

Delayed-Pilot Sampling for Mixture Kalman Filter With Application in Fading Channels

Xiaodong Wang, *Member, IEEE*, Rong Chen, *Member, IEEE*, and Dong Guo

Abstract—Sequential Monte Carlo (SMC) methods are powerful techniques for online filtering of nonlinear and non-Gaussian dynamic systems. Typically dynamic systems exhibit strong memory effects, i.e., future observations can reveal substantial information about the current state. Recently, the delayed-sample sampling method has been proposed in the context of mixture Kalman filter (MKF), which makes use of future observations in generating samples of the current state. Although this method is highly effective in producing accurate filtering results, its computational complexity is exponential in terms of the delay, due to the need to marginalize the future states. In this paper, we address this difficulty by developing two new sampling schemes for delayed estimation, namely, delayed-pilot sampling and hybrid-pilot sampling. The basic idea of delayed-pilot sampling is that instead of exploring the entire space of future states, we generate a number of random pilot streams, each of which indicates what would happen in the future if the current state takes a particular value. The sampling distribution of the current state is then determined by the incremental importance weight associated with each pilot stream. The delayed-pilot sampling can be used in conjunction with the delayed-sample method, resulting in a hybrid scheme. This new sampling technique is then applied to solve the problem of adaptive detection and decoding in flat-fading communication channels. Simulation results are provided to demonstrate the performance of the new low-complexity sampling techniques for delayed estimation and to compare with the delayed-sample method.

Index Terms—Adaptive decoding, delayed-pilot, delayed-sample, fading channel, mixture Kalman filter, sequential Monte Carlo.

I. INTRODUCTION

THE SEQUENTIAL Monte Carlo (SMC) methods recently emerged in the fields of statistics and engineering have shown a great promise in solving a wide range of nonlinear filtering/prediction problems associated with many highly complex dynamic systems [1], [3], [8], [12], [15], [18], [21]–[23], [28]. SMC can be loosely defined as a family of methodologies that use Monte Carlo simulation to solve on-line estimation problems in dynamic systems. By recursively generating Monte

Carlo samples of the state variables or some other latency variables, these methods can flexibly adapt to the dynamics of the underlying stochastic systems. A complete theoretical framework for the SMC has appeared in [23]. Many successful applications of SMC in diverse areas of science and engineering can be found in the recent books [9], [20].

Recently, we have applied the SMC technique, and in particular, the mixture Kalman filter (MKF) for conditional dynamic linear models [3], to the problem of adaptive detection and decoding in flat-fading channels [4] in the presence of Gaussian or non-Gaussian ambient noise. This problem is of fundamental importance in communication theory, and it is known that the optimal solution has a prohibitively high computational complexity [13], [25], [26]. Various suboptimal approaches have been proposed in the literature to bear on this problem [2], [5]–[7], [10], [11], [14], [16], [17], [19], [24], [27], [29], [30]. Our approach in [4] is based on a Bayesian formulation of the problem of on-line detection/decoding in flat-fading channels and the SMC methodology. The basic idea is to sequentially impute multiple samples of the transmitted symbols based on the current observation. Associated with each sequence of imputed symbols is its importance weight. The imputed symbol sequences, together with their importance weights, are used to compute the Bayesian estimates of the transmitted information. It is shown in [4] through simulations that the performance of the SMC-based receivers can be remarkably close to the so-called genie-aided bound in fading channels for both uncoded and coded systems without the use of any training/pilot symbols or decision feedback.

Dynamic systems often possess strong memory, i.e., future observations can reveal substantial information about the current state. For example, in a flat-fading communication channel, the channel fading coefficients are modeled as a correlated complex Gaussian process. Even if the transmitted symbols are independent, since the observations in the near future contain future channel states that are highly correlated with the current channel state, it is beneficial to make use of these future observations in detecting the current symbol. Moreover, when channel coding is employed, the transmitted symbols are also highly correlated, and it becomes crucial to exploit future observations to extract useful information contained in both future symbols and future channel states. However, an SMC method usually does not go back to regenerate past samples in view of new observations, although the past estimation can be adjusted by using the new importance weights. To overcome this difficulty, in [4], we have developed the *delayed-sample* method, which makes use of future observations in generating samples of the current state. It is seen there that this method is especially effective in improving

Manuscript received January 31, 2001; revised October 3, 2001. This work was supported in part by the National Science Foundation under Grants CCR-9980599, DMS-9982846, DMS-0073651, and DMS-0073601. The associate editor coordinating the review of this paper and approving it for publication was Dr. Petar M. Djurić.

X. Wang was with the Department of Electrical Engineering, Texas A&M University, College Station, TX 77843 USA. He is now with Department of Electrical Engineering, Columbia University, New York, NY 10027 USA.

R. Chen is with the Department of Information and Decision Science, University of Illinois at Chicago, Chicago, IL 60607 USA.

D. Guo is with the Department of Electrical Engineering, Texas A&M University, College Station, TX 77843 USA.

Publisher Item Identifier S 1053-587X(02)00563-9.

the receiver performance when the signals are channel-coded and/or when the channel ambient noise is non-Gaussian.

However, the computational complexity of the delayed-sample method developed in [4] is quite high, due to the need of marginalizing out the future states. For example, in a Gaussian noise flat-fading channel, for a Δ -step delayed-sample method, the algorithmic complexity is $O(m|\mathcal{A}|^\Delta)$ (where $\mathcal{A}$ is the finite set of signal constellations, and m is the number of Monte Carlo samples) for an uncoded system, and it is $O(m2^{k_0(\Delta+1)})$ for a coded system with a rate k_0/n_0 convolutional code. The complexity is even much higher if the ambient noise is non-Gaussian. Hence, it becomes important to develop computationally efficient delayed estimation techniques.

In this paper, new low-complexity techniques are developed for delayed estimation under the MKF framework. In order to draw a new sample of the current state, instead of marginalizing the future states of Δ steps, we generate $|\mathcal{A}|$ random pilot streams of these future states. Each pilot stream is obtained by starting with one of the possible value $a_j \in \mathcal{A}$ and propagating Δ steps of SMC. The incremental importance weight of this pilot stream is then given to the value a_j and with probability proportional to which a sample of the current state is drawn. Finally, to correct the bias introduced by the pilot approximation, the usual incremental weight is multiplied by the inverse of the pilot weight. We term this method of delayed estimation the *delayed-pilot* method. Moreover, such an idea can be combined with the delayed-sample method developed in [4] to yield a *delayed-hybrid* method. That is, in order to achieve a $(\Delta + \Delta')$ -step delayed estimation, we implement the combination of a Δ -step delayed-sample sampling and a Δ' -step delayed-pilot sampling. The delayed-pilot method and the delayed-hybrid method are then applied to the problem of adaptive detection and decoding in flat-fading channels treated in [4].

The rest of this paper is organized as follows. In Section II, we briefly summarize the MKF algorithm, as well as the delayed-weight method and delayed-sample method for MKF. In Section III, we develop the delayed-pilot method and the delayed-hybrid method for MKF. In Section IV, we treat the problems of adaptive detection and adaptive decoding in flat-fading channels using the proposed new sampling methods. Simulation results are provided in Section V. Finally, Section VI contains the conclusion.

II. BACKGROUND

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

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

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

Consider the following conditional dynamic linear model (CDLM) of the form

$$\mathbf{x}_t = \mathbf{F}_{\lambda_t} \mathbf{x}_{t-1} + \mathbf{G}_{\lambda_t} \mathbf{u}_t \quad (1)$$

$$\mathbf{y}_t = \mathbf{H}_{\lambda_t} \mathbf{x}_t + \mathbf{K}_{\lambda_t} \mathbf{v}_t \quad (2)$$

where $\mathbf{u}_t \sim \mathcal{N}_c(0, \mathbf{I})$ and $\mathbf{v}_t \sim \mathcal{N}_c(0, \mathbf{I})$ are the state and observation noise, respectively, and $\{\lambda_t\}$ is a sequence of random indicator variables that 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 . The indicator λ_t takes values from a finite set $\mathcal{A}$.

The MKF in [3] is a novel sequential Monte Carlo method for online filtering and prediction of CDLMs; it exploits the conditional Gaussian property and utilizes a marginalization operation to improve the algorithmic efficiency. Let $\mathbf{Y}_t = (\mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_t)$ and $\boldsymbol{\Lambda}_t = (\lambda_0, \lambda_1, \dots, \lambda_t)$. By recursively generating a set of properly weighted random samples $\{(\boldsymbol{\Lambda}_t^{(j)}, w_t^{(j)})\}_{j=1}^m$ to represent $p(\boldsymbol{\Lambda}_t | \mathbf{Y}_t)$, the MKF approximates the target distribution $p(\mathbf{x}_t | \mathbf{Y}_t)$ by a random mixture of Gaussian distributions

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

where

$$\boldsymbol{\mu}_t^{(j)} = \boldsymbol{\mu}_t(\boldsymbol{\Lambda}_t^{(j)}) \quad \text{and} \quad \boldsymbol{\Sigma}_t^{(j)} = \boldsymbol{\Sigma}_t(\boldsymbol{\Lambda}_t^{(j)})$$

are obtained with a Kalman filter on the system (1), (2) for the given indicator trajectory $\boldsymbol{\Lambda}_t^{(j)}$. Denote

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

Thus, a key step in the MKF is the production at time t of the weighted samples of the indicators and the Kalman filter outputs $\{(\boldsymbol{\Lambda}_t^{(j)}, \kappa_t^{(j)}, w_t^{(j)})\}_{j=1}^m$, 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 $(t-1)$ [3], [4].

The model (1) and (2) 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 $\boldsymbol{\Lambda}_t$ or $\mathbf{x}_t$ instantaneously with the posterior distribution $p(\boldsymbol{\Lambda}_t | \mathbf{Y}_t)$ or $p(\mathbf{x}_t | \mathbf{Y}_t)$, we delay this inference to a later time $(t + \Delta)$, $\Delta > 0$ with the distribution $p(\boldsymbol{\Lambda}_t | \mathbf{Y}_{t+\Delta})$ and $p(\mathbf{x}_t | \mathbf{Y}_{t+\Delta})$, respectively. In [4], two delayed estimation techniques are proposed, namely, the delayed-weight method and the delayed-sample method.

The idea behind the simple *delayed-weight* method is as follows. If the set $\{(\boldsymbol{\Lambda}_{t+\delta}^{(j)}, w_{t+\delta}^{(j)})\}_{j=1}^m$ is properly weighted with respect to $p(\boldsymbol{\Lambda}_{t+\delta} | \mathbf{Y}_{t+\delta})$, 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 $p(\lambda_t | \mathbf{Y}_{t+\delta})$. Then, any inference about the indicator λ_t , $E\{h(\lambda_t) | \mathbf{Y}_{t+\delta}\}$ can be approximated by

$$E\{h(\lambda_t) | \mathbf{Y}_{t+\delta}\} \cong \frac{1}{W_{t+\delta}} \sum_{j=1}^m h(\lambda_t^{(j)}) w_{t+\delta}^{(j)} \quad (3)$$

with

$$W_{t+\delta} = \sum_{j=1}^m w_{t+\delta}^{(j)}. \quad (4)$$

Since the weights $\{w_{t+\delta}^{(j)}\}_{j=1}^m$ contain information about the future observations $(\mathbf{y}_{t+1}, \dots, \mathbf{y}_{t+\delta})$, the estimate in (3) is usually more accurate than the concurrent estimate. This approach only requires slight additional memory and no additional computation. The inference about the state variable $\mathbf{x}_t$ can be similarly made at time $t + \delta$.

In the *delayed-sample* method, both the samples and the weights $\{(\lambda_t^{(j)}, w_t^{(j)})\}_{j=1}^m$ are generated based on the observations $\mathbf{Y}_{t+\Delta}$, hence making $p(\Lambda_t | \mathbf{Y}_{t+\Delta})$ the target distribution at time $(t + \Delta)$. The procedure provides better Monte Carlo samples since it utilizes the future observations $(\mathbf{y}_{t+1}, \dots, \mathbf{y}_{t+\Delta})$ 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 $\lambda_{t+1}, \dots, \lambda_{t+\Delta}$. At any time t , the delayed-sample algorithm involves growing and trimming a tree of depth Δ . That is, we grow the tree by adding $|\mathcal{A}|^{\Delta+1}$ leaves at the top of tree, then keep one the $|\mathcal{A}|$ main branches from the root corresponding to the selected $\lambda_t^{(j)}$, and then removing the rest of the branches. Fig. 1 illustrates the delayed-sample sampling operation, with $\mathcal{A} = \{1, 0\}$ and $\Delta = 2$.

III. DELAYED-PILOT SAMPLING IN MIXTURE KALMAN FILTER

A. Simple Delayed-Pilot Sampling

The delayed-sample MKF algorithm can achieve significant improvement in performance compared with the concurrent MKF algorithm, as demonstrated in [4]. However, such improvement comes at the expense of significant increase in computational complexity since it requires full exploration of the space of future states $\lambda_{t+1}, \dots, \lambda_{t+\Delta}$. On the other hand, the delay-weight method utilizes the future observation in $\mathbf{y}_{t+1}, \dots, \mathbf{y}_{t+\delta}$ only through the weight adjustments. The sample of λ_t generated is still based only on the observation $\mathbf{y}_t$ up to time t .

Because of the desire of utilizing future data and because of the high computational complexity associated with a full exploration of the space of the future states, in this paper, we develop a low-complexity algorithm that partially explores (but hopefully the most important part of) the space of future states $\lambda_{t+1}, \dots, \lambda_{t+\Delta}$ and utilizes this information to generate better samples of λ_t . Specifically, since λ_t takes $|\mathcal{A}|$ possible discrete values in $\mathcal{A}$, we generate $|\mathcal{A}|$ “pilots,” each starting with one of the possible values λ_t and propagating to $(t + \Delta)$ using the concurrent MKF steps. Then, the incremental weight (from t to $t + \Delta$) of the pilot starting with $\lambda_t = a_i$ reveals what would happen in the future $(t + 1$ to $t + \Delta)$ if λ_t is chosen to be a_i , given $\mathbf{y}_{t+1}, \dots, \mathbf{y}_{t+\Delta}$. Hence, these incremental weights can be used as the sampling distribution in selecting λ_t . This sampling distribution helps us to generate better samples for the current state, even though we did not fully explore the entire space of future states. We call this algorithm the *delayed-pilot sampling* method. Fig. 2 illustrates the delayed-pilot sampling

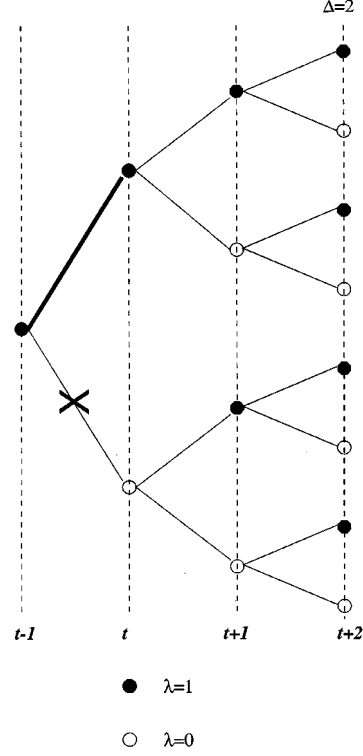


Fig. 1. Illustration of the delayed-sample sampling.

operation, with $\mathcal{A} = \{1, 0\}$ and $\Delta = 2$. The algorithm is summarized as follows.

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

For $j = 1, \dots, m$:

- For each $a_i \in \mathcal{A}$, run a one-step Kalman filter update assuming $\lambda_t = a_i$ to obtain

$$\begin{aligned} \kappa_{t-1}^{(j)} \xrightarrow{\mathbf{y}_t, \lambda_t = a_i} \kappa_{t,i}^{(j)} \\ \triangleq \left[\boldsymbol{\mu}_t \left(\Lambda_{t-1}^{(j)}, \lambda_t = a_i \right), \boldsymbol{\Sigma}_t \left(\Lambda_{t-1}^{(j)}, \lambda_t = a_i \right) \right] \end{aligned} \quad (5)$$

and compute

$$\begin{aligned} \gamma_{t,i}^{(j)} &\triangleq P \left(\lambda_t = a_i | \Lambda_{t-1}^{(j)}, \mathbf{Y}_t \right) \\ &\propto p \left(\mathbf{y}_t | \Lambda_{t-1}^{(j)}, \lambda_t = a_i, \mathbf{Y}_{t-1} \right) P \left(\lambda_t = a_i | \Lambda_{t-1}^{(j)} \right). \end{aligned} \quad (6)$$

- For each $a_i \in \mathcal{A}$, do the following:

For $s = t + 1, \dots, t + \Delta$, repeat

— Let

$$\Lambda_{s-1}^{(i,j)} = \left[\Lambda_{t-1}^{(j)}, \lambda_t = a_i, \lambda_{t+1} = \lambda_{t+1}^{(i,j)}, \dots, \lambda_{s-1} = \lambda_{s-1}^{(i,j)} \right].$$

For each $a_k \in \mathcal{A}$, run a one-step Kalman filter update assuming $\lambda_s = a_k$ to obtain

$$\begin{aligned} \kappa_{s-1,i}^{(j)} \xrightarrow{\mathbf{y}_s, \lambda_s = a_k} \kappa_{s,i}^{(k,j)} \\ \triangleq \left[\boldsymbol{\mu}_s \left(\Lambda_{s-1}^{(i,j)}, \lambda_s = a_k \right), \boldsymbol{\Sigma}_s \left(\Lambda_{s-1}^{(i,j)}, \lambda_s = a_k \right) \right]. \end{aligned} \quad (7)$$

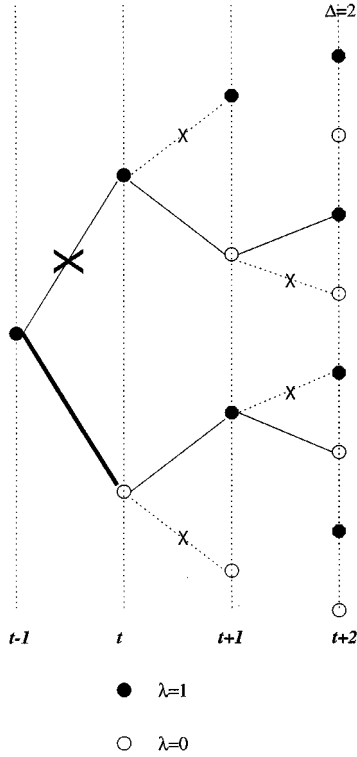


Fig. 2. Illustration of the delayed-pilot sampling.

- For each $a_k \in \mathcal{A}$, compute the sampling density

$$\begin{aligned} \gamma_{s,i}^{(k,j)} &\triangleq P(\lambda_s = a_k | \Lambda_{s-1}^{(i,j)}, \mathbf{Y}_s) \\ &\propto p(\mathbf{y}_s | \Lambda_{s-1}^{(i,j)}, \lambda_s = a_k, \mathbf{Y}_{s-1}) \\ &\quad \times P(\lambda_s = a_k | \Lambda_{s-1}^{(i,j)}). \end{aligned} \quad (8)$$

Draw a sample $\lambda_s^{(i,j)}$ according to the above sampling density. If $\lambda_s^{(i,j)} = a_\ell$ is drawn, then set $\kappa_{s,i}^{(j)} = \kappa_{s,i}^{(\ell,j)}$.

- Compute the incremental importance weight

$$\begin{aligned} \gamma_{s,i}^{(j)} &= \frac{p(\Lambda_s^{(i,j)} | \mathbf{Y}_s)}{p(\Lambda_{s-1}^{(i,j)} | \mathbf{Y}_{s-1}) p(\lambda_s^{(j)} | \Lambda_{s-1}^{(i,j)}, \mathbf{Y}_s)} \\ &\propto p(\mathbf{y}_s | \Lambda_{s-1}^{(i,j)}, \mathbf{Y}_{s-1}) \\ &= \sum_{k=1}^{|\mathcal{A}|} p(\mathbf{y}_s | \lambda_s = a_k, \Lambda_{s-1}^{(i,j)}, \mathbf{Y}_{s-1}) \\ &\quad \times P(\lambda_s = a_k | \Lambda_{s-1}^{(i,j)}) \\ &\propto \sum_{k=1}^{|\mathcal{A}|} \gamma_{s,i}^{(k,j)}. \end{aligned} \quad (9)$$

- For each $a_i \in \mathcal{A}$, compute

$$\rho_{t,i}^{(j)} = \gamma_{t,i}^{(j)} \prod_{s=t+1}^{t+\Delta} \gamma_{s,i}^{(j)}. \quad (10)$$

Using

$$\tilde{\rho}_{t,i}^{(j)} \triangleq \frac{\rho_{t,i}^{(j)}}{\sum_{k=1}^{|\mathcal{A}|} \rho_{t,i}^{(k)}}, \quad i = 1, \dots, |\mathcal{A}| \quad (11)$$

as the trial sampling distribution, draw a sample $\lambda_t^{(j)}$. Append $\lambda_t^{(j)}$ to $\Lambda_{t-1}^{(j)}$ to obtain $\Lambda_t^{(j)}$. If $\lambda_t^{(j)} = a_i$ is drawn, then set $\kappa_t^{(j)} = \kappa_{t,i}^{(j)}$.

- Compute the importance weight. If $\lambda_t^{(j)} = a_i$, then

$$\begin{aligned} w_t^{(j)} &= w_{t-1}^{(j)} \frac{p(\Lambda_t^{(j)} | \mathbf{Y}_t)}{p(\Lambda_{t-1}^{(j)} | \mathbf{Y}_{t-1}) \tilde{\rho}_{t,i}^{(j)}} \\ &\propto w_{t-1}^{(j)} \frac{p(\mathbf{y}_t | \mathbf{Y}_{t-1}, \Lambda_t^{(j)}) P(\lambda_t^{(j)} = a_i | \Lambda_{t-1}^{(j)})}{\tilde{\rho}_{t,i}^{(j)}} \\ &= w_{t-1}^{(j)} \frac{\gamma_{t,i}^{(j)}}{\tilde{\rho}_{t,i}^{(j)}}. \end{aligned} \quad (12)$$

- Do resampling if the effective sample size is below certain threshold, as discussed later in Remark 4.

Remark 1 (Multiple Pilots): In the above algorithm, we can also use multiple pilots for each group. That is, we can generate $|\mathcal{A}|$ groups of pilots, each with K members. Each group starts with one of the possible values in $\mathcal{A}$. The sum of the incremental weights of each group (propagated from t to $t + \Delta$) is then used as the sampling probability for λ_t . This is particularly useful when $|\mathcal{A}|$ is large.

Remark 2 (Inference): The above algorithm generates, at time $(t + \Delta)$, a weighted sample $(\Lambda_t^{(j)}, w_t^{(j)})$ that is properly weighted with respect to $p(\Lambda_t | \mathbf{Y}_t)$. This can be easily seen from

$$\begin{aligned} w_t^{(j)} &= w_{t-1}^{(j)} \frac{p(\Lambda_t^{(j)} | \mathbf{Y}_t)}{p(\Lambda_{t-1}^{(j)} | \mathbf{Y}_{t-1}) \tilde{\rho}_{t,i}^{(j)}} \\ &= \prod_{s=1}^t \frac{p(\Lambda_s^{(j)} | \mathbf{Y}_s)}{p(\Lambda_{s-1}^{(j)} | \mathbf{Y}_{s-1}) \tilde{\rho}_{s,i}^{(j)}} \\ &= \frac{p(\Lambda_t^{(j)} | \mathbf{Y}_t)}{p(\Lambda_0^{(j)} | \mathbf{Y}_0) \prod_{s=1}^t \tilde{\rho}_{s,i}^{(j)}} \end{aligned} \quad (13)$$

where the numerator of (14) is the target distribution, and the denominator is the sampling distribution of $\Lambda_t^{(j)}$ according to the algorithm.

Unfortunately, $(\Lambda_t^{(j)}, w_t^{(j)})$ is *not* properly weighted with respect to $p(\Lambda_t | \mathbf{Y}_{t+\Delta})$. For the latter, one would have to explore the entire future space, as in the delayed-sample algorithm. Hence, the weighted average

$$\frac{1}{W_t} \sum_{j=1}^m h(\lambda_t^{(j)}) w_t^{(j)}$$

is an estimate of $E\{h(\lambda_t)|\mathbf{Y}_t\}$ but *not* of $E\{h(\lambda_t)|\mathbf{Y}_{t+\Delta}\}$. Hence, it seems that our purpose of using future data to help making inference is not served at all. Although the sample $\lambda_t^{(j)}$ is closer to the distribution $p(\lambda_t|\mathbf{Y}_{t+\Delta})$, the weight adjusts them backward to $p(\lambda_t|\mathbf{Y}_t)$, hence giving smaller weights to those close to $p(\lambda_t|\mathbf{Y}_{t+\Delta})$ but away from $p(\lambda_t|\mathbf{Y}_t)$ and giving larger weight to those close to $p(\lambda_t|\mathbf{Y}_t)$ but away from $p(\lambda_t|\mathbf{Y}_{t+\Delta})$.

Fortunately, there is indeed an easy way that allows us to make better inferences on λ_t at time $(t+\Delta)$. Suppose $\lambda_t^{(j)} = a_i$. Then, we calculate the following *auxiliary weight*:

$$\tilde{w}_t^{(j)} = w_t^{(j)} \prod_{s=t+1}^{t+\Delta} \gamma_{s,i}^{(j)}. \quad (15)$$

This is the original weight $w_t^{(j)}$ multiplied by the incremental weight of the sampled pilot propagated from $(t+1)$ to $(t+\Delta)$. Let

$$\Lambda_{t+\Delta}^{(i,j)} \triangleq [\Lambda_{t-1}^{(j)}, \lambda_t = a_i, \lambda_{t+1} = \lambda_{t+1}^{(i,j)}, \dots, \lambda_{t+\Delta} = \lambda_{t+\Delta}^{(i,j)}]$$

be the selected pilot. Then, $(\Lambda_{t+\Delta}^{(i,j)}, \tilde{w}_t^{(j)})$ is properly weighted with respect to $p(\Lambda_{t+\Delta}|\mathbf{y}_{t+\Delta})$. This can be easily seen from the following:

$$\begin{aligned} \tilde{w}_t^{(j)} &= w_t^{(j)} \prod_{s=t+1}^{t+\Delta} \gamma_{s,i}^{(j)} \\ &= \prod_{s=1}^t \frac{p(\Lambda_s^{(j)}|\mathbf{Y}_s)}{p(\Lambda_{s-1}^{(j)}|\mathbf{Y}_{s-1}) \tilde{\rho}_{s,i}^{(j)}} \\ &\quad \times \prod_{s=t+1}^{t+\Delta} \frac{p(\Lambda_s^{(i,j)}|\mathbf{Y}_s)}{p(\Lambda_{s-1}^{(i,j)}|\mathbf{Y}_{s-1}) p(\lambda_s^{(i,j)}|\mathbf{Y}_s, \Lambda_{s-1}^{(i,j)})} \\ &= \frac{p(\Lambda_{t+\Delta}^{(j)}|\mathbf{Y}_{t+\Delta})}{p(\Lambda_0^{(j)}|\mathbf{Y}_0) \prod_{s=1}^t \tilde{\rho}_{s,i}^{(j)} \prod_{s=t+1}^{t+\Delta} p(\lambda_s^{(i,j)}|\mathbf{Y}_s, \Lambda_{s-1}^{(i,j)})} \end{aligned} \quad (16)$$

where the numerator of (16) is the target distribution, and the denominator is the sampling distribution of $\Lambda_{t+\Delta}^{(i,j)}$, the j th stream up to time t , and the selected i th pilot path.

Hence, $(\lambda_t^{(j)}, \tilde{w}_t^{(j)})$ is properly weighted with respect to $p(\lambda_t|\mathbf{y}_{t+\Delta})$. Therefore, we have

$$E\{h(\lambda_t)|\mathbf{Y}_{t+\Delta}\} \cong \frac{1}{\tilde{W}_t} \sum_{j=1}^m h(\lambda_t^{(j)}) \tilde{w}_t^{(j)}$$

with

$$\tilde{W}_t \triangleq \sum_{j=1}^m \tilde{w}_t^{(j)}. \quad (17)$$

Note that no additional computation is needed to obtain the auxiliary weights; hence, it is recommended to keep two sets of weights in the propagation. However, the auxiliary weight are computed at every time step and *cannot* be used for further propagation (i.e., to replace w_t). It is solely used for inference of λ_t at time $(t+\Delta)$.

Remark 3 (Combination With Delayed-Weight): It is desirable to combine the delayed-pilot method with the delayed-weight method. Specifically, suppose at time $(t+\Delta)$ we obtain a sample $(\Lambda_t^{(j)}, w_t^{(j)})$ using the delayed-pilot method. Then, with an additional delay δ , we can use the auxiliary weight $\tilde{w}_{t+\delta}$ at time $(t+\Delta+\delta)$ to make inference on λ_t , i.e.,

$$E\{h(\lambda_t)|\mathbf{Y}_{t+\Delta+\delta}\} \cong \frac{1}{\tilde{W}_{t+\delta}} \sum_{j=1}^m h(\lambda_t^{(j)}) \tilde{w}_{t+\delta}^{(j)}$$

with

$$\tilde{W}_{t+\delta} = \sum_{j=1}^m \tilde{w}_{t+\delta}^{(j)}. \quad (18)$$

This is because $\{(\Lambda_{t+\delta}^{(j)}, \tilde{w}_{t+\delta}^{(j)})\}_{j=1}^m$ is properly weighted with respect to $p(\Lambda_{t+\delta}|\mathbf{Y}_{t+\Delta+\delta})$.

Alternatively, if one does not maintain two sets of weights and is willing to wait additional δ ($\delta \geq \Delta$) steps, then the weight $w_{t+\delta}^{(j)}$ generated at time $(t+\delta)$ from the above algorithm can be used for inference. At that time, $(\Lambda_{t+\delta}^{(j)}, w_{t+\delta}^{(j)})$ is a properly weighted sample with respect to $p(\Lambda_{t+\delta}|\mathbf{Y}_{t+\delta})$. Then, $(\lambda_t^{(j)}, w_{t+\delta}^{(j)})$ is properly weighted with respect to $p(\lambda_t|\mathbf{Y}_{t+\delta})$. Since $\delta \geq \Delta$, we still take full advantage of the delay Δ .

Remark 4 (Resampling): Resampling is an important step in SMC. It rejuvenates the sampler when the effective sample size becomes small, hence providing more efficient samples for the future states. For more details, see [23]. Specifically, suppose at time t we have a set of random samples $\{(\Lambda_t^{(j)}, w_t^{(j)})\}_{j=1}^m$ properly weighted with respect to $p(\Lambda_t|\mathbf{Y}_t)$. By treating the samples as a discrete approximation of the posterior distribution, we can generate another discrete representation as follows.

- Select a set of positive numbers $a^{(j)}$, $j = 1, \dots, m$.
- For $j_* = 1, \dots, m$
 - Randomly select $\tilde{\Lambda}_t^{(j_*)}$ from the set $\{\Lambda_t^{(j)}\}_{j=1}^m$ with probability proportional to $a^{(j)}$.
 - If $\tilde{\Lambda}_t^{(j_*)} = \Lambda_t^{(j_*)}$, then set the new weight associated with this sample to be $\tilde{w}_t^{(j_*)} = w_t^{(j_*)}/a^{(j_*)}$.
- Return the new representation $\{(\tilde{\Lambda}_t^{(j_*)}, \tilde{w}_t^{(j_*)})\}_{j_*=1}^m$.

The new weight is also properly weighted with respect to $p(\Lambda_t|\mathbf{Y}_t)$.

The choice of the resampling probability $a^{(j)}$ directly affects the efficiency of the algorithm. As in [23], one choice is $a^{(j)} = w_t^{(j)}$, which makes the resulting resampled set have equal weights. In the delay-pilot algorithm, the auxiliary weight $a^{(j)} = \tilde{w}_t^{(j)}$ is a better choice. As shown above, w_t gives small weights to samples that are close to the future distribution $p(\Lambda_t|\mathbf{Y}_{t+\Delta})$ but away from the current distribution $p(\Lambda_t|\mathbf{Y}_t)$. Using w_t as the resampling probability is shortsighted. It tends to remove the samples that are less important at time t but may be important in the future (e.g., at time $t+\Delta$). On the other hand, the auxiliary weight $\tilde{w}_t$ is based on the future distribution. This resampling procedure does not reduce the variance of the current weight distribution as much as resampling according to w_t . However, since it keeps better samples (in the eyes of future), the weight distribution at latter time will be better. Note that with $a^{(j)} = \tilde{w}_t^{(j)}$, the new weight is proportional to

$1/\prod_{s=t+1}^{t+\Delta} \gamma_{s,i}^{(j)}$, where the inverse of the incremental weight of the sampled pilot is propagated from $t+1$ to $t+\Delta$.

When the weight distribution becomes very skewed, resample according to the original weights may result in severe loss of diversity of the sample. In this situation, it is recommended to use $a^{(j)} = [w_t^{(j)}]^\alpha$, or $a^{(j)} = [\tilde{w}_t^{(j)}]^\alpha$, where $0 < \alpha < 1$ can vary according to the coefficient of variation of w_t . For the delayed pilot algorithm, $\alpha = 0.5$ seems to be a good choice.

Remark 5 (Complexity): Note that the computation required for the above simple delayed-pilot sampling is the same as that for the one-step delayed-sample method. Hence, the delayed-pilot MKF algorithm is recommended only for $\Delta \geq 2$. At each time, this algorithm requires $m[|\mathcal{A}| + (|\mathcal{A}| - 1)|\mathcal{A}|\Delta]$ one-step Kalman filter updates to be calculated, whereas the delayed-sample MKF requires $m|\mathcal{A}|^{\Delta+1}$ Kalman updates at each time.

B. Delayed-Hybrid Sampling

The delayed-hybrid sampling method is a combination of the delayed-sample method and the delayed-pilot method. Here, we fully explore the state space of the immediate future and generate pilots to explore the space beyond that.

Let $\Delta + \Delta'$ be the total delay time. A delayed-hybrid sampling method generates $|\mathcal{A}|^{\Delta+1}$ pilots, where each takes one of the $|\mathcal{A}|^{\Delta+1}$ possible paths of $\lambda_t, \dots, \lambda_{t+\Delta}$ and makes additional Δ' concurrent MKF steps to time $t + \Delta + \Delta'$. Then, the probability of selecting $\lambda_t = a_i$ is proportional to the sum of the incremental importance weights of all pilots, starting with $\lambda_t = a_i$. Since we have explored the entire space of $\lambda_t, \dots, \lambda_{t+\Delta}$, the weight calculation will be based on the target distribution $p(\Lambda_t | \mathbf{Y}_{t+\Delta})$. Fig. 3 illustrates the delayed-hybrid sampling operation, with $\mathcal{A} = \{1, 0\}$, $\Delta = 1$, and $\Delta' = 2$.

Specifically, for each possible “future” (relative to time $t-1$) symbol sequence at time $(t + \Delta - 1)$, i.e.,

$$(\lambda_t, \lambda_{t+1}, \dots, \lambda_{t+\Delta-1}) \in \mathcal{A}^\Delta$$

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

$$\kappa_{t+\tau}^{(j)}(\lambda_t^{t+\tau}) \triangleq \left[\mu_{t+\tau}^{(j)}(\Lambda_{t-1}^{(j)}, \lambda_t^{t+\tau}), \Sigma_{t+\tau}^{(j)}(\Lambda_{t-1}^{(j)}, \lambda_t^{t+\tau}) \right]_{\tau=0, \dots, \Delta-1}. \quad (19)$$

Denote

$$\kappa_{t-1}^{(j)} \triangleq \left\{ \kappa_{t-1}^{(j)}, \left\{ \kappa_{t+\tau}^{(j)}(\lambda_t^{t+\tau}) \right\}_{\tau=0}^{\Delta-1} : \lambda_t^{t+\tau} \in \mathcal{A}^{\tau+1} \right\}. \quad (20)$$

The delayed-hybrid sampling MKF algorithm recursively propagates the samples properly weighted for $p(\Lambda_{t-1} | \mathbf{Y}_{t+\Delta-1})$ to those for $p(\Lambda_t | \mathbf{Y}_{t+\Delta})$ and is summarized as follows.

Algorithm 2 [Delayed-Hybrid MKF]: Suppose at time $(t + \Delta + \Delta' - 1)$, a set of properly weighted samples $\{(\Lambda_{t-1}^{(j)}, \kappa_{t-1}^{(j)}, w_{t-1}^{(j)})\}_{j=1}^m$ with respect to $p(\Lambda_{t-1} | \mathbf{Y}_{t+\Delta-1})$ are available. Then, at time $(t + \Delta + \Delta')$, as the new data $\mathbf{y}_{t+\Delta+\Delta'}$ become available, the following steps are implemented to update each weighted sample.

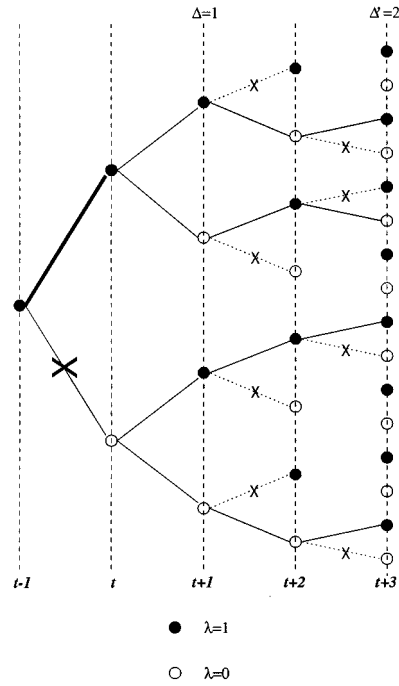


Fig. 3. Illustration of the delayed-hybrid sampling.

For $j = 1, 2, \dots, m$:

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

$$\kappa_{t+\Delta-1}^{(j)}(\lambda_t^{t+\Delta-1}) \xrightarrow{\mathbf{y}_{t+\Delta}, \lambda_{t+\Delta}=a_i} \kappa_{t+\Delta}^{(j)}(\lambda_t^{t+\Delta-1}, \lambda_{t+\Delta}=a_i). \quad (21)$$

Compute

$$\begin{aligned} \gamma_t^{(j)}(\lambda_t^{t+\Delta}) &\triangleq p(\lambda_t^{t+\Delta} | \mathbf{Y}_{t+\Delta}, \Lambda_{t-1}^{(j)}) \\ &\propto p(\lambda_t^{t+\Delta}, \mathbf{y}_t, \dots, \mathbf{y}_{t+\Delta} | \mathbf{Y}_{t-1}, \Lambda_{t-1}^{(j)}) \\ &= \prod_{\tau=0}^{\Delta} p(\mathbf{y}_{t+\tau} | \mathbf{Y}_{t+\tau-1}, \Lambda_{t-1}^{(j)}, \lambda_t^{t+\tau}) \\ &\quad \times \prod_{\tau=0}^{\Delta} p(\lambda_{t+\tau} | \Lambda_{t-1}^{(j)}, \lambda_t^{t+\tau-1}). \end{aligned} \quad (22)$$

- For every path of $\lambda_t^{t+\Delta} \in \mathcal{A}^{\Delta+1}$, do the following:
For $s = t + \Delta + 1, \dots, t + \Delta + \Delta'$, repeat:

— Denote

$$\begin{aligned} \Lambda_{s-1}^{(j)}(\lambda_t^{t+\Delta}) &\triangleq [\Lambda_{t-1}^{(j)}, \lambda_t^{t+\Delta}, \lambda_{t+\Delta+1} = \lambda_{t+\Delta+1}^{(j)}(\lambda_t^{t+\Delta}), \dots \\ &\quad \lambda_{s-1} = \lambda_{s-1}^{(j)}(\lambda_t^{t+\Delta})]. \end{aligned} \quad (23)$$

For each $a_k \in \mathcal{A}$, run a one-step Kalman filter update assuming $\lambda_s = a_k$ to obtain

$$\begin{aligned} \kappa_{s-1}^{(j)}(\lambda_t^{t+\Delta}) \xrightarrow{\mathbf{y}_s, \lambda_s=a_k} \kappa_s^{(j)}(\lambda_t^{t+\Delta}) &\triangleq [\mu_s^{(j)}(\Lambda_{s-1}^{(j)}(\lambda_t^{t+\Delta}), \lambda_s = a_k) \\ &\quad \Sigma_s^{(j)}(\Lambda_{s-1}^{(j)}(\lambda_t^{t+\Delta}), \lambda_s = a_k)]. \end{aligned} \quad (24)$$

- For each $a_k \in \mathcal{A}$, compute the sampling density

$$\begin{aligned} \gamma_s^{(k,j)}(\lambda_t^{t+\Delta}) &\triangleq P\left[\lambda_s = a_k \mid \Lambda_{s-1}^{(j)}(\lambda_t^{t+\Delta}), \mathbf{Y}_s\right] \\ &\propto p\left[\mathbf{y}_s \mid \lambda_s = a_k, \Lambda_{s-1}^{(j)}(\lambda_t^{t+\Delta}), \mathbf{Y}_{t-1}\right] \\ &\quad \times P\left[\lambda_s = a_k \mid \Lambda_{s-1}^{(j)}(\lambda_t^{t+\Delta})\right]. \end{aligned} \quad (25)$$

Draw a sample $\lambda_s^{(j)}(\lambda_t^{t+\Delta})$ according to the above sampling density. If $\lambda_s^{(j)}(\lambda_t^{t+\Delta}) = a_\ell$, set

$$\kappa_s^{(j)}(\lambda_t^{t+\Delta}) = \kappa_s^{(\ell,j)}(\lambda_t^{t+\Delta}).$$

- Compute the incremental importance weight

$$\begin{aligned} \gamma_s^{(j)}(\lambda_t^{t+\Delta}) &= p\left[\mathbf{y}_s \mid \Lambda_{s-1}^{(j)}(\lambda_t^{t+\Delta}), \mathbf{Y}_{s-1}\right] \\ &\propto \sum_{k=1}^{|\mathcal{A}|} \gamma_s^{(k,j)}(\lambda_t^{t+\Delta}). \end{aligned} \quad (26)$$

- For each $a_i \in \mathcal{A}$, compute

$$\begin{aligned} \rho_{t,i}^{(j)} &= \sum_{\lambda_{t+1}^{t+\Delta} \in \mathcal{A}^\Delta} \left[\gamma_t^{(j)}(\lambda_t = a_i, \lambda_{t+1}^{t+\Delta}) \right. \\ &\quad \times \left. \prod_{s=t+\Delta+1}^{t+\Delta+\Delta'} \gamma_s^{(j)}(\lambda_t = a_i, \lambda_{t+1}^{t+\Delta}) \right]. \end{aligned} \quad (27)$$

Using

$$\tilde{\rho}_{t,i}^{(j)} \triangleq \frac{\rho_{t,i}^{(j)}}{|\mathcal{A}| \sum_{k=1}^{|\mathcal{A}|} \rho_{t,k}^{(j)}}, \quad i = 1, \dots, |\mathcal{A}| \quad (28)$$

as the trial sampling distribution, draw a sample $\lambda_t^{(j)}$. Append $\lambda_t^{(j)}$ to $\Lambda_{t-1}^{(j)}$ to obtain $\Lambda_t^{(j)}$. Based on this sample, form $\kappa_t^{(j)}$ based on the results from the previous step.

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

$$\begin{aligned} w_t^{(j)} &= w_{t-1}^{(j)} \frac{p(\Lambda_t^{(j)} | \mathbf{Y}_{t+\Delta})}{p(\Lambda_{t-1}^{(j)} | \mathbf{Y}_{t+\Delta-1}) \tilde{\rho}_{t,i}^{(j)}} \\ &\propto \frac{w_{t-1}^{(j)}}{\tilde{\rho}_{t,i}^{(j)}} \\ &\quad \times \frac{\sum_{\lambda_{t+1}^{t+\Delta} \in \mathcal{A}^\Delta} \left[\prod_{\tau=0}^{\Delta} p(\mathbf{y}_{t+\tau} | \mathbf{Y}_{t+\tau-1}, \Lambda_t^{(j)}, \lambda_{t+1}^{t+\tau}) \right]}{\sum_{\lambda_{t+\Delta-1}^{t+\Delta} \in \mathcal{A}^\Delta} \left[\prod_{\tau=0}^{\Delta-1} p(\mathbf{y}_{t+\tau} | \mathbf{Y}_{t+\tau-1}, \Lambda_{t-1}^{(j)}, \lambda_{t+\tau}^{t+\tau}) \right]} \\ &\quad \times \frac{\prod_{\tau=0}^{\Delta} p(\lambda_{t+\tau} | \Lambda_t^{(j)}, \lambda_{t+1}^{t+\tau-1})}{\prod_{\tau=0}^{\Delta-1} p(\lambda_{t+\tau} | \Lambda_{t-1}^{(j)}, \lambda_{t+\tau-1}^{t+\tau-1})} \end{aligned}$$

$$\begin{aligned} &\sum_{\lambda_{t+1}^{t+\Delta} \in \mathcal{A}^\Delta} \gamma_t^{(j)}(\lambda_t^{(j)} = a_i, \lambda_{t+1}^{t+\Delta}) \\ &= \frac{w_{t-1}^{(j)}}{\tilde{\rho}_{t,i}^{(j)}} \frac{\sum_{\lambda_{t-1}^{t+\Delta-1} \in \mathcal{A}^\Delta} \gamma_{t-1}^{(j)}(\lambda_{t-1}^{(j)} = a_k, \lambda_t^{t+\Delta-1})}{\sum_{\lambda_{t-1}^{t+\Delta-1} \in \mathcal{A}^\Delta} \gamma_{t-1}^{(j)}(\lambda_{t-1}^{(j)} = a_k, \lambda_t^{t+\Delta-1})} \\ &\quad \times p(\mathbf{y}_{t-1} | \mathbf{Y}_{t-2}, \Lambda_{t-1}^{(j)}) P(\lambda_{t-1}^{(j)} | \Lambda_{t-2}^{(j)}). \end{aligned} \quad (29)$$

- Do resampling if the effective sample size is below certain threshold, as discussed in Remark 4 (Section III-B).

Note that at each time the above delayed-hybrid MKF algorithm involves calculating $m|\mathcal{A}|^\Delta[|\mathcal{A}| + (|\mathcal{A}| - 1)|\mathcal{A}|^\Delta]$ one-step Kalman filter updates, whereas the full delayed-sample MKF algorithm requires $m|\mathcal{A}|^{\Delta+\Delta'+1}$ Kalman updates at each time.

Similar to the proof of (14), it can be easily shown that the above algorithm generates, at time $(t + \Delta + \Delta')$, a weighted sample $(\Lambda_t^{(j)}, w_t^{(j)})$, which is properly weighted with respect to $p(\Lambda_t | \mathbf{Y}_{t+\Delta})$ but not to $p(\Lambda_t | \mathbf{Y}_{t+\Delta+\Delta'})$.

Remark 6 (Auxiliary Weight): Similar to the delay-pilot algorithm, we also need to resort to the *auxiliary weight* to make better inferences on λ_t at time $(t + \Delta + \Delta')$. However, it is slightly more complicated for the delay-hybrid algorithm.

Suppose $\lambda_t^{(j)} = a_i$. Define the concurrent weight

$$\begin{aligned} \bar{w}_t^{(j)} &= \bar{w}_{t-1}^{(j)} \frac{p(\Lambda_t^{(j)} | \mathbf{Y}_t)}{p(\Lambda_{t-1}^{(j)} | \mathbf{Y}_{t-1}) \tilde{\rho}_{t,i}^{(j)}} \\ &= \bar{w}_{t-1}^{(j)} \frac{p(\mathbf{y}_t | \mathbf{Y}_{t-1}, \lambda_t = a_i, \Lambda_{t-1}^{(j)}) p(\lambda_t = a_i | \Lambda_{t-1}^{(j)})}{\tilde{\rho}_{t,i}^{(j)}} \end{aligned} \quad (30)$$

where $\tilde{\rho}_{t,i}^{(j)}$ is the sampling probability (28). Then, $(\Lambda_t^{(j)}, \bar{w}_t^{(j)})$ is properly weighted with respect to $p(\Lambda_t | \mathbf{Y}_t)$, which is the concurrent posterior distribution.

Let the ℓ th path in $\mathcal{A}^\Delta$ be $\lambda_{t+1,\ell}^{t+\Delta}$, and $\lambda_{s,\ell}^{(j)}$, where $s = t + \Delta + 1, \dots, t + \Delta + \Delta'$ is the pilot path started from $\lambda_{t+1,\ell}^{t+\Delta}$ in the j th sample stream. Define $\Lambda_{s,\ell}^{(j)} \triangleq \{\Lambda_t^{(j)}, \lambda_{s,\ell}^{(j)}\}$ for $s = t + 1, \dots, t + \Delta$ and $\Lambda_{s,\ell}^{(j)} \triangleq \{\Lambda_t^{(j)}, \lambda_{t+1,\ell}^{t+\Delta}, \lambda_{s,\ell}^{(j)}\}$ for $s = t + \Delta + 1, \dots, t + \Delta + \Delta'$.

For every possible value of $\ell \in |\mathcal{A}|^\Delta$, define the *single path weight* as

$$\begin{aligned} \tilde{w}_{t,\ell}^{(j)} &= \bar{w}_t^{(j)} \frac{p(\Lambda_{t+\Delta+\Delta',\ell}^{(j)} | \mathbf{Y}_{t+\Delta+\Delta'})}{p(\Lambda_t^{(j)} | \mathbf{Y}_t) \prod_{s=t+\Delta+1}^{t+\Delta+\Delta'} p(\lambda_{s,\ell}^{(j)} | \Lambda_{s-1,\ell}^{(j)}, \mathbf{Y}_s)} \\ &= \bar{w}_t^{(j)} \prod_{s=t+1}^{t+\Delta} \frac{p(\Lambda_{s,\ell}^{(j)} | \mathbf{Y}_s)}{p(\Lambda_{s-1,\ell}^{(j)} | \mathbf{Y}_{s-1})} \\ &\quad \times \prod_{s=t+\Delta+1}^{t+\Delta+\Delta'} \frac{p(\Lambda_{s,\ell}^{(j)} | \mathbf{Y}_s)}{p(\Lambda_{s-1,\ell}^{(j)} | \mathbf{Y}_{s-1}) p(\lambda_{s,\ell}^{(j)} | \Lambda_{s-1,\ell}^{(j)}, \mathbf{Y}_s)} \\ &\propto \bar{w}_t^{(j)} \prod_{s=t+1}^{t+\Delta} p(\mathbf{y}_s | \mathbf{Y}_{s-1}, \Lambda_{s,\ell}^{(j)}) p(\lambda_{s,\ell}^{(j)} | \Lambda_{s-1,\ell}^{(j)}) \\ &\quad \times \prod_{s=t+\Delta+1}^{t+\Delta+\Delta'} p(\mathbf{y}_s | \mathbf{Y}_{s-1}, \Lambda_{s-1,\ell}^{(j)}) \end{aligned} \quad (31)$$

$$\begin{aligned}
&= \bar{w}_t^{(j)} \frac{\gamma_t^{(j)}(\lambda_t = a_i, \lambda_{t+1,\ell}^{t+\Delta})}{p(\mathbf{y}_t | \mathbf{Y}_{t-1}, \Lambda_t^{(j)}) p(\lambda_t = a_i | \Lambda_{t-1}^{(j)})} \\
&\quad \times \prod_{s=t+\Delta+1}^{t+\Delta+\Delta'} \gamma_s^{(j)}(\lambda_t = a_i, \lambda_{t+1,\ell}^{t+\Delta}) \quad (32)
\end{aligned}$$

$$\begin{aligned}
&= \frac{\bar{w}_{t-1}^{(j)}}{\tilde{\rho}_{t,i}^{(j)}} \gamma_t^{(j)}(\lambda_t = a_i, \lambda_{t+1,\ell}^{t+\Delta}) \\
&\quad \times \prod_{s=t+\Delta+1}^{t+\Delta+\Delta'} \gamma_s^{(j)}(\lambda_t = a_i, \lambda_{t+1,\ell}^{t+\Delta}) \quad (33)
\end{aligned}$$

where $\gamma_t^{(j)}(\cdot)$, $\gamma_s^{(j)}(\cdot)$, and $\tilde{\rho}_{t,i}^{(j)}$ are defined by (22), (26), and (28), respectively.

Define the auxiliary weight as

$$\tilde{w}_t^{(j)} \triangleq \sum_{\ell=1}^{A_\Delta} \tilde{w}_{t,\ell}^{(j)}. \quad (34)$$

We next show that $\{\Lambda_t^{(j)}, \tilde{w}_t^{(j)}\}$ is properly weighted with respect to $p(\Lambda_t | \mathbf{Y}_{t+\Delta+\Delta'})$. First, for any measurable function h on Λ_t , define an extended function h^* on $\Lambda_{t+\Delta+\Delta'}$ such that $h^*(\Lambda_t, \Lambda_{t+1}^{t+\Delta+\Delta'}) = h(\Lambda_t)$ for all $\Lambda_{t+1}^{t+\Delta+\Delta'}$.

Let c be the constant that makes (31) an equality. It is easily seen c is a constant for all ℓ . By the construction of $\tilde{w}_{t,\ell}$, we have

$$\begin{aligned}
&\frac{1}{m} \sum_{j=1}^m h(\Lambda_t^{(j)}) \cdot c \cdot \tilde{w}_{t,\ell}^{(j)} \\
&= \frac{1}{m} \sum_{j=1}^m h^*(\Lambda_{t+\Delta+\Delta',\ell}^{(j)}) \cdot c \cdot \tilde{w}_{t,\ell}^{(j)} \\
&\rightarrow \int h^*(\Lambda_{t+\Delta+\Delta'}^{(j)}) p(\Lambda_{t+\Delta+\Delta',\ell}^{(j)} | \mathbf{Y}_{t+\Delta+\Delta'}) d\Lambda_{t+\Delta+\Delta'}^{t+\Delta+\Delta'} \\
&= p(\Lambda_{t+1}^{t+\Delta} = \lambda_{t+1,\ell}^{t+\Delta} | \mathbf{Y}_{t+\Delta+\Delta'}) \\
&\quad \times \int \left\{ h(\Lambda_t^{(j)}) p(\Lambda_t^{(j)}, \Lambda_{t+1}^{t+\Delta+\Delta'} | \Lambda_{t+1}^{t+\Delta} \right. \\
&\quad \left. = \lambda_{t+1,\ell}^{t+\Delta}, \mathbf{Y}_{t+\Delta+\Delta'}) \right\} d\Lambda_{t+\Delta+\Delta'}^{t+\Delta+\Delta'} \\
&= p(\Lambda_{t+1}^{t+\Delta} = \lambda_{t+1,\ell}^{t+\Delta} | \mathbf{Y}_{t+\Delta+\Delta'}) \\
&\quad \times E[h(\Lambda_t) | \mathbf{Y}_{t+\Delta+\Delta'}, \Lambda_{t+1}^{t+\Delta} = \lambda_{t+1,\ell}^{t+\Delta}]. \quad (35)
\end{aligned}$$

Hence

$$\begin{aligned}
&E[h(\Lambda_t) | \mathbf{Y}_{t+\Delta+\Delta'}] \\
&= \sum_{\ell} \left\{ E[h(\Lambda_t) | \mathbf{Y}_{t+\Delta+\Delta'}, \Lambda_{t+1}^{t+\Delta} = \lambda_{t+1,\ell}^{t+\Delta}] \right. \\
&\quad \left. \times p(\Lambda_{t+1}^{t+\Delta} = \lambda_{t+1,\ell}^{t+\Delta}) \right\} \\
&\cong \sum_{\ell} \frac{1}{m} \sum_{j=1}^m h(\Lambda_t^{(j)}) \cdot c \cdot \tilde{w}_{t,\ell}^{(j)} \\
&= \frac{c}{m} \sum_{j=1}^m h(\Lambda_t^{(j)}) \sum_{\ell} \tilde{w}_{t,\ell}^{(j)} \\
&= \frac{c}{m} \sum_{j=1}^m h(\Lambda_t^{(j)}) \tilde{w}_t^{(j)}. \quad (36)
\end{aligned}$$

By the same argument as in standard SMC [23], we replace c/m by $\tilde{W}_t = \sum_{j=1}^m \tilde{w}_t^{(j)}$.

IV. ADAPTIVE DETECTION/DECODING IN FADING CHANNELS VIA DELAYED-PILOT MKF

A. System Description

We consider a communication system signaling through a flat-fading channel with additive white Gaussian noise. The complex data symbols $\{s_t\}$ take values from a finite alphabet set $\mathcal{A} = \{a_1, \dots, a_{|\mathcal{A}|}\}$, and each symbol is transmitted over a flat-fading channel whose input-output relationship is given by

$$y_t = \alpha_t s_t + n_t, \quad t = 0, 1, \dots \quad (37)$$

where y_t , α_t , s_t , and n_t are the received signal, the fading channel coefficient, the transmitted symbol, and the additive Gaussian noise at time t , respectively. The processes $\{\alpha_t\}$, $\{s_t\}$, and $\{n_t\}$ are assumed to be mutually independent.

As in [4], it is assumed that the channel-fading process is Rayleigh. That is, the fading coefficients $\{\alpha_t\}$ form a complex Gaussian process that can be modeled by the output of a lowpass Butterworth filter of order r driven by white Gaussian noise

$$\{\alpha_t\} = \frac{\Theta(z^{-1})}{\Phi(z^{-1})} \{u_t\} \quad (38)$$

where $\Phi(z) \triangleq \phi_r z^r + \dots + \phi_1 z + 1$; $\Theta(z) \triangleq \theta_r z^r + \dots + \theta_1 z + \theta_0$; and $\{u_t\}$ is a white complex Gaussian noise sequence with unit variance and independent real and complex components. We can rewrite (37) and (38) in the following state-space model form:

$$\mathbf{x}_t = \mathbf{F} \mathbf{x}_{t-1} + \mathbf{g} u_t \quad (39)$$

$$y_t = s_t \mathbf{h}^H \mathbf{x}_t + \sigma v_t \quad (40)$$

where

$$\mathbf{F} \triangleq \begin{pmatrix} -\phi_1 & -\phi_2 & \dots & -\phi_r & 0 \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{pmatrix}$$

and

$$\mathbf{g} \triangleq \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad \mathbf{h} \triangleq \begin{pmatrix} \theta_0^* \\ \theta_1^* \\ \vdots \\ \theta_r^* \end{pmatrix}$$

$u_t \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}_c(0, 1), \quad v_t \stackrel{\text{i.i.d.}}{\sim} \mathcal{N}_c(0, 1).$

B. Adaptive Detection in Uncoded System

Denote $\mathbf{Y}_t \triangleq (y_0, \dots, y_t)$ and $\mathbf{S}_t \triangleq (s_0, \dots, s_t)$. For the case of the uncoded system, the transmitted symbols are assumed to be independent, i.e.,

$$P(s_t = a_i | \mathbf{S}_{t-1}) = P(s_t = a_i), \quad a_i \in \mathcal{A}. \quad (41)$$

When no prior information about the symbols is available, the symbols are assumed to take each possible value in $\mathcal{A}$ with equal

probability, i.e., $P(s_t = a_i) = 1/|\mathcal{A}|$ for $i = 1, \dots, |\mathcal{A}|$. We are interested in computing the symbol *a posteriori* probabilities

$$P(s_t = a_i | \mathbf{Y}_{t+\Delta}), \quad a_i \in \mathcal{A}, \quad t = 1, 2, \dots \quad (42)$$

In [4], we have solved this problem based on the concurrent MKF algorithm and the delayed-sample MKF algorithm. Here, we consider solving the same problem using the simple delayed-pilot MKF algorithm and the delayed-hybrid MKF algorithm discussed in Section III. By the state-space mode (39) and (40), this is quite straightforward—we only need to make the following substitution of variables in Algorithm 1 and Algorithm 2 respectively, namely

$$\lambda_t \leftarrow \mathbf{s}_t, \quad \Lambda_t \leftarrow \mathbf{S}_t, \quad \mathbf{y}_t \leftarrow \mathbf{y}_t$$

and

$$\mathbf{F}_{\lambda_t} = \mathbf{F}, \quad \mathbf{G}_{\lambda_t} = \mathbf{G}, \quad \mathbf{H}_{\lambda_t} = s_t \mathbf{h}^H, \quad \mathbf{K}_{\lambda_t} = \sigma. \quad (43)$$

C. Adaptive Decoding in Coded System

We next consider the problem of adaptive decoding in a convolutionally coded system signaling through a flat-fading channel using the delay-pilot sampling technique.

Consider a binary rate k_0/n_0 convolutional encoder of overall constraint length $k_0\nu_0$. Suppose the encoder starts with an all-zero state at time $t = 0$. The input to the encoder at time t is a block of information bits $\mathbf{d}_t = (d_{t,1}, \dots, d_{t,k_0})$; the encoder output at time t is a block of code bits $\mathbf{b}_t = (b_{t,1}, \dots, b_{t,n_0})$. For simplicity, here, we assume that BPSK modulation is employed. Then, the transmitted symbols at time t are $\mathbf{s}_t = (s_{t,1}, \dots, s_{t,n_0})$, where $s_{t,l} = 2b_{t,l} - 1$, $l = 1, \dots, n_0$. (That is, $s_{t,l} = 1$ if $b_{t,l} = 1$, and $s_{t,l} = -1$ if $b_{t,l} = 0$.) Since $\mathbf{b}_t$ is determined by $(\mathbf{d}_t, \mathbf{d}_{t-1}, \dots, \mathbf{d}_{t-\nu_0})$, so is $\mathbf{s}_t$. Hence, we can write

$$\mathbf{s}_t = \psi(\mathbf{d}_t, \mathbf{d}_{t-1}, \dots, \mathbf{d}_{t-\nu_0}) \quad (44)$$

for some function $\psi(\cdot)$ determined by the structure of the encoder. We also assume the information bits $\mathbf{d}_t$ are independent.

Let $\mathbf{y}_t = (y_{t,1}, \dots, y_{t,n_0})$ be the received signals at time t , and let $\alpha_t = (\alpha_{t,1}, \dots, \alpha_{t,n_0})$ be the channel states corresponding to $\mathbf{b}_t$ and $\mathbf{d}_t$. Recall that $\alpha_{t,l} = \mathbf{h}^H \mathbf{x}_{t-1,l}$. In addition, denote $\mathbf{D}_t \triangleq (\mathbf{d}_0, \dots, \mathbf{d}_t)$; $\mathbf{S}_t \triangleq (\mathbf{s}_0, \dots, \mathbf{s}_t)$; $\mathbf{Y}_t \triangleq (\mathbf{y}_0, \dots, \mathbf{y}_t)$. The Monte Carlo samples recorded at time t are $\{(\mathbf{D}_t^{(j)}, \kappa_t^{(j)}, w_t^{(j)})\}_{j=1}^m$, where

$$\kappa_t^{(j)} \triangleq [\mu_t^{(j)}, \Sigma_t^{(j)}]$$

contains the mean and covariance matrix of the channel state vector $\mathbf{x}_t$ conditioned on $\mathbf{D}_t^{(j)}$ and $\mathbf{Y}_t$. That is

$$p(\mathbf{x}_t | \mathbf{D}_t^{(j)}, \mathbf{Y}_t) \sim \mathcal{N}(\mu_t^{(j)}, \Sigma_t^{(j)}). \quad (45)$$

As before, given the information bit sequence $\mathbf{D}_t^{(j)}$, the corresponding $\kappa_t^{(j)}$ is obtained by a Kalman filter.

The adaptive decoding algorithms based on the delayed-pilot MKF is as follows.

Algorithm 3 [Delayed-Pilot Adaptive Decoder]: Initialization (For $j = 1, \dots, m$): Each Kalman filter is initialized as

$\kappa_0^{(j)} = (\mu_0^{(j)}, \Sigma_0^{(j)})$, with $\mu_0^{(j)} = \mathbf{0}$ and $\Sigma_0^{(j)} = 2\Sigma$, where Σ is the stationary covariance of $\mathbf{x}_t$ and is computed analytically from (39). All importance weights are initialized as $w_0^{(j)} = 1$. The initial $\mathbf{D}_0^{(j)}$ are randomly generated from the set $\{0, 1\}^{k_0}$.

Suppose at time $(t + \Delta - 1)$, a set of properly weighted samples $\{(\mathbf{D}_{t-1}^{(j)}, \kappa_{t-1}^{(j)}, w_{t-1}^{(j)})\}_{j=1}^m$ with respect to $p(\mathbf{D}_{t-1} | \mathbf{Y}_{t-1})$ and corresponding transmitted symbols $\{\mathbf{S}_{t-1}^{(j)}\}_{j=1}^m$ are available. Then, at time $(t + \Delta)$, as the new data $\mathbf{y}_{t+\Delta}$ become available, the following steps are implemented to update each weighted sample.

For $j = 1, \dots, m$:

- For each possible information bit vector $\mathbf{d}_t = \mathbf{a}_i \in \{0, 1\}^{k_0}$, compute the corresponding $\mathbf{s}_t$ vector to obtain

$$\mathbf{s}_t^{(j)}(\mathbf{a}_i) = \psi(\mathbf{d}_t = \mathbf{a}_i, \mathbf{d}_{t-1}^{(j)}, \dots, \mathbf{d}_{t-\nu_0}^{(j)}). \quad (46)$$

Then, run n_0 steps of Kalman filter update to obtain

$$\begin{aligned} \kappa_t^{(i,j)} \mathbf{y}_t, \mathbf{s}_t^{(j)}(\mathbf{a}_i) &\xrightarrow{\kappa_{t,i}^{(j)}} \kappa_{t,i}^{(j)} \\ &\triangleq [\mu_t(\mathbf{D}_{t-1}^{(j)}, \mathbf{d}_t = \mathbf{a}_i), \Sigma_t(\mathbf{D}_{t-1}^{(j)}, \mathbf{d}_t = \mathbf{a}_i)] \end{aligned} \quad (47)$$

and compute

$$\begin{aligned} \gamma_{t,i}^{(j)} &\triangleq P(\mathbf{d}_t = \mathbf{a}_i | \mathbf{D}_{t-1}^{(j)}, \mathbf{Y}_t) \\ &\propto p(\mathbf{y}_t | \mathbf{d}_t = \mathbf{a}_i, \mathbf{D}_{t-1}^{(j)}, \mathbf{Y}_{t-1}) P(\mathbf{d}_t = \mathbf{a}_i). \end{aligned} \quad (48)$$

- For each $\mathbf{a}_i \in \{0, 1\}^{k_0}$, do the following:
For $l = t + 1, \dots, t + \Delta$, repeat

— Denote

$$\mathbf{D}_{l-1}^{(i,j)} = [\mathbf{D}_{t-1}^{(j)}, \mathbf{d}_t = \mathbf{a}_i, \mathbf{d}_{t+1} = \mathbf{d}_{t+1}^{(i,j)}, \dots, \mathbf{d}_{l-1} = \mathbf{d}_{l-1}^{(i,j)}]$$

and the corresponding transmitted symbols

$$\mathbf{S}_{l-1}^{(i,j)} = [\mathbf{S}_{t-1}^{(j)}, \mathbf{s}_t^{(j)}(\mathbf{a}_i), \mathbf{s}_{t+1} = \mathbf{s}_{t+1}^{(i,j)}, \dots, \mathbf{s}_{l-1} = \mathbf{s}_{l-1}^{(i,j)}].$$

For each $\mathbf{d}_l = \mathbf{a}_k \in \{0, 1\}^{k_0}$, compute the corresponding transmitted symbols $\mathbf{s}_l$, and run n_0 steps of Kalman filter update to obtain

$$\begin{aligned} \kappa_{l-1}^{(j)} \mathbf{y}_l, \mathbf{s}_l &\xrightarrow{\kappa_l^{(k,j)}} \kappa_l^{(k,j)} \\ &\triangleq [\mu_l(\mathbf{D}_{l-1}^{(i,j)}, \mathbf{d}_l = \mathbf{a}_k), \Sigma_l(\mathbf{D}_{l-1}^{(i,j)}, \mathbf{d}_l = \mathbf{a}_k)]. \end{aligned} \quad (49)$$

- For each $\mathbf{a}_k \in \{0, 1\}^{k_0}$, compute the sampling density

$$\begin{aligned} \gamma_{l,i}^{(k,j)} &\triangleq P(\mathbf{d}_l = \mathbf{a}_k | \mathbf{D}_{l-1}^{(i,j)}, \mathbf{Y}_l) \\ &\propto p(\mathbf{y}_l | \mathbf{d}_l = \mathbf{a}_k, \mathbf{D}_{l-1}^{(i,j)}, \mathbf{Y}_{l-1}) P(\mathbf{d}_l = \mathbf{a}_k). \end{aligned} \quad (50)$$

Draw a sample $\mathbf{d}_l^{(i,j)}$ according to the above sampling density. If $\mathbf{d}_l^{(i,j)} = \mathbf{a}_\ell$ is drawn, then set $\kappa_l^{(j)} = \kappa_{l,i}^{(\ell,j)}$.

- Compute the increment importance weight

$$\begin{aligned} \gamma_{l,i}^{(j)} &= p(\mathbf{y}_l | \mathbf{D}_{l-1}^{(i,j)}, \mathbf{Y}_{l-1}) \\ &\propto \sum_{k=1}^{2^{k_0}} \gamma_{l,i}^{(k,j)}. \end{aligned} \quad (51)$$

- For each $\mathbf{a}_i \in \{0, 1\}^{k_0}$, compute

$$\rho_{t,i}^{(j)} = \gamma_{t,i}^{(j)} \prod_{l=t+1}^{t+\Delta} \gamma_{l,i}^{(j)}. \quad (52)$$

Using

$$\tilde{\rho}_{t,i}^{(j)} \triangleq \frac{\rho_{t,i}^{(j)}}{2^{k_0}}, \quad i = 1, \dots, 2^{k_0} \quad (53)$$

as the trial sampling distribution, draw a sample $\mathbf{d}_t^{(j)}$. Append $\mathbf{d}_t^{(j)}$ to $\mathbf{D}_{t-1}^{(j)}$ to obtain $\mathbf{D}_t^{(j)}$. If $\mathbf{d}_t^{(j)} = \mathbf{a}_i$ is drawn, then set $\kappa_t^{(j)} = \kappa_{t,i}^{(j)}$.

- Compute the importance weight. If $\mathbf{d}_t^{(j)} = \mathbf{a}_i$, then

$$\begin{aligned} w_t^{(j)} &= w_{t-1}^{(j)} \cdot \frac{p(\mathbf{D}_t^{(j)} | \mathbf{Y}_t)}{p(\mathbf{D}_{t-1}^{(j)} | \mathbf{Y}_{t-1}) \tilde{\rho}_{t,i}^{(j)}} \\ &= w_{t-1}^{(j)} \cdot \frac{\gamma_{t,i}^{(j)}}{\tilde{\rho}_{t,i}^{(j)}}. \end{aligned} \quad (54)$$

- Do resampling if the effective sample size is below certain threshold.

The adaptive decoding algorithms based on the delayed-hybrid MKF is as follows.

Algorithm 4 [Delayed-Hybrid Adaptive Decoder]: Initialization: Same as in Algorithm 4.

Suppose that at time $(t + \Delta + \Delta' - 1)$, a set of properly weighted samples $\{(\mathbf{D}_{t-1}^{(j)}, \kappa_{t-1}^{(j)}, w_{t-1}^{(j)})\}_{j=1}^m$ with respect to $p(\mathbf{D}_{t-1} | \mathbf{Y}_{t+\Delta-1})$ and the corresponding transmitted signals $\{\mathbf{S}_{t-1}^{(j)}\}_{j=1}^m$ are available; then, at time $(t + \Delta + \Delta')$, as the new data $\mathbf{y}_{t+\Delta+\Delta'}$ become available, the following steps are implemented to update each weighted sample.

For $j = 1, 2, \dots, m$:

- For each $\mathbf{d}_{t+\Delta} = \mathbf{a}_i \in \{0, 1\}^{k_0}$, and for each $\mathbf{d}_t^{t+\Delta-1} \in \{0, 1\}^{k_0\Delta}$, compute the corresponding transmitted symbols $\mathbf{s}_{t+\Delta}$. Perform n_0 steps update on the corresponding Kalman filter $\kappa_{t+\Delta-1}^{(j)}(\mathbf{d}_t^{t+\Delta-1})$, i.e.,

$$\kappa_{t+\Delta-1}^{(j)}(\mathbf{d}_t^{t+\Delta-1}) \xrightarrow{\mathbf{y}_{t+\Delta}, \mathbf{s}_{t+\Delta}} \kappa_{t+\Delta}^{(j)}(\mathbf{d}_t^{t+\Delta-1}, \mathbf{d}_{t+\Delta} = \mathbf{a}_i) \quad (55)$$

and compute

$$\begin{aligned} \gamma_t^{(j)}(\mathbf{d}_t^{t+\Delta}) &\triangleq p(\mathbf{d}_t^{t+\Delta} | \mathbf{D}_{t-1}^{(j)}, \mathbf{Y}_{t+\Delta}) \\ &= \prod_{\tau=0}^{\Delta} p(\mathbf{y}_{t+\tau} | \mathbf{Y}_{t+\tau-1}, \mathbf{D}_{t-1}^{(j)}, \mathbf{d}_t^{t+\tau}) \\ &\quad \times \prod_{\tau=0}^{\Delta} p(\mathbf{d}_{t+\tau}). \end{aligned} \quad (56)$$

- For every path of $\mathbf{d}_t^{t+\Delta} \in \{0, 1\}^{k_0(\Delta+1)}$ do the following:
For $l = t + \Delta + 1, \dots, t + \Delta + \Delta'$, repeat:

— Denote

$$\begin{aligned} \mathbf{D}_{l-1}^{(j)}(\mathbf{d}_t^{t+\Delta}) &\triangleq [\mathbf{D}_{t-1}^{(j)}, \mathbf{d}_t^{t+\Delta}, \mathbf{d}_{t+\Delta+1} = \mathbf{d}_{t+\Delta+1}^{(j)}(\mathbf{d}_t^{t+\Delta}), \dots, \\ &\quad \mathbf{d}_{l-1} = \mathbf{d}_{l-1}^{(j)}(\mathbf{d}_t^{t+\Delta})] \end{aligned} \quad (57)$$

and the corresponding transmitted symbols

$$\begin{aligned} \mathbf{S}_{l-1}^{(j)}(\mathbf{d}_t^{t+\Delta}) &\triangleq [\mathbf{S}_{t-1}^{(j)}, \mathbf{s}_t^{t+\Delta}, \mathbf{s}_{t+\Delta+1} = \mathbf{s}_{t+\Delta+1}^{(j)}(\mathbf{d}_t^{t+\Delta}), \dots, \\ &\quad \mathbf{s}_{l-1} = \mathbf{s}_{l-1}^{(j)}(\mathbf{d}_t^{t+\Delta})]. \end{aligned} \quad (58)$$

For each $\mathbf{d}_l = \mathbf{a}_k \in \{0, 1\}^{k_0}$, compute the corresponding transmitted symbols $\mathbf{s}_l$, and run n_0 steps Kalman filter update to obtain

$$\begin{aligned} \kappa_{l-1}^{(j)}(\mathbf{d}_t^{t+\Delta}) &\xrightarrow{\mathbf{y}_l, \mathbf{s}_l} \kappa_l^{(k,j)}(\mathbf{d}_t^{t+\Delta}) \\ &\times [\mu_l(\mathbf{D}_{l-1}^{(j)}(\mathbf{d}_t^{t+\Delta}), \mathbf{d}_l = \mathbf{a}_k) \\ &\quad \Sigma_l(\mathbf{D}_{l-1}^{(j)}(\mathbf{d}_t^{t+\Delta}), \mathbf{d}_l = \mathbf{a}_m)]. \end{aligned} \quad (59)$$

- For each $\mathbf{a}_k \in \{0, 1\}^{k_0}$, compute the sampling density

$$\begin{aligned} \gamma_l^{(k,j)}(\mathbf{d}_t^{t+\Delta}) &\triangleq P[\mathbf{d}_l = \mathbf{a}_k | \mathbf{D}_{l-1}^{(j)}(\mathbf{d}_t^{t+\Delta}), \mathbf{Y}_l] \\ &\propto p[\mathbf{y}_l | \mathbf{d}_l = \mathbf{a}_k, \mathbf{D}_{l-1}^{(j)}(\mathbf{d}_t^{t+\Delta}), \mathbf{Y}_{l-1}] P(\mathbf{d}_l = \mathbf{a}_k). \end{aligned} \quad (60)$$

Draw a sample $\mathbf{d}_l^{(j)}(\mathbf{d}_t^{t+\Delta})$ according to the above sampling density. If $\mathbf{d}_l^{(j)}(\mathbf{d}_t^{t+\Delta}) = \mathbf{a}_\ell$, set

$$\kappa_s^{(j)}(\mathbf{d}_t^{t+\Delta}) = \kappa_s^{(\ell,j)}(\mathbf{d}_t^{t+\Delta}).$$

- Compute the increment importance weight

$$\begin{aligned} \gamma_t^{(j)}(\mathbf{d}_t^{t+\Delta}) &= p(\mathbf{y}_t | \mathbf{D}_{t-1}^{(j)}(\mathbf{d}_t^{t+\Delta}), \mathbf{Y}_{t-1}) \\ &\propto \sum_{k=1}^{2^{k_0}} \gamma_s^{(k,j)}(\mathbf{d}_t^{t+\Delta}). \end{aligned} \quad (61)$$

- For each $\mathbf{a}_i \in \{0, 1\}^{k_0}$, compute

$$\begin{aligned} \rho_{t,i}^{(j)} &= \sum_{\mathbf{d}_{t+\Delta}^{t+\Delta} \in \{0,1\}^{k_0\Delta}} \left[\gamma_t^{(j)}(\mathbf{d}_t = \mathbf{a}_i, \mathbf{d}_{t+\Delta}^{t+\Delta}) \prod_{l=t+\Delta+1}^{t+\Delta+\Delta'} \right. \\ &\quad \left. \times \gamma_l^{(j)}(\mathbf{d}_t = \mathbf{a}_i, \mathbf{d}_{t+\Delta}^{t+\Delta}) \right]. \end{aligned} \quad (62)$$

Using

$$\hat{\rho}_{t,i}^{(j)} \triangleq \frac{\rho_{t,i}^{(j)}}{\sum_{k=1}^{2^{k_0}} \rho_{t,k}^{(j)}}, \quad i = 1, \dots, 2^{k_0} \quad (63)$$

as the trial sampling distribution, draw a sample $\mathbf{d}_t^{(j)}$ according to the above sampling density. Append $\mathbf{d}_t^{(j)}$ to $\mathbf{D}_{t-1}^{(j)}$ to obtain $\mathbf{D}_t^{(j)}$. Based on this sample, form $\kappa_t^{(j)}$ using the results from the first step.

- Compute the importance weight. Suppose $\mathbf{d}_{t-1}^{(j)} = \mathbf{a}_k$ and $\mathbf{d}_t^{(j)} = \mathbf{a}_i$; then

$$\begin{aligned} w_t^{(j)} &= w_{t-1}^{(j)} \frac{p(\mathbf{D}_t^{(j)} | \mathbf{Y}_{t+\Delta})}{p(\mathbf{D}_{t-1}^{(j)} | \mathbf{Y}_{t+\Delta-1}) \hat{\rho}_{t,i}^{(j)}} \\ &\propto \frac{w_{t-1}^{(j)}}{\hat{\rho}_{t,i}^{(j)}} \\ &\quad \times \frac{\sum_{\mathbf{d}_{t+1}^{t+\Delta} \in \{0,1\}^{k_0\Delta}} \left[\prod_{\tau=0}^{\Delta} p(\mathbf{y}_{t+\tau} | \mathbf{Y}_{t+\tau-1}, \mathbf{D}_t^{(j)}, \mathbf{d}_{t+1}^{t+\tau}) \right]}{\sum_{\mathbf{d}_t^{t+\Delta-1} \in \{0,1\}^{k_0\Delta}} \left[\prod_{\tau=0}^{\Delta-1} p(\mathbf{y}_{t+\tau} | \mathbf{Y}_{t+\tau-1}, \mathbf{D}_t^{(j)}, \mathbf{d}_{t+1}^{t+\tau}) \right]} \\ &\quad \times \frac{\prod_{\tau=0}^{\Delta} p(\mathbf{d}_{t+\tau})}{\prod_{\tau=0}^{\Delta-1} p(\mathbf{d}_{t+\tau})} \\ &= \frac{w_{t-1}^{(j)}}{\hat{\rho}_{t,i}^{(j)}} \\ &\quad \times \frac{\sum_{\mathbf{d}_{t+1}^{t+\Delta} \in \{0,1\}^{k_0\Delta}} \gamma_t^{(j)}(\mathbf{d}_t = \mathbf{a}_i, \mathbf{d}_{t+1}^{t+\Delta})}{\sum_{\mathbf{d}_t^{t+\Delta-1} \in \{0,1\}^{k_0\Delta}} \gamma_{t-1}^{(j)}(\mathbf{d}_{t-1} = \mathbf{a}_k, \mathbf{d}_t = \mathbf{a}_i, \mathbf{d}_{t+1}^{t+\Delta-1})} \\ &\quad \times p(\mathbf{y}_{t-1} | \mathbf{Y}_{t-2}, \mathbf{D}_{t-1}^{(j)}) P(\mathbf{d}_{t-1} = \mathbf{a}_k). \end{aligned} \quad (64)$$

- Do resampling if the effective sample size is below certain threshold.

V. SIMULATION RESULTS

In this section, we provide some computer simulation examples to demonstrate the performance of the adaptive receiver in fading channels that employs the proposed delayed-pilot sampling techniques. As in [4], the fading process is modeled by the output of a Butterworth filter of order $r = 3$ driven by a complex white Gaussian noise process. The cut-off 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

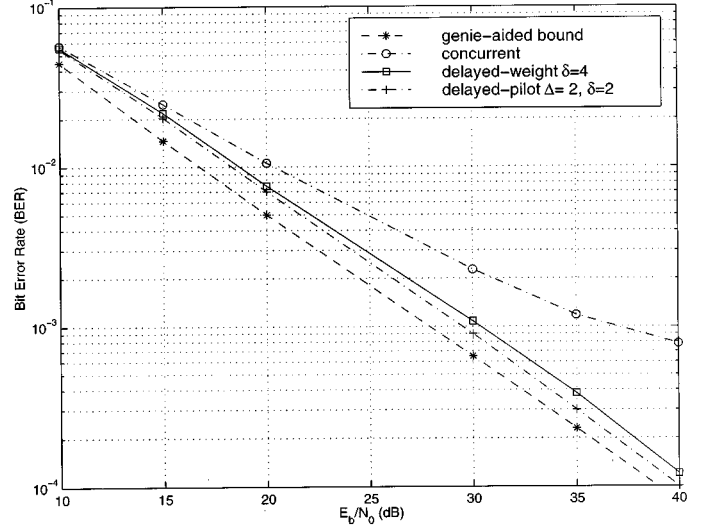


Fig. 4. BER performance of the concurrent MKF receiver, the delayed-weight MKF receiver with $\delta = 2$, and the simple delayed-pilot MKF receiver with $\Delta = 2$, $\delta = 2$ in an uncoded system.

fast fading scenario. Specifically, the fading coefficients $\{\alpha_t\}$ are modeled by the following ARMA(3,3) process:

$$\begin{aligned} \alpha_t &- 2.37409\alpha_{t-1} + 1.92936\alpha_{t-2} - 0.53208\alpha_{t-3} \\ &= 10^{-2}(0.89409u_t + 2.68227u_{t-1} + 2.68227u_{t-2} \\ &\quad + 0.89409u_{t-3}) \end{aligned} \quad (65)$$

where $u_t \sim \mathcal{N}_c(0, 1)$. The filter coefficients in (65) are chosen such that $\text{Var}\{\alpha_t\} = 1$. It is assumed that BPSK modulation is employed, i.e., the transmitted symbols $s_t \in \{+1, -1\}$. In all algorithms, 100 Monte Carlo samples are used, i.e., $m = 100$.

The performance of the adaptive receivers based on the concurrent MKF, the delayed-weight MKF, and the delayed-sample MKF is studied in [4], which will serve as baseline for comparison here. Note that as illustrated in [4], for uncoded systems, the performance of the simple delayed-weight method is already very close to the “genie-aided lower bound,”¹ and the delayed-sample method offers not much further improvement. Hence, it can be inferred that in this case, both the delayed-pilot method and the delayed-hybrid method will also perform close to this lower bound, which is indeed confirmed by simulations, as shown in Fig. 4, where it is seen that the performance of the combined delayed-pilot method and delayed-weight method with $\Delta = 2$ and $\delta = 2$ offers a slightly better performance than the delayed-weight method with a delay $\delta = 4$.

It is also shown in [4] that for coded systems, the delayed-weight method is no longer effective in exploiting the strong memory of the system, and it is necessary to employ the delayed-sample method for adaptive decoding, whose performance steadily improves as the delay Δ increases. In what follows, we illustrate the performance of the delayed-pilot

¹The 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, i.e., $\tilde{y}_t = \alpha_t + \tilde{n}_t$, 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, i.e., it computes $\hat{\alpha}_t = E\{\alpha_t | \mathbf{Y}_t\}$. The symbol is then demodulated according to $\hat{s}_t = \text{sign}(\Re\{\hat{\alpha}_t^* y_t\})$.

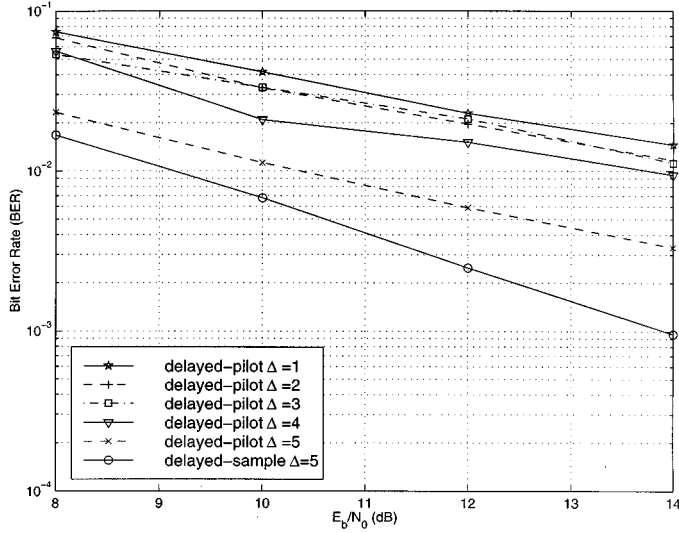


Fig. 5. BER performance of the simple pilot-delayed MKF receiver with different delays in a coded system.

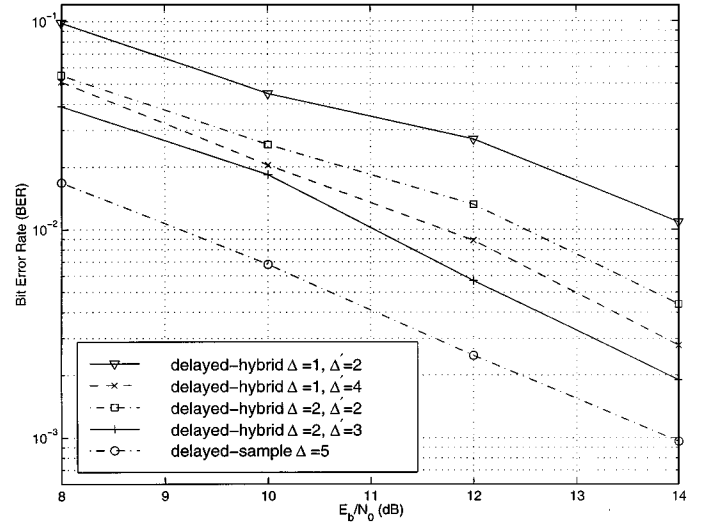


Fig. 7. BER performance of the delayed-hybrid MKF receiver with different delays in a coded system.

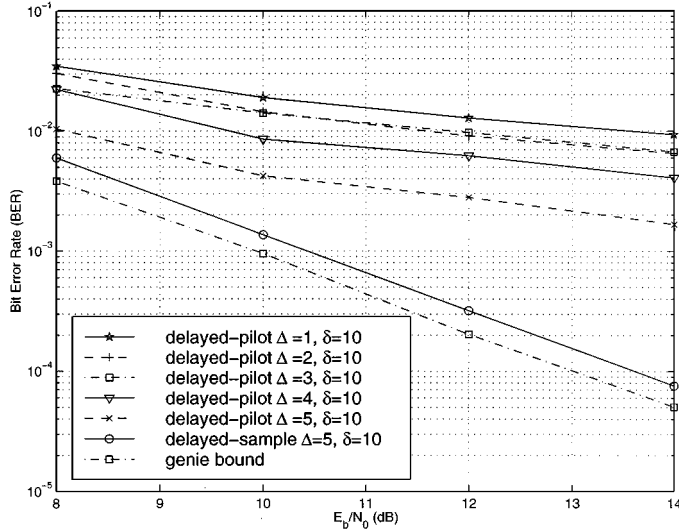


Fig. 6. BER performance of the receiver employing the combined delayed-pilot method (with $\Delta = 1, 2, 3, 4, 5$) and delayed-weight method (with $\delta = 10$) in a coded system.

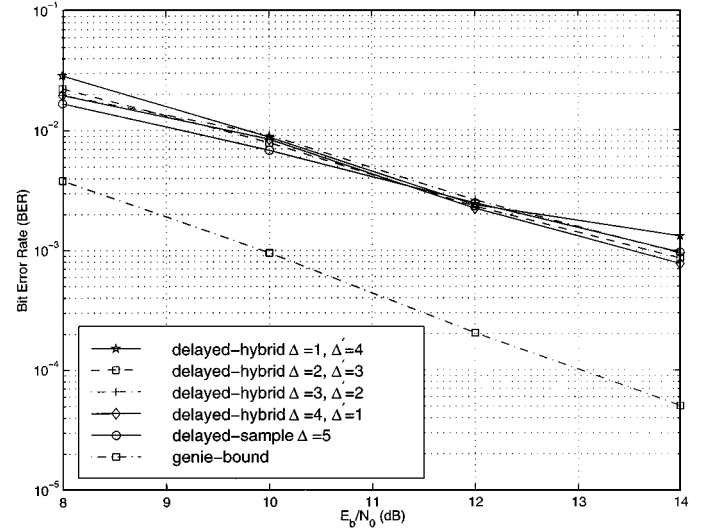


Fig. 8. BER performance of the delayed-hybrid MKF receiver with a fixed total delay ($\Delta + \Delta' = 5$) in a coded system.

method and that of the delayed-hybrid method in coded systems. The information bits are encoded using a rate 1/2 constraint length 5 convolutional code (with generators 23 and 25 in octal notation). The receiver implements the adaptive decoding algorithms discussed in Section IV in combination with the delayed-weight method. That is, the information bits samples $\{d_t^{(j)}\}_{j=1}^m$ are drawn by using the delayed-pilot method with delay Δ or the delayed-hybrid method with delay $\Delta + \Delta'$, whereas the importance weights $\{w_{t+\Delta+\Delta'+\delta}^{(j)}\}_{j=1}^m$ are obtained after a further delay of δ .

Figs. 5 and 6 illustrate the performance of receiver employing the simple delayed-pilot method with or without combining with the delayed-weight method. In the same figures, the performance of the concurrent MKF, as well as that of the delayed-sample MKF with a delay $\Delta = 5$, is shown. It is seen that the proposed delayed-pilot method is quite

effective in improving the receiver performance. In addition, increased performance improvement is obtained as the length of the pilot stream increases. Moreover, a further performance enhancement is achieved by using the delayed-weight method in conjunction with the delayed-pilot method. Figs. 7 and 8 illustrate the performance of receiver employing the delayed-hybrid method. Inferences are made using the auxiliary weight $\tilde{w}_{t+\Delta+\Delta'+\delta}$ at time $(t + \Delta + \Delta' + \delta)$. It is seen that this scheme offers further performance improvement over the delayed-pilot method (at the expense of increased computational complexity). Finally, the performance of the combined delayed-hybrid method and the delayed-weight method is shown in Fig. 9.

VI. CONCLUSIONS

In this paper, we have developed a new sequential Monte Carlo sampling method—the delayed-pilot sampling—under

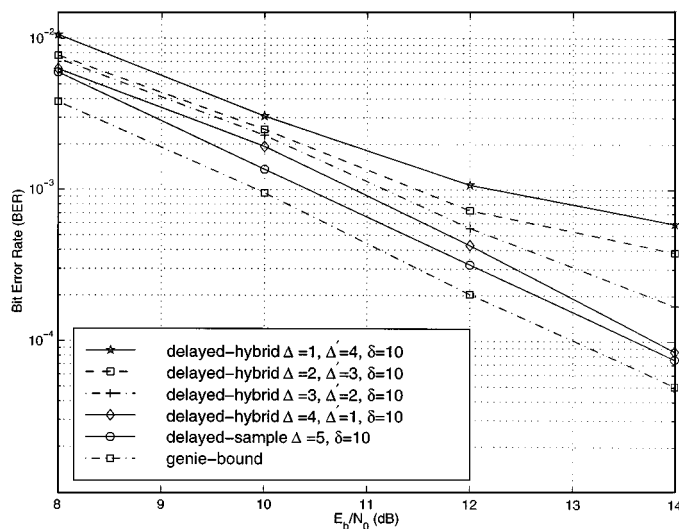


Fig. 9. BER performance of the receiver employing the combined delayed-hybrid method and delayed-weight method in a coded system.

the framework of mixture Kalman filter for delayed estimation in conditional dynamic linear systems. Instead of marginalizing the future states, as in the delayed-sample method [4], this new scheme generates random pilot streams using sequential important sampling, based on the future observations. The incremental importance weights of these pilot streams are then used to form the sampling density of the current state as well as the importance weights associated with the samples of the current state. The computational complexity of the delayed-pilot MKF algorithms is $O(m|\mathcal{A}|^2\Delta)$ for a Δ -step delay (where $\mathcal{A}$ is the finite set of possible states), whereas the corresponding complexity of the delayed-sample MKF algorithm in [4] is $O(m|\mathcal{A}|^\Delta)$. This technique can also be used in conjunction with the delayed-sample method, resulting in a hybrid sampling technique. For a Δ -step delayed-sample followed by a Δ' -step delayed-pilot, the complexity of the corresponding delayed-hybrid algorithm is $O(m|\mathcal{A}|^\Delta \cdot |\mathcal{A}|^2\Delta')$, whereas the full delayed-sample MKF algorithm has a complexity of $O(m|\mathcal{A}|^{\Delta+\Delta'})$ for a total delay of $\Delta + \Delta'$ steps. Moreover, the performance of both the delayed-pilot method and the delayed-hybrid method can be further enhanced when employed in conjunction with the delayed-weight method.

REFERENCES

- [1] D. Avitzour, "A stochastic simulation Bayesian approach to multitarget tracking," *Proc. Inst. Elect. Eng., Radar, Sonar, Navigat.*, vol. 142, pp. 41–44, 1995.
- [2] J. K. Cavers, "An analysis of pilot symbol assisted modulation for Rayleigh fading channels," *IEEE Trans. Veh. Technol.*, vol. 40, pp. 686–693, Nov. 1991.
- [3] R. Chen and J. S. Liu, "Mixture Kalman filters," *J. R. Statist. Soc. B*, vol. 62, no. 3, pp. 493–509, 2000.
- [4] R. Chen, X. Wang, and J. S. Liu, "Adaptive joint detection and decoding in flat-fading channels via mixture Kalman filtering," *IEEE Trans. Inform. Theory*, vol. 46, pp. 2079–2094, Sept. 2000.
- [5] I. B. Collings and J. B. Moore, "An HMM approach to adaptive demodulation of QAM signals in fading channels," *Int. J. Adaptive Contr. Signal Process.*, vol. 8, pp. 457–474, 1994.

- [6] —, "An adaptive hidden Markov model approach to FM and M-ary DPSK demodulation in noisy fading channels," *Signal Process.*, vol. 47, pp. 71–84, 1995.
- [7] A. N. D'Andrea, A. Diglio, and U. Mengali, "Symbol-aided channel estimation with nonselective Rayleigh fading channels," *IEEE Trans. Commun.*, vol. 44, pp. 41–48, Feb. 1995.
- [8] A. Doucet, "On sequential simulation-based methods for Bayesian filtering," *Statist. Comput.*, 2001, to be published.
- [9] A. Doucet, J. F. G. de Freitas, and N. Gordon, *Sequential Monte Carlo in Practice*. Cambridge, U.K.: Cambridge Univ. Press, 2001.
- [10] C. N. Georgiades and J. C. Han, "Sequence estimation in the presence of random parameters via the EM algorithm," *IEEE Trans. Commun.*, vol. 45, pp. 300–308, Mar. 1997.
- [11] M. J. Gertsman and J. H. Lodge, "Symbol-by-symbol MAP demodulation of CPM and PSK signals on Rayleigh flat fading channels," *IEEE Trans. Commun.*, vol. 45, pp. 788–799, July 1997.
- [12] N. J. Gordon, D. J. Salmon, and A. F. M. Smith, "A novel approach to nonlinear/non-Gaussian Bayesian state estimation," *Proc. Inst. Elect. Eng., Radar Signal Process.*, vol. 140, pp. 107–113, 1993.
- [13] R. Haeb and H. Meyr, "A systematic approach to carrier recovery and detection of digitally phase modulated signals on fading channels," *IEEE Trans. Commun.*, vol. 37, pp. 748–754, July 1989.
- [14] P. Hoeher and J. H. Lodge, "Turbo DPSK: Iterative differential PSK demodulation and channel decodings," *IEEE Trans. Commun.*, vol. 47, pp. 837–843, June 1999.
- [15] M. Hürzeler and H. R. Künsch, "Monte Carlo approximations for general state space models," *J. Comput. Graph. Statist.*, vol. 7, pp. 175–193, 1998.
- [16] G. T. Irvine and P. J. McLane, "Symbol-aided plus decision-directed reception for PSK/TCM modulation on shadowed mobile satellite fading channels," *IEEE J. Select. Areas Commun.*, vol. 10, pp. 1289–1299, Oct. 1992.
- [17] P. Y. Kam and H. M. Ching, "Sequence estimation over the slow nonselective Rayleigh fading channel with diversity reception and its application to Viterbi decoding," *IEEE J. Select. Areas Commun.*, vol. 10, pp. 562–570, Mar. 1992.
- [18] A. Kong, J. S. Liu, and W. H. Wong, "Sequential imputations and Bayesian missing data problems," *J. Amer. Statist. Assoc.*, vol. 89, pp. 278–288, 1994.
- [19] H. Kong and E. Shwedyk, "On channel estimation and sequence detection of interleaved coded signals over frequency nonselective Rayleigh fading channels," *IEEE Trans. Veh. Technol.*, vol. 47, pp. 558–565, May 1998.
- [20] J. S. Liu, *Monte Carlo Strategies for Scientific Computing*. New York: Springer-Verlag, 2001.
- [21] J. S. Liu, R. Chen, and T. Logvinenko, "A theoretical framework for sequential importance sampling and resampling," in *Sequential Monte Carlo in Practice*, A. Doucet, N. Freitas, and N. Gordon, Eds. New York: Springer-Verlag, 2001, pp. 225–246.
- [22] J. S. Liu and R. Chen, "Blind deconvolution via sequential imputations," *J. Amer. Statist. Assoc.*, vol. 90, pp. 567–576, 1995.
- [23] —, "Sequential Monte Carlo methods for dynamic systems," *J. Amer. Statist. Assoc.*, vol. 93, pp. 1032–1044, 1998.
- [24] Y. Liu and S. D. Bolstein, "Identification of frequency nonselective fading channels using decision feedback and adaptive linear prediction," *IEEE Trans. Commun.*, vol. 43, pp. 1484–1492, Feb./Mar./Apr. 1995.
- [25] J. H. Lodge and M. L. Moher, "Maximum likelihood sequence estimation of CPM signals transmitted over Rayleigh flat-fading channels," *IEEE Trans. Commun.*, vol. 38, pp. 787–794, June 1990.
- [26] D. Makrakis, P. T. Mathiopoulos, and D. P. Boursas, "Optimal decoding of coded PSK and QAM signals in correlated fast fading channels and AWGN: A combined envelop, multiple differential and coherent detection approach," *IEEE Trans. Commun.*, vol. 42, pp. 63–75, Jan. 1994.
- [27] M. L. Moher and J. H. Lodge, "TCMP—A modulation and coding strategy for Rician fading channels," *IEEE J. Select. Areas Commun.*, vol. 7, pp. 1347–1355, Dec. 1989.
- [28] M. K. Pitt and N. Shephard, "Filtering via simulation: Auxiliary particle filters," *J. Amer. Statist. Assoc.*, vol. 94, pp. 590–599, 1999.
- [29] R. Raheli, A. Polydoros, and C. Tzou, "Per-survivor processing: A general approach to MLSE in uncertain environments," *IEEE Trans. Commun.*, vol. COM-43, pp. 354–364, Feb./Mar./Apr. 1995.
- [30] 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. Commun.*, vol. 43, pp. 2750–2758, Nov. 1995.



Xiaodong Wang (M'98) 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 engineering from Purdue University, West Lafayette, IN, in 1995, and the Ph.D. degree in electrical engineering from Princeton University, Princeton, NJ, in 1998.

From July 1998 to December 2001, he was an Assistant Professor with the Department of Electrical Engineering, Texas A&M University, College Station. In January 2002, he joined the Department of Electrical Engineering, Columbia University, New York, NY, as an Assistant Professor. His research interests fall in the general areas of computing, signal processing, and communications. He has worked in the areas of digital communications, digital signal processing, parallel and distributed computing, nanoelectronics, and quantum computing. His current research interests include multiuser communications theory and advanced signal processing for wireless communications. He was with AT&T Labs—Research, Red Bank, NJ, during the summer of 1997.

Dr. Wang is a member of the American Association for the Advancement of Science. He received the 1999 NSF CAREER Award and the 2001 IEEE Communications Society and Information Theory Society Joint Paper Award. He currently serves as an Associate Editor for the IEEE TRANSACTIONS ON COMMUNICATIONS and for the IEEE TRANSACTIONS ON SIGNAL PROCESSING.



Rong Chen (M'92) received the B.S. degree in mathematics from Peking University, Beijing, China, in 1985 and the M.S. and Ph.D. degrees in statistics from Carnegie Mellon University, Pittsburgh, PA, in 1987 and 1990, respectively.

He 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, College Station. 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*, the *Journal of Business and Economic Statistics*, *Statistica Sinica*, and *Computational Statistics*.



Dong Guo received the B.S. degree in geophysics and computer science (minor) in 1993 from the Chinese University of Mining and Technology (CUMT), Xuzhou, China, and the M.S. degree in geophysics in 1998 from the Graduate School of Research Institute of Petroleum Exploration and Development (RIPED), Beijing, China, in 1996. He received the Ph.D. degree in applied mathematics from Peking University, Beijing, China, in 1999. He is pursuing a second Ph.D. degree with the Department of Electrical Engineering, Texas A&M

University, College Station.

His research interests are in the area of statistical signal processing and its applications, primarily in digital communications.