# Markov Chain Monte Carlo Algorithms for CDMA and MIMO Communication Systems

Behrouz Farhang-Boroujeny, *Senior Member, IEEE*, Haidong (David) Zhu, and Zhenning Shi, *Member, IEEE*

***Abstract*—In this paper, we develop novel Bayesian detection methods that are applicable to both synchronous code-division multiple-access and multiple-input multiple-output communication systems. Markov chain Monte Carlo (MCMC) simulation techniques are used to obtain Bayesian estimates (soft information) of the transmitted symbols. Unlike previous reports that widely use statistical inference to estimate *a posteriori* probability (APP) values, we present alternative statistical methods that are developed by viewing the underlying problem as a multidimensional Monte Carlo integration. We show that this approach leads to results that are similar to those that would be obtained through a proper Rao–Blackwellization technique and thus conclude that our proposed methods are superior to those reported in the literature. We also note that when the channel signal-to-noise ratio is high, MCMC simulator experiences some very slow modes of convergence. Thus accurate estimation of APP values requires simulations of very long Markov chains, which may be infeasible in practice. We propose two solutions to this problem using the theory of importance sampling. Extensive computer simulations show that both solutions improve the system performance greatly. We also compare the proposed MCMC detection algorithms with the sphere decoding and minimum mean square error turbo detectors and show that the MCMC detectors have superior performance.**

***Index Terms*—Code-division multiple access (CDMA), detection algorithms, Markov chain Monte Carlo, multiple-input multiple-output (MIMO).**

## I. Introduction

In the past, a significant amount of research has addressed various detection methods for multiple-access interference suppression in code-division multiple-access (CDMA) communication systems [1]–[3]. The optimum receivers that jointly detect multiuser information sequences have also been derived [4] and have shown that each user can approach a performance close to that of a single-user channel. Since the optimum detector has a prohibitive complexity, a number of its suboptimum versions have been proposed [5]–[8]. Wang and Poor [8], in particular, have proposed a receiver structure for joint detection of a set of $K$ user coded sequences. The proposed structure consists of a soft-in soft-out (SISO) multiuser detector and $K$ parallel (independent) SISO forward error correction (FEC) decoders. The multiuser detector and FEC decoders exchange soft information in a turbo loop, resulting in an improved performance over iterations. The multiuser detector, for each user, uses the soft information from the FEC decoders to cancel interferences from other users. This is followed by a minimum mean square error (MMSE) estimator that further reduces the residual interference. The MMSE estimator requires computation of the estimator coefficients for each user symbol separately. This involves inversion of a $K$-by-$K$ matrix for each user symbol. Since this is a very large complexity, in [8] a procedure that significantly reduces the computational complexity of this step has been proposed. This involves a sequence of inversion steps through the matrix inversion lemma. However, a careful examination of this procedure reveals that it is only applicable to the cases of underloaded and fully loaded systems, i.e., when the number of users $K$ is smaller than or equal to the spreading gain $N$. For cases of overloaded systems, i.e., when $K > N$, the underlying matrices become ill-conditioned and thus the desired inverses do not exist. To avoid this problem, Trichard *et al.* [9] and also Schlegel and Shi [10] have given slightly different formulations of the multiuser detector that are applicable to the case $K > L$ as well.

Communication systems using multiple antennas at both transmitter and receiver sides, known as multiple-input multiple-output (MIMO) systems, on the other hand, have become very popular in recent years. Early works on the subject [11]–[13] have shown that such channels have a capacity that grows linearly with the minimum of the number of antennas at the transmitter and receiver sides. The original layering techniques adopted in MIMO communication systems [12] resemble a synchronous CDMA system with hard cancellation [3]. MIMO detectors with soft cancellation that resembles the MMSE detector of [8] have also been proposed [48]. Each transmit antenna may be viewed as a user and the channel gains between each transmit antenna and the receive antennas is a vector that may be viewed as the spreading code of the associated user. Accordingly, the number of receive antennas plays the same role as the spreading gain $N$ in a multiuser system, and the number of transmit antennas plays the same role as the number of users, $K$. Hence, the methods developed for multiuser detection can be immediately applied to a MIMO channel with $K$ transmit and $N$ receive antennas. Moreover, following the CDMA terminology [2], we refer to the ratio $K/N$ as the system load and, accordingly, refer to a MIMO channel as underloaded when $K < N$, fully loaded when $K = N$, and overloaded when $K > N$.

Manuscript received October 22, 2004; revised July 2, 2005. The work of Z. Shi was supported in part by the Australian government under the "Backing Australia's Ability" initiative and in part by the Australian Research Council. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Xiaodong Wang.

B. Farhang-Boroujeny and H. Zhu are with the Electrical and Computer Engineering Department, University of Utah, Salt Lake City, UT 84112 USA (e-mail: farhang@ece.utah.edu; haidongz@eng.utah.edu).

Z. Shi is with National ICT, Braddon, ACT 2612, Australia (e-mail: zhenning.shi@nicta.com.au).

Digital Object Identifier 10.1109/TSP.2006.872539

Statistical methods that are based on Markov chain simulation have recently been applied to various communication systems including multiuser and MIMO detection. The Markov chain Monte Carlo (MCMC)-based detectors, in particular, have been more popular and widely studied [15]–[30]. In MCMC methods, statistical inferences are developed by simulating the underlying processes through Markov chains. Such simulations are found useful in reducing the exponential complexity of the multidimensional systems to a linear or at most a polynomial complexity. As an example, in a MIMO channel with $K$ transmit antennas and quadrature phase-shift keying (QPSK) signaling from each antenna, the constellation size of each space-time symbol will be $4^K$. For a moderate size system with $K = 8$, the constellation size is 65 536. Obviously, a maximum likelihood search over such constellation is prohibitive and thus is infeasible. Application of the MCMC methods in such problems allows extraction of the desired statistical inferences with a complexity that does not grow exponentially with the signal dimensions.

Huang and Djurić [18], [19] noted that the transient time of an MCMC channel model can be very long and introduced the concept of a Gibbs coupler which aims at obtaining each sample from the target distribution. However, it was noted in [19] that the computational complexity of the Gibbs coupler algorithm increases rapidly in a correlated search space, such as what is often encountered in a fully or overloaded CDMA system. More recently, Guo and Wang [31] proposed detection methods for MIMO systems based on sequential Monte Carlo (SMC) method. Dong *et al.* [32] applied the same method to a BLAST-type receiver and demonstrated that this can significantly improve the performance of MIMO detectors.

Most of the works on MCMC/SMC application to communication systems have proposed counting the number of $+1$ (logical 1) and $-1$ (logical 0) occurrences of an information bit, say, $b_k$, in an associated Markov chain for evaluating the probability of $b_k$ being equal to $+1$ or $-1$ (i.e., to obtain $P(b_k = +1)$ and/or $P(b_k = -1)$) and thus to infer the log-likelihood ratio (LLR) $\ln(P(b_k = +1)/P(b_k = -1))$. This method, although it apparently works for many cases presented in the previous reports, may fail in other cases, as we demonstrate through an example in Section X. It may be particularly noted that when $P(b_k = +1)$ is close to zero or one, reliable evaluation of the LLR value ($L$-value, for short) may require a very long simulation run of MCMC; we need to observe many occurrences of $b_k$ in order to obtain an accurate estimate of $P(b_k = +1)$ and/or $P(b_k = -1)$. This clearly defeats the goal of implementing a system with an acceptable computational complexity.

In this paper, we develop a Bayesian/SISO detection procedure based on MCMC method by approaching the problem from a different angle than the previous works. We note that the computation of $L$-values in a SISO multiuser detector requires a multidimensional summation over all combinations of the users' bits. The Monte Carlo integration and importance sampling [35] are introduced and applied for dealing with the computational complexity of this problem. MCMC simulation [35], [36] is then introduced as an effective method for obtaining *important* samples in the Monte Carlo integration. With this approach, we find that for the same number of MCMC simulation (Markov chain) steps, the proposed method results in much more reliable values of LLR compared to those obtained through the methods proposed in the previous works [16]–[30]. An exception is [15], where the authors have derived a SISO multiuser detector by adopting a graphical Bayesian model. The procedure presented in [15] is similar to one of the three algorithms that are introduced in this paper—the most trivial of the three and the most inferior in performance.

This paper is organized as follows. The channel model that is used throughout the paper is introduced in Section II. This covers both multiuser and MIMO channels. For both simplicity and brevity, in Sections III–XI we use the term *multiuser* only, even though in all cases the discussions and results are applicable to both multiuser and MIMO systems. The problem of iterative multiuser detection is introduced in Section III, where we show how extrinsic information can be exchanged between a SISO multiuser detector and a set of parallel SISO FEC decoders. Section IV reviews briefly the background materials from Monte Carlo statistical methods and sets the foundation for the rest of this paper. Implementation of the multiuser detector based on an MCMC channel model is developed in Section V. An extension of these results to the general case of $L$-ary signaling is presented in an Appendix. In Section VI, we briefly discuss the method of statistical inference based on MCMC simulation and highlight a fundamental difference between the majority of the previous publications and the approach of this paper. In Section VII, we summarize the results of Sections IV–VI and present the multiuser detection algorithms. The max-log version of the proposed algorithms is introduced in Section VIII. In Section IX, we point out a shortcoming of the MCMC in high SNR regime and propose a solution to it. Computer simulations that evaluate the performance of the proposed algorithms are presented in Section X. The concluding remarks and some further comments are made in Section XI.

Throughout this paper the following notations are adhered to. Vectors are denoted by lowercase bold letters. Matrices are denoted by uppercase bold letters. The $ij$th element of a matrix, say, $\mathbf{A}$, is denoted by $a_{ij}$. The superscripts $T$ and $H$ are used to denote matrix or vector transpose and Hermitian, respectively. Integer subscripts are used to distinguish different users (or, in MIMO channels, data streams transmitted through different transmit antennas). Integer time indexes are put in parentheses. For example, in a multiuser communication system, $d_k(i)$ is used to denote the $i$th symbol of the user $k$. In most parts of this paper, we drop the time index $i$ for brevity. The notations $P(\cdot)$ and $p(\cdot)$ denote the probability of discrete random variables and the probability density of continuous random variables, respectively.

## II. Channel Model

For the purpose of theoretical developments in this paper, we consider a multiuser communication system with $K$ users. We assume that users are synchronous and have independent spreading sequences of length $N$, i.e., a spreading gain of $N$. Using $\mathbf{d}(i) = [d_1(i)\ d_2(i)\ \ldots\ d_K(i)]^T$ to denote data symbols transmitted by the $K$ users in the $i$th time slot and $\mathbf{A}(i)$ to denote

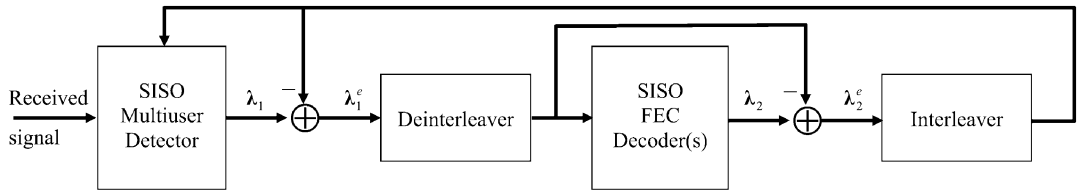



Fig. 1. Receiver structure. The multilayer detector and FEC decoder(s) are SISO blocks that exchange soft information $\boldsymbol{\lambda}_1^e$ and $\boldsymbol{\lambda}_2^e$ in a turbo loop.

the matrix containing spreading code (scaled with the channel gain) of the $k$th user in its $k$th column, the received signal vector, sampled at chip-rate, is given by

$$\mathbf{y}(i) = \mathbf{A}(i)\mathbf{d}(i) + \mathbf{n}(i) \tag{1}$$

where $\mathbf{n}(i)$ is the channel additive noise. We assume that the samples of $\mathbf{n}(i)$ are zero-mean and independent identically distributed (i.i.d.), and $E[\mathbf{n}(i)\mathbf{n}^H(i)] = \sigma_n^2\mathbf{I}$. In the discussion that follows, we assume that data symbols are transmitted and decoded in blocks of length $M$ and for each block the time index $i$ takes the values of 1 to $M$.

It is worth noting that the channel model (1) is that of a flat fading channel. When the channel is frequency selective, the orthogonal frequency-division multiplexing (OFDM) technique could be used to divide it into a number of parallel flat fading subchannels. The results derived in this paper can then be applied by processing the signals of each subchannel separately. The channel code may run across the subchannels, as in coded OFDM [56].

## III. Iterative Multiuser Detection

Fig. 1 presents a block diagram of an iterative multiuser detector, similar to the one proposed by Wang and Poor [8]. It consists of an SISO multiuser detector and a set of parallel SISO channel/FEC decoders. The SISO multiuser detector generates a set of soft output sequences for the information sequences $d_1(i), d_2(i), \ldots, d_K(i)$, based on the observed input vector sequence $\mathbf{y}(i)$ and the a priori (soft) information from the previous iteration of the FEC decoders. After subtracting the a priori information from the output of the multiuser detector, the remaining information which is new (extrinsic) to the FEC decoders is passed to them for further processing. Similarly, the soft input information to each FEC decoder is subtracted from its output to generate the new (extrinsic) information before being fed back to the SISO multiuser detector.

Let us use $\lambda_1(d_k(i))$ to denote the soft output of SISO multiuser detector for $d_k(i)$, $\lambda_2(d_k(i))$ to denote the *a posteriori* output of the FEC decoder for $d_k(i)$, and $\lambda_1^e(d_k(i))$ and $\lambda_2^e(d_k(i))$ to denote the corresponding extrinsic information, respectively. Moreover, we define the vectors $\boldsymbol{\lambda}_2^e(i) = [\lambda_2^e(d_1(i))\, \lambda_2^e(d_2(i)) \cdots \lambda_2^e(d_K(i))]^T$ and $\boldsymbol{\lambda}_1^e(k) = [\lambda_1^e(d_k(1))\, \lambda_1^e(d_k(2)) \cdots \lambda_1^e(d_k(M))]^T$. Note that $\boldsymbol{\lambda}_2^e(i)$ denotes the extrinsic information of all the users at time slot $i$, while $\boldsymbol{\lambda}_1^e(k)$ denotes the extrinsic information of the user $k$ across one data block.

When data symbols are binary, taking values of $+1$ and $-1$, each symbol information is quantified by the LLR of a transmitted "$+1$" and a transmitted "$-1$," given the input information, i.e.,

$$\lambda_1(d_k(i)) = \ln \frac{P\left(d_k(i) = +1 | \mathbf{y}(i), \boldsymbol{\lambda}_2^e(i)\right)}{P\left(d_k(i) = -1 | \mathbf{y}(i), \boldsymbol{\lambda}_2^e(i)\right)} \tag{2}$$

and

$$\lambda_2(d_k(i)) = \ln \frac{P\left(d_k(i) = +1 | \boldsymbol{\lambda}_1^e(k), \text{decoding}\right)}{P\left(d_k(i) = -1 | \boldsymbol{\lambda}_1^e(k), \text{decoding}\right)}. \tag{3}$$

The $L$-values in (3) are obtained by following a standard turbo decoding algorithm. The major problem is finding the values of $\lambda_1(d_k(i))$ in a computationally efficient manner. To see this difficulty we note that[1]

$$\begin{aligned} &P\left(d_k(i) = +1 | \mathbf{y}(i), \boldsymbol{\lambda}_2^e(i)\right) \\ &\quad = \sum_{\mathbf{d}_{-k}(i)} P\left(d_k(i) = +1, \mathbf{d}_{-k}(i) | \mathbf{y}(i), \boldsymbol{\lambda}_2^e(i)\right) \\ &\quad = \sum_{\mathbf{d}_{-k}(i)} P\left(d_k(i) = +1 | \mathbf{y}(i), \mathbf{d}_{-k}(i), \boldsymbol{\lambda}_2^e(i)\right) \\ &\qquad \times P\left(\mathbf{d}_{-k}(i) | \mathbf{y}(i), \boldsymbol{\lambda}_2^e(i)\right) \end{aligned} \tag{4}$$

where the second identity follows by applying the chain rule [37], $\mathbf{d}_{-k}(i) = [d_1(i) \ \ldots \ d_{k-1}(i)\ d_{k+1}(i) \ \ldots \ d_K(i)]^T$ and the summation is overall possible values of $\mathbf{d}_{-k}(i)$. The number of combinations that $\mathbf{d}_{-k}(i)$ takes grows exponentially with $K$ and thus becomes prohibitive for large values of $K$. This prohibitive complexity can be avoided by adopting the Monte Carlo statistical methods that are reviewed in Section IV.

In the rest of this paper, for brevity, we drop the time index $i$ from all the involved variables, e.g., $P\left(d_k | \mathbf{y}, \boldsymbol{\lambda}_2^e\right)$ is a shorthand notation for $P\left(d_k(i) | \mathbf{y}(i), \boldsymbol{\lambda}_2^e(i)\right)$.

## IV. Monte Carlo Statistical Methods

### A. Monte Carlo Integration

Consider the generic problem of evaluating the weighted mean of a function $h(x)$ of the random variable $X$, given the weighting function $f(x)$, i.e.,

$$E_f[h(X)] = \int_{\mathcal{X}} h(x) f(x) dx \tag{5}$$

where $\mathcal{X}$ is the domain of $X$ and $f(\cdot)$ is a proper density function, i.e., $f(x) \geq 0$ for $x \in \mathcal{X}$ and $\int_{\mathcal{X}} f(x)dx = 1$.

[1]One may note that in $P\left(d_k(i) = +1 | \mathbf{y}(i), \mathbf{d}_{-k}(i), \boldsymbol{\lambda}_2^e(i)\right)$, $\boldsymbol{\lambda}_2^e(i)$ may be replaced by $\lambda_2^e(d_k(i))$ since when $\mathbf{d}_{-k}(i)$ is specified, the associated extrinsic information become irrelevant. We have avoided this simplification here. However, in later sections, we will take note of this and remove $\boldsymbol{\lambda}_2^e(i)$ whenever all elements of $\mathbf{d}_k(i)$ are known and hence all the elements of $\boldsymbol{\lambda}_2^e(i)$ become irrelevant.

According to the classical Monte Carlo integration [33], [34], an estimate of (5) can be obtained by evaluating the empirical average

$$\bar{h} = \frac{1}{N_s} \sum_{n=1}^{N_s} h(x_n) \tag{6}$$

where $x_n$s are samples from the distribution $f(x)$. Moreover, the speed of convergence of the method can be evaluated by estimating the variance of $\bar{h}$ which can be obtained through [34]

$$\sigma_{\bar{h}}^2 = \frac{1}{N_s(N_s-1)} \sum_{n=1}^{N_s} |h(x_n) - \bar{h}|^2. \tag{7}$$

Here, $x$ may be a scalar or a vector variable. When $x$ is a vector, the integral in (5) is a multiple integral whose direct computation may become prohibitive as the dimension of $x$ increases. From (7), one may note that the accuracy of the estimate (6) reduces with the square of the number of sample points. Moreover, numerical studies show that the dimension of $x$ and the size $N_s$ are very weakly related in the sense that even though the dimension of $x$ may increase, the order of magnitude of $N_s$ remains unchanged. Hence, the exponential growth of the computational complexity that is commonly encountered in performing multiple integrals may be avoided by using the Monte Carlo methods.

### B. Importance Sampling

In the method of importance sampling $E_f[h(X)]$ is evaluated by performing the empirical average [34]

$$E_f[h(X)] \approx \frac{1}{N_s} \sum_{n=1}^{N_s} \frac{f(x_n)}{f_a(x_n)} h(x_n) \tag{8}$$

where the samples $x_n$ are chosen from the auxiliary distribution $f_a(x)$. Equation (8) follows from the alternative representation of (5)

$$E_f[h(X)] = \int_{\mathcal{X}} h(x) \frac{f(x)}{f_a(x)} f_a(x) dx \tag{9}$$

and performing the integral using the regular Monte Carlo integration based on the distribution $f_a(x)$. Clearly, similar to $f(x)$, $f_a(x)$ also should be a proper density function, i.e., it should satisfy the conditions $f_a(x) \geq 0$ for $x \in \mathcal{X}$ and $\int_{\mathcal{X}} f_a(x)dx = 1$. Obviously, when $f_a(x) = f(x)$, (8) reduces to the regular Monte Carlo integration.

It turns out that, for a fixed value of $N_s$, many choices of $f_a(x)$ that are different from $f(x)$ result in more accurate evaluation of $E_f[h(x)]$. In particular, the optimum function $f_a(x)$ that allows accurate evaluation of $E_f[h(x)]$ with minimum number of samples is $f_{a,o}(x) = |h(x)|f(x)/\int_{\mathcal{X}} |h(x)|f(x)dx$. If available, this would allow exact evaluation of $E_f[h(x)]$ by using only one sample [34]. However, $f_{a,o}(x)$ depends on the integral $\int_{\mathcal{X}} |h(x)|f(x)dx$, which may not be available. In fact, when $h(x) > 0$ for $x \in \mathcal{X}$, the latter is the integral that we seek.

A practical approximation that has been motivated from the importance sampling and usually results in a better approximation than (8) is [35]

$$\bar{h} = \frac{\sum_{n=1}^{N_s} \frac{f(x_n)}{f_a(x_n)} h(x_n)}{\sum_{n=1}^{N_s} \frac{f(x_n)}{f_a(x_n)}} \tag{10}$$

where here, also, $x_n$ is chosen from the distribution $f_a(x)$. A special case of (10), of particular interest in this paper, is obtained when $f_a(x)$ is uniform over a *selected range* of $x$. In order to minimize $N_s$, this range should be chosen such that it covers only the values of $x$ for which $f(x)h(x)$ is significant. In other words, we should limit $x$ to only the *important* samples that contribute to the desired integral. When $f_a(x)$ is uniform, (10) simplifies to

$$\bar{h} = \frac{\sum_{n=1}^{N_s} f(x_n)h(x_n)}{\sum_{n=1}^{N_s} f(x_n)}. \tag{11}$$

### C. Markov Chain Monte Carlo (MCMC) Simulation

MCMC simulation is a statistical method that allows generation of samples of a random process when such process could be modeled as a Markov chain. This method has recently become popular and widely used in various scientific explorations [35], [36]. In the application of interest to this paper, we use MCMC to generate samples of $\mathbf{d}$ from the distribution $P(\mathbf{d}|\mathbf{y}, \boldsymbol{\lambda}_2^e)$. Accordingly, we define a Markov chain in which each state corresponds one selection of $\mathbf{d}$. The kernel $K(\mathbf{d}, \mathbf{d}')$, a conditional probability, determines the movement from the state specified by $\mathbf{d}$ to the state specified by $\mathbf{d}'$. Since the number of states in the Markov chain grows exponentially with the size of $\mathbf{d}$, the method of Gibbs sampling is commonly used to alleviate the problem of complexity [35], [36].

One possible version of Gibbs sampler for the multiuser detector may be summarized as follows:

- Initialize $\mathbf{d}^{(-N_b)}$ (randomly),
- for $n = -N_b + 1$ to $N_s$

draw sample $d_1^{(n)}$ from $P\left(d_1 | d_2^{(n-1)}, \ldots, d_K^{(n-1)}, \mathbf{y}, \boldsymbol{\lambda}_2^e\right)$

draw sample $d_2^{(n)}$ from $P\left(d_2 | d_1^{(n)}, d_3^{(n-1)}, \ldots, d_K^{(n-1)}, \mathbf{y}, \boldsymbol{\lambda}_2^e\right)$

$\vdots$

draw sample $d_K^{(n)}$ from $P\left(d_K | d_1^{(n)}, \ldots, d_{K-1}^{(n)}, \mathbf{y}, \boldsymbol{\lambda}_2^e\right)$

In this routine, $\mathbf{d}^{(-N_b)}$ is initialized randomly, possibly taking into account the a priori information $\boldsymbol{\lambda}_2^e$. The "for" loop examines the state variables $d_k$ in order, $N_b + N_s$ times. The first $N_b$ iterations of the loop, called *burn-in* period, are to let the Markov chain converge to near its stationary distribution. The samples used for LLR computations are those of the last $N_s$ iterations, i.e., $\mathbf{d}^{(n)} = \left[d_1^{(n)}\ d_2^{(n)}\ \ldots\ d_K^{(n)}\right]^T$, for $n = 1, 2, \ldots, N_s$.

Other versions of Gibbs samplers have also been proposed [35]. Of particular interest to us is the possibility of running

a number of Gibbs samplers in parallel. This introduces parallelism in the detector implementation, which in turn allows drawing more Gibbs samples when speed is a practical limitation. Moreover, since in a single Markov chain the samples drawn in successive iterations are *correlated*, the parallel Gibbs samplers draw a pool of samples which are *less correlated*; only the samples that belong to the same Markov chain are correlated. However, we note that parallel Gibbs samplers may be inefficient if a large burn-in period $N_b$ is required. Therefore, the choice between single and parallel Markov chains is application dependent and even for a particular application, this choice may vary as the details of the detector implementation vary. We comment more on this later as we present simulation results.

## V. Implementation of the Multiuser Detector

From Section III, we recall that to obtain $\lambda_1(d_k)$, we need to perform the summation on the right-hand side of (4). When $\mathbf{d}$ has a large dimension, the exact evaluation of this summation may be infeasible and thus other alternatives have to be adopted. In this section, we use Monte Carlo integration methods and the Gibbs sampling procedure of the previous section and develop computationally efficient methods for calculation of $P\left(d_k=+1|\mathbf{y},\boldsymbol{\lambda}_2^e\right)$. We divide our discussion into Section V-A and B. In Section V-A, the approximations (6) and (11) are adopted and the corresponding expressions for the multiuser detector are presented. In Section V-B, numerical procedures involved in the computation of the extrinsic information from the multiuser detector are derived.

### A. Monte Carlo Summations

To obtain a summation similar to (6) for computation of $P\left(d_k=+1|\mathbf{y},\boldsymbol{\lambda}_2^e\right)$, we treat $P\left(\mathbf{d}_{-k}|\mathbf{y},\boldsymbol{\lambda}_2^e\right)$ as the density function $f(x)$ and $P\left(d_k=+1|\mathbf{y},\mathbf{d}_{-k},\boldsymbol{\lambda}_2^e\right)$ as the function whose weighted sum is to be obtained, $h(x)$. An estimate of $P\left(d_k=+1|\mathbf{y},\boldsymbol{\lambda}_2^e\right)$ is thus obtained by evaluating the empirical average

$$P\left(d_k=+1|\mathbf{y},\boldsymbol{\lambda}_2^e\right)\approx\frac{1}{N_s}\sum_{n=1}^{N_s}P\left(d_k=+1|\mathbf{y},\mathbf{d}_{-k}^{(n)},\boldsymbol{\lambda}_2^e\right) \tag{12}$$

where $\mathbf{d}_{-k}^{(n)}$ are the samples that are chosen from the distribution $P\left(\mathbf{d}_{-k}|\mathbf{y},\boldsymbol{\lambda}_2^e\right)$, and, when in equations like (12), $\mathbf{d}_{-k}^{(n)}$ is the shorthand notation for $\mathbf{d}_{-k}=\mathbf{d}_{-k}^{(n)}$.

Starting with (11), we treat $P(\mathbf{d}_{-k}|\mathbf{y},\boldsymbol{\lambda}_2^e)$ as $f(x)$ and $P(d_k=+1|\mathbf{y},\mathbf{d}_{-k},\boldsymbol{\lambda}_2^e)$ as $h(x)$. This gives

$$P\left(d_k=+1|\mathbf{y},\boldsymbol{\lambda}_2^e\right)\approx\frac{\sum_{n=1}^{N_s}P\left(d_k=+1|\mathbf{y},\mathbf{d}_{-k}^{(n)},\boldsymbol{\lambda}_2^e\right)P\left(\mathbf{d}_{-k}^{(n)}|\mathbf{y},\boldsymbol{\lambda}_2^e\right)}{\sum_{n=1}^{N_s}P\left(\mathbf{d}_{-k}^{(n)}|\mathbf{y},\boldsymbol{\lambda}_2^e\right)} \tag{13}$$

where the samples $\mathbf{d}_{-k}^{(n)}$ are chosen from a uniform distribution.

### B. Computation of $L$-Values

In this section, we develop numerical procedures for computation of the extrinsic $L$-value

$$\lambda_1^e(d_k)=\lambda_1(d_k)-\lambda_2^e(d_k). \tag{14}$$

The estimates (12) and (13) are treated separately.

*1) Computation Based on (12):* Here, we need to evaluate $P\left(d_k=+1|\mathbf{y},\mathbf{d}_{-k}^{(n)},\boldsymbol{\lambda}_2^e\right)$, for $n=1,2,\ldots,N_s$. For this, we define

$$\lambda_1^{(n)}(d_k)=\ln\frac{P\left(d_k=+1|\mathbf{y},\mathbf{d}_{-k}^{(n)},\boldsymbol{\lambda}_2^e\right)}{P\left(d_k=-1|\mathbf{y},\mathbf{d}_{-k}^{(n)},\boldsymbol{\lambda}_2^e\right)} \tag{15}$$

and expand it as

$$\begin{aligned}\lambda_1^{(n)}(d_k)&=\ln\frac{p\left(\mathbf{y}|\mathbf{d}_{-k}^{(n)},d_k=+1\right)P\left(\mathbf{d}_{-k}^{(n)},d_k=+1|\boldsymbol{\lambda}_2^e\right)}{p\left(\mathbf{y}|\mathbf{d}_{-k}^{(n)},d_k=-1\right)P\left(\mathbf{d}_{-k}^{(n)},d_k=-1|\boldsymbol{\lambda}_2^e\right)}\\&=\ln\frac{p\left(\mathbf{y}|\mathbf{d}_{-k}^{(n)},d_k=+1\right)}{p\left(\mathbf{y}|\mathbf{d}_{-k}^{(n)},d_k=-1\right)}+\lambda_2^e(d_k)\\&=\frac{1}{2\sigma_n^2}\left(\left|\mathbf{y}-\mathbf{A}_{-k}\mathbf{d}_{-k}^{(n)}+\mathbf{a}_k\right|^2-\left|\mathbf{y}-\mathbf{A}_{-k}\mathbf{d}_{-k}^{(n)}-\mathbf{a}_k\right|^2\right)+\lambda_2^e(d_k)\\&=\frac{2}{\sigma_n^2}\Re\left\{\mathbf{a}_k^H\left(\mathbf{y}-\mathbf{A}_{-k}\mathbf{d}_{-k}^{(n)}\right)\right\}+\lambda_2^e(d_k)\\&=\frac{2}{\sigma_n^2}\Re\left\{y_k^{\mathrm{MF}}-\sum_{\substack{l=1\\l\neq k}}^{K}\rho_{kl}d_l^{(n)}\right\}+\lambda_2^e(d_k)\end{aligned} \tag{16}$$

where $\Re\{\cdot\}$ denotes the real part of, $\mathbf{A}_{-k}$ is $\mathbf{A}$ with its $k$th column $\mathbf{a}_k$ removed, $y_k^{\mathrm{MF}}=\mathbf{a}_k^H\mathbf{y}$ is the matched filter output for the $k$th user, and $\rho_{kl}=\mathbf{a}_k^H\mathbf{a}_l$ is the cross-correlation between users $k$ and $l$. Moreover, the second line follows from the fact that, because of interleaving effect, the extrinsic information provided by each element of $\boldsymbol{\lambda}_2^e$ is independent of those provided by its other elements, and we assume that the elements of $\mathbf{d}$ are independent of one another. This implies that

$$\begin{aligned}&\ln\frac{P\left(\mathbf{d}_{-k}^{(n)},d_k=+1|\boldsymbol{\lambda}_2^e\right)}{P\left(\mathbf{d}_{-k}^{(n)},d_k=-1|\boldsymbol{\lambda}_2^e\right)}\\&=\ln\frac{P\left(\mathbf{d}_{-k}^{(n)}|\boldsymbol{\lambda}_{2,-k}^e\right)}{P\left(\mathbf{d}_{-k}^{(n)}|\boldsymbol{\lambda}_{2,-k}^e\right)}+\ln\frac{P\left(d_k=+1|\lambda_2^e(d_k)\right)}{P\left(d_k=-1|\lambda_2^e(d_k)\right)}\\&=\ln\frac{P\left(d_k=+1|\lambda_2^e(d_k)\right)}{P\left(d_k=-1|\lambda_2^e(d_k)\right)}=\lambda_2^e(d_k)\end{aligned} \tag{17}$$

where $\boldsymbol{\lambda}_{2,-k}^e$ is $\boldsymbol{\lambda}_2^e$ with $\lambda_2^e(d_k)$ dropped from it and the last identity follows from the definition of $L$-value. The third line in (16) follows since $p(\mathbf{y}|\mathbf{d})=\left(1/\left(2\pi\sigma_n^2\right)^{N/2}\right)e^{-|\mathbf{y}-\mathbf{A}\mathbf{d}|^2/2\sigma_n^2}$.

Once $\lambda_1^{(n)}(d_k)$ is obtained, recalling that $P\left(d_k=-1|\mathbf{y},\mathbf{d}_{-k}^{(n)},\boldsymbol{\lambda}_2^e\right) = 1 - P\left(d_k=+1|\mathbf{y},\mathbf{d}_{-k}^{(n)},\boldsymbol{\lambda}_2^e\right)$ and solving (15) for $P\left(d_k=+1|\mathbf{y},\mathbf{d}_{-k}^{(n)},\boldsymbol{\lambda}_2^e\right)$, we obtain

$$P\left(d_k=+1|\mathbf{y},\mathbf{d}_{-k}^{(n)},\boldsymbol{\lambda}_2^e\right)=\frac{1}{1+\exp\left(-\lambda_1^{(n)}(d_k)\right)}. \quad (18)$$

This is used in (12) for computation $P(d_k=+1|\mathbf{y},\boldsymbol{\lambda}_2^e)$. Next, we calculate $\lambda_1(d_k) = \ln\left(P(d_k=+1|\mathbf{y},\boldsymbol{\lambda}_2^e)/1-P(d_k=+1|\mathbf{y},\boldsymbol{\lambda}_2^e)\right)$ and substitute in (14) to obtain $\lambda_1^e(d_k)$.

*2) Computation Based on (13):* Here, we begin with evaluating the $L$-value

$$\lambda_1(d_k)=\ln\frac{P(d_k=+1|\mathbf{y},\boldsymbol{\lambda}_2^e)}{P(d_k=-1|\mathbf{y},\boldsymbol{\lambda}_2^e)}. \quad (19)$$

Substituting (13) and its dual when $d_k=-1$ in (19), we obtain

$$\lambda_1(d_k)=\ln\frac{\sum_{n=1}^{N_s}P\left(d_k=+1|\mathbf{y},\mathbf{d}_{-k}^{(n)},\boldsymbol{\lambda}_2^e\right)P\left(\mathbf{d}_{-k}^{(n)}|\mathbf{y},\boldsymbol{\lambda}_2^e\right)}{\sum_{n=1}^{N_s}P\left(d_k=-1|\mathbf{y},\mathbf{d}_{-k}^{(n)},\boldsymbol{\lambda}_2^e\right)P\left(\mathbf{d}_{-k}^{(n)}|\mathbf{y},\boldsymbol{\lambda}_2^e\right)}. \quad (20)$$

We note that

$$P\left(\mathbf{d}_{-k}^{(n)}|\mathbf{y},\boldsymbol{\lambda}_2^e\right)=\frac{p\left(\mathbf{d}_{-k}^{(n)},\mathbf{y}|\boldsymbol{\lambda}_2^e\right)}{p\left(\mathbf{y}|\boldsymbol{\lambda}_2^e\right)}=\frac{p\left(\mathbf{y}|\mathbf{d}_{-k}^{(n)},\boldsymbol{\lambda}_2^e\right)P^e\left(\mathbf{d}_{-k}^{(n)}\right)}{p\left(\mathbf{y}|\boldsymbol{\lambda}_2^e\right)}. \quad (21)$$

## VI. LLR Computation Based on Statistical Inference: The Conventional Approach

Most of the works on MCMC detectors have considered using the statistical inference for obtaining $L$-values. In this approach, $P(d_k=+1|\mathbf{y},\boldsymbol{\lambda}_2^e)$ is obtained by collecting samples of $d_k$ through a Gibbs sampler and then evaluating the statistical average [16]–[30]

$$P(d_k=+1|\mathbf{y},\boldsymbol{\lambda}_2^e)\approx\frac{1}{N_s}\sum_{n=1}^{N_s}\delta\left(d_k^{(n)}=+1\right) \quad (25)$$

where $\delta(\cdot)$ is an indicator function that takes value of 1 if its argument is true and value of 0 otherwise.

To compare the approach of (25) and the detection algorithms that are proposed in this paper, we note that (12) may be viewed as a *Rao–Blackwellization* [35] of (25). Rao–Balckwellization reduces the variance of the estimates when additional side information are available; see [35, p. 311]. It follows from the identity $\mathrm{var}(U)=\mathrm{var}(E[U|V])+E[\mathrm{var}(U|V)]$, where $\mathrm{var}(\cdot)$ denotes the variance of the argument, which implies that $\mathrm{var}(E[U|V])\le\mathrm{var}(U)$. The desired result is obtained if we replace $U$ by the estimate (25), i.e., $(1/N_s)\sum_{n=1}^{N_s}\delta\left(d_k^{(n)}=+1\right)$ and $V$ by $\left(\mathbf{y},\mathbf{d}_{-k}^{(n)},\boldsymbol{\lambda}_2^e\right)$, and note that

$$P\left(d_k=+1|\mathbf{y},\mathbf{d}_{-k}^{(n)},\boldsymbol{\lambda}_2^e\right)=E\left[\delta(d_k=+1)|\mathbf{y},\mathbf{d}_{-k}^{(n)},\boldsymbol{\lambda}_2^e\right].$$

where $P^e\left(\mathbf{d}_{-k}^{(n)}\right)$ is shorthand notation for $P\left(\mathbf{d}_{-k}^{(n)}|\boldsymbol{\lambda}_2^e\right)$, i.e., the probability of $\mathbf{d}_{-k}=\mathbf{d}_{-k}^{(n)}$ given the available extrinsic information. Also, using the Bayes rule, we obtain

$$P\left(d_k=+1|\mathbf{y},\mathbf{d}_{-k}^{(n)},\boldsymbol{\lambda}_2^e\right)=\frac{p\left(\mathbf{y}|\mathbf{d}_{-k}^{(n)},d_k=+1\right)P^e(d_k=+1)}{p\left(\mathbf{y}|\mathbf{d}_{-k}^{(n)},\boldsymbol{\lambda}_2^e\right)}. \quad (22)$$

Substituting (22), the dual of (22) with $d_k=+1$ replaced by $d_k=-1$, and (21) in (20), we get

$$\begin{aligned}\lambda_1(d_k)&=\ln\frac{\sum_{n=1}^{N_s}p\left(\mathbf{y}|\mathbf{d}_{-k}^{(n)},d_k=+1\right)P^e\left(\mathbf{d}_{-k}^{(n)}\right)}{\sum_{n=1}^{N_s}p\left(\mathbf{y}|\mathbf{d}_{-k}^{(n)},d_k=-1\right)P^e\left(\mathbf{d}_{-k}^{(n)}\right)}\cdot\frac{P^e(d_k=+1)}{P^e(d_k=-1)}\\&=\ln\frac{\sum_{n=1}^{N_s}p\left(\mathbf{y}|\mathbf{d}_{-k}^{(n)},d_k=+1\right)P^e\left(\mathbf{d}_{-k}^{(n)}\right)}{\sum_{n=1}^{N_s}p\left(\mathbf{y}|\mathbf{d}_{-k}^{(n)},d_k=-1\right)P^e\left(\mathbf{d}_{-k}^{(n)}\right)}+\lambda_2^e(d_k).\end{aligned} \quad (23)$$

Recalling that $\lambda_1^e(d_k)=\lambda_1(d_k)-\lambda_2^e(d_k)$, from (23), we obtain

$$\lambda_1^e(d_k)=\ln\frac{\sum_{n=1}^{N_s}p\left(\mathbf{y}|\mathbf{d}_{-k}^{(n)},d_k=+1\right)P^e\left(\mathbf{d}_{-k}^{(n)}\right)}{\sum_{n=1}^{N_s}p\left(\mathbf{y}|\mathbf{d}_{-k}^{(n)},d_k=-1\right)P^e\left(\mathbf{d}_{-k}^{(n)}\right)}. \quad (24)$$

In Section X, through computer simulations, we show that the accuracy of the $L$-values obtained through (25) is much inferior to that obtained through the methods of this paper.

## VII. Summary of the Detection Algorithms

Summarizing our results so far, we list three multiuser/MIMO detection algorithms:

- statistical inference (SI); an implementation based on (25);
- Markov chain Rao–Blackwellization (MCRB); an implementation based on (12);
- Markov chain Rao–Blackwellization with uniform sampling (MCRB-U); an implementation based on (13).

Each of these algorithms can have a number of different implementations depending on how MCMC simulations are performed, e.g., using a single Markov chain or a set of parallel Markov chains.

Implementations of the SI and MCRB algorithms are obvious. To implement the SI algorithm, one should realize one or more Gibbs samplers, allow sufficient time for the Markov chain(s) to converge, collect sufficient samples of $d_k^{(n)}$, obtain $P(d_k=+1|\mathbf{y},\boldsymbol{\lambda}_2^e)$ according to (25), and finally calculate $\lambda_1(d_k) = \ln\left(P(d_k=+1|\mathbf{y},\boldsymbol{\lambda}_2^e)/1-P(d_k=+1|\mathbf{y},\boldsymbol{\lambda}_2^e)\right)$. The MCRB is implemented similarly, with the difference that (12) is used instead of (25).

In the MCRB-U algorithm, the samples of $\mathbf{d}_{-k}$ should be from a uniform distribution that covers the domain of $P\left(\mathbf{d}_{-k}^{(n)}|\mathbf{y},\boldsymbol{\lambda}_2^e\right)$. Vaguely speaking, the feasible implementation that we adopt takes samples from a uniform distribution over a range where $P\left(\mathbf{d}_{-k}^{(n)}|\mathbf{y},\boldsymbol{\lambda}_2^e\right)$ is significant. How such samples could be obtained is not obvious since the function $P\left(\mathbf{d}_{-k}^{(n)}|\mathbf{y},\boldsymbol{\lambda}_2^e\right)$ itself is unknown; it is why we use MCMC simulation. Fortunately, our basic knowledge of the multiuser signals and the underlying Markov chain allow us to suggest an ad hoc method for generation of the desired samples of $\mathbf{d}_{-k}$. Starting from any arbitrary initial value $\mathbf{d}^{(-N_b)}$, $\mathbf{d}^{(n)}$ converges toward and walks through the combinations of data symbols for which $P\left(\mathbf{d}_{-k}^{(n)}|\mathbf{y},\boldsymbol{\lambda}_2^e\right)$ is significant. One can expand the range of the samples $\mathbf{d}_{-k}^{(n)}$ by running a number of Gibbs samplers in parallel. However, these samples will not be uniformly distributed. To resolve this problem, we ignore the repetitions of $\mathbf{d}_{-k}^{(n)}$ in the obtained sample set before their insertion in (24). Moreover, since any sample $\mathbf{d}_{-k}^{(n)}$ can improve the result of (24), we ignore the burn-in period, i.e., we set $N_b = 0$, and thus all the Gibbs samples (except repetitions) are used in performing (24).

Fig. 2 presents a pictorial presentation of the process of choosing samples in the MCRB-U algorithm. The horizontal axis indicates states of the Markov chain and the curve shows the probability of states. The circles/particles show the samples. The first sample of each state is kept in the pool of samples, while the subsequent repetitions are ignored. If sufficient number of samples are collected, the final sample pool will cover almost all the states with significant probability. This may be thought as a set of uniformly (and perfectly) distributed samples over the range of the significant states. At the same time, we should acknowledge that when the size of the underlying state space is large and the number of Gibbs samples is small, as is the case for most of the results presented in Section X, the Gibbs sampler may fail to include many of the states with significant probabilities in the sample pool. This, mixed with the fact that the samples generated by a Gibbs sampler are correlated, can lead to a sample pool that may be far from uniformly distributed. Nevertheless, the computer simulations presented in this paper and also in [57] show that the proposed MCRB-U algorithm, with small number of samples, performs very close to the capacity. This observation may be explained as follows. The LLR computation involves the ratio of the probabilities $P\left(d_k=+1|\mathbf{y},\boldsymbol{\lambda}_2^e\right)$ and $P\left(d_k=-1|\mathbf{y},\boldsymbol{\lambda}_2^e\right)$. Using a small and possibly biased set of samples affects both terms equally (at least, approximately) and therefore the computed LLR remains accurate. It turns out that an analysis/proof of this proposition is quite difficult and should remain a subject of future studies.

## VIII. Max-Log Approximation

We recall from the literature of turbo decoding [49] that a log ratio such as (24) could be approximated as

$$\lambda_1^e(d_k) \approx \ln \frac{\max_n p\left(\mathbf{y}|\mathbf{d}_{-k}^{(n)}, d_k=+1\right) P^e\left(\mathbf{d}_{-k}^{(n)}\right)}{\max_n p\left(\mathbf{y}|\mathbf{d}_{-k}^{(n)}, d_k=-1\right) P^e\left(\mathbf{d}_{-k}^{(n)}\right)} \quad (26)$$

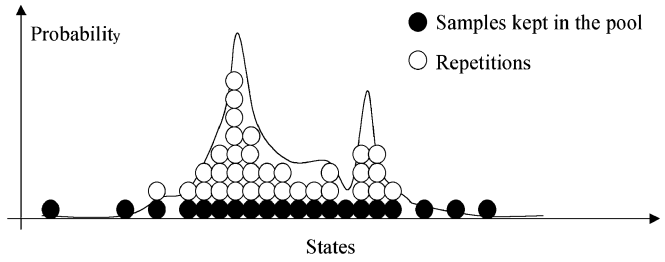



Fig. 2. A pictorial presentation of choosing samples in the MCRB-U algorithm.

without significant loss in the performance of the detector. This, which is known as max-log approximation, simplifies the MCRB-U algorithm with two respects. First, it avoids the summations in (24). Secondly, and more importantly, it avoids the process of looking for repetitions of $\mathbf{d}^{(n)}$. This can be a great saving, since the number of comparisons required for deleting repetitions grows with the square of $N_s$.

## IX. Gibbs Sampler Stalling and its Remedy: Modified MCRB-U Algorithm

A serious problem that may limit the applicability of MCMC techniques in many applications is what we wish to refer to as the "stalling problem." In certain situations, particularly, when signal-to-noise ratio (SNR) is high, one may find that in a Markov chain, transition from some states to others can only occur with a very low probability. This particularly is a consequence of the use of Gibbs sampler which was adopted to simplify the realization of the MCMC simulator. To see this, consider a case where for a selected sample $\mathbf{d}^{(n)}$, $|\mathbf{y}-\mathbf{A}\mathbf{d}^{(n)}|$, for any $k=1,2,\ldots,K$, is significantly smaller than $\left|\mathbf{y}-\mathbf{A}\mathbf{d}_k^{(n)}\right|$, where $\mathbf{d}_k^{(n)}$ is the same $\mathbf{d}^{(n)}$ with its $k$th element changed, i.e., the intermediate samples of $\mathbf{d}$ while walking through one iteration of the Gibbs sampler. In this situation when SNR is high, one may find that $P\left(\mathbf{d}=\mathbf{d}^{(n)}|\mathbf{y},\boldsymbol{\lambda}_2^{(n)}\right) \approx 1$ and $P\left(\mathbf{d}=\mathbf{d}_k^{(n)}|\mathbf{y},\boldsymbol{\lambda}_2^{(n)}\right) \approx 0$, for $k=1,2,\ldots,K$. In this situation, obviously, the Gibbs sampler most likely holds on to the sample $\mathbf{d}^{(n)}$ and thus the Markov chain may remain at the same state for a very long length. Hence, to obtain meaningful results, the MCMC simulator should be run for a very long time. This leads to a computationally expensive system whose implementation may become prohibitive. Therefore, development of elegant methods that avoid the problem of stalling is of great interest.

A general philosophy that one may adopt to prevent the stalling problem is to amend the probability of transitions in Gibbs sampler such that to avoid very small probabilities. Among various possible choices, one possible solution that we found effective is to run the Gibbs sampler with a higher noise level but use the correct value of the noise variance in the computation of the probabilities in (24). Computer simulations presented in Section X reveal that this modified version of the MCRB-U indeed results in significant improvement, especially in highly loaded cases when $K > 2N$. We refer to this algorithm as *modified* MCRB-U.

It is worth noting that the above philosophy has similarity with change of scaling, also called *temperature,* in simulated annealing [35]. Here, an increased temperature allows for faster moves on the surface of a performance function whose minimization/maximization is the goal. In the context of this paper, noise power plays the role of temperature.

## X. Simulation Results

In this section, we present a range of simulation results with the following goals:

- to show that the methods developed in this paper outperform the widely used bit counting method [16]–[30];
- to study the relative behaviors of MCRB and MCRB-U algorithms;
- to compare the proposed detection algorithms with the state-of-the-art techniques. In particular, comparisons will be made with the MMSE-based turbo receiver of [8] and the sphere decoding of [51].

### A. Comparison With the Bit Counting Method

To compare the bit counting method with the methods proposed in this paper, we consider a synchronous CDMA system with 16 users and a spreading gain of 8. We assume binary data and SNR of 3 dB. We assume that the extrinsic information fed back from the FEC decoder(s) to the multiuser detector has a reliability of 50%, where reliability is defined as the mutual information between the soft input information that is passed to the detector and the actual information bits [47], i.e., $I\left(\lambda_2^e(d_k); d_k\right)$. In order to have full control over the reliability values, there is no turbo loop here. The $L$-values are generated randomly, following the method discussed in [41].

Fig. 3 shows the outcome of the multiuser detector for 1000 iterations of Gibbs sampler. From these we assume that the first ten iterations are for burn-in, and after the burn-in period, we have either counted the number of occurrences of $+1$s and $-1$s to obtain the $L$-values (SI algorithm) or have averaged the respective probabilities according to (12) (MCRB algorithm). The expected $L$-values are also indicated. The following observations are made.

- The results obtained by using (12) are relatively close to the expected $L$-values, virtually from the beginning of all the curves. This shows that after the burn-in period, averaging over a small number of samples may be sufficient for reliable estimates.
- The LLR estimates obtained by bit counting, on the other hand, are quite unreliable. In this particular example, for four of the bits (plots in the first and second columns of the top row and plots in the third and fourth columns of the second row), the $L$-values have remained very high and far from their true values even after 1000 iterations of the Gibbs sampler. The actual $L$-values in these cases were $\pm\infty$ (no occurrence of the opposite bits has been observed). We have limited these values to $\pm 26$ in order not to drop out of the diagrams. For six of the bits, we need 100 or more Gibbs samples for the $L$-values to converge. Only those bits whose $L$-values are small (in magnitude) give good results with bit counting. This observation may be explained as follows. A positive large $L$-value implies that the associated information bit, say, $d_k$, takes value of $+1$ with a probability that is close to one, and thus takes the value of $-1$ with a probability that is close to zero. Therefore, to find an accurate estimate of the $L$-value $\ln(P(d_k=+1)/P(d_k=-1))$, one needs to observe occurrences of $d_k$ in a very long simulated Markov chain before seeing sufficient number of occurrences of $d_k=-1$. Obviously, a similar argument applies when an $L$-value is a large negative number. On the other hand, a small (positive or negative) $L$-value implies that $P(d_k=+1)$ and $P(d_k=-1)$ are not very different. Hence, a short Markov chain may be sufficient for obtaining good estimates of $P(d_k=+1)$ and $P(d_k=-1)$ and thus to obtain an acceptable estimate of $\ln(P(d_k=+1)/P(d_k=-1))$.

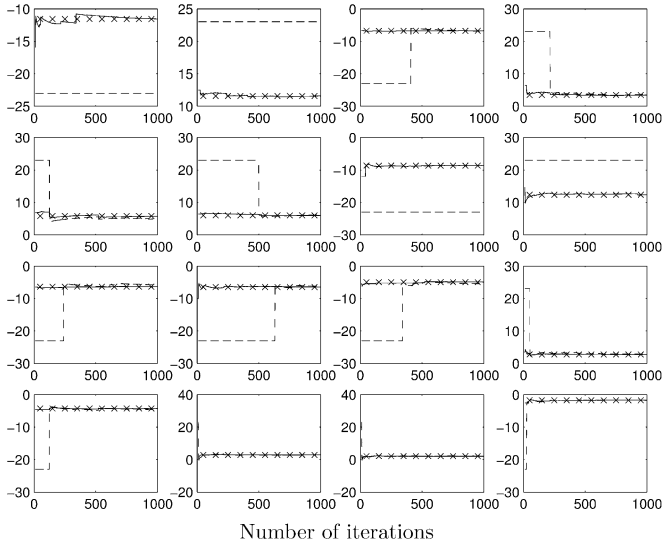



Fig. 3. Comparison of the $L$-values obtained using bit counting and those obtained by using (12). Dashed line plots are bit counting results; full line plots are results obtained from (12); crosses show the expected $L$-values. The very large $L$-values are limited to $\pm 26$ in order not to drop out of the figures.

### B. Comparison of MCRB and MCRB-U Algorithms

To compare the relative behavior of MCRB and MCRB-U algorithms, we evaluate the extrinsic $L$-values of the information bits at the output of the multiuser/MIMO detector for a variety of system setups. Our experiments, which were repeated for both multiuser and MIMO setups, had the same conclusion consistently [58]. Here, we present those of an MIMO system with four transmit and two receive antennas. QPSK symbols are used. The channel code is a rate 1/2 convolutional code with the generator polynomials $1+D+D^2$ and $1+D^2$. MCRB algorithm is implemented in three different ways: i) based on a single Markov chain with a burn-in period of 20; ii) based on four parallel Markov chains with a burn-in period of 5; iii) based on a set of parallel Markov chains, each of length 10 and burn-in period of 5. MCRB-U and modified MCRB-U algorithms are implemented based on a set of parallel Markov chains; each runs for four iterations of the Gibbs sampler. For each case the exact $L$-values are calculated and compared with the estimated $L$-values obtained from the algorithms. The difference between the exact and estimated $L$-values is then squared

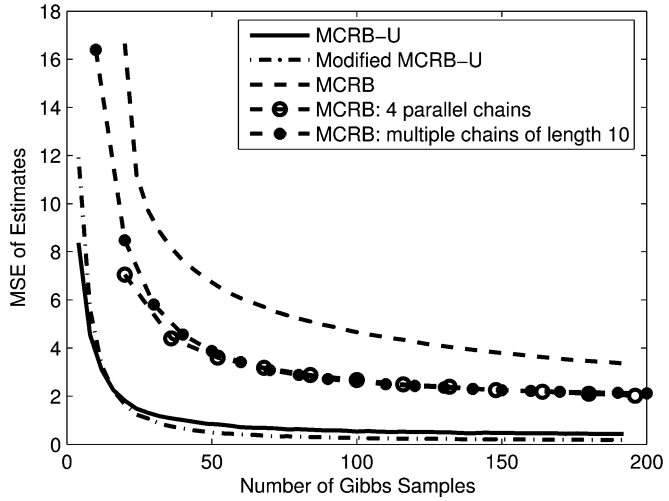



Fig. 4. Comparison of accuracy of MCRB, MCRB-U and modified MCRB-U algorithms in estimating the $L$-values. Presented are the MSE of the estimates asa function of the number of Gibbs samples used for each estimate.

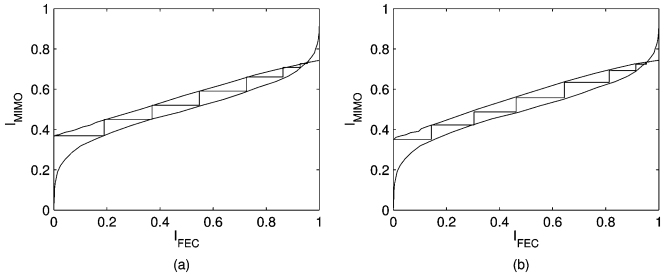



Fig. 5. EXIT charts of (a) MCRB and (b) MCRB-U algorithms in a MIMO setup with four transmit and two receive antennas at SNR of 8 dB. MCRB algorithm is implemented using a single Markov chain with 192 Gibbs samples for each data bit. A burn-in period of 20 Gibbs samples is assumed. MCRB-U algorithm uses 12 Gibbs samples from three independent Markov chains to obtain the $L$-value of each information bit. The similar performance of the two algorithms for the selected parameters matches the prediction made by MSE results of Fig. 4.

and ensemble averaged over a large number of the channel realizations to obtain an estimate of the mean square error (MSE) of the estimates. The SNR at the receiver input is set equal to 8 dB. In the implementation of MCRB and MCRB-U algorithms, the true value of SNR is used to generate the Gibbs samples. For modified MCRB-U algorithm, SNR of 5 dB is considered for generation of the Gibbs samples. The reliability of the $L$-values received from the FEC decoder is set equal to 0.2. This reliability value is mapped to the $L$-values statistics (mean and variance) as discussed in [41] and accordingly the values of $\lambda_1^e(\cdot)$ are generated randomly for each realization of the channel independently. Fig. 4 shows the results of this experiment. Note that for MCRB algorithm with single Markov chain, the curve starts from the twenty-first iteration of the Gibbs sampler since we assume a burn-in period of 20 iterations. Also, recall that MCRB-U and modified MCRB-U algorithms do not have any burn-in period. As seen, the MCRB-U algorithm and its modified version perform significantly better than MCRB algorithm. Running MCRB algorithm with parallel Markov chains improves the results, but, still, MCRB-U algorithm performs significantly better.

Why MCRB-U performs significantly better than MCRB algorithm is not clear at this time. It remains a subject of future study. Our guess, at this time, is that the importance sampling approximation (10) plays the key role. The presence of the terms $(f(x_n)/f_a(x_n))$ in both numerator and denominator improves the estimates of the probabilities and thus results in more accurate $L$-values.

To further confirm the above observation, we present extrinsic information transfer (EXIT) chart analysis and bit-error-rate (BER) results of the two algorithms. EXIT chart is a popular method for approximate analysis of convergence of iterative decoders [40]–[42]. It was invented by ten Brink [40] and subsequently used by many authors for better understanding of the turbo and low-density parity check codes behaviors as well as for code designs [41], [43]–[45]. Moreover, the method was found applicable to other detection schemes that work based on the so-called turbo principles [46]. We also found EXIT chart analysis very helpful in this paper. In EXIT charts, the idea is to analyze the input/output characteristic of each decoding/detection block, where the input is the mutual information between the transmitted bits and the available $L$-values to the block (the extrinsic information from the other block) and the output is the extracted extrinsic mutual information at the block output; thus, the name extrinsic information transfer chart. The EXIT curves corresponding to the two blocks of an iterative decoder are plotted in one graph and a zig-zag trajectory that jumps horizontally and vertically between the two curves predicts the evolution of the decoding progression. We note that EXIT curves are obtained by making certain assumptions on the distribution of $L$-values [41], [42]. However, since in an actual decoder the distribution of the $L$-values may be different from what has been assumed, EXIT charts are only capable of predicting an approximate behavior of the decoder. Here, we use EXIT charts to predict the behavior of the proposed algorithms first. The corresponding BER curves are then presented in order to confirm the accuracy of the conclusions drawn from EXIT charts.

According to Fig. 4, if we choose 12 Gibbs samples for MCRB-U algorithm, to obtain the same performance, MCRB algorithm (with single Markov chain) requires 192 Gibbs samples (20 for burn-in and 172 samples for averaging the probabilities). Fig. 5(a) and (b) shows the EXIT chart analysis of the two algorithms based on these number of Gibbs samples. SNR values and system parameters here are also similar to those in Fig. 4. As one would expect, both algorithms show similar performance, albeit, MCRB requires significantly more Gibbs samples and hence is significantly more complex.

To confirm the accuracy of the EXIT charts, Fig. 6(a) and (b) shows the BER curves of the MCRB and MCRB-U algorithms. Similar to Fig. 5, the number of Gibbs samples used for each information bit is 192 and 12 for MCRB and MCRB-U, respectively. BER curves are shown for successive iterations of the turbo detector loop. The results of iterations 1, 3, 5, 7, and 9 are given. As predicted by the MSE estimates and EXIT chart analysis, the two algorithms perform very similarly for $E_b/N_o$ of 5 dB, equivalent to SNR of 8 dB—Figs. 4 and 5 were generated assuming an SNR of 8 dB. However, at higher SNR, MCRB

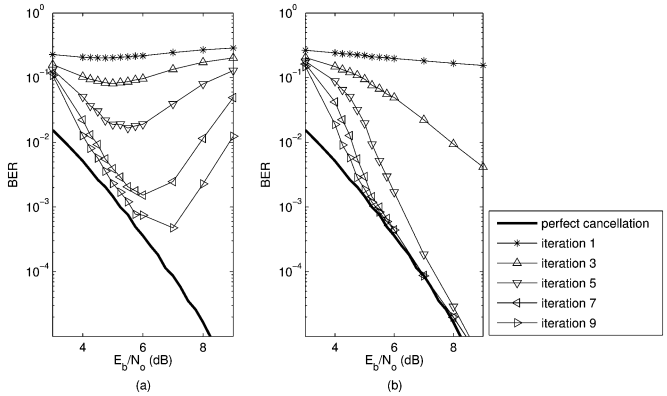



Fig. 6. BER results of (a) MCRB and (b) MCRB-U algorithms in a MIMO setup with four transmit and two receive antennas. The "perfect cancellation" plots correspond to the case where data symbols from other transmit antennas are perfectly cancelled before detection of each symbol. MCRB algorithm operates based on one Markov chain with a burn-in period of 20 and averaging over 172 Gibbs samples. MCRB-U algorithm obtains each L-value by averaging over 12 Gibbs samples from three independent Markov chains.

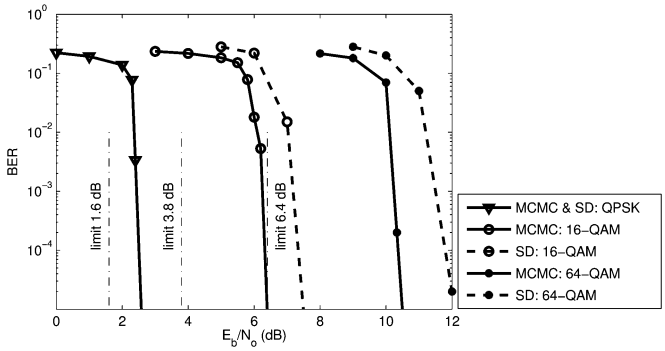



Fig. 7. BER results of the MCMC detector and sphere decoding. The vertical lines show the capacity limits (1.6 dB for QPSK, 3.8 dB for 16-QAM, and 6.4 dB for 64-QAM).

fails and BERs increase as SNR increases. This may be explained as follows. As noted in Section VII, at higher values of SNR some of the transition probabilities in the underlying Markov chain may become very small and as a result the MCMC simulator may get trapped in a state for many iterations before moving to another state. Hence, only a very small number of distinct samples of $\mathbf{d}$ may be observed, and therefore the estimated $L$-values may become very inaccurate. Clearly, the results could be improved by increasing the number of samples. However, this would be at a (much) higher computational cost, which, of course, is undesirable.

### C. Comparison With Sphere Decoding

The MCMC simulator may be viewed as a statistical tool for searching and finding the combinations of the transmitted information bits/symbols that correspond to significant terms in the summation (4). By using these combinations only, one can obtain a good estimate of the associated probabilities and hence the desired $L$-values. An alternative method that also finds the combinations of the transmitted information bits/symbols with significant contribution to the desired probabilities, but from a deterministic point of view, uses the search algorithm of [50]. This algorithm that is commonly known as *sphere decoding* (SD) has recently been studied and applied to the detection of MIMO signals by many researchers, e.g., [51]–[55]. In [51], in particular, a number of simulation results that show the performance of SD for MIMO detection in a receiver structure similar to Fig. 1 are presented. Other authors that have proposed alternative versions of the SD [53], [55] have also confirmed a similar performance. In this section, we present the results of MCRB-U algorithm for an MIMO setup similar to [51] and contrast our results with those presented there. We present the results for an MIMO channel with eight transmit and eight receive antennas. As in [51], the cases of QPSK, 16-QAM, and 64-QAM signaling are tested. Also, the channel code is a rate 1/2 parallel concatenated (turbo) code with feedforward polynomial $1+D^2$ and feedback polynomial $1+D+D^2$ and interleaver size of 9216 information bits.

Fig. 7 compares the BER results of MCRB-U algorithm and the SD algorithm of [51]. The number of Gibbs samples used for each LLR estimation in the case of QPSK is 40, obtained from eight parallel Markov chains. This increases to 400 samples, obtained from 20 parallel Markov chains, in the case of 16-QAM, and to 900, obtained from 30 parallel Markov chains, in the case of 64-QAM. In order to give compatible results with those in [51], the max-log version of the MCRB-U algorithm, i.e., an implementation based on (31), is used to obtain the results shown in Fig. 7. As seen, the MCRB-U algorithm is able to reduce the gap between the BER curves and the capacity limit from 3.7 to 2.6 dB in the case of 16-QAM and from 5.6 dB to 4 dB in the case of 64-QAM signaling. For the case of QPSK, the MCMC detector performs similar to SD. In the latter case, SD, the way it is set up in [51], searches over all possible values of symbols, i.e., it is an exact (max-log) MAP detector. For all cases we have studied, replacement of (31) by (30) results in 0.2–0.5 dB improvement.

### D. Highly Loaded Cases

As was noted earlier, we differentiate between the cases of underloaded ($K < N$), fully loaded ($K = N$), and overloaded ($K > N$). Generally speaking, all detection algorithms deteriorate in performance and grow in complexity as $K/N$ grows. For $K \geq N$, the complexity of SD grows exponentially with $K - N$ and thus becomes prohibitive for moderate and large values of $K - N$. In [51] and [55], the case $K > N$ is avoided by using linear dispersive codes which effectively converts an overloaded MIMO system to an under- or fully loaded system. This approach usually results in an increase in the detector complexity.

The MMSE (with soft cancellation) detector as originally proposed in [8] is applicable to the cases where $K \leq N$. Later developments showed that this method could also be extended and used when $K > N$, at the cost of higher complexity; see the comments in the first paragraph of Section I.

In this section, we present some simulation results that reveal the behavior of the proposed MCMC detectors as the system load increases. We also present the results of the MMSE detector for comparison. We consider a CDMA setup with a spreading gain of $N = 10$. The spreading gain vectors $\mathbf{a}_k$ (i.e., the columns of $\mathbf{A}$) are real vectors with i.i.d. entries from

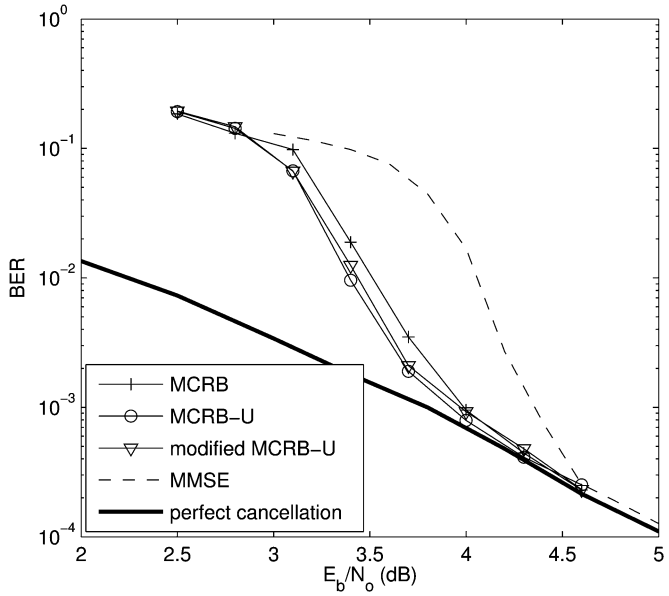



Fig. 8. BER results of MCRB, MCRB-U, and modified MCRB-U algorithms in a CDMA setup with $K = 18$ and $N = 10$. The results of the MMSE detector and those of the case where signals from other users are perfectly cancelled (equivalent to that of a single user channel) are presented for comparison.

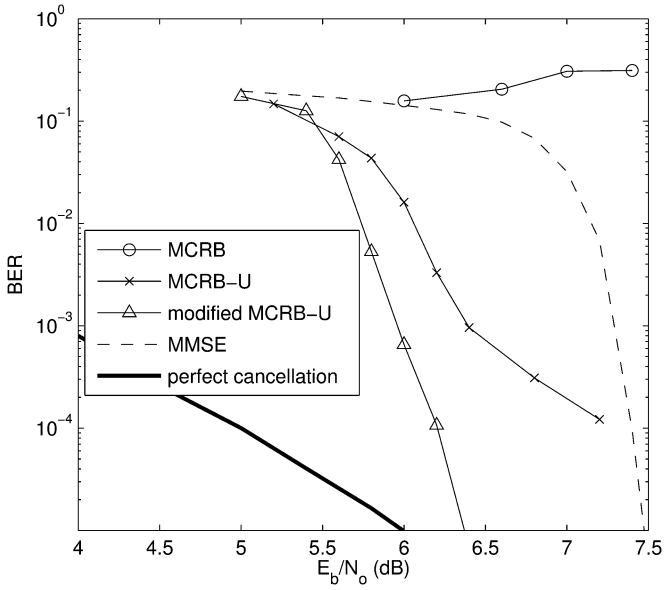



Fig. 9. BER results of MCRB, MCRB-U, and modified MCRB-U algorithms in a CDMA setup with $K = 24$ and $N = 10$. The results of the MMSE detector and those of the case where signals from other users are perfectly cancelled (equivalent to that of a single user channel) are presented for comparison.

$\pm(1/\sqrt{N})$. The channel code is the convolutional code that was used to generate the results of Figs. 5 and 6.

Fig. 8 shows the results for the case when $K = 18$, after the tenth iteration of the turbo loop. For MCRB algorithm, one Markov chain with $N_b = 30$ and $N_s = 30$ (60 Gibbs samples per transmit bit) is used. Four parallel Markov chains with $N_b = 0$ and $N_s = 5$ (20 Gibbs samples per transmit bit) are used for MCRB-U and modified MCRB-U algorithms. Hence, here, the latter algorithms are three times less complex than MCRB algorithm. Also, shown for comparison is the performance of a receiver where signals from other users are perfectly cancelled while detecting each user data. This, which is also known as single user performance, provides a lower bound for the system performance. As seen, the MMSE and all versions of the MCMC detectors approach the performance of such an ideal receiver at higher values of SNR. However, here, the MCMC detectors approach this lower bound by a margin of 0.5 dB when compared with the MMSE detector.

Fig. 9 shows the results of the same experiment for the case when $K$ is increased to 24. Here also the results are those obtained after ten iterations of the turbo loop. For MCRB algorithm, three Markov chains with $N_b = 100$ and $N_s = 100$ (total of 600 Gibbs samples per user bit) are used. Eight parallel Markov chains with $N_b = 0$ and $N_s = 30$ (240 Gibbs samples per user bit) are used for MCRB-U and modified MCRB-U algorithms. As seen, even though we have increased the number of Gibbs samples significantly, MCRB algorithm fails. MCRB-U performs better than the MMSE detector at lower values of SNR, but shows a trend of loosing this gain at higher values of SNR. The modified MCRB-U, on the other hand, shows a better and more robust behavior over a wider range of SNR values. It has a gain of 1.2 dB over the MMSE detector and approaches the single user capacity around 6.5 dB.

We recall that in the modified MCRB-U algorithm, in order to increase the mobility among the states, the underlying Markov chains are run with an assumed higher noise level than the actual channel noise. Namely, if the true noise variance in the channel is $\sigma_n^2$, a noise variance of $\sigma_n'^2$ is assumed while running the MCMC simulations. The ratio $\sigma_n'^2/\sigma_n^2$ is thus a design parameter that has to be selected properly. At this stage of our research we do not have any theoretical analysis that can be used for the selection of this ratio. However, computer simulations that we have performed suggest that for SNR values in the range of 10 dB or less, an initial choice of $\sigma_n'^2$ two to three decibels above $\sigma_n^2$ for the first iteration of the turbo loop, and gradual reduction of it to $\sigma_n^2$, leads to good results. For the results in Fig. 8, we started the modified MCRB-U algorithm with $\sigma_n'^2/\sigma_n^2 = 1.58$ ($\sim$2 dB) and linearly reduced it to 1 over ten iterations. This was increased to 2 ($\sim$3 dB) in Fig. 9.

The mobility of a Markov chain and its capability of convergence to a desired distribution should be balanced. In the first few iterations of the turbo loop, when the a priori information from the FEC decoder(s) is small, a larger value of $\sigma_n'^2$ helps in the mobility of the Markov chain, allowing coverage of more states, thus increasing the chance of visiting most of the states that have significant contribution to the summations such as (4). On the other hand, in the later iterations, one finds that the Markov chains are dominantly driven by the extrinsic information (bit/symbol probabilities) from the FEC decoder(s). In this situation, augmentation of the noise variance becomes counterproductive and results in some bias in the estimated $L$-values.

## XI. Conclusion and Additional Comments

Application of Markov chain Monte Carlo simulation to soft detection of multiuser and MIMO signals was studied. Unlike

the previous reports that widely work based on statistical inference (i.e., the $L$-values are obtained by counting the number occurrences of each information bit value in an MCMC simulator), we presented novel statistical methods that were developed by viewing the underlying problem as a multidimensional Monte Carlo integration. We found that this approach leads to results that are similar to those which would be obtained through a proper Rao–Blackwellization technique [35]. A number of detection algorithms were developed and their performance behaviors studied and compared against some of the well-known techniques in the literature. In particular, we found that the proposed algorithms were superior to those that have been widely adopted in the past literature of MCMC techniques. Moreover, comparison with the existing results of SD and MMSE detectors showed that the proposed algorithms lead to superior performance.

Although we developed the basic foundation of a new class of detection algorithms and showed their superior performance when compared with the most recent results in the literature, the study reported in this paper did not cover many aspects of these algorithms. One important issue that needs attention is theoretical analysis of the proposed algorithms. We revealed the differences between the algorithms through computer simulations only. Theoretical analyses can be helpful in better understanding of the algorithms and may also lead to new algorithms with improved performance.

Another aspect of the developed algorithms that is equally important but we did not mention in this paper is the computational complexity—in particular, how the proposed algorithms compare with the existing algorithms, e.g., SD and MMSE detectors. In a recent work [59], [60], we have presented a detailed comparison of the MCMC detection method and the list SD algorithm of [51]. The general conclusion is that as the system size (the number of users in the multiuser case, and the number of transmit antennas in the MIMO case) increases, MCMC becomes significantly less complex than SD. Our ongoing research on the implementation of MCMC detection algorithms shows that they can be implemented in a very efficient and highly pipelined architecture with a complexity that is even far lower than those reported in [59] and [60]. The results of this research will be reported in the future.

## APPENDIX
## EXTENSION TO COMPLEX SYMBOLS

Throughout this paper, for convenience of presentation, we limited our discussion to the cases where data symbols were binary, i.e., one information bit was carried by each symbol. Here, we consider the case where data symbols are chosen from a (possibly complex) alphabet $\mathcal{A}=\{\alpha_1,\alpha_2,\ldots,\alpha_L\}$ and present the relevant detection formulas.

The Gibbs sampler that was discussed is equally applicable to the general case of $L$-ary data symbols. Extensions of the detection formulae (16) and (24) are also rather straightforward. Here, for brevity, we only review an extension of (24). We also limit our discussion to the simple but important case where $J=\log_2 L$ is an integer, i.e., where an integer number of coded bits is carried by each symbol.

When $J$ is an integer, the first line of (23) may be generalized as in (27), shown at the bottom of the page, where $b_j(d_k)$ is the $j$th bit of $d_k$ and $U_j^+(\mathcal{A})$ and $U_j^-(\mathcal{A})$ are the subsets of the alphabet $\mathcal{A}$ in which the $j$th bit of each element is $+1$ and $-1$, respectively. We also note that because of the interleaving effect in the channel decoder, the extrinsic information of the

$$\lambda_1(b_j(d_k))=\ln\frac{\sum_{U_j^+(\mathcal{A})}\sum_{n=1}^{N_s}P\left(\mathbf{y}|\mathbf{d}_{-k}^{(n)},d_k=\alpha_i\right)P^e\left(\mathbf{d}_{-k}^{(n)}\right)P^e(d_k=\alpha_i)}{\sum_{U_j^-(\mathcal{A})}\sum_{n=1}^{N_s}P\left(\mathbf{y}|\mathbf{d}_{-k}^{(n)},d_k=\alpha_i\right)P^e\left(\mathbf{d}_{-k}^{(n)}\right)P^e(d_k=\alpha_i)} \tag{27}$$

$$\lambda_1(b_j(d_k))=\ln\frac{\sum_{U_j^+(\mathcal{A})}\sum_{n=1}^{N_s}P\left(\mathbf{y}|\mathbf{d}_{-k}^{(n)},d_k=\alpha_i\right)P^e\left(\mathbf{d}_{-k}^{(n)}\right)P^e_{-j}(d_k=\alpha_i)}{\sum_{U_j^-(\mathcal{A})}\sum_{n=1}^{N_s}P\left(\mathbf{y}|\mathbf{d}_{-k}^{(n)},d_k=\alpha_i\right)P^e\left(\mathbf{d}_{-k}^{(n)}\right)P^e_{-j}(d_k=\alpha_i)}\cdot\frac{P^e(b_j(d_k)=+1)}{P^e(b_j(d_k)=-1)} \tag{29}$$

$$\lambda_1^e(b_j(d_k))=\ln\frac{\sum_{U_j^+(\mathcal{A})}\sum_{n=1}^{N_s}P\left(\mathbf{y}|\mathbf{d}_{-k}^{(n)},d_k=\alpha_i\right)P^e\left(\mathbf{d}_{-k}^{(n)}\right)P^e_{-j}(d_k=\alpha_i)}{\sum_{U_j^-(\mathcal{A})}\sum_{n=1}^{N_s}P\left(\mathbf{y}|\mathbf{d}_{-k}^{(n)},d_k=\alpha_i\right)P^e\left(\mathbf{d}_{-k}^{(n)}\right)P^e_{-j}(d_k=\alpha_i)} \tag{30}$$

$$\lambda_1^e(b_j(d_k)) \approx \ln \frac{\max_{U_j^+(\mathcal{A}),n} P\left(\mathbf{y}|\mathbf{d}_{-k}^{(n)}, d_k=\alpha_i\right) P^e\left(\mathbf{d}_{-k}^{(n)}\right) P_{-j}^e(d_k=\alpha_i)}{\max_{U_j^+(\mathcal{A}),n} P\left(\mathbf{y}|\mathbf{d}_{-k}^{(n)}, d_k=\alpha_i\right) P^e\left(\mathbf{d}_{-k}^{(n)}\right) P_{-j}^e(d_k=\alpha_i)} \tag{31}$$

bits within each symbol may be assumed independent of each other. This implies that

$$P^e(d_k=\alpha_i) = \prod_{j=1}^{J} P^e(b_j(d_k)=b_j(\alpha_i)). \tag{28}$$

Using (28), (27) may be written as shown in (29) at the bottom of the previous page, where $P_{-j}^e(d_k=\alpha_i) = \prod_{l=1,l\neq j}^{L} P^e(b_l(d_k) = b_l(\alpha_i))$. Next, recalling that $\lambda_1(b_j(d_k)) = \lambda_1^e(b_j(d_k)) + \lambda_2^e(b_j(d_k))$ and $\lambda_2^e(b_j(d_k)) = \ln\left(P^e(b_j(d_k)=+1)/P^e(b_j(d_k)=-1)\right)$, we obtain, from (29), (30), shown at the bottom of the previous page.

Following the max-log approximation (26), from (30), we obtain (31) as shown at the top of the page.

## REFERENCES


[1] A. J. Viterbi, *CDMA: Principles of Spread Spectrum Communication*. Englewood Cliffs, NJ: Prentice-Hall, 1995.

[2] S. Verdù, *Multiuser Detection*. Cambridge, U.K.: Cambridge Univ. Press, 1998.

[3] F. Swarts, P. van Rooyen, I. Oppermann, and M. P. Lötter, *CDMA Techniques for Third Generation Mobile Systems*. Norwood, MA: Kluwer Academic, 1999.

[4] S. Verdú, "Minimum probability of error for asynchronous Gaussian multiple-access channels," *IEEE Trans. Inf. Theory*, vol. IT-32, no. 1, pp. 85–96, Jan. 1986.

[5] P. D. Alexander, M. C. Reed, J. A. Asenstorfer, and C. B. Schlegel, "Iterative multiuser interference reduction: Turbo CDMA," *IEEE Trans. Commun.*, vol. 47, no. 7, Jul. 1999.

[6] M. C. Reed, C. B. Schlegel, P. D. Alexander, and J. A. Asenstorfer, "Iterative multiuser detection for CDMA with FEC: Near single-user performance," *IEEE Trans. Commun.*, vol. 46, no. 12, pp. 1693–1699, Dec. 1998.

[7] M. Moher, "An iterative multiuser decoder for near-capacity communications," *IEEE Trans. Commun.*, vol. 47, no. 7, pp. 870–880, Jul. 1998.

[8] X. Wang and H. V. Poor, "Iterative (Turbo) soft interference cancellation and decoding for coded CDMA," *IEEE Trans. Commun.*, vol. 47, no. 7, pp. 1046–1061, Jul. 1999.

[9] L. Trichard, J. Evans, and I. Collings, "Large system analysis of linear multistage parallel interference cancellation," *IEEE Trans. Commun.*, vol. 50, no. 11, pp. 1778–1786, Nov. 2002.

[10] C. Schlegel and Z. Shi, "Performance and complexity of CDMA iterative multiuser detection," in *Proc. IEEE Inf. Theory Workshop*, Mar. 2003, pp. 111–114.

[11] G. J. Foschini, "Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas," *Bell Labs Tech. J.*, vol. 1, pp. 41–59, Aug. 1996.

[12] G. J. Foschini and M. J. Gans, "On limits of wireless communications in a fading environment," *Wirelss Commun. Mag.*, vol. 6, pp. 311–335, Mar. 1998.

[13] E. Telatar, "Capacity of multi-antenna Gaussian channels," *Eur. Trans. Telecomm.*, vol. 10, no. 6, pp. 585–595, Nov.–Dec. 1999.

[14] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo codes," *IEEE Trans. Commun.*, vol. 44, no. 10, pp. 1261–1271, Oct. 1996.

[15] V. Buchoux, O. Cappe, and E. Moulines, "Turbo multiuser detection for coded DS-CDMA systems: A Gibbs sampling approach," in *Conf. Rec. 34th Asilomar Conf. Signals, Systems Computing*, vol. 2, Nov. 1, 2000, pp. 1426–1430.

[16] R. Chen, J. S. Liu, and X. Wang, "Convergence analyses and comparisons of Markov chain Monte Carlo algorithms in digital communications," *IEEE Trans Signal Process.*, vol. 50, no. 2, pp. 255–270, Feb. 2002.

[17] T. M. Gatherer, A. X. Wang, and R. Chen, "Interference cancellation using the Gibbs sampler," in *Proc. IEEE Vehicular Technology Symp. (VTS)-Fall 2000*, vol. 1, Sep. 24–28, 2000, pp. 429–433.

[18] Y. Huang and P. M. Djurić, "Multiuser detection of synchronous CDMA signals by the Gibbs coupler," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process. (ICASSP)*, vol. 4, May 7–11, 2001, pp. 2273–2276.

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

[20] P. M. Y. Huang and T. Ghirmai, "Perfect sampling: A review and applications to signal processing," *IEEE Trans. Signal Process.*, vol. 50, no. 2, pp. 345–356, Feb. 2002.

[21] X. Wang and R. Chen, "Adaptive Bayesian multiuser detection for synchronous CDMA with Gaussian and impulsive noise," *IEEE Trans. Signal Process.*, vol. 48, no. 7, pp. 2013–2028, Jul. 2000.

[22] ——, "Adaptive Bayesian multiuser detection," in *Proc. IEEE Int. Symp. Inf. Theory 2000*, Jun. 25–30, 2000, pp. 273–273.

[23] X. Wang and V. D. Phan, "Turbo multiuser detection for nonlinearly modulated CDMA in unknown channels," in *IEEE 6th Int. Symp. Spread Spectrum Technology Application*, vol. 2, Sep. 6–8, 2000, pp. 598–602.

[24] L. Wu, G. Liao, and H. Huang, "A novel nonlinear group-blind multiuser detection technique," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, vol. 4, Apr. 6–10, 2003, pp. 49–52.

[25] Z. Yang, B. Lu, and X. Wang, "Bayesian Monte Carlo multiuser receiver for space-time coded multicarrier CDMA systems," in *IEEE Vehicular Technology Symp. (VTS)-Fall 2000*, vol. 1, Sep. 24–28, 2000, pp. 429–433.

[26] Z. Yang and X. Wang, "Blind multiuser detection for long-code multipath CDMA," in *Conf. Rec. 34th Asilomar Conf. Signals, Systems, Computing*, vol. 2, Oct.–Nov. 29–1, 2000, pp. 1148–1152.

[27] ——, "Blind turbo multiuser detection for long-code uplink CDMA with unknown interferences," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing (ICASSP)*, vol. 4, May 7–11, 2001, pp. 2305–2308.

[28] Z. Yang, B. Lu, and X. Wang, "Bayesian Monte Carlo multiuser receiver for space-time coded multicarrier CDMA systems," *IEEE J. Sel. Areas Commun.*, vol. 19, no. 8, pp. 1625–1637, Aug. 2001.

[29] ——, "Blind Bayesian multiuser receiver for space-time coded MC-CDMA system over frequency-selective fading channel," in *Global Telecommunications Conf. (GLOBECOM'01)*, vol. 2, Nov. 25–29, 2001, pp. 781–785.

[30] Z. Yang and X. Wang, "Blind turbo multiuser detection for long-code multipath CDMA," *IEEE Trans. Commun.*, vol. 50, no. 1, pp. 112–125, Jan. 2002.

[31] D. Guo and X. Wang, "Blind detection in MIMO systems via sequential Monte Carlo," *IEEE J. Sel. Areas Commun.*, vol. 21, no. 3, pp. 464–473, Apr. 2003.

[32] B. Dong, X. Wang, and A. Doucet, "A new class of MIMO demodulation algorithms," *IEEE Trans. Signal Process.*, vol. 51, no. 11, pp. 2752–2763, Nov. 2003.

[33] J. Hammersley and D. Handscomb, *Monte Carlo Methods*. New York: Methuen, 1964.

[34] G. Fishman, *Monte Carlo: Concepts, Algorithms and Applications*. New York: Springer-Verlag, 1996.

[35] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*. New York: Springer-Verlag, 1999.

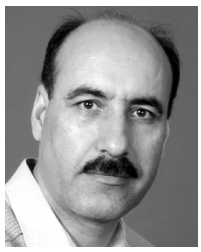
[36] O. Häggström, *Finite Markov Chains and Algorithmic Applications*. Cambridge, U.K.: Cambridge Univ. Press, 2002.

[37] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, 3rd ed. New York: McGraw-Hill, 1991.

[38] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Machine Intell.*, no. 6, pp. 721–741, 1984.

[39] A. E. Gelfand and A. F. M. Smith, "Sampling-based approaches to calculating marginal densities," *J. Amer. Stat. Assoc.*, vol. 85, pp. 398–409, Jun. 1990.

[40] S. ten Brink, "Convergence of iterative decoding," *Electron. Lett.*, vol. 35, no. 10, pp. 806–808, May 1999.
[41] ——, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Trans. Commun.*, vol. 49, no. 10, pp. 1727–1737, Oct. 2001.
[42] ——, "Code characteristic matching for iterative decoding of serially concatenated codes," *Ann. Des Télécommun.*, vol. 56, no. 7-8, pp. 394–408, 2001.
[43] ——, "Rate one-half code for approaching the Shannon limit by 0.1 dB," *Electron. Lett.*, vol. 36, pp. 1293–1294, Jul. 2000.
[44] S. ten Brink, G. Kramer, and A. Ashikhmin, "Design of low-density parity-check codes for modulation and detection," *IEEE Trans. Commun.*, vol. 52, no. 4, pp. 670–678, Apr. 2004.
[45] C. Schlegel and A. Grant, "Differential space-time turbo codes," *IEEE Trans. Inf. Theory*, vol. 49, no. 9, pp. 2298–2306, Sep. 2003.
[46] J. Hagenauer and R. Bauer, "The turbo principle in joint source channel decoding of variable length codes," in *Proc. IEEE Information Theory Workshop (ITW2001)*, Cairns, Australia, Sep. 2–7, 2001, pp. 33–35.
[47] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.
[48] H. E. Gamal and A. R. Hammons Jr, "A new approach to layered space-time coding and signal processing," *IEEE Trans Inf. Theory*, vol. 47, no. 6, pp. 2321–2334, Sep. 2001.
[49] P. Robertson, E. Villebrun, and P. Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain," in *IEEE Int. Conf. Commun. (ICC)*, vol. 2, Seattle, WA, Jun. 18–22, 1995, pp. 1009–1013.
[50] U. Fincke and M. Pohst, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis," *Math. Comp.*, vol. 44, pp. 463–471, 1985.
[51] B. M. Hochwald and S. ten Brink, "Achieving near-capacity on a multiple-antenna channel," *IEEE Trans. Commun.*, vol. 51, no. 3, pp. 389–399, Mar. 2003.
[52] D. Pham, K. R. Pattipati, P. K. Willett, and J. Luo, "An improved complex sphere decoder for V-BLAST systems," *IEEE Signal Process. Lett.*, vol. 11, no. 9, pp. 748–751, Sep. 2004.
[53] R. Wang and G. B. Giannakis, "Approaching MIMO channel capacity with reduced-complexity soft sphere decoding," in *Proc. IEEE Wireless Commun. Networking Conf. (WCNC 2004)*, vol. 3, Mar. 21–25, 2004, pp. 1620–1625.
[54] A. M. Chan and I. Lee, "A new reduced-complexity sphere decoder for multiple antenna systems," in *IEEE Int. Conf. Commun. (ICC)*, vol. 1, Apr.–May 28–2, 2002, pp. 460–464.
[55] H. Vikalo and B. Hassibi, "Modified fincke-pohst algorithm for low-complexity iterative decoding over multiple antenna channels," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT 2002)*, pp. 390–390.
[56] R. Van Nee and R. Prasad, *OFDM for Wireless Multimedia Communications*. Boston, MA: Artech House, 2000.
[57] R.-R. Chen, B. Farhang-Boroujeny, and A. Ashikhmin, "Capacity-approaching LDPC codes based on Markov chain Monte Carlo MIMO detection," in *SPAWC 2005: 6th IEEE Int. Workshop Signal Process. Advances Wireless Communication*, New York, Jun. 5–8, 2005.
[58] H. Zhu, "MIMO data detection methods and channel estimation in flat fading environments," Ph.D. dissertation, Electrical and Computer Eng. Dept., Univ. of Utah, Salt Lake City, UT, May 2006.
[59] H. Zhu, B. Farhang-Boroujeny, and R.-R. Chen, "On performance of sphere decoding and Markov chain Monte Carlo detection methods," presented at the SPAWC 2005: 6th IEEE Int. Workshop Signal Process. Advances Wireless Commun., New York, Jun. 5–8, 2005.
[60] ——, "On performance of sphere decoding and Markov chain Monte Carlo detection methods," *IEEE Signal Process. Lett.*, vol. 12, no. 10, pp. 669–672, Oct. 2005.



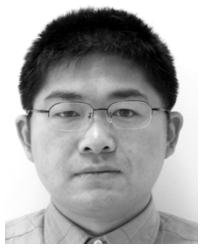
**Behrouz Farhang-Boroujeny** (M'84–SM'90) received the B.Sc. degree in electrical engineering from Tehran University, Iran, in 1976, the M.Eng. degree from the University of Wales Institute of Science and Technology, U.K., in 1977, and the Ph.D. degree from Imperial College, University of London, U.K., in 1981.

From 1981 to 1989, he was with Isfahan University of Technology, Isfahan, Iran. From 1989 to 2000, he was with National University of Singapore. Since August 2000, he has been with the University of Utah, Salt Lake City, where he is now a Professor and Associate Chair of the department. He is an expert in the general area of signal processing. His current scientific interests are adaptive filters, multicarrier communications, detection techniques for space-time coded systems, and signal processing applications to optical devices. In the past, he has worked and has made significant contribution to areas of adaptive filters theory, acoustic echo cancellation, magnetic/optical recoding, and digital subscriber line technologies. He is the author of *Adaptive Filters: Theory and Applications* (New York: Wiley, 1998).

Dr. Farhang-Boroujeny received the UNESCO Regional Office of Science and Technology for South and Central Asia Young Scientists Award in 1987. He was Associate Editor of IEEE TRANSACTIONS ON SIGNAL PROCESSING from July 2002 to July 2005. He has also been involved in various IEEE activities. He is currently Chairman of the Signal Processing/Communications chapter of IEEE in Utah.

**Haidong (David) Zhu** received the B.S. and M.S. degrees in electrical engineering from Fudan University, Shanghai, China, in 1998 and 2001, respectively. He is currently pursuing the doctoral degree at the University of Utah, Salt Lake City.

Since 2001, he has been a Research Assistant with the University of Utah. In summer 2005, he was a Research Intern with Nokia Research Center, San Diego, CA. His research interests include statistical signal processing for MIMO/CDMA systems, iterative decoding, and channel estimation for wireless communication systems.



**Zhenning Shi** (S'01–M'04) received the B.S. degree in electronics engineering from Tsinghua University, Beijing, China, in 1998 and the Ph.D. degree in electrical and computer engineering from the University of Utah, Salt Lake City, in 2003.

From 2002 to 2004, he was a Postdoctoral Fellow with the Electrical and Computer Engineering Department, University of Utah. In 2004, he joined the Wireless Signal Processing (WSP) program, National ICT, Australia. He is also an Adjunct Research Fellow with the Research School of Information Science and Engineering, Australian National University. His primary research interests include multiple-access systems, iterative detection, Markov chain Monte Carlo methods, channel estimation, and carrier recovery.