Multilevel Mixture Kalman Filter

Dong Guo

Department of Electrical Engineering, Columbia University, New York, NY 10027, USA
Email: author please provide

Xiaodong Wang

Department of Electrical Engineering, Columbia University, New York, NY 10027, USA
Email: wangx@ee.columbia.edu

Rong Chen

Department of Information and Decision Sciences, University of Illinois at Chicago, Chicago, IL 60607-7124, USA
Department of Business Statistics & Econometrics, Peking University, Beijing 100871, China
Email: rongchen@uic.edu

The mixture Kalman filter is a general sequential Monte Carlo technique for conditional linear dynamic systems. It generates samples of some indicator variables recursively based on sequential importance sampling (SIS) and integrates out the linear and Gaussian state variables conditioned on these indicators. Due to the marginalization process, the complexity of the mixture Kalman filter is quite high if the dimension of the indicator sampling space is high. In this paper, we address this difficulty by developing a new Monte Carlo sampling scheme, namely, the multilevel mixture Kalman filter. The basic idea is to make use of the multilevel or hierarchical structure of the space from which the indicator variables take values. That is, we draw samples in a multilevel fashion, beginning with sampling from the highest-level sampling space and then draw samples from the associate subspace of the newly drawn samples in a lower-level sampling space, until reaching the desired sampling space. Such a multilevel sampling scheme can be used in conjunction with the delayed estimation method, such as the delayed-sample method, resulting in delayed multilevel mixture Kalman filter. Examples in wireless communication, specifically the coherent and noncoherent 16-QAM over flat-fading channels, are provided to demonstrate the performance of the proposed multilevel mixture Kalman filter.

Keywords and phrases: sequential Monte Carlo, mixture Kalman filter, multilevel mixture Kalman filter, delayed-sample method.

1. INTRODUCTION

Recently there have been significant interests in the use of the sequential Monte Carlo (SMC) methods to solve on-line estimation and prediction problems in dynamic systems. Compared with the traditional filtering methods, the simple, flexible—yet powerful—SMC provides effective means to overcome the computational difficulties in dealing with nonlinear dynamic models. The basic idea of the SMC technique is the recursive use of the sequential importance sampling (SIS). There also have been many recent modifications and improvements on the SMC methodology [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12].

Among these SMC methods, the mixture Kalman filter (MKF) [3] is a powerful tool to deal with conditional dynamic linear models (CDLMs) and finds important applications in digital wireless communications [3, 13, 14]. A similar method is also discussed in [15] for CDLM system. The CDLM is a direct generalization of the dynamic linear model (DLM) [16] and it can be generally described as follows:

$$
\begin{aligned}
\mathbf{x}_t &= \mathbf{F}_{\lambda_t}\mathbf{x}_{t-1} + \mathbf{G}_{\lambda_t}\mathbf{u}_t, \\
\mathbf{y}_t &= \mathbf{H}_{\lambda_t}\mathbf{x}_t + \mathbf{K}_{\lambda_t}\mathbf{v}_t,
\end{aligned}
\tag{1}
$$

where $\mathbf{u}_t \sim \mathcal{N}(0, \mathbf{I})$ and $\mathbf{v}_t \sim \mathcal{N}(0, \mathbf{I})$ are the state and observation noise, respectively, and λ_t is a sequence of random indicator variables which may form a Markov chain, but are independent of $\mathbf{u}_t$ and $\mathbf{v}_t$ and the past $\mathbf{x}_s$ and $\mathbf{y}_s$, $s < t$. The matrices $\mathbf{F}_{\lambda_t}$, $\mathbf{G}_{\lambda_t}$, $\mathbf{H}_{\lambda_t}$, and $\mathbf{K}_{\lambda_t}$ are known, given λ_t.

An important feature of CDLM is that, given the trajectory of the indicator $\{\lambda_t\}$, the system becomes Gaussian and linear, for which the Kalman filter can be used. Thus, by using the marginalization technique for Monte Carlo computation [17], the MKF focuses on the sampling of the indicator variable λ_t other than the whole state variable $\{\mathbf{x}_t, \lambda_t\}$. This method can drastically reduce Monte Carlo variances associated with a standard sequential importance sampler applied directly to the space of the state variable.
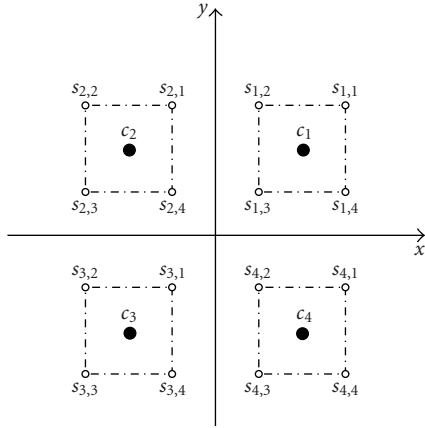
FIGURE 1: The multilevel structure of the 16-QAM modulation used in digital communications. The set of transmitted symbol $\mathcal{A}_1 = \{s_{i,j}, i = 1,\ldots,4, j = 1,\ldots,4\}$ is the original sampling space, and the centers $\mathcal{A}_2 = \{c_1, c_2, c_3, c_4\}$ constitute a higher-level sampling space.

However, the computational complexity of the MKF can be quite high, especially in the case of high-dimensional indicator space, due to the need of marginalizing out the indicator variables. Fortunately, often the space from which the indicator variables take values exhibits multilevel or hierarchical structures, which can be exploited to reduce the computational complexity of the MKF. For example, a multilevel structure of the 16-QAM modulation used in digital communications is shown in Figure 1. The set of transmitted symbols $\mathcal{A}_1 = \{s_{i,j}, i = 1,\ldots,4, j = 1,\ldots,4\}$ is the original sampling space, and the centers $\mathcal{A}_2 = \{c_1, c_2, c_3, c_4\}$ constitute a higher-level sampling space. Thus, based on the observed data, for every sample stream, we first draw a sample (say c_1) from the higher-level sampling space $\mathcal{A}_2$ and then draw a new sample from the associated subspaces $s_{1,1}$, $s_{1,2}, s_{1,3}, s_{1,4}$ of c_1 in the original sampling space $\mathcal{A}_1$. In this way, we need not sample from the entire original sampling space, and many Kalman filter update steps associated with the standard MKF can be saved.

This kind of hierarchical structure imposed on the indicator space is also employed in the partitioned sampling strategy [18], which greatly improved the efficiency and the accuracy for multiple target tracking over the original SMC methods. However, in this paper, the hierarchical structure is employed to reduce the computational load associated with MKF, especially for high-dimensional indicator space, while retaining the desirable properties of MKF.

Dynamic systems often possess strong memories, that is, future observations can reveal substantial information about the current state. Therefore, it is often beneficial to make use of these future observations in sampling the current state. However, an MKF method usually does not go back to regenerate past samples in view of the new observations, although the past estimation can be adjusted by using the new importance weights. To overcome this difficulty, a *delayed-sample* method is developed [13]. It makes use of future observations in generating samples of the current state. It is seen

there that this method is especially effective in improving the performance of the MKF. However, the computational complexity of the delayed-sample method is very high. For example, for a Δ-step delayed-sample method, the algorithmic complexity is $\mathcal{O}(|\mathcal{A}_1|^\Delta)$, where $|\mathcal{A}_1|$ is the cardinality of the original sampling space. Here, we also provide a delayed multilevel MKF by exploring the multilevel structure of the indicator space. Instead of exploring the original entire space of future states, we only sample the future states in a higher-level sampling space, thus significantly reducing the dimension of the search space and the computational complexity.

In recent years, the SMC methods have been successfully employed in several important problems in communications, such as the detection in flat-fading channels [13, 19, 20], space-time coding [21, 22], OFDM system [23], and so on. To show the good performance of the proposed novel receivers, we apply them into the problem of adaptive detection in flat-fading channels in the presence of Gaussian noise.

The remainder of the paper is organized as follows. Section 2 briefly reviews the MKF algorithm and its variants. In Section 3, we present the multilevel MKF algorithm. In Section 4, we treat the delayed multilevel MKF algorithm. In Section 5, we provide simulation examples. Section 6 concludes the paper.

2. BACKGROUND OF MIXTURE KALMAN FILTER

2.1. Mixture Kalman filter

Consider again the CDLMs defined by (1). The MKF exploits the conditional Gaussian property conditioned on the indicator variable and utilizes a marginalization operation to improve the algorithmic efficiency. Instead of dealing with both $\mathbf{x}_t$ and λ_t, the MKF draws Monte Carlo samples only in the indicator space and uses a mixture of Gaussian distributions to approximate the target distribution. Compared with the generic SMC method, the MKF is substantially more efficient (e.g., it produces more accurate results with the same computational resources).

First we define an important concept that is used throughout the paper. A set of random samples and the corresponding weights $\{(\eta^{(i)}, w^{(i)})\}_{i=1}^m$ is said to be *properly weighted* with respect to the distribution $\pi(\cdot)$ if, for any measurable function h, we have

$$\frac{\sum_{j=1}^m h(\eta^{(j)}) w^{(j)}}{\sum_{j=1}^m w^{(j)}} \longrightarrow E_\pi\{h(\eta)\} \quad \text{as } m \longrightarrow \infty. \quad (2)$$

In particular, if $\eta^{(j)}$ is sampled from a trial distribution $g(\cdot)$ which has the same support as π, and if $w^{(j)} = \pi(\eta^{(j)})/g(\eta^{(j)})$, then $\{(\eta^{(j)}, w^{(j)})\}_{j=1}^m$ is properly weighted with respect to $\pi(\cdot)$.

Let $\mathbf{Y}_t = (\mathbf{y}_0, \mathbf{y}_1, \ldots, \mathbf{y}_t)$ and $\mathbf{\Lambda}_t = (\lambda_0, \lambda_1, \ldots, \lambda_t)$. By recursively generating a set of properly weighted random samples $\{(\mathbf{\Lambda}_t^{(j)}, w_t^{(j)})\}_{j=1}^m$ to represent $p(\mathbf{\Lambda}_t \mid \mathbf{Y}_t)$, the MKF approximates the target distribution $p(\mathbf{x}_t \mid \mathbf{Y}_t)$ by a random mixture of Gaussian distributions

$$\sum_{j=1}^m w_t^{(j)} \mathcal{N}_c\left(\boldsymbol{\mu}_t^{(j)}, \boldsymbol{\Sigma}_t^{(j)}\right), \quad (3)$$

where

$$\boldsymbol{\mu}_t^{(j)} = \boldsymbol{\mu}_t\left(\boldsymbol{\Lambda}_t^{(j)}\right), \qquad \boldsymbol{\Sigma}_t^{(j)} = \boldsymbol{\Sigma}_t\left(\boldsymbol{\Lambda}_t^{(j)}\right) \qquad (4)$$

are obtained with a Kalman filter on the system (1) for the given indicator trajectory $\boldsymbol{\Lambda}_t^{(j)}$. Denote

$$\kappa_t^{(j)} \triangleq \left[\boldsymbol{\mu}_t^{(j)}, \boldsymbol{\Sigma}_t^{(j)}\right]. \qquad (5)$$

Thus, a key step in the MKF is the production at time t of the weighted samples of indicators $\{(\boldsymbol{\Lambda}_t^{(j)}, \kappa_t^{(j)}, w_t^{(j)})\}_{j=1}^m$ based on the set of samples $\{(\boldsymbol{\Lambda}_{t-1}^{(j)}, \kappa_{t-1}^{(j)}, w_{t-1}^{(j)})\}_{j=1}^m$ at the previous time $(t-1)$. Suppose that the indicator λ_t takes values from a finite set $\mathcal{A}_1$. The MKF algorithm is as follows.

Algorithm 1 (MKF). Suppose at time $(t-1)$, a set of property weighted samples $\{(\boldsymbol{\Lambda}_{t-1}^{(j)}, \kappa_{t-1}^{(j)}, w_{t-1}^{(j)})\}_{j=1}^m$ is available with respect to $p(\boldsymbol{\Lambda}_{t-1} \mid \mathbf{Y}_{t-1})$. Then at time t, as the new data $\mathbf{y}_t$ becomes available, the following steps are implemented to update each weighted sample.

For $j = 1, \ldots, m$, the following steps are applied.

(i) Based on the new data $\mathbf{y}_t$, for each $a_i \in \mathcal{A}_1$, run a one-step Kalman filter update assuming $\lambda_t = a_i$ to obtain

$$\kappa_{t-1}^{(j)} \xrightarrow{\mathbf{y}_t, \lambda_t = a_i} \kappa_{t,i}^{(j)} \triangleq \left[\boldsymbol{\mu}_t\left(\boldsymbol{\Lambda}_{t-1}^{(j)}, \lambda_t = a_i\right), \boldsymbol{\Sigma}_t\left(\boldsymbol{\Lambda}_{t-1}^{(j)}, \lambda_t = a_i\right)\right]. \qquad (6)$$

(ii) For each $a_i \in \mathcal{A}_1$, compute the sampling density

$$\begin{aligned}\rho_{t,i}^{(j)} &\triangleq P\left(\lambda_t = a_i \mid \boldsymbol{\Lambda}_{t-1}^{(j)}, \mathbf{Y}_t\right) \\ &\propto p\left(\mathbf{y}_t \mid \lambda_t = a_i, \boldsymbol{\Lambda}_{t-1}^{(j)}, \mathbf{Y}_{t-1}\right) P\left(\lambda_t = a_i \mid \boldsymbol{\Lambda}_{t-1}^{(j)}\right).\end{aligned} \qquad (7)$$

Note that by the model (1), the first density in (7) is Gaussian and can be computed based on the Kalman filter update (6). Draw a sample $\lambda_t^{(j)}$ according to the above sampling density. Append $\lambda_t^{(j)}$ to $\boldsymbol{\Lambda}_{t-1}^{(j)}$ and obtain $\boldsymbol{\Lambda}_t^{(j)}$. If $\lambda_t^{(j)} = a_i$, then set $\kappa_t^{(j)} = \kappa_{t,i}^{(j)}$.

(iii) Compute the importance weight

$$\begin{aligned}w_t^{(j)} &= w_{t-1}^{(j)} \cdot p\left(\mathbf{y}_t \mid \boldsymbol{\Lambda}_{t-1}^{(j)}, \mathbf{Y}_{t-1}\right) \\ &\propto w_{t-1}^{(j)} \cdot \sum_{i=1}^{|\mathcal{A}_1|} \rho_{t,i}^{(j)}.\end{aligned} \qquad (8)$$

The new sample $\{\boldsymbol{\Lambda}_t^{(j)}, \kappa_t^{(j)}, w_t^{(j)}\}$ is then properly weighted with respect to $p(\boldsymbol{\Lambda}_t \mid \mathbf{Y}_t)$.

(iv) Perform a resampling step as discussed below.

2.2. Resampling procedure

The importance sampling weight $w_t^{(j)}$ measures the "quality" of the corresponding imputed indicator sequence $\boldsymbol{\Lambda}_t^{(j)}$. A relatively small weight implies that the sample is drawn far from the main body of the posterior distribution and has a small contribution in the final estimation. Such a sample is said

to be ineffective. If there are too many ineffective samples, the Monte Carlo procedure becomes inefficient. To avoid the degeneracy, a useful resampling procedure, which was suggested in [7, 11], may be used. Roughly speaking, resampling is to multiply the streams with the larger importance weights, while eliminating the ones with small importance weights. A simple, but efficient, resampling procedure consists of the following two steps.

(1) Sample a new set of streams $\{\tilde{\boldsymbol{\Lambda}}_t^{(j)}, \tilde{\boldsymbol{\mu}}_t^{(j)}, \tilde{\boldsymbol{\Sigma}}_t^{(j)}\}_{j=1}^m$ from $\{\boldsymbol{\Lambda}_t^{(j)}, \boldsymbol{\mu}_t^{(j)}, \boldsymbol{\Sigma}_t^{(j)}\}_{j=1}^m$ with probability proportional to the importance weights $\{w_t^{(j)}\}_{j=1}^m$.

(2) To each stream in $\{\tilde{\boldsymbol{\Lambda}}_t^{(j)}, \tilde{\boldsymbol{\mu}}_t^{(j)}, \tilde{\boldsymbol{\Sigma}}_t^{(j)}\}_{j=1}^m$, ==assign equal weight,== that is, $\tilde{w}_t^{(j)} = 1/m$, $j = 1, \ldots, m$. ☐ 5

Resampling can be done at every fixed-length time interval (say, every five steps) or it can be conducted dynamically. The *effective sample size* can be used to monitor the variation of the importance weights of the sample streams and to decide when to resample as the system evolves. The effective sample size is defined as in [13]:

$$\bar{m}_t \triangleq \frac{m}{1 + v_t^2}, \qquad (9)$$

where v_t, the coefficient of variation, is given by

$$v_t^2 = \frac{1}{m} \sum_{j=1}^m \left(\frac{w_t^{(j)}}{\bar{w}_t} - 1\right)^2, \qquad (10)$$

with $\bar{w}_t = \sum_{j=1}^m w_t^{(j)}/m$. In dynamic resampling, a resampling step is performed once the effective sample size $\bar{m}_t$ is below a certain threshold.

Heuristically, resampling can provide chances for good sample streams to amplify themselves and hence "rejuvenate" the sampler to produce a better result for future states as the system evolves. It can be shown that the samples drawn by the above resampling procedure are also indeed properly weighted with respect to $p(\boldsymbol{\Lambda}_t \mid \mathbf{Y}_t)$, provided that m is sufficiently large. In practice, when small to modest m is used (we use $m = 50$ in this paper), the resampling procedure can be seen as a tradeoff between the bias and the variance. That is, the new samples with their weights resulting from the resampling procedure are only approximately proper, and this introduces small bias in the Monte Carlo estimation. On the other hand, however, resampling significantly reduces the Monte Carlo variance for future samples.

2.3. Delayed estimation

The model (1) is often observed to be highly ==correlated.== As a result, future observations often contain information about the current state. Hence a delayed estimate is usually more accurate than the concurrent estimate. In delayed estimation, instead of making inference on $\boldsymbol{\Lambda}_t$ instantaneously with the posterior distribution $p(\boldsymbol{\Lambda}_t \mid \mathbf{Y}_t)$, we delay this inference to a later time $(t + \Delta)$, $\Delta > 0$, with the distribution $p(\boldsymbol{\Lambda}_t \mid \mathbf{Y}_{t+\Delta})$. As discussed in [13], there are primarily two approaches to ☐ 6

delayed estimation, namely, the delayed-weight method and the delayed-sample method.

2.3.1. Delayed-weight method

In the concurrent MKF algorithm, if the set $\{(\Lambda_{t+\delta}^{(j)}, w_{t+\delta}^{(j)})\}_{j=1}^m$ is properly weighted with respect to $p(\Lambda_{t+\delta} \mid \mathbf{Y}_{t+\delta})$, then when we focus our attention on λ_t at time $(t+\delta)$, we have that $\{(\lambda_t^{(j)}, w_{t+\delta}^{(j)})\}_{j=1}^m$ is properly weighted with respect to $p(\lambda_t \mid \mathbf{Y}_{t+\delta})$. Then any inference about the indicator λ_t, $E\{h(\lambda_t) \mid \mathbf{Y}_{t+\delta}\}$, can be approximated by

$$
\begin{aligned}
&E\{h(\lambda_t) \mid \mathbf{Y}_{t+\delta}\} \\
&\quad \cong \frac{1}{W_{t+\delta}} \sum_{j=1}^m h(\lambda_t^{(j)}) w_{t+\delta}^{(j)}, \quad W_{t+\delta} = \sum_{j=1}^m w_{t+\delta}^{(j)}.
\end{aligned}
\tag{11}
$$

Since the weights $\{w_{t+\delta}^{(j)}\}_{j=1}^m$ contain information about the future observations $(\mathbf{y}_{t+1}, \ldots, \mathbf{y}_{t+\delta})$, the estimate in (11) is usually more accurate than the concurrent estimate. Note that such a delayed estimation method incurs no additional computational cost (i.e., CPU time), but it requires some extra memory for storing $\{\lambda_t^{(j)}, \ldots, \lambda_{t+\delta}^{(j)}\}_{j=1}^m$. For most systems, this simple delayed-weight method is quite effective for improving the performance over the concurrent method. However, if this method is not sufficient for exploiting the constraint structures of the indicator variable, we must resort to the delayed-sample method, which is described next.

2.3.2. Delayed-sample method

An alternative method of delayed estimation is to generate both the delayed samples and the weights $\{(\lambda_t^{(j)}, w_t^{(j)})\}_{j=1}^m$ based on the observations $\mathbf{Y}_{t+\Delta}$, hence making $p(\Lambda_t \mid \mathbf{Y}_{t+\Delta})$ the target distribution at time $(t + \Delta)$. The procedure will provide better Monte Carlo samples since it utilizes the future observations $(\mathbf{y}_{t+1}, \ldots, \mathbf{y}_{t+\Delta})$ in generating the current samples of λ_t. But the algorithm is also more demanding both analytically and computationally because of the need of marginalizing out $\lambda_{t+1}, \ldots, \lambda_{t+\Delta}$.

For each possible "future" (relative to time $t - 1$) symbol sequence at time $(t + \Delta - 1)$, that is,

$$
(\lambda_t, \lambda_{t+1}, \ldots, \lambda_{t+\Delta-1}) \in \mathcal{A}_1^\Delta,
\tag{12}
$$

we keep the value of a Δ-step Kalman filter $\{\kappa_{t+\tau}^{(j)}(\lambda_t^{t+\tau})\}_{\tau=0}^{\Delta-1}$, where

$$
\begin{aligned}
&\kappa_{t+\tau}^{(j)}(\lambda_t^{t+\tau}) \\
&\triangleq \left[\mu_{t+\tau}(\Lambda_{t-1}^{(j)}, \lambda_t^{t+\tau}), \Sigma_{t+\tau}(\Lambda_{t-1}^{(j)}, \lambda_t^{t+\tau}) \right], \quad \tau = 0, \ldots, \Delta-1,
\end{aligned}
\tag{13}
$$

with $\lambda_a^b \triangleq (\lambda_a, \lambda_{a+1}, \ldots, \lambda_b)$. Denote

$$
\kappa_{t-1}^{(j)} \triangleq \left\{ \kappa_{t-1}^{(j)}, \{\kappa_{t+\tau}^{(j)}(\lambda_t^{t+\tau})\}_{\tau=0}^{\Delta-1} : \lambda_t^{t+\tau} \in \mathcal{A}_1^{\tau+1} \right\}.
\tag{14}
$$

The delayed-sample MKF algorithm recursively propagates

the samples properly weighted for $p(\Lambda_{t-1} \mid \mathbf{Y}_{t+\Delta-1})$ to those for $p(\Lambda_t \mid Y_{t+\Delta})$ and is summarized as follows.

Algorithm 2 (delayed-sample MKF). Suppose, at time $(t+\Delta-1)$, a set of properly weighted samples $\{(\Lambda_{t-1}^{(j)}, \kappa_{t-1}^{(j)}, w_{t-1}^{(j)})\}_{j=1}^m$ is available with respect to $p(\Lambda_{t-1} \mid \mathbf{Y}_{t+\Delta-1})$. Then at time $(t + \Delta)$ as the new data $\mathbf{y}_{t+\Delta}$ becomes available, the following steps are implemented to update each weighted sample.

For $j = 1, 2, \ldots, m$, the following steps are performed.

(i) For each $\lambda_{t+\Delta} = a_i \in \mathcal{A}_1$, and for each $\lambda_t^{t+\Delta-1} \in \mathcal{A}_1^\Delta$, perform a one-step update on the corresponding Kalman filter $\kappa_{t+\Delta-1}^{(j)}(\lambda_t^{t+\Delta-1})$, that is,

$$
\kappa_{t+\Delta-1}^{(j)}(\lambda_t^{t+\Delta-1}) \xrightarrow{\mathbf{y}_{t+\Delta}, \lambda_{t+\Delta}=a_i} \kappa_{t+\Delta}^{(j)}(\lambda_t^{t+\Delta-1}, \lambda_{t+\Delta} = a_i).
\tag{15}
$$

(ii) For each $a_i \in \mathcal{A}_1$, compute the sampling density

$$
\begin{aligned}
\rho_{t,i}^{(j)} &\triangleq P\left(\lambda_t = a_i \mid \Lambda_{t-1}^{(j)}, \mathbf{Y}_{t+\Delta}\right) \\
&= P\left(\lambda_t = a_i \mid \Lambda_{t-1}^{(j)}\right) \\
&\quad \times \sum_{\lambda_{t+1}^{t+\Delta} \in \mathcal{A}_1^\Delta} \left[\prod_{\tau=0}^\Delta p\left(\mathbf{y}_{t+\tau} \mid \mathbf{Y}_{t+\tau-1}, \Lambda_{t-1}^{(j)}, \lambda_t = a_i, \lambda_{t+1}^{t+\tau}\right) \right. \\
&\quad \left. \times \prod_{\tau=1}^\Delta P\left(\lambda_{t+\tau} \mid \Lambda_{t-1}^{(j)}, \lambda_t = a_i, \lambda_{t+1}^{t+\tau-1}\right) \right].
\end{aligned}
\tag{16}
$$

Note that the second density in (16) is Gaussian and can be computed based on the results of the Kalman filter updates in (15). Draw a sample $\lambda_t^{(j)}$ according to the above sampling density. Append $\lambda_t^{(j)}$ to $\Lambda_{t-1}^{(j)}$ and obtain $\Lambda_t^{(j)}$. Based on this sample, form $\kappa_t^{(j)}$ using the results from the previous step.

(iii) Compute the importance weight. If $\lambda_{t-1}^{(j)} = a_k$ and $\lambda_t^{(j)} = a_i$, then

$$
\begin{aligned}
w_t^{(j)} &= w_{t-1}^{(j)} \frac{p\left(\Lambda_t^{(j)} \mid \mathbf{Y}_{t+\Delta}\right)}{p\left(\Lambda_{t-1}^{(j)} \mid \mathbf{Y}_{t+\Delta-1}\right) P\left(\lambda_t^{(j)} \mid \Lambda_{t-1}^{(j)}, \mathbf{Y}_{t+\Delta}\right)} \\
&\propto w_{t-1}^{(j)} \frac{\sum_{\lambda_t^{t+\Delta} \in \mathcal{A}_1^{\Delta+1}} \left[\prod_{\tau=0}^\Delta p\left(\mathbf{y}_{t+\tau} \mid \mathbf{Y}_{t+\tau-1}, \Lambda_{t-1}^{(j)}, \lambda_t^{t+\tau}\right) \right.}{\sum_{\lambda_t^{t+\Delta-1} \in \mathcal{A}_1^\Delta} \left[\prod_{\tau=0}^{\Delta-1} p\left(\mathbf{y}_{t+\tau} \mid \mathbf{Y}_{t+\tau-1}, \Lambda_{t-1}^{(j)}, \lambda_t^{t+\tau}\right) \right.} \\
&\quad \times \frac{\prod_{\tau=0}^\Delta P\left(\lambda_{t+\tau} \mid \Lambda_{t-1}^{(j)}, \lambda_t^{t+\tau-1}\right)}{\prod_{\tau=0}^{\Delta-1} P\left(\lambda_{t+\tau} \mid \Lambda_{t-1}^{(j)}, \lambda_t^{t+\tau-1}\right)} \\
&\propto \frac{w_{t-1}^{(j)}}{\rho_{t-1,k}^{(j)}} p\left(\mathbf{y}_{t-1} \mid \mathbf{Y}_{t-2}, \Lambda_{t-1}^{(j)}\right) \\
&\quad \times P\left(\lambda_{t-1} = a_k \mid \Lambda_{t-2}^{(j)}\right) \sum_{i=1}^{|\mathcal{A}_1|} \rho_{t,i}^{(j)}.
\end{aligned}
\tag{17}
$$

(iv) Resample if the effective sample size is below a certain threshold, as discussed in Section 2.2.

Finally, as noted in [13], we can use the above delayed-sample method in conjunction with the delayed-weight method. For example, using the delayed-sample method, we generate delayed samples and weights $\{(\Lambda_t^{(j)}, w_t^{(j)})\}_{j=1}^m$ based on observations $\mathbf{Y}_{t+\Delta}$. Then with an additional delay δ, we can use the following delayed-weight method to approximate any inference about the indicator λ_t:

$$
\begin{aligned}
&E\{h(\lambda_t) \mid \mathbf{Y}_{t+\Delta+\delta}\} \\
&\cong \frac{1}{W_{t+\delta}} \sum_{j=1}^m h(\lambda_t^{(j)}) w_{t+\delta}^{(j)}, \quad W_{t+\delta} = \sum_{j=1}^m w_{t+\delta}^{(j)}.
\end{aligned}
\tag{18}
$$

3. MULTILEVEL MIXTURE KALMAN FILTER

Suppose that the indicator space has a multilevel structure. For example, consider the following scenario of a two-level sampling space. The first level is the original sampling space Ω with $\Omega = \{\lambda_i, i = 1, 2, \ldots, N\}$. We also have a higher-level sampling space $\tilde{\Omega}$ with $\tilde{\Omega} = \{c_i, i = 1, 2, \ldots, \tilde{N}\}$. The higher-level sampling space can be obtained as follows. Define the elements (say c_i) in the higher-level sampling space as the centers of subset ω_i in the original sampling space. That is,

$$
c_i = \frac{1}{|\omega_i|} \sum_j \lambda_j \mathbb{1} \quad (\lambda_j \in \omega_i) \ i = 1, 2, \ldots, \tilde{N},
\tag{19}
$$

where ω_i, $i = 1, 2, \ldots, \tilde{N}$, is the subset in the original sampling space

$$
\Omega = \bigcup_i^{\tilde{N}} \omega_i, \quad \omega_i \bigcap \omega_j = \varnothing, \quad i, j = 1, \ldots, \tilde{N}, \ i \neq j.
\tag{20}
$$

We call c_i the parent of the elements in subset ω_i and ω_i the child set of c_i. We can also iterate the above merging procedure on the newly created higher-level sampling space to get an even higher-level sampling space.

For example, we consider the 16-QAM modulation system often used in digital communications. The values of the symbols are taken from the set

$$
\Omega = \{(a, b) : a, b = \pm 0.5, \pm 2.5\}.
\tag{21}
$$

As shown in Figure 1, the sampling space Ω can be divided into four disjoint subsets:

$$
\begin{aligned}
\omega_1 &= \{(0.5, 0.5), (0.5, 2.5), (2.5, 0.5), (2.5, 2.5)\}, \\
\omega_2 &= \{(-0.5, 0.5), (-0.5, 2.5), (-2.5, 0.5), (-2.5, 2.5)\}, \\
\omega_3 &= \{(-0.5, -0.5), (-0.5, -2.5), (-2.5, -0.5), (-2.5, -2.5)\}, \\
\omega_4 &= \{(0.5, -0.5), (0.5, -2.5), (2.5, -0.5), (2.5, -2.5)\}.
\end{aligned}
\tag{22}
$$

Moreover, the centers of these four subspaces are $c_1 = (1.5, 1.5)$, $c_2 = (-1.5, 1.5)$, $c_3 = (-1.5, -1.5)$, and $c_4 = (1.5, -1.5)$. Thus, we have obtained a higher-level sampling space composed of four elements. Then the MKF can draw samples first from the highest-level sampling space and then from the associated child set in the next lower-level sampling space. The procedure is iterated until reaching the original sampling space.

For simplicity, we will use $c_{i,l}$ to represent the ith symbol value in the lth-level sampling space. Assume that there are, in total, L levels of sampling space and the number of elements at the lth level is $|\mathcal{A}_l|$. The original sampling space is defined as the first level. Then the multilevel MKF is summarized as follows.

Algorithm 3 (multilevel MKF). Suppose, at time $(t-1)$, a set of properly weighted samples $\{(\Lambda_{t-1}^{(j)}, \kappa_{t-1}^{(j)}, w_{t-1}^{(j)})\}_{j=1}^m$ is available with respect to $p(\Lambda_{t-1} \mid \mathbf{Y}_{t-1})$. Then at time t, as the new data $\mathbf{y}_t$ becomes available, the following steps are implemented to update each weighted sample.

For $j = 1, \ldots, m$, perform the following steps.

(A1) *Draw the sample $c_{t,L}^{(j)}$ in the Lth-level sampling space.*

(a) Based on the new data $\mathbf{y}_t$, for each $c_{i,L} \in \mathcal{A}_L$, perform a one-step update on the corresponding Kalman filter $\kappa_{t-1}^{(j)}(\lambda_{t-1})$ assuming $c_{t,L} = c_{i,L}$ to obtain

$$
\kappa_{t-1}^{(j)}(\lambda_{t-1}) \xrightarrow{\mathbf{y}_t, c_{i,L}} \kappa_{t,i}^{(j)}(\lambda_{t-1}, c_{i,L}).
\tag{23}
$$

(b) For each $c_{i,L} \in \mathcal{A}_L$, compute the Lth-level sampling density

$$
\begin{aligned}
\rho_{t,L}^{(i,j)} &\triangleq P\left(c_{t,L} = c_{i,L} \mid \Lambda_{t-1}^{(j)}, Y_t\right) \\
&= p\left(\mathbf{y}_t \mid c_{i,L}, \Lambda_{t-1}^{(j)}, \mathbf{Y}_{t-1}\right) P\left(c_{i,L} \mid \Lambda_{t-1}^{(j)}, \mathbf{Y}_{t-1}\right).
\end{aligned}
\tag{24}
$$

Note that by the model (1), the first density in (24) is Gaussian and can be computed based on the Kalman filter update (23).

(c) Draw a sample $c_{t,L}^{(j)}$ from the Lth-level sampling space $\mathcal{A}_L$ according to the above sampling density, that is,

$$
P\left(c_{t,L}^{(j)} = c_{i,L}\right) \propto \rho_{t,L}^{(i,j)}, \quad c_{i,L} \in \mathcal{A}_L.
\tag{25}
$$

(A2) *Draw the sample $c_{t,l}^{(j)}$ in the lth-level sampling space.* First, find the child set $\omega_l^{(j)}$ in the current lth-level sampling space for the drawn sample $c_{t,l+1}^{(j)}$ in the $(l+1)$th-level sampling space, then proceed with three more steps.

(a) For each $c_{i,l}^{(j)}, i = 1, 2, \ldots, |\omega_l^{(j)}|$ in the child set $\omega_l^{(j)}$, perform a one-step update on the corresponding Kalman filter $\kappa_{t-1}^{(j)}(\lambda_{t-1})$ to obtain

$$
\kappa_{t-1}^{(j)}(\lambda_{t-1}) \xrightarrow{\mathbf{y}_t, c_{i,l}^{(j)}} \kappa_{t,i}^{(j)}(\lambda_{t-1}, c_{i,l}^{(j)}).
\tag{26}
$$

(b) For each $c_{i,l}^{(j)}$, compute the sampling density

$$
\begin{aligned}
\rho_{t,l}^{(i,j)} &\triangleq P\big(c_{t,l} = c_{i,l}^{(j)} \mid \Lambda_{t-1}^{(j)}, Y_t\big) \\
&= p\big(\mathbf{y}_t \mid c_{i,l}^{(j)}, \Lambda_{t-1}^{(j)}, \mathbf{Y}_{t-1}\big) P\big(c_{i,l}^{(j)} \mid \Lambda_{t-1}^{(j)}, \mathbf{Y}_{t-1}\big).
\end{aligned}
\tag{27}
$$

Note that the first density in (27) is also Gaussian and can be computed based on the Kalman filter update (26).

(c) Draw a sample $c_{t,l}^{(j)}$ according to the sampling density, that is,

$$
P\big(c_{t,l}^{(j)} = c_{i,l}^{(j)}\big) \propto \rho_{t,l}^{(i,j)}, \quad c_{i,l}^{(j)} \in \omega_l^{(j)}.
\tag{28}
$$

Repeat the above steps for the next level until we draw a sample $\lambda_t^{(j)} = c_{t,1}^{(j)}$ from the original sampling space.

(A3) *Append the symbol $\lambda_t^{(j)}$ to $\Lambda_{t-1}^{(j)}$ and obtain $\Lambda_t^{(j)}$.*

(A4) *Compute the trial sampling probability.* Assuming the drawn sample $c_{t,l}^{(j)} = c_{t,l}^{i,j}$ in the lth-level sampling space and the associated sampling probability is $\rho_{t,l}^{(i,j)}$, then the effective sampling probability $\tilde{P}(\lambda_t^{(j)} \mid \Lambda_{t-1}^{(j)}, \mathbf{Y}_t)$ can be computed as follows:

$$
\tilde{P}\big(\lambda_t^{(j)} \mid \Lambda_{t-1}^{(j)}, \mathbf{Y}_t\big) = \prod_{l=1}^{L} \rho_{t,l}^{(i,j)}.
\tag{29}
$$

(A5) *Compute the importance weight*

$$
\begin{aligned}
w_t^{(j)} &= w_{t-1}^{(j)} \frac{p\big(\Lambda_{t-1}^{(j)}, \lambda_t^{(j)} \mid \mathbf{Y}_t\big)}{p\big(\Lambda_{t-1}^{(j)} \mid \mathbf{Y}_{t-1}\big) \tilde{P}\big(\lambda_t^{(j)} \mid \Lambda_{t-1}^{(j)}, \mathbf{Y}_t\big)} \\
&= w_{t-1}^{(j)} \frac{p\big(\mathbf{y}_t \mid \lambda_t^{(j)}, \mathbf{Y}_{t-1}\big) P\big(\lambda_t^{(j)} \mid \Lambda_{t-1}^{(j)}, \mathbf{Y}_{t-1}\big)}{\prod_{l=1}^{L} \rho_{t,l}^{(i,j)}}.
\end{aligned}
\tag{30}
$$

(A6) *Resample* if the effective sample size is below a certain threshold, as discussed in Section 2.2.

Remark 1 (complexity). Note that the dominant computation required for the above multilevel MKF is the update of the Kalman filter in (23) and (26). Denote $J \triangleq |\mathcal{A}_1|$ and $K_l \triangleq |\omega_l|$. The number of one-step Kalman filter updates in the multilevel MKF is $N = \sum_{l=1}^{L} K_l$. Consider the 16-QAM and its corresponding two-level sampling space shown in Figure 1. There are $N = 8$ one-step Kalman filter updates needed by the multilevel MKF, whereas, the original MKF requires $J = 16$ Kalman updates for each Markov stream. Hence, the computation complexity is reduced by half by the multilevel MKF.

Remark 2 (properties of the weighted samples). From (30), we have

$$
w_{t-1}^{(j)} = w_{t-2}^{(j)} \frac{p\big(\Lambda_{t-2}^{(j)}, \lambda_{t-1}^{(j)} \mid \mathbf{Y}_{t-1}\big)}{p\big(\Lambda_{t-2}^{(j)} \mid \mathbf{Y}_{t-2}\big) \tilde{P}\big(\lambda_{t-1}^{(j)} \mid \Lambda_{t-2}^{(j)}, \mathbf{Y}_{t-1}\big)}.
\tag{31}
$$

Substituting it into (30), and repeating the procedure with $w_{t-2}^{(j)}, \ldots, w_1^{(j)}$, respectively, we finally have

$$
\begin{aligned}
& w_t^{(j)} \\
&= \frac{p\big(\Lambda_{t-1}^{(j)}, \lambda_t^{(j)} \mid \mathbf{Y}_t\big)}{\tilde{P}\big(\lambda_1^{(j)} \mid \Lambda_0^{(j)}, \mathbf{Y}_1\big) \cdots \tilde{P}\big(\lambda_{t-1}^{(j)} \mid \Lambda_{t-2}^{(j)}, \mathbf{Y}_{t-1}\big) \tilde{P}\big(\lambda_t^{(j)} \mid \Lambda_{t-1}^{(j)}, \mathbf{Y}_t\big)}.
\end{aligned}
\tag{32}
$$

Consequently, the samples $\{\Lambda_t^{(j)}, w_t^{(j)}\}_{j=1}^m$ drawn by the above procedure are properly weighted with respect to $p(\Lambda_t \mid \mathbf{Y}_t)$ provided that $\{\Lambda_{t-1}^{(j)}, w_{t-1}^{(j)}\}_{j=1}^m$ are properly weighted with respect to $p(\Lambda_{t-1} \mid \mathbf{Y}_{t-1})$.

4. DELAYED MULTILEVEL MIXTURE KALMAN FILTER

The delayed-sample method is used to generate samples $\{(\lambda_t^{(j)}, \omega_t^{(j)})\}_{j=1}^m$ based on the observations $\mathbf{Y}_{t+\Delta}$, hence making $p(\Lambda_t \mid \mathbf{Y}_{t+\Delta})$ the target distribution at time $(t + \Delta)$. The procedure will provide better Monte Carlo samples since it utilizes the future observations $(\mathbf{y}_{t+1}, \ldots, \mathbf{y}_{t+\Delta})$ in generating the current samples of λ_t. But the algorithm is also more demanding both analytically and computationally because of the need of marginalizing out $\lambda_{t+1}, \ldots, \lambda_{t+\Delta}$.

Instead of exploring the "future" symbol sequences in the original sampling space, our proposed delayed multilevel MKF will marginalize out the future symbols in a higher-level sampling space. That is, for each possible "future" (relative to time $t - 1$) symbol sequence at time $(t + \Delta)$, that is,

$$
(\lambda_t, c_{t+1,l}, \ldots, c_{t+\Delta,l}) \in \mathcal{A}_1 \times \mathcal{A}_l^\Delta \quad (l > 1),
\tag{33}
$$

where $c_{t,l}$ is the symbol in the lth-level sampling space, we compute the value of a Δ-step Kalman filter $\{\kappa_{t+\tau}^{(j)}(\lambda_t, c_{t+1,l}^{t+\tau})\}_{\tau=1}^\Delta$, where

$$
\begin{aligned}
& \kappa_{t+\tau}^{(j)}(\lambda_t, c_{t+1,l}^{t+\tau}) \\
&\triangleq \big[\boldsymbol{\mu}_{t+\tau}\big(\Lambda_{t-1}^{(j)}, \lambda_t, c_{t+1,l}^{t+\tau}\big), \boldsymbol{\Sigma}_{t+\tau}\big(\Lambda_{t-1}^{(j)}, \lambda_t, c_{t+1,l}^{t+\tau}\big)\big], \quad \tau = 1, \ldots, \Delta,
\end{aligned}
\tag{34}
$$

with $c_{a,l}^b \triangleq (c_{a,l}, c_{a+1,l}, \ldots, c_{b,l})$. The delayed multilevel MKF recursively propagates the samples properly weighted for $p(\Lambda_{t-1} \mid \mathbf{Y}_{t+\Delta-1})$ to those for $p(\Lambda_t \mid \mathbf{Y}_{t+\Delta})$ and is summarized as follows.

Algorithm 4 (delayed multilevel MKF). Suppose, at time $(t - 1)$, a set of properly weighted samples $\{(\Lambda_{t-1}^{(j)}, \kappa_{t-1}^{(j)}, w_{t-1}^{(j)})\}_{j=1}^m$ is available with respect to $p(\Lambda_{t-1} \mid \mathbf{Y}_{t+\Delta-1})$. Then at time t, as the new data $\mathbf{y}_{t+\Delta}$ becomes available, the following steps are implemented to update each weighted sample.

For $j = 1, \ldots, m$, apply the following steps.

(A1) For each $\lambda_t = a_i \in \mathcal{A}_1$, and for each $c_{t+1,l}^{t+\Delta} \in \mathcal{A}_l^\Delta$, perform the update on the corresponding Kalman filter

$\kappa_{t+\Delta-1}^{(j)}(\lambda_t, c_{t+1,l}^{t+\Delta-1})$, that is,

$$\kappa_{t+\tau-1}^{(j)}(\lambda_t, c_{t+1,l}^{t+\tau-1}) \xrightarrow{\mathbf{y}_{t+\tau}, c_{t,l}} \kappa_{t+\tau}^{(j)}(\lambda_t, c_{t+1,l}^{t+\tau}). \quad (35)$$

(A2) For each $a_i \in \mathcal{A}_1$, compute the sampling density

$$\begin{aligned}
\rho_{t,i}^{(j)} &\triangleq P\left(\lambda_t = a_i \mid \mathbf{\Lambda}_{t-1}^{(j)}, \mathbf{y}_{t+\Delta}\right) \\
&= \sum_{c_{t+1,l}^{t+\Delta} \in \mathcal{A}_l^{\Delta}} p\left(\mathbf{y}_t^{t+\Delta}, \lambda_t, c_{t+1,l}^{t+\Delta} \mid \mathbf{\Lambda}_{t-1}^{(j)}, \mathbf{Y}_{t-1}^{(j)}\right) \\
&\propto P\left(\lambda_t = a_i \mid \mathbf{\Lambda}_{t-1}^{(j)}\right) \\
&\quad \times \sum_{c_{t+1,l}^{t+\Delta} \in \mathcal{A}_l^{\Delta}} \left[\prod_{\tau=0}^{\Delta} p\left(\mathbf{y}_{t+\tau} \mid \mathbf{Y}_{t+\tau-1}, \mathbf{\Lambda}_{t-1}^{(j)}, \lambda_t = a_i, c_{t+1,l}^{t+\tau}\right) \right. \\
&\quad \left. \times \prod_{\tau=1}^{\Delta} P\left(c_{t+\tau,l} \mid c_{t+1,l}^{t+\tau-1}, \lambda_t = a_i, \mathbf{\Lambda}_{t-1}^{(j)}\right) \right].
\end{aligned}$$
$$(36)$$

(A3) Draw a sample $\lambda_t^{(j)}$ according to the above sampling density, that is,

$$P\left(\lambda_t^{(j)} = \lambda_i\right) \propto \rho_{t,i}^{(j)}. \quad (37)$$

(A4) Append the sample $\lambda_t^{(j)}$ to $\mathbf{\Lambda}_{t-1}^{(j)}$ and obtain $\mathbf{\Lambda}_t^{(j)}$.

(A5) Compute the importance weight

$$\begin{aligned}
w_t^{(j)} &= w_{t-1}^{(j)} \frac{p\left(\mathbf{\Lambda}_{t-1}^{(j)}, \lambda_t^{(j)} \mid \mathbf{Y}_{t+\Delta}\right)}{p\left(\mathbf{\Lambda}_{t-1}^{(j)} \mid \mathbf{Y}_{t+\Delta-1}\right) P\left(\lambda_t^{(j)} \mid \mathbf{\Lambda}_{t-1}^{(j)}, \mathbf{Y}_{t+\Delta}\right)} \\
&\propto w_{t-1}^{(j)} \frac{\sum_{C_{t+1,l}^{t+\Delta} \in \mathcal{A}_l^{\Delta}} p\left(\mathbf{y}_t^{t+\Delta}, c_{t+1,l}^{t+\Delta}, \lambda_t^{(j)} \mid \mathbf{\Lambda}_{t-1}^{(j)}, \mathbf{Y}_{t-1}\right)}{\sum_{C_{t,l}^{t+\Delta-1} \in \mathcal{A}_l^{\Delta}} p\left(\mathbf{y}_t^{t+\Delta-1}, c_{t,l}^{t+\Delta-1} \mid \mathbf{\Lambda}_{t-1}^{(j)}, \mathbf{Y}_{t-1}\right) \rho_{t,i}^{(j)}} \\
&= w_{t-1}^{(j)} \frac{\sum_{C_{t+1,l}^{t+\Delta} \in \mathcal{A}_l^{\Delta}} \left[\prod_{\tau=1}^{\Delta} p\left(y_{t+\tau} \mid \mathbf{Y}_{t+\tau-1}, C_{t+1,l}^{t+\tau}, \lambda_t^{(j)}, \mathbf{\Lambda}_{t-1}^{(j)}\right) \right.}{\sum_{c_{t,l}^{t+\Delta-1} \in \mathcal{A}_l^{\Delta}} \left[\prod_{\tau=0}^{\Delta-1} p\left(y_{t+\tau} \mid \mathbf{Y}_{t+\tau-1}, C_{t,l}^{t+\tau}, \mathbf{\Lambda}_{t-1}^{(j)}\right) \right.} \\
&\quad \frac{\left. P\left(\lambda_t^{(j)} \mid \mathbf{\Lambda}_{t-1}^{(j)}, \mathbf{Y}_{t-1}\right) \prod_{\tau=1}^{\Delta} P\left(c_{t+\tau,l} \mid c_{t+1,l}^{t+\tau-1}, \mathbf{\Lambda}_t^{(j)}\right) \right]}{\left. \prod_{\tau=0}^{\Delta-1} P\left(c_{t+\tau,l} \mid C_{t,l}^{t+\tau-1}, \mathbf{\Lambda}_{t-1}^{(j)}\right) \rho_{t,i}^{(j)} \right]}.
\end{aligned}$$
$$(38)$$

(A6) Do resampling if the effective sample size is below a certain threshold, as discussed in Section 2.2.

Remark 3 (properties of the weighted samples). Similar to the multilevel sampling algorithm in Section 3, it can be shown that the samples $\{\mathbf{\Lambda}_t^{(j)}, w_t^{(j)}\}_{j=1}^m$ drawn by the above procedure are properly weighted with respect to $p(\mathbf{\Lambda}_t \mid \mathbf{Y}_{t+\Delta})$ provided that $\{\mathbf{\Lambda}_{t-1}^{(j)}, w_{t-1}^{(j)}\}_{j=1}^m$ are properly weighted with respect to $p(\mathbf{\Lambda}_{t-1} \mid \mathbf{Y}_{t+\Delta-1})$. However, the likelihood function $p(y_{t+\tau} \mid \mathbf{Y}_{t+\tau-1}, C_{t,l}^{t+\tau}, \mathbf{\Lambda}_{t-1}^{(j)})$ is not simply a Gaussian distribution any more, but a mixture of Gaussian components. Since the mixture of Gaussian distribution is implausible to be achieved within the rough sampling space, it has to be approximated with a Gaussian distribution computed by the Kalman filter assuming that the elements in the higher-level

sampling space are transmitted. Therefore, some bias will be introduced into the computation of the weight. On the other hand, we make use of more information in the approximation of better distribution $p(\mathbf{\Lambda}_{t-1}^{(j)}, \lambda_t^{(j)} \mid \mathbf{Y}_{t+\Delta})$, which makes the algorithm more efficient than the original MKF.

Remark 4 (properly weighted samples). To mitigate the bias problem introduced in the weight computation, instead of (38), we can use the following importance weight:

$$\begin{aligned}
w_t^{(j)} &= w_{t-1}^{(j)} \frac{p\left(\mathbf{\Lambda}_{t-1}^{(j)}, \lambda_t^{(j)} \mid \mathbf{Y}_t\right)}{p\left(\mathbf{\Lambda}_{t-1}^{(j)} \mid \mathbf{Y}_{t-1}\right) \tilde{P}\left(\lambda_t^{(j)} \mid \mathbf{\Lambda}_{t-1}^{(j)}, \mathbf{Y}_{t+\Delta}\right)} \\
&= w_{t-1}^{(j)} \frac{p\left(\mathbf{y}_t \mid \lambda_t^{(j)}, \mathbf{Y}_{t-1}\right) P\left(\lambda_t^{(j)} \mid \mathbf{\Lambda}_{t-1}^{(j)}, \mathbf{Y}_{t-1}\right)}{\rho_{t,i}^{(j)}}.
\end{aligned}$$
$$(39)$$

Similar to the multilevel sampling algorithm in Section 3, it is easily seen that the samples $\{\mathbf{\Lambda}_t^{(j)}, w_t^{(j)}\}_{j=1}^m$ drawn by the above procedure are properly weighted with respect to $p(\mathbf{\Lambda}_t \mid \mathbf{Y}_t)$ provided that $\{\mathbf{\Lambda}_{t-1}^{(j)}, w_{t-1}^{(j)}\}_{j=1}^m$ are properly weighted with respect to $p(\mathbf{\Lambda}_{t-1} \mid \mathbf{Y}_{t-1})$. Since the whole procedure is just to get better samples based on the future information, delayed weight may be very effective. Furthermore, there is no bias anymore in the weight computation although we still approximate the mixture Gaussian distribution with a Gaussian distribution as in Remark 3.

Remark 5 (complexity). Note that the dominant computation required for the above delayed multilevel MKF is mainly in the first step. Denote $J \triangleq |\mathcal{A}_1|$ and $M \triangleq |\mathcal{A}_l|$. The number of one-step Kalman filter updates in the delayed multilevel MKF can be computed as follows:

$$N = J + JM + \cdots + JM^{\Delta} = J \frac{M^{\Delta+1} - 1}{M - 1}. \quad (40)$$

Note that the delayed-sample MKF requires $J^{\Delta+1}$ Kalman updates at each time. Since usually $M \ll J$, compared with the delayed-sample MKF, the computational complexity of the delayed multilevel MKF is significantly reduced.

Remark 6 (alternative sampling method). To further reduce the computational complexity in the first step, we can take the alternative sampling method composed of the following steps.

Algorithm 5 (alternative sampling method). (1) At time $t + \Delta$, for each $c_{t,l}^{t+\Delta} \in \mathcal{A}_l^{\Delta}$, perform the update on the corresponding Kalman filter $\kappa_{t+\Delta-1}^{(j)}(c_{t,l}^{t+\Delta-1})$, that is,

$$\kappa_{t+\tau-1}^{(j)}(c_{t,l}^{t+\tau-1}) \xrightarrow{\mathbf{y}_{t+\tau}, c_{t+\tau,l}} \kappa_{t+\tau}^{(j)}(c_{t,l}^{t+\tau}). \quad (41)$$

(2) Select K paths from $c_{t+1,l}^{t+\Delta}$ based on the computed weight

$$\prod_{\tau=0}^{\Delta} p\left(\mathbf{y}_{t+\tau} \mid \mathbf{Y}_{t+\tau-1}, \mathbf{\Lambda}_{t-1}^{(j)}, c_{t,l}^{t+\tau}\right) P\left(c_{t+\tau,l} \mid c_{t,l}^{t+\tau-1}, \mathbf{\Lambda}_{t-1}^{(j)}\right). \quad (42)$$

(3) For each $\lambda_t = a_i \in \mathcal{A}_1$, and for each path k in the selected K paths, perform the update on the corresponding Kalman filter $\kappa_{t+\Delta-1}^{(j)}(\lambda_t, c_{t+1,l}^{t+\Delta-1,k})$, that is,

$$\kappa_{t+\tau-1}^{(j)}(\lambda_t, c_{t+1,l}^{t+\tau-1,k}) \xrightarrow{\mathbf{y}_{t+\tau}, c_{t+\tau,l}k} \kappa_{t+\tau}^{(j)}(\lambda_t, c_{t+1,l}^{t+\tau,k}). \qquad (43)$$

(4) For each $a_i \in \mathcal{A}_1$, compute the sampling density

$$\rho_{t,i}^{(j)} \triangleq P\left(\lambda_t = a_i \mid \mathbf{\Lambda}_{t-1}^{(j)}\right)$$
$$\times \sum_{k=1}^{K} \left[\prod_{\tau=0}^{\Delta} p\left(\mathbf{y}_{t+\tau} \mid \mathbf{Y}_{t+\tau-1}, \mathbf{\Lambda}_{t-1}^{(j)}, \lambda_t = a_i, c_{t+1,l}^{t+\tau,k}\right) \right.$$
$$\left. \times \prod_{\tau=1}^{\Delta} P\left(c_{t+\tau,l}^{k} \mid c_{t+1,l}^{t+\tau-1,k}, \lambda_t = a_i, \mathbf{\Lambda}_{t-1}^{(j)}\right) \right]. \qquad (44)$$

The weight calculation can be computed by (40) for the target distribution $p(\lambda_t \mid \mathbf{Y}_t)$ or (45) for the target distribution $p(\lambda_t \mid \mathbf{Y}_{t+\Delta})$. Besides the bias that resulted from the Gaussian approximation in Remark 3, the latter also introduces new bias because of the summation over K selected paths, other than the whole sampling space in higher level. However, the first one does not introduce any bias as in Remark 4. Denote $J \triangleq |\mathcal{A}_1|$ and $M \triangleq |\mathcal{A}_l|$. The number of one-step Kalman filter updates in the delayed multilevel MKF can be computed as $N = JK + M^{\Delta}$.

Remark 7 (the choice of the parameters). Note that the performance of the delayed multilevel MKF is mainly dominated by the two important parameters: the number of prediction steps Δ and the specific level of the sampling space $\mathcal{A}_l$. With the same computation, the delayed multilevel MKF can see further "future" steps (larger Δ) with a coarser-level sampling space (larger l), whereas it can see the "future" samples in a finer sampling space (smaller l), but with smaller Δ steps.

Remark 8 (multiple sampling space). In Algorithm 5, we can also use different sampling spaces for different delay steps. That is, we can gradually increase the sampling space from the lower level to the higher level with an increase in the delay step.

5. SIMULATIONS

We consider the problem of adaptive detection in flat-fading channels in the presence of Gaussian noise. This problem is of fundamental importance in communication theory and an array of methodologies have been developed to tackle this problem. Specifically, the optimal detector for flat-fading channels with known channel statistics is studied in [24, 25], which has a prohibitively high complexity. Suboptimal receivers in flat-fading channels employ a two-stage structure, with a channel estimation stage followed by a sequence detection stage. Channel estimation is typically implemented by a Kalman filter or a linear predictor, and is facilitated by per-survivor processing (PSP) [26], decision feedback [27], pilot symbols [28], or a combination of the above [29].

Other suboptimal receivers for flat-fading channels include the method based on a combination of a hidden Markov model and a Kalman filtering [30], and the approach based on the expectation-maximization (EM) algorithm [31].

In the communication system, the transmitted data symbols $\{s_t\}$ take values from a finite alphabet set $\mathcal{A}_1 = \{a_1,\ldots,a_{|\mathcal{A}_1|}\}$, and each symbol is transmitted over a Rayleigh fading channel. As shown in [32, 33], the fading process is adequately represented by an ARMA model, whose parameters are chosen to match the spectral characteristics of the fading process. That is, the fading process is modelled by the output of a lowpass Butterworth filter of order r driven by white Gaussian noise

$$\{\alpha_t\} = \frac{\Theta(D)}{\Phi(D)}\{u_t\}, \qquad (45)$$

where D is the back-shift operator D^k, $u_t \triangleq u_{t-k}$; $\Phi(z) \triangleq \phi_r z^r + \cdots + \phi_1 z + 1$; $\Theta(z) \triangleq \theta_r z^r + \cdots + \theta_1 z + \theta_0$; and $\{u_t\}$ is a white complex Gaussian noise sequence with unit variance and independent real and complex components. The coefficients $\{\phi_i\}$ and $\{\theta_i\}$, as well as the order r of the Butterworth filter, are chosen so that the transfer function of the filter matches the power spectral density of the fading process, which in turn is determined by the channel Doppler frequency. On the other hand, a simpler method, which uses a two-path model to build ARMA process, can be found in [34]; the results there closely approximate more complex path models. Then such a communication system has the following state-space model form:

$$\mathbf{x}_t = \mathbf{F}\mathbf{x}_{t-1} + \mathbf{g}u_t, \qquad (46)$$
$$y_t = s_t \mathbf{h}^H \mathbf{x}_t + \sigma v_t, \qquad (47)$$

where

$$\mathbf{F} \triangleq \begin{pmatrix} -\phi_1 & -\phi_2 & \cdots & -\phi_r & 0 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix}, \qquad \mathbf{g} \triangleq \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \qquad (48)$$

The fading coefficient sequence $\{\alpha_t\}$ can then be written as

$$\alpha_t = \mathbf{h}^H \mathbf{x}_t, \quad \mathbf{h} \triangleq \begin{bmatrix} \theta_0 & \theta_1 & \cdots & \theta_r \end{bmatrix}. \qquad (49)$$

In the state-space model, $\{u_t\}$ in (46) and $\{v_t\}$ in (47) are the white complex Gaussian noise sequences with unit variance and independent real and imaginary components:

$$u_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}_c(0,1), \qquad v_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}_c(0,1). \qquad (50)$$

In our simulations, the fading process is specifically modeled by the output of a Butterworth filter of order $r = 3$ driven by a complex white Gaussian noise process. The cutoff frequency of this filter is 0.05, corresponding to a normalized Doppler frequency (with respect to the symbol rate $1/T$) $f_d T = 0.05$, which is a fast-fading scenario. That is, the fading
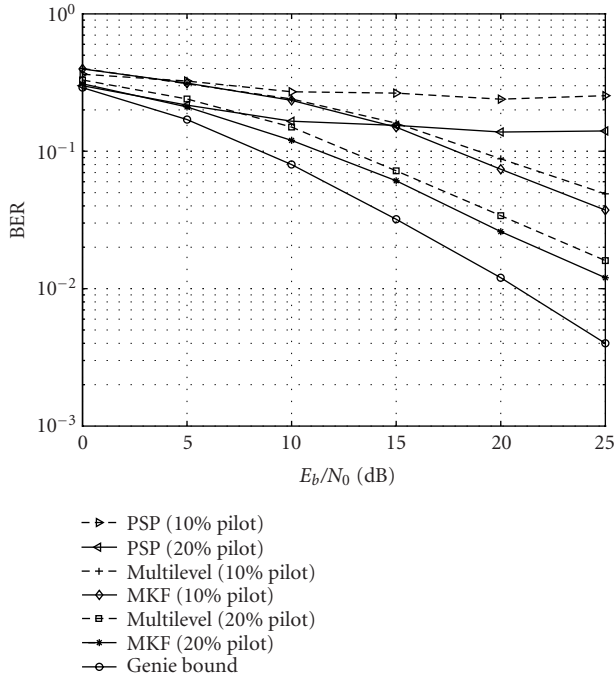
FIGURE 2: The BER performance of the PSP, MKF, and multilevel MKF for pilot-aided 16-QAM over flat-fading channels.
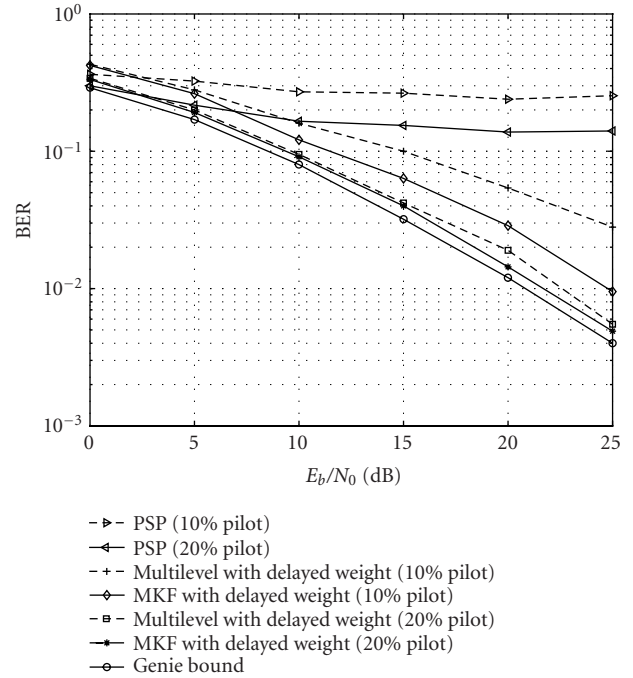
FIGURE 3: The BER performance of the PSP, the MKF with delayed weight ($\delta = 10$), and the multilevel MKF with delayed weight ($\delta = 10$) for pilot-aided 16-QAM over flat-fading channels.

coefficients $\{\alpha_t\}$ are modeled by the following ARMA(3,3) process:

$$\alpha_t - 2.37409\alpha_{t-1} + 1.92936\alpha_{t-2} - 0.53208\alpha_{t-3}$$
$$= 10^{-2}(0.89409u_t + 2.68227u_{t-1} + 2.68227u_{t-2} \quad (51)$$
$$+ 0.89409u_{t-3}),$$

where $u_t \sim \mathcal{N}_c(0, 1)$. The filter coefficients in (51) are chosen such that $\mathrm{Var}\{\alpha_t\} = 1$. However, a simpler method, which uses a two-path model to build ARMA process, can be found in [34].

We then apply the proposed multilevel MKF methods to the problem of online estimation of the a posteriori probability of the symbol s_t based on the received signals up to time t. That is, at time t, we need to estimate

$$P(s_t = a_i \mid \mathbf{Y}_t), \quad a_i \in \mathcal{A}_1. \quad (52)$$

Then a hard maximum a posteriori (MAP) decision on symbol s_t is given by

$$\hat{s}_t = \arg \max_{a_i \in \mathcal{A}_1} P(s_t = a_i \mid \mathbf{Y}_t). \quad (53)$$

If we obtain a set of Monte Carlo samples of the transmitted symbols $\{(\mathbf{S}_t^{(j)}, w_t^{(j)})\}_{j=1}^m$, properly weighted with respect to $p(\mathbf{S}_t \mid \mathbf{Y}_t)$, then the a posteriori symbol probability in (52) is approximated by

$$p(s_t = a_i \mid \mathbf{Y}_t) \cong \frac{1}{W_t} \sum_{j=1}^m \mathbb{1}(s_t^{(j)} = a_i) w_t^{(j)}, \quad a_i \in \mathcal{A}_1, \quad (54)$$

with $W_t \triangleq \sum_{j=1}^m w_t^{(j)}$.

In this paper, we use the 16-QAM modulation. Note that the 16-QAM modulation has much more phase ambiguities than the BPSK in [13]. We have provided the following two schemes to resolve phase ambiguities.

Scheme 1 (pilot-assisted). Pilot symbols are inserted periodically every fixed length of symbols; the similar scheme was used in [20]. In this paper, 10% and 20% pilot symbols are studied.

Scheme 2 (differential 16-QAM). We view the 16-QAM as a pair of QPSK symbols. Then two differential QPSKs will be used to resolve the phase ambiguity. Given the transmitted symbol s_{t-1} and information symbol d_t, they can be represented by the QPSK symbol pair as $(r_{s_{t-1},1}, r_{s_{t-1},2})$ and $(r_{d_t,1}, r_{d_t,2})$, respectively. Then the differential modulation scheme is given by

$$r_{s_t,1} = r_{s_{t-1},1} \cdot r_{d_t,1}, \qquad r_{s_t,2} = r_{s_{t-1},2} \cdot r_{d_t,2}. \quad (55)$$

The two-QPSK symbol pair $(r_{s_t,1}, r_{s_t,2})$ will be mapped to the 16-QAM symbols and then transmitted through the fading channel. The traditional differential receiver performs the following steps:

$$r_{d_t,1} = r_{d_t,1} \cdot r_{s_{t-1},1}^*, \qquad r_{d_t,2} = r_{s_t,2} \cdot r_{s_{t-1},2}^*. \quad (56)$$

We next show the performance of the multilevel MKF algorithm for detecting 16-QAM over flat-fading channels. The receiver implements the decoding algorithms discussed
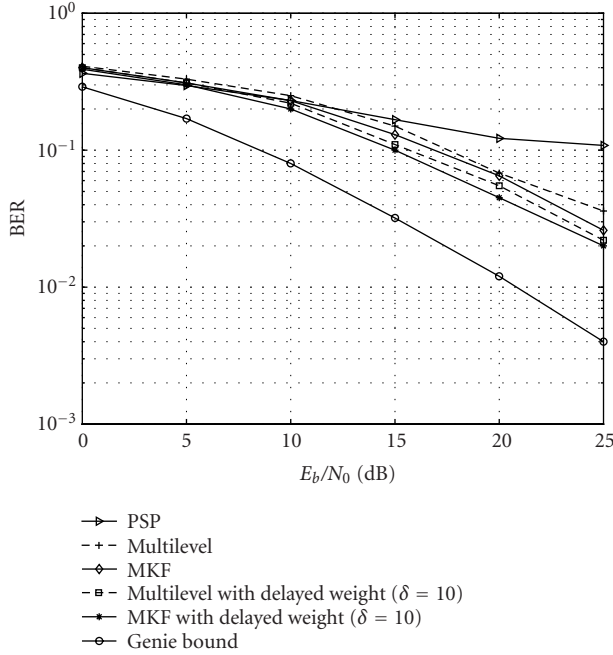
FIGURE 4: The BER performance of the PSP, the MKF, and the multilevel MKF for differential 16-QAM over flat-fading channels. The delayed-weight method is used with $\delta = 10$.

in Section 3 in combination with the delayed-weight method under Schemes 1 and 2 discussed above. In our simulations, we take $m = 50$ Markov streams. The length of the symbol sequence is 256. We first show the bit error rate (BER) performance versus the signal-to-noise (SNR) by the PSP, the multilevel MKF, and the MKF receiver under different pilot schemes without delayed weight in Figure 2 and with delayed weight in Figure 3. The numbers of bit errors were collected over 50 independent simulations at low SNR or more at high SNR. In these figures, we used the "genie-aided lower bound."[1] The BER performance of PSP is far from the genie bound at SNR higher than 10 dB. But the performance of the multilevel MKF with 20% pilot symbol is very close to the genie bound. Furthermore, with the delayed-weight method, the performance of the multilevel MKF can be significantly improved. We next show the BER performance in Figure 4 under the differential coding scheme. As in the pilot-assisted scheme, the BER performance of PSP is far from the genie bound at SNR higher than 15 dB. On the contrary, the performance of the multilevel MKF is very close to that of the original MKF although there is a 5 dB gap between the genie bound and the performance of the multilevel MKF.
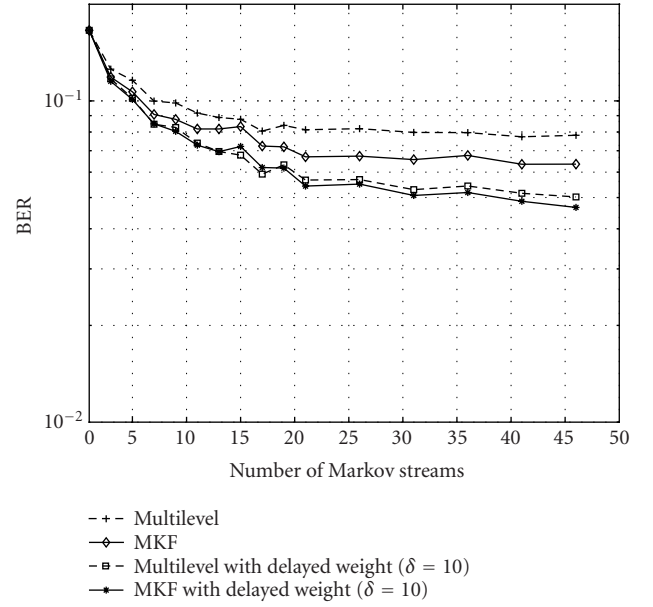

FIGURE 5: The BER performance of the MKF, and the multilevel MKF for differential 16-QAM over flat-fading channels versus the number of Markov streams. The delayed-weight method is used with $\delta = 10$.
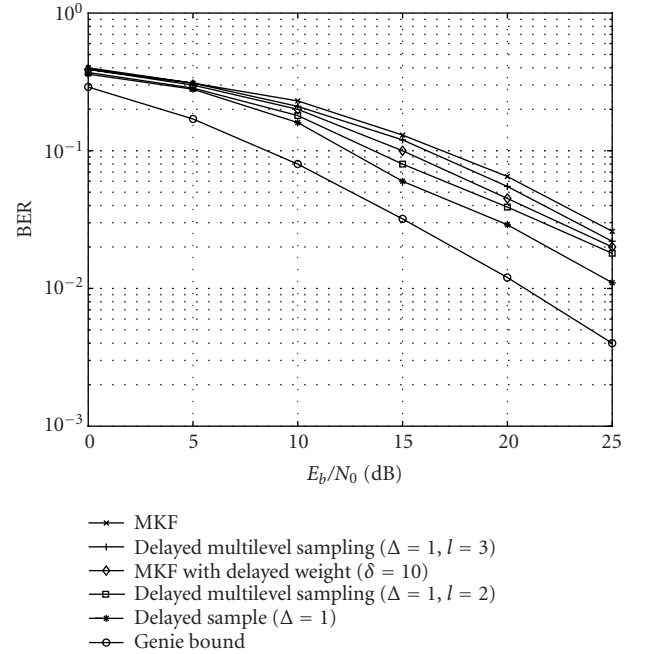

FIGURE 6: The BER performance of the delayed multilevel MKF with the second ($l = 2$)-level or the third ($l = 3$)-level sampling space for the differential 16-QAM system over flat-fading channels. The BER performance of the delayed-sample method is also shown.

The BER performance of the MKF and the multilevel MKF with or without delayed weight versus the number of Markov streams is shown in Figure 5 under the differential coding scheme. The BER is gradually improved from the

---

[1] The genie-aided lower bound is obtained as follows. We assume that at each time t, a genie provides the receiver with an observation of the modulation-free channel coefficient corrupted by additive white Gaussian noise with the same variance, that is, $\tilde{y}_t = \alpha_t + \tilde{n}_t$, where $\tilde{n}_t \sim \mathcal{N}_c(0, \sigma^2)$. The receiver employs a Kalman filter to track the fading process based on the information provided by the genie, that is, it computes $\hat{\alpha}_t = E\{\alpha_t \mid \tilde{y}_t\}$. An estimate of the transmitted 16-QAM is obtained by slicing $(\mathbf{y}_t \hat{\alpha}_t^\star)$.

value 0.16 to the value about 0.08 for multilevel MKF, 0.07 for MKF, 0.065 for multilevel MKF with delayed weight, and 0.062 for MKF with delayed weight with 25 Markov streams. However, the BER performance can not be improved anymore with more than 25 streams. Therefore, the optimal number of Markov streams will be 25 in this example.

Next, we illustrate the performance of the delayed multilevel MKF. The receiver implements the decoding algorithms discussed in Section 4. We show the BER performance versus SNR in Figure 6, computed by the delayed-sample or the delayed multilevel MKF with one delayed step ($\Delta = 1$). The BER performance of the MKF and the MKF with delayed weight is also plotted in the same figure. In the delayed multilevel MKF, we implement two schemes for choosing the sampling space for the "future" symbols. In the first scheme, we choose the second-level ($l = 2$) sampling space $\mathcal{A}_2 = \{c_1, c_2, c_3, c_4\}$, where c_i, $i = 1, \ldots, 4$, are the solid circles shown in Figure 1; and in the second scheme, we choose the third-level ($l = 3$) sampling space $\mathcal{A}_3 = \{c_1 + c_2, c_3 + c_4\}$. It is seen that the BER of the multilevel MKF is very close to that of the delayed-sample algorithm. It is also seen that the performance of the multilevel MKF method is conditioned on the specific level sampling space. The performance of the delayed multilevel MKF based on the second-level sampling space $\mathcal{A}_2$ is nearly 2 dB better than that based on the third-level sampling space $\mathcal{A}_3$.

6. CONCLUSION

In this paper, we have developed a new sequential Monte Carlo (SMC) sampling method—the multilevel mixture Kalman filter (MKF)—under the framework of MKF for conditional dynamic linear systems. This new scheme generates random streams using sequential importance sampling (SIS), based on the multilevel or hierarchical structure of the indicator random variables. This technique can also be used in conjunction with the delayed estimation methods, resulting in a delayed multilevel MKF. Moreover, the performance of both the multilevel MKF and the delayed multilevel MKF can be further enhanced when employed in conjunction with the delayed-weight method.

We have also applied the multilevel MKF algorithm and the delayed multilevel MKF algorithm to solve the problem of adaptive detection in flat-fading communication channels. It is seen that compared with the receiver algorithm based on the original MKF, the proposed multilevel MKF techniques offer very good performance. It is also seen that the receivers based on the delayed multilevel MKF can achieve similar performance as that based on the delayed-sample MKF, but with a much lower computational complexity.

ACKNOWLEDGMENT

REFERENCES

[1] C. Andrieu, N. de Freitas, and A. Doucet, "Robust full Bayesian learning for radial basis networks," *Neural Computation*, vol. 13, no. 10, pp. 2359–2407, 2001.

[2] E. Beadle and P. Djuric, "A fast-weighted Bayesian bootstrap filter for nonlinear model state estimation," *IEEE Trans. on Aerospace and Electronics Systems*, vol. 33, no. 1, pp. 338–343, 1997.

[3] R. Chen and J. S. Liu, "Mixture Kalman filters," *Journal of the Royal Statistical Society: Series B*, vol. 62, no. 3, pp. 493–508, 2000.

[4] C.-M. Cho and P. Djuric, "Bayesian detection and estimation of cisoids in colored noise," *IEEE Trans. Signal Processing*, vol. 43, no. 12, pp. 2943–2952, 1995.

[5] P. M. Djuric and J. Chun, "An MCMC sampling approach to estimation of nonstationary hidden Markov models," *IEEE Trans. Signal Processing*, vol. 50, no. 5, pp. 1113–1123, 2002.

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

[7] 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.

[8] W. Fong, S. J. Godsill, A. Doucet, and M. West, "Monte Carlo smoothing with application to audio signal enhancement," *IEEE Trans. Signal Processing*, vol. 50, no. 2, pp. 438–449, 2002.

[9] Y. Huang and P. Djurić, "Multiuser detection of synchronous code-division multiple-access signals by perfect sampling," *IEEE Trans. Signal Processing*, vol. 50, no. 7, pp. 1724–1734, 2002.

[10] A. Kong, J. S. Liu, and W. H. Wong, "Sequential imputations and Bayesian missing data problems," *Journal of the American Statistical Association*, vol. 89, no. 425, pp. 278–288, 1994.

[11] 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.

[12] M. K. Pitt and N. Shephard, "Filtering via simulation: auxiliary particle filters," *Journal of the American Statistical Association*, vol. 94, no. 446, pp. 590–599, 1999.

[13] R. Chen, X. Wang, and J. S. Liu, "Adaptive joint detection and decoding in flat-fading channels via mixture Kalman filtering," *IEEE Transactions on Information Theory*, vol. 46, no. 6, pp. 2079–2094, 2000.

[14] X. Wang, R. Chen, and D. Guo, "Delayed-pilot sampling for mixture Kalman filter with application in fading channels," *IEEE Trans. Signal Processing*, vol. 50, no. 2, pp. 241–254, 2002.

[15] A. Doucet, N. J. Gordon, and V. Kroshnamurthy, "Particle filters for state estimation of jump Markov linear systems," *IEEE Trans. Signal Processing*, vol. 49, no. 3, pp. 613–624, 2001.

[16] M. West and J. Harrison, *Bayesian Forecasting and Dynamic Models*, Springer-Verlag, New York, NY, USA, 1989.

[17] R. Y. Rubinstein, *Simulation and the Monte Carlo Method*, John Wiley & Sons, New York, NY, USA, 1981.

[18] J. MacCormick and A. Blake, "A probabilistic exclusion principle for tracking multiple objects," *International Journal of Computer Vision*, vol. 39, no. 1, pp. 57–71, 2000.

[19] D. Guo, X. Wang, and R. Chen, "Wavelet-based sequential Monte Carlo blind receivers in fading channels with unknown channel statistics," *IEEE Trans. Signal Processing*, vol. 52, no. 1, pp. 227–239, 2004.

[20] E. Punskaya, C. Andrieu, A. Doucet, and W. Fitzgerald, "Particle filtering for demodulation in fading channels with non-Gaussian additive noise," *IEEE Trans. Communications*, vol. 49, no. 4, pp. 579–582, 2001.

[21] D. Guo and X. Wang, "Blind detection in MIMO systems via sequential Monte Carlo," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 3, pp. 464–473, 2003.

[22] J. Zhang and P. Djurić, "Joint estimation and decoding of space-time trellis codes," *EURASIP Journal on Applied Signal Processing*, vol. 2002, no. 3, pp. 305–315, 2002.

[23] Z. Yang and X. Wang, "A sequential Monte Carlo blind receiver for OFDM systems in frequency-selective fading channels," *IEEE Trans. Signal Processing*, vol. 50, no. 2, pp. 271–280, 2002.

[24] R. Haeb and H. Meyr, "A systematic approach to carrier recovery and detection of digitally phase modulated signals of fading channels," *IEEE Trans. Communications*, vol. 37, no. 7, pp. 748–754, 1989.

[25] D. Makrakis, P. T. Mathiopoulos, and D. P. Bouras, "Optimal decoding of coded PSK and QAM signals in correlated fast fading channels and AWGN: a combined envelope, multiple differential and coherent detection approach," *IEEE Trans. Communications*, vol. 42, no. 1, pp. 63–75, 1994.

[26] G. M. Vitetta and D. P. Taylor, "Maximum likelihood decoding of uncoded and coded PSK signal sequences transmitted over Rayleigh flat-fading channels," *IEEE Trans. Communications*, vol. 43, no. 11, pp. 2750–2758, 1995.

[27] Y. Liu and S. D. Blostein, "Identification of frequency nonselective fading channels using decision feedback and adaptive linear prediction," *IEEE Trans. Communications*, vol. 43, no. 2/3/4, pp. 1484–1492, 1995.

[28] H. Kong and E. Shwedyk, "On channel estimation and sequence detection of interleaved coded signals over frequency nonselective Rayleigh fading channels," *IEEE Trans. Vehicular Technology*, vol. 47, no. 2, pp. 558–565, 1998.

[29] G. T. Irvine and P. J. McLane, "Symbol-aided plus decision-directed reception for PSK/TCM modulation on shadowed mobile satellite fading channels," *IEEE Journal on Selected Areas in Communications*, vol. 10, no. 8, pp. 1289–1299, 1992.

[30] I. B. Collings and J. B. Moore, "An adaptive hidden Markov model approach to FM and M-ary DPSK demodulation in noisy fading channels," *Signal Processing*, vol. 47, no. 1, pp. 71–84, 1995.

[31] C. N. Georghiades and J. C. Han, "Sequence estimation in the presence of random parameters via the EM algorithm," *IEEE Trans. Communications*, vol. 45, no. 3, pp. 300–308, 1997.

[32] G. L. Stüber, *Principles of Mobile Communication*, Kluwer Academic Publishers, Boston, Mass, USA, 2000.

[33] R. A. Ziegler and J. M. Cioffi, "Estimation of time-varying digital radio channels," *IEEE Trans. Vehicular Technology*, vol. 41, no. 2, pp. 134–151, 1992.

[34] A. Duel-Hallen, "Decorrelating decision-feedback multiuser detector for synchronous code-division multiple-access channel," *IEEE Trans. Communications*, vol. 41, no. 2, pp. 285–290, 1993.

Dong Guo received the B.S. degree in geophysics and computer science from China University of Mining and Technology (CUMT), Xuzhou, China, in 1993; the M.S. degree in geophysics from the Graduate School, Research Institute of Petroleum Exploration and Development (RIPED), Beijing, China, in 1996. In 1999, he received the Ph.D. degree in applied mathematics from Beijing University, Beijing, China. He is now pursuing a second Ph.D. degree in the Department of Electrical Engineering, Columbia University, New York. His research interests are in the area of statistical signal processing and communications.

Xiaodong Wang received the B.S. degree in electrical engineering and applied mathematics (with the highest honor) from Shanghai Jiao Tong University, Shanghai, China, in 1992; the M.S. degree in electrical and computer engineering from Purdue University in 1995; and the Ph.D. degree in electrical engineering from Princeton University in 1998. From July 1998 to December 2001, he was an Assistant Professor in the Department of Electrical Engineering, Texas A&M University. In January 2002, he joined the Department of Electrical Engineering, Columbia University, as an Assistant Professor. Dr. Wang's research interests fall in the general areas of computing, signal processing, and communications. He has worked in the areas of wireless communications, statistical signal processing, parallel and distributed computing, and bioinformatics. He has published extensively in these areas. Among his publications is a recent book entitled *Wireless Communication Systems: Advanced Techniques for Signal Reception*, published by Prentice Hall. His current research focuses include Bayesian Monte Carlo signal processing, multiuser communication theory, and wireless communications. Dr. Wang has received the 1999 NSF CAREER Award. He has also received the 2001 IEEE Communications Society and Information Theory Society Joint Paper Award. He currently serves as an Associate Editor for the IEEE Transactions on Signal Processing, the IEEE Transactions on Communications, the IEEE Transactions on Wireless Communications, and IEEE Transactions on Information Theory.
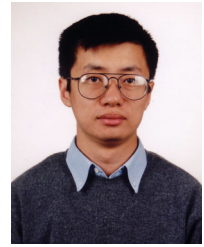
Rong Chen is a Professor at the Department of Information and Decision Sciences, College of Business Administration, University of Illinois at Chicago (UIC). Before joining UIC in 1999, he was with the Department of Statistics, Texas A&M University. Dr. Chen received his B.S. degree in mathematics from Peking University, China, in 1985; his M.S. and Ph.D. degrees in statistics from Carnegie Mellon University in 1987 and 1990, respectively. His main research interests are in time series analysis, statistical computing and Monte Carlo methods in dynamic systems, and statistical applications in engineering and business. He is an Associate Editor for the Journal of American Statistical Association, Journal of Business and Economic Statistics, Statistica Sinica, and Computational Statistics.

COMPOSITION COMMENTS

1   We added the e-mail addresses of the last two authors. Please provide the e-mail address of the first author affiliated to the university.

2   We completed the second postal address of the last author. Please check.

3   We changed "$1 \cdots 4$" to "$1, \ldots, 4$" throughout. Please check.

4   We changed "let" to "if". Otherwise, we will change "let" to "letting". Please check.

5   We changed the format of the highlighted part to avoid bad line break. Please check.

6   Please replace the word "correlated" with some other word to make the meaning clearer.

7   We removed "with" from (11) and (18). Please check.

8   We added the highlighted "$\times$" in (16). Please check.

9   We changed "level" to "levels". Otherwise, we will change "are" to "is". Please check.

10   We added the highlighted "$\times$" in (36) and (38). Similar cases are highlighted throughout. Please check.

11   Please specify where Remark 4 ends.

12   We changed "sample" to "samples". Please check.

13   Please specify where Remark 6 ends.

14   We added "(PSP)" after "per-survivor processing". Please check.

15   We added the highlighted comma. Please check.

16   We changed "sequences" to "sequence". Otherwise, we will change "is" to "are". Please check.

17   We removed "where" from (49). Please check.

18   We added "we used". Otherwise, please complete the sentence as it seems that something is missing.

19   We changed the initials of the second author's name in [1]. Please check.

20   We changed the last page number in [3]. Please check.

21   We changed the authors in [6] to be the editors. Please check.

22   We changed the publisher name and the initials of the second editor in [6]. Please check.

23   We changed "simulation-based" to "Monte Carlo sampling" in the title of [7]. Please check.

24   We changed the order of the third and fourth authors and the last page number in [8]. Please check.

25   We changed the last page number in [12]. Please check.

26   We changed "Krishnamurthy" to "Kroshnamurthy" in [15]. Please check.

27   We changed the range of pages in [21]. Please check.

28   We changed "on" to "of" in [24]. Please check.

29   We changed "envelop" in [25] to "envelope". Please check.

30   We changed "Priniciples" to "Principles" in [32]. Please check.

31   We changed the last page number in [33]. Please check.

32   Please note that the biographies should not exceed 200 words. Consequently, please reduce the biography of "Xiaodong Wang" to the required number of words.