i = 1,\ldots,N\}$. These particles have equal weights $\tilde{w}^{(i)}(1:t) = 1/N$.

(2) Metropolis-Hastings smoothing. For $i = 1,\ldots,N$, sample $\Theta^{\star(i)}(1:t) = (\tilde{\Theta}^{(i)}(1:t-1)\theta^\star(t))$ from step (1) to step (3) in the Bayesian importance sampling step with starting parameters given by $\tilde{\Theta}^{(i)}(1:t)$. For $i = 1,\ldots,N$, set an acceptance possibility by (31). For $i = 1,\ldots,N$, accept $\Theta^{\star(i)}(1:t)$ (i.e., substitute $\tilde{\Theta}^{(i)}(1:t)$ by $\Theta^{\star(i)}(1:t)$) with probability $r^{(i)}(t) \sim U(0,1)$. The particles after the step are $\{\hat{\Theta}^{(i)}(1:t);\ i = 1,\ldots,N\}$ with equal weights $\hat{w}^{(i)}(1:t) = 1/N$.

[5] Generating $\theta^\star(t)$ involves sampling speech state $s_t^\star$ from $\tilde{s}_{1:t}^{(i)}$ according to a first-order Markovian transition probability $p(s_t^\star|\tilde{s}_t^{(i)})$ in the graphical model in Figure 2. Usually, this transition probability matrix is not symmetric; that is, $p(s_t^\star|\tilde{s}_t^{(i)}) \neq p(\tilde{s}_t^{(i)}|s_t^\star)$. Our assumption of symmetric proposal distribution $q(\theta^\star(t)|\tilde{\theta}^{(i)}(t))$ is for simplicity in calculating an acceptance possibility.

Table 1: State estimation experiment results. The results show the mean and variance of the mean squared error (MSE) calculated over 100 independent runs.

Algorithm	MSE		Averaged execution time (s)
	Mean	Variance	
Particle filter	8.713	49.012	5.338
Extended Kalman particle filter	6.496	34.899	13.439
Rao-Blackwellized particle filter	4.559	8.096	6.810

Noise parameter estimation

(1) Noise Parameter Estimation. With the above generated particles at each time t, estimation of the noise parameter $\mu_n^l(t)$ may be acquired by MMSE. Since each particle has the same weight, MMSE estimation of $\hat{\mu}_n^l(t)$ can be easily carried out as

$$\hat{\mu}_n^l(t) = \frac{1}{N}\sum_{i=1}^{N}\hat{\mu}_n^{l(i)}(t). \tag{35}$$

The computational complexity of the algorithm at each time t is $O(2N)$ and is roughly equivalent to $2N$ EKFs. These steps are highly parallel, and if resources permit, can be implemented in a parallel way. Since the sampling is based on BIS, the storage required for the calculation does not change over time. Thus the computation is efficient and fast.

Note that the estimated $\hat{\mu}_n^l(t)$ may be biased from the true physical mean vector for log-spectral noise power $N^l(t)$, because the function plotted in Figure 1 has zero derivative with respect to $n_j^l(t)$ in regions where $n_j^l(t)$ is much smaller than $x_j^l(t)$. For those $\hat{\mu}_n^{l(i)}(t)$ which are initialized with values larger than speech mean vector $\mu^{l(i)}_{s_t^{(i)}k_t^{(i)}}$, updating by EKF may be lower bounded around the speech mean vector. As a result, the updated $\hat{\mu}_n^l(t) = 1/N\sum_{i=1}^{N}\hat{\mu}_n^{l(i)}(t)$ may not be the true noise log-spectral power.

Remark 4. The above problem, however, may not hurt a model-based noisy speech recognition system, since it is the modified likelihood in (10) that is used to decode speech signals.[6] But in a speech enhancement system, noisy speech spectrum is directly processed on the estimated noise parameter. Therefore, biased estimation of the noise parameter may hurt performances more apparently than in a speech recognition system.

4. EXPERIMENTS

We first conducted synthetic experiments in Section 4.1 to compare three types of particle filters presented in Sections 3.2 and 3.3. Then, in the following sections, we present applications of the above noise parameter estimation method based on Rao-Blackwellized particle filter (27). We consider particularly difficult tasks for speech processing, speech enhancement, and noisy speech recognition in nonstationary noisy environments. We show in Section 4.2 that the method can track noise dynamically. In Section 4.3, we show that the method improves system robustness to noise in an ASR system. Finally, we present results on speech enhancement in Section 4.4, where the estimated noise parameter is used in a time-varying linear filter to reduce noise power.

4.1. Synthetic experiments

This section[7] presents some experiments[8] to show the validity of Rao-Blackwellized filter applied to the state-space model in (7) and (8). A sequence of $\mu_n^l(1:t)$ was generated from (8), where state-process noise variance V_n^l was set to 0.75. Speech mean vector $\mu^l_{s_tk_t}(t)$ in (7) was set to a constant 10. The observation noise variance $\Sigma^l_{s_tk_t}$ was set to 0.00005. Given only the noisy observation $Y^l(1:t)$ for $t = 1,\ldots,60$, different filters (particle filter by (26), extended Kalman particle filter by (28), and Rao-Blackwellized particle filter by (27)) were used to estimate the underlying state sequence $\mu_n^l(1:t)$. The number of particles in each type of filter was 200, and all the filters applied residual resampling [28]. The experiments were repeated for 100 times with random reinitialization of $\mu_n^l(1)$ for each run. Table 1 summarizes the mean and variance of the MSE of the state estimates, together with the averaged execution time of each filter. Figure 3 compares the estimates generated from a single run of the different filters. In terms of MSE, the extended Kalman particle filter performed better than the particle filter. However, the execution time of the extended Kalman particle filter was the longest (more than two times longer than that of particle filter (26)). Performance of the Rao-Blackwellized particle filter of (27) is clearly the best in terms of MSE. Notice that its averaged execution time was comparable to that of particle filter.

4.2. Estimation of noise parameter

Experiments were performed on the TI-Digits database downsampled to 16 kHz. Five hundred clean speech utterances from 15 speakers and 111 utterances unseen in the training set were used for training and testing, respectively.

[6] The likelihood of the observed signal $Y^l(t)$, given speech model parameter and a noise parameter, is the same as long as the noise parameter is much smaller than the speech parameter $\mu^{l(i)}_{s_t^{(i)}k_t^{(i)}}(t)$.

[7] A Matlab implementation of the synthetic experiments is available by sending email to the corresponding author.

[8] All variables in these experiments are one dimensional.

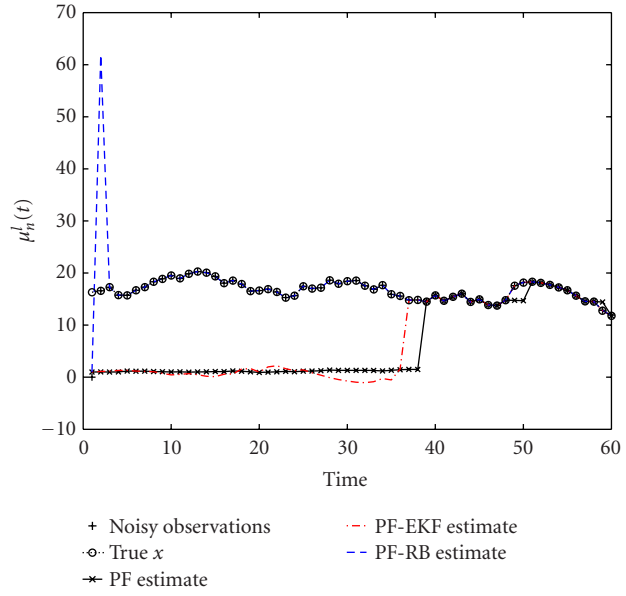


Figure 3: Plot of estimates generated by the different filters on the synthetic state estimation experiment versus true state. PF denotes particle filter by (26). PF-EKF denotes particle filter with EKF proposal sampling by (28). PF-RB denotes Rao-Blackwellized particle filter by (27).

Digits and silence were respectively modeled by 10-state and 3-state whole-word HMMs with 4 diagonal Gaussian mixtures in each state.

The window size was 25.0 milliseconds with a 10.0 milliseconds shift. Twenty-six filter banks were used in the binning stage; that is, $J = 26$. Speech feature vectors were Mel-scaled frequency cepstral coefficients (MFCCs), which were generated by transforming log-spectral power spectra vector with discrete Cosine transform (DCT). The baseline system had 98.7% word accuracy for speech recognition under clean conditions.

For testing, white noise signal was multiplied by a chirp signal and a rectangular signal in the time domain. The time-varying mean of the noise power as a result changed either continuously, denoted as experiment A, or dramatically, denoted as experiment B. SNR of the noisy speech ranged from 0 dB to 20.4 dB. We plotted the noise power in the 12th filter bank versus frames in Figure 4, together with the estimated noise power by the sequential method with the number of particles N set to 120 and the environment driving noise variance V_n^l set to 0.0001. As a comparison, we also plotted in Figure 5 the noise power and its estimate by the method with the same number of particles but larger driving noise variance set to 0.001.

Four seconds of contaminating noise were used to initialize $\hat{\mu}_n^l(0)$ in the noise estimation method. Initial value $\hat{\mu}_n^{l(i)}(0)$ of each particle was obtained by sampling from $\mathcal{N}(\hat{\mu}_n^l(0) + \zeta(0), 10.0)$, where $\zeta(0)$ was distributed in $U(-1.0, 9.0)$. To apply the estimation algorithm in Section 3.5, observation vectors were transformed into log-spectral domain.

Based on the results in Figures 4 and 5, we make the following observations. First, the method can track the evolution of the noise power. Second, the larger driving noise variance V_n^l will make faster convergence but larger estimation error. Third, as discussed in Section 3.5, there was large bias in the region where noise power changed from large to small. Such observation was more explicit in experiment B (noise multiplied with a rectangular signal).

4.3. Noisy speech recognition in time-varying noise

The experiment setup was the same as in the previous experiments in Section 4.2. Features for speech recognition were MFCCs plus their first- and second-order time differentials. Here, we compared three systems. The first was the baseline trained on clean speech without noise compensation (denoted as Baseline). The second was the system with noise compensation, which transformed clean speech acoustic models by mapping clean speech mean vector $\mu_{s_t k_t}^l$ at each state s_t and Gaussian density k_t with the function [8]

$$\hat{\mu}_{s_t k_t}^l = \mu_{s_t k_t}^l + \log\left(1 + \exp\left(\hat{\mu}_n^l - \mu_{s_t k_t}^l\right)\right), \tag{36}$$

where $\hat{\mu}_n^l$ was obtained by averaging noise log-spectral in noise-alone segments in the testing set. This system was denoted as stationary noise assumption (SNA). The third system used the method in Section 3.5 to estimate the noise parameter $\hat{\mu}_n^l(t)$ without training transcript. The estimated noise parameter was plugged into $\hat{\mu}_n^l$ in (36) for adapting acoustic mean vector at each time t. This system was denoted according to the number of particles and variance of the environment driving noise V_n^l.

4.3.1. Results in the simulated nonstationary noise

In terms of recognition performance in the simulated nonstationary noise described in Section 4.2, Table 2 shows that the method can effectively improve system robustness to the time-varying noise. For example, with 60 particles and the environment driving noise variance V_n^l set to 0.001, the method improved word accuracy from 75.3%, achieved by SNA, to 94.3% in experiment A. The table also shows that the word accuracies can be improved by increasing the number of particles. For example, given driving noise variance V_n^l set to 0.0001, increasing the number of particles from 60 to 120 could improve word accuracy from 77.1% to 85.8% in experiment B.

4.3.2. Speech recognition in real noise

In this experiment, speech signals were contaminated by highly nonstationary machine gun noise in different SNRs. The number of particles was set to 120, and the environment driving noise variance V_n^l was set to 0.0001. Recognition performances are shown in Table 3, together with Baseline and SNA. It is observed that, in all SNR conditions, the method in Section 3.5 further improved system performances in comparison with SNA. For example, in 8.9 dB SNR, the method improved word accuracy from 75.6% by SNA to 83.1%. As a whole, it reduced the word error rate by 39.9% more than SNA.

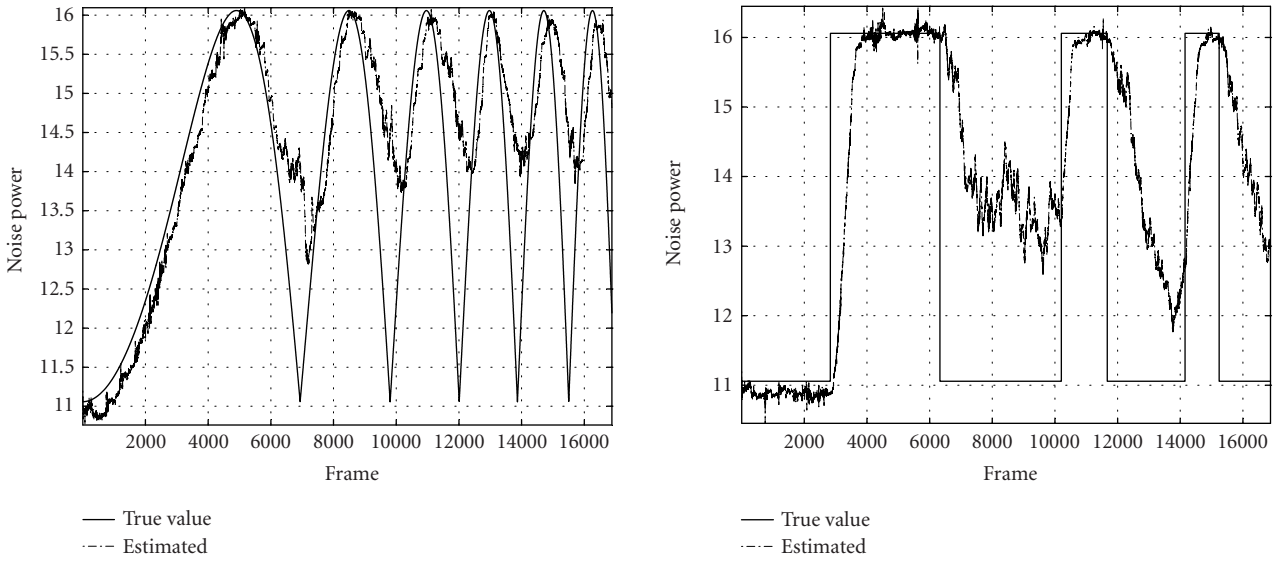


Figure 4: Estimation of the time-varying parameter $\mu_n^l(t)$ by the sequential Monte Carlo method at the 12th filter bank in experiment A. The number of particles is 120. The environment driving noise variance is 0.0001. The solid curve is the true noise power, whereas the dash-dotted curve is the estimated noise power.

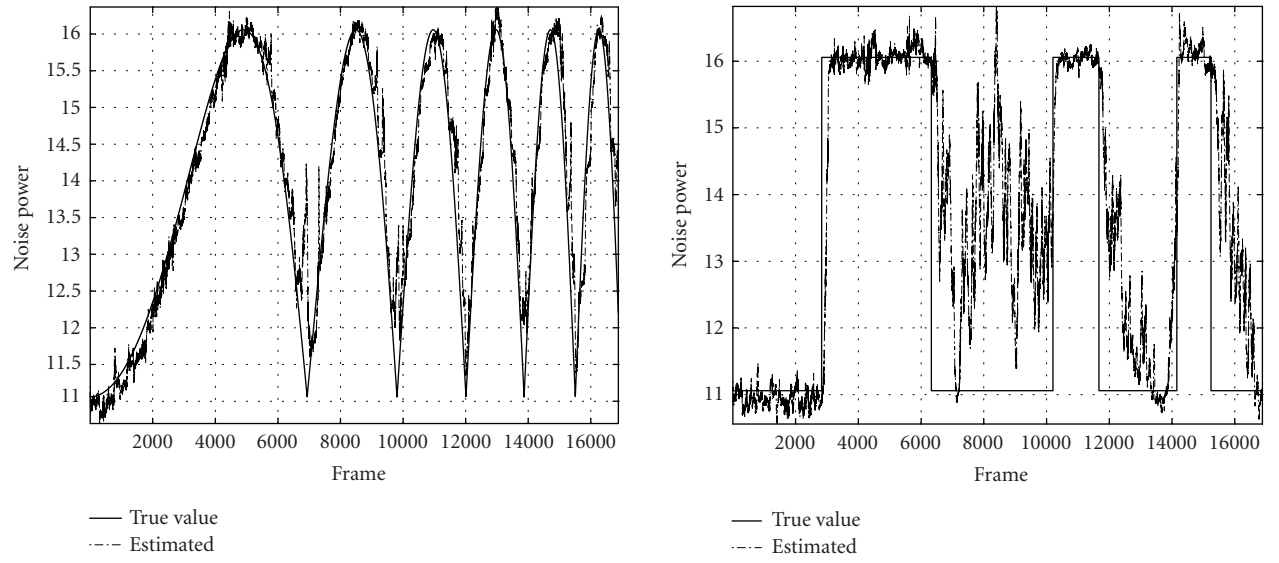


Figure 5: Estimation of the time-varying parameter $\mu_n^l(t)$ by the sequential Monte Carlo method at the 12th filter bank in experiment A. The number of particles is 120. The environment driving noise variance is 0.001. The solid curve is the true noise power, whereas the dash-dotted curve is the estimated noise power.

4.4. Perceptual speech enhancement

Enhanced speech $\hat{x}(t)$ is obtained by filtering the noisy speech sequence $y(t)$ via a time-varying linear filter $h(t)$; that is,

$$\hat{x}(t) = h(t) * y(t). \tag{37}$$

This process can be studied in the frequency domain as multiplication of the noisy speech power spectrum $y_j^{\text{lin}}(t)$ by a time-varying linear coefficient at each filter bank; that is,

$$\hat{x}_j^{\text{lin}}(t) = h_j(t) \cdot y_j^{\text{lin}}(t), \tag{38}$$

where $h_j(t)$ is the gain at filter bank j at time t. Referring to (2), we can expand it as

$$\hat{x}_j^{\text{lin}}(t) = h_j(t)x_j^{\text{lin}}(t) + h_j(t)n_j^{\text{lin}}(t). \tag{39}$$

We are left with two choices for linear time-varying filters.

Table 2: Word accuracy (%) in simulated nonstationary noise, achieved by the sequential Monte Carlo method in comparison with baseline without noise compensation, denoted as Baseline, and noise compensation assuming stationary noise, denoted as stationary noise assumption.

Experiment	Baseline	Stationary noise assumption	No. of particles = 60		No. of particles = 120	
			V_n^l		V_n^l	
			0.001	0.0001	0.001	0.0001
A	48.2	75.3	94.3	94.0	94.3	94.6
B	53.0	78.0	82.2	77.1	85.8	85.8

Table 3: Word accuracy (%) in machine gun noise, achieved by the sequential Monte Carlo method in comparison with baseline without noise compensation, denoted as Baseline, and noise compensation assuming stationary noise, denoted as stationary noise assumption.

SNR (dB)	Baseline	Stationary noise assumption	No. of particles = 120, $V_n^l = 0.0001$
28.9	90.4	92.8	97.6
14.9	64.5	76.8	88.3
8.9	56.0	75.6	83.1
1.6	50.0	69.0	72.9

(1) Wiener filter constructs the coefficient as

$$h_j(t) = 1 - \frac{\hat{n}_j^{\text{lin}}(t)}{y_j^{\text{lin}}(t)}, \tag{40}$$

where $\hat{n}_j^{\text{lin}}(t)$ is the estimate of noise power spectrum.

(2) The criterion for perceptual filter is to construct $h_j(t)$ so that the amplitude of the filtered noise power spectra $h_j(t) \cdot n_j^{\text{lin}}(t)$ is below the masking threshold of the denoised speech; that is,

$$h_j(t) \cdot n_j^{\text{lin}}(t) \le T_j(t), \tag{41}$$

where $T_j(t)$ is the masking threshold of the denoised speech signal. The threshold is a function of clean speech spectrum $x_j^{\text{lin}}(t)$. Since $x_j^{\text{lin}}(t)$ is not directly observed, the following equation is used instead, which makes the masking threshold a function of the estimated noise power spectra $\hat{n}_j^{\text{lin}}(t)$:

$$\hat{x}_j^{\text{lin}}(t) = y_j^{\text{lin}}(t) - \hat{n}_j^{\text{lin}}(t). \tag{42}$$

The perceptual filter exploits the masking properties of the human auditory system, and it has been employed by many researchers (e.g., [14]) in order to provide improved performance over the Wiener filter in low SNR conditions. Masking occurs because the auditory system is incapable of distinguishing two signals close in time or frequency domain. This is manifested by an evaluation of the minimum threshold of audibility due to a masker signal. Masking has been widely applied to speech and audio coding [15]. We consider frequency masking [15] when a weak signal is made inaudible by a stronger signal occurring simultaneously.

Both Wiener filter and perceptual filter require the estimated noise power spectrum $\hat{n}_j^{\text{lin}}(t)$. Under the assumption of stationary noise, the noise power spectrum can be estimated from noise-alone segments provided by explicit VAD, for example, speech enhancement scheme in [7]. However, in real applications, we encounter time-varying noise, which may change its statistics during speech utterances.

The objective of this section is to test the above devised method in Section 3.5 for speech enhancement in time-varying noise. The estimated $\hat{\mu}_n^l(t)$ is converted to linear spectral domain $\hat{\mu}_n^{\text{lin}}(t)$ by exponential operation. Corresponding jth element in $\hat{\mu}_n^{\text{lin}}(t)$ substitutes $\hat{n}_j^{\text{lin}}(t)$ in (40) and (42), respectively, to construct Wiener filter and perceptual filter. Therefore, the proposed speech enhancement algorithm is a combination of sequential noise parameter estimation by sequential Monte Carlo method and speech enhancement method with time-varying linear filtering. Diagram of the algorithm is shown in Figure 6. At each frame t, the algorithm carries out the noise parameter estimation in the log-spectral domain and perceptual enhancement of noisy speech in the time domain. Noise parameter estimation in the module "Noise parameter estimation" works in the log-spectral domain of input speech signals. Estimation of noise parameter is given by Algorithm 1. With the estimated noise parameter at the current frame, the module "wiener filter" outputs the enhanced speech spectrum in linear spectral domain, and the enhanced speech spectrum is used in "masking threshold calculation." Perceptual filter based on masking threshold and the estimated noise parameter is constructed in the module "perceptual filter." With the time-varying perceptual filter constructed, input noisy speech signal is filtered in time domain in the module "filtering" to obtain perceptually enhanced signal $\hat{x}(t)$. A detailed description of this algorithm is provided in the following sections.

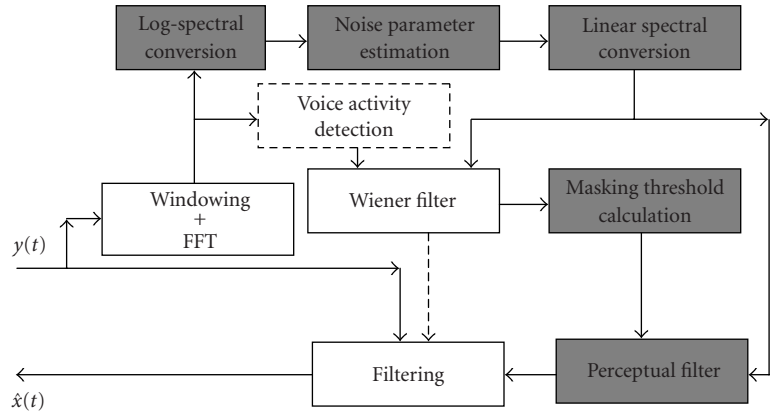


Figure 6: Diagram of the proposed speech enhancement method. Noisy signal $y(t)$ is converted into linear spectral amplitude in "windowing + FFT." Noise parameter is sequentially estimated in "noise parameter estimation." The estimated noise parameter is converted back into linear spectral domain and is fed into "Wiener filter" to obtain enhanced linear power spectrum. The enhanced spectrum is inputted to "masking threshold calculation," and the obtained masking threshold is used in perceptual filter with the estimated noise parameter in linear spectral domain. Module "perceptual filter" outputs filter coefficients for speech enhancement in "filtering," which outputs the enhanced signal $\hat{x}(t)$.

4.4.1. Masking threshold calculation

The masking threshold $T_j(t)$ is obtained through modeling the frequency selectivity of the human ear and its masking property. This paper applies a computational model of masking by Johnston [15].

Frequency masking threshold calculation

(1) Frequency analysis. According to a mapping between linear frequency and Bark frequency [14], power spectrum $x_j^{\text{lin}}(t)$ after short-time Fourier transform (STFT) of input speech signal is combined in each Bark bank $b\,(1 \le b \le B)$ by

$$x_b^{\text{lin}}(t) = \sum_{j=b_L}^{b_H} x_j^{\text{lin}}(t), \tag{43}$$

where b_L and b_H denote the lowest and the highest frequency for the bark index b.

(2) Convolution with spreading function. The spreading function $\mathbf{S}$ is used to estimate the effects of masking across critical bands. One example of the spreading function B_b at $b = 2$ is plotted in Figure 7. The spread Bark spectrum at bark index b is denoted as $C_b^{\text{lin}}(t) = B_b x_b^{\text{lin}}(t)$.

(3) Relative threshold calculation based on tone-like or noise-like determination. The tone-like or noise-like is determined by spectral flatness measure (SFM), which is calculated by measuring the decibel (dB) of the ratio of the geometric mean of the power spectrum to the arithmetic mean of the power spectrum.

(4) Masking threshold calculation. The relative threshold is subtracted from the spread critical band spectrum to yield the spread threshold estimate.

(5) Renormalization and including absolute threshold information [15].

(6) Converting the masking threshold from Bark frequency to linear frequency domain. The masking threshold in linear spectral domain $T_j(t)$ is obtained as a result.

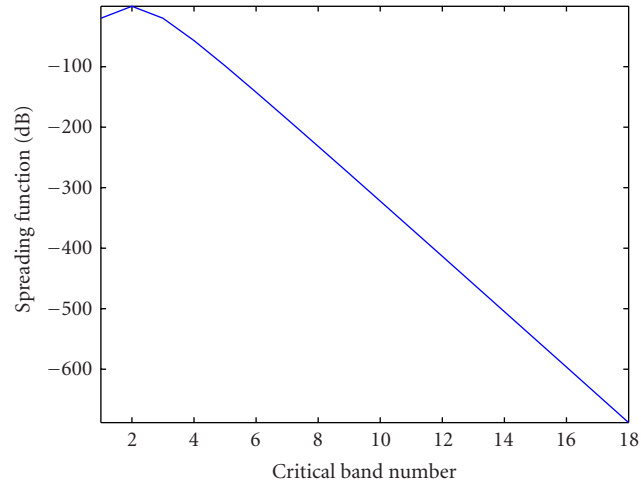


Figure 7: Spreading function for the noise masking threshold calculation. The plot shows the spreading function applied to critical band at 2.

An example of masking threshold in linear spectral domain for a given input spectrum is plotted in Figure 8. The sampling frequency is 8 kHz. Therefore, the total number of critical bands is $B = 18$. In the method presented above, the masking threshold is calculated from the clean speech signal.

4.4.2. Wiener filter and perceptual filter

We apply the method in Section 3.5 for time-varying noise parameter estimation. The jth element in $\hat{\mu}_n^l(t)$ is converted to linear spectral domain by exponential operation and then substitutes $\hat{n}_j^{\text{lin}}(t)$ in (40) and (42), respectively, for Wiener filter and perceptual filter. Masking threshold of the perceptual filter is obtained from Section 4.4.1.

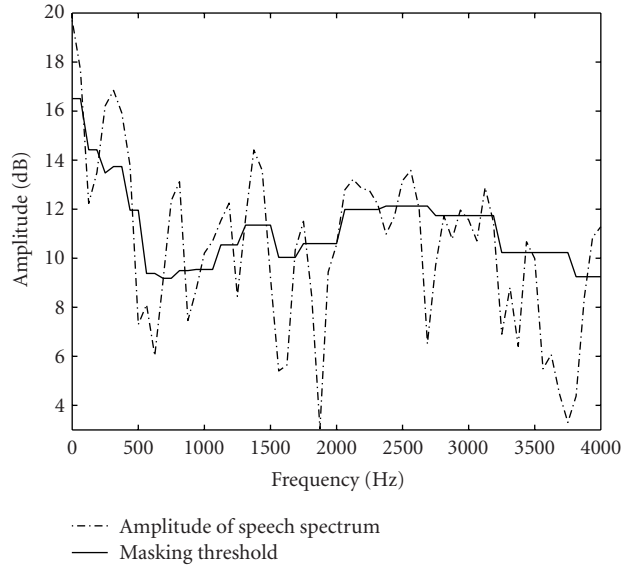


Figure 8: Example of the masking threshold $T_b(t)$.

4.4.3. Experimental results

Experiments were performed on Aurora 2 database. Speech models were trained on 8840 clean speech utterances. The model was an HMM with 18 states and 8 Gaussian mixtures in each state. Noise model was a single Gaussian density with time-varying mean vector. Window size was 25.0 milliseconds with a 10.0 milliseconds shift. J was set to 65.

We compared three systems. The first system, denoted as Baseline, was a speech enhancement system based on ETSI proposal [7], in which a VAD is used for decision of speech/nonspeech segments. Noise parameters were estimated from segmented noise-alone frames. The second system, denoted as Known, differs from the first system in that the Wiener filter was designed with noise parameters estimated by the proposed method. The third system, denoted as Perceptual, was a perceptual filter with noise parameter estimated by the proposed method.

VAD was initialized during the first three frames in each utterance. Driving variance V_n^l in (9) was set to 0.0003. Number of particles (N in (35)) was set to 800.

Noise signals were (1) simulated nonstationary noise, generated by multiplying white noise with a time-varying continuous factor in time domain, (2) Babble noise, and (3) Restaurant noise.

4.4.4. Performance evaluation

Spectrogram

An example of the original clean speech signals, noisy signals in the simulated nonstationary noise, and enhanced signals are shown in Figure 9. The contrast is more evident by viewing their corresponding spectrogram in Figure 10. It is observed that the noise power appeared after 0.4 seconds, which

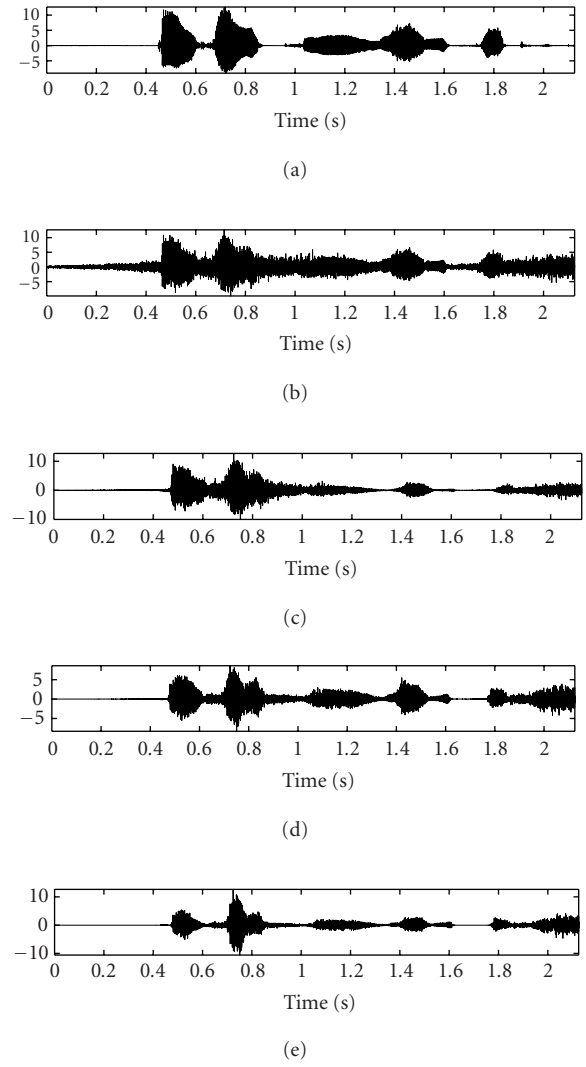


Figure 9: An example of signals. (a) Clean speech signal in English "Oh Oh Two One Six." (b) Noisy signal (noise is the simulated nonstationary noise and SNR is −0.2 dB). (c) Enhanced speech signal by Wiener filter (system Baseline). (d) Enhanced speech signal by Wiener filter with noise parameters estimated by the proposed method (system Known). (e) Enhanced speech signal by perceptual filter with noise parameters estimated by the proposed method (system Perceptual).

was almost at the time when the speech segments occurred. Figure 10c shows that Baseline cannot handle this nonstationarity of the noise, and the enhanced signal by the system still contains much noise power in the speech segments. On the contrary, with the proposed method, the enhanced signal by Known has reduced the noise power in speech segments (shown in Figure 10d). Perceptual reduces noise in the enhanced signal to a greater extent than the other two systems (shown in Figure 10e). An example in Babble noise is shown in Figure 11, and the corresponding spectrogram is shown in Figure 12.

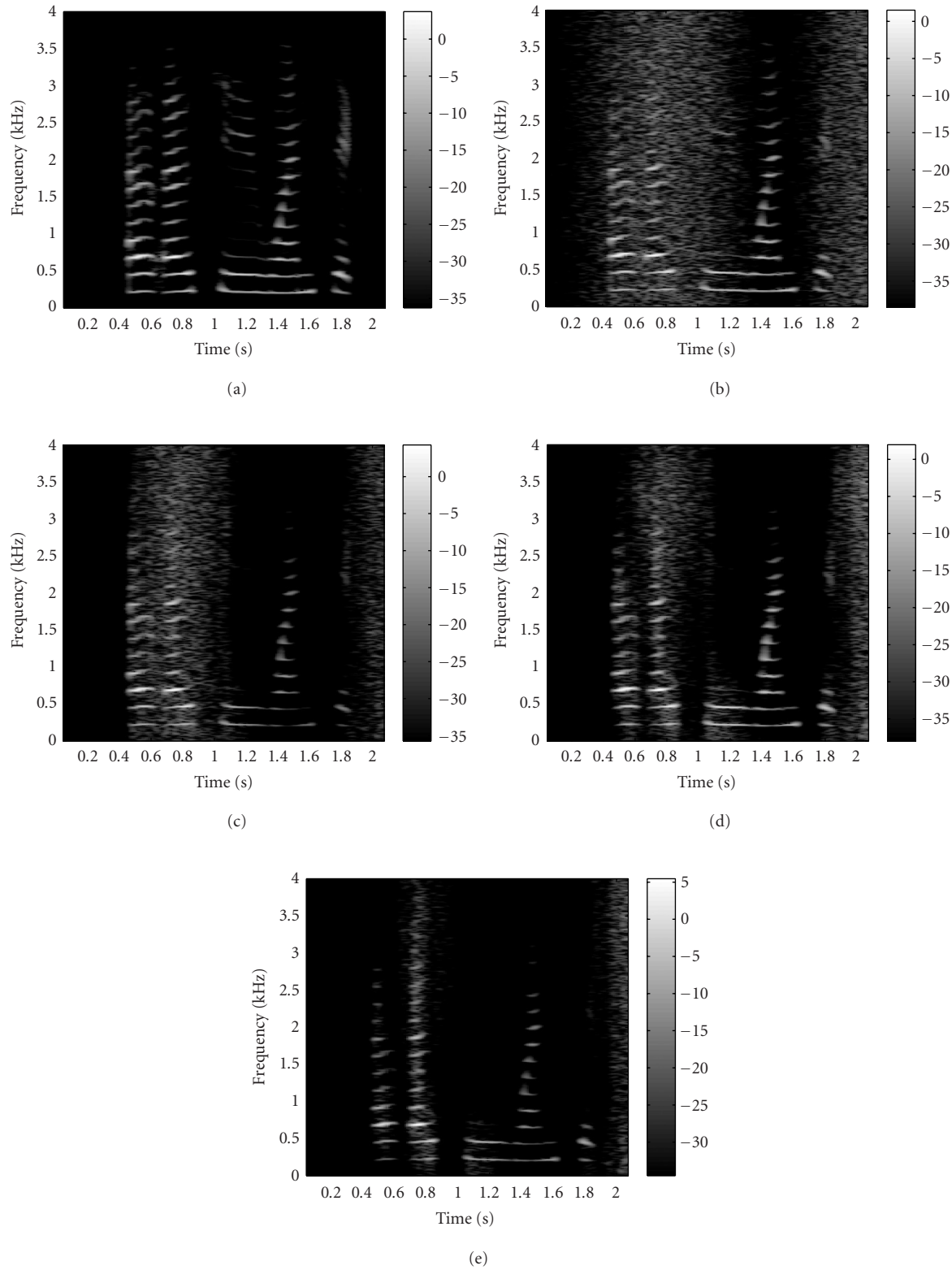


Figure 10: An example of the spectrum of the signals (from top to down). (a) Spectrogram of the clean speech signal in English "Oh Oh Two One Six." (b) Spectrogram of the noisy signal (noise is the simulated nonstationary noise and SNR is −0.2 dB). (c) Spectrogram of the enhanced signal by Wiener filter (system Baseline). (d) Spectrogram of the enhanced signal by Wiener filter with noise parameters estimated by the proposed method (system Known). (e) Spectrogram of the enhanced signal by perceptual filter with noise parameters estimated by the proposed method (system Perceptual).

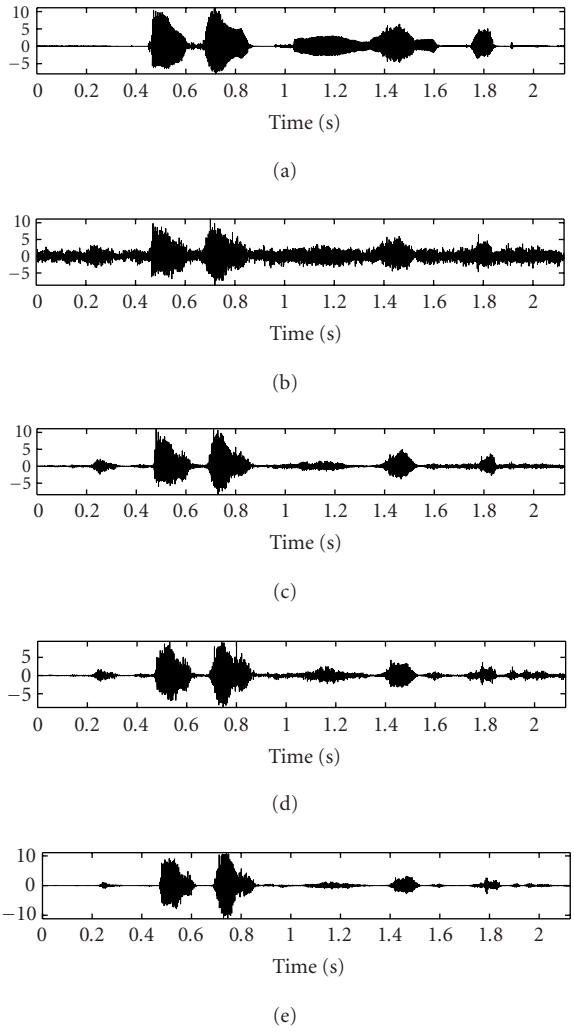


Figure 11: An example of signals. (a) Clean speech signal in English "Oh Oh Two One Six." (b) Noisy signal (noise is babble and SNR is −1.86 dB). (c) Enhanced speech by Wiener filter (system Baseline). (d) Enhanced speech by Wiener filter with noise parameter estimated by the proposed method (system Known). (e) Enhanced speech by perceptual filter with noise parameter estimated by the proposed method (system Perceptual).

However, the nonstationary noise was not perfectly removed in the final part of the sequence in Figure 10. This was in part due to inefficiency in the proposal distribution. Note that the speech states and mixtures were sampled according to the proposal distribution in (25). Thus, at the end of an utterance, the proposed speech states might not yet reach the states of silence. As a result, the speech parameter $\hat{\mu}^{l(i)}_{s_t^{(i)}k_t^{(i)}}(t)$ might still mask (be larger than) the noise parameter $\hat{\mu}^{l}_{n}(t)$. In this situation, the noise parameter may not have been updated (remained small if previously estimated noise parameter was smaller than speech parameters $\hat{\mu}^{l(i)}_{s_t^{(i)}k_t^{(i)}}$) because the Kalman gain in EKF was (approaching) zero. Therefore, noise in the final part of the sequence cannot be perfectly removed in some utterances.

Another direction in which the method needs to be improved is obvious in Figure 12. In this figure, high-frequency components are attenuated more than necessary. Since the Mel scale and Bark scale are wider in higher-frequency components than those in the lower-frequency components, noise parameters may not be accurately estimated due to frequency uncertainty between linear frequency and Mel scale (or Bark scale). Constructing speech enhancement algorithms that work directly in linear spectral domain (not Bark-scaled log-spectral domain in this work) may achieve higher frequency resolution and hence better enhancement results.

SNR improvement

The amount of noise reduction is generally measured with the segmental SNR (SegSNR) improvement—the difference between input and output SegSNR:

$$G_{\text{SNR}}=\frac{1}{B}\sum_{b=0}^{B-1}10\cdot\log_{10}\frac{(1/D)\sum_{d=0}^{D-1}n^2(d+Db)}{(1/D)\sum_{d=0}^{D-1}\left[x(d+Db)-\hat{x}(d+Db)\right]^2}, \tag{44}$$

where B represents the number of frames in the signal. D is the number of observation samples per frame, and it is set to 256.

Figure 13 shows the SegSNR improvement obtained from various noise types and at various noise levels. We can see that the system Known with the sequential Monte Carlo method has improved SegSNR over system Baseline. Figure 13 also shows that both systems Known and Perceptual benefit from the sequential Monte Carlo method. Furthermore, Perceptual shows much greater improvement than Known, which implies that it is effective to employ human auditory properties for speech enhancement.[9]

5. CONCLUSIONS AND DISCUSSIONS

We have presented a sequential Monte Carlo method for a Bayesian estimation of time-varying noise parameters. This method is derived from the general sequential Monte Carlo method for time-varying parameter estimation, but with particular considerations on time-varying noise parameter estimation. The estimated noise parameters are used in a Wiener filter and a perceptual filter for speech enhancement in nonstationary noisy environments. We also demonstrate that, with the estimated noise parameters, a sequential modification of the time-varying mean vector of speech models can improve speech recognition performance in nonstationary noise. The results show that it is a promising approach to handle speech signal processing in nonstationary noise scenarios.

[9] However, as discussed in Section 4.4.4, because the system Perceptual attenuated higher-frequency components more than traditional Wiener filters, the subjective quality of the perceptually enhanced speech signal in human hearing was in fact no better than that by Wiener filters.

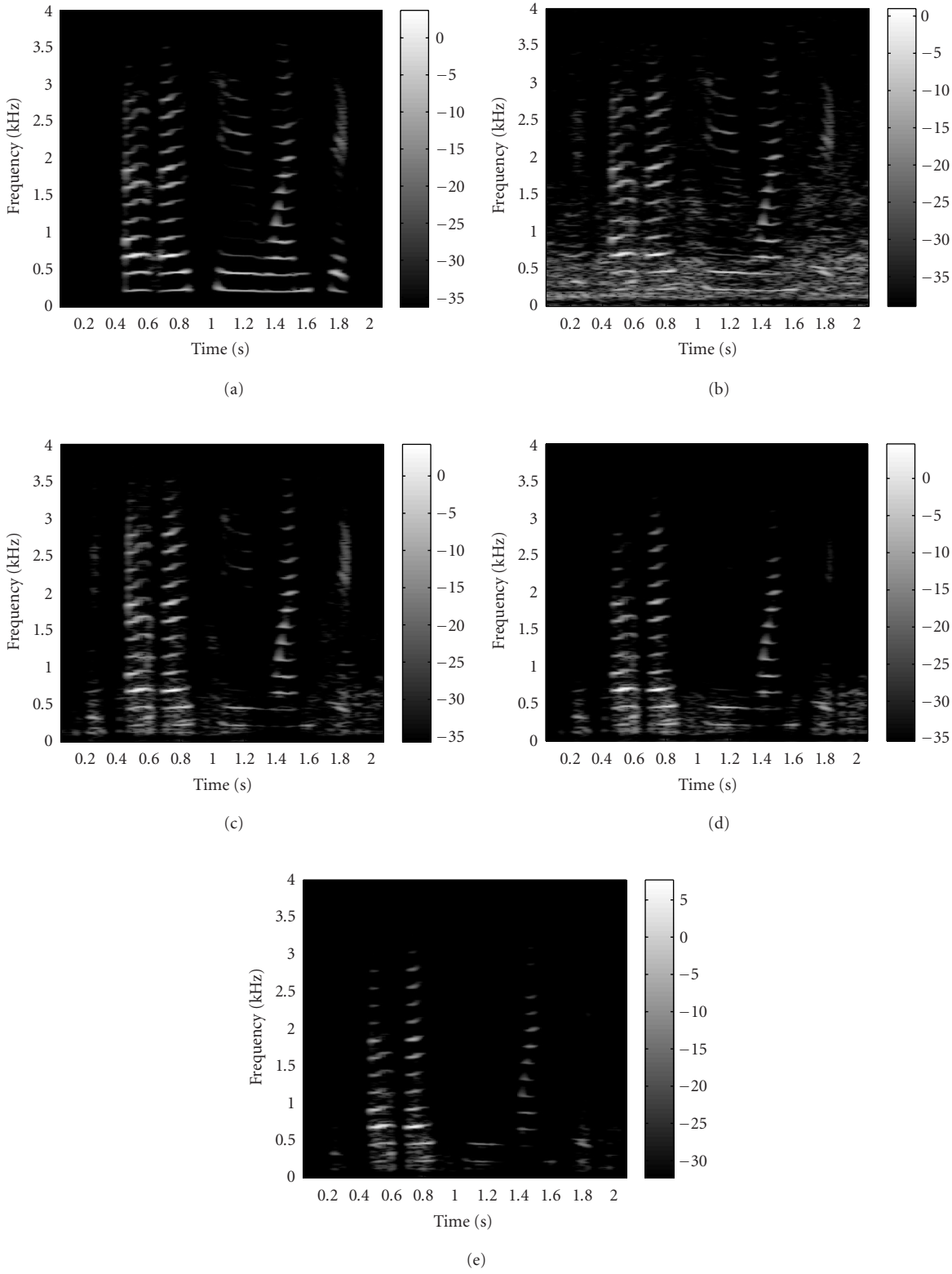
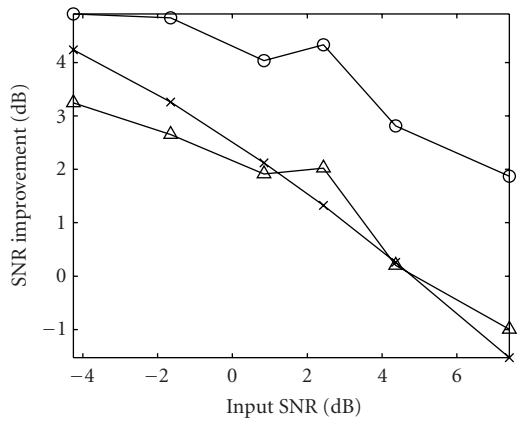
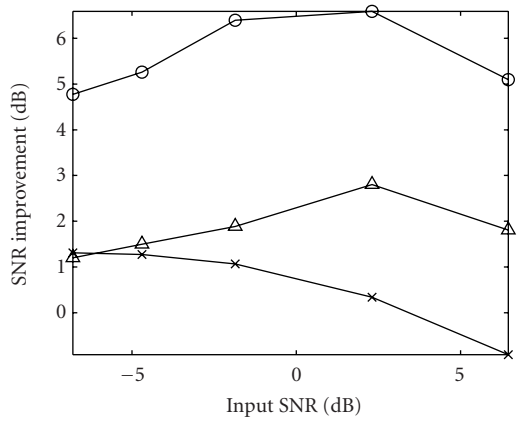


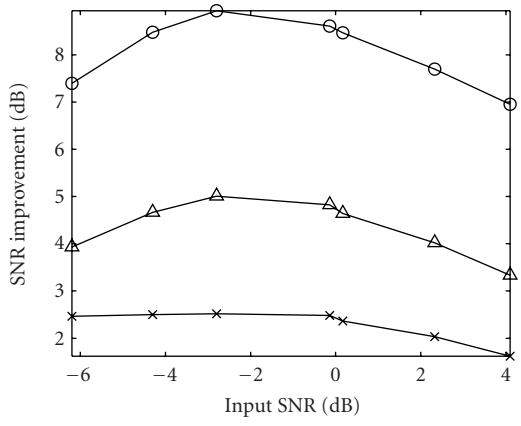
Figure 12: An example of the spectrum of the signals (from top to bottom). (a) Spectrogram of the clean speech signal in English "Oh Oh Two One Six." (b) Spectrogram of the noisy signal (noise is babble and SNR is −1.86 dB). (c) Spectrogram of the enhanced speech by Wiener filter (system Baseline). (d) Spectrogram of the enhanced speech by Wiener filter with noise parameter estimated by the proposed method (system Known). (e) Spectrogram of the enhanced speech by perceptual filter with noise parameter estimated by the proposed method (system Perceptual).



(a)



(b)



(c)

Figure 13: Segmental SNR improvement in the following noise: (a) simulated nonstationary noise; (b) babble noise; (c) restaurant noise. The tested systems are the following: (×) Wiener filter (system Baseline); (△) Wiener filter with noise parameter estimated by the proposed method (system Known); (o) Perceptual filter with noise parameter estimated by the proposed method (system Perceptual).

The sequential Monte Carlo method in this paper is successfully applied to two seemingly different areas in speech processing, speech enhancement, and speech recognition. This is possible because the graphical model shown in Figure 2 is applicable to the above two areas. The graphical model incorporates two hidden state sequences: one is the speech state sequence for modeling transition of speech units, and the other is a continuous-valued state sequence for modeling noise statistics. With the sequential Monte Carlo method, noise parameter estimation can be conducted via sampling the speech state sequences and updating continuous-valued noise states with $2N$ EKFs at each time. The highly parallel scheme of the method allows an efficient parallel implementation.

We are currently considering the following steps for improved performance: (1) making use of more efficient proposal distribution, for example, auxiliary sampling [29], (2) accurate training of speech models, and (3) design of algorithms working directly in linear spectral domain for speech enhancement. Improvements may be achieved if explicit speech modeling, for example, autocorrelation modeling of speech signals [11], pitch model [30], and so forth, can be incorporated in the framework. Because there is nonlinear function involved, we also believe that incorporating smoothing techniques recently proposed for nonlinear time series [31] may achieve improved performances.

APPENDICES

A. APPROXIMATION OF THE ENVIRONMENT EFFECTS ON SPEECH FEATURES

Effects of additive noise on speech power at the jth filter bank can be approximated by (2), where $y_j^{\text{lin}}(t)$, $x_j^{\text{lin}}(t)$, and $n_j^{\text{lin}}(t)$ denote noisy speech power, speech power, and additive noise power in filter bank j [8, 9].

In the log-spectral domain, this equation can be written below as

$$\begin{aligned}&\log\left(x_j^{\text{lin}}(t)+n_j^{\text{lin}}(t)\right)\\&\quad=\log x_j^{\text{lin}}(t)+\log\left(1+\frac{n_j^{\text{lin}}(t)}{x_j^{\text{lin}}(t)}\right)\\&\quad=\log x_j^{\text{lin}}(t)+\log\left(1+\exp\left(\log n_j^{\text{lin}}(t)-\log x_j^{\text{lin}}(t)\right)\right).\end{aligned}\tag{A.1}$$

Substituting $x_j^l(t)=\log x_j^{\text{lin}}(t)$, $n_j^l(t)=\log n_j^{\text{lin}}(t)$, and $y_j^l(t)=\log y_j^{\text{lin}}(t)$, we have (3).

B. EXTENDED KALMAN FILTER

The prediction likelihood of the EKF is given by [24]

$$\begin{aligned}&p\left(Y^l(t)|s_t^{(i)},k_t^{(i)},\mu_{s_t^{(i)}k_t^{(i)}}^{l(i)}(t),\hat{\mu}_n^{l(i)}(t-1),Y^l(t-1)\right)\\&=\int_{\mu_n^{l(i)}(t)}p\left(Y^l(t)|\theta^{(i)}(t)\right)p\left(\mu_n^{l(i)}(t)|\hat{\mu}_n^{l(i)}(t-1)\right)d\mu_n^{l(i)}(t)\\&\propto\exp\left(-\frac{1}{2}\alpha^{(i)}(t)^T\left(C^{(i)}(t)\Sigma^l_{s_t^{(i)}k_t^{(i)}}C^{(i)}(t)^T+V_n^l\right)^{-1}\alpha^{(i)}(t)\right),\end{aligned}\tag{B.1}$$

where, respectively, the innovation vector $\alpha^{(i)}(t)$, one-step ahead prediction of noise parameter $\hat{\mu}_n^{l(i)}(t|t-1)$, correlation matrix of the error in $\hat{\mu}_n^{l(i)}(t)$, correlation matrix of the error in $\hat{\mu}_n^{l(i)}(t|t-1)$, measurement matrix at time t (obtained by the first-order differentiation of (7) with respect to $\mu_n^{l(i)}(t)$), and gain function $G^{(i)}(t)$ are given as

$$\alpha^{(i)}(t) = Y^l(t) - \mu^{l(i)}_{s_t^{(i)}k_t^{(i)}}(t) - \log\left(1 + \exp\left(\hat{\mu}_n^{l(i)}(t-1) - \mu^{l(i)}_{s_t^{(i)}k_t^{(i)}}(t)\right)\right), \tag{B.2}$$

$$\hat{\mu}_n^{l(i)}(t|t-1) = \hat{\mu}_n^{l(i)}(t-1) + G^{(i)}(t)\alpha^{(i)}(t-1), \tag{B.3}$$

$$K^{(i)}(t) = K^{(i)}(t,t-1) - G^{(i)}(t)C^{(i)}(t)K^{(i)}(t,t-1), \tag{B.4}$$

$$K^{(i)}(t,t-1) = K^{(i)}(t-1) + V_n^l, \tag{B.5}$$

$$C^{(i)}(t) = \frac{\exp\left(\hat{\mu}_n^{l(i)}(t|t-1) - \mu^l_{s_t^{(i)}k_t^{(i)}}(t)\right)}{1 + \exp\left(\hat{\mu}_n^{l(i)}(t|t-1) - \mu^l_{s_t^{(i)}k_t^{(i)}}(t)\right)}, \tag{B.6}$$

$$G^{(i)}(t) = K^{(i)}(t,t-1)C^{(i)}(t)^T\left[C^{(i)}(t)K^{(i)}(t,t-1)C^{(i)}(t)^T + \Sigma^l_{s_t^{(i)}k_t^{(i)}}\right]^{-1}. \tag{B.7}$$

ACKNOWLEDGMENTS

The authors thank anonymous reviewers for their helpful comments on this paper, which substantially improved its presentation. The corresponding author thanks Dr. S. Nakamura (ATR SLT) for helpful discussions. Part of the work was performed when K. Yao was with ATR SLT.

REFERENCES

[1] J. S. Lim, *Speech Enhancement*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1983.

[2] K. K. Paliwal and A. Basu, "A speech enhancement method based on Kalman filtering," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, vol. 12, pp. 177–180, Dallas, Tex, USA, April 1987.

[3] J. S. Lim and A. V. Oppenheim, "All-pole modeling of degraded speech," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 26, no. 3, pp. 197–210, 1978.

[4] Y. Ephraim, "A Bayesian estimation approach for speech enhancement using hidden Markov models," *IEEE Trans. Signal Processing*, vol. 40, no. 4, pp. 725–735, 1992.

[5] Y. Ephraim and D. Malah, "Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator," *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. 32, no. 6, pp. 1109–1121, 1984.

[6] M. G. Hall, A. V. Oppenheim, and A. S. Willsky, "Time-varying parametric modeling of speech," *Signal Processing*, vol. 5, no. 3, pp. 267–285, 1983.

[7] ETSI, "Speech processing, transmission and quality aspects (STQ); distributed speech recognition; advanced front-end feature extraction algorithm; compression algorithms," Tech. Rep. ETSI ES 202 050, European Telecommunications Standards Institute, Sophia Antipolis, France, 2002.

[8] 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, no. 4, pp. 289–307, 1995.

[9] A. Acero, *Acoustical and environmental robustness in automatic speech recognition*, Ph.D. thesis, Carnegie Mellon University, Pittsburgh, Pa, USA, September 1990.

[10] S. V. Vaseghi and B. P. Milner, "Noise compensation methods for hidden Markov model speech recognition in adverse environments," *IEEE Trans. Speech, and Audio Processing*, vol. 5, no. 1, pp. 11–21, 1997.

[11] J. Vermaak, C. Andrieu, A. Doucet, and S. J. Godsill, "Particle methods for Bayesian modeling and enhancement of speech signals," *IEEE Trans. Speech, and Audio Processing*, vol. 10, no. 3, pp. 173–185, 2002.

[12] K. Yao and S. Nakamura, "Sequential noise compensation by sequential Monte Carlo method," in *Advances in Neural Information Processing Systems*, vol. 14, pp. 1205–1212, MIT Press, Cambridge, Mass, USA, 2001.

[13] D. Burshtein and S. Gannot, "Speech enhancement using a mixture-maximum model," *IEEE Trans. Speech, and Audio Processing*, vol. 10, no. 6, pp. 341–351, 2002.

[14] N. Virag, "Single channel speech enhancement based on masking properties of the human auditory system," *IEEE Trans. Speech, and Audio Processing*, vol. 7, no. 2, pp. 126–137, 1999.

[15] J. D. Johnston, "Estimation of perceptual entropy using noise masking criteria," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, vol. 5, pp. 2524–2527, New York, NY, USA, April 1988.

[16] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1993.

[17] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, "An introduction to variational methods for graphical models," in *Learning in Graphical Models*, pp. 105–162, Kluwer Academic, Dordrecht, 1998.

[18] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, 1967.

[19] W. K. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.

[20] G. Casella and C. P. Robert, "Rao-Blackwellisation of sampling schemes," *Biometrika*, vol. 83, no. 1, pp. 81–94, 1996.

[21] A. Doucet, N. de Freitas, and N. J. Gordon, Eds., *Sequential Monte Carlo in Practice*, Springer-Verlag, New York, NY, USA, 2001.

[22] J. M. Bernardo and A. F. M. Smith, *Bayesian Theory*, John Wiley & Sons, New York, NY, USA, 1994.

[23] V. S. Zaritskii, V. B. Svetnik, and L. I. Shimelevich, "Monte-Carlo techniques in problem of optimal information processing," *Automation and Remote Control*, vol. 36, no. 3, pp. 2015–2022, 1975.

[24] A. Doucet, S. J. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, no. 3, pp. 197–208, 2000.

[25] E. L. Lehmann and G. Casella, *Theory of Point Estimation*, Springer-Verlag, New York, NY, USA, 1998.

[26] R. Merwe, A. Doucet, N. Freitas, and E. Wan, "The unscented particle filter," Tech. Rep. CUED/F-INFENG/TR. 380, Signal Processing Group, Engineering Department, Cambridge University, Cambridge, UK, 2000.

[27] B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap*, Chapman & Hall, New York, NY, USA, 1993.

[28] J. S. Liu and R. Chen, "Sequential Monte Carlo methods for dynamic systems," *Journal of the American Statistical Association*, vol. 93, no. 443, pp. 1032–1044, 1998.

[29] M. K. Pitt and N. Shephard, "Filtering via simulation: auxiliary particle filters," *Journal of the American Statistical Association*, vol. 94, no. 446, pp. 590–699, 1999.

[30] M. Davy and S. J. Godsill, "Bayesian harmonic models for musical pitch estimation and analysis," Tech. Rep. CUED/F-INFENG/TR.431, Signal Processing Group, Engineering Department, Cambridge University,, Cambridge, UK, 2002.
[31] S. J. Godsill, A. Doucet, and M. West, "Monte Carlo smoothing for nonlinear time series," *Journal of the American Statistical Association*, vol. 99, no. 465, pp. 156–168, 2004.

Kaisheng Yao is a Postgraduate Researcher at the Institute for Neural Computation, University of California, San Diego. Dr. Yao received his B. Eng. and M. Eng. in electrical engineering from Huazhong University of Science & Technology (HUST), Wuhan, China, in 1993 and 1996. In 2000, he received his Dr. Eng. degree for work performed jointly at the State Key Laboratory on Microwave and Digital Communications, Department of Electronic Engineering, Tsinghua University, China, and at the Human Language Technology Center (HLTC), Department of Electrical and Electronic Engineering, Hong Kong University of Science and Technology, Hong Kong. From 2000 to 2002, he was a Researcher in Spoken Language Translation Research Laboratories at the Advanced Telecommunications Research Institute International (ATR), Japan. Since August 2002, he has been with the Institute for Neural Computation, University of California, San Diego. His research interests are mainly in speech recognition in noise, acoustic modeling, dynamic Bayesian networks, statistical signal processing, and some general problems in pattern recognition. He has severed as a Reviewer of IEEE Transactions on Speech and Audio Processing and ICASSP.

Te-Won Lee is an Associate Research Professor at the Institute for Neural Computation, University of California, San Diego, and a Collaborating Professor in the Biosystems Department at the Korea Advanced Institute of Science and Technology (KAIST). Dr. Lee received his diploma degree in March 1995 and his Ph.D. degree in October 1997 (summa cum laude) in electrical engineering from the University of Technology Berlin. During his studies, he received the Erwin-Stephan Prize for excellent studies from the University of Technology Berlin and the Carl-Ramhauser Prize for excellent dissertations from the DaimlerChrysler Corporation. He was a Max-Planck Institute Fellow from 1995 till 1997 and a Research Associate at the Salk Institute for Biological Studies from 1997 till 1999. Dr. Lee's research interests include machine learning algorithms with applications in signal and image processing. Recently, he has worked on variational Bayesian methods for independent component analysis, algorithms for speech enhancement and recognition, models for computational vision, and classification algorithms for medical informatics.