

Soft output multiuser detection via a Markov chain Monte Carlo approach

Soren Henriksen, Brett Ninness, and Steven R. Weller

Abstract— This paper investigates the use of computational Bayesian methods for multiuser detection (MUD) of synchronous direct-sequence code-division multiple access (DS-CDMA) systems. The Markov chain Monte Carlo (MCMC) multiuser detection methods proposed in this paper are iterative, and offer near-optimal performance with manageable complexity even at high system loads. Moreover, posterior symbol probabilities produced by the detector are suitable for use as prior probabilities for soft-input channel decoders.

I. INTRODUCTION

It is well known that while the maximum-likelihood (ML) detector for code-division multiple access (CDMA) systems is optimal in the sense of minimizing the probability of incorrectly detecting the user symbols jointly, the optimal detector typically exhibits a computational complexity which increases exponentially with the number of users [1]. Due to the prohibitive complexity of the ML detector, enormous effort has been expended over the past two decades to find suboptimal detectors which strike a balance between performance and computational complexity [2].

This paper investigates the use of computational Bayesian methods for multiuser detection (MUD) of symbol synchronous direct-sequence code-division multiple-access (DS-CDMA) systems. In particular, this paper explores the use of *Markov chain Monte Carlo (MCMC)* methods for multiuser detection. Sampling-based methods, especially MCMC algorithms, have generated enormous recent interest in the mathematical and applied statistics communities [3], [4] and have a history spanning decades in other fields [5], [6].

The essential idea underlying MCMC methods is the construction of a random number generator which produces samples from the posterior density of interest which, in this paper, are the probabilities of the symbols transmitted by each user. Sample histograms of the generator output are used to estimate the density values, with the strong law of large numbers guaranteeing estimates of arbitrary accuracy given sufficient realizations from the generator. The key principle is to construct the generator via implementation of a Markov chain whose limiting density is the desired posterior.

The methods employed here are based on the Metropolis–Hastings method. A particular special case of this method is the so-called Gibbs sampler, which has recently been proposed for joint channel parameter estimation and multiuser detection

[7], [8]. In related work, a class of algorithms known as sequential importance sampling methods, of which an increasingly well known instance is the particle filter, has been applied to multiuser detection; see [9] and the references therein. These algorithms also approximate the optimal detector sample histograms, but they are substantially different to the methods proposed in this paper.

This paper builds on the Gibbs sampler approach to MUD [7], [8] in two ways. First, we examine the use of the more general Metropolis–Hastings framework. Second, we address a problem identified in [8], [9] in which the performance of Gibbs sampling methods is shown to degrade at high signal-to-noise ratios.

II. SIGNAL MODEL

This paper considers the following baseband CDMA signal model, in which the received signal consists of the sum of antipodally modulated synchronous signature waveforms embedded in additive white Gaussian noise (AWGN):

$$y(t) = \sum_{n=1}^K A_n x_n(t) s_n(t) + \sigma \nu(t), \quad 0 \leq t \leq T. \quad (1)$$

Here K is the number of active users, A_n is the received amplitude of the n -th user, $x_n(t) \in \{-1, +1\}$ is the symbol transmitted by the n -th user, T is the symbol period, and $s_n(t)$ is the normalized signature waveform assigned to the n -th user. The additive noise $\nu(t)$ in (1) is white Gaussian, with unit power spectral density and the noise source single-sided spectral density $N_0 = 2\sigma^2$. The normalized signature waveform of the n -th user is given by

$$s_n(t) = \sum_{j=0}^{N-1} a_n(t) p(t - jT_c), \quad (2)$$

where T_c is the chip interval, $p(t)$ is a rectangular pulse of duration T_c and amplitude such that $\int_0^{T_c} s_n^2(t) dt = 1$. The signature waveform for the n -th user is defined by the sequence $\{a_n(t), n = 0, 1, \dots, N-1\}$ of pseudorandom chips $\in \{-1, +1\}^N$, where the spreading factor (or processing gain) N is the number of pseudorandom chips in one symbol period, so that $T = NT_c$.

At the front end of the receiver, a bank of filters matched to the signature waveform of the users is employed. After sampling at the end of each symbol period, this results in the output y_n of the n -th matched filter being

$$y_n = \int_0^T y(t) s_n(t) dt = A_n x_n + \sum_{j \neq n} A_j x_j \rho_{jn} + \nu_n, \quad (3)$$

The authors are with the School of Electrical Engineering and Computer Science, The University of Newcastle, Callaghan, NSW 2308, Australia, email: {eesjh, brett, steve}@ee.newcastle.edu.au

where

$$\rho_{ij} = \int_0^T s_i(t)s_j(t)dt, \quad \nu_n = \sigma \int_0^T n(t)s_n(t)dt. \quad (4)$$

It is convenient to express (3)–(4) in vector form:

$$y = RAx + \nu. \quad (5)$$

Here, the normalized crosscorrelation matrix $R = \{\rho_{ij}\}$ has diagonal elements equal to 1 and is symmetric nonnegative definite with $y \triangleq [y_1, \dots, y_K]^T$, $x \triangleq [x_1, \dots, x_K]^T$, and $A \triangleq \text{diag}\{A_1, \dots, A_K\}$, and with ν a zero-mean Gaussian random vector with covariance matrix $\mathbf{E}\{\nu\nu^T\} = \sigma^2 R$.

III. APPROXIMATELY OPTIMAL DETECTION

Given the observation y , we seek to estimate the vector of transmitted symbols x of each of the K users. The approach considered here is that of maximum a-posteriori (MAP) detection, so that the estimate \hat{x} is taken as

$$\hat{x} = \arg \max_{x \in \{-1, +1\}^K} p(x | y) \quad (6)$$

with

$$p(x | y) = \frac{p(x)}{p(y)} p_\nu(y - RAx) \quad (7)$$

where $p_\nu(\cdot)$ is the joint probability density function of the elements in the noise vector ν . When a Gaussian distribution is involved, this leads to the well known explicit form

$$p(x | y) = \frac{\text{const}}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2} (y - RAx)^T R^{-1} (y - RAx)\right). \quad (8)$$

The MAP estimate \hat{x} is therefore, in the Gaussian case, given by

$$\hat{x} = \arg \min_{x \in \{-1, +1\}^K} (y - RAx)^T R^{-1} (y - RAx). \quad (9)$$

The minimization of (9) is a combinatorial optimization problem requiring, in general, an exhaustive search over 2^K possible combinations of transmitted symbols, motivating the search for computationally cheaper solutions.

The contribution of this paper is to address the issue via a Markov chain Monte Carlo (MCMC) approach. This involves generating a realization $\{x_1, \dots, x_k, \dots, x_M\}$ with limiting distribution equal to the desired posterior; viz.

$$\lim_{k \rightarrow \infty} p(x_k = x | x_0) = p(x | y).$$

This simulated realization $\{x_k\}$ is then used as if it were a random sample from $p(x | y)$. Provided the required distributional convergence holds, then via a law of large numbers argument, this leads to consistent estimates of various quantities. For example, for an arbitrary measurable function g , it allows the numerical computation and consistent estimation of the conditional expectation $\mathbf{E}\{g(x) | y\}$ as

$$\mathbf{E}\{g(x) | y\} = \sum_x g(x)p(x | y) \approx \frac{1}{M} \sum_{k=\ell+1}^{\ell+M} g(x_k). \quad (10)$$

Here ℓ is the length of an initial ‘‘burn-in’’ period of the algorithm, so that right hand side of (10) is simply the sample histogram of $g(x)$.

While this may seem like a reasonable approach, it may also appear to be impossible to implement due to the difficulty of sampling from the rather arbitrary and perhaps poorly specified multivariable posterior density $p(x | y)$. Perhaps surprisingly, a Markov chain can be constructed with limiting density equal to the required posterior $p(x | y)$, and in a manner that is far more straightforward than might prima facie be thought possible. In this paper, it will be achieved by means of the following *Metropolis–Hastings algorithm* [5], [6].

Algorithm III.1—Metropolis–Hastings:

- 1) Initialize x_0 at some value such that $p(x_0 | y) > 0$
- 2) At iteration k , consider a candidate value ξ_k which is drawn from a proposal density $\gamma(\xi_k | x_{k-1})$. That is, find a possible realization for x_k as

$$\xi_k \sim \gamma(\cdot | x_{k-1}) \quad (11)$$

- 3) Compute the acceptance probability

$$\alpha(\xi_k | x_{k-1}) = \min \left\{ 1, \frac{p(\xi_k | y)}{p(x_{k-1} | y)} \cdot \frac{\gamma(x_{k-1} | \xi_k)}{\gamma(\xi_k | x_{k-1})} \right\} \quad (12)$$

- 4) Accept the proposed $x_k = \xi_k$ with probability $\alpha(\xi_k | x_{k-1})$, otherwise leave x_k unchanged
- 5) Increment k and return to step 2

While this provides a completely general definition of the essential algorithm to be employed here, some important specializations and embellishments will be employed here in order to maximize its performance for the MUD application studied in this paper.

In particular, note that in the special case of a proposal density which is symmetric in that $\gamma(\xi | x) = \gamma(x | \xi)$ then

$$\frac{\gamma(x_{k-1} | \xi_k)}{\gamma(\xi_k | x_{k-1})} = 1, \quad (13)$$

and hence the acceptance probability (12) simplifies to

$$\alpha(\xi_k | x_{k-1}) = \min \left\{ 1, \frac{p(\xi_k | y)}{p(x_{k-1} | y)} \right\}. \quad (14)$$

In this case, Algorithm III.1 is known as the ‘‘Metropolis Algorithm’’.

A further specialization of Algorithm III.1 occurs in the situation where it is possible to employ the proposal density

$$\gamma(x_k^i | x_{k-1}) = p(x_k^i | x_k^{-i}, y). \quad (15)$$

Here x^i denotes an i -th sub-block of the vector x , and x^{-i} denotes the complement of this, namely everything *except* the i -th sub-block. Note that the term in the acceptance probability (14) affected by this choice becomes

$$\frac{p(x_k | y)}{p(x_{k-1} | y)} \cdot \frac{\gamma(x_{k-1}^i | x_k)}{\gamma(\xi_k^i | x_{k-1})} = \frac{p(x_k^i | x_k^{-i}, y)p(x_k^{-i} | y)}{p(x_{k-1}^i | x_{k-1}^{-i}, y)p(x_{k-1}^{-i} | y)} \cdot \frac{p(x_{k-1}^i | x_{k-1}^{-i}, y)}{p(x_k^i | x_k^{-i}, y)} = 1.$$

In this calculation, the definition of conditional probability $p(A, B) = p(A|B)p(B)$ and the fact that by design $x_k^{-i} = x_{k-1}^{-i}$ have both been used. This implies that the acceptance probability $\alpha(x_k | x_{k-1})$ in (14) is one, and hence the proposals drawn from the density $p(x_k^i | x_k^{-i}, y)$ are always retained. In this special case, Algorithm III.1 becomes an instance of the *Gibbs sampling* algorithm, whose application to multiuser detection method is presented in [8], [10].

IV. UNDERLYING THEORY

This section is intended as a brief presentation of the fundamental ideas underlying Algorithm III.1 and its properties. The material is derived from several excellent detailed treatments that are available [4], [11], [3], [12].

We assume the reader is familiar with the idea of a Markov chain, which is a stochastic process $\{x_k\}$ for which the probability density function of a realization x_k at time $t = k$, and given all previous observations $\{\dots, x_{k-2}, x_{k-1}\}$ reduces to a function $K(x_k | x_{k-1})$ of x_k and x_{k-1} only, and which is termed the transition probability density function.

A crucial point is that Algorithm III.1 implements a Markov chain in which the realizations $\{x_k\}$ have an underlying transition probability which is very simply characterized. Namely, the probability of observing x_k given x_{k-1} is simply the probability $\gamma(x_k | x_{k-1})$ of proposing x_k times the probability $\alpha(x_k | x_{k-1})$ of accepting this proposal:

$$K(x_k | x_{k-1}) = \alpha(x_k | x_{k-1}) \cdot \gamma(x_k | x_{k-1}) + \delta(x_k - x_{k-1})r(x_{k-1}), \quad (16)$$

where

$$r(x_{k-1}) = \left[1 - \sum_{\xi \in \{-1, +1\}^N} \alpha(\xi | x_{k-1}) \cdot \gamma(\xi | x_{k-1}) \right]. \quad (17)$$

The term $\delta(x_k - x_{k-1})r(x_{k-1})$ in (16) accounts for the (usually) rare event that the proposal density may provide the realization $x_k = x_{k-1}$.

While it is fundamental that Algorithm III.1 implements a Markov chain, it is also essential that this chain converges in the distributional sense that for some density $\pi(x)$

$$\pi(x) = \lim_{k \rightarrow \infty} p(x_k = x | x_0). \quad (18)$$

To investigate this, note that by straightforward computation, assuming that such a $\pi(x)$ exists

$$\begin{aligned} \pi(x) &= \lim_{k \rightarrow \infty} p(x_k = x | x_0) \\ &= \lim_{k \rightarrow \infty} \sum_{x_{k-1}} K(x_k = x | x_{k-1}) \cdot p(x_{k-1} | x_0) \\ &= \sum_{x_{k-1}} \lim_{k \rightarrow \infty} K(x_k = x | x_{k-1}) \cdot \lim_{k \rightarrow \infty} p(x_{k-1} | x_0) \\ &= \sum_{\lambda} K(x | \lambda) \cdot \pi(\lambda). \end{aligned}$$

Therefore, if the Markov chain realized by Algorithm III.1 converges, then it does so to an *invariant* density π which (if it exists) must satisfy the defining equation

$$\pi(x) = \sum_{\lambda} K(x | \lambda)\pi(\lambda). \quad (19)$$

Suppose further that for some density π , the Markov transition probability satisfies the reversibility condition, also called the *detailed balance* condition:

$$\pi(\lambda)K(x | \lambda) = \pi(x)K(\lambda | x). \quad (20)$$

That is, the probability of randomly drawing the value λ from $\pi(\cdot)$, and then the Markov chain moving from λ to x is identical to the reverse set of moves (randomly drawing x and moving from there to λ).

If the reversibility condition (20) holds,

$$\sum_{\lambda} K(x | \lambda)\pi(\lambda) = \sum_{\lambda} K(\lambda | x)\pi(x) \quad (21)$$

$$= \pi(x) \sum_{\lambda} K(\lambda | x) = \pi(x). \quad (22)$$

Hence the reversibility condition (20) on π is a sufficient condition for π to be an invariant density with respect the kernel K , in which case it is a candidate for the ergodic limit (18).

Now here is a vital point. The acceptance probability $\alpha(x | \lambda)$ given by (14) is designed to yield a kernel $K(x | \lambda)$ for which (20) and hence (22) holds for $\pi(x) = p(x | y)$.

Lemma IV.1: Algorithm III.1 yields a Markov chain with invariant density $\pi(x) = p(x | y)$.

Proof: Suppose that $x_k \neq x_{k-1}$. Then according to (16) and substituting into (12)

$$\begin{aligned} &p(x_{k-1} | y)K(x_k | x_{k-1}) \\ &= \alpha(x_k | x_{k-1})\gamma(x_k | x_{k-1})p(x_{k-1} | y) \\ &= \min \{ \gamma(x_k | x_{k-1})p(x_{k-1} | y), \gamma(x_{k-1} | x_k)p(x_k | y) \}. \end{aligned}$$

On the other hand, by the same reasoning

$$\begin{aligned} &p(x_k | y)K(x_{k-1} | x_k) \\ &= \alpha(x_{k-1} | x_k)\gamma(x_{k-1} | x_k)p(x_k | y) \\ &= \min \{ \gamma(x_{k-1} | x_k)p(x_k | y), \gamma(x_k | x_{k-1})p(x_{k-1} | y) \}. \end{aligned}$$

That is,

$$p(x_{k-1} | y)K(x_k | x_{k-1}) = p(x_k | y)K(x_{k-1} | x_k)$$

and hence according to the reasoning leading to (22), $p(x | y)$ is an invariant density for the Markov Chain implemented by Algorithm III.1 The case of $x_k = x_{k-1}$ follows by an identical argument. ■

This establishes that Algorithm III.1 implements a Markov Chain that, if it converges, must converge (in a distributional sense) to the desired posterior density $p(x | y)$. In fact, a stronger result is possible.

Theorem IV.1—Law of Large Numbers: Algorithm III.1 generates a realization sequence $\{x_k\}$ for which, with $f : \mathbf{R}^K \rightarrow \mathbf{R}$ being an arbitrary bounded function

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{k=1}^N f(x_k) = \sum_x f(x) \pi(x) \quad (23)$$

with probability one.

Proof: See [13]. ■

In particular, choosing f as the Kronecker delta function $f(x) = \delta(x - x_k)$ implies that the sample histogram formed from realizations of Algorithm III.1 converges to the invariant density $p(x | y)$, and thereby provides a means for estimating it.

V. MULTIUSER DETECTION VIA MCMC

With the essential theoretical underpinnings of the general Algorithm III.1 established, the paper now turns to the application of these techniques to the specific problem of multiuser detection. This involves the study of two specializations of Algorithm III.1, namely the Gibbs sampler and the Metropolis method.

A. Multiuser Detection via the Gibbs Sampler

Recall that the Gibbs sampler is achieved as a special case of Algorithm III.1 whereby the proposal density is chosen as

$$\gamma(x_k^i | x_{k-1}) = p(x_k^i | x_{k-1}^{-i}, y) \quad (24)$$

and the acceptance probability is then always equal to one. This requires that the k -th realization of a symbol vector x_k be divided into a variety of sub-blocks. Each sub-block may comprise a different number of bits, and sub-blocks may overlap. However, the choice and size of the sub-blocks involves a trade-off between the computation complexity required at each iteration, and the total number of iterations required to form the required sample histograms to sufficient accuracy.

Lemma V.1: In the case where x_k^i represents the (antipodally modulated) symbol of a single user in the baseband CDMA system (1), the proposal density is readily calculated as:

$$p(x_k^i = +1 | x_{k-1}^{-i}, y) = \frac{1}{1 + \beta^{-1}}, \quad (25)$$

where

$$\beta = \frac{p(y | x_{k-1}^{-i}, x_k^i = +1)}{p(y | x_{k-1}^{-i}, x_k^i = -1)}. \quad (26)$$

With $[B]^i$ and $[B]^{-i}$ denoting, respectively, the i -th sub-block of matrix B , and everything except the i -th sub-block of B ,

$$p(y | x_{k-1}^{-i}, x_k^i) = p_\nu (y - [RA]^i x_k^i - [RA]^{-i} x_{k-1}^{-i}) \quad (27)$$

$$= \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2} \|y - [RA]^i x_k^i - [RA]^{-i} x_{k-1}^{-i}\|_R^2\right), \quad (28)$$

where for $z \in \mathbf{R}^K$, $\|z\|_R^2 \triangleq z^T R^{-1} z$.

Proof: See [13]. ■

Algorithm V.1—Gibbs Sampler for MUD:

- 1) Initialize x_0 at some value such that $p(x_0 | y) > 0$
- 2) At iteration k , for a randomly chosen index i , compute the proposal density value

$$\gamma(x_k^i = +1 | x_{k-1}) = p(x_k^i = +1 | x_{k-1}^{-i}, y) \quad (29)$$

according to (25)–(28)

- 3) Draw a realization $z \sim \mathcal{U}[0, 1]$, where $\mathcal{U}[0, 1]$ represents a uniform distribution on the interval $[0, 1]$
- 4) Set $x_k^i = +1, x_k^{-i} = x_{k-1}^{-i}$ if

$$\gamma(x_k^i = +1 | x_{k-1}) < z, \quad (30)$$

otherwise set $x_k^i = -1, x_k^{-i} = x_{k-1}^{-i}$

- 5) Increment k and return to step 2

B. Multiuser Detection via the Metropolis Algorithm

The Metropolis algorithm is another specialization of Algorithm III.1. However, it is a situation in which the proposal density γ may be chosen much more generally than is dictated in the Gibbs sampling case.

For the purposes of multiuser detection the simplest, and most practical (from a computational load point of view) choice of proposal density again involves the consideration of sub-blocks of a realization x_k . The proposal density γ is then taken as one in which a tentative new realization ξ_k is formed by the bits in the randomly chosen i -th sub-block having their signs reversed with probability one.

Clearly, in this case $\gamma(\xi_k | x_{k-1}) = \gamma(x_{k-1} | \xi_k) = 1$ which, according to (13) is necessary for the associated general Metropolis–Hastings Algorithm III.1 to reduce to the simpler Metropolis case. It is then necessary to compute the acceptance probability $\alpha(\xi_k | x_{k-1})$ according to (14).

The most suitable approach computationally is to use 1-bit sub-blocks, where there is probability $p = 1$ of inverting the selected bit. Thus the proposal step requires negligible computation. In the case of larger sub-blocks, bits within the sub-block may randomly be replaced or inverted.

The Metropolis–Hastings algorithm is a more general MCMC technique than the Gibbs Sampler, and offers different Markov chain transition probabilities, yet retains the same limiting distribution. As it allows a choice of proposal density, it can be adjusted to best suit the MUD application.

Algorithm V.2—Metropolis–Hastings for MUD:

- 1) Initialize x_0 at some value such that $p(x_0 | y) > 0$
- 2) At iteration k , for a randomly chosen index i , set

$$\xi_k = -x_{k-1}^i \cup x_{k-1}^{-i} \quad (31)$$

- 3) Compute the acceptance probability

$$\alpha(x_k | x_{k-1}) = \min\{1, \beta\} \quad (32)$$

where β is given by (26)–(28)

- 4) Accept the proposed $x_k = \xi_k$ with probability $\alpha(x_k | x_{k-1})$
- 5) Increment k and return to step 2

Figure 1 compares the detection performance (bit error rate (BER) versus E_b/N_0) of the Gibbs sampler and Metropolis–Hastings detectors (Algorithms V.1 and V.2, respectively) with the linear minimum mean-squared error (MMSE) detector [2], single-user bound, an iterative parallel decision-feedback detector (P-DFD), and a 2-stage iterative successive DFD (IS-DFD), the latter two detectors being presented in [14]. Spreading sequences are randomly assigned, and at least 5000 error events are observed for each E_b/N_0 of interest. For the computational Bayesian detectors, 5000 iterations of the corresponding algorithm were employed for the detection of each symbol.

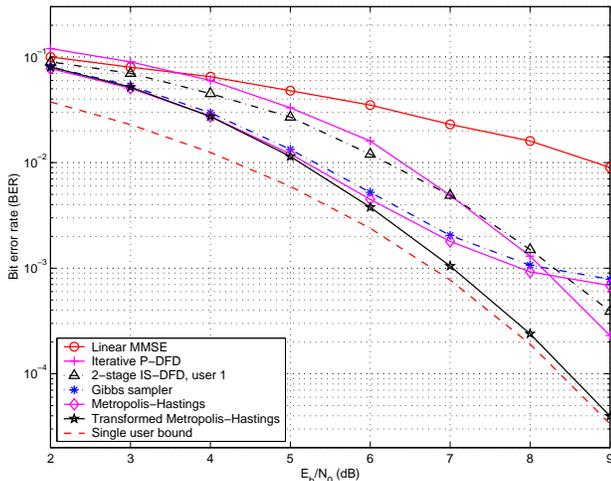


Fig. 1. Performance comparison of different receivers on an AWGN channel, with $K = 96$ users and spreading factor $N = 128$

For bit-error rates exceeding 10^{-3} , the Gibbs sampler and Metropolis–Hastings detectors clearly outperform the MMSE, P-DFD and IS-DFD receivers, with the Metropolis–Hastings detector providing a modest improvement on the Gibbs sampler of around 0.1 dB. For E_b/N_0 exceeding around 6 dB, however, a clear flattening of the BER curves is apparent for both of the computational Bayesian detectors, a phenomenon also observed in previous applications of the Gibbs sampler to MUD [7], suggesting that the algorithms in their current form are primarily useful in low signal-to-noise ratio (SNR) environments.

At high SNR, the shape of $p(x|y)$ is poorly suited to sampling-based algorithms, resulting in excessively long convergence times and the flattening of the BER performance curves. In the following we describe a method to overcome this problem.

A. Search Space Transformation

At higher signal-to-noise ratios, the conditional probability in (8) is influenced by the lower values of estimated noise variance σ^2 . The effect on the Metropolis algorithm is that the acceptance probability is much more sensitive to changes in the x vector, and hence the transition probabilities are much lower at local minima.

To overcome this problem, the algorithm can be made more efficient by operating on a transformation of $p(x|y)$. A new

probability density function $p_\eta(x|y) = f(p(x|y))$ will be used instead for the Metropolis algorithm. We define this transformed function via a mapping of the marginal probability

$$p_\eta(x^i | x^{-i}, y) = K_i p(x^i | x^{-i}, y)^\eta, \quad (33)$$

where $\eta > 0$ is a design constant, typically chosen in the range $[0.2, 2]$, and K_i is a normalizing constant such that

$$\sum_{x^i \in \{-1, 1\}} K_i p_\eta(x^i | x^{-i}, y) = 1 \quad (34)$$

for all i . It is not necessary to calculate these constants for the Metropolis algorithm, although they are estimated for the recovery of the soft-output probabilities after the algorithm is complete.

Selecting $\eta < 1$ causes a higher proportion of the iterations to occur in areas that would otherwise be poorly explored. This transformation requires essentially no additional computational load. Figure 1 shows the performance of the transformed Metropolis–Hastings detector corresponding to $\eta = 0.4$, in which the effectiveness of the transformed detector at eliminating the flattening of the BER curve at high SNR is apparent.

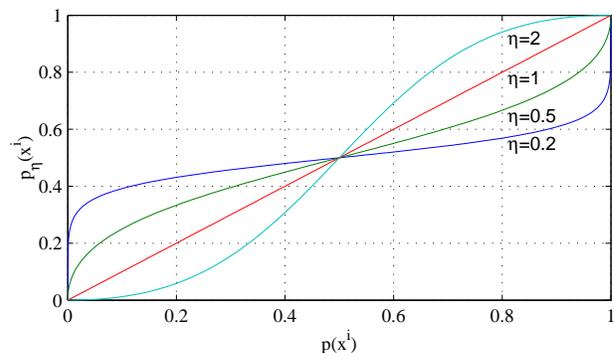


Fig. 2. Transformation of the bit marginal probability

For binary phase-shift keying (BPSK), Figure 2 shows the transformation that is applied to the marginal probability distribution for various values of η . Values of $\eta < 1$ force the algorithm to spend a higher proportion of the iterations exploring areas of the space with bit probabilities of less than 0.5.

As the transformation function is monotonic for all values of η , the MAP estimate for each bit will be the same for both marginal probabilities, i.e.

$$\arg \max_x p_\eta(x^i | y) = \arg \max_x p(x^i | y). \quad (35)$$

Thus for hard-output decisions, the transformed conditional probability may be used in place of the original, and the the same maximum likelihood result will be found. In addition, correct soft output probabilities may still be recovered by applying an inverse transformation to the output.

Apart from this final re-mapping, this transformation does not increase the complexity of the algorithm. The only change is in the evaluation of the acceptance probability and, computationally, this is equivalent to modifying the value of σ^2 that is provided to the algorithm, so no additional operations are required per iteration.

Figure 3 shows the BER performance of the Metropolis algorithm simulated for a range of SNRs and values of η . For a given spreading gain and number of users, the optimal choice of η is such that the ratio $\frac{\sigma^2}{\eta}$ is may be kept a constant.

The consequence of this is that the algorithm itself does not require an estimate of the noise variance for optimum calculation of the hard estimates. Knowledge of the signal-to-noise ratio is only required in order to obtain accurate soft outputs.

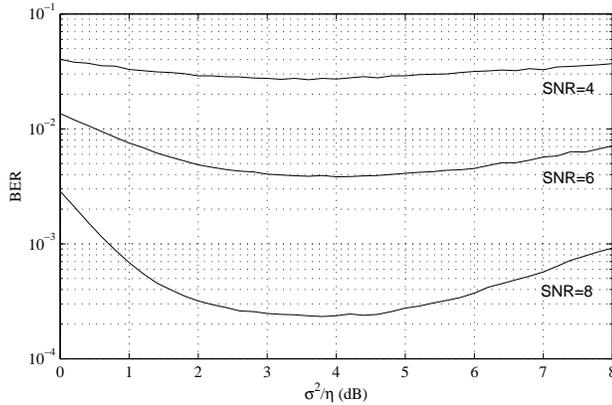


Fig. 3. BER as a function of η and SNR after 5000 iterations; spreading factor $N = 128$, and $K = 96$ users

B. Soft Outputs

A key benefit of the MCMC schemes is that the outputs take the form of posteriori probabilities rather than hard decisions. These probabilities may be fed to soft-input channel decoders including turbo and low-density parity-check decoders.

When the density function is transformed, (34) is used to calculate the necessary constants K_i for re-scaling from the estimated marginal distribution, \hat{p}_η :

$$\hat{p}(x^i | x^{-i}, y) = \frac{1}{K_i} \hat{p}_\eta(x^i | y)^{\frac{1}{\eta}} = \frac{\hat{p}_\eta(x^i | y)^{\frac{1}{\eta}}}{\sum_{x^i} \hat{p}_\eta(x^i | y)^{\frac{1}{\eta}}}. \quad (36)$$

Figure 4 shows the performance of the soft outputs from the decoder in the form of a histogram showing the ensemble averages of a transmitted bit being a logic one, as a function of the probability ascribed to it by the algorithm. This histogram is entirely consistent with the expected form, as out of N bits x_0, \dots, x_{N-1} , each of which has an output probability of $p_1 = p\{x_k = 1\}$, the expected number of bits that are 1 is $p_1 N$. The straight-line relationship seen in Figure 4 verifies that the output probabilities are indeed an accurate measure of the reliability of the estimates.

REFERENCES

- [1] S. Verdú, "Computational complexity of optimum multiuser detection," *Algorithmica*, vol. 4, no. 3, pp. 303–312, 1989.
- [2] S. Verdú, *Multiuser Detection*, Cambridge University Press, Cambridge, United Kingdom, 1998.
- [3] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, Eds., *Markov Chain Monte Carlo in Practice*, Chapman & Hall, Boca Raton, FL, 1996.
- [4] P. J. Green, "A Primer on Markov Chain Monte Carlo," in *Complex Stochastic Systems*, O. E. Barndorff-Nielsen, D. R. Cox, and C. Klüppelberg, Eds., number 87 in Monographs on Statistics and Applied Probability, chapter 1, pp. 1–62. Chapman & Hall/CRC, 2000.

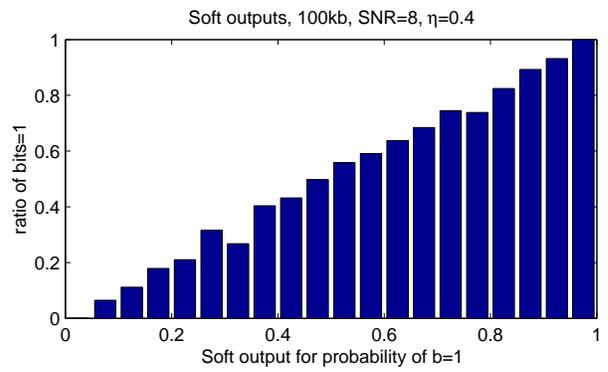


Fig. 4. Soft output performance.

- [5] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *J. Chem. Phys.*, vol. 21, no. 6, pp. 1087–1092, June 1953.
- [6] W. K. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, April 1970.
- [7] X. Wang and R. Chen, "Adaptive Bayesian multiuser detection for synchronous CDMA with Gaussian and impulsive noise," *IEEE Trans. Signal Processing*, vol. 47, no. 7, pp. 2013–2028, July 2000.
- [8] R. Chen, J. S. Liu, and X. Wang, "Convergence analyses and comparisons of Markov chain Monte Carlo algorithms in digital communications," *IEEE Trans. Signal Processing*, vol. 50, no. 2, pp. 255–270, February 2002.
- [9] P. M. Djurić, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Míguez, "Particle filtering," *IEEE Sig. Proc. Magazine*, vol. 20, no. 5, pp. 19–38, September 2003.
- [10] X. Wang, R. Chen, and J. S. Liu, "Monte Carlo Bayesian signal processing for wireless communications," *J. VLSI Sig. Proc.*, vol. 30, no. 1–3, pp. 89–105, Jan–Mar 2002.
- [11] R. L. Tweedie, "Markov chains: Structure and applications," in *Handbook of Statistics*, D. N. Shanbhag and C. R. Rao, Eds., vol. 19, pp. 817–851. Elsevier, Amsterdam, 2000.
- [12] L. Tierney, "Markov chains for exploring posterior distributions," *Ann. Statist.*, vol. 22, no. 4, pp. 1701–1762, December 1994, With discussion and a rejoinder by the author.
- [13] S. Henriksen, B. Ninness, and S. R. Weller, "Multiuser detection via a computational Bayesian approach," Tech. Rep., University of Newcastle, Callaghan, NSW, 2308 Australia, May 2004.
- [14] G. Woodward, R. Ratasuk, M. L. Honig, and P. B. Rapajic, "Minimum mean-squared error multiuser decision-feedback detectors for DS-SS-CDMA," *IEEE Trans. Commun.*, vol. 50, no. 12, pp. 2104–2112, December 2002.