Journal of VLSI Signal Processing 30, 89–105, 2002

Monte Carlo Bayesian Signal Processing for Wireless Communications*

XIAODONG WANG
Electrical Engineering Department, Texas A&M University, College Station, TX 77843, USA

RONG CHEN
Information and Decision Science Department, University of Illinois at Chicago, Chicago, IL 60607, USA

JUN S. LIU
Statistics Department, Harvard University, Cambridge, MA 02138, USA

Received April 4, 2000; Accepted June 6, 2001

Abstract. Many statistical signal processing problems found in wireless communications involves making inference about the transmitted information data based on the received signals in the presence of various unknown channel distortions. The *optimal* solutions to these problems are often too computationally complex to implement by conventional signal processing methods. The recently emerged Bayesian Monte Carlo signal processing methods, the relatively simple yet extremely powerful numerical techniques for Bayesian computation, offer a novel paradigm for tackling wireless signal processing problems. These methods fall into two categories, namely, Markov chain Monte Carlo (MCMC) methods for batch signal processing and sequential Monte Carlo (SMC) methods for adaptive signal processing. We provide an overview of the theories underlying both the MCMC and the SMC. Two signal processing examples in wireless communications, the blind turbo multiuser detection in CDMA systems and the adaptive detection in fading channels, are provided to illustrate the applications of MCMC and SMC respectively.

1. Introduction

The projections of rapidly escalating demand for advanced wireless services such as multimedia present major challenges to researchers and engineers in the field of telecommunications. Meeting these challenges will require sustained technical innovations on many fronts. Of paramount importance in addressing these challenges is the development of suitable receivers for wireless multiple-access communications in non-stationary and interference-rich environments. While considerable previous work has addressed many aspects of this problem separately, e.g., single-user channel equalization, interference suppression for multiple-access channels, and tracking of time-varying channels, to name a few, the demand for a unified approach to addressing the problem of *jointly* combating the various impairment in wireless channels has only recently become significant. This is due in part to the fact that practical high data-rate multiple-access systems are just beginning to emerge in the market place. Moreover, most of the proposed receiver design solutions are suboptimal techniques whose performances are still far below the performance achieved by the theoretically optimal procedures. On the other hand, the optimal solutions mostly can not be implemented in practice because of their prohibitively high computational complexities.

The Bayesian Monte Carlo methodologies recently emerged in statistics have provided a promising new

*This work was supported in part by the U.S. National Science Foundation (NSF) under grants CCR-9980599, DMS-0073651, DMS-0073601 and DMS-0094613.

paradigm for the design of low-complexity signal processing techniques with performance approaching the theoretical optimum for fast and reliable communication in the highly severe and dynamic wireless environment. Over the past decade, a large body of methods in the field of statistics has emerged based on iterative Monte Carlo techniques and they are especially useful for computing the Bayesian solutions to the optimal signal reception problems encountered in wireless communications. These powerful statistical tools, when employed in the signal processing engines of the digital receivers in wireless networks, hold the potential of closing the giant gap between the performance of the current state-of-art wireless receivers and the ultimate optimal performance predicted by statistical communication theory. This advance will not only strongly influence the development of theory and practice of wireless communications and signal processing, but also yield tremendous technological and commercial impacts on the telecommunication industry.

In this paper, we provide an overview of the theories and applications of Monte Carlo signal processing methods. These methods in general fall into two categories, namely, Markov chain Monte Carlo (MCMC) methods for batch signal processing, and sequential Monte Carlo (SMC) methods for adaptive signal processing. For each category, we outline the general theory and provide a signal processing example in wireless communications to illustrate its application. The rest of this paper is organized as follows. In Section 2, we discuss the general optimal signal processing problem under the Bayesian framework; In Section 3, we describe the MCMC signal processing techniques; In Section 4, we describe the SMC signal processing methods. We conclude the article in Section 5.

2. Bayesian Signal Processing

2.1. The Bayesian Framework

A typical statistical signal processing problem can be stated as follows: Given some observations $\boldsymbol{Y} \triangleq [\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n]$, we want to make statistical inference about some unknown quantities $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m]$. Typically the observations $\boldsymbol{Y}$ are functions of $\boldsymbol{X}$ and some unknown "nuisance" parameters, $\boldsymbol{\Theta} = [\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_l]$, plus some noise. The following examples are three well-known wireless signal processing problems.

Example 1 (Equalization). Suppose we want to transmit binary symbols $x_1, x_2, \ldots, x_n$, $x_i \in \{+1, -1\}$, through a bandlimitted communication channel whose input-output relationship is given by

$$y_t = \sum_{l=0}^{L-1} h_l x_{t-l} + v_t, \tag{1}$$

where $\{h_l\}_{l=0}^{L-1}$ represents the unknown complex channel response; $\{v_t\}_{t=1}^{n}$ are i.i.d. Gaussian noise samples, $v_t \sim \mathcal{N}_c(0, \sigma^2)$. The inference problem is to estimate the transmitted symbols $\boldsymbol{X} = \{x_t\}_{t=1}^{n}$, based on the received signals $\boldsymbol{Y} = \{y_t\}_{t=1}^{n}$. The nuisance parameters are $\boldsymbol{\Theta} = \{h_0, \ldots, h_{L-1}, \sigma^2\}$.

Example 2 (Multiuser Detection). In code-division multiple-access (CDMA) systems, multiple users transmit information symbols over the same physical channel through the use of distinct spreading waveforms. Suppose there are K users in the system. The k-th user employs a spreading waveform $\boldsymbol{s}_k$ of the form

$$\boldsymbol{s}_k = \frac{1}{\sqrt{N}} [s_{1,k}\ s_{2,k} \cdots s_{N,k}]^T, \quad s_{n,k} \in \{+1, -1\}, \tag{2}$$

and transmit binary information symbols $x_{k,1}, x_{k,2}, \ldots, x_{k,n}$, $x_{k,t} \in \{+1, -1\}$. The received signal in this system is given by

$$\boldsymbol{y}_t = \sum_{k=1}^{K} A_k x_{k,t} \boldsymbol{s}_t + \boldsymbol{v}_t, \tag{3}$$

where A_k is the unknown complex amplitude of the k-th user; $\{\boldsymbol{v}_t\}_{t=1}^{n}$ are i.i.d. Gaussian noise vectors, $\boldsymbol{v}_t \sim \mathcal{N}_c(\boldsymbol{0}, \sigma^2 \boldsymbol{I})$. The inference problem is to estimate the transmitted multiuser symbols $\boldsymbol{X} = \{\{x_{k,t}\}_{t=1}^{n}\}_{k=1}^{K}$, based on the received signals $\boldsymbol{Y} = \{\boldsymbol{y}_t\}_{t=1}^{n}$. The nuisance parameters are $\boldsymbol{\Theta} = \{A_1, \ldots, A_k, \sigma^2\}$.

Example 3 (Fading Channel). Suppose we want to transmit binary symbols $x_1, \ldots, x_n$, $x_t \in \{+1, -1\}$, through a fading channel whose input-output relationship is given by

$$y_t = \alpha_t x_t + v_t, \tag{4}$$

where $\{\alpha_t\}_{t=1}^{n}$ represents the unknown Rayleigh fading process, which can be modeled as the output of a low-pass filter of order r driven by white Gaussian noise,

$$\{\alpha_t\} = \frac{\Psi(D)}{\Phi(D)} \{u_t\}, \tag{5}$$

where D is the back-shift operator $D^k u_t \triangleq u_{t-k}$; $\Phi(z) \triangleq \phi_r z^r + \cdots + \phi_1 z + 1$; $\Psi(z) \triangleq \psi_r z^r + \cdots + \psi_1 z + \psi_0$; and $\{u_t\}_{t=1}^n$ is a white complex Gaussian noise sequence with independent real and imaginary components, $u_t \sim \mathcal{N}_c(0, \sigma^2)$. The inference problem is to estimate the transmitted symbols $\boldsymbol{X} = \{x_t\}_{t=1}^n$, based on the received signals $\boldsymbol{Y} = \{y_t\}_{t=1}^n$. The nuisance parameters are $\boldsymbol{\Theta} = \{\alpha_1, \ldots, \alpha_n, \sigma^2\}$.

In Bayesian approach to a statistical signal processing problem, all unknown quantities are treated as random variables with some prior distribution $p(\boldsymbol{X}, \boldsymbol{\Theta})$. The Bayesian inference is made based on the joint posterior distribution of these unknowns

$$p(\boldsymbol{X}, \boldsymbol{\Theta} \mid \boldsymbol{Y}) \propto p(\boldsymbol{Y} \mid \boldsymbol{\Theta}, \boldsymbol{X})\, p(\boldsymbol{X}, \boldsymbol{\Theta}). \tag{6}$$

Note that typically the joint posterior distribution is completely known up to a normalizing constant. If we are interested in making inference about the i-th component $\boldsymbol{x}_i$ of $\boldsymbol{X}$, say, we need to compute $E\{h(\boldsymbol{x}_i) \mid \boldsymbol{Y}\}$ for some function $h(\cdot)$, then this is given by

$$\begin{aligned} E\{h(\boldsymbol{x}_i) \mid \boldsymbol{Y}\} &= \int h(\boldsymbol{x}_i)\, p(\boldsymbol{x}_i \mid \boldsymbol{Y})\, d\boldsymbol{x}_i \qquad (7)\\ &= \int h(\boldsymbol{x}_i) \int p(\boldsymbol{X}, \boldsymbol{\Theta} \mid \boldsymbol{Y})\, d\boldsymbol{X}_{[-i]}\, d\boldsymbol{\Theta}\, d\boldsymbol{x}_i, \qquad (8) \end{aligned}$$

where $\boldsymbol{X}_{[-i]} \triangleq \boldsymbol{X} \setminus \boldsymbol{x}_i$. Neither of these computations is easy in practice.

2.2. Batch Processing versus Adaptive Processing

Depending on how the data are processed and the inference is made, most signal processing methods fall into one of the two categories: batch processing and adaptive (i.e., sequential) processing. In batch signal processing, the entire data block $\boldsymbol{Y}$ is received and stored before it is processed; and the inference about $\boldsymbol{X}$ is made based on the entire data block $\boldsymbol{Y}$. In adaptive processing, however, inference is made sequentially (i.e., on-line) as the data being received. For example, at time t, after a new sample $\boldsymbol{y}_t$ is received, then an update on the inference about some or all elements of $\boldsymbol{X}$ is made. In this paper, we focus on the *optimal* signal processing under the Bayesian framework for both batch processing and adaptive processing. We next illustrate the batch and adaptive Bayesian signal processings, respectively, by two examples.

Example 2 (continued). Consider the multiuser detection problem in Section 2.1. Denote $\boldsymbol{S} = [\boldsymbol{s}_1, \ldots, \boldsymbol{s}_K]$, $\boldsymbol{A} = \mathrm{diag}\,(A_1, \ldots, A_K)$ and $\boldsymbol{x}_t = [x_{1,t}, \ldots, x_{K,t}]^T$. Then (3) can be written as

$$\boldsymbol{y}_t = \boldsymbol{S}\boldsymbol{A}\boldsymbol{x}_t + \boldsymbol{v}_t. \tag{9}$$

The optimal *batch* processing procedure for this problem is as follows. Let $\boldsymbol{a} = [A_1, \ldots, A_K]^T$. Assume that the unknown quantities $\boldsymbol{a}$, σ^2 and $\boldsymbol{X}$ are independent of each other and have prior distributions $p(\boldsymbol{a})$, $p(\sigma^2)$ and $p(\boldsymbol{X})$, respectively. Since $\{\boldsymbol{v}_t\}_{t=1}^n$ is a sequence of independent Gaussian vectors, the joint posterior distribution of these unknown quantities $(\boldsymbol{a}, \sigma^2, \boldsymbol{X})$ based on the received signal $\boldsymbol{Y}$ takes the form of

$$\begin{aligned} p(\boldsymbol{a}, \sigma^2, \boldsymbol{X} \mid \boldsymbol{Y}) &= p(\boldsymbol{Y} \mid \boldsymbol{a}, \sigma^2, \boldsymbol{X})\, p(\boldsymbol{a})\, p(\sigma^2) \\ &\qquad \times p(\boldsymbol{X})/p(\boldsymbol{Y}) \qquad (10)\\ &\propto \left(\frac{1}{\sigma^2}\right)^{\frac{Nn}{2}} \exp\left(-\frac{1}{2\sigma^2} \sum_{t=1}^{n} \|\boldsymbol{y}_t - \boldsymbol{S}\boldsymbol{A}\boldsymbol{x}_t\|^2\right) \\ &\qquad \times p(\boldsymbol{a}) p(\sigma^2)\, p(\boldsymbol{X}). \qquad (11) \end{aligned}$$

The a posteriori probabilities of the transmitted symbols can then be calculated from the joint posterior distribution (11) according to

$$\begin{aligned} P[x_{k,t} = +1 \mid \boldsymbol{Y}] &= \sum_{\boldsymbol{X}: x_{k,t} = +1} p(\boldsymbol{X} \mid \boldsymbol{Y}) \qquad (12)\\ &= \sum_{\boldsymbol{X}: x_{k,t} = +1} \int p(\boldsymbol{a}, \sigma^2, \boldsymbol{X} \mid \boldsymbol{Y})\, d\boldsymbol{a}\, d\sigma^2. \qquad (13) \end{aligned}$$

Clearly the computation in (13) involves 2^{Kn-1} multidimensional integrals, which is certainly infeasible for any practical implementations.

Example 3 (continued). Consider the fading channel problem with optimal *adaptive* processing. System (4) and (5) can be rewritten in the state-space model form, which is instrumental in developing the sequential signal processing algorithm. Define

$$\{z_t\} \triangleq \Psi^{-1}(D)\{\alpha_t\} \Rightarrow \Phi(D)\{z_t\} = \{u_t\}. \tag{14}$$

Denote $\boldsymbol{z}_t \triangleq [z_t, \ldots, z_{t-r+1}]^T$. By (5) we then have

$$\boldsymbol{z}_t = \boldsymbol{F}\boldsymbol{z}_{t-1} + \boldsymbol{g}u_t, \qquad u_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}_c(0, 1), \tag{15}$$

where

$$\boldsymbol{F} \triangleq \begin{pmatrix} -\phi_1 & -\phi_2 & \cdots & -\phi_r & 0 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix},$$

$$\text{and} \quad \boldsymbol{g} \triangleq \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

Because of (14), the fading coefficient sequence $\{\alpha_t\}$ can be written as

$$\alpha_t = \boldsymbol{h}^H \boldsymbol{z}_t, \quad \text{where} \quad \boldsymbol{h} \triangleq [\psi_0 \quad \psi_1 \cdots \psi_r]^H. \tag{16}$$

Then we have the following state-space model for the system defined by (4) and (5):

$$\boldsymbol{z}_t = \boldsymbol{F}\boldsymbol{z}_{t-1} + \boldsymbol{g}u_t, \tag{17}$$

$$y_t = x_t \boldsymbol{h}^H \boldsymbol{z}_t + v_t. \tag{18}$$

We now look at the problem of on-line estimation of the symbol x_t based on the received signals up to time t, $\{y_i\}_{i=1}^t$. This problem is the one of making Bayesian inference with respect to the posterior distribution

$$\begin{aligned} & p(z_1, \ldots, z_t, x_1, \ldots, x_t \mid y_1, \ldots, y_t) \\ & \quad \propto \prod_{j=1}^{t} p(z_j \mid \boldsymbol{z}_{j-1}) p(y_j \mid \boldsymbol{z}_j, x_j) \\ & \propto \prod_{j=1}^{t} \exp\left(- \left\| z_j + \sum_{i=1}^{r} \phi_i z_{j-i} \right\|^2 \right) \\ & \times \exp\left(-\frac{1}{\sigma^2} \left\| y_j - x_j \boldsymbol{h}^H \boldsymbol{z}_j \right\|^2 \right), \quad t = 1, 2, \ldots. \end{aligned} \tag{19}$$

For example, an on-line symbol estimation can be obtained from the marginal posterior distribution

$$p(x_t \mid y_1, \ldots, y_t) = \sum_{x_1 \in \{+1,-1\}} \cdots \sum_{x_{t-1} \in \{+1,-1\}} \int \cdots \int p(z_1, \ldots, z_t, x_1, \ldots, x_t \mid y_1, \ldots, y_t)\, dz_1 \cdots dz_t. \tag{20}$$

Again we see that direct implementation of the optimal sequential Bayesian inference is computationally prohibitive.

It is seen from the above discussions that although the Bayesian procedures achieve the optimal performance in terms of the minimum mean squared error on symbol detections, they exhibit prohibitively high computational complexity and thus are not implementable in practice. Various suboptimal solutions have been proposed to tackle these signal processing problems. For example, for the equalization problem, one may first send a training sequence to estimate the channel response; symbol detection can then be carried out based on the estimated channel [1]. For the multiuser detection problem, one may apply linear or nonlinear interference cancelation methods [2]. And for the fading channel problem, one may use a combination of training and decision-directed approach to estimate the fading process, based on which symbol detection can be made [3]. Although these suboptimal approaches provide reasonable performance, it is far from the best attainable performance. Moreover, the use of training sequence in a communication system incurs a significant loss in spectral efficiency. The recently developed Monte Carlo methods for Bayesian computation have provided a viable approach to solving many optimal signal processing problems (such as the ones mentioned above) at a reasonable computational cost.

2.3. Monte Carlo Methods

In many Bayesian analyses, the computation involved in eliminating the nuisance parameters and missing data is so difficult that one has to resort to some analytical or numerical approximations. These approximations were often case-specific and were the bottleneck that prevented the Bayesian method from being widely used. In late 1980s and early 1990s, statisticians discovered that a wide variety of Monte Carlo strategies can be applied to overcome the computational difficulties encountered in almost all likelihood-based inference procedures. Soon afterwards, this “rediscovery” of the Monte Carlo method as one of the most versatile and powerful computational tools began to invade other quantitative fields such as artificial intelligence, computational biology, engineering, financial modeling, etc. [4].

Suppose we can generate random samples (either independent or dependent)

$$\left(\boldsymbol{X}^{(1)}, \boldsymbol{\Theta}^{(1)}\right), \left(\boldsymbol{X}^{(1)}, \boldsymbol{\Theta}^{(1)}\right), \ldots, \left(\boldsymbol{X}^{(\nu)}, \boldsymbol{\Theta}^{(\nu)}\right),$$

from the joint distribution (6). We can approximate the marginal distribution $p(\boldsymbol{x}_i \mid \boldsymbol{Y})$ by the empirical distribution (i.e., the histogram) based on $\boldsymbol{x}_i^{(1)}, \boldsymbol{x}_i^{(2)}, \ldots, \boldsymbol{x}_i^{(\nu)}$, the component of $\boldsymbol{x}_i$ in $\boldsymbol{X}^{(j)}$, and approximate the posterior mean (8) by

$$E\{h(\boldsymbol{x}_i) \mid \boldsymbol{Y}\} \cong \frac{1}{\nu}\sum_{j=1}^{\nu} h\big(\boldsymbol{x}_i^{(j)}\big). \tag{21}$$

Most Monte Carlo techniques fall into one of the following two categories, Markov chain Monte Carlo (MCMC) methods, corresponding to batch processing, and sequential Monte Carlo (SMC) methods, corresponding to adaptive processing.

3. Markov Chain Monte Carlo Signal Processing

3.1. General Markov Chain Monte Carlo Algorithms

Markov chain Monte Carlo (MCMC) is a class of algorithms that allow one to draw (pseudo-) random samples from an arbitrary target probability distribution, $p(\boldsymbol{x})$, known up to a normalizing constant. The basic idea behind these algorithms is that one can achieve the sampling from p by running a Markov chain whose equilibrium distribution is exactly p. Two basic types of MCMC algorithms, the Metropolis algorithm and the Gibbs sampler, have been widely used in diverse fields. The validity of the both algorithms can be proved by the basic Markov chain theory.

3.1.1. Metropolis–Hastings Algorithm. Let $p(\boldsymbol{x}) = c\exp\{-f(\boldsymbol{x})\}$ be the target probability distribution from which we want to simulate random draws. The normalizing constant c may be unknown to us. Metropolis et al. [5] first introduced the fundamental idea of evolving a Markov process in Monte Carlo sampling. Their algorithm is as follows. Starting with any configuration $\boldsymbol{x}^{(0)}$, the algorithm evolves from the current state $\boldsymbol{x}^{(t)} = \boldsymbol{x}$ to the next state $\boldsymbol{x}^{(t+1)}$ as follows:

Algorithm 1 (*Metropolis Algorithm—Form I*)**.**

1. *A small, random, and "symmetric" perturbation of the current configuration is made. More precisely, $\boldsymbol{x}'$ is generated from a symmetric proposal function $T(\boldsymbol{x}^{(t)} \to \boldsymbol{x}')$ [i.e., $\sum_{\boldsymbol{y}} T(\boldsymbol{x} \to \boldsymbol{y}) = 1$ for all $\boldsymbol{x}$ and $T(\boldsymbol{x} \to \boldsymbol{x}') = T(\boldsymbol{x}' \to \boldsymbol{x})$].*
2. *The "gain" (or loss) of an objective function (corresponding to $\log p(\boldsymbol{x}) = f(\boldsymbol{x})$) resulting from this perturbation is computed.*
3. *A random number u is generated independently.*
4. *The new configuration is accepted if $\log(u)$ is smaller than or equal to the "gain" and rejected otherwise.*

Heuristically, this algorithm is constructed based on a "trial-and-error" strategy. Metropolis et al. restricted their choice of the "perturbation" function to be the symmetric ones, i.e., Intuitively, this means that there is no "flow bias" at the proposal stage. Hastings generalized the choice of T to all those that satisfies the property: $T(\boldsymbol{x} \to \boldsymbol{x}') > 0$ if and only if $T(\boldsymbol{x}' \to \boldsymbol{x}) > 0$ [6]. This generalized form can be simply stated as follows. At each iteration:

Algorithm 2 (*Metropolis–Hastings Algorithm—Form II*)**.**

1. *Propose a random "perturbation" of the current state, i.e., $\boldsymbol{x} \to \boldsymbol{x}'$, where $\boldsymbol{x}'$ is generated from a transition function $T(\boldsymbol{x}^{(t)} \to \boldsymbol{x}')$, which is nearly arbitrary (of course, some are better than others in terms of efficiency) and is completely specified by the user.*
2. *Compute the Metropolis ratio*

$$r(\boldsymbol{x}, \boldsymbol{x}') = \frac{p(\boldsymbol{x}')T(\boldsymbol{x}' \to \boldsymbol{x})}{p(\boldsymbol{x})T(\boldsymbol{x} \to \boldsymbol{x}')}. \tag{22}$$

3. *Generate a random number $u \sim$ uniform (0,1). Let $\boldsymbol{x}^{(t+1)} = \boldsymbol{x}'$ if $u \le r(\boldsymbol{x}, \boldsymbol{x}')$, and let $\boldsymbol{x}^{(t+1)} = \boldsymbol{x}^{(t)}$ otherwise.*

It is easy to prove that the M–H transition rule results in an "actual" transition function $A(\boldsymbol{x}, \boldsymbol{y})$ (it is different from T because a acceptance/rejection step is involved) that satisfies the detailed balance condition

$$p(\boldsymbol{x})A(\boldsymbol{x}, \boldsymbol{y}) = p(\boldsymbol{y})A(\boldsymbol{y}, \boldsymbol{x}), \tag{23}$$

which necessarily leads to a *reversible* Markov chain with $p(\boldsymbol{x})$ as its invariant distribution.

The Metropolis algorithm has been extensively used in statistical physics over the past 40 years and is the cornerstone of all MCMC techniques recently adopted and generalized in the statistics community. Another class of MCMC algorithms, the Gibbs sampler [7], differs from the Metropolis algorithm in that it uses conditional distributions based on $p(\boldsymbol{x})$ to construct Markov chain moves.

3.1.2. Gibbs Sampler. Suppose $\boldsymbol{x} = (x_1, \ldots, x_d)$, where x_i is either a scalar or a vector. In the Gibbs sampler, one randomly or systematically choose a coordinate, say x_i, and then update its value with a new sample x_i' drawn from the conditional distribution $p(\cdot \mid \boldsymbol{x}_{[-i]})$. Algorithmically, the Gibbs sampler can be implemented as follows:

Algorithm 3 (*Random Scan Gibbs Sampler*)**.** *Suppose currently* $\boldsymbol{x}^{(t)} = (x_1^{(t)}, \ldots, x_d^{(t)})$. *Then*

1. *Randomly select i from the index set* $\{1, \ldots, d\}$ *according to a given probability vector* $(\pi_1, \ldots, \pi_d)$.
2. *Draw* $x_i^{(t+1)}$ *from the conditional distribution* $p(\cdot \mid \boldsymbol{x}_{[-i]}^{(t)})$, *and let* $\boldsymbol{x}_{[-i]}^{(t+1)} = \boldsymbol{x}_{[-i]}^{(t)}$.

Algorithm 4 (*Systematic Scan Gibbs Sampler*)**.** *Let the current state be* $\boldsymbol{x}^{(t)} = (x_1^{(t)}, \ldots, x_d^{(t)})$.

For $i = 1, \ldots, d$, *we draw* $x_i^{(t+1)}$ *from the conditional distribution*

$$p\big(x_i \mid x_1^{(t+1)}, \ldots, x_{i-1}^{(t+1)}, x_{i+1}^{(t)}, \ldots, x_d^{(t)}\big).$$

It is easy to check that *every* individual conditional update leaves p invariant. Suppose currently $\boldsymbol{x}^{(t)} \sim p$. Then $\boldsymbol{x}_{[-i]}^{(t)}$ follows its marginal distribution under p. Thus,

$$p\big(x_i^{(t+1)} \mid \boldsymbol{x}_{[-i]}^{(t)}\big) \cdot p\big(\boldsymbol{x}_{[-i]}^{(t)}\big) = p\big(x_i^{(t+1)}, \boldsymbol{x}_{[-i]}^{(t)}\big), \quad (24)$$

which implies that the joint distribution of $(\boldsymbol{x}_{[-i]}^{(t)}, x_i^{(t+1)})$ is unchanged at p after one update.

The Gibbs sampler's popularity in statistics community stems from its extensive use of *conditional distributions* in each iteration. The data augmentation method [8] first linked the Gibbs sampling structure with missing data problems and the EM-type algorithms. The Gibbs sampler was further popularized by [9] where it was pointed out that the conditionals needed in Gibbs iterations are commonly available in many Bayesian and likelihood computations. Under regularity conditions, one can show that the Gibbs sampler chain converges geometrically and its convergence rate is related to how the variables correlate with each other [10]. Therefore, grouping highly correlated variables together in the Gibbs update can greatly speed up the sampler.

3.1.3. Other Techniques. A main problem with all the MCMC algorithms is that they may, for some problems, move very slowly in the configuration space or may be trapped in a local mode. This phenomena is generally called *slow-mixing* of the chain. When chain is slow-mixing, estimation based on the resulting Monte Carlo samples becomes very inaccurate. Some recent techniques suitable for designing more efficient MCMC samplers include parallel tempering [11], multiple-try method[12], and evolutionary Monte Carlo [13].

3.2. Application of MCMC—Blind Turbo Multiuser Detection

In this section, we illustrate the application of MCMC signal processing (in particular, the Gibbs sampler) to the multiuser detection problem (cf. Example 2), taken from [14]. Specifically, we show that the MCMC technique leads to a novel blind turbo multiuser receiver structure for a coded CDMA systems. The block diagram of the transmitter end of such a system is shown in Fig. 1. The binary information bits $\{d_{k,\tau}\}$ for user k are encoded using some channel code (e.g., block code, convolutional code or turbo code), resulting in a code bit stream. A code-bit interleaver is used to reduce the influence of the error bursts at the input of the channel decoder. The interleaved code bits are then mapped to binary symbols, yielding symbol stream $\{x_{k,t}\}$. Each data symbol is then modulated by a spreading waveform $\boldsymbol{s}_k$ and transmitted through the channel. The received signal is given by (3).

The task of the receiver is to decode the transmitted information bits $\{d_{k,\tau}\}$ of each user k. We consider a "turbo" receiver structure which iterates between a multiuser detection stage and a decoding stage to successively refine the performance. Such a turbo processing scheme has received considerable recent attention in the fields of coding and signal processing, and has been successively applied to many problems in these areas [15]. The turbo multiuser receiver structure is shown in Fig. 2. It consists of two stages: a Bayesian multiuser detector, followed by a bank of maximum a posteriori probability (MAP) channel decoders. The two stages are separated by deinterleavers and interleavers. The Bayesian multiuser detector computes the a posteriori symbol probabilities $\{P[x_{k,t} = +1 \mid \boldsymbol{Y}]\}_{k,t}$. Based on these, we first compute the a posteriori log-likelihood ratios (LLR's) of a transmitted "+1" symbol and a transmitted "−1" symbol,

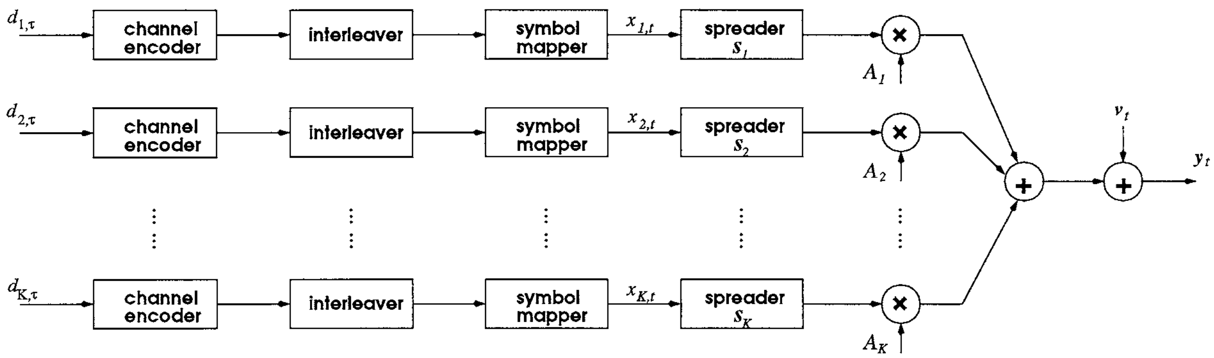


Figure 1. A coded synchronous CDMA communication system.

$$\Lambda_1[x_{k,t}] \triangleq \log \frac{P[x_{k,t} = +1 \mid \boldsymbol{Y}]}{P[x_{k,t} = -1 \mid \boldsymbol{Y}]}. \tag{25}$$

Using the Bayes' rule, (25) can be written as

$$\Lambda_1[x_{k,t}] = \underbrace{\log \frac{p[\boldsymbol{Y} \mid x_{k,t} = +1]}{p[\boldsymbol{Y} \mid x_{k,t} = -1]}}_{\lambda_1[x_{k,t}]} + \underbrace{\log \frac{P[x_{k,t} = +1]}{P[x_{k,t} = -1]}}_{\lambda_2^p[x_{k,t}]}, \tag{26}$$

where the second term in (26), denoted by $\lambda_2^p[x_{k,t}]$, represents the *a priori* LLR of the code bit $x_{k,t}$, which is computed by the channel decoder in the previous iteration, interleaved and then fed back to the Bayesian multiuser detector. (The superscript p indicates the quantity obtained from the previous iteration). For the first iteration, assuming equally likely code bits, i.e., no prior information available, we then have $\lambda_2^p[x_{k,t}] = 0$. The first term in (26), denoted by $\lambda_1[x_{k,t}]$, represents the *extrinsic* information delivered by the Bayesian multiuser detector, based on the received signals $\boldsymbol{Y}$, the structure of the multiuser signal given by (9) and the prior information about all other code bits. The extrinsic information $\lambda_1[x_{k,t}]$, which is not influenced by the *a priori* information $\lambda_2^p[x_{k,t}]$ provided by the channel decoder, is then reverse interleaved and fed into the channel decoder. Based on the extrinsic information of the code bits $\{\lambda_1^p[x_{k,\tau}]\}_\tau$ and the structure of the channel code, the k-th user's MAP channel decoder [16] computes the a posteriori LLR of each code bit, based on which the extrinsic information $\{\lambda_2^p[x_{k,t}]\}_t$ is extracted and fed back the Bayesian multiuser detector as the a priori information in the next iteration.

From the above discussion, it is seen that the key computation involved in the turbo multiuser receiver is calculating the a posteriori symbol probabilities $P[x_{k,t} = +1|\boldsymbol{Y}]$. Note that we do *not* assume that the receiver has the knowledge about the channels, e.g., the user amplitudes $A_1, \ldots, A_K$ or the noise variance σ^2. Hence the receiver is termed "blind". In Bayesian paradigm, all unknowns are considered as random quantities with some prior distributions. We first briefly summarize the principles for choosing those priors, as follows.

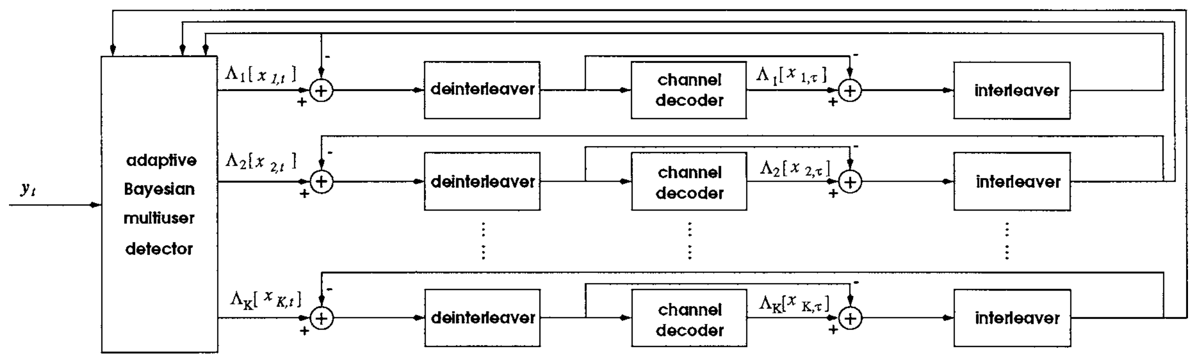


Figure 2. Blind turbo multiuser receiver structure.

- *Noninformative Priors*: In Bayesian analysis, prior distributions are used to incorporate the prior knowledge about the unknown parameters. When such prior knowledge is limited, the prior distributions should be chosen such that they have a minimal impact on the posterior distribution. Such priors are termed as *non-informative*. The rationale for using noninformative prior distributions is to "let the data speak for themselves", so that inferences are unaffected by information external to current data [17, 18].
- *Conjugate Priors*: Another consideration in the selection of the prior distributions is to simplify computations. To that end, *conjugate priors* are usually used to obtain simple analytical forms for the resulting posterior distributions. The property that the posterior distribution belongs to the same distribution family as the prior distribution is called conjugacy. The conjugate family of distributions is mathematically convenient in that the posterior distribution follows a known parametric form [17, 18]. Finally, to make the Gibbs sampler more computationally efficient, the priors should also be chosen such that the conditional posterior distributions are easy to simulate.

For an introductory treatment of the Bayesian philosophy, including the selection of prior distributions, see the textbooks [17–19]. An account of criticism of the Bayesian approach to data analysis can be found in [20, 21]; and a defense of "The Bayesian Choice" can be found in [22].

We next outline the MCMC procedure for solving this problem of blind Bayesian multiuser detection using the Gibbs sampler. Consider the signal model (9). The unknown quantities $\boldsymbol{a}$, σ^2 and $\boldsymbol{X}$ are regarded as realizations of random variables with the following prior distributions. For the unknown amplitudes $\boldsymbol{a}$, a truncated Gaussian prior distribution is assumed,

$$p(\boldsymbol{a}) \sim \mathcal{N}(\boldsymbol{a}_0, \Sigma_0) I_{\{\boldsymbol{a}>\boldsymbol{0}\}}, \tag{27}$$

where $I_{\{\boldsymbol{a}>\boldsymbol{0}\}}$ is an indicator that is 1 if all elements of $\boldsymbol{a}$ are positive and it is 0 otherwise. Note that large value of Σ_0 corresponds to the less informative prior. For the noise variance σ^2, an inverse chi-square prior distribution is assumed,

$$p(\sigma^2) \sim \chi^{-2}(\nu_0, \lambda_0). \tag{28}$$

Small value of ν_0 corresponds to the less informative priors. Finally since the symbols $\{x_{k,t}\}_{k,t}$ are assumed to be independent, the prior distribution $p(\boldsymbol{X})$ can be expressed in terms of the prior symbol probabilities $\rho_{k,t} \triangleq P[x_{k,t} = +1]$, as

$$p(\boldsymbol{X}) = \prod_t \prod_k \rho_{k,t}^{\delta_{k,t}} (1 - \rho_{k,t})^{1-\delta_{k,t}}, \tag{29}$$

where $\delta_{k,t}$ is the indicator such that $\delta_{k,t} = 1$ if $x_{k,t} = +1$ and $\delta_{k,t} = 0$ if $x_{k,t} = -1$.

The blind Bayesian multiuser detector based on the Gibbs sampler is summarized as follows. For more details and some other related issues, see [14].

Algorithm 5 (*Blind Bayesian Multiuser Detector* (B^2*MUD*))**.** *Given the initial values of the unknown quantities* $\{\boldsymbol{a}^{(0)}, \sigma^{2^{(0)}}, \boldsymbol{X}^{(0)}\}$ *drawn from their prior distributions, and for* $j = 1, 2, \ldots$

1. *Draw* $\boldsymbol{a}^{(j)}$ *from*

$$p(\boldsymbol{a} \mid \sigma^2, \boldsymbol{X}, \boldsymbol{Y}) \sim \mathcal{N}(\boldsymbol{a}_*, \boldsymbol{\Sigma}_*) I_{\{\boldsymbol{a}>0\}}, \tag{30}$$

$$\textit{with} \quad \boldsymbol{\Sigma}_*^{-1} \triangleq \boldsymbol{\Sigma}_0^{-1} + \frac{1}{\sigma^2} \sum_{t=1}^{n} \boldsymbol{B}_t \boldsymbol{R} \boldsymbol{B}_t, \tag{31}$$

$$\textit{and} \quad \boldsymbol{a}_* \triangleq \boldsymbol{\Sigma}_* \left(\boldsymbol{\Sigma}_0^{-1} \boldsymbol{a}_0 + \frac{1}{\sigma^2} \sum_{t=1}^{n} \boldsymbol{B}_t \boldsymbol{S}^T \boldsymbol{y}_t \right), \tag{32}$$

where in (31) $\boldsymbol{R} \triangleq \boldsymbol{S}^T \boldsymbol{S}$, *and* $\boldsymbol{B}_t \triangleq \mathrm{diag}\,(x_{1,t}, \ldots, x_{K,t})$.

2. *Draw* $\sigma^{2^{(j)}}$ *from*

$$p(\sigma^2 \mid \boldsymbol{a}, \boldsymbol{X}, \boldsymbol{Y}) \sim \chi^{-2}\left(\nu_0 + Nn, \frac{\nu_0 \lambda_0 + s^2}{\nu_0 + Nn} \right), \tag{33}$$

$$\textit{with} \quad s^2 \triangleq \sum_{t=1}^{n} \| \boldsymbol{y}_t - \boldsymbol{S}\boldsymbol{A}\boldsymbol{x}_t \|^2. \tag{34}$$

3. *For* $t = 1, 1, \ldots, n$

   For $k = 1, 2, \ldots, K$

   Draw $x_{k,t}^{(j)}$ *based on the following probability ratio*

$$\frac{P\left[x_{k,t} = +1 \,\middle|\, \boldsymbol{a}, \sigma^2, \boldsymbol{X}_{[-k,t]}, \boldsymbol{Y}\right]}{P\left[x_{k,t} = -1 \,\middle|\, \boldsymbol{a}, \sigma^2, \boldsymbol{X}_{[-k,t]}, \boldsymbol{Y}\right]} = \frac{\rho_k(i)}{1 - \rho_k(i)} \cdot \exp\left\{ \frac{2A_k}{\sigma^2} \boldsymbol{s}_k^T \left[\boldsymbol{y}_t - \boldsymbol{S}\boldsymbol{A}\boldsymbol{x}_{k,t}^0 \right] \right\}, \tag{35}$$

where $\boldsymbol{x}^0_{k,t} \triangleq [x_{1,t}, \ldots, x_{k-1,t}, 0, x_{k+1,t}, \ldots, x_{K,t}]^T$.

where $\boldsymbol{X}^{(j)}_{[-k,t]} \triangleq \{\boldsymbol{x}^{(j)}_1, \ldots, \boldsymbol{x}^{(j)}_{t-1}, x^{(j)}_{1,t}, \ldots, x^{(j)}_{k-1,t}, x^{(j-1)}_{k+1,t}, \ldots, x^{(j-1)}_{K,t}, \boldsymbol{x}^{(j-1)}_{t+1}, \ldots, \boldsymbol{x}^{(j-1)}_n\}$.

To ensure convergence, the above procedure is usually carried out for $(\nu_0 + \nu)$ iterations and samples from the last ν iterations are used to calculate the Bayesian estimates of the unknown quantities. In particular, the a posteriori symbol probabilities are approximated as

$$P[x_{k,t} = +1 \mid \boldsymbol{Y}] \cong \frac{1}{\nu} \sum_{j=\nu_0+1}^{\nu_0+\nu} \delta^{(j)}_{k,t}, \tag{36}$$

where $\delta^{(j)}_{k,t}$ is the indicator such that $\delta^{(j)}_{k,t} = 1$ if $x^{(j)}_{k,t} = +1$ and $\delta^{(j)}_{k,t} = 0$ if $x^{(j)}_{k,t} = -1$.

3.2.1. Simulation Example. We now illustrate the performance of the blind turbo multiuser receiver. We consider a 5-user ($K = 5$) CDMA system with processing gain $N = 10$. All users have the same amplitudes. The channel code for each user is a rate $\frac{1}{2}$ constraint length-5 convolutional code (with generators 23, 35 in octal notation). The interleaver of each user is independently and randomly generated, and fixed for all simulations. The block size of the information bits is 128. (i.e., the code bit block size is $n = 256$.) All users have the same amplitudes. In computing the symbol probabilities, the Gibbs sampler is iterated 100 runs for each data block, with the first 50 iterations as the "burn-in" period. The symbol posterior probabilities are computed according to (36) with $\nu_0 = \nu = 50$.

Figure 3 illustrates the bit error rate performance of the blind turbo multiuser receiver for user 1 and user 3. The code bit error rate at the output of the blind Bayesian multiuser detector (B^2MUD) is plotted for the first three iterations. The curve corresponding to the first iteration is the uncoded bit error rate at the output of the B^2MUD. The uncoded and coded bit error rate curves in a single-user additive white Gaussian noise (AWGN) channel are also shown in the same figure (as respectively the dash-dotted and the dashed lines). It is seen that by incorporating the extrinsic information provided by the channel decoder as the prior symbol probabilities, the performance of the blind turbo multiuser receiver approaches that of the single-user in an AWGN channel within a few iterations.

4. Sequential Monte Carlo Signal Processing

4.1. General Sequential Monte Carlo Algorithms

4.1.1. Sequential Importance Sampling. Importance sampling is perhaps one of the most elementary, well-known, and versatile Monte Carlo techniques. Suppose we want to estimate $E\{h(\boldsymbol{x})\}$ (with respect to p), using Monte Carlo method. Since directly sampling from $p(\boldsymbol{x})$ is difficult, we want to find a *trial* distribution, $q(\boldsymbol{x})$, which is reasonably close to p but is easy to draw samples from. Because of the simple identity

$$E\{h(\boldsymbol{x})\} = \int h(\boldsymbol{x})\, p(\boldsymbol{x})\, d\boldsymbol{x} \tag{37}$$

$$= \int h(\boldsymbol{x})\, w(\boldsymbol{x})\, q(\boldsymbol{x})\, d\boldsymbol{x}, \tag{38}$$

where

$$w(\boldsymbol{x}) \triangleq \frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}, \tag{39}$$

is the importance weight, we can approximate (38) by

$$E\{h(\boldsymbol{x})\} \cong \frac{1}{W} \sum_{j=1}^{\nu} h\big(\boldsymbol{x}^{(j)}\big) w\big(\boldsymbol{x}^{(j)}\big), \tag{40}$$

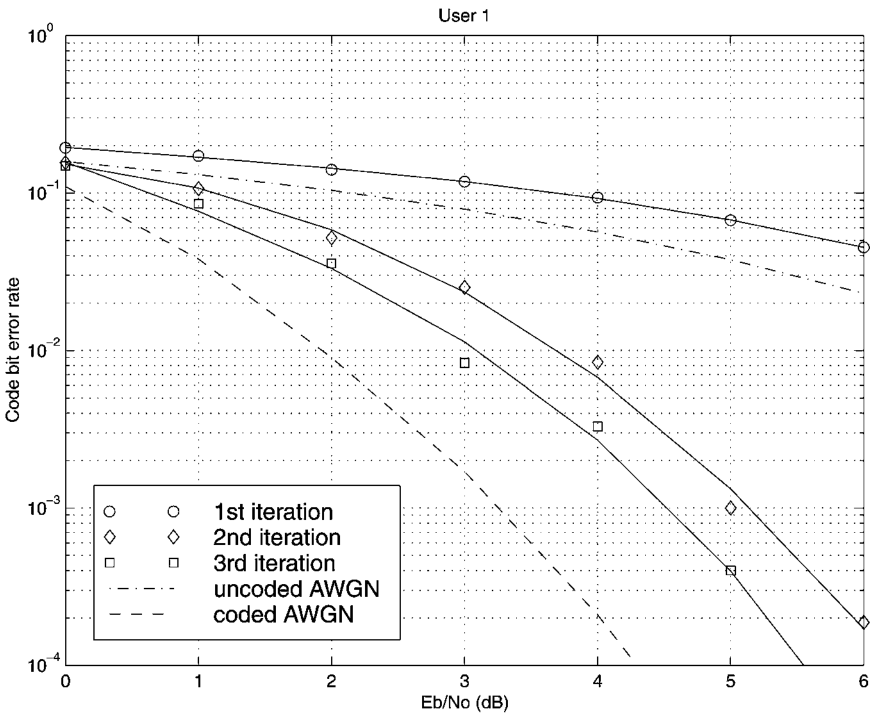
where $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(\nu)}$ are random samples from q, and $W = \sum_{j=1}^{n} w(\boldsymbol{x}^{(j)})$. In using this method, we only need to know the expression of $p(\boldsymbol{x})$ up to a normalizing constant, which is the case for all the signal processing problems we have studied. Each $\boldsymbol{x}^{(j)}$ is said to be properly weighted by $w(\boldsymbol{x}^{(j)})$ with respect to p.

However, it is usually difficult to design a good trial density function in high dimensional problems. One of the most useful strategies in these problems is to build up the trial density sequentially. Suppose we can decompose $\boldsymbol{x}$ as $(x_1, \ldots, x_d)$ where each of the x_j may be multidimensional. Then our trial density can be constructed as

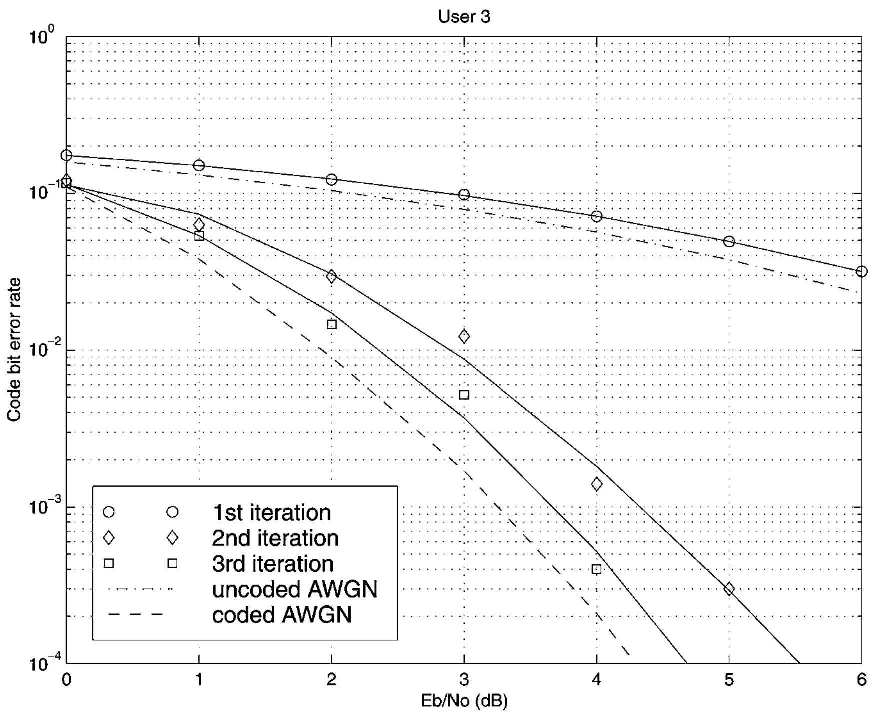
$$q(\boldsymbol{x}) = q_1(x_1) q_2(x_2 \mid x_1) \cdots q_d(x_d \mid x_1, \ldots, x_{d-1}), \tag{41}$$

by which we hope to obtain some guidance from the target density while building up the trial density. Corresponding to the decomposition of $\boldsymbol{x}$, we can rewrite the target density as

$$p(\boldsymbol{x}) = p(x_1) p(x_2 \mid x_1) \cdots p(x_d \mid x_1, \ldots, x_{d-1}), \tag{42}$$



(a)



(b)

Figure 3. Bit error rate performance of the blind turbo multiuser receiver.

and the importance weight as

$$w(\boldsymbol{x}) = \frac{p(x_1)p(x_2 \mid x_1)\cdots p(x_d \mid x_1,\ldots,x_{d-1})}{q_1(x_1)q_2(x_2 \mid x_1)\cdots q_d(x_d \mid x_1,\ldots,x_{d-1})}. \tag{43}$$

Equation (43) suggests a recursive way of computing and monitoring the importance weight. That is, by denoting $\boldsymbol{x}_t = (x_1,\ldots,x_t)$ (thus, $\boldsymbol{x}_d \equiv \boldsymbol{x}$), we have

$$w_t(\boldsymbol{x}_t) = w_{t-1}(\boldsymbol{x}_{t-1})\frac{p(x_t \mid \boldsymbol{x}_{t-1})}{q_t(x_t \mid \boldsymbol{x}_{t-1})}. \tag{44}$$

Then w_d is equal to $w(\boldsymbol{x})$ in (43). Potential advantages of this recursion and (42) are (a) we can stop generating further components of $\boldsymbol{x}$ if the *partial weight* derived from the sequentially generated *partial sample* is too small, and (b) we can take advantage of $p(x_t|\boldsymbol{x}_{t-1})$ in designing $q_t(x_t \mid \boldsymbol{x}_{t-1})$. In other words, the marginal distribution $p(\boldsymbol{x}_t)$ can be used to guide the generation of $\boldsymbol{x}$.

Although the "idea" sounds interesting, the trouble is that expressions (42) and (43) are not useful at all! The reason is that in order to get (42), one needs to have the marginal distribution

$$p(\boldsymbol{x}_t) = \int p(x_1,\ldots,x_d)\,dx_{t+1}\ldots dx_d, \tag{45}$$

which is perhaps more difficult than the original problem.

In order to carry out the sequential sampling idea, we need to find a sequence of "auxiliary distributions," $\pi_1(x_1), \pi_2(\boldsymbol{x}_2),\ldots,\pi_d(\boldsymbol{x})$, so that $\pi_t(\boldsymbol{x}_t)$ is a reasonable approximation to the marginal distribution $p(\boldsymbol{x}_t)$, for $t = 1,\ldots,d-1$, and $\pi_d = p$. We want to emphasize that the π_t are only required to be known up to a normalizing constant and they *only* serve as "guides" to our construction of the whole sample $\boldsymbol{x} = (x_1,\ldots,x_d)$. The *sequential importance sampling* (SIS) method can then be defined as the following recursive procedure.

Algorithm 6 (*Sequential Importance Sampling* (*SIS*))**.** *For* $t = 2,\ldots,d$:

1. *Draw* x_t *from* $q_t(x_t|\boldsymbol{x}_{t-1})$, *and let* $\boldsymbol{x}_t = (\boldsymbol{x}_{t-1}, x_t)$.
2. *Compute*

$$u_t = \frac{\pi_t(\boldsymbol{x}_t)}{\pi_{t-1}(\boldsymbol{x}_{t-1})q_t(x_t \mid \boldsymbol{x}_{t-1})}, \tag{46}$$

and let $w_t = w_{t-1}u_t$. *Here* u_t *is called an incremental weight.*

It is easy to show that $\boldsymbol{x}_t$ is properly weighted by w_t with respect to π_t provided that $\boldsymbol{x}_{t-1}$ is properly weighted by w_{t-1} with respect to π_{t-1}. Thus, the whole sample $\boldsymbol{x}$ obtained by SIS is properly weighted by w_d with respect to the target density $p(\boldsymbol{x})$. The "auxiliary distributions" can also be used to help construct a more efficient trial distribution:

- We can build q_t in light of π_t. For example, one can choose (if possible)

$$q_t(x_t \mid \boldsymbol{x}_{t-1}) = \pi_t(x_t \mid \boldsymbol{x}_{t-1}). \tag{47}$$

  Then the incremental weight becomes

$$u_t = \frac{\pi_t(\boldsymbol{x}_t)}{\pi_{t-1}(\boldsymbol{x}_{t-1})}. \tag{48}$$

  In the same token, we may also want q_t to be $\pi_{t+1}(x_t \mid \boldsymbol{x}_{t-1})$, where the latter involves integrating out x_{t+1}.
- When we observe that w_t is getting too small, we may want to reject the sample half-way and restart. In this way we avoid wasting time on generating samples that are doomed to have little effect in the final estimation. However, as an outright rejection incurs bias, techniques such as the rejection control are needed [23].
- Another problem with the SIS is that the resulting importance weights are often very skewed, especially when d is large. An important recent advance in sequential Monte Carlo to address this problem is the resampling technique [24–26].

4.1.2. SMC for Dynamic Systems. Consider the following dynamic system modeled in a state-space form as

$$\begin{aligned} \text{state equation } z_t &= f_t(z_{t-1}, \boldsymbol{u}_t) \\ \text{observation equation } \boldsymbol{y}_t &= g_t(z_t, \boldsymbol{v}_t), \end{aligned} \tag{49}$$

where z_t, $\boldsymbol{y}_t$, $\boldsymbol{b}_t$ 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.

Let $\boldsymbol{Z}_t = (z_0, z_1,\ldots,z_t)$ and let $\boldsymbol{Y}_t = (\boldsymbol{y}_0, \boldsymbol{y}_1,\ldots,\boldsymbol{y}_t)$. Suppose an on-line 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 $E\{h(\boldsymbol{Z}_t) \mid \boldsymbol{Y}_t\} = \int h(\boldsymbol{Z}_t)p(\boldsymbol{Z}_t \mid \boldsymbol{Y}_t)\,d\boldsymbol{Z}_t$.

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, suppose a set of random samples $\{\boldsymbol{Z}_t^{(j)}\}_{j=1}^{\nu}$ is generated from the trial distribution $q(\boldsymbol{Z}_t \mid \boldsymbol{Y}_t)$. By associating the weight

$$w_t^{(j)} = \frac{p(\boldsymbol{Z}_t^{(j)} \mid \boldsymbol{Y}_t)}{q(\boldsymbol{Z}_t^{(j)} \mid \boldsymbol{Y}_t)} \tag{50}$$

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

$$E\{h(\boldsymbol{Z}_t) \mid \boldsymbol{Y}_t\} \cong \frac{1}{W_t} \sum_{j=1}^{\nu} h(\boldsymbol{Z}_t^{(j)}) w_t^{(j)}, \tag{51}$$

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

To implement Monte Carlo techniques for a dynamic system, a set of random samples properly weighted with respect to $p(\boldsymbol{Z}_t \mid \boldsymbol{Y}_t)$ is needed for any time t. Because the state equation in system (49) possesses a Markovian structure, we can implement a a SMC strategy [26]. Suppose a set of properly weighted samples $\{(\boldsymbol{Z}_{t-1}^{(j)}, w_{t-1}^{(j)})\}_{j=1}^{\nu}$ (with respect to $p(\boldsymbol{Z}_{t-1} \mid \boldsymbol{Y}_{t-1})$) is given at time $(t-1)$. A *Monte Carlo filter* (MCF) generates from the set a new one, $\{\boldsymbol{Z}_t^{(j)}, w_t^{(j)}\}_{j=1}^{\nu}$, which is properly weighted at time t with respect to $p(\boldsymbol{Z}_t \mid \boldsymbol{Y}_t)$, according to the following algorithm.

Algorithm 7 (*Sequential Monte Carlo Filter for Dynamic Systems*). *For* $j = 1, \ldots, \nu$:

1. *Draw a sample* $z_t^{(j)}$ *from a trial distribution* $q(z_t \mid \boldsymbol{Z}_{t-1}^{(j)}, \boldsymbol{Y}_t)$ *and let* $\boldsymbol{Z}_t^{(j)} = (\boldsymbol{Z}_{t-1}^{(j)}, z_t^{(j)})$;
2. *Compute the importance weight*

$$w_t^{(j)} = w_{t-1}^{(j)} \cdot \frac{p(\boldsymbol{Z}_t^{(j)} \mid \boldsymbol{Y}_t)}{p(\boldsymbol{Z}_{t-1}^{(j)} \mid \boldsymbol{Y}_{t-1}) q(z_t^{(j)} \mid \boldsymbol{Z}_{t-1}^{(j)}, \boldsymbol{Y}_t)}. \tag{52}$$

The algorithm is initialized by drawing a set of i.i.d. samples $z_0^{(1)}, \ldots, z_0^{(m)}$ from $p(z_0 \mid \boldsymbol{y}_0)$. When $\boldsymbol{y}_0$ represents the "null" information, $p(z_0 \mid \boldsymbol{y}_0)$ corresponds to the prior of z_0.

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*. Specifically, a useful choice of the trial distribution $q(z_t \mid \boldsymbol{Z}_{t-1}^{(j)}, \boldsymbol{Y}_t)$ for the state space model (49) is of the form

$$\begin{aligned} q(z_t \mid \boldsymbol{Z}_{t-1}^{(j)}, \boldsymbol{Y}_t) &= p(z_t \mid \boldsymbol{Z}_{t-1}^{(j)}, \boldsymbol{Y}_t) \\ &= \frac{p(\boldsymbol{y}_t \mid z_t)\, p(z_t \mid z_{t-1}^{(j)})}{p(\boldsymbol{y}_t \mid z_{t-1}^{(j)})}. \end{aligned} \tag{53}$$

For this trial distribution, the importance weight is updated according to

$$w_t^{(j)} \propto w_{t-1}^{(j)} \cdot p(\boldsymbol{y}_t \mid z_{t-1}^{(j)}). \tag{54}$$

See [26] for the general sequential MCF framework and a detailed discussion on various implementation issues.

4.1.3. Mixture Kalman Filter. Many dynamic system models belong to the class of conditional dynamic linear models (CDLM) of the form

$$\begin{aligned} \boldsymbol{x}_t &= F_{\lambda_t} \boldsymbol{x}_{t-1} + G_{\lambda_t} \boldsymbol{u}_t, \\ \boldsymbol{y}_t &= H_{\lambda_t} \boldsymbol{x}_t + K_{\lambda_t} \boldsymbol{v}_t, \end{aligned} \tag{55}$$

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" 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 the complete statistical characterization of the system dynamics. Recently a novel sequential Monte Carlo method, the mixture Kalman filter (MKF), was proposed in [27] for on-line filtering and prediction of CDLM's; it exploits the conditional Gaussian property and utilizes a marginalization operation to improve the algorithmic efficiency. Instead of dealing 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 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 $\{(\boldsymbol{\Lambda}_t^{(j)}, w_t^{(j)})\}_{j=1}^{\nu}$ to represent $p(\boldsymbol{\Lambda}_t \mid \boldsymbol{Y}_t)$, the MKF approximates the target distribution $p(\boldsymbol{x}_t \mid \boldsymbol{Y}_t)$ by a random mixture of Gaussian distributions

$$\frac{1}{W_t}\sum_{j=1}^{\nu} w_t^{(j)} \mathcal{N}_c\Big(\boldsymbol{\mu}_t^{(j)}, \boldsymbol{\Sigma}_t^{(j)}\Big), \tag{56}$$

where $\kappa_t^{(j)} \triangleq [\boldsymbol{\mu}_t^{(j)}, \Sigma_t^{(j)}]$ is obtained by implementing a Kalman filter for the given indicator trajectory $\boldsymbol{\Lambda}_t^{(j)}$ and $W_t = \sum_{j=1}^{\nu} w_t^{(j)}$. A key step in the MKF is the production at time t of a weighted sample of indicators, $\{(\boldsymbol{\Lambda}_t^{(j)}, \kappa_t^{(j)}, w_t^{(j)})\}_{j=1}^{\nu}$, based on the set of samples, $\{(\boldsymbol{\Lambda}_{t-1}^{(j)}, \kappa_{t-1}^{(j)}, w_{t-1}^{(j)})\}_{j=1}^{\nu}$, at the previous time $(t-1)$ according to the following algorithm.

Algorithm 8 (*Mixture Kalman Filter for Conditional Dynamic Linear Models*). *For $j = 1, \ldots, \nu$:*

1. *Draw a sample $\lambda_t^{(j)}$ from a trial distribution $q\,(\lambda_t \mid \boldsymbol{\Lambda}_{t-1}^{(j)}, \kappa_{t-1}^{(j)}, \boldsymbol{Y}_t)$.*
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*

$$w_t^{(j)} \propto w_{t-1}^{(j)} \cdot \frac{p\big(\boldsymbol{\Lambda}_{t-1}^{(j)}, \lambda_t^{(j)} \mid \boldsymbol{Y}_t\big)}{p\big(\boldsymbol{\Lambda}_{t-1}^{(j)} \mid \boldsymbol{Y}_{t-1}\big)\, q\big(\lambda_t^{(j)} \mid \boldsymbol{\Lambda}_{t-1}^{(j)}, \kappa_{t-1}^{(j)}, \boldsymbol{Y}_t\big)}. \tag{57}$$

4.2. Application of SMC—Adaptive Detection in Fading Channels

Consider the flat-fading channel with additive Gaussian noise, given by (17) and (18). Denote $\boldsymbol{Y}_t \triangleq (y_1, \ldots, y_t)$ and $\boldsymbol{X}_t \triangleq (x_1, \ldots, x_t)$. We are interested in estimating the symbol x_t at time t based on the observation $\boldsymbol{Y}_t$. The Bayes solution to this problem requires the posterior distribution

$$p(z_t, x_t \mid \boldsymbol{Y}_t) = \int p(z_t \mid \boldsymbol{X}_t, \boldsymbol{Y}_t)\, p(\boldsymbol{X}_t \mid \boldsymbol{Y}_t)\, d\boldsymbol{X}_{t-1}. \tag{58}$$

Note that with a given $\boldsymbol{X}_t$, the state-space model (17)–(18) becomes a linear Gaussian system. Hence,

$$p(z_t \mid \boldsymbol{X}_t, \boldsymbol{Y}_t) \sim \mathcal{N}_c(\boldsymbol{\mu}_t(\boldsymbol{X}_t), \boldsymbol{\Sigma}_t(\boldsymbol{X}_t)), \tag{59}$$

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

In order to implement the MKF, we need to obtain a set of Monte Carlo samples of the transmitted symbols, $\{(\boldsymbol{X}_t^{(j)}, w_t^{(j)})\}_{j=1}^{\nu}$, properly weighted with respect to the distribution $p(\boldsymbol{X}_t|\boldsymbol{Y}_t)$. Then the a posteriori symbol probability can be estimated as

$$P[x_t = +1 \mid \boldsymbol{Y}_t] \cong \frac{1}{W_t}\sum_{j=1}^{\nu} 1\big(x_t^{(j)} = +1\big)\, w_t^{(j)}, \tag{60}$$

where $1(\cdot)$ is an indicator function such that $1(x_t = +1) = 1$ if $x_t = +1$ and 0 otherwise.

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 $\kappa_t^{(j)} \triangleq [\boldsymbol{\mu}_t^{(j)}, \boldsymbol{\Sigma}_t^{(j)}]$. The following algorithm, which is based on the mixture Kalman filter and first appeared in [28], generates properly weighted Monte Carlo samples $\{(\boldsymbol{S}_t^{(j)}, \kappa_t^{(j)}, w_t^{(j)})\}_{j=1}^{\nu}$.

Algorithm 9 (*Adaptive Blind Receiver in Flat Fading Channels*).

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 $w_0^{(j)} = 1$, $j = 1, \ldots, \nu$. Since the data symbols are assumed to be independent, initial symbols are not needed.*

   Based on the state-space model (17)–(18), *the following steps are implemented at time t to update each weighted sample. For $j = 1, \ldots, \nu$:*
2. *Compute the one-step predictive update of each Kalman filter $\kappa_{t-1}^{(j)}$:*

$$\boldsymbol{K}_t^{(j)} = \boldsymbol{F}\boldsymbol{\Sigma}_{t-1}^{(j)}\boldsymbol{F}^H + \boldsymbol{g}\boldsymbol{g}^H, \tag{61}$$

$$\gamma_t^{(j)} = \boldsymbol{h}^H\boldsymbol{K}_t^{(j)}\boldsymbol{h} + \sigma^2, \tag{62}$$

$$\eta_t^{(j)} = \boldsymbol{h}^H\boldsymbol{F}\boldsymbol{\mu}_{t-1}^{(j)}. \tag{63}$$

3. *Compute the trial sampling density: For $b \in \{+1, -1\}$, compute*

$$\rho_{t,b}^{(j)} \triangleq P\big[x_t = b \,\big|\, \boldsymbol{X}_{t-1}^{(j)}, \boldsymbol{Y}_t\big] \propto p\big(y_t \,\big|\, x_t = b, \boldsymbol{X}_{t-1}^{(j)}, \boldsymbol{Y}_{t-1}\big)\, P[x_t = b], \tag{64}$$

   with

$$p\big(y_t \,\big|\, x_t = b, \boldsymbol{X}_{t-1}^{(j)}, \boldsymbol{Y}_{t-1}\big) \sim \mathcal{N}_c\big(b\eta_t^{(j)}, \gamma_t^{(j)}\big). \tag{65}$$

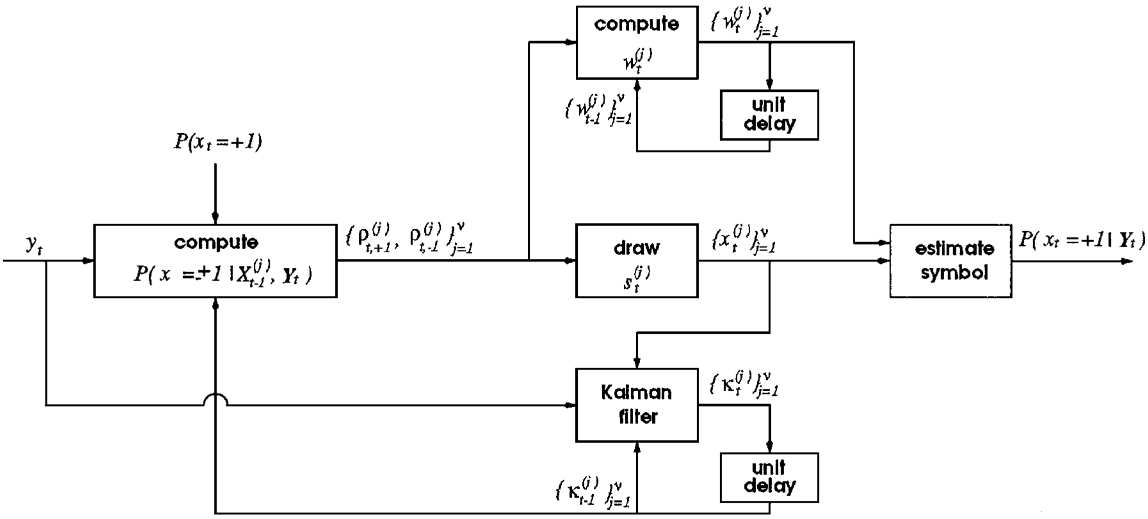


Figure 4. Adaptive Bayesian receiver in a flat-fading channel based on the mixture Kalman filtering.

4. *Impute the symbol x_t: Draw $x_t^{(j)}$ from the set $\{+1, -1\}$ with probability*

$$P\big[x_t^{(j)} = b\big] \propto \rho_{t,b}^{(j)}, \quad b \in \{+1, -1\}. \tag{66}$$

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

5. *Compute the importance weight*:

$$\begin{aligned} w_t^{(j)} &= w_{t-1}^{(j)} \cdot p\big(y_t \big| \boldsymbol{X}_{t-1}^{(j)}, \boldsymbol{Y}_{t-1}\big) \\ &\propto w_{t-1}^{(j)} \cdot \big[\rho_{t,+1}^{(j)} + \rho_{t,-1}^{(j)}\big]. \end{aligned} \tag{67}$$

6. *Compute the one-step filtering update of the Kalman filter $\kappa_{t-1}^{(j)}$: Based on the imputed symbol $x_t^{(j)}$ 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*:

$$\boldsymbol{\mu}_t^{(j)} = \boldsymbol{F}\boldsymbol{\mu}_{t-1}^{(j)} + \frac{1}{\gamma_t^{(j)}}\big(y_t - x_t^{(j)}\eta_t^{(j)}\big)\boldsymbol{K}_t^{(j)}\boldsymbol{h}x_t^{(j)}, \tag{68}$$

$$\boldsymbol{\Sigma}_t^{(j)} = \boldsymbol{K}_t^{(j)} - \frac{1}{\gamma_t^{(j)}}\boldsymbol{K}_t^{(j)}\boldsymbol{h}\boldsymbol{h}^H\boldsymbol{K}_t^{(j)}. \tag{69}$$

The above algorithm is depicted in Fig. 4. It is seen that at any time t, the only quantities that need to be stored are $\{\kappa_t^{(j)}, w_t^{(j)}\}_{j=1}^{\nu}$. At each time t, the dominant computation in this receiver involves the ν one-step Kalman filter updates. Since the ν 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 [29].

Since the fading process is highly correlated, the future received signals contain information about current data and channel state. Hence a delayed estimate is usually more accurate than the concurrent estimate. From the recursive procedure described above, we note by induction that if the set $\{(\boldsymbol{X}_{t+\delta}^{(j)}, w_{t+\delta}^{(j)})\}_{j=1}^{\nu}$ is properly weighted with respect to $p(\boldsymbol{X}_t \mid \boldsymbol{Y}_t)$, then the set $\{(\boldsymbol{X}_{t+\delta}^{(j)}, w_{t+\delta}^{(j)})\}_{j=1}^{\nu}$ is properly weighted with respect to $p(\boldsymbol{X}_{t+\delta} \mid \boldsymbol{Y}_{t+\delta})$, for any $\delta > 0$. Hence, if we focus our attention on $\boldsymbol{X}_t$ at time $(t+\delta)$, we obtain the following delayed estimate of the symbol

$$P[x_t = +1 \mid \boldsymbol{Y}_{t+\delta}] \cong \frac{1}{W_{t+\delta}} \sum_{j=1}^{\nu} 1\big(x_t^{(j)} = +1\big)\, w_{t+\delta}^{(j)}. \tag{70}$$

Since the weights $\{w_{t+\delta}^{(j)}\}_{j=1}^{\nu}$ contain information about the signals $(y_{t+1}, \ldots, y_{t+\delta})$, the estimate in (70) 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 $\{(x_{t+1}^{(j)}, \ldots, x_{t+\delta}^{(j)})\}_{j=1}^{\nu}$. For more details and some other issues related to SMC, such as resampling, see [28].

4.2.1. Simulation Example. In this simulation, the fading process is modeled by the output of a Butterworth filter of order $r = 3$ driven by a complex white Gaussian noise process. The cutoff frequency of this filter is 0.05, corresponding to a normalized

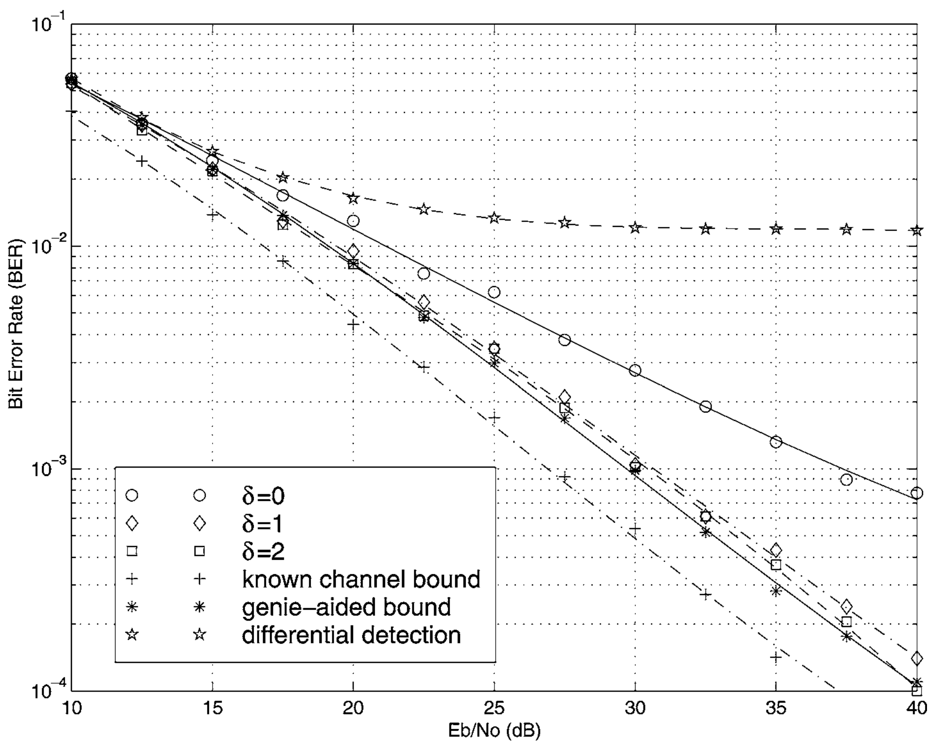


Figure 5. Performance of the SMC adaptive receiver in a flat fading channel.

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:

$$\begin{aligned}\alpha_t - 2.37409\alpha_{t-1} + 1.92936\alpha_{t-2} - 0.53208\alpha_{t-3} \\ = 10^{-2}(0.89409u_t + 2.68227u_{t-1} \\ + 2.68227u_{t-2} + 0.89409u_{t-3}),\end{aligned} \tag{71}$$

where $u_t \sim \mathcal{N}_c(0, 1)$. The filter coefficients in (71) are chosen such that $\text{Var}\{\alpha_t\} = 1$. Differential encoding and decoding are employed to resolve the phase ambiguity. The number of Monte Carlo samples drawn at each time was empirically set as $\nu = 50$. In each simulation, the sequential Monte Carlo algorithm was run on 10000 symbols, (i.e., $t = 1, \ldots, 10000$). In counting the symbol detection errors, the first 50 symbols were discarded to allow the algorithm to reach the steady state. In Fig. 5, the bit error rate (BER) performance versus the signal-to-noise ratio (defined as $\text{Var}\{\alpha_t\}/\text{Var}\{v_t\}$) 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 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.

5. Concluding Remarks

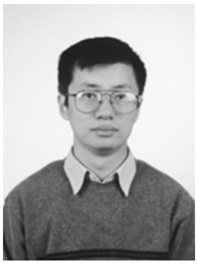
We have presented an overview on the theories and applications of the emerging field of Monte Carlo Bayesian signal processing. The optimal solutions to many statistical signal processing problems, especially those found in wireless communications, are computationally prohibitive to implement by conventional signal processing methods. The Monte Carlo paradigm offers an novel and powerful approach to tackling these problems at a reasonable computational cost. We have outlined two families of Monte Carlo signal processing methodologies, namely, Markov chain Monte Carlo (MCMC) for batch signal processing, and sequential

Monte Carlo (SMC) for adaptive signal processing. Although research on Monte Carlo signal processing has just started, we anticipate that in the near future, an array of *optimal* signal processing problems found in wireless communications, such as mitigation of various types of radio-frequency interference, tracking of fading channels, resolving multipath channel dispersion, space-time processing by multiple transmitter and receiver antennas, exploiting coded signal structures, to name a few, will be solved under the powerful Monte Carlo signal processing framework. Indeed, a number of recent works have addressed applications of MCMC methods in receiver design for various communication systems [30–35]. For additional recent references on theory and applications of Monte Carlo signal processing, see the two journal special issues [36, 37], the books [4, 38, 39], and the websites `http://www.statslab.cam.ac.uk/~mcmc` and `http:// www-sigproc.eng.cam.ac.uk/smc`

References

1. J.G. Proakis, *Digital Communications*, 3rd edn., New York: McGraw-Hill, 1995.
2. S. Verdú, *Multiuser Detection*, Cambridge, UK: Cambridge University Press, 1998.
3. R. Haeb and H. Meyr, "A Systematic Approach to Carrier Recovery and Detection of Digitally Phase Modulated Signals on Fading Channels," *IEEE Trans. Commun.*, vol. 37, no. 7, 1989, pp. 748–754.
4. J.S. Liu, *Monte Carlo Methods in Scientific Computing*, New York: Springer-Verlag, 2001.
5. N. Metropolis, A.W. Rosenbluth, A.H. Teller, and E. Teller, "Equations of State Calculations by Fast Computing Machines," *J. Chemical Physics*, vol. 21, 1953, pp. 1087–1091.
6. W.K. Hastings, "Monte Carlo Sampling Methods Using Markov Chains and Their Applications," *Biometrika*, vol. 57, 1970, pp. 97–109.
7. S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distribution, and the Bayesian Restoration of Images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, no. 11, 1984, pp. 721–741.
8. M.A. Tanner and W.H. Wong, "The Calculation of Posterior Distribution by Data Augmentation (with Discussion)," *J. Amer. Statist. Assoc.*, vol. 82, 1987, pp. 528–550.
9. A.E. Gelfand and A.F.W. Smith, "Sampling-Based Approaches to Calculating Marginal Densities," *J. Amer. Stat. Assoc.*, vol. 85, 1990, pp. 398–409.
10. J.S. Liu, "The Collapsed Gibbs Sampler with Applications to a Gene Regulation Problem," *J. Amer. Statist. Assoc*, vol. 89, 1994, pp. 958–966.
11. C.J. Geyer, "Markov Chain Monte Carlo Maximum Likelihood," in *Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface*, E.M. Keramigas (Ed.), Fairfax: Interface Foundation, 1991, pp. 156–163.
12. J.S. Liu, F. Ling, and W.H. Wong, "The Use of Multiple-Try Method and Local Optimization in Metropolis sampling," *J. Amer. Statist. Assoc*, vol. 95, 2000, pp. 121–134.
13. F. Liang and W.H. Wong, "Evolutionary Monte Carlo: Applications to c_p Model Sampling and Change Point Problem," *Statistica Sinica*, vol. 10, 2000, pp. 317–342.
14. X.Wang and R. Chen, "Adaptive Bayesian Multiuser Detection for Synchronous CDMA in Gaussian and Impulsive Noise," *IEEE Trans. Sign. Proc*, vol. 48, no. 7, 2000, pp. 2013–2028.
15. J. Hagenauer, "The Turbo Principle: Tutorial Introduction and State of the Art," in *Proc. International Symposium on Turbo Codes and Related Topics*, Brest, France, Sept. 1997, pp. 1–11.
16. L.R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate," *IEEE Trans. Inform. Theory*, vol. IT-20, no. 3, 1974, pp. 284–287.
17. G.E. Box and G.C. Tiao, *Bayesian Inference in Statistical Analysis*, Reading, MA: Addison-Wesley, 1973.
18. A. Gelman, J.B. Carlin, H.S. Stern, and D.B. Rubin, *Bayesian Data Analysis*, Chapman & Hall, 1995.
19. E.L. Lehmann and G. Casella, *Theory of Point Estimation*, 2nd edn., New York: Springer-Verlag, 1998.
20. J.O. Berger, *Statistical Decision Theory and Bayesian Analysis*, 2nd edn., New York: Springer-Verlag, 1985.
21. T.J. Rothenberg, "The Bayesian Approach and Alternatives in Econometrics," in *Studies in Bayesian Econometrics and Statistics*, vol. 1, S. Fienberg and A. Zellners (Eds.), Amsterdam: North-Holland, 1977, pp. 55–75.
22. C.P. Robert, *The Bayesian Choice: A Decision-Theoretic Motivation*, New York: Springer-Verlag, 1994.
23. J.S. Liu, R. Chen, and W.H. Wong, "Rejection Control for Sequential Importance Sampling," *Journal of the American Statistical Association*, vol. 93, 1998, pp. 1022–1031.
24. N.J. Gordon, D.J. Salmon Salmon, and A.F.M. Smith, "A Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation," *IEE Proceedings on Radar and Signal Processing*, vol. 140, 1993, pp. 107–113.
25. J.S. Liu and R. Chen, "Blind Deconvolution Via Sequential Imputations," *Journal of the American Statistical Association*, vol. 90, 1995, pp. 567–576.
26. J.S. Liu and R. Chen, "Sequential Monte Carlo Methods for Dynamic Systems," *Journal of the American Statistical Association*, vol. 93, 1998, pp. 1032–1044.
27. R. Chen and J.S. Liu, "Mixture Kalman Filters," *Journal of Royal Statistical Society (B)*, vol. 62, no. 3, 2000, pp. 493–509.
28. R. Chen, X. Wang, and J.S. Liu, "Adaptive Joint Detection and Decoding in Flat-Fading Channels Via Mixture Kalman Filtering," *IEEE Trans. Inform. Theory*, vol. 46, no. 6, 2000, pp. 2079–2094.
29. S.Y. Kung, *VLSI Array Processing*, Englewood Cliffs, NJ: Prentice Hall, 1988.
30. B. Lu and X. Wang, "Bayesian Blind Turbo Receiver for Coded OFDM Systems with Frequency Offset and Frequency-Selective Fading," *IEEE J. Select. Areas Commun.*, vol. 19, no. 12, 2001, Special issue on Signal Synchronization in Digital Transmission Systems.
31. V.D. Phan and X. Wang, "Bayesian Turbo Multiuser Detection for Nonlinearly Modulated CDMA," *Signal Processing*, to appear.
32. X. Wang and R. Chen, "Blind Turbo Equalization in Gaussian and Impulsive Noise," *IEEE Trans. Vehi. Tech.*, vol. 50, no. 4,

2001, pp. 1092–1105.
33. Z. Yang, B. Lu, and X. Wang, "Bayesian Monte Carlo Multiuser Receiver for Space-Time Coded Multi-Carrier CDMA Systems," *IEEE J. Select. Areas Commun.*, vol. 19, no. 8, 2001, pp. 1625–1637, Special Issue on Multiuser Detection Techniques with Application to Wired & Wireless Communication System.
34. Z. Yang and X. Wang, "Blind Turbo Multiuser Detection for Long-Code Multipath CDMA," *IEEE Trans. Commun.*, to appear.
35. Z. Yang and X. Wang, "Turbo Equalization for GMSK Signaling Over Multipath Channels Based on the Gibbs Sampler," *IEEE J. Select. Areas Commun.*, vol. 19, no. 9, 2001, pp. 1753–1763, Special Issue on the Turbo Principle: From Theory to Practice.
36. *IEEE Trans. Sig. Proc*., Special Issue on Monte Carlo Methods for Statistical Signal Processing, 2002.
37. *Signal Processing*, Special Issue on Markov Chain Monte Carlo (MCMC) Methods for Signal Processing, vol. 81, no. 1, 2001.
38. A. Doucet, N. de Freitas, and N. Gordon (Eds.), *Sequential Monte Carlo Methods in Practice*, New York: Springer-Verlag, 2001.
39. W.R. Wilks, S. Richardson, and D.J. Spiegelhalter (Eds.), *Monte Carlo Monte Carlo in Practice*, London: Chapman & Hall, 1998.



Xiaodong Wang received the B.S. degree in Electrical Engineering and Applied Mathematics (with the highest honor) from Shanghai Jiao Tong University, Shanghai, China, in 1992; the M.S. degree in Electrical and Computer Engineering from Purdue University in 1995; and the Ph.D degree in Electrical Engineering from Princeton University in 1998. In July 1998, he joined the Department of Electrical Engineering, Texas A&M University, as an Assistant Professor.

Dr. Wang's research interests fall in the general areas of computing, signal processing and communications. He has worked in the areas of digital communications, digital signal processing, parallel and distributed computing, nanoelectronics and quantum computing. His current research interests include multiuser communications theory and advanced signal processing for wireless communications. He worked at the AT&T Labs–Research, in Red Bank, NJ, during the summer of 1997. Dr. Wang is a member of the IEEE and a member of the American Association for the Advancement of Science. Dr. Wang has received the 1999 NSF CAREER Award. He has also received the 2001 IEEE Communications Society and Information Theory Society Joint Paper Award.

He currently serves as an Associate Editor for the IEEE Transactions on Communications, and for the IEEE Transactions on Signal Processing.

Rong Chen is a Professor at the Department of Information and Decision Sciences, College of Business Administration, University of Illinois at Chicago. Before joining UIC in 1999, he was at Department of Statistics, Texas A&M University.

Dr. Chen received his B.S. (1985) in Mathematics from Peking University, P.R. China, his M.S. (1987) and Ph.D. (1990) in Statistics from Carnegie Mellon University. His main research interests are in time series analysis, statistical computing and Monte Carlo methods in dynamic systems, and statistical applications in engineering and business. He is an Associate Editor for Journal of American Statistical Association, Journal of Business and Economic Statistics, Statistica Sinica and Computational Statistics.

Jun S. Liu received the B.S. degree in Mathematics in 1985 from Peking University, Beijing, China; and the Ph.D. degree in Statistics in 1991 from the University of Chicago. He was Assistant Professor of Statistics at Harvard from 1991 to 1994 and joined the Statistics Department of Stanford University in 1994, as Assistant Professor. In 2000, he returned to Harvard Statistics Department as Professor.

Dr. Liu's main research interests are Bayesian modeling and computation, Monte Carlo methods, bioinformatics, dynamic system and nonlinear filtering. Dr. Liu has worked with collaborators on designing novel Gibbs sampling-based algorithms to find subtle repetitive patterns in biopolymer sequences. These algorithms have been successfully applied to predict protein functions and to understand gene regulation mechanisms. Dr. Liu has also worked on theoretical analyses of Markov chain Monte Carlo algorithms, such as efficiency comparison of different algorithms, novel algorithmic designs, and convergence rate studies. Dr. Liu's research recently focuses on designing more efficient sequential Monte Carlo-based filtering methods for signal processing and developing statistical tools for genomic data analysis.

Dr. Liu received the CAREER award from the National Science Foundation in 1995 and the Mitchell Prize from the International Society of Bayesian Analysis in 2000.
Jliu@STAT.harvard.edu