Adaptive Joint Detection and Decoding in Flat-Fading Channels via Mixture Kalman Filtering

Rong Chen, Xiaodong Wang, *Member, IEEE*, and Jun S. Liu

***Abstract*—A novel adaptive Bayesian receiver for signal detection and decoding in fading channels with known channel statistics is developed; it is based on the sequential Monte Carlo methodology that recently emerged in the field of statistics. The basic idea is to treat the transmitted signals as "missing data" and to sequentially impute multiple samples of them based on the observed signals. The imputed signal sequences, together with their importance weights, provide a way to approximate the Bayesian estimate of the transmitted signals and the channel states. Adaptive receiver algorithms for both uncoded and convolutionally coded systems are developed. The proposed techniques can easily handle the non-Gaussian ambient channel noise. It is shown through simulations that the proposed sequential Monte Carlo receivers achieve near-bound performance in fading channels for both uncoded and coded systems, without the use of any training/pilot symbols or decision feedback. Moreover, the proposed receiver structure exhibits massive parallelism and is ideally suited for high-speed parallel implementation using the very large scale integration (VLSI) systolic array technology.**

***Index Terms*—Adaptive decoding, adaptive detection, coded system, flat-fading channel, mixture Kalman filter, non-Gaussian noise, sequential Monte Carlo methods.**

I. INTRODUCTION

NARROW-BAND mobile communications for voice and data can be modeled as signaling over frequency-nonselective (flat) Rayleigh fading channels. A considerable amount of research has recently been devoted to signal detection in such channels. Specifically, various techniques for the maximum-likelihood sequence estimation (MLSE) in flat-fading channels have been proposed. The optimal solutions under several fading models are studied in [11], [21], and [22]. The exact implementation of these optimal solutions, however, involves prohibitively time-consuming high-dimensional filtering. A number of suboptimal algorithms have thus been proposed, most of which employ a two-stage receiver 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 [30], [32], decision feedback [11], [14], [20], pilot symbols [3], [7], [16], [29], or a combination of all the above [13]. Other suboptimal solutions to MLSE in flat-fading channels include the method based on a combination of hidden Markov modeling and Kalman filtering [5], [6] and the method based on the expectation–maximization (EM) algorithm [8]. Furthermore, joint channel estimation and decoding techniques are developed in [9], [12] for coded systems based on iterative (turbo) processing.

In this paper, we propose a new *adaptive* receiver technique for signal reception and decoding in flat-fading channels based on a Bayesian formulation of the problem and the sequential Monte Carlo methodology that recently emerged in the field of statistics [19]. The basic idea is to treat the transmitted signals as "missing data" and to sequentially impute multiple samples of them based on the current observation. The importance weight for each of the imputed signal sequences is computed according to its relative ability in predicting the future observation. Then the imputed signal sequences, together with their importance weights, can be used to approximate the Bayesian estimates of the transmitted signals and the fading coefficients of the channel. The novel features of such an approach include the following:

- The algorithm is self-adaptive and no training/pilot symbols or decision feedback are needed.
- The tracking of fading channels and the estimation of data symbols are naturally integrated.
- The ambient channel noise can be either Gaussian or non-Gaussian.
- If the system employs channel coding, the coded signal structure can be easily exploited to substantially improve the accuracy of both channel and data estimation.
- The resulting receiver structure exhibits massive parallelism and is ideally suited for high-speed parallel implementation using very large scale integration (VLSI) systolic array technology.

This paper is organized as follows. In Section II, the communication system under study is described and the Bayesian formulation of the problem is stated. In Section III, some background material on Monte Carlo filtering methods for

Manuscript received October 28, 1999; revised March 4, 2000. This work was supported in part by the Interdisciplinary Research Initiatives Program, Texas A&M University. The work of R. Chen was supported in part by the U.S. National Science Foundation (NSF) under Grants DMS-9626113 and DMS-9982846. The work of X. Wang was supported in part by the NSF under CAREER Grant CCR-9875314. The work of J. S. Liu was supported in part by the NSF under Grant DMS-9803649. The material in this paper was presented in part at the 33rd Annual Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, October 1999; and in part at the IEEE International Symposium on Information Theory, Sorrento, Italy, June 2000.

R. Chen was with the Department of Statistics, Texas A&M University, College Station, TX 77843 USA. He is now with the Department of Information and Decision Sciences, The University of Illinois at Chicago, Chicago, IL 60607 USA.

X. Wang is with the Department of Electrical Engineering, Texas A&M University, College Station, TX 77843 USA (e-mail: wangx@ee.tamu.edu).

J. S. Liu was with the Department of Statistics, Stanford University, Stanford, CA 94305 USA. He is now with the Department of Statistics, Harvard University, Cambridge, MA 02138 USA.

Communicated by M. L. Honig, Associate Editor for Communications.

Publisher Item Identifier S 0018-9448(00)06985-6.

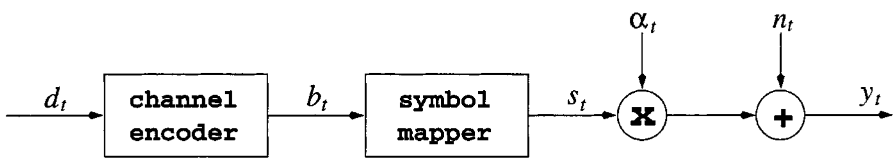

Fig. 1. A coded communication system signaling through a flat-fading channel.

sequential Bayesian inference is provided. In Section IV, we develop an adaptive Bayesian receiver algorithm for concurrent channel and data estimation in fading Gaussian noise channels for uncoded systems. In Section V, techniques for delayed estimation are discussed. In Section VI, we develop adaptive Bayesian sequential decoding methods for convolutionally coded systems in fading Gaussian noise channels. In Section VII, we discuss adaptive Bayesian receivers for fading non-Gaussian noise channels. Simulation results are provided in Section VIII and a brief summary is given in Section IX. Necessary mathematical proofs are contained in the Appendices A–C.

II. System Description

We consider a channel-coded communication system signaling through a flat-fading channel with additive ambient noise. The block diagram of such a system is shown in Fig. 1. The input binary information bits $\{d_t\}$ are encoded using some channel code, resulting in a code bit stream $\{b_t\}$. The code bits are passed to a symbol mapper, yielding complex data symbols $\{s_t\}$, which take values from a finite alphabet set $\mathcal{A} = \{a_1, \ldots, a_{|\mathcal{A}|}\}$. Each symbol is then transmitted through a flat-fading channel whose input–output relationship is given by

$$y_t = \alpha_t s_t + n_t, \qquad t = 0, 1, \ldots \tag{1}$$

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

It is assumed that the additive noise $\{n_t\}$ is a sequence of independent and identically distributed (i.i.d.) zero-mean complex random variables. In this paper we consider two types of noise distributions. In the first type, n_t assumes a complex Gaussian distribution

$$n_t \sim \mathcal{N}_c(0, \sigma^2) \tag{2}$$

whereas in the second type, n_t follows a two-term mixture Gaussian distribution

$$n_t \sim (1-\epsilon)\mathcal{N}_c(0, \sigma_1^2) + \epsilon\mathcal{N}_c(0, \sigma_2^2) \tag{3}$$

where $0 < \epsilon < 1$ and $\sigma_2^2 > \sigma_1^2$. Here the term $\mathcal{N}_c(0, \sigma_1^2)$ represents the nominal ambient noise, and the term $\mathcal{N}_c(0, \sigma_2^2)$ represents an impulsive component. The probability that impulses occur is ϵ. Note that the overall variance of the noise is $(1-\epsilon)\sigma_1^2 + \epsilon\sigma_2^2$. This model serves as an approximation to the more fundamental Middleton Class A noise model [26], [27], [34], and has been used extensively to model physical noise arising in radio and acoustic channels.

It is further 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

$$\{\alpha_t\} = \frac{\Theta(D)}{\Phi(D)}\{u_t\} \tag{4}$$

where D is the back-shift operator $D^k u_t \triangleq u_{t-k}$

$$\Phi(z) \triangleq \phi_r z^r + \ldots + \phi_1 z + 1$$
$$\Theta(z) \triangleq \theta_r z^r + \ldots + \theta_1 z + \theta_0$$

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. In this paper, we assume that the statistical properties of the fading process are known *a priori*. Consequently, the order and the coefficients of the Butterworth filter in (4) are known.

We next write system (1) and (4) in the state-space model form, which is instrumental in developing the adaptive receiver proposed in this paper. Define

$$\{x_t\} \triangleq \Theta^{-1}(D)\{\alpha_t\} \implies \Phi(D)\{x_t\} = \{u_t\}. \tag{5}$$

Denote $\boldsymbol{x}_t \triangleq [x_t, \ldots, x_{t-r+1}]^T$. By (4) we then have

$$\boldsymbol{x}_t = \boldsymbol{F}\boldsymbol{x}_{t-1} + \boldsymbol{g}u_t, \qquad u_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}_c(0, 1) \tag{6}$$

where

$$\boldsymbol{F} \triangleq \begin{pmatrix} -\phi_1 & -\phi_2 & \ldots & -\phi_r & 0 \\ 1 & 0 & \ldots & 0 & 0 \\ 0 & 1 & \ldots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & 1 & 0 \end{pmatrix}, \quad \text{and } \boldsymbol{g} \triangleq \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

Because of (5), the fading coefficient sequence $\{\alpha_t\}$ can be written as

$$\alpha_t = \boldsymbol{h}^H \boldsymbol{x}_t, \qquad \text{where } \boldsymbol{h} \triangleq [\theta_0\, \theta_1\, \ldots\, \theta_r]^H. \tag{7}$$

If the additive noise in (1) is Gaussian, i.e., $n_t \sim \mathcal{N}_c(0, \sigma^2)$, then we have the following state-space model for the system defined by (1) and (4):

$$\boldsymbol{x}_t = \boldsymbol{F}\boldsymbol{x}_{t-1} + \boldsymbol{g}u_t \tag{8}$$
$$y_t = s_t \boldsymbol{h}^H \boldsymbol{x}_t + \sigma v_t \tag{9}$$

where $\{v_t\}$ in (9) is a white complex Gaussian noise sequence with unit variance and independent real and imaginary components.

On the other hand, if the additive noise in (1) is non-Gaussian and is modeled by (3), we introduce an indicator random variable I_t, $t = 0, 1, \ldots$

$$I_t \triangleq \begin{cases} 1, & \text{if } n_t \sim \mathcal{N}_c(0, \sigma_1^2) \\ 2, & \text{if } n_t \sim \mathcal{N}_c(0, \sigma_2^2) \end{cases} \tag{10}$$

with $P(I_t = 1) = \epsilon$ and $P(I_t = 2) = 1 - \epsilon$. Because n_t is an i.i.d. sequence, so is I_t. We then have the state-space signal model for this case given by

$$\boldsymbol{x}_t = \boldsymbol{F}\boldsymbol{x}_{t-1} + \boldsymbol{g}u_t \tag{11}$$

$$y_t = s_t \boldsymbol{h}^H \boldsymbol{x}_t + \sigma_{I_t} v_t. \tag{12}$$

We now look at the problem of online estimation of the symbol s_t and the channel coefficient α_t based on the received signals up to time t, $\{y_i\}_{i=0}^t$. Consider the simple case when the ambient channel noise is Gaussian and the symbols are i.i.d. uniformly *a priori*, i.e., $p(s_i) = 1/|\mathcal{A}|$. Then the problem becomes one of making Bayesian inference with respect to the posterior distribution

$$\begin{aligned} &p(x_0, \ldots, x_t, s_0, \ldots, s_t | y_0, \ldots, y_t) \\ &\quad\propto \prod_{j=1}^{t} p(x_j|\boldsymbol{x}_{j-1}) p(s_j) p(y_j|\boldsymbol{x}_j, s_j) \\ &\quad\propto \prod_{j=1}^{t} \exp\left(-\|x_j + \sum_{i=1}^{r} \phi_i x_{j-i}\|^2 - \frac{1}{\sigma^2} \left\| y_j - s_j \boldsymbol{h}^T \boldsymbol{x}_j \right\|^2 \right), \\ &\qquad\qquad t = 0, 1, \ldots. \end{aligned} \tag{13}$$

For example, an online symbol estimation can be obtained from the marginal posterior distribution $p(s_t|y_0, \ldots, y_t)$, and an online channel state estimation can be obtained from the marginal posterior distribution $p(\boldsymbol{x}_t|y_0, \ldots, y_t)$. Although the joint distribution (13) can be written out explicitly up to a normalizing constant, the computation of the corresponding marginal distributions involves very-high-dimensional integration and is infeasible in practice. Our approach to this problem is the sequential Monte Carlo filtering technique.

III. Sequential Monte Carlo Methods

In this section, the general framework of sequential Monte Carlo methods for updating a dynamic system is described. Of particular interest is the mixture Kalman filtering technique described in [4], which will be used for designing adaptive Bayesian receivers in fading channels.

A. Sequential Monte Carlo Filtering

Consider the following dynamic system modeled in a state-space form as

$$\begin{array}{rl} \text{state equation} & \boldsymbol{z}_t = f_t(\boldsymbol{z}_{t-1}, \boldsymbol{u}_t) \\ \text{observation equation} & \boldsymbol{y}_t = g_t(\boldsymbol{z}_t, \boldsymbol{v}_t) \end{array} \tag{14}$$

where $\boldsymbol{z}_t$, $\boldsymbol{y}_t$, $\boldsymbol{u}_t$, and $\boldsymbol{v}_t$ are, respectively, the state variable, the observation, the state noise, and the observation noise at time t. They can be either scalars or vectors. In the communication system described in the previous section (e.g. (13)), the state variable $\boldsymbol{z}_t$ corresponds to $(\boldsymbol{x}_t, s_t)$, representing both the unobserved symbol and the unknown fading channel at time t.

Let $\boldsymbol{Z}_t=(\boldsymbol{z}_0, \boldsymbol{z}_1, \ldots, \boldsymbol{z}_t)$ and let $\boldsymbol{Y}_t = (\boldsymbol{y}_0, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_t)$. Suppose an online inference of $\boldsymbol{Z}_t$ is of interest; that is, at current time t we wish to make a timely estimate of a function of the state variable $\boldsymbol{Z}_t$, say $h(\boldsymbol{Z}_t)$, based on the currently available observation, $\boldsymbol{Y}_t$. With the Bayes theorem, we realize that the optimal solution to this problem is

$$E\{h(\boldsymbol{Z}_t)|\boldsymbol{Y}_t\} = \int h(\boldsymbol{Z}_t) p(\boldsymbol{Z}_t|\boldsymbol{Y}_t) \mathrm{d}\boldsymbol{Z}_t.$$

In most cases, an exact evaluation of this expectation is analytically intractable because of the complexity of such a dynamic system. Monte Carlo methods provide us with a viable alternative to the required computation. Specifically, if we can draw m random samples $\{\boldsymbol{Z}_t^{(j)}\}_{j=1}^m$ from the distribution $p(\boldsymbol{Z}_t|\boldsymbol{Y}_t)$, then we can approximate $E\{h(\boldsymbol{Z}_t)|\boldsymbol{Y}_t\}$ by

$$E\{h(\boldsymbol{Z}_t)|\boldsymbol{Y}_t\} \cong \frac{1}{m} \sum_{j=1}^{m} h(\boldsymbol{Z}_t^{(j)}). \tag{15}$$

Very often direct simulation from $p(\boldsymbol{Z}_t|\boldsymbol{Y}_t)$ is not feasible, but drawing samples from some *trial* distribution is easy. In this case, we can use the idea of *importance sampling*. Suppose a set of random samples $\{\boldsymbol{Z}_t^{(j)}\}_{j=1}^m$ is generated from the trial distribution $q(\boldsymbol{Z}_t|\boldsymbol{Y}_t)$. By associating the weight

$$w_t^{(j)} = \frac{p(\boldsymbol{Z}_t^{(j)}|\boldsymbol{Y}_t)}{q(\boldsymbol{Z}_t^{(j)}|\boldsymbol{Y}_t)} \tag{16}$$

to the sample $\boldsymbol{Z}_t^{(j)}$, we can approximate the quantity of interest, $E\{h(\boldsymbol{Z}_t)|\boldsymbol{Y}_t\}$, as

$$E_p\{h(\boldsymbol{Z}_t)|\boldsymbol{Y}_t\} \cong \frac{1}{W_t} \sum_{j=1}^{m} h(\boldsymbol{Z}_t^{(j)}) w_t^{(j)} \tag{17}$$

where $W_t = \sum_{j=1}^m w_t^{(j)}$. The pair $(\boldsymbol{Z}_t^{(j)}, w_t^{(j)})$, $j = 1, \ldots, m$, is called a *properly weighted sample* with respect to distribution $p(\boldsymbol{Z}_t|\boldsymbol{Y}_t)$. A trivial but important observation is that the $\boldsymbol{z}_t^{(j)}$ (one of the components of $\boldsymbol{Z}_t^{(j)}$) is also properly weighted by the $w_t^{(j)}$ with respect to the marginal distribution $p(\boldsymbol{z}_t|\boldsymbol{Y}_t)$.

Another possible estimate of $E\{h(\boldsymbol{Z}_t)|\boldsymbol{Y}_t\}$ is

$$\tilde{h} = \frac{1}{m} \sum_{j=1}^{m} h\left(\boldsymbol{Z}_t^{(j)}\right) w_t^{(j)}. \tag{18}$$

The main reasons for preferring the ratio estimate (17) to the unbiased estimate (18) in an importance sampling framework are that a) estimate (17) usually has a smaller mean-squared error than $\tilde{h}$ in (18); and b) the normalizing constants of both the trial and the target distributions are not required in using (17) (where these constants are canceled out); in such cases, the weights $w_t^{(j)}$ are evaluated only up to a multiplicative constant. For example, the target distribution $p(\boldsymbol{Z}_t|\boldsymbol{Y}_t)$ in a typical dynamic system (and many Bayesian models) can be evaluated easily up to a normalizing constant (e.g., the likelihood multiplied by a prior dis-

TABLE I

A SEQUENTIAL MONTE CARLO ALGORITHM FOR PROPAGATING A SET OF PROPERLY WEIGHTED SAMPLES FROM TIME $(t-1)$ TO TIME t. NOTE THAT IN MOST APPLICATIONS WE ARE ONLY ABLE TO EVALUATE $p(\boldsymbol{Z}_t|\boldsymbol{Y}_t) \propto p(\boldsymbol{Z}_t, \boldsymbol{Y}_t)$ UP TO A NORMALIZING CONSTANT, WHICH IS SUFFICIENT FOR USING (17) IN MONTE CARLO ESTIMATION

FOR $j = 1, \cdots, m$ DO

1. Draw a sample $\boldsymbol{z}_t^{(j)}$ from a trial distribution $q(\boldsymbol{z}_t|\boldsymbol{Z}_{t-1}^{(j)}, \boldsymbol{Y}_t)$ and let $\boldsymbol{Z}_t^{(j)} = (\boldsymbol{Z}_{t-1}^{(j)}, \boldsymbol{z}_t^{(j)})$;
2. Compute the importance weight $w_t^{(j)} = w_{t-1}^{(j)} \cdot p(\boldsymbol{Z}_t^{(j)}|\boldsymbol{Y}_t) / \left[p(\boldsymbol{Z}_{t-1}^{(j)}|\boldsymbol{Y}_{t-1})\, q(\boldsymbol{z}_t^{(j)}|\boldsymbol{Z}_{t-1}^{(j)}, \boldsymbol{Y}_t)\right]$

END

tribution), whereas sampling from the distribution directly and evaluating the normalizing constant analytically are impossible.

To implement Monte Carlo techniques for a dynamic system, a set of random samples properly weighted with respect to $p(\boldsymbol{Z}_t|\boldsymbol{Y}_t)$ is needed for any time t. Because the state equation in system (14) possesses a Markovian structure, we can implement a recursive importance sampling strategy, which is the basis of all sequential Monte Carlo techniques [19]. Suppose a set of properly weighted samples $\{(\boldsymbol{Z}_{t-1}^{(j)}, w_{t-1}^{(j)})\}_{j=1}^m$ (with respect to $p(\boldsymbol{Z}_{t-1}|\boldsymbol{Y}_{t-1})$) is given at time $(t-1)$. A *Monte Carlo filter* (MCF) generates from the set a new one, $\{\boldsymbol{Z}_t^{(j)}, w_t^{(j)}\}_{j=1}^m$, which is properly weighted at time t with respect to $p(\boldsymbol{Z}_t|\boldsymbol{Y}_t)$. The algorithm is described in Table I.

The algorithm is initialized by drawing a set of i.i.d. samples $\boldsymbol{z}_0^{(1)}, \ldots, \boldsymbol{z}_0^{(m)}$ from $p(\boldsymbol{z}_0|\boldsymbol{y}_0)$. When $\boldsymbol{y}_0$ represents the "null" information, $p(\boldsymbol{z}_0|\boldsymbol{y}_0)$ corresponds to the prior distribution of $\boldsymbol{z}_0$. We show in Appendix A that the weighted samples generated by this algorithm satisfy

$$E\left\{h\left(\boldsymbol{Z}_t^{(j)}\right) w_t^{(j)}\right\} = E\{h(\boldsymbol{Z}_t)|\boldsymbol{Y}_t\} \tag{19}$$

$$E\left\{w_t^{(j)}\right\} = 1. \tag{20}$$

Hence, by the law of large numbers

$$\frac{1}{W_t}\sum_{j=1}^m h\left(\boldsymbol{Z}_t^{(j)}\right) w_t^{(j)} = \frac{\sum_{j=1}^m h\left(\boldsymbol{Z}_t^{(j)}\right)/m}{W_t/m} \xrightarrow{\text{a.s.}} E\{h(\boldsymbol{Z}_t)|\boldsymbol{Y}_t\}, \quad \text{as } m \longrightarrow \infty. \tag{21}$$

There are a few important issues regarding the design and implementation of a sequential MCF, such as the choice of the trial distribution $q(\cdot)$ and the use of *resampling* (cf. Section IV-B). Specifically, a useful choice of the trial distribution $q(\boldsymbol{z}_t|\boldsymbol{Z}_{t-1}^{(j)}, \boldsymbol{Y}_t)$ for the state-space model (14) is of the form

$$\begin{aligned} q&\left(\boldsymbol{z}_t \,\middle|\, \boldsymbol{Z}_{t-1}^{(j)}, \boldsymbol{Y}_t\right) \\ &= p\left(\boldsymbol{z}_t \,\middle|\, \boldsymbol{Z}_{t-1}^{(j)}, \boldsymbol{Y}_t\right) \\ &= p\left(\boldsymbol{y}_t \,\middle|\, \boldsymbol{z}_t, \boldsymbol{Z}_{t-1}^{(j)}, \boldsymbol{Y}_{t-1}\right) \\ &\quad \cdot p\left(\boldsymbol{z}_t \,\middle|\, \boldsymbol{Z}_{t-1}^{(j)}, \boldsymbol{Y}_{t-1}\right) \Big/ p\left(\boldsymbol{y}_t \,\middle|\, \boldsymbol{Z}_{t-1}^{(j)}, \boldsymbol{Y}_{t-1}\right) \\ &= \frac{p(\boldsymbol{y}_t|\boldsymbol{z}_t)\, p\left(\boldsymbol{z}_t \,\middle|\, \boldsymbol{z}_{t-1}^{(j)}\right)}{p(\boldsymbol{y}_t|\boldsymbol{z}_{t-1}^{(j)})} \end{aligned} \tag{22}$$

where in (22) we used the fact that

$$p(\boldsymbol{y}_t|\boldsymbol{z}_t, \boldsymbol{Z}_{t-1}^{(j)}\boldsymbol{Y}_{t-1}) = p(\boldsymbol{y}_t|\boldsymbol{z}_t)$$

and

$$p(\boldsymbol{z}_t|\boldsymbol{Z}_{t-1}^{(j)}, \boldsymbol{Y}_{t-1}) = p(\boldsymbol{z}_t|\boldsymbol{z}_{t-1}^{(j)})$$

both following directly from the state-space model (14). For this trial distribution, the importance weight is updated according to

$$\begin{aligned} w_t^{(j)} &= w_{t-1}^{(j)} \cdot \frac{p\left(\boldsymbol{Z}_t^{(j)} \,\middle|\, \boldsymbol{Y}_t\right)}{p\left(\boldsymbol{Z}_{t-1}^{(j)} \,\middle|\, \boldsymbol{Y}_{t-1}\right) p\left(\boldsymbol{z}_t^{(j)} \,\middle|\, \boldsymbol{Z}_{t-1}^{(j)}, \boldsymbol{Y}_t\right)} \\ &= w_{t-1}^{(j)} \cdot \frac{p\left(\boldsymbol{Z}_{t-1}^{(j)} \,\middle|\, \boldsymbol{Y}_t\right)}{p\left(\boldsymbol{Z}_{t-1}^{(j)} \,\middle|\, \boldsymbol{Y}_{t-1}\right)} \end{aligned} \tag{23}$$

$$\begin{aligned} &= w_{t-1}^{(j)} \cdot \frac{p\left(\boldsymbol{y}_t, \boldsymbol{Z}_{t-1}^{(j)}, \boldsymbol{Y}_{t-1}\right) p(\boldsymbol{Y}_{t-1})}{p\left(\boldsymbol{Z}_{t-1}^{(j)}, \boldsymbol{Y}_{t-1}\right) p(\boldsymbol{Y}_t)} \\ &\propto w_{t-1}^{(j)} \cdot p\left(\boldsymbol{y}_t \,\middle|\, \boldsymbol{Z}_{t-1}^{(j)}, \boldsymbol{Y}_{t-1}\right) \\ &= w_{t-1}^{(j)} \cdot p\left(\boldsymbol{y}_t \,\middle|\, \boldsymbol{z}_{t-1}^{(j)}\right) \end{aligned} \tag{24}$$

where (23) follows from the fact that

$$p\left(\boldsymbol{Z}_t^{(j)} \,\middle|\, \boldsymbol{Y}_t\right) = p\left(\boldsymbol{Z}_{t-1}^{(j)} \,\middle|\, \boldsymbol{Y}_t\right) p\left(\boldsymbol{z}_t^{(j)} \,\middle|\, \boldsymbol{Z}_{t-1}^{(j)}, \boldsymbol{Y}_t\right)$$

and the last equality is due to the conditional independence property of the state-space model (14). See [19] for the general sequential MCF framework and a detailed discussion on various implementation issues.

B. The Mixture Kalman Filter

Many dynamic system models, including the flat-fading channel models (8), (9) and (11), (12) belong to the class of conditional dynamic linear models (CDLM) of the form

$$\begin{aligned} \boldsymbol{x}_t &= F_{\lambda_t}\boldsymbol{x}_{t-1} + G_{\lambda_t}\boldsymbol{u}_t \\ \boldsymbol{y}_t &= H_{\lambda_t}\boldsymbol{x}_t + K_{\lambda_t}\boldsymbol{v}_t \end{aligned} \tag{25}$$

where $\boldsymbol{u}_t \sim \mathcal{N}_c(0, I)$, $\boldsymbol{v}_t \sim \mathcal{N}_c(0, I)$ (here I denotes an identity matrix), and λ_t is a random indicator variable. The matrices F_{λ_t}, G_{λ_t}, H_{λ_t}, and K_{λ_t} are known given λ_t. In this model, the "state variable" $\boldsymbol{z}_t$ corresponds to $(\boldsymbol{x}_t, \lambda_t)$.

We observe that for a given trajectory of the indicator λ_t in a CDLM, the system is both linear and Gaussian, for which the Kalman filter provides a complete statistical characterization of the system dynamics. Recently, a novel sequential Monte Carlo method, the mixture Kalman filter (MKF), was proposed in [4] for online filtering and prediction of CDLMs; it exploits the conditional Gaussian property and utilizes a marginalization operation to improve the algorithmic efficiency. Instead of dealing

TABLE II
MIXTURE KALMAN FILTERING ALGORITHM FOR UPDATING A SET OF PROPERLY WEIGHTED SAMPLES FROM $(t-1)$ TO t INITIALIZED BY AN i.i.d. SAMPLE, $\{\lambda_0^{(j)}\}_{j=1}^m$, DRAWN FROM $p(\lambda_0|\boldsymbol{y}_0) \propto \int p(\boldsymbol{y}_0|\boldsymbol{x}_0, \lambda_0)p(\boldsymbol{x}_0, \lambda_0)d\boldsymbol{x}_0$

FOR $j = 1, \cdots, m$ DO
1. Draw a sample $\lambda_t^{(j)}$ from a trial distribution $q(\lambda_t|\boldsymbol{\Lambda}_{t-1}^{(j)}, \kappa_{t-1}^{(j)}, \boldsymbol{Y}_t)$
2. Run a one-step Kalman filter based on $\lambda_t^{(j)}$, $\kappa_{t-1}^{(j)}$, and $\boldsymbol{y}_t$ to obtain $\kappa_t^{(j)}$;
3. Compute the weight $w_t^{(j)} = w_{t-1}^{(j)} \cdot p(\boldsymbol{\Lambda}_{t-1}^{(j)}, \lambda_t^{(j)}|\boldsymbol{Y}_t) / \left[p(\boldsymbol{\Lambda}_{t-1}^{(j)}|\boldsymbol{Y}_{t-1})\, q(\lambda_t^{(j)}|\boldsymbol{\Lambda}_{t-1}^{(j)}, \kappa_{t-1}^{(j)}, \boldsymbol{Y}_t)\right]$
END

with both $\boldsymbol{x}_t$ 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 MCF method described in Section III-A, the MKF is substantially more efficient (e.g., giving more accurate results with the same computing resources). However, the MKF often needs more "brain power" for its proper implementation, as the required formulas are more complicated. Additionally, the MKF requires the CDLM structure which may not be applicable to other problems.

Let $\boldsymbol{Y}_t = (\boldsymbol{y}_0, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_t)$ and let $\boldsymbol{\Lambda}_t = (\lambda_0, \lambda_1, \ldots, \lambda_t)$. By recursively generating a set of properly weighted random samples $\{(\boldsymbol{\Lambda}_t^{(j)}, w_t^{(j)})\}_{j=1}^m$ to represent $p(\boldsymbol{\Lambda}_t|\boldsymbol{Y}_t)$, the MKF approximates the target distribution $p(\boldsymbol{x}_t|\boldsymbol{Y}_t)$ by a random mixture of Gaussian distributions $\sum_{j=1}^m w_t^{(j)}\mathcal{N}_c(\mu_t^{(j)}, \Sigma_t^{(j)})$, where $\kappa_t^{(j)} \triangleq [\mu_t^{(j)}, \Sigma_t^{(j)}]$ is obtained by implementing a Kalman filter for the given indicator trajectory $\boldsymbol{\Lambda}_t^{(j)}$. 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 $(t-1)$. The algorithm is given in Table II. The application of this algorithm to the problem of detection/estimation in fading channels is described in the next section where the correctness of the algorithm is also proved. The MKF can be extended to handle the so-called partial CDLM, where the state variable has a linear component and a nonlinear component. See [4] for a detailed treatment of the MKF and the extended MKF.

IV. ADAPTIVE RECEIVER IN FADING GAUSSIAN NOISE CHANNELS—UNCODED CASE

A. MKF-Based Sequential Monte Carlo Receiver

Consider the flat-fading channel with additive Gaussian noise, given by (8) and (9). Denote $\boldsymbol{Y}_t \triangleq (y_0, \ldots, y_t)$ and $\boldsymbol{S}_t \triangleq (s_0, \ldots, s_t)$. We first consider the case of uncoded system, where the transmitted symbols are assumed to be independent, i.e.,

$$P(s_t = a_i|\boldsymbol{S}_{t-1}) = P(s_t = a_i), \qquad a_i \in \mathcal{A}. \tag{26}$$

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., $P(s_t = a_i) = \frac{1}{|\mathcal{A}|}$ for $i = 1, \ldots, |\mathcal{A}|$. We are interested in estimating the symbol s_t and the channel coefficient $\alpha_t = \boldsymbol{h}^H\boldsymbol{x}_t$ at time t based on the observation $\boldsymbol{Y}_t$. The Bayes solution to this problem requires the posterior distribution

$$p(\boldsymbol{x}_t, s_t|\boldsymbol{Y}_t) = \int p(\boldsymbol{x}_t|\boldsymbol{S}_t, \boldsymbol{Y}_t)p(\boldsymbol{S}_t|\boldsymbol{Y}_t)\, d\boldsymbol{S}_{t-1}. \tag{27}$$

Note that with a given $\boldsymbol{S}_t$, the state-space model (8), (9) becomes a linear Gaussian system. Hence

$$p(\boldsymbol{x}_t|\boldsymbol{S}_t, \boldsymbol{Y}_t) \sim \mathcal{N}_c\left(\boldsymbol{\mu}_t(\boldsymbol{S}_t), \boldsymbol{\Sigma}_t(\boldsymbol{S}_t)\right) \tag{28}$$

where the mean $\boldsymbol{\mu}_t(\boldsymbol{S}_t)$ and covariance matrix $\boldsymbol{\Sigma}_t(\boldsymbol{S}_t)$ can be obtained by a Kalman filter with the given $\boldsymbol{S}_t$.

In order to implement the MKF, we need to obtain a set of Monte Carlo samples of the transmitted symbols $\{(\boldsymbol{S}_t^{(j)}, w_t^{(j)})\}_{j=1}^m$ properly weighted with respect to the distribution $p(\boldsymbol{S}_t|\boldsymbol{Y}_t)$. Then for any integrable function $h(\boldsymbol{x}_t, s_t)$, we can approximate the quantity of interest $E\{h(\boldsymbol{x}_t, s_t)|\boldsymbol{Y}_t\}$ as follows:

$$\begin{aligned} & E\left\{h(\boldsymbol{x}_t, s_t)|\boldsymbol{Y}_t\right\} \\ &= \iint h(\boldsymbol{x}_t, s_t)p(\boldsymbol{x}_t, s_t|\boldsymbol{Y}_t)\, d\boldsymbol{x}_t\, ds_t \\ &= \iint h(\boldsymbol{x}_t, s_t)\; p(\boldsymbol{x}_t|\boldsymbol{S}_t, \boldsymbol{Y}_t)p(\boldsymbol{S}_t|\boldsymbol{Y}_t)\, d\boldsymbol{x}_t\, d\boldsymbol{S}_t && (29) \\ &= \int \underbrace{\left[\int h(\boldsymbol{x}, s_t)\phi\left(\boldsymbol{x}; \boldsymbol{\mu}_t(\boldsymbol{S}_t), \boldsymbol{\Sigma}_t(\boldsymbol{S}_t)\right)\, d\boldsymbol{x}\right]}_{\xi(\boldsymbol{S}_t)} p(\boldsymbol{S}_t|\boldsymbol{Y}_t)\, d\boldsymbol{S}_t && (30) \\ &\cong \frac{1}{W_t}\sum_{j=1}^m \xi\left(\boldsymbol{S}_t^{(j)}\right) w_t^{(j)} && (31) \end{aligned}$$

where $W_t = \sum_{j=1}^m w_t^{(j)}$, (29) follows from (27), (30) follows from (28), and in (30), $\phi(\cdot; \mu, \Sigma)$ denotes a complex Gaussian density function with mean μ and covariance matrix Σ. In particular, the MMSE channel estimate is given by

$$\begin{aligned} E\{\alpha_t|\boldsymbol{Y}_t\} &= \boldsymbol{h}^H E\{\boldsymbol{x}_t|\boldsymbol{Y}_t\} \\ &\cong \frac{1}{W_t}\boldsymbol{h}^H\left[\sum_{j=1}^m \boldsymbol{\mu}_t(\boldsymbol{S}_t^{(j)})w_t^{(j)}\right]. \end{aligned} \tag{32}$$

In other words, we can let $h(\boldsymbol{x}_t, s_t) \triangleq \boldsymbol{h}^H\boldsymbol{x}_t$, implying that $\xi(\boldsymbol{S}_t) \triangleq \boldsymbol{h}^H\boldsymbol{\mu}_t(\boldsymbol{S}_t)$ in (30). Moreover, the *a posteriori* symbol probability can be estimated as

$$\begin{aligned} & P(s_t = a_i|\boldsymbol{Y}_t) \\ &\quad = E\{1(s_t = a_i)|\boldsymbol{Y}_t\} \\ &\quad \cong \frac{1}{W_t}\sum_{j=1}^m 1(s_t^{(j)} = a_i)w_t^{(j)}, \qquad i = 1, \ldots, |\mathcal{A}| \end{aligned} \tag{33}$$

where $1(\cdot)$ is an indicator function such that $1(s_t = a_i) = 1$ if $s_t = a_i$ and $1(s_t = a_i) = 0$ otherwise. This corresponds to having $h(\boldsymbol{x}_t, s_t) \triangleq 1(s_t = a_i)$ and $\xi(\boldsymbol{S}_t) \triangleq 1(s_t = a_i)$.

Note that a hard decision on the symbol s_t is obtained by

$$\begin{aligned}\hat{s}_t &= \arg\max_{a_i \in \mathcal{A}} P(s_t = a_i | \boldsymbol{Y}_t) \\ &\cong \arg\max_{a_i \in \mathcal{A}} \sum_{j=1}^{m} 1\left(s_t^{(j)} = a_i\right) w_t^{(j)}. \end{aligned} \tag{34}$$

When M-ary phase-shift keying (MPSK) signals are transmitted, i.e.,

$$a_i = \exp\left(\jmath \frac{2\pi i}{|\mathcal{A}|}\right), \qquad \text{for } i = 0, \ldots, |\mathcal{A}| - 1$$

where $\jmath = \sqrt{-1}$, the estimated symbol $\hat{s}_t$ may have a phase ambiguity. For instance, for binary phase-shift keying (BPSK) signals, $s_t \in \{+1, -1\}$. It is easily seen from (1) that if both the symbol sequence $\{s_t\}$ and the channel value sequence $\{\alpha_t\}$ are phase-shifted by π (resulting in $\{-s_t\}$ and $\{-\alpha_t\}$, respectively), no change is incurred on the observed signal $\{y_t\}$. Alternatively, in the state-space model (8), (9), a phase shift of π on both the symbol sequence $\{s_t\}$ and the state sequence $\{\boldsymbol{x}_t\}$ yields the same model for the observations. Hence such a phase ambiguity necessitates the use of differential encoding and decoding.

Hereafter, we let $\boldsymbol{\mu}_t^{(j)} \triangleq \boldsymbol{\mu}_t(\boldsymbol{S}_t^{(j)})$, $\boldsymbol{\Sigma}_t^{(j)} \triangleq \boldsymbol{\Sigma}_t(\boldsymbol{S}_t^{(j)})$, and $\kappa_t^{(j)} \triangleq [\boldsymbol{\mu}_t^{(j)}, \boldsymbol{\Sigma}_t^{(j)}]$. By applying the techniques outlined in Section III to the flat-fading channel system, we describe a recursive procedure for generating properly weighted Monte Carlo samples $\{(\boldsymbol{S}_t^{(j)}, \kappa_t^{(j)}, w_t^{(j)})\}_{j=1}^m$.

1) *Initialization:* Each Kalman filter is initialized as $\kappa_0^{(j)} = (\boldsymbol{\mu}_0^{(j)}, \boldsymbol{\Sigma}_0^{(j)})$, with $\boldsymbol{\mu}_0^{(j)} = \boldsymbol{0}$, $\boldsymbol{\Sigma}_0^{(j)} = 2\boldsymbol{\Sigma}$, $j = 1, \ldots, m$, where $\boldsymbol{\Sigma}$ is the stationary covariance of $\boldsymbol{x}_t$ and is computed analytically from (6). (The factor 2 is to accommodate the initial uncertainty). All importance weights are initialized as $w_0^{(j)} = 1$, $j = 1, \ldots, m$. Since the data symbols are assumed to be independent, initial symbols are not needed.

   Based on the state-space model (8), (9), the following steps are implemented at time t to update each weighted sample. For $j = 1, \ldots, m$,

2) *Compute the one-step predictive update of each Kalman filter* $\kappa_{t-1}^{(j)}$

$$\boldsymbol{K}_t^{(j)} = \boldsymbol{F}\boldsymbol{\Sigma}_{t-1}^{(j)}\boldsymbol{F}^H + \boldsymbol{g}\boldsymbol{g}^H \tag{35}$$

$$\gamma_t^{(j)} = \boldsymbol{h}^H \boldsymbol{K}_t^{(j)} \boldsymbol{h} + \sigma^2 \tag{36}$$

$$\eta_t^{(j)} = \boldsymbol{h}^H \boldsymbol{F} \boldsymbol{\mu}_{t-1}^{(j)}. \tag{37}$$

3) *Compute the trial sampling density:* For each $a_i \in \mathcal{A}$, compute

$$\begin{aligned}\rho_{t,i}^{(j)} &\triangleq P\left(s_t = a_i \,\middle|\, \boldsymbol{S}_{t-1}^{(j)}, \boldsymbol{Y}_t\right) \\ &\propto p\left(y_t, \boldsymbol{Y}_{t-1}, s_t = a_i, \boldsymbol{S}_{t-1}^{(j)}\right) \\ &= p\left(y_t \,\middle|\, s_t = a_i, \boldsymbol{S}_{t-1}^{(j)}, \boldsymbol{Y}_{t-1}\right) \\ &\quad \cdot P\left(s_t = a_i \,\middle|\, \boldsymbol{S}_{t-1}^{(j)}, \boldsymbol{Y}_{t-1}\right) \\ &= p\left(y_t \,\middle|\, s_t = a_i, \boldsymbol{S}_{t-1}^{(j)}, \boldsymbol{Y}_{t-1}\right) P(s_t = a_i) \end{aligned} \tag{38}$$

   where (38) holds because s_t is independent of $\boldsymbol{S}_{t-1}$ and $\boldsymbol{Y}_{t-1}$. Furthermore, we observe that

$$p\left(y_t \,\middle|\, s_t = a_i, \boldsymbol{S}_{t-1}^{(j)}, \boldsymbol{Y}_{t-1}\right) \sim \mathcal{N}_c\left(a_i \eta_t^{(j)}, \gamma_t^{(j)}\right). \tag{39}$$

4) *Impute the symbol* s_t: Draw $s_t^{(j)}$ from the set $\mathcal{A}$ with probability

$$P\left(s_t^{(j)} = a_i\right) \propto \rho_{t,i}^{(j)}, \qquad a_i \in \mathcal{A}. \tag{40}$$

   Append $s_t^{(j)}$ to $\boldsymbol{S}_{t-1}^{(j)}$ and obtain $\boldsymbol{S}_t^{(j)}$.

5) *Compute the importance weight:*

$$\begin{aligned} w_t^{(j)} &= w_{t-1}^{(j)} \cdot p\left(y_t \,\middle|\, \boldsymbol{S}_{t-1}^{(j)}, \boldsymbol{Y}_{t-1}\right) \\ &= w_{t-1}^{(j)} \cdot \sum_{a_i \in \mathcal{A}} p\left(y_t \,\middle|\, s_t = a_i, \boldsymbol{S}_{t-1}^{(j)}, \boldsymbol{Y}_{t-1}\right) P(s_t = a_i) \\ &\propto w_{t-1}^{(j)} \cdot \sum_{a_i \in \mathcal{A}} \rho_{t,i}^{(j)} \end{aligned} \tag{41}$$

   where (41) follows from (38).

6) *Compute the one-step filtering update of the Kalman filter* $\kappa_{t-1}^{(j)}$: Based on the imputed symbol $s_t^{(j)}$ and the observation y_t, complete the Kalman filter update to obtain $\kappa_t^{(j)} = [\boldsymbol{\mu}_t^{(j)}, \boldsymbol{\Sigma}_t^{(j)}]$, as follows:

$$\boldsymbol{\mu}_t^{(j)} = \boldsymbol{F}\boldsymbol{\mu}_{t-1}^{(j)} + \frac{1}{\gamma_t^{(j)}}\left(y_t - s_t^{(j)}\eta_t^{(j)}\right)\boldsymbol{K}_t^{(j)}\boldsymbol{h} \tag{42}$$

$$\boldsymbol{\Sigma}_t^{(j)} = \boldsymbol{K}_t^{(j)} - \frac{1}{\gamma_t^{(j)}}\boldsymbol{K}_t^{(j)}\boldsymbol{h}\boldsymbol{h}^H\boldsymbol{K}_t^{(j)}. \tag{43}$$

It is shown in Appendix B that the samples $\{(\boldsymbol{S}_t^{(j)}, \kappa_t^{(j)}, w_t^{(j)})\}_{j=1}^m$ drawn by the above procedure are indeed properly weighted with respect to $p(\boldsymbol{S}_t | \boldsymbol{Y}_t)$ provided that $\{(\boldsymbol{S}_{t-1}^{(j)}, \kappa_{t-1}^{(j)}, w_{t-1}^{(j)})\}_{j=1}^m$ are proper at time $(t-1)$. The above algorithm is depicted in Fig. 2. It is seen that at any time t, the only quantities that need to be stored are $\{\kappa_t^{(j)}, w_t^{(j)}\}_{j=1}^m$. At each time t, the dominant computation in this receiver involves the m one-step Kalman filter updates. Since the m samplers operate independently and in parallel, such a sequential Monte Carlo receiver is well suited for massively parallel implementation using the VLSI systolic array technology [17].

B. Resampling Procedures

The importance sampling weight $w_t^{(j)}$ measures the "quality" of the corresponding imputed signal sequence $\boldsymbol{S}_t^{(j)}$. 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. This can be detected by observing a large *coefficient of variation* in the importance weight. Suppose $\{w_t^{(j)}\}_{j=1}^m$ is a sequence of importance weights. Then the coefficient of variation, υ_t is defined as

$$\upsilon_t^2 = \frac{\sum_{j=1}^m \left(w_t^{(j)} - \overline{w}_t\right)^2 / m}{\overline{w}_t^2} = \frac{1}{m}\sum_{j=1}^m \left(\frac{w_t^{(j)}}{\overline{w}_t} - 1\right)^2 \tag{44}$$

where $\overline{w}_t = \sum_{j=1}^m w_t^{(j)} / m$. Note that if the samples are drawn exactly from the target distribution, then all the weights are

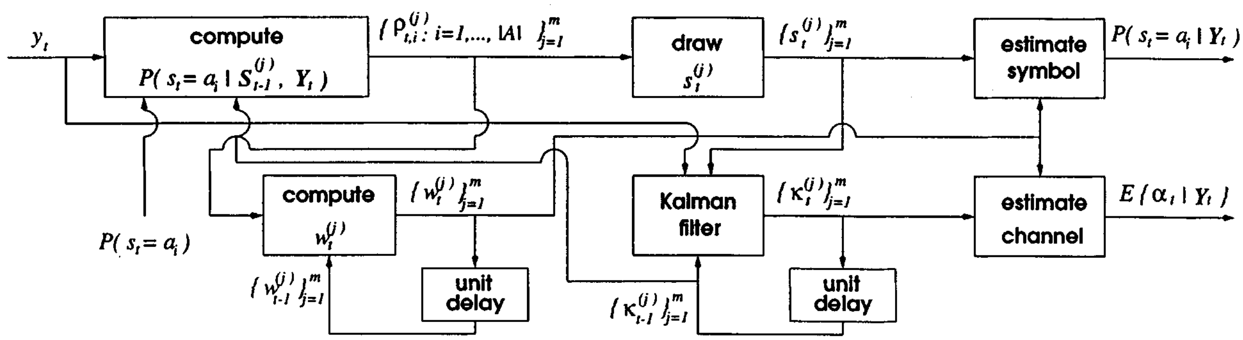

Fig. 2. An adaptive Bayesian receiver in flat-fading Gaussian noise channels based on mixture Kalman filtering.

equal, implying that $\upsilon_t = 0$. It is shown in [15] that the importance weights resulting from a sequential Monte Carlo filter form a martingale sequence. As more and more data are processed, the coefficient of variation of the weights increases; that is, the number of ineffective samples rapidly increases.

A useful method for reducing ineffective samples and enhancing effective ones is *resampling*, which was suggested in [10], [18] under the MCF setting. Roughly speaking, resampling allows those "bad" samples (with small importance weights) to be discarded and those "good" ones (with large importance weights) to replicate so as to accommodate the dynamic change of the system. Specifically, let $\{(\boldsymbol{S}_t^{(j)}, \kappa_t^{(j)}, w_t^{(j)})\}_{j=1}^m$ be the original properly weighted samples at time t. A *residual resampling* strategy forms a new set of weighted samples $\{(\tilde{\boldsymbol{S}}_t^{(j)}, \tilde{\kappa}_t^{(j)}, \tilde{w}_t^{(j)})\}_{j=1}^m$ according to the following algorithm (assume that $\sum_{j=1}^m w_t^{(j)} = m$):

1) For $j = 1, \ldots, m$, retain $k_j = \lfloor w_t^{(j)} \rfloor$ copies of the sample $(\boldsymbol{S}_t^{(j)}, \kappa_t^{(j)})$. Denote $K_r = m - \sum_{j=1}^m k_j$.
2) Obtain K_r i.i.d. draws from the original sample set $\{(\boldsymbol{S}_t^{(j)}, \kappa_t^{(j)})\}_{j=1}^m$, with probabilities proportional to
$$(w_t^{(j)} - k_j), \qquad j = 1, \ldots, m.$$
3) Assign equal weight, i.e., set $\tilde{w}_t^{(j)} = 1$, for each new sample.

It is shown in Appendix C that the samples drawn by the above residual resampling procedure are properly weighted with respect to $p(\boldsymbol{S}_t | \boldsymbol{Y}_t)$, provided that m is sufficiently large. In practice, when small to modest m is used (we used $m = 50$ in our simulations), the resampling procedure can be seen as trading off bias against variance. That is, the new samples with their weights resulting from the resampling procedure are only approximately proper, which introduces small bias in Monte Carlo estimation. On the other hand, however, resampling greatly reduces Monte Carlo variance for the future samples.

Resampling can be done at any time. However, resampling also often adds computational burden and decreases "diversities" of the Monte Carlo filter (i.e., it decreases the number of distinctive filters and loses information). On the other hand, resampling may also sometimes result in loss of efficiency. It is thus desirable to give guidance on when to do resampling. A measure of the efficiency of an importance sampling scheme is the *effective sample size* $\overline{m}_t$, defined as
$$\overline{m}_t \triangleq \frac{m}{1 + \upsilon_t^2}. \tag{45}$$
Heuristically, $\overline{m}_t$ reflects the equivalent size of a set of i.i.d. samples for the set of m weighted ones. It is suggested in [19] that resampling should be performed when the effective sample size becomes small, e.g., $\overline{m}_t \leq \frac{m}{10}$. Alternatively, one can conduct resampling at every fixed-length time interval (say, every five steps).

V. Delayed Estimation

Since the fading process is highly correlated, the future received signals contain information about current data and channel state. A delayed estimate is usually more accurate than the concurrent estimate. This is true for any channel with memory, such as the intersymbol interference channel, and is especially prominent when the transmitted symbols are coded, in which case not only the channel states but also the data symbols are highly correlated. In delayed estimation, instead of making inference on $(\boldsymbol{x}_t, s_t)$ instantaneously with the posterior distribution $p(\boldsymbol{x}_t, s_t | \boldsymbol{Y}_t)$, we delay this inference to a later time $(t + \Delta)$, $\Delta \geq 0$, with the distribution $p(\boldsymbol{x}_t, s_t | \boldsymbol{Y}_{t+\Delta})$. Here we discuss two methods for delayed estimation: the delayed-weight method and the delayed-sample method.

A. Delayed-Weight Method

From the recursive procedure described in Section IV-A, we note by induction that if the set $\{(\boldsymbol{S}_t^{(j)}, w_t^{(j)})\}_{j=1}^m$ is properly weighted with respect to $p(\boldsymbol{S}_t | \boldsymbol{Y}_t)$, then the set $\{(\boldsymbol{S}_{t+\delta}^{(j)}, w_{t+\delta}^{(j)})\}_{j=1}^m$ is properly weighted with respect to $p(\boldsymbol{S}_{t+\delta} | \boldsymbol{Y}_{t+\delta})$, $\delta > 0$. Hence, if we focus our attention on $\boldsymbol{S}_t$ at time $(t + \delta)$ and let $h(\boldsymbol{x}_t, s_t) = 1(s_t = a_i)$ as in (33), we obtain a delayed estimate of the symbol
$$P(s_t = a_i | \boldsymbol{Y}_{t+\delta}) \cong \frac{1}{W_{t+\delta}} \sum_{j=1}^m 1\left(s_t^{(j)} = a_i\right) w_{t+\delta}^{(j)}, \qquad i = 1, \ldots, |\mathcal{A}|. \tag{46}$$
Since the weights $\{w_{t+\delta}^{(j)}\}_{j=1}^m$ contain information about the signals $(y_{t+1}, \ldots, y_{t+\delta})$, the estimate in (46) is usually more accurate. Note that such a delayed estimation method incurs no additional computational cost (i.e., CPU time), but it requires some

extra memory for storing $\{(s_{t+1}^{(j)}, \ldots, s_{t+\delta}^{(j)})\}_{j=1}^{m}$. As will be seen in Section VIII, for uncoded systems, this simple delayed-weight method is quite effective for improving the detection performance over the concurrent method. However, for coded systems, this method is not sufficient for exploiting the constraint structures of both the channel and the symbols, and we must resort to the delayed-sample method, which is described next.

B. Delayed-Sample Method

An alternative method is to generate both the delayed samples and the weights $\{(s_t^{(j)}, w_t^{(j)})\}_{j=1}^{m}$ based on the signals $\boldsymbol{Y}_{t+\Delta}$, hence making $p(\boldsymbol{S}_t|\boldsymbol{Y}_{t+\Delta})$ the target distribution at time $(t+\Delta)$. This procedure will provide better Monte Carlo samples since it utilizes the future information $(y_{t+1}, \ldots, y_{t+\Delta})$ in generating the current sample of s_t. But the algorithm is also more demanding both analytically and computationally because of the need of marginalizing out s_{t+d} for $d = 1, \ldots, \Delta$.

For each possible "future" symbol sequence at time $t+\Delta-1$, i.e.,

$$(s_t, s_{t+1}, \ldots, s_{t+\Delta-1}) \in \mathcal{A}^{\Delta}$$

(a total of $|\mathcal{A}|^{\Delta}$ possibilities), we keep the value of a Δ-step Kalman filter $\{\kappa_{t+\tau}^{(j)}(\underline{s}_t^{t+\tau})\}_{\tau=0}^{\Delta-1}$, where

$$\kappa_{t+\tau}^{(j)}\left(\underline{s}_t^{t+\tau}\right) \triangleq \left[\boldsymbol{\mu}_{t+\tau}\left(\boldsymbol{S}_{t-1}^{(j)}, \underline{s}_t^{t+\tau}\right), \boldsymbol{\Sigma}_{t+\tau}\left(\boldsymbol{S}_{t-1}^{(j)}, \underline{s}_t^{t+\tau}\right)\right], \qquad \tau = 0, 1, \ldots, \Delta-1$$

with $\underline{s}_a^b \triangleq (s_a, s_{a+1}, \ldots, s_b)$. Denote

$$\boldsymbol{\kappa}_{t-1}^{(j)} \triangleq \left\{\kappa_{t-1}^{(j)}; \left\{\kappa_{t+\tau}^{(j)}\left(\underline{s}_t^{t+\tau}\right)\right\}_{\tau=0}^{\Delta-1} : \underline{s}_t^{t+\tau} \in \mathcal{A}^{\tau+1}\right\}.$$

The following recursive procedure is implemented.

1) *Initialization:* Each Kalman filter is initialized as $\kappa_0^{(j)} = (\boldsymbol{\mu}_0^{(j)}, \boldsymbol{\Sigma}_0^{(j)})$, with $\boldsymbol{\mu}_0^{(j)} = \boldsymbol{0}$ and $\boldsymbol{\Sigma}_0^{(j)} = 2\boldsymbol{\Sigma}$, $j = 1, \ldots, m$, where $\boldsymbol{\Sigma}$ is the stationary covariance of $\boldsymbol{x}_t$ and is computed analytically from (6). All importance weights are initialized as $w_0^{(j)} = 1, j = 1, \ldots, m$. Since the data symbols are assumed to be independent, initial symbols are not needed.

   At time $(t+\Delta)$, we perform the following updates for $j = 1, \ldots, m$ to propagate from the sample $\{(\boldsymbol{S}_{t-1}^{(j)}, \kappa_{t-1}^{(j)}, w_{t-1}^{(j)})\}_{j=1}^{m}$, properly weighted for $p(\boldsymbol{S}_{t-1}|\boldsymbol{Y}_{t+\Delta-1})$, to that for $p(\boldsymbol{S}_{t-1}|\boldsymbol{Y}_{t+\Delta})$.

2) *Compute the one-step predictive update for each of the $|\mathcal{A}|^{\Delta}$ Kalman filters*: For each $\underline{s}_t^{t+\Delta-1} \in \mathcal{A}^{\Delta}$, perform the update on the Kalman filter $\kappa_{t+\Delta-1}^{(j)}(\underline{s}_t^{t+\Delta-1})$, according to equations (35), (37) to obtain $\boldsymbol{K}_{t+\Delta}^{(j)}(\underline{s}_t^{t+\Delta-1})$, $\gamma_{t+\Delta}^{(j)}(\underline{s}_t^{t+\Delta-1})$, and $\eta_{t+\Delta}^{(j)}(\underline{s}_t^{t+\Delta-1})$. (Here we make it explicit that these quantities are functions of $\underline{s}_t^{t+\Delta-1}$.)

3) *Compute the trial sampling density:* For each $a_i \in \mathcal{A}$ compute (47) at the bottom of this page.

4) *Impute the symbol s_t:* Draw $s_t^{(j)}$ with probability

$$P\left(s_t^{(j)} = a_i\right) \propto \rho_{t,i}^{(j)}, \qquad a_i \in \mathcal{A}. \tag{48}$$

   Append $s_t^{(j)}$ to $\boldsymbol{S}_{t-1}^{(j)}$ and obtain $\boldsymbol{S}_t^{(j)}$.

5) *Compute the importance weight:* See (49) and (50), at the top of the following page, where

$$p\left(y_{t+\tau}\,\middle|\,\boldsymbol{Y}_{t+\tau-1}, \boldsymbol{S}_{t-1}^{(j)}, \underline{s}_t^{t+\tau}\right) \sim \mathcal{N}_c\left(s_{t+\tau}\gamma_{t+\tau}^{(j)}\left(\underline{s}_t^{t+\tau-1}\right), \eta_{t+\tau}^{(j)}\left(\underline{s}_t^{t+\tau-1}\right)\right).$$

6) *Compute the one-step filtering update for each of the $|A|^{\Delta}$ Kalman filters:* Using the values of $s_t^{(j)}$ and $y_{t+\Delta}$, for each $\underline{s}_{t+1}^{t+\Delta} \in \mathcal{A}^{\Delta}$ perform a one-step filtering update on the Kalman filter $\kappa_{t+\Delta-1}^{(j)}(\underline{s}_t^{t+\Delta-1})$ according to (42) and (43) to obtain

$$\kappa_{t+\Delta}^{(j)}\left(\underline{s}_{t+1}^{t+\Delta}\right) \triangleq \left[\boldsymbol{\mu}_{t+\Delta}\left(\boldsymbol{S}_t^{(j)}, \underline{s}_{t+1}^{t+\Delta}\right), \boldsymbol{\Sigma}_{t+\Delta}\left(\boldsymbol{S}_t^{(j)}, \underline{s}_{t+1}^{t+\Delta}\right)\right].$$

   With this and the subset of $\{\kappa_{t+\tau}^{(j)}(\underline{s}_{t+1}^{t+\tau})\}_{\tau=0}^{\Delta-1}$ corresponding to the sample $s_t^{(j)}$, which has been obtained in the previous iteration, we form the new filter class $\boldsymbol{\kappa}_t^{(j)}$.

7) Do resampling as described in Section IV-B when $\overline{m}_t$ in (45) is below a threshold.

The dominant computation of the above delayed-sample method at each time t involves the $(m|\mathcal{A}|^{\Delta})$ one-step Kalman filter updates, which, as before, can be carried out in parallel. Finally, we note that we can use the delayed-sample method in conjunction with the delayed-weight method. For example, using the delayed-sample method, we generate delayed samples and weights $\{(s_t^{(j)}, w_t^{(j)})\}_{j=1}^{m}$ based on the signals $\boldsymbol{Y}_{t+\Delta}$. Then with an additional delay δ, we can use the following

$$\begin{aligned}
\rho_{t,i}^{(j)} &\triangleq P\left(s_t = a_i|\boldsymbol{S}_{t-1}^{(j)}, \boldsymbol{Y}_{t+\Delta}\right) \propto p\left(\boldsymbol{Y}_{t+\Delta}, \boldsymbol{S}_{t-1}^{(j)}, s_t = a_i\right) \\
&= \sum_{\underline{s}_{t+1}^{t+\Delta} \in \mathcal{A}^{\Delta}} p\left(\boldsymbol{Y}_{t+\Delta}, \boldsymbol{S}_{t-1}^{(j)}, s_t = a_i, \underline{s}_{t+1}^{t+\Delta}\right) \\
&\propto \sum_{\underline{s}_{t+1}^{t+\Delta} \in \mathcal{A}^{\Delta}} \prod_{\tau=0}^{\Delta} \underbrace{p\left(y_{t+\tau}\,\middle|\,\boldsymbol{Y}_{t+\tau-1}, \boldsymbol{S}_{t-1}^{(j)}, s_t = a_i, \underline{s}_{t+1}^{t+\tau}\right)}_{\mathcal{N}_c\left(s_{t+\tau}\gamma_{t+\tau}^{(j)}\left(\underline{s}_t^{t+\tau-1}\right), \eta_{t+\tau}^{(j)}\left(\underline{s}_t^{t+\tau-1}\right)\right)} p(s_t = a_i) \cdot \prod_{\tau=0}^{\Delta} p(s_{t+\tau}).
\end{aligned} \tag{47}$$

$$
\begin{aligned}
w_t^{(j)} &= w_{t-1}^{(j)} \cdot \frac{p\left(\boldsymbol{S}_t^{(j)} \middle| \boldsymbol{Y}_{t+\Delta}\right)}{p\left(\boldsymbol{S}_{t-1}^{(j)} \middle| \boldsymbol{Y}_{t+\Delta-1}\right) p\left(s_t^{(j)} \middle| \boldsymbol{S}_{t-1}^{(j)}, \boldsymbol{Y}_{t+\Delta}\right)} \\
&= w_{t-1}^{(j)} \cdot \frac{p\left(\boldsymbol{S}_{t-1}^{(j)} \middle| \boldsymbol{Y}_{t+\Delta}\right)}{p\left(\boldsymbol{S}_{t-1}^{(j)} \middle| \boldsymbol{Y}_{t+\Delta-1}\right)} \\
&\propto w_{t-1}^{(j)} \cdot \frac{p\left(\boldsymbol{Y}_{t+\Delta}, \boldsymbol{S}_{t-1}^{(j)}\right)}{p\left(\boldsymbol{Y}_{t+\Delta-1}, \boldsymbol{S}_{t-1}^{(j)}\right)} \\
&= w_{t-1}^{(j)} \cdot \frac{\sum_{\underline{s}_t^{t+\Delta} \in \mathcal{A}^{\Delta+1}} p\left(\underline{s}_t^{t+\Delta}, \boldsymbol{S}_{t-1}^{(j)}, \boldsymbol{Y}_{t+\Delta}\right)}{\sum_{\underline{s}_t^{t+\Delta-1} \in \mathcal{A}^{\Delta}} p\left(\underline{s}_t^{t+\Delta-1}, \boldsymbol{S}_{t-1}^{(j)}, \boldsymbol{Y}_{t+\Delta-1}\right)}
\end{aligned} \tag{49}
$$

$$
\propto w_{t-1}^{(j)} \cdot \frac{\sum_{\underline{s}_t^{t+\Delta} \in \mathcal{A}^{\Delta+1}} \left[\prod_{\tau=0}^{\Delta} p\left(y_{t+\tau} \middle| \boldsymbol{Y}_{t+\tau-1}, \boldsymbol{S}_{t-1}^{(j)}, \underline{s}_t^{t+\tau}\right) \cdot \prod_{\tau=0}^{\Delta} p(s_{t+\tau})\right]}{\sum_{\underline{s}_t^{t+\Delta-1} \in \mathcal{A}^{\Delta}} \left[\prod_{\tau=0}^{\Delta-1} p\left(y_{t+\tau} \middle| \boldsymbol{Y}_{t+\tau-1}, \boldsymbol{S}_{t-1}^{(j)}, \underline{s}_t^{t+\tau}\right) \cdot \prod_{\tau=0}^{\Delta-1} p(s_{t+\tau})\right]} \tag{50}
$$

delayed-weight method to estimate the symbol *a posteriori* probability:

$$
P(s_t = a_i | \boldsymbol{Y}_{t+\Delta+\delta}) \cong \frac{1}{W_{t+\delta}} \sum_{j=1}^{m} 1(s_t^{(j)} = a_i) w_{t+\delta}^{(j)}, \quad i = 1, \ldots, |\mathcal{A}|. \tag{51}
$$

VI. Adaptive Receiver in Fading Gaussian Noise Channels—Coded Case

So far we have considered the problem of detecting uncoded independent symbols in flat-fading channels. In what follows we extend the adaptive receiver technique presented in Section IV and address the problem of sequential decoding of information bits in a convolutionally coded system signaling through a flat-fading channel.

Consider a binary rate $\frac{k_0}{n_0}$ 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 $\boldsymbol{d}_t = (d_{t,1}, \ldots, d_{t,k_0})$; the encoder output at time t is a block of code bits $\boldsymbol{b}_t = (b_{t,1}, \ldots, b_{t,n_0})$. For simplicity here we assume that BPSK modulation is employed. Then the transmitted symbols at time t are $\boldsymbol{s}_t = (s_{t,1}, \ldots, s_{t,n_0})$, where $s_{t,l} = 2b_{t,l} - 1$, $l = 1, \ldots, n_0$. (That is, $s_{t,l} = 1$ if $b_{t,l} = 1$, and $s_{t,l} = -1$ if $b_{t,l} = 0$.) Since $\boldsymbol{b}_t$ is determined by $(\boldsymbol{d}_t, \boldsymbol{d}_{t-1}, \ldots, \boldsymbol{d}_{t-\nu_0})$, so is $\boldsymbol{s}_t$. Hence we can write

$$
\boldsymbol{s}_t = \psi(\boldsymbol{d}_t, \boldsymbol{d}_{t-1}, \ldots, \boldsymbol{d}_{t-\nu_0}) \tag{52}
$$

for some function $\psi(\cdot)$ which is determined by the structure of the encoder.

Let $\boldsymbol{y}_t = (y_{t,1}, \ldots, y_{t,n_0})$ be the received signals at time t and let $\boldsymbol{\alpha}_t = (\alpha_{t,1}, \ldots, \alpha_{t,n_0})$ be the channel states corresponding to $\boldsymbol{b}_t$ and $\boldsymbol{d}_t$. Recall that $\alpha_{t-1,n_0} = \boldsymbol{h}^H \boldsymbol{x}_{t-1,n_0}$. Denote also

$$
\begin{aligned}
\boldsymbol{D}_t &\triangleq (\boldsymbol{d}_0, \ldots, \boldsymbol{d}_t) \\
\boldsymbol{S}_t &\triangleq (\boldsymbol{s}_0, \ldots, \boldsymbol{s}_t) \\
\boldsymbol{Y}_t &\triangleq (\boldsymbol{y}_0, \ldots, \boldsymbol{y}_t).
\end{aligned}
$$

The Monte Carlo samples recorded at time $(t-1)$ are $\{(\boldsymbol{D}_{t-1}^{(j)}, \kappa_{t-1,n_0}^{(j)}, w_{t-1}^{(j)})\}_{j=1}^m$ where

$$
\kappa_{t-1,n_0}^{(j)} \triangleq [\boldsymbol{\mu}_{t-1,n_0}^{(j)}, \boldsymbol{\Sigma}_{t-1,n_0}^{(j)}]
$$

contains the mean and covariance matrix of the state vector channel $\boldsymbol{x}_{t-1,n_0}$ conditioned on $\boldsymbol{D}_{t-1}^{(j)}$ and $\boldsymbol{Y}_{t-1}$. That is,

$$
p\left(\boldsymbol{x}_{t-1,n_0} \middle| \boldsymbol{D}_{t-1}^{(j)}, \boldsymbol{Y}_{t-1}\right) \sim \mathcal{N}_c\left(\boldsymbol{\mu}_{t-1,n_0}^{(j)}, \boldsymbol{\Sigma}_{t-1,n_0}^{(j)}\right). \tag{53}
$$

As before, given the information bit sequence $\boldsymbol{D}_{t-1}^{(j)}$, the corresponding $\kappa_{t-1,n_0}^{(j)}$ is obtained by a Kalman filter. Our algorithm is as follows.

1) *Initialization:* Each Kalman filter is initialized as $\kappa_{0,n_0}^{(j)} = (\boldsymbol{\mu}_{0,n_0}^{(j)}, \boldsymbol{\Sigma}_{0,n_0}^{(j)})$, with $\boldsymbol{\mu}_{0,n_0}^{(j)} = \boldsymbol{0}$ and $\boldsymbol{\Sigma}_{0,n_0}^{(j)} = 2\boldsymbol{\Sigma}$, $j = 1, \ldots, m$, where $\boldsymbol{\Sigma}$ is the stationary covariance of $\boldsymbol{x}_t$ and is computed analytically from (6). All importance weights are initialized as $w_0^{(j)} = 1$, $j = 1, \ldots, m$. The initial $\boldsymbol{D}_0^{(j)}$ are randomly generated from the set $\{0, 1\}^{k_0}$, $j = 1, \ldots, m$.

   At time t, we implement the following steps to update each sample j, $j = 1, \ldots, m$.

2) *Compute the n_0-step update of the Kalman filter:* For each possible code vector $\boldsymbol{d}_t = \boldsymbol{a}_i \in \{0, 1\}^{k_0}$, compute the corresponding symbol vector $\boldsymbol{s}_t$ using (52) to obtain

$$
\boldsymbol{s}_t^{(j)}(\boldsymbol{a}_i) = \psi\left(\boldsymbol{d}_t = \boldsymbol{a}_i, \boldsymbol{d}_{t-1}^{(j)}, \ldots, \boldsymbol{d}_{t-\nu_0}^{(j)}\right). \tag{54}
$$

Let

$$\boldsymbol{\Sigma}_{t,0}^{(j)}(\boldsymbol{a}_i) \triangleq \boldsymbol{\Sigma}_{t-1,n_0}^{(j)}$$

and

$$\mu_{t,0}^{(j)}(\boldsymbol{a}_i) \triangleq \mu_{t-1,n_0}^{(j)}.$$

Perform n_0 steps of Kalman filter update, using $\boldsymbol{s}_t^{(j)}(\boldsymbol{a}_i)$ and $\boldsymbol{y}_t$, as follows: for $l = 1, \ldots, n_0$, compute

$$\boldsymbol{K}_{t,l}^{(j)}(\boldsymbol{a}_i) = \boldsymbol{F}\boldsymbol{\Sigma}_{t,l-1}^{(j)}(\boldsymbol{a}_i)\boldsymbol{F}^H + \boldsymbol{g}\boldsymbol{g}^H, \tag{55}$$

$$\gamma_{t,l}^{(j)}(\boldsymbol{a}_i) = \boldsymbol{h}^H\boldsymbol{K}_{t,l}^{(j)}(\boldsymbol{a}_i)\boldsymbol{h} + \sigma^2, \tag{56}$$

$$\eta_{t,l}^{(j)}(\boldsymbol{a}_i) = \boldsymbol{h}^H\boldsymbol{F}\boldsymbol{\mu}_{t,l-1}^{(j)}(\boldsymbol{a}_i), \tag{57}$$

$$\begin{aligned}\boldsymbol{\mu}_{t,l}^{(j)}(\boldsymbol{a}_i) = &\boldsymbol{F}\boldsymbol{\mu}_{t,l-1}^{(j)}(\boldsymbol{a}_i)\\ &+ \frac{1}{\gamma_{t,l}^{(j)}(\boldsymbol{a}_i)}\left[y_{t,l} - s_{t,l}^{(j)}(\boldsymbol{a}_i)\eta_{t,l}^{(j)}(\boldsymbol{a}_i)\right]\\ &\cdot \boldsymbol{K}_{t,l}^{(j)}(\boldsymbol{a}_i)\boldsymbol{h}\end{aligned} \tag{58}$$

$$\begin{aligned}\boldsymbol{\Sigma}_{t,l}^{(j)}(\boldsymbol{a}_i) = &\boldsymbol{K}_{t,l}^{(j)}(\boldsymbol{a}_i)\\ &- \frac{1}{\gamma_{t,l}^{(j)}(\boldsymbol{a}_i)}\boldsymbol{K}_{t,l}^{(j)}(\boldsymbol{a}_i)\boldsymbol{h}\boldsymbol{h}^H\boldsymbol{K}_{t,l}^{(j)}(\boldsymbol{a}_i).\end{aligned} \tag{59}$$

In (54)–(59) it is made explicit that the quantity on the left side of each equation is a function of the code bit vector $\boldsymbol{a}_i$. We therefore obtain $\{\gamma_{t,l}^{(j)}(\boldsymbol{a}_i),\ \eta_{t,l}^{(j)}(\boldsymbol{a}_i)\}_{l=1}^{n_0}$ and $[\boldsymbol{\mu}_{t,n_0}^{(j)}(\boldsymbol{a}_i),\ \boldsymbol{\Sigma}_{t,n_0}^{(j)}(\boldsymbol{a}_i)]$ for each $\boldsymbol{a}_i \in \{0, 1\}^{k_0}$.

3) *Compute the trial sampling density:* For each $\boldsymbol{a}_i \in \{0,1\}^{k_0}$, compute (60)–(62) at the bottom of this page, where (61) follows from the fact that $\boldsymbol{d}_t$ is independent of $\boldsymbol{D}_{t-1}$ and $\boldsymbol{Y}_{t-1}$.

4) *Impute the code bit vector* $\boldsymbol{d}_t$: Draw $\boldsymbol{d}_t^{(j)}$ from the set $\{0, 1\}^{k_0}$ with probability

$$P\left(\boldsymbol{d}_t^{(j)} = \boldsymbol{a}_i\right) \propto \rho_{t,i}^{(j)}, \qquad \boldsymbol{a}_i \in \{0, 1\}^{k_0}. \tag{63}$$

Append $\boldsymbol{d}_t^{(j)}$ to $\boldsymbol{D}_{t-1}^{(j)}$ and obtain $\boldsymbol{D}_t^{(j)}$. Pick the updated Kalman filter values

$$\boldsymbol{\mu}_{t,n_0}^{(j)} \triangleq \boldsymbol{\mu}_{t,n_0}^{(j)}(\boldsymbol{d}_t^{(j)})$$

and

$$\boldsymbol{\Sigma}_{t,n_0}^{(j)} = \boldsymbol{\Sigma}_{t,n_0}^{(j)}(\boldsymbol{d}_t^{(j)})$$

from the results in Step 1), according to the value of the sample $\boldsymbol{d}_t^{(j)}$. We obtain $\kappa_{t,n_0}^{(j)} = [\boldsymbol{\mu}_{t,n_0}^{(j)},\ \boldsymbol{\Sigma}_{t,n_0}^{(j)}]$.

5) *Compute the importance weight:*

$$\begin{aligned}w_t^{(j)} &= w_{t-1}^{(j)} \cdot p\left(\boldsymbol{y}_t \,\middle|\, \boldsymbol{D}_{t-1}^{(j)}, \boldsymbol{Y}_{t-1}\right)\\ &= w_{t-1}^{(j)} \cdot \sum_{\boldsymbol{a}_i \in \{0,1\}^{k_0}} p\left(\boldsymbol{y}_t, \boldsymbol{d}_t = \boldsymbol{a}_i \,\middle|\, \boldsymbol{D}_{t-1}^{(j)}, \boldsymbol{Y}_{t-1}\right)\\ &\propto w_{t-1}^{(j)} \sum_{\boldsymbol{a}_i \in \{0,1\}^{k_0}} \rho_{t,i}^{(j)}\end{aligned} \tag{64}$$

where (64) follows from (60).

6) Do resampling as described in Section IV-B when $\overline{m}_t$ in (45) is below a threshold.

Following the same line of proof as in Appendix B, it can be shown that $\{(\boldsymbol{D}_t^{(j)},\ \kappa_{t,n_0}^{(j)},\ w_t^{(j)})\}_{j=1}^m$ drawn by the above procedure are properly weighted with respect to $p(\boldsymbol{D}_t|\boldsymbol{Y}_t)$ provided that the samples $\{(\boldsymbol{D}_{t-1}^{(j)},\ \kappa_{t-1,n_0}^{(j)},\ w_{t-1}^{(j)})\}_{j=1}^m$ are properly weighted with respect to $p(\boldsymbol{D}_{t-1}|\boldsymbol{Y}_{t-1})$. Note that in the coded case, the phase ambiguity is prevented by the code constraint (52), and differential encoding is not needed.

At each time t, the major computation involved in the above adaptive decoding algorithm is the $(m\, n_0\, 2^{k_0})$ one-step Kalman filter updates, which can be carried out by $(m\, 2^{k_0})$ processing units, each computing an n_0-step update. (Note that $\boldsymbol{d}_t$ contains k_0 bits of information.) Furthermore, if the delayed-sample method outlined in Section V-B is employed for delayed estimation, then for a delay of Δ time units, a total of $(m\, n_0\, 2^{k_0(\Delta+1)})$ one-step Kalman filter updates are needed at each time t, which can be distributed among $(m\, 2^{k_0(\Delta+1)} t)$ processing units, each computing an n_0-step update.

VII. Adaptive Receivers in Fading Non-Gaussian Noise Channels

To date, most of the work on signal detection in fading channels assumes that the additive ambient channel noise has a Gaussian distribution. In practice, however, the ambient noise in many mobile communication channels is impulsive, due to various natural and man-made impulsive sources [1], [2], [24], [25], [28]. In [33], a technique is developed for signal detection in fading channels with impulsive noise based on the Masreliez nonlinear filtering [23] and making use of pilot symbols and decision feedback. In this section, we develop an adaptive receiver for flat-fading channels with non-Gaussian ambient noise, using the mixture Kalman filtering technique.

$$\rho_{t,i}^{(j)} \triangleq P\left(\boldsymbol{d}_t = \boldsymbol{a}_i \,\middle|\, \boldsymbol{D}_{t-1}^{(j)}, \boldsymbol{Y}_t\right) \propto p\left(\boldsymbol{y}_t, \boldsymbol{Y}_{t-1}, \boldsymbol{d}_t = \boldsymbol{a}_i, \boldsymbol{D}_{t-1}^{(j)}\right) \propto p\left(\boldsymbol{y}_t, \boldsymbol{d}_t = \boldsymbol{a}_i \,\middle|\, \boldsymbol{D}_{t-1}^{(j)}, \boldsymbol{Y}_{t-1}\right) \tag{60}$$

$$= P\left(\boldsymbol{d}_t = \boldsymbol{a}_i \,\middle|\, \boldsymbol{D}_{t-1}^{(j)}, \boldsymbol{Y}_{t-1}\right) p\left(\boldsymbol{y}_t \,\middle|\, \boldsymbol{d}_t = \boldsymbol{a}_i, \boldsymbol{D}_{t-1}^{(j)}, \boldsymbol{Y}_{t-1}\right)$$

$$= P(\boldsymbol{d}_t = \boldsymbol{a}_i) p\left(\boldsymbol{y}_t \,\middle|\, \boldsymbol{s}_t^{(j)}(\boldsymbol{a}_i) = \psi\left(\boldsymbol{d}_t = \boldsymbol{a}_i, \boldsymbol{d}_{t-1}^{(j)}, \ldots, \boldsymbol{d}_{t-\nu_0}^{(j)}\right), \boldsymbol{S}_{t-1}^{(j)}, \boldsymbol{Y}_{t-1}\right) \tag{61}$$

$$\propto P(\boldsymbol{d}_t = \boldsymbol{a}_i) \prod_{l=1}^{n_0} \underbrace{p\left(y_{t,l} \,\middle|\, \boldsymbol{S}_{t-1}^{(j)}, s_{t,1}^{(j)}(\boldsymbol{a}_i), \ldots, s_{t,l}^{(j)}(\boldsymbol{a}_i), \boldsymbol{Y}_{t-1}, y_{t,1}, \ldots, y_{t,l-1}\right)}_{\mathcal{N}_c\left(s_{t,l}^{(j)}(\boldsymbol{a}_i)\eta_{t,l}^{(j)}(\boldsymbol{a}_i),\, \gamma_{t,l}^{(j)}(\boldsymbol{a}_i)\right)} \tag{62}$$

As in the case of Gaussian fading channels, we first develop adaptive receivers for uncoded systems. Consider the state-space system given by (11)–(12). Note that given both the symbol sequence $\boldsymbol{S}_t \triangleq (s_0, \ldots, s_t)$, and the noise indicator sequence $\boldsymbol{I}_t \triangleq (I_0, \ldots, I_t)$, this system is linear and Gaussian. Hence

$$p(\boldsymbol{x}_t|\boldsymbol{S}_t, \boldsymbol{I}_t, \boldsymbol{Y}_t) \sim \mathcal{N}_c(\boldsymbol{\mu}_t(\boldsymbol{S}_t, \boldsymbol{I}_t), \boldsymbol{\Sigma}_t(\boldsymbol{S}_t, \boldsymbol{I}_t)) \tag{65}$$

where the mean $\boldsymbol{\mu}_t(\boldsymbol{S}_t, \boldsymbol{I}_t)$ and the covariance matrix $\boldsymbol{\Sigma}_t(\boldsymbol{S}_t, \boldsymbol{I}_t)$ can be obtained by a Kalman filter with given $\boldsymbol{S}_t$ and $\boldsymbol{I}_t$. As before, we seek to obtain properly weighted samples $\{\boldsymbol{S}_t^{(j)}, \boldsymbol{I}_t^{(j)}, \kappa_t^{(j)}, w_t^{(j)}\}_{j=1}^m$, with respect to the distribution $p(\boldsymbol{S}_t, \boldsymbol{I}_t|\boldsymbol{Y}_t)$. These samples are then used to estimate the transmitted symbols and channel parameters.

1) *Initialization:* This step is the same as that in the Gaussian case. Note that no initial values for $I_0^{(j)}$ are needed due to independence.

   At time t, the following updates are implemented for each sample j, $j = 1, \ldots, m$.

2) *Compute the one-step predictive update of the Kalman filter $\kappa_{t-1}^{(j)}$:*

$$\boldsymbol{K}_t^{(j)} = \boldsymbol{F}\boldsymbol{\Sigma}_{t-1}^{(j)}\boldsymbol{F}^H + \boldsymbol{g}\boldsymbol{g}^H \tag{66}$$

$$\tilde{\gamma}_t^{(j)} = \boldsymbol{h}^H \boldsymbol{K}_t^{(j)} \boldsymbol{h} \tag{67}$$

$$\eta_t^{(j)} = \boldsymbol{h}^H \boldsymbol{F} \boldsymbol{\mu}_{t-1}^{(j)}. \tag{68}$$

   Conditioned on $\boldsymbol{S}_t^{(j)}$ and $\boldsymbol{I}_t^{(j)}$, the predictive distribution is then given by

$$p\left(y_t \middle| \boldsymbol{S}_t^{(j)}, \boldsymbol{I}_t^{(j)}, \boldsymbol{Y}_{t-1}\right) \sim \mathcal{N}_c\left(s_t^{(j)}\eta_t^{(j)}, \tilde{\gamma}_t^{(j)} + \sigma^2_{I_t^{(j)}}\right). \tag{69}$$

3) *Compute the trial sampling density:* For each $(a, \delta)_i \in \mathcal{A} \times \{1, 2\}$, compute (70) at the bottom of this page.

4) *Impute the symbol and the noise indicator* (s_t, I_t): Draw $(s_t^{(j)}, I_t^{(j)})$ from the set $\mathcal{A} \times \{1, 2\}$ with probability

$$P\left(\left(s_t^{(j)}, I_t^{(j)}\right) = (a, \delta)_i\right) \propto \rho_{t,i}^{(j)}, \qquad (a, \delta)_i \in \mathcal{A} \times \{1, 2\}. \tag{71}$$

   Append $(s_t^{(j)}, I_t^{(j)})$ to $(\boldsymbol{S}_{t-1}^{(j)}, \boldsymbol{I}_{t-1}^{(j)})$ and obtain $(\boldsymbol{S}_t^{(j)}, \boldsymbol{I}_t^{(j)})$.

5) *Compute the importance weight:*

$$\begin{aligned} w_t^{(j)} &= w_{t-1}^{(j)} \cdot p\left(y_t \middle| \boldsymbol{S}_{t-1}^{(j)}, \boldsymbol{I}_{t-1}^{(j)}, \boldsymbol{Y}_{t-1}\right) \\ &= w_{t-1}^{(j)} \cdot \sum_{(a,\delta)_i \in \mathcal{A}\times\{1,2\}} \\ &\quad \cdot p\left(y_t \middle| (s_t, I_t) = (a, \delta)_i, \boldsymbol{S}_{t-1}^{(j)}, \boldsymbol{I}_{t-1}^{(j)}, \boldsymbol{Y}_{t-1}\right) \\ &\quad \cdot P((s_t, I_t) = (a, \delta)_i) \\ &= w_{t-1}^{(j)} \cdot \sum_{(a,\delta)_i \in \mathcal{A}\times\{1,2\}} \rho_{t,i}^{(j)}, \end{aligned} \tag{72}$$

   where (72) follows from (70).

6) *Compute the one-step filtering update of the Kalman filter:* Based on the imputed symbol and indicator $(s_t^{(j)}, I_t^{(j)})$, and the observation y_t, complete the Kalman filter update to obtain $\kappa_t^{(j)} = [\boldsymbol{\mu}_t^{(j)}, \boldsymbol{\Sigma}_t^{(j)}]$ according to (42) and (43) with $\gamma_t^{(j)} = \tilde{\gamma}_t^{(j)} + \sigma^2_{I_t(j)}$.

7) Do resampling as described in Section IV-B when the effective sample size $\tilde{m}_t$ in (45) is below a threshold.

The proof that the above algorithm gives the properly weighted samples is similar to that for the Gaussian fading channels in Appendix B. The dominant computation involved in the above algorithm at each time t includes m one-step Kalman filter updates. If the delayed-sample method is employed for delayed estimation with a delay of Δ time units, then at each time t, $(m\,(2|\mathcal{A}|)^{\Delta})$ one-step Kalman filter updates are needed because $|\mathcal{A} \times \{1, 2\}| = 2|\mathcal{A}|$, which can be implemented in parallel.

Moreover, we can also develop the adaptive receiver algorithm for coded systems in non-Gaussian noise flat-fading channels, similar to the one discussed in Section VI. For a rate $\frac{k_0}{n_0}$ convolutional code, if the delayed-sample method is used with a delay of Δ time units, then at each time t a total of $(m\,n_0\,2^{(k_0+n_0)(\Delta+1)})$ one-step Kalman filter updates are needed, which can be distributed among $(m\,2^{(k_0+n_0)(\Delta+1)})$ processors, each computing one n_0-step update. (With a delay of Δ units, there are $2^{k_0(\Delta+1)}$ possible code vectors, and there are $2^{n_0(\Delta+1)}$ possible noise indicator vectors.)

VIII. Simulation Results

In this section, we provide some computer simulation examples to demonstrate the performance of the proposed sequential Monte Carlo receivers in fading channels under various conditions. The fading process is modeled by the output of a Butter-

$$\begin{aligned} \rho_{t,i}^{(j)} &\triangleq P\left((s_t, I_t) = (a, \delta)_i \middle| \boldsymbol{S}_{t-1}^{(j)}, \boldsymbol{I}_{t-1}^{(j)}, \boldsymbol{Y}_t\right) \\ &\propto p\left(y_t, \boldsymbol{Y}_{t-1}, (s_t, I_t) = (a, \delta)_i, \boldsymbol{S}_{t-1}^{(j)}, \boldsymbol{I}_{t-1}^{(j)}\right) \\ &= \underbrace{p\left(y_t \middle| (s_t, I_t) = (a, \delta)_i, \boldsymbol{S}_{t-1}^{(j)}, \boldsymbol{I}_{t-1}^{(j)}, \boldsymbol{Y}_{t-1}\right)}_{\mathcal{N}_c\left(s_t\,\eta_t^{(j)}, \tilde{\gamma}_t^{(j)} + \sigma^2_{I_t^{(j)}}\right)} P((s_t, I_t) = (a, \delta)_i). \end{aligned} \tag{70}$$

worth 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 $\frac{1}{T}$) $f_d T = 0.05$, which is a fast fading scenario. Specifically, the fading coefficients $\{\alpha_t\}$ is modeled by the following ARMA $(3, 3)$ process

$$\begin{aligned}\alpha_t - 2.37409\alpha_{t-1} + 1.92936\alpha_{t-2} - 0.53208\alpha_{t-3} \\ = 10^{-2}(0.89409u_t + 2.68227u_{t-1} + 2.68227u_{t-2} \\ + 0.89409u_{t-3})\end{aligned} \tag{73}$$

where $u_t \sim \mathcal{N}_c(0, 1)$. The filter coefficients in (73) are chosen such that $\mathrm{Var}\{\alpha_t\} = 1$. It is assumed that BPSK modulation is employed, i.e., the transmitted symbols $s_t \in \{+1, -1\}$.

In order to demonstrate the high performance of the proposed adaptive receiver, in the following simulation examples we compare the performance (in terms of bit-error rate (BER)) of the proposed sequential Monte Carlo receivers with that of the following three receiver schemes.

- *Known channel lower bound:* In this case, we assume that the fading coefficients $\{\alpha_t\}$ are known to the receiver. Then by (1), the optimal coherent detection rule is given by $\hat{s}_t = \mathrm{sign}\,(\Re\{\alpha_t^* y_t\})$ for both the Gaussian noise case (2) and the non-Gaussian noise case (3).
- *Genie-aided lower bound:* In this case, we assume that a genie provides the receiver with an observation of the modulation-free channel coefficient corrupted by additive noise with the same variance, i.e., $\tilde{y}_t = \alpha_t + \tilde{n}_t$, where $\tilde{n}_t \sim \mathcal{N}_c(0, \sigma^2)$ for the Gaussian noise case and $\tilde{n}_t \sim \mathcal{N}_c(0, \sigma_{I_t}^2)$ for the non-Gaussian noise case. In case of non-Gaussian noise, the genie also provides the receiver with the noise indicator I_t. The receiver then uses a Kalman filter to track the fading process based on the information provided by the genie; i.e., it computes $\hat{\alpha}_t = E\{\alpha_t | \tilde{\boldsymbol{Y}}_t, \boldsymbol{I}_t\}$. The transmitted symbols are then demodulated according to $\hat{s}_t = \mathrm{sign}\,(\Re\{\hat{\alpha}_t^* y_t\})$. It is clear that such a genie-aided bound is lower-bounded by the known channel bound. It should also be noted that the genie is used only for calculating the lower bound. Our proposed algorithms estimate the channel and the symbols simultaneously with no help from the genie.
- *Differential detector:* In this case, no attempt is made to estimate the fading channel. Instead, the receiver detects the phase difference in two consecutively transmitted bits by using the simple rule of differential detection: $\widehat{b_t b_{t-1}} = \mathrm{sign}\,(\Re\{y_t^* y_{t-1}\})$.

First we consider the performance of the sequential Monte Carlo receiver in a fading Gaussian noise channel without coding. In this case, differential encoding and decoding are employed to resolve the phase ambiguity. The adaptive receiver implements the algorithm described in Section IV-A. The number of Monte Carlo samples drawn at each time was empirically set as $m = 50$. Simulation results showed that the performance did not improve much when m was increased to 100, while it degraded notably when m was reduced to 20. The resampling procedure discussed in Section IV-B was employed to maintain the efficiency of the algorithm, in which the effective sample size threshold is $\overline{m}_t = m/10$. The *delayed-weight* method discussed in Section V-A was used to extract further information from future received signals, which resulted in an improved performance compared with concurrent estimation. In each simulation, the sequential Monte Carlo algorithm was run on 10 000 symbols, (i.e., $t = 1, \ldots, 10\,000$). In counting the symbol detection errors, the first 50 symbols were discarded to allow the algorithm to reach the steady state. In Fig. 3, the BER performance versus the signal-to-noise ratio (defined as $\mathrm{Var}\{\alpha_t\}/\mathrm{Var}\{n_t\}$) corresponding to delay values $\delta = 0$ (concurrent estimate), $\delta = 1$, and $\delta = 2$ is plotted. In the same figure, we also plot the known channel lower bound, the genie-aided lower bound, and the BER curve of the differential detector. From this figure it is seen that for the uncoded case, with only a small amount of delay, the performance of the sequential Monte Carlo receiver can be significantly improved by the delayed-weight method compared with the concurrent estimate. Even with the concurrent estimate, the proposed adaptive receiver does not exhibit an error floor, as does the differential detector. Moreover, with a delay $\delta = 2$, the proposed adaptive receiver essentially achieves the genie-aided lower bound. We have also implemented the delayed-sample method for this case and found that it offers little improvement over the delayed-weight method.

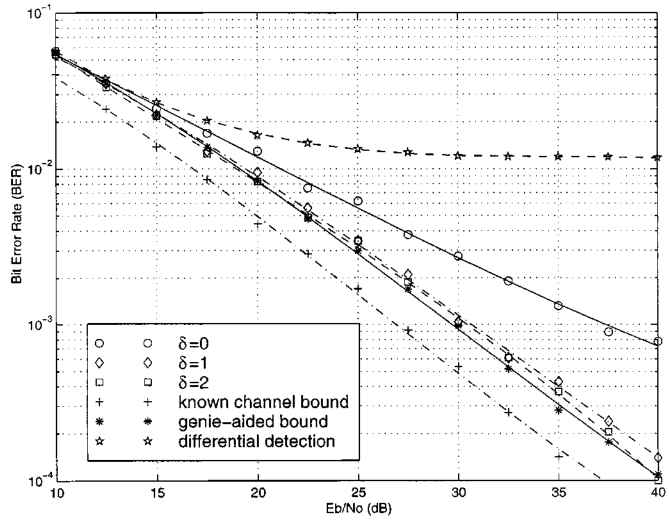

Fig. 3. BER performance of the sequential Monte Carlo receiver in a fading channel with Gaussian noise and without coding. The delayed-weight method is used. The BER curves corresponding to delays $\delta = 0$, $\delta = 1$, and $\delta = 2$ are shown. Also shown in the same figure are the BER curves for the known channel lower bound, the genie-aided lower bound, and the differential detector.

The uncoded BER performance of the proposed adaptive receiver, together with that of the other three receiver schemes, in a fading channel with non-Gaussian ambient noise is shown in Fig. 4. The noise distribution is given by the two-term Gaussian mixture model (3) with $\kappa = 100$ and $\epsilon = 0.1$. As mentioned earlier in this case for the genie-aided bound, the genie not only provides the observation of the noise-corrupted modulation-free channel coefficients, but also the true noise indicator $\{I_t\}$ to the channel estimator. It is seen from this figure that, again, the delayed-weight method offers significant improvement over the concurrent estimate, although in this case the BER curve for

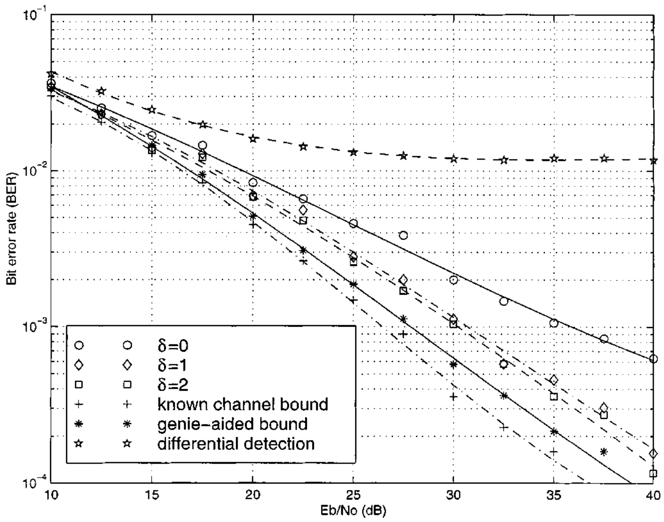

Fig. 4. BER performance of the sequential Monte Carlo receiver in a fading channel with non-Gaussian noise and without coding. $\epsilon = 0.1$, $\kappa = 100$. The delayed-weight method is used. The BER curves corresponding to delays $\delta = 0$, $\delta = 1$, and $\delta = 2$ are shown. Also shown in the same figure are the BER curves for the known channel lower bound, the genie-sided lower bound, and the differential detector.

$\delta = 2$ is slightly off the genie-aided lower bound. Furthermore, the proposed adaptive receiver does not have the error floor exhibited by the simple differential detector.

We next show the performance of the proposed sequential Monte Carlo receiver in a coded system. 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 algorithm discussed in Section VI with a combination of delayed-sample and delayed-weight method. That is, the information bits samples $\{\boldsymbol{d}_t^{(j)}\}_{j=1}^m$ are drawn by using the delayed-sample method with delay Δ, whereas the importance weights $\{w_{t+\delta}^j\}_{j=1}^m$ are obtained after a further delay of δ. The coded BER performance of this adaptive receiver with different delays, together with that of the known channel lower bound, the genie-aided lower bound, and the differential detector, is plotted in Fig. 5. It is seen that unlike the uncoded case, for coded systems the delayed-sample method is very effective in improving the receiver performance. With a sample delay of $\Delta = 5$ and weight delay $\delta = 10$, the receiver performance is close to the genie-aided lower bound.

IX. Conclusions

We have developed a new adaptive Bayesian receiver for signal detection and decoding in flat-fading channels with known channel statistics based on the sequential Monte Carlo methodology. Specifically, we have derived adaptive receiver algorithms for both uncoded and coded systems, where the delayed-weight method, the delayed-sample method, and a combination of both are employed to improve estimation accuracy. The proposed sequential Monte Carlo receiver techniques can also handle the non-Gaussian ambient noise. The computational complexities of the various algorithms discussed in this paper are summarized in Table III. It is shown through simulations that the performance of the proposed sequential Monte Carlo receivers can be remarkably close to the so-called genie-aided lower bound in fading channels for both uncoded and coded systems without the use of any training/pilot symbols or decision feedback. Moreover, the proposed receiver structure exhibits massive parallelism and is ideally suited for high-speed parallel implementation using the VLSI systolic array technology. Future explorations of sequential Monte Carlo methods include the development of computationally efficient delayed-sample techniques, which will find wide applications in channels with strong memory (e.g., intersymbol interference channels [31]).

Appendix A
Proof of (19)

Note that

$$\begin{aligned} w_t &= w_{t-1} \cdot \frac{p(\boldsymbol{Z}_t|\boldsymbol{Y}_t)}{p(\boldsymbol{Z}_{t-1}|\boldsymbol{Y}_{t-1})q(\boldsymbol{z}_t|\boldsymbol{Z}_{t-1}, \boldsymbol{Y}_t)} \\ &\quad (\text{with } w_0 = 1) \\ &= \prod_{i=1}^{t} \frac{p(\boldsymbol{Z}_i|\boldsymbol{Y}_i)}{p(\boldsymbol{Z}_{i-1}|\boldsymbol{Y}_{i-1})q(\boldsymbol{z}_i|\boldsymbol{Z}_{i-1}, \boldsymbol{Y}_i)} \\ &= \frac{p(\boldsymbol{Z}_t|\boldsymbol{Y}_t)}{p(\boldsymbol{z}_0|\boldsymbol{y}_0)\prod_{i=1}^{t} q(\boldsymbol{z}_i|\boldsymbol{Z}_{i-1}, \boldsymbol{Y}_i)}. \end{aligned} \tag{74}$$

The numerator in (74) is the target distribution, and the denominator is the sampling distribution from which $\boldsymbol{Z}_t$ was generated. Hence, for any measurable function $h(\cdot)$, we have

$$\begin{aligned} E\left\{h\left(\boldsymbol{Z}_t^{(j)}\right) w_t^{(j)}\right\} &= \int h(\boldsymbol{Z}_t) \frac{p(\boldsymbol{Z}_t|\boldsymbol{Y}_t)}{p(\boldsymbol{z}_0|\boldsymbol{y}_0)\prod_{i=1}^{t} q(\boldsymbol{z}_i|\boldsymbol{Z}_{i-1}, \boldsymbol{Y}_i)} \\ &\quad \cdot \left[p(\boldsymbol{z}_0|\boldsymbol{y}_0)\prod_{i=1}^{t} q(\boldsymbol{z}_i|\boldsymbol{Z}_{i-1}, \boldsymbol{Y}_i)\right] d\boldsymbol{Z}_t \\ &= \int h(\boldsymbol{Z}_t)\, p(\boldsymbol{Z}_t|\boldsymbol{Y}_t)\, d\boldsymbol{Z}_t \\ &= E\left\{h(\boldsymbol{Z}_t)|\boldsymbol{Y}_t\right\}. \end{aligned} \tag{75}$$

Finally, note that both (19) and (20) are special cases of (75).

Appendix B
Proof That (40) and (41) Give Properly Weighted Samples

To show that the sample $(s_t^{(j)}, w_t^{(j)})$ given by (40) and (41) is a properly weighted sample with respect to $p(\boldsymbol{S}_t|\boldsymbol{Y}_t)$, we need to verify that (41) gives the correct weight. Assume that at time $(t-1)$, we have a properly weighted sample $(s_{t-1}^{(j)}, w_{t-1}^{(j)})$ with respect to $p(\boldsymbol{S}_{t-1}|\boldsymbol{Y}_{t-1})$. That is, assume that $s_{t-1}^{(j)}$ is drawn from some trial distribution $q(\boldsymbol{S}_{t-1}|\boldsymbol{Y}_{t-1})$, and that importance weight is given by $w_{t-1}^{(j)} = \omega_{t-1}(\boldsymbol{S}_{t-1}^{(j)}|\boldsymbol{Y}_{t-1})$, with

$$\omega_{t-1}(\boldsymbol{S}_{t-1}|\boldsymbol{Y}_{t-1}) \triangleq \frac{p(\boldsymbol{S}_{t-1}|\boldsymbol{Y}_{t-1})}{q(\boldsymbol{S}_{t-1}|\boldsymbol{Y}_{t-1})}. \tag{76}$$

By (38) and (40), $s_t^{(j)}$ is drawn from the distribution $p(s_t|\boldsymbol{S}_{t-1}^{(j)}, \boldsymbol{Y}_t)$. Hence, the sampling distribution for $\boldsymbol{S}_t^{(j)}$ is given by $q(\boldsymbol{S}_{t-1}|\boldsymbol{Y}_{t-1})\; p(s_t|\boldsymbol{S}_{t-1}, \boldsymbol{Y}_t)$. Since the target

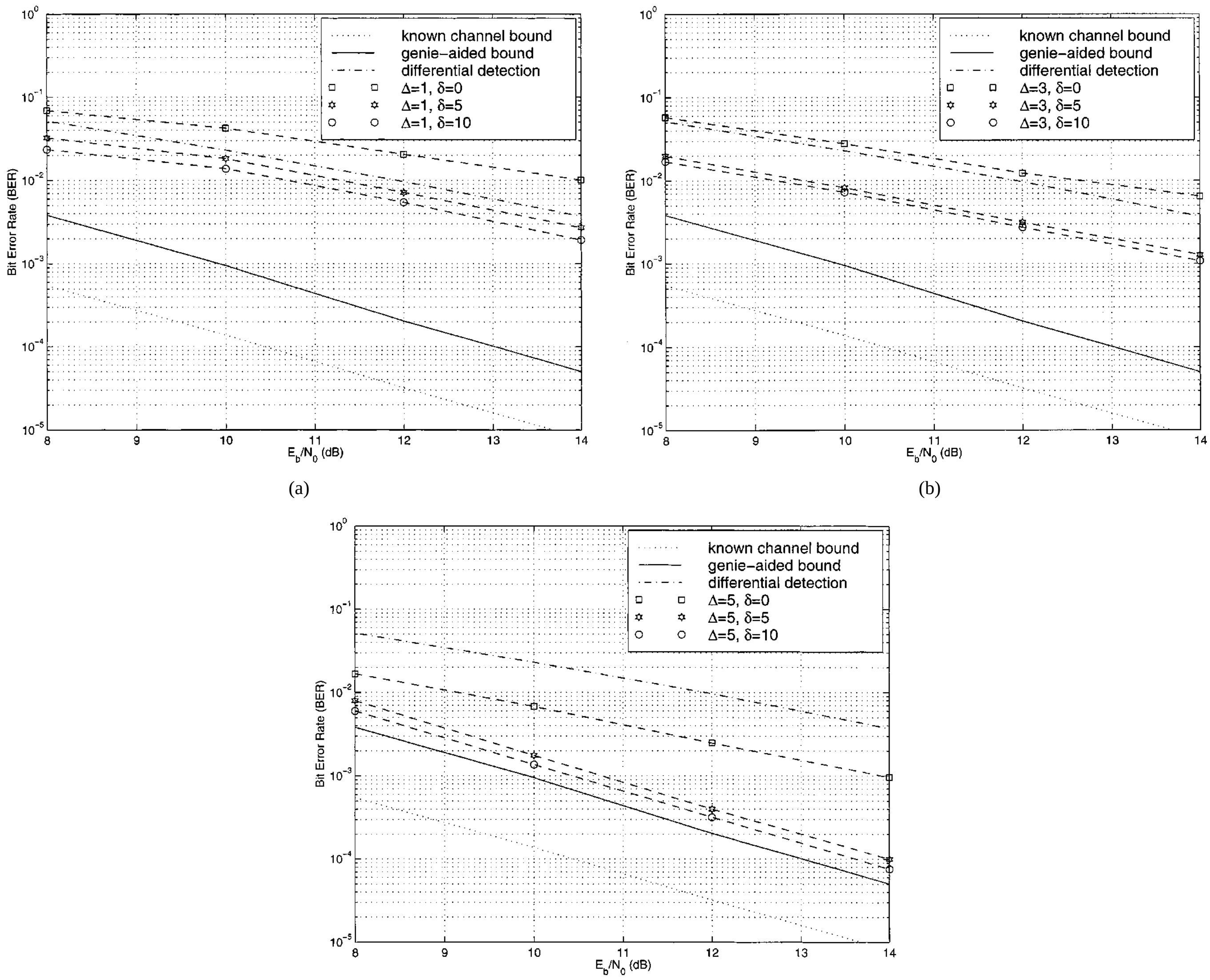

Fig. 5. BER performance of the sequential Monte Carlo receiver in a fading channel with Gaussian noise for a convolutionally coded system. The convolutional code has rate $1/2$ and constraint length five. A combination of delayed-sample (with delay Δ) and delayed-weight (with delay δ) method is used. The BER curves corresponding to delays $\Delta = 1$, $\Delta = 3$, and $\Delta = 5$ are shown. Also shown in the same figure are the BER curves for the known channel lower bound, the genie-aided lower bound, and the differential detector.

distribution is $p(\boldsymbol{S}_t|\boldsymbol{Y}_t)$, the weight function at time t is then give by

$$\begin{aligned}
&\omega_t(\boldsymbol{S}_t|\boldsymbol{Y}_t)\\
&\quad= \frac{p(\boldsymbol{S}_t|\boldsymbol{Y}_t)}{q(\boldsymbol{S}_{t-1}|\boldsymbol{Y}_{t-1})\, p(s_t|\boldsymbol{S}_{t-1}, \boldsymbol{Y}_t)}\\
&\quad= \frac{p(\boldsymbol{S}_{t-1}|\boldsymbol{Y}_{t-1})}{q(\boldsymbol{S}_{t-1}|\boldsymbol{Y}_{t-1})} \cdot \frac{p(\boldsymbol{S}_{t-1}|\boldsymbol{Y}_t)}{p(\boldsymbol{S}_{t-1}|\boldsymbol{Y}_{t-1})}\\
&\quad= \omega_{t-1}(\boldsymbol{S}_{t-1}|\boldsymbol{Y}_{t-1})\\
&\qquad \cdot \frac{p(y_t|\boldsymbol{Y}_{t-1}, \boldsymbol{S}_{t-1})\, p(\boldsymbol{Y}_{t-1}|\boldsymbol{S}_{t-1})\, p(\boldsymbol{S}_{t-1})/p(\boldsymbol{Y}_t)}{p(\boldsymbol{Y}_{t-1}|\boldsymbol{S}_{t-1})\, p(\boldsymbol{S}_{t-1})/p(\boldsymbol{Y}_{t-1})}\\
&\quad\propto \omega_{t-1}(\boldsymbol{S}_{t-1}|\boldsymbol{Y}_{t-1}) \cdot p(y_t|\boldsymbol{Y}_{t-1}, \boldsymbol{S}_{t-1})\\
&\quad= \omega_{t-1}(\boldsymbol{S}_{t-1}|\boldsymbol{Y}_{t-1})\\
&\qquad \cdot \sum_{a_i \in \mathcal{A}} p(y_t|s_t = a_i, \boldsymbol{S}_{t-1}, \boldsymbol{Y}_{t-1})\\
&\qquad \cdot p(s_t = a_i|\boldsymbol{S}_{t-1}, \boldsymbol{Y}_{t-1})\\
&\quad= \omega_{t-1}(\boldsymbol{S}_{t-1}|\boldsymbol{Y}_{t-1}) \cdot \sum_{a_i \in \mathcal{A}} \rho_{t,i}.
\end{aligned} \tag{77}$$

Hence

$$w_t^{(j)} = \omega_t(\boldsymbol{S}_t^{(j)}|\boldsymbol{Y}_t) = w_{t-1}^{(j)} \sum_{a_i \in \mathcal{A}} \rho_{t,i}^{(j)}.$$

This verifies that (41) gives the correct importance weight at time t.

Appendix C

In this appendix we verify the correctness of the residual resampling under a general setting. Let $(x_t^{(j)}, w_t^{(j)})$ be a properly weighted sample with respect to $p(x_t|Y_t)$. Without loss of generality, we assume that $\sum_{j=1}^m w_t^{(j)} = m$. Let $\{\tilde{x}_t^{(j)}\}_{j=1}^m$ be the set of samples generated from the residual resampling scheme. The new set consists of $k_j = \lfloor w_t^{(j)} \rfloor$ copies of the sample $x_t^{(j)}$ for $j = 1, \ldots, m$, and $K_r = m - \sum_{j=1}^m k_j$ i.i.d. samples drawn from set $\{x_t^{(j)}\}_{j=1}^m$ with probability proportional to

TABLE III

THE COMPUTATIONAL COMPLEXITIES OF THE PROPOSED SEQUENTIAL MONTE CARLO RECEIVER ALGORITHMS UNDER DIFFERENT CONDITIONS IN TERMS OF THE NUMBER OF ONE-STEP KALMAN FILTER UPDATES NEEDED AT EACH TIME t. THE DEGREE OF PARALLELISM REFERS TO THE MAXIMUM NUMBER OF COMPUTING UNITS THAT CAN BE EMPLOYED TO IMPLEMENT THE ALGORITHM IN PARALLEL. IT IS ASSUMED THAT THE DELAYED-SAMPLE METHOD IS USED WITH A DELAY OF Δ TIME UNITS. THE NUMBER OF SAMPLES DRAWN AT EACH TIME IS m. FOR UNCODED SYSTEM, THE CARDINALITY OF THE SYMBOL ALPHABET IS $|\mathcal{A}|$. FOR CODED SYSTEM, A $\frac{k_0}{n_0}$ CONVOLUTIONAL CODE IS USED. THE NON-GAUSSIAN NOISE IS MODELED BY A TWO-TERM GAUSSIAN MIXTURE

	Uncoded system		Coded system	
	Complexity	Deg. of Parallelism	Complexity	Deg. of Parallelism
Gaussian	$m\,\vert\mathcal{A}\vert^{\Delta}$	$m\,\vert\mathcal{A}\vert^{\Delta}$	$m\,n_0\,2^{k_0(\Delta+1)}$	$m\,2^{k_0(\Delta+1)}$
non-Gaussian	$m\,(2\vert\mathcal{A}\vert)^{\Delta}$	$m\,(2\vert\mathcal{A}\vert)^{\Delta}$	$m\,n_0\,2^{(k_0+n_0)(\Delta+1)}$	$m\,2^{(k_0+n_0)(\Delta+1)}$

$(w_t^{(j)} - \lfloor w_t^{(j)} \rfloor)/K_r$. The weights for the new samples are set to 1. Hence

$$
\begin{aligned}
E&\left[\frac{1}{m}\sum_{j=1}^{m} h\left(\tilde{x}_t^{(j)}\right)\right]\\
&= E\left\{E\left[\frac{1}{m}\sum_{j=1}^{m} h\left(\tilde{x}_t^{(j)}\right) \middle| \left\{x_t^{(j')},\, w_t^{(j')}\right\}_{j'=1}^{m}\right]\right\}\\
&= \frac{1}{m}E\left\{\sum_{j=1}^{m} h\left(x_t^{(j)}\right)\left\lfloor w_t^{(j)}\right\rfloor + \sum_{j=m-K_r+1}^{m}\right.\\
&\qquad \left. \cdot E\left[h\left(\tilde{x}_t^{(j)}\right) \middle| \left\{x_t^{(j')},\, w_t^{(j')}\right\}_{j'=1}^{m}\right]\right\}\\
&= \frac{1}{m}E\left\{\sum_{j=1}^{m} h\left(x_t^{(j)}\right)\left\lfloor w_t^{(j)}\right\rfloor + K_r E\right.\\
&\qquad \left. \cdot \left[h(\tilde{x}_t) \middle| \left\{x_t^{(j')},\, w_t^{(j')}\right\}_{j'=1}^{m}\right]\right\}\\
&= \frac{1}{m}E\left\{\sum_{j=1}^{m} h\left(x_t^{(j)}\right)\left\lfloor w_t^{(j)}\right\rfloor\right.\\
&\qquad \left. + K_r \sum_{j=1}^{m} h\left(x_t^{(j)}\right)\frac{w_t^{(j)} - \left\lfloor w_t^{(j)}\right\rfloor}{K_r}\right\}\\
&= \frac{1}{m}E\left[\sum_{j=1}^{m} h\left(x_t^{(j)}\right) w_t^{(j)}\right] = E[h(x_t)|Y_t].
\end{aligned}
\tag{78}
$$

Furthermore,

$$
\begin{aligned}
\mathrm{Var}&\left[\frac{1}{m}\sum_{j=1}^{m} h\left(\tilde{x}_t^{(j)}\right)\right]\\
&= \mathrm{Var}\left\{E\left[\frac{1}{m}\sum_{j=1}^{m} h\left(\tilde{x}_t^{(j)}\right) \middle| \left\{x_t^{(j')},\, w_t^{(j')}\right\}_{j'=1}^{m}\right]\right\}\\
&\quad + E\left\{\mathrm{Var}\left[\frac{1}{m}\sum_{j=1}^{m} h\left(\tilde{x}_t^{(j)}\right) \middle| \left\{x_t^{(j')},\, w_t^{(j')}\right\}_{j'=1}^{m}\right]\right\}\\
&= \mathrm{Var}\left\{\frac{1}{m}\sum_{j=1}^{m} h\left(x_t^{(j)}\right) w_t^{(j)}\right\}\\
&\quad + E\left\{\frac{K_r}{m^2}\mathrm{Var}\left[h(\tilde{x}_t) \middle| \left\{x_t^{(j')},\, w_t^{(j')}\right\}_{j'=1}^{m}\right]\right\}\\
&\le \frac{1}{m}\mathrm{Var}[h(x_t)w_t]\\
&\quad + E\left[\frac{K_r}{m^2}\sum_{j=1}^{m}\left(h\left(x_t^{(j)}\right)\right)^2 \frac{\left(w_t^{(j)} - \left\lfloor w_t^{(j)}\right\rfloor\right)}{K_r}\right]\\
&\le \frac{1}{m}\mathrm{Var}[h(x_t)w_t] + \frac{1}{m^2}\\
&\quad \cdot E\left[\sum_{j=1}^{m}\left(h\left(x_t^{(j)}\right)\right)^2 \min\left\{1,\, w_t^{(j)}\right\}\right]\\
&\le \frac{1}{m}\mathrm{Var}[h(x_t)w_t] + \frac{1}{m}\\
&\quad \cdot E[(h(x_t))^2 w_t] \to 0, \qquad \text{as } m \to \infty.
\end{aligned}
\tag{79}
$$

Here we assume that $\mathrm{Var}[h(x_t)w_t] < \infty$. Hence

$$\frac{1}{m}\sum_{j=1}^{m} h(\tilde{x}_t^{(j)}) \to E(h(x_t)|\boldsymbol{Y}_t)$$

in probability.

REFERENCES

[1] K. L. Blackard, T. S. Rappaport, and C. W. Bostian, “Measurements and models of radio frequency impulsive noise for indoor wireless communications,” *IEEE J. Select. Areas Commun.*, vol. 11, pp. 991–1001, Sept. 1993.

[2] T. K. Blankenship, D. M. Krizman, and T. S. Rappaport, “Measurements and simulation of radio frequency impulsive noise in hospitals and clinics,” in *Proc. 1997 IEEE Vehicular Technology Conf. (VTC’97)*, 1997, pp. 1942–1946.

[3] 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.

[4] R. Chen and J. S. Liu, “Mixture Kalman filters,” *J. Roy. Statist. Soc. (B)*, 2000, to be published.

[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 Processing*, 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 Processing*, 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, Jan. 1995.

[8] C. N. Georghiades 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.

[9] M. J. Gertsman and J. H. Lodge, "Symbol-by-symbol MAP demodulation of CPM and PSK signals on Rayleigh flad fading channels," *IEEE Trans. Commun.*, vol. 45, pp. 788–799, July 1997.

[10] N. J. Gordon, D. J. Salmon, and A. F. M. Smith, "A novel approach to nonlinear/non-Gaussian Bayesian state estimation," *IEE Proc. Radar and Signal Processing*, vol. 140, pp. 107–113, 1993.

[11] R. Haeb and H. Meyr, "A systematic approach to carrier recovery and detection of digitally phase modulxated signals on fading channels," *IEEE Trans. Commun.*, vol. 37, pp. 748–754, July 1989.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] S. Y. Kung, *VLSI Array Processing*. Englewood Cliffs, NJ: Prentice Hall, 1988.

[18] J. S. Liu and R. Chen, "Blind deconvolution via sequential imputations," *J. Amer. Statist. Assoc.*, vol. 90, pp. 567–576, 1995.

[19] ——, "Sequential Monte Carlo methods for dynamic systems," *J. Amer. Statist. Assoc.*, vol. 93, pp. 1032–1044, 1998.

[20] Y. Liu and S. D. Blostein, "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.

[21] 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.

[22] 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 envelop, multiple differential and coherent detection approach," *IEEE Trans. Commun.*, vol. 42, pp. 63–75, Jan. 1994.

[23] C. J. Masreliez, "Approximate non-Gaussian filtering with linear state and observation relations," *IEEE Trans. Automat. Contr.*, vol. AC-20, pp. 107–110, Jan. 1975.

[24] D. Middleton, "Man-made noise in urban environments and transportation systems: Models and measurement," *IEEE Trans. Commun.*, vol. COM-21, pp. 1232–1241, Nov. 1973.

[25] ——, "Statistical-physical models of electromagnetic interference," *IEEE Trans. Electromag. Compat.*, vol. EMC-19, pp. 106–127, 1977.

[26] ——, "Channel modeling and threshold signal processing in underwater acoustics: An analytical overview," *IEEE J. Oceanic Eng.*, vol. OE-12, pp. 4–28, 1987.

[27] ——, "Non-Gaussian noise models in signal processing for telecommunications: New methods and results for class A and class B noise models," *IEEE Trans. Inform. Theory*, vol. 45, pp. 1122–1129, May 1999.

[28] D. Middleton and A. D. Spaulding, "Elements of weak signal detection in non-Gaussian noise," in *Advances in Statistical Signal Processing Vol. 2: Signal Detection*, H. V. Poor and J. B. Thomas, Eds. Greenwich, CT: JAI, 1993.

[29] 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.

[30] R. Raheli, A. Polydoros, and C. Tzou, "Per-survivor processing: A general approach to MLSE in uncertain environments," *IEEE Trans. Commun.*, vol. 43, pp. 354–364, Feb./Mar./Apr. 1995.

[31] J. Thielecke, "A soft-decision state-space equalizer for FIR channels," *IEEE Trans. Commun.*, vol. 45, pp. 1208–1217, Oct. 1997.

[32] 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.

[33] X. Wang and H. V. Poor, "Joint channel estimation and symbol detection in Rayleigh flat fading channels with impulsive noise," *IEEE Commun. Lett.*, vol. 1, pp. 19–21, Jan. 1997.

[34] S. M. Zabin and H. V. Poor, "Efficient estimation of the class A parameters via the EM algorithm," *IEEE Trans. Inform. Theory*, vol. 37, pp. 60–72, Jan. 1991.